

Persoonlijke aanbevelingen voor video's gebaseerd op gezichtsherkenning

Ine Coppens

Promotoren: prof. dr. ir. Luc Martens, dr. ir. Toon De Pessemier

Begeleider: dr. ir. Toon De Pessemier

Masterproef ingediend tot het behalen van de academische graad van
Master of Science in de industriële wetenschappen: informatica

Vakgroep Informatietechnologie
Voorzitter: prof. dr. ir. Bart Dhoedt
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2017-2018



Woord vooraf

Na een intense en leerrijke periode van vier maanden leg ik nu de laatste hand aan deze masterproef. Het onderwerp van de masterproef was iets dat me heel erg boeide waardoor ik me er helemaal heb in kunnen verdiepen en er mijn eigen persoonlijke ideeën en creativiteit in kwijt kon. Ik heb veel bijgeleerd, zowel over informatica- en wiskundegerelateerde onderwerpen als over de mens en zijn expressies. Hieronder wil ik graag alle personen bedanken die hebben meeg geholpen bij de realisatie van dit eindwerk.

Ik wil ten eerste vooral mijn promotoren prof. Luc Martens en dr. ir. Toon De Pessemier bedanken voor dit onderwerp en het vertrouwen. Ik wil meneer De Pessemier ook bedanken om mij steeds een duwtje in de juiste richting te geven. Ook wil ik mevrouw Pollefliet bedanken voor haar interessante en leerrijke sessies gedurende de volledige opleiding.

Daarnaast wil ik ook mijn ouders bedanken om mij steeds te steunen en helpen waar mogelijk. Ik wil mijn moeder bedanken voor de grammaticale (en morele) ondersteuning en mijn vader voor de Engelstalige en structurele hulp.

Tenslotte ook een woord van dank aan alle vrienden en familie die bereid waren de applicatie te testen door enkele leuke (en ook minder leuke) video's te bekijken. Zonder jullie zou het niet gelukt zijn om zo veel te leren over de menselijke emoties en gedragingen. Ook jullie aanwezigheid en enthousiasme heeft me veel geholpen. Dus aan iedereen, bedankt.

Ine Coppens

Gent - 8 juni 2018

De auteur geeft de toelating deze masterproef voor consultatie beschikbaar te stellen en delen van de masterproef te kopiëren voor persoonlijk gebruik. Elk ander gebruik valt onder de bepalingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit deze masterproef.

The author gives permission to make this master dissertation available for consultation and to copy parts of this master dissertation for personal use. In the case of any other use, the copyright terms have to be respected, in particular with regard to the obligation to state expressly the source when quoting results from this master dissertation.

8 juni 2018

Samenvatting

Deze masterproef beschrijft de ontwikkeling van een aanbevelingssysteem voor video's dat enkel gebaseerd is op gezichtsherkenning en -detectie. Impliciete feedback bestaande uit emotiedetectie, aandachtsmeting (hoofd- en oogpositie) en gedragsanalyse (gesloten ogen, bedekte mond en ogen) wordt verzameld terwijl de gebruikers naar een video kijken. Een Androidapplicatie werd ontwikkeld en volgt drie stappen: de automatische login, het genereren van de aanbeveling en het verzamelen van de impliciete feedback. Er wordt een oplossing gegeven voor het cold start probleem zodat ook gebruikers zonder geschiedenis een aanbeveling kunnen ontvangen gebaseerd op hun geschatte leeftijd en geslacht. Het systeem genereert een groepsaanbeveling met de Average without Misery-strategie en elimineert onaanvaardbare films op basis van minimumleeftijd en reeds bekeken films. De identificatie door gezichtsherkenning staat centraal zodat de app op elk moment weet aan welke gebruiker de feedback toebehoort. Om de identificatie goed te laten verlopen, leert het systeem elke gebruiker goed kennen door bij elke aanmelding het gezicht extra te trainen en leeftijd en geslacht opnieuw in te schatten. De app gebruikt meerdere gezichtsdetectieservices: Microsoft Face API, Face++ en Kairos. Hun output wordt, op basis van een evaluatie, samengenomen om de waarderingsscore van elke individuele kijker te schatten. De term emotiefingerprint wordt geïntroduceerd en definieert de verwachte emotieprobabiliteiten in de loop van de tijd. Het is de grote uitdaging van de masterproef om de emotiefingerprints, uniek aan elke video, zodanig te bepalen dat de waardering op basis van emoties goed kan worden ingeschat. De app werd getest op twintig testpersonen en zes video's. Op basis van hun getoonde emoties worden vier verschillende sets van verwachte emotiefingerprints uitgewerkt en uitgetest om verschillende oplossingsmethoden te onderzoeken. Set 4 geeft de beste resultaten en kan vooral films met de emotie blijheid goed onderscheiden. De kwaliteit van set 4 is te danken aan de focus op enkel de belangrijkste scènes en emoties. Verdriet, boosheid, afkeer, angst en verrassing werden doorheen alle testen slechts met een maximale waarschijnlijkheid van 0,68 gedetecteerd en vaak lagen hun waarden dicht bij nul. Bij blijheid was dit bij 90 % van de testpersonen groter dan 0,95. Dit toont aan dat blijheid en video's met deze emotie het makkelijkst te uiten, te detecteren en in te schatten zijn. De andere emoties, met kleinere waarden aanwezig, zijn bijgevolg moeilijker te linken aan de verwachte emoties en hierbij kunnen metingen van de factoren aandacht en gedrag bijdragen aan de waarderingsscore.

Abstract

This master's thesis describes the development of a recommendation system for videos that is only based on face recognition and detection. Implicit feedback consisting of emotion detection, attention measurement (head and eye pose) and behaviour analysis (occlusion of the eyes and mouth, closure of the eyes) is collected while users are watching a video. An Android application was developed and follows three steps: the automated login, the generating of the recommendation and the collecting of the implicit feedback. A solution is proposed for the cold start problem so users without history can also receive a recommendation based on their estimated age and gender. The system generates the group recommendation using the Average without Misery strategy and eliminates unacceptable movies based on minimum age and already watched movies. The app revolves around the identification with face recognition so it knows exactly to which user the implicit feedback belongs. To make sure the identification works well, the system trains the users' faces at each login and estimates their age and gender again. The app uses multiple face detection services: Microsoft Face API, Face++ and Kairos. Their output is aggregated, based on their evaluation, to estimate the appreciation score of every individual user. The term emotion fingerprint is introduced to define the expected emotion probabilities in function of the time. The big challenge in this master's thesis is to determine the emotion fingerprints, which are unique to each video, in a way that the appreciation based on emotions can be estimated correctly. The app was tested on twenty persons who watched six videos. Based on their expressed emotions, four different collections of expected emotion fingerprints were developed and tested. The fourth set gives the best results and can distinguish videos with happiness very well. The effectiveness of set 4 is explained by its focus on only the most important scenes and emotions. Sadness, anger, disgust, fear and surprise were detected throughout all the tests with a maximum probability of only 0,68 and often, their values were close to zero. For happiness, the maximum value with 90 % of the test persons was larger than 0,95. This proves that happiness and videos with happiness are the easiest to express, detect and estimate. The other emotions, which are present with lower values, are consequently harder to link to the expected emotions and their appreciation score can be assisted by the attention and behaviour measurement.

Personal Suggestions for Video Content Based on Face Recognition

Ine Coppens

Supervisor(s): prof. Luc Martens, dr. ir. Toon De Pessemier

Abstract—In this article, a new method is discussed for a personal video recommendation system based on implicit feedback. With facial recognition it can be determined who is in front of the screen at each point in time to link their implicit feedback to their personal profiles to make a personal recommendation in the future and gain an automated login. The implicit feedback is collected during viewings of the users and consists of emotion detection, attention measurement and behavior analysis. An Android application was created which follows three steps: an automated login to decide who is present, the creating of the video recommendation (to present to the users before choosing a video) and the collecting of the implicit feedback by analyzing the viewers while the chosen video is displayed. The app uses different services for different aspects of the face detection and combines their results into an estimation of the experience of each individual viewer. A solution for the cold start problem, which occurs when the system can't give a recommendation to new users without history, is proposed based on the estimated age and gender of the new users. The term *emotion fingerprint* is introduced and is defined as a unique spectrum of expected emotions for a video which is used to estimate the experience score of the user. The app was tested with a test panel to do research for making an optimal expected emotions fingerprint. The fourth set gives the best results and is best at distinguishing videos with happiness. The effectiveness of set 4 is explained by its focus on only the most important scenes and emotions.

Keywords—Face detection, emotion detection, attention measurement, face recognition, recommender system, Android

I. INTRODUCTION

A lot of streaming services generate a personal recommendation for their users to give a focused suggestion in the abundance of all the possible videos. In the context of watching a video, often more than one person are choosing the video content but the recommendation is not adjusted to everyone. A personal recommendation based on all the people present would be more accurate. It is user unfriendly to manually log in every single person and give a personal explicit feedback for every viewer to the recommender system. A better approach would use facial recognition to automatically log in every viewer and fetch their personal preferences to make a group recommendation. Their preferences could be derived from their implicit feedback, based on face detection, given during previous viewings.

This master's thesis examines which implicit feedback can be measured to calculate the most optimal estimation of the content experience for each viewer. An Android application is developed to display a video, take pictures of all the people in front of the screen and send requests to different services at the same time. The app applies face recognition on every photo to firstly identify each face, followed by emotion detection, attention measurement and behavior analysis. The similarity between the expected emotions and the expressed emotions is calculated to determine the content experience of the user. The term *emotion fingerprint* is introduced and is defined as a unique spectrum of expected emotions for a video. The emotion fingerprint specifies for each moment in the video the probability value of each

of the six detected emotions in the app: anger, disgust, fear, happiness, sadness and surprise. An additional research consists of creating the best expected emotion fingerprints for each video.

The two main components are described in II. Three services - Microsoft Face API, Face++ and Kairos - will be tested to determine which service provides the best results for their face detection. The results are shown in III. With this information, an Android app can be developed which will automatically log users in (step 1), create a recommendation for every user present (step 2) and combine the output of the services and link it to the personal profile of every user while they are watching the video (step 3). The architecture of the steps and components are described in IV. Finally, the app will be tested with a test panel of twenty people and the results are described in V, followed by a conclusion in VI.

II. TWO MAIN COMPONENTS

To achieve its goal, the app consists of two main components, a recommender system and a face detection system, as shown in Figure 1. Step 1 (login) and step 3 (collecting of the implicit feedback) both use face detection and recognition. Step 2 uses a recommendation system which gives a suggestion for multiple videos after the users are logged in. After the creating of the recommendation, the users can choose a video from the proposed videos. Once the video is started, step 3 can start to collect the implicit feedback. When the video is stopped, a re-evaluation of the persons present can be made for a possible new recommendation, based on the current logged-in users, and the process can start over.

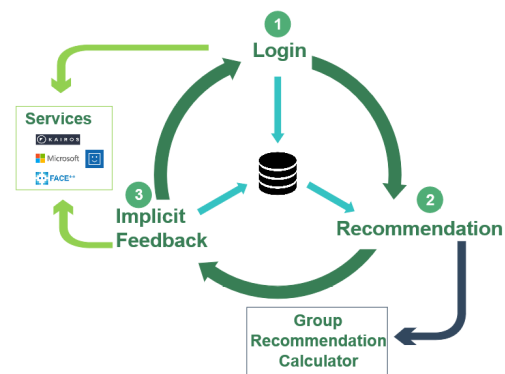


Fig. 1. The three steps use two main components: a recommendation system and face detection services

A. Recommendation system

The first main component is a recommender system who has to take into account the preferences of each logged-in user. The

recommending process can be represented as an estimated score for each person to each video, based on what the system thinks the user likes or dislikes. Judith Masthoff proposes eleven aggregation strategies to aggregate every user's personal preference into a recommendation for the whole group [1]. In this application, the Average without Misery strategy is used because it takes into account every user's estimated score and misery will be avoided by eliminating videos who are unacceptable for some users [1]. This way, when one person has a low preference for a video, this video will not be recommended.

Other criteria can be added to the recommendation whereof one is to determine whether a person is too young to see a video. Every video has a minimum age, obtained from the Common Sense website [2]. When at least one user's age is below this minimum, the video will be marked as unacceptable. Also when one user has already seen a video, this video will also be unacceptable for the group because the app knows that this person probably doesn't want to see that video again.

When a new user is present and no personal preferences are known, the system will have to deal with the cold start problem. No initial recommendation can be made because there is a shortage of information about the preferences of new users [3]. The goal in this application is to make it as user friendly as possible so the app will not ask the new users about their preferences. The app can cope with the cold start problem by making a recommendation based on the user's estimated age and gender (determined by face detection services). With the demographic information, the system can recommend videos that other people of the same demographic group rated with a high score relative to the global rating of the video, using demographic ratings from IMDb [4]. The more history a user collects, the more his recommendation will be based on history and the less it will be based on demographics.

B. Face detection and face recognition

Face detection is the locating of the faces in an image and is actually the starting point of all facial analysis and different approaches to execute the detection exist [5]. Face recognition is matching a face to a person saved by the system. In the paper of Yang et al. it is also called face authentication and is described as an identification of the individual in the picture while facial expression recognition is described as the identification of the emotion [6]. The face recognition software has a collection of possible faces in a database linked to a person and the face it has to identify is compared to each existing face to see if they can belong to the same person [6]. The candidate whose face is the most likely to belong to the input face, is returned.

III. EVALUATION OF THE SERVICES

Existing services for face detection and face recognition will be used in the application. Three emotion detection services are compared to determine which would be best to use in the app. The services will be used in two steps of the application: the login step and the implicit feedback step. The services are: Microsoft Face API [7], Face++ [8] and Kairos [9].

A. Login

The age and gender has to be determined in order to give a demographic recommendation. An age test was performed with the three services on a created dataset that contains pictures of 46 people and their exact known ages. The results of the test are visible in Figure 2 and show the differences between the estimated ages and real ages. The closer the difference is to zero, the better the estimated age. Microsoft has the best results in the age test so only this service will be used to determine the age of the users. The output of the gender estimation of the three services performs equally well and will be aggregated to determine the gender of the users with a voting process. This way, a more reliable gender is estimated.

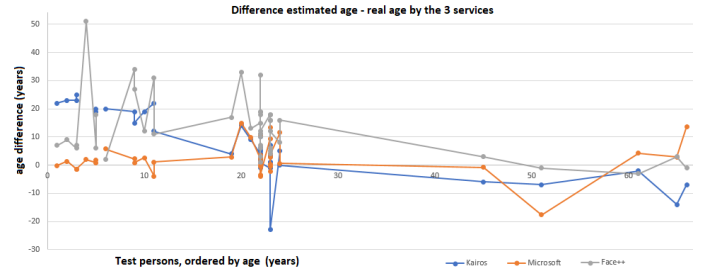


Fig. 2. The results from the three services who estimated the age of people based on their faces in a dataset

The login also requires face recognition and this will also be executed by Microsoft. Microsoft provides a Face Storage that saves persons with their faces in a Person Group which has to be trained [7]. The influence of the three services during the login step can be seen in Figure 3.

B. Implicit feedback

Emotion detection is the main component of the estimation of the implicit feedback. The app will detect six base emotions: anger, disgust, fear, happiness, sadness and surprise. An emotion test was executed on the three services using the Cohn Kanade dataset which contains photos of people showing different emotions evolving from neutral to a very explicit emotion [10], [11]. In the emotion tests, Microsoft has proven to perform best so again this service will be chosen for the emotion detection together with the identification of every person in the picture. The emotion test also clarified that - even with the most explicit emotion photos - anger, disgust and fear are always detected with a low probability value. Happiness can be detected with high probability values and therefore it can be concluded that happiness is the easiest emotion to detect and anger, disgust and fear are the hardest.

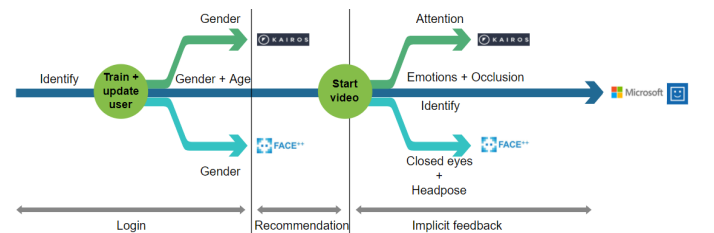


Fig. 3. The three steps use different services

Microsoft has an additional detection for the occluded areas of the face and this will be used to determine the negative feedback when users respond to the video content by holding their hands in front of their face. Face++ is the only service who can detect closed eyes and Kairos will be chosen for the attention measurement. Another factor for the attention detection is determined by the head pose of the viewers. Face++ has proven to give the best results in the head pose estimation. Figure 3 shows the co-operation of the three services to collect the implicit feedback in step 3.

IV. ARCHITECTURE OF THE APPLICATION

The basic architecture of the application can be represented as in Figure 1. The process consists of three steps, which are explained in detail below.

A. Step 1: automated login

In order to log everyone in, the users have to show their face to the camera and push the start button to make a photo and apply face recognition. Users already known to the app will be logged-in and new users will be added to the database to make new user profiles. Their faces will be sent to the service to (re-) evaluate their age and gender and this way the app can update the user profiles. Every time a user logs in, the existing age will count for 90 % of the updated age and the new estimated age will count for 10 %. When somebody ages or his age was estimated with a large error, the system can compensate this with the new age. This is especially interesting for children because when they get older, more videos will become available due to their allowed minimum age. Every new login time, the face is also sent to the Face Storage of Microsoft to train the person again. This way, the service is more likely to recognize the person in the future and can cope with the aging of the users.

B. Step 2: generating a group recommendation

With the help of step 1, the system now knows exactly which users are present. To generate the recommendation, the system fetches information from the local database about the history and demographics of the user.

The calculated preferences are a mix of demographic recommendation scores and history based recommendation scores and can be demonstrated with Figure 4. Users that are already known to the system and who have history, like users 1 and 8, will add their history from the local database to the separate history based recommendation system. This history recommendation takes into account the previous watched videos and the estimated appreciation score from each user to these videos. This awarded score is based on the implicit feedback collected in step 3. The system will recommend similar videos based on this score. Similar videos are determined by the tags that are defined as a connecting thread through all the videos present. An example of such a tag is the genre comedy or the keyword ghosts. By allocating different tags to the movies in the database, some movies show overlapping properties with each other. The tags have to be tuned to the provided video dataset because when there are too much tags, all videos will somehow be interpreted as similar and the connections are too wide. For each user and

each watched video, all the tags are iterated and for each similar video, the average of the awarded scores belonging to the tags is calculated. This results in a collection of videos with the average score on a total of ten, determined for all the users.

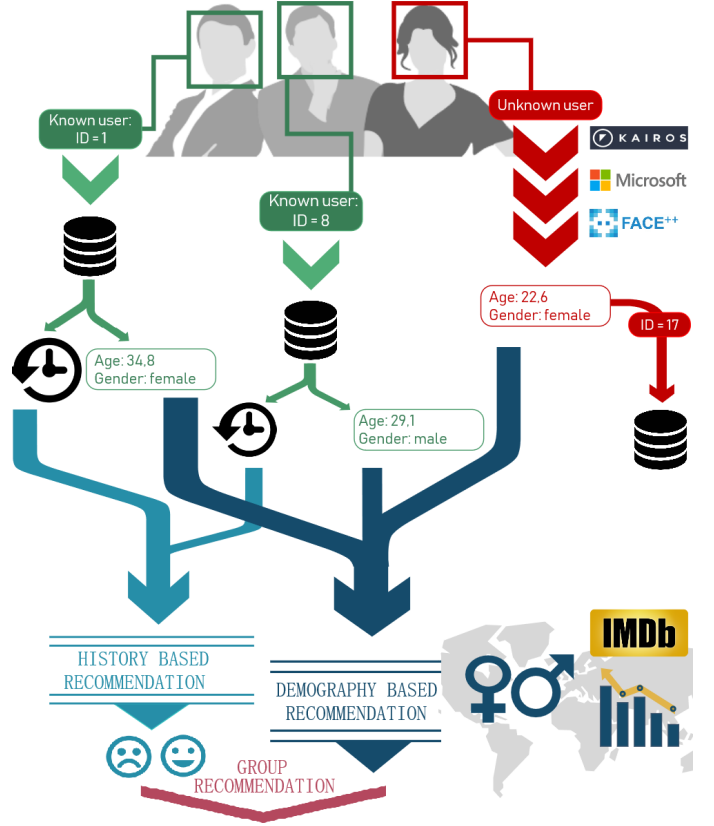


Fig. 4. The flow of step 2 demonstrated with an example

User 17 is new to the system and at this moment, only its calculated age and gender are known. The age and gender of all the logged-in users will be added to the demography based recommendation system which uses demographic ratings as explained in II-A and generates a score on ten point for every possible video.

The output of the two recommendation systems is a score on a maximum of ten points and represents the estimated interest of each user in each video. The scores are combined for each video to create a group recommendation with the Average without Misery strategy with the additional criteria (the minimum age and the already seen videos) to eliminate the unacceptable videos.

C. Step 3: collecting the implicit feedback

Next, the recommendation is provided to the users and they can choose a video from the list. When the video starts, the app takes a picture every two seconds and step 3 can begin. Every picture is immediately sent to the Microsoft service for face detection and identification of every face present. Every individual detected face is then sent to the other detection services for additional information. During the video, all the implicit feedback is collected for each photo and saved into the local database. Because face recognition is always the basis to identify the person,

the system can save the information in the appropriate history of this user.

For each user and each watched video, an estimated feedback score on ten points is estimated. The similarity with the expected emotions counts for six of the ten points. The attention level counts for four of the ten points. Negative feedback determined from the behavior of the user can subtract points of this total score. For example when a user didn't fully watch the video, the application can register this resulting in a lower score. When a person was holding his hand in front of his mouth or eyes a lot or closing his eyes, the application also registers this as a negative feedback because this means the content was probably too scary for the person. This way, a score on a maximum of ten points can be composed based on the emotions, attention level and behavior.

To determine the similarity with the expected emotions, two types of score systems are used. The first is the distance score wherefore the Formula 1 is used which is based on the Euclidean distance between two functions for every point in time [12]. For every emotion value of the expressed emotions, the equivalent expected emotion value of the same time is fetched. The sum function represents the sum of the quadratic differences between the expressed emotion value and the expected emotion value. This sum is divided by the number of time points that are available in the expressed emotion spectrum. In Formula 1, the square root of this value is calculated to get the Euclidean distance.

$$distance = \sqrt{\frac{1}{n} \sum_{i=0}^n (x_i - y_i)^2} \quad (1)$$

The similarity grows as the Euclidean distance becomes smaller. For every emotion distance value, the similarity value is calculated with Formula 2 which results in a value between zero and one. When the similarity is equal to one, this means the two functions are exactly alike and 100 % similar and the distance is equal to zero.

$$similarity = \frac{1}{1 + distance} \quad (2)$$

The other score system is based on margins. Every time an expressed emotion value falls in the margins of the expected emotions, that emotion receives a point. At the end, the points of every emotion are divided by the total possible earned points which also results in a number from zero to one for each separate emotion.

A total emotion score of the six emotions together can be obtained by calculating the sum of all the emotions similarities. When a video contains an emotion that has to have more weight, the similarity of this emotion can for example be multiplied by 2 and the other emotion similarities by 0,8 which also results in a total of 6. An other possibility is to only take into account the expected strongest emotion of that video. For example when there are two prominent emotions, they each receive a weight of 3 resulting in a total emotion score of 6.

The attention level score results in a score on a maximum of four points. For every picture, it is determined whether the person has attention. The total number of pictures with attention

is divided by the total amount of pictures, which results in a number between zero and one, and is then multiplied by four to receive a number from zero to four. The factors that contribute to state that the person is paying attention are: when his both eyes are open, when the attention detection of Kairos has an attention probability of 0,5 or more and when the head pose angles are between the margins (30 degrees for the yaw angle, 20 degrees for the pitch angle).

The behavioral score can subtract points from the total emotion and attention score. The percentage of photos where the person has closed both eyes or occluded his mouth or eyes, is determined. When a person didn't fully watch the movie, the percentage of the amount of photos where he wasn't present is determined to reduce the appreciation score.

V. RESULTS OF THE TESTING OF THE APP

The test panel consisted of twenty people between the ages of 5 and 72. Six videos are chosen to play during the tests and every video contains one of the six base emotions as its strongest emotion. The two children of 5 years old didn't have to watch the movies with a minimum age older than their age. For every person and every video they watch, an emotion, an attention and a shared score are calculated (the shared score is the sum of the attention score and the behavioral score). The test persons are asked to not stop or pause the video but they can look away or do what they want when the content is too scary. Every two seconds during the video, a picture is made and the detection services are called on this picture. Every face detection answer is saved in the local database on the device and after the six videos, all the database info is printed into an CSV file. This way, when new methods are implemented in the app, the old data can be re-read into the database.

During and after the testing with the twenty people, different expected emotion fingerprint sets for all the videos will be developed and re-implemented in the app to recalculate the results. These expected emotions define the emotion score because this score represents how well the expressed emotions of each user match the expected emotions. An expected emotion fingerprint for a specific video performs well when it can determine, given only the expressed emotions of a user, which movie that person was watching. To give a measure of performance, every expected emotion set is being tested with the so called switch test. For this test, six video history's CSV files from six different test users are chosen, one for each of the six videos. Only users who gave a high explicit score to the videos are chosen because this gives a good indication of what an expected emotion spectrum would look like. Every history collection of one of the videos will be disguised as one of the six videos. The calculation of the emotion score will now compare the probabilities to each of the expected emotions it thinks it has. A good set would result in the highest emotion score for the right video.

Figure 5 shows the expressed emotions from person 13, 3, 8 and 4 while watching video 1. They all show different spectra and values for the emotion happiness. Person 3 has explicitly said that she didn't like video 1 and this can also be seen in her spectrum because she didn't show that much happiness.

Figure 6 shows the spectra of two persons while watching video 5 where the emotion fear is expected as the strongest emo-

tion. It shows the difference in expressed emotions between someone who is extravagant in his emotions and someone who is not. The figure also shows that even with an extravagant personality, the probability of the detected emotions anger, disgust, fear, sadness and surprise isn't higher than 0,6. This can conclude that the appreciation of the user by the emotions of a video with happiness are easier to determine.

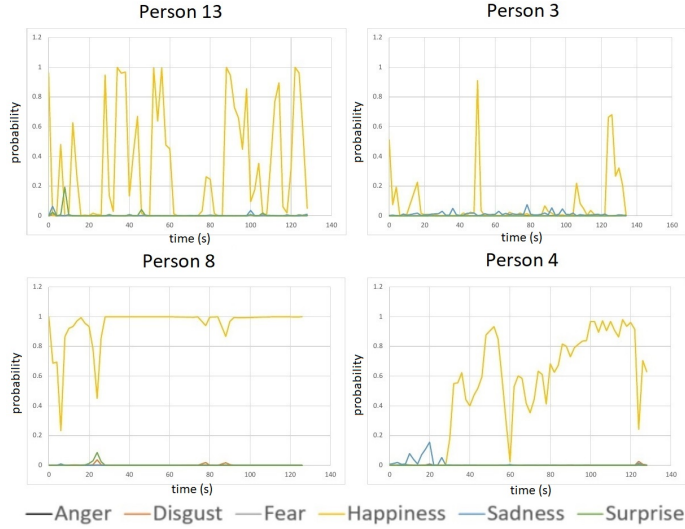


Fig. 5. The expressed emotions by four test persons watching video 1

A. Results from set 1

Set 1 was made from showing acted emotions while watching the video multiple times, recording them, merging them and then smoothing the results. This set was created before the testing was done and had the purpose as a starting set with expected emotions at the right time and to look what the results would be. This set doesn't perform that good when only looking at the emotion score of the strongest emotion of the video. When the expressed emotions of video 1 are compared to the expected videos of video 1, it scores less than the comparison of the expected emotions of video 2 and 5. Because video 1 is all about happiness emotions, the similarity value of the strongest emotion should have been the largest. When the six emotions are merged into the emotion score, the set guesses the right video two out of six times so this is slightly better. It can be concluded that set 1 still can be improved.

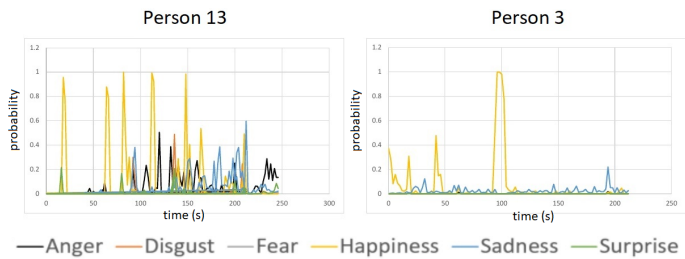


Fig. 6. The expressed emotions by two test persons watching video 5

B. Results from set 2

Set 2 was made after the tests were taken. It was made from the averages of the shown emotions from some of the test per-

sons. This set should have more realistic values because it is based on the emotions of real people. Although it guesses four out of six video in the switch test with the combined emotion score, set 2 still doesn't perform well. No single emotion score guesses video 1, which should be the easiest to guess and the differences between the emotion scores is so little that almost no distinction can be made between the highest and lowest scores. The diversity between the same videos for different persons is also very low which means set 2 also needs some improvement.

C. Results from set 3

The third set is the same as set 2 but with some adjustments. Parts of the video that are not interesting are omitted by giving the emotion probabilities a value of zero. This way, the key scenes that are important for emotion detection are emphasized and only emotions during those periods will be calculated in the emotion score. Also, this set was smoothed to eliminate peaks and declines that set 2 had. The set resembles set 2 a lot so there is also a low distinction between the scores in the switch test. Another set with better values has to be created.

D. Results from set 4

The final set, set 4, was made with the results learned from the previous sets. Key scenes are even more defined which has to result in a better definition of the important parts. Also, videos can have more than one expected emotion. The switch test has better results for this set resulting in 3/6 well guessed videos when looked at the sum of only the strongest emotion similarities. This includes video 1, with a large difference to distinguish itself from the other videos. The diversity between videos for different persons is also better. This is what was aimed for, so currently set 4 is the best set. But there is still room for improvement because video 3 gets the highest emotion score of all videos for 83,33 % of the persons (75,00 % when the two children are included) and highest total score for 77,78 % of the persons, resulting in a similar recommendation throughout almost the whole test panel.

E. General conclusions of all the sets

Figure 7 shows the four different sets for the expected emotions of video 1 and 2. Set 3 shows the smoothening of set 2 by taking the maximum value in every range of ten seconds. Set 4 demonstrates the more defined interesting moments of every video.



Fig. 7. The expected emotions from the different sets for video 1 at the top and video 2 at the bottom

During the testing it became clear that people don't express their emotions a lot, even during the intense six videos. Only

happiness is expressed with a maximum probability value of 1,00 throughout all the test persons and videos (90 % reached a maximum of 0,95 for happiness) so videos with happiness easily achieve high probabilities. When the other emotions are expressed, the detection service often registers only a small probability, making videos with these emotions harder to estimate. The maximum value of sadness was 0,68 with only 15 % of the test persons scoring a maximum of 0,60 or higher and for fear the maximum of all test persons was only 0,27.

The real challenge is to define the right expected emotions and their probability values. The following principles and guidelines can be determined:

1. **Take into account all the emotions of a video and ignore emotions that don't matter.** Focus on emotions that have to be present and define emotions that have to be absent so that deviations and distances with these emotions can be calculated. Ignore emotions that don't matter so they don't influence the score too much.

2. **Take into account the emotions that are difficult to express and detect.** Adjust the expected emotion probabilities for emotions that are harder to detect. When the expected probability values are raised too much, it is almost impossible for the viewers to reach those values and their estimated emotion scores will be low. This doesn't apply for the emotion happiness because this probability is easy to reach. When the expected values are reduced, it is too easy to reach a high similarity and emotion score. With an expected value that is too low, every video's score can be mistaken for another video in the switch test. A good middle value has to be determined.

3. **Define only the most important scenes of the video.** Focus on the scenes where the emotions are mostly expected and ignore parts where no emotions will be expressed.

The attention level and behavior analysis have proven to be determined correctly by the face detection services but it can be considered to give their results more weight when the emotion score doesn't suffice.

VI. CONCLUSION

The app is based on estimating an appreciation score for each user who has watched a video based on detected emotions, attention level and behavior. With this estimated score, other movies that are alike (by being connected by tags, the connecting threads for different movies) can be determined as videos that this person might like. A group recommendation takes into account these estimated preferences of the everyone present during the login part. The cold start problem is solved by estimating the age and gender and calculating the most optimal recommendation by using demographic ratings of IMDb.

Defining the appreciation of the video solely by analyzing the emotions of users, is the challenge of this master's thesis. The emotion test on the services showed that anger, disgust and fear are the hardest to detect with a high probability. However, the tests with the test panel showed that every emotion except happiness never has higher probabilities than 0,68. For happiness, the maximum value throughout all the tests was with 90 % of the test persons larger than 0,95. This proves that happiness and videos with happiness are the easiest to express, detect and estimate. The other emotions, which are present with lower val-

ues, are consequently harder to link to the expected emotions and their appreciation score can be assisted by the attention and behavior measurement.

For videos where happiness is the strongest emotion, expected emotions set 4 has proven to be effective. The effectiveness of set 4 can be explained by its fine definition of key scenes and balanced probabilities (not too high so it's impossible to reach and not too low so every expressed emotion can reach it). Set 4 also takes into account only the strongest emotions of the videos resulting in a score that's only based on key scenes and key emotions. Set 4 is still not ideal because the video with anger gets the highest emotion score throughout all the videos for 83,33 % of the test persons resulting in a similar recommendation. However, set 4 shows that three principles and guidelines can be determined for an emotion fingerprint: 1) take into account all the emotions and ignore emotions that don't matter, 2) take into account the emotions that are difficult to express and detect, 3) define only the most important scenes of the video.

Finding a good balance of the probabilities in the emotion fingerprint of a video is key to a good recommender system. This is because everything in the recommendation depends on the assigned implicit score and the emotion score plays a big role in this score. Other factors like attention level and behavior analysis can also contribute to the implicit score to gain a more precise recommendation. Additional research with more test persons can be done to even further improve the effectiveness of set 4, based on the determined principles and guidelines of this master's thesis.

REFERENCES

- [1] Judith Masthoff, "Group recommender systems: Combining individual models," in *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, Eds., pp. 677–702. Springer US, Boston, MA, 2011.
- [2] "Common sense media," <https://www.common-sense-media.org/about-us/our-mission>, 2018, Accessed: March 17, 2018.
- [3] Blerina Lika, Kostas Kolomvatsos, and Stathes Hadjiefthymiades, "Facing the cold start problem in recommender systems," in *Expert Systems with Applications: An International Journal*, Tarrytown, NY, USA, March 2014, vol. 41, pp. 2065–2073, Pergamon Press, Inc.
- [4] "Internet movie database (imdb)," <http://www.imdb.com/>, 2018, Accessed: March 29, 2018.
- [5] Mayank Singh Chauhan and Mukesh Sakle, "Study and analysis of different face detection techniques," *International Journal of Computer Science and Information Technologies*, vol. 5, 2014.
- [6] Ming-Hsuan Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, Jan 2002.
- [7] "Microsoft cognitive services," <https://azure.microsoft.com/en-us/services/cognitive-services/>, 2018, Accessed: March 29, 2018.
- [8] "Face++ cognitive services," <https://www.faceplusplus.com>, 2018, Accessed: March 29, 2018.
- [9] "Kairos," <https://www.kairos.com>, 2018, Accessed: March 29, 2018.
- [10] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00)*, Grenoble, France, pp. 46–53, 2000.
- [11] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete expression dataset for action unit and emotion-specified expression," *Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010)*, San Francisco, USA, pp. 94–101, 2010.
- [12] Toby Segaran, *Programming collective intelligence: building smart web 2.0 applications*, pp. 9–28, O'Reilly Media, December 2008.

Inhoudsopgave

Woord vooraf

Samenvatting

Abstract

1	Inleiding	1
1.1	Huidige situatie: een overvloed aan films en series	1
1.1.1	Expliciete feedback geven, niet iedereen doet het	1
1.2	Doelstelling van de masterproef	2
1.2.1	Onderzoeksvraag en deelonderzoeksvragen	2
1.2.2	Methode waarmee de applicatie werkt	2
1.3	Hoe er te werk wordt gegaan en hoe deze scriptie is ingedeeld	3
2	Componenten en bestaand werk	4
2.1	Aanbevelingssystemen	5
2.1.1	Soorten systemen	5
2.1.2	Ratings	6
2.1.3	Cold start probleem	7
2.1.4	Groepsaanbevelingen	8
2.2	Gezichtsdetectie	9
2.2.1	Gezichtsdetectie versus gezichtsherkenning	10
2.2.2	Werking	11
2.2.3	Emoties	13
2.3	Bestaand werk	14
2.3.1	Aanbevelingen op basis van emoties	14
2.3.2	Aanbevelingen op basis van aandachtsniveau	14
2.3.3	Aanbevelingssystemen	15
2.3.4	Gelijkaardige applicaties in de actualiteit	16
3	Evaluatie services	17
3.1	Gezichtsdetectieservices: een eerste vergelijking	17
3.1.1	Microsoft Cognitive Services	17
3.1.2	Face++	19
3.1.3	Kairos	19
3.1.4	Overzicht	19
3.1.5	Hoe de services gebruikt worden	20
3.2	Testen voor leeftijd en geslacht	21
3.3	Emotietesten	23
3.4	Testen voor bedekte gezichtsdelen, gesloten ogen, hoofdpositie en aandachtsniveau	26
3.5	Besluit voor het gebruik van de services	27
4	Applicatie	29
4.1	Androidapplicatie en externe benodigdheden	29
4.1.1	Android Studio	29
4.1.2	Room Database	30
4.1.3	YouTube API	30

4.1.4	Invulling databank	30
4.1.5	Gebruikte smartphone en tablet	31
4.2	Stap 1: Automatische aanmelding	31
4.3	Stap 2: Genereren van de groepsaanbeveling	33
4.3.1	Groepsaanbeveling op basis van demografische gegevens	34
4.3.2	Groepsaanbeveling op basis van impliciete feedback uit de persoonlijke geschiedenis	36
4.3.3	Aggregeren van de twee afzonderlijke groepsaanbevelingen	38
4.4	Stap 3: Impliciete feedback verzamelen	38
4.4.1	Berekenen van de overeenkomst met de verwachte emoties	41
4.4.2	Geschatte score van de film op basis van de impliciete feedback	44
4.4.3	Uitzonderingen en speciale gevallen	46
4.5	Updaten en trainen van gebruikers	46
4.6	Architectuur	47
4.6.1	Activiteitendiagram	48
4.6.2	Databankmodel voor de Roomdatabank	48
4.6.3	UML- of klassendiagram	49
4.6.4	Sequentiediagrammen	52
4.7	Testplan	54
5	Resultaten: uittesten van de app met testpersonen	55
5.1	Het uitvoeren van de testen	55
5.1.1	Voorafgaand aan de testen	55
5.1.2	Het testproces	56
5.2	Het gebruik van een eerste set: set 1	58
5.2.1	Resultaten met het gebruik van set 1 en moeilijkheden van de testen	58
5.2.2	Kwaliteit van set 1 met de switchtest	63
5.3	Nieuwe emotieset op basis van de emoties van de testpersonen: set 2	64
5.3.1	Resultaten met het gebruik van set 2	64
5.3.2	Kwaliteit van set 2 met de switchtest	65
5.4	Verbeteren van de verwachte emoties met set 3	66
5.4.1	Resultaten met het gebruik van set 2	66
5.4.2	Kwaliteit van set 3 met de switchtest	67
5.5	Optimaliseren resultaten met set 4	67
5.5.1	Resultaten met het gebruik van set 4	68
5.5.2	Set 4.2	69
5.5.3	Kwaliteit van set 4 met de switchtest	70
5.6	Resultaten bevraging testpersonen en bedenkingen	71
5.7	Besluiten	72
5.7.1	Emotiedetectie	72
5.7.2	Aandachtsniveau	74
5.7.3	Gedragsanalyse	74
5.7.4	Combinatie van factoren	75
5.7.5	Algemeen besluit en uiteindelijke instellingen	75
6	Bedenkingen en mogelijke uitbreidingen	76
6.1	Privacyaspect	76
6.1.1	Het vertrouwen van de kijkers	76
6.1.2	Authenticatie	76
6.2	Mogelijke uitbreidingen	77
6.2.1	Extra mogelijkheden	77
6.2.2	Mogelijke problemen	78
7	Conclusie	80
	Referentielijst	83

Bijlage A: Installatiehandleiding	87
1 Installatie met APK	87
2 Installatie via Android Studio	89
3 Vereiste keys en vereiste libraries	90
4 Bestanden in het project	91
 Bijlage B: Gebruikershandleiding	 92
1 Opstartscherm	92
2 Inlogscherm	93
3 Scherm waarin de gebruiker een video kan kiezen	94
4 Instellingenscherm	96
4.1 Persistente data bekijken	96
4.2 Niet-persistente data bekijken	97
4.3 Data in de databank zetten en data uit de app verwijderen	97
4.4 Uitprinten van data uit de databank	97
4.5 Evaluatietesten services	97
4.6 Herinlezen van geschiedenisrecords	98
4.7 De gemiddelde waarden van verschillende emotiespectra laten bepalen	99
5 De verschillende modi om de app te laten draaien	99
6 Standaardinstellingen	100
7 Uitvoeren unittesten	101
8 Mogelijke problemen	101
 Bijlage C: Grafieken, diagrammen, code en tabellen	 102

Lijst van figuren

2.1	De drie stappen gebruiken twee hoofdcomponenten: een aanbevelingssysteem en gezichts- detectieservices	4
2.2	Een illustratie van de service Face++ met 106 landmarks aangeduid op een gezicht [1] . .	10
2.3	Haar-like features van een menselijk gezicht gedemonstreerd in de paper van B. Ali et al. [2]	12
2.4	Het Viola-Jones algoritme in werking volgens een video gemaakt door Adam Harvey [3, 4]	13
3.1	Het json antwoord van Microsoft Face API als voorbeeld gegeven op hun site [5]	18
3.2	Vergelijking van de leeftijdsinschatting van de drie services toegepast op de niet-ideale dataset gesorteerd op leeftijd	22
3.3	Vergelijking van de services per leeftijdsgroep	23
3.4	Vergelijking tussen de drie services en de emotiedetectie van blijheid op de laatste (a) en middelste (b) foto uit elke reeks	24
3.5	Vergelijking tussen de drie services voor de emotiedetectie van afkeer, boosheid en angst, toegepast op de laatste foto uit de reeks	25
3.6	Vergelijking tussen de drie services voor de emotiedetectie van verrassing en verdriet, toe- gepast op de middelste foto uit de reeks	25
3.7	De flow van de verschillende services doorheen het gebruik van de applicatie	28
4.1	Stap 1: Automatisch inloggen van de aanwezige gebruikers met gezichtsherkenning	32
4.2	De lijst van aanbevolen films in de app	33
4.3	Stap 2: De groepsaanbeveling wordt gegenereerd op basis van de verkregen informatie . .	34
4.4	Stap 3: Het verzamelen van impliciete feedback met gezichtsdetectie	39
4.5	De emoties van video 2, een fragment uit de film ‘Up’: a) zoals ze gedetecteerd werden en b) zoals ze aangepast werden	42
4.6	Het vergelijken van de verwachte emoties met de getoonde emoties voor elk gedetecteerd tijdstip, gedemonstreerd op zowel de distance- als de margins-methode	43
4.7	De punten die te verdienen zijn op basis van de impliciete feedback samen met de punten die te verliezen zijn	45
4.8	Databankmodel van de applicatie	49
4.9	Deel van het UML-diagram waar de services worden opgeroepen	50
4.10	Deel van het UML-diagram waar de databankmethoden en Globals behandeld worden . .	51
5.1	De getoonde emoties van testpersoon 6 bij videofragment 1	61
5.2	De getoonde emoties van testpersoon 1 bij videofragment 1	65
A.1	Het logo van de app op het toestel	87
A.2	Het scherm ‘My Files’ geeft toegang tot de installeerbare bestanden die aanwezig zijn op toestel	88
A.3	Het Androidtoestel begeleidt de gebruiker meteen naar de correcte instellingen waar hij de beveiliging tijdelijk kan uitschakelen	88
A.4	De gebruiker krijgt nog een laatste waarschuwing alvorens een installeerbaar bestand, dat niet van de officiële Google Play Store afkomstig is, te installeren	89
A.5	Het scherm dat aan de gebruiker vraagt of de applicatie op het toestel mag geïnstalleerd worden, met rechts onderaan de bevestigingsknop	89
B.1	Het scherm dat te zien is bij het allereerste gebruik van de app	92
B.2	Bij het eerste gebruik moet de gebruiker toestemming aan de app geven om de camera en de opslagruimte te kunnen gebruiken	93
B.3	Het inlogscherm met de verschillende knoppen om de app te bedienen	93

B.4	Het stuk van het inlogscherf dat een laadicoon toont wanneer de gebruiker de knop van het logo heeft ingedrukt	94
B.5	Het filmovezicht in de applicatie met de aanbevolen films voor de ingelogde gebruikers . .	95
B.6	Een informatie-overzicht van de geselecteerde film	95
B.7	Het instellingenscherf met links de knoppen om verschillende acties uit te voeren: a) volledig omhoog gescrold en b) volledig omlaag gescrold	96
B.8	Het instellingenscherf met links de knoppen om acties uit te voeren op de databank . . .	97
C.1	Vergelijking van de emotiedetectie bij de drie services toegepast op de laatste foto uit elke fotoreeks (deel 1)	115
C.2	Vergelijking van de emotiedetectie bij de drie services toegepast op de laatste foto uit elke fotoreeks (deel 2)	116
C.3	Vergelijking van de emotiedetectie bij de drie services toegepast op de middelste foto uit elke fotoreeks (deel 1)	117
C.4	Vergelijking van de emotiedetectie bij de drie services toegepast op de middelste foto uit elke fotoreeks (deel 2)	118
C.5	Vergelijking van de emotiedetectie bij de drie services toegepast op een hele reeks foto's van één emotie	119
C.6	Het UML-diagram van de volledige applicatie	120
C.7	Het activiteitendiagram van de applicatie	121
C.8	Het sequentiediagram voor het inlogproces	122
C.9	Het sequentiediagram voor genereren van de aanbeveling	123
C.10	Het sequentiediagram voor het verzamelen van de impliciete feedback	124
C.11	De spectra van de vier sets voor video 1, 2 en 3	125
C.12	De spectra van de vier sets voor video 4, 5 en 6	126
C.13	De emoties van testpersonen 13, 3, 8 en 4 bij video 1, 2 en 3	127
C.14	De emoties van testpersonen 13, 3, 8 en 4 bij video 4, 5 en 6	128

Lijst van tabellen

2.1	IMDb ratings per demografische groep voor de film ‘Madagascar’ [6]	7
2.2	De elf soorten aggregatiestrategieën volgens het werk van Judith Masthoff [7]	8
2.3	Ratings op tien per gebruiker voor verschillende films	9
3.1	Vergelijking van de drie gebruikte gezichtsdetectieservices	20
3.2	De resultaten van de leeftijds- en geslachtstesten voor de drie services toegepast op de twee datasets	22
3.3	De resultaten van de drie services toegepast op de dataset voor gezichtsdetectie van gesloten ogen, bedekte gezichtsdel en, aandachtsniveau en hoofdpositie	27
5.1	Een overzicht van de zes video’s die getoond worden aan de testpersonen	56
5.2	Een overzicht van de berekende scores	57
5.3	De maximale waarschijnlijkheidswaarden per gebruiker over alle video’s gesorteerd per emotie. De emotiegetallen betekenen: 0 = anger, 1 = disgust, 2 = fear, 3 = happiness, 4 = sadness, 5 = surprise	59
5.4	Resultaat van de switchtest bij de verwachte emoties uit set 1 voor video 1	64
A.1	Links naar sites voor het verkrijgen van de vereiste keys	90
C.1	Overzicht van de gebruikte nummering van emoties doorheen de tabellen en code	102
C.2	Overzicht van de resultaten van alle testpersonen per video bij de testen. De emotiegetallen betekenen: 0 = anger, 1 = disgust, 2 = fear, 3 = happiness, 4 = sadness, 5 = surprise	107
C.3	Overzicht van de resultaten van alle testpersonen per video bij set 3 (deel 1): de verwachte emotie staat voor elke film bovenaan in de ‘highest emotion’-kolom vermeld met het emotiegetal (0 = anger, 1 = disgust, 2 = fear, 3 = happiness, 4 = sadness, 5 = surprise)	108
C.4	Overzicht van de resultaten van alle testpersonen per video bij set 3 (deel 2)	109
C.5	Overzicht van de resultaten van alle testpersonen per video bij set 4 en 4.2 (deel 1): de verwachte emotie staat voor elke film bovenaan in de ‘highest emotion’-kolom vermeld met het emotiegetal (0 = anger, 1 = disgust, 2 = fear, 3 = happiness, 4 = sadness, 5 = surprise)	110
C.6	Overzicht van de resultaten van alle testpersonen per video bij set 4 en 4.2 (deel 2)	111
C.7	Verschil in geschatte leeftijd en geslacht voor de 20 testpersonen. Telkens schatte elke service de leeftijd vijf keer.	112
C.8	Antwoorden van de testpersonen op de vragenlijst over hun mening over de privacy en hoe goed ze expliciete beoordelingen zouden geven	113
C.9	Antwoorden van de testpersonen op de vragenlijst over hun impliciete feedback	114

Hoofdstuk 1

Inleiding

1.1 Huidige situatie: een overvloed aan films en series

Tegenwoordig zijn er talloze bedrijven die streamingdiensten aanbieden (zoals bijvoorbeeld Netflix en Yelo Play van Telenet) waarmee mensen vanuit hun zetel films en series kunnen bestellen en bekijken. Het enige wat de klant nog moet doen, is zijn film of serie kiezen. Gebruikers van deze diensten worden vaak geconfronteerd met een overvloed aan films en series waardoor het maken van deze keuze soms moeilijk kan zijn. Indien ze niet goed weten wat ze willen bekijken, moeten ze verschillende films overlopen en de beschrijving ervan lezen om een keuze te kunnen maken. Dit kan, tot frustratie van vele mensen, lang duren en leiden tot conflicten tussen de mensen die willen meekijken.

Om hen te helpen bij de zoektocht naar het gewenste product in deze chaos en bij het maken van hun beslissing, worden aanbevelingssystemen ingeschakeld die een persoonlijke selectie voorstellen. Deze selectie bevat de films of series waarin de gebruiker waarschijnlijk is geïnteresseerd en wordt opgebouwd op basis van impliciete feedback (bijvoorbeeld voorgaand kijkgedrag en zoekgeschiedenis) en expliciete feedback (bijvoorbeeld een beoordeling op tien punten of een sterrenrating). De impliciete feedback vertelt weinig over wat de gebruiker echt van de film vond en kan dus verbeterd worden met expliciete feedback. Het systeem kan zo om expliciete feedback vragen nadat de gebruiker de film heeft bekeken en beslissen of iets gelijkaardig in de toekomst kan worden aangeraden. Deze vorm van feedback is dus expliciet omdat de gebruiker moet nadenken wat hij van de film vond en dan zelf de score moet aanduiden of intypen.

1.1.1 Expliciete feedback geven, niet iedereen doet het

Het ingeven van de persoonlijke expliciete feedback kan zorgen voor een meer nauwkeurige aanbeveling. Er ontstaan hierbij enkele problemen. Vaak slaan de gebruikers de stap om expliciet hun feedback te geven over of wordt de feedback niet correct ingevuld omdat ze er niet altijd zin in hebben. Een film bekijken gebeurt ook vaak in groep maar de aanwezige kijkers willen achteraf zichzelf niet elk afzonderlijk aanmelden om hun feedback te geven. Ook bij de keuze van de film ontstaat er een probleem wanneer meerdere mensen samen een film willen kijken. Niet alle aanwezige kijkers willen zich manueel aanmelden zodat het systeem weet op wie hij de groepsaanbeveling moet baseren. Indien slechts één persoon is aangemeld op het account, zal de suggestie van films niet aangepast zijn aan alle kijkers.

Films worden vaak op een televisie afgespeeld en het navigeren met de afstandsbediening maakt aanmelden en feedback geven minder gebruiksvriendelijk. Dit kan als gevolg hebben dat gebruikers zich op accounts van andere mensen aanmelden waardoor de eigenaars van die accounts later aanbevelingen krijgen die niet op hun eigen kijkgedrag gebaseerd zijn. Al deze gevallen zorgen voor onvoldoende of foutieve data waardoor het aanbevelingssysteem geen nauwkeurige en persoonlijke aanbevelingen kan genereren. Een applicatie die op andere manieren aanwezige kijkers kan aanmelden, impliciete feedback over hen kan verzamelen en deze automatisch linken aan de persoon, zou zowel het gebruiksgemak als de nauwkeurigheid van de aanbeveling en het systeem verhogen.

1.2 Doelstelling van de masterproef

In deze masterproef zal een Androidapplicatie ontwikkeld worden waarin video's worden afgespeeld en de camera impliciete feedback verzamelt over de kijkers. De app heeft als doel om een persoonlijke en nauwkeurige aanbeveling gebaseerd op gezichtsherkenning en gezichtsdetectie te genereren met een minimum aan gebruikersinput. Gebruikers moeten zich niet manueel aanmelden aangezien de app de aanwezige gezichten automatisch aanmeldt met behulp van gezichtsherkenning. De impliciete feedback kan verzameld worden door de emoties van de kijkers te meten met behulp van emotiedetectie terwijl ze de film aan het bekijken zijn. Ook kan hun gedrag en aandachtsniveau worden gemeten met gezichtsdetectie zodat het systeem weet wie er geïnteresseerd is in de film. Deze feedbackgeschiedenis moet per gebruiker worden bijgehouden en later kan er dan, bij het detecteren van deze gebruiker, een aanbeveling gegenereerd worden op basis van zijn geschiedenis. Indien een gebruiker nog geen geschiedenis heeft - waardoor het systeem geconfronteerd wordt met het cold start probleem - kan er een aanbeveling gegenereerd worden op basis van zijn geschatte geslacht en leeftijd. De applicatie moet met alle aanwezige gebruikers rekening houden en genereert zo een groepsaanbeveling waarmee iedereen tevreden is. Een applicatie die dit kan realiseren zou de aanbevelingen nauwkeuriger en gebruiksvriendelijker maken.

1.2.1 Onderzoeksvraag en deelonderzoeksvragen

In deze masterproef wordt onderzocht hoe een persoonlijke aanbeveling kan worden gegenereerd enkel en alleen op basis van gezichtsherkenning- en gezichtsdetectieservices. Hiervoor wordt er allereerst onderzocht welke soorten impliciete feedback op basis van gezichtsdetectie en gezichtsherkenning het best kunnen bijdragen tot een persoonlijke aanbeveling. De impliciete feedback zal bestaan uit emotiedetectie, aandachtsdetectie en gedragsanalyse. Hiervoor wordt de output van verschillende gezichtsdetectieservices geaggregeerd en wordt er onderzocht welke van de services het meest geschikt is voor welk aspect. Een Androidapplicatie wordt ontwikkeld, getest op twintig testpersonen en aan de hand van de resultaten wordt onderzocht of de gezichtsdetectie de impliciete feedback correct registreert. Ten tweede wordt een oplossing voor het cold start probleem gezocht zodat gebruikers zonder geschiedenis ook reeds een aanbeveling kunnen krijgen.

Ten derde wordt er ook uitgebreid onderzocht hoe de getoonde emoties van kijkers kunnen bijdragen aan een inschatting van een waarderingsscore voor de video. Hiervoor wordt de mate van overeenkomst tussen de getoonde emoties en de verwachte emoties berekend per tijdstip. Er wordt nagegaan of het systeem op basis van de getoonde emoties van een gebruiker kan afleiden welke video de gebruiker heeft bekeken. Een onderzoek wordt tenslotte uitgevoerd naar hoe deze verwachte emoties het best bepaald worden.

1.2.2 Methode waarmee de applicatie werkt

Bij het opstarten van de applicatie wordt allereerst bepaald via de camera wie er allemaal aanwezig is. Voor deze eerste stap is dus gezichtsherkenning vereist. Met deze informatie weet het systeem op welke gebruikers de groepsaanbeveling gebaseerd moet worden. Gebruikers die reeds feedbackgeschiedenis hebben, worden aangemeld en hun geschiedenis wordt opgehaald als input voor de aanbeveling. Gebruikers die nieuw zijn voor het systeem en nog geen geschiedenis hebben, zullen een aanbeveling ontvangen gebaseerd op hun geschatte geslacht en leeftijd. Alle gebruikers van het systeem krijgen automatisch een profiel waarin hun geschatte geslacht, leeftijd en feedbackgeschiedenis wordt opgeslagen. Een gebruiker moet nooit zelf informatie geven aan de applicatie over zichzelf, het enige wat hij moet doen is een film kiezen uit de aanbevolen lijst.

In de tweede stap wordt de aanbeveling gegenereerd. De samenstelling van de groep kijkers zal worden bepaald en hiermee houdt het systeem rekening voor de aanbeveling. Zo wordt er geen horrorfilm aangeraden indien er twee volwassenen en een kind van zes jaar aanwezig zijn, aangezien het kind geen films aangeraden krijgt waarvan de minimumleeftijd te hoog ligt. Er wordt ook geen film aangeraden die één van de aanwezige mensen al gezien heeft. De groepsaanbeveling geeft een suggestie van video's die gemiddeld het hoogste scoren bij alle aanwezige mensen en onaanvaardbare video's worden uitgesloten uit de selectie. Op deze manier wordt er rekening gehouden met de voorkeuren van alle aanwezige personen.

Zoals reeds gezegd kan de aanbeveling op zowel geschiedenis als demografie van de kijkers gebaseerd worden. Bij de aanbeveling op basis van geschiedenis wordt er gekeken naar video's waar de gebruikers in het

verleden impliciet een goede waarderingsscore hebben gegeven. Films die hierop lijken doordat ze tags gemeenschappelijk hebben - zoals een rode draad doorheen vergelijkbare films -, zullen vervolgens worden aangeraden. Het schatten van de impliciete waarderingsscore gebeurt in stap 3. Door naar ratings van mensen uit dezelfde demografische groep te kijken, wordt de demografische aanbeveling gemaakt. Naarmate een persoon meer feedbackgeschiedenis verzamelt, wordt er steeds minder naar de demografische aanbeveling gekeken en steeds meer naar zijn eigen persoonlijke feedbackgeschiedenis.

Stap drie zorgt voor het verzamelen van impliciete feedback van de kijkers en het berekenen van de geschatte waarderingsscore van de kijker voor de film. Wanneer de kijkers naar de film aan het kijken zijn, maakt de camera beelden van hun gezichten. Uit deze beelden haalt de applicatie:

- Waar er gezichten aanwezig zijn in de foto (gezichtsdetectie).
- Elk gezicht identificeren (gezichtsherkenning). Deze stap is cruciaal en wordt steeds als basis uitgevoerd omdat het op elk moment moet gekend zijn over wie de gedetecteerde feedback gaat. Personen die nog niet ingelogd waren, worden nu ook ingelogd en ook voor hen wordt informatie verzameld.
- De waarschijnlijkheid van de emoties (boosheid, angst, blijheid, verdriet, verrassing en afkeer), het gedrag van de persoon (onder andere het sluiten van de ogen, het bedekken van mond en ogen en het vroegtijdig weggaan bij de film) en het aandachtsniveau.

De identificatie van de gebruiker wordt samen met zijn impliciete feedback vervolgens opgeslagen in de persoonlijke geschiedenis van de gebruiker. Op basis hiervan kan afgeleid worden of elke specifieke kijker het een goede of slechte film vond en wordt de geschatte score opgebouwd. Komt de emotie op dat moment in de film overeen met de verwachte emotie? Behoudt de kijker tijdens de volledige film zijn aandacht of kijkt hij vaak weg? Bedekt de persoon vaak zijn ogen of mond omdat hij de film te eng vindt? Heeft de persoon vaak de ogen dicht omdat hij de film niet aangenaam vindt om naar te kijken? Al deze informatie kan met behulp van gezichtsdetectiesoftware bepaald worden en deze informatie vertelt veel over welke gebruikers de film goed vonden en welke gebruikers niet. Als er bijvoorbeeld een komedie afgespeeld wordt en twee van de drie kijkers lachen veel, dan kan er besloten worden dat deze kijkers de film leuk vinden en dat de derde persoon een negatieve feedback geeft.

Het bepalen van de overeenkomst in getoonde emoties met de verwachte emoties is de grootste uitdaging in deze masterproef. De verwachte emoties eigen aan een film worden hier ook wel de *emotiefingerprint* genoemd en deze term slaat op het unieke patroon van de zes gedetecteerde emoties en hun probabiliteit in functie van de tijd. Voor elke film moet bepaald worden welke emoties wanneer verwacht worden en met welke probabiliteit. Met behulp van de getoonde emoties van testpersonen bij enkele zelfgekozen korte videofragmenten uit films waarbij op korte tijd veel emoties worden opgewekt, kunnen realistische emotiespectra onderzocht worden en als basis dienen voor emotiefingerprints.

Het hele systeem werkt dus eigenlijk rond het schatten van hoe goed elke gebruiker de bekeken film vond, schatten hoe sterk de gebruikers gelijkaardige films zouden appreciëren en op basis van de schatting de groepsaanbeveling genereren voor alle aanwezige kijkers.

1.3 Hoe er te werk wordt gegaan en hoe deze scriptie is ingedeeld

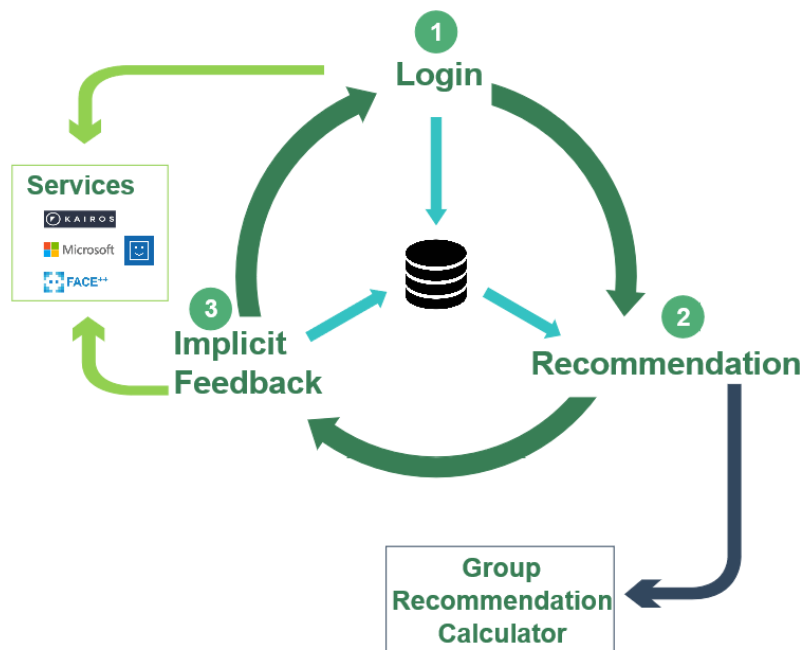
De indeling van de masterproef beschrijft de volgorde van de stappen die werden ondernomen in het onderzoek. Hoofdstuk 2 behandelt een onderzoek en literatuurstudie naar aanbevelingssystemen en gezichtsdetectiesystemen, de twee grote basiscomponenten die nodig zijn voor deze applicatie. Er worden enkele bestaande gelijkaardige systemen besproken. De principes die daarin gebruikt worden komen aan bod en waarom ze wel of niet kunnen worden toegepast in deze masterproef. In Hoofdstuk 3 worden de verschillende services met elkaar vergeleken alvorens kan bepaald worden welke services het best gebruikt worden in de applicatie. Ook wordt onderzocht hoe goed de emoties en andere feedback kunnen geregistreerd worden om er rekening mee te houden bij het bouwen van de applicatie. Hoofdstuk 4 beschrijft gedetailleerd de werking van de applicatie en de implementatie van de afzonderlijke onderdelen. Nadat de applicatie afgewerkt is, wordt deze getest op twintig testpersonen en de resultaten worden beschreven in Hoofdstuk 5. Hierin wordt de nauwkeurigheid van de impliciete feedback en de gegenereerde aanbeveling geëvalueerd. Bedenkingen over de app worden besproken in Hoofdstuk 6. Tenslotte worden de conclusies besproken in Hoofdstuk 7. Deze scriptie bevat ook drie bijlagen: een installatiehandleiding, een gebruikershandleiding en een bijlage die alle bijkomende grafieken, tabellen en diagrammen bevat.

Hoofdstuk 2

Componenten en bestaand werk

Zoals reeds verteld in de inleiding zal het genereren van de aanbeveling steeds uit drie stappen bestaan. De uiteindelijke app zal werken volgens de volgorde zoals getoond in Figuur 2.1. De eerste stap is steeds het inloggen van de aanwezige personen zodat het systeem weet op wie de aanbeveling moet gebaseerd worden. Het berekenen van de groepsaanbeveling gebeurt in stap 2 op basis van data uit de databank, waarna de gebruikers een film kunnen kiezen uit de aanbevolen lijst. Stap 2 maakt gebruik van een aanbevelingssysteem zoals getoond in Figuur 2.1 met een pijl naar *Group Recommendation Calculator*. Wanneer de film gekozen is, speelt deze af en begint stap 3 waarin tijdens het kijken naar de film informatie verzameld wordt over de kijkers. Deze informatie is de impliciete feedback zoals hun emoties, aandachtsniveau en gedrag. Tijdens het kijken worden bijkomende gebruikers, die initieel niet aanwezig waren, ook ingelogd zodat het systeem dit weet voor een eventuele volgende aanbeveling. Na het bekijken van de video kunnen gebruikers ervoor kiezen om bijkomende gebruikers in te loggen, opnieuw voor een eventuele volgende aanbeveling zodat de drie stappen weer opnieuw doorlopen worden.

Stap 1 (het inloggen) en stap 3 (het verzamelen van impliciete feedback van de kijkers) zijn volledig gebaseerd op gezichtsdetectie en gezichtsherkenning, namelijk identificatie van de persoon en het bepalen van leeftijd, geslacht, emoties en aandachtsniveau. Beide stappen zijn afhankelijk van gezichtsdetectieservices zoals Kairos, Microsoft Face API en Face++ en dit wordt in Figuur 2.1 aangeduid met een pijl naar *Services*. De services worden besproken in Hoofdstuk 3.



Figuur 2.1: De drie stappen gebruiken twee hoofdcomponenten: een aanbevelingssysteem en gezichtsdetectieservices

In de uiteindelijke applicatie zijn er dus twee grote systemen nodig: een eerste voor het genereren van een aanbeveling en een tweede voor de gezichtsdetectie. Dit hoofdstuk beschrijft deze twee hoofdcomponenten van het systeem. De verschillende manieren om een aanbeveling te maken en de algemene werking van aanbevelingssystemen wordt beschreven in 2.1. In 2.2 wordt er uitgelegd hoe gezichtsdetectie werkt en welke software en services er zullen gebruikt worden. Tenslotte beschrijft 2.3 reeds bestaande systemen die ook gebruik maken van deze principes.

2.1 Aanbevelingssystemen

Aanbevelingen zijn persoonlijke suggesties voor producten aan een gebruiker gebaseerd op zijn gekende voorkeuren. Een aanbevelingssysteem genereert deze suggesties en helpt zo de gebruiker bij het beslissen welk product hij zou kopen. Vele gebruikers worden namelijk geconfronteerd met een overvloed aan producten op online webshops, zoals Bol.com of Amazon, of een overvloed aan films of series op streamingdiensten zoals Netflix en Yelo Play. Door hen een beperkte selectie uit alle producten voor te stellen en deze selectie op te bouwen op basis van de gekende voorkeuren van de gebruiker, kan de gebruiker sneller zijn gewenste product vinden. Het bedrijf achter de online webshop haalt er ook voordelen uit omdat de gebruikers sneller zullen overgaan tot aankopen van het product als ze sneller het gewenste product vinden. Het bedrijf houdt er bovendien tevreden klanten aan over aangezien hun user experience wordt verhoogd. Als het systeem van bijvoorbeeld een online kledingshop weet dat de gebruiker een vrouw is, kan de selectie beperkt worden tot enkel vrouwenkleding. Het systeem kan nog meer leren over de gebruiker, door bijvoorbeeld aankoopgeschiedenis of feedback over producten bij te houden, en zo de aanbeveling nog persoonlijker maken.

Een aanbevelingssysteem wordt beschreven door Ricci [8] als een softwarehulpmiddel om suggesties voor producten te geven aan gebruikers om hen te helpen bij het maken van een beslissing. De aanbevelingen zijn gepersonaliseerd waardoor verschillende soorten gebruikers andere aanbevelingen zullen krijgen. Er bestaan ook niet-gepersonaliseerde aanbevelingen, zoals bijvoorbeeld de top 10 populairste boeken, maar dit wordt vaak niet gebruikt in een aanbevelingssysteem. Dietmar Jannach et al. [9] beschrijven dat een aanbevelingssysteem twee verschillende doelen kan hebben. Allereerst worden ze gebruikt om gebruikers aan te zetten tot het kopen van een product of tot het bekijken van een film. Ten tweede dienen ze ook om de gebruikers een beperkt aanbod te geven binnen het veel te grote aanbod van producten en informatie, de *information overload*.

Gepersonaliseerde suggesties vereisen dat er informatie gekend is over de gebruiker [9]. Dit is nodig voor elke gebruiker afzonderlijk. Bijgevolg moet het aanbevelingssysteem een profiel bijhouden voor elke gebruiker. In dit profiel wordt dan allerlei informatie bijgehouden, zoals voorkeuren van de gebruiker, aankoopgeschiedenis en kijkgeschiedenis. Op deze manier kan er met deze informatie later voorspeld worden welke andere producten de gebruiker nog zou willen kopen. Ook in deze masterproef zal er gewerkt worden met een geschatte waarderingsscore die de gebruiker aan producten (hier dus video's) zou geven om op basis hiervan een aanbevolen lijst van producten te genereren.

2.1.1 Soorten systemen

Welke informatie en hoe deze informatie wordt verkregen, hangt af van het aanbevelingssysteem. Zo kunnen de voorkeuren van de gebruiker expliciet worden gevraagd, bijvoorbeeld door de gebruiker een sterrating te laten geven, of impliciet, bijvoorbeeld door zijn aankoopgedrag te analyseren. Er zijn ook verschillen in welke soort informatie er wordt gevraagd. Indien er gewerkt wordt met het gedrag en voorkeuren van een grote gemeenschap van andere gebruikers, spreken Dietmar Jannach et al. over *community based* of *collaborative recommendation* en dat het idee hierachter is dat “als gebruikers dezelfde interesses deelden in het verleden - als ze bijvoorbeeld dezelfde boeken bekeken of gekocht hebben - ze ook dezelfde smaken in de toekomst zullen hebben” [9]. Zo kan een boek dat gebruiker A heeft gekocht aan gebruiker B worden aangeraden indien gebruikers A en B in het verleden ooit hetzelfde boek hebben gekocht. Dit is zeker herkenbaar voor vele mensen als ze op het internet aan het zoeken zijn naar producten: “Klanten die dit product kochten, kochten ook: ...”. Deze collaboratieve filter methode is dan ook de meest gebruikte manier om aanbevelingen te genereren. Het voordeel van deze methode is dat er geen extra informatie over de producten zelf moet worden opgeslagen, zoals wie de schrijver van het boek is of welke

acteurs meespelen in de film. Anderzijds kan deze informatie wel interessant zijn om later een aanbeveling te geven van een boek of film die sterk lijkt op vorige aankopen, bijvoorbeeld een film met dezelfde acteurs.

