KU LEUVEN

FACULTY OF ENGINEERING TECHNOLOGY

Group T Leuven Campus

EMBARGO

This master's thesis is under embargo until 02-10-2030

Product condition evaluation using multi sensor vision systems

Kavishk KATHAYAT

Jan VAN HUFFEL

Supervisor: Prof. Dr. Ing. MSc. Jef Peeters

Co-supervisor: Yifan Wu

Master's Thesis submitted to obtain the degree of Master of Science in Electromechanical Engineering Technology

Academic year 2024-2025

Preface

Completing this master's thesis has been a shared journey, made possible through the guidance, collaboration, and encouragement of many individuals and organisations.

We are grateful to prof. Jef Peeters, our promotor, whose strategic direction, insightful feedback and advice kept the project on course. Our deepest thanks goes to Yifan Wu, our daily supervisor and co-promotor. Her open-door attitude, patient advice, and practical problem-solving were invaluable at every stage; this work would not exist without her dedication.

We also thank our industrial partner Altrad, with special appreciation to Emiel Tormans and Stefan Knaepkens for providing the crucial dataset and for generously sharing practical insights that grounded the research in real-world requirements.

We are fortunate to have been supported by a wider technical and academic community. Special thanks to Mohammad Mahin Shoaib, Dieter De Marelle, Anthony Coopman, Joren Vandenbosch, Hao Qin, Niels Griffioen, and Terrin Pulikottil for their technical assistance, constructive discussions, and ongoing support that helped us overcome both conceptual and implementation hurdles.

Finally, we extend heartfelt thanks to our families and friends for their unwavering encouragement and understanding. Their support sustained us through long days in the lab and late-night revision sessions.

To everyone who contributed, directly or indirectly, we offer our sincere gratitude.

Summary

Product condition evaluation determines whether a component is in good condition, safe to use, and compliant with quality standards. This process prevents failures, ensures safety, and supports operational efficiency. As a result, condition evaluation technologies are increasingly used across industrial sectors like manufacturing and construction, where reliability and precision are essential.

This thesis investigates advanced multi-sensor vision systems that combine 3D, RGB, and near-infrared (NIR) imaging to automate the inspection of industrial components. Two case studies were conducted: recycled e-bike battery housings and scaffolding pipes. These contrasting use cases reflect structured electronics in controlled recycling settings versus safety-critical infrastructure in construction environments. The first case established a baseline in a predictable domain, while the second tested system robustness under harsher, less structured conditions.

For e-bike battery inspection, the growing need for sustainable recycling—given over 200,000 tonnes of lithium-ion batteries expected to be processed annually in Europe by 2030—makes manual inspection increasingly unscalable. To address this, an automated pipeline was developed using eight multi-view RGB-D scans per module. Background removal and point cloud registration were performed using deep learning and classical geometry, achieving a root mean square error (RMSE) of 0.0384 mm in 0.124 seconds. YOLOv11 was trained to detect missing screws with a mAP@0.5 of 0.963. A complementary anomaly scoring algorithm supports downstream steps such as robotic reassembly.

The pipeline was then extended to scaffolding pipe inspection—representing a less structured, high-risk scenario. The global construction sector is projected to reach $17.05 trillion by 2025, with 65% of workers operating on scaffolding and fall-related incidents comprising 38.4% of fatalities. Manual defect detection for issues like rust, bent pipes, or mortar stains is slow and error-prone. A conveyor-mounted setup was built using calibrated RGB+NIR and 3D line-scan cameras with halogen lighting and a custom GUI interface. Sixteen pipes were rotated and scanned, yielding 64 RGB and 64 NIR images. These were expanded to 1455 training samples using an adapted Slice Aided Hyper Inference (SAHI) algorithm.

Evaluation showed the RGB-only YOLOv11 model achieved a mAP@0.5 of 0.970, while the RGB+NIR version slightly improved to 0.982—demonstrating that multi-modal fusion enhances detection of subtle defects such as mortar stains. However, when detections were passed to SAM2 for segmentation, over-segmentation limited further gains.

This work lays the foundation for intelligent, scalable inspection systems adaptable to different domains. One critical question explored is whether it is necessary to invest heavily in training task-specific models, or whether comparable results can be achieved using zero-shot techniques like open-world object detection (OWOD), which eliminate the need for retraining. This thesis examines that question while demonstrating a modular, transferable framework for real-world inspection challenges across diverse use cases.

Keywords: Automated registration, optimized illumination, multimodal defect detection and segmentation, Zero-shot object detection, YOLOv11, Segment Anything Model, Industrial vision inspection

TABLE OF CONTENTS

Preface ..... **i**

Summary ..... **ii**

List of Figures ..... **ix**

List of tables ..... **xi**

List of Abbreviations ..... **xiii**

1 Introduction **1**

1.1 Motivation *1*

1.2 Introduction *3*

1.2.1 Case-study instantiation: datasets and branch utilisation 3

1.3 Research Question *5*

1.4 Contributions *7*

2 Literature Study **9**

2.1 Introduction to Machine Vision for Industrial Inspection *9*

2.1.1 Image Acquisition 10

2.2 Foundational Concepts in Computer Vision Tasks *11*

2.2.1 Image Classification 11

2.2.2 Object Detection 12

2.2.3 Image Segmentation 12

2.3 Types of Image Segmentation *13*

2.3.1 Semantic Segmentation 13

2.3.2 Instance Segmentation 14

2.3.3 Panoptic Segmentation 14

2.4 Traditional Point Cloud Processing for Geometric Analysis *15*

2.4.1 Stages 15

2.4.2 Optimization-Based Registration and Comparison Techniques 16

2.4.3 Other Traditional 3D Techniques 18

2.4.4 Limitations 18
2.5 Traditional Image Segmentation Methods (Pre-Deep Learning) 19
2.5.1 Intensity-Based Methods 19
2.5.2 Edge-Based Methods 20
2.5.3 Region-Based Methods 20
2.5.4 Other Approaches 21
2.5.5 Limitations 21
2.6 Deep Learning for Image Segmentation: The Convolutional Era 23
2.6.1 Early CNN Approaches 23
2.6.2 Encoder-Decoder Architectures 24
2.6.3 Enhancing Context and Resolution 24
2.6.4 Instance Segmentation Advances 25
2.6.5 The YOLO Framework for Real-Time Object Detection 26
2.7 Deep Learning for Image Segmentation: Transformer Revolution ... 28
2.7.1 Vision Transformer (ViT) 28
2.7.2 Adapting Transformers for Segmentation 29
2.7.3 Hierarchical and Efficient Vision Transformers 30
2.8 Foundation Models and Zero-Shot Segmentation 31
2.8.1 The Concept of Foundation Models in Vision 31
2.8.2 Self-Supervised Learning (SSL) 31
2.8.3 Vision-Language Models (VLMs) 32
2.8.4 Open-World Recognition and Detection (OWOD) 33
2.8.5 The Segment Anything Model (SAM) 34
2.9 Leveraging Multi-Modal Data (RGB + NIR)..... 37
2.9.1 Motivation behind using multi-modal data 37
2.9.2 Multi-Modal Fusion Strategies 38
2.10 Evaluation Metrics for Detection and Segmentation 39
2.10.1 3-D Geometry Metrics 39

2.10.2 Pixel-Level Metrics 40
2.10.3 Object/Instance-Level Metrics..... 40
2.11 Summary and Research Gap 42
3 Materials & Methods 43
3.1 Imaging & Acquisition Hardware 43
3.1.1 Hardware setup capturing e-bike battery..... 43
3.1.2 Hardware setup capturing scaffolding pipe..... 47
3.2 Datasets & Capture Pipelines..... 51
3.2.1 E-bike battery capture pipeline..... 51
3.2.2 E-bike battery capture examples..... 52
3.2.3 Pipe capture pipeline..... 53
3.2.4 Pipe capture examples..... 56
3.3 Overview..... 57
3.3.1 Proposed hybrid pipeline: architecture and guiding principles 57
3.4 Pre-processing & Data Augmentation 62
3.4.1 Background Removal & Classical Augmentations..... 62
3.4.2 Four-Channel Image Slicing (Custom SAHI Adaptation) 62
3.5 Point-Cloud Registration & 3-D Analysis (E-bike battery) 64
3.5.1 End to end point cloud registration..... 64
3.5.2 Anomaly Map..... 66
3.6 OWOD + SAM Zero-Shot Detector..... 68
3.6.1 OWLv2 configuration and text-prompts..... 68
3.6.2 Bounding-box to SAM-prompt conversion..... 68
3.6.3 Post-processing of SAM masks..... 68
3.7 Supervised Trained Detector - YOLO Fine-Tuned..... 70
3.7.1 Motivation Reiteration 70
3.7.2 Dataset Splits..... 70
3.7.3 Training recipe e-bike battery..... 70

3.8 Multichannel Fusion Implementation 72
3.8.1 Early Fusion Layer Modification 72
3.8.2 Dataset Configurations 72
3.8.3 Training Protocol Parity 73
3.9 Design Decisions & Rationale 74
3.9.1 Open-World Detection 74
3.9.2 Supervised Detector Fallback 75
3.9.3 Segmentation Engine 75
3.9.4 Sensor-Fusion Strategy 76
3.9.5 Annotation & Labelling Environment 76
3.9.6 Hardware and Performance Constraints 76
3.9.7 Consolidated Overview Table 78
3.10 Visualisation & Operator Feedback GUI 79
3.10.1 Software Architecture and Data Flow 80
3.10.2 Overlay Rendering and Visual Cues 80
3.10.3 Interactive Threshold Adjustment 80
3.10.4 Two-Dimensional Bounding-Box Detection 81
3.10.5 Two-Dimensional Bounding-Box Detection 81
3.10.6 Hardware and Performance Constraints 81
3.11 Evaluation Protocol 83
3.11.1 Two-Dimensional Bounding-Box Detection 83
3.11.2 Segmentation Masks Generated by SAM 83
3.11.3 Operator-Level Classification on Scaffolding Pipes 84
3.11.4 Three-Dimensional Anomaly Detection on Bike Batteries .. 84
3.11.5 Hardware used for training and inference 84
4 Results 85
4.1 3-D Point-Cloud Registration & Anomaly Scoring: E-Bike Battery Dataset 85
4.1.1 3-D Point-Cloud Registration 85

4.1.2 Anomaly map 87
4.2 2-D Detection & Segmentation — E-Bike Battery Dataset 89
4.2.1 Zero-Shot Baseline — OWLv2 + SAM 89
4.2.2 Supervised Baseline — YOLOv11 (RGB) + SAM 92
4.2.3 Multichannel Variant — YOLOv11 (RGB + Depth) + SAM 94
4.3 2-D Detection & Segmentation — Scaffolding-Pipe Dataset 96
4.3.1 Zero-Shot Baseline — OWLv2 + SAM 96
4.3.2 Supervised Baseline — YOLOv11 (RGB) + SAM 98
4.3.3 Multispectral Variant — YOLOv11 (RGB + NIR) + SAM 100
4.4 Discussion 102
4.4.1 3D combination of deep learning and traditional methods 102
4.4.2 OWOD vs Supervised models 103
4.4.3 RGB vs RGBD 104
4.4.4 RGB vs RGB NIR 105
4.4.5 Preliminary Validation on Crack-Type Defects 106
4.5 Future work 108
5 Conclusion 109
References 110
Appendix A: Additional Literature image acquisition 122
Appendix B: Technical Files of lights and cameras 127
Appendix C: Conveyor system setup 136

List of Figures

Figure 2-a: Three core formulations in computer vision (*Image Segmentation Detailed Overview [Updated 2024]*, n.d.) ----------11

Figure 2-b: Types of image segmentation (*Image Segmentation*, n.d.) ----------13

Figure 2-c: Iterative closest point protocol (Jaykumaran, 2025)----------16

Figure 2-: Otsu thresholding (*3.3.9.7. Otsu Thresholding — Scipy Lecture Notes*, n.d.) ------19

Figure 2-: Sobel edge detection (*Figure 10*, n.d.)----------20

Figure 2-: Region (seed) Growing Segmentation (Alaa, n.d.)----------21

Figure 2-: FCN workflow ('(PDF) Comparison of Fully Convolutional Networks (FCN) and U-Net for Road Segmentation from High Resolution Imageries', 2025)----------23

Figure 2-: U-Net Architecture (*U-Net - an Overview | ScienceDirect Topics*, n.d.)----------24

Figure 2-: Atrous spatial pyramid pooling ('(PDF) A Deep Neural Network for Oil Spill Semantic Segmentation in Sar Images', 2025)----------25

Figure 2-: PSPNet workflow (Zhao et al., 2017b) ----------25

Figure 2-: The Mask-RCNN framework (He et al., 2018) ----------26

Figure 2-: Vision Transformer architecture (Dosovitskiy et al., 2021b) ----------29

Figure 2-: SETR workflow (Zheng et al., 2021b) ----------30

Figure 2-: MAE workflow (He et al., 2021)----------32

Figure 2-: SAM workflow (Ravi et al., 2024b) ----------34

Figure 2-: Examples of RGB + NIR images and annotations in the dataset (J. Luo et al., 2025) 37

Figure 3-a: Overview of pipeline starting from image acquisition to software pipeline ---------43

Figure 3-b: e-bike battery dataset setup with soft box ----------44

Figure 3-c: Mech-Eye Pro-S: Field of view (*PRO S-GL and PRO M-GL*, n.d.)----------44

Figure 3-d: Mech-Eye LSR-S: Field of view (*LSR S-GL*, n.d.) ----------45

Figure 3-e: (a) SW-4000Q-10GE 4-CMOS prism-based line scan camera (b) C5-2040cs 3D sensor ----------47

Figure 3-f: Pipe image capturing setup----------48

Figure 3-g: RGB+NIR camera spectral response----------49

Figure 3-h: Graphical representation of capture viewpoints for e-bike battery dataset (MATHEW & CHHABRA, n.d.)----------51

Figure 3-i: (a) RGB images of various e-bike batteries (b) corresponding single channel depth images ----------52

Figure 3-j: Pipe dataset capturing workflow ----------53

Figure 3-k: Pipe dataset creator GUI ----------54

Figure 3-l presents high-resolution images (to ~700 × 5000 pixels) of captured scaffolding pipes. Pipes (a) and (b) are clean, while (c) and (d) exhibit substantial mortar residue along their length. ----------56

Figure 3-m: Branch A of the developed software pipeline ----------58
Figure 3-n: Branch B of the developed software pipeline ----------59
Figure 3-o: (a) Raw RGB image of e-bike battery (b) Filtered RGB image using DIS (c) Binary mask ----------65
Figure 3-p: Real time pipe defect detection system GUI ----------79
Figure 4-a: Comparison of ICP methods: RMSE and Runtime ----------85
Figure 4-b: Final registered e-bike battery----------86
Figure 4-c: Anomaly map highlighting 6 missing screws in e-bike battery ----------87
Figure 4-d: Missing screws in e-bike battery close up ----------88
Figure 4-e: PR curve for OWOD e-bike battery pipeline performance ----------89
Figure 4-f: e-bike battery correctly mapped examples with legend ----------91
Figure 4-g: e-bike battery with model errors ----------91
Figure 4-h: PR curve for pipe OWOD pipeline inference----------96
Figure 4-i: AP at different IoU thresholds for pipe----------97
Figure 4-j: Ground truth vs predicted result for YOLOv11 model trained on RGB data------ 100
Figure 4-k: Performance comparison of YOLOv11 with RGB and RGB-D inputs for e-bike battery case---------- 105
Figure 4-l: Performance comparison of YOLOv11 with RGB and RGB+NIR inputs for pipe case ---------- 106
Figure 4-m: Extra analysis for e-bike battery case with multiple defects ---------- 107

List of tables

Table 3-1: Summary hardware setup – e-bike battery 46
Table 3-2: Training hyperparameters 71
Table 3-3: Design Decisions overview 78
Table 4-1: Comparison and benchmarking of different ICP protocols for 3D registration pipeline 85
Table 4-2: YOLOv11 Results Screw – RGB 93
Table 4-3: YOLOv11 + SAM Results Screw - RGB 93
Table 4-4: YOLOv11 Results Screw – RGBD 94
Table 4-5: YOLOv11 + SAM Results Screw - RGBD 94
Table 4-6: YOLOv11 Results Mortar - RGB 99
Table 4-7: YOLOv11 + SAM Results Mortar - RGB 99
Table 4-8: YOLOv11 Results Mortar – RGB NIR 100
Table 4-9: YOLOv11 + SAM Results Mortar – RGB NIR 101

List of Abbreviations

Acronym	Definition
2D	Two-Dimensional
3D	Three-Dimensional
ABS	Acrylonitrile Butadiene Styrene
AI	Artificial Intelligence
CLIP	Contrastive Language-Image Pretraining
CNN	Convolutional Neural Network
COCO	Common Objects in Context
CSP	Cross Stage Partial
DETR	Detection Transformer
DGCNN	Dynamic Graph CNN
DINO	DEtection with TRansformers
DLP	Deep Learning Pipeline
DOF	Degrees of Freedom
DCP	Deep Closest Point
FP	False Positive
FGR	Fast Global Registration
FLOPs	Floating Point Operations Per Second
FN	False Negative
FPS	Frames Per Second
FPH	Fast Point Feature Histograms
FCN	Fully Convolutional Network
GFLOPs	Giga Floating Point Operations
GUI	Graphical User Interface
HSV	Hue Saturation Value

ICP	Iterative Closest Point
IoU	Intersection over Union
IR	Infrared
LED	Light Emitting Diode
LLM	Large Language Model
LR	Learning Rate
LSR	Line Structured Recording
MAE	Masked Autoencoder
ML	Machine Learning
mAP	Mean Average Precision
MS-COCO	Microsoft Common Objects in Context
MVS	Machine Vision System
NDT	Non-Destructive Testing
NIR	Near-Infrared
NMS	Non-Maximum Suppression
OWL-ViT	Open-World Learning Vision Transformer
OWOD	Open-World Object Detection
PA	Pixel Accuracy
PLY	Polygon File Format (for point clouds)
PNG	Portable Network Graphics
PSA	Polarized Self Attention
RGB	Red Green Blue
RGB-D	Red Green Blue + Depth
RMSE	Root Mean Square Error
RANSAC	Random Sample Consensus
SAM	Segment Anything Model
SAM2.1	Segment Anything Model version 2.1
SAHI	Slicing Aided Hyper Inference

SDK	Software Development Kit
SSL	Self-Supervised Learning
SWIN	Shifted Window Transformer
TP	True Positive
TIF	Tagged Image File Format
TXT	Plain Text File
ViT	Vision Transformer
VLM	Vision-Language Model
YOLO	You Only Look Once

1 INTRODUCTION

1.1 Motivation

Product condition refers to the measurable state of an item's physical, functional, and aesthetic integrity, including factors such as wear, damage, corrosion, and structural degradation. Accurate condition assessment ensures that components continue to perform as intended, meet safety requirements, and remain economically viable for reuse or rental.

The need for reliable condition assessment becomes critical when products are subjected to repeated use in harsh environments. Without precise evaluation, undetected defects can lead to operational failures, increased maintenance costs, and severe safety risks. This is particularly true in industries where equipment longevity and user safety are paramount.

The construction industry presents a particularly representative case for condition assessment challenges. Driven by cost efficiency and project flexibility considerations, this sector has widely adopted rental models for heavy equipment such as scaffolding pipes. This shift introduces the critical challenge of accurately assessing and managing the condition of scaffolding components between rentals. Scaffolding pipes, subjected to rigorous use in construction environments, experience significant wear and tear, corrosion, and structural degradation, which pose safety risks if undetected or improperly evaluated. Consequently, ensuring the reliability and safety of these components through precise and objective condition assessment is paramount.

To validate the universality of the assessment methodology, this study selects two distinct categories of objects—construction equipment and consumer products as comparative cases. Currently, scaffolding pipe condition evaluation predominantly relies on manual visual inspections by trained personnel. This subjective approach introduces several limitations, including inconsistent quality assessments, human error, and inefficiencies resulting from labour-intensive processes (Construction Industry Institute, 2023). Besides, the e-bike battery housing serves as a representative consumer product case, where plastic components are prone to defects such as cracks and missing screws (parts) during outdoor use. Although their failure modes differ from construction materials, these components similarly require high-precision defect detection—making them an ideal testbed for verifying the adaptability of the assessment method across different materials and defect types.Recent advancements in sensor technology, non-destructive testing (NDT), and machine vision systems present opportunities to significantly enhance the accuracy, consistency, and efficiency of condition assessments. These technological solutions have successfully demonstrated their potential in

related fields, such as defect detection in manufacturing and infrastructure monitoring, yet their application within the scaffolding rental and e-bike battery domain remains underexplored (Chen et al., 2021).

To bridge this gap, this thesis introduces a hybrid inspection pipeline and evaluates it through two industrial case studies. Each case study is designed to activate different functional aspects of the methodology, illustrating both its versatility and its practical relevance.

The first case study revisits e-bike battery housings, leveraging a high-resolution RGB-D dataset collected in an earlier project. This scenario enables a full demonstration of the pipeline's capabilities, including 3D point-cloud alignment for geometric inspection and 2D defect localisation using machine vision techniques.

The second case study focuses on scaffolding pipes, directly addressing the construction rental context. Here, the primary concerns are surface-level defects such as corrosion and mortar stains, aligning closely with real-world inspection needs in the scaffolding industry.

The inclusion of two distinct case studies serves a dual purpose:

1. Demonstrating versatility – showcasing how the proposed pipeline adapts to different object types, defect modalities, and inspection goals.
2. Highlighting trade-offs – illustrating how varying sensor configurations (e.g., RGB vs. RGB-D) impact detection accuracy, performance, and feasibility in industrial settings.

Together, these case studies provide a comprehensive validation of the scope of the thesis for laying down a foundational concept of product condition evaluation.

1.2 Introduction

1.2.1 Case-study instantiation: datasets and branch utilisation

To ground the hybrid pipeline in concrete industrial scenarios, the thesis evaluates it on two different datasets. Each dataset activates a specific subset of the pipeline's functionality, thereby illustrating both the breadth of the framework and the trade-offs that arise when sensor modalities are added or withheld. Detailed acquisition protocols and annotation statistics are deferred to Chapter 3.4; here the intention is simply to map each dataset to the branch or branches it exercises.

1.2.1.1 Bike-battery housings

The first case study uses a high-resolution dataset collected during an earlier master's project. A six-axis robot positions a dual-mode sensor head around each e-bike battery casing, capturing eight canonical viewpoints that jointly provide full surface coverage. For every viewpoint the sensor outputs an RGB image and a registered depth map, yielding a paired RGB-D representation. Because both photometric appearance and geometric integrity are available, the dataset enables a complete demonstration of the pipeline: Branch A performs 3-D point-cloud alignment against a non-defect point cloud of the same item, while Branch B handles 2-D defect localisation and segmentation. Within Branch B a further experiment contrasts a purely RGB-trained YOLOv11 detector with an early-fusion variant that ingests the full RGB-D tensor. For the purpose of this thesis the primary defect of interest is the presence or absence of screws, taken as a proxy for assembly completeness and ingress-protection reliability.

1.2.1.2 Scaffolding pipes

The second case study introduces a dataset collected specifically for the present work. Steel pipes travel beneath a prism-based line-scan camera mounted on a conveyor, generating continuous scroll images that contain both visible-light and near-infra-red channels. Unfortunately, depth capture was postponed, so only the RGB-NIR pair is available. Consequently, the evaluation on this dataset invokes Branch B exclusively: OWL-V2 followed by SAM furnishes a zero-shot baseline, and a YOLOv11 detector fine-tuned on the same imagery offers a supervised comparison. The immediate inspection target is mortar or cement residue on the pipe surface, a fault that impedes coupler tightening and therefore compromises scaffold stability. Rust and dent detection remain on the research agenda but lie outside the annotated scope of this thesis. Branch A is deferred until stable depth hardware can be re-integrated; the software hooks required for a future 3-D extension are, however, already present.

1.2.1.3 Rationale for dual instantiation

Juxtaposing these two case studies serves three analytic purposes. First, it demonstrates the portability of the pipeline across markedly different production contexts: an electronics housing manipulated by a stationary robot versus tubular steel transported on a moving conveyor. The second part tests the prompt able zero-shot detector in an unseen domain—the scaffolding pipes—after initial development on the battery casings, thereby providing a realistic measure of generalisation. Third, it establishes an empirical basis for discussing when geometry-based inspection offers added value over spectral based segmentation alone. The ability to toggle Branch A on or off according to sensor availability is itself a testament to the modularity claimed in § 3.1.2.

1.3 Research Question

The research investigates how to build an automated inspection system that employs multi-modal imaging and machine learning for detecting and segmenting industrial components, particularly scaffolding pipes and e-bike batteries, while prioritizing safety and precision. This study aims to tackle both the technical barriers and operational challenges related to data source alignment and processing, which are essential for achieving precise visual assessment. A central question guiding this effort is:

Can traditional computer vision techniques and deep learning methods be effectively combined to develop a robust registration system for processing diverse image types, including 3D and RGB data?

This question explores whether integrating the geometric precision of conventional approaches with the learning capabilities of neural networks through a hybrid method can lead to the accurate spatial alignment required for downstream tasks such as defect classification and segmentation.

The research also analyses how multi-modal data fusion influences the overall performance of machine learning models. By integrating RGB data with Near-Infrared (NIR) and depth (RGB-D) imaging, the system aims to capture richer surface information that improves defect visibility across various material types and lighting conditions. This leads to the second major research question:

Can the fusion of multi-modal inputs, such as RGB+NIR or RGB-D, significantly enhance the performance of machine learning models in both defect detection and segmentation tasks?

The study evaluates whether such multi-source data inputs result in more accurate and reliable model outputs, particularly when identifying complex and subtle defects such as rust, bent pipes, or mortar stains.

Finally, the research considers the cost and complexity associated with training highly specialized models, and whether more scalable alternatives could offer similar performance. To this end, it poses the third key question:

Is it necessary to invest significant time, resources, and effort into building and training task-specific models, or can comparable performance be achieved using emerging zero-shot techniques such as open-world object detection (OWOD), which eliminate the need for custom training?

This question evaluates the practicality and efficiency of new, general-purpose detection methods when applied in real-world industrial inspection scenarios, where reducing development time and operational cost is critical.

By investigating these three interconnected research questions, the study aims to develop intelligent, multi-modal inspection systems capable of both detecting and segmenting defects.

1.4 Contributions

Industrial inspection systems must be modular enough to accept new sensors or algorithms yet fast enough for shop-floor decision-making. This thesis delivers a generic framework that meets both aims: a single acquisition, processing and visualisation stack that can be re-configured for very different products and sensing modalities. The bike-battery (robot-mounted RGB-D) and scaffolding-pipe (conveyor-based RGB + NIR) case studies serve only as proofs of concept; the underlying architecture is agnostic to object shape, size or spectral band. The work contributes five main deliverables:

1. **Modular multi-sensor inspection platform**: A hardware–software skeleton was designed and calibrated to accept heterogeneous imagers—ranging from stereo RGB-D rigs on a six-axis robot to RGB + NIR line-scan cameras on a conveyor—while keeping a unified timing, calibration and data-exchange protocol. This makes later swaps (e.g. LiDAR, thermal IR) a matter of configuration rather than redesign.
2. **Custom GUI for high-resolution pipe dataset acquisition**: A real-time operator interface was developed for conveyor-based capture, allowing the user to control speed, exposure, and run duration. The GUI produces audit-ready scroll images in both RGB and NIR, aligned by design.
3. **Multi-channel image slicing tool**: A lightweight adaptation of the SAHI library was developed to slice large multi-channel scroll images (≈ 2 448 × 42 000 px) into 1 455 overlapping crops. The tool preserves channel alignment and prepares the dataset for multi-modal training and evaluation.
4. **Unified real-time processing pipeline with visual feedback**: Zero-shot detection (OWLv2 + SAM 2.1), supervised fallback (YOLOv11 + SAM 2.1) and 3-D ICP-based anomaly mapping are orchestrated by a single controller that streams results to a live GUI, allowing operators to adjust confidence thresholds and view masks in real time.
5. **Systematic experimental comparison of sensing and learning choices**: Extensive tests contrast (i) zero-shot versus supervised detection, (ii) RGB versus RGB + Depth on bike batteries, (iii) RGB versus RGB + NIR on pipes, and (iv) five ICP variants for 3-D alignment. Metrics include mAP, Dice, recall, RMSE and runtime, all reproducible on an RTX 3090 within a 24 GB VRAM budget.

Together these contributions provide a reusable platform, a novel dataset, a complete real-time pipeline and a rigorously benchmarked evidence base for future industrial inspection research.

2 LITERATURE STUDY

2.1 Introduction to Machine Vision for Industrial Inspection

Modern industrial automation depends on machine vision as its essential foundation which enables systems to duplicate human visual inspection abilities for multiple manufacturing operations. The combination of cameras with sensors and lighting and intelligent image processing software enables real-time visual data interpretation from production lines. Machine vision systems (MVS) provide non-contact high-speed inspection with high accuracy and repeatability because it operates independently of human limitations such as fatigue and inconsistency and human errors.

MVS demonstrates its ability to work with multiple materials and surface finishes as well as different product geometries and can be tailored to meet different industrial requirements.

The core pipeline or operational workflow of Machine Vision Systems (MVS) remains consistent across different industrial and non-industrial domains (Golnabi & Asadpour, 2007). The system workflow starts with data collection and ends with decision-making through several critical stages.

1. Image Acquisition: The process begins with capturing visual data. The system acquires digital images of real-world objects through cameras or sensors which convert physical scenes into data for system processing.
2. Image Preprocessing: The unprocessed images show imperfections that need correction. The system applies preprocessing methods which include noise reduction and geometric corrections together with contrast enhancement and colour normalization to improve image quality before moving forward.
3. Feature Extraction: The system identifies and extracts essential features or patterns from the enhanced image which are important for the task. Machine learning algorithms receive these features as input to achieve accurate classification and detection or analysis outcomes.
4. Decision-Making: The system uses extracted features to make decisions about object conformity to desired criteria during its final stage. The system generates feedback through defect counts and object classification and pass/fail results and alerts which enable operators and automated systems to take corrective action when needed.

Machine vision has progressed beyond quality control functions to become a fundamental driver of smart manufacturing as industries adopt Industry 4.0 principles. The combination of

artificial intelligence and machine learning with machine vision enables real-time decision-making and predictive maintenance and continuous process optimization (Çınar et al., 2020). The evolving nature of machine vision demonstrates its expanding role as both an inspection tool and a strategic component in data-driven production systems connected through networks.

The basic operation of machine vision systems stays consistent, but every application requires unique system design because no single design works for all cases. The implementation needs to be customized for each operational context by considering defect size and nature along with environmental factors and processing speed needs and object dimensions and system integration requirements with larger automation systems (Steger et al., 2018). Vision systems require extensive customization throughout their entire pipeline starting from image acquisition through decision-making to fulfil specific requirements of each application.

2.1.1 Image Acquisition

The acquisition of images stands as the essential foundation for any machine vision pipeline. The current emphasis on AI and deep learning techniques does not change the fact that high-quality input images remain essential because poor image quality makes advanced algorithms useless (Steger et al., 2018). The performance of any vision-based system depends directly on the visual data quality because poor image input produces unreliable results regardless of algorithm complexity.

2.1.1.1 Illumination & lighting

Machine vision systems heavily depend on illumination because it determines both the visibility of essential features and the elimination of unwanted image artifacts. The effective use of lighting technology improves both the feature contrast and surface anomaly visibility and minimizes shadow effects and glare and uneven illumination that hide important details (Z. Ren et al., 2022). The development of a reliable inspection system requires complete control over lighting conditions because uncontrolled illumination sources create system variability which decreases inspection accuracy. The detection of small defects in low-contrast situations requires precise optimization of lighting systems. The enhanced visibility of small irregularities through proper lighting allows traditional and AI-based image processing algorithms to identify defects more accurately and consistently (Schraml & Notni, 2024). The fundamental understanding of light behaviour through reflection and refraction and absorption and scattering effects on various materials and surfaces is essential.

2.1.1.2 Fundamentals of Light-Material Interaction

When electromagnetic radiation strikes a material surface, its energy is partitioned among four primary pathways—reflection, absorption, transmission, and scattering—whose proportions

are governed by the material's complex refractive index, surface morphology, and the incident wavelength. Smooth, metallic surfaces favour specular reflection, preserving the angle of incidence and often dominating the return signal, whereas rough or dielectric surfaces exhibit diffuse reflection and volumetric subsurface scattering that blur directional cues yet encode micro-geometric detail. Absorbed photons excite electronic or vibrational states, producing diagnostic spectral signatures; transmitted photons may emerge refracted, phase-shifted, or wavelength-filtered according to Beer–Lambert attenuation. Because different materials modulate these pathways differently at distinct wavelengths, tailoring illumination and sensing beyond the visible band (e.g., near-infrared or ultraviolet) can reveal otherwise hidden contrasts—an asset that multispectral vision systems exploit to boost defect visibility under varied inspection conditions.

2.2 Foundational Concepts in Computer Vision Tasks

Rapid advances in computer vision stem from three core problem formulations that build upon one another in complexity: *image classification*, *object detection*, and *image segmentation*. Clarifying these concepts provides the groundwork for understanding later architectural choices and evaluation strategies.

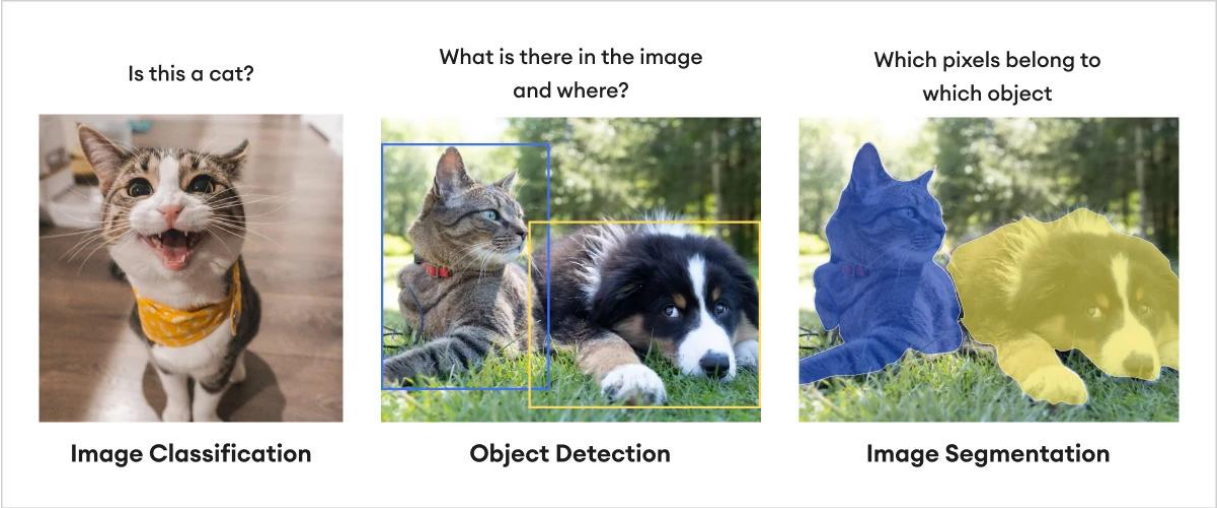


Figure 2-a: Three core formulations in computer vision
(*Image Segmentation Detailed Overview [Updated 2024]*, n.d.)

2.2.1 Image Classification

The process of image classification identifies a single dominant label which represents the main semantic meaning of an image such as "cat," "traffic sign" or "human." The traditional image classification methods depended on manually designed features including gradients and key points. Modern approaches employ deep neural networks that automatically learn hierarchical feature representations from raw pixel data. The evaluation of performance relies

on two main metrics which are top 1 and top 5 accuracy. The top 1 accuracy evaluates the model's ability to predict the correct label when the model shows its highest confidence, but the top 5 accuracy evaluates the model's ability to place the correct label among its top five most confident predictions. Image classification models function as basic components which enable multiple downstream applications including object detection and segmentation and medical diagnostic systems. The learned feature encoders from these models serve as essential components for transfer learning because they allow adaptation to new tasks with minimal available data.

2.2.2 Object Detection

Object detection goes beyond classification by identifying all relevant instances in images while assigning each detected object to its corresponding category. Modern detectors use a single fully trainable pipeline to predict both bounding boxes and class scores which enables both real-time edge device inference and high-precision offline analytics. The performance of object detection systems is typically evaluated through mean Average Precision (mAP) which considers multiple intersection-over-union thresholds across datasets that contain images with diverse scales and aspect ratios and different levels of occlusion. The current research focuses on solving open-set recognition problems and handling long-tailed distributions and domain shift robustness because these factors significantly impact safety-critical deployment scenarios.

2.2.3 Image Segmentation

Image segmentation requires assigning specific categories to individual pixels in images which produces more detailed visual understanding than classification or detection methods. Segmentation produces detailed predictions which show the exact boundaries and positions of objects and regions instead of using bounding boxes or assigning single image labels. Applications that need spatial precision such as medical imaging and autonomous driving and industrial inspection require pixel-level understanding. Modern segmentation models strive to achieve precise boundary detection while maintaining an understanding of the entire image context through hierarchical features and multiscale information.

2.3 Types of Image Segmentation

Image segmentation can be divided into three main types—semantic, instance, and panoptic segmentation. These are briefly outlined below.

