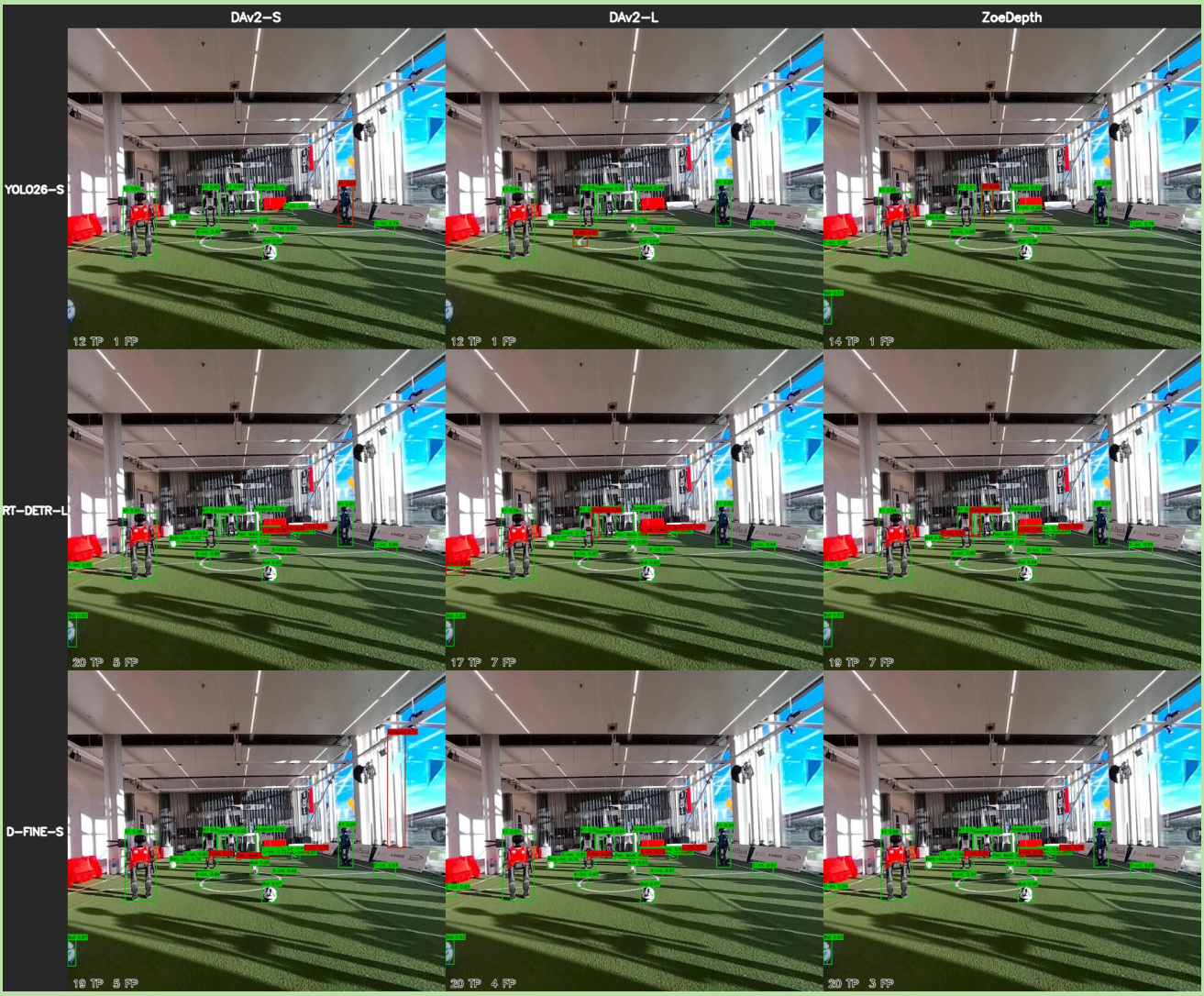
RGB-D Object Detection for RoboCup

The Impact of Monocular Depth Estimation on Object Detection Performance in the RoboCup HSL Environment: A Multi-Pipeline Evaluation



Irem Afacan

Layout: typeset by the author using LaTeX.
Cover illustration: Irem Afacan

RGB-D Object Detection for RoboCup

The Impact of Monocular Depth Estimation on Object Detection Performance in the RoboCup HSL Environment: A Multi-Pipeline Evaluation

Irem Afacan
15189163

Bachelor thesis
Credits: 18 EC

Bachelor *Kunstmatige Intelligentie*

University of Amsterdam
Faculty of Science
Science Park 900
1098 XH Amsterdam

Supervisor
Dr. A. Visser

Informatics Institute
Faculty of Science
University of Amsterdam
Science Park 900
1098 XH Amsterdam

Semester II, 2025-2026

Abstract

Reliable object detection is fundamental for autonomous robotic perception on platforms such as the K1 Booster competing in the RoboCup Humanoid Soccer League (HSL). While RGB-only detection methods are standard, they struggle with challenging conditions such as motion blur, overexposure, and distant objects, which are all prevalent in RoboCup match environments. Depth information, as an auxiliary modality, may mitigate these limitations by providing geometric structure that complements RGB, appearance-based features.

This thesis investigates whether augmenting RGB input with monocular depth estimation (MDE) improves object detection performance in the HSL context, with a particular focus on whether depth benefits the detection of objects at varying scales. Since the HSL Objects V1 dataset contains no ground-truth depth data, depth maps were generated using three MDE models: DepthAnythingV2-Small, DepthAnythingV2-Large, and ZoeDepth. These were combined with three object detectors (YOLO26-S, RT-DETR-L, and D-FINE-S) via early fusion, resulting in nine RGB-D pipeline combinations evaluated against RGB-only baselines.

The results show that the effect of adding depth varies substantially across detector architectures. D-FINE-S showed negligible differences between RGB-only and RGB-D pipelines, with ablation results confirming the model effectively disregarded the depth channel. YOLO26-S and RT-DETR-L show performance degradation compared to their RGB-only baselines, with average mAP50-95 drops of 0.202 and 0.107 respectively. Analysis of scale-based AP metrics reveals that depth consistently harms detection of small objects across pipelines, with the largest drops observed at AP_S across both YOLO26 and RT-DETR RGB-D pipelines. These findings suggest that MDE-generated depth maps do not provide reliable complementary information for RGB-D object detection in the RoboCup HSL context, likely due to limitations in early fusion and depth map quality.

Contents

Chapter 1

Introduction

1.1 RoboCup

This thesis will focus on RGB-D based object detection specifically in the context of the Humanoid Soccer League of the RoboCup. The RoboCup is an international initiative with the purpose of promoting robotics and AI research, organizing football games in which the teams consist of autonomous robots. The creators of the RoboCup have set the goal of fielding a team of robots capable of winning against the human soccer World Cup champions by 2050. These ambitious goals are currently being developed in a standardized framework: the fields must have a green carpet, white lines, and specific ball patterns. The structured environment of the RoboCup provides an ideal benchmark for object detection, as it allows variables to be isolated and performance to be consistently compared across different platforms.

Most teams participating in the Humanoid Soccer League, a sub-division of the RoboCup, utilize the Booster K1 and T1 as their main robot platform. WhIRLwind Amsterdam, the team taking part in the RoboCup on behalf of the Intelligent Robotics Lab of the University of Amsterdam, employs the Booster K1 as well (Honkoop et al., 2026), making the K1 the target deployment platform for the object detection pipeline developed in this thesis. The official RoboCup rules do not impose specific lighting conditions, making lighting variability a persistent challenge for RGB-only detection systems (RoboCup Humanoid Soccer League, 2026). Additionally, RoboCup scenes are characterized by clutter, such as multiple robots, goalposts, field markings, and a ball present simultaneously, creating occlusion challenges that RGB-only detectors may struggle with. These characteristics motivated the exploration of depth as a lighting-invariant auxiliary modality that may provide additional geometric context in such challenging scenes.

1.2 RGB-Depth Based Object Detection

The humanoid robots participating in the RoboCup often need to process a noisy football scene in real-time. In computer vision, object detection entails the task in which AI systems are meant to correctly localize and classify objects. Traditional object detection models were solely reliant on RGB images to achieve their objective. However, while unimodal input may suffice for AI systems operating in purely digital domains (e.g., content recommendation), it can fall short for embodied

agents that interact physically with their surroundings (e.g., robotic grasping, autonomous indoor navigation, or augmented reality). This need for richer, multisensory features therefore gave rise to multimodal-based (e.g., RGB-Depth, RGB-Thermal, etc.) object detection (Gao et al., 2019). In this thesis, the scope is narrowed to specifically focus on RGB-D based object detection.

Since an RGB image is a 2D projection of the 3D world, depth serves as an auxiliary modality that mitigates this spatial information loss by providing explicit 3D structural information. Furthermore, in contrast to RGB images, depth data is lighting and color invariant, as depth sensors (such as LiDAR, Structured Light, etc.) are often active sensors capable of working in darkness (Li et al., 2022).

While active sensing technologies such as LiDAR and structured light cameras can provide accurate depth measurements, they introduce additional hardware cost, weight, and power consumption that may be prohibitive in resource-constrained platforms such as the K1 Booster (Cadena et al., 2016). Monocular Depth Estimation (MDE) instead infers depth from a single RGB image using only passive visual sensing, making it an attractive alternative for platforms already equipped with a camera (Arampatzakis et al., 2024). Although the K1 employs stereo vision-based depth estimation onboard, the dataset used in this thesis consists of monocular RGB images, necessitating the use of monocular depth estimation as the depth source.

1.3 Research Question & Thesis Overview

This thesis investigates whether incorporating monocular depth estimation alongside the RGB modality improves object detection performance in the RoboCup HSL context. Target objects are defined as robot-relevant objects such as the ball, penalty marks, field line intersections (X/L/T), goalposts, and the K1 robot itself. A particular focus is placed on whether depth information differentially impacts detection across object scales, as small and distant objects present a persistent challenge for RGB-only detectors and depth may either aid or hinder their detection depending on depth map quality. This thesis therefore aims to answer the following research question:

> *Does augmenting RGB input with monocular depth estimation improve object detection accuracy in the RoboCup HSL environment, and does this improvement vary as a function of object scale?*

This is further decomposed into three sub-questions:

1. Does adding an estimated depth channel to RGB input improve overall object detection performance, as measured by mAP50-95, compared to RGB-only baselines?
2. Which combination of depth estimator and object detector yields the best detection performance, as measured by mAP50-95, on the HSL Objects V1 dataset?
3. Does the addition of depth differentially impact detection performance across object scales, as measured by AP_S, AP_M, and AP_L?

To address these questions, this thesis is structured as follows. Chapter 2 establishes the theoretical background, covering depth estimation, object detection, and multimodal fusion. Chapter 3 describes the methodology, including pipeline design, model selection, and training strategy. Chapter 4 presents and discusses the experimental results, and Chapter 5 concludes with a summary of findings and directions for future work.

Chapter 2

Theoretical Background

This chapter provides the theoretical foundation for RGB-D based object detection, covering depth estimation and object detection as individual research fields before examining the specific models used in the pipeline: DepthAnythingV2 (Yang, Kang, Huang, Zhao, et al., 2024) and ZoeDepth (Bhat et al., 2023) as depth estimators, an YOLO26 (Sapkota et al., 2026), RT-DETR (Zhao et al., 2024), and D-FINE (Peng et al., 2025) as object detectors. Together, these foundations motivate the RGB-D pipeline evaluated in this thesis.

2.1 RGB-D Based Object Detection

2.1.1 RGB-D Data

RGB-D data can essentially be viewed in terms of the depth map as an additional single-channel image alongside the traditional three-channel color data (Lai et al., 2011). Each pixel of the depth map represents the distance from the camera to the corresponding surface point. While RGB data provides robust representations for the visual appearance of objects and textures, depth information is crucial for capturing geometric structure and the spatial layout of a scene (Ferreri et al., 2021). Because the depth modality compensates in what RGB images lack, namely lighting and texture invariance (Shotton et al., 2011), the combination of RGB and depth data significantly boosts the quality of results in computer vision tasks such as object recognition and detection (Lai et al., 2011). More specifically, depth provides surface normals, object boundaries in 3D, and relative distances between objects, allowing explicit reasoning about spatial structure instead of deriving this from appearance alone. A concrete example is, for instance, a dark object against a dark background, which is nearly invisible to RGB, but fully present in a depth map.

2.1.2 Depth Representations

There are multiple methods for representing depth. A raw depth map consists of the noisy depth data obtained from active sensors, which often contain holes where the sensor was unable to measure the distance. The utilization of this kind of depth map therefore requires additional preprocessing. Gupta et al. (2014) designed a more informative representation of depth by augmenting the single-channel depth map to a three-channel image comprising Horizontal disparity, Height above ground,

and Angle with gravity. This Horizontal-Height-Angle (HHA) encoding allows Convolutional Neural Networks (CNN) pre-trained on three-channel RGB images to process HHA geocentric features without requiring architectural modifications. However, HHA-encoded depth maps can form a computational bottleneck in downstream tasks (Rahman et al., 2026). Alternatively, depth maps generated by monocular depth estimation (MDE) models form an attractive solution, because these models produce dense depth maps, as the estimation process inherently predicts a value for every pixel. Figure 2.1 shows a side-by-side comparison of sparse and depth maps. Consequently, MDE generated depth maps do not require further preprocessing, and are computationally lighter than HHA-embeddings, making them an appealing depth representation option. A more extensive review of MDE models and principles can be found in Section 2.2.2.

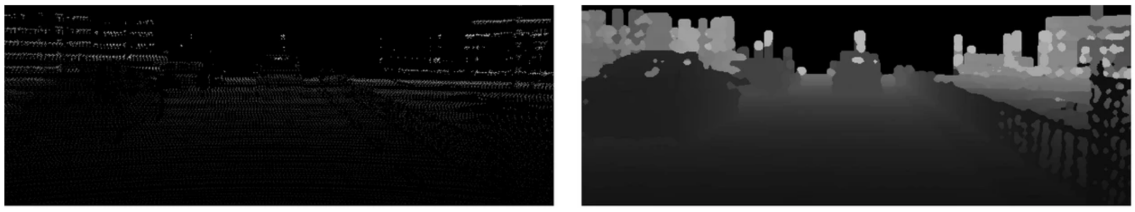


Figure 2.1: Comparison of a sparse depth map obtained from LiDAR (left) and a dense depth map generated by monocular depth estimation (right). Taken from Z. Chen et al. (2024).

2.1.3 RGB-D Fusion Schemes in Object Detection

A suitable fusion scheme is fundamental to meaningfully combine extracted features from two or more modalities, such as RGB and depth, for an object detection model (Gao et al., 2019). Most multimodal fusion schemes can broadly be described as an early, late or multi-scale fusion strategy (A. Chen et al., 2024), as visualized in Figure 2.2.

Early Fusion

An early fusion strategy entails that the modalities are fused at the earliest possible moment of the object detection pipeline. This can be implemented by directly concatenating RGB and depth channels when feeding them into the model, creating a four-channel RGB-D input. Early fusion can furthermore be implemented by first extracting low-level features of both modalities, before concatenation of RGB and depth channels. Early fusion strategies usually require minimal architectural modification of the object detection network, and are computationally lightweight compared to dual-network approaches (Mahjourian & Nguyen, 2025). However, noisy depth maps can corrupt the input representations at pixel level, before any feature extraction has taken place, potentially disrupting the model's ability to learn useful features.

Late Fusion

In contrast to early fusion, a late fusion scheme combines the two modalities at a later stage of the pipeline. A late fusion-based object detection model first processes both modalities separately using two parallel network streams. Features are concatenated after high-level feature extraction and subsequently fed into the object detector that generates the predictions. Alternatively, the

two streams can produce independent predictions for each modality, which are then combined into a final prediction. Late feature fusion therefore allows each modality to develop independent feature representations without interference from the other modality, while sacrificing the potential benefits of cross-modal interactions and requiring a more computationally demanding dual-stream architecture.

Multi-scale Fusion

Unlike late fusion, which processes modalities independently, multi-scale fusion exploits correlations between RGB and depth across multiple stages of the network. One approach uses cross-modal interaction modules, such as attention mechanisms (Orfaig et al., 2026), that allow RGB and depth features to mutually inform each other before being fed into the main feature extraction network. Alternatively, RGB and depth features are extracted at multiple scales simultaneously (Zhou et al., 2021), with fusion occurring at each corresponding scale before being combined in a decoder network. While multi-scale fusion enables richer cross-modal feature learning than early or late fusion, it introduces significantly greater architectural complexity.

