Report on

Coding Assignment

Stereo-Vision Boat Detection, Depth Estimation & Geo-Tracking

Table of contents

Task 3.1

Introduction

The assignment asks to detect vessels (RIB, yachts, catamarans) in two synchronized video streams. The core challenge here is the dataset size. With only 167 frames per camera, training a deep learning model from scratch is hardly possible, moreover free Kaggle resources provide only 30 hours for GPU using.

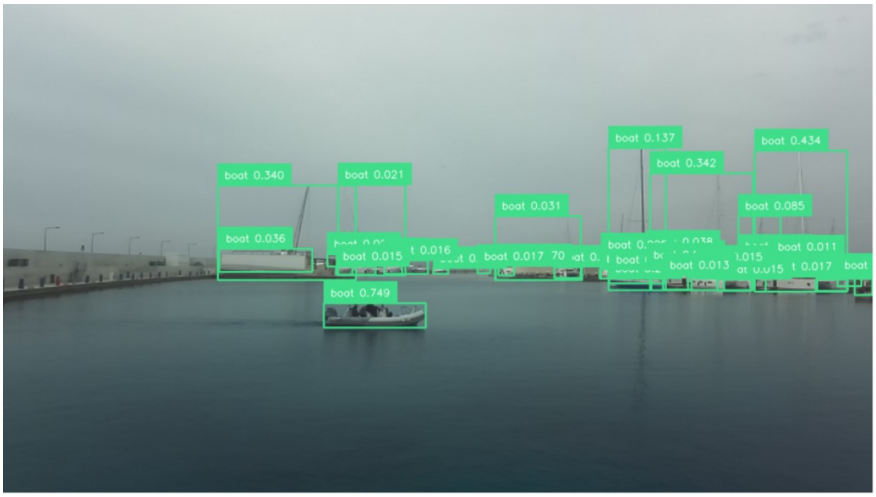
Therefore, the probable strategy could be using the suggested in assignment ways:

- Off-the-shelf use of pretrained detectors (YOLOvXX, Faster-RCNN, etc.).
- Off-the-shelf use of zero-shot / open-vocabulary detection. This is the most modern and flexible approach. It leverages large-scale pre-trained models that can detect objects based on natural language descriptions (e.g., "a boat," "a white yacht").
- Fine-tuning a pre-trained detector. This is a more traditional approach. We would take a detector like YOLOv8, already trained on a large dataset like COCO, and fine-tune it on a public maritime-specific dataset. This can yield high performance but requires finding and curating a suitable external dataset

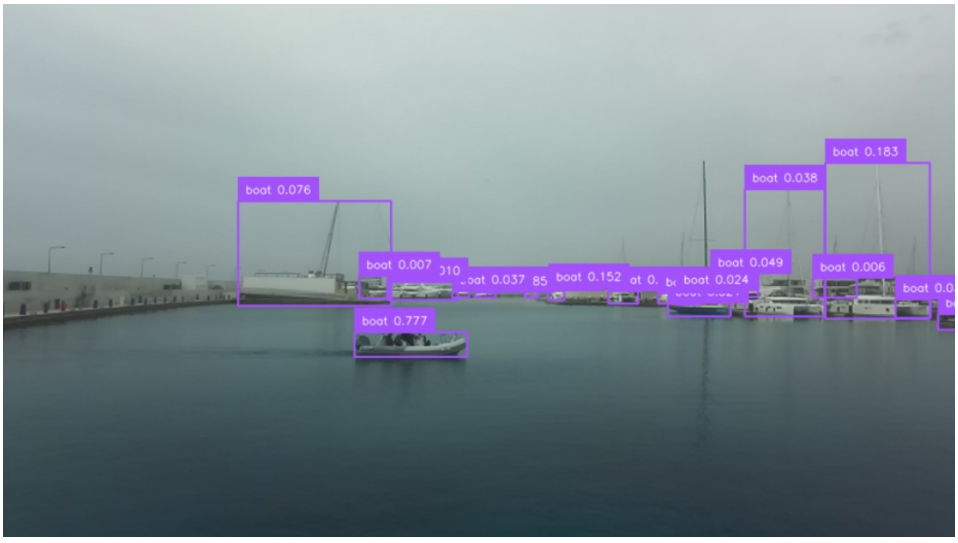
Due to the lack of time we will test only first and second approach

Testing some models

Here are the outcomes of using YOLO12x model and YOLOWORLD:



YOLO12 Outcomes: TARGET_CLASS = 'boat', TARGET_CLASS_ID = 8 # COCO ID for 'boat', CONFIDENCE_VALUE = 0.01, IoU_VALUE = 0.25



YOLOWORLD Outcomes: CONFIDENCE_VALUE = 0.005, IoU_VALUE = 0.05, classes = ["boat"]

So, there are several models, which can be used for the detection task and we need to choose the best one for existing environment:

- YOLO8x
- YOLO8l
- YOLO12x
- YOLO12l
- YOLO8x-World-v2
- YOLO8l-World-v2
- YOLO8x-World
- YOLO8l-World
- RT-DETR-X
- RT-DETR-L
- YOLO-NAS-L
- YOLO-NAS-M

Comparison models

As a next step we should select metrics in order to compare models with each other. Models can be evaluated and examined across three pillars:

1. Qualitative visual analysis: a structured human-in-the-loop review.
2. Quantitative proxy metrics: metrics that don't require labels but can indicate model behavior and stability.
3. Semi-supervised evaluation (the "gold standard" method): creating a small, targeted test set to enable true metric calculation.

Metrics for comparison

Qualitative visual analysis

It directly answers "Does this look right?" and helps to spot subtle but critical failure modes that proxy metrics can't capture.

Quantitative proxy metrics

These are numerical measures one can calculate without labels. They are imperfect but can reveal trends. Such metrics provide objective numbers that can complement qualitative review and are easy to plot and compare. Possible proxy metrics are:

- detections count per frame,
- average confidence of detections per frame,
- histogram of all detection confidences,
- detection count distribution.

Average confidence of detections can tell that some model that is consistently more "certain" will have a higher average confidence. At the same time a bad model could be "confidently wrong.", if a model has high confidence and it is visually confirmed that its detections are good, that could be a strong positive signal.

Number of detections per frame could be an indicator of stability and false positives. The number of static moored boats should be constant. The number of total boats should only change by one as the RIB moves out from field of view. A model whose detection count spikes up and down erratically is likely unstable or generating many false positives. The most stable line is likely the best model.

Confidence histogram: one can aggregate all detection confidences from all frames and plot a histogram. A good model often shows a bimodal distribution: a peak of high-confidence detections (the things it's sure about) and another peak at low-confidence (the "maybe" detections). A model with all its confidences clustered in the middle (e.g., 0.4-0.6) may be uncalibrated or uncertain.

Warning about Proxy Metrics: Never rely on a single proxy metric. A model can easily "game" them. For example, a model that only outputs one perfect detection with 99% confidence and misses everything else would have a fantastic average confidence but terrible recall. You must use them as a suite of indicators.

Semi-Supervised Evaluation

This is the most robust and defensible approach. It bridges the gap between unlabeled and fully labeled evaluation, but the only way to calculate true performance metrics like mAP50, mAP75, mAP5090, F1-score, Precision and Recall is creating test set with labeled data.

At first, to understand the metrics, we need to define a "correct" detection. This is determined by the Intersection over Union (IoU) threshold.

Intersection over Union (IoU): A value from 0 to 1 that measures how much a predicted bounding box overlaps with a ground-truth bounding box. An IoU of 1 means a perfect match.

True Positive (TP): A correct detection. The model predicts a box that has an IoU with a ground-truth box above a certain threshold (e.g., > 0.5) and is of the correct class.

False Positive (FP): An incorrect detection. A predicted box that either has an IoU below the threshold with any ground-truth box or is a duplicate detection of an already-detected object.

False Negative (FN): A missed detection. A ground-truth object that was not detected by the model.

Now we can describe Precision, Recall and F1-Score:

- Precision = TP / (TP + FP): the fraction of model's positive predictions that were actually correct. It answers the question: "When model predicts a boat, how often is it right?" A high precision score (e.g., 0.95) means the model has a low false positive rate; it rarely makes mistakes.This is a measure of the quality of the detections.
- Recall = TP / (TP + FN): the fraction of all actual positive instances in the dataset that model successfully identified. It answers the question: "Of all the actual boats in the scene, what percentage did model find?" A high recall score (e.g., 0.95) means the model has a low false negative rate; it rarely misses objects. This is a measure of the completeness or sensitivity of the detections.
- F1-score = 2 * (Precision * Recall) / (Precision + Recall): the harmonic mean of Precision and Recall, providing a single score that balances both metrics. The F1-Score is useful when one need to find an optimal blend of precision and recall. It punishes models that are extremely good at one metric at the great expense of the other. It provides a single, balanced measure of a model's overall accuracy.

mAP metrics

There are also advanced mAP metrics (the COCO standard): the mean average precision (mAP) metrics are the primary metrics for object detection challenges. They provide a comprehensive evaluation of a model's performance across all classes and various confidence levels:

mAP50 (or mAP@0.5): the mean Average Precision calculated with the IoU threshold set to a lenient 0.50. This metric measures how well the model can correctly classify and roughly locate objects. Because the IoU threshold is low, it does not heavily penalize small inaccuracies in bounding box placement. It is a good indicator of the model's ability to identify objects correctly, even if the boxes aren't perfectly tight.