Een andere methode is *content-based recommendation* en is, in tegenstelling tot de collaboratieve filtering, wel gebaseerd op beschrijvingen van de producten en de profielen [9]. Zo kan een profiel de informatie bevatten dat die persoon liever producten koopt die voldoen aan bepaalde eigenschappen, bijvoorbeeld een bepaalde regisseur of acteur. De profielen kunnen automatisch ingevuld worden op basis van aankoopkeuzes van de gebruiker of door feedback te vragen met bijvoorbeeld een sterrating. Het principe steunt op eigenschappen van producten en voorkeuren van gebruikers voor bepaalde eigenschappen en het aanbevelingssysteem moet de producten bepalen die het beste passen bij de voorkeuren van de gebruiker [9]. Het systeem raadt dus producten aan die lijken op producten die de gebruiker in het verleden leuk vond. Het voordeel van de methode is dat nieuwe producten, die nog nooit door een gebruiker aangekocht zijn, ook al meteen kunnen worden aangeraden aangezien hun beschrijving en informatie over de eigenschappen al werd ingevuld. Het nadeel is dan weer dat deze informatie manueel moet worden ingevuld.

Indien er geen geschiedenis bekend is, kunnen bovenstaande methoden niet worden toegepast. Er kan dan een niet-gepersonaliseerde aanbeveling gegeven worden, bijvoorbeeld de top 10 populairste boeken, maar dit wordt liever vermeden. Een andere optie is om met extra informatie over de gebruiker een aanbeveling te maken en dit heet *knowledge-based recommendation* [9]. Zo kan er gevraagd worden aan de gebruiker naar welk genre hij of zij op zoek is, welke specificaties er vereist zijn enzovoort. Op deze manier heeft het systeem toch nog iets van informatie, aangezien het de informatie niet uit de geschiedenis van het profiel kan halen. Het systeem gaat dan deze informatie koppelen aan de eigenschappen van de producten om op die manier een aanbeveling te genereren. Het nadeel hiervan is dat er veel interactie met de gebruiker nodig is om al zijn vereisten in te geven.

De applicatie in de masterproef streeft naar zo weinig mogelijk interactie met de gebruiker en bijgevolg zal er geen informatie aan de gebruiker gevraagd worden zoals in de knowledge-based methode. De applicatie zal daarentegen werken volgens het content-based principe en *tags* toekennen aan video's die de eigenschappen van de video bevatten. Een waarderingsscore voor verschillende tags per gebruiker wordt bepaald op basis van voorgaande bekeken video's die deze tags bevatten. Het aanbevelingssysteem zal vervolgens video's aanraden die het beste passen bij de gebruiker op basis van de waarderingsscore van de tags zodat de aanbevolen video's lijken op video's die de gebruiker leuk vond in het verleden. Indien de gebruiker nog geen geschiedenis heeft en het systeem bijgevolg geconfronteerd wordt met het *cold start* probleem, kan de content-based methode niet worden toegepast. Een oplossing voor dit probleem wordt gegeven in 2.1.3.

2.1.2 Ratings

Aanbevelingssystemen kunnen gebruik maken van ratings die door de gebruikers worden gegeven aan de aangekochte producten of bekeken films. Hierdoor weet het systeem hoe goed de gebruiker het product vond en heeft het systeem een indicatie van de interesses van de persoon zodat er later een aanbeveling met deze informatie kan gegenereerd worden. Het expliciet vragen aan de gebruikers wat ze van het product vonden en welke score ze eraan toekennen, is de meest nauwkeurige manier. Zo kunnen ze een sterrating op vijf geven of een score op tien. Het probleem bij expliciete ratings is dat de gebruikers extra moeite moeten doen om na te denken over hun mening en om deze in te geven. Op een smartphone, tablet of laptop is het geven van zo een rating vrij makkelijk. De gebruiker klikt of raakt het aantal sterren aan op het scherm dat hij wilt geven, en klikt vervolgens op verzenden. Het ingeven van de rating op een smart TV is al wat lastiger omdat een afstandsbediening niet zo snel werkt als een touchscreen of een computermuis.

In de context van het bekijken van films zal de smart TV optie vaker gebruikt worden en is het mogelijk dat gebruikers het geven van de rating overslaan omdat dit niet gebruiksvriendelijk is. Hierdoor kan het aantal ratings te klein worden om nuttige aanbevelingen te geven en zal het aanbevelingssysteem minder nauwkeurig werken. Een andere optie is dus om impliciet feedback van de gebruikers te gaan verzamelen. Een voorbeeld hiervan is de aankoopgeschiedenis van de gebruiker. Als een gebruiker een product aangekocht heeft of een film besteld heeft, krijgt dit product in het systeem een positieve rating. Er wordt dus uitgegaan van het feit dat een gebruiker dingen koopt of bestelt waarin hij waarschijnlijk geïnteresseerd is. Het systeem kan ook het gedrag van de gebruiker gaan analyseren. Als een gebruiker lang op de

informatiepagina van een product blijft of een trailer van een bepaalde film volledig bekijkt, kan er van uit gegaan worden dat de gebruiker deze producten een positieve rating geeft. Deze impliciete ratings vragen dus geen extra inspanningen van de gebruiker maar het kan zijn dat de verzamelde informatie niet correct is, bijvoorbeeld een de gebruiker die na het kijken van de trailer de film toch niet wilt bekijken.

In deze masterproef zal onderzocht worden hoe impliciete feedback nauwkeuriger kan bepaald worden dan enkel op basis van aankoopgeschiedenis. Er wordt gekeken naar welke emoties de gebruikers tonen tijdens het kijken naar een film. Ook wordt er gekeken naar het aandachtsniveau en het gedrag. Met deze informatie kan geanalyseerd worden of de getoonde emoties op dat moment in de film overeenkomen met de verwachte emoties. Er kan ook bepaald worden wie continu aandachtig de film aan het volgen is en wie er vaak weg kijkt of niet meer aanwezig is voor de camera en de tv. Deze informatie vertelt veel over hoe goed de gebruikers de film vonden en dient dus eigenlijk als een impliciete rating voor de film waarmee het systeem de waardering van de kijkers kan inschatten. Gebruikers moeten dus zelf geen expliciete rating of score gaan ingeven na het kijken van de film want het systeem schat hun waarderingsscore op basis van hun impliciete feedback.

2.1.3 Cold start probleem

Zoals reeds aangehaald, bestaat er steeds een probleem bij het aanbevelen van iets aan een gebruiker waarvan nog geen geschiedenis bekend is. Dit probleem staat ook wel bekend als het *cold start* probleem dat ontstaat wanneer er een gebrek is aan informatie over de voorkeuren van nieuwe gebruikers [10]. Er zijn drie soorten cold start problemen: aanbevelingen voor nieuwe gebruikers, aanbevelingen voor nieuwe producten en aanbevelingen voor nieuwe gebruikers en nieuwe producten. In het artikel van Blerina Lika [10] gaan ze uit van het eerste probleem waarbij nieuwe gebruikers het systeem gebruiken. In een eerste fase worden demografische gegevens gebruikt (bijvoorbeeld leeftijd en geslacht) samen met de voorkeuren van de gebruikers in het model. Het idee is “dat mensen met een gemeenschappelijke achtergrond meer kans hebben dezelfde voorkeuren te hebben” [10]. Het doel van deze fase is om burens te vinden, gebaseerd op informatie van bestaande gebruikers, voor elke nieuwe gebruiker. De burens zijn bestaande gebruikers die tot dezelfde categorie behoren, en dus dezelfde eigenschappen hebben als de nieuwe gebruiker. In de tweede fase wordt een gewogen gemiddelde genomen van de demografische gegevens waarmee de gelijkheid wordt geschat tussen elke nieuwe gebruiker en elke buur. In de laatste fase wordt een voorspelling gemaakt voor elk product in het systeem door de gelijkheidsscores uit de tweede fase te combineren met de score van de burens van het mogelijks aanbevolen product.

In deze masterproef zal het cold start probleem worden opgelost door de leeftijd en het geslacht van de personen zonder geschiedenis te bepalen met gezichtsdetectie. In een databank worden verschillende ratings per film bijgehouden. De ratings worden gegroepeerd per geslacht- en leeftijdsgroep zoals te zien in Tabel 2.1 voor de animatiefilm ‘Madagascar’. De demografische ratings in deze tabel zijn afkomstig van de IMDb.com-site [6]. Deze site voorziet informatie over films, televisieseries, video games enzovoort. Gebruikers van de site kunnen een rating geven aan een film of serie waardoor de film of serie een gemiddelde globale rating krijgt op 10. Hiermee kunnen mensen die een bepaalde film willen bekijken, opzoeken wat andere mensen van over de hele wereld van de film vonden. Wanneer een gebruiker een rating geeft, wordt ook zijn leeftijd en geslacht bij die rating opgeslagen. De site voorziet dus naast de globale rating ook een overzicht van de ratings verspreid over de verschillende demografische groepen.

Tabel 2.1: IMDb ratings per demografische groep voor de film ‘Madagascar’ [6]

Leeftijd (jaren):	<18	18-29	29-45	>45
Mannen	7,2	6,9	6,8	7,3
Vrouwen	7,3	7,2	7,1	7,5

Deze informatie is nuttig om het cold start probleem in dit project op te lossen. De leeftijd en het geslacht van aanwezige gebruikers wordt bepaald en op basis hiervan worden een aantal films aangeraden die een hoge score krijgen in de ratingtabel voor hun leeftijdsgroep en geslacht. Het globale gemiddelde van alle ratings van de film ‘Madagascar’ is 6,9/10. Indien er dan voor elke demografische groep het verschil wordt genomen met dit globale gemiddelde, verschijnt er een duidelijk verschil tussen de demografische

groepen. Zo wordt het duidelijk dat personen onder de 18 jaar gemiddeld 0,3 punten voor mannen en 0,4 punten voor vrouwen boven het globale gemiddelde scoren, terwijl mannen boven de 30 jaar gemiddeld 0,1 punten lager scoren. Deze relatieve rating kan dan gebruikt worden bij het aanbevelen van een film afhankelijk van het geslacht en de leeftijd van de aanwezige personen. Indien de groep kijkers maar uit één persoon bestaat, wordt er een persoonlijke aanbeveling gegeven op basis van geslacht en leeftijd van die persoon. Indien de groep kijkers uit meerdere personen bestaat, zal de leeftijd en het geslacht van alle personen moeten meetellen om de aanbeveling te maken. Er zal dan een groepsaanbeveling moeten worden gegenereerd die films aanbiedt die geschikt zijn voor alle aanwezige personen. Meer informatie over de groepsaanbevelingen wordt hieronder in 2.1.4 gegeven.

2.1.4 Groepsaanbevelingen

Als er een goede aanbeveling kan gemaakt worden voor individuele gebruikers, is de volgende vraag hoe men dit ook kan doen voor een groep gebruikers. Online aanbevelingen voor producten op webshops zijn vaak gericht op één persoon. Het kijken van een film speelt zich echter vaak af in groepsverband en moet men rekening gehouden worden met wie er zich allemaal voor het scherm bevindt. Het uiteindelijke doel van de groepsaanbeveling is dat alle aanwezige kijkers tevreden zijn met de producten die worden aanbevolen [7]. De aanbeveling moet zich dus aanpassen aan de volledige groep gebaseerd op informatie van de voorkeuren van de individuele gebruikers.

Groepsaanbevelingen kunnen op verschillende manieren geclassificeerd worden, bijvoorbeeld afhankelijk van of de voorkeuren van de gebruikers gekend zijn op voorhand of ontwikkeld worden met de tijd, hoe de individuele voorkeuren worden bepaald, of er gewerkt wordt met het aggregeren van de aanbevelingen of van de profielen. Dit aggregeren is een belangrijk punt bij groepsaanbevelingen. Hierbij worden er eerst individuele aanbevelingen gemaakt die daarna worden samengevoegd tot één grote groepsaanbeveling. Bij het aggregeren van de profielen worden de individuele voorkeuren van alle relevante gebruikers samengevoegd. Deze verzameling van voorkeuren wordt dan gebruikt om een groepsaanbeveling te maken. In het werk van Judith Masthoff [7] worden elf aggregatiestrategieën verklaard. Een overzicht van de elf aggregatiestrategieën wordt hieronder gegeven in Tabel 2.2.

Tabel 2.2: De elf soorten aggregatiestrategieën volgens het werk van Judith Masthoff [7]

Strategie	Hoe het werkt
Plurality Voting	Er wordt gestemd op de items en het item met de meeste stemmen wordt verkozen
Average	Neemt het gemiddelde van alle scores
Multiplicative	Vermenigvuldigt alle scores per item om een totaalscore per item te krijgen
Borda Count	De som van de rankingposities per persoon wordt bepaald per item
Copeland Rule	Telt hoe vaak een item een ander item verslaat en vermindert dit aantal met het aantal keer het verliest
Approval Voting	Telt het aantal personen met stemmen boven een bepaalde drempel voor dit item
Least Misery	Neemt het minimum van de individuele ratings
Most Pleasure	Neemt het maximum van de individuele ratings
Average without Misery	Neemt het gemiddelde van de individuele ratings, waarvan de ratings onder een bepaalde drempel uit werden weggelaten waardoor er nooit iets kan worden aangeraden dat minstens één persoon heel slecht vindt
Fairness	De items worden gesorteerd in de volgorde waarin de gebruikers ze elk om de beurt zouden kiezen
Most respected person	Gebruikt de score van de meest gerespecteerde persoon

Het gemiddelde nemen met de *Average*-strategie bepaalt voor de scores van alle gebruikers het gemiddelde per film. De film met het grootste gemiddelde wordt gekozen als de beste keuze voor de groep. De *Average without Misery*-strategie is een manier om ervoor te zorgen dat er nooit iets wordt aanbevolen dat minstens één persoon uit de groep echt niet wilt. Met de voorbeelddata uit Tabel 2.3 kan een groepsaanbeveling gemaakt worden voor de drie gebruikers volgens deze strategie. De tabel toont de geschatte rating die de gebruiker zou hebben voor elk van de vier films. Om deze rating te schatten, moet het systeem een interesseniveau per gebruiker per film kunnen bepalen op basis van voorgaand kijkgedrag of andere persoonlijke informatie. Er is dus op dit moment per gebruiker al een persoonlijke aanbeveling gegenereerd in de vorm van een rating op tien per gebruiker per film. De volgende stap is nu om deze informatie te aggregeren zodat de beste film voor alle drie de gebruikers samen kan worden aangeraden. De onderste rij in de tabel toont de gemiddelde rating per film. Film A en film C hebben beide een score van 7/10. Eén van deze films zal dus gekozen moeten worden als beste film.

Tabel 2.3: Ratings op tien per gebruiker voor verschillende films

	film A	film B	film C	film D
gebruiker 1	10	6	7	2
gebruiker 2	9	5	8	7
gebruiker 3	2	6	6	8
gemiddelde	7	5,7	7	5,7

Film A scoort heel hoog bij gebruiker 1 en 2 maar heel laag bij gebruiker 3. Film C scoort vrij goed bij alle drie de gebruikers maar bij niemand heel goed. Bij de strategie *Average without Misery* kan er een drempelwaarde opgelegd worden voor scores. Films met ratings die onder deze drempelwaarde liggen worden sowieso verworpen uit de lijst van mogelijke beste films. Als de drempel op 5/10 ligt, betekent dit dat film A als onaanvaardbaar voor gebruiker 3 zou worden beschouwd. Bijgevolg is film C de beste keuze aangezien deze bij geen enkele gebruiker onder de drempel scoort. Film D scoort voor gebruiker 1 ook onder de drempelwaarde en wordt ook als onaanvaardbaar beschouwd indien meerdere films kunnen worden aangeraden. Uiteindelijk zou film C op plaats 1 komen te staan en film B op plaats 2 in de aanbeveling.

Deze strategie kan ook in dit project worden toegepast. Als er een film aanwezig is in de lijst van mogelijke films waarvan zeker is dat minstens één gebruiker die niet leuk zou vinden, wordt deze film automatisch geëlimineerd. Er kunnen nog bijkomende criteria worden opgelegd waarmee onderscheid kan gemaakt worden of een film tot de kandidaat-films mag behoren of tot de onaanvaardbare. Een extra criterium in de applicatie zal bijvoorbeeld de minimumleeftijd van een film zijn. Als er bijvoorbeeld vier mensen voor de televisie zitten, drie volwassen personen en één kind, wordt er best geen horrorfilm aangeraden. De horrorfilm heeft een vooraf ingestelde minimumleeftijd en wanneer het kind jonger is dan de minimumleeftijd, wordt de horrorfilm als onaanvaardbaar beschouwd. De horrorfilm wordt vervolgens uit de mogelijk kandidaten weggelaten. Dit criterium zal zowel voor de aanbeveling voor mensen zonder geschiedenis als voor mensen met geschiedenis worden toegepast.

Een ander criterium voor onaanvaardbaarheid is wanneer sommige gebruikers een film reeds bekeken hebben. Dit kan afgeleid worden uit hun kijkgeschiedenis en reeds bekeken films zullen als onaanvaardbaar worden beschouwd. In Hoofdstuk 4 wordt uitgebreid beschreven hoe de groepsaanbevelingsstrategie praktisch zal worden uitgewerkt.

2.2 Gezichtsdetectie

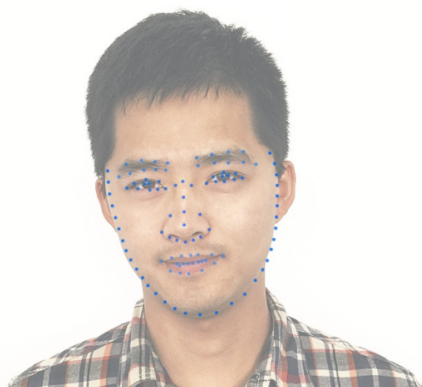
Gezichtsdetectie is een technologie die sinds enige tijd bestaat en talloze toepassingen kent, onder andere: herkenning van emoties, inschatten leeftijd en geslacht, bepalen van unieke gezichtsfeatures en gezichtsherkenning. Voor deze masterproef zullen bestaande services gebruikt worden die *cloud based* zijn waardoor het rekenwerk voor de detectie niet lokaal op het toestel wordt uitgevoerd. De foto, waarop de gezichtsdetectie zal worden toegepast, wordt verstuurd over het internet. Vervolgens ontvangt de applicatie alle informatie over die foto. Hieronder wordt de algemene werking van gezichtsdetectie beschreven. De verschillende services die gebruikt worden in de applicatie worden besproken in Hoofdstuk 3 waar ze met elkaar worden vergeleken.

2.2.1 Gezichtsdetectie versus gezichtsherkenning

Gezichtsdetectie is niet hetzelfde als gezichtsherkenning. Bij gezichtsdetectie gaat het over het vinden van gezichten van mensen in digitale foto's. De computersoftware moet de volledige foto overlopen en bepalen of er één of meerdere gezichten aanwezig zijn en waar ze zich bevinden. Gezichtsdetectie is eigenlijk de basis van alle gezichtsanalyses, zoals gezichtslokalisatie, bepaling van *facial features*, gezichtsherkenning, verificatie van gezichten en herkennen van gezichtsuitdrukkingen [11]. Het zijn allemaal zaken die nauw verwant zijn met gezichtsdetectie maar gezichtsdetectie is steeds de eerste stap [12]. Gezichtsdetectie kan ook gebruikt worden om het geslacht en de leeftijd van een persoon in te schatten.

Gezichtsdetectie is ook de eerste stap in het bepalen van de *facial features* of *gezichtsfeatures*. Deze features zijn de interessante delen van het gezicht, bijvoorbeeld ogen, neus en mond. Bij bepaling van gezichtsfeatures wordt de locatie van deze features bepaald en veel computervisie-algoritmen maken gebruik van deze informatie [13]. Soms worden deze ook *landmarks* genoemd. Figuur 2.2 demonstreert de verschillende landmarks gedetecteerd door de service Face++ op een gezicht.

Doc Center / Landmarks Illustration (106 Points)



Figuur 2.2: Een illustratie van de service Face++ met 106 landmarks aangeduid op een gezicht [1]

Als de verschillende features gelokaliseerd zijn, kan de software hier nog extra features uit berekenen, zoals de afstand tussen de ogen, de breedte van de neus, de lengte van de kaaklijn enzovoort. Deze getallen en de combinatie van de verschillende features zijn uniek aan een gezicht en kunnen gebruikt worden voor gezichtsherkenning. Priyanka R. et al. beschrijven verschillende methoden om de locatie van de gezichtsfeatures te bepalen, zoals low-level methoden met hoek- en randdetectie en methoden gebaseerd op beweging, vormen en gekende informatie over het gezicht [13]. Verschillende methoden worden verder uitgelegd in 2.2.2

Bij gezichtsherkenning bepaalt de software van wie de gezichten zijn in de foto. Het gaat dus om het vastleggen van de identiteit van de persoon en het is dus niet hetzelfde als gezichtsdetectie. Hiervoor moet de software een verzameling van mogelijke gezichten bijhouden in een databank en het te identificeren gezicht testen op elk bestaand gezicht om te kijken of het ermee overeenkomt [12]. Dit gebeurt door unieke gezichtsfeatures van de foto's te bepalen en op te slaan in de databank in plaats van de foto's zelf. Priyanka R. et al. verduidelijken dat de gezichtsherkenning uit vier stappen bestaat [13]. Eerst wordt de ingegeven afbeelding onderverdeeld in segmenten. Daarna worden de grenzen van het gezicht in de afbeelding en de gezichtsfeatures gelokaliseerd. In de derde stap worden de verkregen features vergeleken met de features van de andere gezichten aanwezig in de databank. Als er features van een bestaande persoon overeenkomen, weet de software dat het te identificeren gezicht overeenkomt met die persoon. Tenslotte wordt bepaald tot welke persoon het gezicht behoort.

Gezichtsherkenning slaat gezichten of gezichtsfeatures op, wat gezichtsdetectie niet doet. In deze masterproef wordt zowel gezichtsdetectie als gezichtsherkenning gebruikt. Gezichtsherkenning is een stap die bij elke genomen foto zal worden toegepast aangezien het systeem steeds moet weten over wie het gaat. De identificatie speelt dus een belangrijke rol bij het maken van de aanbeveling want elke emotie die ge-

registreerd wordt, moet correct gekoppeld worden aan de juiste persoon zodat persoonlijke feedback kan worden opgeslagen. Ook bij het aanmelden moet de identiteit van de aanwezige personen automatisch bevestigd worden met gezichtsherkenning. Er kan dus gezegd worden dat de gezichtsherkenning centraal staat in de applicatie en aanvullend wordt gezichtsdetectie afzonderlijk gebruikt om leeftijd, geslacht, gesloten ogen, emoties, aandachtsniveau enzovoort te bepalen.

2.2.2 Werking

Er zijn verschillende manieren om een gezicht te lokaliseren in een foto. Hier worden kort de verschillende concepten beschreven. Yang et al. geven een onderverdeling in vier groepen [12]:

- **Knowledge-based methoden** waarbij er wordt uitgegaan van menselijke kennis over het gezicht. Een gezicht bestaat uit twee ogen, een neus en een mond met onderlinge afstanden en in deze methode wordt hiernaar op zoek gegaan.
- **Feature invariant methoden** proberen features te zoeken in verschillende situaties zoals variërende belichting en positie.
- **Template matching methoden** waarbij standaard patronen van gezichten worden opgeslagen en de correlatie met een input afbeelding wordt berekend om te bepalen of er een gezicht in aanwezig is.
- **Appearance-based methoden** waarbij de patronen worden aangeleerd uit een dataset met afbeeldingen voor training. Deze methoden gebruiken statistische analyse en *machine learning* om patronen van gezichten en van niet-gezichten te vinden.

De laatste categorie van methoden, de Appearance-based methoden, blijken de beste performantie te hebben volgens de paper van C. Zhang et al. [14]. De methoden gebaseerd op leeralgoritmes hebben al bewezen dat ze uitstekende resultaten geven [12]. Verder in dit hoofdstuk wordt het principe van machine learning verder uitgelegd samen met de valkuilen. Bij de feature invariant methode wordt er uitgegaan van het feit dat een computer ook gezichten moet herkennen in omstandigheden met minder belichting en andere posities, aangezien mensen dit ook kunnen. Er wordt aangenomen dat er features moeten bestaan van het gezicht die invariant zijn aan de omstandigheden. Ze gaan dus eerst features zoals wenkbrauwen en ogen detecteren en dan pas de aanwezigheid van een gezicht [12]. Voorbeelden van zo een feature zijn textuur en huidskleur.

Low-level analyse houdt zich bezig met de opdeling in eigenschappen van pixels zoals kleur [15]. Indien de foto in kleur is, kan de kleur van de huid gebruikt worden om een gezicht te detecteren in de zogenaamde *color-based* methoden. Rein-Lien Hsu et al. beschrijven in hun artikel hoe hun algoritme gebieden van huid detecteert en zo mogelijke gezichten verzamelt op basis van hun ruimtelijke plaatsing [16]. Voor elk mogelijk gezicht worden dan de ogen, mond en gezichtsranden gelokaliseerd om te bepalen waar de gezichten zich bevinden. Het algoritme werkt met variaties in kleur en grootte van gezichten maar heeft moeilijkheden met drukke achtergronden en variaties in belichting.

Indien de software een foto ontvangt met een gezicht en een achtergrond dat slechts uit één kleur bestaat, moet de software enkel de achtergrond weglaten en wordt het gezicht snel gevonden. De meeste databanken met gezichten zijn dan ook gemaakt om het zoeken naar gezichten te vereenvoudigen door gecontroleerde achtergronden te gebruiken. Een andere uitdaging bestaat er dus in om gezichten te zoeken in foto's zonder beperkingen, zoals foto's uit het dagelijkse leven. Dit heet *unconstrained face recognition* en wordt toegepast op bijvoorbeeld de databank 'Labeled Faces in the Wild' waar alle foto's natuurlijke variaties in belichting, achtergrond en andere factoren bevatten [17].

Machine learning en zijn valkuilen

Eerder in dit hoofdstuk werd machine learning al eens vermeld als een mogelijke manier om gezichts-detectie uit te voeren. Het wordt over het algemeen vaak gebruikt in toepassingen waar manueel programmeren niet mogelijk of niet efficiënt is. De computer, een machine, neemt dan dit werk over. Het machine learning algoritme kan zelf uitzoeken hoe hij dingen moeten geprogrammeerd en geïnterpreteerd

worden. Dit doet het algoritme door voorbeelden aan te leren en hiervan een veralgemening te maken [18]. Machine learning is dus eigenlijk letterlijk een machine iets aanleren. Gezichtsdetectie is een van de vele toepassingen ervan en de detector van Viola-Jones, verder in dit hoofdstuk uitgelegd, maakt hier ook gebruik van.

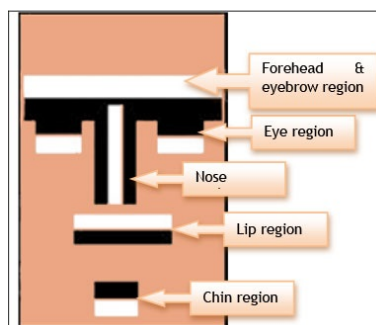
Er bestaan verschillende soorten van leeralgoritmen. Eén ervan is *classification*. Dit classifier systeem krijgt een zekere input, bijvoorbeeld een verzameling van allerlei features, en genereert een output in een discrete vorm zoals bijvoorbeeld: “Ja, het is een gezicht” of “Nee, het is geen gezicht” [18]. Om te zorgen dat de classifier de juiste output geeft, moet deze getraind worden. Dit gebeurt door een *learner* die een verzameling trainingsdata als input geeft en bij elke input een verwachte outputwaarde geeft. De learner verwerkt de input naar gegenereerde output en kijkt of deze overeenkomt met de verwachte outputwaarde. Door het proces te herhalen met verschillende trainingsdata zal de classifier voldoende getraind zijn om nieuwe, ongeziene inputdata te verwerken tot een (meestal) correct geklasseerde discrete output. Het hoofddoel van machine learning is om zodanig te veralgemenen voorbij de aangeleerde voorbeelden van de trainingsset [18]. Het herkennen van gezichten werkt ook op deze manier. Een eerste gezicht moet als input worden meegegeven en hieruit kan het algoritme leren. Toekomstige gezichten kunnen dan als input gegeven worden en er wordt bepaald of het gezicht wel of niet tot een bepaalde persoon behoort.

Eén van de problemen bij machine learning is *overfitting*. Dit gebeurt als het algoritme te veel getraind wordt waardoor het te hard lijkt op of exact hetzelfde is als een deel van de trainingsdata. Daardoor kan het algoritme nieuwe data niet correct klasseren. De classifier past “té hard” op de trainingsdata. Het model begint de trainingsdata te herinneren in plaats van te veralgemenen. Het systeem zal nu falen als het dingen moet klasseren dat het nog nooit heeft gezien.

Viola-Jones gezichtsherkenning

De laatste jaren heeft de Viola-Jones face detector de meeste impact gehad en wordt deze in veel applicaties gebruikt [19]. Het algoritme is robuust, real time en werkt enkel voor gezichtsdetectie, niet voor gezichtsherkenning. Het gebruikt onder andere het AdaBoost leeralgoritme dat gebruikt wordt om goede features te selecteren en *classifiers* te trainen. De paper van Paul Viola en Michael J. Jones beschrijft de drie bijdragen aan het algoritme [19]:

- De *Integral Image* dat gebruikt wordt voor snelle evaluatie van de features. Ze gebruiken features die lijken op de *Haar* functies. Dit zijn Haar-like features die hun naam te danken hebben aan de *Haar wavelets* die er sterk op lijken en werden besproken door Papageorgiou et al. in hun paper voor object detection [20]. De rechthoekige Haar-like feature worden, in alle mogelijke grootten, op de afbeelding gelegd, zoals getoond in Figuur 2.3. Hierin bestaan sommige rechthoeken uit twee gebieden, een *two-rectangle feature*, en sommige uit drie gebieden, een *three-rectangle feature*. De two-rectangle feature berekent het verschil tussen de som van pixels tussen de twee rechthoekige gebieden. De three-rectangle feature berekent de som binnen de twee buitenste rechthoeken verminderd met de som van de binnenste rechthoek [19].

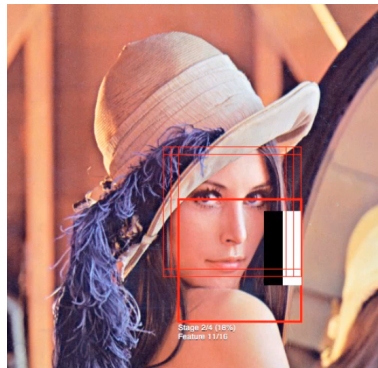


Figuur 2.3: Haar-like features van een menselijk gezicht gedemonstreerd in de paper van B. Ali et al. [2]

Deze berekende waarden duiden op specifieke eigenschappen van dat deel in de afbeelding. Zo zijn de Haar features gebaseerd op het feit dat alle menselijke gezichten dezelfde eigenschappen hebben,

bijvoorbeeld dat het hoogste stuk van de neus lichter is dan de ogen. Indien zo een eigenschap wordt waargenomen in een feature weet het systeem dat er misschien een neus en twee ogen aanwezig kunnen zijn. In Figuur 2.3 worden de verschillende mogelijke Haar-like features op een menselijk gezicht getoond. Als de combinatie van de verschillende features wordt gevonden in een regio van een afbeelding, wordt de regio gemarkeerd als een potentieel gezicht [2]. Dankzij de Integral Image, een tussenliggende representatie, kunnen de Haar-like features heel snel berekend worden.

- Een eenvoudige en snelle classifier die een selectie van de belangrijkste features maakt uit een grote verzameling gerealiseerd door het AdaBoost leeralgoritme. Indien dit niet zou gebeuren, zouden er te veel features zijn. De beste worden dus geselecteerd en de classifiers die ze gebruiken, worden getraind.
- Het combineren van complexe classifiers in een cascade. Hierdoor wordt de focus gelegd op gebieden in de foto waar er meer kans is een gezicht aan te treffen. De stappen in de cascade worden gesorteerd op hun complexiteit en deelaafbeeldingen worden erdoor gestuurd. Elke stap in de cascade bepaalt of er zeker geen gezicht aanwezig is in de deelaafbeelding, waardoor de deelaafbeelding verwijderd wordt, of er misschien een gezicht aanwezig is, waardoor het door mag naar de volgende stap in de cascade.



Figuur 2.4: Het Viola-Jones algoritme in werking volgens een video gemaakt door Adam Harvey [3, 4]

Dit algoritme wordt het meest gebruikt voor gezichtsdetectie en in de meeste digitale camera's en telefoons. Figuur 2.4 toont een fragment uit een video van Adam Harvey, artiest en onderzoeker in computervisie-technologieën, en demonstreert het algoritme in werking. De rode rechthoek toont het gebied van de afbeelding dat wordt gescand om mogelijke gezichten te detecteren. Per kader worden verschillende rechthoeken gezocht om Haar features te detecteren, zoals de zwart-witte rechthoek die te zien is in de rode kader. Alle potentiële gezichten worden in een dunnere rode kader aangeduid. De aanwezigheid van een gezicht is nu nog niet bevestigd. Na het verwerken worden alle potentiële rechthoeken onderzocht op overlap. Meestal vormen twee of drie rechthoeken die overlappen één gezicht en potentiële rechthoeken die met niets overlappen, worden gemarkeerd als false-positive [3, 4].

2.2.3 Emoties

Om emotiedetectie in computervisie te begrijpen, wordt eerst uitgelegd hoe emotiedetectie bij de mens werkt. Er zijn verschillende visies op hoe een mens emoties interpreteert en of dit iets universeel is, onafhankelijk van wat er aangeleerd wordt in een cultuur. Charles Darwin zei in zijn boek *The Expression of the Emotions in Man and Animals* dat emoties voortkomen uit de evolutie en bij elke mens hetzelfde zijn [21]. Hierover twijfelen Rachael E. Jack et al. in hun paper waarin ze de visuele perceptie van mensen uit verschillende culturen reconstrueren op de zes basisemoties, namelijk blijheid, verdriet, boosheid, angst, afkeer (in de zin van walging, in het Engels *disgust*) en verrassing [22]. Er zijn dus verschillende visies op dit onderwerp maar wat hier belangrijk is, is dat alle voor- en tegenstanders het eens zijn dat er verschillende basisemoties kunnen onderscheiden worden. In deze masterproef worden de emoties boosheid, angst, afkeer, blijheid, verdriet en verrassing als de basisemoties gebruikt.

Ook emotiedetectie start met gezichtsdetectie. Ilias Maglogiannis et al. stellen een systeem voor dat gezichtsdetectie en emotieherkenning toepast en veronderstellen hierbij dat de ogen en mond de delen van het gezicht zijn die het meeste de emotie weergeven [23]. Hierbij is het dus noodzakelijk de mond en ogen te kunnen detecteren en lokaliseren hetgeen ze oplossen met een randdetectietechniek. De classificatie gebeurt volgens vijf emoties: blij, droevig, neutraal, verrast en boos. Voor elk van de emoties leggen ze drempels vast voor de posities van ogen en mond om te bepalen over welke emotie het gaat. Verder in deel 3.1 worden de verschillende gemeten emoties besproken die door verschillende gezichtsdetectiesoftware gedetecteerd wordt.

In Hoofdstuk 3 worden de verschillende services die gebruikt worden in deze applicatie vergeleken met elkaar. Ze hebben allemaal een verschillend aantal emoties dat gedetecteerd wordt. Bij sommige services wordt neutraal ook als emotie gerekend, maar eigenlijk kan dit eerder gezien worden als het afwezig zijn van enige andere emotie. In 3.1 wordt eenzelfde dataset als input gegeven voor de emotiedetectie van de drie services en wordt hun output vergeleken.

2.3 Bestaand werk

Het systeem gebruikt twee grote componenten: het aanbevelingssysteem en de gezichtsdetectie. De emoties en het aandachtsniveau bepalen de interesse van de kijker in de film en geven zo informatie aan het aanbevelingssysteem. Hieronder worden enkele bestaande systemen beschreven die gebruik maken van dezelfde principes.

2.3.1 Aanbevelingen op basis van emoties

Artikel [24] beschrijft een systeem dat multimedia aanraadt op basis van de gedetecteerde emoties van de mensen. De gemoedstoestand van de persoon wordt bepaald en op basis hiervan wordt bijvoorbeeld een vrolijk lied afgespeeld, indien de gebruiker een droevige gezichtsuitdrukking toont. Hiervoor moet de gebruiker op voorhand zijn voorkeuren instellen of hij liever naar vrolijke of droevige muziek luistert als hij droevig is. Het systeem kan verschillende soorten multimedia aanraden zoals muziek, films en andere video's. In het artikel wordt beschreven hoe gezichtsuitdrukkingen kunnen gedetecteerd worden door een kleine verandering in plaats en vorm van neus, lippen en ogen. De verandering wordt veroorzaakt door samentrekkingen van spieren in het gezicht van de persoon doordat de persoon een bepaalde emotie ervaart.

L. Pauly et al. beschrijven in [25] een aanbevelingssysteem dat feedback krijgt uit de emoties die de gebruikers tonen. Het systeem heeft als doel online producten aan te bevelen en het detecteert de gezichten van de online gebruikers die voor hun camera zitten. Het geslacht wordt gebruikt om aanbevelingen te genereren. De emoties van de gebruiker worden ook geregistreerd om impliciete feedback te verkrijgen over het aanbevolen product. Als de gebruiker een blijde emotie toonde, zal het systeem hetzelfde product aanraden in de toekomst. Met dit systeem worden producten op een meer gebruiksvriendelijke manier op de markt gebracht.

2.3.2 Aanbevelingen op basis van aandachtsniveau

De richting waar mensen naar staren kan gebruikt worden om te bepalen of de personen aandachtig zijn voor wat er gebeurt. Zo kan de interesse van personen die naar iets kijken gemeten worden op basis van de hoofdpositie. In het artikel van S. Clippingdale et al. nemen ze aan dat als een frontale hoofdpositie wordt waargenomen, de persoon naar het scherm kijkt en dus meer geïnteresseerd is [26]. In combinatie met de gezichtsuitdrukkingen wordt het interesseniveau bepaald en op basis van deze informatie worden gerichte advertenties aangeboden op de webpagina waar de gebruiker zit en wordt de gebruikersinterface gepersonaliseerd. In de context van televisie zitten er vaak meerdere kijkers voor het scherm en dus ook hier bestaat het probleem dat er geen volledig gepersonaliseerde gebruikersinterfaces bestaan voor televisies. Door elke kijker apart te identificeren, kan men voor elke kijker een apart profiel aanmaken om op die manier gepersonaliseerde content aan te bevelen.

Een andere manier om het visuele aandachtsniveau te meten, is met behulp van *gaze estimators* [27] die een inschatting maken van de mate waarin de gebruiker “staart” naar het scherm. Het systeem mapt de oog features en staarpunten met *Support Vector Machines* waardoor het staarpunt kan gevonden worden. In het artikel van R. Valenti et al. bespreken ze dat het inschatten van het staarpunt een combinatie is van twee factoren, namelijk de hoofdpositie en de locatie van de ogen [28]. Het vinden van de ooglocatie vereist vaak dat de hoofdpositie stabiel blijft en de hoofdpositie alleen slaagt er niet in om fijne veranderingen in staarpunt waar te nemen. In het artikel stellen ze dus voor om een systeem te ontwikkelen waar ze beide principes combineren en uit hun resultaten blijkt dat de nauwkeurigheid van de afzonderlijke systemen wordt verbeterd door de combinatie. Ook in artikel [29] bespreken ze de voordelen van een hybride methode die de hoofdpositie gebruikt en de staarvector met een geometrische aanpak berekent.

Indien de camera op een grotere afstand van de persoon staat, wordt het inschatten van de oogpositie al moeilijker. D. Johnson et al. tonen in hun paper aan dat enkel de *yaw*- en *pitch*-positie van het hoofd kunnen helpen bij het bepalen van de staarrichting zonder naar de ogen te kijken [30]. In het onderzoek van D. Johnson et al. vergelijken ze de gemeten hoofdpositie van testpersonen met de werkelijke staarpositie. Hieruit blijkt dat er lineaire relaties bestaan tussen de yaw en pitch van de hoofdpositie en de yaw en pitch van de staarrichting [30]. Met hun formules kan de staarrichting geschat worden met de data uit de gemeten hoofdpositie.

In de applicatie zal het aandachtsniveau bepaald worden door een service die dit kan meten maar ook door de inschatting van de yaw- en pitch-richting van het hoofd. Bijkomend wordt er ook gebruik gemaakt van detectie van gesloten ogen.

2.3.3 Aanbevelingssystemen

Aanbevelingssystemen gebruiken informatie van profielen om later een aanbeveling te genereren voor wat deze mensen zouden kunnen leuk vinden. Deze informatie bestaat uit de geschiedenis van de gebruikers: welke producten ze een goede rating geven, welke producten ze gekocht hebben of welke films ze besteld hebben. Indien een nieuwe gebruiker een aanbeveling wenst en er dus geen geschiedenis gekend is van deze gebruiker, ontstaat het *cold start* probleem waarover reeds verteld werd in 2.1. Hieronder worden enkele bestaande methoden beschreven om aanbevelingssystemen te maken.

In artikel [31] worden modellen op twee niveaus gegenereerd om aanbevelingen te maken. Het laagste niveau brengt specifieke gebruikers en hun eigenschappen en specifieke films met elkaar in verband om aanbevelingen te maken met een associatieve classificatiemethode. Hiervoor is er nood aan gebruikers die genoeg films geëvalueerd hebben, en dus genoeg geschiedenis hebben, en aan films die ooit al geëvalueerd geweest zijn. Op het hoogste niveau wordt het verband gemaakt op basis van het soort film en het soort gebruiker in plaats van op specifieke films en gebruikers. Deze methode kan gebruikt worden om aanbevelingen te maken voor producten die nog geen score gekregen hebben. Op dezelfde manier kunnen nieuwe gebruikers zonder geschiedenis aanbevelingen krijgen op basis van hun profiel dat bepaalt tot welk soort gebruiker ze behoren. Een aanbevelingssysteem kan dus op deze manier in alle gevallen een aanbeveling geven.

Le Hoang Son vergelijkt in het artikel [32] enkele oplossingsmethoden voor het cold start probleem. Verschillende onderzoeken worden vergeleken en worden in drie groepen onderverdeeld: methoden die aanvullende gegevensbronnen gebruiken, methoden die de groep van meest gelijkende mensen selecteert en methoden die de twee methoden combineren. Van elk van deze groepen worden specifieke algoritmen beschreven en hun efficiëntie werd onderzocht door ze toe te passen op enkele *Recommendation System*-datasets zoals MovieLens [33]. Uit de vergelijkingen van de paper blijkt dat het NHSM algoritme, dat wordt gebruikt voor de tweede groep van methoden, het meest accurate is en het minst computertijd vereist [34]. Dit NHSM algoritme maakt gebruik van een heuristische methode die gelijkenissen kan meten en is enkel gebaseerd op het gebruik van de gegevens over de ratings en niet van aanvullende gegevens zoals demografische gegevens.

2.3.4 Gelijkaardige applicaties in de actualiteit

De meest besproken streamingdienst van het moment is ongetwijfeld Netflix met zijn aanbevelingsalgoritme. Het algoritme haalt zijn informatie op zowel expliciete als impliciete feedback van gebruikers. De impliciete feedback in dit geval is bijvoorbeeld de frequentie waarin mensen naar een serie kijken en ander kijkgedrag. Voor het algoritme werd in 2006 een prijs - de Netflix Prize die uiteindelijk gewonnen werd in 2009 - aangekondigd voor het maken van het beste collaboratieve filtering algoritme om gebruikersratings voor films in te schatten met minstens 10 % grotere accuraatheid dan het toen gebruikte systeem van het bedrijf [35]. The Telegraph vertelt in een artikel dat Netflix gebruik maakt van duizenden soorten subgenres om de aanbeveling op te baseren [36]. De genres zijn gemaakt door een ingewikkeld algoritme dat data van het kijkgedrag van de gebruikers gebruikt zodat kan bepaald worden welk subgenre een gebruiker lijkt leuk te vinden. Enkel de hoofdgenres van films zijn dus niet genoeg in deze masterproef wordt er ook een fijnere onderverdeling gemaakt op basis van tags.

Het Netflixalgoritme heeft al bewezen goed te werken en goede aanbevelingen te geven aan zijn gebruikers. Één Netflixaccount kan gedeeld worden over meerdere gebruikers en Netflix zorgt voor aanbevelingen gebaseerd op het gebruikte gebruikersprofiel en bijhorend kijkgedrag. Om een correcte persoonlijke aanbeveling te krijgen, moet de gebruiker dus steeds op het correcte profiel zijn aangemeld. Dit kan een potentiële valkuil zijn in het aanbevelingsalgoritme aangezien er met één login en één wachtwoord toegang gegeven kan worden tot meerdere gebruikersprofielen. Indien er op een profiel bijvoorbeeld ooit een kind aanmeldt en naar kleutertekenfilms kijkt op profiel van persoon A, houdt het algoritme hier rekening mee bij de volgende aanbeveling van persoon A. Hierdoor kan het zijn dat persoon A plots allemaal kinderfilms krijgt aangeraden in plaats van zijn normale series. De app die in deze masterproef wordt voorgesteld, zou een oplossing voor dit probleem kunnen bieden omdat gebruikers nooit foutief kunnen worden aangemeld dankzij de gezichtsherkenning. De aanbeveling van een gebruiker zal enkel gebaseerd worden op films waar hij echt zelf naar gekeken heeft.

Hoofdstuk 3

Evaluatie services

De applicatie gebruikt bestaande machine learning technieken van verschillende platformen die de gezichtsdetectieservice aanbieden. Alvorens met deze bestaande services gewerkt kan worden, wordt er eerst onderzoek gedaan naar welke service het best voor welke aspect van de gezichtsdetectie wordt ingeschakeld. De drie services worden onderworpen aan verschillende testen en de resultaten worden met elkaar vergeleken in dit hoofdstuk. De testen werden uitgevoerd via de app en hoe dit opnieuw gedaan kan worden, wordt beschreven in [Bijlage B: Gebruikershandleiding](#).

3.1 Gezichtsdetectieservices: een eerste vergelijking

Bij het maken van de applicatie werd gekozen om met drie verschillende gezichtsdetectieservices te werken. Hiervoor zijn drie redenen. Ten eerste om de verschillen in de gebruikte software te onderzoeken, wordt er onderzocht welke van de drie services de beste resultaten geven voor verschillende onderdelen van de gezichtsdetectie. Ten tweede om hun verschillende functionaliteiten door elkaar in de applicatie te kunnen gebruiken. De drie services kunnen allemaal gezichten detecteren maar elk van de drie heeft zijn eigen extra featuredetecties die interessant kunnen zijn voor deze masterproef. Microsoft Face API kan bijvoorbeeld ook bedekte ogen, mond en voorhoofd detecteren, wat Kairos en Face++ niet kunnen. Face++ kan zeggen of de ogen gesloten of open zijn en Kairos kan extra informatie voorzien over het aandachtsniveau. De derde reden waarom er met drie services gewerkt wordt, is zodat ze elkaars nadelen en beperkingen kunnen opvangen, zoals prijs en requestlimieten. De verschillende services zijn namelijk niet volledig gratis en er moet rekening gehouden worden met de beperkingen van een gratis account.

Bij het inschatten van leeftijd en geslacht van nieuwe personen worden de services ook geaggregeerd gebruikt om tot een gemiddelde te komen van de drie resultaten. De geschatte leeftijd en geslacht vertrouwen dus op het resultaat van niet één maar van meerdere services waardoor de geschatte waarden meer kans hebben aan te leunen bij de realiteit. Ook is het handig om een oneven aantal services te hebben voor de *voting* over het geslacht. Uiteindelijk werd slechts één service gekozen voor de inschatting van de leeftijd en de reden hiervoor wordt hieronder verder verklaard.

Hieronder wordt een kort overzicht gegeven per service, elk met zijn eigen extra features en functionaliteit. Er werden ook nog andere gezichtsdetectieservices overwogen zoals Vision API van Google. Deze werden niet gekozen voor deze masterproef omdat ze sommige onderdelen niet bevatten zoals bijvoorbeeld gezichtsherkenning (het identificeren van gezichten) of geen aanvullende informatie kunnen bieden. Een overzicht in tabelvorm van de drie services en wat ze aanbieden is te vinden in [3.1.4](#).

3.1.1 Microsoft Cognitive Services

Microsoft Azure is een platform van Microsoft dat verschillende services aanbiedt, waaronder ook *Cognitive Services* om API toepassingen toe te voegen aan programma's om "contextuele interacties mogelijk te maken" [5]. Microsoft voorziet verschillende API's voor onder andere tekstanalyse, computervisie en gezichtsdetectie. Deze laatste heet Face API en hiermee is het mogelijk om gezichten te detecteren, te identificeren (gezichtsherkenning), te analyseren, te organiseren en te taggen. Vroeger heette dit 'Project

Oxford’ en onder deze naam is er ook veel informatie over te vinden. Doorheen de masterproef en de code worden zowel de benaming Microsoft als Face API voor deze service van Microsoft gebruikt.

Microsoft biedt twee soorten gratis accounts aan. Met de eerste, de *free Face API* kunnen 30000 aanvragen per maand doorgestuurd worden aan een maximale frequentie van 20 transacties per minuut. Er kunnen tot 1000 afbeeldingen opgeslagen worden in de *Face Storage*, een persoonlijke opslagplaats waar gezichten afgebeeld worden op personen om zo gezichten te kunnen identificeren. Het tweede is een free trial Azure account waarmee er toegang verleend wordt tot het gebruiken van de *standard Face API* waarmee een maximale frequentie van 10 transacties per seconde beschikbaar is [5]. Beide gratis accounts hebben een maximale gebruikperiode van 7 dagen zonder kredietkaart voor de free Face API en 30 dagen voor de free trial met kredietkaartnummer. Na de free trial periode bedraagt de kostprijs 0,844 euro voor 1000 transacties en 0,211 euro per 1000 opgeslagen afbeeldingen per maand bij de standard Face API [5] of kan er met een nieuw e-mailadres en kredietkaartnummer een nieuwe trial gestart worden. Aangezien Microsoft op deze manier de enige service is waarmee gratis een hoog aantal transacties beschikbaar is en ook de beste resultaten haalt op de evaluatietesten die hier verder beschreven worden, wordt deze service in de applicatie gekozen als basis voor gezichtsdetectie en -herkenning.

Elke foto die geanalyseerd wordt, wordt eerst doorgestuurd in een request naar de Microsoft Face API en hierbij worden alle gezichten gedetecteerd. De response vertelt waar de aanwezige gezichten zich bevinden en voorziet de coördinaten voor een kader rond de gezichten, zoals te zien in Figuur 3.1. Microsoft identificeert dan ook meteen van wie welk gezicht is. In elke stap is het vereist dat de identificatie gebeurt omdat het belangrijk is te weten van wie de bijhorende emoties en aandachtsniveau zijn. De identificatie gebeurt door het te identificeren gezicht te gaan vergelijken met elk gezicht uit de *Person Group* van de persoonlijke Face Storage. Zo een Person Group bevat een verzamelingen van opgeslagen gezichten per geregistreerde persoon. De gelijkenis met het te detecteren gezicht en de gezichten in de Person Group worden berekend en de kandidaten voor wie de persoon kan zijn, worden gesorteerd teruggegeven in mate van gelijkenis. Een Person Group moet aangemaakt worden met een group-ID bij het initiële gebruik. Telkens nieuwe gezichten en nieuwe personen aan de groep worden toegevoegd, moet de groep getraind worden zodat het later de identificatie kan uitvoeren [37]. Het trainen gebeurt door het group-ID mee te geven aan de Face Storage.

De afzonderlijke, geïdentificeerde gezichten kunnen dan doorgestuurd worden voor verdere verwerking, bijvoorbeeld naar Face++ voor detectie van de gesloten ogen en naar Kairos voor detectie van het aandachtsniveau. Er gebeuren twee transacties per foto, gezichtsdetectie en gezichtsherkenning, en wanneer de app elke twee seconden een foto neemt, vinden er 60 transacties per minuut plaats. Bijgevolg wordt het aangeraden om de app te gebruiken met een standard Face API versie om te voldoen aan de vereiste limiet. De app kan ook gebruikt worden aan 20 transacties per minuut met een free versie alhoewel er dan een “rate limit exceeded”-error zal verschijnen bij het overschrijden van de limiet. Een oplossing hiervoor is om slechts filmfragmenten van 20 seconden te voorzien of de frequentie van foto’s verlagen naar elke 6 seconden. Indien de gratis service toch gebruikt wordt, kan men een “exceeded rate limit”-waarschuwing krijgen indien de limiet bereikt werd. Deze waarschuwing kan ook verkregen worden bij het uitvoeren van de emotietesten uit dit hoofdstuk aangezien de limiet van de standard Face API nog steeds “maar” 10 transacties per seconde is. Bijgevolg is de limiet steeds iets waar rekening mee moet worden gehouden.



Figuur 3.1: Het json antwoord van Microsoft Face API als voorbeeld gegeven op hun site [5]

Microsoft Face API detecteert acht emoties (boosheid, minachting, afkeer, angst, blijheid, neutraal, verdriet en verrassing) en hun probabiliteit, weergegeven in een json formaat zoals te zien in Figuur 3.1. Geslacht en leeftijd worden ook geschat. Face API detecteert ook of mond, voorhoofd en mond bedekt zijn. Dit is te zien in het json antwoord onder *occlusion* in Figuur 3.1. Het aantal landmarks of features in het gezicht dat gedetecteerd wordt is 27, waaronder pupil-, wenkbrauw- en lipposities [5].

3.1.2 Face++

Face++ Cognitive Services is een platform van Megvii Technology Inc dat ook computervisietechnologieën voorziet: het omvat afbeeldingsherkenning, Optical Character Recognition (OCR), lichaamsherkenning en gezichtsherkenning [1]. Face++ kan 83 of 106 landmarks van een gezicht weergeven, afhankelijk van de wens van de gebruiker. Een illustratie van de site van Face++ is te zien in Figuur 2.2 uit Hoofdstuk 2 waar 106 landmarks aangeduid zijn op een gezicht. Ook geslacht en leeftijd worden geschat en de probabiliteit van de emoties verdriet, afkeer, boosheid, verrassing, angst, blijheid en neutraal worden bepaald. Face++ heeft nog een bijkomende detectiefeature die voor beide ogen van het gezicht bepaalt of het oog open of dicht is en of de persoon al dan niet een bril draagt. Het gratis account is beperkt tot één request per seconde en er is geen gratis trial periode beschikbaar waarin de volledige capaciteit kan getest worden. Het betalende account bedraagt ongeveer 0,0017 euro per request, wat goedkoper is dan Microsoft. Toch wordt deze service niet gebruikt voor leeftijd- en emotiedetectie omdat onderstaande evaluatie uitwees dat Face++ hiervoor slechter scoorde dan de andere services.

Sinds 6 juni 2018 werd opgemerkt dat Face++ op zijn site aankondigt dat het tijdelijk geen service meer voorziet aan klanten in Europa [1]. Gedurende de volledige ontwikkeling van de applicatie, de evaluatietesten en de testen met testpersonen kon er gebruik gemaakt worden van de service. Alle resultaten van de testen zijn terug te vinden op de bijgeleverde USB-stick. De service kan elk moment weer actief worden, maar in de tussentijd kan de app er geen gebruik van maken. Dit heeft geen grote gevolgen voor het verzamelen van de impliciete feedback. De app houdt er rekening mee dat services kunnen falen en vult vervolgens de resultaten van Face++ niet in en vertrouwt voor de aandachtsdetectie op Kairos. De evaluatietesten kunnen niet door Face++ worden uitgevoerd, maar de service heeft in het verleden bewezen te werken en de resultaten hiervan werden opgeslagen.

3.1.3 Kairos

Kairos is een bedrijf dat zich bezig houdt met kunstmatige intelligentie en dat gespecialiseerd is in gezichtsherkenning [38]. Kairos kan gezichten identificeren, emoties herkennen en demografische gegevens zoals leeftijd en geslacht schatten, zowel in afbeeldingen als in video's. De emoties die Kairos detecteert zijn blijheid, boosheid, afkeer, verdriet, angst en verrassing. De extra feature die interessant is voor deze masterproef is de *Attention Measurement*. Dit berekent de totale aandachtstijd, het aantal blikken en aantal keer dat de persoon geknipperd heeft met de ogen. Het aantal features dat gedetecteerd wordt is 68. Met een gratis account kunnen er maximum 25 transacties per minuut gebruikt worden met een limiet van 1500 per dag. Het goedkoopste maandabonnement bedraagt ongeveer 425 euro per maand en voorziet maximum 75 transacties per minuut met een limiet van 50000 per dag [38]. Wegens de beperkingen van het gratis account kunnen er geen video's, maar enkel foto's geanalyseerd worden.

3.1.4 Overzicht

In onderstaande tabel, Tabel 3.1, staan de drie gebruikte services naast elkaar met hun eigenschappen. Face++ voorziet, net zoals Kairos, een soort aandachtsniveaumeting in de vorm van *eye gaze estimation*. Dit schat in waar de ogen naar kijken en wordt dus ook vermeld in de tabel. Aangezien Kairos meteen het aandachtsniveau, gebaseerd op de ogen en hoofdpositie, in een waarde van 0 tot 1 teruggeeft, wordt Kairos hiervoor gebruikt waardoor geen extra berekeningen vereist zijn. De gedetecteerde emoties zijn bij de drie services steeds de basisemoties, eventueel uitgebreid met nog extra emoties. Bij Microsoft Face API en Face++ zit neutraal er ook bij, maar zoals eerder gezegd, is dit eerder een bevestiging van het afwezig zijn van de andere emoties. Minachting wordt enkel door Microsoft gedetecteerd maar zal niet gebruikt worden in de applicatie aangezien het niet als universele basisemotie wordt beschouwd.

Tabel 3.1: Vergelijking van de drie gebruikte gezichtsdetectieservices

	Microsoft Face API	Face++	Kairos
Gezichtsherkenning	✓	✓	✓
Geslacht en leeftijd	✓	✓	✓
Bedekte gezichtsdelen	✓	✗	✗
Gesloten ogen	✗	✓	✗
Aandachtsniveau	✗	✓	✓
Emotiedetectie	Basisemoties + minachting en neutraal	Basisemoties + neutraal	Basisemoties
Aantal gedetecteerde features	27	83 of 106	68
Beperkingen gratis account	30 000 req/maand 20 req/minuut	1 req/seconde	1500 req/dag 25 req/minuut

Voor de inschatting van de hoofdpositie wordt Microsoft niet gebruikt aangezien dit de enige is die de pitch-draairichting van het hoofd niet detecteert. Kairos en Face++ detecteren wel de drie richtingen, waarvan pitch en yaw de twee relevante zijn voor het meten van het aandachtsniveau. In de applicatie zal gekozen worden om Face++ als hoofdpositiedetectie en detectie van gesloten ogen te gebruiken. Microsoft zal worden gebruikt voor de inschatting van de leeftijd en Kairos voor de inschatting van het aandachtsniveau, aangevuld met de hoofdpositie en de detectie van gesloten ogen.

De waarden die de emotiedetectie teruggeeft, zijn steeds de probabiliteiten dat het over deze emotie gaat. Microsoft en Face++ verdelen een probabiliteit van 100 % over al hun detecteerbare emoties, inclusief de neutrale emotie. Hierdoor is de som van alle emotieprobabiliteiten steeds gelijk aan 100 % en zijn op die manier de waarschijnlijkheden genormaliseerd. Indien een persoon geen emoties toont, is de probabiliteit van de emotie neutraal gelijk aan 100 %. Kairos doet deze normalisatie niet aangezien het geen neutrale emoties detecteert en de som dus niet steeds kan gelijk zijn aan 100 %. Indien de evaluatie van de emotiedetectie uitgevoerd wordt, moet er rekening worden gehouden met een eventuele extra normalisatie.

3.1.5 Hoe de services gebruikt worden

Voor alle drie de services is het zo dat een detectie-aanvraag steeds even moet wachten op een antwoord na het opsturen van de foto. Om te voorkomen dat de applicatie vastloopt, worden asynchrone taken en threads gebruikt om de hoofdthread niet te overbelasten. Een aanvraag voor de detectie bij Microsoft wordt in de code steeds verstuurd via asynchrone taken en de code hiervoor werd gehaald van de Github-pagina van Microsoft [39]. De code roept de methoden op van FaceServiceClient, een klasse voorzien door Face API en doet al het werk voor het verzenden van de aanvragen en de service over het internet [40]. De operaties van FaceServiceClient zijn *long running operations*, operaties die lang duren, dus deze moeten uitgevoerd worden in een worker thread. Een dergelijke worker thread wordt het beste aangemaakt met behulp van de *AsyncTask*-klasse (door een nieuwe klasse van deze AsyncTask te laten afleiden) en door de *doInBackground* methode te implementeren met de methode van FaceServiceClient. Indien dit niet gedaan wordt, krijgt de applicatie een *NetworkOnMainThreadException* [41]. De AsyncTask-klasse werd gebruikt omdat deze aangeraden wordt wanneer het antwoord op een service even duurt en de applicatie kan laten vastlopen. Hierdoor worden de zware taken in de achtergrond uitgevoerd. AsyncTasks worden best gebruikt voor korte operaties van maximaal enkele seconden [41]. Ze zijn dus geschikt voor gezichts-detectie en -herkenning aangezien deze, indien de foto verkleind wordt, maximaal een seconde duren.

Ook de detectie van Face++ werd verwerkt in een asynchrone taak en voor het versturen wordt er gebruik gemaakt van een bibliotheek van Face++ [42] waarover meer uitleg vermeld staat in [Bijlage A: Installatiehandleiding](#). Voor de detectie van Kairos wordt code gebruikt onder een Gnu License [43] en

wordt de Threadklasse van Java gebruikt. Zoals vermeld, worden de foto's verkleind opgestuurd zodat het verzenden van de foto en het krijgen van het antwoord sneller verloopt. Er moet een afweging worden gemaakt tussen performantie en de nauwkeurigheid van de detectie. De foto mag niet te veel verkleind worden anders zou de detectieservice geen goede detectie meer uitvoeren.

3.2 Testen voor leeftijd en geslacht

De drie services werden alle drie vergeleken op de mate waarin ze leeftijd en geslacht goed kunnen schatten. Ze werden toegepast op zowel een databank met foto's genomen in een studio als op foto's genomen met de camera van een smartphone. De studiofoto's werden gebruikt uit de Chicago Face Database [44] en bevatten gezichten van mensen met leeftijd van 16 tot 56 jaar en van verschillend geslacht en etniciteit. De databank voorziet een Excelbestand met per foto het bijhorende geslacht, de etniciteit en de geschatte leeftijd. De leeftijd is niet de exacte leeftijd van de persoon maar is geschat door verschillende mensen. Het testen op deze databank kan dus het verschil in leeftijd tonen tussen wat een computer schat en wat de gemiddelde menselijke inschatting is. Het nadeel van deze databank is dat de foto's met ideale belichting genomen zijn en dus niet representatief voor het doel van deze masterproef. De inschatting van leeftijd en geslacht werd op 128 foto's uit de Chicago Face Database toegepast, waarvan de helft vrouwen en de helft mannen en vier verschillende etniciteiten.

In de context van de masterproef zitten de mensen in een woonkamer of andere ruimte in het huis waar de belichting niet ideaal is en worden de foto's genomen met de camera van de smartphone. De camera van een smartphone heeft niet dezelfde goede kwaliteit van een camera waarmee studiofoto's getrokken worden. De frontcamera van een smartphone heeft vaak zelfs nog minder megapixels dan de camera op de achterkant. Hierdoor kunnen leeftijd en geslacht verkeerd worden ingeschat. Daarom is er gekozen om nog een tweede, zelfgemaakte dataset te gebruiken met foto's van mensen in realistische omgevingen en niet-ideale belichting. Het is namelijk belangrijk ook de prestaties van de services te kennen bij alledaagse en niet-ideale foto's. Wanneer de uiteindelijke applicatie zal getest worden, zullen de testpersonen ook niet in een perfect belichte studio zitten en telkens poseren voor goede foto's. Alle foto's werden genomen met de camera van een smartphone zonder de flits te gebruiken en zonder de belichting in de ruimte aan te passen. Deze dataset bevat foto's van 46 gezichten, zowel mannen (24) als vrouwen (22) en met een leeftijd variërend van 0 tot 66 jaar. Bij elk gezicht zijn de exacte leeftijd en geslacht bekend.

Tabel 3.2 toont de resultaten van de twee datasets voor de geschatte leeftijd en geschatte geslacht. Voor de geschatte leeftijd wordt per schatting het verschil genomen met de echte leeftijd. Deze verschillen worden omgezet naar hun absolute waarde zodat ze de grootte van de fout aanduiden. Het gemiddelde van alle verschillen wordt berekend en in de tabel getoond. Voor het geschatte geslacht passen de drie services voting toe. Ze stemmen elk op het geslacht ze vermoeden dat het is en het geslacht met de meeste stemmen bepaalt het uiteindelijke geslacht. Aangezien er drie services gebruikt worden bij de bepaling van het geslacht, is er steeds één van de twee geslachten in de meerderheid met het aantal stemmen. De resultaten van deze voting worden ook in de tabel getoond aan de hand van het percentage fouten.

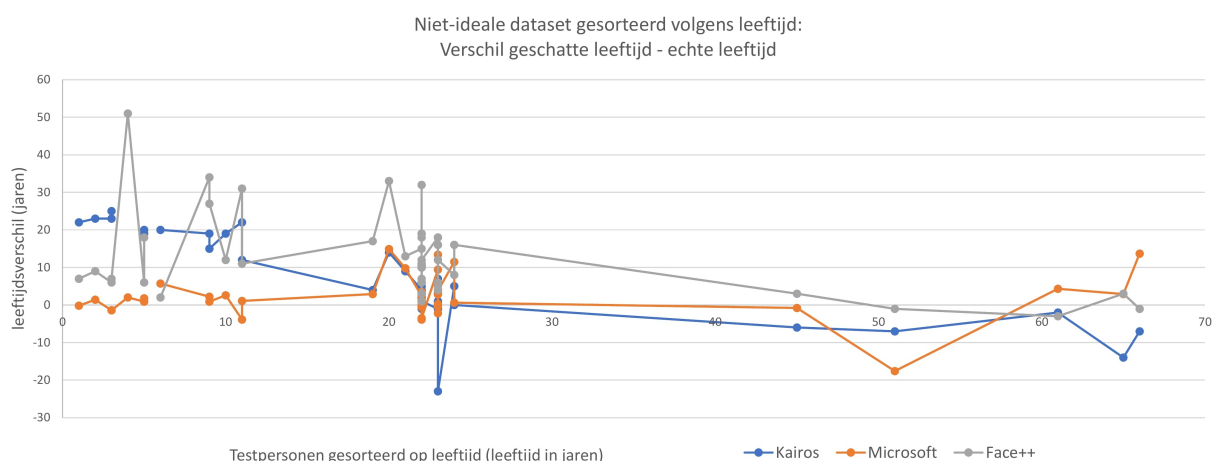
De tabel toont dat, zoals verwacht, de geschatte leeftijden en geslachten grotere fouten tonen bij de niet-ideale smartphone foto's. Zowel de gemiddelde fout op de leeftijd als het percentage foute geslachten is telkens hoger dan de resultaten van de ideale foto's. Ook het aantal foto's waarin de services geen gezicht detecteerden ligt hoger. Dit is te verwachten aangezien de smartphone foto's geen ideale belichting en lage kwaliteit hebben. De tabel toont ook dat Face++ telkens per dataset het slechtst de leeftijden inschat met een gemiddelde van de absolute verschillen van 8,26 en 13,14 jaar voor respectievelijk de ideale dataset en de niet-ideale dataset. Als de gemiddelde leeftijd genomen wordt door de drie services samen te nemen, komt het gemiddelde van al deze resultaten uit op 4,28 en 7,91 jaar. Bij de ideale dataset wordt het gemiddelde dus slechter dan de schatting van leeftijd bij Kairos en Microsoft apart. Bij de inschatting van geslacht haalt Face++ wel betere resultaten.

Tabel 3.2: De resultaten van de leeftijds- en geslachtstesten voor de drie services toegepast op de twee datasets

	Chicago Face Database [44] (ideale omgeving)	Smartphone foto's (niet-ideale omgeving)
Geschatte leeftijd (in jaren)		
Gemiddelde absolute verschil met Microsoft	3,56	4,31
Gemiddelde absolute verschil met Kairos	3,29	8,88
Gemiddelde absolute verschil met Face++	8,26	13,14
Gemiddelde absolute verschil met gemiddelde	4,28	7,91
Geschatte geslacht (in % fout)		
Percentage fout bij Microsoft	7,81 %	15,22 %
Percentage fout bij Kairos	7,03 %	10,87 %
Percentage fout bij Face++	4,69 %	13,04 %
Percentage fout bij voting	5,47 %	10,87 %
Niet gedetecteerde gezichten (in %)		
Microsoft	0,00 %	4,35 %
Kairos	0,00 %	8,70 %
Face++	0,00 %	4,35 %

Indien het resultaat van Face++ niet wordt meegeteld bij het berekenen van de gemiddelde geschatte leeftijd, komt de fout op 3,61 jaar voor de Chicago dataset en 6,05 jaar voor de smartphone dataset, wat een beter resultaat geeft dan het gemiddelde van de drie services en dit voor beide datasets. Er moet dus overwogen worden of Face++ geschikt is om te laten meetellen voor de berekening van de gemiddelde leeftijd. Aangezien de drie services allemaal goede resultaten geven voor inschatting van het geslacht, worden de drie services wel geaggregeerd voor de bepaling van het geslacht en zullen ze met behulp van voting het uiteindelijke geslacht bepalen in de applicatie.

Bij de niet-ideale dataset verduidelijkt Figuur 3.2 het probleem van de leeftijdsinschatting bij mensen onder 20 jaar. De resultaten zijn bekomen uit de geschatte leeftijd van de dataset met smartphone-foto's aangezien deze dataset de meeste variatie had in leeftijden. Deze figuur toont de verschillende testpersonen gesorteerd op hun leeftijd op de x-as. De y-as duidt het verschil aan tussen de geschatte leeftijd en de echte leeftijd van de persoon. Elke stip duidt een persoon aan en per persoon zijn er drie geschatte leeftijden, één voor elke service. Indien een stip onder nul van de y-as ligt, werd de leeftijd van de persoon te jong geschat, en indien de stip erboven ligt, werd de persoon zoveel jaren te oud geschat.