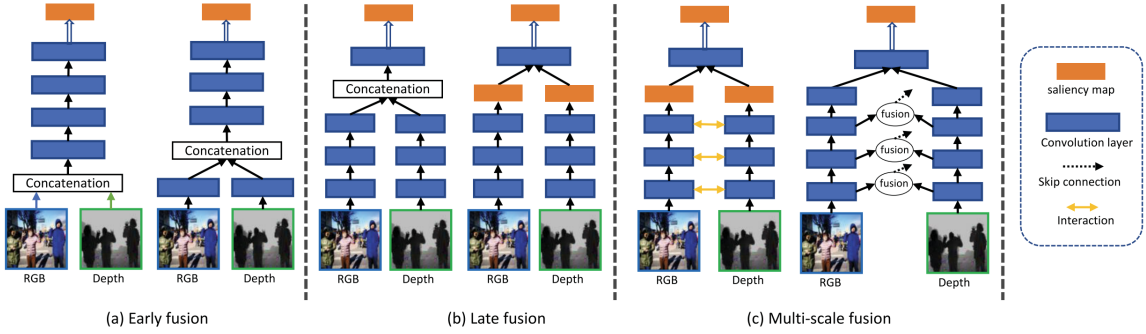


Figure 2.2: Overview of RGB-D fusion strategies. While depicted for salient object detection, the fusion principles apply analogously to object detection by replacing the saliency map output with bounding box predictions. Taken from Zhou et al. (2021).

2.1.4 Prior Work on RGB-D Object Detection

While some studies demonstrate gains over an RGB-only baseline model, the majority of RGB-D based object detection literature focuses on comparing fusion architectures against each other (Zhou et al., 2021), leaving the fundamental question of whether depth consistently helps underexplored. The studies that do assess their RGB-D models against an RGB baseline show that the RGB-D models outperform the RGB-only models. Multi-scale feature fusion is the dominant integration mechanism in the literature, though early and late fusion approaches have also been explored. For instance, Mahjourian and Nguyen (2025) utilize an early fusion approach using Faster R-CNN as the object detector, training three model variants: RGB-only, depth-only, and RGB-D. These pipelines were trained from scratch without pretrained initialization, using a manufacturing dataset with depth maps obtained from a 3D point cloud sensor. The multimodal RGB-D model substantially outperformed both unimodal baselines, achieving a 13% mAP50 improvement over the RGB-only

baseline and an mAP50 score of 0.480, compared to 0.269 for the depth-only variant.

Furthermore, Ophoff et al. (2019) demonstrate significantly higher mAP50 scores against their RGB baseline, utilizing dual RGB and depth network streams to concatenate and feed into the YOLOv2 object detector. The authors trained three types of models: an RGB-only baseline, a depth-only network, and 28 fusion variants. For the RGB subnetwork, standard ImageNet pretrained weights from the DarkNet-19 backbone were used, while for the depth subnetwork the authors used weights from the previously trained depth-only network, as these weights already encode meaningful depth features.

RGBX-DiffusionDet (Orfaig et al., 2026) employs a multi-scale fusion approach with cross-modal attention mechanisms, trained on the KITTI autonomous driving dataset using LiDAR-obtained depth maps. Similarly to Mahjourian and Nguyen (2025) and Ophoff et al. (2019), it demonstrates gains over an RGB-only baseline, improving mAP50-95 from 0.670 to 0.692. Orfaig et al. (2026) identify three core limitations of conventional RGB-D fusion: scale misalignment across feature pyramids, semantic inconsistency between modalities, and unequal reliability of the two input signals. To address these limitations, the authors propose the DCR-CBAM module, which is a channel-wise attention module that dynamically weights modality-specific features. Furthermore, the DMLAB module aggregates features across all pyramid levels rather than selecting a single fixed level, preserving spatial context even when initial bounding boxes are imprecise.

2.2 Depth Estimation

Depth estimation is the task of determining the distances of objects to the camera in a given image. While depth estimation can be achieved by incorporating data streams from active sensors into the robot's working cycle, this spatial awareness can also be accomplished by purely relying on computer vision based approaches. These methods can be categorized as stereo vision-based and monocular depth estimation.

2.2.1 Stereo Vision-Based Depth Estimation

Stereo vision-based approaches utilize two camera's to obtain two different perspectives of the same scene, drawing inspiration from the way in which humans infer depth binocularly. Because the eyes, or lenses, are horizontally separated, they perceive a scene from two distinct vantage points. In order to quantify the resulting displacement, known as binocular disparity, corresponding points in images need to be found. This is also called the Correspondence Problem. Once features in the left and right frames are matched, depth can be inferred by applying triangulation. This geometric process utilizes the known baseline, the distance between the two cameras, and the focal length to calculate the absolute distance to the object (Meng et al., 2021). A schematic representation of this principle is illustrated in Figure 2.3.

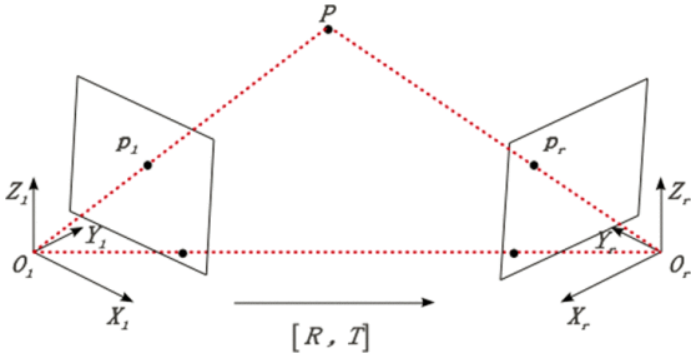


Figure 2.3: Schematic depiction of 3D point triangulation. Taken from Li et al. (2011).

Calculating the binocular disparity for every pixel in an image makes real-time depth estimation, which is essential for the autonomous behavior of robots, computationally prohibitive. Modern stereo vision methods for depth estimation initially relied on machine learning techniques, which streamlined specific building blocks of the stereo pipeline, before shifting to deep neural networks (Poggi et al., 2022). As deep learning advanced (LeCun et al., 2015), such machine learning techniques became obsolete. Rather than optimizing individual steps in the stereo matching pipeline, deep learning allowed for optimizing the entire process end-to-end, effectively automating the traditional pipeline of matching and triangulation. As the dataset does not contain stereo image pairs, stereo vision-based depth estimation is not applicable in this thesis.

2.2.2 Monocular Depth Estimation

In contrast to stereo vision-based methods, Monocular Depth Estimation (MDE) is the task of inferring depth solely from an RGB image (Arampatzakis et al., 2024). This is a particularly complex inverse problem, as the projection of a 3D scene onto a 2D image results in inevitable loss of spatial information, leading to scale ambiguity. An example of such ambiguity in images can be illustrated by comparing a small object closer to the lens and a larger object placed at a greater distance from the camera. When captured as a 2D image, both objects will fallaciously appear to be equal in size. Consequently, MDE models must infer depth from contextual cues as opposed to geometric measurement.

Current state-of-the-art MDE models are predominantly based on foundation models (Xu et al., 2026), which demonstrate emergent zero-shot generalization capabilities across diverse contexts due to the vast size of their training data. While MDE foundation models can be classified into several paradigms, affine-invariant and metric depth models will represent the two most relevant distinctions for this thesis. Models that produce affine-invariant depth generalize well across diverse scenes, but produce depth that cannot be interpreted in physical units, representing the relative depth between objects in a scene (Ranftl et al., 2022). Metric depth estimators, on the other hand, produce depth maps with comparable scale across images, but require camera intrinsics, either as input or through estimation (Piccinelli et al., 2026). In this thesis, models of both paradigms are evaluated to specifically assess whether relative or metric depth yields higher detection performance for the RGB-D pipelines.

DepthAnythingV2

DepthAnythingV2 (DAv2) (Yang, Kang, Huang, Zhao, et al., 2024) is an affine-invariant MDE model, consisting of a DINOv2 Vision Transformer (ViT) backbone followed by a Dense Prediction Transformer (DPT) head. DAv2's training methodology is built upon a teacher-student knowledge distillation approach, as was already established in its predecessor, DAv1 (Yang, Kang, Huang, Xu, et al., 2024). Unlike its predecessor DAv1, which relied on real labeled images, DAv2's teacher was trained entirely on synthetic data. Synthetic training data, generated using Stable Diffusion, was preferred over real sensor data, as the latter often contained noisy labels that propagate into the model's predictions. Synthetic labels, as opposed to real labels, consisted of depth maps with fine detail, even in challenging cases such as transparent objects or reflective surfaces, resulting in stronger generalization across diverse scenes. The teacher in turn generates pseudo-labels for the real, unlabeled images which are used to train the student models. The full student-teacher distillation process is illustrated in Figure 2.4.

The synthetic data used in DAv2 to train its teacher model contains approximately 600K images, while the pseudo-labeled real images used to train the student models consist of 62M images, with both datasets containing indoor and outdoor scenes. DAv2 exhibits robust depth estimation abilities as a result of using synthetic training data, which is free from depth sensor noise. This makes it particularly suitable for challenging scenarios such as textureless surfaces like the green carpet representing the football pitch in RoboCup scenes. DAv2, however, produces affine-invariant depth maps, meaning depth values lack real-world scale and cannot be directly interpreted as physical distances. In this thesis, both the small and large variants of DepthAnythingV2, DAv2-S and DAv2-L respectively, are evaluated.

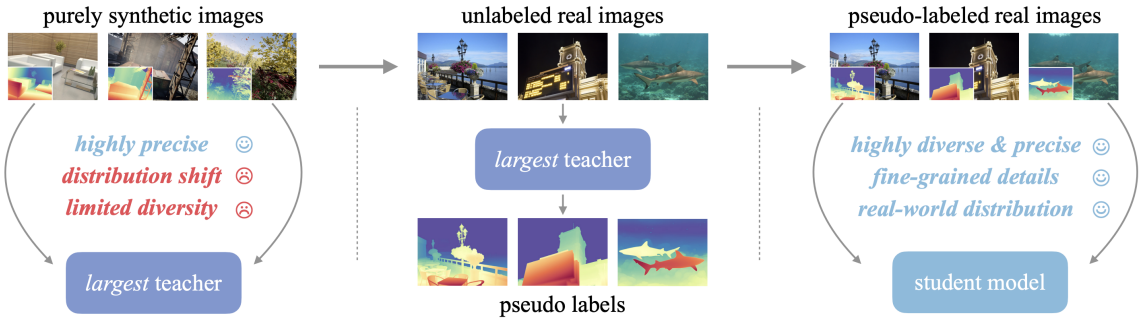


Figure 2.4: Schematic representation of DAv2's teacher-student knowledge distillation process. Taken from Yang, Kang, Huang, Zhao, et al. (2024).

ZoeDepth

In contrast to DepthAnythingV2, ZoeDepth (Bhat et al., 2023) is a metric depth estimator, using an encoder-decoder architecture with a BEiT-L (Bidirectional Encoder from Image Transformers) backbone and DPT head. Instead of directly estimating metric depth, ZoeDepth uses a two-fold approach: pre-training its backbone for affine-invariant depth estimation and fine-tuning the prediction heads for metric depth estimation. Separate prediction heads are used for indoor and outdoor

scenes respectively. Multiple datasets, containing both indoor and outdoor scenes, were used to train the prediction heads. While indoor scenes are limited to a maximum depth of 10 meters, outdoor images can contain scenes of a maximum depth of 80 meters, making metric depth estimation inherently more challenging than relative depth estimation. Pre-training for relative depth estimation was therefore implemented to mitigate the challenges of training across datasets with varying depth scales.

ZoeDepth produces metric depth, providing consistent depth scale across different images, unlike affine-invariant MDE models. However, as ZoeDepth is trained on real data, it may struggle with textureless surfaces and difficult lighting conditions, such as reflections or overexposure. Therefore, ZoeDepth may generate inaccurate depth maps when faced with the textureless, green carpet of the football pitch, or when the camera is exposed to direct overhead lighting. In this thesis, ZoeDepth is used as the metric depth estimator in the RGB-D object detection pipeline.

2.3 Object Detection

The task of object detection refers to locating and identifying objects in a given scene, usually by placing bounding boxes around the recognized objects. Different models of this extensively studied field of research can classically be distinguished by whether the model is based on a two- or one-stage architecture. A two-stage object detection model generates predictions by first locating relevant regions and subsequently classifying the proposed areas. Pioneering two-stage models include R-CNN (Girshick et al., 2014) and its successor Faster R-CNN (Ren et al., 2015), which significantly improved computational efficiency through the introduction of a region proposal network. A single-stage architecture, such as the YOLO (You Only Look Once) models, does both steps simultaneously, exchanging accuracy for higher inference speed. However, since the emergence of transformer-based models, the classic stage-based distinction has become somewhat outdated, as the underlying architecture has become the more meaningful distinguishing criterion (Arkin et al., 2023). The following subsections will therefore be outlined according to this distinction.

2.3.1 CNN-based Detectors

Due to its single-pass design, the YOLO model family has stood out in terms of speed and accuracy. While alternative CNN-based detectors such as the Single Shot Detector (SSD) (Liu et al., 2016) have been widely used, YOLO has emerged as the dominant family in terms of adoption and active development, and will therefore be the focus of this section. The first model of the series, YOLOv1 (Redmon et al., 2016), established the single-pass paradigm, while later versions, primarily YOLOv5 (Jocher, 2020) and YOLOv8 (Jocher et al., 2023), brought the YOLO family to mainstream adoption, substantially improving the speed-accuracy balance. YOLOv8 improved upon YOLOv5 by adopting an anchor-free detection head, eliminating the need for anchor boxes (predefined bounding boxes distributed across the image grid), reducing both the number of candidate predictions per image and the complexity of the detection pipeline. YOLO11 (Jocher & Qiu, 2024) in turn improved upon YOLOv8 by reducing its parameter count, distinguishing itself as a lightweight architecture with a comparable speed-accuracy balance.

YOLO26

Unlike its predecessor YOLO11, YOLO26 (Sapkota et al., 2026) removes the Distribution Focal Loss (DFL) from the detection head, reducing detection head complexity and simplifying regression. This specialized loss function guides the model in drawing precise and tight bounding boxes around the objects by predicting a probability distribution over possible edge locations. Instead, a simpler regression head is used, resulting in weaker localization quality. However, this is compensated by the adoption of Progressive Loss Balancing (ProgLoss) and Small-Target-Aware Label Assignment (STAL) during model training. YOLO26 uses two detection heads: a one-to-many head predicting multiple bounding boxes per object and a one-to-one head predicting a single box per object. During training, ProgLoss dynamically shifts the learning signal weight from the one-to-many head to the one-to-one head, encouraging the model to confidently produce a single prediction per object. As each object receives exactly one prediction, YOLO26 eliminates the need for Non-Maximum Suppression (NMS) in post-processing, reducing inference latency. STAL, introduced alongside ProgLoss, forces the model to take small objects into account during training by enforcing a minimum of four anchor assignments, which is the number of anchor boxes designated to predict a given object, for objects smaller than 8 pixels. The full YOLO26 training pipeline is illustrated in Figure 2.5.

In this thesis, the small variant YOLO26-S is used, offering a compact CNN-based architecture with a reduced parameter count while maintaining competitive detection accuracy. YOLO26 was specifically chosen to represent the CNN-based object detector in the pipeline, and as the most recent and advanced model of the YOLO series (Sapkota et al., 2026), it serves as a strong architecture for CNN-based detection. Its STAL module, is particularly relevant given the size-based AP evaluation in this thesis, where objects at greater distances appear smaller in the image. However, it may struggle with occluded objects, or objects in cluttered scenes, due to the limited global context of CNN-based object detectors.

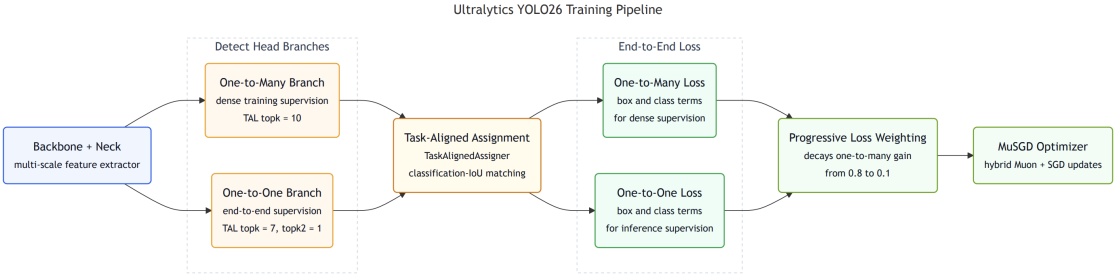


Figure 2.5: Schematic overview of the YOLO26 training pipeline, highlighting the dual detection heads, ProgLoss, and MuSGD Optimizer. Taken from Sapkota et al. (2026).