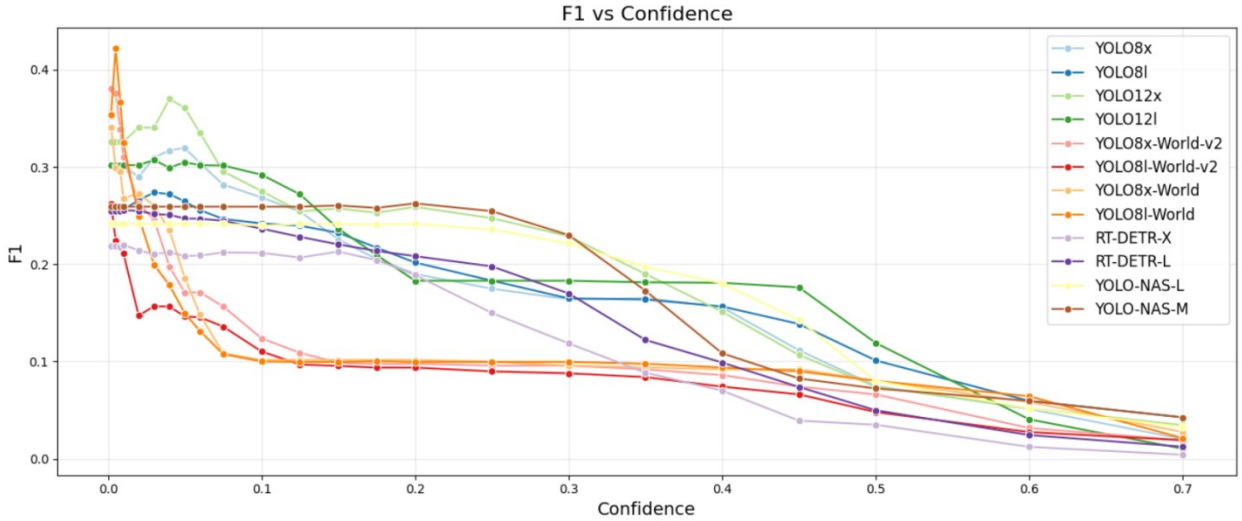
mAP75 (or mAP@0.75): the mean Average Precision calculated with the IoU threshold set to a strict 0.75. It evaluates the model's ability to place bounding boxes with high accuracy. To get a

high score here, a model's predicted boxes must overlap significantly with the ground truth. It is a strong indicator of the model's localization precision.

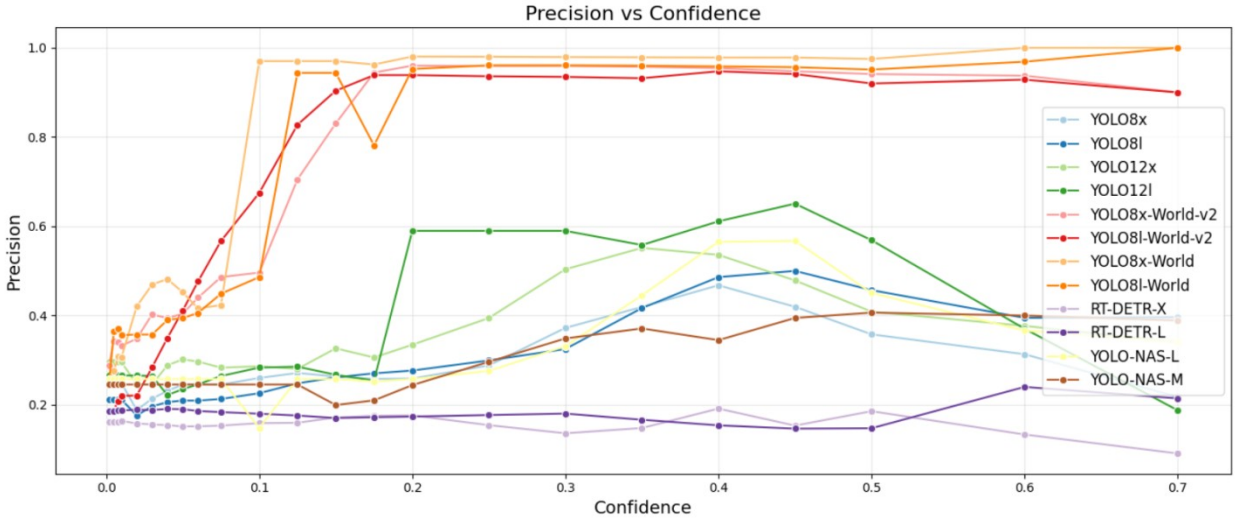
mAP50-95 (The Primary COCO Metric): the mean Average Precision calculated over a range of 10 different IoU thresholds, from 0.50 to 0.95 with a step size of 0.05 (i.e., mAP@0.5, mAP@0.55, ..., mAP@0.95). The final score is the average of these 10 mAP values. This is the most comprehensive and demanding metric. It rewards models that perform well across a wide spectrum of localization requirements, from lenient to pixel-perfect. It is considered the single best overall metric for judging both the classification and localization quality of a detection model. A high mAP50-95 score indicates a robust, high-quality model.

Comparison models on validation dataset with 55 annotated images

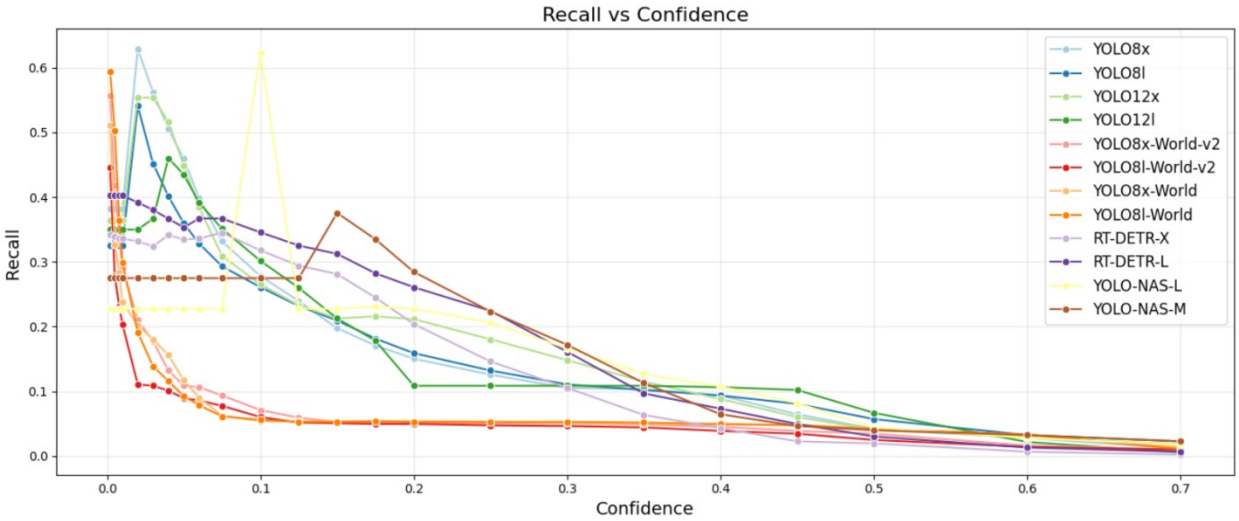
I used Roboflow to annotate at first 55 images and examined models using classical detection metrics. The following results were obtained.



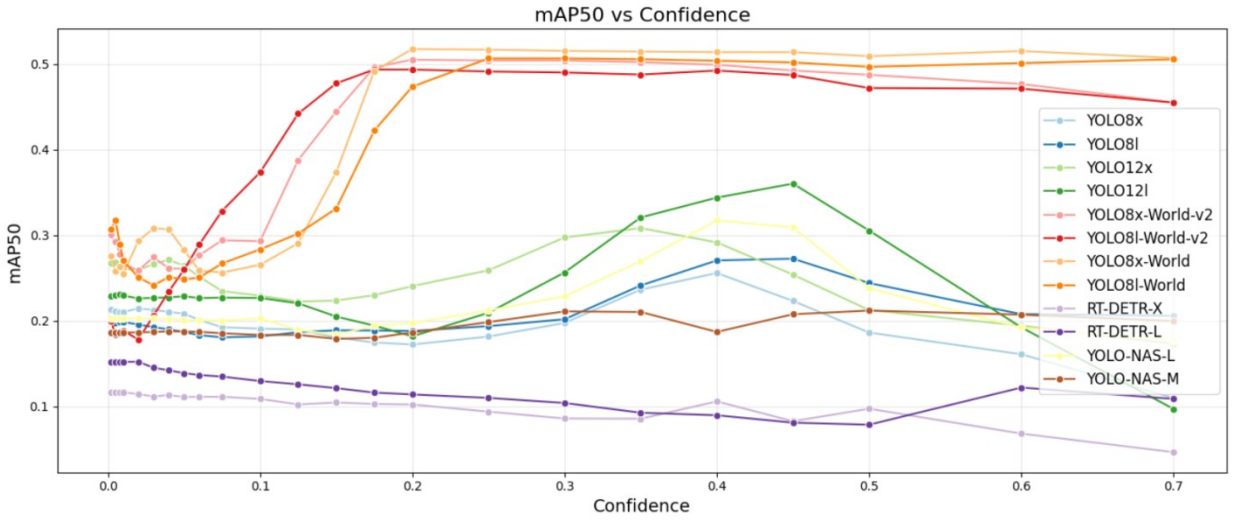
The dependence of F1-score on confidence threshold