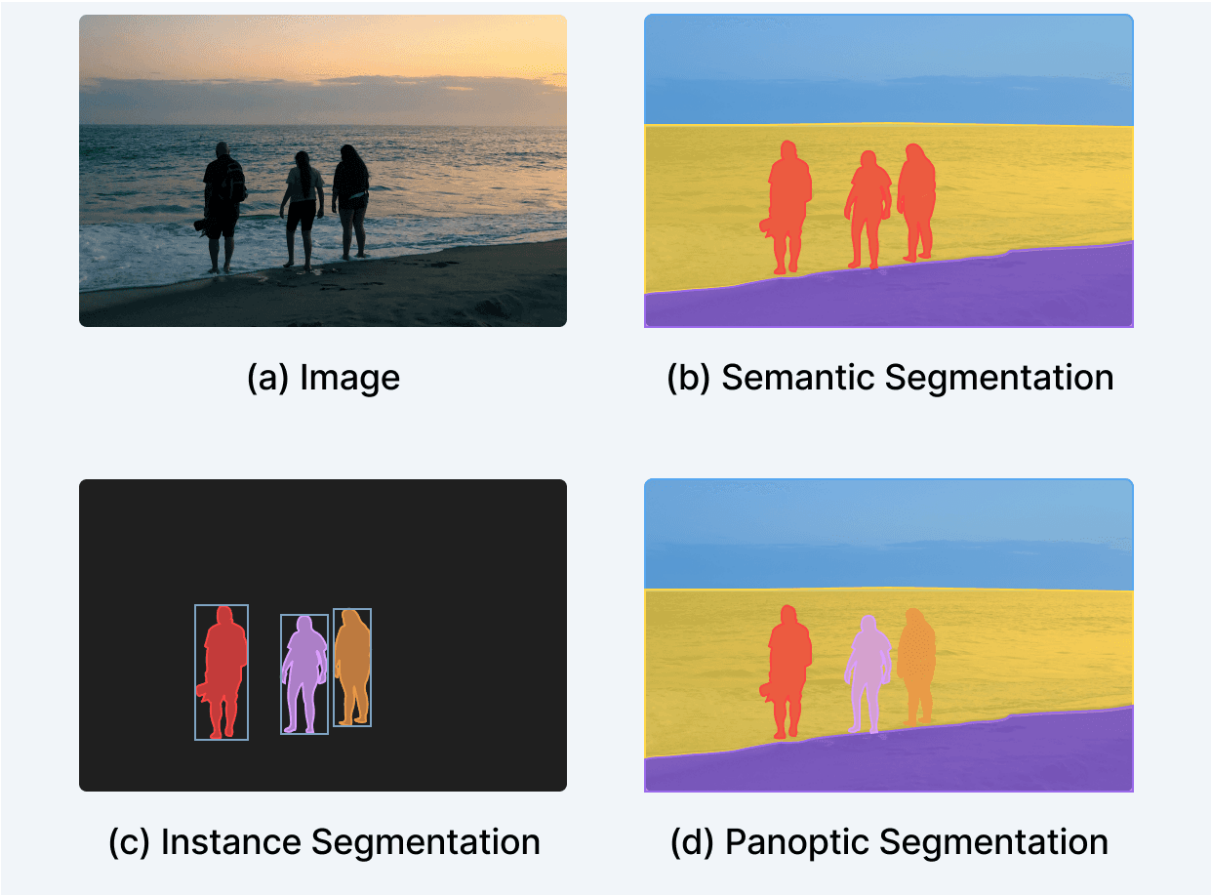


Figure 2-b: Types of image segmentation (*Image Segmentation*, n.d.)

2.3.1 Semantic Segmentation

The process of semantic segmentation assigns one semantic class to each pixel across the entire image. All pixels that belong to the same category receive the same label which means objects of the same type cannot be distinguished from each other. The goal of this process involves generating a detailed categorical map which identifies the specific object type at each pixel location. The output typically consists of a label image that uses predefined class indices. The method delivers robust global context which proves most beneficial for applications requiring spatial class extent over object identification such as drivable area detection and medical organ segmentation. The main weakness of this approach lies in its inability to

distinguish between multiple occurrences of the same class because overlapping or occluded objects become one region.

2.3.2 Instance Segmentation

Instance segmentation extends semantic segmentation by performing both class and object-level separation. The method assigns each pixel both class information and instance identification so it can determine which specific object exists at a particular pixel location. The output consists of either a stack of binary masks or a label image that contains all (class, instance ID) pairs. The ability to distinguish between multiple objects of the same category makes this technique essential for counting, tracking and interaction analysis but the increased representation leads to higher annotation expenses and more complex models. The system ignores amorphous background regions such as sky or road unless they receive explicit inclusion in the label set.

2.3.3 Panoptic Segmentation

Panoptic segmentation combines the previous two tasks by generating one unified map that assigns semantic labels to each pixel and instance identifiers when needed. The system assigns unique (class, instance-ID) pairs to countable “things” including cars and people but uses (class, ID = 0) tuples for uncountable “stuff” categories like grass and sky. The goal is to create a complete scene description which both covers all pixels and maintains object separation.

2.4 Traditional Point Cloud Processing for Geometric Analysis

The detailed geometric surface representations of object surfaces in 3D point clouds make them suitable for detecting defects such as dents, warping, or dimensional deviations in industrial inspection. The traditional point cloud processing methods rely on geometric approaches which have been widely applied for model alignment and structural feature extraction and scanned object comparison against CAD models and ground truth data (Z. Liu et al., 2023).

2.4.1 Stages

The structured pipeline of traditional point cloud processing includes several sequential stages which lead to accurate geometric analysis and defect detection. The process starts with data acquisition followed by preprocessing then registration and surface reconstruction and finally geometric comparison.

1. Data Acquisition: Data acquisition in 3D applications happens through LiDAR and structured light scanners and stereo cameras depending on application requirements and desired resolution levels. Industrial settings mostly use laser scanners because they deliver accurate results and function well under different lighting conditions (Pomerleau et al., 2015).
2. Preprocessing: Point clouds obtained from acquisition contain various types of problems including noise elements as well as superfluous data and irregularities. Statistical outlier removal and voxel grid downsampling and normal estimation functions operate during preprocessing to improve data quality while reducing computational demands (Rusu & Cousins, 2011). The correct estimation of surface normals remains essential for executing both surface fitting operations and curvature analysis procedures.
3. Registration: The process of registration brings together various scans into one unified coordinate frame system. The Iterative Closest Point (ICP) algorithm and feature-based methods serve as common techniques to align scanned data with reference CAD models or other point clouds (Besl & McKay, 1992). The accuracy of registration plays a crucial role for detecting shape deviations and monitoring manufacturing changes in manufactured components.
4. Surface Reconstruction: Some workflows implement surface reconstruction techniques such as Poisson reconstruction or Delaunay triangulation for generating meshes and volumetric representations. The methods convert unprocessed point data into continuous surface models which enable curvature evaluation and visual examination and geometric processing capabilities (Kazhdan et al., 2006).

5. Geometric Comparison and Defect Detection: The last process requires the examination of processed scan data against digital reference or ideal geometry which is usually a CAD model. The evaluation of distance maps and color-coded error visualization helps identify features like dents and warping as well as cracks and dimension shifts (Sabri et al., 2016).

The sequence of these operations creates a complete system to utilize point cloud information in industrial inspection tasks requiring high precision. Traditional methods based on deterministic geometry maintain their strong position by supporting the integration of deep learning algorithms for advanced defect interpretation systems.

2.4.2 Optimization-Based Registration and Comparison Techniques

Unlike traditional alignment methods that depend on basic point correspondences, optimization-based techniques improve registration accuracy by iteratively minimizing error metrics like root mean square error (RMSE) and point-to-plane distances (Besl & McKay, 1992). Once the alignment is complete, pointwise or surface-based deviation maps are generated, enabling detailed analysis of local anomalies and global geometric discrepancies (Kazhdan et al., 2006).

Small inconsistencies can signal critical production faults or material fatigue, making accurate comparison techniques essential (Sabri et al., 2016). The optimization-based registration paradigm not only enhances robustness and accuracy but also supports scalable computation, making it suitable for both offline metrology workflows and real-time quality control systems (Pomerleau et al., 2015).

2.4.2.1 ICP-based

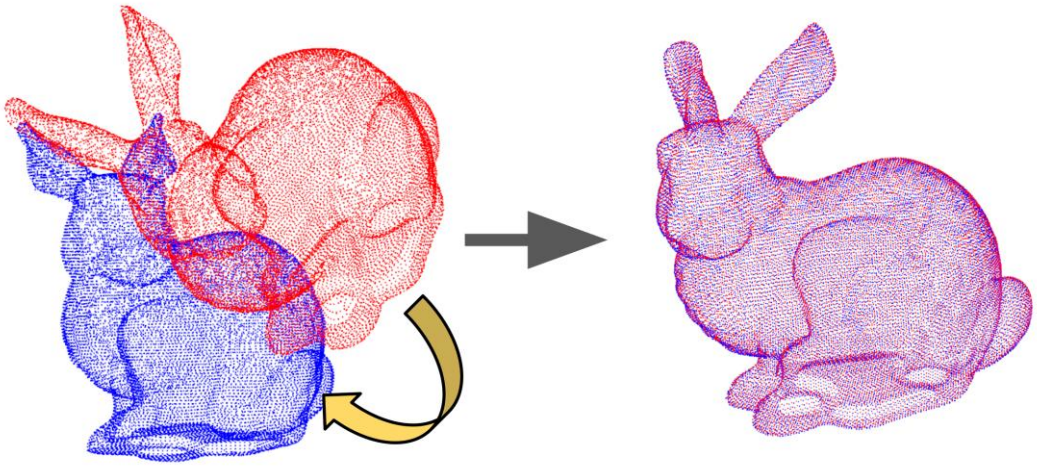


Figure 2-c: Iterative closest point protocol (Jaykumaran, 2025)

The Iterative Closest Point (ICP) algorithm stands as one of the fundamental methods for rigid 3D point cloud registration which works best when dealing with datasets of the same object or scene acquired from different angles. The algorithm was introduced (Besl & McKay, 1992) because of its simple design and successful performance in cases where an initial pose estimate exists.

The algorithm functions through an iterative optimization process that refines the alignment between a source and a target point cloud by minimizing the Euclidean distance between corresponding points. Each iteration of ICP comprises two fundamental steps:

1. Correspondence Estimation: In this phase, for each point in the source point cloud, the algorithm identifies the closest corresponding point in the target cloud based on spatial proximity. This nearest-neighbor search is critical for establishing reliable point correspondences and typically relies on spatial indexing structures such as k-d trees for computational efficiency (Muja & Lowe, 2009).
2. Transformation Estimation: After establishing correspondences, the algorithm determines the rigid transformation including rotation and translation that produces the smallest total alignment error. The optimisation problem is solved through least-squares methods which use either singular value decomposition (SVD) or quaternion-based approaches.

The algorithm performs these two steps repeatedly until it reaches convergence through either a specified iteration limit or an error change threshold. Researchers have developed ICP variants which enhance performance when dealing with noisy data and outliers and partial overlaps through point-to-plane ICP and robust error metrics (Rusinkiewicz & Levoy, 2001). The algorithm stands as a core method in 3D vision because it achieves accurate results with reasonable computational costs although it faces challenges from improper initialization and getting stuck in local optima.

2.4.2.2 DCP-based

The Deep Closest Point (DCP)-based methods implement learning-based optimization techniques into traditional point cloud registration systems. The ICP iterative algorithm differs from DCP because it uses attention mechanisms and neural feature embeddings to create correspondences between two point sets (Y. Wang & Solomon, 2019). The integration of deep learning into geometric processing enables these methods to achieve better resistance against noise and partial occlusions and initial misalignments. The network predicts transformations by minimizing a geometric loss function that uses PointNet or DGCNN architectures to learn global and local features from point correspondences.

2.4.3 Other Traditional 3D Techniques

In addition to registration-based comparison, several traditional geometric techniques are used for surface analysis and shape understanding. The surface fitting approach uses B-splines and least-squares minimization to create smooth surfaces from discrete point cloud samples which benefits reverse engineering and profile analysis (Hoppe et al., 1992).

The RANSAC-based method (Random Sample Consensus) serves as a popular approach for detecting geometric primitives such as planes, cylinders and spheres in noisy point cloud data. The method uses iterative point sampling to create model parameter hypotheses before checking which points match the model within specified tolerance limits (Schnabel et al., 2007). RANSAC techniques excel at segmenting objects and recognizing parts and detecting flat panels and cylindrical rods which frequently appear in industrial assemblies.

2.4.4 Limitations

The widespread application of traditional point cloud processing methods faces multiple significant limitations. Real-world scanning data contains frequent noise and outliers because of reflections and surface inconsistencies and occlusions which make these methods highly sensitive to these issues. The presence of imperfections in data results in substandard normal estimation and incorrect registration and unstable surface reconstruction (Rusu & Cousins, 2011). The processing of large-scale or high-resolution point clouds becomes computationally demanding for these methods. The ICP and RANSAC iterative algorithms need to perform extensive correspondence checks and model hypothesis evaluations which results in longer processing times.

Traditional methods fail to understand the semantic meaning of objects and scenes. The methods function based on geometric data without context which creates difficulties for distinguishing between significant defects and acceptable variations. The limitation becomes most apparent in complex inspection scenarios which require human-level interpretation or categorical recognition (Charles et al., 2017). Machine learning models and deep neural networks need to be integrated to address these gaps because they can detect both semantic features and geometric properties.

2.5 Traditional Image Segmentation Methods (Pre-Deep Learning)

After exploring traditional 3D point cloud methods, the focus shifts back to 2D image data, which continues to play a central role within the pipeline. The simplicity and accessibility of this modality make it still widely prevalent in industrial inspection tasks. Traditional image segmentation techniques function directly on pixel intensities, edges and regions to effectively partition images into analysable segments before deep learning methods became popular. The methods established basic principles for contemporary vision systems, yet they face their own set of difficulties and limitations (Q. Huang et al., 2023). The following section examines traditional segmentation approaches.

2.5.1 Intensity-Based Methods

Thresholding is one of the earliest and most straightforward approaches. It partitions an image into foreground and background based on a predefined intensity value or adaptive criteria. Otsu's method, an influential early method for thresholding, was proposed (Otsu, 1979), which uses minimum intra-class intensity variance to distinguish between foreground and background. The authors presented an alternative threshold selection method through histogram concavity analysis which built upon their initial work (Rosenfeld & De La Torre, 1983). Thresholding techniques commonly use Histogram Analysis as a complementary method. The analysis of intensity distribution peaks and valleys helps identify object boundaries. Intensity-based methods show limitations when dealing with images that have extreme illumination variations or insufficient intensity differences between objects and backgrounds.

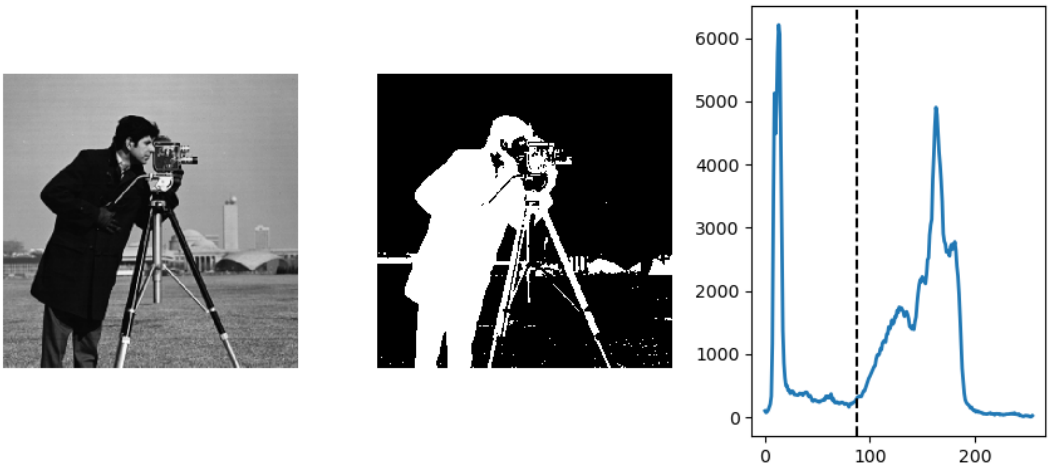


Figure 2-d: Otsu thresholding (*3.3.9.7. Otsu Thresholding — Scipy Lecture Notes*, n.d.)

2.5.2 Edge-Based Methods

The identification of intensity discontinuities which often represent object boundaries depends on edge detection techniques that use gradient operators such as the one in (Sobel, 1970). The methods create edge maps through pixel intensity changes detection but require extra processing to form meaningful edge segments. Active Contours known as Snakes emerged as a segmentation accuracy improvement method when Kass, Witkin, and Terzopoulos introduced them in 1988 (Kass et al., 1988). The method uses energy-minimizing curves which receive internal forces for smoothness control and external forces from image gradients. Active contours demonstrate excellent performance in boundary detection for noisy and incomplete edge conditions, but their performance depends on proper initialization, and they may get stuck in local minima during optimization.

Figure 2-e: Sobel edge detection (*Figure 10*, n.d.)

2.5.3 Region-Based Methods

The Region Growing method starts with seed points which then expand into neighbouring pixels that meet specific homogeneity criteria including intensity similarity or texture features (Adams & Bischof, 1994). The method maintains spatial relationships between pixels which produces segmented regions that remain both connected and logical. The Split-and-Merge algorithm (Horowitz & Pavlidis, 1976) starts by dividing images into quadrants through recursive splitting until each subregion fulfils the uniformity condition. The algorithm merges subregions when they demonstrate comparable features. The Split-and-Merge hierarchical

segmentation method provides better flexibility than basic global thresholding techniques yet faces challenges when processing images with complex textures and irregular structures.

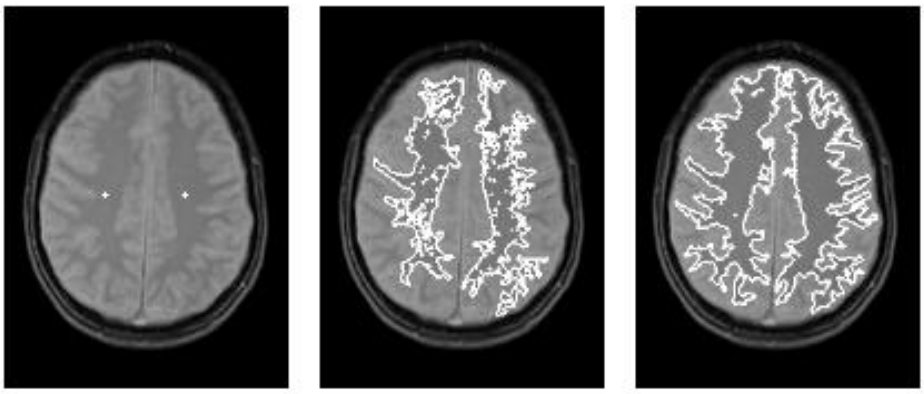


Figure 2-f: Region (seed) Growing Segmentation (Alaa, n.d.)

2.5.4 Other Approaches

The Watershed Algorithm treats an image's intensity surface as if it were a topographic map through which it identifies catchment basins using watersheds (Vincent & Soille, 1991). The method generates complete basin boundaries but requires proper marker control or region-merging heuristics to prevent over-segmentation of images. The clustering technique (k-means) performs unsupervised classification through pixel intensity or colour features to group pixels with matching characteristics. The technique works well for basic image scenes but tends to split objects when lighting conditions are non-uniform or when textures are present. Mathematical Morphology (Serra & Cressie, 1982) employs set-theoretic operations (erosion, dilation, opening, closing) to enhance shapes and eliminate noise while extracting meaningful structures. The technique shows its best results when applied to images with binary or near-binary content. Binary large object (BLOB) Detection methods identify areas with uniform intensity or texture patterns through scale-space analysis. The Difference of Gaussians in the scale-invariant feature transform (SIFT) pipeline (Lowe, 2004) detects interest points by identifying "blobs" that differ from the background. The methods demonstrate strong capabilities for specific feature-extraction operations, yet they may not be appropriate for complete image segmentation tasks.

2.5.5 Limitations

These methods maintain their historical significance yet encounter multiple significant obstacles. The combination of multiple surface attributes with unclear boundaries in complex scenes leads simple intensity- or edge-based measures to produce inaccurate segmentations. Threshold- or region-based methods become less robust when lighting changes occur

because these methods depend on specific conditions. The methods depend on local cues including intensity histograms and edges, but they do not possess strong semantic understanding or complete scene descriptions. The methods tend to produce either excessive segmentation or incorrect boundary establishment. The automation of these methods becomes more difficult because active contours and region growing require manual initialization and parameter adjustments such as seeds and thresholds which reduces their robustness across different datasets. The limitations led to the development of data-driven deep learning-based approaches which use extensive annotated data to discover strong segmentation features that are discussed in following sections.

2.6 Deep Learning for Image Segmentation: The Convolutional Era

Machine learning has developed deep learning as its transformative subset which enables automatic learning of hierarchical data representations during the last ten years. Deep learning models which include neural networks with multiple layers can automatically discover complex patterns from raw inputs without needing hand-designed features. The new paradigm has brought substantial progress to natural language processing and speech recognition and particularly to computer vision (Goodfellow et al., 2016). The combination of large-scale datasets with powerful GPUs has enabled visual perception systems to reach human-level accuracy in multiple tasks. Deep learning introduced a revolutionary change to image segmentation by addressing multiple problems which traditional segmentation methods faced. CNNs achieved unprecedented segmentation accuracies through large, annotated datasets and computational advances which established a foundation for better and more reliable computer vision implementations.

2.6.1 Early CNN Approaches

The initial CNN methods operated through patch-based classification by splitting images into separate patches which each received independent CNN computation. The first deep network architectures emerged with AlexNet (Krizhevsky et al., 2012) before VGGNet (Simonyan & Zisserman, 2015) further developed these networks. The effectiveness of these techniques came with the drawback of redundant patch overlaps and challenges in utilizing advanced image relationships. A breakthrough was achieved with Fully Convolutional Networks (FCNs) by (Shelhamer et al., 2017), which revolutionized segmentation by making end-to-end pixel-level prediction possible. The previously patch-based methods were replaced by FCNs which used fully convolutional layers to process images of any size thus reducing both redundancy and computational costs. The model produced dense pixel-wise predictions which became the foundation for subsequent segmentation advancements.

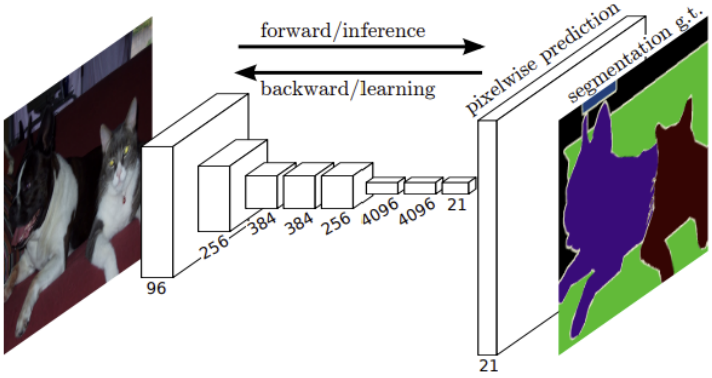


Figure 2-g: FCN workflow (‘(PDF) Comparison of Fully Convolutional Networks (FCN) and U-Net for Road Segmentation from High Resolution Imageries’, 2025)

2.6.2 Encoder-Decoder Architectures

The focus shifted to encoder-decoder architectures after researchers encountered resolution loss problems due to pooling operations in CNNs. The U-Net architecture (Ronneberger et al., 2015) stands out as a notable example which was initially developed for biomedical imaging tasks. The explicit connections between encoder and decoder phases in U-Net enabled the preservation of spatial details at high levels which resulted in better segmentation accuracy. The architecture gained widespread adoption beyond biomedical applications to become a fundamental component for segmentation tasks in various domains. SegNet (Badrinarayanan et al., 2017) implemented encoder-stage pooling indices to control upsample operations which resulted in improved decoder efficiency and real-time segmentation speed.

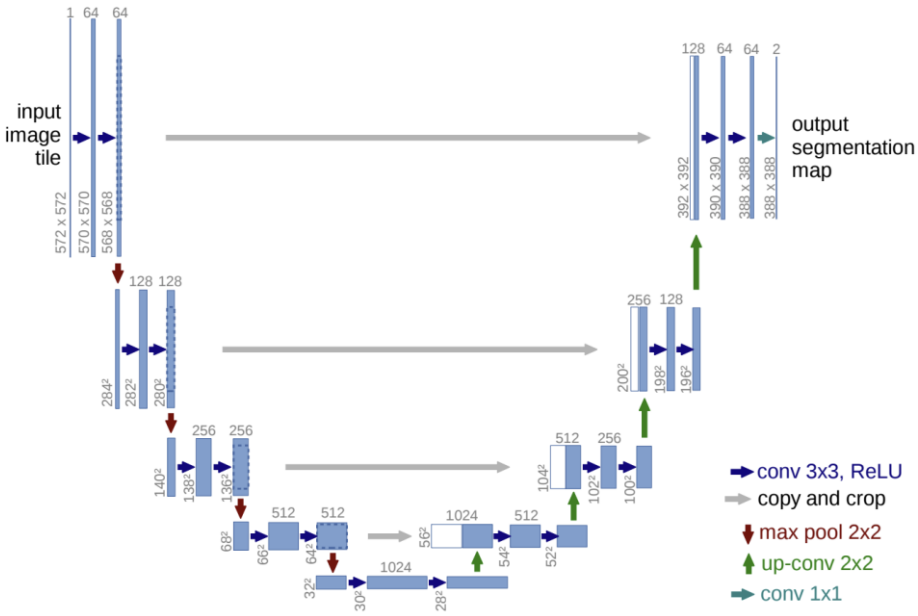


Figure 2-h: U-Net Architecture (*U-Net - an Overview* | *ScienceDirect Topics*, n.d.)

2.6.3 Enhancing Context and Resolution

The development of encoder-decoder architectures led researchers to concentrate on improving contextual understanding while preserving spatial information. The DeepLab series introduced atrous convolution to increase receptive field size without adding many parameters before incorporating Atrous Spatial Pyramid Pooling (ASPP) for enhancing deep multi-scale context understanding (L.-C. Chen et al., 2018). PSPNet introduced global scene perception through pyramid pooling of contextual features across different scales of granularity (Zhao et al., 2017a). High-Resolution Networks advanced this concept by maintaining high-resolution representations throughout the network through parallel multi-scale pathways for precise localization (Sun et al., 2019).

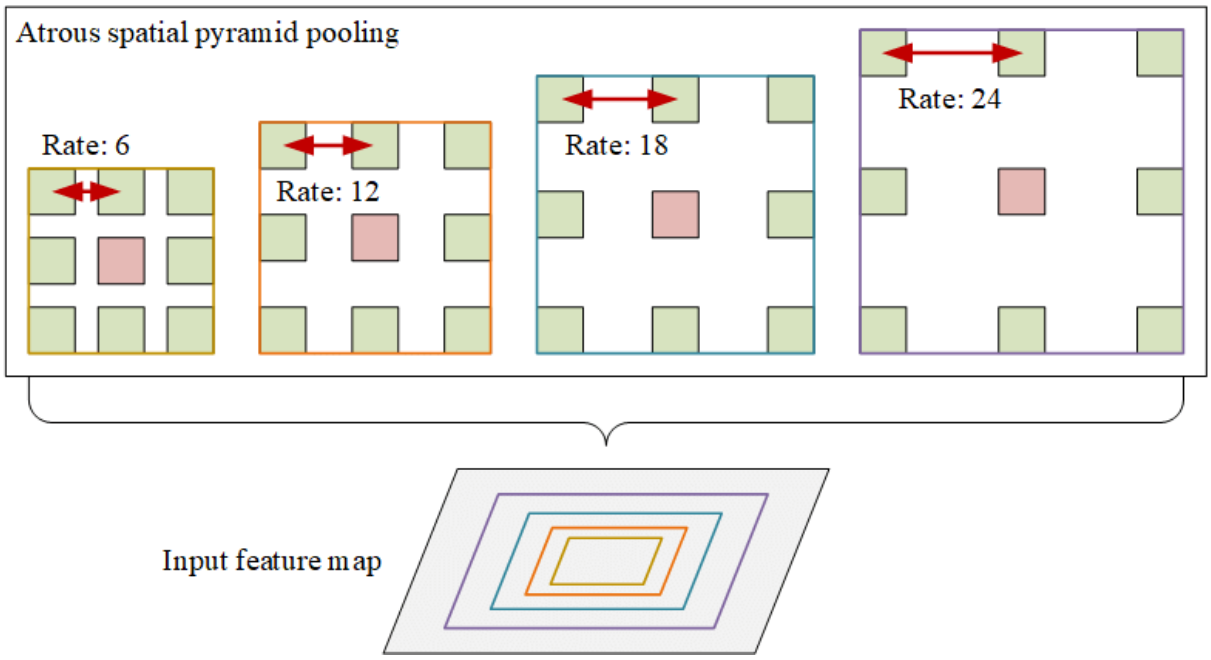


Figure 2-i: Atrous spatial pyramid pooling ('(PDF) A Deep Neural Network for Oil Spill Semantic Segmentation in Sar Images', 2025)

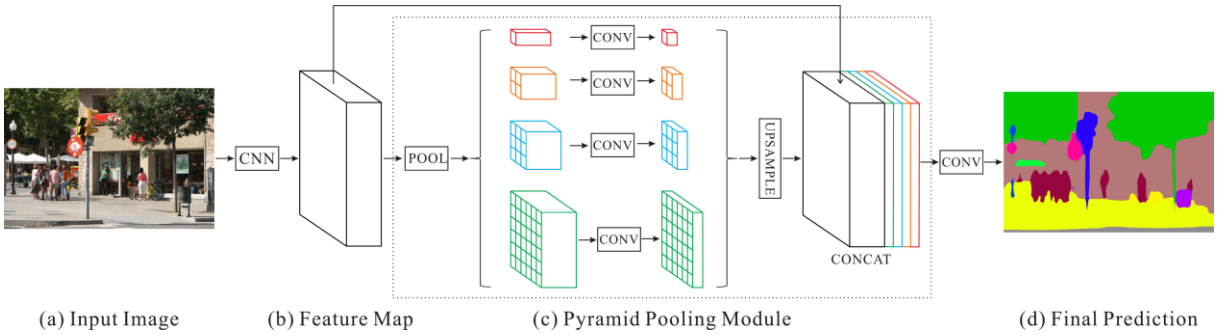


Figure 2-j: PSPNet workflow (Zhao et al., 2017b)

2.6.4 Instance Segmentation Advances

The instance segmentation concept and object detection and pixel-level segmentation merged into one unified architecture through subsequent innovation. Mask R-CNN (He, Gkioxari, et al., 2020) built upon the effective Faster R-CNN (S. Ren et al., 2017) detection model by adding a segmentation mask branch which enabled the detection and segmentation of multiple object instances at once. The breakthrough established new instance segmentation benchmarks which transformed both research and real-world applications.

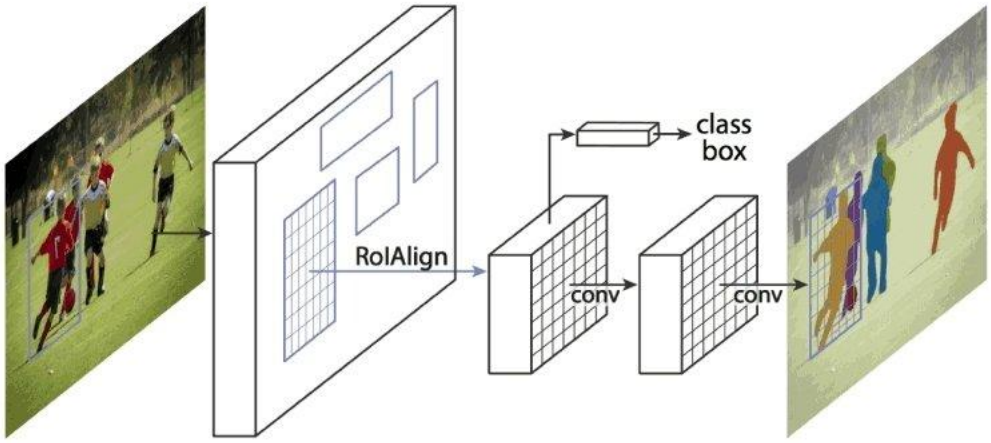


Figure 2-k: The Mask-RCNN framework (He et al., 2018)

2.6.5 The YOLO Framework for Real-Time Object Detection

2.6.5.1 Concept

YOLO reframes object detection as a single-stage regression problem: an image is fed through one convolutional backbone, and in a single forward pass the network simultaneously predicts bounding-box coordinates and class probabilities across a dense grid. Eliminating the region-proposal stage that characterises two-stage detectors (e.g., Faster R-CNN) yields inference latencies measured in milliseconds, making YOLO a natural fit for high-throughput industrial inspection lines where every frame must be analysed in real time.

2.6.5.2 Condensed History

The YOLO family began strictly as a *detection-only* framework and has since expanded to support additional prediction heads, including segmentation in its latest generations. YOLOv1 (2016) introduced a coarse grid-cell formulation for fast bounding-box regression. YOLOv3 (2018) added anchor boxes and multi-scale feature fusion, substantially boosting small-object recall. The v5/6 generation (2020–21) adopted a CSPDarknet backbone and more aggressive data augmentation, matching the accuracy of heavier detectors while remaining lightweight. YOLOv8 (2023) was the first release to include native mask and pose heads, integrating segmentation into the single-stage paradigm. The v10/11 generation (2024) further advanced the architecture: YOLOv10 introduced NMS-free training with dual label assignment for end-to-end detection, while YOLOv11 incorporated C3K2 blocks for efficient feature representation, SPFF modules for enhanced small-object detection via multi-scale pooling, and C2PSA blocks

with spatial attention—collectively achieving higher accuracy and faster inference across diverse tasks.

2.6.5.3 Multi-Head Versatility

Modern YOLO implementations expose a modular prediction head architecture. A bounding-box head delivers classic object detection; an optional segmentation head outputs instance masks; a classification head supports whole-image labelling; and specialised extensions add pose/key-point or monocular-depth estimation streams. All heads share the same backbone and neck, allowing a single model to serve multiple inspection objectives without duplicating feature extraction.

2.7 Deep Learning for Image Segmentation: Transformer Revolution

The field of image segmentation has remained dominated by CNNs for nearly ten years, but their fundamental design limitations became apparent when tasks needed deeper understanding of global context. The main weakness of CNNs was their limited local receptive field because convolutional layers process information through fixed-size windows which made it difficult to capture long-distance dependencies without using multiple layers or creative dilation schemes. The translation invariance and locality inductive biases of CNNs help certain tasks but they limit both flexibility and generalization capabilities. Vision Transformers (ViTs) provided a new option. The transformer architecture emerged from the original paper "Attention Is All You Need" (Vaswani et al., n.d.) which was first developed for language processing. The self-attention mechanism in these models enables the computation of global relationships between all input elements. The transformer architecture which powers modern large language models (LLMs) including GPT and Gemini uses the same fundamental mechanisms to process sequential text data. The attention mechanism achieved significant performance improvements when applied to vision tasks for both classification and segmentation.

2.7.1 Vision Transformer (ViT)

The first Vision Transformer (ViT) model introduced a major breakthrough (Dosovitskiy et al., 2020). Unlike traditional convolutional methods, ViT divides an image into non-overlapping patches (e.g., 16x16) which are then flattened and treated as tokens in a standard transformer encoder. With sufficient data (e.g., JFT-300M) and compute, ViT outperformed state-of-the-art CNNs on benchmarks such as ImageNet. Nevertheless, its quadratic attention scaling and limited data efficiency made it less suitable for smaller datasets or high-resolution inputs without adaptation.

The Detection Transformer (DETR) (Carion et al., 2020) pioneered the application of pure self-attention mechanisms to object detection by formulating the task as a set-prediction problem, removing the need for components like anchor generation and non-maximum suppression. By demonstrating competitive performance in end-to-end bounding-box and class prediction, DETR provided a powerful template for transformer-based dense-prediction and directly inspired segmentation architectures that followed.

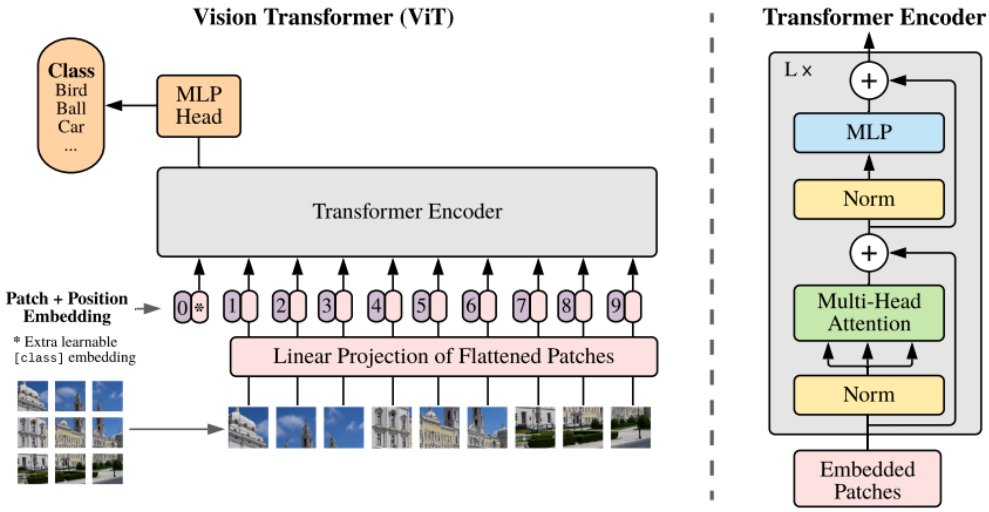


Figure 2-I: Vision Transformer architecture (Dosovitskiy et al., 2021b)

2.7.2 Adapting Transformers for Segmentation

The first attempts to use transformer global modelling capabilities for segmentation involved researchers applying ViTs to dense prediction tasks. The first attempt at this was SETR (Zheng et al., 2021a) which redefined segmentation as a sequence-to-sequence task by replacing the CNN encoder with a transformer and then a decoder that up sampled and reshaped the output. Although a pioneering effort, SETR had difficulty retaining spatial detail. Segmenter (Strudel et al., 2021) built on this method by using a ViT backbone and a mask transformer head for optimizing global context representation while maintaining segmentation fidelity. These methods proved that transformers could be effective segmentation backbones but came with the necessity for structural breakthroughs in addressing computations and resolution.

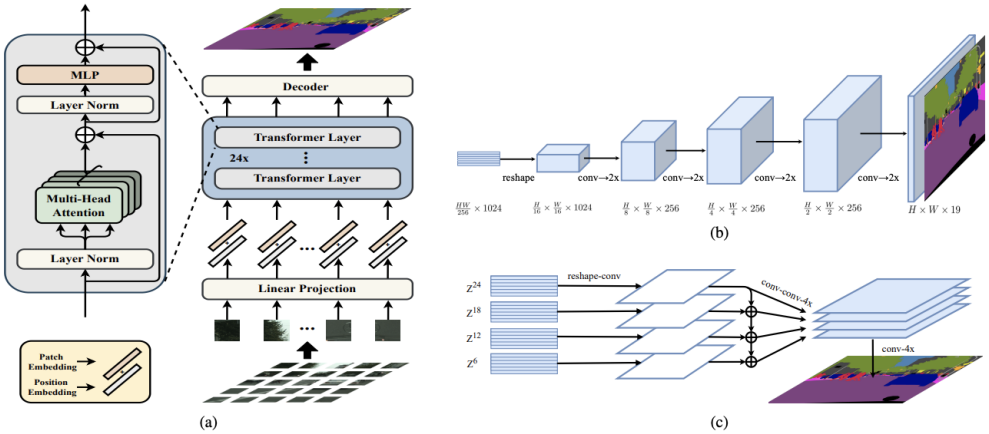


Figure 1. **Schematic illustration of the proposed *SEgmentation TRansformer* (SETR)** (a). We first split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. To perform pixel-wise segmentation, we introduce different decoder designs: (b) progressive upsampling (resulting in a variant called SETR-*PUP*); and (c) multi-level feature aggregation (a variant called SETR-*MLA*).

Figure 2-m: SETR workflow (Zheng et al., 2021b)

.