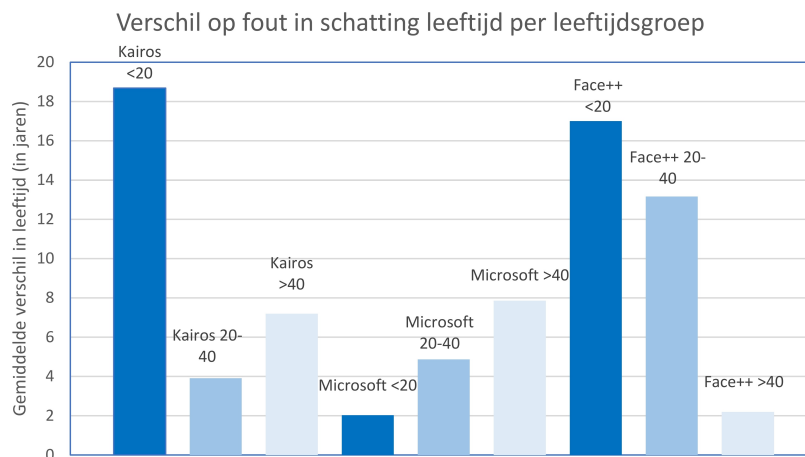


Figuur 3.2: Vergelijking van de leeftijdsinschatting van de drie services toegepast op de niet-ideale dataset gesorteerd op leeftijd

Zowel Kairos als Face++ schatten de personen tussen 0 en 10 jaar vaak ongeveer 20 jaar te oud. Dit is een groot probleem aangezien het vooral voor jonge mensen belangrijk is een goede inschatting van de leeftijd te krijgen zodat er kan nagegaan worden of ze voldoen aan de minimumleeftijd van de aanbevolen

films. Face++ gaat zelfs eens helemaal de mist in door een persoon van 4 jaar meer dan 50 jaar te oud te schatten. Kairos doet het bij deze leeftijdsgroep ook niet goed aangezien het continu jonge kinderen ongeveer 20 jaar te oud schat. Microsoft blijft bij deze leeftijdsgroep steeds rond de 0 jaar verschil schatten, waaruit blijkt dat Microsoft ideaal is voor het schatten van leeftijd bij kinderen. Ook voor mensen tussen de 20 en 40 jaar blijft Microsoft stabiel de leeftijden inschatten terwijl Face++ en Kairos grote fouten blijven maken.

Dezelfde dataset wordt hieronder in Figuur 3.3 voorgesteld in een histogram met per service de gemiddelde absolute fout in jaren op de leeftijd per leeftijdsgroep. Er werd gekozen voor drie leeftijdsgroepen (mensen jonger dan 20, tussen 20 en 40 jaar en ouder dan 40) zodat de dataset van 46 gezichten ongeveer in drie werd verdeeld. Hoe hoger de balk in het histogram, hoe slechter de inschatting bij die leeftijdsgroep.



Figuur 3.3: Vergelijking van de services per leeftijdsgroep

Zoals opnieuw te zien is in het histogram, scoren Kairos en Face++ heel slecht bij gezichten van mensen jonger dan 20, terwijl Microsoft dan goede resultaten behaalt. Face++ is dan weer beter in het schatten van leeftijd van mensen ouder dan 40. Dit is ook wat Face++ heel vaak doet, namelijk mensen oud inschatten. In het geval van mensen boven de 40 jaar, geeft dit goede resultaten. Maar wanneer Face++ mensen onder de 40 schat, valt de geschatte waarde ook vaak boven hun echte leeftijd. Er wordt bijgevolg gekozen om enkel Microsoft te gebruiken voor de inschatting van de leeftijd en de drie services enkel te aggregeren voor het schatten van het geslacht om een overeenkomst te krijgen.

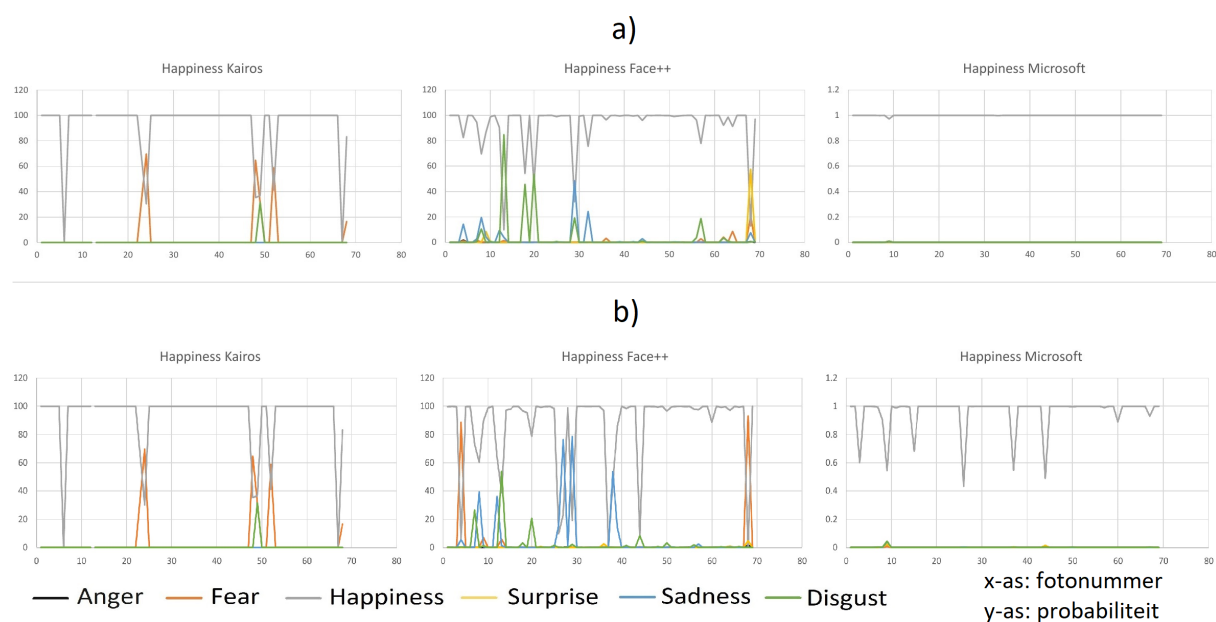
3.3 Emotietesten

De drie services werden getest voor emotiedetectie op de dataset van Cohn Kanade [45, 46]. Dit is een dataset met verschillende foto's van mensen die starten met een neutrale emotie te tonen en dan overgaan tot een uitgesproken emotie zoals lachen, verdriet, boosheid ... De dataset deelt foto's van verschillende personen op per emotie. Zo krijgt elke persoon meerdere reeksen (één reeks is één emotie) en meerdere foto's per emotiereeks (foto's die van neutraal overgaan tot de expressieve emotie). De testen werden telkens uitgevoerd op alle reeksen van eenzelfde emotie. Zo kan er per emotie worden nagegaan hoe goed de drie services die emotie detecteren. Een overzicht van alle grafieken is achteraan te vinden bij [Bijlage C: Grafieken, diagrammen, code en tabellen](#). De emotie die bovenaan de grafiek staat, is degene die de services zouden moeten detecteren, liefst met een probabilliteit van 100 % terwijl de andere emoties een zo laag mogelijke probabilliteit zouden moeten hebben. De figuren tonen per service en per emotie een grafiek met op de x-as het fotonummer en op de y-as de probabilliteit voor elke aanwezige emotie. Microsoft geeft steeds de probabilliteit op een maximumwaarde van 1 terwijl Face++ en Kairos dit voorstellen op een waarde van 100. Bovendien werden de waarden van Kairos steeds genormaliseerd alvorens ze voor te stellen in een grafiek. Dit wilt zeggen dat de som van alle emotiewaarden, net zoals Face++ en Microsoft, gelijk zal zijn aan 100 % tenzij er geen enkele emotie detecteerbaar is (want dan wordt neutraal, hier niet afgebeeld, gelijk aan 100 %). De waarden van Face++ en Microsoft zijn automatisch reeds genormaliseerd in hun detectie-antwoord. De resultaten van de testen zijn in de gegenereerde CSV-bestanden

verzameld in volgende volgorde: boosheid, angst, blijheid, verrassing, verdriet en afkeer, wat afwijkt van de alfabetische Engelstalige volgorde in de rest van de applicatie en code.

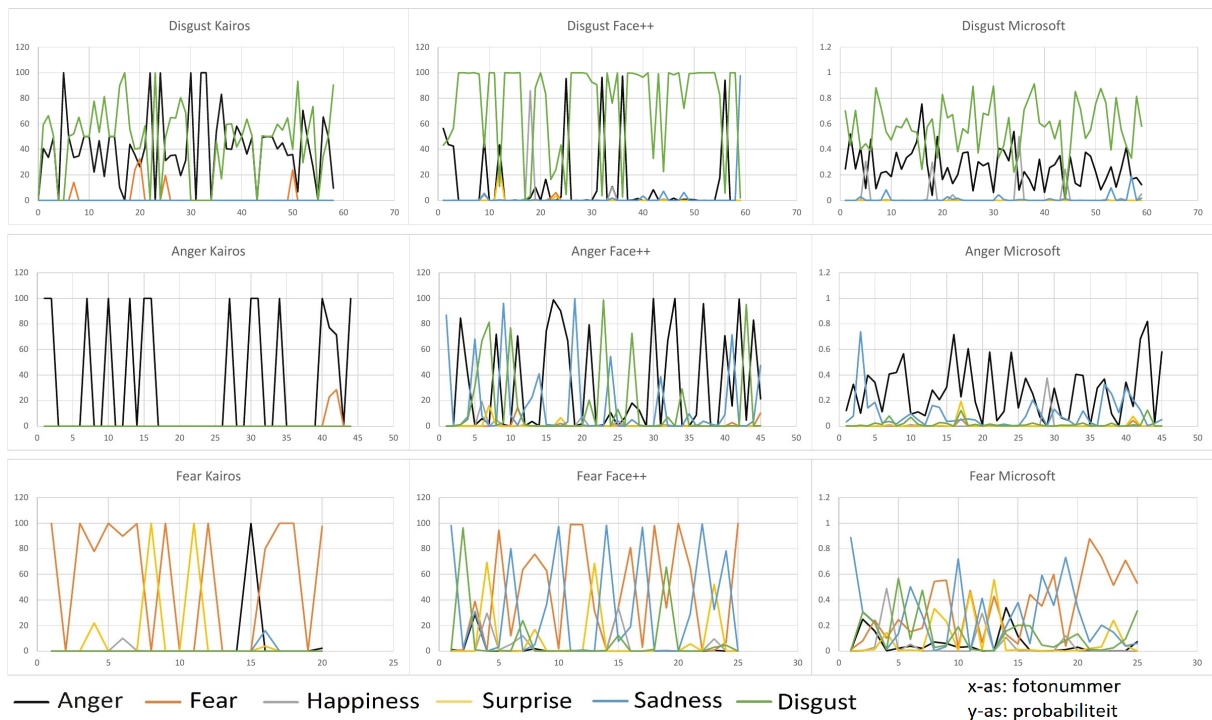
De emotiedetectietesten werden voor de zes emoties en voor de drie services uitgevoerd. Een eerste test werd toegepast op telkens de laatste foto uit een fotoreeks van de Cohn Kanade dataset. De laatste foto van zo een reeks bevat steeds de meest uitgesproken emotie. Bij deze testen zou de emotie heel duidelijk aanwezig moeten zijn en kunnen de services vergeleken worden op hoe goed ze heel aanwezige emoties detecteren. Een tweede test nam steeds de middelste foto, met andere woorden een foto waarop de persoon een gezichtsuitdrukking toont tussen neutraal en de meest uitgesproken versie ervan. Bij deze test zou het wat moeilijker moeten zijn om de juiste emotie te detecteren. Elk van deze twee testen beschouwt dus één foto uit elke emotiereeks. De foto's van reeksen die dezelfde emotie voorstellen worden samen gezet en krijgen een volgnummer. Het volgnummer is het nummer aangeduid op de x-as en op deze manier kunnen de uitkomsten van de verschillende services vergeleken worden per emotie en per foto.

Figuur 3.4 vergelijkt het verschil in resultaat voor de drie services voor de emotie blijheid bij zowel de detectie op de laatste foto als de middelste. De grafieken zijn afkomstig van het grote emotiegrafieken-overzicht achteraan in [Bijlage C: Grafieken, diagrammen, code en tabellen](#). De figuur toont dat bij de emotie blijheid, zowel bij de eerste als bij de tweede test, Microsoft steeds enkel blijheid detecteert en geen andere emotie. Kairos en Face++ detecteren hier en daar een andere emotie zoals soms eens verdriet of angst, zelfs bij de makkelijkere eerste test waarbij de personen een uitgesproken emotie vertonen. De makkelijkst detecteerbare impliciete feedback van mensen die een film kijken is blijheid aangezien mensen eerder lachen als ze de film leuk vinden. Het is dus belangrijk dat er zeker voor deze emotie de best detecterende service gekozen wordt en in dit geval is dit Microsoft.



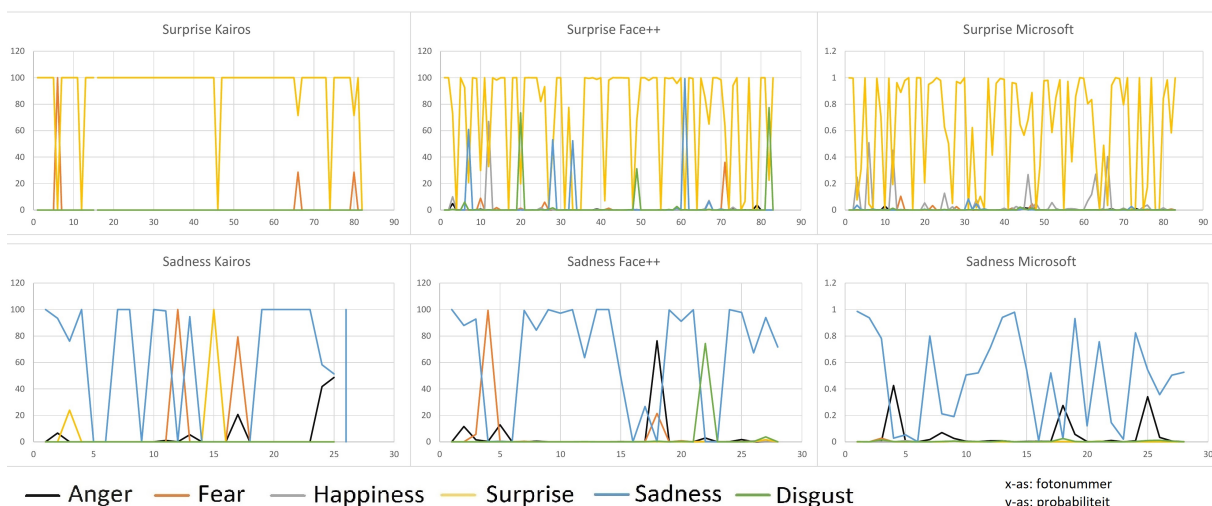
Figuur 3.4: Vergelijking tussen de drie services en de emotiedetectie van blijheid op de laatste (a) en middelste (b) foto uit elke reeks

Indien enkel naar de eerste test met de laatste foto uit de reeks gekeken wordt, wordt duidelijk dat elke service het moeilijk heeft met de detectie van de emoties afkeer, boosheid en angst. Dit wordt duidelijk geïllustreerd in Figuur 3.5. De emotie afkeer (disgust) wordt vaak verward met boosheid (anger). Bij Microsoft zit de probabiteit van afkeer steeds boven die van boosheid waardoor kan besloten worden dat Microsoft opnieuw een betere emotiedetectie uitvoert voor deze emotie. Ook bij boosheid detecteert Microsoft zo goed als continu de emotie met de meeste probabiteit terwijl Kairos soms geen emotie detecteert en bij Face++ de probabiteit van andere emoties soms groter is. De waarschijnlijkheden van Microsoft zijn dan wel kleiner dan die van de andere services maar de overheersende emotie besluit dat het over boosheid gaat, waardoor opnieuw blijkt dat Microsoft betere resultaten geeft.



Figuur 3.5: Vergelijking tussen de drie services voor de emotiedetectie van afkeer, boosheid en angst, toegepast op de laatste foto uit de reeks

Angst blijkt de moeilijkste te detecteren emotie te zijn aangezien alle drie de services de emotie vaak verwarren met verdriet, afkeer en verrassing, zelfs voor de meest expressieve foto uit elke reeks. De emoties verrassing en verdriet worden door de drie services vrij goed bepaald bij de laatste foto uit elke reeks, blijkt uit de grafieken uit Figuur C.4 achteraan in [Bijlage C: Grafieken, diagrammen, code en tabellen](#). Wanneer er gekeken wordt naar de middelste foto uit elke reeks, waarbij de emoties moeilijker te detecteren worden omdat ze minder expressief zijn, vertoont de detectie van de services een grafiek zoals in Figuur 3.6. Kairos detecteert nog steeds het best de emotie verrassing, maar opnieuw is het Microsoft met de probabiliteit van verdriet die steeds overheerst bij de verwachte emotie van verdriet.



Figuur 3.6: Vergelijking tussen de drie services voor de emotiedetectie van verrassing en verdriet, toegepast op de middelste foto uit de reeks

De expressies in een reeks foto's evolueren van neutraal naar de volledig uitgesproken emotie. Het is ook interessant eens te kijken hoe de services deze evolutie detecteren. Figuur C.5 in [Bijlage C: Grafieken](#),

[diagrammen, code en tabellen](#) toont de grafieken voor de verschillende services. Telkens werd er voor elk van de zes emoties één reeks gekozen waarop de gezichtsdetectie werd toegepast en de resultaten worden opnieuw voorgesteld in grafieken. Op de x-as worden de verschillende opeenvolgende fotonummers van eenzelfde reeks getoond en de y-as toont opnieuw de probabiliteit van elke emotie. De fotoreeks gekozen voor de emotie boosheid bevat iets meer dan 70 foto's en daarom vertonen alle grafieken van deze reeks meerdere meetpunten. De fotoreeks voor de emotie verrassing vertoont meer rechte lijnen omdat deze reeks slechts 15 foto's bevat, net zoals de reeks voor afkeer.

Kairos vertoont bij de reeks voor de emotie angst ook veel vlakke lijnen en dit komt door het normaliseren van de waarden. Omdat er geen andere emoties dan angst aanwezig zijn, krijgt angst vanaf foto 19 een probabiliteit van 100 %. Hetzelfde geldt voor de emoties blijheid, verrassing, verdriet en afkeer. De andere emotiegrafieken van Kairos werden ook genormaliseerd maar hebben meer “grillige” lijnen omdat er ook nog andere emoties werden gedetecteerd. Bij Face++ en Microsoft wordt het duidelijk dat verdriet vaak gedetecteerd wordt met een kleine probabiliteit bij boosheid en angst als verwachte emotie. Microsoft is de enige waarbij de grafieken voor alle zes de emoties op het einde toe overgaan naar de juiste verwachte emotie zonder dat er verwarring is met een andere emotie. Bij de grafiek van verdriet van Face++ bijvoorbeeld gaat de lijn voor verdriet naar de laatste foto toe wel over naar een hoge waarschijnlijkheid voor verdriet maar bij de laatste foto's wordt de probabiliteit van afkeer groter. De afkeergrafiek van Kairos toont op het einde naar de laatste foto toe een verwarring van boosheid. Opnieuw kan er dus besloten worden dat Microsoft de beste emotiedetectieresultaten behaalt voor alle zes de emoties.

Indien naar alle grafieken van Microsoft wordt gekeken, valt het op dat er voor de emoties boosheid, angst en afkeer nooit een probabiliteit van 100 % werd waargenomen, zelfs niet bij de meest expressieve foto's die de laatste foto's van elke reeks bevatten. Bij verdriet, blijheid en verrassing wordt er wel vaak een waarschijnlijkheid van 100 % vastgesteld wat wilt zeggen dat deze drie emoties makkelijker te detecteren zijn. Dit verschil in probabiliteitdetectie vormt een probleem voor de applicatie aangezien deze de emoties niet gelijk mag behandelen. Bijgevolg wordt het probleem opgelost door in de applicatie rekening te houden dat de emoties boosheid, angst en afkeer moeilijker te detecteren zijn en zal er nooit verwacht worden dat diezelfde getoonde emoties van de gebruiker met 100 % probabiliteit zullen gedetecteerd worden. De app zal dus minder streng zijn op de detectie van deze emoties en bij de berekening van de similariteit in [4.4.1](#) zal blijken op welke manier dit verwezenlijkt wordt.

3.4 Testen voor bedekte gezichtsdelen, gesloten ogen, hoofdpositie en aandachtsniveau

De drie services worden ook onderworpen aan een dataset van 76 foto's waarin ze elk hun eigen features detecteren. Microsoft test drie soorten bedekkingen: de ogen, het voorhoofd en de mond. De services Face++ en Kairos testen beide hoe goed ze het aandachtsniveau op basis van de hoofdpositie kunnen bepalen. Wanneer de absolute waarden van de gedetecteerde yaw- en pitch-richting van het hoofd (uitgedrukt in graden) respectievelijk buiten de 30 en 20 graden ligt, kan er vastgesteld worden dat het gezicht weggedraaid is en de persoon geen aandacht houdt. Ook in de code wordt met deze waarden gewerkt. Face++ detecteert op de foto's of er gesloten ogen aanwezig zijn en wat de kans is voor het linker- en rechteroog. Indien zowel linker- als rechteroog een probabiliteit van 40 of meer hebben, wordt vastgesteld dat de persoon beide ogen gesloten heeft. Kairos detecteert het aandachtsniveau door een getal tussen 0 en 1 weer te geven, met 0 als geen aandacht en 1 als volledige aandacht. Hierbij houdt Kairos naast de hoofdpositie ook rekening met de ogen en waarop ze gericht zijn. De drempelwaarde voor het wel of geen aandacht hebben, wordt hierbij op 0,5 gelegd.

De dataset van 76 foto's werd opgebouwd met de foto's van de testpersonen bij het testen van de applicatie en enkele foto's waarop personen expliciet een gezichtsdeel bedekken of geen aandacht tonen. Manueel worden de 76 foto's bekeken en wordt er bepaald of de persoon op de foto de aandacht houdt, of er bedekkingen van het gezicht zijn en of er ogen gesloten zijn. Vervolgens worden de drie services erop losgelaten en worden hun resultaten vergeleken met de echte, manueel bepaalde waarden in een Excelbestand. Tabel [3.3](#) geeft een overzicht van de resultaten van de drie services en vermeldt voor elke service het percentage fouten en het percentage van de foto's waarin geen gezicht werd gedetecteerd door die service. Het percentage fouten werd bepaald op de foto's waarin wel een gezicht gedetecteerd werd.

Microsoft heeft in 2,63 % en 5,26 % van de foto's de bedekking van respectievelijk de ogen en de mond verkeerd ingeschat. De meeste foto's in de dataset bevatten geen gezichten waarvan er een deel bedekt is, maar als er wel een deel bedekt is, detecteert Microsoft het meestal wel. De bedekking van het voorhoofd staat hier ter info bij maar wordt verder in de applicatie niet gebruikt aangezien een bedekt voorhoofd niet wijst op een positieve of negatieve feedback van de gebruiker voor de film. In de app zijn bijgevolg enkel bedekte ogen en een bedekte mond van belang omdat dit kan wijzen op een gebruiker die de handen voor het gezicht houdt indien hij de content te eng vindt, en dus op die manier een impliciete feedback aan de film geeft. Indien er een te groot deel van het gezicht bedekt is, kan Microsoft geen gezicht detecteren. De service detecteert ook geen gezicht als het te veel is weggedraaid van de camera. Beide gevallen zorgen ervoor dat er in 7,89 % van de foto's geen gezicht gedetecteerd werd door Microsoft.

Tabel 3.3: De resultaten van de drie services toegepast op de dataset voor gezichtsdetectie van gesloten ogen, bedekte gezichtsdelens, aandachtsniveau en hoofdpositie

Service en gedetecteerde feature	% foutieve detectie
Microsoft	
Bedekking ogen	2,63 %
Bedekking mond	5,26 %
Bedekking voorhoofd	1,32 %
% geen gezicht gedetecteerd	7,89 %
Face++	
Gesloten ogen	2,63 %
Hoofdpositie voor aandachtsbepaling	27,63 %
% geen gezicht gedetecteerd	2,36 %
Kairos	
Aandachtsniveau	17,11 %
Hoofdpositie voor aandachtsbepaling	39,47 %
% geen gezicht gedetecteerd	11,84 %

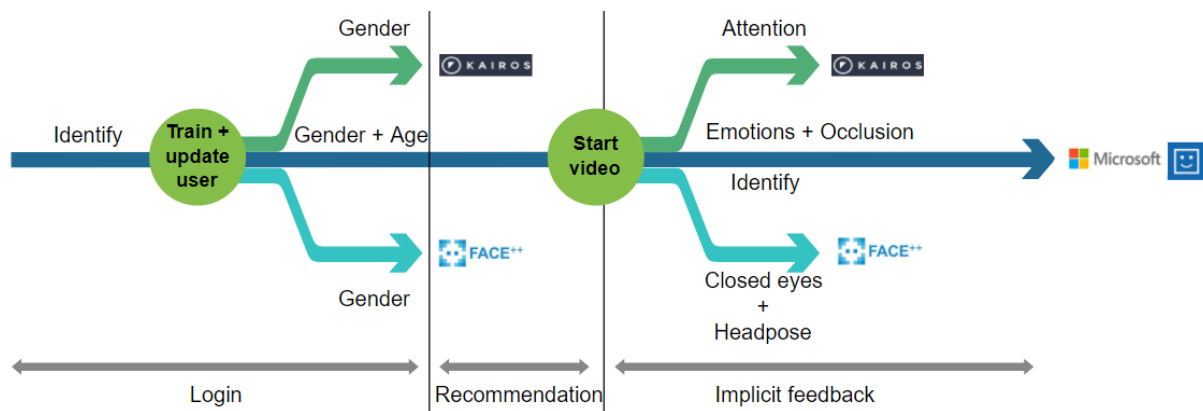
Face++ behaalt goede resultaten bij de detectie van de gesloten ogen. 2,63 % van de foto's wordt fout ingeschat op het aanwezig zijn van gesloten ogen. In 27,63 % van de foto's wordt het aandachtsniveau op basis van de hoofdpositie verkeerd ingeschat met de yaw-marge van 30 en pitch-marge van 20 graden. In 2,63 % van de foto's werd geen gezicht gedetecteerd door Face++. De aandachtsdetectie van Kairos registreert bij 17,11 % van de foto's een foutief ingeschat aandachtsniveau. De aandachtsinschatting op basis van de hoofdpositie bij Kairos behaalt een fout in 39,47 % van de foto's. De hoofdpositie-inschatting van Face++ geeft dus betere resultaten dan die van Kairos. De bepaling van het aandachtsniveau van Kairos geeft betere resultaten dan de bepaling van het aandachtsniveau van Face++ op basis van de hoofdpositie, er zijn namelijk 10,52 % minder fouten. Kairos detecteert in 11,84 % van de foto's geen gezicht, wat hoger ligt dan de ongedetecteerde gezichten van Face++ en Microsoft.

3.5 Besluit voor het gebruik van de services

Nu de services op hun verschillende featuredetecties uitgebreid getest zijn, kan er besloten worden welke service voor welke detectie in de app zal worden gebruikt. Allereerst is het duidelijk dat Microsoft de enige is die zal instaan voor de inschatting van de leeftijden. De drie services zullen door elkaar gebruikt worden voor de inschatting van het geslacht. Voor de emotiedetectie wordt eveneens Microsoft gebruikt aangezien deze bij de emotietesten ook de beste resultaten haalde. Bij de bepaling van de emoties van de kijkers zal er rekening worden gehouden met het feit dat de emoties boosheid, angst en afkeer moeilijker met een hoge probabiliteit te detecteren zijn. De app zal ervoor zorgen dat het milder is bij de detectie van deze emotie door aangepaste emotiefingerprints te voorzien met lagere probabiliteiten.

Microsoft is ook degene die steeds zal instaan voor de gezichtsherkenning met behulp van de voorziene Face Storage. De gezichtsherkenning is een noodzakelijke stap die continu gedurende het gebruik van de applicatie zal uitgevoerd worden en centraal staat in het gebruik van de app omdat elke handeling van elke gebruiker moet gelinkt worden aan die gebruiker. Figuur 3.7 toont Microsoft met de donkerblauwe

pijl centraal gedurende de drie fasen van de applicatie. Face++ en Kairos worden af en toe gedurende het gebruik ingezet om extra features te analyseren waarin zij goed zijn. De services worden allemaal asynchroon opgeroepen en worden daarom met parallelle pijlen voorgesteld.



Figuur 3.7: De flow van de verschillende services doorheen het gebruik van de applicatie

Aangezien de aandachtsdetectie van Kairos slechts bij 17,11 % van de foto's een fout besluit heeft genomen, wordt deze service gebruikt voor het bepalen van de aandacht. Niettemin kan er niet enkel op deze detectie vertrouwd worden en zal er nog een factor in beschouwing gebracht moeten worden om het aandachtsniveau te bevestigen. Het gratis account van Kairos heeft slechts een transactiefrequentie van 25 aanvragen per minuut ter beschikking. Er worden 30 foto's genomen per minuut wat wilt zeggen dat Kairos gemiddeld 5 keer per minuut geen aandachtsdetectie zal kunnen uitvoeren. Bovendien faalde Kairos bij 11,84 % van de foto's doordat het geen gezicht kon detecteren op de foto. Indien Kairos dus geen aandachtsdetectie kan uitvoeren, wordt een inschatting van de hoofdpositie als tweede detectie ingeschakeld. Er wordt gekozen voor de hoofdpositiedetectie van Face++ aangezien deze bij de bovenstaande testen bewezen heeft beter te presteren dan die van Kairos. Een derde detectie is het kijken naar de gesloten ogen. Wanneer een persoon gesloten ogen heeft op een foto kan worden vastgesteld dat de persoon op dat moment ook geen aandacht heeft. Dit kan bijvoorbeeld gebeuren wanneer de persoon de content te eng vindt om naar te kijken of wanneer een persoon in slaap gevallen is, wat beide wijst op een laag aandachtsniveau. Wanneer een persoon net toevallig met de ogen knippert, is dit slechts één foto op de zovele en wordt deze foto zonder aandacht verwaarloosbaar ten opzichte van de andere.

Hoofdstuk 4

Applicatie

In Hoofdstuk 2 kwam reeds aan bod hoe de algemene architectuur van de applicatie er uitziet, geïllustreerd door Figuur 2.1. Een aanbeveling wordt gegenereerd (stap 2) door te analyseren wie er aanwezig is (stap 1) en wat deze mensen in het verleden van een vorige film vonden (door informatie uit de databank te halen die bij stap 3 werd verzameld bij vorige kijkbeurten). Stap 3 is namelijk verantwoordelijk voor het verzamelen van impliciete feedback terwijl gebruikers naar een film kijken. Deze impliciete feedback moet de waardering van de gebruikers schatten in een score. De geschatte score wordt ook wel de waarderingsscore genoemd en is gebaseerd op de getoonde emoties van de kijkers, hun aandachtsniveau en hun gedrag. Met deze geschatte waarderingsscore over voorgaande video's weet het systeem wat voor andere, gelijkende video's de gebruikers ook zouden willen bekijken en hierop kan het systeem zijn aanbeveling baseren.

In dit hoofdstuk wordt beschreven hoe de applicatie gecreëerd is: welke keuzes er gemaakt zijn voor de verschillende onderdelen, hoe mensen automatisch worden ingelogd, hoe het aanbevelingssysteem gemaakt wordt, hoe het cold start probleem wordt opgelost, hoe impliciete feedback in de vorm van ratings verzameld wordt en hoe de waarderingsscore wordt opgebouwd. De drie grote stappen worden overlopen samen met hun extra uitbreidingen. Hierna worden verschillende diagrammen gedemonstreerd om de architectuur te bespreken. Tenslotte komt het testplan aan bod. Het hoofdstuk bespreekt ook de code aan de hand van enkele codevoorbeelden in listings. De listings die langer zijn en meer plaats innemen, worden in [Bijlage C: Grafieken, diagrammen, code en tabellen](#) getoond.

4.1 Androidapplicatie en externe benodigdheden

In deze masterproef werd gekozen om een app te maken voor een Androidtoestel. Androidtoestellen beschikken over camera's die nodig zijn om de beelden te maken voor de gezichtsdetectie. Tijdens het analyseren van de beelden, kan de applicatie een film op het scherm van het Androidtoestel tonen terwijl de camera op dat moment gericht is op de gezichten van de kijkers. Op smartphones en tablets van Android is hiervoor een front camera vereist. Op Android smart TV's is het vereist dat er een camera aanwezig is op of in de tv. De gezichtsdetectie zal gerealiseerd worden met een service die internet vereist en op een Androidtoestel is er de mogelijkheid verbinding te maken met een wifi-netwerk of met het mobiele netwerk. Kortom, een Androidtoestel voorziet alle benodigdheden om de aanbeveling te genereren en video's af te spelen.

4.1.1 Android Studio

Voor de ontwikkeling van de app wordt Android Studio gebruikt, de officiële *Integrated Development Environment* (IDE) om apps te maken. Android is een besturingssysteem ontwikkeld door Google voor mobiele toestellen zoals smartphones, tablets, smart watches en smart TV's [47]. De programmeertaal is gebaseerd op Java. Gebruikers kunnen op hun Androidtoestel applicaties of apps installeren die verspreid worden via de Google Play Store, de tegenhanger van de App Store voor iOS.

4.1.2 Room Database

De Room Persistence Library behoort tot de *Android Architecture Components*, een verzameling bibliotheken van Android die helpen bij het maken van apps [48]. Deze bibliotheek zorgt voor een abstractielaag bovenop SQLite waardoor de app kan gebruik maken van een lokale SQLite databank. Dankzij Room kan data van de app opgeslagen worden en kunnen gebruikers een consistente weergave zien van wat deze data zijn zonder dat er nood is aan een verbinding met het internet. In de applicatie zal gebruik gemaakt worden van deze bibliotheek om de verschillende gebruikers op te slaan, hun kijkgeschiedenis en de informatie over de beschikbare films zoals genre, regisseur en demografische ratings. Het databank-model van deze verschillende tabellen wordt uitgebreid beschreven in 4.6.2. Net zoals alle langdurende operaties in de app, moeten de databankmethoden opgeroepen worden met een asynchrone taak zodat ze de hoofdthread niet overbelasten. In de code wordt dit uitgevoerd met de AsyncTask-klasse.

4.1.3 YouTube API

De video's die de gebruiker van de app zal zien, zullen met behulp van streaming worden getoond in de app. De *YouTube Android Player API* voorziet functionaliteit om video's in de Androidapp te laten afspelen [49]. Hierdoor moeten de video's niet lokaal aanwezig zijn op het toestel en niet gedownload worden voor de gebruiker ze kan bekijken.

Er worden fragmenten uit films gebruikt die op YouTube te vinden zijn. Om volledige films te kunnen afspelen, moet de film eerst aangekocht worden of moet een abonnement gebruikt worden van een streamingsdienst zoals Netflix. In deze masterproef wordt gekozen voor korte fragmenten uit films zodat een gebruiker op korte tijd meer video's kan bekijken en zo snel meer data en meer geschiedenis verzamelt dan wanneer testpersonen volledige films bekijken. De fragmenten variëren van één tot vijf minuten en werden zo gekozen dat ze de overheersende emotie van de film bevatten. Voor komedies wordt er bijvoorbeeld een hele grappige scène gekozen terwijl er voor horrorfilms een scène wordt gekozen waarbij de kijkers vooral emoties van afkeer en angst zullen tonen.

Om de YouTube API te gebruiken moet de applicatie geregistreerd worden in de Google Developers Console en moet er een Android API sleutel verkregen worden. Het verkrijgen van deze sleutel wordt beschreven in [Bijlage A: Installatiehandleiding](#). Op het Androidtoestel is het ook vereist dat de YouTube-app ook apart geïnstalleerd is. Dit is geen groot probleem aangezien Android deze vaak al standaard installeert op de smartphones.

4.1.4 Invulling databank

Om de applicatie van video's te voorzien, werden dertig interessante films en bijhorende filmfragmenten gekozen. De films werden zo gekozen zodat de verzameling alle genres, tags en leeftijdsgroepen bevat. Ook werd ervoor gezorgd dat de bijhorende filmfragmenten steeds de overheersende emoties van de film bevatten. Bij het initialiseren van de Roomdatabank wordt een json bestand uitgelezen met alle nodige informatie over elke film zoals jaar, genres, beschrijving, ratings, minimumleeftijd en de URL van het YouTubefragment. Dit bestand heet `movies.json` en is te vinden in map `EyeVideo/app/src/main/assets` van de applicatie. De genres, de beschrijving en de ratings worden van de site van Internet Movie Database (IMDb) gehaald [6]. Deze site bevat een gemiddelde globale rating op tien verzameld uit recensies en beoordelingen van mensen over de hele wereld. Voor elk van de dertig films werd bovenstaande informatie reeds voorzien, maar slechts voor de eerste zes (de video's gebruikt voor de testen) werd een bijhorend spectrum van verwachte emoties voorzien aangezien enkel hiervoor testen en onderzoek werd gedaan. De andere video's kunnen ook bekeken worden maar bij de bepaling van de waarderingsscore zal geen rekening kunnen worden gehouden met de verwachte emoties aangezien deze er (nog) niet zijn.

De gemiddelde globale rating op tien van IMDb geeft een goed beeld van de kwaliteit van de film maar is nutteloos voor een aanbevelingssysteem aangezien sommige mensen een film met een goede rating, ook heel slecht kunnen vinden. Daarom wordt er ook gebruik gemaakt van de demografische ratings van IMDb. Hier geeft de site een overzicht van de mensen die gestemd hebben op de film, verdeeld volgens hun leeftijd en geslacht. Er zijn vier leeftijdsgroepen te onderscheiden: -18 jaar, 18 tot 29 jaar, 30 tot 44 jaar en 45 jaar en ouder. Indien er een aanbeveling moet gemaakt worden voor een persoon waarvan de

leeftijd en het geslacht gekend zijn, kan er een lijst van films worden gegeven waarvan de demografische rating voor zijn demografische groep hoger ligt dan de gemiddelde rating over alle demografische groepen. Er kan dan van uit gegaan worden dat deze persoon een hoge kans heeft deze lijst van films leuk te vinden. Op deze manier kan er een aanbeveling gegeven worden aan mensen die nog geen feedbackgeschiedenis hebben en is het cold start probleem opgelost. Bij het kiezen van de films en filmfragmenten in de app werd er ook voor gezorgd dat er veel verschil was in demografische ratings zodat mensen van verschillende demografische groepen duidelijke verschillen in demografische aanbeveling zien. De ratings per demografische groep worden in databank bijgehouden in de Ratingstabel.

De aanbeveling houdt ook rekening met films die onaanvaardbaar zijn voor sommige mensen die aanwezig zijn. Bijvoorbeeld als het systeem weet dat er bij het aanmelden een kind van zes jaar aanwezig is, zal het geen film aanbevelen die enkel geschikt is voor 16+. Elke film wordt dus ook voorzien van een minimumleeftijd. Om te bepalen waar de grens ligt voor een minimumleeftijd werd er opnieuw beroep gedaan op een site die informatie verzamelt uit beoordelingen en reviews van mensen over de hele wereld. Common Sense is een organisatie die families en kinderen helpt in het maken van keuzes voor media en technologie [50]. Ze voorzien informatie over ratings en reviews voor onder andere films, boeken en muziek. Deze informatie bevat de gemiddelde leeftijd waarop ouders vinden dat kinderen de film mogen bekijken en analoog de leeftijd waarop kinderen zelf vinden dat deze film geschikt is om naar te kijken. Dankzij deze data wordt er een goed beeld gegeven van een film wanneer deze te veel geweld, grof taalgebruik of andere ongeschikte inhoud bevat. Op deze manier kan een beslissing gemaakt worden voor de minimumleeftijd van een film.

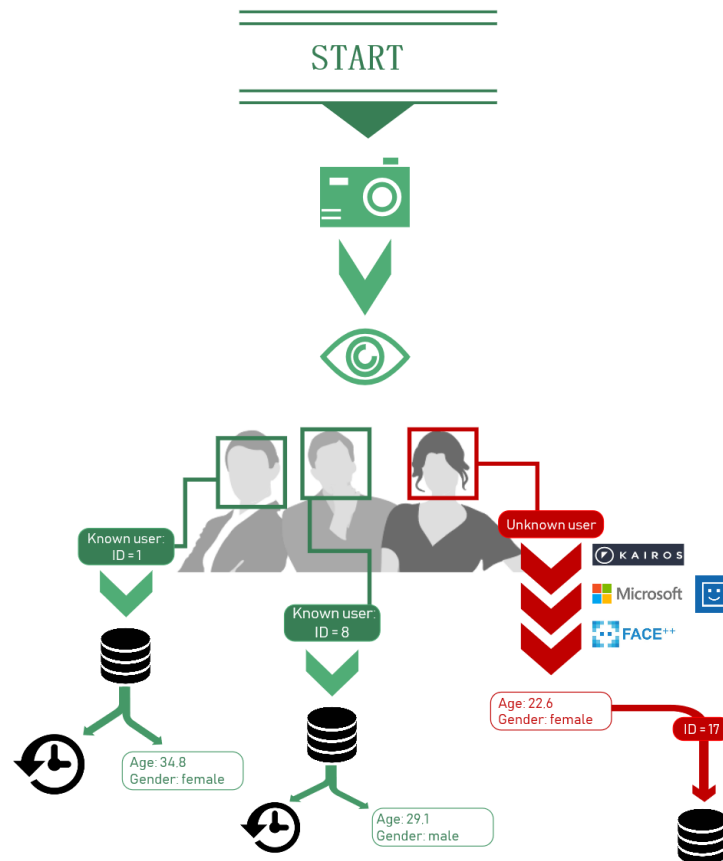
4.1.5 Gebruikte smartphone en tablet

Voor het uittesten van de software geschreven in Android Studio werden twee toestellen gebruikt, een smartphone en een tablet. Bij beide is telkens enkel de front camera van belang aangezien deze steeds gericht is op de gebruikers wanneer ze een film kijken. De Samsung Galaxy S7 is een smartphone met een 5MP front camera en een scherm met diameter van 5,1 inch (12,95 cm). Als tablet werd de Samsung Galaxy Tab S3 gebruikt met scherm van 9,7 inch (24,64 cm) diameter en ook met een 5MP front camera [51]. Het minimum SDK versie nummer van de app is API level 18.

4.2 Stap 1: Automatische aanmelding

Om de aanbeveling te kunnen maken, moet de applicatie weten wie er aanwezig is. Gebruikers moeten zich niet zelf één voor één aanmelden. Het volstaat dat ze hun gezichten tonen voor de camera die een foto neemt wanneer ze op de startknop duwen. De app toont in de achtergrond een voorbeeld van wat de camera ziet zodat de gebruikers ervoor kunnen zorgen dat hun gezicht correct op de foto staat. Het systeem detecteert alle aanwezige gezichten en gaat ze één voor één af om te kijken of hij de persoon herkent. Er wordt dus gezichtsherkenning toegepast. Indien het systeem iemand herkent, haalt hij het ID van deze persoon op en logt hij de persoon in. Indien er iemand onbekend voor de camera zit, wordt de persoon aangemaakt in het systeem. Hiervoor moet zijn leeftijd en geslacht geschat worden. Figuur 4.1 toont van boven naar beneden de verschillende stappen. De twee meest linkse gezichten worden herkend door het systeem als gebruikers met ID 1 en 8. Deze personen worden automatisch aangemeld. Via hun ID heeft het systeem toegang tot de persoonlijke informatie van elke gebruiker uit de databank zoals geslacht, leeftijd en de kijkgeschiedenis met hun impliciete feedback.

De meest rechtse persoon is nieuw voor het systeem. De leeftijd en het geslacht van de persoon worden nu geschat door de drie verschillende services te raadplegen. Ook de leeftijd en het geslacht van de bestaande gebruikers worden opnieuw geschat zodat de gebruikersprofielen worden geüpdatet (verder uitgelegd in 4.5). Enkel de leeftijd geschat door Microsoft zal meetellen voor de leeftijdsbepaling en het geslacht wordt bepaald door de drie services te laten stemmen, zoals besproken in Hoofdstuk 3. Het systeem slaat de nieuwe gebruiker op met ID 17 samen met de geschatte leeftijd en geslacht en meldt ook deze nieuwe gebruiker aan.



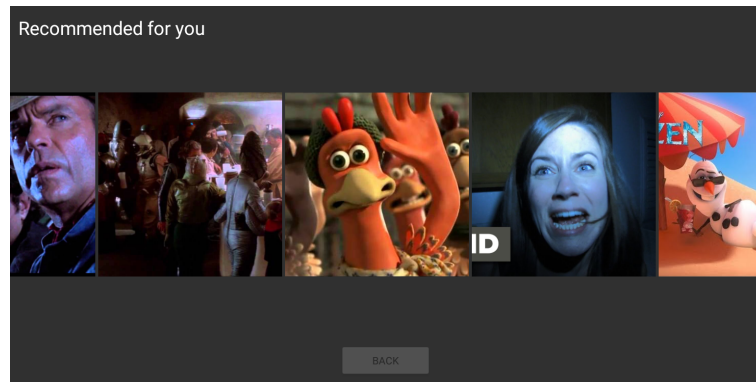
Figuur 4.1: Stap 1: Automatisch inloggen van de aanwezige gebruikers met gezichtsherkenning

De wachttijd voor het antwoord van de services vormt een probleem. Het zou namelijk niet efficiënt zijn om deze drie services steeds te laten samenwerken bij elke aanmelding en steeds te wachten op hun antwoord alvorens ze kunnen worden aangemeld en hun aanbeveling kan worden berekend. Dit wordt opgelost door de services asynchroon op te roepen en ze asynchroon de databank te laten aanvullen en ze zelf de uitslag van het stemmen (met de informatie die reeds aanwezig is in de databank) te laten berekenen. Het is Microsoft Face API die initieel zorgt voor de identificatie van elke persoon en de informatie hierbij bevat reeds de geschatte leeftijd en het geschatte geslacht door Microsoft. Op dat moment wordt de persoon aangemeld of wordt een nieuwe persoon aangemaakt in de databank met een toegewezen ID en geschatte geslacht en leeftijd van Microsoft. De gebruikers worden nu al aangemeld zodat ze niet moeten wachten op de resultaten van de andere services. Hierdoor kan de gebruiker reeds zijn persoonlijke aanbeveling zien en wordt de user experience verhoogd. Terwijl de gebruiker verder kan vanaf het moment waarop Microsoft klaar is met aanmelden, wordt de gezichtsdetectie van Face++ en Kairos opgeroepen in een asynchrone taak. Deze passen dan asynchroon ook de databank aan met hun info (hun geschatte informatie wordt ter info bijgehouden in de databank) en deze herberekenen hun uitkomst voor het geslacht. Indien één van de services zou falen, is dit geen probleem aangezien de gebruiker verder kan zonder alle informatie van de drie services.

De gezichtsherkenning wordt uitgevoerd met de identificatie van Microsoft Face API. De service herkent gezichten die reeds in de Face Storage zitten en haalt hun naam op. Er is een één-op-één relatie tussen de naam in de lokale Roomdatabank en de Face Storage van Microsoft zodat deze makkelijk kunnen samenwerken en er geen verwarring bestaat over het ID van de betreffende gebruiker. Een gezicht in de Face Storage krijgt dus nooit de echte naam van een persoon aangezien dit zou vereisen dat de persoon zijn naam manueel ingeeft. Het doel van deze applicatie is dat de gebruikers zichzelf niet manueel moeten aanmelden dus vraagt het systeem ook niet naar hun naam. Het systeem heeft genoeg informatie met een ID van de gebruiker dat overeenkomt met het ID in de lokale databank.

4.3 Stap 2: Genereren van de groepsaanbeveling

Nu het systeem weet wie er aanwezig is, kan de aanbeveling gegenereerd worden. De gebruikers kunnen dan op de knop ‘Show Recommendation’ duwen die tevoorschijn komt vanaf er iemand aangemeld is. Er wordt een lijst van de tien meest aanbevolen video’s voorgesteld aan de gebruikers, gesorteerd op mate van aanbeveling voor de volledige aanwezige groep. Een screenshot van de app met enkele aanbevolen films wordt getoond in Figuur 4.2. Hieronder wordt uitgelegd hoe deze aanbeveling tot stand komt.

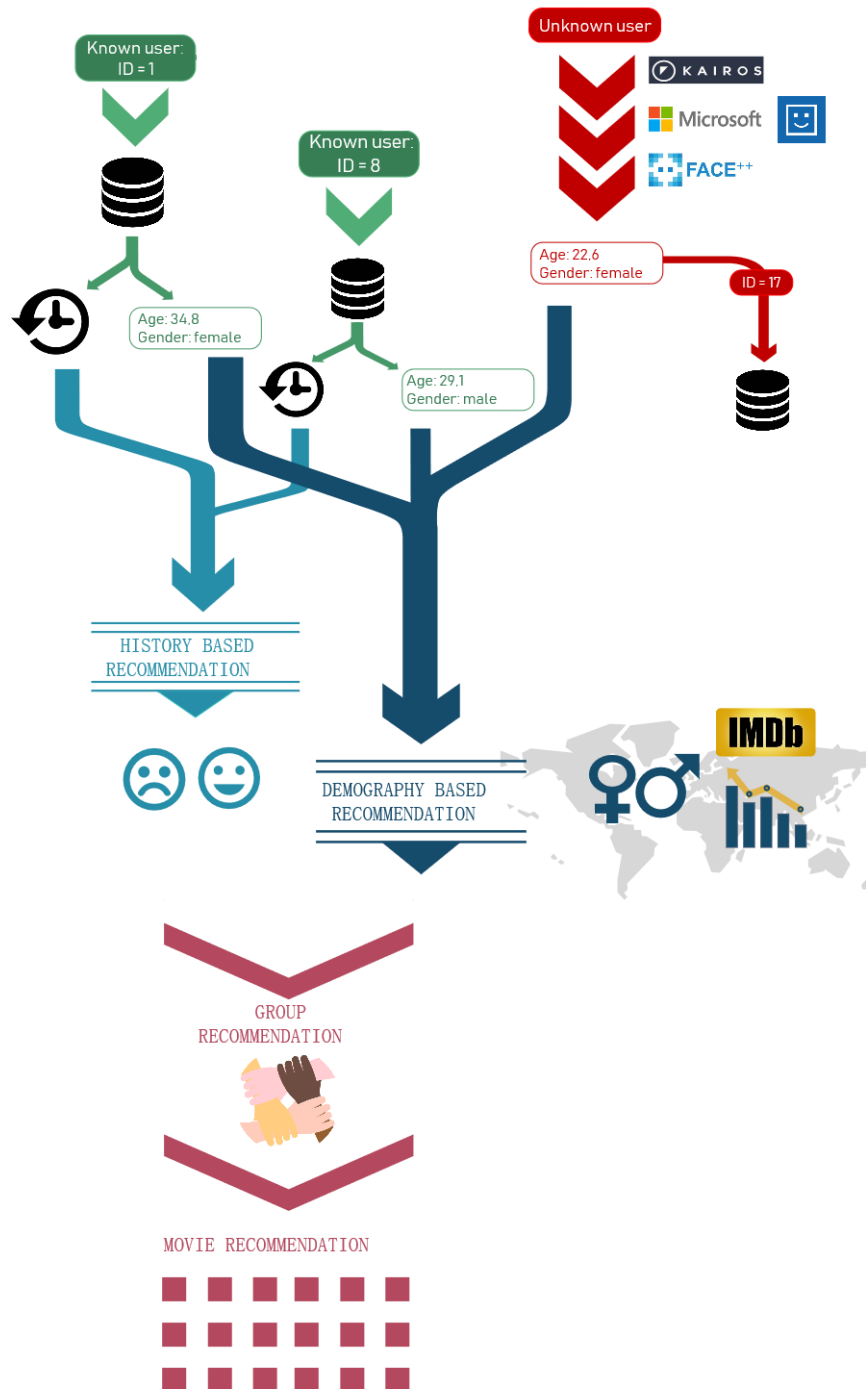


Figuur 4.2: De lijst van aanbevolen films in de app

Er wordt van uit gegaan dat degenen die een film gaan bekijken, de personen zijn die aanwezig waren wanneer de foto bij de inlogstap genomen is. De verzameling films die zal worden aanbevolen, moet gebaseerd zijn op de voorkeuren van alle aanwezige personen. Er moet dus een groepsaanbeveling berekend worden. Enerzijds zal er een groepsaanbeveling berekend worden voor de mensen die reeds feedback hebben in hun kijkgeschiedenis. Dit heet de *History Based Recommendation*. Anderzijds wordt een groepsaanbeveling berekend voor de mensen waarvan nog geen geschiedenis bekend is. Van deze mensen is enkel hun geslacht en leeftijd bekend en wordt er een *Demography Based Recommendation* berekend. Ook voor mensen die wel al geschiedenis hebben, wordt een demografische aanbeveling gemaakt aangezien het kan zijn dat deze mensen nog niet voldoende feedback hebben verzameld om een nuttige geschiedenisgebaseerde aanbeveling te genereren. Hoe meer feedbackgeschiedenis een persoon verzamelt, hoe meer de geschiedenis-aanbeveling zal doorwegen en hoe minder gewicht de demografische aanbeveling zal krijgen voor deze persoon. Het schema in Figuur 4.3 verduidelijkt deze stap en toont de verwerking van twee soorten informatie met behulp van twee pijlen:

- Demografische gegevens van alle aanwezige gebruikers (zowel degenen zonder geschiedenis als degenen met geschiedenis)
- Impliciete feedback uit de geschiedenis van de gebruikers met geschiedenis (gerealiseerd met een geschatte waarderingsscore per gebruiker per bekeken video)

Alle demografische gegevens worden per gebruiker uit de databank gehaald en worden samengenomen voor een groepsaanbeveling op basis van de demografie. De geschiedenis van gebruiker 1 en 8 wordt opgehaald uit de databank en wordt samengenomen voor een groepsaanbeveling op basis van de geschiedenis. Deze twee groepsaanbevelingen worden verder uitgelegd in respectievelijk 4.3.1 en 4.3.2. Voor alle gebruikers, zowel degenen met als zonder geschiedenis, wordt een schatting van de score (hoe goed deze gebruiker de film zou vinden) voor alle mogelijk aanbevolen films gemaakt. Deze geschatte score wordt uitgedrukt in een kommagetal op tien. Op deze manier wordt het mogelijk de History Based Recommendation en de Demography Based Recommendation uniform samen te voegen tot één groepsaanbeveling. In de volgende subhoofdstukken wordt uitgelegd hoe deze score op tien berekend wordt.



Figuur 4.3: Stap 2: De groepsaanbeveling wordt gegenereerd op basis van de verkregen informatie

4.3.1 Groepsaanbeveling op basis van demografische gegevens

Over gebruikers die nog geen films hebben bekeken in de applicatie heeft het systeem nog geen voorkeursinformatie op basis van geschiedenis. Een aanbevelingssysteem zonder voorkeursinformatie kan zo geen persoonlijke aanbeveling genereren voor die personen. Het systeem heeft wel informatie over het geslacht en de leeftijd van de persoon en kan zich dus baseren op de demografische ratings van IMDb. Als een persoon op IMDb een rating voor een film geeft, wordt zijn geslacht en leeftijd mee verwerkt in deze informatie. Zo kan IMDb per film de gemiddelde rating per demografische groep weergeven. Deze informatie kan gebruikt worden om het cold start probleem op te lossen, zoals beschreven in 2.1.3.

Berekening van de geschatte score bij de demografische aanbeveling

Als het systeem bijvoorbeeld weet dat er zich een vrouwelijke gebruiker A van 37 jaar voor de camera bevindt, kan het systeem kijken naar de films op IMDb die relatief hoog scoren bij de demografische groep voor vrouwen van 30-44 jaar. Met relatief hoog scoren wordt bedoeld: ten opzichte van het algemene gemiddelde van de film. Als het globale gemiddelde van een film bijvoorbeeld 7,0 is en voor de demografische groep is het gemiddelde bijvoorbeeld 8,4, dan weet het systeem dat de film voor deze groep hoger scoort dan het globale gemiddelde. De film scoort dus voor gebruiker A 1,4 punten hoger dan het globale gemiddelde en men kan zeggen dat dit de geschatte score op tien is voor deze persoon. Diezelfde film scoort bij een mannelijke gebruiker B van 20 jaar bijvoorbeeld slechts 6,8 waardoor zijn geschatte score op tien nu -0,2 wordt. Om niet met kleine of negatieve getallen te werken, worden de verschillen steeds opgeteld bij een basiswaarde voor de geschatte score van 8. Zo wordt de geschatte score van gebruiker A 9,4/10 en van gebruiker B wordt dit 7,8/10. Er werd gekozen voor een basiswaarde van 8 zodat de score dicht genoeg bij 10 ligt en omdat het verschil tussen de globale ratings en demografische ratings bij de voorbeeldvideo's nooit groter is dan 2 (een score groter dan 10 kan dus niet).

Onaanvaardbare films bij de demografische aanbeveling

Indien er meer dan één persoon aanwezig is, wordt een groepsaanbeveling berekend met de strategie Average without Misery, zoals uitgelegd in 2.1.4. Indien een film voor een bepaalde persoon te laag scoort met zijn relatieve rating (gebaseerd op een drempelwaarde), wordt deze film als onaanvaardbaar beschouwd voor de volledige groep. Alle mogelijke films worden afgegaan, hun gemiddelde wordt berekend over alle aanwezige gebruikers en alle onaanvaardbare films worden weggelaten. Uiteindelijk blijft er een selectie van films over, gesorteerd op hoe sterk deze door de volledige groep wordt aangeraden, gebaseerd op de gemiddelde waarde over alle aanwezige gebruikers.

Een nieuw probleem duikt op wanneer er bijvoorbeeld in de aanwezige personen ook een kind van 7 jaar aanwezig is, jonger dan de minimumleeftijd van een bepaalde film in de aanbeveling. Het systeem wilt voorkomen dat er bijvoorbeeld een horrorfilm met minimumleeftijd 16 jaar wordt aangeraden. Dit wordt opgelost door films die onaanvaardbaar zijn wegens minimumleeftijd ook uit de aanbeveling te verwijderen.

Het codevoorbeeld in Listing 4.1 toont het maken van de aanbevolen lijst met geschatte scores voor alle gebruikers in de variabele `users`. Dit is de inhoud van de methode `giveRecommendation` van de `NoHistoryRecommendation` klasse. De onaanvaardbare films worden aan de instantievariabele `unacceptable` toegevoegd. De uiteindelijke lijst met alle films bevat ook deze onaanvaardbare maar deze worden door de `RecommendationCalculator` klasse later eruit gefilterd, wat uitgelegd wordt in 4.3.3.

```
public Map<Integer, Double> giveRecommendation(List<User> users) { 1
    Map<Integer, Double> videoToRating = new HashMap<>(); 2
    Map<Integer, Integer> amounts = new HashMap<>(); 3
    for (User user : users) { 4
        Rating[] ratings = global.getDatabaseRatingHelper().getAllRatingsWithAgeAndGender( 5
            getAgeGroup(user.getAge()), user.getGender()); 6
        for (Rating rating : ratings) { 7
            int id = rating.getVideoId(); 8
            Video video = global.getDatabaseVideoHelper().getVideoWithId(id); 9
            double difference = rating.getRating() - video.getRating(); 10
            double score = BEGINSORE + difference; 11
            if (!fromHistoryRecommendation) { 12
                if (difference < THRESHOLD || video.getMinAge() > user.getAge()) { 13
                    unacceptable.add(id); 14
                } 15
            } else { 16
                if (video.getMinAge() > user.getAge()) { 17
                    unacceptable.add(id); 18
                } 19
            } 20
        } 21
    } 22
    if (videoToRating.containsKey(id)) { 23
        videoToRating.put(id, videoToRating.get(id) + score); 24
    } 25
}
```

```

        amounts.put(id, amounts.get(id)+1);
    } else {
        videoToRating.put(id, score);
        amounts.put(id, 1);
    }
}

for(Map.Entry<Integer,Double> video : videoToRating.entrySet()){
    int amount = amounts.get(video.getKey());
    double newvalue = video.getValue()/amount;
    videoToRating.put(video.getKey(), newvalue);
}
return videoToRating;
}

```

Listing 4.1: Codevoorbeeld uit NoHistoryRecommendation: berekenen van demografische geschatte scores voor de films en de onaanvaardbare films voor de groep

De databanktabel Ratings bevat de demografische ratings van alle films. De methode `getAllRatings-WithAgeAndGender` haalt van die specifieke demografische groep en van alle video's in de databank de demografische rating op. De for-lus op lijn 7 zorgt er dus voor dat alle video's in de databank overlopen worden.

4.3.2 Groepsaanbeveling op basis van impliciete feedback uit de persoonlijke geschiedenis

Een gebruiker die reeds één of meer films heeft bekeken, heeft feedback in zijn persoonlijk geschiedenis over deze film. De klasse `HistoryRecommendation` kan nu een aanbeveling op basis van deze geschiedenis berekenen. Indien deze persoon bijvoorbeeld een komedie heeft gekeken en zijn gedetecteerde emoties kwamen overeen met de verwachte emoties, in dit geval dus lachen, en zijn aandachtsniveau was ook steeds hoog, dan kan ervan uitgegaan worden dat hij het genre komedie leuk vindt. Als deze gebruiker ook nog een film uit het genre western heeft gekeken waarbij zijn aandachtsniveau zeer laag was, heeft deze gebruiker een negatieve feedback over dit genre. Het genres krijgen bijgevolg een score gebaseerd op hoe positief of hoe negatief de feedback was. Voor elke persoon en elke reeds bekeken film door deze persoon, wordt zo per persoon per genre een score berekend van hoe graag die persoon het genre heeft.

Tags invoeren die moeten zorgen voor voldoende, maar niet te veel, overlap tussen de films

Films uit eenzelfde genre kunnen onderling nog verschillen en een gebruiker vindt niet steeds alle films van éénzelfde genre leuk. Omgekeerd geldt dit ook want het is niet omdat een gebruiker één film uit een bepaald genre niet leuk vindt, dat hij alle films uit dat genre niet wilt zien. Het vormt dus een probleem indien aanbevelingen enkel aan elkaar gelinkt worden op basis van hun genre. Daarom wordt er gekozen om niet alleen met genres maar ook met tags te werken die de film beschrijven. Ze kunnen verschillende films met elkaar verbinden als de films op elkaar lijken. Ze handelen als een rode draad doorheen alle vergelijkbare films en zorgen voor een veel fijnere definiëring van genres en subgenres. Voorbeelden van zo'n tags zijn: geweld, dansen, emotionele monologen, bloed enzovoort.

De tags werden deels gehaald van de *Plot Keywords* die IMDb voorziet bij elke film [6]. De meest voorkomende tags werden gebruikt en gelijkaardige, zoals heksen/demonen/vloek/spoken ..., werden samen gezet onder één tag. De 24 genres die IMDb voorstelt werden ook toegevoegd aan de tagslijst en zo werd een lijst van 48 tags bekomen. De keuze van de filmfragmenten en tags werd op elkaar afgestemd zodat ze elkaar genoeg overlappen maar ook niet te veel. Zo komt de tag 'crying' in de vier filmfragmenten uit de dertig voor waarin er geweend wordt. De tag 'female protagonist' bijvoorbeeld werd niet meegenomen in de tagslijst aangezien deze ongeveer in 1/2 van de filmfragmenten zou voorkomen en zo zorgt voor te veel overlap. Een goed evenwicht moet gevonden worden om zinvolle aanbevelingen te krijgen. Elk filmfragment heeft ook meerdere tags, namelijk de genres die overeenkomen met dat fragment (niet met de volledige film) en de tags die voldoen aan dat fragment. Ook hier werd ervoor gezorgd dat een film niet te veel tags heeft zodat de gegenereerde aanbeveling niet opeens alle films bevat doordat alle film

rechtstreeks met elkaar verbonden zijn door gemeenschappelijke tags. De applicatie bevat momenteel slechts 30 video's en 48 tags. De applicatie kan dus nog veel uitgebreid worden en hoe meer video's aanwezig zijn, hoe specifieker de tags kunnen gemaakt worden.

Berekening van de geschatte score bij de geschiedenisgebaseerde aanbeveling

Alle personen waarvan geschiedenis bekend is, worden meegegeven aan de `giveRecommendation`-methode en worden hierin overlopen in een `for-lus`. De code hiervan is te zien in Listing C.1 en Listing C.2 die respectievelijk deel 1 en deel 2 van de methodecode van de klasse `HistoryRecommendation` bevatten en enkel de meest interessante delen tonen (de berekening van het gemiddelde wordt bijvoorbeeld niet getoond). De `ViewHistoryrecords` van deze gebruikers, die een record bevatten voor elke film die de gebruiker heeft gezien met bijhorende geschatte score, worden opgehaald en worden ook overlopen. Voor elke reeds bekeken film uit elk `ViewHistoryrecord` kan het video-ID aan de instantievariabele `unacceptable` van `HistoryRecommendation` worden toegevoegd en worden alle tags overlopen. Voor elke tag van elke film uit de `ViewHistoryrecords` wordt de geschatte waarderingsscore van die kijkbeurt toegekend aan die tag. Dit wordt gerealiseerd door een `map tagScoresMap` bij te houden die als sleutel het tag-ID heeft en als waarde alle geschatte scores voor die tag. Vervolgens wordt per tag-ID het gemiddelde van alle geschatte scores genomen om zo een geschatte score voor die tag te bekomen voor de huidige gebruiker (met behulp van de `map amounts`).

De `map candidatesForThisUser` beeldt elk video-ID af op de berekende score die deze user voor die video zou hebben. Deze score wordt berekend door voor alle tags alle video's af te gaan die deze tag bevatten en de geschatte tagscore toe te kennen aan dit video-ID in de `map candidatesForThisUser`. Vervolgens wordt er opnieuw per video-ID het gemiddelde genomen van alle toegekende tagscores. Voor elke user wordt zo een `map candidatesForThisUser` bijgehouden.

Elke gebruiker die geschiedenis heeft, heeft vaak nog niet voldoende geschiedenis om een correcte aanbeveling te genereren. Daarom worden de demografische geschatte scores, berekend door `NoHistoryRecommendation`, vermengd met de net gegenereerde geschiedenisgebaseerde scores. Naarmate een gebruiker meer en meer geschiedenis verzamelt, zal zijn geschiedenisgebaseerde score meer en meer doorwegen. In de app werd gekozen om de geschiedenisgebaseerde scores $10 \cdot X \%$ te laten doorwegen, met X het aantal `ViewHistoryrecords` dat er bestaat voor deze gebruiker. Indien een gebruiker bijvoorbeeld al vier films bekeken heeft, zal de geschiedenisgebaseerde score 40% doorwegen en de demografische score 60% .

Alle gebruikers worden nogmaals overlopen om nu, met de kennis over hoe goed ze elke film zouden vinden, een groepsaanbeveling te maken. Dit wordt gerealiseerd in de code vanaf lijn 35 in Listing C.2 door de score van deze gebruikers voor alle films in de `map candidatesScores` te stoppen met als sleutel het video-ID en als waarde de score van alle gebruikers voor deze films. Tenslotte wordt opnieuw voor elk video-ID de gemiddelde score berekend waardoor men een `map` bekomt die alle video-ID's afbeeldt op hun overeenkomstige geschatte score voor alle gebruikers.

Onaanvaardbare films bij de geschiedenisgebaseerde aanbeveling

Een film wordt in de `unacceptable`-lijst ondergebracht als één van de gebruikers uit `users` deze film reeds heeft bekeken of indien één van de gebruikers te jong is om de film te bekijken. Deze laatste voorwaarde kan nagegaan worden door de `unacceptable`-lijst van `NoHistoryRecommendation` op te vragen wanneer deze de demografische scores berekent om er de gewichten op toe te passen. Het is een bijkomend voordeel om deze klasse te gebruiken aangezien deze klasse alle films uit de databank overloopt en dus zijn `unacceptable`-lijst baseert op alle mogelijke films en hun minimumleeftijden. `HistoryRecommendation` doet dit niet aangezien die enkel films overloopt die voorkomen in de reeds bekeken tags van die gebruiker. Het nadeel aan de `NoHistoryRecommendation` is dat deze ook kijkt naar onaanvaardbaarheid door een te lage demografische score. Hier wilt het systeem op dit moment geen rekening meer mee houden aangezien de aanbeveling meer gebaseerd kan zijn op geschiedenis. Het systeem wilt bij een `HistoryRecommendation` enkel rekening houden met de reeds bekeken films en de films die onaanvaardbaar zijn wegens hun minimumleeftijd. Hiervoor geeft de `HistoryRecommendation` mee aan `NoHistoryRecommendation` dat het wordt opgeroepen vanuit een geschiedenisgebaseerde aanbeveling, waardoor `unacceptable` van `NoHistoryRecommendation` hier rekening mee zal houden. Dit is te zien

in Listing 4.1 op lijn 14 waar er wordt nagegaan of de aanbeveling dient voor de HistoryRecommendationklasse of niet. De unacceptable-lijst van NoHistoryRecommendation wordt tenslotte toegevoegd aan die van HistoryRecommendation. Opnieuw worden alle unacceptable-lijsten pas verwerkt in de RecommendationCalculatorklasse die alle scores verwerkt tot één grote aanbeveling. Deze stap wordt uitgelegd hieronder in 4.3.3.

4.3.3 Aggregeren van de twee afzonderlijke groepsaanbevelingen

De twee voorgaande groepsaanbevelingen genereren elk afzonderlijk een map met als sleutel het video-ID en als waarde de geschatte score voor die film, met andere woorden hoe sterk de aanbeveling is. Het samenvoegen van de twee lijsten gebeurt door de klasse RecommendationCalculator die beide mappen samenvoegt in één map op basis van de sleutel. Indien een video in beide mappen voorkomt, neemt RecommendationCalculator het gemiddelde van beide waarden als waarde bij de video-ID-sleutel. Vervolgens haalt RecommendationCalculator alle video-ID's uit unacceptable uit de samengevoegde map, zowel van NoHistoryRecommendation als van HistoryRecommendation. Tenslotte wordt de lijst gesorteerd op de waarden zodat de gebruikers de sterkst aanbevolen video's als eerste zien.