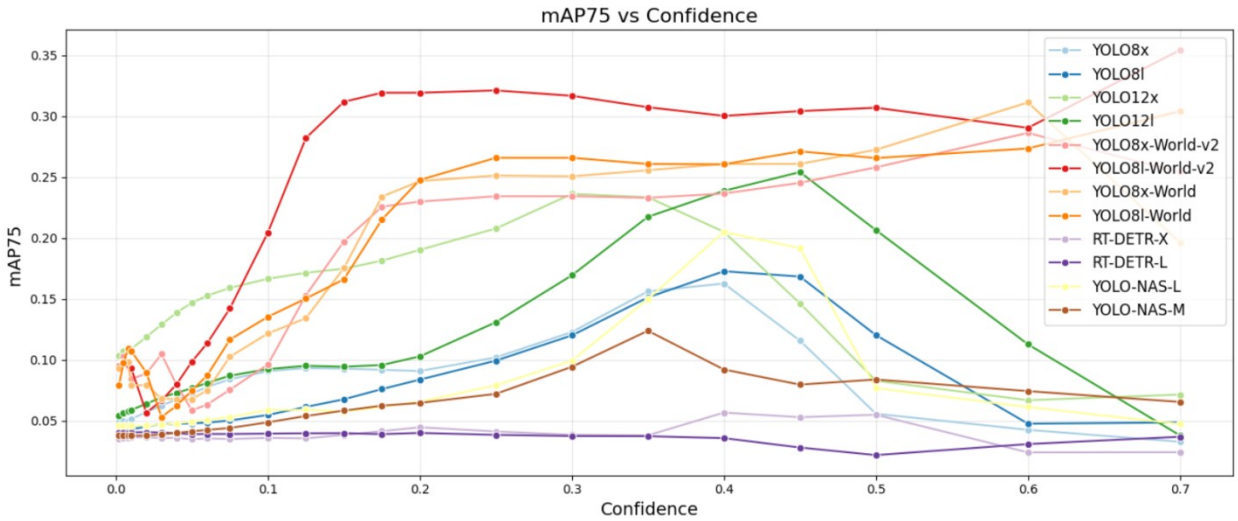
The dependence of Precision on confidence threshold



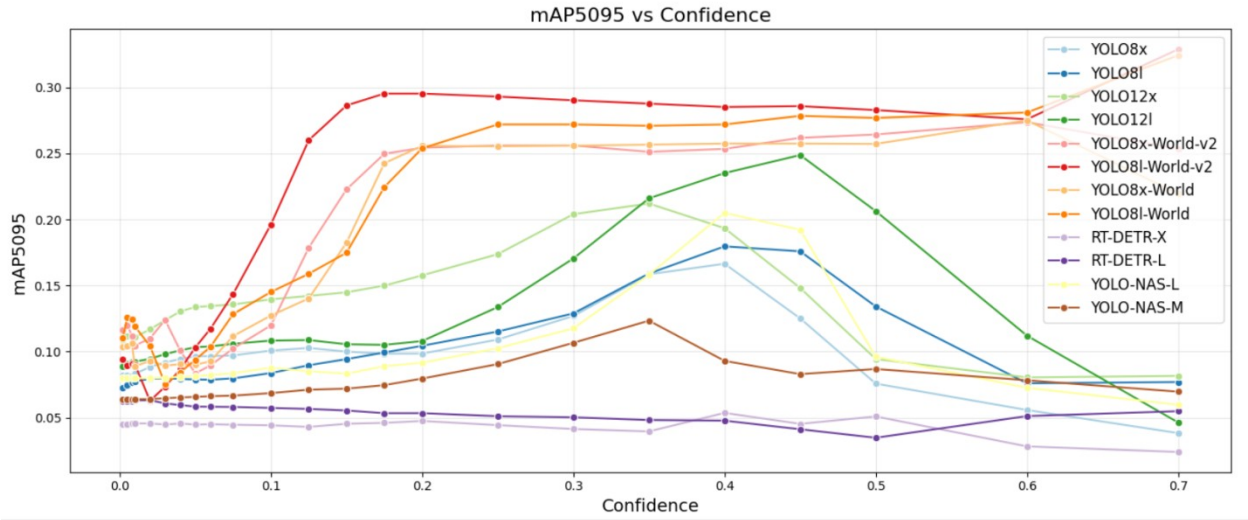
The dependence of Recall on confidence threshold



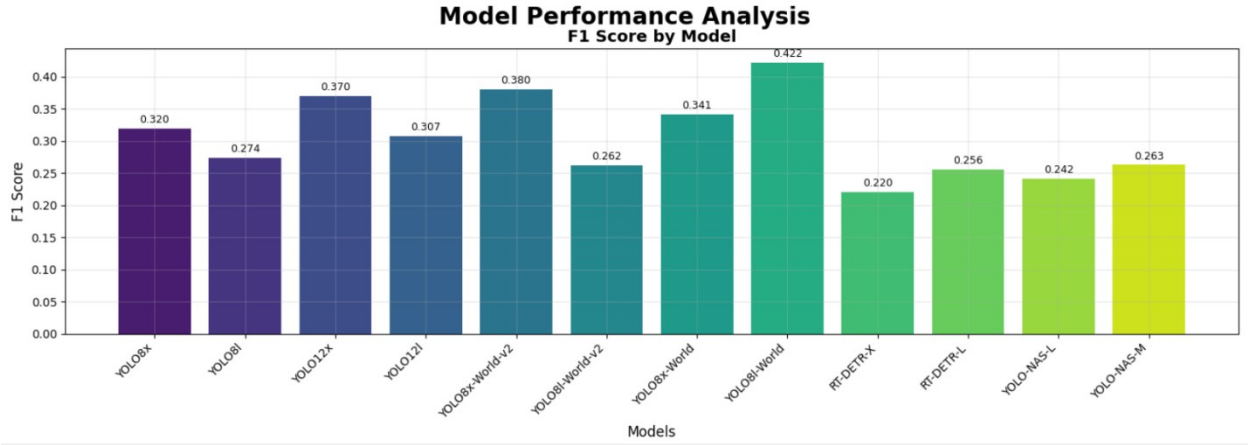
The dependence of mAP50 on confidence threshold



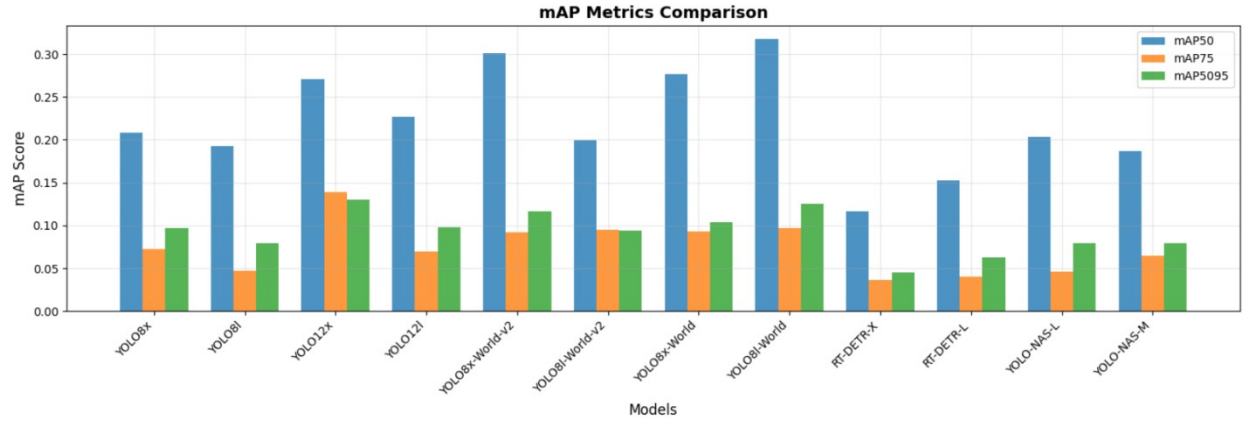
The dependence of mAP75 on confidence threshold



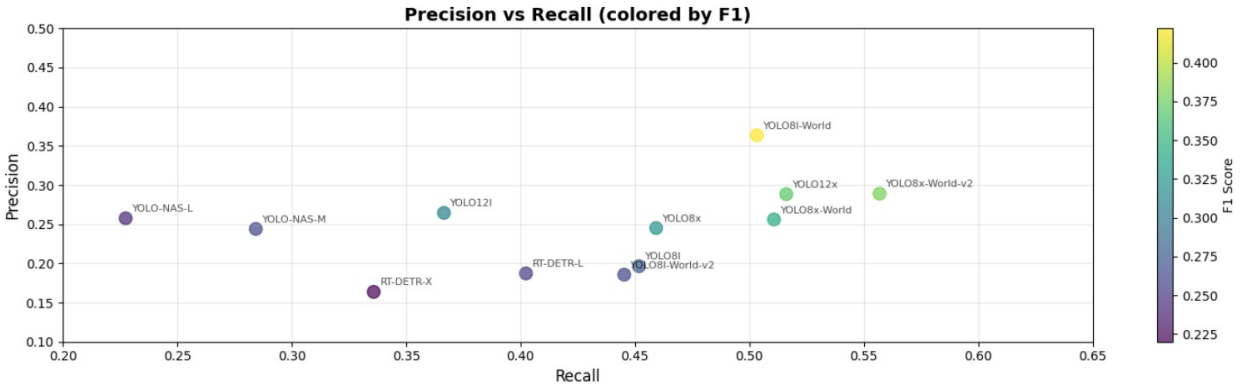
The dependence of mAP5095 on confidence threshold



Model performance analysis



mAP metrics comparison



Precison vs Recal for various models

Based on an evaluation across multiple metrics on the 55-image validation set, YOLO12x, YOLO8l-World, YOLO8x-World, YOLO8x-World-v2 are the probably the top models for detection task in existing environment. These models achieve the highest overall performance, demonstrating the best balance of precision and recall. Furthermore, they exhibit robust performance across all mAP thresholds and maintains their superiority over a wide range of confidence settings. The optimal operating point for these models appears to be at a confidence threshold in-between 0 and 0.20, which maximizes its mAP and F1 performance.