2.3.2 Transformer-based Detectors

Transformers, as introduced in the seminal paper "Attention Is All You Need" from Vaswani et al. (2017), represent a pivotal departure from CNN-based architectures. Whereas CNNs excel in extracting features through local convolutional operations, transformers use self-attention to model relationships between all image patches simultaneously, capturing global patterns that CNNs can

only approximate through increasing network depth. This global reasoning is particularly powerful for object detection models, where understanding an object often requires scene-level context.

The Detection Transformer (DETR) (Carion et al., 2020) was the first to apply transformers to end-to-end object detection, framing it as a set prediction problem and eliminating hand-designed components such as anchor boxes and NMS. While this significantly simplified the detection pipeline, DETR suffered from slow convergence, motivating subsequent works built upon its framework.

RT-DETR

Real Time-DETR (RT-DETR) (Zhao et al., 2024) maintains the NMS-free design of DETR and aims to improve speed by tackling DETR's main computational bottleneck. DETR experienced slow inference, as it applied self-attention across all image tokens at every encoder layer. RT-DETR addresses this by using a hybrid encoder, separating intra-scale interactions and cross-scale fusion, thereby reducing the number of image tokens the self-attention operates on. Specifically, the Attention-based Intra-scale Feature Interaction (AIFI) module is responsible for processing intra-scale interactions, while the CNN-based Cross-scale Feature Fusion (CCFF) module handles the cross-scale fusion. Additionally, RT-DETR's backbone can be replaced with a lighter alternative to further reduce inference latency. The speed improvement of RT-DETR does not happen at the expense of accuracy, as RT-DETR outperforms the contemporary YOLO models (YOLOv5 and YOLOv8) in both mAP and latency.

In this thesis, the large variant RT-DETR-L is used, representing the hybrid encoder transformer-based detector in the architectural comparison, contrasting with D-FINE-S's distribution-based regression approach, as detailed in Section 2.3.2. With 33M parameters (Zhao et al., 2024), RT-DETR-L is expected to demonstrate strong class discrimination. However, RT-DETR-L predicts fixed bounding box coordinates rather than modeling localization uncertainty, potentially resulting in less precise bounding box localization compared to distribution-based approaches such as D-FINE-Ss.

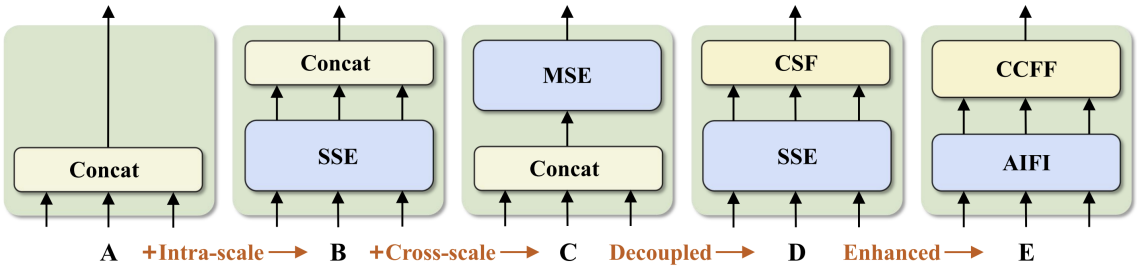


Figure 2.6: Evolution of the RT-DETR encoder design, where SSE and MSE denote single- and multi-scale transformer encoders respectively, culminating in the hybrid encoder (E) which decouples intra-scale interaction (AIFI) and cross-scale fusion (CCFF). Taken from Zhao et al. (2024).

D-FINE

D-FINE (Peng et al., 2025) is a DETR-based real-time object detection model that builds upon RT-DETR, surpassing contemporary detectors in both inference speed and accuracy. A key component in D-FINE's architecture is its Fine-grained Distribution Refinement (FDR) approach, which models probability distributions instead of predicting fixed box coordinates. Unlike fixed-coordinate regression approaches used in models such as RT-DETR, FDR models probability distributions over possible edge locations, explicitly capturing localization uncertainty. D-FINE furthermore introduces Global Optimal Localization Self-Distillation (GO-LSD). GO-LSD transfers localization knowledge from deeper decoder layers, which produce progressively refined distributions, to shallower layers, guiding early layers to learn more accurate bounding boxes and substantially accelerating convergence. Figure 2.7 illustrates the student-teacher distillation process.

In this thesis, the small variant D-FINE-S is used, representing the distribution-based regression transformer-based detector alongside RT-DETR-L's hybrid encoder approach. The FDR approach to bounding box prediction is particularly relevant for small or distinctively shaped objects in RoboCup scenes, such as the ball, where precise localization directly impacts detection accuracy. However, as the small variant, D-FINE-S has fewer parameters than larger variants, potentially limiting its capacity in scenes with many objects or fine-grained class distinctions.

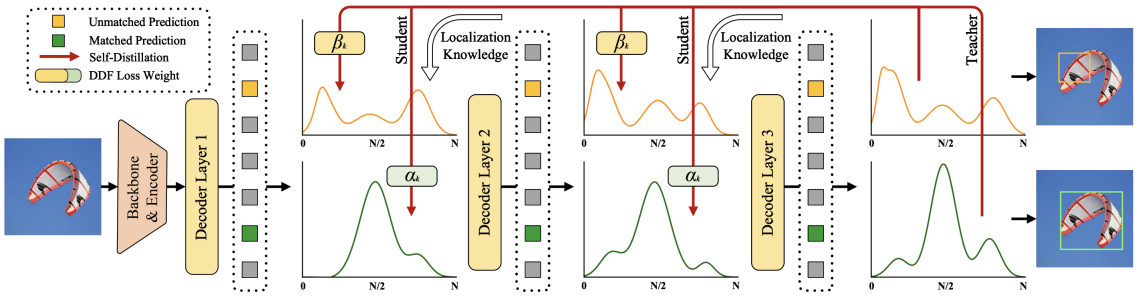


Figure 2.7: The GO-LSD mechanism distilling localization knowledge from deeper decoder layers (teacher) to shallower layers (student) using refined probability distributions. Taken from Peng et al. (2025).

2.4 Weight Initialization Approaches

In transfer learning scenarios where a pretrained model is extended with additional input channels, the newly introduced channels cannot be initialized from pre-trained weights. In these settings, the choice of initialization strategy for these weights is therefore paramount for model performance, as poorly initialized weights can lead to vanishing or exploding gradients during training (Narkhede et al., 2022).

2.4.1 Xavier Initialization

The Xavier weight initialization technique (Glorot & Bengio, 2010) was derived assuming approximately linear activations around zeros. Xavier initialization accounts for both the number of input

and output units to preserve variance in both the forward and backward pass and thereby preventing vanishing or exploding gradients. Neural networks utilizing the sigmoid or tanh activation function, both being symmetric functions with linear behavior around 0, are therefore suitable for this type of initialization. When drawing from a uniform distribution, the weights are assigned as follows:

$$W \sim U\left[-\sqrt{\frac{6}{n_j + n_{j+1}}}, \sqrt{\frac{6}{n_j + n_{j+1}}}\right]$$

Where W represents the weights, while n_j and n_{j+1} represent respectively the number of input and output paths towards the neuron.

2.4.2 He Initialization

Despite Xavier initialization's improvements, sigmoid and tanh remained susceptible to vanishing gradients in very deep networks, motivating the adoption of the Rectified Linear Unit (ReLU) (Agarap, 2026) activation function. The ReLU function zeroes all negative values, keeping only the positive values of the distribution, which substantially speeds computations up. However, this breaks the symmetry assumption which Xavier weight initialization relies on. He initialization (He et al., 2015) addresses this directly by scaling the weight variance by a factor of 2 to compensate for the half of the distribution zeroed out by ReLU, using only fan-in rather than both fan-in and fan-out. Weights assigned from He initialization can both be drawn from a Normal distribution and a uniform distribution. When drawing from a Normal distribution, weights are assigned as follows:

$$W \sim \mathcal{N}\left(0, \sqrt{\frac{2}{n_j}}\right)$$

Alternatively, when drawing from a uniform distribution, weights are assigned using:

$$W \sim U\left[-\sqrt{\frac{6}{n_j}}, \sqrt{\frac{6}{n_j}}\right]$$

2.5 Summary

This chapter outlined several concepts in the broader field of RGB-D based object detection, including various fusion schemes, with subsequent sections focusing on depth estimation, object detection, and weight initialization techniques. Of the three discussed fusion strategies, early fusion was adopted as a method of combining RGB and depth modalities. Additionally, two depth estimation methods were highlighted: stereo vision-based and monocular depth estimation. As the dataset used in this thesis does not contain stereo image pairs, stereo vision-based depth estimation is not explored further in this thesis. Instead, the three MDE models: DepthAnythingV2-Small, DepthAnythingV2-Large, and ZoeDepth are used as depth estimators in the implemented RGB-D pipelines. Furthermore, YOLO26-S, RT-DETR-L, and D-FINE-S were selected as the three object detectors employed in the RGB-D pipelines. Lastly, of the two discussed approaches, Xavier and He initialization, the latter is implemented for the depth channel weights. The early fusion scheme, six selected models, and He initialization together form the foundation of the pipeline described in Chapter 3.

Chapter 3

Method

This chapter describes the methodology of the RGB-D object detection pipeline evaluated in this thesis. The pipeline combines three monocular depth estimators, DepthAnythingV2-Small, DepthAnythingV2-Large, and ZoeDepth, with three object detectors, YOLO26-S, RT-DETR-L, and D-FINE-S, via early fusion, resulting in nine RGB-D pipeline combinations evaluated against three RGB-only baselines.

3.1 RGB-D Pipeline Setup

The full pipeline consists of two sequential phases: depth map generation via monocular depth estimation (MDE), followed by early fusion of the RGB image and depth map into a four-channel input. The object detection model then processes this four-channel RGB-D input, generating green bounding boxes for detected objects. The pipeline is visualized in Figure 3.1.

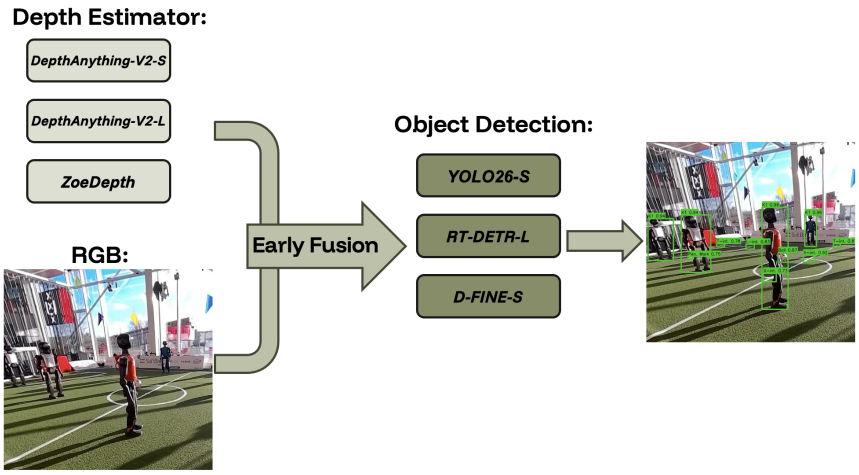
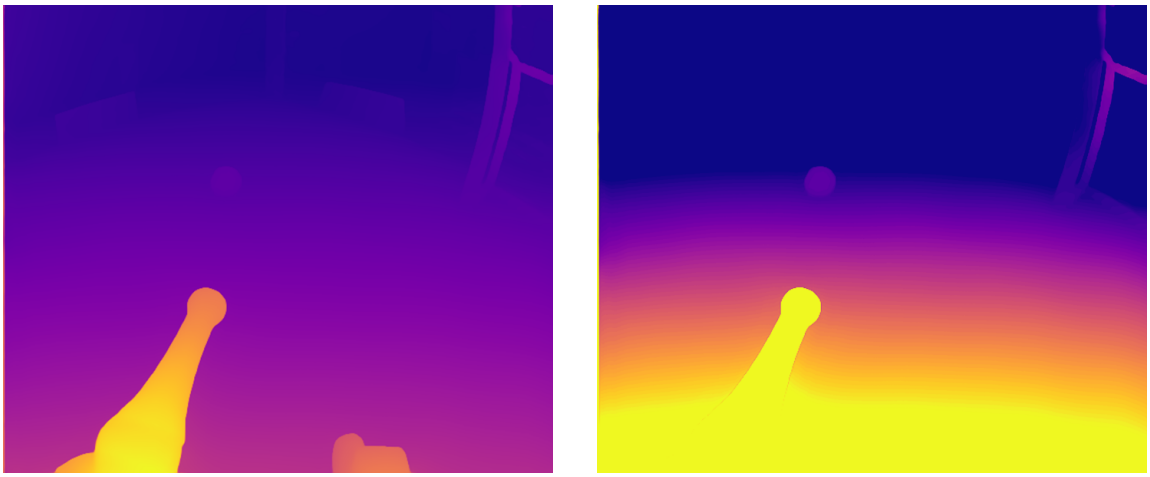


Figure 3.1: Schematic visualization of the implemented RGB-D pipeline.

3.1.1 Input & Preprocessing

Since all three object detectors expect 640×640 pixel input, images and their corresponding depth maps are resized to this resolution before entering the pipeline. Data augmentation is additionally applied during training to improve detection robustness, with common geometric transformations such as horizontal flip, rotation, and image translation randomly applied per sample. Since the aforementioned techniques are all geometric transformations, applying the same operation to the corresponding depth map is not problematic. However, color space augmentations, such as hue, saturation, and brightness adjustment, do pose a problem when applied to the matching depth map, because the pixel value of the depth image encodes spatial information which color augmentations could interfere with. An example of the effect of color jitter on a depth map is visualized in Figure 3.2. Therefore, color augmentations were specifically only applied to RGB images, and not to the corresponding depth maps.



(a) Original depth map.

(b) Depth map with color jitter augmentations.

Figure 3.2: The effect of hypothetical color jitter augmentation on a DAv2-S depth map. Color jitter was applied exclusively to the RGB channels during training and deliberately excluded from the depth channel, as it distorts depth information.

3.1.2 Early Fusion

An early fusion approach was adopted as it requires minimal architectural modification while allowing the backbone to learn joint RGB-D representations from the earliest layers. Concretely, the one-channel depth tensor was concatenated with the three-channel RGB tensor, producing a four-channel RGB-D tensor for the downstream object detector to process. The depth maps are normalized prior to the object detection stage to bring depth values into a compatible range with the normalized RGB channels. This ensures that neither modality dominates the other during training. Depth maps are first scaled to [0, 1] via min-max normalization, after which per-dataset mean and standard deviation are computed for standardization.

3.1.3 Output

The final output of the RGB-D based object detection pipeline is an RGB image with bounding boxes around detected object instances, whereby the depth channel is discarded from the output. Each bounding box is paired with a class label and a confidence score. An example of model predictions is visualized in Figure 3.3.

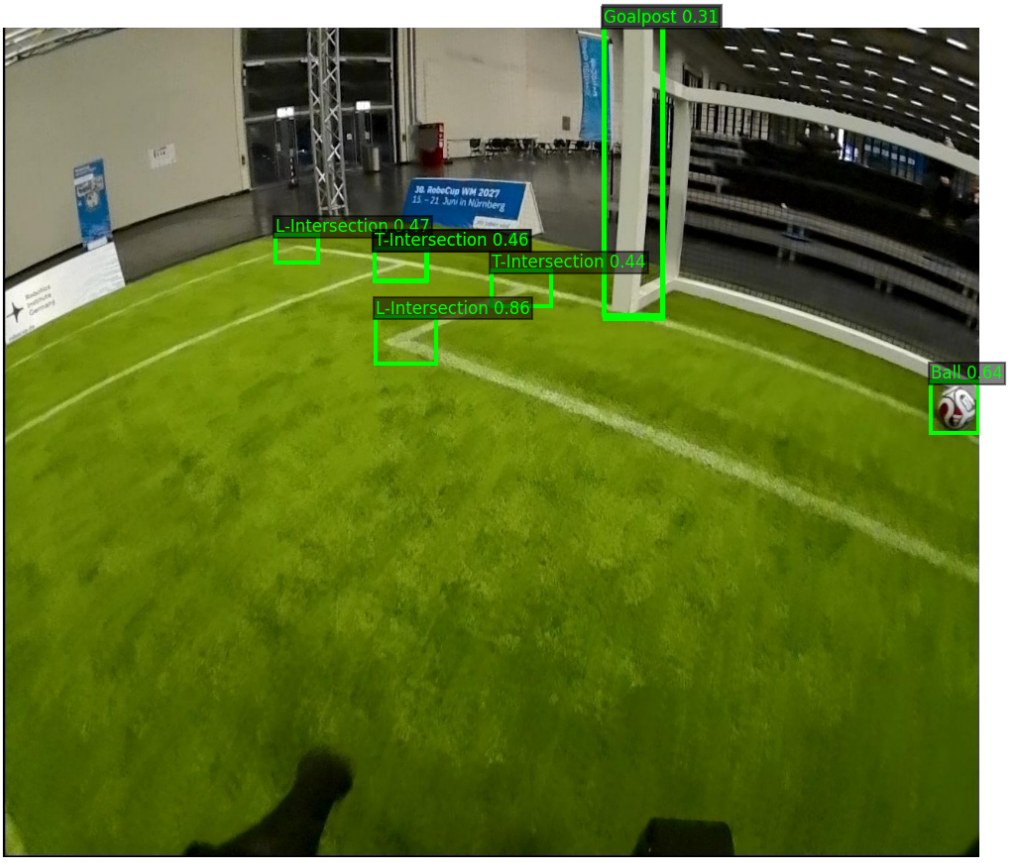


Figure 3.3: Example predictions of the DAv2-S YOLO26 RGB-D model on test image 417.jpg.