2.7.3 Hierarchical and Efficient Vision Transformers

Many hierarchical and hybrid transformer models were developed to reduce the computational cost and improve scalability. The Swin Transformer (Z. Liu et al., 2021)proposed a new method based on shifted windows for efficient local attention computation that facilitated the model upscaling to higher-resolution images. Swin's hierarchical feature maps made it especially suitable for segmentation and detection.

The Pyramid Vision Transformer (W. Wang et al., 2021) presented a pyramid framework with spatial-reduction attention for simplifying the model while giving rise to multi-scale feature extraction like that of CNNs.

LeViT (Graham et al., 2021) followed a different approach with the design of a more lightweight hybrid model that mixed convolutional early stages with attention blocks. The combination of these created rapid inference with competitive accuracy, which made them better suited for edge devices.

2.8 Foundation Models and Zero-Shot Segmentation

2.8.1 The Concept of Foundation Models in Vision

The main purpose of foundation models is to train large neural networks on extensive datasets for developing general-purpose representations which can transfer between different applications (Bommasani et al., 2022). The machine learning field has undergone a significant transformation because foundation models function as versatile backbones which enable direct application or fine-tuning for numerous vision tasks (Khan et al., 2022). The ability to use foundation models has decreased the need for task-specific labelled datasets that previously needed to train separate models for each application (Radford et al., 2021). The outstanding success of foundation models in natural language processing through GPT architecture has motivated researchers to develop similar advancements in computer vision (Brown et al., 2020). The cross-domain influence has resulted in transformative performance improvements which enable zero-shot inference capabilities for models; which is to perform tasks and recognize categories without requiring extra training (Radford et al., 2021).

2.8.2 Self-Supervised Learning (SSL)

Self-supervised learning (SSL) serves as a fundamental method in computer vision because it provides scalable alternatives to supervised methods through model training on unlabeled data to learn meaningful visual representations. SSL trains models to recognize semantic structure and spatial relationships and visual patterns through pretext tasks which are automatically generated from intrinsic data properties (Jing & Tian, 2021). The techniques of contrastive learning (T. Chen et al., 2020; He, Fan, et al., 2020) and masked image modeling (He et al., 2022) show that models can match or outperform supervised counterparts especially when labels are scarce. The progress in vision foundation models has become essential for reducing annotation requirements while enabling zero-shot and transfer learning across multiple downstream tasks.

The Masked Autoencoder (MAE) represents a key example of self-supervised learning (SSL) which enables foundation models to discover meaningful representations from unlabeled data through inherent patterns and structures without manual labels. The Masked Autoencoder (MAE) serves as a key component in ConvNeXt V2 (Woo et al., 2023) because it helps models predict missing parts of input images. The reconstruction process forces the model to understand visual content as a whole which leads to strong feature learning without needing explicit supervision. The Segment Anything Model (Kirillov et al., 2023) and its successors SAM2 (Ravi et al., 2024a) and SAM2.1 implement SSL principles at scale through training on extensive segmentation datasets including SA-1B and its expanded versions.

The models use prompt-based mask prediction to enable adaptable segmentation across various prompts. These approaches learn from data through label-efficient methods which decreases human annotation costs while reaching performance levels equivalent to fully supervised methods.

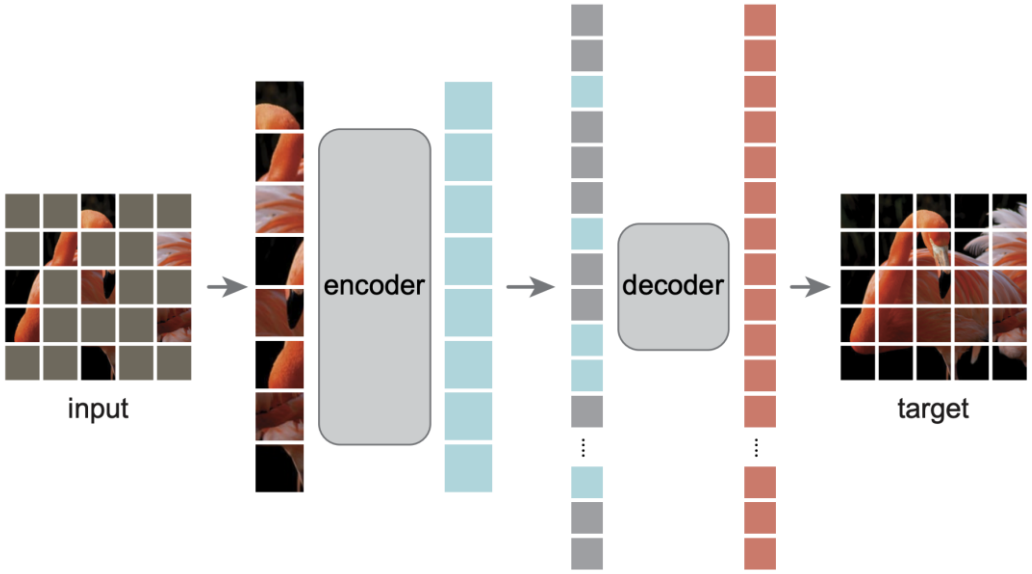


Figure 2-n: MAE workflow (He et al., 2021)

2.8.3 Vision-Language Models (VLMs)

The vision–language models CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) learn a common embedding space through training on extensive image–text pair datasets where similar visual and textual meanings result in close spatial positions. The models use this joint space during inference to perform open-vocabulary zero-shot classification by determining the alignment quality between images and image regions and natural language class names. The method allows task-independent fine-tuning and enables classification across numerous concepts without requiring specific training. The strong dependence on the prompt set emerges because of this approach because categories outside the prompt list remain unidentified. Standard VLM architectures generate predictions only at the image level without object localization or segmentation capabilities. Researchers have recently combined VLM-derived embeddings with specialized localization and segmentation heads to create object detectors and segmenters which maintain open-vocabulary generalization from the vision–language backbone.

2.8.4 Open-World Recognition and Detection (OWOD)

The Open-World Object Detection (OWOD) system combines three essential detection functions into one system.

1. Recognition of known objects: Accurately detecting and classifying objects from a predefined set of categories.
2. Identification of unknown objects: Distinguishing previously unseen objects at inference time without misclassifying them as one of the known categories.
3. Incremental learning: Integrating newly discovered classes over time without erasing or degrading previously learned knowledge (i.e., avoiding catastrophic forgetting).

The Open-World Recognition (ORE) framework introduced this task according to (Joseph et al., 2021). Early OWOD models used visual features to identify unknown objects, but these models did not possess semantic understanding or descriptive flexibility.

Open-vocabulary detection systems now use vision-language model (VLM) features to overcome the limitations of previous systems. OWL-ViT (Minderer et al., 2022), Detic (X. Zhou et al., 2022), RegionCLIP (Zhong et al., 2022) and Grounding DINO (S. Liu et al., 2024) are models that enable region-level grounding based on arbitrary textual queries. OWLv2 (Minderer et al.,), the successor of OWL-ViT, achieves zero-shot object localization and classification through its Vision Transformer (ViT) backbone which combines with CLIP-style text embeddings during a single forward pass.

The models demonstrate superior performance in text-based flexible object detection, yet they do not achieve full OWOD functionality because they do not include mechanisms for adapting to newly labelled classes. The open-vocabulary detection system provides a solid base for OWOD yet achieving the full open-world protocol demands the implementation of continual learning approaches.

2.8.5 The Segment Anything Model (SAM)

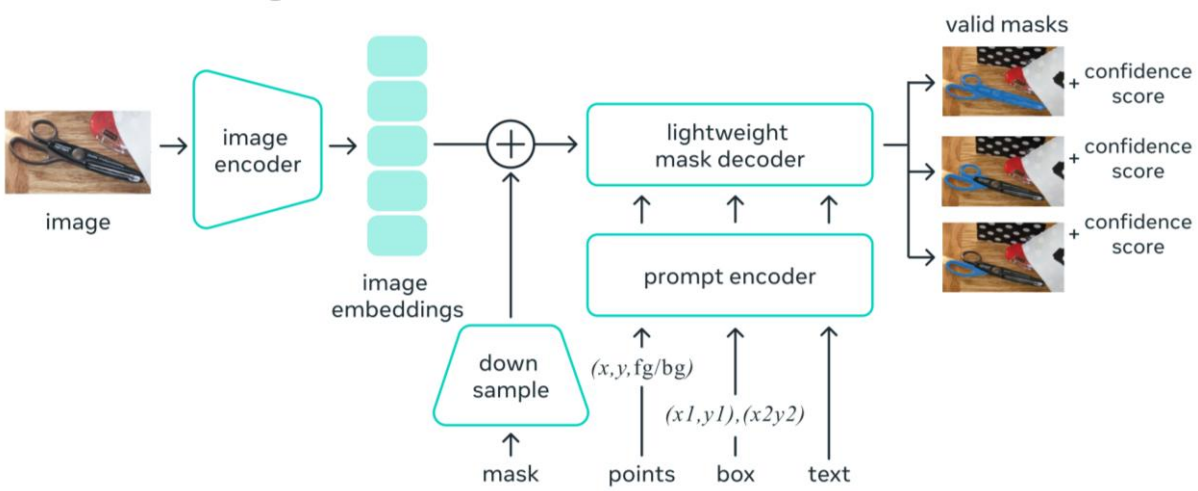


Figure 2-o: SAM workflow (Ravi et al., 2024b)

2.8.5.1 Architecture, SA-1B dataset, prompt engineering (points, boxes, text).

The introduction of the Segment Anything Model (SAM) serves as a major breakthrough for zero-shot segmentation through its highly flexible and promptable architecture (Kirillov et al., 2023). The core functionality of SAM uses transformer-based processing which divides into two separate image and user prompt streams. The image encoder which uses Vision Transformer (ViT) processes input images once to produce dense embeddings that contain extensive visual information. The prompt encoder transforms user inputs such as points and bounding boxes and textual descriptions into a unified format which matches the model requirements. The mask decoder receives the encoded visual and prompt information from separate streams to generate the final segmentation mask. The modular structure of SAM enables it to share image features between different prompts which improves both speed and interactive segmentation flexibility.

The large-scale pretraining of SAM on the SA-1B dataset which contains more than one billion high-quality segmentation masks from diverse image domains enables its powerful capabilities.

The flexible nature of SAM stems from its prompt-based interaction capabilities. Users can steer segmentation through three different prompt categories including point prompts for exact region control and bounding box prompts for fast object segmentation and text prompts for semantic-based segmentation. The prompt engineering framework of SAM establishes it as a versatile tool which can handle a wide range of applications from interactive image editing to automated scene analysis.

2.8.5.2 Capabilities: Zero-Shot segmentation performance

SAM has demonstrated state-of-the-art performance in zero-shot segmentation across a wide range of tasks, exhibiting exceptional generalization to scenarios that differ significantly from its training conditions. Unlike traditional segmentation models that require extensive task-specific fine-tuning and annotated data, SAM leverages its large-scale pretraining and prompt-based inference architecture to generate high-quality segmentation masks without additional supervision. This capability is particularly advantageous in domains where labeled data is scarce, costly, or difficult to acquire, such as medical imaging (Z. Huang et al., 2023).

Empirical evaluations show that SAM achieves competitive, and in some cases superior, segmentation performance compared to fully supervised models tailored to specific tasks. Its robustness in zero-shot settings underscores its practical utility, allowing for the rapid deployment of segmentation solutions in new environments without the typical overhead of data collection and model retraining (Kirillov et al., 2023; X. Li et al., 2023).

2.8.5.3 SAM 2 and SAM 2.1

SAM 2 (Ravi et al., 2024a, p. 2) replaces the original image-only pipeline with a unified transformer that segments both images and videos, adds a streaming-memory module for temporal consistency, and achieves ~6× faster image inference while requiring roughly one-third the user interactions for comparable quality.

The subsequent SAM 2.1 checkpoints sharpen accuracy on small, visually similar and partially occluded objects through richer augmentations, longer training sequences and refined positional encodings, and they ship with an open-source developer suite (training + demo code) that makes task-specific fine-tuning straightforward.

2.8.5.4 Known Limitations

The generalization power of SAM remains strong, but the model demonstrates significant weaknesses when dealing with highly specialized or unseen domains. The geospatial mapping tasks present a challenge to SAM because the model fails to segment ice-wedge polygons and thaw slumps due to the large domain difference between its general training data and the target domain (W. Li et al., 2024).

The zero-shot capability of SAM does not automatically transfer to tasks that have different structures or fine-grained visual variations. The applications of industrial defect detection and medical imaging face challenges because accuracy stands as a critical requirement. The quality of user-provided prompts represents a major challenge for SAM because it depends on these inputs to function. The model produces substandard segmentations when users provide unclear or imprecise prompts because it remains highly responsive to the wording of input

prompts. The segmentation process of SAM lacks the ability to create semantic labels for its output segments which reduces its effectiveness in classification-based applications.

The limitations of SAM can be addressed through the use of outside detectors or specific fine-tuning for particular fields when operating in complex industrial environments.

2.9 Leveraging Multi-Modal Data (RGB + NIR)

The three visible RGB channels in inspection systems detect only a restricted set of optical material characteristics. The addition of near-infrared (NIR) imaging to standard systems which uses silicon-based sensors from 700 nm to 1,000 nm and InGaAs arrays up to 1,700 nm enables the detection of molecular vibrations and moisture absorption and subsurface scattering (Argirusis et al., 2025). Standard cameras cannot detect these spectral cues which reveal early-stage or subtle material anomalies that would otherwise remain undetected. The combination of effective NIR data fusion with traditional RGB imagery produces a substantially enhanced feature set which improves automated inspection capabilities. (Zhang & Zhu, 2023).

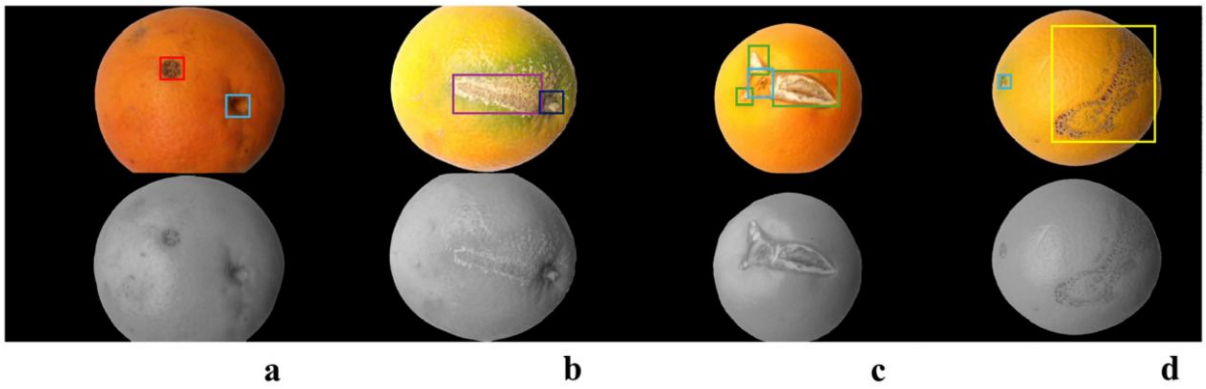


Figure 2-p: Examples of RGB + NIR images and annotations in the dataset (J. Luo et al., 2025)

Red rectangular box (a): example of canker annotation; purple rectangular box (b): example of pest annotation; green rectangular box: example of crack annotation (c); yellow rectangular box (d): example of melanose annotation; blue rectangular box (a,c,d): example of navel annotation; and black rectangular box (b): example of peduncle annotation.

2.9.1 Motivation behind using multi-modal data

Human beings perceive colour through three channels yet convolutional and transformer-based networks operate without these biological limitations. Standard 3D RGB data becomes more informative when neural networks receive additional wavelength information which results in better decisions and improved outcomes.

The combination of RGB and NIR data offers three main advantages. First, NIR light penetrates deeper than visible light through paints, lacquers, and thin polymer coatings, enabling the detection of latent damage before any visual changes appear. Second, the unique reflectance properties in the NIR spectrum help distinguish between materials that appear similar in RGB, allowing more accurate classification based on spectral signatures and surface texture. Third, NIR reflectance remains more stable under varying lighting conditions,

improving detection performance in challenging environments such as nighttime, fog, or low-angle sunlight.

Because absorption in NIR arises from overtone and combination bands of fundamental molecular vibrations, small changes in chemistry or porosity alter a surface's reflectance curve. Portable spectrometers have long exploited this fact for in-field agro-food analysis, measuring moisture, sugar and fat without destructive sampling (dos Santos et al., 2013). This can detect unwanted things in food before they are visible with visible light. For pipeline inspection, oxide layers, under-film corrosion or trapped moisture pockets generate analogous spectral deviations that RGB cannot register until damage is advanced.

2.9.2 Multi-Modal Fusion Strategies

A vision backbone can incorporate a fourth near-infrared (NIR) channel through three fundamental architectural methods. The early fusion method combines NIR data with RGB channels through concatenation to enable a single network that learns cross-modal features from the start. Real-time systems implement this method because it maintains simplicity and causes minimal disruption to the system architecture. The intermediate fusion method divides RGB and NIR data between separate branches for multiple layers before uniting their feature representations through methods like feature concatenation and gated convolutions and cross-modal attention. The RGB and NIR modalities operate independently in late fusion methods which combine their outputs through weighted averaging or voting of logits or segmentation masks. The strategy demonstrates excellent resistance in automotive night vision applications because NIR detectors function as supplements to RGB pipelines (Y. Luo & Luo, 2023).

The current generation of deep learning frameworks enable these fusion methods through small adjustments to their architecture. The addition of a fourth modality in convolutional networks requires increasing the input channel count from three to four while Vision Transformers need a patch embedding projection modification to support the new modality. Advanced hybrid architectures use squeeze-and-excitation and global response normalization layers to dynamically adjust RGB and NIR input contributions because modern hardware provides memory efficiency. The intermediate fusion method produces the most significant improvements in mAP and IoU metrics, but early fusion remains suitable for environments where latency and power consumption and model size are essential factors. Vision systems gain extended sensory capabilities through NIR data integration which enables earlier and more reliable defect detection and classification beyond what visible-spectrum imaging can achieve.

2.10 Evaluation Metrics for Detection and Segmentation

2.10.1 3-D Geometry Metrics

The evaluation of detection accuracy, segmentation quality, and registration performance in 3D systems relies on several fundamental geometric metrics. The fundamental metric between two points in space is Euclidean distance which serves as the basis for multiple complex calculations.

$$d = ||p_i - q_i|| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Where:

- $\boldsymbol{p_i}$ is a point in the defective cloud
- $\boldsymbol{q_i}$ is its closest neighbour in the reference cloud
- d is the deviation at point

The Root Mean Square Error (RMSE) functions as a standard registration accuracy metric as it determines the average squared distance between matching points from two point clouds.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} ||p_i - q_i||^2}$$

Where,

- N = total number of matched point pairs
- p_i= point iii in the source (defective) point cloud
- q_i= corresponding point in the target (reference) point cloud
- $||\boldsymbol{p_i} - \boldsymbol{q_i}||$= Euclidean distance between the two points

The fitness score evaluates alignment quality through its calculation of inlier correspondences which represent points within a specified distance threshold relative to the total source points (Q.-Y. Zhou et al., 2018)

$$\text{Fitness} = \frac{|\text{Correspondences with distance} < d_{\max}|}{|\text{Total points in source point cloud}|}$$

The fundamental metrics establish a base for assessing 3D vision systems and serve as critical indicators to measure how well models represent real-world geometric data.

2.10.2 Pixel-Level Metrics

Pixel-wise correspondence between a predicted mask P and its ground truth G is quantified first by Intersection-over-Union (IoU), also known as the Jaccard Index:

$$\boldsymbol{IoU(P,G) = \frac{|P \cap G|}{|P \cup G|}}$$

This statistic, standardised by (Everingham et al., 2010), penalises both over- and under-segmentation and is averaged across classes to yield mean IoU (mIoU).

Because IoU can become overly harsh when the positive class occupies only a minute fraction of the image, a second measure is reported, the Dice Coefficient (Dice, 1945):

$$\boldsymbol{Dice(P,G) = \frac{2|P \cap G|}{|P| + |G|}}$$

The Dice Coefficient increases the weight of true-positive pixels and is therefore less sensitive to severe class imbalance—an advantage when corrosion speckles cover less than one percent of the frame. Its utility in such scenarios has contributed to its widespread adoption, notably in fields like medical imaging where precise segmentation of small structures is critical (Milletari et al., 2016).

A third statistic, Pixel Accuracy (PA), provides a coarse overall indication of labelling correctness, often noted in early influential works on semantic segmentation (Shelhamer et al., 2017). Although PA can be dominated by large background regions, it quickly exposes gross prediction failures.

$$\boldsymbol{PA = \frac{\#correctly\ labeled\ pixels}{\#total\ pixels}}$$

2.10.3 Object/Instance-Level Metrics

Instance-level performance begins with Precision and Recall, defined as

$$\boldsymbol{Precision = \frac{TP}{TP + FP}}$$

$$\boldsymbol{Recall = \frac{TP}{TP + FN}}$$

where TP, FP, and FN denote true positives, false positives and false negatives, respectively. As the confidence threshold varies, these two quantities trace a Precision–Recall curve; the area under that curve is the Average Precision (AP),

$$AP = \int_0^1 p(r)\, dr \approx \sum_k (r_k - r_{k-1})\, p_{interp}(r_k).$$

In this expression, $p(r)$ denotes the precision at recall level r, r_k and r_{k-1} are successive sampled recall values, and $p_{interp}(r_k)$ is the interpolated precision at recall threshold r_k.

MS-COCO (Lin et al., 2014) evaluates AP at ten IoU thresholds from 0.50 to 0.95 and averages the resulting values to produce mAP@[0.5:0.95]; the legacy PASCAL VOC protocol reports only the single threshold mAP@[0.5]. This single threshold is still the most reported in object detection. Additionally, COCO distinguishes between bounding-box mAP—which measures detection quality via box IoU—and mask mAP—which measures instance segmentation quality via mask IoU. So, these metrics can be applied to both object detection and segmentation tasks.

When a single scalar is required at a fixed operating point, the F1 score offers a concise summary:

$$\boldsymbol{F1} = \mathbf{2}\,\frac{\boldsymbol{Precision} \times \boldsymbol{Recall}}{\boldsymbol{Precision} + \boldsymbol{Recall}}$$

2.11 Summary and Research Gap

Industrial inspection has progressed from rule-based 2-D/3-D techniques—thresholding, watershed, ICP—to deep CNNs and, more recently, Vision-Transformer–based foundation models such as SAM and OWL-ViT. These promptable models extend coverage to unseen classes without retraining, while task-specific networks like YOLOv11 still deliver top accuracy when labelled data are available.

Current systems, however, rarely unite flexibility, precision, and modality breadth: RGB-only inputs overlook spectral or depth cues, and any single model struggles with both obvious and subtle defect modes. Our thesis therefore advances a hybrid strategy: (i) enrich image capture with additional spectral and range channels to expose defects that are harder to see; (ii) keep ICP at the core for deviations best described by geometry; and (iii) run two complementary detectors— a zero-shot, prompt-driven foundation model for rapid class expansion and a supervised YOLO alternative when annotations permit. This combination targets the outstanding need for an adaptable, high-accuracy pipeline that can be deployed with minimal re-engineering across varied defect scenarios.

3 MATERIALS & METHODS

3.1 Imaging & Acquisition Hardware

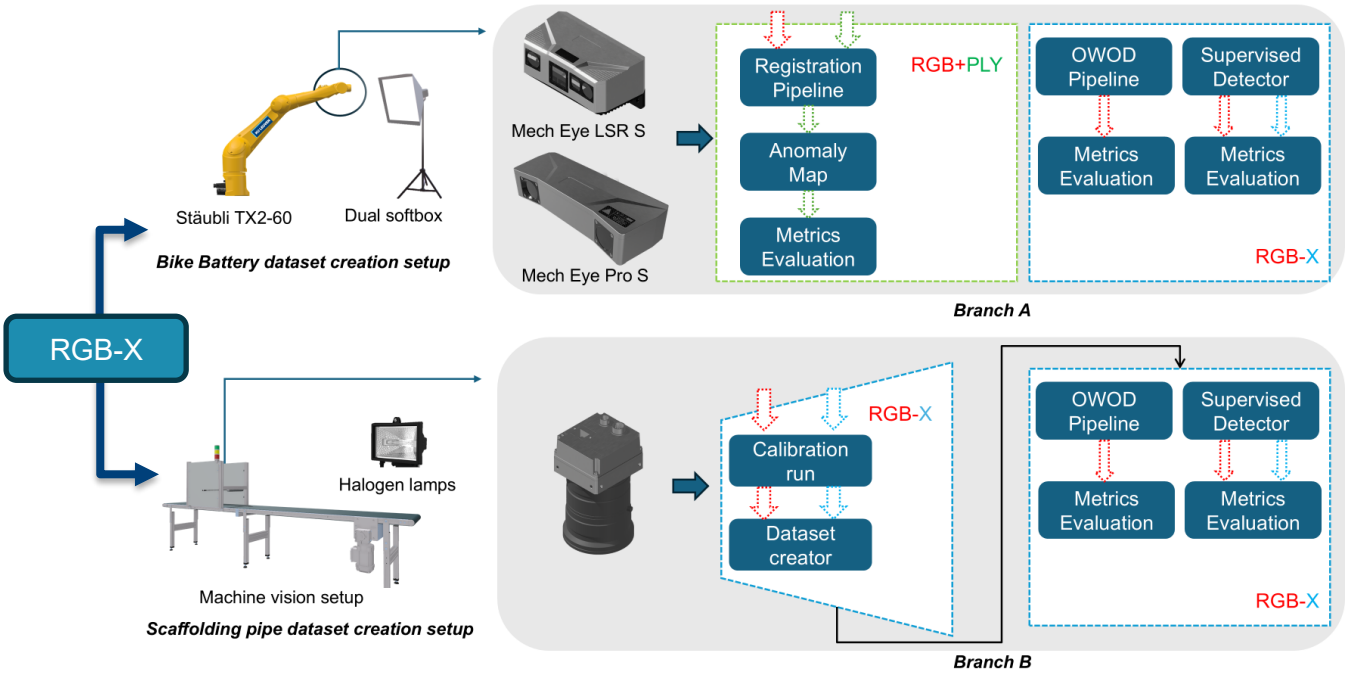


Figure 3-a: Overview of pipeline starting from image acquisition to software pipeline

3.1.1 Hardware setup capturing e-bike battery

The initial development of our inspection pipeline was carried out on the high-resolution e-bike battery dataset collected in previous years. To ensure reproducibility and facilitate subsequent transfer to the scaffolding-pipe dataset, the original acquisition setup and its operating parameters are briefly summarised below.

Figure 3-b: e-bike battery dataset setup with soft box

The acquisition platform integrated two industrial cameras—Mech Eye Pro S and Mech Eye LSR S—which were secured rigidly to the Stäubli TX2-60 six-axis robot flange. The fixed stereo baseline and hand–eye transformation consistency throughout the sequence were achieved through this configuration.

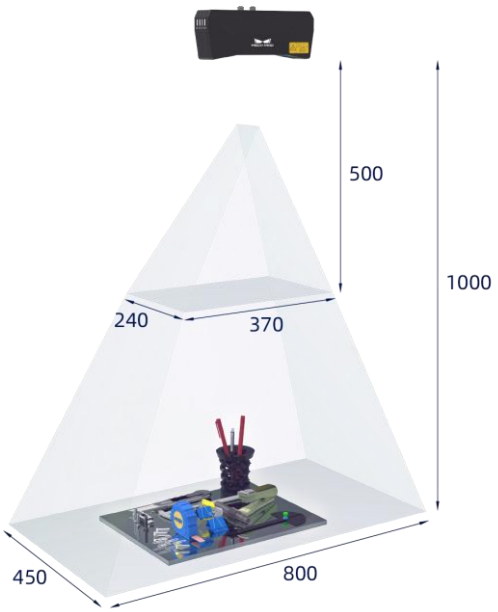


Figure 3-c: Mech-Eye Pro-S: Field of view (*PRO S-GL and PRO M-GL*, n.d.)

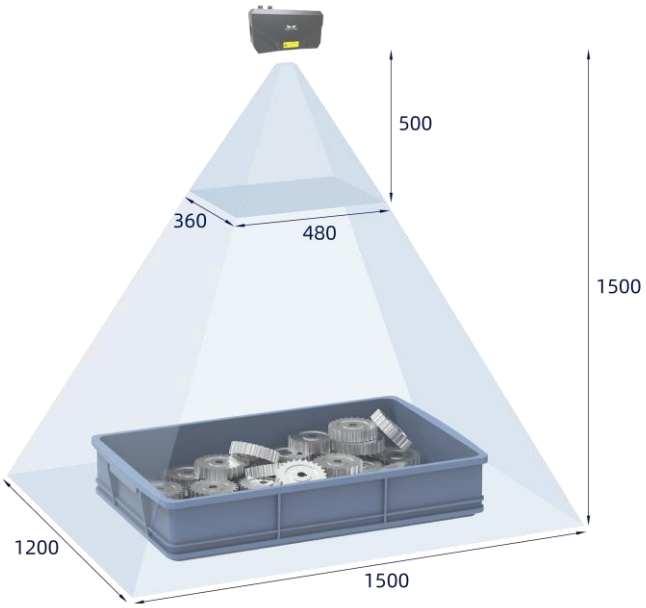


Figure 3-d: Mech-Eye LSR-S: Field of view (*LSR S-GL*, n.d.)

A Python script operated the robot to execute a pre-programmed trajectory which started at the true top-down position. The robot performed two 180° circular motions which rotated about its body-fixed X-axis and Y-axis while stopping at 5° intervals. The eight canonical viewpoints per battery cycle provided complete line-of-sight coverage for all external surfaces including screw heads and electrical contacts.

The illumination system consisted of two Aputure Light Storm 600d Pro LED fixtures which produced uniform glare-free light. The fixtures were placed at a height of 1.2 m above the sample plane while being directed at ±35° off-axis and operating at 50% power output. The high CRI daylight spectrum of these lights reduced color variations in RGB frames while reducing specular reflection artefacts in the depth sensor.

The Mech Eye Viewer “calib” wizard was used to perform camera calibration before starting data collection. The software processed multi-view checkerboard images to obtain full intrinsic matrices and distortion coefficients as well as the robot-relative hand–eye transformation. The final reprojection error stayed below 0.2 pixels throughout the process which allowed sub-millimetre alignment between RGB and depth channels.

The acquisition process generated three data streams simultaneously from each camera system:

a) a 24-bit RGB image (.png, 4016 × 3036 px, via on-sensor upsampling)

b) a single-channel depth map (.tiff, 2048 × 1536 px)

c) a textured point cloud (.ply)

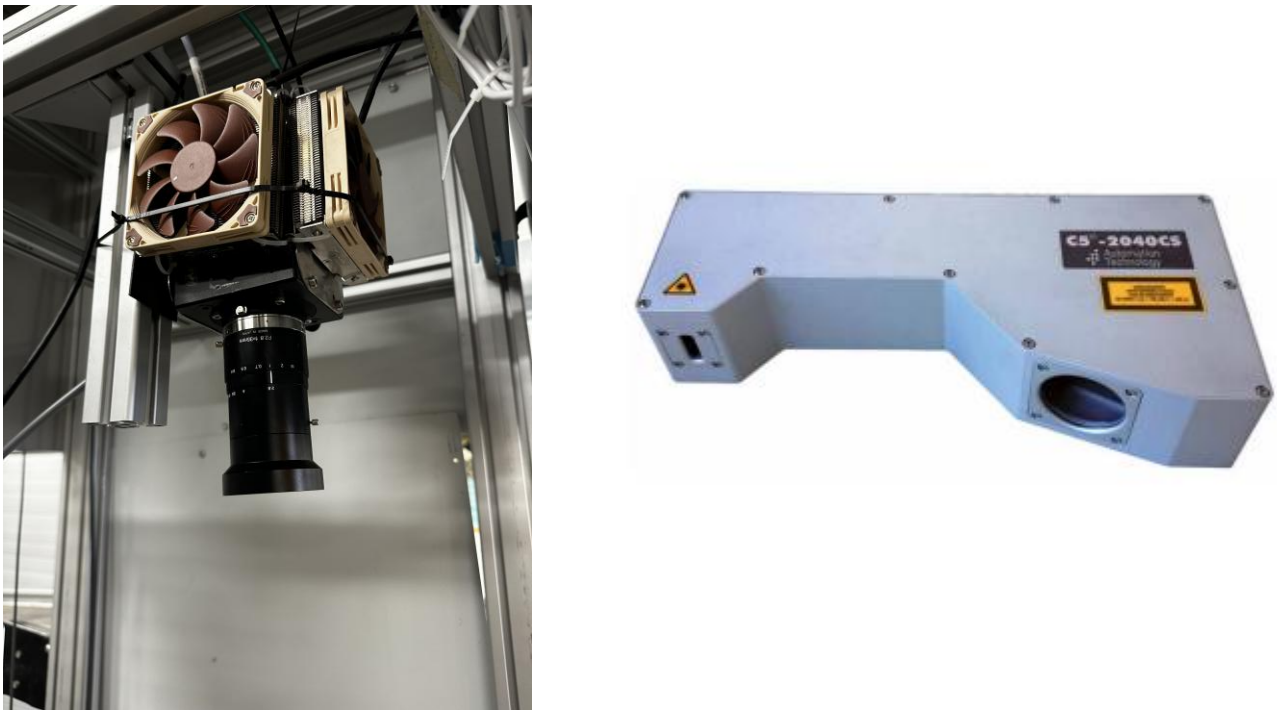
The LSR S depth sensor operated with dual exposure times of 88 ms and 24 ms to detect dark ABS battery casings. The exposure time for reflective aluminium casings was reduced to 60 ms and 24 ms and the RGB exposure time was reduced to 8 ms to prevent blooming. The depth sensing range was limited to 0.10–1.00 m to match the operational space of the robot.

Table 3-1: Summary hardware setup – e-bike battery

Component / setting	Value
Robot	Stäubli TX2-60, 6 DOF
Cameras (RGB-D)	Mech-Eye Pro-S (1920 × 1200 RGB, 0.1 mm @ 1 m) & Mech-Eye LSR-S (2048 × 1536 depth, 1 mm @ 1.5 m)
Viewpoint trajectory	Top-down + two 180 ° arcs (X & Y axes) sampled every 30 ° → 8 views
Illumination	2 × Aputure 600d Pro LEDs, 50 % power, 35 ° off-axis, height ≈ 1.2 m
Exposure settings (LSR-S)	Black ABS: 3D 88 ms + 24 ms, 2D 20 ms
Depth range	0.10 – 1.00 m
Captured modalities	RGB (.png, 4016 × 3036), depth (.tiff, 2048 × 1536), textured point cloud (.ply)

3.1.2 Hardware setup capturing scaffolding pipe

The imaging setup consisted of a conveyor system with a vision box mounted overhead, designed to capture both visual and structural features of scaffolding pipes as they moved beneath.



(a) (b)

Figure 3-e: (a) SW-4000Q-10GE 4-CMOS prism-based line scan camera (b) C5-2040cs 3D sensor

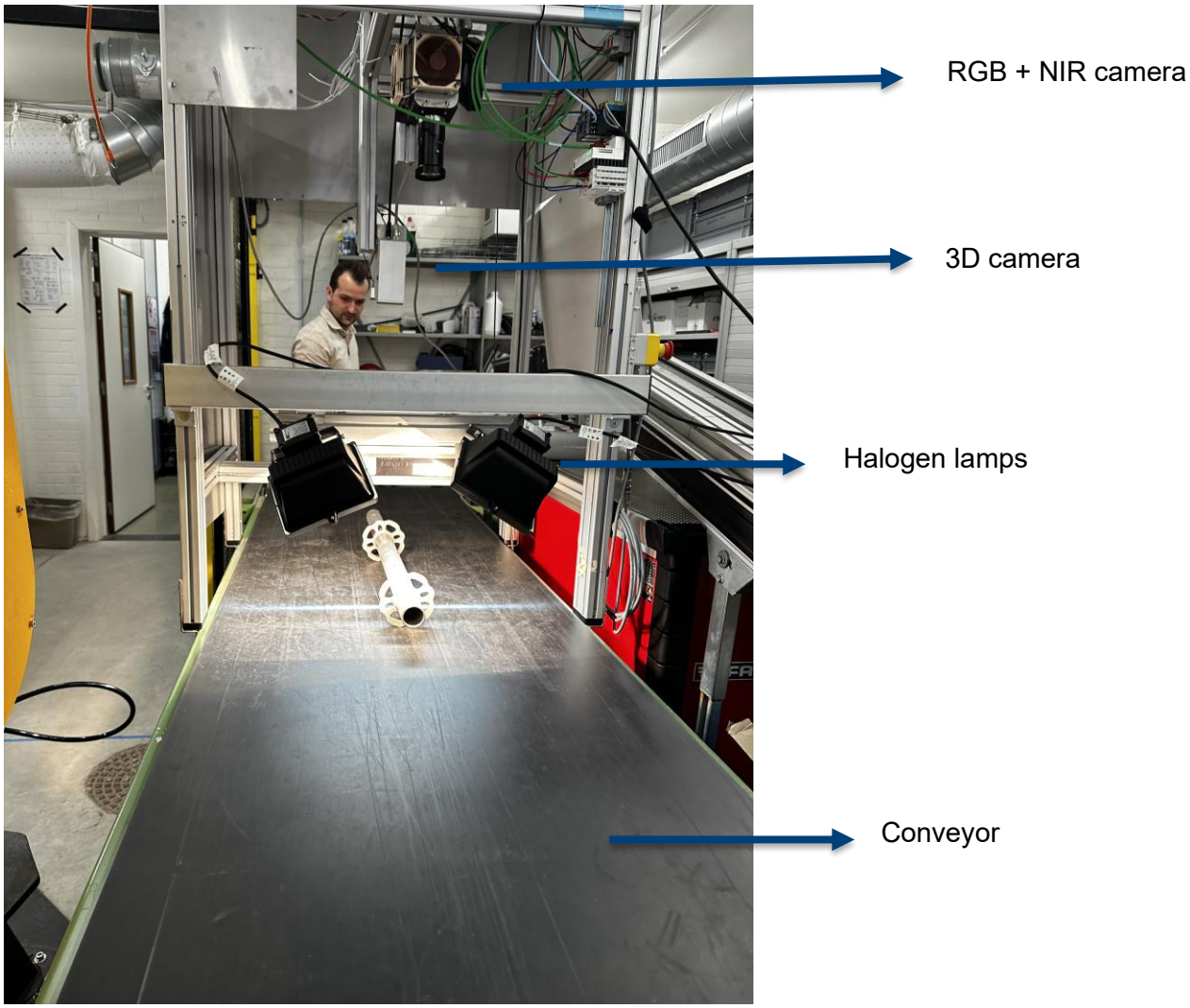


Figure 3-f: Pipe image capturing setup

Two cameras were used in the configuration: a SW-4000Q-10GE 4-CMOS prism-based line scan camera capable of RGB and Near-Infrared (NIR) imaging, and a C5-2040(-4M*)-GigE sensor responsible for acquiring 3D line-scan data for geometric profiling.