Precision vs. Recall visualizes the trade-offs: the ideal model is in the top-right corner. The mentioned models are positioned furthest towards this corner, confirming their great balance. They achieve a high recall without sacrificing as much precision as other models.

mAP metrics show the following. YOLO8l, YOLO8x, YOLO8x-World-v2 and YOLO8l-World-v2 lead with the highest mAP50 score. This means they are the best models at simply finding and identifying the correct objects, even with less-than-perfect bounding boxes. The gap between models at mAP75 shrinks at this stricter IoU threshold, and YOLO8l-World-v2 is the top performer. This shows it is not only good at finding objects but also at localizing them accurately. In mAP50-95 YOLO8l-World-v2 has the highest overall COCO metric, proving its status as the most robust model across all levels of localization difficulty.

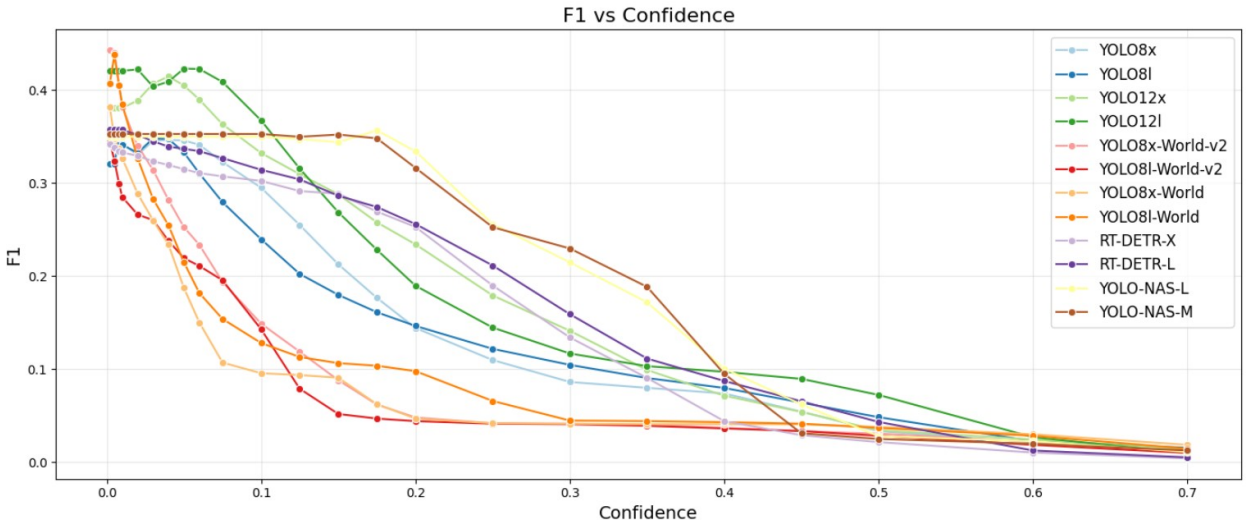
Analysis of line charts shows that across all three mAP plots (mAP50, mAP75, mAP50-95), the lines representing YOLO8l, YOLO8x, YOLO8x-World-v2 and YOLO8l-World-v2 consistently form the highest peaks. These peaks occur at a confidence threshold between 0.1 and 0.25. Selecting a confidence value in this range will yield the model's maximum possible accuracy on this dataset. Setting the confidence too low (<0.1) or too high (>0.4) results in a significant performance degradation.

As expected, precision for all models generally increases with the confidence threshold. The World models demonstrate a remarkable ability to achieve very high precision (>0.9) at moderate confidence levels, indicating that when they are confident, they are very rarely wrong. Conversely, recall drops as confidence increases. At a very low confidence, models find almost every object, but as the threshold becomes stricter, models begin to miss more challenging ones. The key is to find the balance.

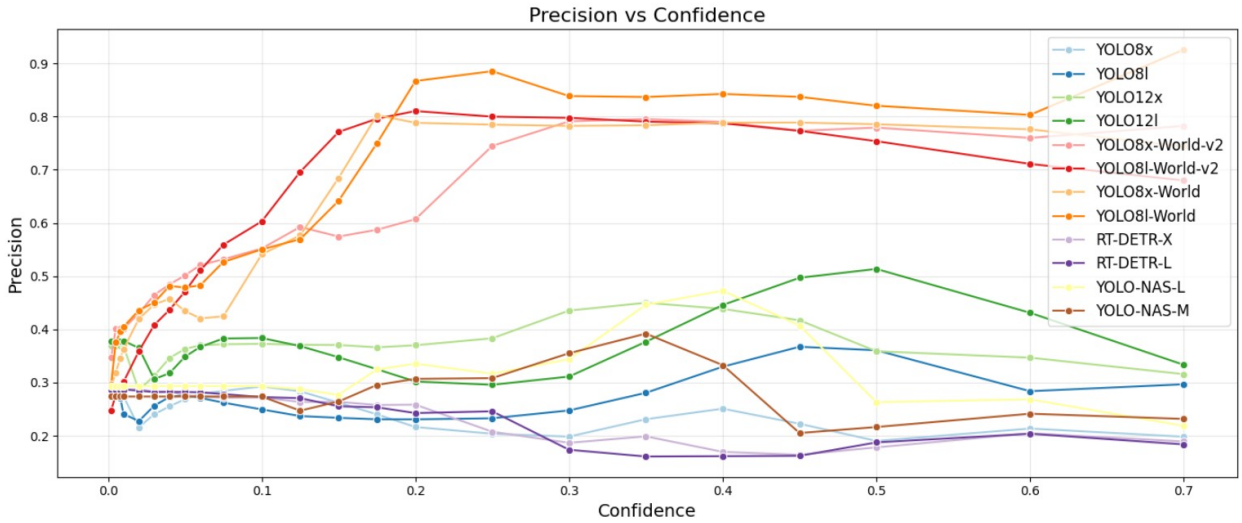
The F1 vs. Confidence plot synthesizes the precision-recall trade-off. Peaks of the F1 curves for selected models occur squarely around a confidence of <0.05, where the initial spikes are. However, a more stable high-performance plateau is achieved around 0.1-0.2, confirming this as the optimal threshold for balanced performance.

Comparison models on validation dataset with 200 annotated images

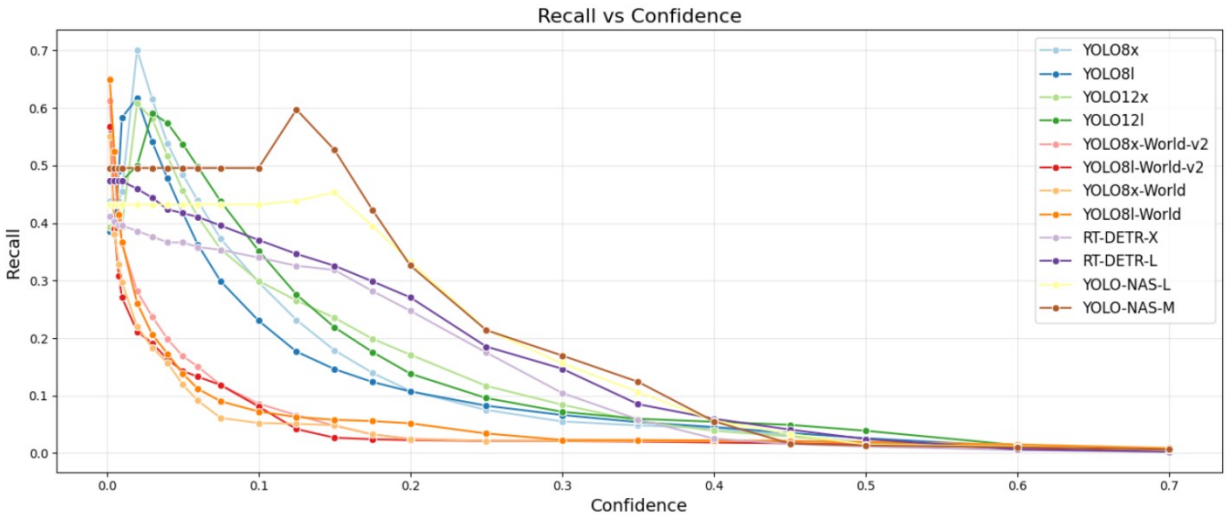
In order to prove the selection of model for generation final predictions the bigger annotated dataset was prepared. The obtained outcomes are the following.