3.2 Model Selection

Six models were selected for integration into the pipeline. Three of these models are depth estimators, while the remaining three architectures are object detection models. Model selection was guided by several criteria, with parameter count serving as a key selection criterion. While larger models may yield superior accuracy, their computational demands may exceed the resource constraints of the K1 platform, where the object detection pipeline must run alongside other core processes. Consequently, lightweight models were an important criterion for both object detectors.

The lightweight criterion was applied primarily to the object detectors, where inference directly impacts the detection pipeline. For depth estimators, model capability and availability were the primary selection criteria. An affine-invariant and a metric depth estimator were selected to investigate whether metric depth yields improved detection performance over relative depth, while a large affine-invariant model was included to assess whether depth map quality has a measurable impact on downstream detection performance. Additionally, since most object detection models expect a three-channel RGB image as input, modifying the backbone to accept a four-channel RGB-D tensor requires access to the underlying source code, making source code accessibility a prerequisite for object detector selection. The resulting selection consists of DepthAnythingV2-Small and DepthAnythingV2-Large as affine-invariant depth estimators, ZoeDepth as the metric depth estimator, and YOLO26-S, RT-DETR-L, and D-FINE-S as the object detectors.

3.2.1 Depth Estimators

Following the criteria outlined in Section 3.2, three depth estimators were selected. DepthAnythingV2-Small (Yang, Kang, Huang, Zhao, et al., 2024) and ZoeDepth (Bhat et al., 2023) were selected to isolate affine-invariant depth type from metric depth DepthAnythingV2-Large, on the other hand, serves as a high-capacity upper bound for affine-invariant depth estimation. Furthermore, the shared model architecture of DepthAnythingV2-Small and -Large enables the attribution of differences in downstream detection performance to model capacity rather than architectural variation. An overview of the selected depth estimation models is displayed in Table 3.1.

Table 3.1: Overview of selected depth estimators

Model	**Type**	**Parameters**	**Role**
ZoeDepth	Metric	345M	Metric baseline
DepthAnythingV2-Small	Affine-invariant	25M	Lightweight relative baseline
DepthAnythingV2-Large	Affine-invariant	335M	High-capacity upper bound

3.2.2 Object Detectors

This subsection outlines the selected object detectors. An overview of the key characteristics of the models is displayed in Table 3.2.

YOLO26-S

YOLO26 (Sapkota et al., 2026) is the most recently released object detection model from the YOLO family, of which the small variant, comprising 9.5M parameters, was selected. Its lightweight, CNN-based architecture satisfies the computational constraints outlined in Section 3.2, while its openly available source code enabled the backbone modifications required for four-channel RGB-D input, as described in Section 3.3. The STAL module, specifically designed to boost small object detection performance, was a key motivation for including YOLO26-S in the pipeline, as small objects such as long-range field markings and distant robots are prevalent in RoboCup scenes.

RT-DETR-L

The large variant of RT-DETR (Zhao et al., 2024) was selected, consisting of 33M parameters, serving as the hybrid encoder-based object detector, to be compared against the distribution-based regression approach introduced in the following subsection. As RT-DETR is also available through the Ultralytics framework, the RGB-D backbone modifications were identical to those applied to YOLO26, substantially simplifying the four-channel input adaptation. Its higher parameter count and global context modelling via the hybrid encoder are expected to yield strong class discrimination in the cluttered RoboCup scene, where multiple objects frequently occlude each other.

D-FINE-S

D-FINE (Peng et al., 2025) was selected as the distribution-based regression counterpart to RT-DETR-L, enabling comparison between two distinct transformer-based detection approaches within the same pipeline. The Fine-grained Distribution Refinement approach is expected to be particularly beneficial for small or distinctively shaped objects in RoboCup scenes, such as the ball, where precise bounding box localization directly impacts detection accuracy. Designed for real-time object detection, D-FINE's small variant, comprising 10M parameters, was selected for integration into the pipeline. Source code was publicly available on GitHub, enabling the backbone modifications required for four-channel RGB-D input. Crucially, D-FINE was the best-performing model in terms of mAP in Catarrinho (2026).

Table 3.2: Overview of selected object detectors

Model	**Variant**	**Parameters**	**Backbone**
YOLO26	Small	9.5M	CSP-Darknet
RT-DETR	Large	33M	HGNetv2
D-FINE	Small	10M	HGNetv2

3.3 Training Strategy

3.3.1 Data Preparation

Dataset preparation was essential as the three object detectors required different dataset organization. YOLO26 and RT-DETR expect a split-level organization, i.e. a corresponding `labels/` folder alongside every `images/` folder within each data split. D-FINE, on the other hand, expects a top-level `annotations/` folder residing at the same level as the split folders themselves. The annotations formats differ as well: YOLO26 and RT-DETR demand per-image `.txt` files, whereas D-FINE requires a dedicated COCO-format (Lin et al., 2014) `.json` file for every data split. Since both annotation formats were derived from the same source annotations, a unified directory structure was implemented instead of maintaining separate dataset copies for each detector, as illustrated in Figure A.1. Additionally, the ground-truth depth maps are organized per-model within each split, meaning each of the three depth estimators has its own subdirectory, keeping depth maps separated by model for easy switching during experiments. As YOLO26 and RT-DETR require YOLO-style

configuration files, two configuration files were included: `rgb.yaml` and `rgbd.yaml`. These point to the RGB-only and RGB-D versions of the dataset respectively, enabling simple switching between the baseline and RGB-D pipelines. The dataset unification approach was inspired by Catarrinho (2026), who dealt with similar dataset organization challenges.

3.3.2 Transfer Learning

Due to the relatively small size of the dataset ($\sim$ 5K images), training from scratch posed a risk of overfitting. Therefore, transfer learning was implemented by initializing all object detection models using weights pre-trained on the COCO dataset. However, as COCO is a general-purpose dataset, fine-tuning on the HSL dataset was necessary to learn HSL features such as field markings and K1 robots. As pre-trained weights exist only for RGB channels, the depth channel weights required separate initialization. For all models, the depth channel weights were initialized using He Normal initialization (He et al., 2015), which is specifically suited to layers followed by ReLU-family activation functions, as used across all three object detectors, preserving gradient stability during early training.

This instability was further mitigated for YOLO26 by freezing the backbone for 35 epochs. Preliminary training runs of YOLO26 without a frozen backbone showed no measurable mAP improvement after 50 epochs, suggesting that the model failed to converge under these circumstances. Freezing the backbone for 35 epochs resolved this, as this protected the RGB weights from noisy gradients. RT-DETR and D-FINE were fully fine-tuned from the start, as these architectures trained stably without requiring a frozen backbone phase.

3.3.3 Backbone Modification

As the used object detection models are designed to accept a three-channel RGB image, various backbone modifications were required to accept four-channel input. The implemented modifications are identical across all three architectures. However, the modification strategies differed, as YOLO26 and RT-DETR are Ultralytics-based, and D-FINE uses a custom, non-Ultralytics training loop.

Each model's architecture is defined by a YAML configuration file, which specifies the number of input channels, the classes, and image folder paths. The addition of the argument `input_channels: 4` to the model YAML sufficed for accepting a four-channel tensor, instantiating the first convolutional layer of the backbone with shape `(out_channels, 4, k, k)`. Furthermore, the paths to the folders containing the ground truth depth maps of the training and validation splits were added to the YAMLs.

As the original dataloaders were specifically designed to process three-channel RGB images, a custom dataloader class was implemented to feed four-channel RGB-D tensors to the models. The depth paths in the YAML were essential for the custom dataloader to locate the corresponding depth map to a given RGB image. To streamline the dataloading process, depth maps were saved with the same filename of its corresponding RGB image, allowing the dataloader to locate them by name.

The transfer learning strategy entailed copying the pre-trained RGB weights into the first three slots of a four-channel tensor. The depth channel, i.e. the fourth input channel of the weight tensor, was initialized with He Normal weights. This weight surgery, and the custom dataloader class, were implemented inside the training script for the Ultralytics models, whereas for D-FINE, both were implemented directly in the source code.

3.3.4 Convergence and Checkpointing

All models were trained for a maximum of 300 epochs, with an early stopping policy of 30 epochs to prevent overfitting and reduce training time. Early stopping was triggered if the mAP50-95 score on the validation set did not improve for 30 consecutive epochs. The best performing checkpoint, defined by the highest mAP50-95 on the validation set, was saved and used for evaluation.

3.4 Evaluation

3.4.1 Pipeline Comparisons

In this thesis, two comparisons of varying scales will be made in order to evaluate whether the addition of depth improves object detection performance. Zooming out, two entire pipelines, an RGB based and an RGB-D based object detection model, are compared and assessed whether accuracy increases with the addition of the depth modality. Zooming in the pipeline, the second comparison will serve for evaluating the effectiveness of the different combinations of used depth estimation and object detection models. For example, one combination could be DepthAnythingV2 as a depth estimator, and YOLO26 as an object detection model. In order to substantively compare different pipeline combinations, the design choice was made to use three different depth estimators and three different object detectors, constituting of nine distinctive model combinations. Furthermore, as Catarrinho (2026) evaluated D-FINE-S in a similar HSL RoboCup context, his results serve as an additional reference point for the D-FINE-S comparisons, supplementing the RGB-only baseline established within this thesis.

3.4.2 Metrics

These comparisons will be quantified using the mean average precision (mAP), which evaluates predictions against ground truth bounding boxes using intersection over union (IoU), i.e. the ratio of the overlapping area to the combined area of the predicted and ground truth boxes, as a correctness criterion:

- **The mAP50-95** measures how closely the predicted bounding box aligns with the ground truth. This is done by averaging the mAP across intersection over union, thresholds from 0.5 to 0.95, in steps of 0.05.
- **The $\mathbf{AP_S}$, $\mathbf{AP_M}$, and $\mathbf{AP_L}$** metrics represent the mAP50-95 scores for respectively small ($<$1024 px^2), medium (1024–9216 px^2), and large objects ($\geq$ 9216 px^2), according to COCO object size thresholds.

As an addition to mean average precision scores, the confidence distributions of true positive detections were analyzed as a supplementary measure to further characterize detection behavior. Confidence distributions are estimated using kernel density estimation using the SciPy library (Virtanen et al., 2020), applied only to true positive predictions, as false positives and false negatives reflect localization or classification errors rather than confidence in correct detections.

A qualitative analysis will form the last form of evaluation in this thesis to complement the quantitative metrics. This analysis, performed on a representative test set image, provides visual insight into detection behavior across RGB-only and RGB-D pipelines, including the nature of false positive predictions and the impact of depth augmentation on specific object classes.

Chapter 4

Experiments & Results

This chapter presents the experimental setup and results of the nine RGB-D pipelines evaluated in this thesis, each combining one of three depth estimators (DepthAnythingV2-Small, DepthAnythingV2-Large, and ZoeDepth) with one of three object detectors (YOLO26-S, RT-DETR-L, and D-FINE-S) via early fusion. Results are evaluated against three RGB-only baselines using mAP50-95 as the primary metric, with AP_S, AP_M, and AP_L providing scale-specific analysis. Confidence distributions of true positive detections and a qualitative scene analysis serve as supplementary measures. The chapter first describes the experimental setup, followed by quantitative results, ablation study findings, confidence distributions, and a qualitative analysis.

4.1 Experimental Setup

4.1.1 Dataset

Model training was conducted on the HSL Objects V1 (Whirlwind Amsterdam, 2025) dataset, consisting of 5037 monocular RGB images with COCO-style bounding box annotations. Images were specifically gathered for object detection training in the Humanoid Soccer League environment, as the dataset primarily contains images obtained during various RoboCup (Asia Pacific Beijing Masters 2025, Salvador 2025, German Open 2025, and German Open 2026) editions, while the remaining images were acquired from test runs in the Intelligent Robotics Lab. It is important to note that, as the dataset did not include ground-truth depth maps, these were generated using the three monocular depth estimation models listed in Table 3.1. This effectively doubled the dataset size.

The dataset contained seven different classes: ball, goalpost, K1, L-intersection, penalty mark, T-intersection, and X-intersection. These classes represent the key objects a robot must detect during HSL RoboCup matches, including the ball, field markings (L-, T-, and X-intersection), goalposts, and opponent robots (K1). A visualization of object classes with per-class sample counts is shown in Figure 4.1. The figure demonstrates a somewhat disproportionate class distribution, with significantly more instances of the K1, L-intersection, and T-intersection classes.

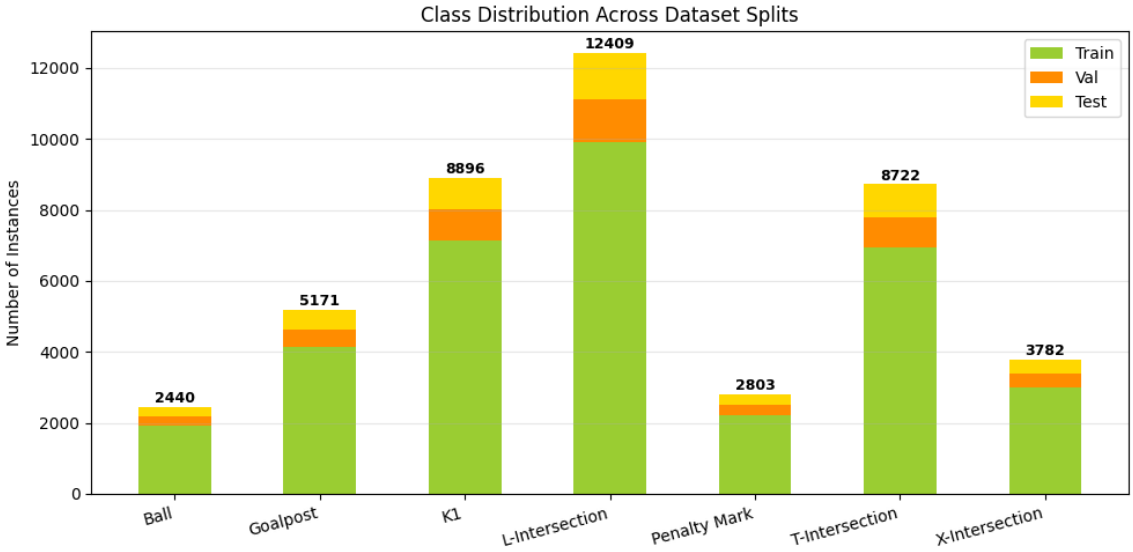


Figure 4.1: Class distribution of HSL Objects V1 dataset.

4.1.2 Dataset Variability

The dataset, while it predominantly contained high-quality RGB images, also included more challenging samples. Some indicators of high-quality RGB images are sharpness, clear line markings, and stable lighting conditions. Challenging samples include images with e.g. overexposure, motion blur, and distant objects. The quality of a depth map is dependent on the quality of the corresponding RGB image, as monocular depth estimators are used to generate depth. For instance, an RGB image with heavy motion blur, also results in a noisy depth map with blurry object outlines and poorly defined depth transitions. High-quality depth maps, on the other hand, display clear object contours and continuous depth surfaces. Examples of high-quality and challenging RGB-depth image pairs of the dataset are respectively shown in Figures 4.2 and 4.3.

Another source of dataset variability stemmed from the wide range of image resolutions, with image samples coming in seven different resolutions, as shown in Table A.1. Since the used object detection models were trained with images resized to 640×640 during preprocessing, this potentially compressed smaller objects in these images, introducing variability in the apparent size of objects across samples. The image resolution distribution across data splits can be found in Table A.1.