To determine the optimal lighting setup for NIR imaging, the spectral response of the SW-4000Q-10GE camera was first analysed. The response curve showed its highest relative sensitivity of 0.6 at 800 nm thus making this wavelength the most suitable for achieving the best signal quality. Halogen parking lamps were selected as an initial prototyping solution due to their broad spectral emission, which includes 800 nm, and their significantly lower cost compared to industrial NIR line lights.

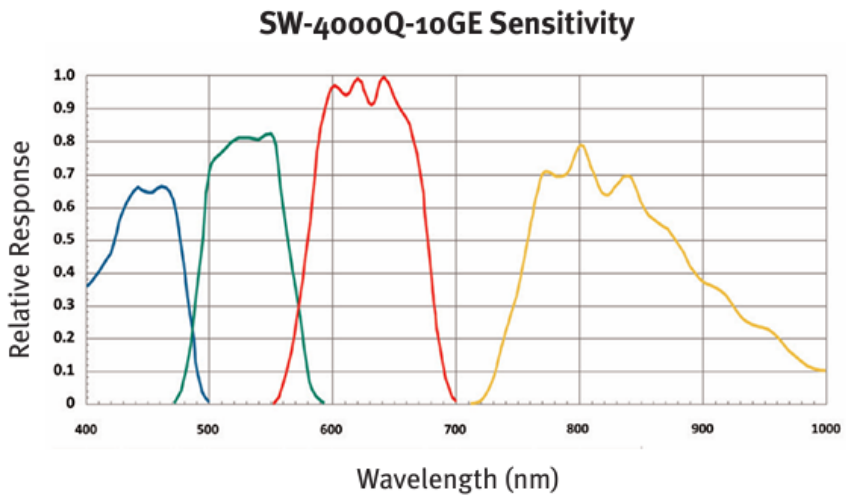


Figure 3-g: RGB+NIR camera spectral response

A lux meter smartphone application measured the illumination patterns on the conveyor belt while manual measurements of light intensity were taken at different angles. The physical light intensity measurements received software validation through the image acquisition system which recorded per-frame statistics about minimum and maximum and average NIR pixel intensities to detect illumination irregularities in acquired data. The results from these two methods were used to make sequential adjustments to the position and direction of the lamps. Multiple test runs showed that positioning the halogen light beam at 40° relative to the sample surface produced the highest NIR reflectance while minimizing shadows and glare. The metallic and coated pipe surfaces likely exhibit specular reflection characteristics which follow a combination of Lambertian and specular reflectance models.

This angle may be empirically modelled using a simplified reflection relation:

$$I(\theta) = I_0 \cdot \cos^n(\theta - \theta_{opt})$$

- $I(\theta)$: observed intensity at angle θ
- I_0: peak intensity
- θ_{opt}: empirically determined optimal angle (≈ 40°)
- n: controls the spread of the intensity falloff

The lighting was adjusted prior to calibrating other imaging parameters that influence scan quality, particularly those affecting motion blur. Motion blur occurs because of the relative movement between the object and the camera during exposure time and can be calculated as:

$$B = v \cdot t_{exp}$$

- B is the blur length (e.g., in mm or pixels)

- v is the conveyor belt speed (m/s)
- t_{exp} is the exposure time (s)

It was observed that even small increases in conveyor speed significantly amplified motion blur at longer exposure times. Therefore, optimal acquisition required joint tuning of:

- Conveyor speed
- Exposure time
- Gain settings (to compensate for shorter exposures)
- Camera acquisition rate

Controlled tests were conducted across varying speeds and exposure settings to map out safe operating regions that preserved visual clarity without overexposure or loss of frame sharpness.

3.2 Datasets & Capture Pipelines

3.2.1 E-bike battery capture pipeline

The e-bike-battery dataset used in this study was inherited from an earlier master's thesis; only a concise recap is given here, as the full hardware specifications appear in § 3.3.2. Data were acquired with a Stäubli TX2-60 robot carrying co-aligned Mech-Eye Pro-S and Mech-Eye LSR-S RGB-D cameras. A Python control script positioned the robot at eight fixed poses per battery: an initial true top-down view followed by the remaining cardinal and diagonal directions. This ensured that every external face, screw head and electrical contact was imaged at least once.

At each pose the cameras simultaneously recorded a 24-bit RGB frame, a registered depth map and a textured point cloud. Illumination came from two high-CRI LED panels mounted above the sample plane (see Fig. 3.3 b); exposure and integration settings were adjusted per battery colour to avoid saturation on reflective aluminium housings.

Intrinsic calibration of both cameras—and depth-to-colour alignment via the vendor's calib routine—was performed offline. The final dataset comprises ≈ 1 100 RGB-D image sets (≈ 137 physical batteries at eight views each) together with their point clouds. Labels inherited from the original project mark screw presence/absence and surface-defect regions. These multi-view, multi-modal assets provide the raw input for both Branch A (3-D geometric alignment) and Branch B (2-D defect localisation and segmentation); the train/validation/test partitioning is detailed later in § 3.7.2.

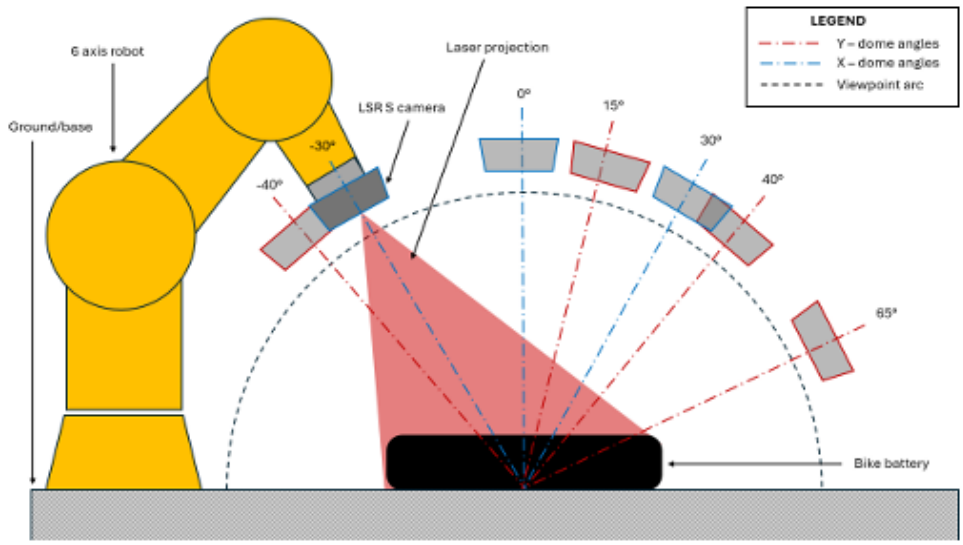


Figure 3-h: Graphical representation of capture viewpoints for e-bike battery dataset (MATHEW & CHHABRA, n.d.)

3.2.2 E-bike battery capture examples

(a) (b)

Figure 3-i: (a) RGB images of various e-bike batteries (b) corresponding single channel depth images

3.2.3 Pipe capture pipeline

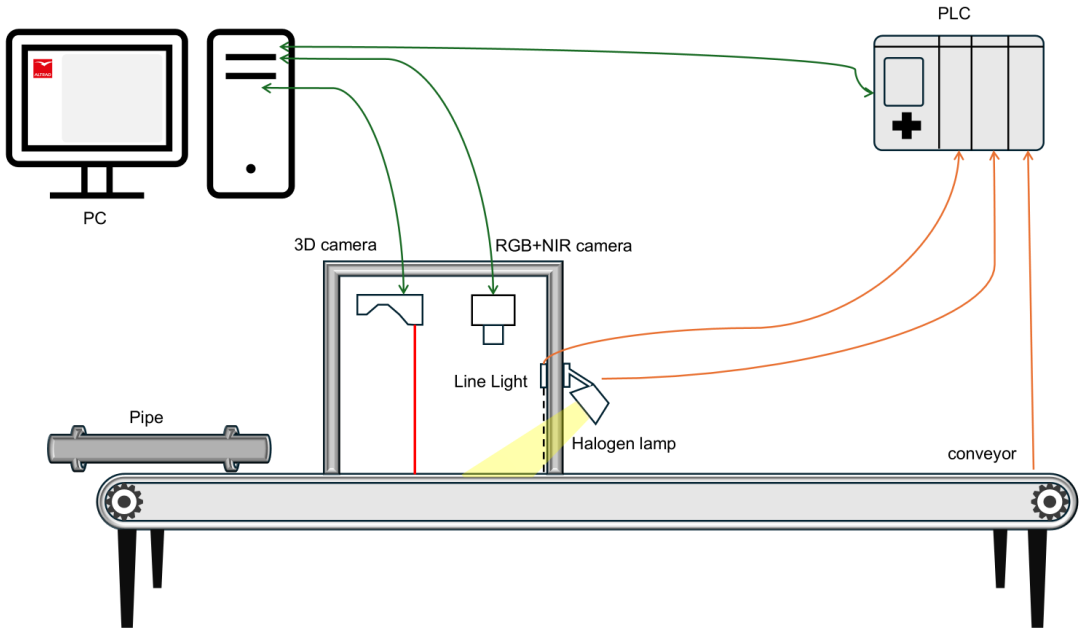


Figure 3-j: Pipe dataset capturing workflow

3.2.3.1 Automated frame acquisition

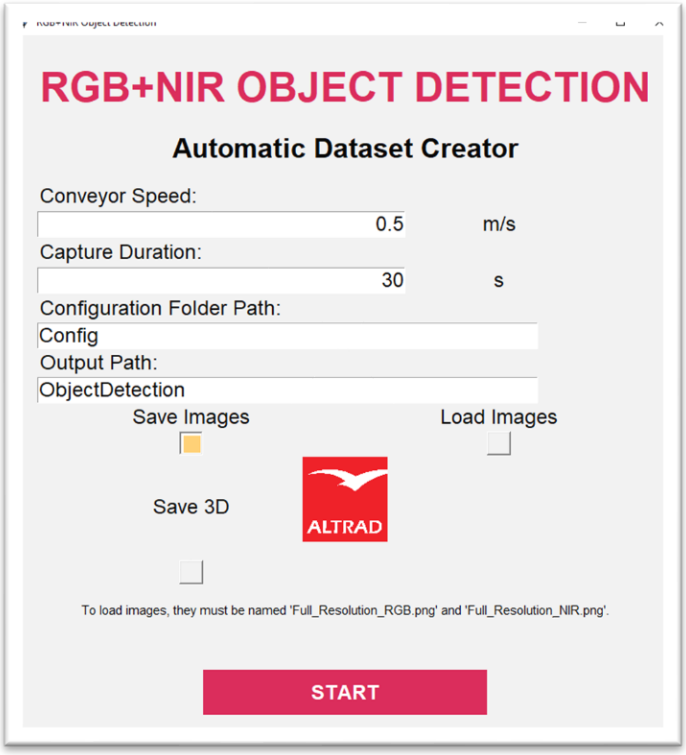


Figure 3-k: Pipe dataset creator GUI

Once the operator confirms conveyor speed and capture duration on the python front-end library Tkinter, the application spawns two child processes. The first streams RGB and NIR frames from a dual-wavelength line-scan sensor; the second is reserved for depth capture but remains inactive for the pipe study. Each slice pair is pushed into shared memory; once the timer expires, the parent process stacks the slices vertically into two full-height mosaics. One RGB and one NIR. Because both bands share the same prism optics, the mosaics are inherently pixel-aligned and need no post-hoc registration.

3.2.3.2 Line-Scan Assembly and Archival

The stitched RGB and NIR images are saved as PNG files named Full_Resolution_RGB.png and Full_Resolution_NIR.png in a run-specific output folder. If the operator disables saving, the images remain in memory for immediate cropping; if saving is enabled, the mosaics provide a verifiable audit trail for data provenance. No external metadata files are generated at this stage because geometric information is implicit in pixel coordinates.

3.2.3.3 Object Segmentation and Cropping

Following assembly, pipe segments are isolated using the extract_objects_from_linescan() routine. This method partitions the image into overlapping horizontal segments to manage memory and computational load. A composite binary mask is generated for each segment by

combining three complementary cues: HSV colour thresholds tuned to typical pipe surface characteristics, pixel intensity deviation from the black conveyor background, and dilated Canny edges to enhance low-contrast boundaries. The resulting mask is refined using morphological closing and opening operations. Contours with an area exceeding 5 000 px² are retained. Bounding boxes that are vertically proximate (less than 100 pixels apart) and horizontally overlapping are merged to avoid redundant crops. The final bounding boxes are padded by 20 pixels on all sides and applied to both RGB and NIR scrolls to produce aligned modality pairs.

3.2.3.4 Crop Storage and Pairing

Crops are saved as lossless PNG files into rgb_crops and nir_crops sub-folders, with a common stem *object_<timestamp>_<idx>_{rgb|nir}.png*. For visual sanity checks a side-by-side composite is written to paired_crops. The routine also exports a binary mask montage and a bounding-box overlay of the full mosaic, supporting qualitative inspection during dataset curation.

3.2.3.5 Dataset Statistics

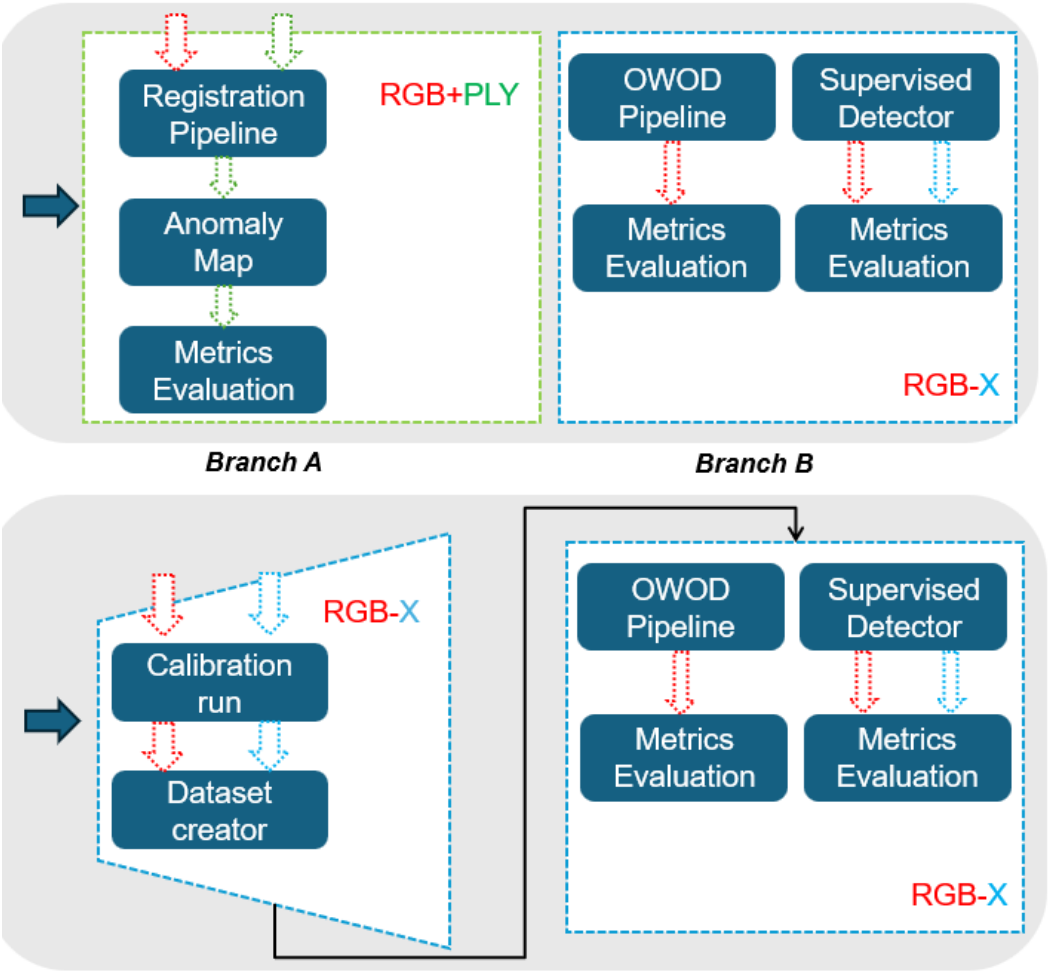
A typical 30-second acquisition at a conveyor speed of $0.5\,\mathrm{m\,s^{-1}}$ yields scroll images approximately 2 448 × 42 000 pixels in size. In total, seven such runs were performed, resulting in 45 paired RGB–NIR samples. Although only 16 physical pipes were available for capture, each was recorded multiple times by rotating it between acquisitions. Pipes equipped with flanges allowed for four distinct and stable orientations, other pipes supported only two. This rotational augmentation enabled a broader sampling of surface conditions and significantly expanded the dataset. Each paired sample has either a “short length” with a length between 50 cm and 100 cm or has a “medium length” with a length between 100 cm and 160 cm. Class-balance metrics and train/validation/test partitioning are presented later in § 3.7.2.

3.2.4 Pipe capture examples

Figure 3-I presents high-resolution images (to ~700 × 5000 pixels) of captured scaffolding pipes. Pipes (a) and (b) are clean, while (c) and (d) exhibit substantial mortar residue along their length.

3.3 Overview

3.3.1 Proposed hybrid pipeline: architecture and guiding principles



The proposed inspection framework in this thesis operates as a modular system which accepts different sensor configurations through a uniform processing architecture. The system accepts various hardware configurations including RGB-D cameras used in e-bike battery inspection and RGB + NIR cameras used in pipe inspection which feed into a single processing pipeline. The processing stages of background removal and detection and segmentation and evaluation maintain a unified structure despite the different image acquisition hardware systems.

The framework contains two main visual analysis branches. The first branch uses task-specific trained models for detection and segmentation through supervised learning. The OWLv2 generates region proposals while SAM 2.1 produces segmentation masks in the second

OWOD pipeline which operates without prior knowledge of new objects. The two branches use identical datasets for processing and share the same evaluation metrics for RGB and NIR inputs to ensure consistent performance assessment.

The e-bike battery use case includes a dedicated 3D processing branch. The system executes geometric registration and anomaly detection through an evaluation module which uses shape-based metrics. The framework maintains a unified structure that enables flexible extension for industrial inspection applications across different settings.

3.3.1.1 Branch A – 3-D geometric analysis

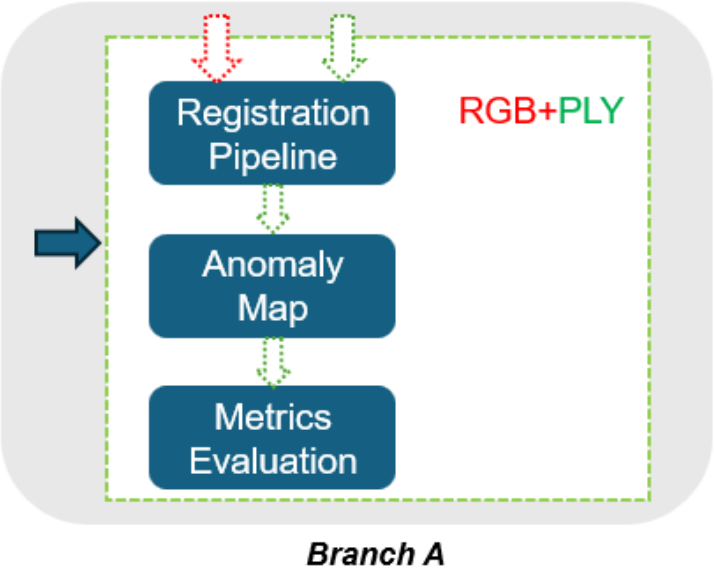


Figure 3-m: Branch A of the developed software pipeline

The proposed inspection framework includes Branch A which conducts 3D geometric analysis for applications requiring exact shape and structural conformity. The e-bike battery inspection utilizes a dual Mech-Mind RGB-D camera setup to capture RGB imagery and dense 3D point clouds of battery units.

The first step involves background removal through HSV-based rule thresholds to classify images as either black or coloured battery representations. The classification result determines whether Otsu's method or adaptive thresholding will be used for segmentation followed by morphological refinement to clean the resulting mask. The refined mask receives projection onto the 3D point cloud through known camera intrinsics and extrinsics to extract only the relevant 3D surface data.

The filtered point cloud requires two registration stages. The first step involves applying a transformation matrix to achieve a basic alignment between the scan data and its reference

model. The ICP algorithm performs fine-grained alignment refinement through point-to-point and point-to-plane variants. The alignment quality and geometric accuracy are evaluated through RMSE and mean distance and standard deviation and maximum deviation calculations before and after refinement.

Following registration, the system performs anomaly detection via differential alignment. The system uses a registered "good" battery point cloud as reference to align with a registered defective battery point cloud through an additional ICP refinement step. The system analyses the residual differences between two-point clouds to produce an anomaly detection map which shows areas of geometric mismatch. The maps function to detect missing screws by showing localized voids or geometric deviations relative to the reference model. The maps provide interpretable outputs which enable operators or automated systems to detect structural incompleteness or omissions with high spatial precision for maintaining mechanical integrity in battery assemblies.

3.3.1.2 Branch B – 2-D zero-shot defect localisation and segmentation

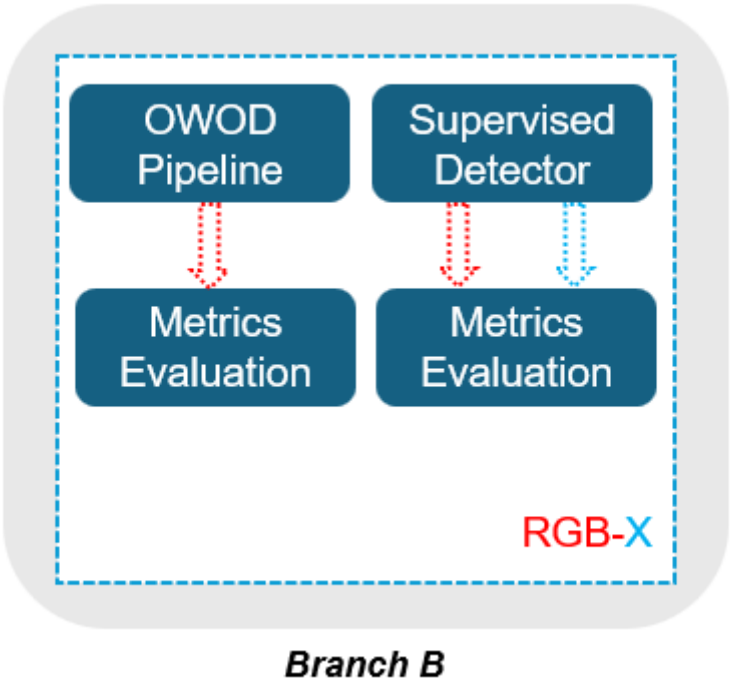


Figure 3-n: Branch B of the developed software pipeline

Branch B of the inspection framework performs 2D defect localisation and segmentation using zero-shot and supervised learning techniques. This branch is applied in the e-bike battery inspection case and is designed to identify and segment surface-level defects, such as missing screws.

The core pipeline begins with a high-resolution RGB image of the part under test, passed to OWLv2, an open-world vision–language model that uses text prompts (e.g., “missing screw”) to detect potential defect regions in a zero-shot fashion. Trained on large-scale image–caption datasets, OWLv2 requires no additional fine-tuning to detect novel defect types, making it ideal for flexible deployment in dynamic inspection environments.

OWLv2 often produces redundant or overlapping bounding boxes, which are refined through a two-step post-processing pipeline:

1. Weighted Box Fusion merges correlated boxes into consensus regions.
2. Per-class Non-Maximum Suppression removes any residual duplicates.

The cleaned bounding boxes are passed to SAM 2.1 (Segment Anything Model), which reuses a cached image embedding to generate pixel-accurate segmentation masks in real time. These masks are further filtered by shape, retaining only circular masks, based on the geometric prior that screws appear circular in the inspection view.

To benchmark zero-shot performance, Branch B includes an additional configuration. All supervised models of the yolov11 family are trained using the complete labelled dataset. All models are evaluated on the same RGB test set using consistent metrics to ensure a fair comparison.

This pipeline is extended to RGB + Depth input for the e-bike battery dataset. The depth channel is concatenated with RGB via an early-fusion strategy, requiring only minor changes to the model’s input layer. The same SAM 2.1 module is reused for all cases, ensuring that segmentation differences are attributable to detection quality and input modality rather than segmentation design.

While the full detection and segmentation pipeline is applied only to the e-bike battery case, the pipe inspection task uses Branch B in a classification-only configuration. Here, the model determines whether a pipe region contains a defect or not, based on either RGB or fused RGB + NIR input.

Branch B, therefore, offers a structured framework for exploring the trade-offs between supervision levels, input modalities, and inference capabilities, especially in contexts where data annotation is costly or impractical.

3.3.1.3 Guiding design principles

The entire architecture depends on four fundamental principles. The use of text- and box-prompted models enables promptability and rapid adaptation while preventing the need for expensive relabelling when the defect definition changes. The second principle uses sensor fusion to combine RGB for color and texture information with NIR for chemical contrast enhancement and depth for geometric encoding. The system maintains scalability through

loose module connections which enable OWLv2 to replace Grounding DINO and ICP to replace feature-based registration without requiring extensive code modifications. The experimental design includes comparative validation through the use of the identical pipeline on two different datasets to show that the approach works across different products rather than being limited to a specific product.

The inspection strategy achieves balance through the combination of geometric and photometric branches. The unified software platform enables real-time operation and easy product transition for both Branch A's metric shape analysis and Branch B's flexible appearance-based segmentation.

3.4 Pre-processing & Data Augmentation

Before any learning stage, each dataset passes through a light pre-processing pipeline. For the bike-battery images this is limited to background removal. The dataset is already large enough that no further augmentation is required. For the pipe images a similar background-removal logic applies, but the small sample count makes extra augmentation essential. Classical 2D transforms were therefore incorporated, along with the multi-channel image slicing described in § 3.5.2.

3.4.1 Background Removal & Classical Augmentations

- Bike batteries RGB frames are classified in HSV space to distinguish black from coloured casings. Black batteries undergo Otsu thresholding; coloured ones use adaptive thresholding. Both masks are refined morphologically and then projected onto the depth map to excise non-battery points (details in § 3.9).
- Scaffolding pipes — Each image is segmented by combining HSV thresholds, intensity deviation from the conveyor background, and dilated Canny edges; small artefacts are removed with closing/opening. After masking, standard photometric augmentations (random HSV jitter, horizontal flip, scale 0.5–1.5) are applied during YOLO training exactly as listed in Table 3-2.

3.4.2 Four-Channel Image Slicing (Custom SAHI Adaptation)

3.4.2.1 Motivation

The scaffolding-pipe dataset contained only 45 paired RGB + NIR scroll images, insufficient for robust YOLOv11 training. Off-the-shelf SAHI could not help because it is hard-wired to 3-channel RGB. To expand the data volume while preserving spectral information, a lightweight slicer was created to cut each 4-channel TIFF into overlapping 640 × 640 windows (50 % stride in both axes).

3.4.2.2 Implementation snapshot

The slicing script loads each 4-channel TIFF using the tifffile library, which supports multi-page and multi-channel formats without loss of metadata. A fixed 640 × 640 pixel sliding window is computed across each scroll image, using a 50 % overlap in both horizontal and vertical directions. For each window position, a crop is extracted that preserves all four channels (RGB + NIR) and saved to disk as a new TIFF file. This layout ensures compatibility with early-fusion YOLO models and allows the auxiliary NIR channel to remain fully aligned with the RGB input. The slicing function supports both channel-last and channel-first formats and writes tiles with filenames that retain traceability to their parent image and window index.

COCO annotations are processed in parallel with the slicing. Each slice is paired with the subset of annotations whose bounding boxes or segmentation polygons intersect the crop region. Bounding boxes are clipped to the slice boundaries, and segmentation polygons are trimmed using shapely to avoid shape distortion. After clipping, any annotation whose retained visible area is less than 10 % of the original is discarded to prevent label noise. Unlike SAHI, which uses object-oriented abstractions (e.g., CocoAnnotation, SliceImageResult), this script modifies COCO dictionaries directly using functional logic. This method can be extended to data that add even more channels than the 4 channels used in this research.

3.4.2.3 Dataset gain

Slicing turned the 45 longer images into 1455 tiles, of which 375 carry at least one valid annotation. The tiles were repartitioned with exactly the same 60 / 20 / 20 train–val–test ratios used elsewhere, ensuring statistical consistency with the original dataset before training the detectors described in § 3.7.

3.5 Point-Cloud Registration & 3-D Analysis (E-bike battery)

3.5.1 End to end point cloud registration

For point cloud registration, the point clouds are first organised into a designated folder:

```
├── Good/
│   ├── PLY
│   ├── RGB
│   └── Registered
└── Missing/
    ├── PLY
    ├── RGB
    └── Registered
```

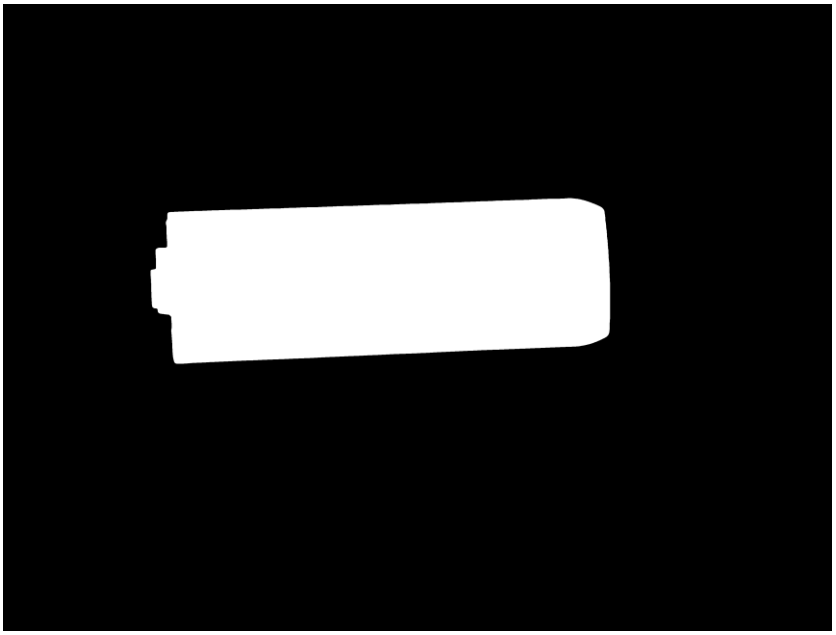
Once this step is completed, the process proceeds to the implementation phase of the point cloud registration algorithm.

To enable efficient and accurate multi-viewpoint cloud registration, a comprehensive pipeline is implemented that integrates deep learning–based segmentation, geometric filtering, and optimized alignment techniques.

The initial step involves assigning each unprocessed 3D point cloud data to its matching RGB image. The ISNetDIS (Dichotomous image segmentation) (Qin et al., 2022) deep segmentation model produces binary segmentation results to identify battery objects from other background elements. The GPU executes the segmentation process at half-precision inference levels to achieve maximum performance. The following mask processing includes morphological operations to remove noise and close small holes before creating an object boundary. A pinhole camera model allows 3D cloud points to be projected onto 2D image space for further processing. The filtering process maintains only points that project inside the segmented area which results in an object-centric point cloud that reduces data volume and enhances relevance.

(a)

(b)



(c)

Figure 3-o: (a) Raw RGB image of e-bike battery (b) Filtered RGB image using DIS (c) Binary mask

The following stage starts the registration process by selecting a fixed reference point cloud from the filtered results. The remaining clouds receive their initial alignment through predefined 4×4 transformation matrices that incorporate sensor pose information. The matrices enable the achievement of a basic alignment between point clouds within a common coordinate framework. The ICP algorithm executes fine alignment procedures. Two ICP methods exist for optimization: The first uses relaxed convergence criteria for speed optimization and the second method uses CUDA acceleration for fast convergence on dense point clouds. The chosen method for alignment depends on RMSE values and fitness scores as well as execution time measurements. A registration approach through Fast Global Registration (FGR) using FPFH features becomes active when both ICP methods fail to produce acceptable outcomes. All view-specific processing tasks including filtering and registration occur simultaneously to decrease the total execution time and enhance hardware performance. The registration process completes by uniting all filtered and registered point clouds into one unified model. Once registration is complete and the results have been saved in their respective folders, these outputs are used to generate the anomaly map. A voxel down sampling operation serves as an optional step which optimizes storage and visualization capabilities while maintaining geometric accuracy in the final output.

3.5.2 Anomaly Map

The workflow includes point cloud preprocessing, registration, distance-based deviation analysis, percentile-driven anomaly segmentation, and screw-focused visualization. For each point cloud, spatial dimensions are calculated, and normal are estimated using a radius-based hybrid search. The search radius r is used for estimating normals is dynamically scaled based on the size of the input point cloud. It is computed as a fixed proportion (5%) of the largest axis-aligned bounding box dimension of the point cloud. The formula is:

$$r = 0.05 \times \max(x_{\max} - x_{\min},\ y_{\max} - y_{\min},\ z_{\max} - z_{\min})$$

Where,

- $x_{\max}$ and $x_{\min}$ are the maximum and minimum coordinates along the **X-axis**,
- $y_{\max}$ and $y_{\min}$ are the maximum and minimum coordinates along the **Y-axis**,
- $z_{\max}$ and $z_{\min}$ are the maximum and minimum coordinates along the **Z-axis**,

The defective cloud is aligned with the reference using a multi-scale Iterative Closest Point (ICP) algorithm. A coarse-to-fine strategy is adopted, starting with a larger correspondence threshold and reducing it progressively to refine the transformation. ICP computes the rigid

transformation T that minimizes the squared Euclidean distances between corresponding points in the two clouds:

$$T = argmin_T \sum_{i=1}^{N} \|p_i - T \cdot q_i\|^2$$

Here, p_i and q_i represent the matched points in the reference and defective clouds, respectively. To evaluate registration performance, two metrics are used: **fitness**, defined as

$$\text{Fitness} = \frac{|\text{Correspondences with distance} < d_{\max}|}{|\text{Total points in source point cloud}|}$$

and **RMSE (Root Mean Square Error)**, calculated as

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} ||p_i - q_i||^2}$$

Where,

- N = total number of matched point pairs
- p_i= point iii in the source (defective) point cloud
- q_i= corresponding point in the target (reference) point cloud
- $||\boldsymbol{p_i} - \boldsymbol{q_i}||$= Euclidean distance between the two points

High fitness (close to 1) and low RMSE confirm accurate alignment. Following registration, the method computes point-wise deviations by measuring the Euclidean distance between each point in the test cloud and its nearest neighbour in the reference:

$$d = ||\boldsymbol{p_i} - \boldsymbol{q_i}|| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Where:

- $\boldsymbol{p_i}$ is a point in the defective cloud
- $\boldsymbol{q_i}$ is its closest neighbour in the reference cloud
- d is the deviation at point

3.6 OWOD + SAM Zero-Shot Detector

3.6.1 OWLv2 configuration and text-prompts

To enable zero-shot object detection within the product condition evaluation pipeline, the OWLv2 model (google/owlv2-base-patch16-ensemble) was utilized. This model supports open-world detection by linking visual inputs with text-based queries using a vision-language transformer. The Owlv2Processor was used to preprocess both the images and the natural language prompts, while Owlv2ForObjectDetection performed the detection.

In this implementation, each input image—was processed with the prompt ["a metallic screw"] for the e-bike battery dataset and ["mortar"] for the scaffolding pipes. This guided OWLv2 to focus only on relevant object categories without requiring explicit training on the dataset, thereby enabling flexible and scalable defect detection across varied scenarios.

3.6.2 Bounding-box to SAM-prompt conversion

Once the zero-shot detector has delivered its bounding boxes, the pipeline first guarantees that every box is expressed in absolute pixel coordinates. If the detector supplies another format, the coordinates are rescaled internally so that x values refer to image columns and y values to rows. Each box is then padded by a small margin—about three percent of the larger image dimension—to provide SAM 2.1 with a little contextual information beyond the object boundary; the expanded rectangle is finally clamped to the image frame.

The padded rectangle is forwarded to SAM 2.1 as a box prompt, and the box centre can be sent as a single positive point when extra spatial guidance is beneficial. All prompts belonging to the same image are batched so that SAM2ImagePredictor processes every object in one forward pass. SAM 2.1 can return several candidate masks per prompt together with stability scores. Masks scoring below 0.90 are rejected immediately; if multiple masks survive, the one covering the largest number of pixels is kept. The outcome is a single, high-confidence binary mask for each detection, already aligned with the original image grid and traceable back to its source box.

3.6.3 Post-processing of SAM masks

The binary masks that emerge from SAM-2 can now follow either—or both—of two parallel post-processing paths.

On the export path, each mask is translated into COCO's polygon format by tracing its outer contour with OpenCV and flattening the coordinates into the list structure mandated by the standard; contours below a minimum area threshold are discarded so they do not pollute the dataset. The corresponding bounding box is re-expressed as [x, y, width, height], the pixel area is measured directly from the mask, and a category ID is retrieved from the prompt-to-

label table (e.g. “metallic screw” or “mortar”). These elements-polygon, box, area, category and the detector’s confidence are then written to a COCO-compliant JSON file.

On the visualisation path, the same masks and bounding boxes are blended onto the original image, or written out as single-channel PNGs, to provide an immediate qualitative check on segmentation quality. No additional geometry processing is needed because both paths consume the same intermediate data structures. Either route can be activated independently: one might generate only the overlaid images for quick sanity checks, only the COCO file for benchmarking and training, or both when a complete record is desired.

3.7 Supervised Trained Detector - YOLO Fine-Tuned

3.7.1 Motivation Reiteration

Although the primary detection pathway leverages open-world capabilities for maximum flexibility, preliminary experiments revealed a precision drop on the very domain-specific images of the bike batteries. To guarantee reliable mask generation in cases where the open-world detector fails to capture more subtle defects, a lightweight, fully supervised fallback is deployed: a YOLOv11-based model fine-tuned on a labelled dataset specific to the application. The choice is pragmatic—previous sections (§ 3.1 and § 3.2) already outlined the motivation, but briefly, YOLO offers rapid convergence, mature training code, and real-time inference, and is relatively easy to train.

3.7.2 Dataset Splits

```
├─── images/
│    ├─── train-550 images
│    ├─── test-100 images
│    └─── val-270 images (strictly unused until final evaluation)
└─── labels/
     ├─── train-550 labels
     ├─── test-100 labels
     └─── val-270 labels (strictly unused until final evaluation)
```

- Training set**:** 550 images
- Validation set**:** 100 images
- Test set**:** 270 images (strictly unused until final evaluation)

The dataset contains multiple battery types, colours, and viewing perspectives, which have been evenly distributed across the three splits (train/val/test) in proportion to their respective sizes. Each subset contains a comparable number of detectable screws, which are the primary labelling target. The annotations are in standard YOLO bounding box format (<class> <x_center> <y_center> <width> <height> in relative coordinates). These were obtained by converting segmentation masks—originally provided in YOLO segmentation format (<class> <x1> <y1> <x2> <y2> ... <xn> <yn> for polygon points)—into bounding boxes.