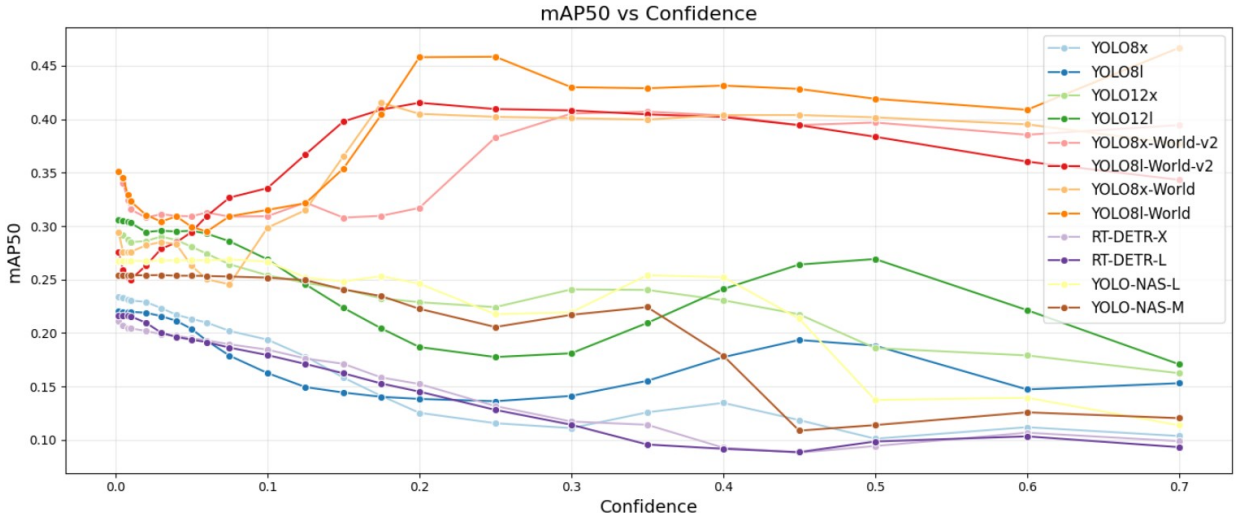
The dependence of F1-score on confidence threshold



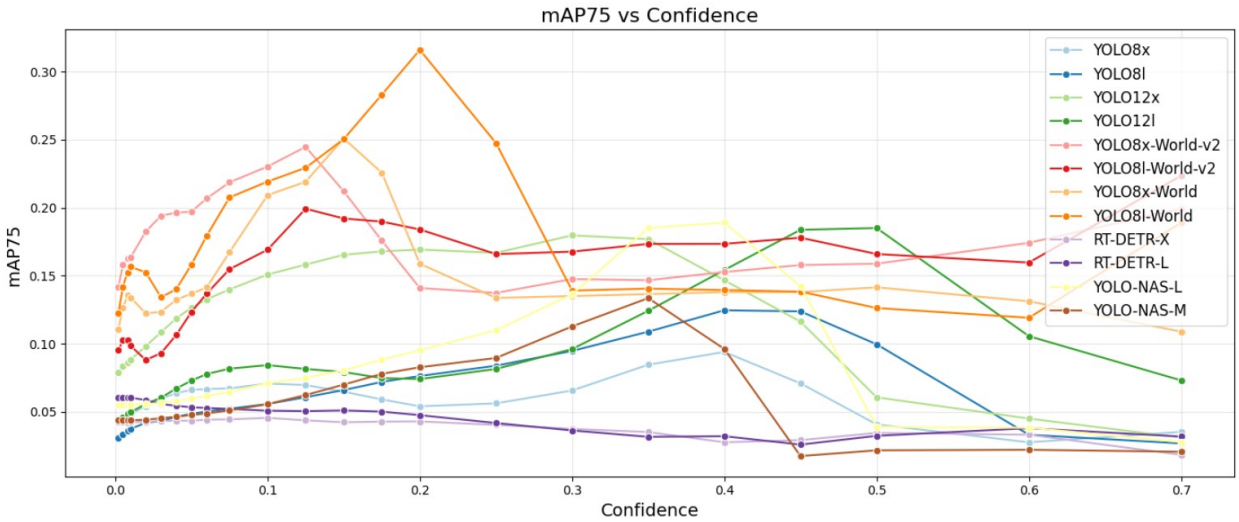
The dependence of Precision on confidence threshold



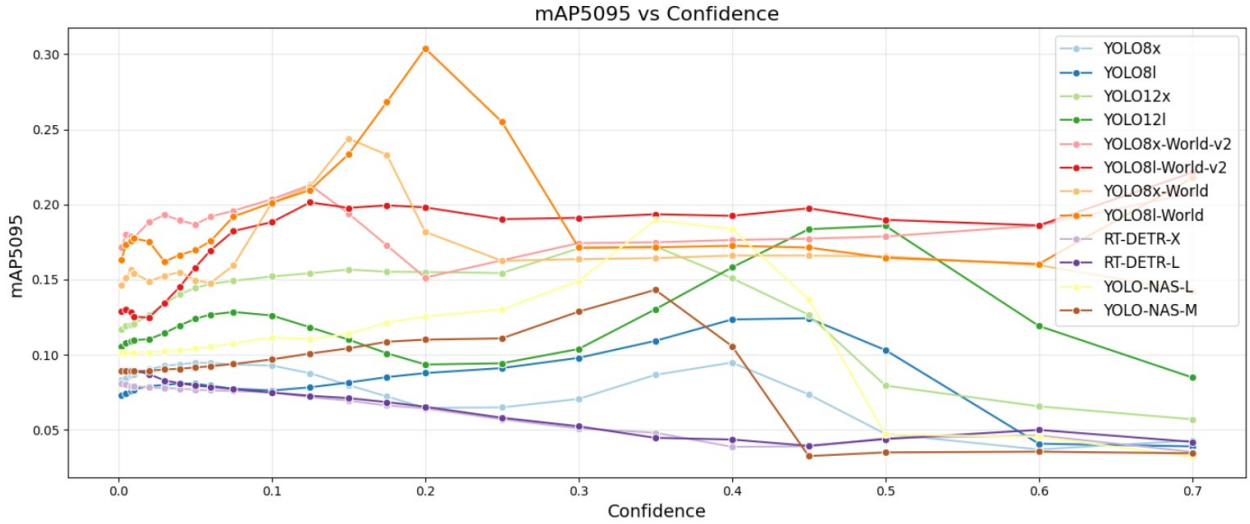
The dependence of Recall on confidence threshold



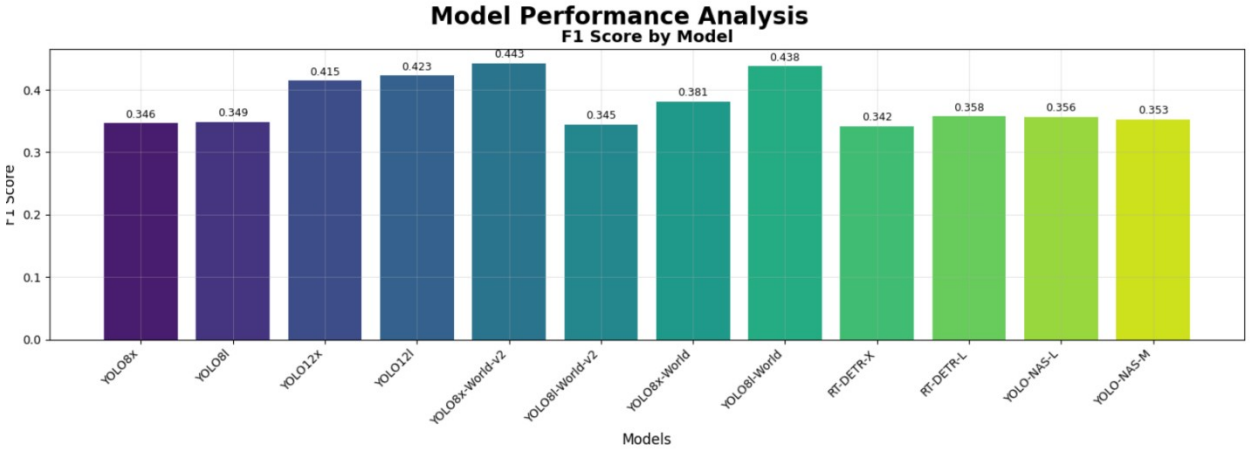
The dependence of mAP50 on confidence threshold



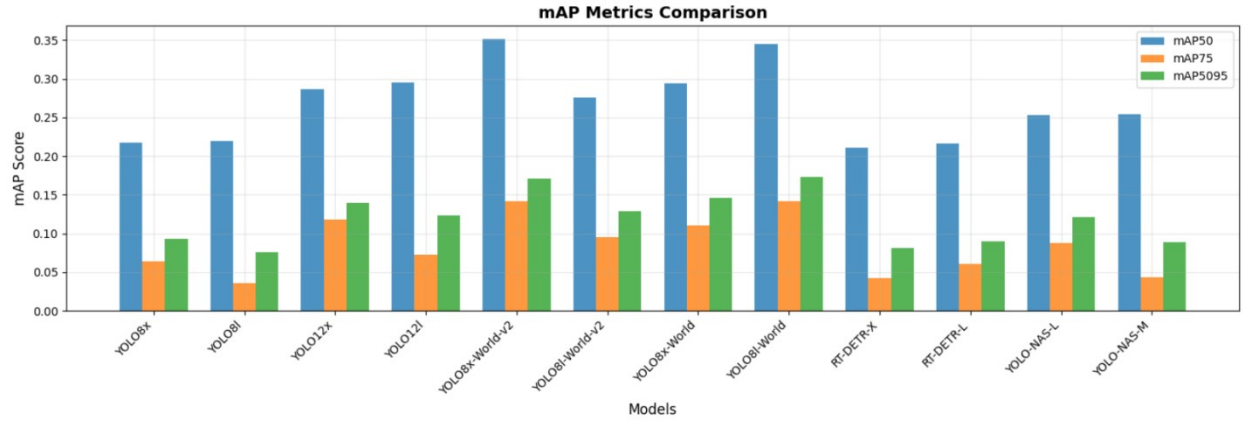
The dependence of mAP75 on confidence threshold