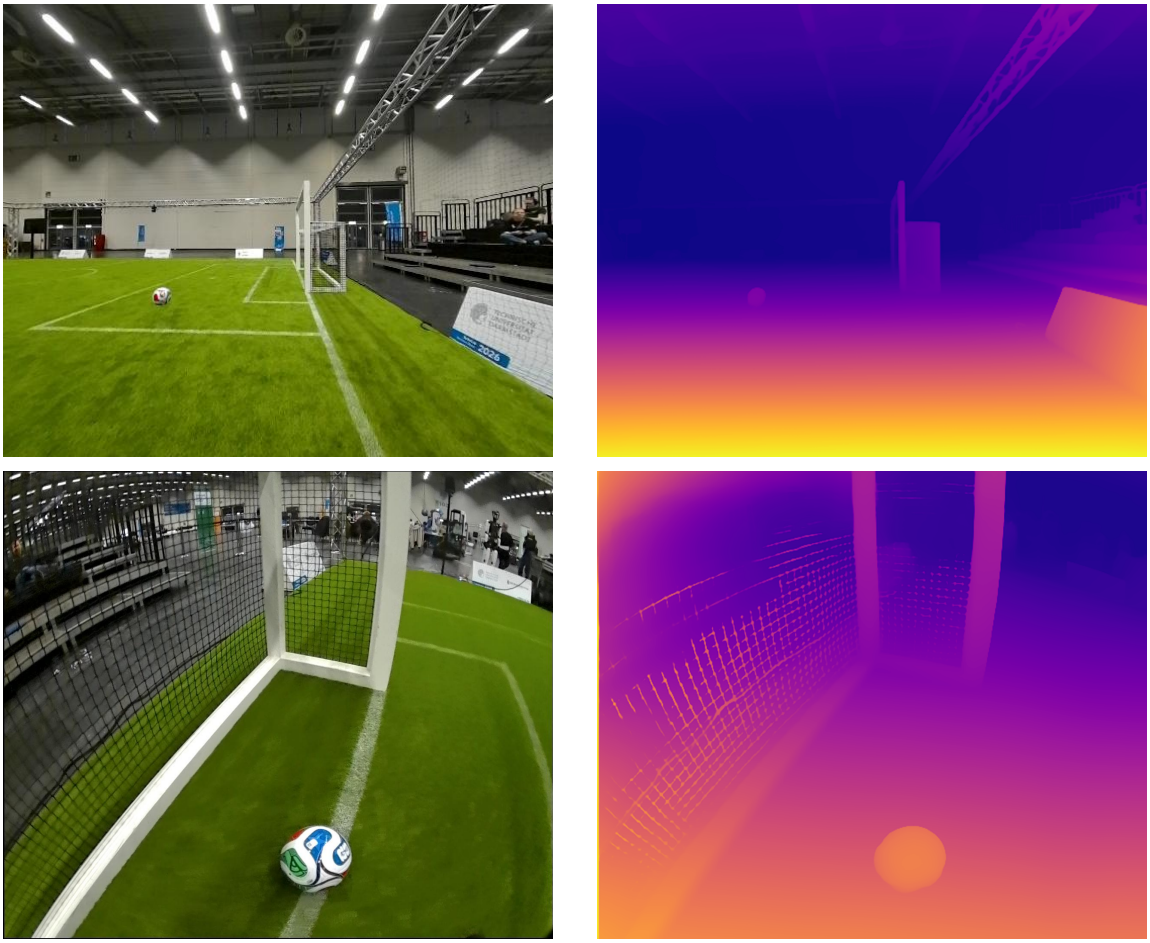


Figure 4.2: Two high-quality image pairs from the HSL Objects V1 dataset, with depth maps generated by DepthAnythingV2-Large.

Figure 4.2 displays two high-quality image pairs from the dataset. The white field markings are clearly visible and high-contrast against the green pitch, with unoccluded objects and stable lighting conditions. The corresponding depth maps include clearly visible balls and goalposts, which appear as sharp depth discontinuities against the background. The green pitch is depicted as a smooth, continuous gradient, indicating the monocular depth estimator's high capability of handling the textureless, green field.

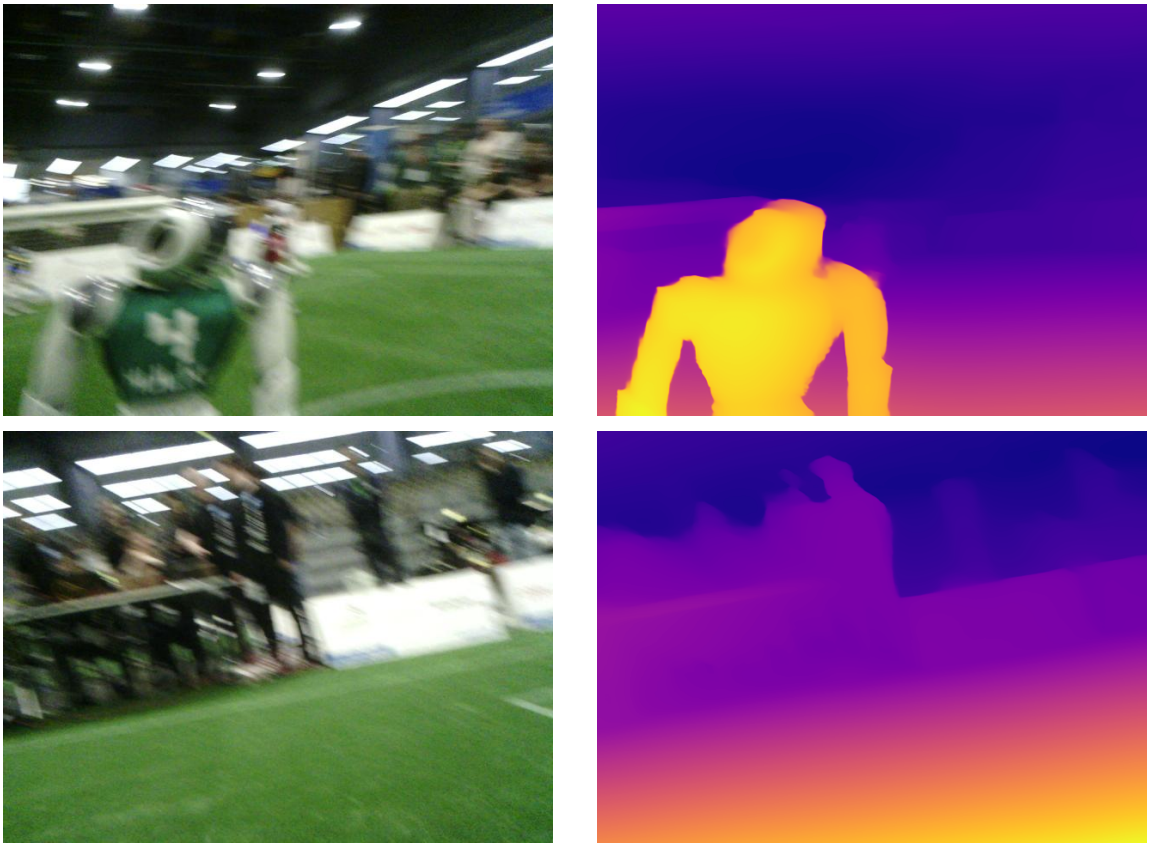


Figure 4.3: Two challenging image pairs from the HSL Objects V1 dataset, with depth maps generated by DepthAnythingV2-Large.

Figure 4.3 displays two challenging image pairs from the dataset. Both RGB images show tilted camera angles and heavy motion blur, which is reflected in depth maps as objects boundaries appear less sharp. However, even in these challenging samples, the green pitch is still represented as a smooth depth gradient. Notably, the depth map of the upper pair retains a clearly distinguishable robot silhouette despite the motion blur in the RGB image, suggesting the depth estimator can partially recover spatial structure under moderate blur. In the lower pair, however, the severe motion blur results in a depth map devoid of object-level detail, with the scene collapsing into broad, undifferentiated depth regions.

4.1.3 Train-Valid-Test Splits

The dataset was divided into train, validation and test sets using a 80/10/10 ratio. A stratified splitting strategy was implemented in order to maintain a fair distribution of images, object instances, and classes across all splits. Details regarding the three data splits are displayed in Tables 4.1 and 4.2.

Table 4.1: Number of images and objects across dataset splits.

	Number of Images		**Number of Objects**	
Split	n	%	n	%
Train	4029	80.0	35262	79.7
Validation	504	10.0	4363	9.9
Test	504	10.0	4598	10.4
Total	5037	100	44223	100

Table 4.2: Class instances across dataset splits.

	Count (n)			**Percentage (%)**		
Class	Train	Val	Test	Train	Val	Test
Ball	1927	261	252	5.5	6.0	5.5
Goalpost	4133	504	534	11.7	11.6	11.6
K1	7134	877	885	20.2	20.1	19.2
L-Intersection	9899	1221	1289	28.1	28.0	28.0
Penalty Mark	2224	280	299	6.3	6.4	6.5
T-Intersection	3001	372	409	8.5	8.5	8.9
X-Intersection	6944	848	930	19.7	19.4	20.2

4.1.4 Training Setup

Similar training hyperparameters across all models were applied in order to keep model comparisons as fair as possible. Training circumstances between RGB and RGB-D models were identical, with YOLO26-S pipelines being the only exception. The first 35 epochs of the YOLO26-S models were trained using a frozen backbone, followed by up to 300 epochs of training on an unfrozen backbone. This multi-phase strategy was implemented, as RGB-D YOLO26-S models struggled with gradient explosion when training without a frozen backbone. Furthermore, since batch size is determined by available GPU memory per model architecture, this hyperparameter differs across all model families, ranging from 4 to 16. All models were trained until convergence on an NVIDIA GeForce RTX 2080 Ti. Visualizations of the training curves are provided in Figures B.1, B.2, and B.3.

Table 4.3: Training hyperparameters for all object detection models.

Hyperparameter	YOLO26-S	RT-DETR-L	D-FINE-S
Optimizer	AdamW	AdamW	AdamW
Batch size	8 / 16[a]	8	4
Frozen epochs	35	—	—
Max epochs (total)	335	300	300
Early stopping	30	30	30
Image size	640	640	640

[a] Batch size 8 during frozen backbone phase (35 epochs), 16 after unfreezing.

4.2 Results

4.2.1 Overall Results across Pipelines

The overview in Table 4.4 displays the mAP50-95 scores across all pipelines, including RGB baselines and RGB-D models. RGB baseline scores for YOLO26 and RT-DETR substantially outperform their respective RGB-D pipelines, with RGB-D YOLO26 models exhibiting a larger performance drop (-0.202 on average) compared to the performance drop of RT-DETR (-0.107 on average). The best performing pipeline belongs to the RGB-D pipeline with DAv2-S as depth estimator and D-FINE-S as object detector, which marginally outperform their RGB-only baseline by 0.002 on average. Additionally, mAP50-95 scores across depth estimators within each RGB-D family remain fairly consistent, with differences not exceeding 0.014 for YOLO26 and RT-DETR, and as small as 0.001 for D-FINE.

Table 4.4: Overall mAP50-95 across all classes for all pipelines.

Depth Estimator	**Object Detector**	**mAP50-95**
RGB Baselines		
—	YOLO26-S	0.724
—	RT-DETR-L	0.749
—	D-FINE-S	0.751
YOLO26-S		
DAv2-S	YOLO26-S	0.519
DAv2-L	YOLO26-S	0.517
ZoeDepth	YOLO26-S	0.531
RT-DETR-L		
DAv2-S	RT-DETR-L	0.648
DAv2-L	RT-DETR-L	0.642
ZoeDepth	RT-DETR-L	0.636
D-FINE-S		
DAv2-S	D-FINE-S	**0.754**
DAv2-L	D-FINE-S	0.753
ZoeDepth	D-FINE-S	0.753

4.2.2 Detection Performance by Object Scale

The overview in Table 4.5 displays the AP_S, AP_M, and AP_L scores across RGB-only and RGB-D pipelines. The RGB-D pipelines with YOLO26 and RT-DETR show significantly lower scores in comparison to their RGB-only baselines, with the RGB-D RT-DETR models exhibiting the highest performance drop on average across object scales (-0.387 for small objects; -0.313 for medium objects; -0.202 for large objects). The DAv2-S D-FINE pipeline performs comparably to the RGB-only D-FINE baseline for small and medium objects, while achieving a marginally higher AP_L score of 0.012. When examining depth estimator choice specifically, ZoeDepth paired with YOLO26 shows a consistent but marginal advantage over DAv2 variants across all object scales, while for RT-DETR and D-FINE, depth estimator choice has negligible impact across.

Table 4.5: AP_S, AP_M, and AP_L averaged across all classes for all pipelines.

Depth Estimator	**Object Detector**	$\mathbf{AP_S}$	$\mathbf{AP_M}$	$\mathbf{AP_L}$
RGB Baselines				
—	YOLO26-S	0.582	0.770	0.788
—	RT-DETR-L	0.623	0.785	0.796
—	D-FINE-S	**0.651***	0.786	0.805
YOLO26-S				
DAv2-S	YOLO26-S	0.305	0.606	0.674
DAv2-L	YOLO26-S	0.303	0.609	0.664
ZoeDepth	YOLO26-S	0.320	0.622	0.684
RT-DETR-L				
DAv2-S	RT-DETR-L	0.227	0.471	0.594
DAv2-L	RT-DETR-L	0.237	0.473	0.590
ZoeDepth	RT-DETR-L	0.244	0.471	0.597
D-FINE-S				
DAv2-S	D-FINE-S	**0.651***	**0.788***	**0.817**
DAv2-L	D-FINE-S	0.646	**0.788***	0.798
ZoeDepth	D-FINE-S	0.649	0.785	0.802

* Indicates a tied score with another pipeline for this metric.

4.3 Ablation Studies & Diagnostic Experiments

4.3.1 Depth Channel Contribution (zero-out)

The following tables display the results of the ablation study where the depth channel weights were zeroed out across all RGB-D pipelines to assess the degree to which each model relied on depth information. These results were compared against the mAP50-95 scores obtained with learned depth channel weights. Table 4.6 exhibits the ablation experiment for RGB-D YOLO26 pipelines, showing a mean performance drop of -0.231, suggesting that the depth-enhanced YOLO26 models substantially relied on depth for predictions. Table 4.7 shows a mean mAP50-95 drop of -0.147 for RT-DETR models, implying that RT-DETR utilized depth maps to make predictions. However, the D-FINE models, as displayed in Table 4.8, exhibit a marginal mean performance drop of -0.006, indicating that the D-FINE models have essentially ignored the depth channel.

Table 4.6: Impact of zeroing the depth channel on YOLO26-S RGBD model performance.

Depth Estimator	**Full RGB-D mAP50-95**	**Depth Zeroed mAP50-95**	**ΔmAP50-95**
DAv2-S	0.5194	0.3011	-0.218
DAv2-L	0.5167	0.2864	-0.230
ZoeDepth	0.5309	0.2858	**-0.245**

Table 4.7: Impact of zeroing the depth channel on RT-DETR-L RGBD model performance.

Depth Estimator	**Full RGB-D mAP50-95**	**Depth Zeroed mAP50-95**	**ΔmAP50-95**
DAv2-S	0.6477	0.5145	-0.133
DAv2-L	0.6416	0.4772	**-0.164**
ZoeDepth	0.6355	0.4908	-0.144

Table 4.8: Impact of zeroing the depth channel on D-FINE-S RGBD model performance.

Depth Estimator	**Full RGB-D mAP50-95**	**Depth Zeroed mAP50-95**	**ΔmAP50-95**
DAv2-S	0.7537	0.7501	-0.004
DAv2-L	0.7525	0.7491	-0.004
ZoeDepth	0.7530	0.7432	**-0.010**

4.3.2 Confidence Distributions of True Positive Detections

The following figures illustrate the confidence distributions of true positive detections across both RGB-only and RGB-D pipelines, with dashed vertical lines representing the mean confidence scores of each pipeline.