3.7.3 Training recipe e-bike battery

All five YOLOv11 variants (n, s, m, l, x) were trained to assess the trade-off between model complexity and accuracy. Both the RGB and RGBD models were initialized with no pre-trained weights.

Table 3-2: Training hyperparameters

Parameter	Value	Rationale
Base weights	yolov11{n,s,m,l,x}.pt	Evaluate across lightweight to large models
Epochs	125 (early-stop patience = 15)	Stable convergence across variants
Batch size	16	Balanced across GPU RAM limits (adjusted if needed)
Optimizer	SGD, momentum 0.937, weight decay 5×10^{-4}	Default YOLO optimizer
LR schedule	One-cycle, max LR 0.01	Effective for fast convergence
Augmentations	Mosaic 0.7, random HSV, horizontal flip 0.5, scale 0.5-1.5	Improve generalization

3.8 Multichannel Fusion Implementation

For both industrial scenarios, scaffolding-pipe inspection and bike-battery quality control, each YOLOv11 variant was trained twice: once in the canonical three-channel configuration and once with an expanded four-channel front end that ingests either an additional NIR or depth plane.

3.8.1 Early Fusion Layer Modification

The Ultralytics fork used in §3.6 was patched to support an arbitrary number of input channels (cf. pull request ultralytics/ultralytics #20223). Concretely, the first convolution of each backbone variant—n, s, m, l, and x—was re-parameterised from C_in = 3 to C_in = 4 while preserving the original kernel spatial dimensions and output width. This microscopic alteration adds exactly one extra kernel slice per output feature map; parameter counts rise by <0.03 % and the theoretical GFLOPs budget by ≤0.2 % (YOLOv11-n: +144 params, 0.0 GFLOPs; YOLOv11-x: +864 params, +0.2 GFLOPs). All subsequent layers remain byte-for-byte identical to their RGB counterparts, safeguarding architectural comparability.

Ultralytics initialises convolution weights according to its default He-style (Kaiming) scheme. Because no pre-trained four-channel checkpoints exist, models start from scratch: the three legacy kernel slices receive random values drawn from the same distribution as the new fourth slice. During preliminary trials the additional channel caused slower early convergence but had negligible effect on final accuracy once training reached steady state.

The dataloader was upgraded in concert with the backbone patch. It now decodes multi-page TIFF files and assembles a four-channel tensor on the fly, leaving the augmentation pipeline unchanged except for adding support for this multi-channel adaptation. Every geometric transform applied to the RGB stack is applied identically to the auxiliary slice; colour-space jitters (HSV) naturally affect only the visible channels.

3.8.2 Dataset Configurations

To ensure apples-to-apples evaluation, the train/validation/test splits defined in §3.6.2 were retained verbatim. Each original JPEG/PNG image of each pipe was paired with an auxiliary modality captured simultaneously by the same physical sensor module. The vendor SDKs deliver intrinsically registered output, so no post-hoc alignment was required.

All three-channel samples continue to reside on disk as ordinary RGB images. Their multispectral twins are stored as single four-channel TIFF files. The dataloader automatically falls back to the RGB asset when only three channels are requested.

3.8.3 Training Protocol Parity

Except for the enlarged input tensor, every hyper-parameter mirrors the recipe in §3.7.3: 125 epochs with early-stopping patience 15, one-cycle learning rate (max 0.01), SGD optimiser (momentum 0.937, weight-decay 5×10^{-4}), batch size 16, and the same augmentation suite (Mosaic 0.7, random HSV, horizontal flip 0.5, scale 0.5–1.5). Holding the schedule constant isolates the effect of the extra modality on both learning dynamics and end-point accuracy. Empirically the four-channel models required roughly 10–15 additional epochs to converge, consistent with their random initialisation, but no adjustment to the early-stopping criterion was necessary in practice.

The resulting model zoo—ten detectors per dataset, spanning five capacities and two different input modalities—forms the basis for the experiments.

3.9 Design Decisions & Rationale

The preceding sections have described what the hybrid inspection pipeline does. This part explains why each constituent block was chosen. The overarching design brief was to achieve a practicable equilibrium between three sometimes-competing objectives: (i) high defect–detection accuracy, (ii) adaptability to new product types or sensing modalities, and (iii) feasibility within the limited time available for completing this thesis. Because the thesis itself is bounded by a fixed completion date, every architectural decision also had to respect the practical limits of implementation effort.

The logic that connects candidate methods to final selections therefore pivots on a four-way trade-off—accuracy, runtime, practical engineering effort, and the calendar time available for thesis work. Whenever quantitative evidence clearly discriminates between alternatives, accuracy is prioritised; in cases where performance margins are narrow, preference is given to methods that satisfy processing demands while maintaining reasonable implementation speed. The discussion that follows proceeds stage by stage through the pipeline, justifying each major choice in turn and noting the principal alternatives that were rejected. A consolidated overview appears at the end of the chapter (Table 3-3).

3.9.1 Open-World Detection

The first component of Branch B (§ 3.1.2.2) must localise previously unseen defect classes from a single textual description or bounding-box cue. This requirement arises from the need for adaptability to different objects or defect types with minimal or no relabelling and training effort. Three state-of-the-art, large-scale models were evaluated on both the scaffolding-pipe and bike-battery validation sets:

- Grounding DINO (SWIN/ViT backbone)
- Qwen 2.5 VL
- OWL-ViT (CLIP with ViT backbone)

Each model was prompted with an identical defect vocabulary and its predictions were inspected qualitatively and quantitatively. Despite Grounding DINO's strong COCO scores, OWL-ViT delivered the highest recall on domain-shifted imagery, especially for the low-contrast mortar residues on steel pipes, while maintaining a VRAM footprint that fits comfortably on the target RTX 3090. The same conclusion was made for Qwen 2.5 VL.

OWL-ViT was therefore selected as the zero-shot detector. At equal confidence thresholds it retrieved a higher proportion of true defects than either competitor yet incurred similar or lower computational cost and required no auxiliary captioning pipeline.

3.9.2 Supervised Detector Fallback

While zero-shot detection minimises annotation overhead, early trials revealed a non-negligible miss-rate on more subtle defects. An optional supervised detector is therefore retained as a fallback, trained on the limited but carefully curated datasets described in § 3.3. For this purpose, Transformer based networks were excluded because they need a lot of training data. The shortlist that was considered comprised out of the following models:

- DETR (ResNet-101 backbone)
- Faster R-CNN (ResNet-101 backbone)
- YOLOv11

The decisive criteria were (i) robust convergence on fewer than 700 labelled images, (ii) inference throughput compatible with real-time preview on an RTX 3090, and (iii) ease of extension to four-channel input (RGB + Depth or RGB + NIR) and (iv) ease of development due to time constraints.

DETR is sensitive to class imbalance and needs a lot of data or heavy augmentation. Faster R-CNN is option but is an outdated technology.

YOLOv11 was ultimately chosen. The Ultralytics implementation converged quickly, is relatively easy to train and implement on a limited dataset and sustained > 30 FPS in single-batch inference. Furthermore, it is adaptable to 4 channel input within the timeframe of the thesis. YOLOv11 thus best satisfied the combined constraints of dataset size, engineering effort, and inference speed.

3.9.3 Segmentation Engine

The mask generator that follows the detectors in Branch B must deliver pixel-accurate boundaries for any region highlighted by a bounding box yet remain agnostic to the precise detector that supplies the prompt. Four candidates were considered:

- Mask R-CNN (ResNet-101 backbone)
- DETR-Seg (Deformable attention head)
- YOLOv11-seg (prototype Ultralytics branch)
- Segment Anything Model 2.1 Hiera Large

Evaluation centred on three axes. Prompt flexibility mattered most, because the detector may change over the project's lifespan; only SAM 2.1 accepts boxes, points and optional text without retraining. Its independence from the detector guarantees that a future switch from OWL-ViT to, say, Grounding DINO or future state of the art OWOD models, requires no change downstream. Finally, zero-shot accuracy is known to be impressively accurate for SAM 2.1. Mask R-CNN, DETR-Seg and YOLOv11-seg would have been viable had the project relied

solely on supervised detectors, but their need for class-specific fine-tuning conflicts with the zero-shot philosophy that underpins the pipeline. SAM 2.1 was therefore adopted as the universal segmentation engine.

3.9.4 Sensor-Fusion Strategy

Both case studies exploit a fourth modality in addition, RGB—depth for bike batteries and near-infra-red for scaffolding pipes (§ 3.8). Three architectural options were considered:

- Early fusion Concatenate the auxiliary channel to RGB and train a single backbone.
- Intermediate fusion Maintain dual branches and merge via cross-attention.
- Late fusion Run modality-specific detectors and ensemble their logits or bounding boxes.

Intermediate fusion promised the highest ceiling on accuracy, but it needs a lot of modifications, which is time-consuming and not flexible for different structure since every time the structure file should be modified. Late fusion, although conceptually appealing, would have increased inference cost significantly and is more complex to implement than early fusion.

Early fusion was therefore selected. A one-line kernel patch allowed YOLOv11 to ingest four channels; the resulting networks incurred < 0.2 % extra GFLOPs and no material memory penalty yet could get us insights into the potential benefits of adding extra modalities into a model originally designed for three channels. Intermediate-fusion prototypes remain a future avenue once more development time becomes available.

3.9.5 Annotation & Labelling Environment

A modest but critical element of the workflow is the tool used to annotate the supervised subset (§ 3.7). Numerous platforms exist—Label Studio (*HumanSignal/Label-Studio*, 2019/2025), CVAT (CVAT.ai Corporation, 2018/2023), SuperAnnotate (*SuperAnnotate | Centralize Data Ops for Multimodal AI*), among others—but the project team already had extensive experience with X-AnyLabeling (W. Wang, 2023/2025).

X-AnyLabeling provides an intuitive interface, integrates SAM-based auto-labelling to accelerate polygon creation, and exports directly to both COCO and YOLO formats required downstream. Given the tight project schedule, the incremental benefit of evaluating and migrating to yet another labelling suite did not justify the effort or risk. X-AnyLabeling thus remained the default environment, satisfying all functional needs while allowing the team to concentrate on model development rather than tooling.

3.9.6 Hardware and Performance Constraints

All architectural choices described above were ultimately bounded by the physical resources that could be secured for the project. Training and inference both ran on an NVIDIA RTX 3090

with 24 GB of GDDR6X memory. The card represents a pragmatic middle ground: it delivers ample tensor-core throughput for mixed-precision training yet remains widely available and therefore indicative of what an industrial integrator might procure off-the-shelf. All memory budgets, batch sizes, and model variants discussed in previous sections were validated on this hardware; heavier backbones or attention mechanisms that exceeded the 24 GB ceiling were ruled out.

Illumination was governed by similar pragmatism. The RGB capture relied on an industrial LED bar already installed on the acquisition rig documented in § 3.3.1 and inherited from earlier master's thesis work. For the NIR channel a dedicated, narrow-band source would have been ideal to complement, yet lead-times conflicted with the thesis schedule. Standard halogen flood-lights were therefore adopted: their filament spectrum provides substantial energy around the NIR spectrum, they are inexpensive, and electrical regulation is straightforward.

The decision to retain the dual-wavelength RGB-NIR line-scan camera from the earlier prototype followed the same logic. Its reuse avoided a new round of calibration and integration while still delivering the spatial resolution required for fine mortar detection. No material performance benefit was predicted from purchasing a newer model because this is a good performing RGB NIR camera, so the available hardware was leveraged as-is.

3.9.7 Consolidated Overview Table

Table 3-3: Design Decisions overview

Pipeline Component	Selected Method	Rationale for Selection	Alternatives Considered & Rejected
Open-World Detection	OWLv2 (ViT backbone + CLIP)	Best zero-shot recall on domain-shifted defects; handles low-contrast cases well; fits in VRAM; no need for captioning pipeline	Grounding DINO (strong COCO scores but lower recall on shifted domains), Qwen 2.5 VL (strong, but similar as with grounding DINO)
Supervised Fallback	YOLOv11	Fast convergence on limited data (<700 images); >30 FPS inference; 4-channel adaptation; feasible under tight timeline	DETR (data-hungry, class imbalance sensitive), Faster R-CNN (outdated), YOLO-head on ConvNeXt V2 (too time-consuming to implement)
Segmentation Engine	SAM 2.1 Hiera Larga	Supports box, point, and text prompts; detector-agnostic; top-tier zero-shot segmentation performance	Mask R-CNN, DETR-Seg, YOLOv11-seg (all require class-specific training, not suitable for zero-shot pipeline)
Sensor Fusion Strategy	Early Fusion (channel concatenation)	Simple kernel patch enables 4-channel input; low computational overhead; viable within time constraints	Intermediate Fusion (more accurate but time-consuming to implement), Late Fusion (high inference cost, harder to maintain)
Annotation Environment	X-AnyLabeling	Fast SAM-based annotation; exports to COCO & YOLO; familiarity; no migration effort	Label Studio, CVAT, SuperAnnotate (functionally solid but no time to migrate or evaluate new toolchains)
Hardware Constraints	RTX 3090 + industrial lighting & existing line-scan camera	RTX 3090 balances power and accessibility; avoids exceeding 24 GB limit; existing lighting and dual-band camera reused to save time and ensure compatibility	Heavier backbones and new cameras were excluded due to VRAM limits and setup lead-times

3.10 Visualisation & Operator Feedback GUI

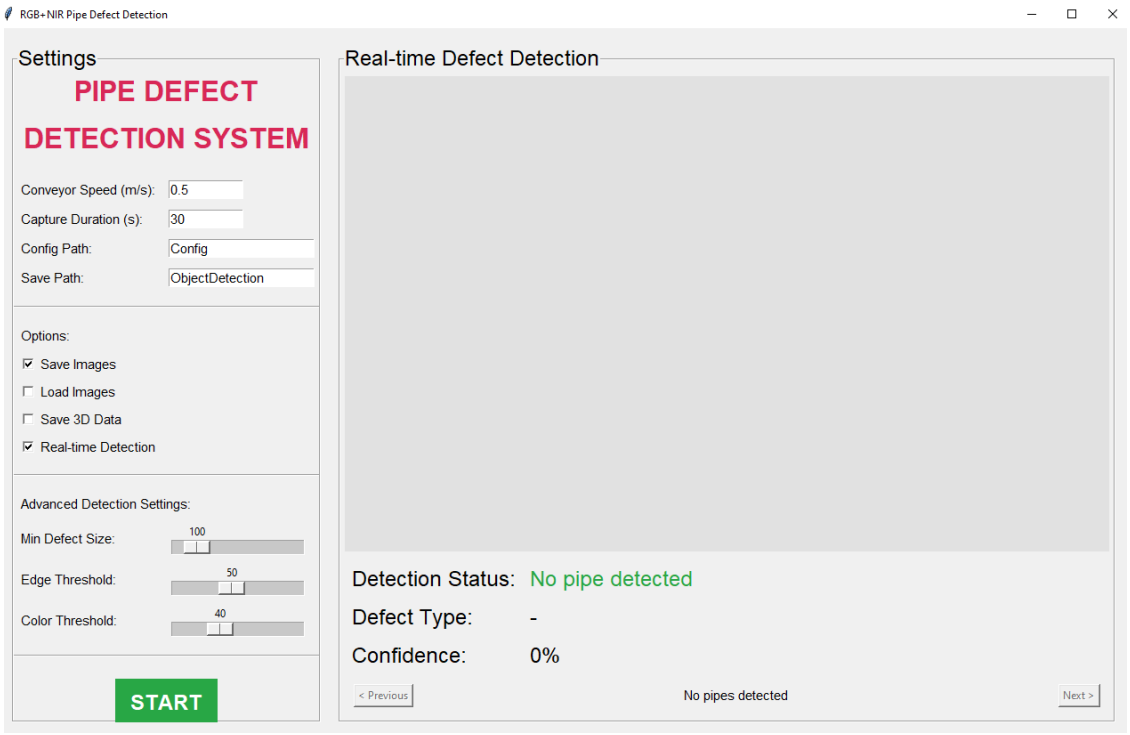


Figure 3-p: Real time pipe defect detection system GUI

A reliable inspection pipeline must culminate in a human-readable interface that presents algorithmic results with sufficient clarity for rapid decision-making on the factory floor. The present section therefore documents a live visualisation module that accompanies the scaffolding-pipe detector introduced in § 3.6. The implementation operationalises the zero-shot OWOD + SAM workflow in an operator-facing tool and fixes the performance envelope within which in the later result section will evaluate accuracy and throughput.

3.10.1 Software Architecture and Data Flow

The visualisation module is an event-driven Python application that bridges the zero-shot detector (§ 3.6) with a Tkinter / PySide6 GUI. RGB + NIR frames enter a shared-memory queue; a background thread pulls them at a configurable stride, feeds the RGB plane to OWLv2, applies weighted-box fusion and class-wise NMS, then pushes a JSON message (box, label, confidence) back to the GUI thread. The interface overlays each box on the live stream with a defect-specific colour (blue = mortar, red = rust, orange = bent) and prints the class tag plus confidence. A slider beneath the viewport exposes the global confidence threshold τ, with adjustments reflected on the next frame. Because inference and rendering live in separate threads, latency is bounded by GPU throughput—9–11 FPS at 1920 × 1200, or ~24 FPS when down-sampled to 1280 × 800 on the evaluation workstation. Persistence and PLC hooks are stubbed out in this prototype to keep the focus on real-time operator feedback.

3.10.2 Overlay Rendering and Visual Cues

Every incoming frame passes through the OWL-ViT branch of the zero-shot pipeline. When detections exceed the user-defined confidence threshold τ the system super-imposes an axis-aligned bounding box and textual class label onto the live video. To maximise perceptual separation between defect categories, the overlay adopts a fixed qualitative colour palette: blue for surface corrosion, green for indentation, and red for puncture. The absence of masks aligns the display with operator expectations formed by legacy 2-D CCD systems and avoids occluding visual texture cues that experts rely upon.

A slim status bar beneath the viewport indicates frame index, real-time FPS, and the current value of τ. If no detections are present the bar is rendered in neutral grey; when a defect is flagged it switches to the colour of the highest-confidence box, offering peripheral awareness even when the operator is focused elsewhere.

3.10.3 Interactive Threshold Adjustment

Domain experience shows that defect appearance varies with surface finish, lighting, and production batch. Rather than hard-coding τ, the GUI exposes a horizontal slider that updates the global threshold at once—no pipeline restart is required. Parameter changes are reflected on the next frame, enabling the operator to calibrate sensitivity during the first seconds of a

shift and to compensate for diurnal lighting drift. In future work the slider could be tied to an adaptive confidence predictor, but manual control was favoured for transparency during the user-acceptance phase.

3.10.4 Two-Dimensional Bounding-Box Detection

The visualisation module is written in Python with PySide6 bindings to Qt 6. OpenCV handles camera capture and basic drawing primitives, while the OWL-ViT inference thread communicates via a thread-safe queue to decouple GUI refresh from model latency. With a YOLOv11-n baseline for comparison, the OWL-ViT branch sustains 9–11 FPS on the workstation's RTX 3090 when processing a resolution of 1280 × 800. No image or overlay data have persisted in the current build; hooks are, however, in place for optional recording to disk should traceability become a requirement.

3.10.5 Two-Dimensional Bounding-Box Detection

Although the present thesis evaluates the interface in a single-camera setting, the software architecture already abstracts the video source and could therefore accommodate the multispectral fusion models of § 3.8 or a depth-augmented viewpoint. Likewise, an audit-trail database and an operator-feedback widget—where false positives/negatives could be tagged for incremental retraining—are envisaged for subsequent iterations.

3.10.6 Hardware and Performance Constraints

All architectural choices described above were ultimately bounded by the physical resources that could be secured for the project. Training and inference both ran on a workstation equipped with an NVIDIA RTX 3090 (24 GB GDDR6X), an Intel Core i7-13700KF CPU, and 64 GB of system RAM. The graphics card represents a pragmatic middle ground: it delivers ample tensor-core throughput for mixed-precision training yet remains widely available and therefore indicative of what an industrial integrator might procure off-the-shelf. All memory budgets, batch sizes, and model variants discussed in previous sections were validated on this hardware; heavier backbones or attention mechanisms that exceeded the 24 GB ceiling were ruled out.

Illumination was governed by similar pragmatism. The RGB capture relied on an industrial LED bar already installed on the acquisition rig documented in § 3.3.1 and inherited from prior work (see [Pablo thesis]). For the NIR channel a dedicated, narrow-band source would have been ideal, yet lead-times conflicted with the thesis schedule. Standard halogen floodlights were therefore adopted: their filament spectrum provides substantial energy around the NIR region, they are inexpensive, and electrical regulation is straightforward.

The decision to retain the dual-wavelength RGB-NIR line-scan camera from the earlier prototype followed the same logic. Its reuse avoided a new round of calibration and integration

while still delivering the spatial resolution required for fine mortar detection. No material performance benefit was predicted from purchasing a newer model, so the available hardware was leveraged as-is.

3.11 Evaluation Protocol

This section explains in detail how performance is quantified for every stage of the proposed pipeline. The numerical outcomes are presented in Chapter 4; this section defines the test conditions, metrics, and comparison axes to ensure unambiguous reproducibility of the evaluation.

3.11.1 Two-Dimensional Bounding-Box Detection

Bounding-box performance is evaluated with the COCO detection protocol. The most widely reported scalar in computer-vision benchmarks is the Average Precision at an IoU threshold of 0.5 (mAP@0.5). For a single class c, it is the area under the precision–recall curve at that threshold:

$$AP_c(0.5) = \int_0^1 p_c(r, 0.5)\, dr \approx \sum_k (r_k - r_{k-1})\, p_{interp,c}(r_k, 0.5).$$

To obtain a dataset-level score, AP values are averaged over all classes:

$$mAP@0.5 = \frac{1}{C}\sum_{c=1}^{C} AP_c(0.5).$$

For a threshold-agnostic perspective, the mean Average Precision is also reported as the average over the ten COCO thresholds from 0.50 to 0.95 in steps of 0.05 (mAP@[0.5:0.95]):

$$mAP@[0.5{:}\,0.95] = \frac{1}{10C} \sum_{t\epsilon\{0.50,\ldots,0.95\}} \sum_{c=1}^{C} AP_c(0.5).$$

Finally, Recall is reported as Recall at IoU 0.5 measures completeness:

$$\boldsymbol{Recall(0.5)} = \frac{\boldsymbol{TP(0.5)}}{\boldsymbol{TP(0.5) + FN(0.5)}}$$

To assess the influence of input modality, each metric is computed for the two variants per dataset: RGB-only and fused input (RGB + NIR for scaffolding pipes, RGB + depth for bike batteries). Additionally, for each modality, the zero-shot detection pipeline is compared with its supervised alternative. This setup allows for a structured comparison of model supervision level and input information content, independently across both datasets.

3.11.2 Segmentation Masks Generated by SAM

Segmentation accuracy is measured with the COCO instance-segmentation protocol. Consistent with detection, the primary scalar reported is mAP@0.5—the average precision of mask predictions at an IoU threshold of 0.5. This is the figure that still dominates leaderboard

summaries in computer vision. For a threshold-agnostic view, mAP@ [0.5:0.95] is also reported, computed over the same set of ten IoU thresholds used for bounding boxes. In essence, mAP for masks evaluates how well the predicted and ground-truth shapes match at the pixel level, while bounding box mAP compares the enclosing rectangles.

All underlying quantities—IoU, precision, recall, AP and their class-averaged means are defined exactly as in the previous section, but then for the area of the masks instead of the area of the bounding boxes.

3.11.3 Operator-Level Classification on Scaffolding Pipes

The scaffolding-pipe application imposes a further requirement: each mask must be translated into an actionable defect category that can be displayed in the maintenance GUI. Although this classification task was not initially planned, exploratory experiments with the zero-shot OWOD model suggested that investigating defect classification could provide valuable insights. This classifier operates by performing a grounding task using the OWOD model, but instead of using the predicted bounding box, it only extracts the textual class label associated with the detection. The performance of this classifier is evaluated using precision, recall, and the F1-score, calculated across the entire scaffolding-pipe test set. As this classification component is specific to the scaffolding-pipe context, no equivalent metrics are reported for the bike-battery dataset. This added classification label can come in handy for an operator who is scanning for specific types of defects.

3.11.4 Three-Dimensional Anomaly Detection on Bike Batteries

For the bike-battery use-case the ultimate deliverable is a geometric deviation score. The reconstructed point cloud of each battery half is rigidly aligned to a reference scan of a non-defective battery using the Iterative Closest Point algorithm. Following this alignment, the root-mean-square distance between corresponding points is computed to quantify the magnitude of deformation. The time taken for both the registration and error calculation are also measured to provide a realistic estimate of throughput on the chosen hardware.

3.11.5 Hardware used for training and inference

All measurements are obtained on the workstation described in § 3.10.6. Inference is executed with a batch size of one and FP16 arithmetic wherever the model supports mixed precision. These settings remain identical across detectors, modalities, and datasets so that differences in accuracy or runtime can be attributed solely to the algorithmic choices outlined above.

4 RESULTS

4.1 3-D Point-Cloud Registration & Anomaly Scoring: E-Bike Battery Dataset

4.1.1 3-D Point-Cloud Registration

Table 4-1: Comparison and benchmarking of different ICP protocols for 3D registration pipeline

Method	RMSE (Mean mm)	Time (Mean)	Relative Speed (%)	Success Rate (%)	Benchmark Score
Point-to-Point ICP (Fast)	0.038407546	0.123881531	100	100	59.83359055
Point-to-Plane ICP (CPU)	0.0383397	1.32867794	9.323668814	100	57.48763107
Point-to-Plane ICP (CUDA+CPU)	0.0383397	2.284084272	5.423684768	100	56.20423849
Coloured ICP	0.0383397	3.05574441	4.054054075	100	55.16767125
Multiscale ICP (CUDA)	0.019469807	14.8887619	0.832047228	100	24.84619153

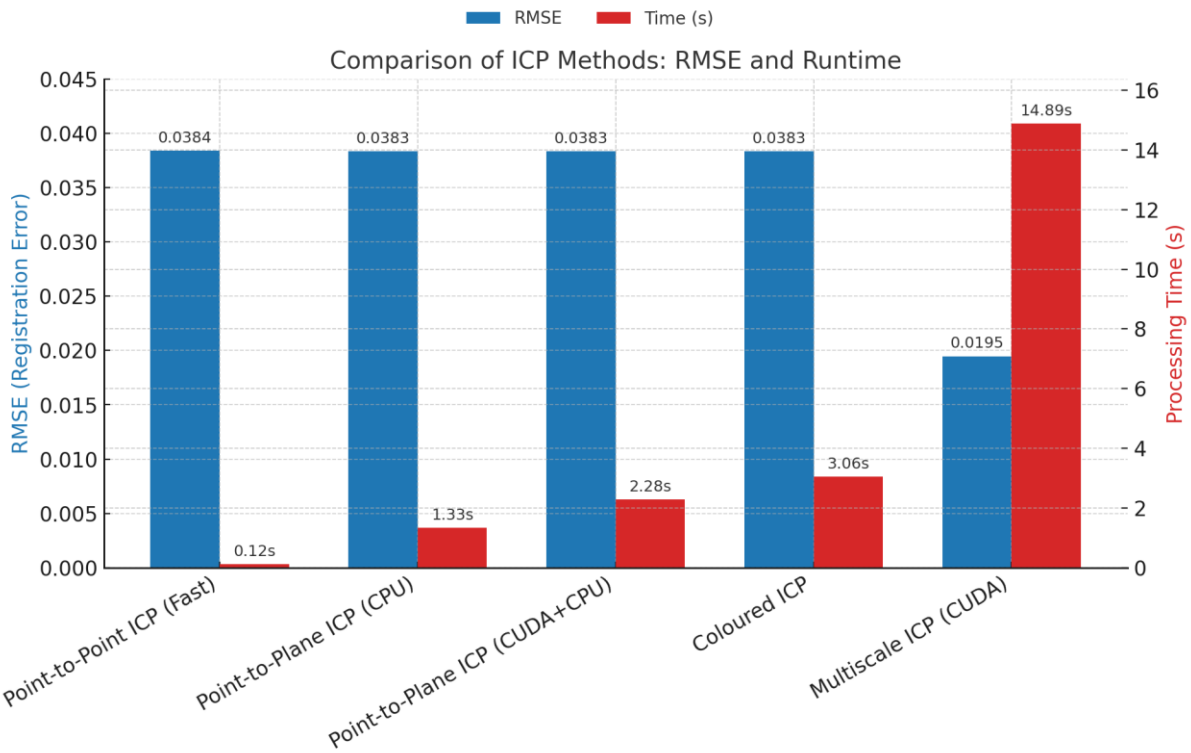


Figure 4-a: Comparison of ICP methods: RMSE and Runtime

The registration performance evaluation of ICP-based methods for e-bike battery multi-viewpoint cloud alignment included benchmarking Fast Point-to-Point ICP and Point-to-Plane ICP (CPU and CUDA versions) alongside Coloured ICP and Multiscale ICP (CUDA). The evaluation of each method included Root Mean Square Error (RMSE) measurements and average processing time results and relative speed assessments and a combined benchmark score that evaluated speed and accuracy performance. The Fast Point-to-Point ICP method delivered the best performance trade-off by achieving a benchmark score of 59.83 with 0.0384 mm RMSE and 0.124 seconds processing time which makes it suitable for real-time or high-throughput operations. The Multiscale ICP with CUDA achieved the lowest RMSE of 0.0195 but required 14.89 seconds of processing time which made it impractical for scalable deployment. The CPU and CUDA versions of Point-to-Plane ICP produced similar accuracy results with RMSE values of 0.0383 mm but the CUDA version failed to deliver its expected speed benefits because of GPU memory usage and poor data batching practices. The Coloured ICP failed to enhance accuracy when applied to industrial scaffolding pipes because it relies on RGB information which provides limited benefits for low-texture surfaces. The registration performance depended on three critical factors which included voxel size set to 0.005 m for optimal results, convergence criteria defined by maximum iterations and RMSE thresholds and the quality of segmentation masks produced by the DIS deep learning model. The masks functioned as essential components for surface object isolation which enhanced the reliability of point cloud filtering and alignment operations. The findings demonstrate how method selection and hyperparameter optimization enable inspection systems to achieve optimal precision-efficiency trade-offs in multi-modal vision-based inspection systems.

Figure 4-b: Final registered e-bike battery

4.1.2 Anomaly map

The binary anomaly detection workflow involved a structured six-step pipeline, beginning with the loading and preprocessing of two high-resolution point clouds—one representing the defective structure and the other a clean reference. The point clouds contained over 2.2 million points each and spanned substantial 3D spatial dimensions. Preprocessing included statistical outlier removal to eliminate noise, followed by voxel downsampling at a resolution of 0.003 mm. This process preserved the full structural integrity of the original scans, with all points retained post-down sampling. Surface normals were estimated to support alignment, and the point clouds were then registered using a multi-scale Iterative Closest Point (ICP) approach, proceeding through decreasing voxel sizes of 0.012, 0.006, and 0.003 mm. The registration resulted in a fitness score of 0.9999, indicating nearly perfect correspondence between the overlapping regions. Following registration, the system computed point-wise distances between the two clouds using a KD-Tree-based accelerated search. This process ran efficiently across 220 batches, completing in 18.21 seconds. The resulting distance distribution was right-skewed, with a mean of 1.27, median of 0.60, and a maximum deviation of 16.90 mm. Most of the points (nearly 80%) were within 1.69 mm, while extreme deviations were concentrated in a very small proportion of the dataset. Key percentiles further revealed that 90% of all points were below 3.32 mm, while the 95th and 99th percentiles reached 5.05 and 8.98 mm, respectively.

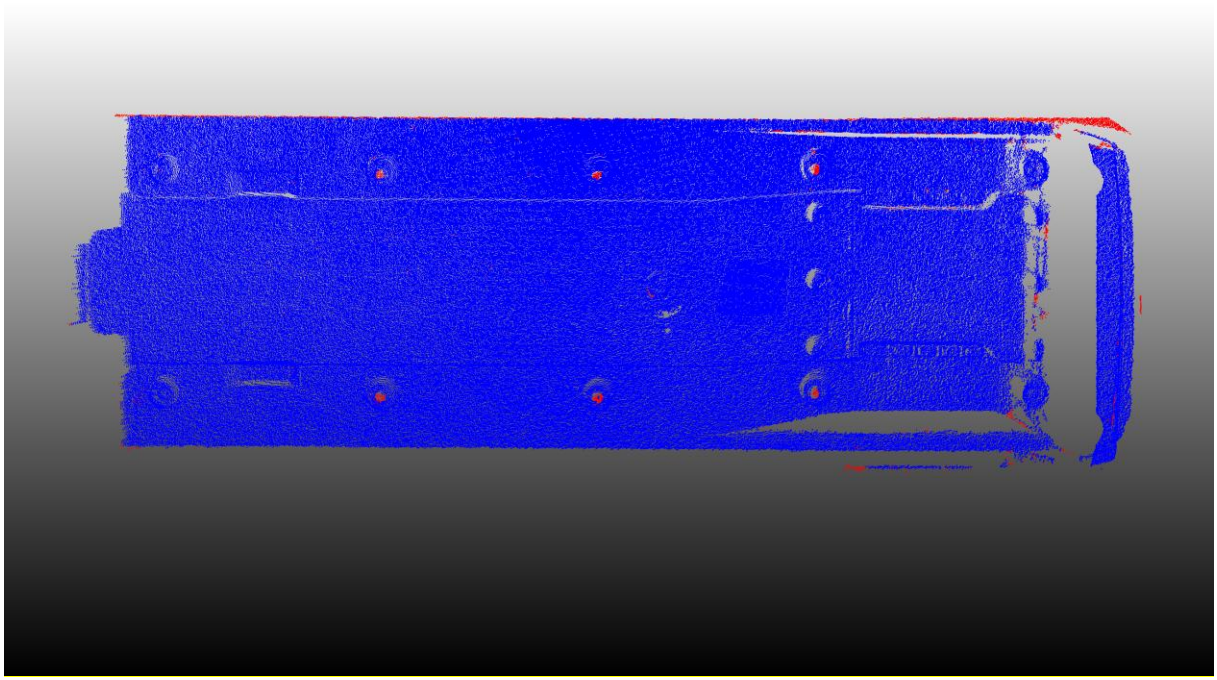


Figure 4-c: Anomaly map highlighting 6 missing screws in e-bike battery

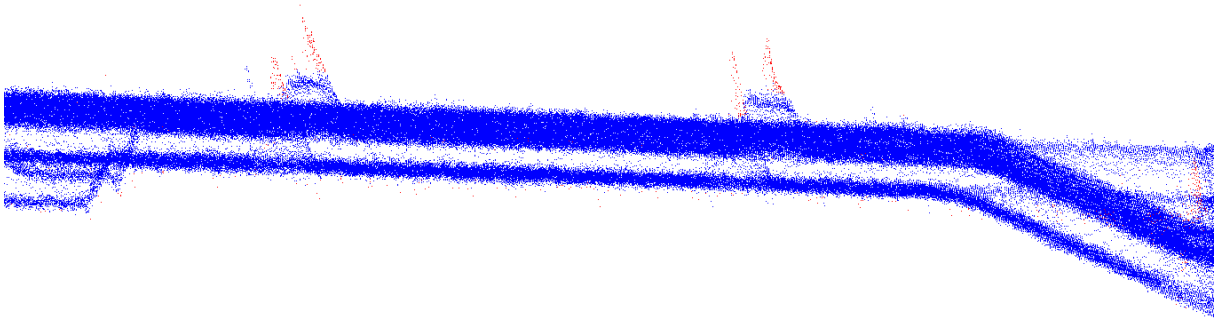


Figure 4-d: Missing screws in e-bike battery close up

To isolate critical anomalies, a 97th percentile threshold of 6.3024 mm was selected. This strategy classified only the top 3% of point deviations as anomalous, providing a robust method for minimizing false positives while highlighting the most significant structural discrepancies. In total, 65,769 anomaly points were flagged, while 2,126,515 points were considered normal, maintaining a practical and interpretable anomaly rate of 3.00%

4.2 2-D Detection & Segmentation — E-Bike Battery Dataset

4.2.1 Zero-Shot Baseline — OWLv2 + SAM

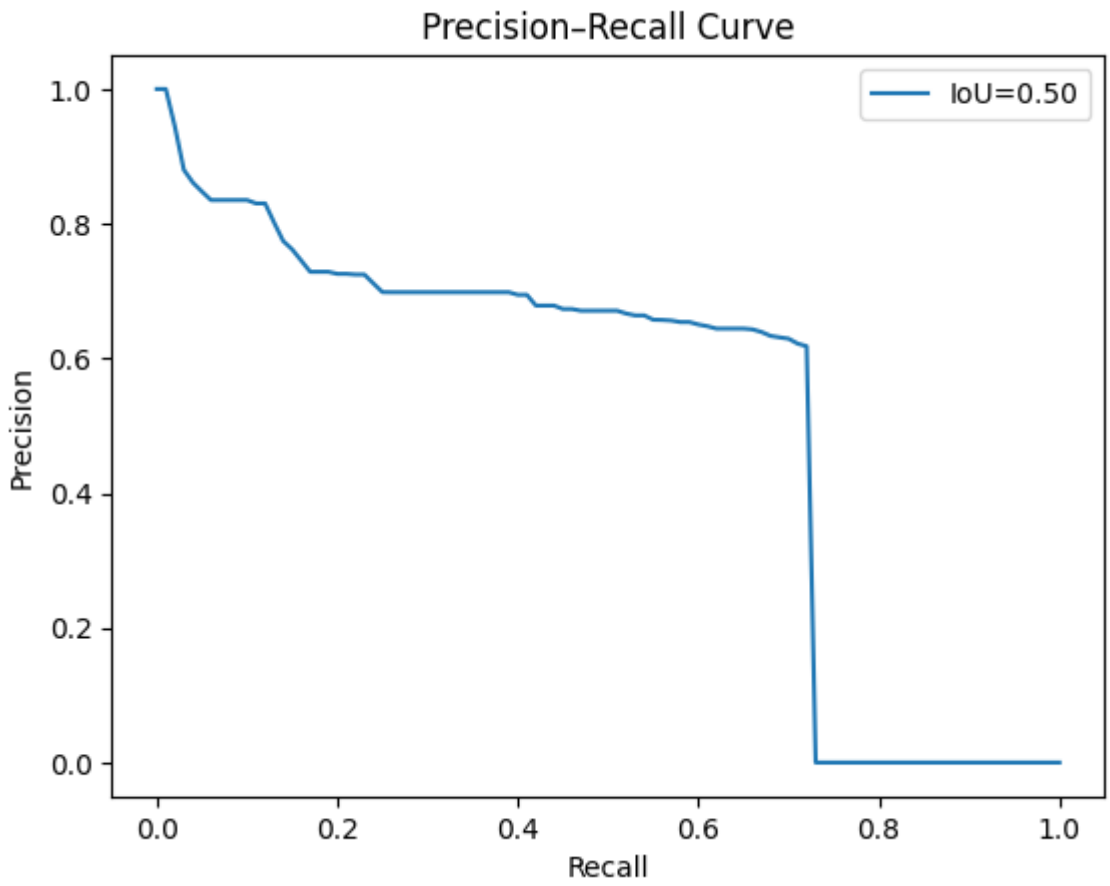


Figure 4-e: PR curve for OWOD e-bike battery pipeline performance

The PR curve in Figure 4-e shows the zero-shot OWOD model's ability to detect screw classification in e-bike battery dataset. COCO-style segmentation metrics evaluated the model at an Intersection over Union threshold of 0.50. Precision begins at 1.0 at the beginning of the recall spectrum, and the model makes highly accurate predictions under strict confidence threshold conditions. However, precision shows a downward trend that becomes severe when recall exceeds 0.75. This drop results from the relaxation of the detection threshold to capture more true positives, which also increases the number of false positives. Additionally, the recall value never reaches 1.0, meaning the model consistently fails to detect all screw instances regardless of the confidence level. This behaviour aligns with the limitations of zero-shot detection, where the model relies on generic objectless cues and semantic similarity rather than class-specific training.