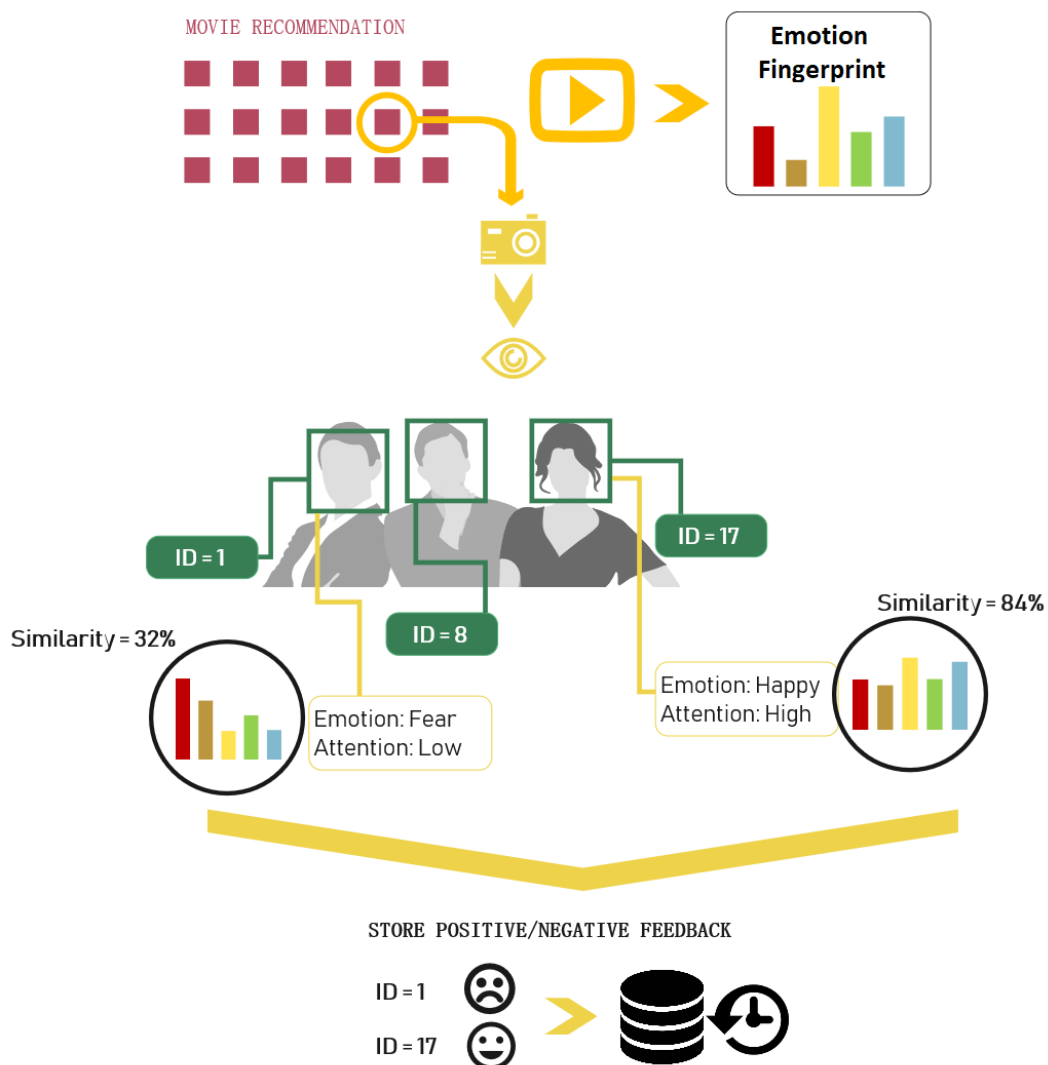
4.4 Stap 3: Impliciete feedback verzamelen

Vanaf het moment dat de gebruikers een film kiezen uit de selectie en de film of het filmfragment start, wordt er om de twee seconden een foto genomen van de gezichten van de kijkers, zoals te zien in Figuur 4.4. Alle aanwezige gezichten worden gedetecteerd, geverifieerd met gezichtsherkenning, hun emoties worden bepaald samen met het aandachtsniveau en het gedrag van de persoon. Op basis van deze info wordt er per persoon zo impliciete feedback over de interesse in de film bepaald en dit wordt opgeslagen in de geschiedenis van de persoon. Stap 3 wordt gedetailleerd uitgelegd volgens een sequentiediagram in 4.6.4. Een overzicht van de verschillende acties die op dit moment plaatsvinden wordt hieronder gegeven:

1. De camera neemt elke twee seconden een foto. De foto wordt verkleind in grootte (minder pixels) zodat de performantie van de app kan worden verhoogd.
2. Microsoft Face API past gezichtsdetectie toe op de foto.
 - (a) De FaceListAdapterklasse maakt een thumbnailbitmap van elk gedetecteerd gezicht dankzij de coördinaten die elk gezicht positioneren.
 - (b) Microsoft Face API past gezichtsherkenning toe om elke gedetecteerde persoon te identificeren. Personen die initieel nog niet aanwezig waren, worden nu ook aangemeld en met hun impliciete feedback wordt vanaf nu ook rekening gehouden.
 - (c) Een HistoryDatarecord wordt aangemaakt in de lokale databank voor de geïdentificeerde persoon en de video die hij aan het bekijken is.
 - (d) De data voor de emoties en bedekkingen in het gezicht, verkregen uit de gezichtsdetectie van Microsoft, wordt ingevuld in het aangemaakte HistoryDatarecord.
3. Kairos en Face++ worden samen opgeroepen op elke thumbnailbitmap en hun methode krijgt het ID mee van het aangemaakte HistoryDatarecord.
 - (a) Kairos zorgt voor de vereiste opslag van de bitmap. Kairos vereist namelijk dat de foto lokaal aanwezig is op het toestel voor de file upload.
 - (b) Kairos en Face++ vullen hun gedetecteerde features asynchroon in in het HistoryDatarecord aan de hand van het meegekregen ID. Kairos vult op deze manier asynchroon het HistoryDatarecord aan met het aandachtsniveau. Face++ vult asynchroon het record aan met de waarden van de gesloten ogen en de hoofdpositie. Op deze manier vullen de drie services elk met hun eigen waarden hetzelfde databankrecord asynchroon aan, vertrouwend op het ID gegenereerd vlak na de identificatie van Microsoft.

4. Wanneer het filmfragment gestopt wordt, wordt de geschatte impliciete feedback berekend door de klasse HistoryProcessor. De HistoryProcessorklasse berekent deze feedback op basis van alle HistoryDatarecords die zojuist ingevuld werden met de resultaten van de verschillende services. De berekende feedbackscore wordt in het bijhorende ViewHistorydatarecord gezet. Listing C.3 en Listing C.4 bevatten een stuk code van de methode updateCalculatedViewHistory van de HistoryProcessorklasse die alle HistoryDatarecords overloopt.

Zoals hierboven te zien, wordt het aantal pixels van de foto vermindert. Hoeveel de foto wordt verkleind is een afweging tussen performantie en accuraatheid van de detectie. Bij een te sterk verkleinde foto zal de gezichtsdetectie niet meer nauwkeurig werken en indien de foto niet verkleind wordt, duurt het opsturen van de foto naar de detectieservices aanzienlijk langer. De frequentie van elke twee seconden wordt vooral bepaald door het feit dat er bij elke foto zowel gezichtsdetectie en gezichtsherkenning van Microsoft moet worden uitgevoerd, gevolgd door gezichtsdetectie van Face++ en Kairos. Alles gebeurt in asynchrone taken maar de applicatie kan slechts een maximum aantal asynchrone taken aan. Dit zorgt ervoor dat er niet te veel nieuwe taken gestart mogen worden alvorens de services hun antwoord hebben gegeven. Het verkleinen van de foto's zorgt ervoor dat de frequentie van elke twee seconden kan worden aangehouden doordat de antwoorden tijdig ontvangen worden.



Figuur 4.4: Stap 3: Het verzamelen van impliciete feedback met gezichtsdetectie

Het wordt ook duidelijk dat steeds alle data wordt opgeslagen, zowel in de vorm van de foto van elke gedetecteerde persoon als in de databank in de vorm van HistoryDatarecords met alle informatie die

de services gedetecteerd hebben. De databankinformatie kan ook worden uitgeschreven naar een CSV-bestand, wat verder wordt uitgelegd in [Bijlage B: Gebruikershandleiding](#). In Figuur 4.4 heeft de persoon met ID 1 weinig aandacht getoond en emoties getoond die niet sterk overeenkomen met de verwachte emoties. In zijn geschiedenis wordt dus bijgehouden dat hij een slechte feedback geeft bij deze film. De persoon met ID 17 heeft veel aandacht getoond en emoties die sterk overeenkomen met de verwachte emoties waardoor een positieve feedback in de geschiedenis wordt opgeslagen. De persoon met ID 8 kwam in het voorbeeld niet of te weinig in beeld tijdens het afspelen van de film. Bijgevolg wordt er van deze gebruiker geen geschiedenis opgeslagen.

Er kan samengevat worden dat de impliciete feedback gebaseerd is op drie grote componenten:

1. Komen de getoonde emoties overeen met de verwachte emoties?

De getoonde emoties van de kijkers worden geanalyseerd. De emoties kunnen wijzen op een positieve impliciete feedback indien er bijvoorbeeld bij een komedie veel blijheid aanwezig is (bijvoorbeeld 70 % van de tijd werd er minimum een probabiliteit van 0,70 gedetecteerd). Men kan dus zeggen dat de hoeveelheden van de getoonde emoties moeten overeenkomen met de hoeveelheden van de verwachte emoties. Schematisch kan de verwachte emotiefingerprint van een video voorgesteld worden zoals bovenaan in Figuur 4.4. De balken geven de verwachte hoeveelheid van elke emotie voor. De hoeveelheden per emotie zijn typerend voor de film (daarom de naam: ‘fingerprint’) en dienen als basis voor de vergelijking met de getoonde emoties. Deze informatie is belangrijk om de emoties van de gebruikers een betekenis te geven voor verdere aanbevelingen. Onderaan worden twee gedetecteerde emotiespectra getoond van twee kijkers samen met de berekende overeenkomst met de emotiefingerprint door de hoeveelheden met elkaar te vergelijken. Deze figuur is enkel illustratief en bevat hier slechts vijf emoties.

Wanneer enkel de hoeveelheden van de emoties bepaald en vergeleken worden, ontstaat het probleem dat er geen rekening wordt gehouden met de tijdstippen waarop de emoties plaatsvinden. Bij sommige video’s is het bijvoorbeeld belangrijk dat de kijker op tijdstip X lacht en op tijdstip Y verdriet toont. Om dit op te lossen wordt er in de code met probabiliteitswaarden per tijdstip gewerkt in de verwachte emotiefingerprints en niet met totale hoeveelheden zoals hierboven afgebeeld. De emotiefingerprint van een film bevat dus de verwachte hoeveelheden per emotie én de verwachte tijdstippen ervan. Verder in 4.4.1 wordt gedetailleerd uitgelegd hoe de overeenkomst tussen wat verwacht wordt en wat getoond wordt, wordt berekend in de code.

2. Wat is het aandachtsniveau van de gebruiker?

Het aandachtsniveau wordt bepaald door meerdere factoren. Allereerst vertelt de hoofdpositie van de persoon of het hoofd van de persoon gericht is naar het scherm. In deze applicatie wordt enkel gekeken naar de yaw- en pitch-richting aangezien enkel deze een indicatie geven van het aandachtsniveau. Indien de hoofdpositie in de yaw- en pitch-richting buiten bepaalde grenzen valt, kan besloten worden dat de persoon het hoofd zeker niet gericht heeft op het scherm en niet aan het kijken is. Dit resulteert in negatieve feedback op het aandachtsniveau. De marges voor de yaw-richting liggen op 30 graden en voor pitch is dit 20 graden, zoals reeds bepaald in Hoofdstuk 3.

Het probleem ontstaat dat de hoofdpositie enkel de richting van het hoofd aangeeft maar niet bepaalt of de ogen gericht zijn op het scherm. Dit wordt opgelost door software te gebruiken die de ogen van de kijker analyseert. Kairos voorziet *attention level*-detectie en kan bepalen of de persoon naar het scherm staart en hoeveel keer er geknipperd geweest is met de ogen in een video [38]. Hier wordt gekozen om enkel met foto’s te werken aangezien het gratis Kairos-account slechts een beperkt aantal transacties per minuut toelaat en het aantal transacties met video’s snel zou overschreden worden. Kairos geeft een waarde tussen 0 en 1 terug en de drempel wordt in de applicatie ingesteld op 0,5. Alles boven deze drempelwaarde wordt gezien als aandacht houden. Indien er ook wordt vastgesteld dat de persoon beide ogen gesloten heeft, dient deze informatie als derde bevestiging dat de persoon geen aandacht aan het houden is.

3. Wat is het gedrag en de houding van de gebruiker?

Er kan informatie over de beleving van de content worden afgeleid uit het gedrag van de kijkers. Vele mensen hebben de neiging hun gezicht te bedekken indien ze naar een film aan het kijken zijn die ze eng vinden. Als dit gedetecteerd wordt, kan er besloten worden dat de getoonde content niet als aangenaam werd beschouwd door de kijker en kan negatieve feedback hiervan worden afgeleid. Microsoft Face API voorziet detectie van bedekte ogen, mond of voorhoofd [40]. De bedekking van het voorhoofd wordt niet gebruikt omdat hierbij geen verband bestaat met de feedback over de video.

Face++ voorziet ook detectie of de ogen gesloten zijn met een zekere probabiteit [1]. Indien een persoon continu met de ogen dicht gedetecteerd werd, kan ervan uitgegaan worden dat hij de film te saai vindt en kan dit ook geregistreerd worden als negatieve feedback.

Elk filmfragment heeft een bepaalde lengte. Indien de kijkers niet gedurende de volledige lengte naar de film hebben gekeken, kan ervan uitgegaan worden dat ze een negatieve feedback geven aan die film. Wanneer een gebruiker weggaat tijdens het kijken of continu zijn volledige gezicht bedekt, wordt er van de gebruiker gedurende die tijd geen gezicht geregistreerd. De totale aanwezigheid van die persoon is dus minder groot en ook hier kan een negatieve feedback aan gelinkt worden.

4.4.1 Berekenen van de overeenkomst met de verwachte emoties

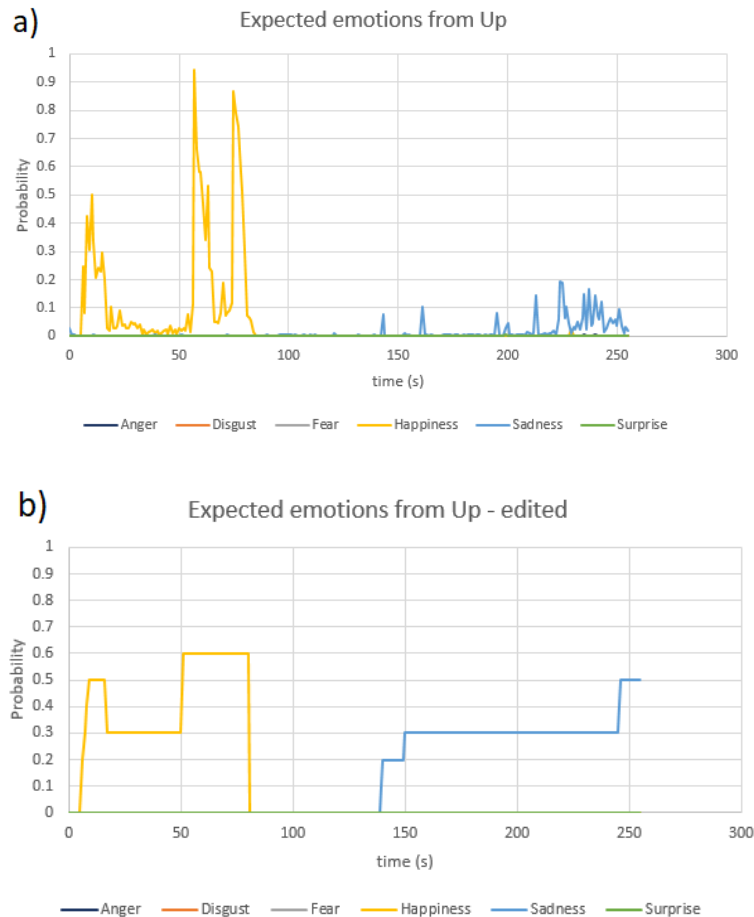
Nadat de gebruikers een film hebben bekeken, wordt ingeschat hoe leuk ze deze film vonden zodat er met deze waarderingsscore later een aanbeveling gegenereerd kan worden. Hiervoor moet er initieel een spectrum aan verwachte emoties, de emotiefingerprint, worden opgebouwd dat in de ExpectedEmotionstabel wordt geïnitieerd. Indien de ExpectedEmotionstabel voor een bepaalde film is ingevuld, kan de similariteit tussen de getoonde en de verwachte emoties berekend worden door de klasse HistoryProcessor. Per HistoryDatarecord wordt het overeenkomstig ExpectedEmotionsrecord emotie per emotie vergeleken, zoals getoond in Listing C.3 en Listing C.4.

Het is belangrijk dat het verwachte emotiespectrum bepaald werd per seconde van het filmfragment. Een film kan namelijk op verschillende tijdstippen verschillende emoties bevatten en het moet per tijdspanne geregistreerd worden of de getoonde emotie overeenkomt met de verwachte emotie. Men dient er rekening mee te houden dat het systeem bij het verzamelen van impliciete feedback (dat elke twee seconden de probabiteitswaarden bijhoudt) een seconde kan overslaan. Dit vormt geen probleem wanneer er per seconde een verwachte emotiewaarde voorzien is aangezien de frequentie van de verwachte emotiespectra hoger ligt dan die van de getoonde emotiespectra.

Opbouwen van de verwachte emoties

Er werden in de applicatie, gedurende de ontwikkeling van de app en het uitvoeren van de testen, verschillende reeksen verwachte emoties gemaakt (deze worden set 1, 2, 3 en 4 genoemd). Een eerste set werd gemaakt voor het starten van de testen en was gebaseerd op het zelf uiten van de emoties tijdens het bekijken van de zes testvideo's (door de PHOTO_MODUS-constante in te stellen, zoals uitgelegd in Hoofdstuk 5 en in Bijlage B: Gebruikershandleiding). Deze emotiespectra werden afgevlakt zodat ze geen te grote verschillen toonden. Een voorbeeld wordt gedemonstreerd in Figuur 4.5 met op de x-as de tijd in seconden en op de y-as de probabiteit van de emotie.

De bovenste grafiek in Figuur 4.5 bij a) zijn de emoties zoals ze werden waargenomen door de Microsoft Face API emotiedetectie. De onderste grafiek bij b) is de bewerking die manueel op de data is toegepast om de grafiek meer af te vlakken en emotiegebieden duidelijker te markeren. Beide grafieken tonen zowel blijheid in het begin als verdriet op het einde. De onderste grafiek benadrukt de droevige scène op het einde van het fragment. De droevige emoties werden wat verhoogd zodat kijkers enkel een grote similariteit hebben met de verwachte emoties als ze een duidelijke expressie van verdriet tonen.



Figuur 4.5: De emoties van video 2, een fragment uit de film ‘Up’: a) zoals ze gedetecteerd werden en b) zoals ze aangepast werden

Uit de evaluatie van de services leert men dat de emoties boosheid, angst en afkeer zelfs bij de meest expressieve foto’s nooit een probabilliteit van 100 % haalden. Daarom wordt er ook bij de verwachte emoties rekening gehouden dat de verwachte probabilliteit van deze emoties niet zo hoog mag liggen en ze worden bijgevolg standaard tussen ongeveer 0,3 en 0,8 gelegd indien de emotie verwacht wordt.

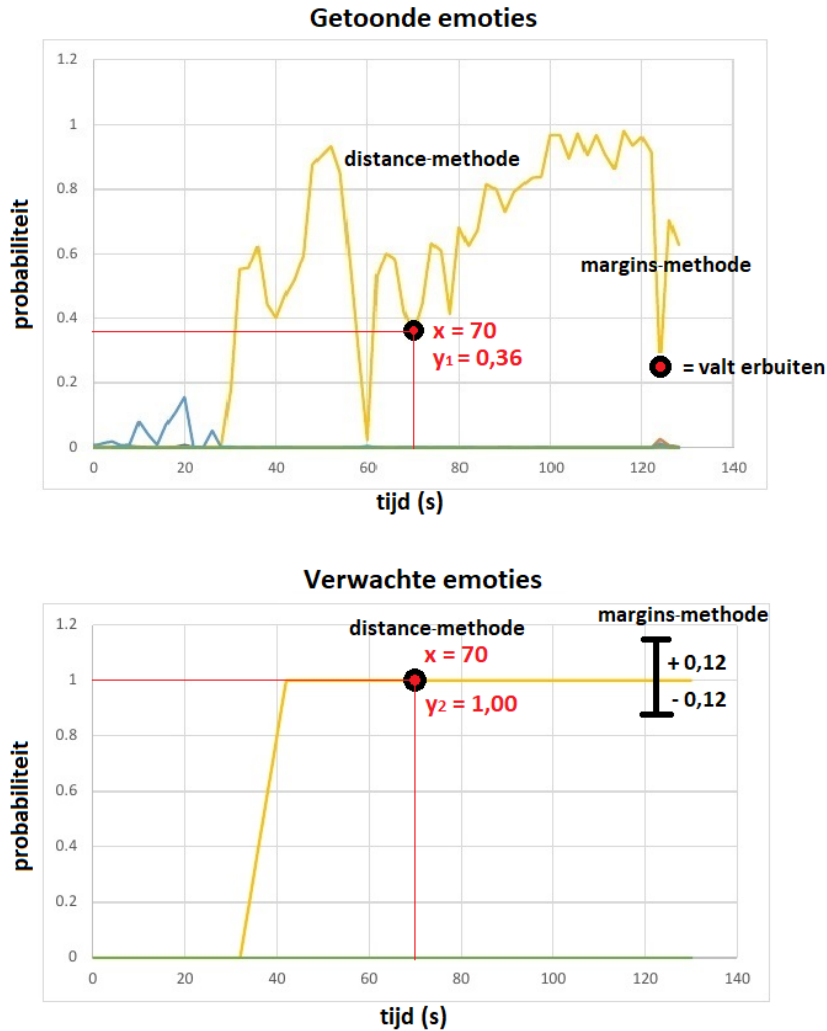
De eigen emoties als verwachte emoties gebruiken, houdt geen rekening met de gemiddelde emotie van verschillende kijkers. Elke kijker die de film leuk vindt, toont namelijk een andere emotie. Nadat de testen met de testpersonen zijn uitgevoerd, kunnen andere sets ontwikkeld worden op basis van de getoonde emoties van de testpersonen. Hoofdstuk 5 beschrijft hoe en waarom de verschillende sets zijn gemaakt en overloopt de resultaten wanneer ze worden toegepast op de getoonde emoties van de testpersonen, de problemen en de mogelijke oplossingen. De resultaten tonen aan hoe belangrijk de verwachte emoties zijn aangezien dit het enige is waarmee de getoonde emoties vergeleken worden.

Bepalen van de similariteit

Voor elk HistoryDatarecord in de databank is de relatieve tijd in het filmfragment gekend samen met de probabilliteit van elke emotie op dat moment. Elk record kan dus vergeleken worden met de waarden van de emoties van het ExpectedEmotionrecord op dat zelfde moment. Er moeten bijgevolg twee functies met elkaar worden vergeleken: de getoonde emoties op elk moment t en de verwachte emoties op t . Deze vergelijking wordt dan voor elk van de zes emoties uitgevoerd. De app voorziet twee methoden om de overeenkomst met de verwachte emoties te bepalen. De eerste methode in deze masterproef heet de distance-methode en is gebaseerd op het berekenen van de similariteit tussen twee functies. De tweede heet de margins-methode en is gebaseerd op het tellen van het aantal keer dat een emotie binnen de marges van de emotiefingerprint van de film valt. In de applicatie wordt vooral gewerkt met de

distance-methode en deze wordt uitgevoerd door de klasse HistoryProcessor. Een codevoorbeeld die de distance-methode illustreert wordt gegeven in Listing C.3 en Listing C.4.

Voor de distance-methode kunnen twee functies worden vergeleken door een similariteitswaarde te bepalen. Een geïllustreerd voorbeeld die twee spectra vergelijkt is te vinden in Figuur 4.6 waar de distance-methode staat aangeduid.



Figuur 4.6: Het vergelijken van de verwachte emoties met de getoonde emoties voor elk gedetecteerd tijdstip, gedemonstreerd op zowel de distance- als de margins-methode

Segaran beschrijft in zijn boek een methode om de similariteitscore te bepalen die gebruik maakt van de euclidische afstand, berekend voor elk van de punten [52]. De euclidische afstand tussen twee functies is hierin gelijk aan

$$sum = \sum_{i=0}^n (y_{1,i} - y_{2,i})^2. \quad (4.1)$$

Hierbij is:

- n = het aantal HistoryDatarecords van die persoon voor die film.
- $y_{1,i}$ = de getoonde emotieprobabiliteit van dat record (uit HistoryDatatablel)
- $y_{2,i}$ = de verwachte emotieprobabiliteit op dat tijdstip (uit ExpectedEmotionstabel)

De afstand, genaamd distance in de formule, is deze som, genomen voor alle punten n die in de getoonde emoties voorkomen. De n bijhorende punten uit de verwachte emoties worden hierbij vergeleken. Het is dus niet zo dat ook alle punten uit de verwachte emoties overlopen worden. Deze distance is verschillend voor elke kijkbeurt aangezien n telkens kan verschillen. Daarom wordt de som gedeeld door n alvorens hieruit de wortel wordt genomen om de distance te kennen:

$$distance = \sqrt{\frac{1}{n} sum}. \quad (4.2)$$

De similariteit wordt nu volgens Segaran berekend door volgende formule toe te passen:

$$similarity = \frac{1}{1 + distance} \quad (4.3)$$

waardoor men een getal krijgt tussen 0 en 1 [52]. Dit getal duidt dan aan hoe sterk de twee functies op elkaar lijken, waarbij 0 betekent dat ze volledig verschillen en de distance oneindig groot is. De maximale waarde van de distance is in deze applicatie gelijk aan 1 aangezien de probabiliteiten van elke emotie steeds tussen 0 en 1 liggen. Indien de distance gelijk is aan 1, wordt de similariteit met formule 4.3 gelijk aan 0,5. 0,5 is bijgevolg de minimale waarde van de similariteit. De similariteit is 1 als de distance gelijk is aan 0 en dit wilt zeggen dat de twee functies volledig hetzelfde zijn. Voor de zes emoties samen krijgt men op die manier steeds een distance-waarde tussen 3 en 6. Later in de applicatie worden nog andere factoren hierbij opgeteld, zoals het aandachtsniveau. De totale geschatte score die gebruik maakt van de distance-methode wordt in de ViewHistorytabel opgeslagen onder de naam similarityDistance.

Voor de margins-methode wordt er telkens gekeken of de getoonde emoties binnen boven- en ondergrenzen vallen van de verwachte emoties, zoals geïllustreerd in Figuur 4.6 waar de margins-methode staat aangegeven. Deze boven- en ondergrenzen worden berekend door telkens een margins-waarde bij de verwachte waarde op te tellen of af te trekken. Om een persoonlijk aspect hierin op te nemen en niet te werken met vaste marges, wordt eerst per gebruiker zijn gemiddelde emotie bepaald waarna met de formule

$$margin = \frac{1 - average}{5} \quad (4.4)$$

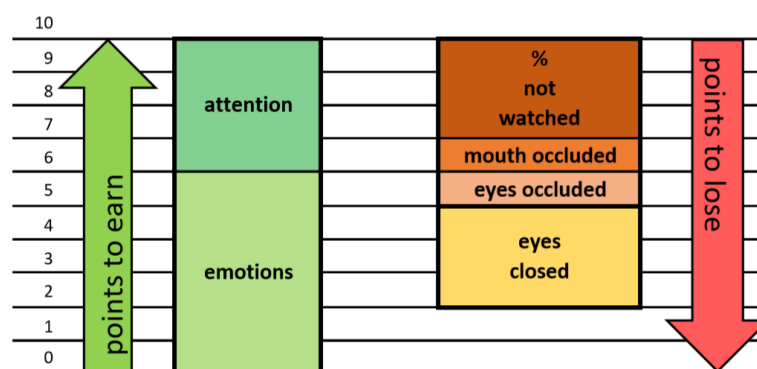
wordt bepaald wat de marge moet zijn van deze emotie. Formule 4.4 zorgt ervoor dat de gemiddelde waarde, dat een getal is tussen 0 en 1, wordt afgebeeld op hoe goed die persoon emoties kan uiten. Op deze manier krijgt de persoon een persoonlijke marge voor die film op basis van hoe extravert of introvert de persoon is. Indien bijvoorbeeld de gemiddelde waarde van blijheid van een extraverte persoon gelijk is aan 0,5, wordt de margins-waarde gelijk aan 0,1. Dit is een kleine marge dus een extraverte persoon behaalt minder snel de verwachte emoties binnen de marges dan een introverte persoon. Een introverte persoon heeft bijvoorbeeld bij die film een gemiddelde blijheidswaarde van 0,1 waardoor de margins-waarde gelijk is aan 0,18, wat al een veel grotere marge is. De bedoeling hiervan is om een minder extraverte persoon ook een kans te geven te voldoen aan de verwachte emoties en milder te zijn voor zijn similariteitsberekening. Bovendien zorgt deze methode ervoor dat het systeem milder omgaat met emoties die moeilijker te detecteren zijn, zoals boosheid, afkeer en angst, besloten in Hoofdstuk 3. De berekende waarde zal opnieuw een waarde tussen 0 en 6 zijn en hierbij worden later nog andere factoren bijgeteld. De geschatte score op basis van de margins-methode wordt in de ViewHistorydatabank opgeslagen per kijkbeurt onder de naam similarityMargins. De berekening van de totale score wordt hieronder verder uitgelegd.

4.4.2 Geschatte score van de film op basis van de impliciete feedback

De uiteindelijke app maakt een schatting van hoe goed de persoon de film vond. Dit doet hij door de zes similariteiten van elke emotie op te tellen en zo een getal te verkrijgen met een maximale waarde van zes. De app streeft naar een geschatte score op tien, dus er kunnen nog vier punten verdiend worden als een persoon de volledige aandacht behouden heeft. Een HistoryDatarecord waarin een persoon de ogen open heeft, een hoofdpositie binnen de grenzen en Kairos een aandachtsinschatting van groter dan 0,5 heeft gemaakt, voldoet aan de eigenschap attention. Listing C.4 toont deze berekening vanaf lijn 29 en roept op

het einde de methode `calculateSimilarityFromDB` op die de aantallen verwerkt: alle `HistoryDatarecords` die voldoen aan het aandachtsniveau worden opgeteld en gedeeld door het totale aantal records van de persoon bij die film. Dit zal een getal zijn tussen 0 en 1 en wordt vermenigvuldigd met vier. Hoe meer records er geregistreerd zijn waar de persoon aandachtig is, hoe hoger dit getal op vier. De emotiescore op zes en de aandachtsscore op vier worden opgeteld om een getal op tien te verkrijgen. De maximale impliciete feedback score die een gebruiker op een film kan krijgen is dus tien indien deze volledig voldoet aan de verwachte emoties, continu de aandacht houdt en geen strafpunten door negatieve feedback verkrijgt. Een persoon kan ook een negatieve score krijgen indien de positieve feedback kleiner is dan de negatieve feedback. Hoe de negatieve feedback bepaald wordt, wordt verder uitgelegd.

Figuur 4.7 toont hoe de zes emoties een score op zes vormen. De score wordt bepaald door de som te nemen van alle similariteiten van de zes emoties en elk van deze similariteiten te vermenigvuldigen met een factor om de typerende emotie meer te laten doorwegen. Er zijn drie modi beschikbaar, verder uitgelegd in [Bijlage B: Gebruikershandleiding](#). Zo is er een modus om alle emoties evenveel te laten meetellen. Vervolgens is er een modus om enkel de meest typerende emotie(s) het gehele gewicht te geven, dit is bijvoorbeeld het gewicht zes indien er één typerende emotie is en het gewicht drie indien er twee typerende emoties zijn. En tenslotte is er een modus om de meest typerende emotie(s) een groter gewicht te geven dan de rest, bijvoorbeeld 2 voor de typerende emotie en de andere 0,8 (wat op een totaal van zes uitkomt). Figuur 4.7 toont ook hoe een score op 10 kan behaald worden door de aandachtspunten op vier erbij te tellen.



Figuur 4.7: De punten die te verdienen zijn op basis van de impliciete feedback samen met de punten die te verliezen zijn

Negatieve feedback, die men van de totale score op 10 punten kan aftrekken, wordt ook geregistreerd in de vorm van bedekken van de ogen, bedekken van de mond en het sluiten van de ogen. Het aantal records waarin één van deze eigenschappen gedetecteerd werd, wordt bepaald door de `HistoryProcessor`klasse en gedemonstreerd in [Listing C.4](#) vanaf lijn 48. De `HistoryProcessor`klasse roept vervolgens de methode `calculateSimilarityFromDB` op waarin het aantal van elke gedragsfactor wordt gedeeld door het totaal aantal records van deze persoon voor de film (en vermenigvuldigd met een gewicht). Het niet volledig bekijken van de film wordt ook als negatieve feedback gezien. Dit wordt in een getal gegoten door het aantal ontbrekende records te delen met het aantal verwachte records en dit getal te vermenigvuldigen met een gewicht. De gewichten duiden op de maximale te verliezen score, net zoals de aandachtspunten op vier stonden, en worden voor de negatieve feedback gevisualiseerd in [Figuur 4.7](#) aan de rechterkant. Indien bijvoorbeeld een persoon continu de ogen gesloten heeft, bedraagt het percentage van gesloten ogen 100 % en wordt dit vermenigvuldigd met 3, wat uitkomt op een totaal van 3 strafpunten. Wanneer een persoon gedurende 50 % van de tijd de ogen gesloten heeft, komt dit uit op een totaal van 1,5 strafpunten. De gewichten zijn in te stellen door de waarden van de constanten in te stellen. Op deze manier wordt de inschatting van de score meer of minder door de verschillende componenten beïnvloed. Momenteel staan het bedekken van de ogen en mond op elk één strafpunt, het gesloten houden van de ogen op een maximale straf van 3 en het niet volledig bekijken van de film ook op 3.

De som van deze negatieve feedbackwaarden wordt in [Listing 4.2](#) bijgehouden in de variabele `shared-Score` samen met de `attentionScore`. De listing toont een stuk code van de methode `calculateSimi-`

larityFromDB van de HistoryProcessorklasse die instaat voor het bepalen van de geschatte score van de kijker. De som van de negatieve waarden wordt afgetrokken van de totale score om zo een inschatting te krijgen van de belevenis van de content van de gebruiker. Van de tien punten die te verdienen zijn met het uiten van de juiste emoties en met het houden van de aandacht, zijn er dus ook weer acht punten te verliezen om uiteindelijk een totale score op tien te krijgen.

```

double notWatched = ((totalMovie - total)*1.0/totalMovie)*PENALTYNOTFULLYWATCHED; 1
if(notWatched >= BOUNDARY) notWatched = 0; 2
sharedScore += notWatched; 3

double percEyeOccluded = history.getEyeOccluded()/total*1.0; 4
sharedScore += (PENALTYOCCLUSION*percEyeOccluded); 5
double mouthOccluded = history.getMouthOccluded()/total*1.0; 6
sharedScore += (PENALTYOCCLUSION*mouthOccluded); 7

double eyesClosed = history.getEyesClosed()/total*1.0; 8
if(eyesClosed >= BOUNDARY) eyesClosed = 0; 9
sharedScore += (PENALTYEYESCLOSED*eyesClosed); 10

```

Listing 4.2: Codevoorbeeld van de HistoryProcessorklasse: berekenen van negatieve feedback

Alle deelscores, emotionScore, typicalEmotionScore, attentionScore en sharedScore, worden apart opgeslagen in de databank in een ViewHistoryrecord. De totale score, in de app bepaald door de som van emotionScore en sharedScore, wordt ook opgeslagen onder scoreDistance. typicalEmotionScore wordt momenteel enkel gebruikt om te vergelijken bij de testen welke som het beste resultaat geeft. Indien gewenst, kan per ViewHistoryrecord een nieuwe distanceScore berekend worden door typicalEmotionScore en sharedScore op te tellen. De scoreMargins wordt op analoge wijze bepaald door sharedScore op te tellen bij de berekende margins-score op zes. Hoofdstuk 5 geeft in Tabel 5.2 een overzicht van al deze scores om de resultaten te kunnen begrijpen.

4.4.3 Uitzonderingen en speciale gevallen

Stap drie kan ook om met afwijkingen in de samenstelling van mensen die aanwezig zijn bij het kijken naar de film. Mensen die initieel bij de eerste startfoto aanwezig waren bij het inloggen, kunnen tijdens het kijken van de film verdwijnen. Er moet nu bepaald worden of deze mensen de film dan bekeken hebben en of ze zijn weggegaan omdat ze de film niet leuk vonden. Deze informatie zegt veel over de feedback die de gebruiker aan de film geeft. Indien de persoon de film niet heeft bekeken, mag er eigenlijk geen feedback worden opgeslagen en mag het systeem niet denken dat de persoon de film al gezien heeft. De film moet namelijk nog steeds in aanmerking komen voor de toegelaten films en ongeziene films van die gebruiker. Dit wordt opgelost door in RecommendationCalculator een methode (deleteLowPhotoHistory) te voorzien die de ViewHistoryrecords verwijdert indien het aantal foto's ervan kleiner is dan een bepaald getal, bijvoorbeeld bij registratie van slechts 5 foto's (momenteel ook zo ingesteld in de applicatie). Indien de persoon de film wel bekeken heeft en bijvoorbeeld in de helft is weggegaan omdat hij het geen leuke film vond, moet deze film wel worden opgeslagen in de feedback van de gebruiker, namelijk als negatieve feedback. Dit wordt opgelost door het aantal foto's te vergelijken met het verwachte totale aantal foto's bij de film en het verschil te vermenigvuldigen met strafpunten die worden toegekend aan de score van die persoon.

Het is ook mogelijk dat er mensen zijn bijgekomen tijdens het kijken van de film die initieel bij de startfoto niet aanwezig waren. Aangezien het systeem tijdens het kijken van de film steeds identificatie uitvoert van alle aanwezig personen, worden nieuw aanwezige personen ook herkend. Indien ze lang genoeg aanwezig waren, kan het geregistreerd worden dat de persoon de film ook heeft bekeken met zijn bijhorende impliciete feedback op basis van zijn getoonde emoties en aandachtsniveau.

4.5 Updaten en trainen van gebruikers

De app werkt op basis van gezichtsherkenning en indien de gezichtsherkenning van Microsoft faalt, ontstaat het probleem dat de app de persoon niet correct herkent maar bijvoorbeeld een bestaande persoon

aanmaakt als een nieuwe gebruiker. Hierdoor kan de app niet de persoonlijke aanbeveling genereren of persoonlijke geschiedenis gaan opslaan van de juiste gebruiker. Dit probleem wordt opgelost door de Face Storage van Microsoft extra te trainen met meerdere gezichten van eenzelfde persoon. Telkens een gebruiker zich opnieuw aanmeldt, wordt er een foto genomen voor identificatie maar ook voor een extra training. Deze foto wordt toegevoegd aan de Face Storage van deze gebruiker waarna deze getraind wordt. Dit is nodig opdat het systeem zo de gebruiker beter en beter leert “kennen”. Hierdoor zal de persoon in de toekomst sneller correct herkend worden, ook in variërende omstandigheden zoals andere gezichtsuitdrukking of andere belichting. Ook wanneer gebruikers ouder worden, is dit interessant. Het gezicht van kinderen verandert snel en op deze manier kan het systeem hier langzaamaan mee rekening houden.

Het probleem kan zich ook voordoen dat de leeftijd en het geslacht van een gebruiker bij het eerste gebruik verkeerd geschat werden en de app met de fout blijft verder werken. Dit wordt opgelost door bij elke nieuwe aanmelding ook opnieuw geslachts- en leeftijdsherkenning op de foto toe te passen zodat de leeftijd en het geslacht van de persoon kunnen worden aangepast. De geüpdatet leeftijd zal voor 90 % uit de oude leeftijd bestaan en voor 10 % uit de net geschatte leeftijd. De leeftijd van de persoon in de databank zal nu geüpdatet worden naar de nieuwe samengestelde leeftijd. De app blijft dus nooit bij zijn eerste inschatting van leeftijd en geslacht maar past echt continu de profielen van zijn gebruikers aan. Hierdoor wordt er rekening gehouden met het falen van sommige gezichtsdetectie en worden steeds nauwkeurige aanbevelingen gegenereerd, inclusief de correcte onaanvaardbare films. Opnieuw is dit interessant voor kinderen omdat hun leeftijd wordt aangepast naarmate ze ouder worden. Hierdoor zal het systeem hen op de juiste leeftijd films kunnen aanraden die ze enkele jaren ervoor nog niet mochten bekijken wegens hun minimumleeftijd. Bij kinderen komt het ook vaak voor dat hun geslacht foutief bepaald werd. Dankzij de update van het geslacht, door de drie services opnieuw te laten stemmen voor het geslacht, krijgt de persoon uiteindelijk het juiste geslacht toegekend.

Ook bij de eerste aanmelding van een nieuwe gebruiker worden de leeftijd en geslacht geschat en wordt het gezicht getraind. Het wordt aangeraden om een nieuwe gebruiker 5 à 6 keer op de startknop te laten drukken zodat er meerdere foto's genomen worden. Op die manier kunnen er verschillende foto's opgestuurd worden voor een extra inschatting van leeftijd en geslacht en training van het gezicht meteen vanaf het eerste gebruik.

De Face Storage heeft een beperkte capaciteit voor het gratis account. Het betalende account kan tot één miljoen personen in een Face Group opslaan maar rekent 0,5 dollar per 1000 gezichten in de opgeslagen afbeeldingen aan per dag [5]. Bovendien is het aangewezen de Face Group niet te overtrainen en oude foto's te verwijderen (bijvoorbeeld kinderen die er na enkele jaren helemaal anders uitzien). Hiervoor is het *persistend face ID* nodig van de opgeslagen foto en gezicht. Deze ID's worden toegekend door Microsoft Face API aan de opgeslagen gezichten en de app houdt deze nog eens extra bij in de databank, gelinkt aan de gebruiker van dat gezicht. Zo heeft de app steeds toegang tot dit ID om het gezicht later te kunnen verwijderen. De Facestabel houdt deze ID's bij en wanneer het aantal persistent face-ID's van een gebruiker boven een bepaalde limiet zit, wordt het oudste gezicht zowel uit de databank als uit Face Storage zelf verwijderd.

Figuur 3.7 op het einde van Hoofdstuk 3 toont de samenwerking van de verschillende services doorheen het gebruik van de applicatie, met in het begin ook de training en update van de gebruikers. De training en update gebeuren telkens wanneer er op de startknop gedruwd wordt. Het valt buiten de drie grote fasen en is niet noodzakelijk voor het systeem maar zorgt er wel voor dat de aanbeveling nauwkeuriger kan worden berekend en de gezichtsherkenning beter kan worden uitgevoerd.

4.6 Architectuur

Hieronder wordt de architectuur van de applicatie beschreven met verschillende diagrammen en modellen zoals een activiteitendiagram, een databankmodel, een UML-diagram en sequentiediagrammen. Dit deel bevat de technische details over hoe de code in elkaar zit. Hoe er met de app moet gewerkt worden en hoe de code kan worden aangepast, wordt beschreven in [Bijlage B: Gebruikershandleiding](#).

4.6.1 Activiteitendiagram

Figuur C.7 in [Bijlage C: Grafieken, diagrammen, code en tabellen](#) toont het activiteitendiagram van de applicatie. Bij de start moet er eerst een foto worden genomen om gezichtsherkenning toe te kunnen passen op alle aanwezige personen. Gezichtsdetectie zoekt alle gezichten in de foto en gezichtsherkenning kijkt dan of de gezichten reeds gekend zijn binnen het systeem. Als de persoon reeds gekend is, identificeert de applicatie de persoon en meldt deze aan. Vervolgens update en hertraint de applicatie deze persoon ook. Als er zich een nieuwe persoon voor de camera bevindt, wordt zijn leeftijd en geslacht door de drie services bepaald en berekend. Vervolgens wordt deze persoon ook toegevoegd en aangemeld.

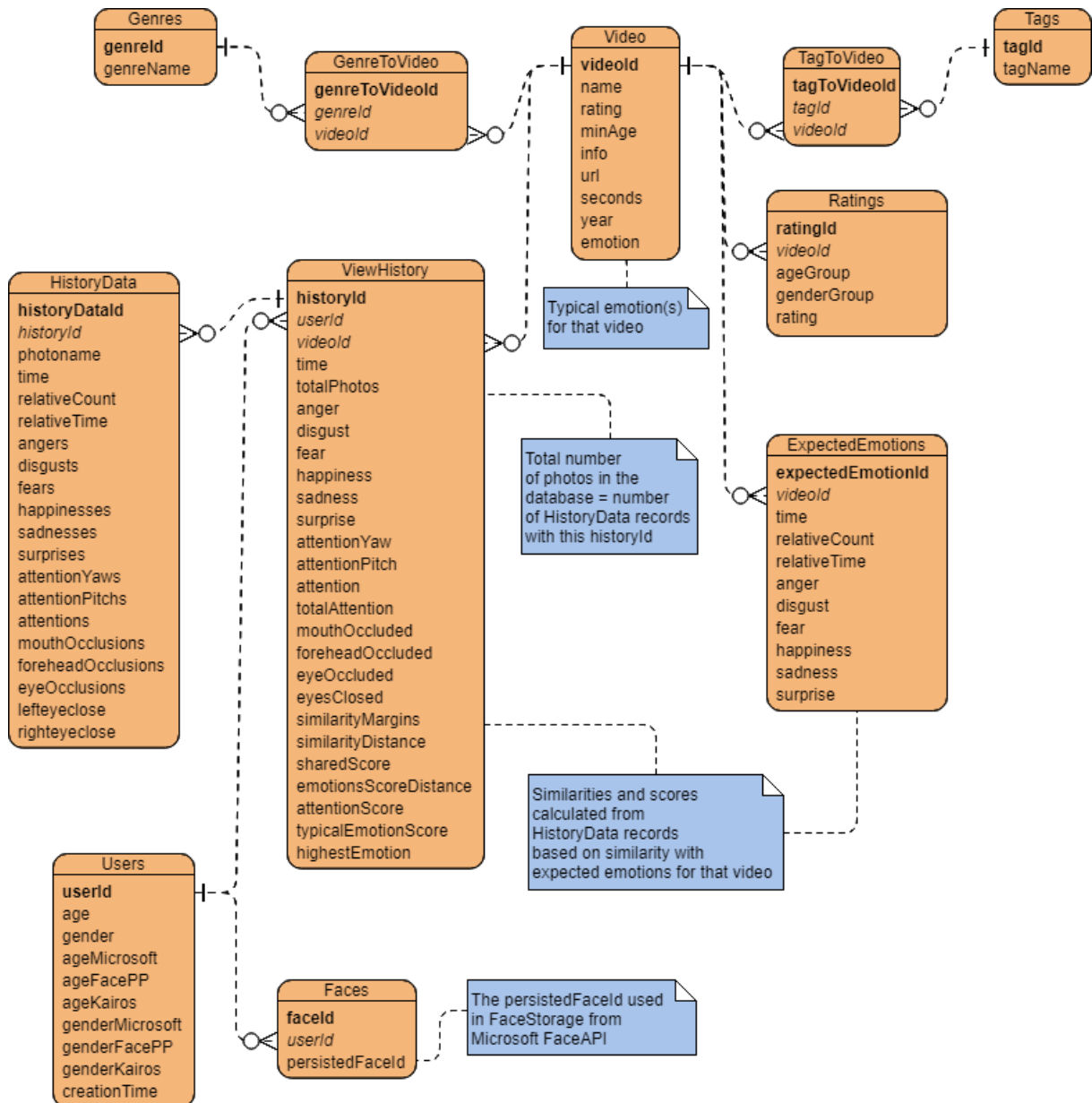
Nu alle aanwezige personen gekend zijn, kan de aanbeveling berekend worden. De personen kiezen een film en terwijl deze film speelt, wordt er om de twee seconden een foto genomen van de kijkers. Telkens wanneer er een foto genomen wordt, wordt er per persoon bepaald wie de persoon is (identificatie), wat zijn emoties, gedrag en aandachtsniveau is en wordt deze persoonlijk impliciete feedback opgeslagen in het systeem. Personen die nog niet aangemeld waren, worden nu ook aangemeld zodat er bij een eventuele nieuwe aanbeveling ook rekening kan gehouden worden met deze gebruikers. Tenslotte, wanneer de film gedaan is, wordt de applicatie afgesloten, kunnen de kijkers een nieuwe aanbeveling krijgen als ze een nieuwe film willen bekijken of kunnen ze de ingelogde personen opnieuw synchroniseren door een nieuwe foto te trekken en de nieuw aanwezige mensen zo ook in te loggen.

4.6.2 Databankmodel voor de Roomdatabank

In Figuur 4.8 worden de verschillende tabellen van de databank voorgesteld met hun kolommen en relaties. Zoals hierin te zien, heeft een video verbindingen met meerdere tags en genres. De genres dienen om in de app een overzicht van de filmgenres te kunnen geven wanneer de gebruiker een informatie-overzicht over de geselecteerde film opvraagt. De tags bevatten zowel genres als keywords die eigen zijn aan de film en dienen enkel voor het linken van gelijkaardige films bij het maken van een aanbeveling. Dit werd reeds uitgelegd in 4.3.2. Een video heeft ook per seconde een ExpectedEmotionrecord die de verwachte hoeveelheid boosheid, afkeer, angst, blijheid, verdriet en verrassing beschrijft. De demografische ratings van een film worden voorgesteld door een tabel Ratings die voor elke demografische groep, bepaald door leeftijdsgroep (ageGroup) en geslacht (genderGroup), een bepaalde ratingwaarde heeft (rating). De tabel wordt gebruikt om het cold start probleem op te lossen.

De gebruikers worden opgeslagen in de Userstabel samen met hun geschatte geslacht en leeftijd, zowel geaggregeerd als apart, en hun aanmaaktijdstip. De persistent face-ID's die gebruikt worden in Face Storage van Microsoft Face API worden bijgehouden en gelinkt aan de Userstabel in de Facetabel. Omdat de persistent face-ID's worden bijgehouden is het mogelijk om (de oudste) gezichten van een persoon te verwijderen uit de Face Storage gedurende het gebruik van de app.

De ViewHistorytabel houdt een record bij voor elke gebruiker die een film bekijkt. Er bestaat dus per combinatie gebruiker en film maximum slechts één record. Indien een gebruiker een film twee keer bekijkt (wat enkel kan gebeuren als de gebruiker niet aanwezig was bij het aanmelden, aangezien de aanbeveling rekening houdt met welke film de gebruiker al heeft gezien, of wanneer hij een film expliciet bekijkt via de 'Show all videos'-knop), komt de extra informatie onder hetzelfde ViewHistoryrecord en ViewHistory-ID. Elk ViewHistoryrecord heeft meerdere HistoryDatarecords. Deze HistoryDatarecords houden per foto het tijdstip bij samen met de getoonde probabiliteiten van de emotiewaarden, headposerichtingen, gezichtsbedekkingen en geslotenheid van de ogen. De emotiewaarden worden uitgedrukt in een kommagetal tussen 0 en 1 dat de probabieliteit van elk van de zes basisemoties aanduidt. De similariteitswaarden in ViewHistory worden berekend op deze waarden en doordat de HistoryDatarecords worden bijgehouden, kan het systeem de similariteitswaarden steeds opnieuw berekenen indien drempelwaarden, expected emoties of de records zelf worden aangepast. Hoe dit gerealiseerd kan worden, wordt uitgelegd in [Bijlage B: Gebruikershandleiding](#). Op deze manier kunnen alle data opnieuw verwerkt worden door het systeem, wat een belangrijk element is bij de testen.



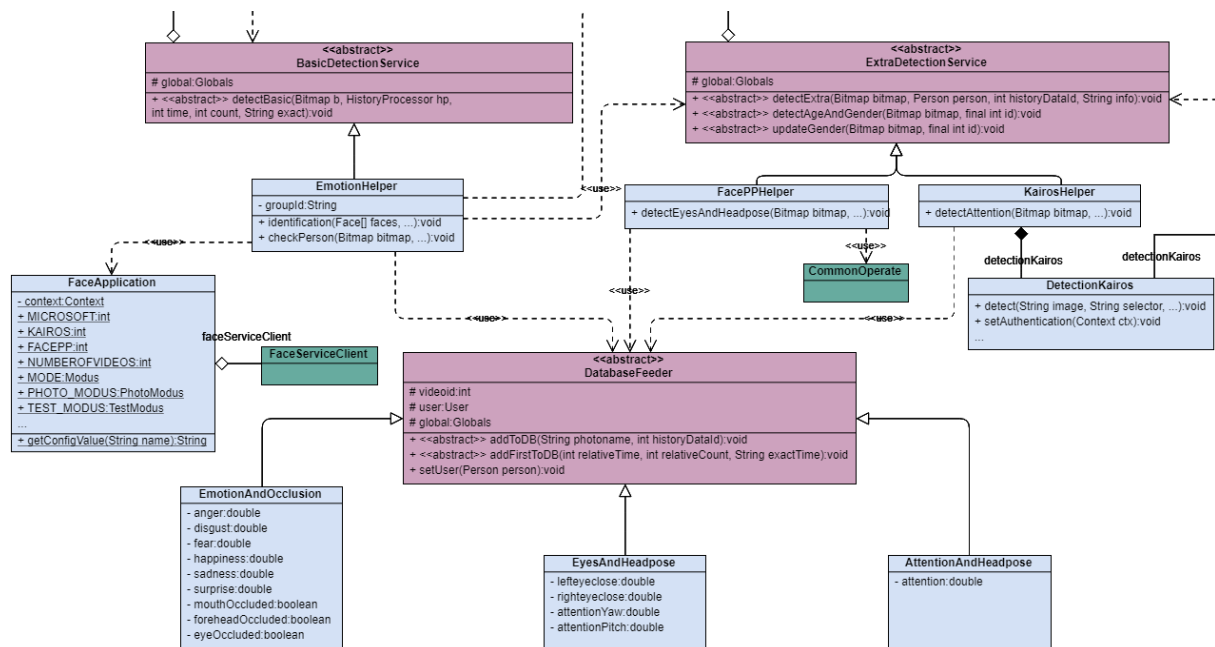
Figuur 4.8: Databankmodel van de applicatie

4.6.3 UML- of klassendiagram

De volledige applicatie is ingedeeld zoals te zien op het UML-diagram in [Bijlage C: Grafieken, diagrammen, code en tabellen](#). Het diagram toont alle klassen die de applicatie bevat en ook enkele klassen die gebruikt worden maar niet zelf geïmplementeerd werden. Deze klassen worden aangeduid met een donkerdere kleur. Om overzichtelijk te blijven, bevat het UML-diagram niet alle constructors, getters, setters en constanten. Ook de dataklassen worden niet gelinkt omdat dit reeds duidelijk is met het databankmodel dat de relaties ertussen uitlegt. De lifecyclemethoden van de Activityklassen en alle methoden van de databankklassen worden ook niet getoond omdat de bedoeling van dit UML-diagram vooral is om de relatie tussen alle klassen duidelijk te maken. De belangrijkste methoden worden vermeld en alle statische methoden en variabelen worden onderlijnd weergegeven.

Figuur 4.9 toont een deel van het volledige UML-diagram. Dit deel behandelt de drie services en hoe ze samenwerken doorheen de applicatie, zowel bij het verzamelen van impliciete feedback in stap 3 als bij het schatten van geslacht en leeftijd in stap 1. Aangezien Microsoft steeds als basis gebruikt wordt

(omdat Microsoft zorgt voor de noodzakelijke identificatiestap bij elke foto), wordt er in het antwoord van Microsoft steeds de eerste stap gezet tot het maken van geschiedenisrecords. EmotionHelper, de klasse die zorgt voor connectie met de Microsoftservice, is afgeleid van BasicDetectionService hetgeen instaat voor het uitvoeren van de basis, namelijk het detecteren wie de persoon is en het aanmaken van een HistoryDatarecord met de initiële informatie zoals het tijdstip. Microsoft voorziet ook de emotie en de bedekking van gezichtsdelten. Deze kenmerken worden meteen ook opgevraagd bij de basisdetectie met de methode detectBasic. In detectBasic worden asynchrone methoden aangeroepen voor de identificatie en detectie en wanneer deze een antwoord ontvangen, wordt een EmotionAndOcclusionobject aangemaakt. Dit object is overgeërfd van DatabaseFeeder en staat in voor het verzamelen van de info van Microsoft en het wegschrijven naar de databank. Analooft doen EyesAndHeadpose en AttentionAndHeadpose dit voor respectievelijk Face++ en Kairos wanneer zij aan de beurt zijn. De methode addFirstToDB van EmotionAndOcclusion wordt opgeroepen hetgeen zorgt voor het aanmaken van het HistoryDatarecord en het teruggeven van het aangemaakte HistoryData-ID. Hierna kunnen Kairos en Face++ aan dit record hun informatie gaan toevoegen. Om dit te doen, worden FacePPHelper en KairosHelper opgeroepen, die beiden afgeleid zijn van ExtraDetectionService, wat instaat voor het detecteren van “extra” features in de foto. FacePPHelper en KairosHelper maken beide ook een DatabaseFeederobject aan, respectievelijk EyesAndHeadpose en AttentionAndHeadpose en de methode addToDB wordt opgeroepen om de gedetecteerde informatie weg te schrijven naar de databank.



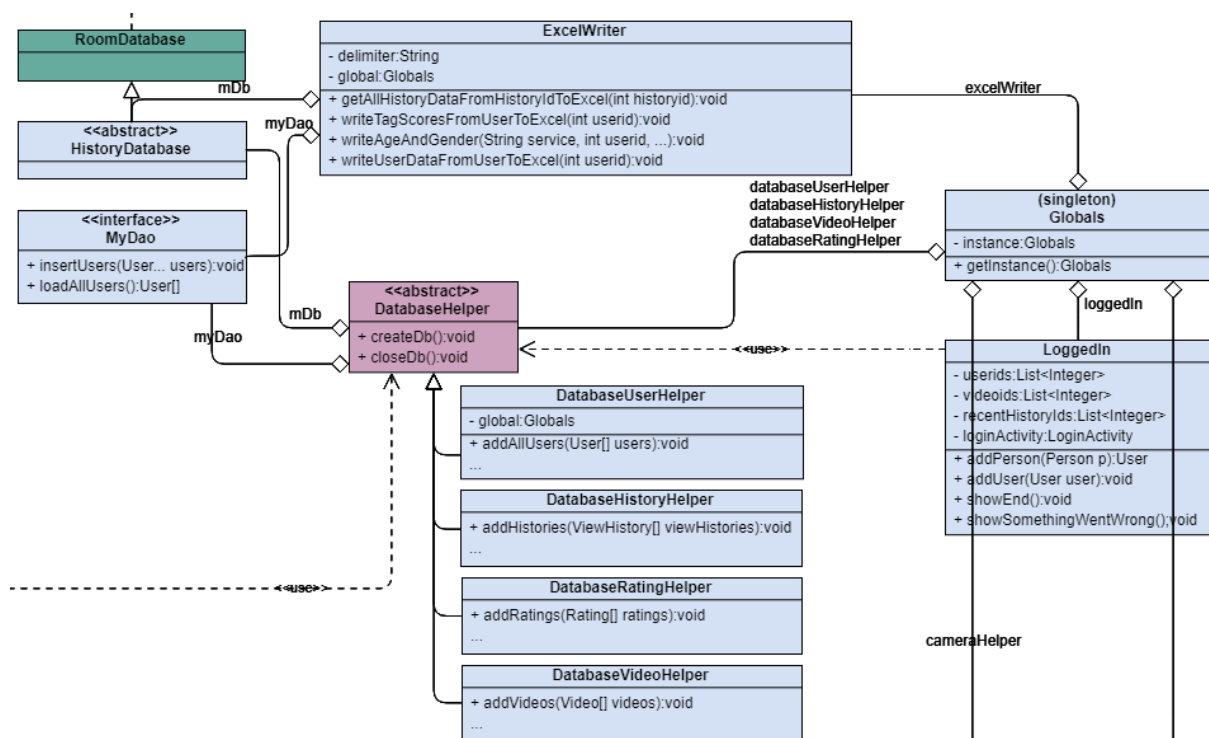
Figuur 4.9: Deel van het UML-diagram waar de services worden opgeroepen

Het pattern dat hierin te herkennen is, is het Abstract Factory pattern [53]. De drie services (EmotionHelper, FacePPHelper en KairosHelper) produceren elk een object van het type DatabaseFeeder, afhankelijk van wat zij detecteren en wat voor output de services produceren. De drie services zijn elk afzonderlijk afhankelijk van hun eigen hulpklassen om de detectierequests te versturen. Zo gebruikt EmotionHelper van Microsoft de FaceServiceClientklasse dat wordt bijgehouden in de globale FaceApplicationklasse. Face++ gebruikt CommonOperate, een klasse beschikbaar door de geïmporteerde bibliotheek, beschreven in [Bijlage A: Installatiehandleiding](#). Kairos gebruikt de klasse DetectionKairos om de HTTP request te versturen.

De FaceApplicationklasse is een klasse afgeleid van Application, de basisklasse die de global application state bijhoudt. Een instantie van deze klasse wordt gemaakt vooraleer een instantie van elke andere klasse wordt aangemaakt [54]. In dit project worden hier enkele globale constanten bijgehouden waar meerdere klassen gebruik van kunnen maken, zoals de verschillende modi (PHOTO_MODUS, TEST_MODUS ...) waarin de applicatie kan gedraaid worden. Over de verschillende modi wordt meer uitgelegd in [Bijlage B: Gebruikershandleiding](#).

Een ander deel van het UML-diagram wordt hieronder in Figuur 4.10 getoond. De klasse Globals is een singleton en er kan dus maar één object van worden aangemaakt. De klasse staat in voor het bijhouden van alle objecten die globaal gedeeld worden onder de verschillende klassen, zoals bijvoorbeeld de LoggedIn-instantie en ExcelWriter-instantie. De Androiddocumentatie zegt dat het niet steeds nodig is af te leiden van de Applicationklasse maar dat voor sommige variabelen een statische singleton dezelfde functie uitoefent maar op een meer modulaire manier [54]. De singleton Globals wordt hier gebruikt om alles wat te maken heeft met databanktoegang en ingelogde mensen te centraliseren. Alle klassen in de applicatie, behalve Initialize en LoggedIn, die gebruik willen maken van de databank moeten dit doen via de instanties in het Globalsobject, zoals databaseUserHelper. Alle klassen die de instantie bevatten van Globals worden niet met een pijl naar Globals zelf voorgesteld om het overzichtelijk te houden, maar krijgen bij hun instantiellijst het veld global mee. LoggedIn en Initialize maken zelf een instantie aan van de databankhelpers omdat deze niet gedurende de hele applicatie gebruikt worden en slechts zelden opgeroepen worden.

De LoggedIn-instantie houdt een lijst bij van de huidig ingelogde gebruikers, de huidig aangeraden video's en de recent bekeken video's met hun ViewHistory-ID. De instantie houdt belangrijke niet-persistente informatie voor de gehele app bij en het is dus belangrijk dat deze instantie globaal wordt bijgehouden in de Globals-singleton. LoggedIn heeft ook toegang tot de LoginActivity opdat die de Toast en RecommendationButton kan controleren. De RecommendationButton is enkel zichtbaar als er mensen ingelogd zijn en de Toast verschijnt wanneer er geen aanbeveling kan gegenereerd worden of er iets mis is gegaan. Deze beide zaken zijn enkel regelbaar vanuit de LoggedInklasse. De ExcelWriterklasse wordt gebruikt om data over de HistoryDatarecords, ViewHistoryrecords, berekende tagscores (een geschatte score van de gebruiker voor die tag) en andere gebruikersdata zoals geslacht en leeftijd uit te schrijven naar CSV-bestand. De CSV-bestanden kunnen uitgeschreven worden door bij 'Settings' op de knoppen te klikken wat verder wordt uitgelegd in [Bijlage B: Gebruikershandleiding](#).



Figuur 4.10: Deel van het UML-diagram waar de databankmethoden en Globals behandeld worden

De Recommendationinterface verwacht dat een aanbevelingsklasse twee methoden kan oproepen, namelijk één om de aanbevolen lijst films afgebeeld op hun scores terug te geven en één om de lijst met onaanvaardbare films terug te geven. Zowel NoHistoryRecommendation als HistoryRecommendation implementeren de methoden op hun manier. De RecommendationCalculatorklasse bezit twee instanties van het type Recommendation, één voor de demografische aanbeveling en één voor de geschiedenisgebaseerde

aanbeveling. Dankzij de uniforme methoden van de Recommendationinterface worden de twee soorten aanbevelingen in hetzelfde formaat afgeleverd aan de RecommendationCalculator en handelt de interface als een soort adapter waarbij beide aanbevelingsklassen streven naar de targetinterface. Ook bezit de RecommendationCalculatorklasse een instantie van de klasse HistoryProcessor om nog eens de scores van de gebruikers voor wie de aanbeveling zal worden gegenereerd, te herberekenen. Meer uitleg over deze klassen wordt verklaard in 4.6.4 met behulp van enkele sequentiediagrammen.

De ExpectedEmotionProcessor doet alle verwerkingen van verwachte emoties. Deze zorgt ervoor dat de eigen getoonde emoties kunnen worden gebruikt als verwachte emotie, dat de gemiddelde emoties van gebruikers kan bepaald worden als verwachte emotie en dat bij initialisatie de bestanden voor de verwachte emoties correct worden ingelezen naar de databank. Alles wat te maken heeft met de databank en de dataklassen staat rechts bovenaan in Figuur C.6 in de bijlage. De MyDao interface en de abstracte klasse HistoryDatabase werden geïmplementeerd zoals vermeld op in de Android developer site over hoe data moet verkregen worden met Room *data access objects* (DAO's) en de code werd deels daarvan gehaald [55].

De klassen CameraHelper, FaceListAdapter, ImageHelper en StorageHelper staan in voor specifieke hulpmethoden, zoals het opslaan van bestanden, en zijn allemaal afkomstig van de Android developer website of de Microsoft Cognitive website. CameraHelper zorgt voor het initialiseren van de camera en heeft een Apache License [56]. FaceListAdapter kan een lijst van bitmaps teruggeven per gedetecteerd gezicht in een foto en maakt hiervoor gebruik van de klasse ImageHelper en de output van de gezichtsdetectie van Microsoft die de coördinaten van de gedetecteerde gezichten teruggeeft. Beide klassen hebben een MIT License en werden gehaald van de Githubpagina van Microsoft [39, 57]. StorageHelper is verantwoordelijk voor het wegschrijven van foto's en bestanden naar het Androidtoestel en is deels overgenomen van de Android developer site [58]. Zo worden tijdens de kijktesten steeds alle gecreëerde thumbnails van alle gedetecteerde personen weggeschreven naar het toestel en worden ook de CSV-bestanden met deze klasse gemaakt. De details voor het oproepen van de services werden reeds in Hoofdstuk 3 uitgelegd met behulp van de nodige asynchrone taken en licenties.

4.6.4 Sequentiediagrammen

De sequentiediagrammen van de drie fasen worden hieronder beschreven. De methoden, zoals gebruikt in de applicatie, worden getoond maar enkel met hun parameterlijst indien die benadrukt moet worden. Alle sequentiediagrammen zijn in Bijlage C: Grafieken, diagrammen, code en tabellen terug te vinden.

Aanmelden

Voor het aanmelden van de personen die zich voor de camera bevinden, moeten verschillende componenten samenwerken. Het sequentiediagram hiervan is terug te vinden in Figuur C.8. Allereerst drukt één van de gebruikers op de startknop onderaan het scherm wanneer de app zich in de LoginActivity bevindt. Dit zorgt ervoor dat er een foto genomen wordt en vanuit de Cameraklasse aan de LoginActivity gegeven wordt. De LoginActivity zet de foto om naar een bitmapformaat en geeft die mee aan de PersonGroupManagementklasse die de methode detect asynchroon oproept van FaceClientService. FaceClientService is de hulpklasse voor identificatie en detectie met de Microsoftservice. Gezichten worden hiernaar opgestuurd in de eerste plaats voor gezichtsherkenning om mensen te identificeren maar in deze loginfase worden leeftijd en geslacht ook al ingeschat. Deze informatie kan namelijk meteen gebruikt worden om, ofwel een nieuwe persoon aan te maken in de databank met alle gegevens ofwel een bestaande persoon zijn gegevens te updaten.

Wanneer FaceClientService een antwoord geeft, stuurt PersonGroupManagement het antwoord, dat de coördinaten van de verschillende gedetecteerde gezichten bevat en de face-ID's van de gezichten, samen met de bitmap naar een FaceListAdapterobject. De klasse FaceListAdapter kan met de methode getFaceThumbnails een lijst van bitmapobjecten van de bitmap maken, één bitmap per gedetecteerd gezicht. Vervolgens roept PersonGroupManagement de identitymethode van FaceClientService op die met de face-ID's de mogelijke kandidaten van het gezicht opzoekt door te vergelijken met de gezichten in de Face Storage. Bij het antwoord van de identitymethode worden alle gezichten overlopen.

Indien er voor een gezicht geen mogelijke kandidaten zijn, wilt dit zeggen dat de persoon nog niet gekend is en wordt de methode addPersonToDBAndFaceStorage opgeroepen die aangeduid is met de donkerdere

achtergrond in het sequentiediagram in Figuur C.8. Hierbij wordt het geslacht in de applicatie door de drie services samen bepaald door middel van voting. Daarom voegen de twee andere services nu ook hun geschatte geslacht toe wanneer ze aangeroepen worden door `PersonGroupManager` met de bitmap. De databanktabel `Users` bevat een kolom geslacht en leeftijd voor de drie services en kolommen `age` en `gender` voor de definitieve uitslag. `FacePPHelper` en `KairosHelper` vullen respectievelijk de recordkolommen `ageFacePP` en `genderFacePP`, en `ageKairos` en `genderKairos` in van de net aangemaakte gebruiker en plaatsen het gewonnen geslacht in de recordkolom `gender`. Met de geschatte leeftijd van `Face++` en `Kairos` doet het systeem niets en houdt ze enkel bij ter informatie in de databank (zie Hoofdstuk 3). Wanneer de gebruiker wel mogelijke kandidaten heeft en dus wel al gekend is in het systeem, wordt zijn gezicht nogmaals toegevoegd aan `Face Storage` om vervolgens extra getraind te worden en geslacht en leeftijd opnieuw te bepalen en updaten. Steeds wordt de persoon, nieuw of bestaand, ingelogd door deze toe te voegen aan de `LoggedIn`-instantie.

Genereren van de aanbeveling

Wanneer de gebruikers zijn ingelogd, kunnen ze op de ‘Show Recommendation’-knop duwen vanuit de `LoginActivity`. Dit zorgt ervoor dat de `LoginActivity` de methode `calculateRecommendation` van de `RecommendationCalculator` oproept, zoals getoond in Figuur C.9. Om de aanbeveling te genereren heeft de `RecommendationCalculator` nood aan alle ingelogde personen van de `LoggedIn`-instantie die wordt bijgehouden door de globale singleton `Globals`. Om zeker te zijn dat de impliciete feedback van de ingelogde gebruiker up-to-date is, vraagt `RecommendationCalculator` aan de `HistoryProcessor` om al deze gebruikers nog eens te updaten met `getHistoryDataForCalculation` (deze methode roept de methode `updateCalculatedViewHistory` op, hetgeen gedemonstreerd wordt in Listing C.3 en Listing C.4). Vervolgens worden alle ingelogde gebruikers overlopen en één voor één bekeken of ze geschiedenisrecords hebben. Afhankelijk hiervan worden ze aan de lijst `usersNoHistory` of `usersWithHistory` toegevoegd. Deze lijsten worden vervolgens meegegeven aan respectievelijk de `NoHistoryRecommendation`-klasse en de `HistoryRecommendation`-klasse die de scores voor de films voor de gebruikers berekent. De berekening van de scores die gemapt zijn op de video-ID's werd uitgelegd in 4.3.1 en 4.3.2. Tenslotte vermengt de `RecommendationCalculator` de verkregen scores per video-ID en verwijdert alle onaanvaardbare films alvorens de uiteindelijke aanbevolen filmlijst in te stellen in de globale `LoggedIn`-instantie. Ook wanneer gebruikers op de knop ‘Recalculate’ duwen in de `InformationActivity` (het scherm waarin de info over de geselecteerde film getoond wordt), wordt de `RecommendationCalculator` opgeroepen op dezelfde manier.