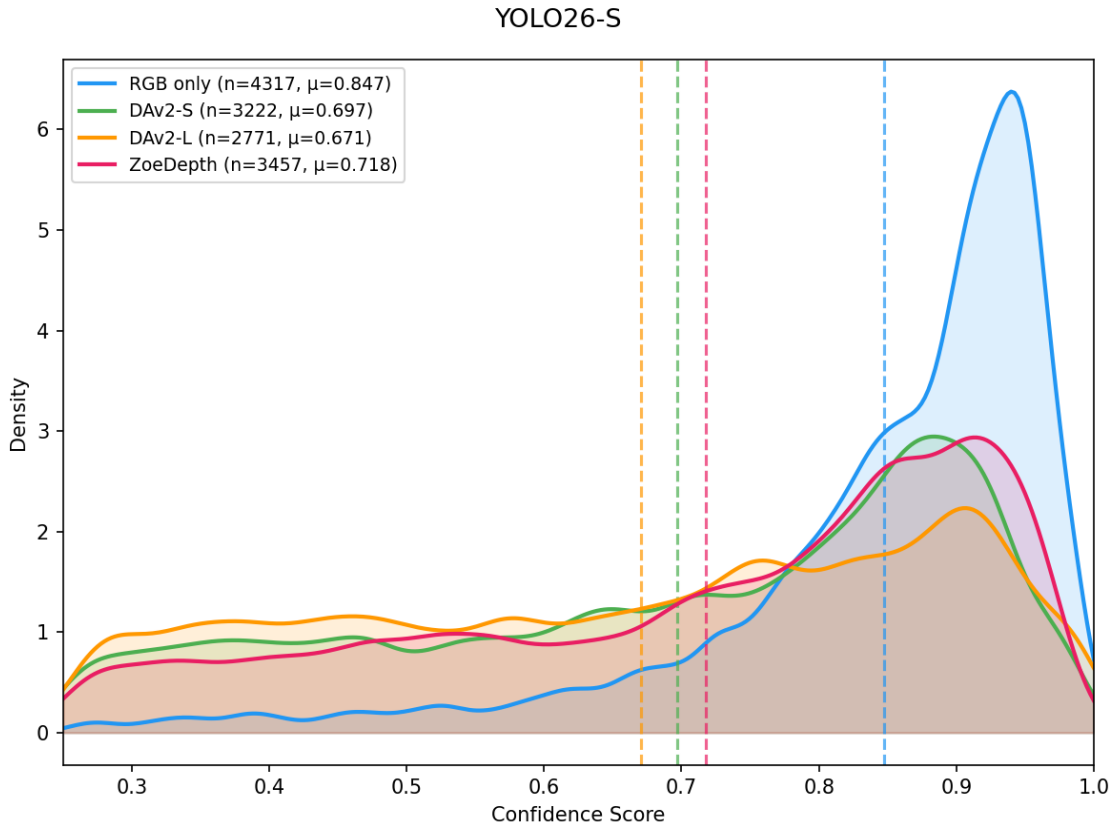


Figure 4.4: Kernel density estimates of true positive detection confidence scores for YOLO26-S across RGB-only and RGB-D input modalities.

As illustrated in the plot above, the confidence distribution of RGB-only YOLO26, with high density areas concentrated in the upper confidence range, contrasts sharply with the RGB-D YOLO26 pipelines, which display significantly lower and more fluctuating peaks. The RGB-only model's substantially higher mean confidence (μ=0.847) compared to its RGB-D counterparts (DAv2-S: μ=0.697, DAv2-L: μ=0.671, ZoeDepth: μ=0.718) further reinforces this difference. Among the RGB-D pipelines, ZoeDepth yields the highest confidence score for true positive detections, along with the highest true positive predictions compared to other RGB-D YOLO26 pipelines. However, the RGB-only baseline remains the model with the highest true positive detections, producing 1167 more true positives on average compared to the RGB-D counterparts.

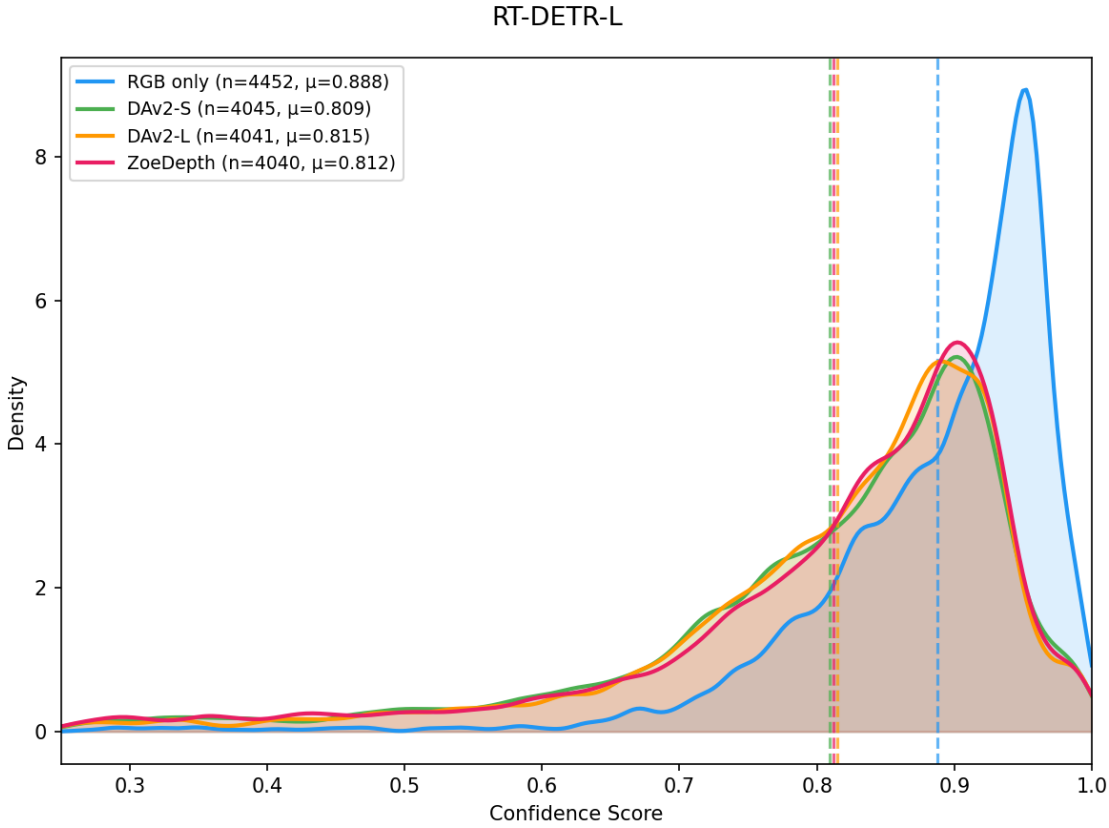


Figure 4.5: Kernel density estimates of true positive detection confidence scores for RT-DETR-L across RGB-only and RGB-D input modalities.

The plot above illustrates the confidence distributions across RT-DETR models. While the RGB-D models display peaks in the upper range of confidence scores, the peak of the RGB-only RT-DETR model is narrower and shifted further right compared to the RGB-D peaks. As shown in Figure 4.4, the confidence gap between RGB-only and RGB-D pipelines is smaller for YOLO26 than observed for RT-DETR, with the RGB-only model's mean confidence (μ=0.888) sitting closer to its RGB-D counterparts (μ=0.809-0.815). Additionally, the RGB-D RT-DETR pipelines roughly follow the same distribution along with essentially equal mean confidence scores, indicating that depth estimator choice has minimal impact on confidence scores for RT-DETR. Furthermore, the RGB-only baseline made approximately 410 more true positive predictions on average compared to the RGB-D pipelines.

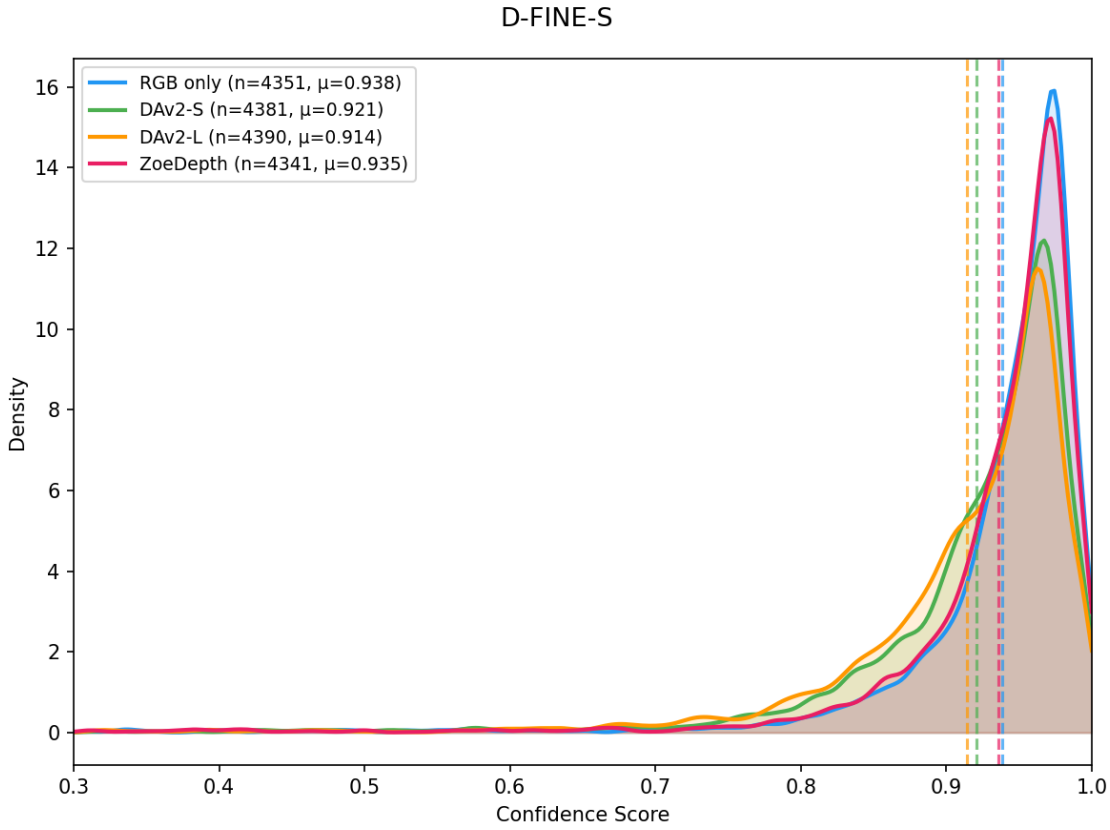


Figure 4.6: Kernel density estimates of true positive detection confidence scores for D-FINE-S across RGB-only and RGB-D input modalities.

The plot above shows the confidence distribution of RGB-only and RGB-D D-FINE models. While the peaks of all models are located at approximately the same upper confidence range, DAv2-S and DAv2-L based RGB-D models have lower density peaks. Additionally, DAv2-S and DAv2-L based D-FINE models follow roughly the same distribution, while ZoeDepth-based D-FINE and RGB-only D-FINE display a similar shaped distribution. This grouping is also reflected in mean confidence scores, with DAv2-S (μ=0.921) and DAv2-L (μ=0.914) scoring lower than ZoeDepth (μ=0.935) and RGB-only (μ=0.938). However, DAv2-S and DAv2-L based D-FINE produce somewhat more true positive detections (approximately 35 more on average). As opposed to the YOLO26 and RT-DETR RGB-D pipelines, the RGB-D pipelines of D-FINE produce very similar distributions. Notably, ZoeDepth-based D-FINE is the only RGB-D pipeline across all nine evaluated in this thesis to achieve a mean confidence score comparable to its RGB-only counterpart (μ=0.935 vs μ=0.938).

4.4 Qualitative Analysis

This section shows detection results for one representative HSL RoboCup scenario, including long-range detections, occluded objects and a cluttered background. The following results are organized per object detector family. For each family, two figures are shown: predictions made by the RGB-only model and predictions made by the best-performing RGB-D pipeline as determined by mAP50-95 in Table 4.4. The predictions made by the remaining RGB-D pipelines are visualized in Figures C.1, C.2, and C.3. Predictions across all pipelines will be made on test image 2973.jpg, as shown in Figure 4.7.

Figure 4.7: Test image 2973.jpg, annotated with ground truth labels.

4.4.1 YOLO26-S

Figures 4.8a and 4.8b depict a direct comparison of RGB-only YOLO26 and ZoeDepth YOLO26 predictions, with green bounding boxes representing true positives, red bounding boxes representing false negatives, and orange bounding boxes representing false positives. The performance gap between the RGB and RGB-D pipeline is clear through the difference in true positive (TP) detections: while RGB-only YOLO26 predicts 21 true positives, ZoeDepth-based YOLO26 predicts only 13. Additionally, RGB-D YOLO26 predicts 3 false positives (FP) while the RGB-only model generates 1 FP. The false positive predictions of both models can be attributed to duplicate predictions for a single object, as visualized by the overlapping orange and green bounding boxes in the figures. This may have been caused by the lack of the Non-Maximum Suppression (NMS) module in YOLO26, resulting in duplicate bounding boxes not being suppressed. Regarding false negatives, the RGB-only model misses one occluded T-intersection near the goalpost, whereas the RGB-D model misses nearly all long-range field markings near the goalposts. The only field marking detected by the RGB-D model is the L-intersection, which was flagged as a false positive due to it falling just below the Intersection over Union (IoU) threshold of 0.5, suggesting an imprecise localization rather than a detection failure.

To reiterate why YOLO26 was one of the assessed models, YOLO26 was included as one of the object detectors in this thesis as it was expected to demonstrate a strong small object detection ability, due to its Small-Target-Aware Label Assignment (STAL) module. The RGB-only predictions indeed corroborate this by correctly detecting small objects, such as the penalty mark, and the T- and L-intersections close to the goalposts. The ZoeDepth-based YOLO26 model failed to detect these field markings, which is consistent with the fact that flat, painted surfaces such as field markings are imperceptible in monocular depth maps. The depth channel therefore provided no useful signal for these objects, potentially introducing noise that disrupted their detection.

(a) RGB-only YOLO26-S predictions. (b) Best RGB-D YOLO26-S predictions (ZoeDepth).

Figure 4.8: Qualitative detection results for YOLO26-S. Green: true positive (TP); orange: false positive (FP); red: false negative (FN).

4.4.2 RT-DETR-L

Unlike YOLO26, the RGB-only baseline of RT-DETR produces zero false positives, correctly detecting the 22 objects, as shown in Figure 4.9a. Additionally, the performance gap between RGB and RGB-D pipelines is smaller for RT-DETR in comparison to YOLO26 pipelines, with DAv2-S based RT-DETR producing 20 true positive detections and 3 false positives. The three false positives stem from localization issues on two objects: two near-identical L-intersection predictions both falling just below the IoU threshold, and one duplicate T-intersection prediction whose ground truth box was already cconvolutionalmed by a true positive detection. Notably, the missed L-intersection simultaneously appears as both a false negative and a false positive, indicating an imprecise localization rather than a detection failure. Similar to the duplicate predictions observed in YOLO26, this behavior can be attributed to RT-DETR's NMS-free architecture, where redundant low-confidence predictions are not suppressed in post-processing.

Reflecting on the motive for including RT-DETR in this thesis, this particular object detection model was chosen as a high-capacity model, expected to demonstrate strong class discrimination. This expectation has been met when analyzing results of the RGB-only model, producing 22 true positive detections with zero FPs and zero FNs. For the RGB-D model, the smaller performance gap between RGB-only and RGB-D for RT-DETR compared to YOLO26 suggests that higher model capacity partially compensated for the noise introduced by the depth channel.

(a) RGB-only RT-DETR-L predictions.

(b) Best RGB-D RT-DETR-L predictions (DAv2-s).

Figure 4.9: Qualitative detection results for RT-DETR-L. Green: true positive (TP); orange: false positive (FP); red: false negative (FN).

4.4.3 D-FINE-S

Figures 4.10a and 4.10b show the RGB-only and DAv2-S D-FINE predictions respectively. Both models detect the same number of objects, with only the RGB-D pipeline predicting one false positive, resulting in the smallest performance gap between RGB-only and RGB-D pipelines across all three object detector families evaluated in this thesis. Similar to the duplicate predictions observed in RT-DETR, this false positive was a duplicate prediction of an L-intersection falling just below the IoU threshold of 0.5, representing an imprecise localization rather than a detection failure, as evidenced by the same L-intersection simultaneously appearing as a false negative. This implies that the false positives across the RT-DETR and D-FINE RGB-D models were a result of strict IoU thresholding rather than genuine detection failures.

D-FINE's Fine-grained Distribution Refinement (FDR) approach to predicting bounding boxes was specifically designed to produce tight bounding boxes around objects. Comparing D-FINE's bounding boxes to those produced by YOLO26 and RT-DETR, no observable difference in localization precision is apparent at this scene resolution. D-FINE was additionally chosen for its strong performance in the HSL RoboCup context, as demonstrated by Catarrinho (2026). This strong performance is corroborated by both the near-identical TP/FP counts between RGB-only and RGB-D D-FINE and the comparable mAP50-95 scores between these two pipelines, as shown in Table 4.4.

(a) RGB-only YOLO26-S predictions.

(b) Best RGB-D YOLO26-S predictions (ZoeDepth).

Figure 4.10: Qualitative detection results for D-FINE-S. Green: true positive (TP); orange: false positive (FP); red: false negative (FN).

Chapter 5

Conclusion & Discussion

5.1 Conclusion

This thesis is centered around the following research question: *"Does augmenting RGB input with monocular depth estimation improve object detection accuracy in the RoboCup HSL environment, and does this improvement vary as a function of object scale?"* This overarching research question is answered **negatively**: augmenting RGB input with monocular depth estimation did not improve object detection accuracy in the RoboCup HSL environment, and the impact of depth varied across object scales.