The qualitative pattern from the PR curve is consistent with the quantitative metrics derived from the COCO evaluation. Under the segmentation evaluation, the model achieves an Average Precision of 0.519 at IoU = 0.50, which drops to 0.264 at IoU = 0.75, highlighting a

reduction in localization accuracy under stricter matching. The mean Average Precision across all IoUs (0.50:0.95) for segmentation is 0.260, suggesting moderate detection performance. The model performs slightly better on small and medium-sized objects, with segmentation AP scores of 0.289 and 0.268, respectively, while the AP for large objects is undefined due to the absence of such instances in the dataset. Average Recall (AR) in segmentation peaks at 0.485 when the maximum number of detections per image is set to 100, and small objects achieve a notably high AR of 0.631.

Bounding box evaluation reveals a similar trend but with slightly weaker results. The model attains an AP of 0.505 at IoU = 0.50, yet this falls sharply to 0.069 at IoU = 0.75, and the overall bounding box mAP is lower at 0.157. The AR under bounding box evaluation reaches a maximum of 0.340 with 100 detections, while small object AR is 0.424. These results suggest that the model is more consistent with pixel-level (mask-based) localization than with bounding box placement, particularly under high IoU thresholds. The greater segmentation scores may reflect the model's ability to roughly capture object shapes even when its bounding box predictions are less precise.

In summary, these findings indicate that a zero-shot OWOD model can effectively detect unseen industrial parts such as screws, especially when they appear in smaller sizes. However, it suffers from rapid precision degradation as recall increases and consistently fails to detect all object instances. The lower mAP and AR values in the bounding box evaluation, compared to segmentation, underscore the challenges of precise object localization without prior class-specific supervision. This emphasizes the broader difficulty of fine-grained object detection in zero-shot settings, particularly for small, visually complex, or sparsely annotated categories.

Figure 4-f: e-bike battery correctly mapped via OWOD model examples with legend

Figure 4-g: e-bike battery with OWOD model errors

4.2.2 Supervised Baseline — YOLOv11 (RGB) + SAM

4.2.2.1 Prediction Examples

Ground Truth

Predicted

Ground Truth

Predicted

Ground Truth

Predicted

4.2.2.2 Quantitative Results

A five-capacity YOLOv11 family (n → x) was fine-tuned on the bike-battery training split and evaluated on the held-out test set. Bounding-box scores at IoU = 0.5 together with the full COCO mean-IoU metric, recall, per-image inference latency, and GFLOPS are listed in Table 4-2. Each detector's outputs were forwarded to SAM 2.1 to obtain pixel-level masks; the corresponding mask mAP values are summarised in Table 4-3.

Table 4-2: YOLOv11 Results Screw – RGB

Model	mAP@0.5 (box)	mAP@0.5:0.95 (box)	Recall	Inference Speed (ms)	GFLOPs
YOLO11n	0.954	0.732	0.905	2.5	6.3
YOLO11s	0.957	0.767	0.919	3.3	21.3
YOLO11m	0.954	0.780	0.909	3.7	67.6
YOLO11l	0.962	0.779	0.906	4.8	86.6
YOLO11x	0.963	0.780	0.911	8.4	194.4

Table 4-3: YOLOv11 + SAM Results Screw - RGB

Model	mAP@0.5 (mask)	mAP@0.5:0.95 (mask)	Dice Score	Inference Speed (ms)
YOLO11n + SAM2.1 Hiera Large	0.932	0.692	0.916	268.5
YOLO11s + SAM2.1 Hiera Large	0.935	0.705	0.913	271.3
YOLO11m + SAM2.1 Hiera Large	0.934	0.847	0.918	273.7
YOLO11l + SAM2.1 Hiera Large	0.947	0.833	0.921	275.8
YOLO11x + SAM2.1 Hiera Large	0.947	0.847	0.920	279.4

All five detectors yield near-saturation detection accuracy (mAP@0.5 ≥ 0.954). Mask performance also remains high: SAM 2.1 converts these boxes into segmentation masks with only a modest drop in mAP, achieving up to 0.947 @ 0.5 IoU. The consistency between detection and segmentation suggests that screws present strong local texture and shape cues that SAM 2.1 can delineate reliably once localised. Per-image inference latency grows with model capacity yet remains < 9 ms for detection; the dominant runtime cost stems from the fixed-size SAM 2.1 call (~270 ms per image), independent of the YOLO variant. These findings establish the RGB-only configuration as a performant baseline against which the depth-enhanced variant can be compared.

4.2.3 Multichannel Variant — YOLOv11 (RGB + Depth) + SAM

4.2.3.1 Prediction Examples

Since the predictions are visually similar to those shown in 4.2.2.1, this section refers the reader to those examples for a representative illustration of the model's behaviour.

4.2.3.2 Quantitative Results

A four-channel (RGB + Depth) version of the YOLOv11 family (n → x) was trained with the same protocol as in § 3.8 and evaluated on the scaffolding-pipe test set. Bounding-box metrics are listed in Table 4-4, followed by mask metrics for the YOLO → SAM 2.1 pipeline in Table 4-5.

Table 4-4: YOLOv11 Results Screw – RGBD

Model	mAP@0.5	mAP@0.5:0.95	Recall	Inference Speed (ms)	GFLOPs
YOLO11n	0.916	0.656	0.856	2.4	6.3
YOLO11s	0.918	0.705	0.893	3.1	21.4
YOLO11m	0.918	0.721	0.887	3.6	67.8
YOLO11l	0.921	0.706	0.879	4.8	86.7
YOLO11x	0.922	0.717	0.874	8.1	194.6

Table 4-5: YOLOv11 + SAM Results Screw - RGBD

Model	mAP@0.5	mAP@0.5:0.95	Dice Score	Inference Speed (ms)
YOLO11n + SAM2.1 Hiera Large	0.916	0.701	0.884	268.7
YOLO11s + SAM2.1 Hiera Large	0.917	0.733	0.896	271.6
YOLO11m + SAM2.1 Hiera Large	0.916	0.732	0.902	273.9
YOLO11l + SAM2.1 Hiera Large	0.920	0.729	0.882	276.0
YOLO11x + SAM2.1 Hiera Large	0.920	0.722	0.900	279.5

Contrary to initial expectations, adding the depth channel did not boost detection accuracy; mAP@0.5 drops by roughly 3–4 pp across capacities compared with the RGB-only baseline. This suggests that depth cues contribute little discriminative signal for the small, largely planar screw heads visible in the camera frame, and may even introduce convergence noise. SAM 2.1 mask metrics track the detector scores closely, maintaining ≈ 0.92 mAP@0.5 but offering no tangible gain over RGB. These findings align with previous research (C. Zhou et al., 2024) where similarly limited benefit when fusing RGB with depth and intensity cues for screw detection was observed. In their study, early fusion underperformed compared to RGB-only

models, and feature-level fusion required careful alignment mechanisms to surpass baseline performance.

Runtime overhead by adding this fourth channel remains negligible for detection (< 0.3 ms difference) and unchanged for SAM 2.1. Overall, the results suggest that in constrained screw inspection settings where texture and reflectance cues dominate over depth. A broader interpretation of this trend is provided in § 4.5.

4.3 2-D Detection & Segmentation — Scaffolding-Pipe Dataset

4.3.1 Zero-Shot Baseline — OWLv2 + SAM

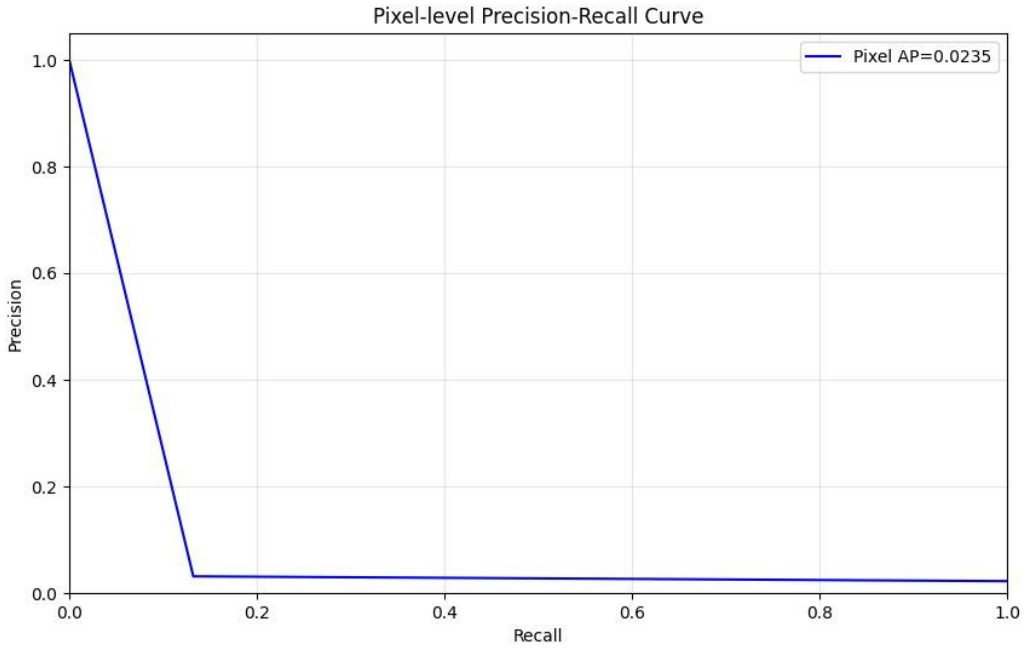


Figure 4-h: PR curve for pipe OWOD pipeline inference

The zero-shot Open World Object Detection (OWOD) baseline which combines OWLv2 with the Segment Anything Model (SAM) was evaluated on the scaffolding pipe dataset for both detection and segmentation tasks. The model's performance was limited because zero-shot methods are difficult to apply to specialized industrial data.

The model reached a mean Average Precision (mAP) bbox of 0.0357 across multiple Intersection over Union (IoU) thresholds for object detection. The AP remained at 0.0357 at the standard IoU threshold of 0.50 which shows that relaxing the spatial matching criterion did not improve detection results. The AP-IoU curve (Figure X) shows that performance steadily declines as the IoU threshold increases, approaching zero beyond 0.75. The model fails to detect objects precisely when the spatial boundaries become tighter.

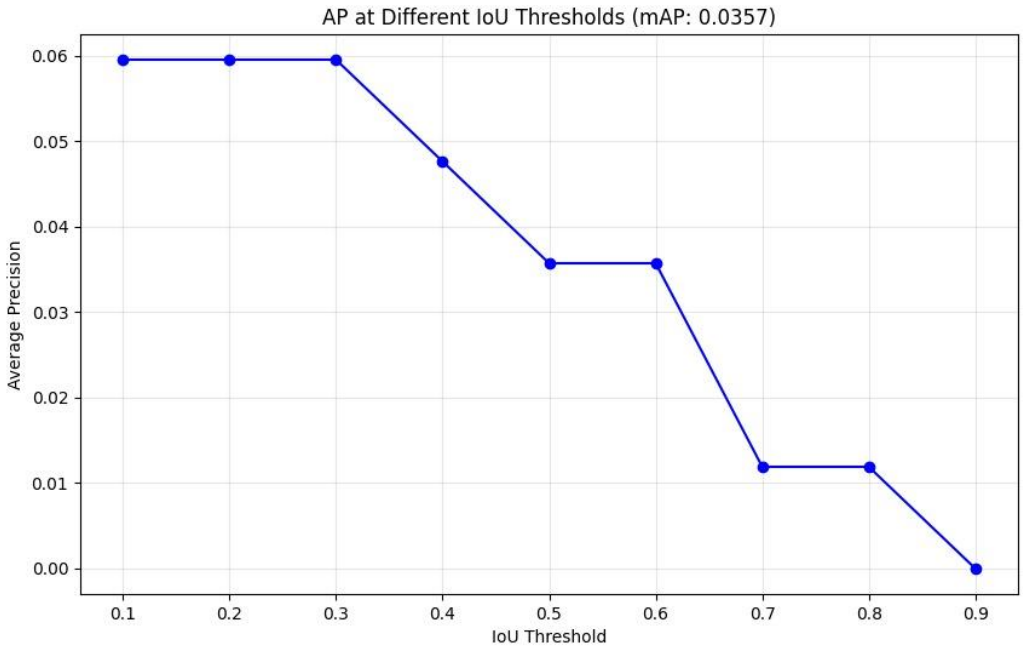
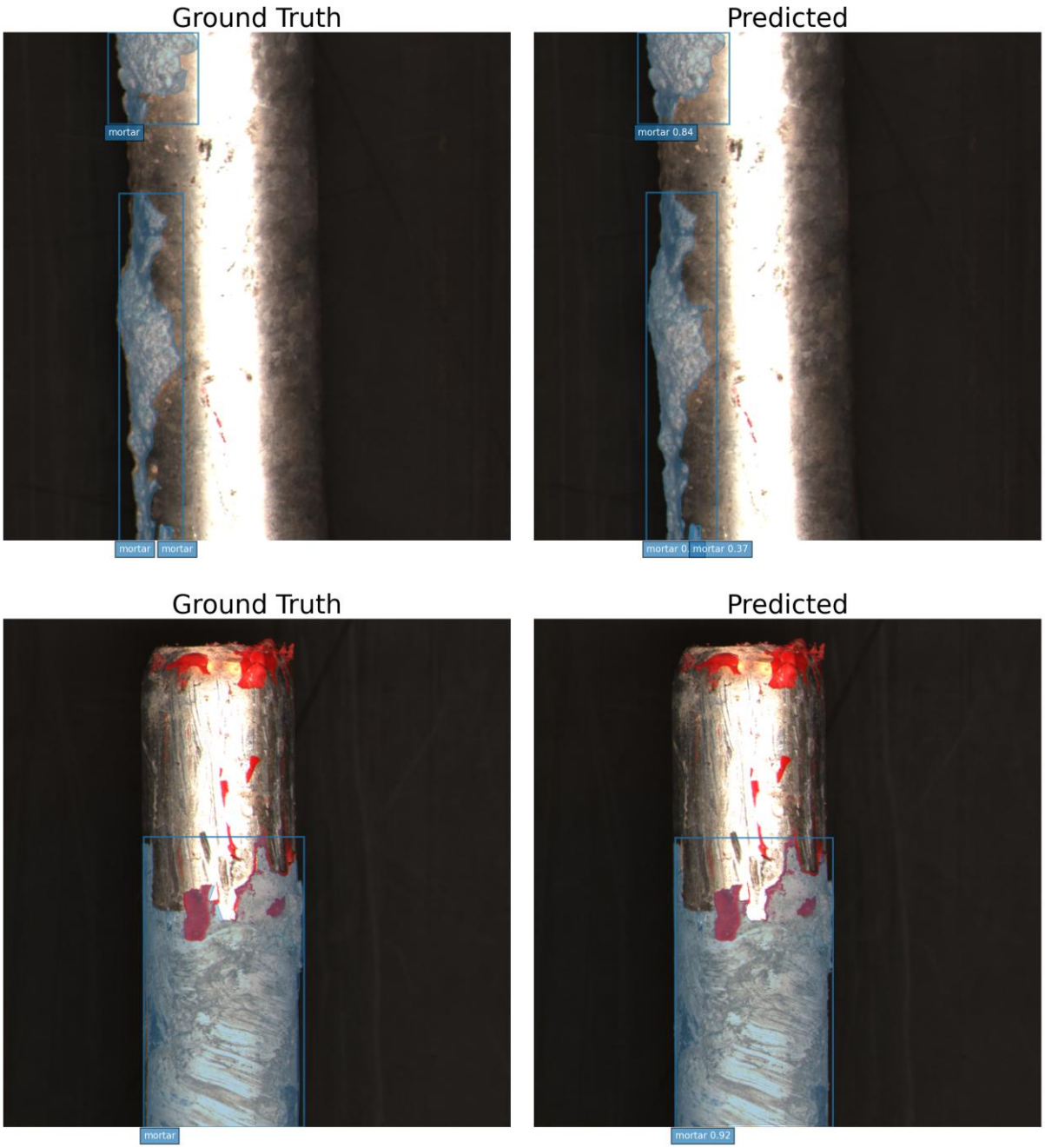


Figure 4-i: AP at different IoU thresholds for pipe

Segmentation performance was also weak. The pixel-level Average Precision segm was measured at 0.0235 as shown by the precision-recall curve in Figure Y. The precision starts at 1.0 for very high-confidence predictions but drops rapidly as recall increases. The model demonstrates high certainty in identifying a few positive instances, yet its recall ability remains severely restricted. The threshold relaxation to detect more positive instances leads to a significant increase in false positives.

4.3.2 Supervised Baseline — YOLOv11 (RGB) + SAM

4.3.2.1 Prediction Examples



4.3.2.2 Quantitative Results

A five-capacity YOLOv11 family (n → x) was fine-tuned on the scaffolding-pipe training split and evaluated on the held-out test set. Bounding-box results at the standard COCO threshold of IoU = 0.5 are summarised in Table 4-6; the table also lists the full COCO mean-IoU score, recall, single-image inference latency, and theoretical GFLOPS. Each detector's predictions were then forwarded to SAM 2.1 to obtain pixel-level masks; the corresponding mask mAP values are reported in Table 4-7.

Table 4-6: YOLOv11 Results Mortar - RGB

Model	mAP@0.5	mAP@0.5:0.95	Recall	Inference Speed (ms)	GFLOPS
YOLO11n	0.926	0.683	0.850	1.6	6.3
YOLO11s	0.953	0.742	0.881	2.7	21.3
YOLO11m	0.957	0.756	0.830	5.9	67.6
YOLO11l	0.962	0.796	0.909	8.6	86.6
YOLO11x	0.970	0.823	0.933	12.8	194.4

Table 4-7: YOLOv11 + SAM Results Mortar - RGB

Model	mAP@0.5	mAP@0.5:0.95	Dice Score	Inference Speed (ms)
YOLO11n + SAM	0.593	0.192	0.675	196.6
YOLO11s + SAM	0.552	0.187	0.660	197.7
YOLO11m + SAM	0.555	0.199	0.681	200.9
YOLO11l + SAM	0.536	0.199	0.675	203.6
YOLO11x + SAM	0.527	0.184	0.681	207.8

A concise inspection of Table 4-6 shows a monotonic gain in mAP@0.5, peaking at 0.970 for YOLOv11-x, with inference latency rising proportionally to model size. In contrast, when propagated through SAM 2.1 (Table 4-7), mask accuracy drops substantially, with mAP@0.5 values hovering around 0.55. This discrepancy highlights a limitation of the zero-shot segmentation approach: even when bounding boxes closely match the ground truth, SAM 2.1 may oversegment, extending the predicted mask beyond the actual defect region. An example of this overshoot—where the box has high IoU but the mask deviates significantly—is provided in Figure 4-j. Extended discussion and cross-case interpretation are deferred to § 4.5.

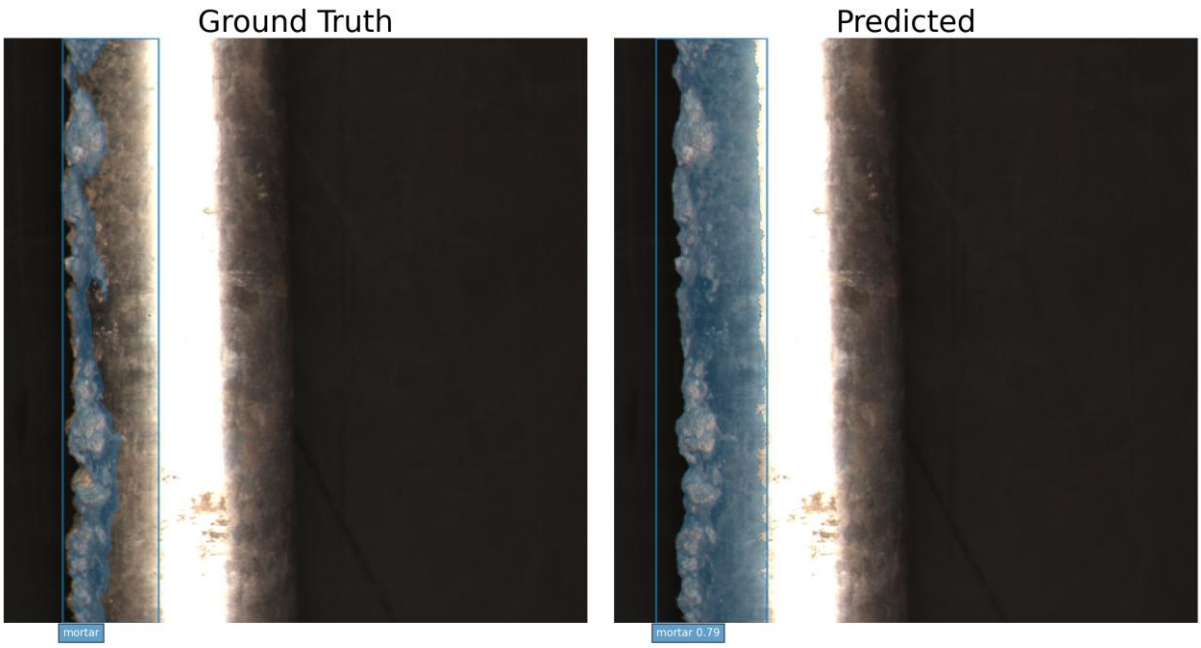


Figure 4-j: Ground truth vs predicted result for YOLOv11 model trained on RGB data

4.3.3 Multispectral Variant — YOLOv11 (RGB + NIR) + SAM

4.3.3.1 Prediction Examples

Since the predictions are visually similar to those shown in § 4.2.3.1, this section refers the reader to those examples for a representative illustration of the model's behaviour.

4.3.3.2 Quantitative Results

A four-channel (RGB + NIR) version of the YOLOv11 family (n → x) was trained with the same protocol as in § 3.8 and evaluated on the scaffolding-pipe test set. Bounding-box metrics are listed in Table 4-8, followed by mask metrics for the YOLO → SAM 2.1 pipeline in Table 4-9

Table 4-8: YOLOv11 Results Mortar – RGB NIR

Model	mAP@0.5	mAP@0.5:0.95	Recall	Inference Speed (ms)	GFLOPS
YOLO11n	0.956	0.769	0.901	1.5	6.3
YOLO11s	0.975	0.824	0.943	2.7	21.4
YOLO11m	0.976	0.817	0.928	5.4	67.8
YOLO11l	0.977	0.841	0.965	6.9	86.7
YOLO11x	0.982	0.869	0.960	13.1	194.6

Table 4-9: YOLOv11 + SAM Results Mortar – RGB NIR

Model	mAP@0.5	mAP@0.5:0.95	Dice Score	Inference Speed (ms)
YOLO11n + SAM	0.573	0.199	0.673	196.5
YOLO11s + SAM	0.537	0.189	0.674	197.7
YOLO11m + SAM	0.547	0.190	0.685	200.4
YOLO11l + SAM	0.550	0.195	0.676	201.9
YOLO11x + SAM	0.562	0.198	0.692	208.1

A concise inspection of Table 4-8 confirms the familiar monotonic rise in mAP@0.5, now topping out at 0.982 for YOLOv11-x. These scores are higher than their RGB-only counterparts and a detailed comparison is reserved for § 4.5. When the same detections are propagated to SAM 2.1 (Table 4-9), mask accuracy remains in the ~0.55 mAP@0.5 range, mirroring the behaviour observed in § 4.3.3.2 and indicating that segmentation accuracy is primarily limited by SAM 2.1 itself rather than the quality of the bounding box detections.

4.4 Discussion

This section examines the principal experimental results that appear throughout the thesis by linking quantitative performance metrics to methodological choices. The research employed both 3D point cloud registration and 2D image-based object detection and segmentation pipelines in the two industrial case studies involving e-bike batteries and scaffolding pipes. Different testing configurations were evaluated through the combination of standard geometric approaches and deep learning algorithms together with RGB and depth and near-infrared (NIR) imaging modalities.

The experimental findings deliver direct answers to all three research questions. The combination of deep learning-based segmentation (using DIS) with traditional point cloud registration methods (including ICP) produced sub-millimetre alignment precision through appropriate voxel sizes and filtering masks. The results demonstrate how precise registration systems can be developed through the combination of traditional and modern vision techniques.

The evaluation between zero-shot detection (OWOD) and supervised models (YOLOv11 variants) revealed different performance characteristics. The potential of OWOD models for detecting small objects without training was observed but their spatial accuracy remained inconsistent while they struggled in complex industrial scenes especially in the scaffolding pipe dataset. Supervised YOLOv11 models delivered strong mAP scores along with dependable segmentation results when used with SAM 2.1.

The analysis of different input modalities demonstrated that RGB+NIR fusion delivered better detection precision for texture-similar defects such as mortar. The combination of RGB and Depth imaging showed either no improvement or sometimes worse results for identifying flat and reflective objects like screws because of signal noise. These findings support earlier research findings and demonstrate that choosing suitable input modalities is crucial to enhance both contrast and structural information.

The results show that method choices along with data selection and model structure have a direct effect on detection precision and both segmentation quality and registration accuracy. The evidence proves that inspection systems can be developed using modular data-efficient approaches by merging deep learning with classical techniques and designing input approaches for specific inspection requirements.

4.4.1 3D combination of deep learning and traditional methods

The thesis began by addressing the first research question about the possibility of merging traditional computer vision techniques with deep learning methods to create a strong

registration pipeline that works with 3D point clouds and RGB images. The end-to-end registration system was developed by integrating DIS (deep learning-based segmentation model) with ICP (Iterative Closest Point) algorithms. The DIS model segments RGB images of e-bike battery modules into high-quality masks which direct the following point cloud filtering and alignment process. The masks function as essential elements for extracting the important surface geometry before registration takes place.

The ICP component of the pipeline operates across multiple scales to refine alignment, and the combination of deep learning segmentation with geometric registration yielded highly accurate results. The system reached a RMSE of 0.038 mm in 0.123 seconds which proves both accuracy and speed. The hybrid approach demonstrates both practicality and scalability for industrial inspection applications in real-world scenarios. The Re-DE manufacturing lab is developing an application which will contain this pipeline through wrapper classes to enable straightforward deployment in production settings.

The designed anomaly scoring algorithm detects structural deviations by evaluating point-wise distance errors following alignment. The algorithm successfully located absent screws in battery modules which demonstrates its ability to assist downstream automation processes. The exact detection of missing components would allow robotic systems to execute autonomous re-screwing operations. The results demonstrate that traditional and deep learning methods can produce highly precise real-time solutions for industrial inspection defect detection.

4.4.2 OWOD vs Supervised models

The assessment between zero-shot and supervised detection systems demonstrates how modern vision systems must choose between general applicability and detection precision through the comparison of e-bike battery and scaffolding pipe dataset results. The OWLv2 + SAM system detection results together with segmentation outcomes reveal specific limitations in the model's capabilities under COCO metric evaluation.

The e-bike battery dataset demonstrated segmentation performance from OWOD at 0.519 when using IoU=0.50 yet this score decreased to 0.264 at IoU=0.75. The model shows successful identification of rough object positions in the results yet fails to achieve precise spatial accuracy which represents a standard problem in zero-shot detection. Bounding box accuracy revealed a similar decline when the evaluation standard shifted to more stringent IoU thresholds. The inherent difficulty of OWOD to detect fine details in objects emerges in screw detection because subtle geometric information is essential. The model shows improved detection performance for small objects although its ability to generalize semantics indicates potential usefulness in limited annotation scenarios and tasks that require unsupervised detection.

The scaffolding pipe dataset presented a significant performance decrease for OWOD system. The model reached 0.0357 mAP at IoU=0.5 and only 0.0235 pixel-level AP for detecting the complex industrial components such as rust patches or mortar stains. Natural image training data for OWOD models fails to provide sufficient transfer learning abilities when detecting domain-specific defects without additional guidance or adaptation.

The YOLOv11 detectors trained with domain-specific data exceeded OWOD performance by a significant margin during both detection and segmentation tasks. All YOLOv11 variants obtained near-saturation accuracy at IoU=0.5 in both datasets while achieving mAP@0.5 scores exceeding 0.96. When using SAM 2.1 for segmentation the detectors produced high-quality mask predictions that reached dice scores of 0.92 on the e-bike battery dataset and 0.68 on the pipe dataset. Supervised models show enhanced performance when trained with RGB+NIR data in scaffolding contexts.

The performance difference between OWOD and supervised models reveals a fundamental weakness of zero-shot methods in achieving precise localization and domain adaptation. The potential of OWOD frameworks remains significant despite their current limitations. OWOD frameworks provide valuable advantages through their ability to detect objects without training by offering flexible deployment in diverse inspection settings. The enhanced recall of small objects together with their shape-level segmentation abilities create possibilities for future development.

4.4.3 RGB vs RGBD

Figure 4-k compares the five YOLOv11 capacities trained on RGB input (blue) with their early-fusion RGB + Depth counterparts (red). The depth-enhanced models exhibit consistently lower performance across all models.

Computationally, the depth channel is almost the same: GFLOPS rise by $< 0.2\,\%$ and single-image inference is also similar. The performance penalty is therefore not a resource issue but a signal-to-noise problem.

The most plausible explanation is noise leakage from the depth channel. The extra plane contributes little meaningful geometry but injects considerable pixel-level variance. Sometimes the depth map can suggest an absence of the screw but it’s clearly visible on the RGB. When concatenated early, this variance forces the first convolutional layer to learn a compromise representation, diluting the clear texture cues of the RGB-only model. Consequently, the network spends capacity reconciling two weakly correlated signals, leading to a lower performance ceiling. Previous research (C. Zhou et al., 2024) shows that a naïve RGB + Depth + Intensity concatenation lags behind a plain RGB detector, but that gap reverses when the auxiliary branches are realigned and re-weighted with their PAlign fusion module, yielding a

modest (~2 mAP-point) boost. Because the screw-head relief in their laptop and HDD datasets is typically < 1 mm, depth alone adds limited information; accordingly, the fusion gain is present but small.

Future work can work towards an algorithm that performs intermediate fusion in a way that is flexible with respect to input modalities—capable of integrating RGB, depth, NIR, or other channels without requiring major architectural changes. This would allow broader applicability across multimodal vision tasks.

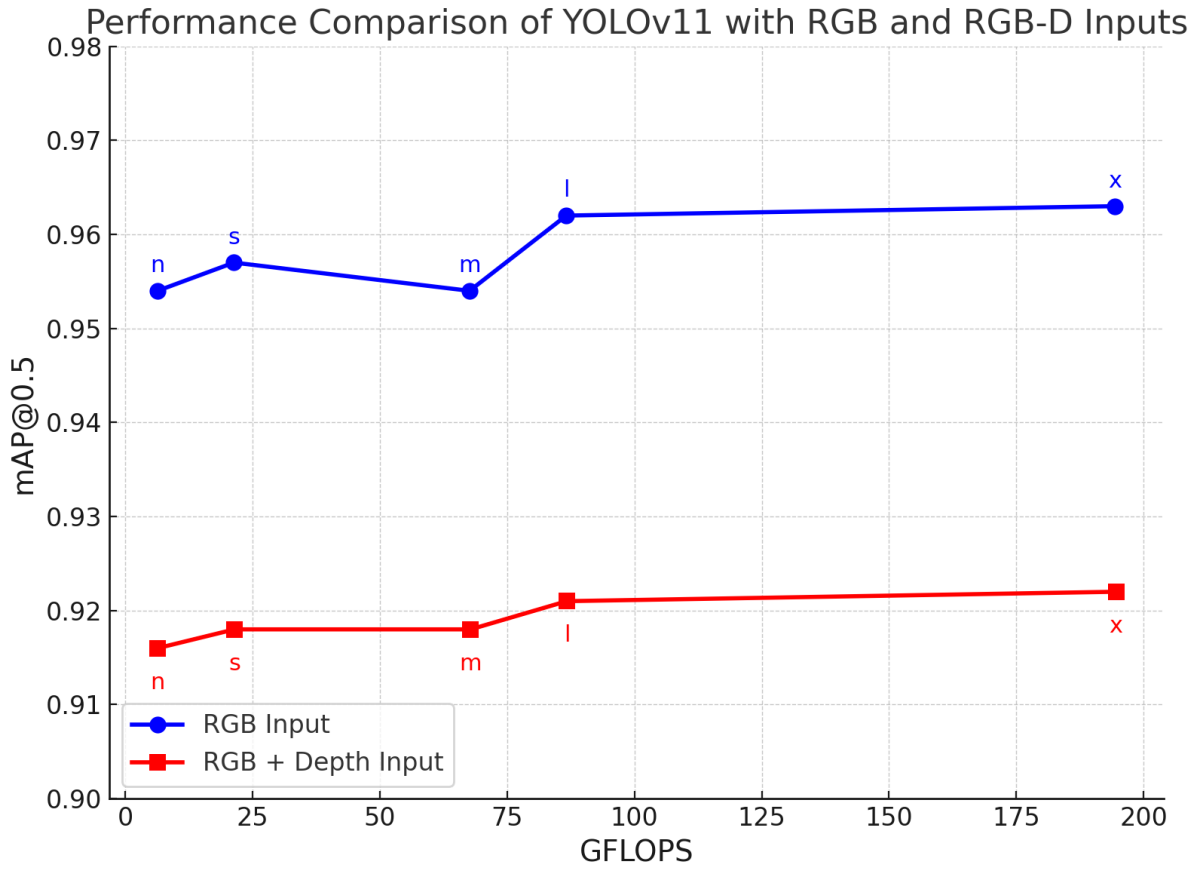


Figure 4-k: Performance comparison of YOLOv11 with RGB and RGB-D inputs for e-bike battery case

4.4.4 RGB vs RGB NIR

Figure 4-l juxtaposes the five YOLOv11 capacities trained on RGB input (blue) with their early-fusion RGB + NIR counterparts (red). In stark contrast to the modest depth performance drops, every model size the multispectral variant yields a consistent gain of ≈ 0.01–0.03 mAP@0.5, with the relative benefit most pronounced for the smallest network (n: +0.03) and tapering off toward the x-capacity (+0.012). Crucially, these accuracy lifts are achieved at almost negligible computational cost: the additional NIR channel increases GFLOPS by <0.2 % and adds ≤0.3 ms to single-image inference.

The improvement can be attributed to the NIR band accentuating mortar residues that are spectrally indistinct from the steel substrate in visible light alone. As model capacity grows, the RGB-only detectors partially recover this contrast through deeper feature hierarchies, explaining the diminishing incremental gain. It is worth noting that this advantage is confined to the detection stage; SAM 2.1 operates on RGB crops only, so no parallel uplift is observed in the mask metrics. Future work could explore multi-spectral aware segmentation back-ends to capitalise on the richer input.

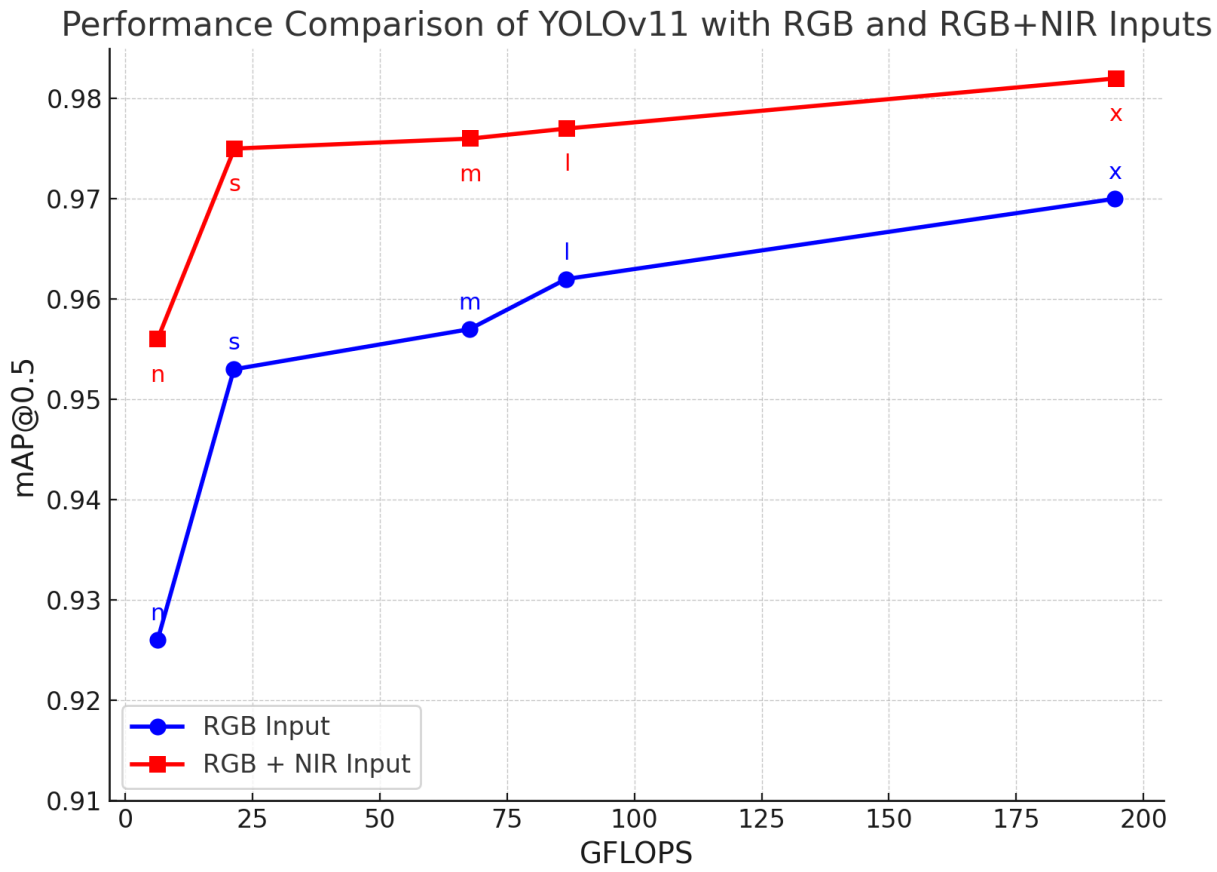


Figure 4-l: Performance comparison of YOLOv11 with RGB and RGB+NIR inputs for pipe case

4.4.5 Preliminary Validation on Crack-Type Defects

To investigate whether the differential-alignment strategy developed for missing-screw detection generalises to other fault types, two artificial cracks were manually introduced into the casing of a single e-bike battery module. The first measured approximately 3.50 mm in width and 23.55 mm in length, penetrating the full wall thickness of the hollow casing. The second was smaller, with dimensions of 1.60 mm width, 21.00 mm length, and 2.40 mm depth. The complete Branch A pipeline was subsequently rerun on the modified module. After coarse–fine ICP alignment, the residual anomaly map (Figure 4-m) revealed that besides the 2 missing screws, a concentrated ridge along the crack path, clearly distinguishable from the surrounding surface was detected as a deviation. While this single exemplar cannot constitute

robust evidence, the successful localisation suggest that the anomaly-mapping approach is sensitive to other small defects as well. It therefore provides encouraging evidence that the method could be extended to a wider spectrum of geometric defects—such as dents, delamination, or excessive wear—provided that sufficient point-density and calibration fidelity are maintained. Systematic validation across a larger defect library remains an important avenue for future work.