Verzamelen van impliciete feedback

Wanneer de gebruiker op ‘Play’ duwt in de `InformationActivity` speelt het gekozen fragment af en begint de app met het nemen van foto's om de twee seconden. Vanaf dat moment start de fase waarin impliciete feedback wordt verzameld. Dit wordt in Figuur C.10 voorgesteld door een *loop* die om de twee seconden de `Cameraklasse` aanspreekt. Hierbij wordt een `HistoryProcessor`-object aangemaakt in de `InformationActivity` die per foto de relatieve teller (`relativeCount`) verhoogt en de starttijd van het kijken bijhoudt om de relatieve tijd in de video van de foto te weten (`relativeTime`). Wanneer een foto wordt gemaakt, ontvangt de `InformationActivity` de foto, bepaalt de huidige tijd (`exactTime`) en berekent de relatieve tijd in het fragment (`relativeTime`). De `InformationActivity` geeft de foto, `relativeTime` en `exactTime` mee aan de `HistoryProcessor` die op zijn beurt de `relativeCount` met één verhoogt. Vervolgens roept de `HistoryProcessor` een `EmotionHelper`-object aan met de foto en de tijdstippen als argument.

`EmotionHelper` is een hulpklasse voor de service van Microsoft om in de eerste plaats de gezichten te identificeren. Terwijl `EmotionHelper` de gezichtsherkenning uitvoert, wordt er ook gezichtsdetectie uitgevoerd om emoties en bedekkingen van het gezicht in foto's te detecteren. Hier kan de app dan ook al meteen gebruik van maken en deze info, samen met de tijdstippen, in een nieuw `HistoryDataRecord` verwerken. Alle gedetecteerde gezichten worden overlopen voor de identificatie in de loop en indien er een gezicht bij zit waarvoor er weer geen kandidaten zijn, zal deze persoon net zoals in de loginfase worden toegevoegd aan de databank en `Face Storage`. Dit gebeurt met de methode `addPersonToDBAndFaceStorage`, die al eerder uitgebreid werd beschreven in het sequentiediagram van de loginfase in Figuur C.8.

Elk resultaat van elke foto wordt bijgehouden in een `HistoryDataRecord`. Hierdoor is het mogelijk achteraf de data opnieuw door het systeem te laten verwerken indien een andere similariteitsmethode ge-

implementeerd wordt. De `EmotionAndOcclusion`-, `EyesAndHeadpose`- en `AttentionAndHeadpose`klassen zijn verantwoordelijk voor het invullen van de `HistoryDataRecords` met de gegevens van respectievelijk Microsoft Face API, Face++ en Kairos. Vooraleer zo een `HistoryDataRecord` kan bestaan, moet er een `ViewHistoryRecord` zijn die bijhoudt welke gebruiker naar welke video kijkt. Aangezien Microsoft centraal steeds eerst voor de identificatie zorgt en dan pas Face++ en Kairos oproept, is het `EmotionAndOcclusion` die verantwoordelijk is voor het aanmaken van het `ViewHistoryRecord`, indien dit nog niet bestaat voor de huidige gebruiker en film. Dit gebeurt met de methode `addHistories` die wordt opgeroepen op het `DatabaseHistoryHelper`-object. Vervolgens is het ook aan `EmotionAndOcclusion` om het `HistoryDataRecord` aan te maken voor de huidige foto. Meteen vult de klasse ook de gedetecteerde emoties en occlusion in met de methode `updateHistoryDataValuesMicrosoft`. Het `HistoryData-ID` wordt teruggegeven waardoor `FacePPHelper` en `KairosHelper` weten op welk record zij nu hun gegevens (asynchroon) moeten invullen. `EmotionAndOcclusion` roept de `detectExtra`-methode op op `FacePPHelper` en `KairosHelper` en deze passen vervolgens ook asynchroon de databank aan door elk respectievelijk een `EyesAndHeadpose`- en `AttentionAndHeadpose`-instantie aan te maken.

4.7 Testplan

De Androidapp wordt getest met basistesten zoals unittesten en integratietesten. In Android zijn er twee soorten unittesten: lokale testen en *instrumented* testen [59]. De lokale testen draaien enkel op de lokale machine, zoals de laptop waarop Android Studio staat geïnstalleerd, waar ze lokaal op de Java Virtual Machine worden uitgevoerd. De instrumented testen worden uitgevoerd op het Androidtoestel zelf waardoor ze toegang hebben tot data op het toestel, zoals de lokale Roomdatabank, en instrumentatie-informatie, zoals de Context van de app. Deze laatste soort test moet gebruikt worden indien de test Android dependencies nodig heeft die zogenaamde *mock* objecten niet kunnen vervangen, bijvoorbeeld wanneer de te testen methode niet kan opgeroepen worden zonder dat deze toegang heeft tot de lokale databank. De instrumented testen worden ook gebruikt wanneer integratie- en functionele User Interface testen worden geschreven. In een Android project zijn de lokale testen te vinden in de map `app/src/test/java` en de instrumented testen in de map `app/src/androidTest/java` [60].

In deze masterproef werden alle testen uitgevoerd op het toestel zelf en staan ze bijgevolg onder de instrumented testen. De reden hiervoor is dat alle data die de geteste methoden gebruiken, afkomstig zijn uit de lokale databank. In het bestand `app/src/androidTest/java/SimpleEntityReadWriteTest.java` worden de databank en de dataklassen getest door een object van elke klasse data-objecten in te lezen in de databank en dan te vergelijken met het uitgelezen resultaat. In deze unittest wordt een software-module apart getest, in dit geval dus de klasse verantwoordelijk voor het wegschrijven naar en lezen van de databank. In het bestand `app/src/androidTest/java/InstrumentedTest.java` staan de integratietesten van de verschillende methoden die zorgen voor het maken van de aanbeveling. Hiervoor worden zowel de `databaseHelper`methoden, om de nodige testdata uit de databank te lezen, als de logicaklassen, om de data te verwerken, aangesproken. Dankzij deze integratietest worden verschillende aparte softwaremodules als een geheel getest en wordt er getest of de aparte componenten op een correcte manier interageren [61]. Het is duidelijk dat ook hier een instrumented test vereist is aangezien de test op het toestel zelf moet draaien. [Bijlage B: Gebruikershandleiding](#) beschrijft hoe deze testen praktisch uitgevoerd kunnen worden in Android Studio. Beide testen slagen op alle voorziene testmethoden.

De app voorziet ook de mogelijkheid om de databank en het aanbevelingssysteem uit te testen door een fictieve gebruiker toe te voegen met zes `ViewHistoryRecords`. Dit kan gerealiseerd worden door in het instellings scherm op 'Initialise' te klikken wat ervoor zorgt dat een gebruiker met ID 99 wordt aangemaakt in de databank met bijhorende `ViewHistoryRecords`. Deze gebruiker wordt ook ingelogd waardoor er op de knop 'Show Recommendation' kan geklikt worden. Indien alle gebruikers worden uitgelogd en enkel deze gebruiker met ID 99 wordt aangemeld, genereert het systeem een aanbeveling op basis van de fictieve geschiedenis. Aangezien de gebruiker drie films met de tags komedie en animatiefilm een hoge score gaf en drie films met de tags horror en drama een lage score gaf, zal de aanbeveling verschillende films bevatten die de tags komedie en animatiefilm bevatten. De details voor het uittesten van deze voorbeelddata zijn terug te vinden in [Bijlage B: Gebruikershandleiding](#).

Hoofdstuk 5

Resultaten: uittesten van de app met testpersonen

De werking van de applicatie werd getest op enkele testpersonen (met leeftijden van 5 tot 72 jaar) die naar zes verschillende filmfragmenten kijken. Hun impliciete feedback en de daarop gebaseerde aanbevelingen worden in dit hoofdstuk geanalyseerd en vergeleken met hun echte feedback. Hiervoor gaven de testpersonen ook een expliciete feedbackscore om te vergelijken met de inschatting van de waarderingsscore door de applicatie. Er zal onderzocht en besproken worden welke factoren (emoties, gedrag en aandachtsniveau) de beste inschatting geven en welke methoden het meest nauwkeurige resultaat geven om vervolgens een antwoord te geven op dit deel van de onderzoeksvraag.

Een ander aspect van de onderzoeksvraag was om te onderzoeken hoe de getoonde emoties kunnen bijdragen tot een inschatting van de waarderingsscore. Bijkomend onderzoek werd uitgevoerd naar hoe de verwachte emoties het best bepaald kunnen worden. Dit hoofdstuk overloopt eerst hoe de testen werden uitgevoerd en gaat dan de vier geïmplementeerde sets van verwachte emoties één voor één af om hun resultaten en de interpretatie te bespreken. Tenslotte worden de meningen van de testpersonen besproken en worden de algemene resultaten overlopen.

5.1 Het uitvoeren van de testen

Zes filmfragmenten werden geselecteerd om te tonen aan de testpersonen. Elk filmfragment heeft één overheersende emotie en aangezien de app met zes gedetecteerde emoties werkt, werd ervoor gekozen één filmfragment per overheersende emotie te voorzien. De informatie over deze video's wordt uitgelezen uit het `movies.json` bestand door de app. De eerste zes video's in dit bestand zijn meteen ook de zes fragmenten voor de testen. Tabel 5.1 geeft een overzicht over de videofragmenten die het doel van elk fragment verduidelijken. De fragmenten zijn afkomstig van YouTube en het `movies.json` bestand bevat de URL van het YouTubefragment. De zes fragmenten van de test werden ook op de bijgevoegde USB-stick gezet in mp4-formaat (in de map *Videos*) voor het geval de link op YouTube niet meer beschikbaar zou zijn.

5.1.1 Voorafgaand aan de testen

Alvorens de testpersonen getest werden, werd er voor elk van de zes fragmenten een eerste verwachte emotiefingerprint voor elke video gemaakt. Set 1 werd opgesteld voor de aanvang van de testen en set 2, 3 en 4 werden achteraf opgesteld op basis van de ondervonden resultaten na de testen. Verder in dit hoofdstuk worden de verschillende sets één voor één besproken. Alle verwachte emotiespectra hebben elke seconde een waarde gedefinieerd. De getoonde emoties van de gebruiker worden elke twee seconden geregistreerd. Hierdoor ligt de frequentie van de verwachte emoties hoger dan die van de getoonde emoties en bestaat er steeds een verwachte waarde voor elke getoonde waarde, zelfs indien een service eens een seconde langer op zich laat wachten bij de detectie van de getoonde emoties. Deze verzameling van fingerprints van alle uiteindelijke sets zijn terug te vinden als grafieken in Figuur C.11 en Figuur C.12 en als CSV-bestanden in de map *ExpectedEmotions/DifferentSets* op de USB-stick.

Tabel 5.1: Een overzicht van de zes video's die getoond worden aan de testpersonen

nr	Film waaruit afkomstig	Overheersende emotie	Korte beschrijving fragment
1	Dude, Where's my car?	Blijheid	Een fragment uit een komedie waarin op korte tijd de kijker aan het lachen wordt gebracht.
2	Up	Verdriet	Een fragment over het leven van een man die op het einde zijn vrouw verliest. De kijker wordt meegetrokken in dit verdriet.
3	Brothers	Boosheid	Het vele roepen en geweld in dit fragment zorgt ervoor dat de kijker gevoelens van boosheid krijgt.
4	The Fly	Afkeer	Op het einde van het fragment ervaren kijkers veel gevoelens van afkeer omdat een man op gruwelijke wijze vermoord wordt.
5	What Happened to Monday	Angst	De spanning wordt opgebouwd naarmate de kijker begrijpt dat de vader de vinger van zijn dochter gaat afsnijden. De scène wordt traag uitgevoerd waardoor de kijker angst toont.
6	The Conjuring 2	Verrassing	Door de opbouwende spanning in deze scène verschiet de kijker telkens er iets onverwachts gebeurt.

Voordat alle testen van start konden gaan, werd ook eerst één persoon apart getest in een 'trial'-test om eventuele onverwachte gebeurtenissen en zwaktes in het testproces te ontdekken. Zo werd er gemerkt dat het aangenamer is om met een oortelefoon naar het fragment te luisteren en de tablet in een houder te plaatsen zodat de kijker de tablet niet moet vasthouden. Het werd ook duidelijk dat niet enkel de totale som van negatieve en positieve feedback moet worden uitgeschreven, maar ook de deelscores. Hierdoor werden de kolommen `attentionScore`, `sharedScore`, `emotionScore` en `typicalEmotionScore` aan de databanktabel `ViewHistory` toegevoegd zodat deze waarden per `ViewHistoryrecord` kunnen uitgeschreven worden naar het CSV-bestand. De betekenis van deze kolommen werd reeds uitgelegd in Hoofdstuk 4 en de scores worden hieronder nog eens verduidelijkt in Tabel 5.2. Dankzij de deelscores kunnen meer gedetailleerde conclusies over de kwaliteit van de scores bepaald worden. Op deze manier kan er afzonderlijk naar de aandachtsscore en de emotiescores gekeken worden. De similariteitberekening, uitgelegd in Hoofdstuk 4, genereert een score op één per emotie. Aan de hand van gewichten wordt de score per emotie afgebeeld op een ander totaal zodat de som van alle emotiesimilariteiten gelijk is aan zes. De uiteindelijke totale waarderingsscore die de applicatie gebruikt is `scoreDistance` en de andere worden enkel opgeslagen voor het onderzoek.

5.1.2 Het testproces

Voor elke testpersoon X wordt er een *Vragenlijst testpersonen $X.xlsx$* -bestand aangemaakt. Er worden in totaal 20 mensen getest met leeftijden van 5 tot 72 jaar. Bij de testpersonen zitten 2 kinderen van 5 jaar en de video's die zij te zien krijgen zijn enkel deze die voldoen aan hun minimumleeftijd (zodat ze geen horrorfragmenten te zien krijgen). Video 2 is de enige video die voldoet aan de minimumleeftijd van 5 jaar, waardoor deze testpersonen enkel geschiedenis voor video 2 zullen hebben. Tabblad 'Info' bevat het geslacht en de leeftijd van de testpersoon samen met de ingevulde vragenlijst. De vragenlijst behandelt vragen die dienen voor een klein onderzoek naar de mate waarin mensen een gelijkaardige app op de markt zouden vertrouwen en akkoord zouden gaan met het maken van foto's. Ook omvat het onderzoek verschillende vragen over het soort emoties en het gedrag dat ze persoonlijk snel zouden tonen als reactie op videocontent. Tenslotte vullen de testpersonen ook bij alle 48 mogelijk tags (of enkel de tags die relevant zijn voor de zes bekeken video's) een score op 10 in gebaseerd op de mate waarin ze films met deze genres en keywords leuk vinden. Het aanbevelingssysteem werkt namelijk op basis van geschatte scores voor tags in te schatten die gelijkaardige films met elkaar verbinden. Dankzij hun expliciete scores voor de tags, kunnen deze vergeleken worden met de tagscores die de applicatie berekend heeft op basis van de zes bekeken video's.

De testpersonen van 5 jaar konden de vragenlijst nog niet invullen en hierdoor komt het totaal aantal testpersonen voor de vragenlijst op 18. Met de verzamelde informatie van de 18 testpersonen kunnen conclusies getrokken worden over welke soort impliciete feedback het meest interessant kan zijn voor dit systeem. De conclusies worden op het einde van dit hoofdstuk besproken.

Tabel 5.2: Een overzicht van de berekende scores

Naam score	Betekenis
emotionScore	De berekende score op basis van alle getoonde emoties met de distance-methode. Deze score staat op een totaal van 6. De overheersende emoties wegen iets meer door dan de andere. Indien er bijvoorbeeld één sterkste overheersende emotie is, krijgt deze gewicht 2 en de andere 0,8.
typicalEmotionScore	De berekende score enkel gebaseerd op de overheersende emoties met de distance-methode. Deze score staat op een totaal van 6. Indien er één sterkste emotie is, krijgt deze gewicht 6. Indien er twee sterkste emoties zijn, krijgen ze elk gewicht 3.
attentionScore	De berekende score enkel gebaseerd op het aandachtsniveau van de kijker. Deze score staat op een totaal van 4. Deze score is onafhankelijk van de gekozen distance- of margins-methode.
sharedScore	De berekende score op basis van het gedrag van de persoon vermeerderd met de attentionScore. Het is de gedeelde gemeenschappelijke score tussen de distance- en de margins-methode omdat deze onafhankelijk is van de gekozen methode en heet daarom de sharedScore.
scoreDistance	De som van emotionScore en sharedScore bij set 1, 2, 3 en 4. De som van typicalEmotionScore en sharedScore bij set 4.2. Deze score is een score op 10 en wordt gebruikt voor het genereren van de aanbeveling in de rest van de applicatie.
scoreMargins	De som van een apart bepaalde margins-score en dezelfde sharedScore. Dit is ook een score op 10. Deze margins-methode wordt enkel gebruikt voor het onderzoek maar wordt verder niet toegepast in de applicatie.

Tabblad ‘HistoryData’ bevat alle opgeslagen HistoryDatarecords van deze persoon voor elke bekeken video. Zoals reeds vermeld wordt er per film een impliciete, geschatte score berekend van wat de gebruiker van de film vond. Deze score wordt opgeslagen in bijhorend ViewHistoryrecord en de berekende waarden hiervan worden per gebruikte set van verwachte emoties (in totaal zullen dit er vier worden) opgeslagen in het tabblad ‘HistoryId’. ‘UserData’ bevat de aanmaaktijd en geschatte leeftijd en geslacht van de gebruiker voor eventuele extra evaluatie van de drie services. Het tabblad ‘Ine’ bevat de waargenomen feedback van de testpersoon, bijvoorbeeld het tijdstip waarop de testpersoon duidelijk lachte, en de expliciete score op tien die weergeeft hoe graag de persoon het fragment heeft bekeken en iets gelijkaardig zou willen zien.

Tenslotte overloopt het ‘Opstart’-tabblad de verschillende stappen waarin de testen worden uitgevoerd. Allereerst moet er worden gezorgd dat de kijker op een goede afstand van de tablet zit en er voldoende belichting in de kamer is. Een nieuwe gebruiker wordt meerdere keren aangemeld door zes tot zeven keer op de startknop te duwen. Hierdoor wordt het gezicht meteen zes tot zeven keer getraind en worden de leeftijd en het geslacht met hogere zekerheid bepaald. Telkens worden de geschatte leeftijd en geslacht per service automatisch toegevoegd aan een gegenereerd CSV-bestand *AgeAndGender_userX.csv* waarvan de uitkomst in het tabblad ‘UserData’ wordt gezet. Op dit moment kan het systeem een aanbeveling op basis van demografie vormen en dit wordt voorgesteld aan de testpersoon die hun mening over de aangeraden video’s geeft in het ‘Info’-tabblad.

De zes vooraf bepaalde filmfragmenten worden getoond door op de knop ‘Show All Videos’ te klikken waardoor de zes testvideo’s als eerste in de lijst getoond worden. De persoon die de testen aflegt, noteert het moment waarop de kijker iets interessants doet en na het kijken noteert die ook per video wat de expliciete score op tien is van de kijker. Als de zes filmfragmenten bekeken zijn, wordt de similariteit in de ViewHistoryrecords berekend door de app. Dit kan met de ‘Recalculate’-knop in het instellingscherm en aangezien er vele records overlopen worden, kan dit enkele seconden duren. Meteen worden ook alle data weggeschreven naar CSV-bestanden door op de ‘Print Excel’-knop te duwen in het instellingscherm zodat er steeds een back-up van de HistoryDatarecords wordt gemaakt en deze later opnieuw kunnen worden ingelezen (meer informatie wordt vermeld in [Bijlage B: Gebruikershandleiding](#)). De testpersonen kunnen na het kijken de vragenlijst invullen in ‘Info’. De resultaten van alle testpersonen zijn in overeenkomende mappen terug te vinden in de map *Testen met testpersonen*, samen met de gegenereerde CSV-bestanden en genomen foto’s in de map *data X*, met X het nummer van de testpersoon. Een overzicht van alle antwoorden en berekende scores van alle testpersonen in één Excelwerkblad wordt gegeven

in het bestand *InfoAllUsers*. De informatie uit dit bestand wordt ook achteraan in [Bijlage C: Grafieken, diagrammen, code en tabellen](#) in aparte tabellen nogmaals gedemonstreerd.

5.2 Het gebruik van een eerste set: set 1

Set 1 werd gecreëerd door zelf meerdere keren naar de fragmenten te kijken en op de juiste momenten de goede emoties uit te beelden. De `PHOTO_MODUS`-constante van de `FaceApplication`-klasse moet hiervoor ingesteld worden op `INITOWNEXPECTEDEMOTIONS` zodat er elke seconde emotiedetectie plaatsvindt en een *expected emotions*-spectrum wordt aangemaakt, waarover meer uitleg in [Bijlage B: Gebruikershandleiding](#). De verschillende spectra werden samengenomen en de belangrijkste emoties voor dat fragment werden afgevlakt. Deze verzameling van fingerprints wordt set 1 genoemd en de grafieken zijn te zien in [Figuur C.11](#) en [Figuur C.12](#). Op basis van deze verwachte emoties wordt de similariteit, een maat van overeenkomst, met de getoonde emoties berekend en de impliciete feedbackscore van de testpersonen geregistreerd. Aangezien de emotiefingerprint de verwachte emoties per tijdstip in de film bepaalt, kan tot op de seconde bepaald worden welke emotie wanneer moet voorkomen en kunnen verschillende emoties op verschillende tijdstippen in eenzelfde video voorkomen.

5.2.1 Resultaten met het gebruik van set 1 en moeilijkheden van de testen

Nadat alle testpersonen de fragmenten bekeken hebben, kunnen er een aantal zaken vastgesteld worden. De ontdekte eigenschappen worden hier overlopen samen met de ondervonden moeilijkheden. [Tabel C.2](#) geeft een overzicht van alle testpersonen nadat alle testen uitgevoerd zijn en is te vinden in [Bijlage C: Grafieken, diagrammen, code en tabellen](#). De tabel toont de scores voor het aandachtsniveau per film en per kijker op een totale score van 4. De absolute waarde van het verschil tussen de aandachtsscore en de expliciete score van de kijker wordt gemaakt en getoond op een resultaat van 10 in de kolom ‘difference attention’. In de laatste kolom toont de tabel per kijker de maximale emotieprobabiliteit over alle films per emotie (in het rood staan de foutieve waarden aangeduid die niet in de uiteindelijke maximale waarde worden opgenomen). Merk op dat deze `attentionScore` voor alle sets dezelfde zal zijn aangezien de score enkel gebaseerd is op het aandachtsniveau en niets te maken heeft met de set van verwachte emoties.

Lage probabiliteitswaarden voor alle emoties behalve blijheid

In [Bijlage C: Grafieken, diagrammen, code en tabellen](#) worden interessante voorbeeldspectra van vier testpersonen getoond in [Figuur C.13](#) en [Figuur C.14](#). Wat hierbij op het eerste zicht meteen opvalt, is dat de emotie blijheid heel vaak gedetecteerd wordt met een hoge probabiliteit. Alle andere vijf emoties worden veel minder snel gedetecteerd en als een persoon de emoties toch toont, hebben ze een lage probabiliteitswaarde.

Per kijker werd vervolgens de maximale emotiewaarde over alle films bepaald en gesorteerd per emotie, zoals te zien in [Tabel 5.3](#), een deel uit [Tabel C.5](#). Het valt op dat blijheid (kolom met nummer 3) steeds hoge waarschijnlijkheidswaarden haalt, behalve bij testpersoon 6 aangezien deze nooit heeft gelachen. Blijheid is bijgevolg de meest makkelijke emotie dat te bepalen was bij de testen. 90 % van alle testpersonen haalt hierbij een maximale waarde van 0,95 of meer doorheen alle bekeken video's.

Verrassing (kolom met nummer 5) lijkt bij meerdere mensen ook hoge waarden te halen maar indien naar de bijhorende foto's gekeken wordt, valt het op dat ze helemaal geen verrassing uiten. Testpersonen 5 en 14 bijvoorbeeld waren op de desbetreffende foto beiden aan het geeuwen en bijgevolg is verrassing verkeerd gedetecteerd. Testpersoon 13 was bij de bijhorende foto in de ogen aan het wrijven en de wenkbrauwen aan het optrekken, waardoor de foto verward werd met verrassing. Tenslotte werd ook naar de foto van testpersoon 18 gekeken en het bleek dat deze persoon (een kind) net met de handen in de mond zat waardoor de foto ook verward werd met verrassing. Indien deze waarden worden weggelaten (in het rood aangeduid in de tabel), bekomt men slechts een maximale waarschijnlijkheidswaarde van 0,58 voor de emotie verrassing.

Tabel 5.3: De maximale waarschijnlijkheidswaarden per gebruiker over alle video's gesorteerd per emotie. De emotiegetallen betekenen: 0 = anger, 1 = disgust, 2 = fear, 3 = happiness, 4 = sadness, 5 = surprise

nr	emotions	maximum emotions values					
		0	1	2	3	4	5
		/1					
1		0.02	0.02	0.02	1.00	0.46	0.03
2		0.00	0.02	0.01	1.00	0.59	0.21
3		0.02	0.11	0.13	1.00	0.42	0.08
4		0.05	0.06	0.01	1.00	0.36	0.20
5		0.40	0.07	0.01	0.98	0.29	0.68
6		0.01	0.01	0.13	0.00	0.52	0.22
7		0.01	0.01	0.01	1.00	0.16	0.15
8		0.20	0.08	0.04	1.00	0.31	0.52
9		0.17	0.36	0.27	1.00	0.66	0.21
10		0.11	0.01	0.00	1.00	0.10	0.01
11		0.16	0.02	0.09	1.00	0.23	0.58
12		0.49	0.26	0.06	1.00	0.53	0.48
13		0.51	0.49	0.13	1.00	0.60	0.62
14		0.50	0.16	0.06	1.00	0.55	0.93
15		0.00	0.01	0.00	1.00	0.10	0.01
16		0.08	0.14	0.25	1.00	0.96	0.12
17		0.26	0.16	0.00	1.00	0.68	0.04
18		0.02	0.02	0.08	0.87	0.11	0.75
19		0.06	0.01	0.00	1.00	0.21	0.33
20		0.04	0.04	0.00	1.00	0.21	0.31
max		0.51	0.49	0.27	1.00	0.96	0.93
		0.51	0.49	0.27	1.00	0.68	0.58

Verdriet (kolom met nummer 4) lijkt ook hoge waarden te scoren bij de emotiedetectie bij testpersoon 16 maar indien naar de bijhorende foto gekeken wordt, is het duidelijk dat de persoon eigenlijk een sterke vorm van afkeer toont. Indien deze persoon wordt weggelaten is de maximale waarde van verdriet doorheen alle testpersonen slechts 0,68. Uit de tabel blijkt zelfs dat over alle testpersonen, slechts 15 % een probabilliteit van 0,60 of hoger haalt doorheen alle video's. Bij de emotie angst (kolom met nummer 2) bedraagt de maximale waarde over alle testpersonen en alle video's slechts 0,27. Er kan besloten worden dat enkel de emotie blijheid grote probabilliteiten vertoont en het ziet er naar uit dat de verwachting dat boosheid, angst en afkeer enkel slecht gedetecteerd konden worden, zal moeten uitgebreid worden met verdriet en verrassing. Er zijn bijgevolg vijf moeilijk te detecteren emoties waarvan de maximale gedetecteerde waarde over alle testpersonen en alle video's slechts 0,68 bedraagt en waarvan de waarden over het algemeen rond 0 liggen.

Distance-score geeft betere resultaten dan margins-score

Aangezien de margins-methode de verwachte emotiewaarden uitbreidt met boven- en ondermarges, gaat er een stuk van de informatie over het effectieve verschil met de verwachte waarde verloren. Hierdoor wordt de berekende similariteit niet zo nauwkeurig ingeschat als bij de distance-methode. De margins-score zal steeds als extra informatie-element worden bijgehouden maar het is vooral de distance-methode waarop er in de volgende paragrafen gefocust wordt.

Emoties duiden geen persoonlijke score aan

Allereerst is het duidelijk dat als iemand een film niet leuk lijkt te vinden, omdat de persoon bijvoorbeeld weinig emoties toont, het niet per se betekent dat de kijker de video niet leuk vindt. Soms gaven mensen met weinig emoties de film toch een hoge expliciete score. Het omgekeerde geldt ook. Wanneer bijvoorbeeld testpersoon nummer 4 naar video 1 keek, kwamen de emoties van blijheid goed overeen met de verwachte emoties waardoor de totale distance-score berekend werd op 9,11/10. Ondanks deze hoge geschatte score gaf de testpersoon toch een expliciete score van slechts 6/10. Wanneer het zelfs voor een buitenstaander leek alsof de testpersoon de film leuk vond, gaf de persoon de film een lage score. Een volledige correcte inschatting maken van de feedback van een gebruiker is dus onmogelijk puur op basis van de emoties aangezien dit zelfs voor de mens al moeilijk in te schatten is.

Aandachtsniveau vertelt meer over de waarderingsscore

Het aandachtsniveau daarentegen geeft wel een goede indicatie van hoe de kijker de content ervaart. Testpersoon 2 bijvoorbeeld heeft gedurende video 6 bijna niet gekeken naar de tablet omdat ze het te eng vond om naar te kijken. Hierdoor werd zowel een laag aandachtsniveau als een groot tekort aan HistoryDatarecords waargenomen waardoor de aandachtsscore op 1,29/4, de sharedScore op -0,22 en de totale distance-score op 4,66/10 berekend werd. De persoon vond de video ook echt niet leuk en dankzij het toevoegen van een aandachtsfactor kon de applicatie dit correct inschatten. Dit bewijst dat impliciete feedback op basis van het aandachtsniveau een positieve invloed heeft op de bepaling van de waarderingsscore van de kijker.

Kairos schatte bij testpersoon 19 in dat deze niet continu de aandacht hield waardoor de testpersoon een score van 1,58/4 behaalde voor het aandachtsniveau. Hier was de aandachtsinschatting van Kairos foutief en de reden hierachter is dat de testpersoon te dicht bij het scherm stond, waardoor het lijkt alsof hij naast het scherm aan het kijken is. De bijdrage van het aandachtsniveau aan de totale waarderingsscore is hierbij ook groot wat bewijst dat de applicatie rekening houdt met de geschatte waarde van Kairos maar dat Kairos de inschatting ook niet steeds correct maakt.

Indien naar Tabel C.2 gekeken wordt, is het duidelijk dat het slechts bij twee kijkbeurten is voorgevallen dat testpersonen voor het aandachtsniveau onder 2/4 scoren. Dit komt omdat er aan de testpersonen werd gevraagd het volledige fragment te bekijken, wat de meesten dus ook deden. Wanneer zij de vrije keuze zouden hebben, zouden sommigen enkele fragmenten reeds vroegtijdig stoppen of wegstijgen omdat ze het niet aangenaam vonden om naar te kijken. Indien dit het geval zou zijn, registreert de app dit als negatieve feedback (zowel aandacht- als gedragsfeedback) en zouden opnieuw de berekende scores meer aansluiten bij de realistische feedback van de kijker. Eventueel zouden testen kunnen worden uitgevoerd waarbij kijkers het fragment mogen stoppen wanneer ze willen.

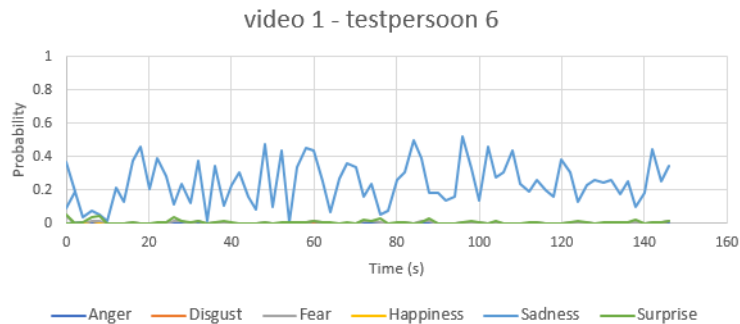
Neutrale expressie van kijkers wordt vaak verward met andere emotie

Wat ook opvalt, is dat kijkers vaak neutraal zijn in hun emoties bij het kijken naar de fragmenten. De emotiedetectie van Microsoft kan neutrale emoties soms verwarren met verdriet. Bij testpersoon 6 bijvoorbeeld komt er bij elke video hetzelfde patroon voor: gedurende het gehele fragment een emotie van verdriet met een probabiliteit tussen 0,1 en 0,4. Figuur 5.1 toont zijn getoonde emoties in de loop van de tijd.

Grote verschillen tussen overeenkomst overheersende emotie (typicalEmotionScore) en alle emoties (emotionScore waarbij overheersende emotie iets meer doorweegt)

Video 1 verwacht veel emoties van blijheid in zijn emotiefingerprint en de typicalEmotionScore werd 100 % gebaseerd op blijheid. Bij testpersoon 6 en video 1 resulteerde de typicalEmotionScore op 3,28/6 omdat de persoon zo goed als geen emoties van blijheid vertoonde, zoals te zien in Figuur 5.1. Wanneer ook de overeenkomst met alle emoties bepaald werd, de emotionScore, komt deze neer op 4,90/6. Hieruit kan besloten worden dat als een persoon niet lacht, wanneer lachen verwacht wordt, de typicalEmotionScore heel laag komt te liggen en de emotionScore (met lachen een zwaarder gewicht van 2 en de rest 0,8) toch hoog komt te liggen. Het kan dus een betere aanpak zijn om enkel op de overheersende emotie

van het fragment te gaan vergelijken. Momenteel werd de distance-score berekend op de emotionScore aangezien de TOTAL_SCORE-constante in de FaceApplication op MOREWEIGHTTOMOST ingesteld staat. Hierdoor wordt de distance-score gelijk aan 8,53. Indien de distance-score zou berekend worden op de typicalEmotionScore zou deze ongeveer 1,7 punten lager moeten liggen, wat beter zou zijn aangezien deze testpersoon niet voldeed aan de verwachte emotie. Dit is dus iets om in het achterhoofd te houden bij latere implementaties.



Figuur 5.1: De getoonde emoties van testpersoon 6 bij videofragment 1

Ook rekening houden met andere emoties dan verwacht

Bij horrorfragmenten waarbij eigenlijk de emotie afkeer verwacht wordt, zoals video 4, gebeurt het ook vaak dat mensen moeten lachen. Nu moet de keuze gemaakt worden of lachen dan als een positieve of negatieve feedback moet gezien worden. In set 1 hoort lachen niet bij de verwachte emoties van video 4 waardoor lachende mensen minder overeenkomst zullen vertonen met set 1. Kijkers die lachten, zoals bijvoorbeeld testpersoon 9, gaven vaak een iets positievere feedback dus het kan overwogen worden om lachen ook als verwachte emotie op te nemen.

Score op basis van het afwezig zijn van emoties

De emotiefingerprints kunnen ook gebruikt worden om de afwezigheid van emoties te testen. Testpersoon 6 heeft bijvoorbeeld een grote emotionScore voor video 5 waarbij hij weinig emoties toonde omdat de emoties afkeer, verdriet, boosheid, angst en verrassing ook heel laag liggen. Bij de verwachte emoties liggen deze emotiewaarden ook laag waardoor een kijker snel voldoet aan de scores van deze emoties, zelfs als de overheersende emotie meer doorweegt. Deze eigenschap kan interessant zijn om bijvoorbeeld bij hele serieuze fragmenten, waarbij er niet gelachen mag worden, te testen of de emotie blijheid effectief afwezig is. Wanneer de getoonde blijheid 0,9 bedraagt en de verwachte emotie 0,0, zal dit zich ook uiten als een grotere distance en kleinere similariteit waardoor de score lager zal liggen.

Bedekkingen van het gezicht en gesloten ogen

Testpersonen die de content te eng vonden om naar te kijken, reageren hier op verschillende manieren op. Testpersoon 4 houdt vaak de handen voor de mond, draait het hoofd en sluit de ogen bij de meest intense stukken van video 4, 5 en 6. Ondanks dat deze persoon dit gedrag wel vertoont en het ook zichtbaar is op de genomen foto's, heeft Microsoft de bedekking van de mond niet correct geregistreerd. Indien Microsoft dit wel correct had geregistreerd, zou de negatieve feedbackscore verhoogd worden door de applicatie waardoor er rekening zou worden gehouden in de totale score. Face++ detecteerde wel correct de gesloten ogen bij deze testpersoon. Testpersoon 20 had bij video 4 bij één foto de handen voor de mond en dit werd wel correct geregistreerd. Na de testen met set 1 werd besloten om gedetecteerde gesloten ogen meer te laten meetellen en een bedekt voorhoofd enkel ter informatie bij te houden, maar er niets mee te doen. Deze aanpassing werd in de code uitgevoerd en de aandachtresultaten van set 1 werden hiernaar aangepast door de app de data van de testpersonen opnieuw te berekenen. Alle sets werken op deze definitieve manier waardoor het aandachtsniveau bij de vier sets steeds exact dezelfde waarde heeft.

Testpersoon 8 hield bij het kijken naar video 6 vaak de handen voor de mond. Wanneer de handen het gezicht te veel bedekken, kan Microsoft geen gezicht detecteren en zal er geen foto per aanwezige persoon worden opgeslagen. Figuur C.14 in [Bijlage C: Grafieken, diagrammen, code en tabellen](#) toont het emotiespectrum van testpersoon 8 bij video 6 waarin te zien is dat er enkele tussenliggende meetpunten ontbreken tussen seconde 70 en 200. Dit is te verklaren door het falen van de gezichtsdetectie van Microsoft waardoor er geen probabiliteiten van foto's op die tijdstippen konden berekend worden. Het tekort aan foto's, ook al heeft de persoon de film volledig bekeken, resulteert in een grotere negatieve feedbackscore van de sharedScore waardoor de totale distance-score op 6,55/10 berekend wordt.

Het kan gebeuren dat gezichten niet gedetecteerd worden doordat mensen met hun handen hun gezicht ondersteunen, in een slechte hoek zitten of omdat er onvoldoende belichting is. Het probleem kan opgelost worden testpersonen uitdrukkelijk te vragen hun gezicht niet te veel te bedekken en enkel te bedekken indien het een reactie is op de content. Indien het een reactie is op de content en het gezicht wordt niet gedetecteerd in de foto, kan het gebruikt worden als negatieve feedback doordat het systeem denkt dat de persoon niet gedurende de volledige video aanwezig was. Bij testpersoon 8 en video 6 maakte het systeem op deze manier een correcte inschatting van de beleving van de content.

Het valt tenslotte op dat de foto's, waarop deze gedragingen getoond worden, vaak slechts een fractie uitmaken van het totale aantal foto's. Hierdoor is hun bijdrage op dit moment verwaarloosbaar en het kan overwogen worden om wanneer deze gedragingen gedetecteerd worden, ze meer te laten doorwegen.

Spectra van eenzelfde film voor verschillende kijkers

Op basis van de geplote emotiespectra van de kijkers kan gekeken worden of verschillende kijkers dezelfde emoties vertonen bij dezelfde film. In [Bijlage C: Grafieken, diagrammen, code en tabellen](#) tonen Figuur C.13 en Figuur C.14 de emotiespectra van vier interessante testpersonen voor de zes video's. Voor video 1 is het duidelijk dat testpersoon 3 de content minder grappig vond dan de andere personen aangezien de probabilliteit van blijheid veel lager ligt. Zowel testpersoon 13, 8 en 4 vertonen een kleine probabilliteit van verdriet op het einde van video 2. Testpersoon 13 is de enige die bij video 3 wat boosheid toont, de emotie die verwacht wordt, en de probabilliteit hiervan ligt ook heel laag. Bij video 4 gebeurt het vaak dat kijkers moeten lachen met het fragment maar bij testpersoon 13 komt boosheid opnieuw wat meer naar boven en ook een kleine probabilliteit voor de verwachte emotie afkeer. Video 5 maakt vaak mensen aan het lachen vanaf seconde 140, maar opnieuw toont testpersoon 13 gedurende het volledige fragment veel verschillende emoties. Niemand vertoont de verwachte emotie angst met hoge probabilliteit. Video 6 zorgt voor enkele plotse kleine pieken voor de emotie verrassing. Het is uitzonderlijk dat een testpersoon zo veel emoties toont zoals testpersoon 13. Meestal bevatten de spectra eerder neutrale emoties zoals bij testpersoon 3 en in het extreme geval zoals testpersoon 6 in Figuur 5.1.

Uit deze spectra kan opnieuw besloten worden dat de emoties afkeer, boosheid en angst vaak gedetecteerd worden met een lage probabilliteit die nog lager ligt bij de foto's van deze testpersonen dan die uit de emotietesten. Ook de emotie verdriet en verrassing worden met lage waarden gedetecteerd terwijl blijheid steeds snel hoge waarschijnlijkheden behaalt.

Overheersende emotie over het gehele fragment bepalen

Voor elke kijkbeurt wordt er per video bepaald wat de sterkste en meest overheersende emotie van de gebruiker was op basis van de hoogste probabilliteitswaarde per seconde. Dit wordt bijgehouden in het veld highestEmotion van de ViewHistoryrecords. Voor testpersonen 1 en 8 bijvoorbeeld blijkt dat hun meest aanwezige emotie lachen was bij video 1. Bij persoon 6, die vaak een hogere probabilliteit voor verdriet had, is verdriet de overheersende emotie. De modus (in te stellen met de METHOD-constante in de FaceApplicationklasse) waarin set 1 wordt uitgevoerd is ALLSECONDS en bijgevolg wordt highestEmotion over alle seconden van het fragment bepaald.

In het ideale geval zou de sterkste emotie voor elke video gelijk zijn aan de verwachte overheersende emotie zoals beschreven in Tabel 5.1. Bij het bekijken van de resultaten van de testpersonen is het duidelijk dat dit niet het geval is. Bij video 1 en 2 komt het het vaakst voor dat de sterkste emotie wel overeenkomt, maar de andere video's vertonen, zoals bij testpersoon 4 bijvoorbeeld, vaak de emotie verdriet aangezien

dit de hoogste probabilliteit gaf indien de gebruiker geen expressie op het gezicht vertoonde. Het kan dus zijn dat de emotie verdriet de toevallig sterkste emotie is in video 2 aangezien verdriet ook veel voorkomt als sterkste emotie in de andere fragmenten.

Vaak voorkomende problemen en hun oplossing

Soms gebeurt het dat een kijker, initieel met bijvoorbeeld ID 1, in de helft van een film opeens wordt aanzien als een nieuwe gebruiker die ID 2 toegekend krijgt. Dit kan gebeuren als de persoon voor de gezichtsherkenning er even anders uitziet dan het systeem kent door bijvoorbeeld een uitgesproken emotie te tonen. Hierdoor maakt het systeem HistoryDatarecords voor twee verschillende gebruikers aan en wordt de similariteit ook voor twee verschillende gebruikers berekend. Het probleem kan voorkomen worden door de gezichten van de gebruikers meerdere keren te trainen zodat de identificatie meer kans heeft op slagen. Indien de identificatie bij de testen toch eens mislukt was, is de oplossing hiervoor om manueel de HistoryDatarecords samen te zetten en te laten doorgaan als dezelfde gebruiker. De data worden dan opnieuw in de app ingelezen en verwerkt om de uiteindelijke similariteit te bepalen.

5.2.2 Kwaliteit van set 1 met de switchtest

Met een emotiefingerprint eigen aan de film kan veel bepaald worden. Mogen emoties van blijheid verwacht worden bij een horrorfilm? Betekent dit dat de gebruiker de film positief of negatief beoordeelt? Worden kijkers ook beoordeeld op het afwezig zijn van emoties bij bepaalde scènes? De initiële emotiefingerprints uit set 1 zijn niet ideaal aangezien ze geen rekening houden met deze opmerkingen en de verschillende mogelijke emoties van gebruikers. Een tweede set kan gemaakt worden door het gemiddelde van de getoonde emoties van de testpersonen te nemen per tijdstip en zo een meer realistische emotiefingerprint te bouwen.

Zorgen dat de geschatte score voor een video zo goed mogelijk overeenkomt met de expliciete score van de persoon over die film is niet de focus van de masterproef. Het is onmogelijk exact af te leiden wat iemand van een film vond, enkel en alleen op basis van zijn emoties. Het is van groter belang te zorgen dat de applicatie op basis van de getoonde emoties van een persoon kan afleiden naar welke film de persoon aan het kijken was en op basis hiervan een score kan toekennen aan de overeenkomst met de verwachte film. De manier waarop dit getest kan worden, is door bijvoorbeeld de getoonde emoties van persoon A bij video 1 te laten doorgaan alsof ze bij video 1, 2, 3, 4, 5 en 6 horen. Hierdoor past de applicatie de bijhorende verwachte emoties van video 1, 2, 3, 4, 5 en 6 één voor één toe op de getoonde emoties van video 1. De applicatie weet niet van welke video de getoonde emoties afkomstig zijn en kan dit enkel “raden” op basis van de berekende similariteitscore, die het hoogst zou moeten zijn bij toepassing van de verwachte emoties van video 1. De score bij de andere verwachte emoties zou lager moeten liggen aangezien de probabilliteiten minder overeenkomen. Deze test wordt de switchtest genoemd en de resultaten zijn terug te vinden in de map *ExpectedEmotions/SwitchTest*. Wanneer de verwachte emoties van video 1 als input van de switchtest worden gegeven, krijgt video 1 een emotionScore van 6/6 aangezien deze exact dezelfde zijn en de similariteit dus 100 % bedraagt.

Elke set zal een switchtest ondergaan waarbij er volledig getest wordt op de emoties, met behulp van typicalEmotionScore en emotionScore, en niet meer op de som met het aandachtsniveau. Voor elke film worden de getoonde emoties gekozen van één testpersoon die goede emoties uitte tijdens het kijken en de film een goede expliciete score gaf. De inputbestanden die hiervoor gemaakt werden, staan in de map *ExpectedEmotions/SwitchTest/Input*. De test wordt op de vier sets toegepast en zal een maat geven van de kwaliteit van de set. De bestanden voor de switchtest zijn als volgt opgebouwd en een deel van bestand *switchTest_set1.xlsx* is te zien in Tabel 5.4. Elke film komt één keer aan bod en wordt telkens voorgedaan alsof het video 1, 2, 3, 4, 5 en 6 is. Het echte video-ID staat in de eerste kolom met daarna de testpersoon van wie de data afkomstig is. Het video-ID dat gebruikt wordt voor de verwachte emotie staat in de derde kolom. In Tabel 5.4 zou de score voor video 1 de hoogste moeten zijn aangezien de data hiervan afkomstig is. De hoogste score wordt in het vet gezet. De kwaliteit van een set wordt bepaald door het aantal keren dat de video met de hoogste score overeenkomt met de video die de hoogste score zou moeten hebben.

Tabel 5.4: Resultaat van de switchtest bij de verwachte emoties uit set 1 voor video 1

videoid	testuserid	expected emotions videoid	emotion	attention	typical emotion
1	1	1	5.6730	3.57576	5.0678
1	1	2	5.6321	3.57576	5.8829
1	1	3	4.8388	3.57576	3.8398
1	1	4	5.1386	3.57576	4.9980
1	1	5	5.3437	3.57576	5.8379
1	1	6	4.7156	3.57576	4.6228

In Tabel 5.4 haalt de geschatte score van typicalEmotionScore van de juiste video niet de hoogste score. Op basis van de typicalEmotionScore kan dus niet bepaald worden over welke video het echt gaat. De emotionScore daarentegen haalt wel voor de juiste video de correcte score. Het totaal aantal keren dat de emotionScore de juiste video raadt is 2/6 en bij typicalEmotionScore is dit 1/6. De typerende emotie is blijheid bij video 1. Dit is de video waarbij de typicalEmotionScore van set 1 gefaald heeft, terwijl dit net de makkelijkste video zou moeten zijn omdat blijheid het makkelijkste te detecteren is. Er kan besloten worden dat set 1 nog kan verbeterd worden.

5.3 Nieuwe emotieset op basis van de emoties van de testpersonen: set 2

Voor de emotiefingerprints van set 2 wordt per tijdstip het gemiddelde van de emoties van enkele testpersonen gebruikt. Enkel de testpersonen die de video leuk vonden en een positieve expliciete feedback gaven, komen in aanmerking aangezien enkel deze emoties kunnen aanzien worden als de verwachte emoties. [Bijlage B: Gebruikershandleiding](#) beschrijft de knop in de app die dit gemiddelde automatisch berekent. De data van alle testpersonen worden opnieuw in de app geladen en met set 2 worden de nieuwe scores berekend. De METHOD-constante in de FaceApplicationklasse, waarin set 2 wordt uitgevoerd is opnieuw ALLSECONDS. De bedoeling van set 2 is om meer realistische waarden te bekomen voor emoties die moeilijk te detecteren zijn en door het gemiddelde te nemen zal de set automatisch ook milder zijn voor moeilijke emoties.

5.3.1 Resultaten met het gebruik van set 2

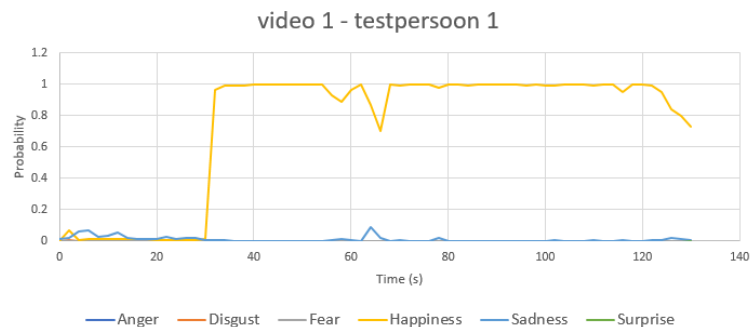
Vanzelfsprekend is elke attentionScore en sharedScore exact dezelfde als deze uit de resultaten van de eerste set. De enige verandering was het toevoegen van een nieuwe emotiefingerprint voor elke video waardoor de margins-score en distance-score aangepast worden.

Lage scorediversiteit tussen de verschillende video's omdat de gemiddelde waarden vaak heel laag liggen

Zoals te zien in de grafieken van set 2 in [Figuur C.11](#) en [Figuur C.12](#) in [Bijlage C: Grafieken, diagrammen, code en tabellen](#), zorgt het gemiddelde ervoor dat eventuele hogere probabiliteiten van één testpersoon verlaagd worden naar een lagere gemiddelde probabiliteit. Hierdoor gaan enkele verwachte emoties wat verloren maar leunen ze wel beter aan bij de realiteit. Door de lagere verwachte waarden voldoen de getoonde emoties heel snel aan de verwachte. Hierdoor scoren alle films vrij hoog en steeds dicht bij een 6/6. Dit geldt niet bij video 1 zoals hieronder verder wordt uitgelegd. Bij alle testpersonen krijgt men dus hetzelfde patroon en dit resulteert in weinig diversiteit.

Video 1 scoort steeds lager door foute verwachte emoties

Set 2 zorgt ervoor dat video 1 een bijna continue probabilliteit tussen 0,1 en 0,8 heeft. Mensen die in het begin van het fragment niet lachen en op het einde wel lachen met een probabilliteit van 1,0 worden hierop afgestraft door een lage similariteit met hetgeen er verwacht wordt. Wanneer de kijker in het begin niet lacht en op het einde wel, zou dit net moeten aanzien worden als positieve feedback. Testpersoon 1 bijvoorbeeld geeft in zijn getoonde emoties bij video 1 aan dat hij veel heeft gelachen op het einde, zoals te zien in Figuur 5.2, maar toch bedraagt zijn typicalEmotionScore slechts 4,49/6 terwijl dit bij de andere video's 5,80 of meer bedraagt. Deze scores komen ook vaak voor bij de andere testpersonen en dus kan er besloten worden dat deze set ook geen goede resultaten geeft.



Figuur 5.2: De getoonde emoties van testpersoon 1 bij videofragment 1

Niet alle stukken uit de video's zijn interessant

De video's bevatten vaak voor meer dan de helft stukken waarin geen emotie verwacht wordt. Aangezien elke seconde voor evenveel score meetelt, worden de belangrijkste scènes hierdoor wat verwaarloosd in de totale score. De verwachte emoties zouden dus beter rekening houden met stukken waarin helemaal geen emoties gedetecteerd worden, zoals saaier stukken in het fragment of intro's, zodat enkel de belangrijkste scènes de totale score beïnvloeden. Bij video 1 kan er zo bijvoorbeeld beter vastgelegd worden dat er tot seconde 50 geen emoties mogen meetellen en daarna wel, zoals bij testpersoon 1 in Figuur 5.2.

Te veel onregelmatigheden in de gemiddelde spectra

Zoals te zien in Figuur C.11 en Figuur C.12, bevatten de grafieken uit set 2 heel wat “grillige” lijnen. Een afgevlakt verwacht spectrum met meer continue lijnen is een meer logische optie.

5.3.2 Kwaliteit van set 2 met de switchtest

Opnieuw wordt een switchtest uitgevoerd op de set die de HistoryDatarecords eens laat voordoen alsof ze van een andere video zijn. Het aantal keer dat deze set de juiste video raadt, is bij de emotionScore gelijk aan 4/6 en bij de typicalEmotionScore is dit 1/6. Dit lijkt een goed resultaat maar toch is deze set niet zo nauwkeurig als ze lijkt. Ten eerste slaagt geen enkele emotiescore erin om video 1 goed te raden. Dit zou de makkelijkste video moeten zijn om te raden aangezien de sterkste emotie blijheid is. Ten tweede, wanneer bij de vier juiste geraden scores wordt gekeken naar het verschil met de tweede hoogste score, wordt het duidelijk dat het gemiddelde verschil tussen deze twee slechts 0,05 bedraagt. Dit zorgt ervoor dat de set niet zorgt voor een duidelijk onderscheid tussen de verschillende scores. Dit probleem is te wijten aan de lage diversiteit in scores voor de verschillende films en hierdoor liggen alle emotionScores heel dicht bij elkaar. De kwaliteit van een set verbetert naarmate het onderscheid tussen de emotionScores groter wordt. Bijgevolg is het nieuwe streefdoel een set te vinden die een groter onderscheid kan maken.

5.4 Verbeteren van de verwachte emoties met set 3

Een set waarin de grafieken rekening houden met de oninteressante stukken van het fragment zou een betere optie zijn. De interessante en veelbepalende stukken worden op deze manier benadrukt en moeten manueel worden aangeduid. Hiervoor wordt set 2 aangepast naar set 3 door manueel de oninteressante emoties op bepaalde stukken op nul te zetten. De code moet er dan voor zorgen dat de oninteressante stukken overgeslagen worden. Dit kan gerealiseerd worden door de `METHOD`-constante in de `FaceApplication`klasse op `SKIPSECONDS` in te stellen waarover meer uitleg wordt gegeven in [Bijlage B: Gebruikershandleiding](#). Wanneer een emotiewaarde in de verwachte emoties gelijk is aan nul, wordt er op die manier geen rekening gehouden met de getoonde emoties van de kijker op dat moment. [Listing C.3](#) (vanaf lijn 38) en [Listing C.4](#) (vanaf lijn 9) tonen hoe dit in de code gerealiseerd wordt.

Set 3 brengt ook een verbetering in het grillige patroon van set 2 door per 10 seconden en per emotie de maximale waarde te bepalen. De maximale waarde wordt dan over diezelfde 10 seconden als de probabilliteit van die emotie ingesteld.

5.4.1 Resultaten met het gebruik van set 2

Aangezien set 2 als basis voor set 3 gebruikt werd, lijken de resultaten sterk op die van set 2. Een overzicht van alle testpersonen met hun berekende waarden is terug te vinden in [Bijlage C: Grafieken, diagrammen, code en tabellen](#) bij [Tabel C.3](#) en [Tabel C.4](#).

Gemiddelden van de verschillen tussen expliciete score en geschatte score

[Tabel C.3](#) geeft per film de resultaten van alle personen per video voor de overheersende emotie bekomen met set 3 (in de kolom `highest emotion`). De tabel toont ook de `emotionScore` en de `typicalEmotionScore` op een totaal van 6 en de grootste waarde werd per persoon in het vet gezet. De tweede tabel van set 3, [Tabel C.4](#), bevat per kijker en per film nog een extra berekening die de absolute waarde van het verschil weergeeft tussen de expliciete score op 10 van de kijker en zowel de `emotionScore` als de `typicalEmotionScore` (beide afgebeeld op een totaalscore van 10). Bij set 3 kan besloten worden dat het gemiddelde van alle absolute verschillen het grootste is bij video 4 (de video met afkeer als overheersende emotie). Dit geldt voor de twee kolommen. Dit betekent dat de app bij een video met afkeer het moeilijkste de expliciete score van de gebruiker kan schatten. Een video met verrassing blijkt het tweede moeilijkste te schatten bij set 3.

Er kan ook geconstateerd worden dat de kijkers verschillen tonen in hun expliciete score. Een kijker die een film half goed vond, geeft bijvoorbeeld een score van 1 terwijl een kijker die dezelfde film slecht vond, bijvoorbeeld een score van 4 gaf. De inschatting van de scores van gebruikers is dus niet iets waarop gebaseerd moet worden en kan worden opgelost door de expliciete scores te normaliseren. Zoals reeds eerder vermeld, ligt de focus van de masterproef niet op het correct inschatten van de expliciete feedback maar deze waarden vertellen wel al veel over de mate waarin de emoties van kijkers overeenkomen met wat ze echt vinden van de film.

Nog steeds lage scorediversiteit

De overeenkomst voor testpersoon 1 bij video 1 is gestegen van 4,49/6 naar 4,73/6, maar nog steeds hebben de andere video's een hogere score. Bij de andere vijf video's liggen de scores opnieuw rond de 5,80/6 en ook bij de andere testpersonen wordt dit patroon verkregen. De diversiteit in scoreverdeling lag beter bij set 1. Het gemiddelde van `typicalEmotionScore` en `emotionScore` per video over alle gebruikers werd berekend in het bestand `Testing/InfoAllUsers` en wordt getoond in [Tabel C.3](#). In deze tabel staat steeds per kijker de film met de hoogste score in het vet aangeduid. Het valt op dat er bij de `emotionScore` en `typicalEmotionScore` steeds één film het meeste aantal hoogste scores heeft wat wijst op een slechte diversiteit. Gemiddeld over alle gebruikers haalt video 3 de hoogste score van alle films, net zoals in set 2, met een `emotionScore` van 5,83/6 en scoort deze video het hoogste bij de testpersonen waardoor vaak dezelfde films, die veel tags gemeenschappelijk hebben met video 3, worden aangeraden.

Overheersende emoties aangepast dankzij andere focus

Dankzij het aanduiden van de interessante stukken van het fragment, wordt ook de overheersende emotie enkel op deze stukken berekend. Bijvoorbeeld bij testpersoon 8 was de overheersende emotie van video 2 in set 1 en set 2 (over het gehele fragment berekend) steeds gelijk aan blijheid. Met de afbakening van de belangrijke scènes en het gebruik van SKIPSECONDS, wordt de overheersende emotie hiervan veranderd naar verdriet. Aangezien de verwachte overheersende emotie van video 2 gelijk is aan verdriet, is dit een verbetering ten opzichte van de ALLSECONDS-methode.

Er kan ook besloten worden dat geen enkele kijker bij video 4, 5 en 6 de verwachte emotie als overheersende emotie toont, zoals te zien in Tabel C.3 bij de 'highest emotion'-kolom. Bij video 1 vertonen 15/18 testpersonen de emotie blijheid en bij video 2 vertonen 18/20 personen de emotie verdriet. Bij video 3 vertonen slechts 2/18 testpersonen de verwachte emotie boosheid. Wanneer een testpersoon niet de verwachte emotie als overheersende emotie heeft, toont de persoon vaak de emotie verdriet of blijheid in de plaats. Dit kan verklaard worden door het feit dat de testpersonen vaak neutraal kijken en deze emotie kan verward worden met verdriet. Bij video 4 lachen de kijkers vaak in plaats van afkeer te tonen waardoor de overheersende emotie wordt bepaald als blijheid. Hieruit kan besloten worden dat video 1 de beste resultaten behaalt aangezien de emotie blijheid niet verward werd met een andere emotie en dat de emoties van de andere fragmenten vaak niet voldoende aanwezig zijn om overheersend te zijn.

Tabblad 'Tags' toegevoegd

Voor de resultaten van set 3 werd het tabblad 'Tags' toegevoegd aan de Excelbestanden. Hierin worden voor alle tags de expliciete score van de kijker voor die tag vergeleken met wat de applicatie denkt dat de kijker van de tag vindt. De applicatie berekent deze waarden door voor elke bekeken film aan de bijhorende tags de scoreDistance van de film te geven en vervolgens het gemiddelde te nemen van alle waarden per tag. Dit is dezelfde berekening die wordt uitgevoerd voor het maken van een aanbeveling, zoals uitgelegd in 4.3.2 in Hoofdstuk 4. Ook wordt in dit tabblad de gegenereerde aanbeveling geplaatst volledig op basis van de net bekeken films van de gebruiker. Voor set 3 komt het heel vaak voor de film 'The Magnificent Seven' wordt aangeraden met de hoogste score. Dit komt omdat deze film veel tags gemeenschappelijk heeft met video 3, namelijk 'War' en 'Soldier/Fight/Violence'. Bovendien krijgt video 3 bij vele testpersonen de grootste similariteit voor het emotiespectrum over alle films. De verwachte emotie boosheid heeft in de emotiefingerprint een heel lage probabilliteit waardoor de emoties van kijkers hier snel aan voldoen, ook al vertonen ze helemaal geen boosheid. Hierdoor worden films die lijken op film 3 met een hogere score aangeraden.

5.4.2 Kwaliteit van set 3 met de switchtest

Opnieuw liggen de emotionScores heel dicht bij elkaar en kan er nog verbetering worden aangebracht aan de verwachte emotiewaarden. Het aantal keer dat de emotionScore de juiste video raadt is 2/6 en bij typicalEmotionScore is dit 1/6. Opnieuw wordt video 1 niet correct geraden terwijl deze de makkelijkste zou moeten zijn. De resultaten van set 3 zijn bijgevolg zeker niet beter dan die van set 2.

5.5 Optimaliseren resultaten met set 4

Er wordt uiteindelijk geopteerd om een volledig nieuwe set aan te maken die nog meer de interessante scènes afbakt en de andere emotiewaarden op nul zet. Set 4 zal streven naar een betere overeenkomst in similariteit tussen de verwachte en getoonde emoties. De set duidt tot op de seconde aan wanneer er een bepaalde emotie aanwezig moet zijn. Enkel met de aangeduide stukken wordt rekening gehouden met de similariteitsbepaling en alle andere stukken worden genegeerd. Een stuk waarin bijvoorbeeld verwacht wordt dat iemand de emotie verrassing toont, ook al is dit slechts 10 seconden, kan op deze manier duidelijk worden afgebakend en wordt bijgevolg benadrukt.

Ook worden meerdere overheersende emoties toegelaten per video. Zo krijgt video 2, naast verdriet, ook de overheersende emotie blijheid waardoor beide emoties gewicht 3 krijgen bij de typicalEmotionScore. Bij video 4 mogen testpersonen ook lachen en krijgen zowel afkeer als blijheid een hoog gewicht. Set 4

krijgt dus als invoer een aangepast movies.json bestand met meerdere overheersende emoties per video. Het gebruik van de SKIPSECONDS-methode werkte voordelig voor set 3 en zal bijgevolg in set 4 opnieuw worden toegepast.

5.5.1 Resultaten met het gebruik van set 4

De HistoryDatarecords van alle testpersonen worden opnieuw verwerkt met de verwachte emoties van set 4 en hieronder worden de resultaten besproken. Een overzicht van de typicalEmotionScore en het verschil met de expliciete score per testpersoon per film is te vinden in Tabel C.5 en Tabel C.6.

Waarden die minder aanleunen tegen de maximale 6/6

Meteen valt het op dat de impliciete scores, zowel emotionScore als typicalEmotionScore, nu lager liggen dan bij set 2 en 3 en al zeker niet meer dichtbij de maximale waarde van 6/6. De verklaring hiervoor is dat de probabiliteiten in set 4 hoger liggen dan in set 2 en 3 waardoor het moeilijker is voor een testpersoon om een hoge similariteit te verkrijgen.

Opnieuw hoge scores voor video 3

De emotionScore en typicalEmotionScore van video 3 ligt bij alle testpersonen hoger dan de score van de andere video's. Vele van de testpersonen gaven aan dit fragment het beste uit de zes te vinden. De hoge score is deels ook te verklaren door de probabiliteiten van set 4. Set 4 voorziet in de belangrijke stukken van video 3 een probabieliteit van 0,3 voor boosheid. De waarde werd zo laag gekozen omdat geen enkele testpersoon hoger dan deze waarde kon scoren voor boosheid. Maar de lage waarde zorgt er ook voor dat de testpersonen opnieuw heel snel voldoen aan de verwachte emoties en op die manier een hoge similariteit bekomen. Het gevolg van de hoge score voor video 3 uit zich opnieuw in de aanbeveling die bij 100,00 % van de testpersonen (90,00 % indien de twee kinderen worden meegerekend) de film 'The Magnificent Seven' aanraadt op plaats 1 doordat deze film enkele tags gemeenschappelijk heeft met video 3. Deze aanbeveling is zeker niet ideaal en zal deels worden opgelost door de aanbeveling enkel te baseren op de typicalEmotionScore, zoals toegepast in set 4.2 en wordt hieronder verder uitgelegd in 5.5.2.

Video 1 scoort correct

Een eigenschap die vaak terugkomt bij set 4 is dat video 1 vaak hoog scoort bij de testpersonen met de typicalEmotionScore, indien de testpersoon veel heeft gelachen. Bij set 2 en 3 behaalde video 1 vaak een lage score doordat de verwachte emotie niet hoog genoeg lag. Bij set 4 ligt de verwachte emotie telkens op 1. Wanneer een persoon video 1 echt leuk vindt en veel moet lachen, behaalt deze steeds probabiliteiten van 1 en krijgt men een hoge similariteitscore. Testpersoon 8 haalt op die manier een typicalEmotionScore van 5,78/6, een hoge similariteit die ook reeds af te leiden was uit Figuur C.13. Testpersoon 3 vond video 1 niet zo aangenaam - wat ook reeds bleek uit Figuur C.13 - en deze testpersoon haalde een typicalEmotionScore van 3,13/6. Er kan dus besloten worden dat set 4 heel goed werkt voor de emotie blijheid dankzij de hoge probabiliteiten en de definiëring van de grappige scènes.

Hogere scorediversiteit

Elke testpersoon heeft meer diverse scores voor de verschillende video's en ook de testpersonen onderling hebben andere score voor eenzelfde video. Vooral de scores voor video 1 verschillen duidelijk tussen de testpersonen. Zoals hierboven reeds aangehaald, bestaat er een verschil van 2,65/6 in typicalEmotionScore voor video 1 tussen testpersoon 8 en 3. Er kan besloten worden dat video 1 en de emotie blijheid in het algemeen de meest detecteerbare emotie is en de beste similariteitwaarden oplevert. Ook kan besloten worden dat typicalEmotionScore betere resultaten geeft dan emotionScore en met behulp van set 4.2, hieronder uitgelegd in 5.5.2, zal deze verbetering worden aangebracht.

Overheersende emoties opnieuw nauwkeuriger bepaald

De nieuwe afbakening van de interessante scènes zorgt voor nieuwe overheersende emoties per video. Zoals te zien in Tabel C.5 in de ‘highest emotion’-kolom, behalen 15/18 personen opnieuw een overheersende emotie blijheid bij video 1. Slechts 10/20 personen behalen de emotie verdriet bij video 2. De reden voor dit aantal is omdat set 4 de droevige scènes in video 2 veel fijner heeft aangeduid waardoor er in totaal minder droevige emoties dan blij emoties worden verwacht, zoals te zien in Figuur C.11. De betere afbakening geeft ook geen betere resultaten bij video 3, 4, 5 en 6 waardoor besloten kan worden dat de kijkers de emoties boosheid, afkeer, angst en verrassing minder expressief tonen.

Gemiddelde verschil tussen expliciete score en typicalEmotionScore

De berekende scores bij set 4 liggen over het algemeen lager dan die bij set 3. Hierdoor is het verschil met de expliciete score ook veel kleiner. Tabel C.6 toont opnieuw voor elke gebruiker en elke film de absolute waarde van het verschil tussen de expliciete score en de typicalEmotionScore om te kijken hoe goed de typicalEmotionScore het doet. De gemiddelde waarde van al deze waarden komt bij video 4 neer op 2,92/10 en is daarmee opnieuw de grootste waarde. Hieruit kan besloten worden dat bij video 4 het het moeilijkste is om de expliciete scores van de kijkers te bepalen. De kleinste gemiddelde waarden zijn te vinden bij video 3, namelijk 0,84/10 verschil, en bij video 1, namelijk 1,56/10 verschil. Ook bij set 3 gaf video 1 het kleinste gemiddelde verschil. Dit betekent dat het bij video's met de emotie blijheid het makkelijkst is om de waarderingsscore van de kijkers in te schatten.

5.5.2 Set 4.2

Bij de resultaten van set 1 werd reeds de opmerking gemaakt dat het misschien beter kon zijn om typicalEmotionScore te gebruiken voor de berekening van de totale distance-score. In de initiële berekening van set 4 werd dit nog niet geïmplementeerd maar set 4.2 past dit wel toe. Set 4.2 gebruikt volledig dezelfde emotieprobabiliteiten per film als set 4 maar houdt bij de totale waarderingsscore van de gebruiker enkel rekening met de emotiescore van de - vooraf bepaalde - typerende, overheersende emotie bij die video. De scoreDistance is bij set 4.2 dus gelijk aan typicalEmotionScore + sharedScore. Bij set 4 was dit de emotionScore + sharedScore en de verschillen in scoreDistance tussen deze twee sets zijn te zien in Tabel C.6. De tabel toont per gebruiker de hoogste score in het vet. De aanbeveling bij set 4.2 is bijgevolg anders dan die bij set 4 aangezien deze enkel en alleen gebaseerd is op de waarde van scoreDistance van set 4.2. Om dit in de code in te stellen, moet de constante TOTAL_SCORE in FaceApplication ingesteld worden op ONLYMOST. In de *Vragenlijst testpersonen X.xlsx*-bestanden wordt dit toegevoegd door in ‘HistoryId’ scoreDistance te bepalen als som van typicalEmotionScore en sharedScore in de kolom na de resultaten van set 4.

Het resultaat van deze aanpassing is het best te demonstreren bij testpersoon 1. Deze testpersoon behaalde bij de typicalEmotionScore de maximale score bij video 1, namelijk 5,63/6. De emotionScore die het hoogste was, kwam voor bij video 3 met een waarde van 4,74. Set 4 bekwam op deze manier, door emotionScore te gebruiken, een aanbeveling met op de eerste plaats ‘The Magnificent Seven’, de film die zo goed als alle testpersonen aangeraden kregen. Set 4.2 werd gebaseerd op de typicalEmotionScore en zorgde voor een aanbeveling met een komedie op de eerste plaats. De hoogste typicalEmotionScore behoorde bij video 1 en aangezien de aangerade film veel tags gemeenschappelijk heeft met video 1, is het een komedie dat werd aangeraden.