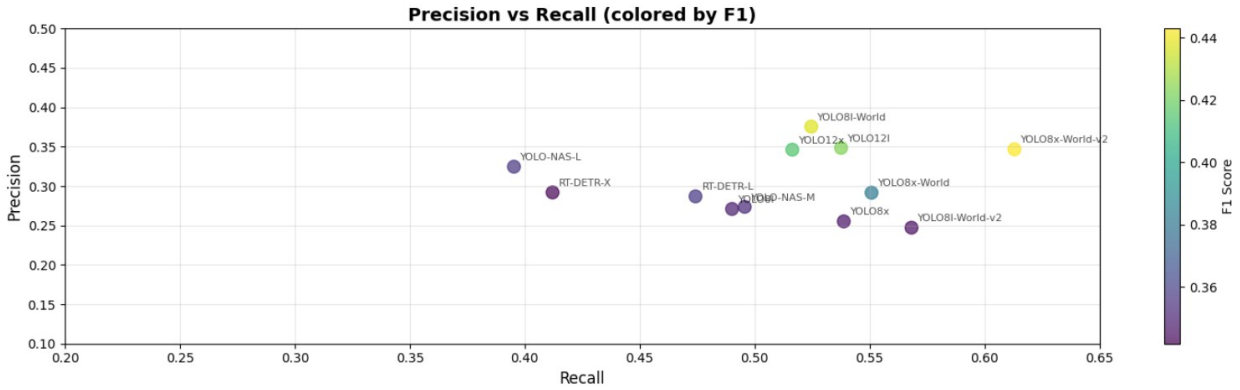
The dependence of mAP5095 on confidence threshold



Model performance analysis



mAP metrics comparison

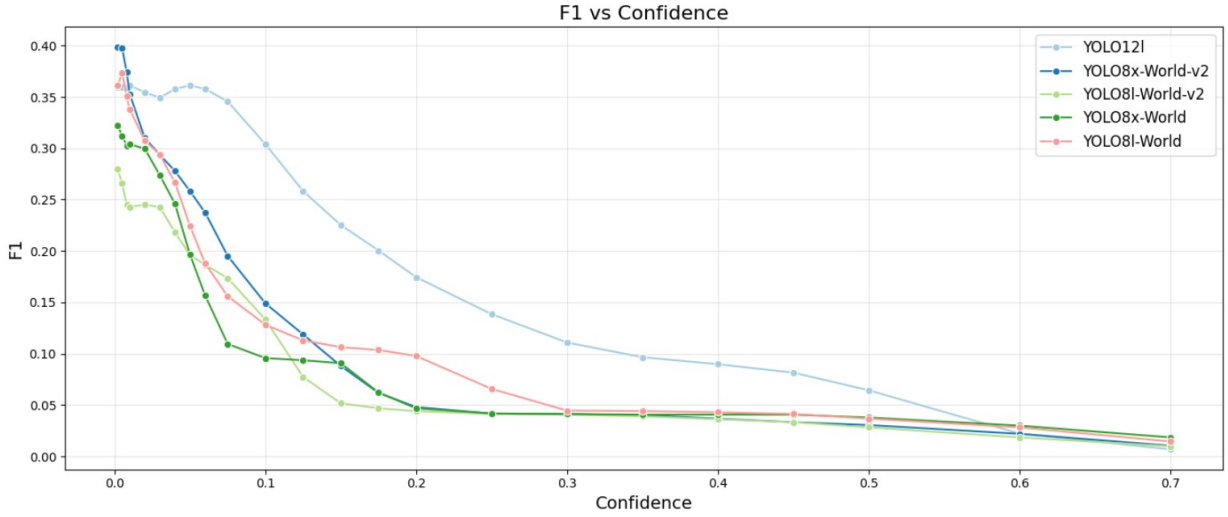


Precison vs Recal for various models

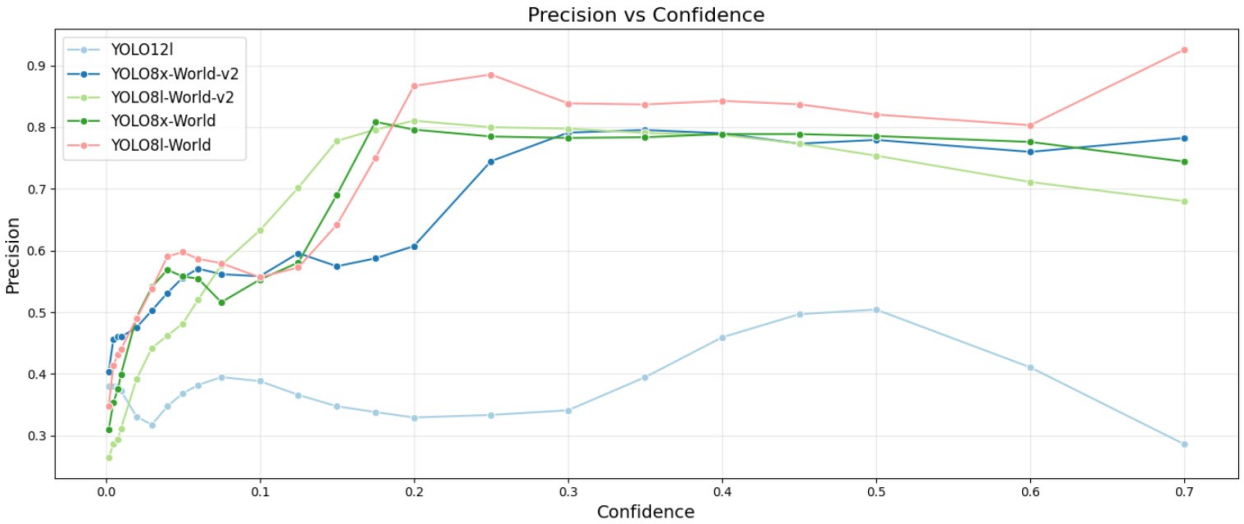
It is clear now that the top model is YOLO8x-World-v2. It has the top performance on mAP metrics [especially in low confidence regions (< 0.1)] as well as F1-score.

Comparison top-models on validation dataset with 200 annotated images and low value of IoU

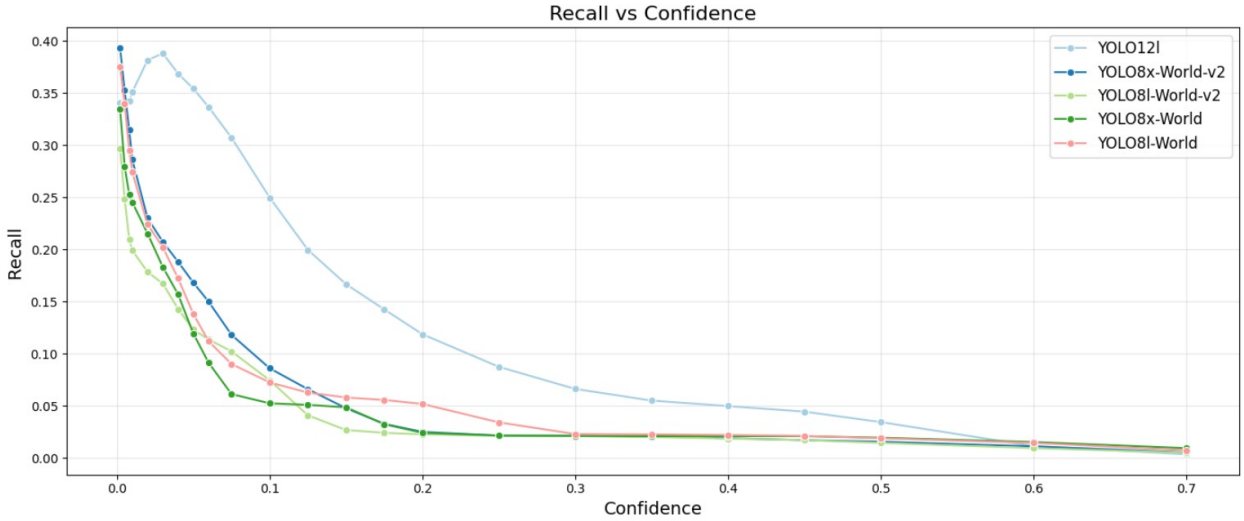
In order to double-check the performance of top-models the additional test with low value of IoU = 0.05 was carried out. The results are the following.



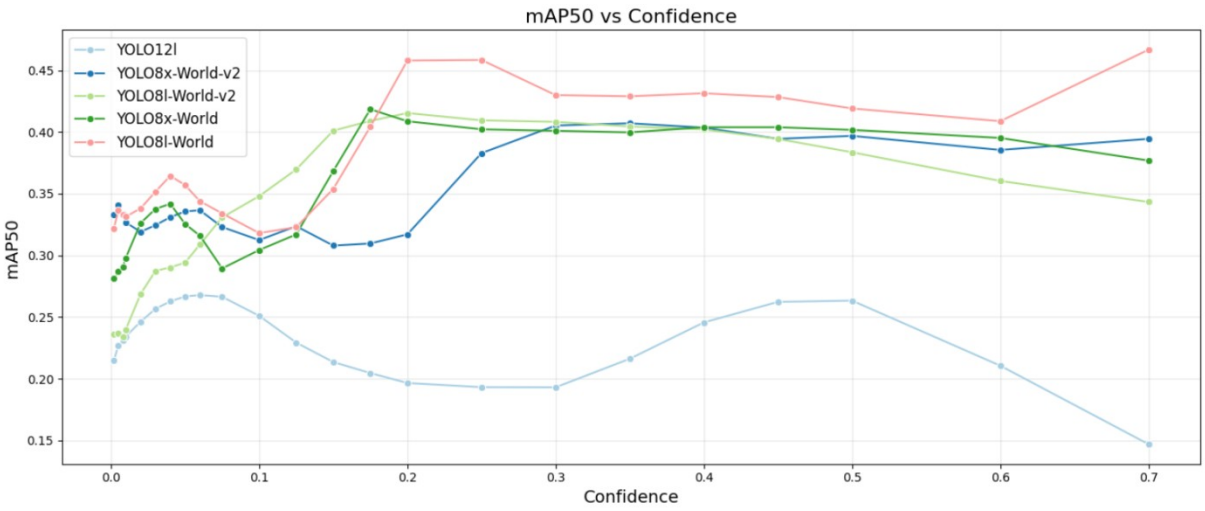
The dependence of F1-score on confidence threshold