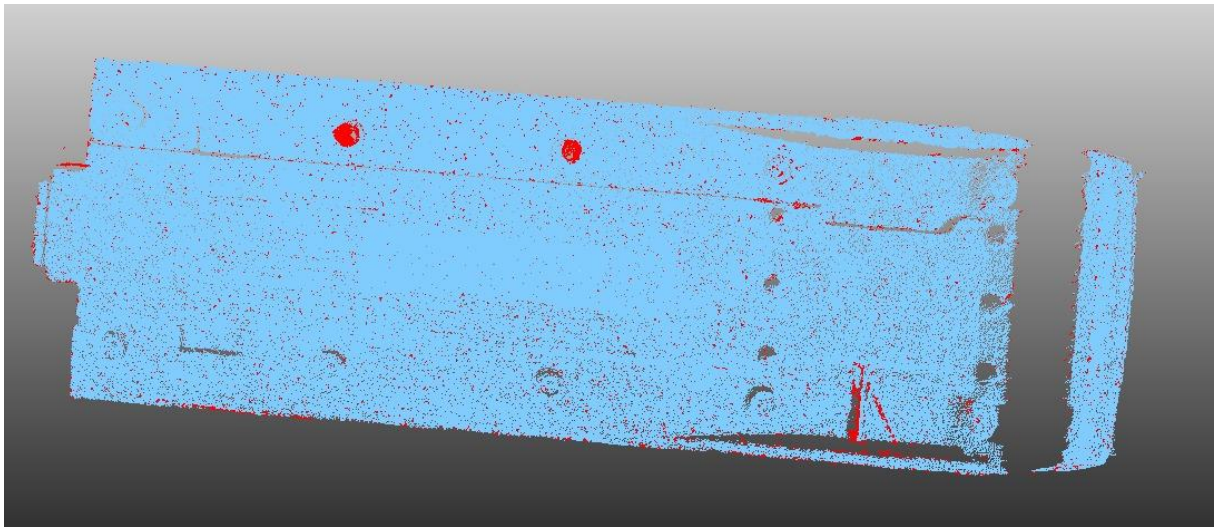


Figure 4-m: Extra analysis for e-bike battery case with multiple defects

4.5 Future work

While the proposed hybrid pipeline achieved strong results across both use cases, several opportunities remain to extend its capabilities. In the 3D inspection branch, the current rough alignment step relies on predefined transformation matrices based on robot pose information. Although effective, this approach limits flexibility and requires precise calibration. Future work could focus on developing a robust and efficient initial alignment method that removes the need for manually encoded transformations while maintaining low runtime. This would simplify deployment in less structured environments and make the 3D pipeline more adaptable to new setups and parts.

For the zero-shot detection and segmentation branch, further improvements could be made by adapting vision-language models with a few labelled examples from the target domain. This few-shot adaptation would enhance precision without sacrificing generalisability to unseen defect types. In addition, prompt engineering strategies that expand or optimise textual inputs could help improve recall in more complex industrial scenes. Finally, extending zero-shot segmentation to operate directly on 3D point clouds using deep learning would remove the dependency on 2D masks and allow for more semantic interpretation of structural defects.

Multimodal fusion also presents opportunities for improvement. While the current early-fusion approach is efficient, intermediate fusion techniques using cross-attention could more effectively combine RGB with auxiliary modalities such as depth or NIR. Developing a model that is flexible on its input channels and does this intermediate fusion enables to assign dynamic importance to each input. Furthermore, the demonstrated benefit of NIR suggests that expanding the spectral range, such as adding ultraviolet (UV), could further improve contrast and detection accuracy, especially for low-visibility defects. The current pipeline is compatible with such extensions, offering a clear path for future upgrades.

5 CONCLUSION

This thesis presented a general, modular hybrid inspection pipeline uniting classical geometry and modern machine vision. Its applicability is demonstrated through two contrasting industrial case studies—e-bike battery housings and scaffolding pipes—showcasing the framework's versatility. Although both branches follow modular design principles, each uses distinct models and methods tailored to its task. The pipeline flexibly accommodates sensor inputs (RGB, depth, NIR) via early-fusion modifications and a custom multi-channel slicer. Branch A combines deep-learning segmentation masks with multi-scale ICP registration for point-cloud alignment, while Branch B integrates zero-shot OWLv2 + SAM with a supervised YOLOv11 fallback for defect localization and segmentation. Key hardware includes conveyor-mounted RGB illumination, halogen lamps for NIR, and a prism-based line-scan RGB+NIR camera. Training and inference ran on an NVIDIA RTX 3090 with standardized calibration and selected hyperparameters for reproducibility and real-time performance.

Empirical evaluation confirmed the approach's efficacy. In 3-D inspection, Fast Point-to-Point ICP balanced speed (0.124 s post-filtering and rough alignment) and accuracy (RMSE ≈ 0.038 mm), while a multiscale CUDA variant halved the error at the cost of runtime. The anomaly-mapping pipeline (voxel downsampling, ICP refinement, KD-Tree scoring) achieved a 0.9999 fitness score and flagged 3 % of points as deviations, correctly identifying missing screws. In 2-D, zero-shot OWLv2 + SAM achieved moderate mask performance (mAP@0.5 ≈ 0.52) on batteries but failed on pipes; supervised YOLOv11 + SAM reached near-saturation box mAP@0.5 (≥ 0.96 on batteries, ≥ 0.97 on pipes) and mask mAP@0.5 up to 0.94, all within sub-10 ms. While OWLv2 performs well on general data, its precision drops in domain-specific tasks, making supervised models essential. Depth inputs offered no benefit and slightly degraded screw detection, while NIR improved pipe results by 1–3 pp at negligible cost.

We showed that integrating deep-learning segmentation with ICP enables the spatial alignment needed for downstream defect analysis. Multi-modal fusion experiments revealed that NIR can improve detection by up to 3 pp in low-contrast pipe defects, whereas depth added little for planar screw inspection, highlighting the context-dependent value of auxiliary modalities. Lastly, while zero-shot OWLv2 + SAM allows out-of-the-box detection, its lower precision in specialized scenes confirms the need for supervised YOLOv11 models in industrial settings. These findings answer our core research questions and provide a foundation for future work in few-shot vision-language adaptation and intermediate-fusion architectures integrating spectral bands or 3-D semantics.

References

3.3.9.7. Otsu thresholding—Scipy lecture notes. (n.d.). Retrieved 20 May 2025, from https://scipy-lectures.org/packages/scikit-image/auto_examples/plot_threshold.html

Adams, R., & Bischof, L. (1994). Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*(6), 641–647. https://doi.org/10.1109/34.295913

Alaa, A. (n.d.). *Week 6: Region Growing and Clustering Segmentation)*. Tutorials for SBME Students. Retrieved 20 May 2025, from https://sbme-tutorials.github.io/2019/cv/notes/6_week6.html

Argirusis, N., Achilleos, A., Alizadeh, N., Argirusis, C., & Sourkouni, G. (2025). IR Sensors, Related Materials, and Applications. *Sensors*, *25*(3), Article 3. https://doi.org/10.3390/s25030673

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615

Besl, P. J., & McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2), 239–256. https://doi.org/10.1109/34.121791

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., Arx, S. von, Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J. Q., Demszky, D., … Liang, P. (2022). *On the Opportunities and Risks of Foundation Models* (No. arXiv:2108.07258). arXiv. https://doi.org/10.48550/arXiv.2108.07258

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G.,

Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language Models are Few-Shot Learners*.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. *Computer Vision – ECCV 2020*, 213–229. https://doi.org/10.1007/978-3-030-58452-8_13

Charles, R. Q., Su, H., Kaichun, M., & Guibas, L. J. (2017). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 77–85. https://doi.org/10.1109/CVPR.2017.16

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2018). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834–848. https://doi.org/10.1109/TPAMI.2017.2699184

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). *A Simple Framework for Contrastive Learning of Visual Representations*.

Çınar, Z. M., Nuhu, A. A., Zeeshan, Q., Korhan, O., Asmael, M., & Safaei, B. (2020). Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0. *Sustainability*, *12*(19), 8211. https://doi.org/10.3390/su12198211

CVAT.ai Corporation. (2023). *Computer Vision Annotation Tool (CVAT)* (Version 2.25.0) [Python]. https://github.com/cvat-ai/cvat (Original work published 2018)

Dice, L. R. (1945). Measures of the Amount of Ecologic Association Between Species. *Ecology*, *26*(3), 297–302. https://doi.org/10.2307/1932409

dos Santos, C. A. T., Lopo, M., Páscoa, R. N. M. J., & Lopes, J. A. (2013). A Review on the Applications of Portable Near-Infrared Spectrometers in the Agro-Food Industry. *Applied Spectroscopy*, *67*(11), 1215–1233. https://doi.org/10.1366/13-07228

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021a). *AN*

IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021b). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* (No. arXiv:2010.11929; Version 2). arXiv. https://doi.org/10.48550/arXiv.2010.11929

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, *88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4

Figure 10: Cameraman image filtered with a Sobel kernel of dimension 3. (n.d.). ResearchGate. Retrieved 20 May 2025, from https://www.researchgate.net/figure/Cameraman-image-filtered-with-a-Sobel-kernel-of-dimension-3_fig9_330502710

Golnabi, H., & Asadpour, A. (2007). Design and application of industrial machine vision systems. *Robotics and Computer-Integrated Manufacturing*, *23*(6), 630–637. https://doi.org/10.1016/j.rcim.2007.02.005

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. https://www.researchgate.net/publication/320703571_Ian_Goodfellow_Yoshua_Bengio_and_Aaron_Courville_Deep_learning_The_MIT_Press_2016_800_pp_ISBN_0262035618

Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jegou, H., & Douze, M. (2021). LeViT: A Vision Transformer in ConvNet's Clothing for Faster Inference. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 12239–12249. https://doi.org/10.1109/ICCV48922.2021.01204

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2021). *Masked Autoencoders Are Scalable Vision Learners* (No. arXiv:2111.06377). arXiv. https://doi.org/10.48550/arXiv.2111.06377

He, K., Chen, X., Xie, S., Li, Y., Dollar, P., & Girshick, R. (2022). Masked Autoencoders Are Scalable Vision Learners. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15979–15988. https://doi.org/10.1109/CVPR52688.2022.01553

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum Contrast for Unsupervised Visual Representation Learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9726–9735. https://doi.org/10.1109/CVPR42600.2020.00975

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2018). *Mask R-CNN* (No. arXiv:1703.06870; Version 3). arXiv. https://doi.org/10.48550/arXiv.1703.06870

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2020). Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(2), 386–397. https://doi.org/10.1109/TPAMI.2018.2844175

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., & Stuetzle, W. (1992). Surface reconstruction from unorganized points. *SIGGRAPH Comput. Graph.*, *26*(2), 71–78. https://doi.org/10.1145/142920.134011

Horowitz, S. L., & Pavlidis, T. (1976). Picture Segmentation by a Tree Traversal Algorithm. *Journal of the ACM*, *23*(2), 368–388. https://doi.org/10.1145/321941.321956

Huang, Q., Tian, H., Jia, L., Li, Z., & Zhou, Z. (2023). A review of deep learning segmentation methods for carotid artery ultrasound images. *Neurocomputing*, *545*, 126298. https://doi.org/10.1016/j.neucom.2023.126298

Huang, Z., Liu, H., Zhang, H., Li, X., Liu, H., Xing, F., Laine, A., Angelini, E., Hendon, C., & Gan, Y. (2023). *Push the Boundary of SAM: A Pseudo-label Correction Framework for Medical Segmentation* (No. arXiv:2308.00883). arXiv. https://doi.org/10.48550/arXiv.2308.00883

HumanSignal/label-studio. (2025). [JavaScript]. HumanSignal. https://github.com/HumanSignal/label-studio (Original work published 2019)

Image Segmentation: Deep Learning vs Traditional [Guide]. (n.d.). Retrieved 20 May 2025, from https://www.v7labs.com/blog/image-segmentation-guide

Image segmentation detailed overview [Updated 2024]. (n.d.). SuperAnnotate. Retrieved 20 May 2025, from https://www.superannotate.com/blog/image-segmentation-for-machine-learning

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q. V., Sung, Y., Li, Z., & Duerig, T. (2021). *Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision* (No. arXiv:2102.05918). arXiv. https://doi.org/10.48550/arXiv.2102.05918

Jing, L., & Tian, Y. (2021). Self-Supervised Visual Feature Learning With Deep Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(11), 4037–4058. https://doi.org/10.1109/TPAMI.2020.2992393

Joseph, K. J., Khan, S., Khan, F. S., & Balasubramanian, V. N. (2021). Towards Open World Object Detection. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5826–5836. https://doi.org/10.1109/CVPR46437.2021.00577

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, *1*(4), 321–331. https://doi.org/10.1007/BF00133570

Kazhdan, M., Bolitho, M., & Hoppe, H. (2006). *Poisson surface reconstruction*.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in Vision: A Survey. *ACM Computing Surveys*, *54*(10s), 1–41. https://doi.org/10.1145/3505244

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). *Segment Anything* (No. arXiv:2304.02643). arXiv. https://doi.org/10.48550/arXiv.2304.02643

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Li, W., Hsu, C.-Y., Wang, S., Yang, Y., Lee, H., Liljedahl, A., Witharana, C., Yang, Y., Rogers, B. M., Arundel, S. T., Jones, M. B., McHenry, K., & Solis, P. (2024). Segment Anything Model Can Not Segment Anything: Assessing AI Foundation Model's Generalizability in Permafrost Mapping. *Remote Sensing*, *16*(5), Article 5. https://doi.org/10.3390/rs16050797

Li, X., Deng, R., Tang, Y., Bao, S., Yang, H., & Huo, Y. (2023). *Leverage Weakly Annotation to Pixel-wise Annotation via Zero-shot Segment Anything Model for Molecular-empowered Learning* (No. arXiv:2308.05785). arXiv. https://doi.org/10.48550/arXiv.2308.05785

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. *Computer Vision – ECCV 2014*, 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., & Zhang, L. (2024). Grounding DINO: Marrying DINO with Grounded Pre-training for Open-Set Object Detection. *Computer Vision – ECCV 2024*, 38–55. https://doi.org/10.1007/978-3-031-72970-6_3

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 9992–10002. https://doi.org/10.1109/ICCV48922.2021.00986

Liu, Z., Xin, X., Xu, Z., Zhou, W., Wang, C., Chen, R., & He, Y. (2023). Robust and Accurate Feature Detection on Point Clouds. *Computer-Aided Design*, *164*, 103592. https://doi.org/10.1016/j.cad.2023.103592

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, *60*(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

LSR S-GL. (n.d.). Retrieved 21 May 2025, from https://docs.mech-mind.net/en/eye-3d-camera/latest/hardware/specifications-lsr-s.html

Luo, J., Yang, Z., Cao, Y., Wen, T., & Li, D. (2025). RT-DETR-MCDAF: Multimodal Fusion of Visible Light and Near-Infrared Images for Citrus Surface Defect Detection in the Compound Domain. *Agriculture*, *15*(6), Article 6. https://doi.org/10.3390/agriculture15060630

Luo, Y., & Luo, Z. (2023). Infrared and Visible Image Fusion: Methods, Datasets, Applications, and Prospects. *Applied Sciences*, *13*(19), 10891. https://doi.org/10.3390/app131910891

MATHEW, E. E., & CHHABRA, D. (n.d.). *Product Condition Evaluation by a 3D Vision System on a Robot for Robotic Disassembly of Electric Vehicle Batteries and Bike Batteries*. KU Leuven.

Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, 565–571. https://doi.org/10.1109/3DV.2016.79

Minderer, M., Gritsenko, A., & Houlsby, N. (2024). *Scaling Open-Vocabulary Object Detection* (No. arXiv:2306.09683). arXiv. https://doi.org/10.48550/arXiv.2306.09683

Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., Wang, X., Zhai, X., Kipf, T., & Houlsby, N. (2022). *Simple Open-Vocabulary Object Detection with Vision Transformers* (No. arXiv:2205.06230). arXiv. https://doi.org/10.48550/arXiv.2205.06230

Muja, M., & Lowe, D. G. (2009). FAST APPROXIMATE NEAREST NEIGHBORS WITH AUTOMATIC ALGORITHM CONFIGURATION: *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, 331–340. https://doi.org/10.5220/0001787803310340

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

(PDF) A Deep Neural Network for Oil Spill Semantic Segmentation in Sar Images. (2025, May 19). *ResearchGate*. https://doi.org/10.1109/ICIP.2018.8451113

(PDF) Comparison of Fully Convolutional Networks (FCN) and U-Net for Road Segmentation from High Resolution Imageries. (2025). *ResearchGate*. https://doi.org/10.30897/ijegeo.737993

Pomerleau, F., Colas, F., & Siegwart, R. (2015). *A Review of Point Cloud Registration Algorithms for Mobile Robotics*. https://ieeexplore.ieee.org/document/8187578

PRO S-GL and PRO M-GL. (n.d.). Retrieved 21 May 2025, from https://docs.mech-mind.net/en/eye-3d-camera/latest/hardware/specifications-pro-s-pro-m.html

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). *Learning Transferable Visual Models From Natural Language Supervision*.

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024a). *SAM 2: Segment Anything in Images and Videos* (No. arXiv:2408.00714). arXiv. https://doi.org/10.48550/arXiv.2408.00714

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024b). *SAM 2: Segment Anything in Images and Videos* (No. arXiv:2408.00714). arXiv. https://doi.org/10.48550/arXiv.2408.00714

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Ren, Z., Fang, F., Yan, N., & Wu, Y. (2022). State of the Art in Defect Detection Based on Machine Vision. *International Journal of Precision Engineering and Manufacturing-Green Technology*, *9*(2), 661–691. https://doi.org/10.1007/s40684-021-00343-6

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

Rosenfeld, A., & De La Torre, P. (1983). Histogram concavity analysis as an aid in threshold selection. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-13*(2), 231–235. https://doi.org/10.1109/TSMC.1983.6313118

Rusinkiewicz, S., & Levoy, M. (2001). Efficient variants of the ICP algorithm. *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, 145–152. https://doi.org/10.1109/IM.2001.924423

Rusu, R. B., & Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). *2011 IEEE International Conference on Robotics and Automation*, 1–4. https://doi.org/10.1109/ICRA.2011.5980567

Sabri, V., Tahan, S. A., Pham, X. T., Moreau, D., & Galibois, S. (2016). Fixtureless profile inspection of non-rigid parts using the numerical inspection fixture with improved definition of displacement boundary conditions. *The International Journal of Advanced Manufacturing Technology*, *82*(5–8), 1343–1352. https://doi.org/10.1007/s00170-015-7425-3

Schnabel, R., Wahl, R., & Klein, R. (2007). Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, *26*(2), 214–226. https://doi.org/10.1111/j.1467-8659.2007.01016.x

Schraml, D., & Notni, G. (2024). Synthetic Training Data in AI-Driven Quality Inspection: The Significance of Camera, Lighting, and Noise Parameters. *Sensors (Basel, Switzerland)*, *24*(2), 649. https://doi.org/10.3390/s24020649

Serra, J., & Cressie. (1982). *Image analysis and mathematical morphology*. https://www.researchgate.net/publication/238273912_Image_analysis_and_mathematical_morphology_by_J_Serra_Academic_Press_London_1982_xviii_610_p_9000

Shelhamer, E., Long, J., & Darrell, T. (2017). Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(4), 640–651. https://doi.org/10.1109/TPAMI.2016.2572683

Simonyan, K., & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition* (No. arXiv:1409.1556). arXiv. https://doi.org/10.48550/arXiv.1409.1556

Sobel, E. (1970). *CAMERA MODELS AND MACHINE PERCEPTION - ProQuest*. https://www.proquest.com/openview/c051205a85f18112e06cb51e57d9379e/1?cbl=18750&diss=y&parentSessionId=%2BAc9JkIQMi%2BjRvMoHUI5gquXRXJuMc7Avs8wz8XPsls%3D&pq-origsite=gscholar&accountid=17215

Steger, C., Ulrich, M., & Wiedemann, c. (2018). *Machine Vision Algorithms and Applications*. https://www.researchgate.net/publication/322754740_Machine_Vision_Algorithms_and_Applications

Strudel, R., Garcia, R., Laptev, I., & Schmid, C. (2021). Segmenter: Transformer for Semantic Segmentation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 7242–7252. https://doi.org/10.1109/ICCV48922.2021.00717

Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep High-Resolution Representation Learning for Human Pose Estimation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5686–5696. https://doi.org/10.1109/CVPR.2019.00584

SuperAnnotate | Centralize Data Ops for Multimodal AI. (n.d.). SuperAnnotate. Retrieved 21 May 2025, from https://www.superannotate.com

U-Net—An overview | ScienceDirect Topics. (n.d.). Retrieved 20 May 2025, from https://www.sciencedirect.com/topics/computer-science/u-net

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (n.d.). *Attention is All you Need*.

Vincent, L., & Soille, P. (1991). Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*(6), 583–598. https://doi.org/10.1109/34.87344

Wang, W. (2025). *X-AnyLabeling* [Python]. https://github.com/ultralytics/ultralytics (Original work published 2023)

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., & Shao, L. (2021). Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 548–558. https://doi.org/10.1109/ICCV48922.2021.00061

Wang, Y., & Solomon, J. (2019). Deep Closest Point: Learning Representations for Point Cloud Registration. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3522–3531. https://doi.org/10.1109/ICCV.2019.00362

Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I. S., & Xie, S. (2023). ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16133–16142. https://doi.org/10.1109/CVPR52729.2023.01548

Zhang, Z., & Zhu, L. (2023). A Review on Unmanned Aerial Vehicle Remote Sensing: Platforms, Sensors, Data Processing Methods, and Applications. *Drones*, *7*(6), 398. https://doi.org/10.3390/drones7060398

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017a). Pyramid Scene Parsing Network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6230–6239. https://doi.org/10.1109/CVPR.2017.660

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017b). *Pyramid Scene Parsing Network* (No. arXiv:1612.01105; Version 2). arXiv. https://doi.org/10.48550/arXiv.1612.01105

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H. S., & Zhang, L. (2021a). Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6877–6886. https://doi.org/10.1109/CVPR46437.2021.00681

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H. S., & Zhang, L. (2021b). *Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers* (No. arXiv:2012.15840; Version 3). arXiv. https://doi.org/10.48550/arXiv.2012.15840

Zhong, Y., Yang, J., Zhang, P., Li, C., Codella, N., Li, L. H., Zhou, L., Dai, X., Yuan, L., Li, Y., & Gao, J. (2022). RegionCLIP: Region-based Language-Image Pretraining. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16772–16782. https://doi.org/10.1109/CVPR52688.2022.01629

Zhou, C., Wu, Y., Sterkens, W., Piessens, M., Vandewalle, P., & Peeters, J. R. (2024). Towards robotic disassembly: A comparison of coarse-to-fine and multimodal fusion screw detection methods. *Journal of Manufacturing Systems*, *74*, 633–646. https://doi.org/10.1016/j.jmsy.2024.04.024

Zhou, Q.-Y., Park, J., & Koltun, V. (2018). *Open3D: A Modern Library for 3D Data Processing* (No. arXiv:1801.09847). arXiv. https://doi.org/10.48550/arXiv.1801.09847

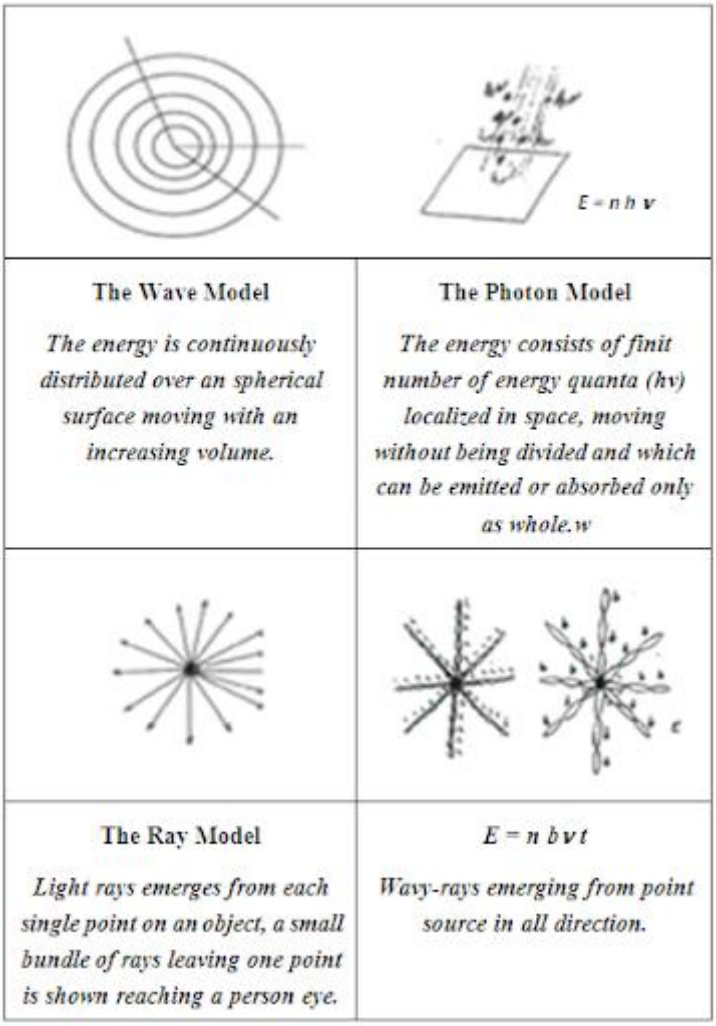
Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., & Misra, I. (2022). Detecting Twenty-Thousand Classes Using Image-Level Supervision. *Computer Vision – ECCV 2022*, 350–368. https://doi.org/10.1007/978-3-031-20077-9_21

Appendices

APPENDIX A: ADDITIONAL LITERATURE IMAGE ACQUISITION

Understanding Light and Its Interaction with Matter

To grasp how we capture images, it's essential first to understand how light makes objects visible. We perceive an object when light emitted or reflected by it reaches our eyes. This light may originate directly from the object (if it's self-luminous) or may be reflected from external illumination sources (Young & Freedman, 2022).



In physics, light can be interpreted through three fundamental models: the particle model, the wave model, and the geometric (ray) model (Figure 1). These models offer distinct but complementary ways to explain optical phenomena. For example, wave theory explains interference and diffraction; particle theory accounts for quantum behaviors like the

photoelectric effect; and ray theory provides practical tools for understanding lenses and reflection (Hecht, 2017).

It's important to note that these frameworks are not mutually exclusive. Instead, they are applied depending on the phenomenon under study and the level of precision required. This interplay of models is often called wave-particle duality, where light exhibits both continuous wave-like and discrete particle-like behavior.

The Spectrum of Light

The electromagnetic spectrum spans an enormous range of wavelengths, from gamma rays (shortest) to radio waves (longest), with only a narrow window known as the visible spectrum detectable by the human eye (roughly 380–750 nm) (Pedrotti & Pedrotti, 2019). Different wavelengths within this range correspond to different perceived colors, with shorter wavelengths appearing bluish and longer ones reddish.

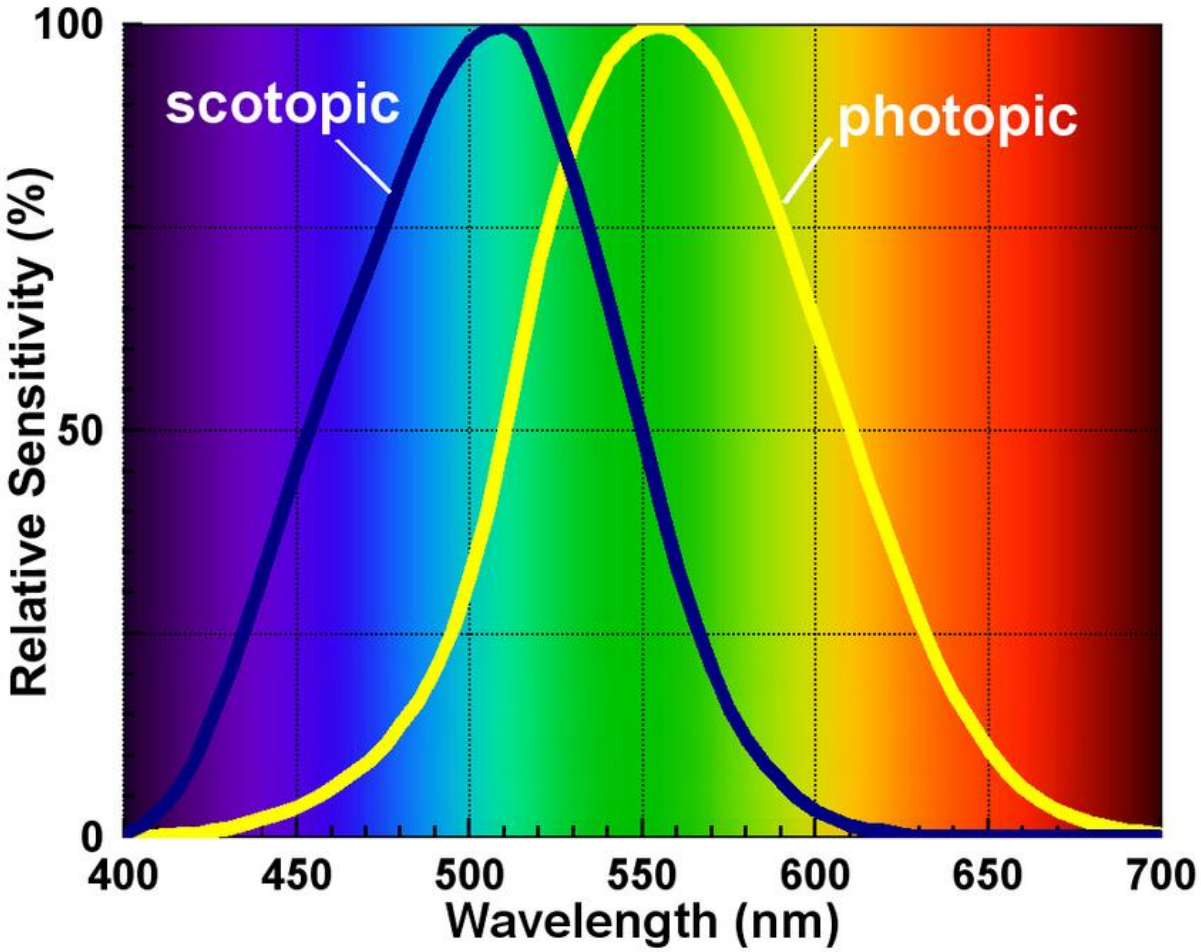
Light also has an intensity component, which can be described in various ways depending on context: energy per unit area, number of photons, or power per unit time (Saleh & Teich, 2019). These quantities become critical when designing sensors or cameras to optimize for lighting conditions.

Measuring Light: Radiometry and Photometry

Two main frameworks describe light measurement:

- Radiometry quantifies all electromagnetic radiation, independent of human perception, using units like watts and joules.
- Photometry adjusts these measurements to account for human visual sensitivity, using weighted measures like lumens and lux (CIE, 2018).

This adjustment matters because human vision does not respond equally to all wavelengths — for example, we are more sensitive to green light under bright conditions (photopic vision) and more sensitive to blue light under low-light (scotopic vision) (Figure 2).



Mathematically, photometric quantities are derived by weighting radiometric measurements with the luminosity function, reflecting the eye's spectral response. This provides practical measures that align better with how humans actually experience brightness and color (Houser & Wei, 2017).

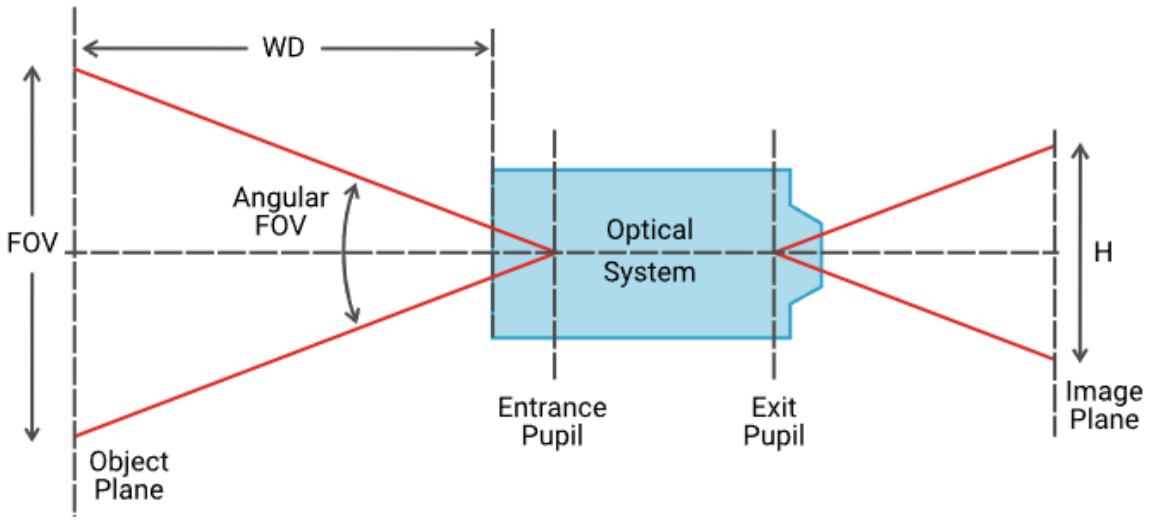
Optical Principles: Lenses, Focus, and Field of View

A key parameter in image capture is the focal length of a lens, which determines how strongly it bends light rays to form an image. This focal length, usually measured in millimeters, defines how magnified or wide the field of view will be.

The field of view (FoV) is inversely related to focal length — longer focal lengths zoom in on a narrow area, while shorter focal lengths capture a broader scene (Jenkins & White, 2001).

Another critical distance is the working distance: the space between the lens and the object required to fit the object fully within the image frame. Working distance depends on focal length,

sensor size, and desired image magnification and can be roughly estimated using geometric optics formulas.

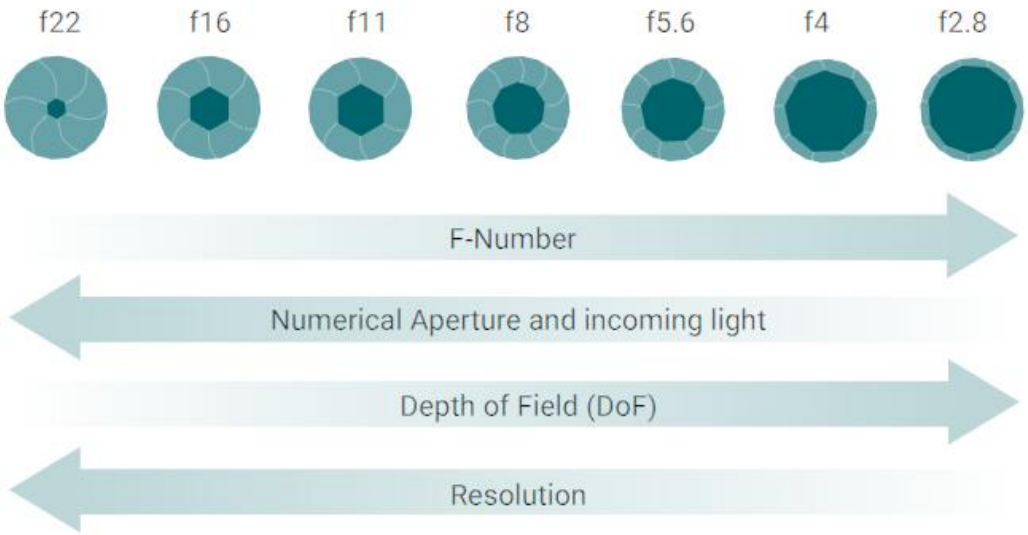


Aperture, Depth of Field, and Light Control

The aperture or opening within the lens controls how much light reaches the sensor. This is described by the f-number (f/#), which is the ratio of the lens's focal length to the aperture diameter.

A lower f-number (wide aperture) allows more light in but produces a shallow depth of field — only a narrow range is sharply in focus. A higher f-number (narrower aperture) reduces light intake but extends the depth of field, bringing more of the scene into sharp focus (Ray, 2002).

In imaging setups, balancing aperture size, exposure time, and illumination is key. Often, strong artificial lighting is used to enable small apertures and deep focus, ensuring that even fine object details remain crisp.



Beyond optics, sensor technology plays a crucial role in image acquisition. Sensors such as CCD or CMOS arrays convert incoming light into electrical signals. Key parameters here include:

- Pixel size: Smaller pixels increase spatial resolution but reduce light sensitivity.
- Dynamic range: The ability to capture both dark and bright areas without losing detail.
- Noise performance: Determines how well the sensor can distinguish signal from background noise, critical under low-light conditions (Janesick, 2001).

Higher resolution sensors can capture more detail, but they also demand careful optical matching to avoid over- or undersampling.

APPENDIX B: TECHNICAL FILES OF LIGHTS AND CAMERAS

UL Collimated Line Light PRODUCT DATA SHEET METAPHASE LIGHTING TECHNOLOGIES

UL-CLL

Using Metaphase's proprietary optics, the UL-Collimated Line Light works like a traditional line light but can project its light at a lower divergence angle. This helps keep a tight, structured beam of light at longer working distances, possibly many meters away while delivering high intensity light and for applications where you cannot have the light close to the object or web. These features are useful for creating clearances for robotics, imaging high tunnel ceilings, creating distances from hot objects or applications requiring precise backlight imaging. The UL-Collimated Line Light is a great tool for any line scan or machine vision lighting application.