Tabel C.5 toont dat de gemiddelde waarde van typicalEmotionScore bij video 3 neerkomt op 4,68/6 en 83,33 % van de testpersonen het hoogste scoort op video 3. Aangezien set 4.2 de typicalEmotionScore gebruikt, wordt het in Tabel C.6 duidelijk dat video 3 bij 77,78 % van de testpersonen de hoogste distance-score uit alle video's heeft in tegenstelling tot 100,00 % bij gebruik van emotionScore in set 4. Bij 3 testpersonen krijgt video 1 nu een hogere score en bij 1 testpersoon is dit video 2. Dankzij het gebruik van scoreDistance van set 4.2 wordt de film ‘The Magnificent Seven’ bij 83,33 % van de testpersonen aangeraden op de eerste plaats (75,00 % indien de twee kinderen worden meegerekend), in tegenstelling tot 100,00 % bij gebruik van scoreDistance van set 4. Dankzij gebruik van set 4.2 werd de aanbeveling dus iets beter gegenereerd maar er kan ook besloten worden dat de set nog steeds niet ideaal is.

Aangezien de kijkers video 3 vaak een hoge expliciete score gaven, is de aanbeveling niet foutief want ‘The Magnificent Seven’ is de film die het sterkst op video 3 lijkt. Er kan bijgevolg niet worden afgeleid of de film door toeval of door een goede verwachte emotieset werd aanbevolen. De eentonige aanbeveling kan ook deels verklaard worden door de beperkte verzameling van video’s in de applicatie (initieel zijn er slechts 30 fragmenten aanwezig). Bij uitbreiding kunnen meerdere video’s en tags toegevoegd worden en kan er een meer divers aanbod gegenereerd worden in de aanbeveling.

5.5.3 Kwaliteit van set 4 met de switchtest

Set 4 werd opgesteld door eerst de scènes te definiëren waarin potentieel de meeste emoties getoond worden. Vervolgens werden verschillende groottes van probabiliteitswaarden uitgetoetst om te kijken wat de beste overeenkomst gaf. De resultaten van de testsets vooraleer de uiteindelijke set 4 bepaald werd, zijn te zien in *switchTest_set4_Highest* en *switchTest_set4_Lower*. De keuze van de grootte van de verwachte emotieprobabiliteit is veelbepalend en hieronder wordt besproken welke besluiten werden vastgesteld.

Hoe lager de verwachte probabiliteiten, hoe sneller andere video’s ook een hoge similariteitscore behalen

Initieel werd de probabiliteit voor verdriet op 0,5 gezet bij video 2. Indien deze verlaagd werd naar 0,2, voldeed video 2 bij de switchtest heel goed aan de score en werd deze makkelijk bepaald. Bij het raden van andere video’s werden deze ook vaak als video 2 geraden omdat ze door de lage verwachte waarden heel snel voldoen aan de emotiefingerprint. Dit is duidelijk te zien in *switchTest_set4_Lower* met de resultaten van een testset met lage waarden dat aan de switchtest werd onderworpen. Hierin is te zien dat video 2 met de typicalEmotionScore 3/6 keer de hoogste waarde krijgt bij een foutieve video. De switchtest raadt dan wel correct video 2, maar ook 3 andere video’s worden als video 2 aanzien. Het aanpassen van de waarden in één emotiefingerprint beïnvloedt dus ook alle andere resultaten. In de switchtest heeft dit in het extreme geval tot gevolg dat alle emotiespectra similariteitswaarden hoge waarden voor alle video’s behalen (zoals de waarden dichtbij 6/6 bij set 2) en er weinig diversiteit bestaat tussen de verschillende video’s. Een te lage verwachte emotiewaarde is dus geen goed idee.

Hoe hoger de verwachte probabiliteiten, hoe moeilijker de similariteit hoog kan score

Het andere uiterste werd getest en bij een belangrijke scène werden alle verwachte emotieprobabiliteiten op 1 gezet. De resultaten hiervan zijn te zien in *switchTest_set4_Highest*. Deze resultaten tonen dat geen enkele persoon met zijn getoonde emoties nog konden voldoen aan de verwachte emoties, zelfs de video die overeen zou moeten komen. Alle waarden liggen hierbij iets boven 3/6. 3/6 is de minimale similariteitswaarde voor de 6 emoties samen aangezien een maximale distance van 1 een minimale similariteit van 1/2 tot gevolg heeft. Bij te hoge verwachte waarden behalen de emoties dus een waarde die te veel naar het minimum neigt. Er kan besloten worden dat er een waarde tussenin moet gevonden worden, gebaseerd op hoe goed een emotie te detecteren is.

Het zoeken naar de ideale waarde en de resultaten

De ideale waarde hangt af van hoe goed de emotie te detecteren is. Bij de emotietesten werd reeds duidelijk dat angst, afkeer en boosheid het moeilijkste te detecteren waren met een hoge probabiliteit. Deze emoties zullen al zeker geen probabiliteit van 1 krijgen aangezien dit onbereikbaar is, maar ook geen 0,2 aangezien dan andere video’s ook te snel hieraan zullen voldoen. Blijheid krijgt wel de waarde van 1 omdat gebleken is dat deze waarde snel behaald wordt door testpersonen. Uiteindelijk werd set 4 bekomen zoals ze is ingesteld in de applicatie, in *ExpectedEmotions/DifferentSets* te vinden is en zoals ze geplotted wordt in Figuur C.11 en Figuur C.12.

De uitgebalanceerde waarden van de uiteindelijke set 4 werden ook onderworpen aan de switchtest en deze resultaten zijn te zien in *switchTest_set4*. Merk op dat de switchtest voor set 4 en set 4.2 exact dezelfde waarden geeft aangezien het ook over exact hetzelfde verwachte emotiespectrum gaat en de switchtest

zowel de typicalEmotionScore (gebruikt in scoreDistance van set 4.2) als de emotionScore (gebruikt in scoreDistance van set 4) berekent. Het aantal correct geraden video's met de typicalEmotionScore is 3/6 en voor emotionScore is dit 2/6. Deze resultaten zijn beter dan die van set 2 want wanneer er opnieuw gekeken wordt naar het verschil met de tweede grootste score bij typicalEmotionScore, krijgt men een gemiddelde verschil van 0,53. Bij set 2 was dit verschil gelijk aan 0,05 en dit bewijst nog eens dat set 4 zorgt voor een veel grotere diversiteit in scores tussen de verschillende video's. Bovendien raadt set 4 video 1 wel goed en met een hoog verschil tegenover de andere videowaarden dankzij de grotere diversiteit.

Het onderscheid wordt het beste gemaakt met de typicalEmotionScore aangezien met deze score meer juist geraden video's voorkwamen. Dit is de reden waarom de TOTAL_SCORE na set 4 werd aangepast naar ONLYMOST voor set 4.2 zodat er enkel rekening wordt gehouden met de typicalEmotionScore. Ook werd er bij deze definitieve set bij de typicalEmotionScore geen grote fout meer gemaakt zoals het geval was in *switchTest_set4_Lower* (doordat 3 verkeerde video's als video 2 werden aanzien met de hoogste scores).

Switchtest met zichzelf

Indien de verwachte emoties van een video als input worden gegeven voor de switchtest met de zes video's, wordt de score 6/6 aangezien deze exact overeenkomen. Wanneer de belangrijke scènes enkele seconden verschoven worden naar een ander moment, zodat een andere film gesimuleerd wordt, krijgt deze een veel lagere score. De totale hoeveelheid emotie is dan wel hetzelfde maar als de scènes verschoven worden, kloppen de tijdstippen met de getoonde emoties niet meer waardoor dit zich zal uiten in een grotere distance en bijgevolg een kleinere similariteit. Dit bewijst dat set 4 erin slaagt video's te herkennen dankzij de fijne definiëring van scènes op verschillende tijdstippen.

5.6 Resultaten bevraging testpersonen en bedenkingen

Er werden aan de testpersonen ook enkele vragen gesteld. De antwoorden van alle testpersonen werden samen gezet in het *InfoAllUsers*-bestand en worden hier deel per deel overlopen. Allereerst werd hun leeftijd en geslacht genoteerd om opnieuw een onderzoek te doen naar de mate waarin de drie services de leeftijd correct schatten. De resultaten hiervan zijn terug te vinden in Tabel C.7 in [Bijlage C: Grafieken, diagrammen, code en tabellen](#) die per testpersoon de vijf geschatte leeftijden toont van elke service. Opnieuw blijkt hieruit dat Microsoft de beste inschattingen maakt. Voor de 20 testpersonen is de gemiddelde fout (op basis van de absolute verschillen tussen de gemiddelde geschatte leeftijd en de echte leeftijd) slechts 5,36 jaar. Voor Kairos is dit 7,66 jaar en voor Face++ zelfs 12,66 jaar. De 'gender'-kolom in de tabel toont het resultaat van het geslacht van die service op basis van voting (vijf keer stemmen op het geslacht en het geslacht met de meeste stemmen staat in de kolom). Geen enkele service schatte het geslacht van een testpersoon foutief in. Deze resultaten bevestigen nogmaals dat het een goede keuze was het geslacht door de drie services te laten bepalen en de leeftijd enkel door Microsoft.

Alle testpersonen, behalve de twee kinderen, beantwoordden ook enkele andere vragen. Het overzicht voor alle testpersonen is terug te vinden in Tabel C.8. Allereerst werd gevraagd of ze een dergelijke app zouden gebruiken indien deze op de markt zou komen. Slechts 16,67 % zou de app gebruiken indien deze werkt met een service die vereist dat de genomen foto's over het internet verstuurd worden. 94,44 % van de testpersonen zou liever hebben dat de foto's lokaal verwerkt worden aangezien het opsturen van foto's over het internet een gevoelig onderwerp blijkt te zijn. De meeste mensen zouden de app vooral gebruiken op de laptop, namelijk 38,89 %. Als het gaat over het vertrouwen in de camera van het toestel, zouden de meeste mensen de camera van hun smart TV en laptop bedekken indien de app niet gebruikt zou worden.

Ook de bereidheid van de personen om expliciete beoordelingen te geven na het kijken van een film aan bestaande systemen zoals Telenet of Netflix werd getest. Het werd duidelijk dat 11/18 testpersonen geen zin hadden en 5/18 slechts soms zin hadden om dit via de afstandsbediening van de smart TV te doen aangezien dit niet gebruiksvriendelijk is. Een film beoordelen op de laptop of tablet lijkt een minder grote drempel voor de testpersonen aangezien dit enkel een muisklik of aanraking op het touchscreen vereist.

Hieruit kan besloten worden dat een app die een aanbeveling kan genereren op basis van impliciete feedback vooral handig kan zijn bij films op een smart TV.

Er werd ook gevraagd of de testpersonen het gevoel hadden dat ze veel aandacht hebben gegeven aan de fragmenten en of ze de zes emoties bij het kijken naar films snel zouden uiten. De resultaten hiervan zijn weergegeven in Tabel C.9. Hieruit bleek dat de emotie die mensen het snelst denken te uiten blijheid is met een percentage van 88,89 %. De minst geuite emotie volgens het testpaneel is boosheid met een percentage van slechts 16,67 %. Alle andere emoties worden volgens meer dan de helft van de testpersonen geuit tijdens het kijken naar films.

Andere interessante resultaten komen voort uit de bevraging van de mogelijke reacties. 77,78 % zou weggijken wanneer de content te eng is en 16,67 % zou het volledige gezicht bedekken. Indien een persoon weggijkt of het volledige gezicht bedekt, zou de app dit kunnen registreren als het afwezig zijn van een gezicht en dit correct zien als negatieve feedback. 33,33 % zou de ogen sluiten, 16,67 % zou de ogen bedekken en 22,22 % zou de mond bedekken wanneer de video te eng is. Ook deze impliciete negatieve feedback kan correct geregistreerd worden Face++ en Microsoft.

Er kan besloten worden dat uit deze antwoorden blijkt dat elke persoon de content op een andere manier kan ervaren. Bijgevolg kan het samenvoegen van alle soorten impliciete feedback een realistische schatting van de ervaring van de content maken. Indien meerdere soorten detectiesystemen voor impliciete feedback worden samengevoegd, wordt de kans verhoogd dat er voor alle personen iets bij zit dat gedetecteerd kan worden om de waarderingsscore nauwkeuriger te schatten. Ook blijkt dat lachen de emotie is die het merendeel van de mensen snel zal uiten. Dit betekent dat het schatten van een impliciete score bij blij en grappige films makkelijker moet zijn dan bij de andere emoties.

5.7 Besluiten

Meerdere factoren bepalen de uiteindelijke ingeschatte waarderingsscore. De resultaten uit de testen met twintig testpersonen vergelijken de deelscores van elke factor (emoties, aandacht en gedrag). Het werd snel duidelijk dat de inschatting met de emotiefactor de moeilijkste is en daarom werd er verder onderzoek gedaan naar een goede set van verwachte emoties waarmee de getoonde emoties kunnen vergeleken worden. Een goede verwachte emotiefingerprint heeft namelijk invloed op alle berekeningen in de applicatie. Wanneer het systeem denkt dat een gebruiker film A heel leuk vindt op basis van de emoties van de gebruiker, raadt het systeem alle films aan die tags gemeenschappelijk hebben met film A. Indien de similariteit voor film A foutief en onterecht was, krijgt de persoon dus films aangeraden die hij eigenlijk niet wilt en faalt het systeem. De applicatie lost dit op door deels een demografische aanbeveling er ook in te verwerken en naarmate een gebruiker meer en meer geschiedenis verzamelt, de aanbeveling meer en meer te baseren op de feedbackgeschiedenis. De resultaten van de testen werden echter enkel gebaseerd op de geschiedenis van de zes video's om zicht te krijgen op hoe goed de impliciete feedbackgeschiedenis kan verzameld en bepaald worden.

5.7.1 Emotiedetectie

Enkel en alleen op basis van de emoties wordt het moeilijk om een waarderingsscore te bepalen van de kijker. Tijdens de testen werd het duidelijk dat niet alle testpersonen bij alle films evenveel emoties tonen. De fragmenten werden zo gekozen dat ze op korte tijd veel emoties moeten opwekken maar bij de testpersonen kwam het vaak voor dat ze eerder een neutrale expressie vertoonden of lachten bij horrorfragmenten. Films met de emotie blijheid worden hierbij het beste geschat aangezien die probabiliteit makkelijk een hoge waarde krijgt. De maximale gedetecteerde waarde van blijheid over alle video's en alle testpersonen bedroeg 1,00, wat de hoogst mogelijke waarde is. Indien ze boosheid, angst, verrassing, afkeer of verdriet toonden, werd dit slechts gedetecteerd met een lage probabiliteit waardoor films met deze emoties moeilijker in te schatten zijn. De maximale waarde doorheen alle video's en alle testpersonen van deze vijf emoties lag maximum op 0,68 voor verdriet en slechts 0,27 voor angst. Er kan dus besloten worden dat deze emoties zowel moeilijk te uiten zijn als moeilijk te detecteren zijn.

De definiëring van verwachte emoties per tijdstip in de film dankzij een emotiespectrum in functie van de tijd (en niet enkel op basis van de totale hoeveelheid emotie) draagt ook bij tot een meer nauwkeurige inschatting van de waarderingsscore. De overheersende emotie voor elke kijkbeurt van elke persoon werd bepaald om te kijken of deze overeenkomt met de verwachte overheersende emotie. De overheersende emotie kwam vooral overeen met de verwachte emotie bij video 1 (blijheid) en video 2 (verdriet). Verdriet kwam bij het merendeel van de fragmenten naar voor als sterkste emotie van de kijker aangezien de emotiedetectie dit snel detecteert wanneer er sprake is van een neutrale emotie. Hierdoor kan er niet met zekerheid gezegd worden of de kijkers bij video 2 de juiste overheersende emotie hebben getoond. Opnieuw blijkt hieruit dat de emotie blijheid als beste kan worden ingeschat.

In het onderzoek werd vervolgens gekeken naar de kwaliteit van de emotiescorebepaling. Verschillende sets met verwachte emoties werden in de app verwerkt om er de similariteiten mee te bepalen. Set 4 verkreeg de beste resultaten bij de switchtest aangezien deze het best de juiste video kon raden met de grootste verschillen in scores tussen de video's onderling. De goede prestatie van set 4 kan verklaard worden door de goede definiëring van de interessante scènes en de goed uitgebalanceerde probabiliteiten. Set 4.2 bevat exact dezelfde verwachte emotiewaarden als set 4 maar houdt enkel rekening met de sterkste emoties van de video waardoor de distance-score enkel gebaseerd is op belangrijke scènes en belangrijke emoties. Set 4 (en dus ook set 4.2) is nog steeds niet ideaal aangezien video 3 nog steeds vaak de hoogste score toegekend krijgt en er daardoor een onverdeelde aanbeveling gegenereerd werd voor de testpersonen. Een betere verzameling van verwachte emoties op verwachte tijdstippen zou ontworpen kunnen worden bij verder onderzoek en bij meerdere testpersonen.

Volgende principes en richtlijnen kunnen gedefinieerd worden om de overeenkomst tussen twee emotiespectra te maximaliseren indien deze de emoties bij eenzelfde film bevatten. Ze bevatten de vaak voorkomende problemen bij emotiedetectie en bieden een oplossing hiervoor aan. Hierbij draait alles om het behalen van een hoge similariteit indien de kijkers de juiste emoties tonen en een lage similariteit indien ze verkeerde emoties tonen of de juiste emoties niet tonen.

1. Hou rekening met alle verwachte emoties in de video en negeer emoties die er niet toe doen

De verbetering van set 4.2 ten opzichte van set 4 bewees dat enkel de meest typerende emoties van een video mogen bijdragen tot de score. Emoties die er niet toe doen, zorgen er alleen maar voor dat de score te veel beïnvloed wordt door deze waarden waardoor de echt belangrijke emoties minder doorwegen in de score. Een video kan zowel meerdere emoties op verschillende tijdstippen verwachten als meerdere emoties op hetzelfde tijdstip. Het aanwezig zijn van meerdere emoties op eenzelfde tijdstip is een oplossing voor:

- Het probleem waarbij sommige emoties kunnen verward worden met een andere: indien meerdere emoties verwacht worden, kunnen ook de verwarde emoties hiermee een hoge score halen. Uit de emotietesten bleek bijvoorbeeld dat angst vaak verward werd met verdriet, afkeer en verrassing. Ook bleek dat afkeer vaak verward werd met boosheid. Indien iemand verwacht wordt boos te kijken, is het bijgevolg handig om ook afkeer te verwachten. Een persoon die boos kijkt maar verward wordt met afkeer, kan op die manier toch ook een hoge score behalen.
- Het feit dat mensen die een film leuk vinden, dit met meerdere emoties kunnen uiten. Bijvoorbeeld bij een horrorfilm kunnen zowel walging als blijheid gezien worden als positieve feedback, aangezien een kijker beide emoties kan uiten en bij beide emoties de film goed kan vinden.
- Het checken op het afwezig zijn van emoties: de afstandsfunctie houdt ook rekening met de verwachte probabiliteiten die gelijk zijn aan nul. Hoe hoger de probabiliteit van de getoonde emotie, hoe verder deze van de verwacht nulwaarde ligt en hoe kleiner de similariteit en bijgevolg en de waarderingsscore. Dit kan gebruikt worden bij fragmenten waarbij er echt niet mag gelachen worden, zoals bijvoorbeeld hele droevige stukken van een film. Een gebruiker die hierbij lacht, vindt dit stuk waarschijnlijk niet leuk en moet een lage waarderingsscore voor deze film krijgen.

Het komt er dus op neer om de focus te leggen op emoties die ertoe doen, zelfs indien het eigenlijk niet verwacht wordt dat die emotie aanwezig is.

2. Hou rekening met de moeilijk te uiten en moeilijk te detecteren emoties

Zoals gebleken uit de emotietesten en de maximale waarden van de testpersonen, wordt enkel blijheid

vaak met een hoge probabilmiteit gedetecteerd. Angst, afkeer, boosheid, verdriet en verrassing werden bij de testen slechts met een maximale waarde van 0,68 gedetecteerd en bijgevolg moet er rekening gehouden worden dat wanneer de verwachte waarden voor deze emoties te hoog ligt, geen enkele persoon een hoge similariteit kan behalen. Er moet een goede waarde worden gekozen, rekening houdend met volgende zaken:

- Indien een te lage waarde wordt gekozen (bijvoorbeeld 0,1), leunt deze waarde misschien wel meer aan bij de getoonde emoties van de gemiddelde persoon (zoals in set 2) maar heeft dit tot gevolg dat als een persoon een probabilmiteit van 0,0 haalt, het verschil met wat er verwacht wordt slechts 0,1 bedraagt waardoor de persoon een relatief hoge similariteit behaalt (ondanks dat die persoon de emotie eigenlijk niet heeft geuit). In de switchtest heeft dit tot gevolg dat alle emotiespectra hoge similariteitswaarden (rond 6/6 zoals in set 2) voor alle video's behalen, de verkeerde video's de hoogste scores krijgen (zoals in set 4 met lage waarden voorafgaand aan de definitieve set 4) en er weinig diversiteit tussen de verschillende video's bestaat.
- Indien een te hoge waarde wordt gekozen, is de verwachte hoge waarde onbereikbaar en kan zelfs de meest expressieve persoon geen hoge similariteit met deze waarde bereiken. In de switchtest heeft dit tot gevolg dat geen enkele emotiespectra nog hoge similariteitswaarden kan behalen, zelfs niet voor de correcte film. Alle waarden liggen dan rond 3/6, wat het minimum is.
- Makkelijk te detecteren emoties zoals blijheid mogen wel verwachte probabilmiteiten van 1 hebben.

3. Laat stukken van video's waarin geen interessante emoties opgewekt worden minder doorwegen

Het is beter om de focus te leggen op de meest interessante stukken in de video voor de emotiedetectie. Zo kunnen de niet-interessante scènes bijvoorbeeld volledig genegeerd worden zodat enkel de meest bepalende stukken doorwegen.

5.7.2 Aandachtsniveau

De detectie van het aandachtsniveau heeft bewezen ook een grote rol te spelen. De score van iemand die niet continu heeft gekeken, lag veel lager dan bij de andere video's. Er kan dus overwogen worden om het aandachtsniveau meer te laten doorwegen. Momenteel telt het aandachtsniveau voor 40 % mee en wordt deze grotendeels door de attention measurement van Kairos bepaald. Deze meting heeft bij de testen af en toe gefaald maar het werd wel duidelijk dat het aandachtsniveau een grote invloed had op de uiteindelijke waarderingsscore.

De aandachtsmeting van Kairos wordt aangevuld door de hoofdpositie-inschatting en gesloten-ogendetectie van Face++. Indien deze services samen perfect de aandacht zouden registreren, zou het aandachtsniveau één van de beste factoren zijn in het inschatten van het aandachtsniveau. Er moet echter rekening mee gehouden worden dat deze services ook fouten kunnen maken.

5.7.3 Gedragsanalyse

Uit de vragenlijst die gegeven werd aan de testpersonen bleek dat vele mensen anders reageren op videocontent indien deze te eng is: sommigen houden de handen voor de ogen, sommigen kijken weg, sommige bedekken het volledige gezicht. Al deze reacties hebben te maken met de analyse van het gedrag. Aangezien iedereen anders kan reageren, houdt het systeem rekening met alle mogelijke reacties. De gezichtsdetectie van Microsoft heeft bewezen bedekte ogen en mond te registreren maar faalt ook af en toe. Indien bij een persoon een bedekt gezichtsdeel werd waargenomen, was dit vaak slechts voor één of enkele foto's waardoor deze invloed op de score verwaarloosbaar werd. Het kan dus overwogen worden om zelfs één gedetecteerde foto met een reactie meer te laten doorwegen.

Een ander onderdeel van de gedragsanalyse is de tijd de persoon heeft gekeken naar de film. Het systeem bepaalt dit door te kijken hoeveel keer het gezicht van die persoon aanwezig was ten opzichte van de totale lengte van de film. De testen werden afgenomen op 20 testpersonen die het volledige fragment

moesten uitkijken maar ze mochten wegkijken wanneer ze de content onaangenaam vonden. Bij realistisch gebruik zou een gebruiker het fragment vroegtijdig kunnen stopzetten en zou het systeem dit als negatieve feedback kunnen registreren. Indien de kijker de film niet leuk vond en wegkeek, kan het systeem dit ook registreren. Bijna alle testpersonen hebben de volledige fragmenten hun gezicht getoond en de fragmenten aandacht gegeven waardoor de gedragsanalyse op dit vlak verwaarloosbaar werd. Hierdoor zijn de resultaten in de test bijna volledig gebaseerd op de emotiescores. Een meer realistische test zou de testpersonen hun ding laten doen waardoor de totale score meer gebaseerd zou zijn op het gedrag (zoals het vroegtijdig stopzetten van het fragment).

5.7.4 Combinatie van factoren

Voor de emoties angst en afkeer zou het aandachtsniveau de gedragsanalyse de grootste factor kunnen uitmaken bij de inschatting van de waarderingsscore aangezien het emoties zijn waarbij kijkers ook snel gaan reageren door weg te kijken, de ogen te sluiten of het fragment vroegtijdig te stoppen. Extra factoren zoals aandachtsniveau en gedrag kunnen ertoe bijdragen dat deze score meer naar de echte waarderingsscore neigt en er kan overwogen worden deze factoren grotere gewichten te geven. Video's met de emotie blijheid worden het makkelijkste ingeschat en de verschillen in belangrijke scènes kunnen makkelijk gedefinieerd worden aangezien lachen vaak een hoge probabiliteit behaalt. De combinatie van verschillende factoren kan er dus voor zorgen dat de voor- en nadelen van elke factor elkaar aanvullen.

5.7.5 Algemeen besluit en uiteindelijke instellingen

De verschillende factoren afzonderlijk hebben bewezen dat ze kunnen bijdragen aan de inschatting van de waarderingsscore. Hun aandeel in de uiteindelijke score kan worden afgestemd door ze grotere of kleinere gewichten toe te kennen. De uiteindelijke gewichten van elke factor zijn deze zoals gebruikt in de testen en zoals ingesteld in de code. Er kan overwogen worden om deze aan te passen aangezien deze nog niet ideaal werden afgestemd.

Er kan besloten worden dat de grote moeilijkheid zit in het zoeken naar een goede emotiefingerprint voor elke film. Wanneer de verwachte emoties van één film aangepast worden, heeft dit onrechtstreeks invloed op de score in de switchtest bij de andere video's. Een te hoge waarde zorgt ervoor dat het verwachte spectrum voor geen enkele testpersoon haalbaar is. Een te lage waarde zorgt ervoor dat alle testpersonen te snel een hoge similariteit behalen. Een goed evenwicht vinden is dus de boodschap en het huidige beste evenwicht werd behaald met set 4. Bovendien houdt set 4 rekening met meerdere emoties in één filmfragment, iets wat set 1, 2 en 3 nog niet deden. De definitieve versie van set 4 besluit ook dat de totale scoreDistance in de ViewHistoryrecords moet gebaseerd zijn op deze meerdere emoties en daardoor wordt vanaf nu hiervoor de typicalEmotionScore gebruikt. Set 4 is nog steeds niet ideaal en kan verder geoptimaliseerd worden door verder onderzoek met meerdere testpersonen en door rekening te houden met de bepaalde principes en richtlijnen uit [5.7.1](#).

De uiteindelijke en voorlopig meest ideale instelling waarin de applicatie kan gedraaid worden, wordt gerealiseerd met set 4 in combinatie met de TOTAL_SCORE-constante ingesteld op ONLYMOST en de METHOD-constante op SKIPSECONDS. Dit zijn de instellingen waarbij set 4.2 werd uitgevoerd en waarin de applicatie uiteindelijk werd ingesteld. De score op basis van de marges wordt verder niet gebruikt en wordt enkel berekend ter informatie en als extra onderzoek. Het movies.json-bestand bevat per video ook meerdere overwegende emoties en dit zijn de emoties die er echt toe doen voor het fragment.

Hoofdstuk 6

Bedenkingen en mogelijke uitbreidingen

Dit hoofdstuk beschrijft alle bedenkingen bij het gebruik van de app. De bedenkingen bespreken het privacyaspect van de applicatie maar ook de mogelijke uitbreidingen die kunnen bijdragen aan verbeteringen. Tenslotte komen de mogelijke problemen (en oplossingen) aan bod die kunnen ontstaan bij het gebruik van de app.

6.1 Privacyaspect

Zoals ook gebleken uit de vragenlijst bij de testpersonen, is het nemen van foto's en het doorsturen ervan over het internet, een gevoelig onderwerp bij vele mensen. Bovendien werkt het aanmelden op basis van gezichten en kan er bijgevolg worden ingelogd op het profiel van een persoon door een foto van deze persoon voor de camera te houden.

6.1.1 Het vertrouwen van de kijkers

Tijdens het gebruik van de app worden er op twee momenten foto's genomen: bij stap 1 en stap 3. Bij de inlogfase (stap 1) wordt een voorbeeld van wat de camera ziet, getoond op het toestel waardoor de gebruikers zekerheid hebben over de foto die over het internet zal verzonden worden. Tijdens het kijken naar de video bij stap 3 weten de kijkers niet hoe de foto er zal uitzien en kan het gebeuren dat ze vergeten dat de app aan staat en om de twee seconden foto's blijft nemen. Het gebruik van zo een app in het dagelijkse leven zou vele vragen en twijfels doen rijzen bij mensen omwille van de privacy.

De app voorgesteld in deze masterproef werkt met externe services waar de gezichtsdetectie berekend wordt. Slechts 16,67 % van de bevroegde testpersonen zou een app gebruiken die foto's van de kijkers over het internet verstuurt, zoals reeds besproken in Hoofdstuk 5. Ook werd reeds besproken dat 88,89 % de camera van hun smart TV en/of laptop zou bedekken indien ze de app niet gebruiken uit angst om onvoldoende privacy te hebben in de woonkamer of achter de laptop. De camera van de tablet en smartphone bedekken zou respectievelijk 66,67 % en 61,11 % van de testpersonen overwegen. De camera op een smart TV en laptop zijn vaak op iets gericht terwijl ze niet gebruikt worden en bij een tablet en smartphone is dit niet het geval. Ook werd duidelijk dat meer mensen zouden openstaan voor gezichtsdetectie die plaatsvindt op het toestel zelf.

6.1.2 Authenticatie

In de inlogfase kan een persoon ook worden aangemeld door een foto van een gebruiker voor de camera te tonen. De gezichtsherkenning weet niet dat dit niet de echte persoon is en meldt de persoon aan. Bij systemen met belangrijke data zoals betaalgegevens, wachtwoorden en persoonlijk informatie, is het onverantwoord om de inlogfase op deze manier te voorzien aangezien dit onvoldoende beveiligd is. De data die in deze app uit deze masterproef worden opgeslagen, bevatten geen persoonlijk gegevens die beveiligd moeten worden. Daardoor is het inlogproces op basis van het getoonde gezicht voldoende.

6.2 Mogelijke uitbreidingen

Een eerste grote mogelijke uitbreiding zou zorgen dat er frequenter foto's worden genomen om geen enkele gezichtsexpressie te missen. Het kan namelijk gebeuren dat kijkers een emotie slechts gedurende een halve seconde tonen en de applicatie dit mist. Indien er elke seconde een foto zou verzonden worden (met de variabele `INTERVALTIME`), zou de app vastlopen. Dit komt omdat er te veel asynchrone taken gestart worden die niet op tijd afgerond worden aangezien het opsturen van de foto's naar de service een zekere tijd duurt. De applicatie heeft maar een maximum aantal beschikbare asynchrone taken. Wanneer het maximum bereikt is, kan er geen nieuwe gestart worden. Dit zou ertoe leiden dat er "gaten" in de `HistoryDatarecords` gaan zitten aangezien er op sommige tijdstippen geen detectie kon uitgevoerd worden.

Momenteel is de app reeds geoptimaliseerd in uitvoeringstijd om elke twee seconden een asynchrone taak te starten doordat de genomen foto's verkleind worden in aantal pixels. De gemiddelde grootte van een opgeslagen foto ligt rond 60 KB. De afweging moet gemaakt worden tussen een snelle verzending van de foto's en een correcte detectie op de foto's. Deze optimalisatie lukt indien er één gebruiker aanwezig is voor de camera aangezien er dan één foto naar Kairos en Face++ wordt opgestuurd nadat Microsoft er alle gezichten heeft uitgehaald. Indien er meerdere personen op een foto staan, maakt Microsoft meerdere thumbnails per foto en stuurt de applicatie elk afzonderlijk gezicht op naar Face++ en Kairos waardoor er opnieuw een teveel aan asynchrone taken gestart wordt. Er kan ook gekozen worden om enkel bij interessante scènes frequenter foto's te nemen en bij de oninteressante stukken geen foto's te nemen. Dit zou deze problemen kunnen oplossen aangezien er op deze manier zuiniger wordt opgesprongen met de asynchrone taken.

De belangrijkste reden waarom de detectie per foto ongeveer twee seconden duurt, is omdat er eerst steeds gezichtsdetectie én gezichtsherkenning door Microsoft wordt toegepast. Dit is de noodzakelijke eerste stap omdat de verkleinde bitmaps van elk gezicht daarna worden opgestuurd naar de andere services en de resultaten hiervan moeten worden gelinkt aan de correcte gebruiker. Indien de app steeds zou gebruikt worden met slechts één persoon, zou de herkenning kunnen uitgeschakeld worden en alle gedetecteerde feedback aan één hoofdgebruiker toekennen. In de app werd gekozen om telkens alle feedback persoonlijk aan de correcte gebruiker toe te kennen. Dit is de basis van een persoonlijke aanbeveling en in de context van tv kijken er vaak meerdere mensen tegelijkertijd. Indien de gezichtsherkenning uitgeschakeld zou zijn, zou het niet meer om een persoonlijke geschiedenis gaan. Een andere optie om frequenter foto's te nemen, is door de foto's enkel op te slaan en pas achteraf de gezichtsdetectie uit te voeren. Dit kan gerealiseerd worden door de `PHOTO_MODUS` aan te passen naar `SAVEPHOTOS`. In de normale uitvoermodus werd gekozen voor de real time verwerking van de gezichtsdetectie omdat de app in staat moet zijn om meteen na het kijken van een film, een nieuwe aanbeveling moet kunnen geven aan de aanwezige gebruikers.

6.2.1 Extra mogelijkheden

Hieronder wordt een lijst gegeven van andere mogelijke uitbreidingen voor de applicatie. Het zijn ideeën die het gebruiksgemak en de nauwkeurigheid van de app nog meer zouden verhogen. Ook zitten er ideeën bij die moeten worden uitgevoerd indien de app in de praktijk gebruikt zou worden. Momenteel is de app namelijk ontwikkeld voor de testen maar nog niet afgestemd op grootschalig gebruik.

- Er kan per profiel worden bijgehouden hoe extravert of introvert de persoon was met zijn emoties in het verleden. De app kan op die manier de persoon leren kennen en weten bij volgende kijkbeurten dat bijvoorbeeld persoon A pas echt aan het lachen is vanaf een probabiliteit van 0,6 of dat persoon B bij 0,2 probabiliteit op verdriet niet echt droevig is, maar eerder een neutraal kijkt.
- Wanneer een gebruiker een video vroegtijdig stopzet, betekent dit niet steeds dat hij de video niet aangenaam vond. Het systeem kan ook rekening houden met de gepauzeerde video's van een gebruiker. Om een onderscheid te maken tussen een gepauzeerde en een gestopte video kan men werken met twee aparte knoppen. Indien een gebruiker een volgende keer aanmeldt, kan het systeem hem de gepauzeerde video's opnieuw aanraden en deze hervatten vanaf het moment dat er het laatst gekeken was. De persoonlijke hervatting is iets wat diensten zoals Netflix ook reeds implementeren maar in deze app wordt het mogelijk een hervatting te baseren op de gezichten van de aanwezige mensen. Indien bijvoorbeeld zowel persoon A als persoon B samen naar een film kijken en deze samen pauzeren, kan de hervatting enkel worden aangeraden indien bij een volgend bezoek zowel

persoon A als B aanwezig zijn. Indien bij een volgend bezoek ook persoon C aanwezig is, wordt diezelfde film niet aangeraden omdat het systeem weet dat persoon C de film niet heeft meegekeken.

- Een andere extra feature die kan worden toegevoegd, is het vermelden van de reden van aanbeveling. Op die manier kan een gebruiker zien op hoeveel procent de aanbeveling gebaseerd is op zijn demografie, hoeveel procent op zijn geschiedenis en met welke score. De app kan een boodschap geven in de vorm van: “Omdat je graag komedie lijkt te kijken”. De gebruiker zou op deze manier feedback kunnen geven aan het aanbevelingssysteem indien het de bal mislaat, wat maakt dat het systeem hiermee rekening kan houden in de toekomst.
- Andere soorten impliciete feedback zouden ook in rekening kunnen gebracht worden. Bijvoorbeeld wanneer mensen in beeld zijn en terwijl aan het praten zijn, kan ook besloten worden dat die personen geen aandacht geven aan film. Het opnemen van omgevingsgeluiden kan ook interessant zijn om te detecteren of mensen wel aandacht houden of praten terwijl de film speelt.
- Gezichtsdetectie die plaatsvindt op het toestel zelf zodat de app geen foto’s over het internet moet versturen, kan ook een optie zijn die meer mensen vertrouwen. Het nadeel hiervan is dat het toestel de detectie zelf lokaal moet uitvoeren en dat dit een belastende berekening kan zijn.
- Momenteel is de app enkel voorzien op de testen en is er nog geen functionaliteit voorzien om een video te pauzeren of terug te spoelen. Indien een gebruiker dit doet, blijft de app om de twee seconden een foto nemen zonder de tijd aan te passen aan die van het fragment. Toch zou het interessant zijn om hier rekening mee te houden. Wanneer een persoon bijvoorbeeld vaak terug spoelt, kan het systeem afleiden dat de gebruiker de video heel interessant vindt.
- Indien de app gebruikt zou worden in de praktijk, is het onnodig om steeds alle HistoryData-records bij te houden. Eén ViewHistoryrecord per gebruiker per film volstaat om een aanbeveling te genereren. De similariteitscore in het ViewHistoryrecords wordt dan enkel vlak na het kijken naar de film berekend op basis van de HistoryData-records waarna deze records erna verwijderd worden of opgestuurd naar een server die hieruit verwachte emotiespectra kan maken. In de praktijk zal het namelijk niet nodig zijn om lokaal alle resultaten per foto bij te houden en al zeker niet om de foto’s zelf ook op het toestel bij te houden.

6.2.2 Mogelijke problemen

De applicatie werkt met vele componenten en services die moeten samenwerken en vanaf er één service faalt, kan het zijn dat de geschiedenis niet correct wordt verzameld. Dit zal zich uiten in het ontbreken van HistoryData-records of foutief berekende waarderingsscores. Bij de testen vertoonde de app geen problemen bij het gebruik van de services maar dit wilt niet zeggen dat er bij toekomstig gebruik nooit iets kan fout gaan. De app vertrouwt bijna continu op het internet en wanneer er geen internetverbinding beschikbaar is, zal de app een foutmelding geven en de actie niet uitvoeren. Wanneer de internetverbinding aanwezig is maar niet sterk genoeg is, kan dit zich opnieuw uiten in mislukte service-aanvragen en het ontbreken van HistoryData-records. Om deze reden werd ervoor gezorgd dat de testen steeds werden uitgevoerd met een sterke en stabiele internetverbinding. Vooral de initialisatie bij het opstartscherm (uitgelegd in [Bijlage B: Gebruikershandleiding](#)) vereist internet aangezien hier de Face Group van de Face Storage van Microsoft wordt aangemaakt. Indien de Face Group niet correct werd aangemaakt of werd verwijderd door op ‘Delete database’ te klikken, zal Microsoft geen identificatie meer kunnen uitvoeren en zal de app niet werken zoals gewenst. Hierdoor moet de app verwijderd worden van het toestel en opnieuw worden geïnstalleerd zodat de Face Group vanaf nul terug kan worden geïnitieerd.

Aangezien de app van drie services gebruik maakt, is de functionaliteit ook afhankelijk van de gebruikte accounts en keys. De gratis accounts hebben beperkingen, zoals besproken in Hoofdstuk 3, en de gebruikte keys kunnen vervallen (uitgelegd in [Bijlage A: Installatiehandleiding](#)). De app is ook afhankelijk van de prestaties van de drie services. Hiervoor werd in Hoofdstuk 3 reeds een oplossing gezocht door de beste aspecten van elke service te gaan gebruiken maar nog steeds kunnen de services falen bij hun gebruik. Microsoft geeft aan dat de accuraatheid van de identificatie kan beïnvloed worden door verschillende factoren: obstructies die één of meerdere ogen bedekken, harde belichting, veranderingen aan kapsel of gezichtsbehaaring, veranderingen door leeftijd en extreme gezichtsuitdrukkingen kunnen er allemaal voor zorgen dat de identificatie niet de juiste persoon herkent [62]. Deze masterproef lost het probleem van

ouder worden op door gezichten van bestaande gebruikers bij elke aanmelding opnieuw te trainen. Hierdoor houdt de applicatie rekening met veranderende gezichten en wordt de kans op verkeerde identificatie ook geminimaliseerd.

Tijdens de ontwikkeling van de app werd ook gemerkt dat de uitgeknipte facethumbnails (op basis van de kader rond het gezicht) te nauw rond het gezicht is bepaald waardoor Kairos er geen gezicht in kan herkennen. Dit werd opgelost door de FaceListAdapterklasse en de ImageHelperklasse de facethumbnails te laten ‘uitknippen’ met rechts, links, boven en onder een marge van enkele pixels waardoor er meer plaats is rond het gezicht. Afhankelijk van hoe sterk de genomen foto verkleind wordt (momenteel wordt de totale foto tot ongeveer 60 KB verkleind, in te stellen met de SCALEDETECTIONVIDEO-constante in de FaceApplicationklasse), moet het aantal pixels in de marge ook aangepast worden. De marge wordt in de FaceApplicationklasse ingesteld met behulp van de constante MARGINSFACETHUMBNAILS die momenteel op 40 pixels ingesteld staat. Hoe groter het aantal bytes van de foto is, hoe groter het aantal pixels en hoe groter deze marge moet worden gemaakt. De marge mag ook niet te groot worden ingesteld aangezien er dan meerdere gezichten op één uitgeknipte foto kunnen staan.

Het draaien van een app op een Androidtoestel impliceert ook dat men afhankelijk is van de capaciteiten van dat toestel. De ‘OutOfMemoryError’ is een vaak voorkomend probleem bij Android. Deze error komt voor wanneer een foto in het geheugen van de applicatie wordt ingelezen als bitmap maar het geheugen van de applicatie dit niet aan kan en de garbage collector geen nieuwe vrije ruimte kan creëren [41]. De app leest bij het draaien van de testen (leeftijd, geslacht, bedekkingen ...) de foto’s uit het geheugen van het toestel in en zet ze om naar bitmaps in het geheugen (byte allocation). Wanneer er te veel foto’s zijn of de foto’s een te groot geheugenvolume innemen, zal de Java Virtual Machine deze error opgooien. De app lost deze error op door een waarschuwing te geven aan de gebruiker dat de test niet kan worden uitgevoerd en vraagt de gebruiker de test opnieuw te laten draaien met minder of kleinere foto’s.

Een ander probleem ontstaat wanneer de similariteiten in de ViewHistoryrecords berekend worden uit de HistoryDatarecords. De berekening vereist dat alle HistoryDatarecords overlopen worden van een bepaalde gebruiker en elk record wordt vergeleken met bijhorend ExpectedEmotionsrecord. Indien een gebruiker van zes video’s geschiedenis heeft, duurt deze bewerking enkele seconden. Deze tijd zal dus nog groter worden indien gebruikers volledige films bekijken en het aantal records nog groter wordt. Momenteel worden de similariteiten herberekend (enkel van de ingelogde gebruikers) bij elke aanvraag voor een aanbeveling aangezien het systeem zeker wilt zijn dat de ViewHistoryrecords zijn ingevuld met de vereiste similariteitscores. Bij praktisch gebruik van deze app zou de berekening uiteindelijk ofwel tijdens het kijken van de film reeds uitgevoerd worden, ofwel op een server ofwel in een asynchrone taak. Dit werd nog niet geïmplementeerd aangezien de app momenteel enkel voor de testen werd gebruikt en nog niet op grote schaal. De berekening wordt momenteel ook telkens opnieuw uitgevoerd bij elke aanbevelingsaanvraag, ook als deze reeds berekend was. De app is op deze manier ingesteld omdat de similariteitberekening in de ontwikkelfase vaak werd aangepast en het systeem zeker wilt zijn van de berekende waarden en niet met oude waarden wilt werken. Er kan dus besloten worden dat dit een werkpunt is aan de app samen met de opslag van alle HistoryDatarecords en genomen foto’s (zoals hierboven ook reeds besproken in sectie 6.2.1) zodat niet steeds alle HistoryDatarecords herwerkt en bijgehouden moeten worden op het toestel zelf.

Een laatste probleem dat zal blijven bestaan, hoe goed de applicatie en de verwachte emotiespectra ook zijn, is het feit dat kijkers niet steeds duidelijk hun emoties uitdrukken en zorgen dat hun gezicht steeds volledig zichtbaar is. Sommige mensen ondersteunen hun hoofd terwijl ze naar een film aan het kijken zijn, liggen neer waardoor hun gezicht in een andere hoek ligt of kijken in een te weinig belichte kamer naar de video waardoor hun gezicht niet of foutief gedetecteerd wordt. Vele mensen kijken ook naar films zonder enige emotie te tonen terwijl dit niet wilt zeggen dat ze de film slecht vonden, maar hierdoor wel een onterechte lage waarderingsscore krijgen.

Hoofdstuk 7

Conclusie

Het ontworpen systeem is gebaseerd op het inschatten van een waarderingsscore voor elke gebruiker en voor elke video die hij bekeken heeft. De waarderingsscore bestaat uit een inschatting van de gedetecteerde emoties, het aandachtsniveau en het gedrag van de kijkers dat bepaald werd met behulp van gezichtsdetectiesoftware. Met de berekende score kunnen andere gelijkaardige films worden aangeraden door de scores toe te kennen aan tags. Deze tags stellen de rode draad voor tussen gelijkaardige video's en werden in het systeem afgestemd op de 30 voorziene video's. De tags bieden een oplossing voor de te brede definitie van genres en zorgen voor een verfijning die gelijkaardige films kan verbinden. Een groepsaanbeveling houdt rekening met de geschatte voorkeuren van iedereen die aanwezig is tijdens de inlogstap door een aanbeveling te maken met de Average without Misery-strategie. Onaanvaardbare films, op basis van minimumleeftijd (gebaseerd op de meningen op de Common Sense website) en reeds bekeken films, worden geëlimineerd uit de selectie. Voor mensen zonder geschiedenis wordt een oplossing geboden voor het cold start probleem door te kijken naar demografische ratings van IMDb die overeenkomen met hun geschatte geslacht en leeftijd. Het systeem leert de gebruikers echt kennen door hun gezichten bij elke aanmelding extra te trainen, leeftijd en geslacht te updaten en alle geschiedenis per gebruikersprofiel bij te houden. Hierdoor wordt het probleem van een verkeerde inschatting van leeftijd en geslacht en een verkeerde identificatie geminimaliseerd waardoor de aanbevelingen nauwkeuriger en de gebruikersprofielen up-to-date zijn.

Drie services, namelijk Microsoft Cognitive Services Face API, Face++ en Kairos, worden door elkaar gebruikt op basis van de gezichtsdetectiefeatures die ze het beste detecteren. Microsoft wordt steeds gebruikt als basis voor de gezichtsherkenning zodat elke aanwezige persoon automatisch geïdentificeerd wordt. De drie services werden onderworpen aan leeftijds- en geslachtstesten om te kijken welke service hiervoor de beste inschatting maakt. Zowel de Chicago Face Database dataset als een zelf gemaakte dataset van foto's van mensen met hun leeftijd en geslacht werden gebruikt voor de testen. Hieruit bleek dat enkel Microsoft een nuttige inschatting gaf voor leeftijd (vooral bij de inschatting van jonge mensen) en daarom wordt enkel deze service hiervoor gebruikt. Bij de testen voor de inschatting van het geslacht presteerden de drie services goed waardoor ze samen zorgen voor een betrouwbare geslachtsbepaling in de applicatie. Voor de analyse van het gedrag werd Microsoft gekozen voor de bedekkingen van de mond en ogen en Face++ voor het detecteren van gesloten ogen. Indien deze gedragingen gedetecteerd worden bij kijkers, worden deze als negatieve feedback geregistreerd. De gedragsanalyse wordt ook bepaald door het aantal keer het gezicht van de gebruiker aanwezig was tijdens het kijken naar de film. Hoe minder zijn gezicht gedetecteerd werd, hoe lager zijn waarderingsscore. De meting van het aandachtsniveau wordt hoofdzakelijk bepaald door Kairos en wordt aangevuld door de hoofdpositie-inschatting van Face++ aangezien deze de beste resultaten gaf. De gedetecteerde emoties van Microsoft behaalden de beste resultaten bij de emotietesten op de Cohn Kanade dataset foto's en daarom wordt Microsoft ook hiervoor ingeschakeld. De emotietesten toonden ook aan dat de emotie blijheid met de meeste probabiliteit te detecteren is en dat boosheid, afkeer en angst niet met hoge probabiliteiten gedetecteerd worden, zelfs niet bij hele expressieve emoties. Initieel werd er dus aangenomen dat er bij deze drie emoties milder moet worden omgegaan bij de detectie.

Twintig testpersonen werden ingeschakeld, met leeftijden van 5 tot 72 jaar, om zes video's te bekijken terwijl hun gezichten elke twee seconden werden gefotografeerd. Elke video heeft één allersterkste emotie en zo wordt er voor de zes emoties elk één video gekozen. Uit de testen kon bepaald worden of de detec-

tieservices de emoties correct inschatten en hoe de getoonde emoties, het aandachtsniveau en het gedrag kunnen bijdragen tot een waarderingsscore. De testen wezen aan dat wanneer iemand geen aandacht geeft aan een film (en Kairos dit correct registreert), de app deze negatieve feedback correct registreert en de waarderingsscore aanzienlijk doet zakken. De applicatie kan ook de waarderingsscore verlagen wanneer een persoon de ogen sluit of het gezicht bedekt bij te enge stukken. Momenteel gaat de bijdrage van het gedrag op basis van gesloten ogen en bedekkingen bijna volledig verloren aangezien dit steeds slechts over een klein deel op het totale aantal foto's gaat. Er kan dus overwogen worden om deze factoren meer te laten doorwegen. Wanneer het gezicht van een persoon niet gedurende de volledige film aanwezig was, wordt dit wel correct geregistreerd als negatieve feedback. In de rest van de masterproef werd geprobeerd om vooral de waarderingsscore goed te bepalen op basis van de getoonde emoties en om een geschikte verwachte emotiefingerprint voor elk van de zes video's te zoeken. Deze emotiefingerprint is typerend voor de video en legt op elk tijdstip de probabiliteitswaarden van de verwachte emoties vast.

De waarderingsscore van de gebruiker voor de video wordt voor 60 % bepaald door de getoonde emoties van de gebruiker en wordt berekend door de similariteit met de emotiefingerprint van de video te bepalen. De overeenkomst met de emotiefingerprint berekenen en hieruit een waarderingsscore bepalen is de grote uitdaging van de masterproef. Allereerst omdat uit de testen bleek dat de meeste testpersonen niet heel veel emoties tonen bij de fragmenten, ondanks de sterke content erin. Enkel bij video 1 waar de emotie blijheid een hoofdrol speelt, tonen de meeste mensen een expressieve emotie. Ten tweede is het een uitdaging omdat, wanneer een persoon een emotie sterk uit, de detectieservice de emotie niet steeds detecteert met een hoge probabiliteit. Initieel werd gedacht dat enkel boosheid, angst en afkeer de moeilijk te detecteren emoties waren en een probleem vormden. Uit de testen met testpersonen blijkt dat de emoties verrassing en verdriet ook een mildere behandeling vereisen. De emotie blijheid is de enige die probabiliteiten van meer dan 0,95 behaalt bij 90 % van de testpersonen over alle bekeken video's. Verdriet behaalt slechts bij 15 % van de testpersonen probabiliteiten hoger dan 0,60 met een maximum van 0,68. Angst behaalt over alle testpersonen en alle video's slechts een maximale waarde van 0,27. Hier werd als oplossing voorzien om de verwachte emotieprobabiliteiten van de andere vijf emoties (boosheid, angst, afkeer, verrassing en verdriet) ook lage waarden te geven zodat gebruikers een hoge emotiesimilariteit kunnen behalen voor films met deze emoties. De margins-methode - die verder niet gebruikt wordt in de uiteindelijke applicatie - biedt een oplossing door de gemiddelde geuite emotie van de gebruiker te bepalen en grote marges toe te kennen en bijgevolg milder te zijn voor emoties met een lage probabiliteit. Het probleem blijft bestaan dat de meeste geschatte emoties steeds een waarschijnlijkheidswaarde hebben van slechts 0,10 op hun hoogste punt of bij sommige mensen deze emoties zelfs afwezig zijn. Door verschillende soorten verwachte emoties uit te proberen, wordt getracht dit probleem op te lossen.

Met de resultaten van de testen werden vier verschillende emotiefingerprintsets ontworpen die per video de verwachte emotiewaarden definiëren. De kwaliteit van elk van de sets werd getest. Set 4 behaalde de beste resultaten bij de switchtest door drie op zes video's correct te raden op basis van de emotiescore met grote diversiteit tegenover de andere scores. Video 1 - met blijheid als overheersende emotie - behaalde hierbij het grootste verschil in score en dit is te verklaren omdat de emotie blijheid het makkelijkste te uiten en detecteren is. De kwaliteit van set 4 is te danken aan de fijne definiëring van interessante stukken uit het fragment en uitgebalanceerde waarschijnlijkheidswaarden van de verwachte emoties. De balans werd hiervoor onderzocht: indien de verwachte waarden te hoog liggen, is het onmogelijk voor de kijkers om een hoge similariteitscore te behalen en wanneer ze te laag liggen, behaalt iedereen - ook mensen die geen emoties tonen - een te hoge similariteitscore. Set 4 houdt ook rekening met meer dan één overheersende emotie bij één video. Set 4.2 bevat exact dezelfde verwachte emotiewaarden als set 4 met het enige verschil dat set 4.2 enkel met de waarden van de sterkste emoties van de video werkt. Dit resulteert in een set waarmee de geschatte waarderingsscore gebaseerd zal zijn op enkel de belangrijke scènes in de fragmenten en op enkel de belangrijkste emoties. Set 4 gaf voor 100,00 % van de testpersonen de hoogste score aan video 3 terwijl set 4.2 hier verbetering aan brengt door voor 77,78 % video 3 de hoogste score te geven, wat een meer realistisch resultaat is. Set 4.2 heeft ook bewezen terechte scores toe te kennen aan de kijkers voor video 1. Er kan besloten worden dat deze set nog niet ideaal is aangezien nog steeds 83,33 % van de testpersonen dezelfde video op de eerste plaats aangeraden krijgt bij de aanbeveling doordat dezelfde video hoge scores haalde. Uit de set leert men wel dat er drie principes en richtlijnen te definiëren zijn voor het opstellen van de emotiefingerprint:

1. Hou rekening met alle verwachte emoties in de video en negeer emoties die er niet toe doen

2. Hou rekening met de moeilijk te uiten en moeilijk te detecteren emoties
3. Laat stukken van video's waarin geen interessante emoties opgewekt worden, minder doorwegen

Een werkpunt aan de applicatie is de frequentie van de gezichtsdetectie. De app is gemaakt voor real time verwerking van de genomen foto's en het opsturen van de aanvragen naar de services zodat ze meteen na het kijken een nieuwe aanbeveling kunnen aanvragen. Er wordt slechts om de twee seconden een foto genomen terwijl de mensen naar de video kijken. De grootste oorzaken hiervoor zijn de wachttijd op de services voor de gezichtsherkenning en gezichtsdetectie en het gelimiteerd aantal asynchrone taken van de Androidapplicatie. Ook de limiet van de gratis accounts bij de service verhindert een hogere frequentie. Bij de testen werden emoties (vooral blijheid) goed gedetecteerd met deze frequentie maar het kan steeds gebeuren dat een kijker slechts gedurende een halve seconde een emotie toont. Een applicatie waarbij de frequentie hoger kan liggen, zou bijgevolg ervoor kunnen zorgen dat geen enkele emotie en expressie gemist wordt. Een blijvend probleem dat echter zal blijven bestaan, hoe goed de applicatie, de frequentie en de verwachte emotiefingerprints ook zijn, is dat gebruikers niet altijd hun emoties tonen bij het kijken naar films. Het wilt namelijk niet zeggen dat wanneer een kijker geen emoties toont, deze de film slecht vond. Bovendien kan het gebeuren dat gezichten niet worden gedetecteerd (door slechte belichting of wanneer mensen de handen voor het gezicht houden) waardoor de personen een onterechte lage feedbackscore krijgen.

Tenslotte kan er samengevat worden dat de applicatie op een vernieuwende wijze meerdere gezichtsdetectieservices en detecteerbare aspecten samenbrengt om de waarderingsscore van de kijker in te schatten. De factoren aandachtsniveau en gedragsanalyse hebben bewezen correct gedetecteerd te worden en toepasbaar te zijn in het opstellen van deze waarderingsscore. De derde factor, emotiedetectie, speelt de grootste rol in deze score en het is dé uitdaging om de verwachte emoties te bepalen zodanig dat ze deze waardering van de kijker correct kunnen inschatten. Hiervoor moet een goede balans gevonden worden tussen de verwachte probabiliteitswaarden van elke emotie op elk moment van het fragment. Alles in het aanbevelingssysteem hangt namelijk af van de geschatte impliciete waarderingsscore. De ontwikkelde applicatie biedt een oplossing voor het verzamelen van deze impliciete feedback op basis van een set verwachte emoties bij elke video die per tijdstip en per emotie de verwachte probabiliteit definieert. Deze gecreëerde set is echter nog steeds niet ideaal. Bijkomend onderzoek met meerdere testpersonen en video's kan uitgevoerd worden om de kwaliteit van set 4 nog verder te verbeteren, gebaseerd op de vastgestelde principes en richtlijnen uit deze masterproef. Andere factoren zoals aandachtsniveau en gedragsanalyse kunnen bijdragen tot het opstellen van de impliciete feedbackscore zodat een meer precieze aanbeveling gegenereerd kan worden. Op deze manier krijgt men een nauwkeurig en gebruiksvriendelijk systeem dat de wereld weer een stap verder brengt in de toekomst van televisie kijken.

Referentielijst

- [1] “Face++ cognitive services,” <https://www.faceplusplus.com>, 2018, accessed: March 29, 2018.
- [2] B. Ali, Y. Ayaz, M. Jamil, S. Gilani, and N. Muhammad, “Improved method for stereo vision-based human detection for a mobile robot following a target person,” *South African Journal of Industrial Engineering*, vol. 26, pp. 102 – 119, May 2015. [Online]. Available: http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S2224-78902015000100009&nrm=iso
- [3] A. Harvey, “Opencv face detection: Visualized,” <https://vimeo.com/12774628>, June 2011, accessed: March 27, 2018.
- [4] —, “Haarcascade visualizer,” <https://github.com/adamhrv/HaarcascadeVisualizer>, 2012, accessed: March 27, 2018.
- [5] “Microsoft cognitive services,” <https://azure.microsoft.com/en-us/services/cognitive-services/>, 2018, accessed: March 29, 2018.
- [6] “Internet movie database (imdb),” <http://www.imdb.com/>, 2018, accessed: March 29, 2018.
- [7] J. Masthoff, “Group recommender systems: Combining individual models,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 677–702.
- [8] F. Ricci, L. Rokach, and B. Shapira, “Introduction to recommender systems handbook,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 1–35.
- [9] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*, November 2010, pp. 1–24.
- [10] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, “Facing the cold start problem in recommender systems,” in *Expert Systems with Applications: An International Journal*, vol. 41. Tarrytown, NY, USA: Pergamon Press, Inc., March 2014, pp. 2065–2073.
- [11] M. S. Chauhan and M. Sakle, “Study and analysis of different face detection techniques,” *International Journal of Computer Science and Information Technologies*, vol. 5, 2014.
- [12] M.-H. Yang, D. J. Kriegman, and N. Ahuja, “Detecting faces in images: a survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, Jan 2002.
- [13] P. R. Borude, S. Gandhe, P. Dhulekar, and G. Phade, “Identification and tracking of facial features,” in *Procedia Computer Science*, vol. 49. Elsevier, 2015, pp. 2–10.
- [14] C. Zhang and Z. Zhang, “A survey of recent advances in face detection,” Tech. Rep., June 2010. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/a-survey-of-recent-advances-in-face-detection/>
- [15] E. Hjelmås and B. K. Low, “Face detection: A survey,” *Computer Vision and Image Understanding*, vol. 83, pp. 236–274, 2001.
- [16] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, “Face detection in color images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696–706, May 2002.

- [17] G. B. Huang, M. A. Mattar, T. L. Berg, and E. G. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” 2008.
- [18] P. Domingos, “A few useful things to know about machine learning,” *Commun. ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2347736.2347755>
- [19] P. Viola and M. J. Jones, “Robust real-time face detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.
- [20] C. P. Papageorgiou, M. Oren, and T. Poggio, “A general framework for object detection,” in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, Jan 1998, pp. 555–562.
- [21] C. Darwin, *The Expression of the Emotions in Man and Animals*. John Murray, 1872.
- [22] R. E. Jack, O. G. B. Garrod, H. Yu, R. Caldara, and P. G. Schyns, “Facial expressions of emotion are not culturally universal,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 19, pp. 7241–7244, 2012. [Online]. Available: <http://www.pnas.org/content/109/19/7241>
- [23] I. Maglogiannis, D. Vouyioukas, and C. Aggelopoulos, “Face detection and recognition of natural human emotion using markov random fields,” *Personal and Ubiquitous Computing*, vol. 13, no. 1, pp. 95–101, Jan 2009. [Online]. Available: <https://doi.org/10.1007/s00779-007-0165-0>
- [24] M. B. Mariappan, M. Suk, and B. Prabhakaran, “Facefetch: A user emotion driven multimedia content recommendation system based on facial expression recognition,” in *2012 IEEE International Symposium on Multimedia*, Dec 2012, pp. 84–87.
- [25] L. Pauly and D. Sankar, “A novel online product recommendation system based on face recognition and emotion detection,” in *2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICT)*, Dec 2015, pp. 329–334.
- [26] S. Clippingdale, M. Okuda, M. Takahashi, M. Naemura, and M. Fujii, “Level-of-interest estimation for personalized tv program recommendation,” in *2013 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2013, pp. 671–672.
- [27] B. Yan, S. Yu, T. Pei, and Y. Hu, “Vision interaction method based on visual attention mechanism,” in *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, June 2017, pp. 930–935.
- [28] R. Valenti, N. Sebe, and T. Gevers, “Combining head pose and eye location information for gaze estimation,” *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 802–815, Feb 2012.
- [29] A. Strupczewski, B. Czupryński, J. Naruniec, and K. Mucha, “Geometric eye gaze tracking,” January 2016, pp. 444–455.
- [30] D. Johnson and R. Cuijpers, “Predicting gaze direction from head pose yaw and pitch,” January 2013, International Conference on Image Processing, Computer Vision and Pattern Recognition.
- [31] M. García, J. Lucas, V. Batista, and M. Vicente, “Semantic based web mining for recommender systems,” in *Distributed Computing and Artificial Intelligence. Advances in Intelligent and Soft Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 17–25.
- [32] L. H. Son, “Dealing with the new user cold-start problem in recommender systems: A comparative review,” in *Information Systems*, vol. 58, June 2016, pp. 87–104.
- [33] “Movielens dataset,” <https://grouplens.org/datasets/movielens/>, accessed: February 20, 2018.
- [34] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, “A new user similarity model to improve the accuracy of collaborative filtering,” in *Knowledge-Based Systems*, vol. 56, January 2014, pp. 156–166.
- [35] B. Hallinan and T. Striplas, “Recommended for you: The netflix prize and the production of algorithmic culture,” vol. 18, pp. 117–137, January 2016.
- [36] J. Titcomb, “Netflix codes: The secret numbers that unlock thousands of hidden films and tv shows.”

- [37] “Face api - v1.0,” <https://westus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395239>, 2018, accessed: May 17, 2018.
- [38] “Kairos,” <https://www.kairos.com>, 2018, accessed: March 29, 2018.
- [39] “Microsoft face api: Android client library and sample,” <https://github.com/Microsoft/Cognitive-Face-Android>, 2018, accessed: May 16, 2018.
- [40] “Face api documentation,” <https://docs.microsoft.com/en-us/azure/cognitive-services/face/>, 2018, accessed: March 5, 2018.
- [41] “Android developer,” <https://developer.android.com/guide/index.html>, 2018, accessed: March 5, 2018.
- [42] “Faceplusplus sdk,” <https://github.com/FacePlusPlus/java-sdk/tree/master/java-sdk>, 2018, accessed: May 16, 2018.
- [43] “Facerecognitionflowpilots,” <https://github.com/code-mc/FacerecognitionFlowpilots/tree/master/app/src/main/java/net/steamcrafted>, 2018, accessed: May 16, 2018.
- [44] “Chicago face database version 2.0.3,” <http://faculty.chicagobooth.edu/bernd.wittenbrink/cfd/index.html>, July 2016, accessed: March 2, 2018.
- [45] T. Kanade, J. F. Cohn, and Y. Tian, “Comprehensive database for facial expression analysis.” *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG’00), Grenoble, France*, pp. 46–53, 2000.
- [46] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, “The extended cohn-kanade dataset (ck+): A complete expression dataset for action unit and emotion-specified expression.” *Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010), San Francisco, USA*, pp. 94–101, 2010.
- [47] “Android operating system by Google,” <https://www.android.com/>, 2018, accessed: March 17, 2018.
- [48] “Room persistence library,” <https://developer.android.com/topic/libraries/architecture/room.html>, 2018, accessed: March 17, 2018.
- [49] “Youtube android player API,” <https://developers.google.com/youtube/android/player/>, 2018, accessed: March 17, 2018.
- [50] “Common sense media,” <https://www.common sense media.org/about-us/our-mission>, 2018, accessed: March 17, 2018.
- [51] “Samsung België,” <http://www.samsung.com/be/>, 2018, accessed: March 17, 2018.
- [52] T. Segaran, *Programming collective intelligence: building smart web 2.0 applications*. O’Reilly Media, December 2008, pp. 9–28.
- [53] J. Bishop, *C# 3.0 Design Patterns*. O’Reilly Media, Inc, Usa, February 2009.
- [54] “Documentation: Application,” <https://developer.android.com/reference/android/app/Application>, 2018, accessed: May 14, 2018.
- [55] “Accessing data using room daos,” <https://developer.android.com/training/data-storage/room/accessing-data>, 2018, accessed: May 16, 2018.
- [56] “Camerahelper.java,” <http://www.androiddocs.com/samples/BasicMediaDecoder/src/com.example.android.common.media/CameraHelper.html>, 2018, accessed: May 16, 2018.
- [57] “Microsoft emotion api: Android client library and sample,” <https://github.com/Microsoft/Cognitive-Emotion-Android>, 2018, accessed: May 16, 2018.
- [58] “Camera api,” <https://developer.android.com/guide/topics/media/camera>, 2018, accessed: May 16, 2018.

- [59] “Build effective unit tests,” <https://developer.android.com/training/testing/unit-testing/>, 2018, accessed: May 3, 2018.
- [60] “Developing android unit and instrumentation tests - tutorial,” <http://www.vogella.com/tutorials/AndroidTesting/article.html>, 2018, accessed: May 3, 2018.
- [61] “Test app component integrations,” <https://developer.android.com/training/testing/integration-testing/>, 2018, accessed: May 3, 2018.
- [62] “Face api frequently asked questions,” <https://docs.microsoft.com/en-us/azure/cognitive-services/face/faq>, 2018, accessed: June 5, 2018.
- [63] “Configure on-device developer options,” <https://developer.android.com/studio/debug/dev-options>, 2018, accessed: May 3, 2018.
- [64] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., *Recommender Systems Handbook*. Boston, MA: Springer US, 2011.

Bijlage A: Installatiehandleiding

Deze bijlage beschrijft hoe de Android applicatie kan geïnstalleerd worden op een Androidtoestel. De app wordt niet via de officiële Google Play Store gedownload maar is in twee formaten terug te vinden op de bijgeleverde USB-stick. De applicatie kan zowel via een Android Package bestand (APK) rechtstreeks op het toestel geïnstalleerd worden, of er kan voor gekozen worden om de app via Android Studio op het toestel te zetten. Deze laatste optie heeft als voordeel dat de code zichtbaar is en kan aangepast worden. De app kan namelijk in verschillende uitvoermodi worden uitgevoerd en de verschillende constanten (drempelwaarden, keys ...) zijn enkel via de code in te stellen. De installatie via de APK heeft dan weer als voordeel dat Android Studio niet geïnstalleerd moet worden. De code voor de applicatie bevat reeds geldige keys voor alle services. Enkel de key van de Microsoft Face API service heeft beperkte geldigheid en vervalt op 1/7/2018.

Deze handleiding beschrijft alle informatie die nodig is om de installatie te realiseren en eventuele keys opnieuw aan te vragen. De verschillende stappen worden overlopen en de nodige programma's en keys worden vermeld. Eens de app geïnstalleerd is, kan deze gebruikt worden aan de hand van de informatie in [Bijlage B: Gebruikershandleiding](#). De geïnstalleerde app heeft op het toestel een groen icoon met een wit logo, zoals getoond in [Figuur A.1](#), met daaronder de naam van de app ('EyeVideo'). Op dit icoon kan er vanuit het thuisscherm geklikt worden om de app te openen.