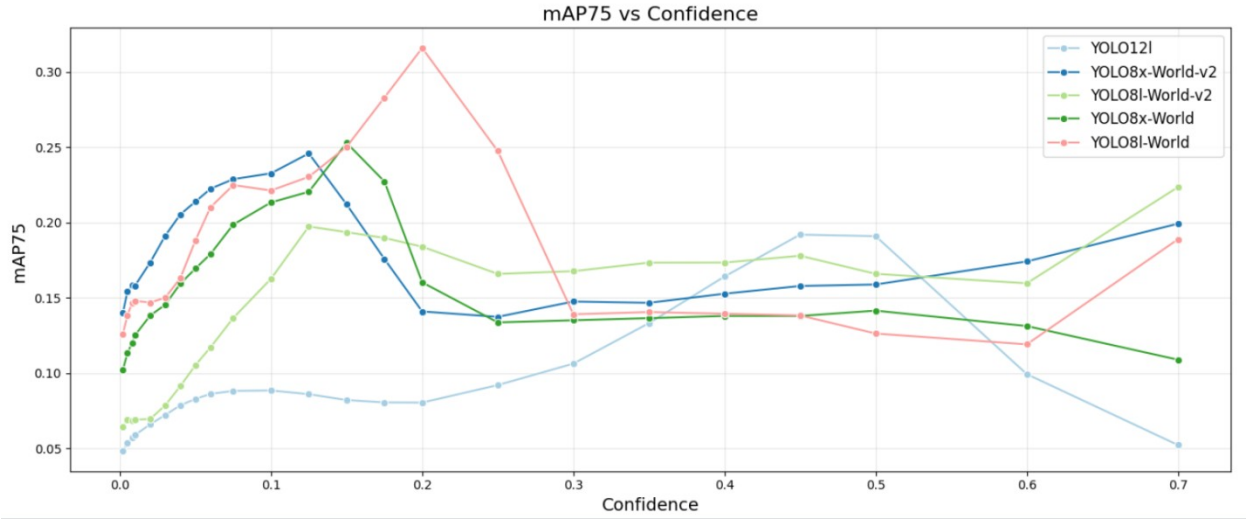
The dependence of Precision on confidence threshold



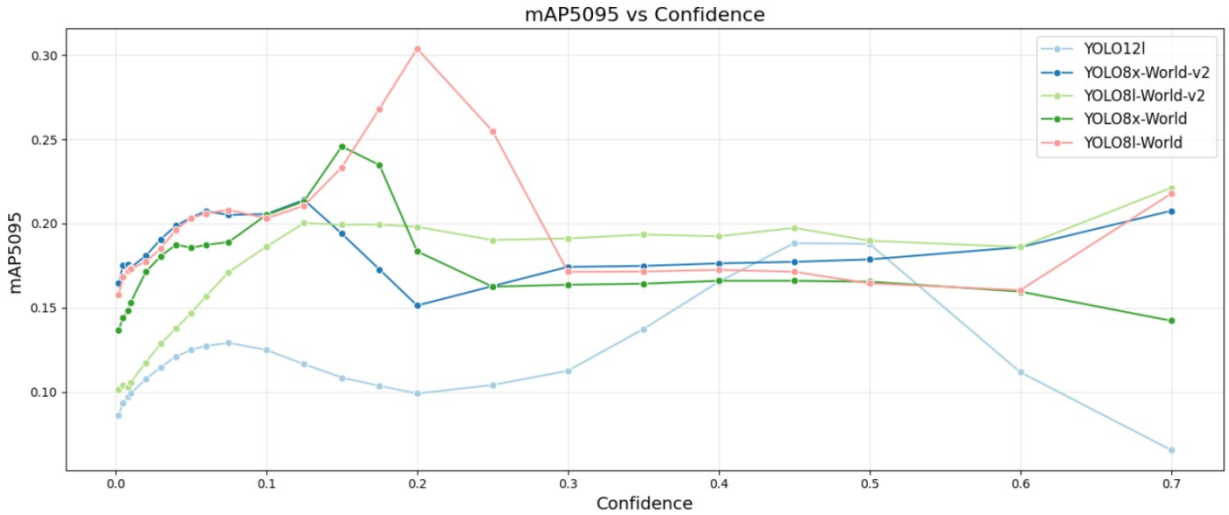
The dependence of Recall on confidence threshold



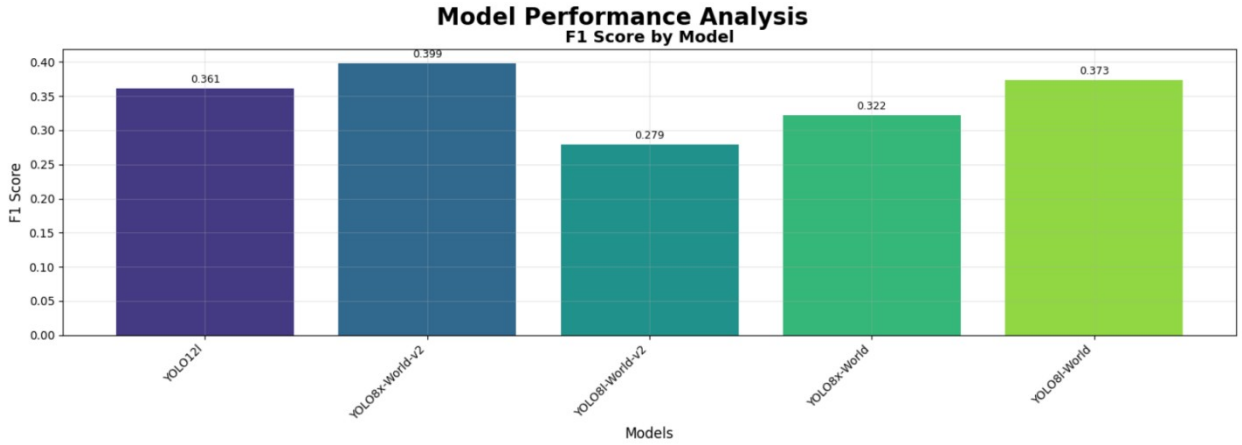
The dependence of mAP50 on confidence threshold



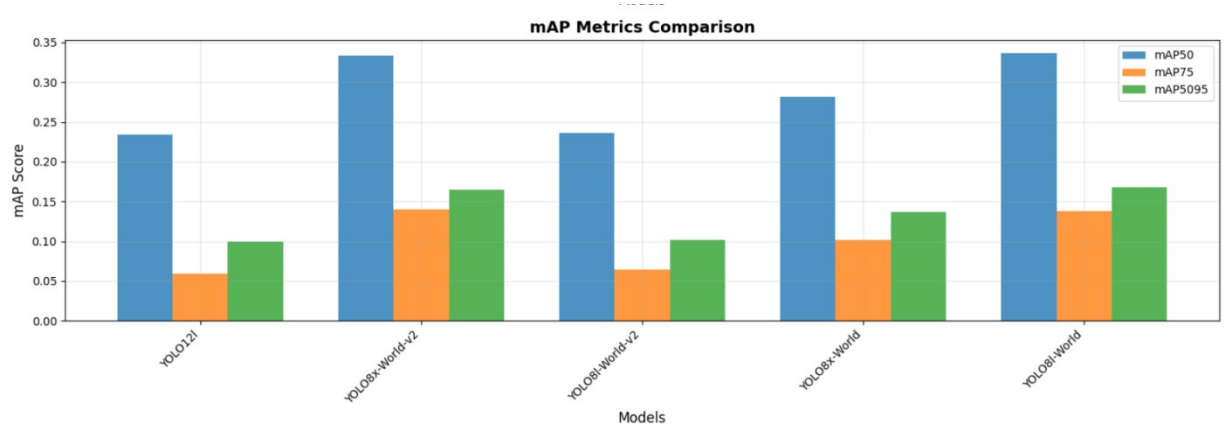
The dependence of mAP75 on confidence threshold



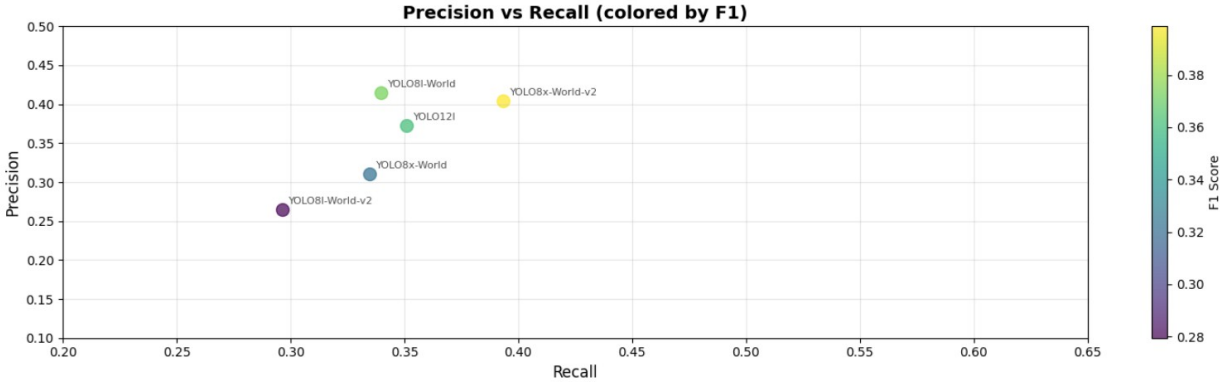
The dependence of mAP5095 on confidence threshold