FEATURES

- Proprietary collimated lens for low divergence angle
- High intensity, over 1.5 million Lux
- Lens options for various beam widths and working distances
- Lengths range from 5 inches to 120 inches
- High Uniformity using Metaphase's unique diffuser technology
- 24Z version has a built-in Constant Current Driver(s) with 0-10VDC Intensity Control (1kHz max gating)
- Compatible with DDC-3 and ILD-35 Dimmers
- Compliance: CE and ROHS

APPLICATIONS

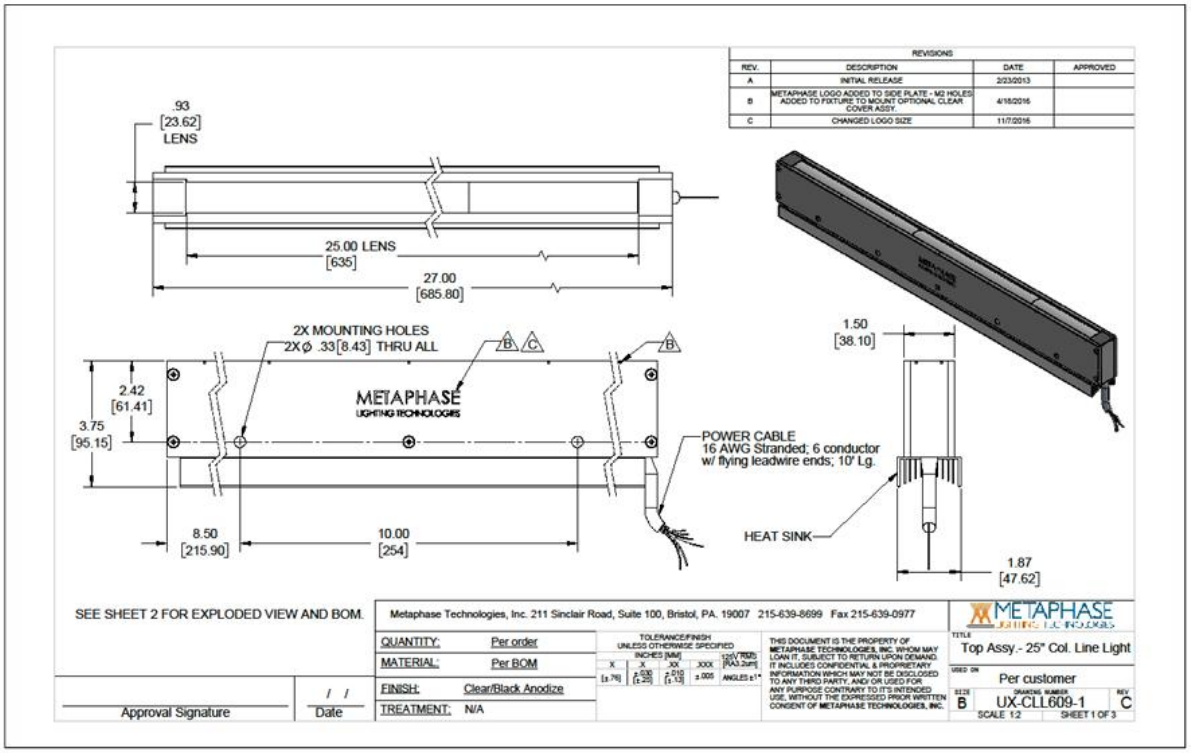
- Geometry measurements of long targets
- Direct position measurement
- Web inspection (foil, plastic film, PCBs, glass, paper, web converting, etc.)
- Printed circuit board inspection
- Long working distance applications such as tunnels, rails, or ceiling inspections
- Tube or bottle inspection
- Quality control in food, chemicals, and textile industry
- Fabrics
- Solar Cells

GENERAL SPECIFICATIONS

Power Source:	24VDC ±5%
Cable (Standard):	10-foot cable with flying leads
Housing:	Black Anodized Aluminum
Ambient Temp:	-20°C to 40°C
Lifetime Expectancy:	75,000 hours (except UV)

UL-CLL

Sample Model Number: **UL-CLL609-W-24Z**



Specifications (PRO S)

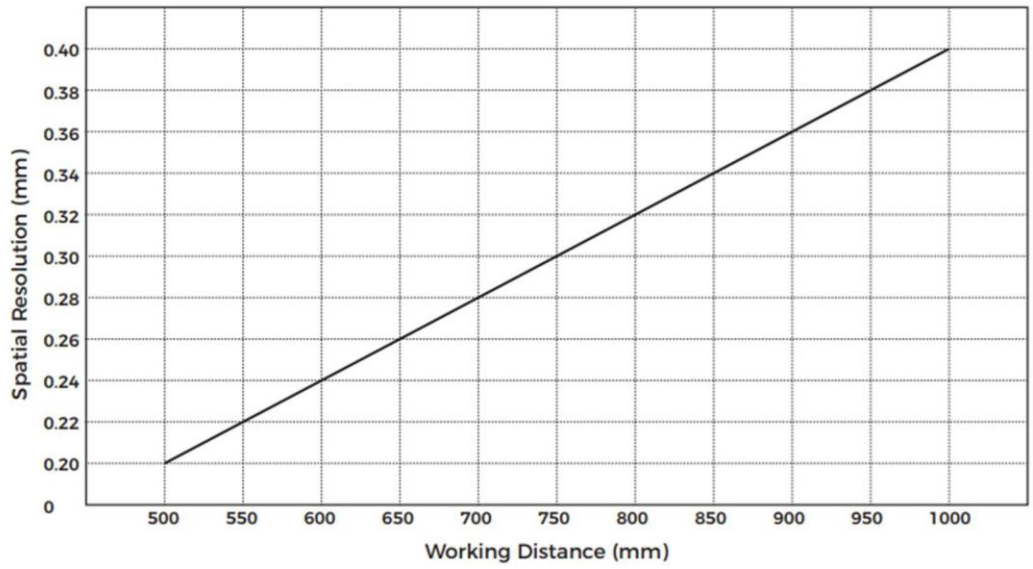
Product name	Mech-Eye Industrial 3D Camera		
Model	PRO S		
Object focal distance[1]	500 mm	700 mm	1000 mm
Recommended working distance	500–600 mm	600–800 mm	800–1000 mm
FOV	370 × 240 mm @ 0.5 m		
	800 × 450 mm @ 1 m		
Resolution	1920 × 1200		
Megapixels	2.3 MP		
Point Z-value repeatability (σ)[2]	0.05 mm @ 1 m		
Measurement accuracy (VDI/VDE)[3]	0.1 mm @ 0.5 m		
Typical capture time	0.3–0.6 s		
Weight	Approx. 1.6 kg		
Baseline	Approx. 180 mm		
Dimensions	Approx. 265 × 57 × 100 mm		
2D image color	Monochrome	Color	
Light source	Blue LED (459 nm, RG2)	White LED (RG2)	
Operating temperature	0–45°C		
Communication interface	Gigabit Ethernet		
Input[4]	24 V ⎓ 3.75 A		
Safety and EMC	CE / FCC / VCCI / UKCA / KC		
IP rating[5]	IP65		
Cooling	Passive		

[1] The camera is available in three object focal distances, each corresponding to a different range of recommended working distance.
[2] One standard deviation of 100 Z-value measurements of the same point. Measurement target was a ceramic plate.
[3] According to VDI/VDE 2634 Part II.
[4] ⎓ : Direct current.
[5] Test implemented based on IEC 60529. 6: dust-tight; 5: protected against water jets.

01

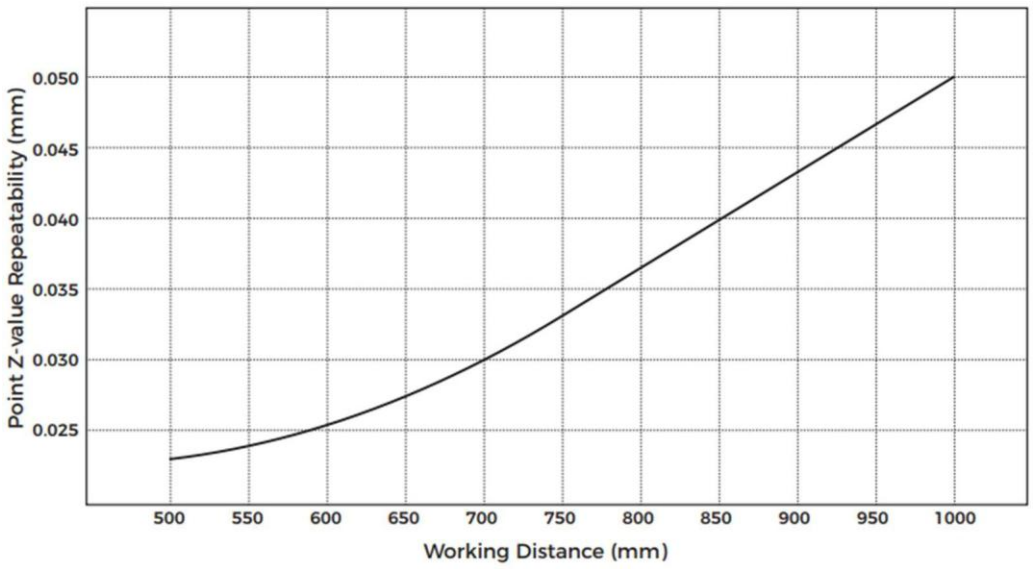
Spatial Resolution

PRO S



Point Z-value Repeatability

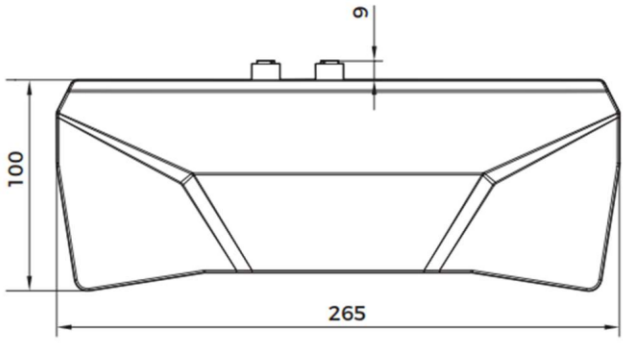
PRO S

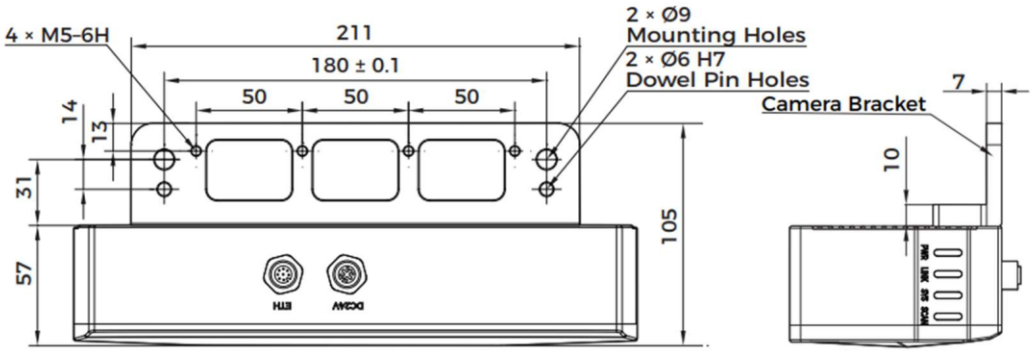


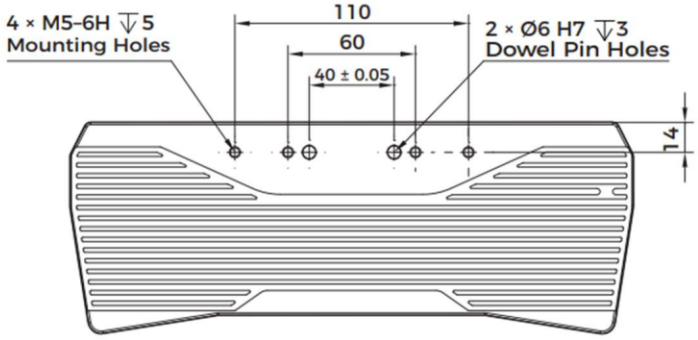
Dimensions

Unit: mm

PRO S







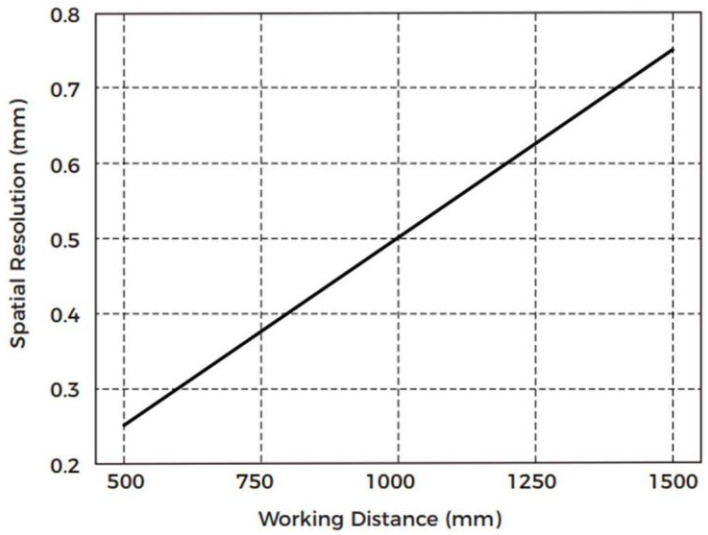
05

Specifications (LSR S)

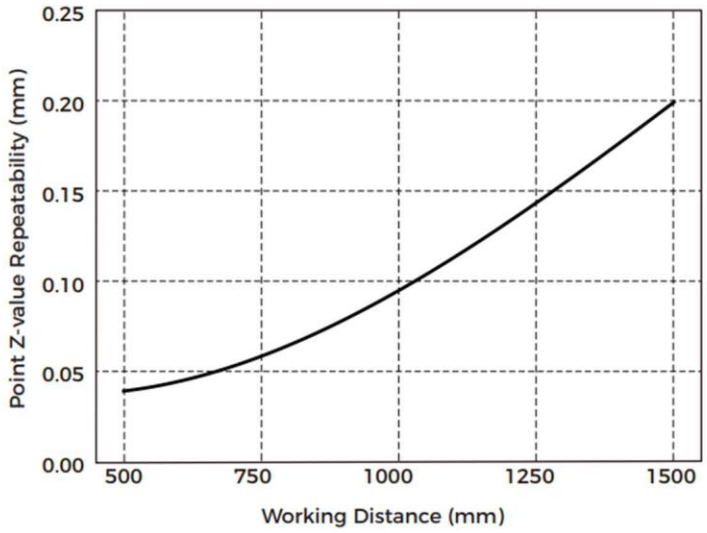
Product name	Mech-Eye Industrial 3D Camera	
Model	LSR S	
Object focal distance[1]	800 mm	1400 mm
Recommended working distance	500–900 mm	900–1500 mm
FOV	480 × 360 mm @ 0.5 m	
	1500 × 1200 mm @ 1.5 m	
Depth map resolution	2048 × 1536	
RGB resolution	4000 × 3000 / 2000 × 1500	
Point Z-value repeatability (σ)[2]	0.2 mm @ 1.5 m	
Measurement accuracy (VDI/VDE)[3]	1.0 mm @ 1.5 m	
Typical capture time	0.5–0.9 s	
Weight[4]	Approx. 1.9 kg	
Baseline	Approx. 140 mm	
Dimensions[4]	Approx. 228 × 77 × 126 mm	
Light source	Red laser (638 nm, Class 2)	
Operating temperature[5]	-10–45°C	
Communication interface	Gigabit Ethernet	
Input[6]	24 V ⎓ 3.75 A	
Safety and EMC	CE / FCC / VCCI / UKCA / KC	
IP rating[7]	IP67	
Cooling	Passive	

[1] The camera is available in two object focal distances, each corresponding to a different range of recommended working distance.
[2] One standard deviation of 100 Z-value measurements of the same point. Measurement target was a ceramic plate.
[3] According to VDI/VDE 2634 Part II.
[4] Excluding the heat-dissipation panel and camera bracket.
[5] The operating temperature of the camera without the heat-dissipation panel.
[6] ⎓ : Direct current.
[7] Test implemented based on IEC 60529. 6: dust-tight; 7 : protected against water jets.

Spatial Resolution



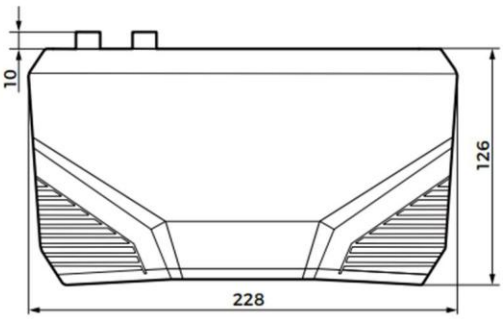
Point Z-value Repeatability

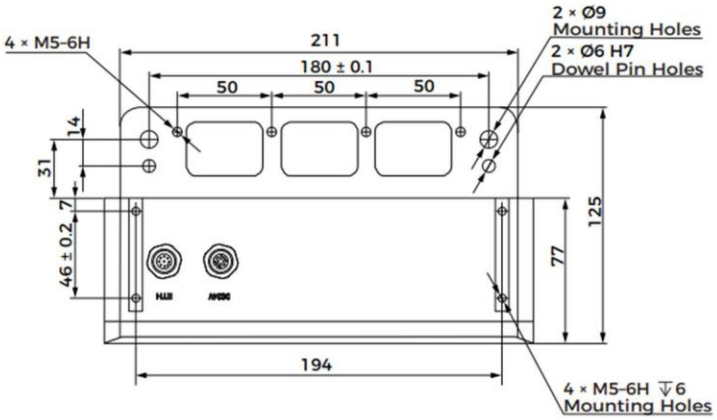


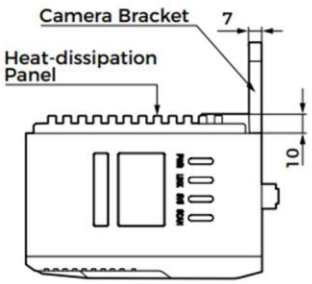
02

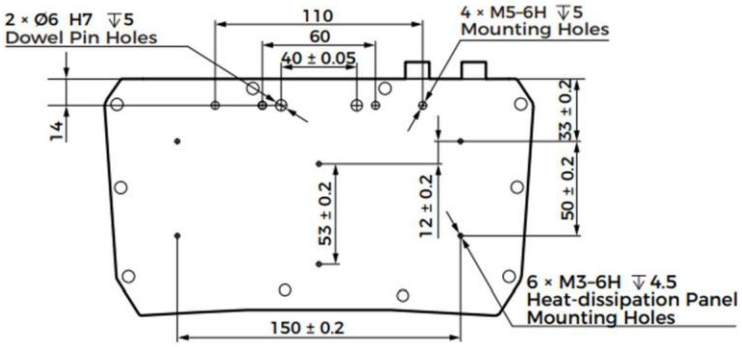
Dimensions

Unit: mm









03

Specifications for SW-4000Q-10GE

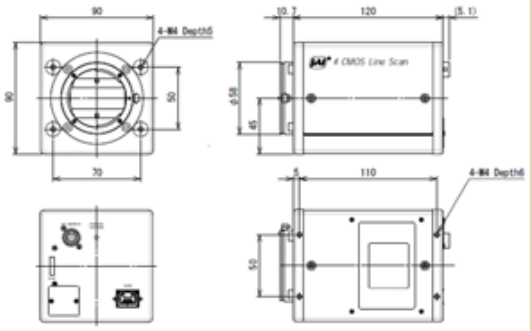
Sweep+ Series

Specifications	SW-4000Q-10GE
Scanning system	4 high-speed CMOS line sensors, prism-mounted
Active pixels	4 x 4096 pixels (R, G, B, NIR)
Line rate (full width)	Up to 72 kHz (variable) for 8-bit RGB + NIR 74 kHz possible with YUV compression
Sensor width	30.72 mm
Pixel size	Mode A: 7.5 µm x 7.5 µm Mode B: 7.5 µm x 10.5 µm
Ethernet speeds	10GBASE-T, 5GBASE-T, 2.5GBASE-T, 1000BASE-T Full backwards compatibility
Video output	Single stream: RGB8 Two streams: RGB8, RGB10V1Packed, RGB10p32, YUV422_8_UVYV, YUV422_8 (visible) Mono8, Mono10Packed (NIR)
Object illuminance (min.)	300 lx @ 7800 K, Mode A (Gain 18 dB, 525 µs exp., 50% video, f/2.8)
Responsivity	RGB: 118 DN/nJ/cm² @ 550 nm (G channel) NIR: 64 DN/nJ/cm² @ 800 nm (Mode A, 10-bit, 0 dB gain)
S/N ratio	>55 dB on green, 10-bit with 0 dB gain
Inputs (Trigger)	1 Opto In + 1 TTL via 12-pin, 2 TTL via 10-pin, Pulse Generator (4), NAND Out (2), Action (4), User Out (4)
Outputs	2 TTL via 12-pin, 2 TTL via 10-pin
Gain	Analog Base Gain: 0 dB / 6 dB / 12 dB Digital Master: 0 to +18 dB, R/B/NIR: -4 to +12 dB Digital Individual: 0 to +24 dB
White balance	Manual/one-push auto by gain or exposure
Gamma	0.45 to 1.0 (9 steps) or 257-point LUT
Image processing	PRNU/DSNU, black level, flat shading and color shading correction, chromatic aberration adjustment, horizontal mirroring
Color space conversion	RGB or RGBa8 to HSI, XYZ (CIE), sRGB, Adobe RGB, or User Custom RGB
Exposure modes	No shutter, timed, and trigger width control
Electronic shutter	3 µs to 13889 µs in 1 µs increments at 72 kHz. Exposure time can be longer at slower line rates.
Pulse width control	1.8 µs to ~1 sec
Time synchronization	Support for Precision Time Protocol (IEEE 1588)
Lens mount	M52 mount or Nikon F-mount (46.5 mm flange back for both mounts)
Operating temp. (ambient)	-5°C to +45°C (20 to 80% non-condensing)
Storage temp. (ambient)	-25°C to +60°C (20 to 80% non condensing)
Vibration	3G (20 Hz to 200 Hz, XYZ directions)
Shock	50G
Regulations	CE (EN61000-6-2, EN61000-6-3) FCC Part 15 Class B, RoHS/WEEE
Power 12-pin PoE	+10V to +25V DC. 19.3 W typical @ 12V Not supported.
Dimensions (H x W x L)	(without connector and lens mount protrusions) 90 mm x 90 mm x 120 mm
Weight	980 g

Ordering Information

SW-4000Q-10GE-F	4-CMOS prism line scan camera with F-mount
SW-4000Q-10GE-M52	4-CMOS prism line scan camera with M52 mount

Dimensions (F-mount)



F-mount model shown. For M52 drawings and dimensions, see manual.

Connector pin-out

DC In / Trigger

HIROSE HR10A-10R-12PB(71)

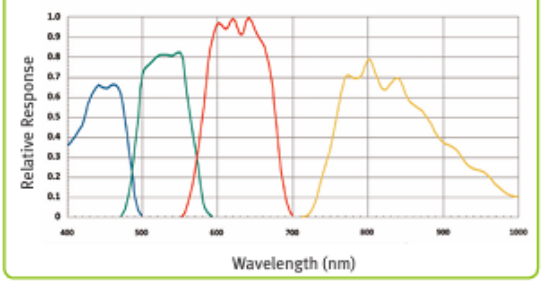
Pin		
	1	Ground
	2	DC in +10V to +25V
	3	Ground
	4	Reserved
	5	Opto in -
	6	Opto in +
	7	TTL out 4
	8	NC
	9	TTL out 1
	10	TTL in 1
	11	DC in +10V to +25V
	12	Ground

GigE Vision Interface

RJ-45 with locking screws

Pin	Signal
1	TRD+ (0)
2	TRD- (0)
3	TRD+ (1)
4	TRD+ (2)
5	TRD- (2)
6	TRD- (1)
7	TRD+ (3)
8	TRD- (3)

Spectral response



APPENDIX C: CONVEYOR SYSTEM SETUP

How to set up the system

C.1 Configuring Ports for PLC and 3D Camera Connection

The RGB+NIR camera (model SW-400Q-10GE by JAI) requires a high data transfer bandwidth (~10 Gbps). Therefore, it is connected directly to the bottom Ethernet adapter card on the lab computer. The PLC (Programmable Logic Controller) and 3D camera, which were initially connected to this bottom adapter through an Ethernet switch, are relocated to the top Ethernet adapter card (1 Gbps ports) to free up the higher-speed connection.

To ensure proper configuration of the Ethernet ports:

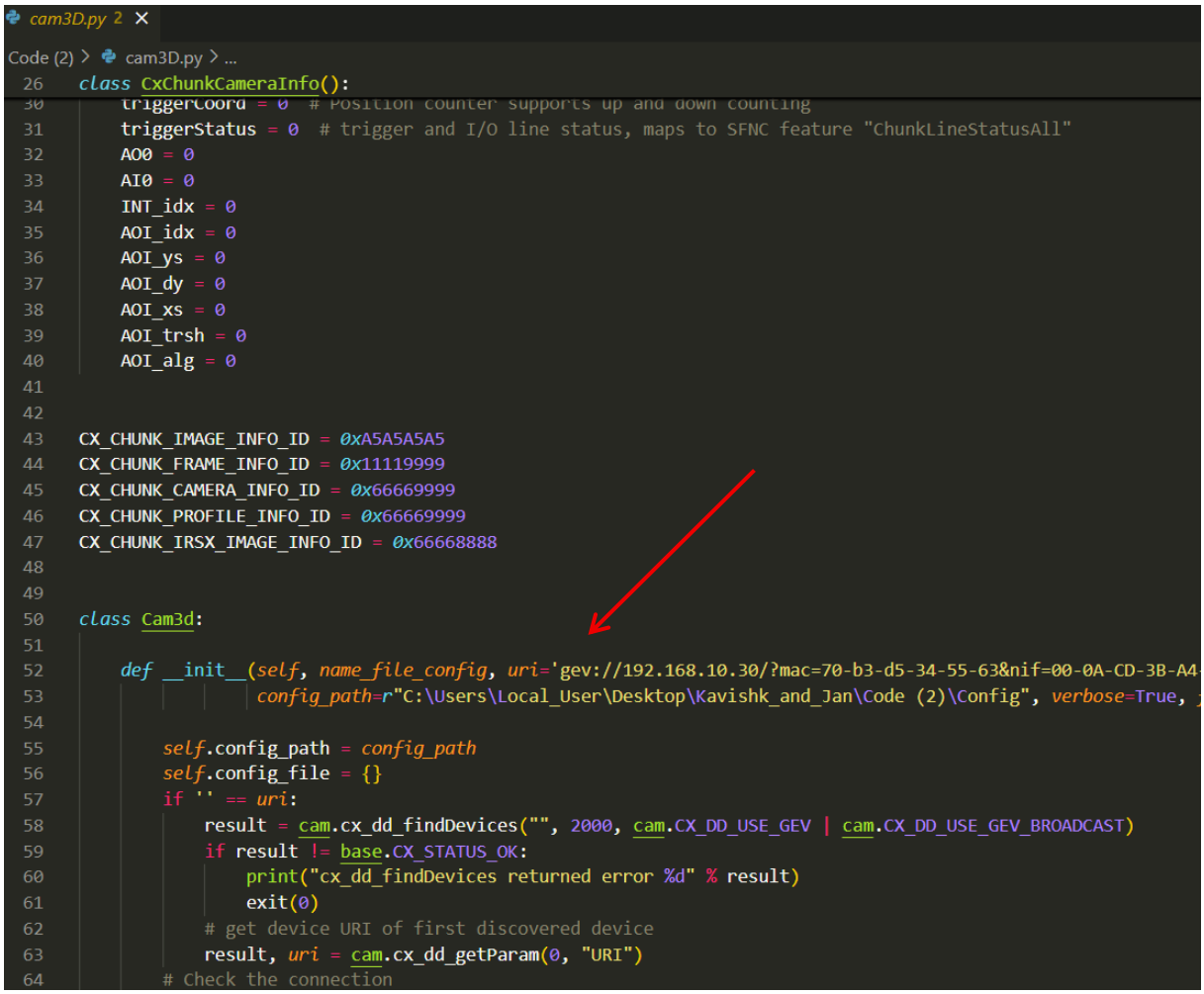
1. Navigate to Ethernet Settings → Change adapter options.
2. Right-click on the relevant port and select Properties.
3. In the pop-up window, highlight Internet Protocol Version 4 (TCP/IPv4) and click Properties.
4. Assign a unique static IP address (e.g., 192.168.10.171) and set the subnet mask to 255.255.255.0. Confirm and close the dialog.
5. Back in the [Port Name] Properties window, click Configure → Advanced, then:
6. Set Jumbo Packet, Receive Buffers, and Transmit Buffers to their maximum values.
7. Set Speed & Duplex to the highest full-duplex speed (typically 1.0 Gbps).
8. To confirm that the PLC and 3D camera are reachable, use the Command Prompt to ping the 3D camera (current address: 192.168.10.30).

C.2 Updating Python Camera Scripts

To update the Python environment for the new network adapter, follow these steps:

Open cxDiscover and copy the camera's URI (e.g., gev://192.168.10.30/?mac=70-b3-d5-34-55-63&nif=F0-2F-74-1E-42-A6).

Open the script cam3D.py and paste this URI into the __init__ method of the Cam3D class.



The RGB+NIR camera is configured and accessed via the Python script camRGB_NIR.py, utilizing the GenICam framework.

C.3 Required Software and Drivers

Use eBus Player for camera configuration, or download the custom software for JAI cameras under:

“Sweep+ Series” → “SW-4000T-10GE” → “eBUS SDK for JAI (64-bit)”.

Install the Harvester Python package, which interfaces with GenICam cameras.

Download the recommended Matrix Vision GenTL Producer from the official site (look for the latest version, e.g., mvGenTL_Acquire-x86_64-2.47.0.exe).

Install the 64-bit Common Vision Blox licensing software and the cx-Support Package to enable Python control over the 3D camera.

Important Note:

To prevent runtime errors with the RGB+NIR camera in the Harvester package, modify the file gentl.py at line 3689 as follows:

Before:

```
def revoke_buffer(self, buffer):
    ret = self._revoke_buffer(buffer)
    return BufferToken(raw_buffer=ret[0], context=ret[1])
```

After:

```
def revoke_buffer(self, buffer):
    pass
# ret = self._revoke_buffer(buffer)
```

For troubleshooting buffer or memory issues, refer to the Harvester package's online documentation.

C.4 Connecting and Controlling the PLC and Conveyor Belt

To interface with the PLC and conveyor belt from Python (tested on Python 3.9), use the Snap7 library:

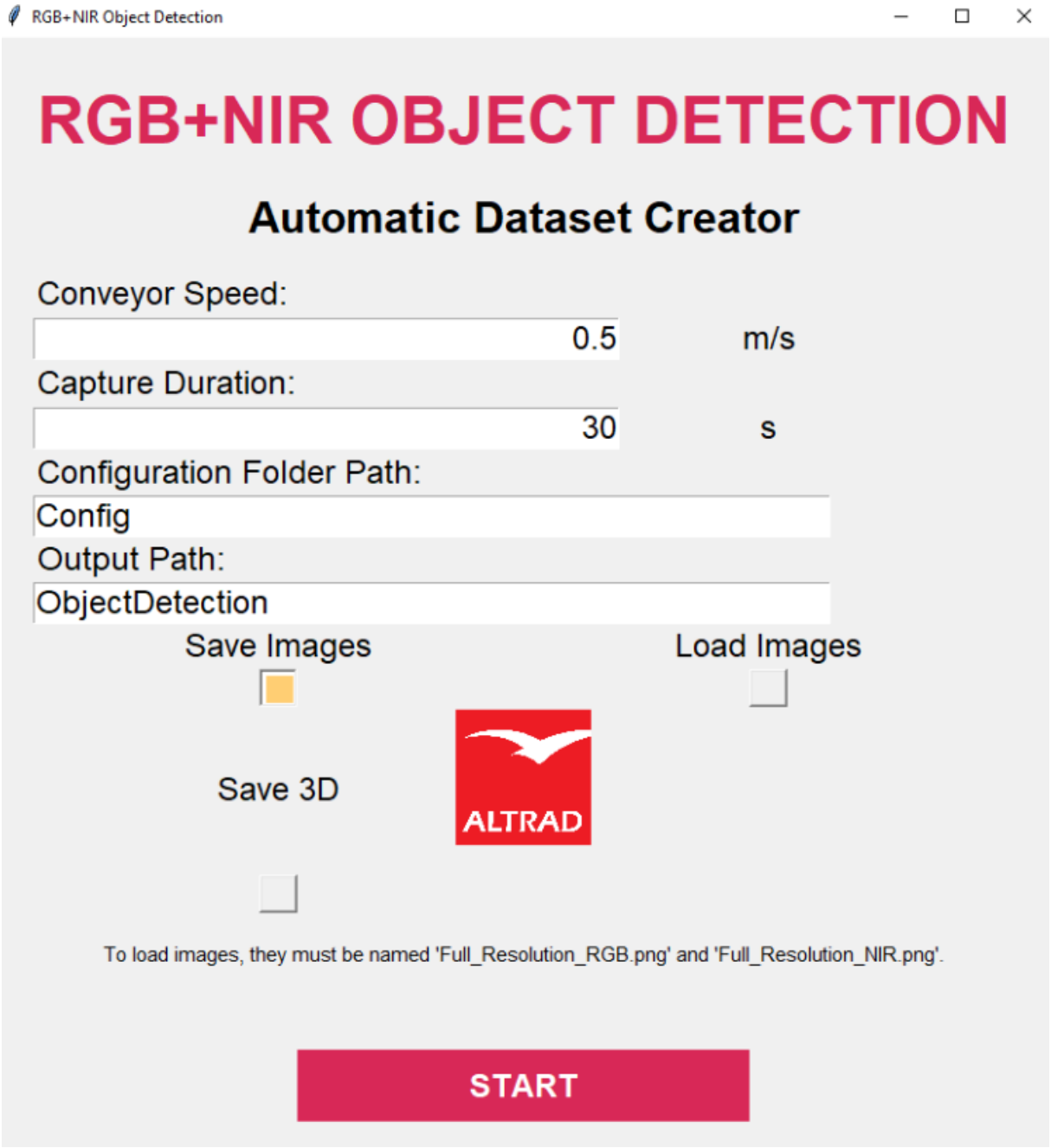
Install Snap7 via Conda or the base environment:

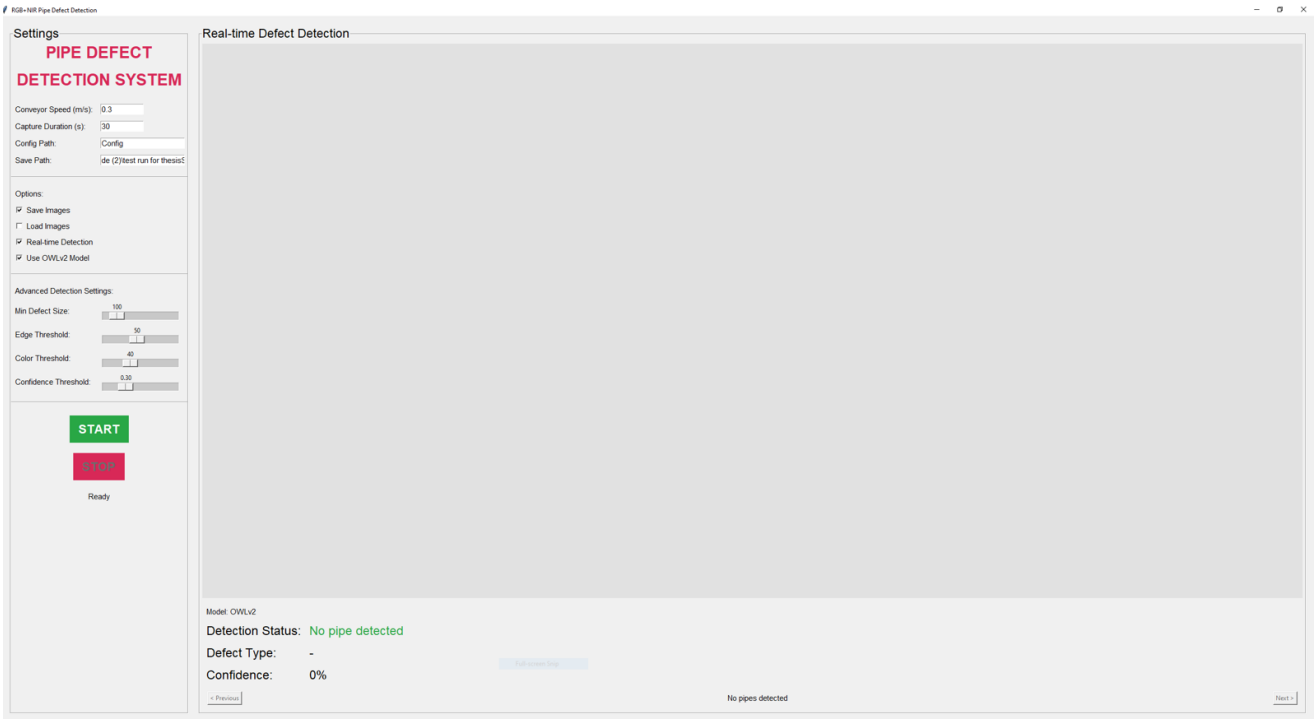
1. *pip install python-snap7*
2. Download the latest Snap7 package from the official Snap7 website.
3. Extract the folder and place the snap7.dll file into a directory included in the system's PATH.

Python interaction with the PLC and conveyor system is managed via the DbPlc class, defined in the script controlPLC.py.

To capture dataset using the RGB+NIR camera, run the Dataset_creator_GUI.py

You will have to set the conveyor speed manually, the config file path (if it is in the same folder as the code for the GUI, you don't need to change anything) and finally the path to save the captures. Ideally, it is suggested to choose a value between 0.3-0.8 m/s, depending on the size and length of the object. For longer objects, a lower value of 0.3 was selected to avoid image warping.





The same applies to the live defect detection system, which can be deployed by running the live_defect_detection_OWOD.py file, where it is pre-loaded with the OWOD model and the prompts for all 3 defects we observe for the pipe case (mortar, rust, bent). The user can easily replace this with say a deep learning model by importing the model as well as its weights.

To make sure all these scripts work, all the helper classes/files need to be present in the same folder as the code mentioned above. This is already ensured in the GitHub of the thesis under the main LCE lab folder.

FACULTY OF ENGINEERING TECHNOLOGY
W. de Croylaan 56
3001 HEVERLEE, Belgium
info.iiw@kuleuven.be
www.fet.kuleuven.be