The quantitative evidence supporting this conclusion is detailed in the sub-question answers below, which address: 1) whether depth improves object detection performance; 2) which depth estimator and object detector yields the best detection performance; 3) what the impact of depth information on detection performance across object scales.

The first sub-question asks whether adding an estimated depth channel improves overall object detection performance, as measured by mAP50-95. As shown in Table 4.4, depth augmentation consistently harmed performance for YOLO26 and RT-DETR, with average mAP50-95 drops of 0.202 and 0.107 respectively compared to their RGB-only baselines. This finding is further supported by the confidence distributions of true positive detections (Figures 4.4 and 4.5), which show substantially lower mean confidence scores for RGB-D YOLO26 and RT-DETR pipelines compared to their RGB-only baselines. D-FINE showed negligible mAP50-95 differences between RGB-only and RGB-D pipelines (+0.002 on average, Table 4.4), though ablation results (Table 4.8) confirmed this was due to the model disregarding the depth channel rather than successfully integrating it. Confidence distributions for RGB-D D-FINE models and the RGB-only baseline also showed no notable differences (Figure 4.6). The sub-question is therefore answered negatively: depth augmentation did not improve object detection performance.

The second sub-question asks which pipeline yields the best detection performance using the mAP50-95 as evaluation metric. The best performing RGB-D pipeline combination is DAv2-S coupled with D-FINE-S as object detector, yielding a mAP50-95 score of 0.754 (Table 4.4). However, ablation results confirmed that D-FINE-S did not substantively utilize the depth channel (Table 4.8). Among the pipelines that did utilize depth, the DAv2-S and RT-DETR-L combination

yielded the best mAP50-95 score of 0.648 (Table 4.4).

The third and last sub-question asks whether the addition of depth information differentially impacts detection performance across object scales, as measured by AP_S, AP_M, and AP_L. Of the models that did substantively use the depth channel, drops are observed across all object scales in comparison to their RGB-only baselines. RT-DETR exhibits substantially larger drops across all object scales than YOLO26, with -0.387 for small objects, -0.313 for medium objects, and -0.202 for large objects. For YOLO26 this drop was -0.273 for small objects, -0.158 for medium objects, and -0.114 for large objects, as shown in Table 4.5. D-FINE, which disregarded the depth channel, showed comparable scores to the RGB-only baseline across all object scales. The sub-question is therefore answered positively: depth augmentation did differentially impact performance across object scales, with the largest drops consistently observed for small objects across both YOLO26 and RT-DETR.

5.2 Discussion

5.2.1 Comparison with Related work

Catarrinho (2026) conducted a similar object detection study in the RoboCup HSL context, using RGB-only pipelines, with D-FINE-S as the best performing model achieving a mAP50-95 score of 0.595. The majority of pipelines implemented in this thesis, both RGB-only and RGB-D, outperform this score, with YOLO26-based RGB-D pipelines forming the only exception. However, this comparison is complicated by the dataset used in the research of Catarrinho (2026), which comprised 2,510 images, while the dataset used in this thesis consisted of 5037 RGB images, being roughly double in size. This notable difference in performance can therefore potentially be attributed to incomparable dataset quality.

Two recent RGB-D object detection studies further contextualise the findings of this thesis. Mahjourian and Nguyen (2025) adopt an early fusion approach on a manufacturing dataset with ground-truth depth from a 3D point cloud sensor, finding that their RGB-D model outperforms their RGB-only baseline by 0.055 mAP50. Orfaig et al. (2026) similarly find that RGBX-DiffusionDet, trained on the KITTI autonomous driving dataset using LiDAR depth, outperforms its RGB-only baseline by 0.022 mAP50-95. Both findings directly contrast with this thesis, where RGB-only pipelines substantially outperformed RGB-D pipelines. A key methodological difference is that both studies used sensor-obtained depth maps, while this thesis relied on monocular depth estimation, suggesting that depth source and quality may critically determine whether RGB-D augmentation benefits object detection.

5.2.2 Methodological Reflection

Several aspects of the methodology could be refined to improve the results of this study. Firstly, the early fusion strategy to combine RGB and depth modalities may have limited the model's ability to extract meaningful depth features. Noisy depth maps may have corrupted the input before the model had a chance to learn discriminative features, as early fusion integrates depth at the pixel level before any feature extraction. A late or multi-scale fusion strategy may have enabled more

meaningful depth feature extraction, as these approaches allow each modality to develop independent feature representations before integration.

Additionally, using pre-trained weights for the RGB channels, while initializing the depth channel with He-initialized weights may have created a disparity between the RGB and depth channel weights. The RGB weights essentially needed to be fine-tuned, whereas the depth channel weights had to be learned from scratch. A more balanced approach would have been to initialize all channels, including RGB, with He-initialized weights, eliminating the disparity between pretrained and randomly initialized weights. Ideally, depth channel weights pretrained on RGB-D data from comparable indoor scenes would provide a more informed starting point than random initialization.

Due to time constraints, no hyperparameter tuning specific to the RGB-D input was performed. Parameters such as differential learning rates for pretrained RGB and randomly initialized depth weights, or tuning of the frozen backbone duration, may have improved the model's ability to integrate depth information effectively.

5.2.3 Impact of Depth Estimator Choice

The three depth estimator models were included for two different reasons. Firstly, an affine-invariant model, DepthAnythingV2-Small, and a metric depth estimator, ZoeDepth, were chosen to assess whether depth type influenced object detection performance. Secondly, DepthAnythingV2-Large was included alongside DepthAnythingV2-Small to assess whether depth estimator capacity had a measurable impact on downstream detection performance.

Regarding depth type, the results in Table 4.4 show no consistent advantage of metric depth over affine-invariant depth across object detectors. For YOLO26-S, ZoeDepth marginally outperforms DAv2-S with a difference of 0.012 mAP50-95, while for RT-DETR-L, ZoeDepth performs marginally worse than DAv2-S (-0.012). For D-FINE-S, all three depth estimators produce near-identical scores, with a maximum difference of 0.001. These findings suggest that depth type did not consistently influence detection performance.

The key motivation behind including both small and large variants of DepthAnythingV2 in the object detection pipelines was to assess whether depth estimator capacity had a differential impact on detection performance. As DAv2-S and DAv2-L share the same architecture, any differences in detection performance can be attributed to model capacity alone. The findings of this thesis indicate that depth estimator capacity did not have a notable impact on detection performance, as evidenced by two observations. Firstly, Table 4.4 shows a marginal mAP50-95 difference between DAv2-S and DAv2-L of 0.003 for YOLO26, 0.006 for RT-DETR, and 0.001 for D-FINE. Secondly, Figures 4.4, 4.5, and 4.6 show similarly shaped confidence distributions for DAv2-S and DAv2-L, indicating that depth estimator capacity had no observable effect on detection confidence either.

5.2.4 Detector-Specific Analysis

In Chapter 3, specific expectations were established for each of the three object detectors. This subsection evaluates whether these expectations are reflected in the results.

YOLO26-S was primarily selected for its Small-Target-Aware Label Assignment module, designed to boost small object detection performance. The RGB-only baseline achieves an AP_S score of 0.582 (Table 4.5), indicating moderate small object detection performance. However, the RGB-D pipelines show a mean AP_S drop to 0.309, suggesting that depth augmentation undermined the STAL module's effectiveness rather than complementing it.

RT-DETR-L was selected as the high-capacity transformer-based detector, expected to demonstrate strong class discrimination. The RGB-only baseline supports this expectation, producing 22 true positive detections with zero false positives in the qualitative analysis. However, RT-DETR-L exhibited the largest performance drops across all object scales when augmented with depth, with a mean AP_S drop of 0.387 (Table 4.5), suggesting that its high capacity did not protect against depth-induced performance degradation.

D-FINE-S was selected for its Finegrained Distribution Refinement approach, expected to produce precise bounding box localization particularly for small objects. The RGB-only baseline achieves the highest AP_S score of 0.651 across all detectors, supporting this expectation. However, ablation results confirmed that D-FINE-S effectively disregarded the depth channel (Table 4.8), meaning the comparable RGB-D performance reflects depth ignorance rather than successful depth integration.

5.2.5 Limitations

This study was subject to several limitations that may have influenced the results. Firstly, given that this study relied on transfer learning and fine-tuning, the size of the dataset (approximately 5K images) may have constrained the model's ability to learn robust depth-aware features. The potential benefits of fine-tuning may have been decreased as the benefits of this strategy can best be observed when a larger dataset is used (Yosinski et al., 2014). In particular, the depth channel weights were He Normal initialized and had to be learned from scratch, a process that benefits disproportionately from larger datasets compared to fine-tuning pre-trained RGB weights.

Additionally, the absence of ground truth depth data in the dataset necessitated the use of monocular depth estimation, which introduces an additional source of uncertainty. Estimated depth maps are inherently imperfect approximations, making it difficult to isolate whether detection performance was limited by depth map quality or by the fusion strategy itself. Furthermore, as the used depth estimators were trained on general indoor and outdoor scenes rather than RoboCup-specific environments, the estimated depth maps may contain domain-specific errors, particularly for objects such as robots that are unlikely to appear in the estimators' training data.

5.3 Future Research

As this thesis essentially touches upon two research fields, depth estimation and object detection, there are several ways to enhance this thesis with additional research. For instance, D4RT (Zhang et al., 2026), the paper crowned as the best CVPR paper of 2026, is a feedforward transformer model that jointly infers depth, spatio-temporal correspondence, and camera parameters from a single video. Rather than using separate monocular depth estimation as a preprocessing step, D4RT could provide integrated depth estimation directly from video streams. This is especially relevant for robots performing in the RoboCup, as they use video streams to navigate in real-time

as opposed to camera frames.

Additionally, the pipeline developed in this thesis could be extended to pose estimation. Of the seven object classes in the HSL RoboCup dataset, only three (the ball, K1 robots, and goalposts) have meaningful three-dimensional structure visible in depth maps, suggesting depth may provide more useful spatial features for estimating 3D joint positions than for 2D bounding box detection. Pose estimation is furthermore directly relevant in the RoboCup context, where knowing whether a surrounding K1 robot is fallen, standing, or in motion is valuable for game strategy. YOLO26-Pose (Sapkota et al., 2026) is a natural extension of this thesis, building on the same YOLO26 architecture with a Residual Log-Likelihood Estimation head for keypoint detection (Chakrabarty, 2026), achieving state-of-the-art performance on the COCO-17 benchmark.

Lastly, future work could explore replacing the Vision Transformer (ViT) backbones of the depth estimators employed in this thesis with a spatially aware encoder such as TIPS (Maninis et al., 2025). As the ViT backbones are primarily trained for global image understanding, TIPS excels in combining global and spatial understanding through image-text contrastive learning combined with self-supervised masked image modeling. This may yield more spatially precise depth maps, particularly for small objects. This is especially relevant for this research as AP_S scores across all RGB-D pipelines underperformed compared to their RGB-only counterparts.

Chapter 6

Acknowledgments

I would like to thank my supervisor, Dr. Arnoud Visser, for his continuous guidance and support throughout the two and a half months of writing this thesis. I would also like to express my sincere gratitude to Joey, coordinator of the Intelligent Robotics Lab, for helping me set up the workstations that made training the twelve implemented pipelines possible.

I would furthermore like to thank my parents, who always left a warm plate waiting whenever I was working late at Science Park, and offered a second perspective whenever I needed one. Lastly, I would like to thank my dear friends and supportive boyfriend, without whose support this thesis would not have been possible.

Appendix A

Dataset Details

A.1 Dataset Organization

```
unified_dataset/
├── annotations/
│   ├── train.json
│   ├── valid.json
│   └── test.json
├── train/
│   ├── images/
│   │   ├── image1.jpg
│   │   ├── image2.jpg
│   │   └── ...
│   ├── labels/
│   │   ├── 1.txt
│   │   ├── 2.txt
│   │   └── ...
│   ├── depth_DepthAnythingV2-Large/
│   ├── depth_DepthAnythingV2-Small/
│   └── depth_ZoeDepth-Small/
├── valid/
│   ├── images/
│   ├── labels/
│   ├── depth_DepthAnythingV2-Large/
│   ├── depth_DepthAnythingV2-Small/
│   └── depth-Small/
├── test/
│   ├── images/
│   ├── labels/
│   ├── depth_DepthAnythingV2-Large/
│   ├── depth_DepthAnythingV2-Small/
│   └── depth_ZoeDepth-Small/
├── rgb.yaml
└── rgbd.yaml
```

Figure A.1: Unified dataset directory structure compatible with YOLO26, RT-DETR, and D-FINE.

A.2 Image Resolutions

Table A.1: Image resolution distribution across dataset splits.

	Train		**Val**		**Test**	
Resolution	n	%	n	%	n	%
1280×720	2241	55.6	284	56.3	266	52.8
1280×1088	930	23.1	111	22.0	125	24.8
640×480	738	18.3	92	18.3	93	18.5
1920×1080	64	1.6	8	1.6	8	1.6
640×360	29	0.7	3	0.6	8	1.6
800×600	19	0.5	5	1.0	4	0.8
544×448	8	0.2	1	0.2	0	0.0
Total	4029	100	504	100	504	100

Appendix B

Training Loss & Validation mAP50-95 Plots

B.1 YOLO26-S Models

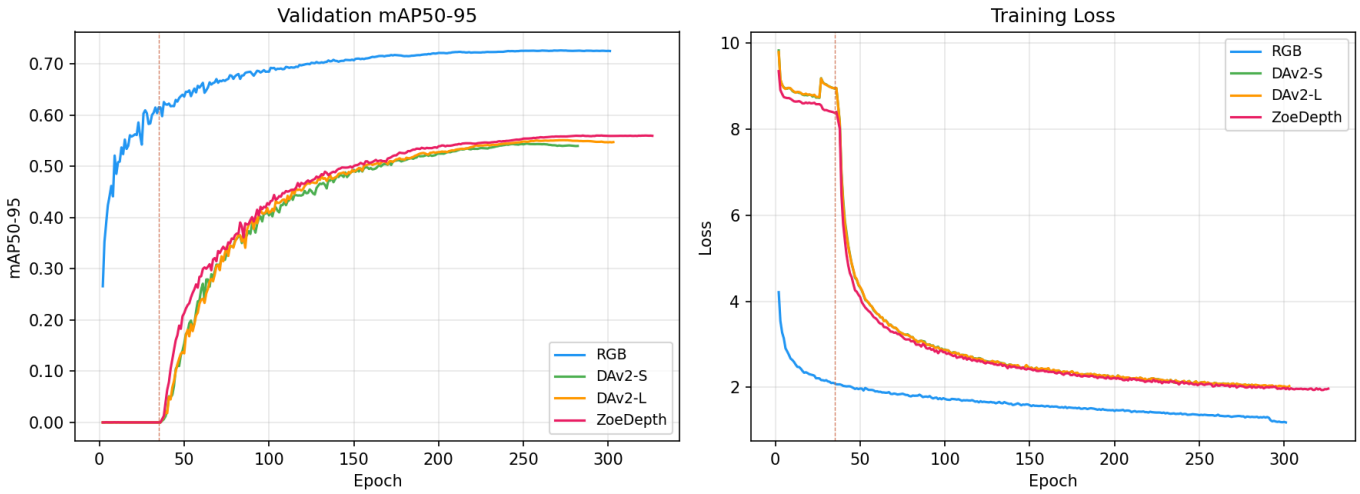


Figure B.1: Training loss and mAP5-95 scores on the validation set for RGB-only and RGB-D YOLO26-S models.