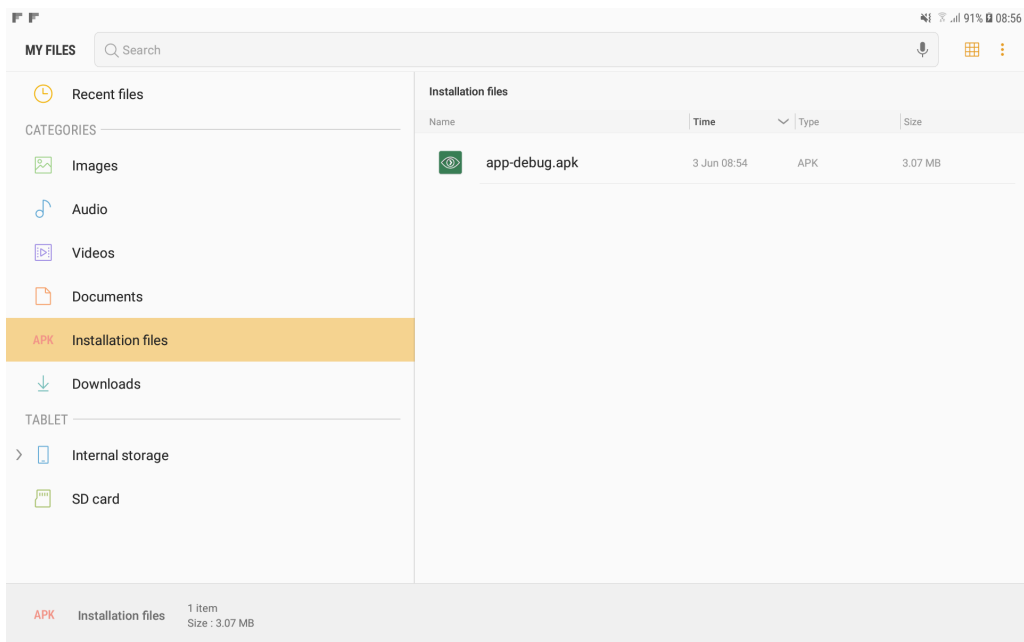


Figuur A.1: Het logo van de app op het toestel

1 Installatie met APK

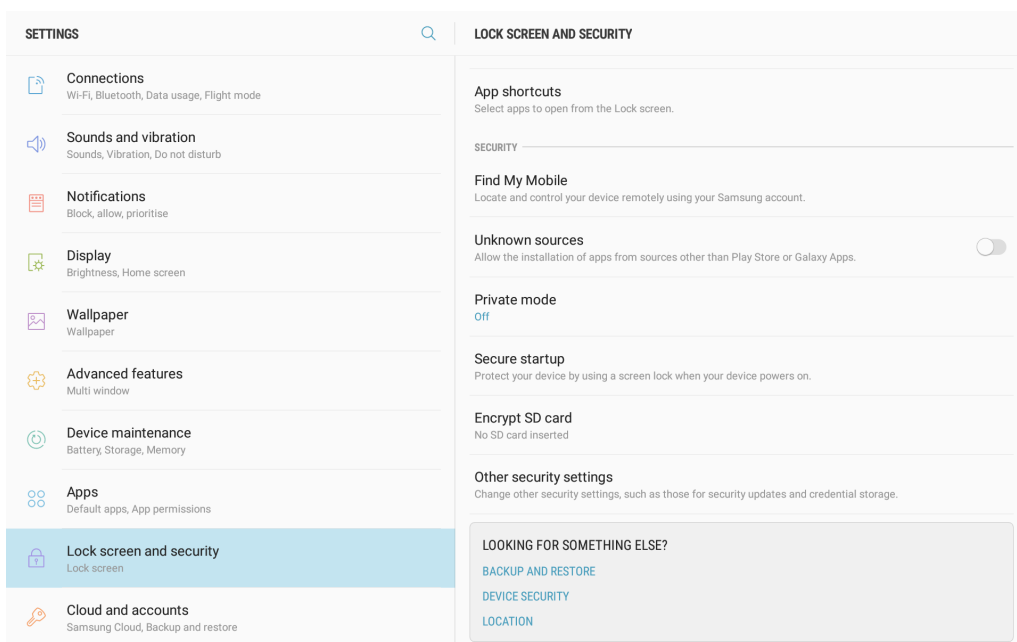
Een APK is een Android Package bestand met .apk als extensie. Dit pakket bevat alles wat nodig is om een Android applicatie te installeren op een Androidtoestel zonder dat de applicatie via de Google Play Store moet gedownload worden. Het .apk-bestand van deze applicatie is te vinden op de bijgeleverde USB-stick in de map *APK*.

Om de app te installeren, moet het APK-bestand op het toestel geplaatst worden. Sluit hiervoor met de USB-kabel het toestel aan op de computer die toegang heeft tot het APK-bestand. Zet het APK-bestand in het geheugen van het toestel. Navigeer op het Androidtoestel naar 'My Files' en klik links, zoals getoond in [Figuur A.2](#), op 'APK Installation files'. Vervolgens verschijnt er rechts een overzicht van alle bestanden die geïnstalleerd kunnen worden op het toestel. Klik op het bestand met het groene logo zoals in [Figuur A.1](#) en met de naam 'app-debug.apk'.



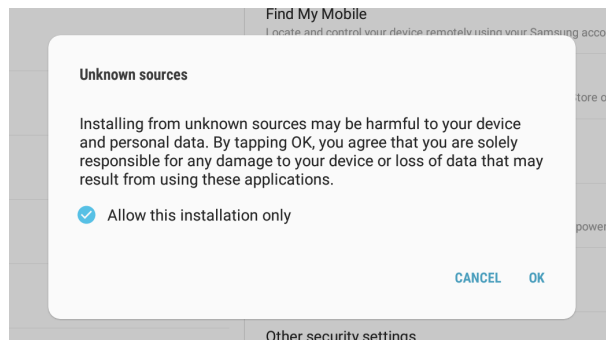
Figuur A.2: Het scherm ‘My Files’ geeft toegang tot de installeerbare bestanden die aanwezig zijn op toestel

Hierna kan er waarschuwing tevoorschijn komen: “For security, your phone is set to block installation of apps obtained from unknown sources.”. Dit betekent dat de standaardinstelling van het Androidtoestel ervoor zorgt dat apps die niet via de officiële Google Play Store worden verkregen, onvertrouwbaar kunnen zijn en bijgevolg moet de beveiliging even gedeactiveerd worden om deze app te kunnen installeren. Klik hiervoor in het waarschuwingenveld op ‘Settings’ zodat het instellingenscherm geopend wordt, zoals te zien in Figuur A.3. Het toestel toont meteen in de instellingenpagina welke knop er moet worden aangezet om tijdelijk de beveiliging uit te schakelen.



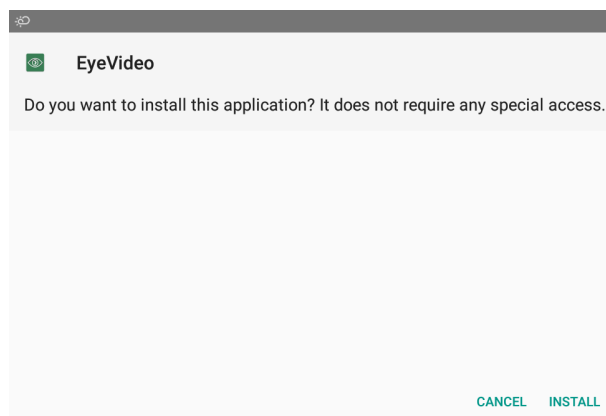
Figuur A.3: Het Androidtoestel begeleidt de gebruiker meteen naar de correcte instellingen waar hij de beveiliging tijdelijk kan uitschakelen

Vink aan de rechterzijde de knop bij ‘Unknown Sources’ aan. Hierdoor verschijnt er weer een nieuwe pop-up, zoals te zien in Figuur A.4. Klik hier opnieuw op ‘OK’ aangezien de app veilig is voor installatie.



Figuur A.4: De gebruiker krijgt nog een laatste waarschuwing alvorens een installeerbaar bestand, dat niet van de officiële Google Play Store afkomstig is, te installeren

Hierna stelt het toestel de vraag of de gebruiker de applicatie wilt installeren zoals te zien in Figuur A.5. Opnieuw mag er bevestigd worden door rechts onderaan op ‘Install’ te klikken (de knop staat helemaal onderaan, soms moeilijk te zien). Vervolgens zal de app worden geïnstalleerd, verschijnt er een laadicoon en wanneer de installatie klaar is, zal het toestel dit laten weten. De app kan vervolgens geopend worden door opnieuw helemaal rechts onderaan op ‘Open’ te klikken.



Figuur A.5: Het scherm dat aan de gebruiker vraagt of de applicatie op het toestel mag geïnstalleerd worden, met rechts onderaan de bevestigingsknop

2 Installatie via Android Studio

Android Studio is te downloaden via <https://developer.android.com/studio/>. Eens Android Studio geïnstalleerd en geopend is, kan via ‘File’ bovenaan het venster op ‘Open...’ geklikt worden. Nu opent er zich een venster waarmee een bestand of project kan geopend worden. Selecteer de map van het meegegeven project met de naam ‘EyeVideo’. Gradle zal nu automatisch het project in Android Studio importeren. Indien het project niet meteen kan geïnstalleerd worden, zal de error verschijnen “Failed to find target with hash string ‘android-X’”. Daaronder staat de optie om de ontbrekende platformen te installeren en het project opnieuw te synchroniseren. Klik op deze optie en wacht tot de installatie beëindigd is. De error “Failed to find Build Tools revision X” kan vervolgens ook verschijnen. Klik dan vervolgens op ‘Install Build Tools X and sync project’. Indien nog gelijkaardige errors verschijnen die met vereiste versies te maken hebben, ga telkens akkoord met het installeren van de vereiste versie.

Zet het Androidtoestel in developermodus (op de gebruikte smartphone en tablet is dit reeds gedaan). Op Android 4.1 en lager is het developerscherm automatisch zichtbaar maar op toestellen met een hogere

platformversie moet dit scherm eerst zichtbaar gemaakt worden. Hiervoor moet er in het ‘Settings’-scherm op ‘System’ geklikt worden bij toestellen met platformversie 8.0 of hoger. Vervolgens moet er op ‘About phone’ of ‘About tablet’ geklikt worden helemaal onderaan het settingsscherm waarna de knop ‘Build number’ zeven keer moet worden ingedrukt. De knop ‘Build number’ kan soms staan onder ‘Software information’ in het ‘About phone’- of ‘About tablet’-scherm. Nadat de ‘Build number’-knop zeven keer is ingedrukt, zal de knop ‘Developer options’ onderaan het settingsscherm zichtbaar zijn. Zet in het developerscherm de ‘Developer options’ aan (bij de ‘On’-knop) en schakel de USB debugging in [63].

Sluit vervolgens het toestel via de USB-kabel aan op de computer en klik vervolgens op de groene ‘Run app’ pijl rechts bovenaan Android Studio. Nu opent er zich een venster om te kiezen op welk toestel of op welk virtueel toestel de app moet draaien. Kies hier het aangesloten Android device dat net in developermodus ingesteld werd.

3 Vereiste keys en vereiste libraries

Om van de drie services en de YouTube API gebruik te kunnen maken, zijn verschillende keys vereist. De keys worden ingesteld in het *config.properties*-bestand in de map *EyeVideo/app/src/main/res*. Voor Microsoft is ook de juiste URL vereist. Bij een free Face API account kan de URL verschillen van die van de standard Face API en de URL wordt gegeven door Microsoft samen met de keys. Face++ levert naast de key ook een secret key af. Kairos verwacht voor verschillende requests verschillende soorten URL’s en heeft zowel een key als een ID per account. Onderstaande tabel geeft een overzicht over waar de keys te verkrijgen zijn. Volg de stappen op de site om de key te verkrijgen. Bij alle services werd de gratis optie gekozen.

Normaal is het niet nodig de keys opnieuw te verkrijgen. De ingediende code bevat namelijk reeds geldige keys voor de drie services en de YouTube API. Bij Microsoft werd gekozen voor de free trial van 30 dagen. Dit is de enige waarbij een kredietkaartnummer moet opgegeven worden en de keys vernieuwd moeten worden om de 30 dagen. Hierdoor heeft de key van Microsoft een beperkte geldigheid tot 1 juli 2018 (de laatste key werd aangevraagd op 1 juni 2018). Na deze datum is het bijgevolg vereist met een nieuw Microsoft account voor de free trial van 30 dagen een nieuwe key aan te vragen indien de app gebruikt wordt. Er kan ook een free Face API key aangevraagd worden (hiervoor is geen kredietkaartnummer nodig) maar deze is slechts 7 dagen geldig en heeft een transactielimiet van slechts 20 aanvragen per minuut. Het wordt dus aangeraden om de app te gebruiken met een standard versie (gratis te verkrijgen via de free trial) aangezien deze versie een transactielimiet van 10 aanvragen per seconde heeft. In de applicatie wordt er om de twee seconden een foto genomen waarop telkens zowel identificatie als emotiedetectie van Microsoft wordt toegepast. Dit zijn twee transacties per foto en zo vinden er dus 60 transacties per minuut plaats vanaf wanneer een gebruiker een minuut naar een filmfragment aan het kijken is. De app kan ook gebruikt worden voor 20 transacties per minuut met een free versie alhoewel er dan wel een “rate limit exceeded”-error zal verschijnen indien een gebruiker te lang naar een film aan het kijken is.

Tabel A.1: Links naar sites voor het verkrijgen van de vereiste keys

Service	Link
Microsoft	https://azure.microsoft.com/en-us/try/cognitive-services/?api=face-api
Face++	https://console.faceplusplus.com/register
Kairos	https://developer.kairos.com/signup
YouTube API	https://developers.google.com/youtube/android/player/register

Ook de gratis versie van Kairos heeft een beperking van 25 transacties per minuut. Bij de normale werking van de app wordt er één foto per twee seconden per gebruiker opgestuurd naar Kairos voor de aandachtsdetectie. Bij elke minuut zijn er dus gemiddeld 5 transacties te kort, maar dit is niet zo een groot probleem aangezien de app dit registreert door het resultaat van het aandachtsniveau te negeren voor die foto’s. In de plaats wordt er enkel naar de hoofdpositie en gesloten ogen, gedetecteerd door Face++, gekeken. Wanneer meerdere mensen aanwezig zijn, worden er meerdere foto’s per twee seconden opgestuurd naar Kairos, één foto per persoon, uitgeknipt door de weergegeven gezichtscoördinaten. Hierbij geeft de limiet een groter probleem maar aangezien de testen steeds met één testpersoon werden uitgevoerd heeft dit weinig invloed op de testen. Een mogelijk oplossing is om een betalend Kairosaccount

te registreren en de keys hiervan in te stellen in het `config.properties` bestand. Een uitgebreid overzicht van de accounts, transactielimieten en prijzen van de drie services werd gegeven in Hoofdstuk 3.

Bij het meegegeven project zitten de twee gebruikte libraries bij de code. Er werd gebruik gemaakt van een library van Face++ zelf die voorziet in enkele methoden om te helpen in het gebruik van de Face++ HTTP API. Het `.jar` bestand van de library is te downloaden op Github via de link <https://github.com/FacePlusPlus/java-sdk/tree/master/java-sdk>. De YouTube Android Player API is nodig in de app om op een simpele manier YouTubevideo's te integreren in de app. Deze is te downloaden via <https://developers.google.com/youtube/android/player/downloads/> en het gebruik ervan vereist een geregistreerde key, zoals hierboven uitgelegd.

4 Bestanden in het project

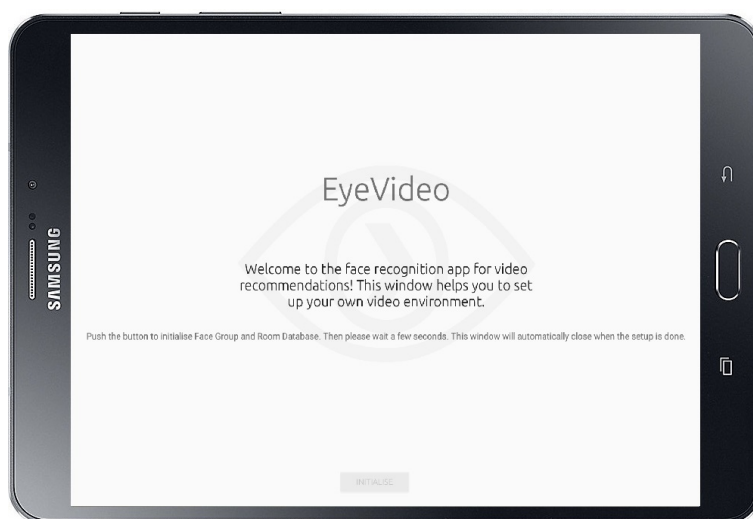
Allereerst is er het `movies.json`-bestand in de map `EyeVideo/app/src/main/assets` dat zorgt voor het inlezen van de standaardvideo's bij het initiële gebruik van de app. Per film bevat dit bestand de titel, het jaartal, de URL van het YouTube-fragment (clip), een string met de ID's van de genres, een string met de ID's van de tags, een string met de verwachte overheersende emoties (de waarde in `emotion`: 0=anger, 1=disgust, 2=fear, 3=happiness, 4=sadness, 5=surprise), de minimumleeftijd, de globale rating, de ratings per demografische groep (F=female, M=male, A=0-17 jaar, B=18-29 jaar, C=30-44 jaar, D=45+), het aantal seconden het fragment duurt en een beschrijving (info). De genres, beschrijving en ratings werden van IMDb gehaald en slaan dus op de volledige film [6]. De tags zijn een uitbreiding van de genres om keywords eigen aan het fragment te beschrijven en gelijkende video's aan elkaar te linken. Deze tags zijn enkel gebaseerd op wat er in het fragment gebeurt en niet op de volledige film. De reden hiervoor is dat de aanbeveling gebaseerd moet zijn op wat de gebruiker in het fragment gezien heeft.

Het inlezen van de tags en genres gebeurt met de klasse `Initialize` die de informatie uit de bestanden `genres.json` en `tags.json` (beide ook terug te vinden in `EyeVideo/app/src/main/assets`) haalt en met hun oplopende ID's in de databank initialiseert. De map `EyeVideo/app/src/main/assets` bevat ook de verwachte emoties van set 4: elke video X heeft een eigen emotiefingerprint, voorgesteld door een CSV-bestand `ExpectedEmotionsVideo_X.csv`. De assetsmap bevat ook het bestand `example.json`. Dit bestand wordt uitgelezen door de applicatie om voorbeelddata in de databank in te laden. Hoe deze data kan ingelezen worden, wordt uitgelegd in [Bijlage B: Gebruikershandleiding](#). Tenslotte bevat de assetsmap ook het bestand `test.csv` waarmee de instrumented testen (beschreven in 4.7 uit Hoofdstuk 4) worden uitgevoerd en waarmee de app kan getest worden op voorbeeldrecords van de `HistoryData` tabel.

De map `EyeVideo/app/src/main/java` bevat de alle Java-klassen van het project, gegroepeerd per package. De bestanden in de map `EyeVideo/app/src/main/res/mipmap-...` bevatten de afbeeldingen met het logo van de app.

Bijlage B: Gebruikershandleiding

De app is eenvoudig in gebruik aangezien de gebruiker nooit zelf iets van informatie moet ingeven. Het enige wat hij moet doen, is de app openen door op het Androidtoestel in het thuisscherm op het groene icoon met het ‘EyeVideo’-logo te klikken waardoor de app opstart. Eens de app geopend is, kan de gebruiker op de knop met het logo duwen midden onderaan het scherm om zichzelf en andere gebruikers aan te melden. Tenslotte kan hij een film kiezen uit de aanbevolen lijst. De kijkers moeten steeds zorgen dat alle gezichten zichtbaar zijn voor de camera en het Androidtoestel niet te dicht op de gezichten staat (anders kan het zijn dat bij het maken van de thumbnails een gezicht er niet volledig op staat). De gebruiker hoeft nooit zijn naam of leeftijd in te geven, het systeem bepaalt dit zelf allemaal voor hem. Een gebruiker die testen wilt uitvoeren en data manipuleren kan dit doen in het instellingscherm. Deze bijlage beschrijft gedetailleerd alle handelingen die via de app op het Androidtoestel kunnen uitgevoerd worden en de acties worden gelinkt aan de code en klassen van het project.



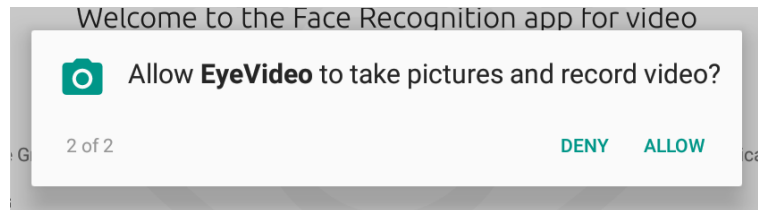
Figuur B.1: Het scherm dat te zien is bij het allereerste gebruik van de app

De app staat standaard in fullscreen-modus en in landschapspositie met de thuisknop naar rechts gericht, zoals getoond in Figuur B.1. Dit werd zo ingesteld omdat films altijd in deze positie worden afgespeeld en zo passen er meerdere gezichten naast elkaar in het beeld van de camera. Het zelfgemaakte logo van de app komt meerdere keren terug in de app zelf en stelt een oog voor van de app die de kijkers “ziet” met een ‘Play’-pijl in voor het starten van de video. Ook in de naam ‘EyeVideo’ zitten deze twee elementen verwerkt. Het startscherm met alle knoppen wordt getoond in Figuur B.3.

1 Opstartscherm

Het opstartscherm dat wordt getoond in Figuur B.1 is een scherm dat enkel bij het eerste gebruik van de app getoond wordt. In de code heet de klasse SetupActivity en de klasse dient voor het instellen van de data in de databank en het aanvragen van de permissies, voorgesteld in Figuur B.2. De app vraagt het gebruik van twee permissies waarvoor de gebruiker toestemming moet geven, namelijk het gebruik van de camera en de toegang tot de opslagplaats op het toestel. Wanneer de gebruiker op de knop ‘Initialise’

duwt, zal de app de basisvideo's, verwachte emoties, genres en tags instellen in de Roomdatabank. De basisvideo's zijn te vinden in het bestand `movies.json` in de map `EyeVideo/app/src/main/assets` van het project. De app maakt ook een Face Group voor de Microsoft Face API en vereist hiervoor toegang tot het internet. Indien er geen internetverbinding is, zal de app een foutmelding geven en de actie pas kunnen uitvoeren bij een internetverbinding.



Figuur B.2: Bij het eerste gebruik moet de gebruiker toestemming aan de app geven om de camera en de opslagruimte te kunnen gebruiken

Bij elk volgend gebruik van de app wordt dit scherm niet meer getoond aangezien er na de initialisatie een variabele wordt ingesteld in de persistente opslagruimte van de app. Deze opslag van applicatiedata wordt verwezenlijkt door de *Shared Preferences* van Android die na de initialisatie bijhoudt dat de databank reeds is ingevuld waardoor de `SetupActivity` niet meer getoond zal worden. Hierdoor kan een gebruiker slechts exact één keer de data in de databank zetten. Wanneer er video's toegevoegd worden aan het `movies.json`-bestand, moet de initialisatie van de databank opnieuw worden uitgevoerd. Dit kan momenteel enkel door de app te verwijderen van het toestel en vervolgens opnieuw te installeren waardoor de databank vanaf nul kan worden ingesteld en er geen foutieve data kan inzitten.

2 Inlogschermb

Het inlogschermb komt overeen met de `LoginActivity`. Dit is steeds de eerste activity van de gehele applicatie (bij het allereerste gebruik opent deze activity de `SetupActivity`). Gebruikers aanmelden kan met de knop met het logo onderaan het scherm. Deze logoknop (ook wel de startknop genoemd) is te zien in de screenshot van de app in Figuur B.4. Elk gezicht van alle kijkers moet voor de camera aanwezig zijn om de foto te maken en iedereen te kunnen aanmelden. Om te kunnen zien hoe de foto er zou uitzien, toont de achtergrond van het inlogschermb een voorbeeld van wat de camera live ziet. Ook de knoppen zijn licht doorschijnend gemaakt zodat de achtergrond voldoende weergeeft wat de camera ziet.



Figuur B.3: Het inlogschermb met de verschillende knoppen om de app te bedienen

Personen kunnen ook in meerdere keren aangemeld worden tijdens één sessie van de applicatie. Door telkens opnieuw de gezichten voor de camera te positioneren en op de logoknop te drukken, worden alle

aanwezige mensen gedetecteerd en aangemeld. Gebruikers die nog onbekend zijn voor de app worden automatisch aangemaakt in de databank en vervolgens ook aangemeld. Ingelogde personen worden bijgehouden door de app zolang de app niet afgesloten wordt. Wanneer de app nog op de achtergrond draait (door bijvoorbeeld op de thuisknop te duwen), is deze niet volledig afgesloten en zullen de gebruikers nog steeds ingelogd blijven. De gebruikers kunnen ook expliciet worden uitgelogd, wat verder wordt uitgelegd in sectie 4.



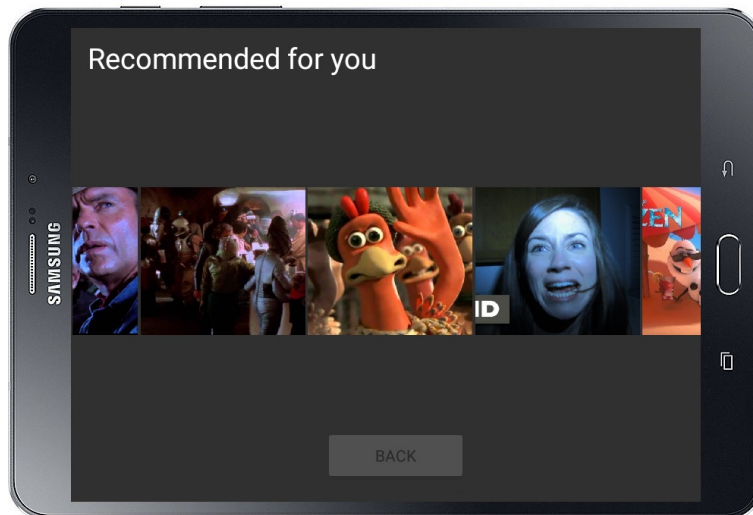
Figuur B.4: Het stuk van het inlogscherm dat een laadicoon toont wanneer de gebruiker de knop van het logo heeft ingedrukt

Een nieuwe gebruiker wordt aangeraden meerdere keren (vijf tot zes keer) op de logoknop te duwen zodat zijn gezicht meerdere malen kan opgestuurd worden voor de training van de gezichtsherkenning. Hierdoor wordt de kans vergroot dat de gezichtsherkenning van Microsoft de gebruiker correct kan identificeren bij het kijken naar de film of een toekomstige aanmelding. Rechts bovenaan staat een knop met een vraagteken. Deze knop geeft informatie over wat de gebruiker moet doen op de inlogpagina.

3 Scherm waarin de gebruiker een video kan kiezen

Indien nog niemand is aangemeld, is enkel de knop 'Show all videos' zichtbaar om naar het filmoverzicht te gaan. Deze knop dient om alle beschikbare films op te vragen die in de app aanwezig zijn (gesorteerd op hun oplopende ID's) maar geeft geen aanbeveling voor de aanwezige personen. Deze functie is vooral handig om te kijken naar alle films of indien een film niet aanwezig is in een aanbeveling en men deze toch wilt bekijken. Als er mensen ingelogd zijn en de globale LoggedIn-instantie de ID's van deze gebruikers bevat, kan er een aanbeveling gegenereerd worden op basis van hun persoonlijke voorkeuren. In dat geval wordt de knop 'Show Recommendation' zichtbaar die leidt naar het filmoverzicht met aanbevolen films, getoond in Figuur B.5.

Voor zowel het tonen van alle films als voor de aanbevolen films wordt een VideoActivity gestart bovenop de LoginActivity. Het overzicht toont de filmthumbnails in een lijst die verticaal kan gescrold worden. Indien er op een thumbnail geklikt wordt, wordt de gebruiker naar het volgende scherm geleid waarin hij alle informatie over de film ziet, zoals de titel, het jaartal, de genres, de globale rating van IMDb en de beschrijving. Deze informatie wordt getoond in een InformationActivity en is afkomstig van IMDb [6].



Figuur B.5: Het filmoverzicht in de applicatie met de aanbevolen films voor de ingelogde gebruikers

Het informatiescherm wordt gedemonstreerd in Figuur B.6. Onderaan staan drie knoppen. De 'Play'-knop is de laatste van de drie en dient voor het starten van het filmfragment. Hierbij wordt er een nieuw scherm geopend dat het filmfragment afspeelt in een nieuwe Activity bovenop de InformationActivity. Bij het beëindigen van het filmfragment klikt de gebruiker op de terugknop van het Androidtoestel en komt hij weer uit bij de InformationActivity.

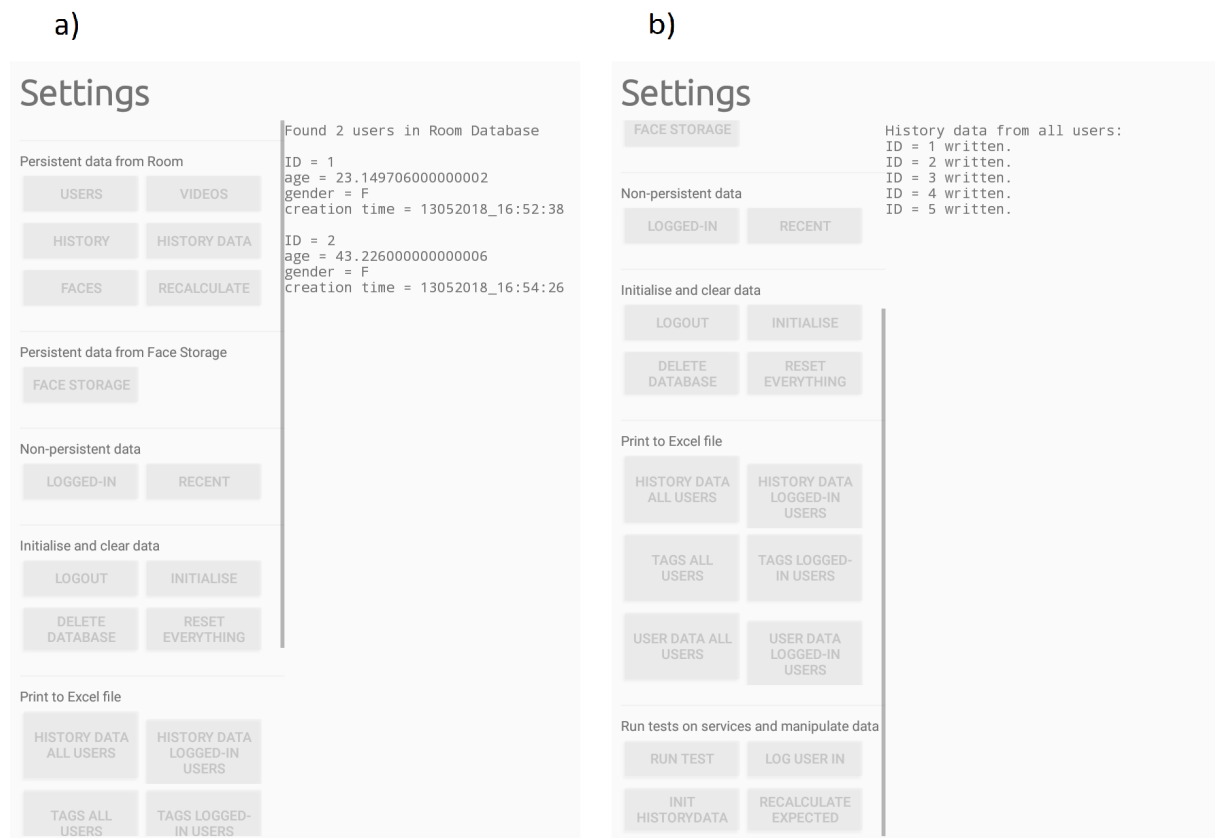


Figuur B.6: Een informatie-overzicht van de geselecteerde film

De eerste knop, de 'Back'-knop, dient voor het terugkeren naar de initiële aanbeveling in het filmoverzicht van de vorige pagina en doet hetzelfde als de terugknop op het Androidtoestel zelf, namelijk terugkeren naar de vorige Activity in de activity stack. De middelste knop dient voor het herberekenen van de aanbeveling. Indien een gebruiker net een film gekeken heeft en op de terugknop van het Androidtoestel heeft gedrukt, komt hij automatisch terug op het informatiescherm aangezien dat de Activity vóór de video was. Als hij nu een nieuwe aanbeveling wilt krijgen, kan hij dit doen door op deze 'Recalculate'-knop te duwen. Hierdoor zal ook rekening gehouden worden met de impliciete feedback van de gebruiker bij de net bekeken film en zal deze film onder andere niet meer in de nieuw berekende aanbevolen lijst voorkomen. De knop zorgt ervoor dat de InformationActivity en de vorige VideoActivity met de initiële aanbeveling worden verwijderd van de activity stack en een nieuwe VideoActivity gestart wordt met de nieuwe aanbeveling.

4 Instellingenschermb

Een gebruiker die wat meer wilt doen, zoals de data in de databank in de app bekijken, kan dit doen door op het tandwielicoon te klikken links bovenaan in het inlogschermb. De gebruiker komt dan terecht op het instellingenschermb, te zien in de screenshots in Figuur B.7. Het scherm komt overeen met de MainActivity in de code. De scroll-lijst van knoppen staat links op het scherm gepositioneerd en de knoppen worden ingedeeld in groep volgens hun functionaliteit. De screenshot in a) in Figuur B.7 toont het eerste deel van de knoppenlijst, volledig naar boven gescrold, zodat de eerste drie knoppengroepen zichtbaar zijn die instaan voor het tonen van informatie in het rechter tekstveld. De screenshot in b) toont alle knoppen die instaan voor het manipuleren van data in de app wanneer de lijst volledig naar onder is gescrold.



Figuur B.7: Het instellingenschermb met links de knoppen om verschillende acties uit te voeren: a) volledig omhoog gescrold en b) volledig omlaag gescrold

4.1 Persistente data bekijken

De eerste groep dient voor het bekijken van de informatie rechtstreeks uit de Roomdatabank op het toestel. De informatie, zoals gebruikers ('Users'), video's ('Videos'), geschiedenis ('History data') en face-ID's ('Faces'), wordt dan in het rechter tekstveld getoond. De knop 'Recalculate' dient om expliciet nog eens alle similariteiten van alle kijkbeurten te herberekenen. Deze knop is handig bij het manueel herinvoeren van de HistoryDatarecords (hieronder verder uitgelegd in sectie 4.6) en herberekenen van de geschatte scores voor de films van de gebruikers. Het herberekenen van alle similariteiten overloopt alle HistoryDatarecords en kan dus vrij lang duren (ongeveer twee tot vijf seconden per gebruiker, afhankelijk van hoeveel video's de gebruiker heeft bekeken).

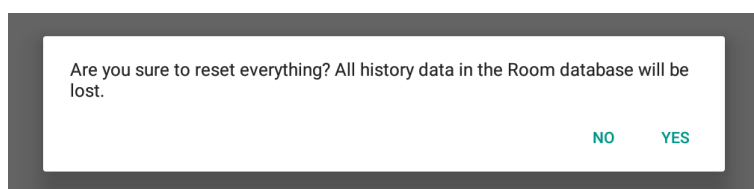
Onder de Roomdatabankgroep staat een knop om de informatie uit de Face Storage van Microsoft te halen, namelijk alle gebruikers die hier geregistreerd staan bij Microsoft met hun toegekende naam en ID.

4.2 Niet-persistente data bekijken

De derde groep knoppen bevat alle informatie die niet persistent is en dus bij het sluiten van de app verloren zal gaan. Als er op deze knoppen gedruwd wordt, wordt de informatie die de LoggedInklasse bijhoudt getoond, namelijk de huidige ingelogde gebruikers en recent aangemaakte ViewHistoryrecords.

4.3 Data in de databank zetten en data uit de app verwijderen

Met de vierde knoppengroep kan een gebruiker iedereen manueel uitloggen door op ‘Logout’ te klikken. De knop ‘Initialise’ dient voor het invullen van de databank met fictieve voorbeelddata van één gebruiker die naar zes video’s gekeken heeft. De knop zorgt ervoor dat er reeds zes ViewHistoryrecords in de databank ingevuld worden op basis van de info uit het bestand *EyeVideo/app/src/main/assets/example.json*. In sectie 6 wordt op deze voorbeelddata verder ingegaan. De ‘Reset Everything’-knop dient voor het leegmaken van alle geschiedenisrecords. De gebruikers en video’s blijven bestaan maar al hun geschiedenis wordt verwijderd. Aangezien de ‘Reset Everything’- en ‘Initialise’-knop riskante operaties uitvoeren, wordt er een dialoogvenster getoond. Dit dialoogvenster vraagt aan de gebruiker of hij zeker is dat hij de actie wilt uitvoeren en ziet er voor de ‘Reset Everything’ uit zoals Figuur B.8.



Figuur B.8: Het instellingenscherf met links de knoppen om acties uit te voeren op de databank

De knop ‘Delete Database’ geeft de instructies om de databank leeg te maken. Aangezien het ledigen van de databank ook alle gebruikers en video’s verwijdert, impliceert dit dat de gebruikers en gezichten ook uit de Face Storage van Microsoft verwijderd moeten worden. De gebruiker wilt namelijk dat wanneer hij de databank verwijdert, ook bijhorende gebruikers en hun gezichten volledig verwijderd worden. Dit is een grote beslissing die niet zomaar mag uitgevoerd worden en daarom verschijnt er opnieuw een dialogbox die vraagt of de gebruiker zeker is van zijn beslissing. Het drukken op de knop zorgt voor het verwijderen van de Face Group en alle personen en gezichten erin. De gebruiker kan nu de volledige app verwijderen zodat de databank ook volledig verwijderd wordt. De app kan opnieuw geïnstalleerd worden met een nieuwe basisdatabank en een lege Face Storage, indien de gebruiker de app terug vanaf het begin wilt gebruiken.

4.4 Uitprinten van data uit de databank

De ‘Print to Excel file’-titel kondigt de volgende knoppengroep aan. Deze groep staat in voor alles wat te maken heeft met het printen naar CSV-bestanden. De CSV-bestanden zullen opgeslagen worden op het toestel onder de map *Pictures* en in de map van de app *FaceRecognitionProject* en kunnen vervolgens in Excel geopend worden om de output in rijen en kolommen te bekijken. Er zijn drie soorten bestanden: geschiedenis per gebruiker (zowel ViewHistory- als HistoryDatarecords), de berekende scores voor de tags per gebruiker en de inhoud van de Userstabel met alle informatie van de gebruikers. Elk bestand krijgt een unieke naam met als identificatie het ID van de gebruiker over wie de informatie gaat. Wanneer de informatie is weggeschreven, verschijnt er in het rechter tekstveld welke ID’s er allemaal zijn weggeschreven. Er kan gekozen worden om de informatie van alle gebruikers weg te schrijven of enkel van degenen die ingelogd zijn. Dit is handig wanneer de app reeds veel gebruikers bevat en enkel de informatie van de ingelogde gebruikers relevant is.

4.5 Evaluatietesten services

De laatste groep bevat knoppen om evaluatietesten om de drie services te laten draaien op het toestel in de app. De testen die kunnen worden uitgevoerd, omvatten de verschillende soorten gezichtsdetectie van

de drie services, zoals het testen van de emoties, leeftijd, geslacht, bedekkingen in het gezicht, gesloten ogen en de hoofdpositie in afbeeldingen. Deze testen werden ook zo uitgevoerd voor de evaluatie van de drie services in Hoofdstuk 3. De methoden hiervoor zitten in de klasse `DetectionForTests`. Alle testen plaatsen de output in een CSV-bestand per service in *Pictures/FaceRecognitionProject* met in elke rij de naam van de foto gevolgd door de resultaten. Wanneer de app iets heeft weggeschreven naar het toestel, kan dit meteen gezien worden als de USB-aansluiting er een opnieuw wordt uitgehaald en ingestoken. Hieronder staat beschreven waar de app de foto's uitleest op het toestel. Hiervoor moet de map *Pictures* van het toestel gevonden worden en hier moeten eventueel enkele nieuwe mappen worden in geplaatst met de foto's waarop de detectie zal worden toegepast.

Het instellen van welke test moet worden uitgevoerd, kan in de code bepaald worden door in de klasse `FaceApplication` de constante `TEST_MODUS` te veranderen naar:

- **AGEANDGENDER:** zorgt voor de inschatting van leeftijd en geslacht door de drie services. Plaats hiervoor foto's rechtstreeks in de map *Pictures/FaceRecognitionProject* op het toestel (maak eventueel zelf de map aan).
- **EMOTIONS LAST:** voert de emotiedetectie uit op de Cohn Kanade dataset [45, 46] voor de drie services en neemt per map steeds de laatste foto die de meeste expressie bevat. De dataset is terug te vinden op de USB-stick in de map *Datasets* of via <http://www.consortium.ri.cmu.edu/ckagree/>. Plaats voor de emotiedetectie de inhoud van de map met de txt-bestanden *Emotion-labels*, namelijk alle mappen S005-S999, rechtstreeks in een map *Pictures/Emotion*. De foto's zelf, in hun mappen S005 tot en met S999 moeten in de map *Pictures/EmotionTest* geplaatst worden. Druk vervolgens op 'Run Test' in de app en bekijk de CSV-bestanden wanneer de services alle foto's hebben behandeld (de duur is afhankelijk van het aantal foto's). De gratis sleutel van Kairos en Microsoft hebben een limiet van 25 en 20 requests per seconde. Het betalende account van Microsoft en de gratis sleutel van Face++ heeft geen limiet hierop en voert in één keer de detectie op alle foto's uit. Voor de beperking van Kairos werd een oplossing voorzien door de reeds gedetecteerde foto's op te slaan in het bestand *already.csv*. De 'Run Test'-functionaliteit moet dan meerdere keren worden uitgevoerd en de applicatie zal automatisch enkel de foto's nemen die hij nog niet heeft behandeld.
- **EMOTIONS MIDDLE:** voert de emotiedetectie uit op de Cohn Kanade dataset [45, 46] voor de drie services en neemt per emotiemap steeds de middelste foto. Hiervoor moeten de foto's op dezelfde manier als de vorige modus op het toestel geplaatst worden.
- **EMOTIONS SEQUENCE:** voert de emotiedetectie uit op de hele reeks foto's uit een volledige emotie-map van de Cohn Kanade dataset [45, 46]. Hiervoor moeten de gewenste fotomappen met de reeks foto's (bijvoorbeeld de map *S005/001* en niet *S005*) in *Pictures/EmotionTest* geplaatst worden. Met deze test wordt de evolutie van de expressies van een persoon zichtbaar die overgaat van geen emotie naar een hele uitgesproken emotie.
- **EXTRA:** voert gezichtsdetectie uit om bedekkingen van het gezicht te detecteren, de probabiliteiten dat de ogen gesloten zijn, het aandachtsniveau en de hoofdpositie te detecteren. Hiervoor moeten de foto's rechtstreeks in *Pictures/FaceRecognitionProject* op het toestel geplaatst worden.

De output bij de drie emotiedetectietesten in de kolommen van het CSV-bestand is van de vorm: naam foto - anger - fear - happy - surprise - sad - disgust. Hier moet rekening mee gehouden worden bij het bekijken van de resultaten aangezien de rest van de applicatie werkt met de alfabetische volgorde (anger, disgust, fear, happiness, sadness, surprise).

4.6 Herinlezen van geschiedenisrecords

Soms is het nodig om opnieuw alle `HistoryData` records van een gebruiker opnieuw in te laden in de databank. Hiervoor kan de knop 'Init HistoryData' gebruikt worden. Deze knop zorgt ervoor dat een bestand met de naam *InitHistoryData_userX.csv* uit de map *Pictures/FaceRecognitionProject* ingelezen wordt, met X het ID van de gebruiker. Na enkele seconden (afhankelijk van de hoeveelheid geschiedenis van de gebruiker) geeft het rechter tekstveld de melding dat het bestand is ingelezen. Alle gegevens van de gebruiker(s) zullen nu toegevoegd zijn aan de databank. Indien men opnieuw de lijst van aanbevolen films voor gebruiker met ID 1 wilt herberekenen met de nieuwe data, kan dit enkel als die gebruiker

is aangemeld. Dit kan gemanipuleerd worden door op de knop ‘Log user in’ te duwen in plaats van de gebruiker in te loggen via de camera in het inlogscherf. Voor elke gebruiker uit de testen zijn er reeds *InitHistoryData_user1.csv*-bestanden voorzien. De bestanden zijn terug te vinden in de map *Testing/X/data_X/recalculated*, met X het volgnummer van de testpersoon, en kunnen gekopieerd worden naar het Androidtoestel om daar opnieuw ingelezen te worden. Elk van de bestanden is ingesteld op het inlezen als gebruiker met ID 1 omdat de testpersonen bij het herinstalleren van de app hun vorige ID verloren en standaard weer op 1 worden gezet. Hierdoor wordt ook enkel de gebruiker met ID 1 ingelogd wanneer er op de ‘Log user in’-knop wordt gedrukt. Het is dus aangeraden de bestanden gebruiker per gebruiker in te lezen in de app als gebruiker 1 en dan de resultaten van die specifieke gebruiker te bekijken.

4.7 De gemiddelde waarden van verschillende emotiespectra laten bepalen

De knop ‘Recalculate expected’ heeft niet veel te maken met de werking van de app maar zorgt ervoor dat een bepaald aantal personen (in te stellen met de variabele *amount* in de *MainActivity*klasse) met hun aanwezige *HistoryData*records per film worden uitgemiddeld tot een spectrum van verwachte emoties, zoals gebruikt werd voor het maken van set 2. Zorg hiervoor dat hun *HistoryData*records zijn ingevuld (bijvoorbeeld door de ‘Init Historydata’-knop te gebruiken, zoals hierboven uitgelegd). Voor het maken van set 2 werden initialisatiebestanden gebruikt uit de map *ExpectedEmotions/DifferentSets/set2_mergedAverage/videoX* voor elk van de zes video’s. Bij deze data staat telkens een bestand *usermapping_videoX.txt* waarin het ID van de gekozen testpersoon vermeld staat die gebruikt werd bij die video. De *HistoryData* van de gebruikers hebben records om de twee seconden. De records worden overlopen en bijgehouden voor alle gebruikers waarna hun gemiddelde waarde per tijdstip en per emotie wordt bepaald. De methode voorziet dat het uitgemiddelde resultaat een emotiespectrum wordt om de één seconde, zoals een emotiefingerprint van een film verwacht. Het resultaat zal in een CSV-bestand met de naam *MergedExpectedEmotionsVideo_X*, met X het ID van de video, worden uitgeschreven. De bestanden kunnen dan door het herinstalleren van de app worden ingelezen in de *ExpectedEmotion*stabel. Hiervoor moeten de bestanden, eventueel manueel hier en daar nog wat worden aangepast zoals in Hoofdstuk 5 en in de map *EyeVideo/app/src/main/assets* geplaatst worden met als naam *ExpectedEmotionsVideo_X.csv*.

5 De verschillende modi om de app te laten draaien

De app kan op meerdere manieren functioneren (met verschillende mogelijke combinaties van instellingen) en naar wens van de gebruiker ingesteld worden. Om geen verschillende versies van de app te maken en de verschillende soorten onnodig ingewikkeld te maken, werd er gekozen om de gebruiker zelf de uitvoermodus te laten instellen met behulp van constanten in de *FaceApplication*klasse. Listing B.1 geeft een overzicht van de standaardinstelling in de *FaceApplication*klasse. Hierboven werd reeds de *TEST_MODUS*-constante besproken en hieronder worden de mogelijke andere instellingen besproken.

Er zijn twee verschillende hoofdmodi waarin de app kan worden uitgevoerd die in te stellen zijn in de *FaceApplication*klasse met de constante *MODE*. Ten eerste is er de *APP*-modus, die zorgt voor de werking zoals de uiteindelijk app zou moeten werken. Ten tweede is er de *TESTING*-modus die gebruikt werd bij het testen met de testpersonen. Deze modus zorgt er onder andere voor dat er reeds na zes kijkbeurten van die persoon een aanbeveling kan worden gemaakt die 100 % gebaseerd is op de geschiedenis en 0 % op de demografische gegevens.

In zijn normale modus werkt de app zodanig dat elke foto om de twee seconden meteen wordt opgestuurd voor detectie. Er is ook de mogelijkheid om elke x seconden een foto te nemen en pas later op te sturen voor detectie. Dit kan ingesteld worden in de klasse *FaceApplication* door de *PHOTO_MODUS*-constante aan te passen. Een gebruiker kan ook zijn eigen emoties opnemen tijdens het kijken naar een filmfragment om later te gebruiken als spectrum van verwachte emoties. Hieronder een overzicht van de verschillende opties van de *PHOTO_MODUS*-constante:

- **NORMAL**: de normale werking van de app. Om de twee seconden wordt een foto genomen en deze wordt opgestuurd naar de drie services voor het verzamelen van impliciete feedback.
- **SAVEPHOTOS**: zorgt ervoor dat er om de x seconden (momenteel ingesteld op 1 seconde) een foto genomen wordt en stuurt deze op naar de *PhotoService* die zorgt voor de opslag op het toestel.

- **INITOWNEXPECTEDEMOTIONS:** in deze modus kan een gebruiker zijn eigen emoties opnemen tijdens het kijken naar een filmfragment. Wanneer het filmfragment wordt gestart, neemt de applicatie elke seconde een foto, detecteert de emoties van de gebruiker en schrijft deze per seconden uit naar het bestand *ExpectedEmotionsVideo_x* met x het ID van de gekozen video. In deze modus vindt er geen gezichtsherkenning plaats omdat er verwacht wordt dat er maar één gebruiker aanwezig is en de getoonde emoties niet gekoppeld zullen worden aan een gebruiker. Hierdoor kan het detecteren van emoties veel sneller gebeuren dan tijdens het verzamelen van impliciete feedback en wordt de timerwaarde op één seconde ingesteld.

De berekening die de HistoryProcessor uitvoert kan ook veranderd worden van instelling. Zo kan bepaald worden hoe de gewichten worden ingesteld voor de getoonde emoties tijdens de similariteitberekening en welke waarde de totale scoreDistance, gebruikt bij de hele werking van de app, toegekend krijgt. Hiervoor moet de constante TOTAL_SCORE veranderd worden als volgt:

- **EQUAL:** zorgt ervoor dat alle zes de emoties evenveel gewicht toegekend krijgen, namelijk elk het gewicht één.
- **ONLYMOST:** zorgt ervoor dat enkel de typerende emotie(s) (ingesteld in het json.movies-bestand) het volle gewicht krijgen. Bij één typerende emotie krijgt deze emotie het gewicht zes, en bij twee emoties krijgen ze elk het gewicht drie.
- **MOREWEIGHTTOMOST:** deze instelling zorgt ervoor dat de typerende emotie(s) standaard het gewicht twee krijgen. De andere emoties krijgen dan elk hun deel van het overblijvende gewicht zodat een totaal van zes verkregen wordt voor de zes emoties. Indien er bijvoorbeeld één typerende emoties is, krijgt deze het gewicht twee en de andere vijf krijgen het gewicht 0,8.

De applicatie kan ingesteld worden zodat het verwachte emoties die gelijk zijn aan nul overslaat bij de berekening van de score. Dit wordt ingesteld door de constante METHOD in FaceApplication op SKIP-SECONDS in te stellen en zorgt ervoor dat scènes met weinig emoties worden overgeslagen tijdens de similariteitberekening. Hierdoor hebben minder belangrijke scènes minder invloed op de geschatte score en worden belangrijke scènes benadrukt om zo een meer nauwkeurige geschatte scores te verkrijgen. Deze instelling werd gebruikt bij set 3 en set 4. De andere instelling is ALLSECONDS en zorgt ervoor dat de getoonde emotie van elke seconde wordt meegeteld en werd gebruikt bij set 1 en set 2.

Tenslotte kan het ook worden ingesteld dat de initiële leeftijd van een nieuwe gebruiker ook door Face++ en Kairos wordt bepaald. Alhoewel het in de uiteindelijk applicatie niet gebruikt wordt, bestaat deze mogelijkheid om de gemiddelde leeftijd van de drie services samen te bepalen. Dit wordt ingesteld door de AGE_MODUS-constante op THREESERVICES te zetten. Standaard staat de constante op ONLYMICROSOFT waardoor enkel de waarde van Microsoft genomen wordt en er geen gemiddelde berekend wordt voor de leeftijd. Voor het geslacht worden standaard wel de drie services gebruikt door middel van voting. Hiervoor bestaat er geen constante aangezien dit vast in de code staat.

6 Standaardinstellingen

Listing B.1 geeft een overzicht van de constanten en hun standaardinstelling in de applicatie. In het APK-installatie-bestand staan deze instellingen vast zoals ze hier zijn ingesteld. Een aanpassing hieraan en de functionaliteit van de app kan volledig veranderd worden via Android Studio (de 'Run'-knop rechts bovenaan zorgt automatisch voor de compilatie en builden van de aanpassingen, waarna de aanpassingen op het toestel kunnen gezet worden). Momenteel zijn ook de verwachte emoties van set 4 ingesteld voor video 1 tot en met 6 (in de map *EyeVideo/app/src/main/assets*).

```
public static final Modus MODE = Modus.TESTING; 1
public static final PhotoModus PHOTO_MODUS = PhotoModus.NORMAL; 2
public static final TestModus TEST_MODUS = TestModus.AGEANDGENDER; 3
public static final Method METHOD = Method.SKIPSECONDS; 4
public static final WeightsEmotions TOTAL_SCORE = WeightsEmotions.ONLYMOST; 5
public static final AgeMethod AGE_MODUS = AgeMethod.ONLYMICROSOFT; 6
```

Listing B.1: Standaardinstellingen van de constanten in de FaceApplicationklasse

Zoals reeds hierboven vermeld, kan er door op de ‘Initialise’-knop in het instellingscherm te duwen, voorbeelddata in de databank worden ingesteld. Deze voorbeelddata wordt uit het *EyeVideo/app/src/main/assets/example.json*-bestand gelezen en bevat data voor één gebruiker met ID 99 en zes bijhorende ViewHistoryrecords. De ViewHistoryrecords bevatten fictieve similariteitscores voor de zes films: drie films (komedies en animatiefilms) krijgen hoge scores en drie films (horrors en drama’s) krijgen lage score. De knop zorgt er ook voor dat de gebruiker met ID 99 wordt ingelogd en hierdoor kan een simulatie van een aanbeveling getoond worden: de gebruiker zal films aanbevolen krijgen die lijken op de films die hij een hoge score gaf (vooral andere animatiefilms en komedies).

7 Uitvoeren unittesten

De testen met bijhorende testdata zitten verwerkt in het Androidproject en zijn terug te vinden in *app/src/androidTest/java/SimpleEntityReadWriteTest.java* en *app/src/androidTest/java/InstrumentedTest.java*. Ze worden uitgevoerd vanuit Android Studio door naar de klassen te navigeren in het linker paneel dat de klassen van het project toont. Door met de rechtermuisknop op de klassen te klikken kan de actie ‘Run InstrumentedTest’ en ‘Run SimpleEntityReadWriteTest’ uitgevoerd worden. Opnieuw opent Android Studio een venster waarin het toestel moet gekozen worden om de test in te laten draaien. De instrumented testen van Android maken namelijk gebruik van de lokale databank waardoor de testen moeten gedraaid worden op het toestel zelf, zoals uitgelegd in 4.7 in Hoofdstuk 4.

8 Mogelijke problemen

Bijna alle handelingen in de app, behalve het raadplegen van de data uit de lokale databank, vereisen het gebruik van internet. Indien er op de startknop, de ‘Show video’-knoppen of de ‘Face Storage’-knop geklikt wordt en er geen internetverbinding is, geeft de app dit aan met een waarschuwing onderaan het scherm en zal de app geen actie uitvoeren. Indien de internetverbinding wel aanwezig is maar te zwak is, zal dit zich ook uiten in de app door het lang laden bij het aanmelden en bij het afspelen van de video’s. Doordat de aanvragen naar de services te lang duren, kan er data in de HistoryDatarecords ontbreken. Om de app ideaal uit te testen is dus een goede internetverbinding vereist. Tijdens de testen met de testpersonen werd er steeds gezorgd voor een goede wifiverbinding.

De app kan ook enkele seconden nodig hebben voor de berekening van de similariteiten op de HistoryDatarecords van gebruikers. De berekening wordt uitgevoerd bij het aanvragen van de aanbeveling op alle HistoryDatarecords van alle ingelogde personen en deze allemaal overlopen kan bijgevolg enkele seconden duren.

Een andere verzameling verwachte emoties kan de volledige werking en scoreverdeling van de app veranderen. Momenteel bevat de app enkel verwachte emoties voor video’s 1 tot en met 6 omdat er enkel naar de emoties bij deze video’s onderzoek werd gedaan met verschillende sets (zoals uitgebreid besproken in Hoofdstuk 5). Alle 30 video’s kunnen bekeken worden terwijl impliciete feedback verzameld wordt: aandachtsniveau, emoties en gedrag worden allemaal gedetecteerd, maar enkel voor video 1 tot en met 6 kan een similariteitscore op basis van emoties berekend worden.

Een laatste en niet onbelangrijk probleem dat sinds 6 juni 2018 opgemerkt is, is dat de service Face++ sinds deze datum op hun site aankondigt dat de service voor klanten binnen Europa tijdelijk onbeschikbaar is. De service kan elk moment weer actief worden maar in de tussentijd zal de app het even zonder Face++ moeten doen. Dit is echter helemaal geen probleem aangezien de impliciete feedback kan blijven verzameld worden zonder dat de app crasht. Face++ vult in dat geval, wanneer hij aan de beurt is, zijn vereiste data niet in waardoor hoofdpositie en gesloten ogen beide de waarde nul krijgen. Dit zorgt ervoor dat het aandachtsniveau iets minder nauwkeurig berekend wordt (enkel nog op basis van Kairos) maar vormt voor de rest geen probleem voor de impliciete feedback. De testen (leeftijd, geslacht, emoties ...) kunnen bijgevolg ook niet uitgevoerd worden door Face++. Tijdens de evaluatietesten van de services en de testen met de testpersonen (februari tot mei 2018, toen Face++ wel nog actief was) heeft deze service reeds bewezen te werken en alle data werden opgeslagen en toegevoegd op de bijgeleverde USB-stick.

Bijlage C: Grafieken, diagrammen, code en tabellen

Deze bijlage toont het overzicht van de verschillende grafieken, diagrammen, codevoorbeelden en tabellen beschreven in de hoofdstukken. Indien nodig, worden afbeeldingen op een aparte pagina geroteerd getoond zodat ze er volledig oppassen.

Emoties worden met hun Engelstalige naam steeds alfabetisch geordend en volgen daarom de nummering zoals in Tabel C.1. Enkel bij de emotietesten wijkt de volgorde (en ook de kleuren in de grafieken) af van de alfabetische volgorde. De code en tabellen met resultaten volgen wel steeds de alfabetische volgorde.

Tabel C.1: Overzicht van de gebruikte nummering van emoties doorheen de tabellen en code

getal	emotie
0	anger
1	disgust
2	fear
3	happiness
4	sadness
5	surprise

De codevoorbeelden bevatten enkel de essentie van de code. Sommige delen worden weggelaten met “...” om aan te duiden dat de focus niet op die regels code ligt of omdat er een vervolg van de code in een volgende listing zal gedemonstreerd worden. Als bijschrift bij elke listing staat de klasse waartoe het codevoorbeeld behoort. De volledige code en alle Javaklassen zijn terug te vinden op de USB-stick in de map *Code*.

```

1 public Map<Integer, Double> giveRecommendation(List<User> users) {
2     Map<Integer,Map<Integer,Double>> candidates = new HashMap<>();
3     Map<Integer,Double> candidatesScores = new HashMap<>();
4     Map<Integer,Integer> amountsCandidatesScores = new HashMap<>();
5
6     for(User user : users){
7         ViewHistory[] histories = global.getDatabaseHistoryHelper().getViewHistoryFromUser(
            user.getUserId());
8         Map<Integer,Double> candidatesForThisUser = new HashMap<>();
9         Map<Integer,Double> tagScoresMap = new TreeMap<>(Collections.reverseOrder());
10        Map<Integer,Integer> amounts = new HashMap<>();
11
12        for(ViewHistory history : histories){
13            unacceptable.add(history.getVideoId());
14
15            for(TagToVideo tag : global.getDatabaseVideoHelper().getAllTagsFromVideoId(
                history.getVideoId())){
16                int id = tag.getTagId();
17                if(!tagScoresMap.containsKey(id)){
18                    tagScoresMap.put(id,history.getSimilarityDistance());
19                    amounts.put(id,1);
20                } else {
21                    tagScoresMap.put(id, tagScoresMap.get(id) + history.
                        getSimilarityDistance());
22                    amounts.put(id, amounts.get(id)+1);
23                }
24            }
25        }
26        ... // calculate average
27
28        amounts = new HashMap<>();
29        Integer[] videoids = global.getDatabaseVideoHelper().getAllVideoIds();
30        for(int videoid : videoids){
31            candidatesForThisUser.put(videoid, 0.0);
32            amounts.put(videoid,0);
33        }
34
35        for (Map.Entry<Integer, Double> entry : tagScoresMap.entrySet()) {
36            TagToVideo[] tagToVideos = global.getDatabaseVideoHelper().getAllTagsFromTagId(
                entry.getKey());
37
38            for(TagToVideo tagToVideo : tagToVideos){
39                if(candidatesForThisUser.containsKey(tagToVideo.getVideoId())){
40                    candidatesForThisUser.put(tagToVideo.getVideoId(),
                        candidatesForThisUser.get(tagToVideo.getVideoId()) + entry.getValue
                            ());
41                }
42                if(amounts.containsKey(tagToVideo.getVideoId())){
43                    amounts.put(tagToVideo.getVideoId(), amounts.get(tagToVideo.getVideoId
                        ())+1);
44                }
45            }
46        }
47        ... // calculate average
48        candidates.put(user.getUserId(), candidatesForThisUser);
49    }
50    ...

```

Listing C.1: Codevoorbeeld uit HistoryRecommendation: berekenen van geschiedenisgebaseerde geschatte scores en de onaanvaardbare films voor de groep (deel 1)

```

...
    for (User user : users) {
        int number = global.getDatabaseHistoryHelper().getViewHistoryFromUser(user.getUserId())
            ().length;
        int weightNoHistory = MIN;
        int weightHistory = MAX;
        if (number < MAX/THRESHOLDWEIGHT) {
            weightHistory = number*THRESHOLDWEIGHT;
            weightNoHistory = MAX - number*THRESHOLDWEIGHT;
        }

        NoHistoryRecommendation noHistoryRecommendation = new NoHistoryRecommendation(true);
        List<User> usersNoHistory = new ArrayList<>();
        usersNoHistory.add(user);
        Map<Integer,Double> listNohistory = noHistoryRecommendation.giveRecommendation(
            usersNoHistory);
        unacceptable.addAll(noHistoryRecommendation.getUnacceptable());
        Map<Integer,Double> listHistory = candidates.get(user.getUserId());

        if (number < MAX/THRESHOLDWEIGHT) {
            for (Map.Entry<Integer,Double> video : listNohistory.entrySet()) {
                double value = video.getValue()*weightNoHistory/MAX;
                listNohistory.put(video.getKey(), value);
            }
            for (Map.Entry<Integer,Double> video : listHistory.entrySet()) {
                double value = video.getValue()*weightHistory/MAX;
                listHistory.put(video.getKey(), value);
            }
        }

        Map<Integer,Double> candidatesForThisUser = listHistory;
        for (Map.Entry<Integer,Double> entry : listNohistory.entrySet()) {
            ...
            candidatesForThisUser.put(entry.getKey(), candidatesForThisUser.get(entry.
                getKey()) + entry.getValue());
        }

        for (Map.Entry<Integer,Double> video : candidatesForThisUser.entrySet()) {
            ...
            candidatesScores.put(video.getKey(), candidatesScores.get(video.getKey())+video.
                getValue());
            amountsCandidatesScores.put(video.getKey(), amountsCandidatesScores.get(video.
                getKey())+1);
        }
    }
    ... // calculate average
    return candidatesScores;
}

```

Listing C.2: Codevoorbeeld uit HistoryRecommendation: berekenen van geschiedenisgebaseerde geschatte scores en de onaanvaardbare films voor de groep (deel 2)

```

public double updateCalculatedViewHistory(int historyId){
    HistoryData[] datas = global.getDatabaseHistoryHelper().
        getAllHistoryDataFromHistoryId(historyId);
    if(datas != null && datas.length != 0){
        int count = 0, countAnger = 0, countDisgust = 0, countFear = 0, countHappiness
            =0, countSadness=0, countSurprise=0;
        double sumAnger=0, sumDisgust=0, sumFear=0, sumHappiness=0, sumSadness=0,
            sumSurprise=0;
        double diff;
        for(HistoryData data : datas){
            int relativeTime = data.getRelativeTime();
            int videoId = global.getDatabaseHistoryHelper().getViewHistoryFromHistoryId
                (historyId).getVideoId();

            // 1. EMOTIONS
            ExpectedEmotion[] expectedEmotion = global.getDatabaseVideoHelper().
                getAllExpectedEmotionsFromVideoIdAndRelativeTime(videoId,relativeTime);
            if(expectedEmotion != null && expectedEmotion.length != 0) {
                ExpectedEmotion expected = expectedEmotion[0];
                switch(method){
                    case ALLSECONDS:
                        count++;
                        ExpectedEmotion expectedBefore = expected;
                        ExpectedEmotion expectedAfter = expected;
                        if(relativeCount != 0){
                            ExpectedEmotion[] expectedBefores = global.
                                getDatabaseVideoHelper().
                                getAllExpectedEmotionsFromVideoIdAndRelativeTime(
                                    videoId,relativeTime-TIMESTEP);
                            if(expectedBefores != null && expectedBefores.length != 0){
                                expectedBefore = expectedBefores[0];
                            }
                        }
                        ExpectedEmotion[] expectedAfters = global.
                            getDatabaseVideoHelper().
                            getAllExpectedEmotionsFromVideoIdAndRelativeTime(videoId,
                                relativeTime+TIMESTEP);
                        if(expectedAfters != null && expectedAfters.length != 0){
                            expectedAfter = expectedAfters[0];
                        }

                        double averageExpected = (expected.getAnger() + expectedAfter.
                            getAnger() + expectedBefore.getAnger())/AVERAGEDIVIDER;
                        diff = (data.getAngers() - averageExpected)*(data.getAngers() -
                            averageExpected);
                        sumAnger+=diff;
                        ... // other emotions
                        ... // calculate highest emotion
                        ... // calculate average for margins method
                        break;
                    case SKIPSECONDS:
                        boolean keyScene = false;
                        count++;
                        if(expected.getAnger()!=0){
                            diff = (data.getAngers() - expected.getAnger()*(data.
                                getAngers() - expected.getAnger()));
                            sumAnger+=diff;
                            countAnger++;
                            keyScene = true;
                        }
                        ... // other emotions
                        ... // calculate average for margins method
                        if(keyScene){
                            ... // calculate highest emotion
                        }
                        break;
                }
            }
        }
    }
    ...
}

```

Listing C.3: Codevoorbeeld uit HistoryProcessor: berekenen van de impliciete feedback op basis van emoties, aandachtsniveau en gedrag (deel 1)

```

...
switch (method) {
    case ALLSECONDS:
        if(count!=0) {
            sumAnger/=count;
            ... // other emotions
        }
        break;
    case SKIPSECONDS:
        if(countAnger!=0) sumAnger/=countAnger;
        else sumAnger = LARGESTDISTANCE;
        ... // other emotions
        break;
}

sumAnger = Math.sqrt(sumAnger);
... // other emotions

sumAnger = 1/(1+sumAnger);
... // other emotions

for(HistoryData data : datas){
    int relativeCount = data.getRelativeCount();
    int videoId = global.getDatabaseHistoryHelper().getViewHistoryFromHistoryId
        (historyId).getVideoId();

    ... // calculate margins score

    // 2. ATTENTION
    boolean hasAttention;
    if(data.getAttentionYaws() > -THRESHOLDYAW && data.getAttentionYaws() <
        THRESHOLDYAW) attentionYaw++;
    if(data.getAttentionPitches() > -THRESHOLDPITCH && data.getAttentionPitches()
        < THRESHOLDPITCH) attentionPitch++;
    if(data.getAttentionYaws() > -THRESHOLDYAW && data.getAttentionYaws() <
        THRESHOLDYAW && data.getAttentionPitches() > -THRESHOLDPITCH && data.
        getAttentionPitches() < THRESHOLDPITCH){
        hasAttention = true;
    } else {
        hasAttention = false;
    }
    if(data.getRelativeCount()%(FaceApplication.INTERVALTIMEKAIROS/
        FaceApplication.INTERVALTIME) == 0 && (data.getPhotoname() != null &&
        data.getAttentions() < THRESHOLDATTENTION)){
        hasAttention = false;
    } else {
        attention++;
    }
    if(data.getLefteyeclose() >= THRESHOLDEYESCLOSED && data.getRighteyeclose()
        >= THRESHOLDEYESCLOSED){
        hasAttention = false;
        eyesClosed++;
    }
    if(hasAttention) totalAttention++;

    // 3. BEHAVIOUR
    if(data.isEyeOcclusions()) eyeOccluded++;
    if(data.isForeheadOcclusions()) foreheadOccluded++;
    if(data.isMouthOcclusions()) mouthOccluded++;

    totalPhotos++;
}
... // calculate highest emotion
... // update info in ViewHistoryrecord
calculateSimilarityFromDB(historyId, sumAnger, sumDisgust, sumFear,
    sumHappiness, sumSadness, sumSurprise);
return sumHappiness;
}
return 0;
}

```

Listing C.4: Codevoorbeeld uit HistoryProcessor: berekenen van de impliciete feedback op basis van emoties, aandachtsniveau en gedrag (deel 2)

Tabel C.2: Overzicht van de resultaten van alle testpersonen per video bij de testen. De emotiegetallen betekenen:

0 = anger, 1 = disgust, 2 = fear, 3 = happiness, 4 = sadness, 5 = surprise

	attention: all sets						difference attention: all sets						maximum emotions values							
	video	1	2	3	4	5	6	1	2	3	4	5	6	emotions	0	1	2	3	4	5
nr		/4						/10							/1					
1		3.58	3.94	3.91	4.00	3.85	3.81	0.94	1.84	0.78	5.00	0.63	3.52		0.02	0.02	0.02	1.00	0.46	0.03
2		3.75	3.84	3.82	3.95	3.91	1.29	1.38	3.60	2.54	8.87	4.78	2.23		0.00	0.02	0.01	1.00	0.59	0.21
3		3.82	3.77	3.68	3.85	3.36	3.62	6.55	2.42	0.20	6.62	1.41	4.04		0.02	0.11	0.13	1.00	0.42	0.08
4		3.63	3.77	3.85	3.58	3.70	3.81	3.08	1.44	1.62	3.95	5.26	7.52		0.05	0.06	0.01	1.00	0.36	0.20
5		4.00	3.90	3.94	4.00	3.93	3.97	2.00	1.76	2.86	3.50	3.81	3.42		0.40	0.07	0.01	0.98	0.29	0.68
6		3.63	3.48	3.85	3.65	3.74	3.94	2.07	2.70	1.63	3.11	2.35	2.84		0.01	0.01	0.13	0.00	0.52	0.22
7		4.00	3.93	3.94	3.74	3.59	3.94	2.00	5.84	2.85	5.35	3.97	3.84		0.01	0.01	0.01	1.00	0.16	0.15
8		4.00	3.97	3.91	3.88	3.89	4.00	3.00	1.91	1.78	3.71	1.74	3.00		0.20	0.08	0.04	1.00	0.31	0.52
9		4.00	3.90	3.88	4.00	3.85	4.00	2.00	3.76	0.71	3.00	2.63	3.00		0.17	0.36	0.27	1.00	0.66	0.21
10		3.82	3.90	4.00	3.90	3.89	3.60	2.55	0.24	2.00	6.75	3.71	8.00		0.11	0.01	0.00	1.00	0.10	0.01
11		4.00	4.00	4.00	3.84	3.96	4.00	3.00	2.00	1.00	8.60	2.39	10.00		0.16	0.02	0.09	1.00	0.23	0.58
12		3.88	4.00	3.91	3.44	3.92	2.31	2.71	1.00	3.78	5.59	6.80	3.79		0.49	0.26	0.06	1.00	0.53	0.48
13		3.82	3.76	3.59	3.69	3.44	3.46	3.04	0.61	1.47	2.23	0.60	0.35		0.51	0.49	0.13	1.00	0.60	0.62
14		3.88	3.91	3.88	3.95	3.77	3.90	2.69	1.77	2.70	4.87	2.43	3.75		0.50	0.16	0.06	1.00	0.55	0.93
15		3.88	3.78	3.73	3.54	3.55	3.55	3.70	1.94	1.33	4.86	3.87	4.37		0.00	0.01	0.00	1.00	0.10	0.01
16		3.51	3.54	3.51	3.67	2.65	3.61	2.77	0.86	0.78	7.18	2.62	8.04		0.08	0.14	0.25	1.00	0.96	0.12
17		3.88	4.00	3.97	3.85	3.85	3.75	1.69	1.00	4.92	8.62	1.62	6.37		0.26	0.16	0.00	1.00	0.68	0.04
18			2.67						2.67						0.02	0.02	0.08	0.87	0.11	0.75
19			1.58						4.04						0.06	0.01	0.00	1.00	0.21	0.33
20		2.48	3.87	3.88	2.68	3.40	2.13	2.79	0.68	3.21	2.71	6.49	4.32		0.04	0.04	0.00	1.00	0.21	0.31
average		3.75	3.68	3.85	3.73	3.68	3.48	2.66	2.10	2.01	5.25	3.17	4.58	max	0.51	0.49	0.27	1.00	0.96	0.93
%															0.51	0.49	0.27	1.00	0.68	0.58

Tabel C.3: Overzicht van de resultaten van alle testpersonen per video bij set 3 (deel 1): de verwachte emotie staat voor elke film bovenaan in de ‘highest emotion’-kolom vermeld met het emotiegetal (0 = anger, 1 = disgust, 2 = fear, 3 = happiness, 4 = sadness, 5 = surprise)

nr	emotion video	highest emotion: set 3						emotionScore: set 3						typicalEmotionScore: set 3					
		1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
		3	4	0	1	2	5	/6						/6					
1		3	4	4	4	4	4	5.55	5.76	5.88	5.49	5.49	5.43	4.73	5.84	5.85	5.78	5.87	5.27
2		3	4	4	4	4	3	5.29	5.73	5.80	5.42	5.42	5.42	4.05	5.82	5.85	5.79	5.88	5.28
3		4	4	4	3	4	4	5.16	5.75	5.88	5.52	5.42	5.45	3.56	5.86	5.85	5.76	5.87	5.27
4		3	4	4	3	4	4	5.61	5.75	5.90	5.49	5.40	5.45	4.90	5.77	5.86	5.79	5.86	5.27
5		3	4	0	3	4	3	5.27	5.78	5.89	5.57	5.47	5.46	3.89	5.82	5.88	5.79	5.87	5.27
6		4	4	4	4	4	4	4.98	5.49	5.86	5.46	5.49	5.35	3.46	5.13	5.85	5.78	5.86	5.32
7		3	4	4	4	4	4	5.58	5.76	5.79	5.49	5.45	5.41	4.83	5.82	5.85	5.78	5.88	5.26
8		3	4	4	3	5	3	5.52	5.69	5.76	5.47	5.52	5.62	4.64	5.76	5.81	5.79	5.89	5.76
9		3	4	3	3	3	3	5.45	5.72	5.82	5.49	5.35	5.44	4.42	5.78	5.85	5.79	5.87	5.26
10		3	3	4	4	4	3	5.22	5.76	5.88	5.47	5.39	5.39	3.74	5.81	5.85	5.78	5.87	5.32
11		4	4	4	3	4	4	5.14	5.73	5.88	5.50	5.55	5.42	3.50	5.75	5.84	5.79	5.88	5.27
12		3	4	4	4	4	4	5.21	5.73	5.85	5.54	5.49	5.41	3.72	5.75	5.78	5.80	5.88	5.26
13		3	4	0	0	4	4	5.27	5.60	5.84	5.49	5.51	5.46	3.89	5.62	5.74	5.84	5.86	5.28
14		3	4	4	4	4	4	5.21	5.67	5.78	5.43	5.41	5.40	3.74	5.62	5.85	5.80	5.86	5.26
15		3	3	3	3	3	3	5.60	5.78	5.73	5.53	5.30	5.36	4.88	5.81	5.85	5.78	5.87	5.27
16		3	4	4	4	4	4	5.16	5.61	5.81	5.38	5.37	5.37	4.02	5.36	5.86	5.74	5.81	5.32
17		3	4	4	3	4	4	5.48	5.71	5.75	5.55	5.44	5.42	4.57	5.75	5.85	5.79	5.87	5.27
18			4						5.70						5.81				
19			4						5.68						5.79				
20		3	4	4	3	3	4	5.46	5.74	5.88	5.47	5.44	5.44	4.44	5.81	5.86	5.78	5.87	5.28
average		15	18	2	0	0	0	5.34	5.71	5.83	5.49	5.44	5.43	4.17	5.72	5.84	5.79	5.87	5.31
/ total																			

Tabel C.4: Overzicht van de resultaten van alle testpersonen per video bij set 3 (deel 2)

video nr	scoreDistance: set 3						difference emotionScore: set 3						difference typEmotionScore: set 3					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
	/10						/10						/10					
1	9.13	9.63	9.79	9.49	9.34	9.22	1.26	1.60	0.80	4.16	0.14	3.05	0.11	1.73	0.75	4.64	0.78	2.78
2	9.04	9.52	9.53	9.37	8.89	5.20	0.82	3.54	2.67	8.04	4.04	8.04	1.25	3.70	2.75	8.64	4.80	7.80
3	8.98	9.36	9.56	9.37	8.78	9.06	5.60	2.59	0.80	6.20	2.03	4.08	2.93	2.76	0.75	6.61	2.78	3.78
4	9.24	9.43	9.68	8.99	8.74	9.26	3.35	1.59	1.83	4.14	5.01	7.08	2.17	1.62	1.76	4.65	5.77	6.78
5	9.27	9.61	9.83	9.57	9.40	9.42	0.79	1.63	2.81	2.78	3.12	2.59	1.51	1.69	2.80	3.14	3.78	2.29
6	8.61	8.85	9.72	9.11	9.22	9.29	1.30	3.15	1.77	3.10	2.14	1.92	1.24	2.54	1.75	3.63	2.77	1.87
7	9.58	9.55	9.68	9.19	8.76	9.32	1.31	5.60	2.64	5.15	4.09	3.02	0.05	5.71	2.75	5.63	4.79	2.77
8	8.55	9.35	9.67	9.00	8.54	7.48	2.20	1.48	1.60	3.11	1.20	2.36	0.73	1.60	1.68	3.65	1.81	2.60
9	9.45	9.51	9.70	9.49	9.20	9.41	1.08	3.53	0.70	2.16	1.92	2.06	0.64	3.63	0.75	2.64	2.78	1.77
10	9.04	9.55	9.88	9.37	9.22	8.39	1.70	0.40	1.79	6.12	2.99	7.98	0.77	0.31	1.75	6.64	3.79	7.87
11	9.05	9.71	9.77	9.23	9.17	8.29	1.57	1.56	0.80	8.17	1.75	9.03	1.17	1.59	0.74	8.65	2.31	8.79
12	9.09	9.73	9.76	8.70	9.27	7.62	1.69	0.54	3.75	6.23	6.15	7.01	0.80	0.58	3.63	6.67	6.79	6.76
13	9.09	9.36	9.43	9.19	8.95	8.92	2.29	0.66	2.24	2.16	1.18	0.09	0.01	0.64	2.07	2.73	1.77	0.20
14	9.04	9.58	9.66	9.38	9.13	9.20	1.69	1.46	2.63	4.05	2.02	2.99	0.77	1.37	2.75	4.67	2.76	2.77
15	9.48	9.49	9.44	9.07	8.82	8.89	3.33	2.13	1.54	5.21	3.83	4.44	2.13	2.19	1.75	5.63	4.78	4.28
16	8.30	8.83	8.14	7.94	6.92	7.98	2.60	1.35	1.68	6.96	4.95	7.96	0.71	0.93	1.76	7.57	5.68	7.87
17	9.36	9.71	9.61	9.39	9.24	9.17	1.14	0.51	4.58	8.24	1.07	6.04	0.38	0.58	4.75	8.64	1.78	5.79
18		6.49						5.50						5.69				
19		6.75						1.47						1.64				
20	7.94	9.51	9.76	8.16	8.81	7.57	0.09	0.56	3.30	5.12	7.06	8.07	1.60	0.69	3.26	5.63	7.78	7.80
average	9.01	9.18	9.59	9.11	8.91	8.54	1.88	2.04	2.11	5.06	3.04	4.88	1.05	2.06	2.12	5.56	3.75	4.70

Tabel C.5: Overzicht van de resultaten van alle testpersonen per video bij set 4 en 4.2 (deel 1): de verwachte emotie staat voor elke film bovenaan in de 'highest emotion'-kolom vermeld met het emotiegetal (0 = anger, 1 = disgust, 2 = fear, 3 = happiness, 4 = sadness, 5 = surprise)

nr	emotion video	highest emotion: set 4						emotionScore: set 4						typicalEmotionScore: set 4.2					
		1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
		3	4	0	1	2	5	/6						/6					
1		3	4	4	3	4	4	3.88	3.37	4.74	3.85	4.51	3.87	5.63	3.56	4.74	4.27	4.51	4.31
2		3	4	4	4	3	3	3.26	3.34	4.63	3.75	4.51	3.87	3.77	3.51	4.63	4.13	4.51	4.31
3		4	4	4	3	4	4	3.04	3.36	4.64	3.88	4.51	3.87	3.13	3.54	4.64	4.32	4.51	4.31
4		3	4	4	3	4	4	3.50	3.42	4.72	3.78	4.63	3.87	4.51	3.63	4.72	4.17	4.63	4.31
5		3	3	0	3	4	3	3.13	3.44	4.63	3.97	4.51	3.87	3.40	3.65	4.63	4.45	4.51	4.31
6		4	4	4	4	4	4	3.00	3.46	4.78	3.68	4.55	3.91	3.00	3.69	4.78	4.02	4.55	4.36
7		3	3	4	3	4	4	3.46	3.47	4.62	3.72	4.51	3.87	4.38	3.71	4.62	4.08	4.51	4.31
8		3	4	4	3	5	3	3.93	4.00	4.79	3.87	4.53	3.91	5.78	4.51	4.79	4.30	4.53	4.37
9		3	3	3	3	3	3	3.39	3.43	4.62	3.74	4.51	3.87	4.16	3.64	4.62	4.11	4.51	4.31
10		3	3	4	3	3	3	3.11	3.41	4.62	3.71	4.51	3.87	3.34	3.61	4.62	4.07	4.51	4.31
11		4	4	4	3	4	3	3.02	3.39	4.66	3.75	4.51	3.87	3.05	3.59	4.66	4.12	4.51	4.31
12		3	3	4	3	4	4	3.07	3.50	4.73	3.78	4.51	3.87	3.21	3.74	4.73	4.17	4.51	4.31
13		3	4	0	0	4	3	3.18	3.36	4.67	3.74	4.55	3.89	3.53	3.53	4.67	4.11	4.55	4.33
14		3	4	4	4	4	4	3.13	3.46	4.80	3.73	4.51	3.88	3.39	3.69	4.80	4.10	4.51	4.32
15		3	3	3	3	3	3	3.62	3.57	4.62	3.92	4.51	3.87	4.86	3.85	4.62	4.38	4.51	4.31
16		3	3	4	4	3	4	3.20	3.53	4.66	3.71	4.52	3.89	3.59	3.80	4.66	4.07	4.52	4.34
17		3	3	3	3	4	4	3.42	3.69	4.66	3.91	4.52	3.87	4.25	4.04	4.66	4.37	4.52	4.31
18			4						3.44						3.66				
19			5						3.40						3.61				
20		3	3	4	3	3	4	3.29	3.34	4.62	3.70	4.51	3.88	3.88	3.51	4.62	4.05	4.51	4.32
average		15	10	2	0	0	0	3.31	3.47	4.68	3.79	4.52	3.88	3.94	3.70	4.68	4.18	4.52	4.32
%								100.00						16.67	83.33				

Tabel C.6: Overzicht van de resultaten van alle testpersonen per video bij set 4 en 4.2 (deel 2)

		difference typEmotionScore: set 4						scoreDistance: set 4						scoreDistance: set 4.2					
nr	video	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
		/10						/10						/10					
	1	1.39	2.07	1.10	2.11	1.49	1.18	7.45	7.24	8.65	7.85	8.36	7.65	9.21	7.42	8.65	8.27	8.36	8.09
	2	1.71	0.15	0.72	5.89	2.51	6.18	7.01	7.13	8.36	7.70	7.97	3.65	7.53	7.31	8.36	8.08	7.97	4.09
	3	2.21	1.11	1.27	4.19	0.51	2.18	6.86	6.96	8.32	7.73	7.87	7.49	6.95	7.14	8.32	8.16	7.87	7.92
	4	1.52	1.95	0.13	1.94	3.72	5.18	7.13	7.10	8.50	7.28	7.97	7.68	8.14	7.31	8.50	7.67	7.97	8.12
	5	2.33	1.91	0.71	0.92	1.52	0.68	7.13	7.27	8.57	7.97	8.44	7.84	7.40	7.49	8.57	8.45	8.44	8.28
	6	2.00	0.14	0.03	0.70	0.58	0.27	6.63	6.82	8.63	7.32	8.29	7.84	6.63	7.05	8.63	7.66	8.29	8.30
	7	0.70	2.18	0.70	2.80	2.51	1.18	7.46	7.26	8.51	7.42	7.81	7.78	8.38	7.50	8.51	7.78	7.81	8.22
	8	2.64	0.49	0.01	1.17	0.45	0.28	6.96	7.66	8.71	7.40	7.55	5.77	8.81	8.17	8.71	7.84	7.55	6.23
	9	1.06	0.07	1.31	0.14	0.51	0.18	7.39	7.21	8.50	7.74	8.36	7.85	8.16	7.43	8.50	8.11	8.36	8.28
	10	1.43	3.98	0.31	3.78	1.51	6.18	6.93	7.19	8.62	7.61	8.34	6.87	7.16	7.40	8.62	7.97	8.34	7.31
	11	1.92	2.02	1.24	5.86	0.01	7.18	6.92	7.37	8.55	7.47	8.13	6.74	6.96	7.56	8.55	7.84	8.13	7.18
	12	1.66	2.76	1.88	3.94	4.51	5.18	6.95	7.50	8.64	6.94	8.29	6.09	7.09	7.74	8.64	7.33	8.29	6.53
	13	0.62	4.11	0.29	0.15	0.41	1.78	6.99	7.11	8.26	7.43	7.99	7.35	7.35	7.29	8.26	7.80	7.99	7.79
	14	1.35	1.85	0.99	1.83	0.51	1.20	6.96	7.37	8.68	7.68	8.22	7.68	7.22	7.60	8.68	8.05	8.22	8.12
	15	2.10	1.08	0.30	3.29	2.51	2.68	7.50	7.27	8.33	7.46	8.02	7.40	8.74	7.56	8.33	7.92	8.02	7.83
	16	0.02	1.67	0.24	4.78	3.54	6.23	6.33	6.75	6.99	6.27	6.08	6.50	6.73	7.01	6.99	6.63	6.08	6.94
	17	0.92	2.27	2.76	6.28	0.47	4.18	7.29	7.69	8.51	7.76	8.31	7.62	8.12	8.04	8.51	8.22	8.31	8.05
	18		2.10						4.23						4.45				
	19		1.99						4.47						4.68				
	20	2.53	3.15	1.20	2.74	5.51	6.21	5.78	7.12	8.51	6.38	7.88	6.01	6.37	7.29	8.51	6.73	7.88	6.45
average		1.56	1.85	0.84	2.92	1.82	3.23	6.98	6.94	8.44	7.41	7.99	6.99	7.61	7.17	8.44	7.81	7.99	7.43
%								100.00						16.67	5.56	77.78			

Tabel C.7: Verschil in geschatte leeftijd en geslacht voor de 20 testpersonen. Telkens schatte elke service de leeftijd vijf keer.

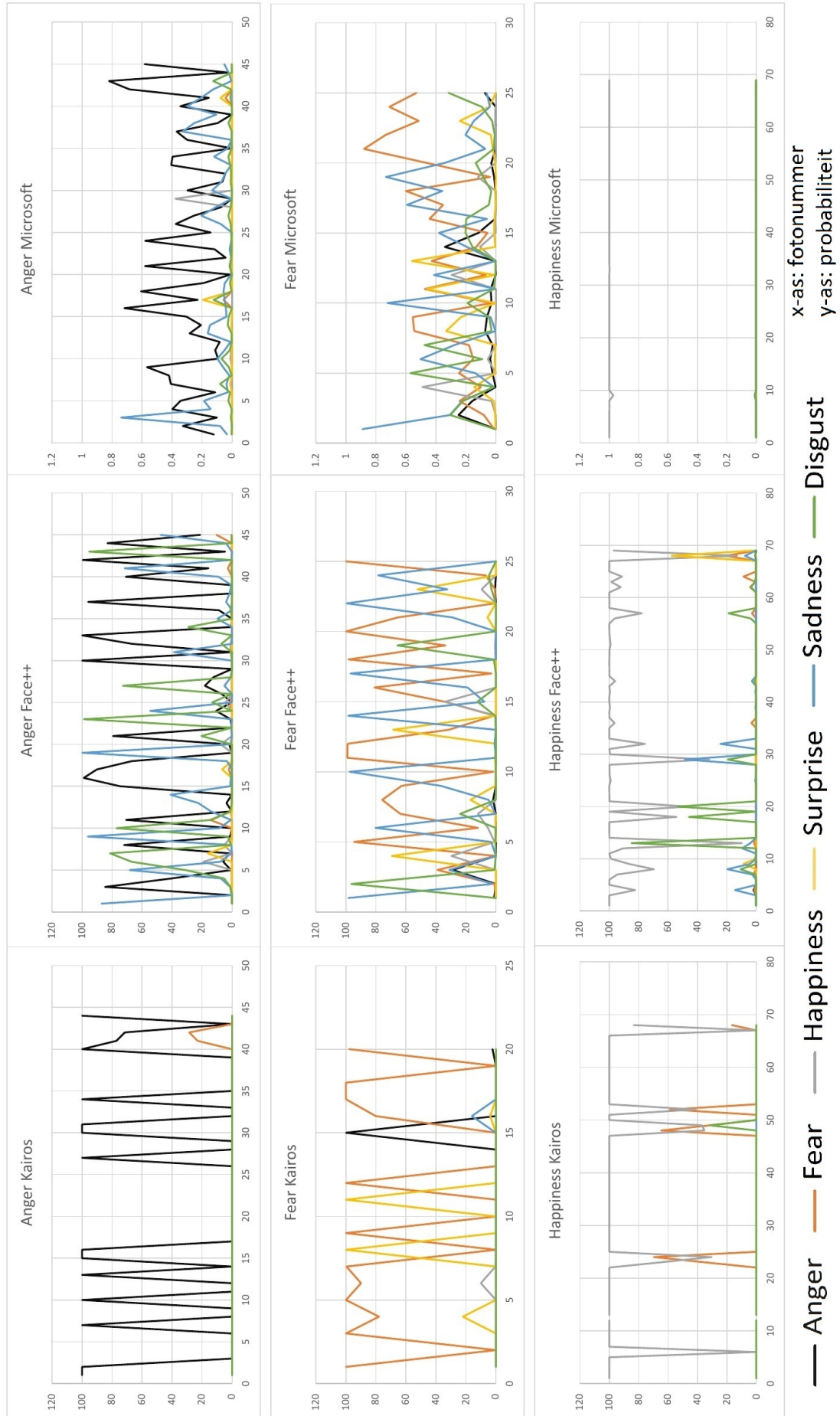
nr	age	gender	Kairos					gender	diff	Face++					gender	diff	Microsoft					gender	diff
1	23	M	30	35	35	32	33	M	10.00	44	46	53	47	40	M	23.00	35.5	39.2	32.9	35.5	36.1	M	12.84
2	22	F	28	31	31	28	28	F	7.20	39	38	42	37	42	F	17.60	31.4	30.0	32.0	27.9	26.3	F	7.52
3	22	F	28	24	24	24	24	F	2.80	29	31	27	31	24	F	6.40	17.9	18.7	21.2	20.4	19.0	F	2.56
4	51	F	43	47	45	46	43	F	6.20	50	49	48	47	48	F	2.60	42.6	42.1	33.3	44.5	38.9	F	10.72
5	51	M	44	42	48	43	45	M	6.60	46	44	59	50	45	M	2.20	39.3	39.4	40.4	40.1	32.9	M	12.58
6	17	M	25	26	24	/	/	M	8.00	21	32	30	27	29	M	10.80	16.6	19.4	18.3	17.5	18.6	M	1.08
7	22	F	33	36	36	37	30	F	12.40	38	38	37	36	39	F	15.60	21.2	21.9	23.1	23.0	25.1	F	0.86
8	22	M	27	24	25	28	26	M	4.00	40	36	37	40	39	M	16.40	25.6	25.6	27.5	30.9	26.0	M	5.12
9	22	M	27	26	26	26	26	M	4.20	33	33	41	41	38	M	15.20	30.3	30.9	31.1	30.7	30.8	M	8.76
10	23	M	22	25	25	23	24	M	0.80	23	33	35	33	27	M	7.20	22.1	24.4	22.1	22.8	21.3	M	0.46
11	22	F	25	25	25	23	25	F	2.60	35	36	40	39	42	F	16.40	22.6	22.1	24.7	23.4	20.2	F	0.60
12	23	F	25	28	30	32	33	F	6.60	41	38	37	43	42	F	17.20	20.4	22.7	23.5	24.9	24.4	F	0.18
13	20	M	27	27	26	27	25	M	6.40	30	30	32	30	32	M	10.80	20.5	21.3	21.9	22.2	23.5	M	1.88
14	22	F	27	27	26	24	25	F	3.80	39	39	40	37	39	F	16.80	27.3	27.6	25.6	25.7	25.6	F	4.36
15	52	M	53	43	54	53	56	M	0.20	50	35	50	51	49	M	5.00	50.6	43.4	50.3	50.4	51.7	M	2.72
16	52	F	41	42	45	41	43	F	9.60	44	39	43	43	0	F	18.20	37.8	37.6	34.9	33.3	41.9	F	14.90
17	72	F	62	63	62	62	61	F	10.00	70	69	70	71	68	F	2.40	77.5	83.7	78.8	81.8	80.2	F	8.40
18	5	F	23	23	38	29	23	F	22.20	18	21	65	9	19	F	21.40	6.0	5.9	5.3	5.4	5.9	F	0.70
19	5	M	28	30	25	31	29	M	23.60	24	24	22	27	16	M	17.60	7.1	7.0	9.0	11.8	6.6	M	3.30
20	22	M	24	28	29	30	29	M	6.00	30	34	34	28	36	M	10.40	30.5	29.2	30.7	30.1	28.0	M	7.70
average									7.66							12.66							5.36

Tabel C.8: Antwoorden van de testpersonen op de vragenlijst over hun mening over de privacy en hoe goed ze expliciete beoordelingen zouden geven

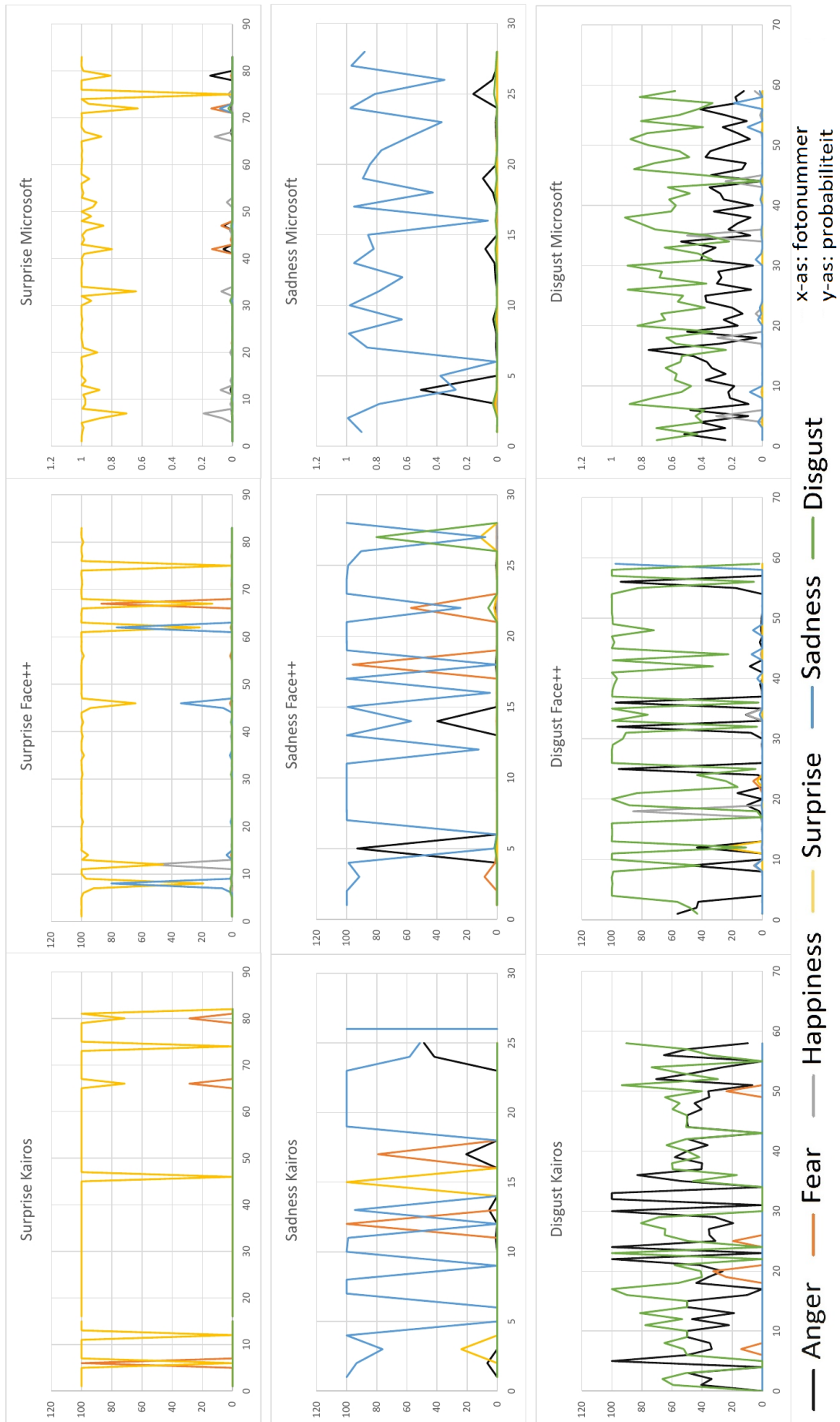
nr	app gebruiken						camera bedekken					beoordelen		
	internet	lokaal	tablet	smartphone	smartTV	laptop	tablet	smartphone	smartTV	laptop		tablet/ smartphone	smartTV	laptop
1		x			x	x	x			x		soms	geen	soms
2		x			x	x	x	x	x	x		altijd	geen	altijd
3		x					x		x	x		soms	soms	soms
4		x	x				x	x	x	x		altijd	soms	altijd
5		x					x	x	x	x		soms	geen	geen
6	x	x					x	x	x	x		soms	geen	soms
7		x	x	x	x	x		x	x	x		soms	soms	soms
8		x		x			x	x	x	x		soms	geen	soms
9		x							x			soms	geen	soms
10		x							x	x		geen	geen	geen
11		x							x	x		altijd	geen	soms
12	x	x	x	x		x	x	x	x	x		altijd	geen	altijd
13	x	x				x			x	x		altijd	soms	altijd
14		x			x	x	x	x	x	x		geen	geen	geen
15												soms	soms	soms
16		x					x	x	x	x		soms	altijd	geen
17		x					x	x	x	x		altijd	altijd	altijd
18														
19														
20		x		x		x	x	x	x	x		geen	geen	geen
average	3	17	3	4	4	7	12	11	16	16	soms	9	5	8
%	16.67	94.44	16.67	22.22	22.22	38.89	66.67	61.11	88.89	88.89	geen	3	11	5
											altijd	6	2	5

Tabel C.9: Antwoorden van de testpersonen op de vragenlijst over hun impliciete feedback

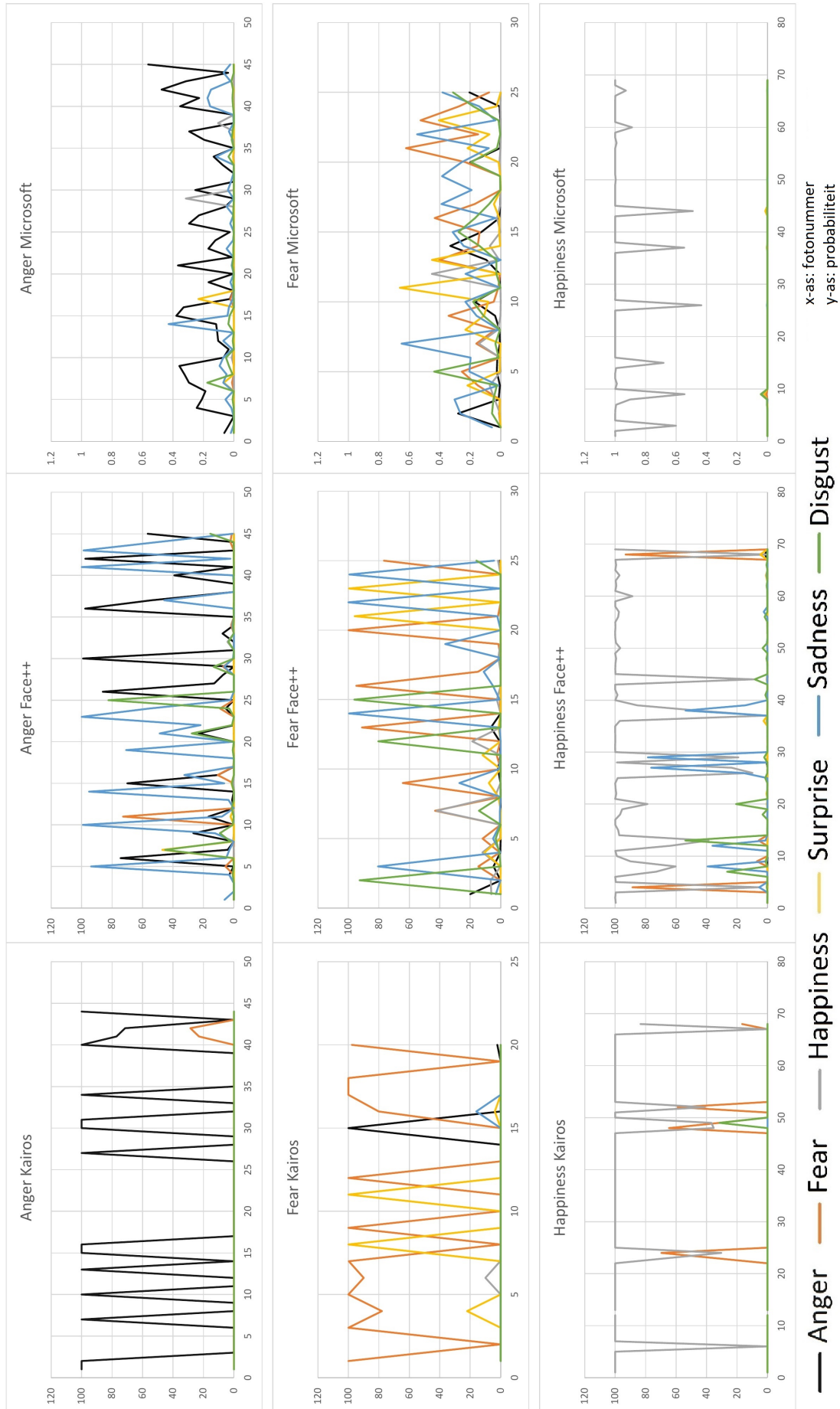
nr	aandacht	tonen emoties						reacties				
		blijheid	boosheid	afkeer	verrassing	angst	verdriet	wegkijken	ogen dicht	ogen bedekken	gezicht bedekken	handen voor mond
1	x	x					x	x				
2		x		x				x				
3	x			x			x	x		x		
4	x			x	x	x	x		x	x	x	
5	x	x			x			x				
6	x	x						x				
7	x	x						x				
8	x	x		x	x	x	x				x	x
9	x	x		x	x		x		x			x
10	x	x		x	x	x	x	x	x	x		
11	x	x		x	x	x	x	x	x			x
12	x	x		x		x	x	x	x			
13	x	x		x	x	x			x			
14	x	x	x	x	x	x	x	x			x	
15	x	x		x	x			x				
16	x	x		x		x	x	x				x
17	x	x	x	x	x	x	x	x				
18												
19												
20	x	x	x	x				x				
average	17	16	3	14	10	9	11	14	6	3	3	4
%	94.44	88.89	16.67	77.78	55.56	50.00	61.11	77.78	33.33	16.67	16.67	22.22



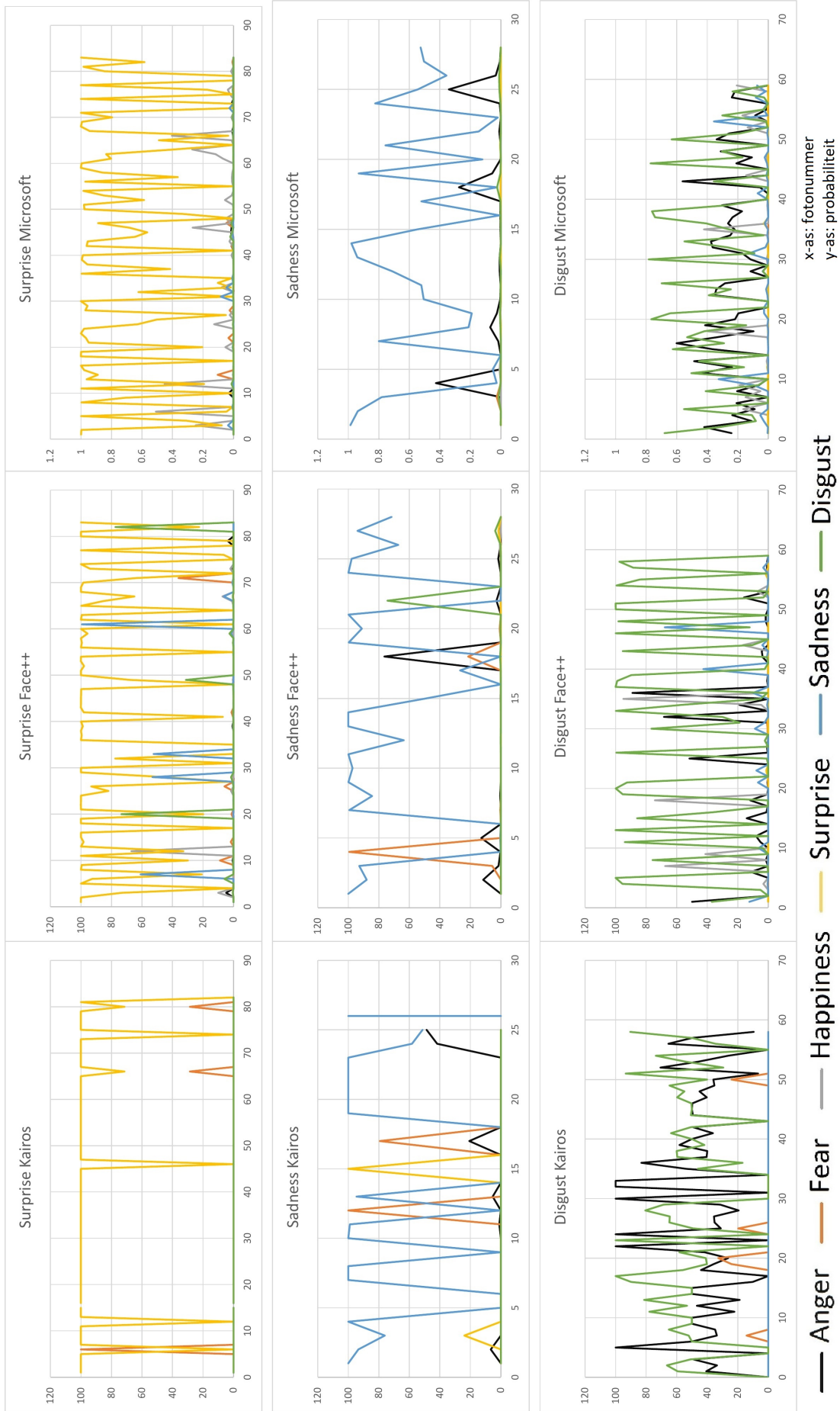
Figuur C.1: Vergelijking van de emotiedetectie bij de drie services toegepast op de laatste foto uit elke fotoreeks (deel 1)



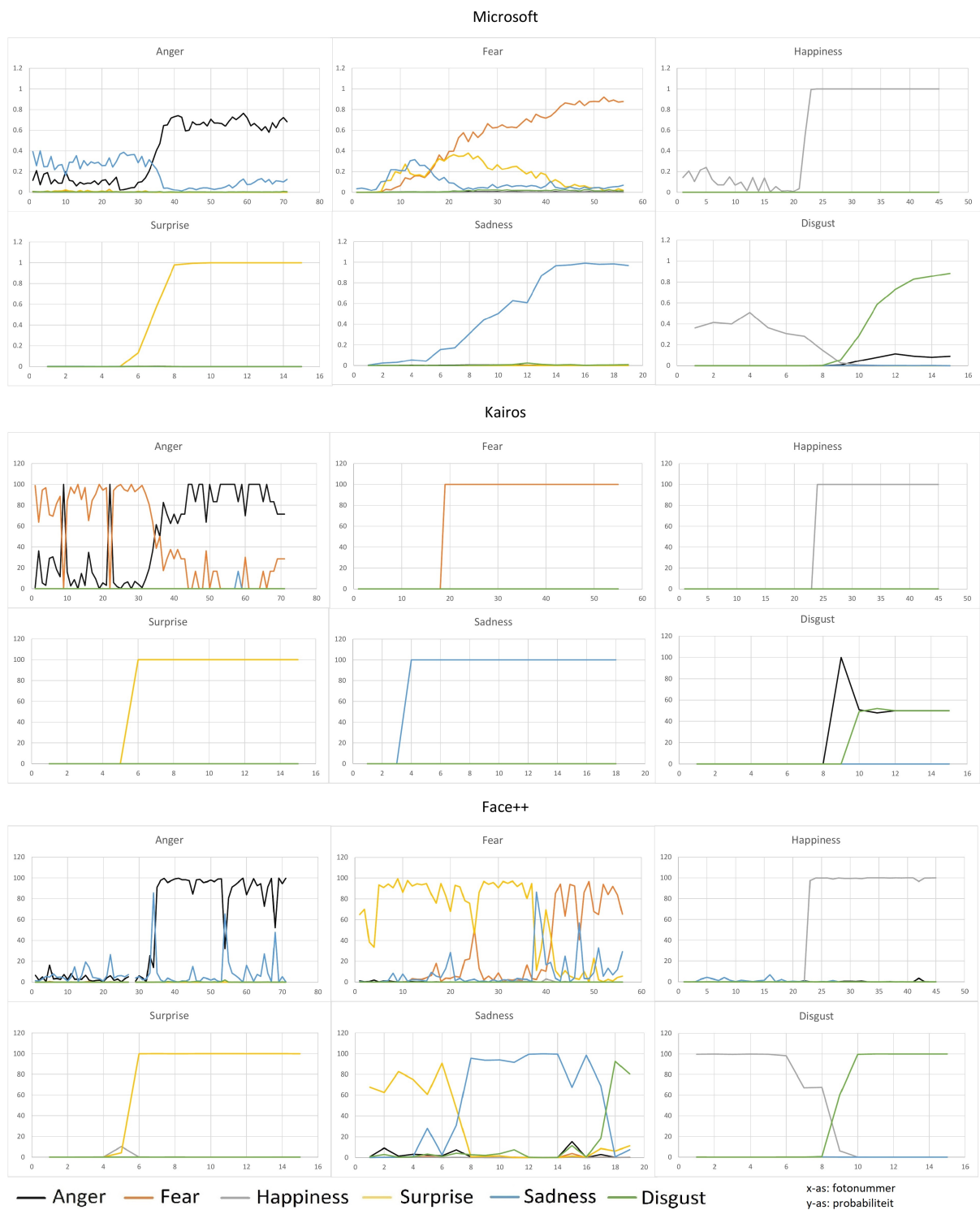
Figuur C.2: Vergelijking van de emotiedetectie bij de drie services toegepast op de laatste foto uit elke fotoreeks (deel 2)



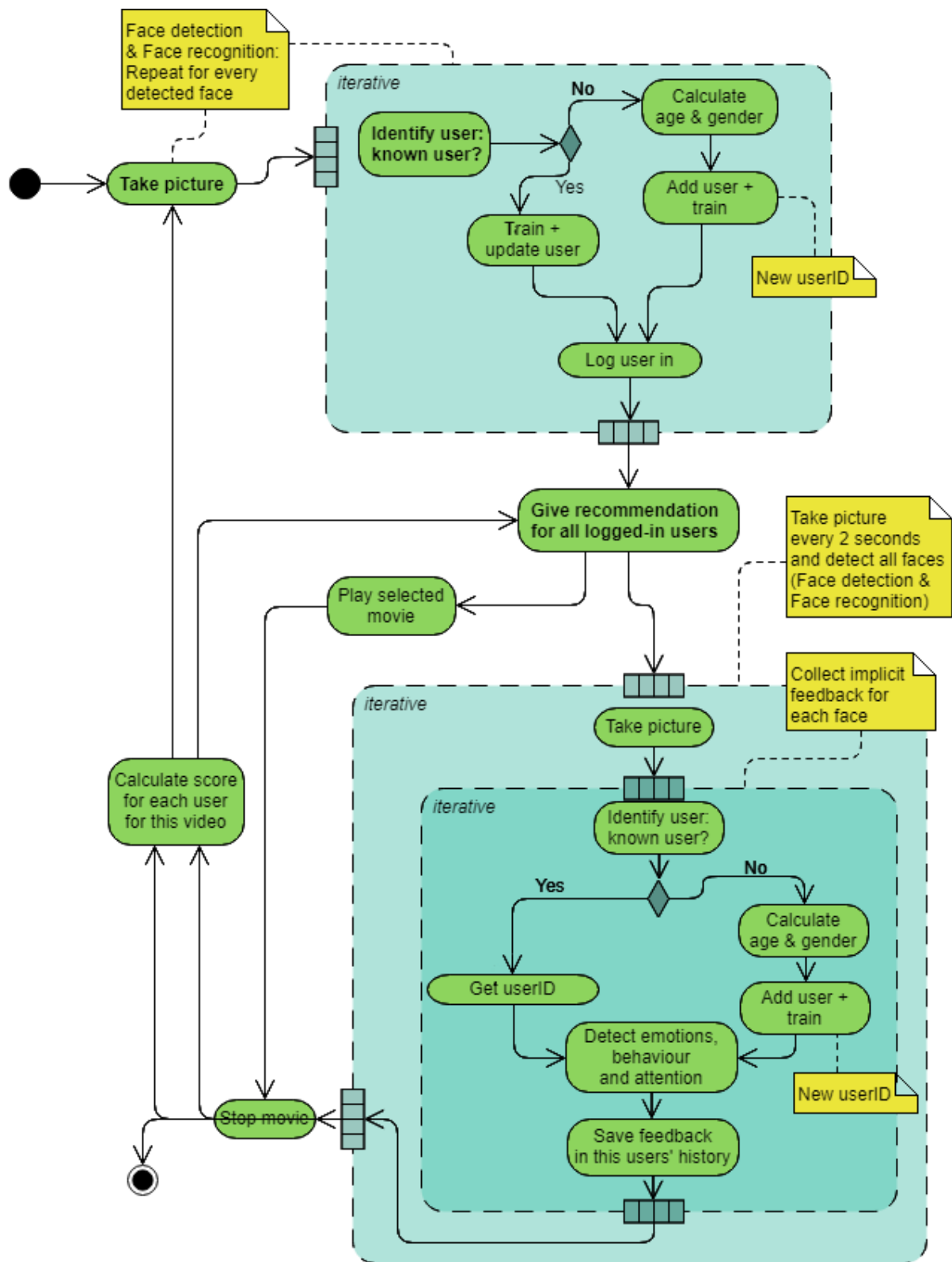
Figuur C.3: Vergelijking van de emotiedetectie bij de drie services toegepast op de middelste foto uit elke fotoreeks (deel 1)



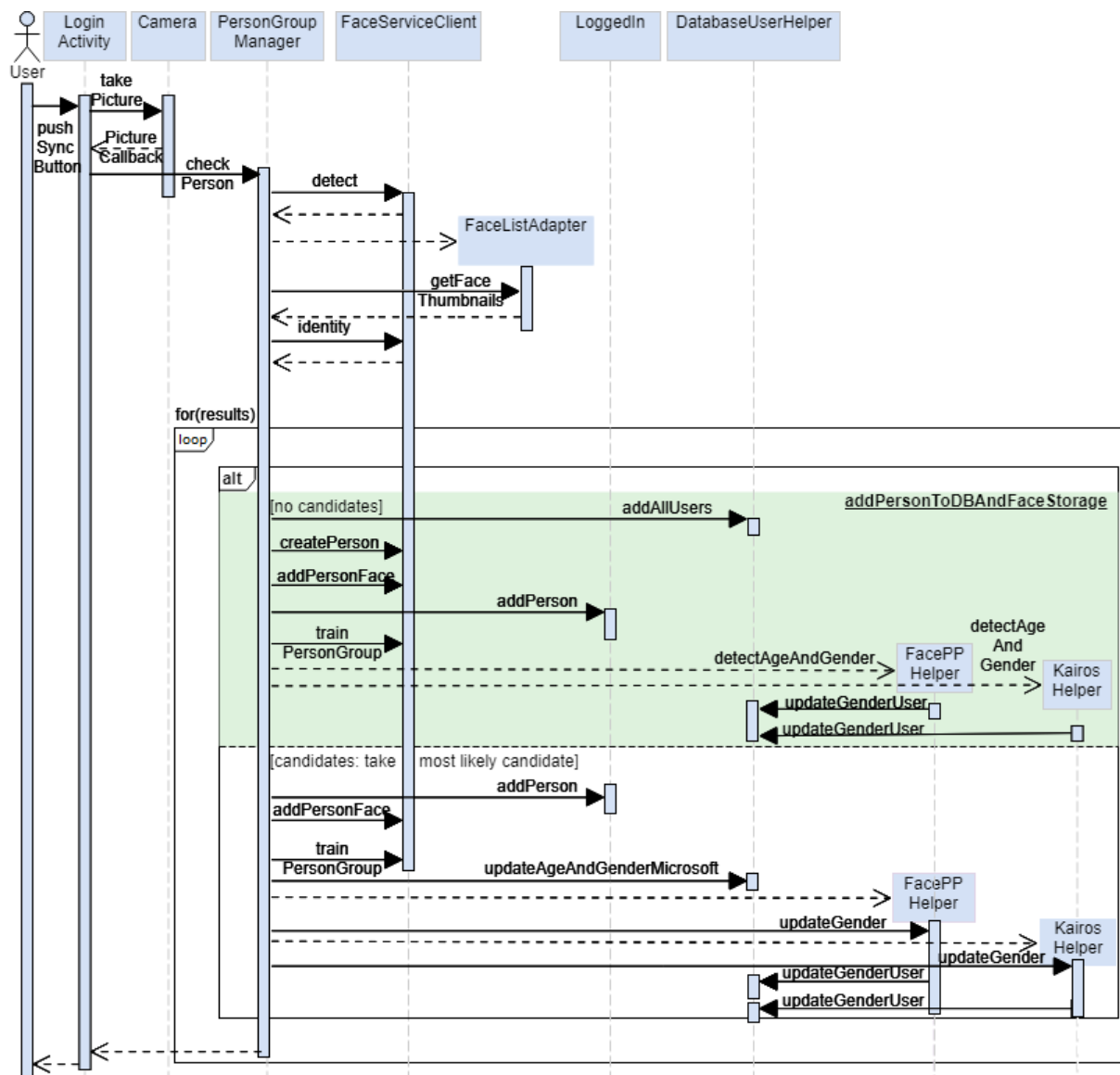
Figuur C.4: Vergelijking van de emotiedetectie bij de drie services toegepast op de middelste foto uit elke fotoreeks (deel 2)



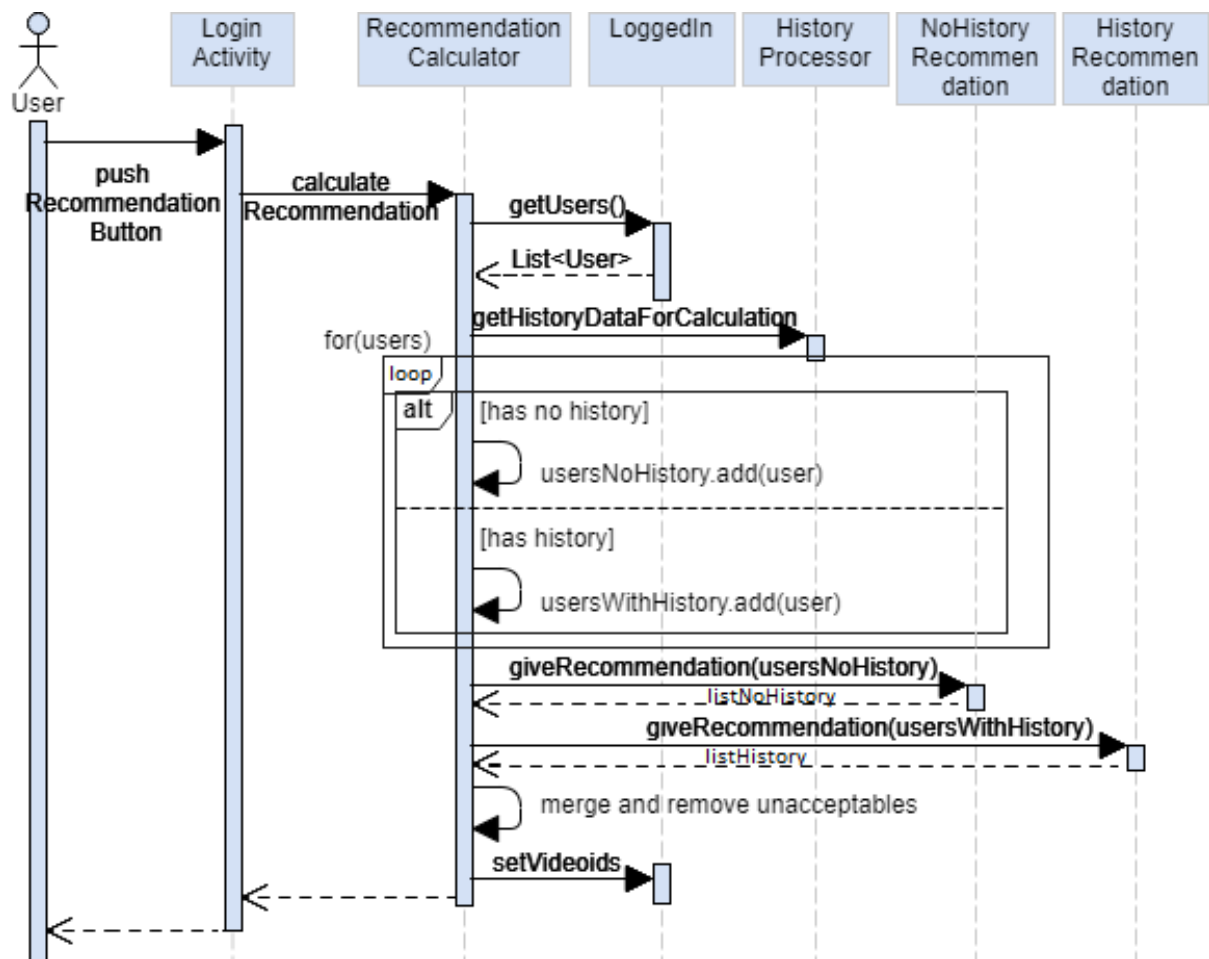
Figuur C.5: Vergelijking van de emotiedetectie bij de drie services toegepast op een hele reeks foto's van één emotie



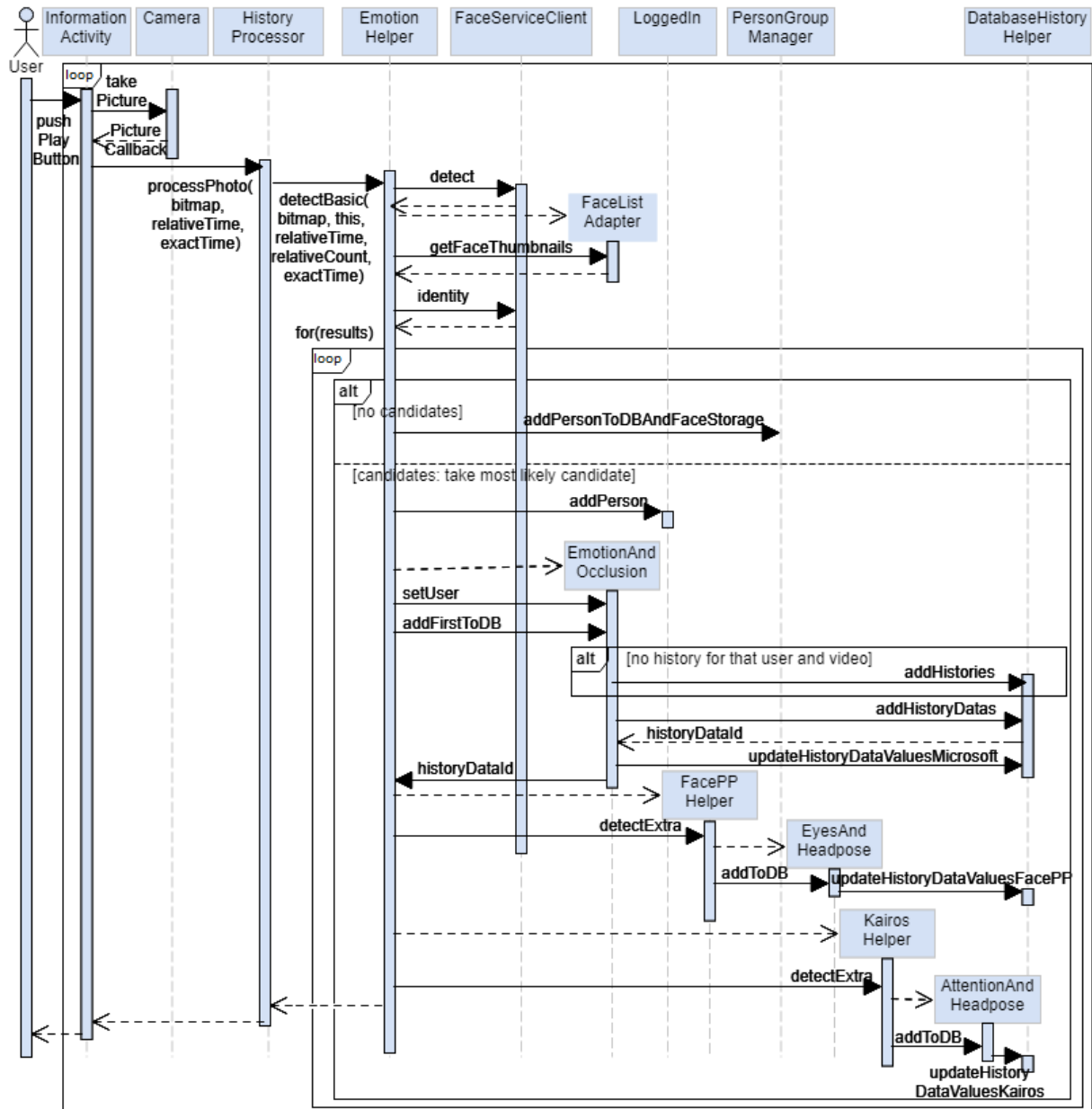
Figuur C.7: Het activiteitendiagram van de applicatie



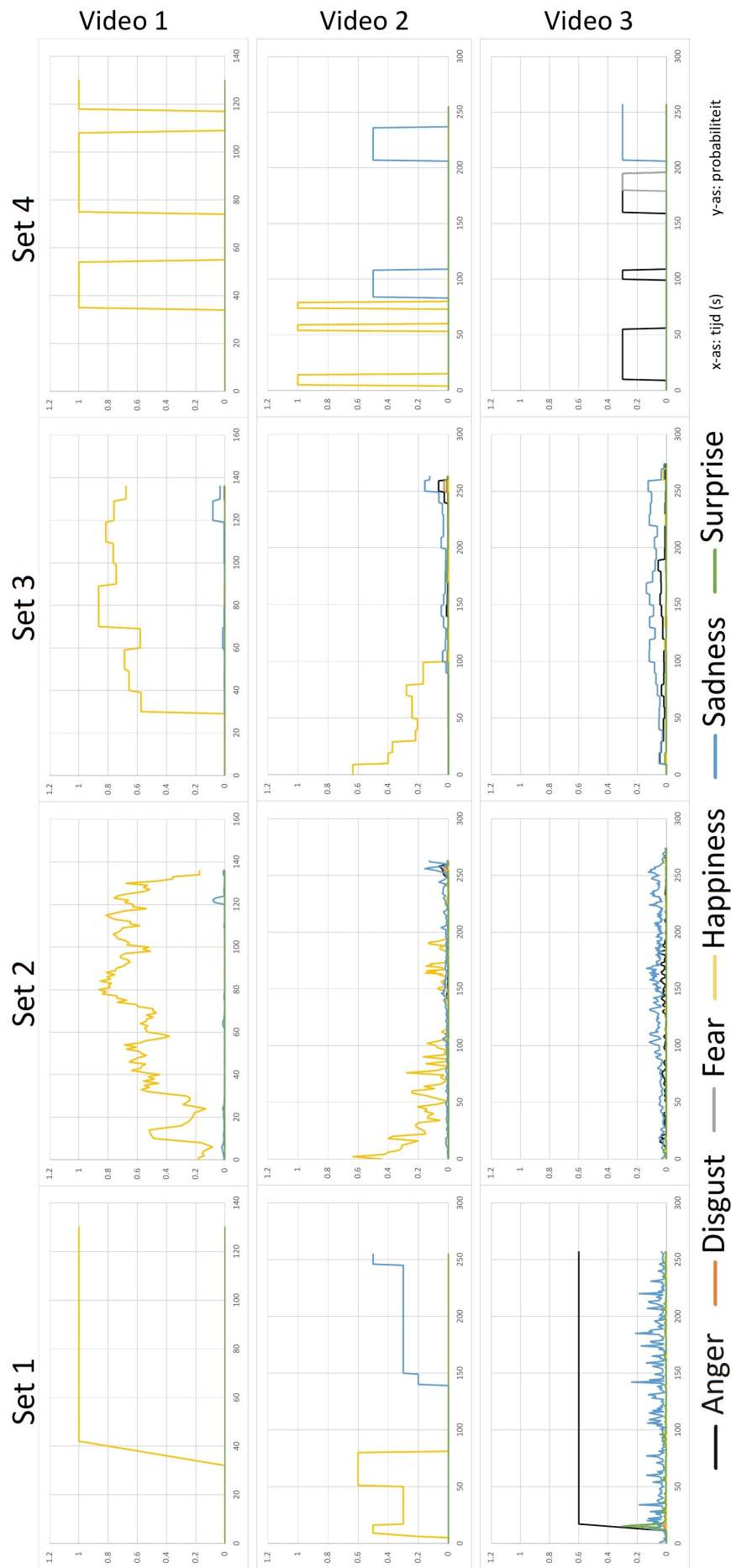
Figuur C.8: Het sequentiediagram voor het inlogproces



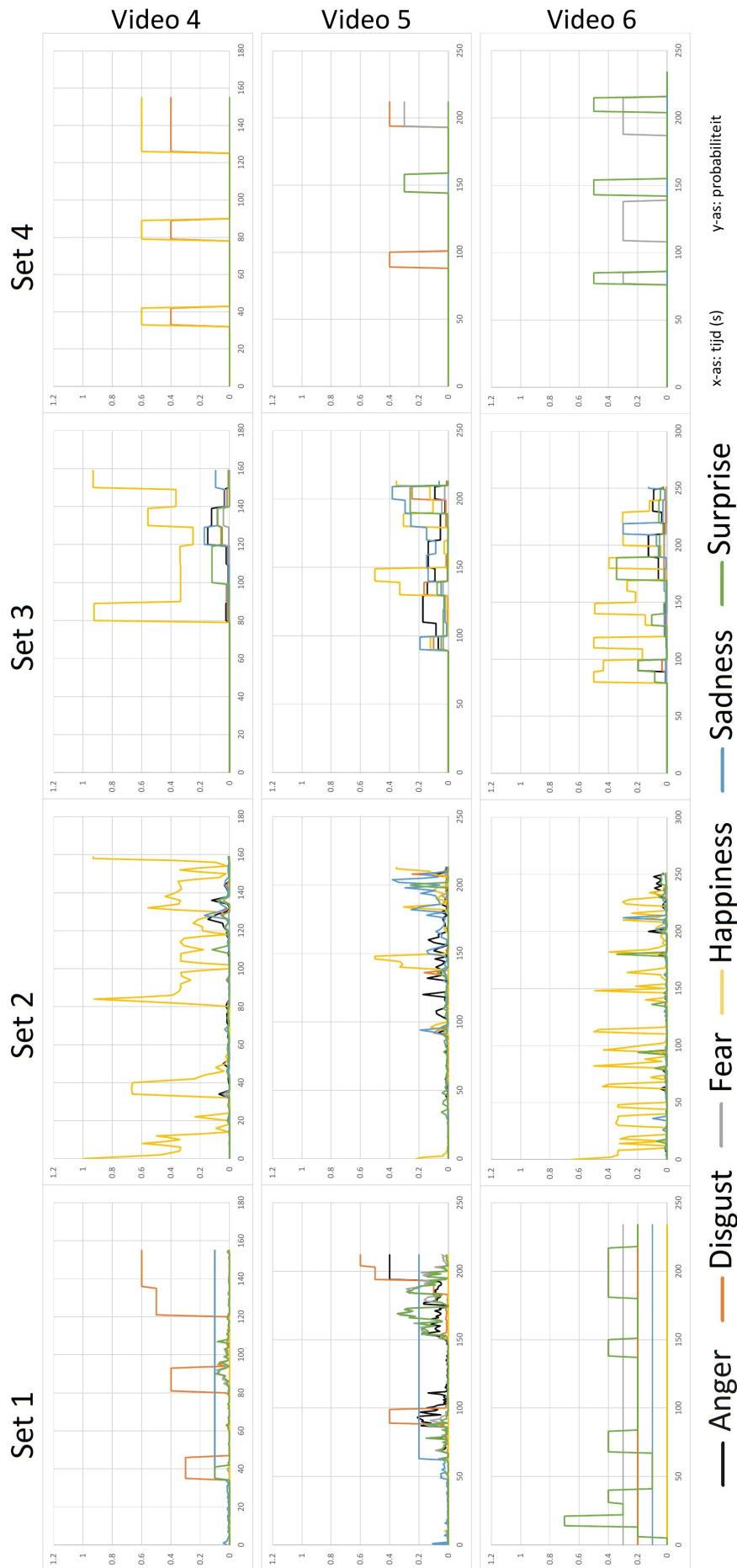
Figuur C.9: Het sequentiediagram voor genereren van de aanbeveling



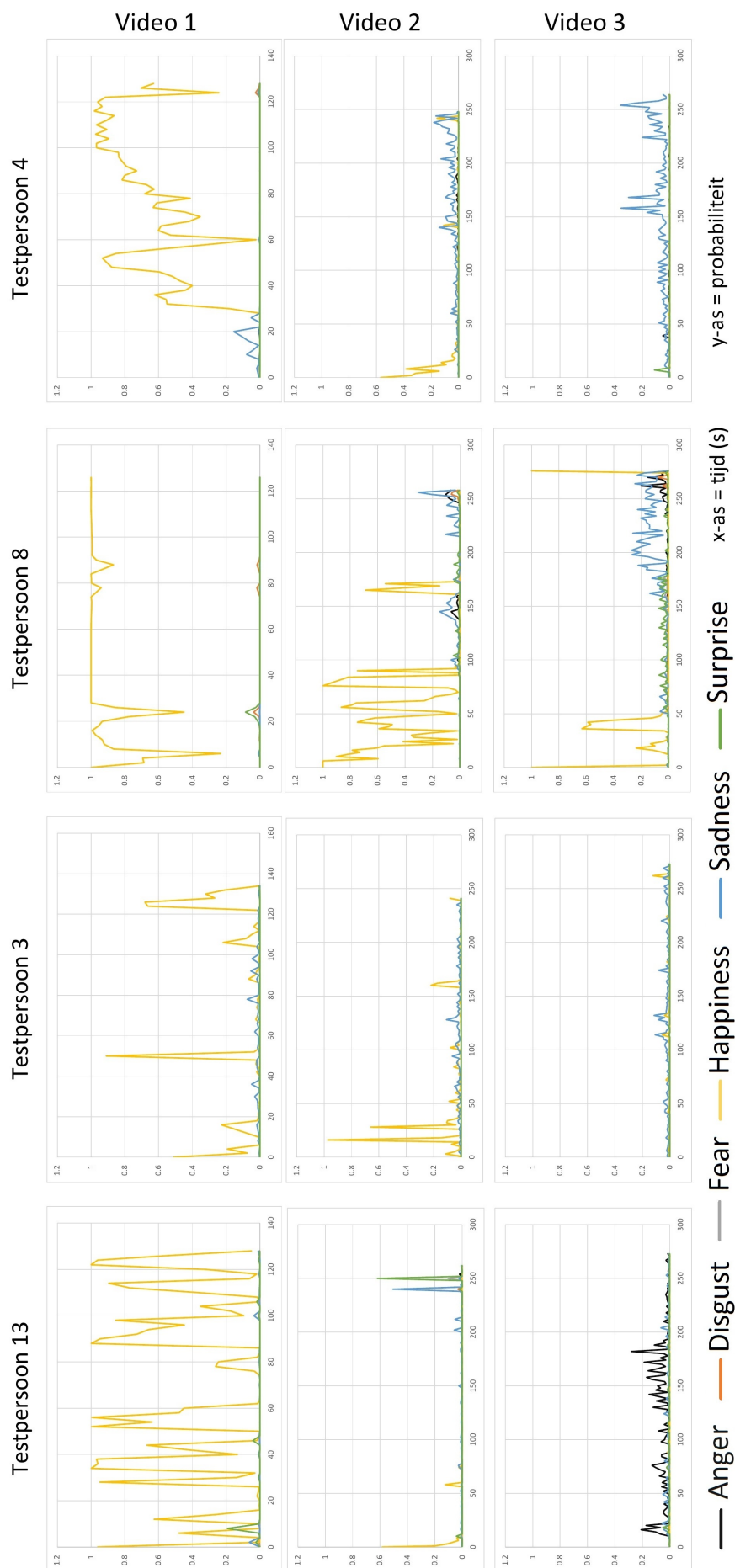
Figuur C.10: Het sequentiediagram voor het verzamelen van de impliciete feedback



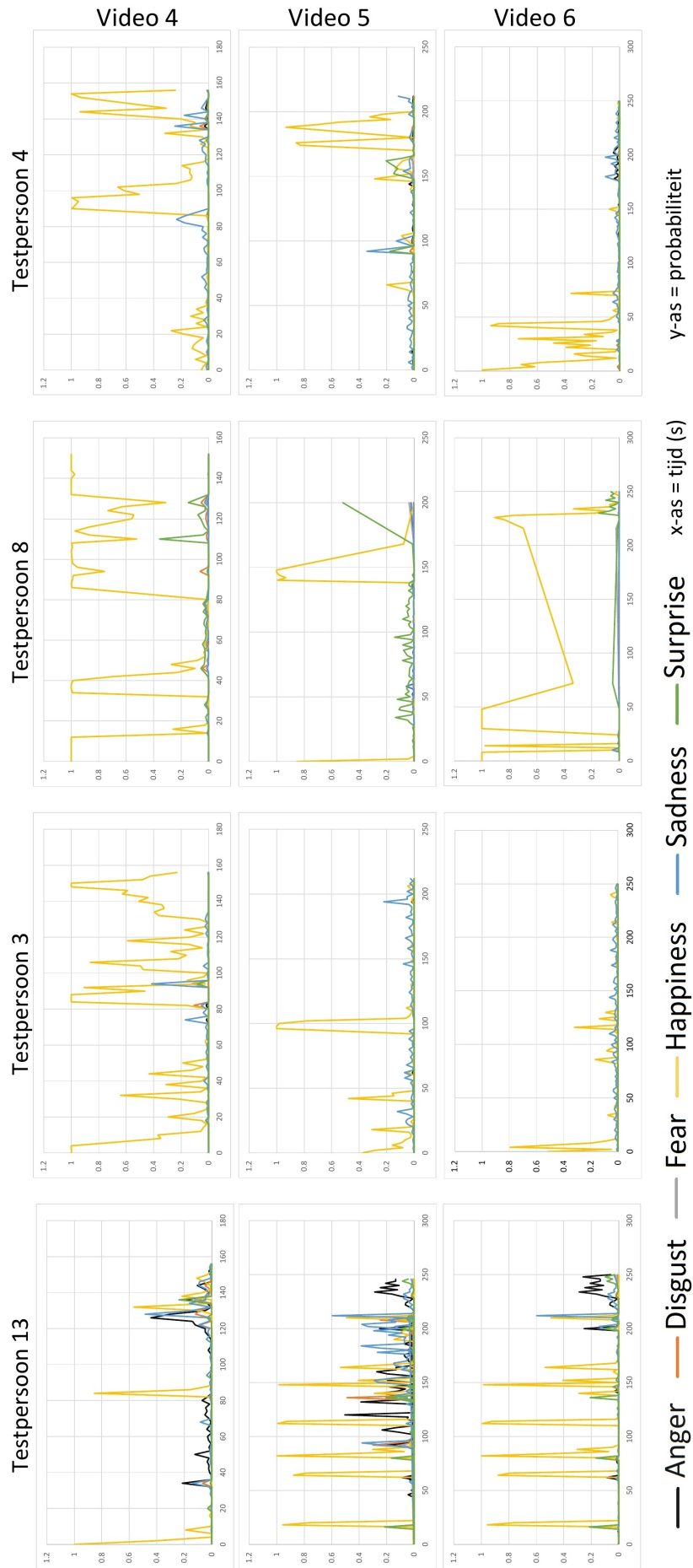
Figuur C.11: De spectra van de vier sets voor video 1, 2 en 3



Figuur C.12: De spectra van de vier sets voor video 4, 5 en 6



Figuur C.13: De emoties van testpersonen 13, 3, 8 en 4 bij video 1, 2 en 3



Figuur C.14: De emoties van testpersonen 13, 3, 8 en 4 bij video 4, 5 en 6