B.2 RT-DETR-L Models

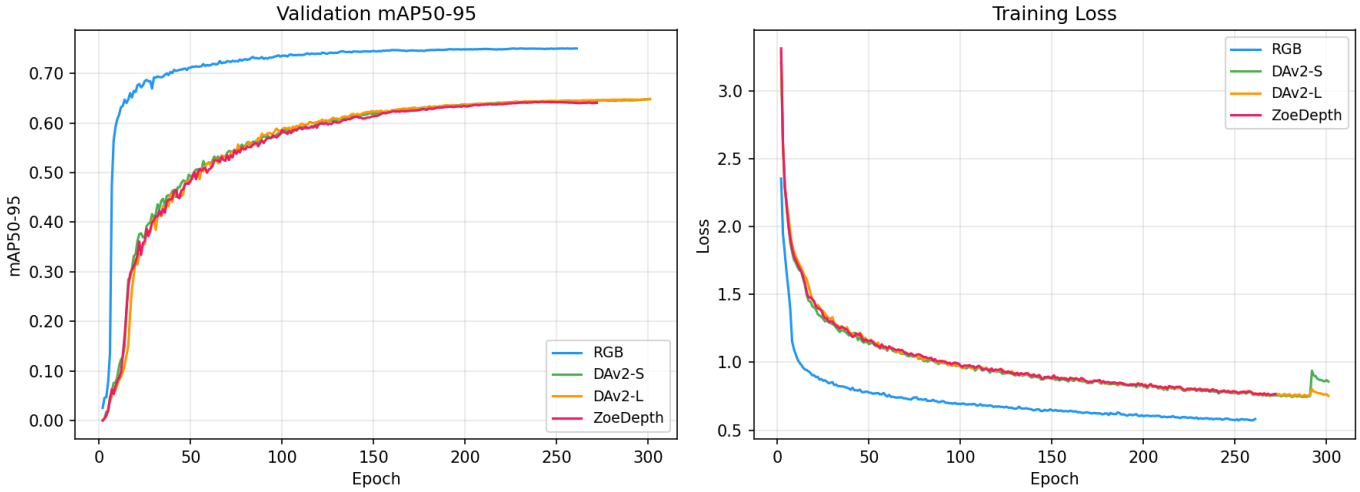


Figure B.2: Training loss and mAP5-95 scores on the validation set for RGB-only and RGB-D RT-DETR-L models.

B.3 D-FINE-S Models

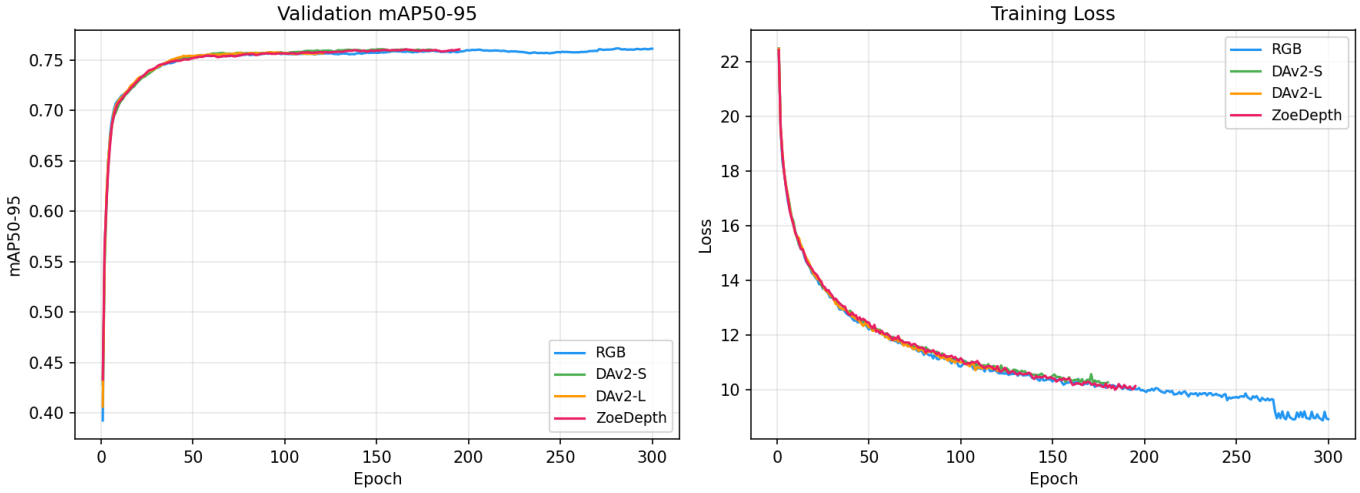


Figure B.3: Training loss and mAP5-95 scores on the validation set for RGB-only and RGB-D D-FINE-S models.

Appendix C

Qualitative Results

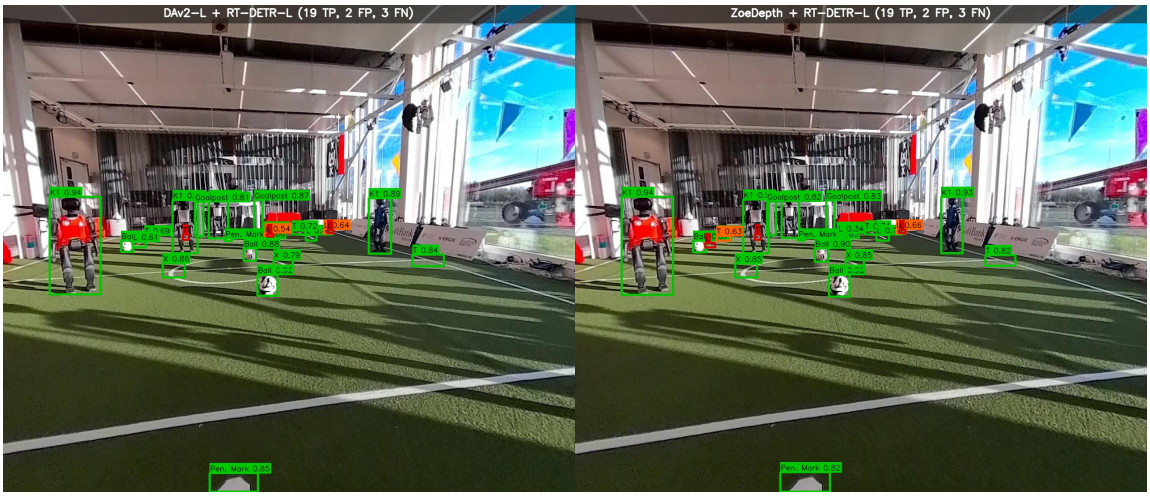
C.1 YOLO26-S

(a) DAv2-S YOLO26 predictions.

(b) DAv2-L YOLO26 predictions.

Figure C.1: Qualitative detection results for YOLO26-S for the remaining pipelines (DAv2-S and DAv2-L combinations).

C.2 RT-DETR-L



(a) DAv2-L RT-DETR predictions. (b) ZoeDepth RT-DETR predictions.

Figure C.2: Qualitative detection results for RT-DETR-L for the remaining pipelines (DAv2-L and ZoeDepth combinations).

C.3 D-FINE-S

(a) DAv2-L D-FINE predictions. (b) ZoeDepth D-FINE predictions.

Figure C.3: Qualitative detection results for D-FINE-S for the remaining pipelines (DAv2-L and ZoeDepth combinations).

Bibliography

Agarap, A. F. (2026). Deep learning using rectified linear units (relu). https://arxiv.org/abs/1803.08375

Arampatzakis, V., Pavlidis, G., Mitianoudis, N., & Papamarkos, N. (2024). Monocular depth estimation: A thorough review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(4), 2396–2414. https://doi.org/10.1109/TPAMI.2023.3330944

Arkin, E., Yadikar, N., Xu, X., Aysa, A., & Ubul, K. (2023). A survey: Object detection methods from cnn to transformer. *Multimedia Tools and Applications*, *82*(14), 21353–21383.

Bhat, S. F., Birkl, R., Wofk, D., Wonka, P., & Müller, M. (2023). Zoedepth: Zero-shot transfer by combining relative and metric depth. https://arxiv.org/abs/2302.12288

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., & Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332. https://doi.org/10.1109/TRO.2016.2624754

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), *Computer vision – eccv 2020* (pp. 213–229). Springer International Publishing.

Catarrinho, D. X. (2026). *Boosting perception: Comparative analysis of transformer, one-stage and two-stage based object detection architectures for real-time object detection on the Booster K1 platform* [Bachelor's thesis]. University of Amsterdam. https://staff.fnwi.uva.nl/a.visser/education/bachelorAI/BSc_Thesis_D%C3%A1rio_Xavier_Catarrinho-2.pdf

Chakrabarty, S. (2026). Yolo26 pose estimation: Real-time keypoint tutorial. https://learnopencv.com/yolo26-pose-estimation-tutorial/

Chen, A., Li, X., He, T., Zhou, J., & Chen, D. (2024). Advancing in rgb-d salient object detection: A survey. *Applied Sciences*, *14*(17). https://doi.org/10.3390/app14178078

Chen, Z., Hu, B.-J., Luo, C., Chen, G., & Zhu, H. (2024). Dense projection fusion for 3d object detection. *Scientific Reports*, *14*(1), 23492.

Ferreri, A., Bucci, S., & Tommasi, T. (2021). Multi-modal rgb-d scene recognition across domains. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2199–2208.

Gao, M., Jiang, J., Zou, G., John, V., & Liu, Z. (2019). Rgb-d-based object recognition using multimodal convolutional neural networks: A survey. *IEEE Access*, *7*, 43110–43136. https://doi.org/10.1109/ACCESS.2019.2907071

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Glorot, X., & Bengio, Y. (2010, 13–15 May). Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh & M. Titterington (Eds.), *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256, Vol. 9). PMLR. https://proceedings.mlr.press/v9/glorot10a.html

Gupta, S., Girshick, R., Arbeláez, P., & Malik, J. (2014). Learning rich features from rgb-d images for object detection and segmentation. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer vision – eccv 2014* (pp. 345–360). Springer International Publishing.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Honkoop, M., de Vries, J., & de Jong, G. (2026). *Team description paper whirlwind amsterdam* (tech. rep.) (RoboCup 2026 Qualification Document). University of Amsterdam. https://staff.fnwi.uva.nl/a.visser/activities/whIRLwind/2026/whIRLwind_RoboCup_2026_qualification_document.pdf

Jocher, G. (2020). *Ultralytics yolov5* (Version 7.0). https://doi.org/10.5281/zenodo.3908559

Jocher, G., Chaurasia, A., & Qiu, J. (2023). *Ultralytics yolov8* (Version 8.0.0). https://github.com/ultralytics/ultralytics

Jocher, G., & Qiu, J. (2024). *Ultralytics yolo11* (Version 11.0.0). https://github.com/ultralytics/ultralytics

Lai, K., Bo, L., Ren, X., & Fox, D. (2011). A large-scale hierarchical multi-view rgb-d object dataset. *2011 IEEE International Conference on Robotics and Automation*, 1817–1824. https://doi.org/10.1109/ICRA.2011.5980382

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

Li, Y., Liu, X., Dong, W., Zhou, H., Bao, H., Zhang, G., Zhang, Y., & Cui, Z. (2022). Deltar: Depth estimation from a light-weight tof sensor and rgb image. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & T. Hassner (Eds.), *Computer vision – eccv 2022* (pp. 619–636). Springer Nature Switzerland.

Li, Y., Ruichek, Y., & Cappelle, C. (2011). 3d triangulation based extrinsic calibration between a stereo vision system and a lidar. *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 797–802. https://doi.org/10.1109/ITSC.2011.6082899

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. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer vision – eccv 2014* (pp. 740–755). Springer International Publishing.

Liu, F., Shen, C., Lin, G., & Reid, I. (2016). Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(10), 2024–2039. https://doi.org/10.1109/TPAMI.2015.2505283

Mahjourian, N., & Nguyen, V. (2025, June). *Multimodal object detection using depth and image data for manufacturing parts* (Vol. Volume 2: Functional Devices/Bioinspired Structures; Sustainability; Semiconductor Manufacturing; Surface Engineering; Clean Energy and E-Mobility Manufacturing; Machining and Deformation Processes; Welding and Joining Processes of Advanced Materials and Structures; Equipment Design, Control and Automation; Human Integration to Smart Manufacturing Systems; Thin Films and Coatings; Meso, Micro, Nano Subtractive and Formative Manufacturing; Explainable AI for Knowledge Discovery). https://doi.org/10.1115/MSEC2025-155144

Maninis, K.-K., Chen, K., Ghosh, S., Karpur, A., Chen, K., Xia, Y., Cao, B., Salz, D., Han, G., Dlabal, J., Gnanapragasam, D., Seyedhosseini, M., Zhou, H., & Araujo, A. (2025). Tips: Text-image pretraining with spatial awareness. In Y. Yue, A. Garg, N. Peng, F. Sha, & R. Yu (Eds.), *International conference on learning representations* (pp. 64256–64279, Vol. 2025). https://proceedings.iclr.cc/paper_files/paper/2025/file/a15a2ece7f0663d1ba7db91103ac61c9-Paper-Conference.pdf

Meng, Z., Kong, X., Meng, L., & Tomiyama, H. (2021). Stereo vision-based depth estimation. In N. N. Chiplunkar & T. Fukao (Eds.), *Advances in artificial intelligence and data engineering* (pp. 1209–1216). Springer Nature Singapore.

Narkhede, M. V., Bartakke, P. P., & Sutaone, M. S. (2022). A review on weight initialization strategies for neural networks. *Artificial intelligence review*, *55*(1), 291–322.

Ophoff, T., Van Beeck, K., & Goedemé, T. (2019). Exploring rgb+depth fusion for real-time object detection. *Sensors*, *19*(4). https://doi.org/10.3390/s19040866

Orfaig, E., Stainvas, I., & Bilik, I. (2026). Rgbx-diffusiondet: A framework for multi-modal rgb-x object detection using diffusiondet. *Pattern Recognition*, *172*, 112460. https://doi.org/https://doi.org/10.1016/j.patcog.2025.112460

Peng, Y., Li, H., Wu, P., Zhang, Y., Sun, X., & Wu, F. (2025). D-fine: Redefine regression task of detrs as fine-grained distribution refinement. In Y. Yue, A. Garg, N. Peng, F. Sha, & R. Yu (Eds.), *International conference on learning representations* (pp. 44015–44031, Vol. 2025). https://proceedings.iclr.cc/paper_files/paper/2025/file/6cf58a87e3097e7d1f9be3e8693a93de-Paper-Conference.pdf

Piccinelli, L., Sakaridis, C., Yang, Y.-H., Segu, M., Li, S., Abbeloos, W., & Van Gool, L. (2026). Unidepthv2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *48*(3), 2354–2367. https://doi.org/10.1109/TPAMI.2025.3628473

Poggi, M., Tosi, F., Batsos, K., Mordohai, P., & Mattoccia, S. (2022). On the synergies between machine learning and binocular stereo for depth estimation from images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(9), 5314–5334. https://doi.org/10.1109/TPAMI.2021.3070917

Rahman, M. A., Das, K., Poovvancheri, J., London, N., & Chen, D. (2026). Rbf weighted hyper-involution for rgb-d object detection. https://arxiv.org/abs/2310.00342

Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., & Koltun, V. (2022). Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(3), 1623–1637. https://doi.org/10.1109/TPAMI.2020.3019967

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 28). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

RoboCup Humanoid Soccer League. (2026). Humanoid soccer league rules. https://github.com/RoboCup-HumanoidSoccerLeague/HSL-Rules/blob/main/Rules.pdf

Sapkota, R., Cheppally, R. H., Sharda, A., & Karkee, M. (2026). Yolo26: Key architectural enhancements and performance benchmarking for real-time object detection. https://arxiv.org/abs/2509.25164

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., & Blake, A. (2011). Real-time human pose recognition in parts from single depth images. *CVPR 2011*, 1297–1304. https://doi.org/10.1109/CVPR.2011.5995316

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Whirlwind Amsterdam. (2025). Humanoid Soccer League (HSL) objects v1 [Hugging Face dataset]. https://huggingface.co/datasets/whirlwind-ams/hsl-objects-v1

Xu, Z., Zhou, H., Peng, S., Lin, H., Guo, H., Shao, J., Yang, P., Yang, Q., Miao, S., He, X., Wang, Y., Wang, Y., Hu, R., Liao, Y., Zhou, X., & Bao, H. (2026). Towards depth foundation models: Recent trends in vision-based depth estimation. *Computational Visual Media*, *12*(2), 243–271. https://doi.org/10.26599/CVM.2025.9450517

Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., & Zhao, H. (2024). Depth anything: Unleashing the power of large-scale unlabeled data. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10371–10381.

Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., & Zhao, H. (2024). Depth anything v2. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, & C. Zhang (Eds.), *Advances in neural information processing systems* (pp. 21875–21911, Vol. 37). Curran Associates, Inc. https://doi.org/10.52202/079017-0688

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, & K. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 27). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/532a2f85b6977104bc93f8580abbb330-Paper.pdf

Zhang, C., Le Moing, G., Koppula, S., Rocco, I., Momeni, L., Xie, J., Sun, S., Sukthankar, R., Barral, J. K., Hadsell, R., Ghahramani, Z., Zisserman, A., Zhang, J., & Sajjadi, M. S. M. (2026). Efficiently reconstructing dynamic scenes one d4rt at a time. *CVPR*.

Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., & Chen, J. (2024). Detrs beat yolos on real-time object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16965–16974.

Zhou, T., Fan, D.-P., Cheng, M.-M., Shen, J., & Shao, L. (2021). Rgb-d salient object detection: A survey. *Computational Visual Media*, *7*(1), 37–69. https://doi.org/10.1007/s41095-020-0199-z