Model performance analysis



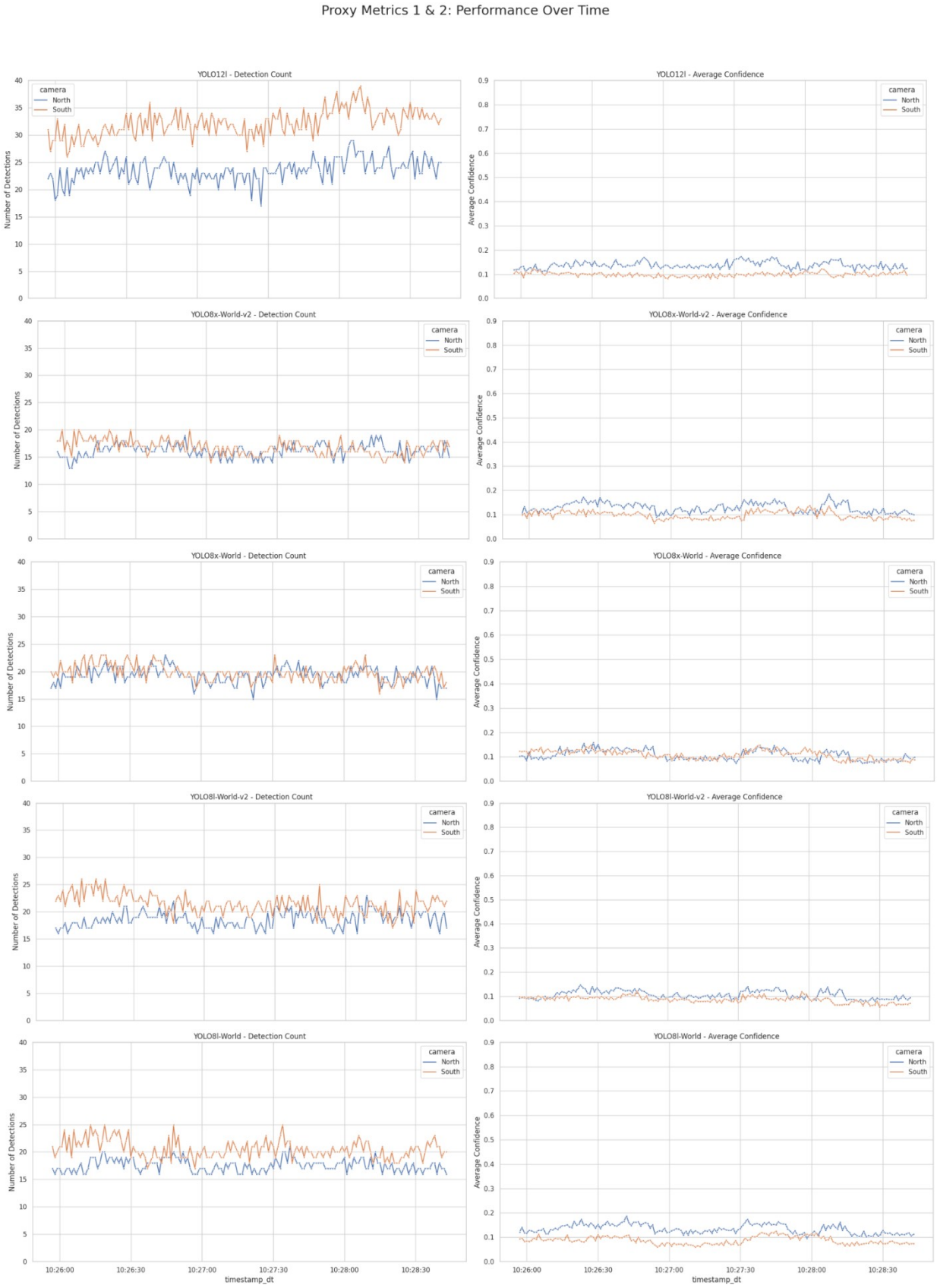
mAP metrics comparison



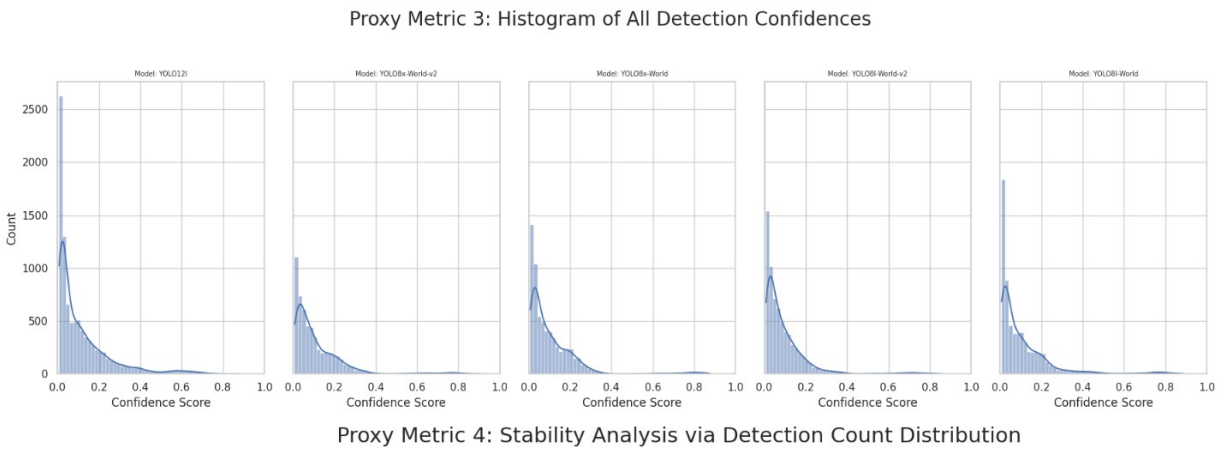
Precison vs Recal for various models

Final evaluation of top-performed models

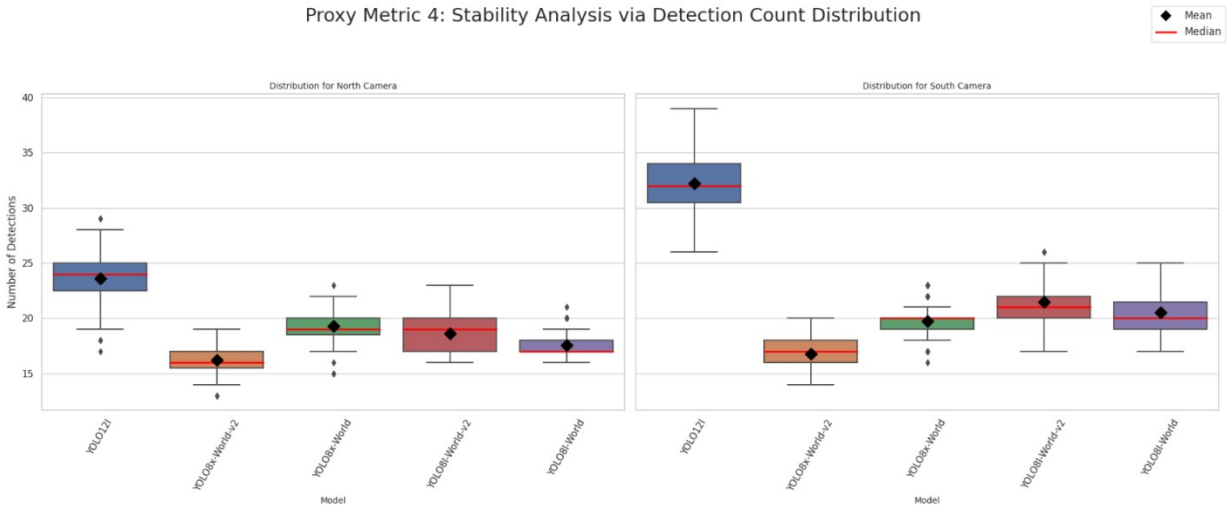
The proxy metrics for selected top model YOLO8x-World-v2 and comparisons with other top-performed models are shown further.



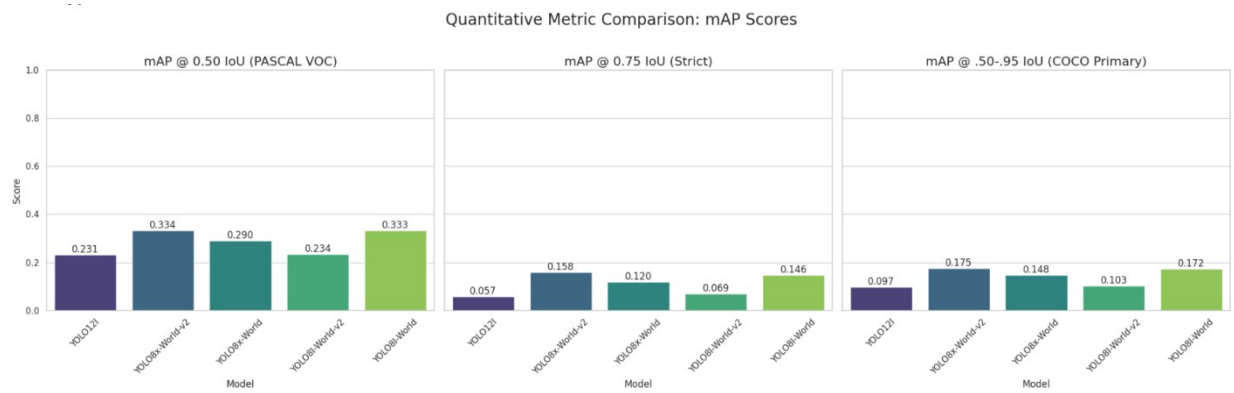
Proxy metrics for top-performed models: Detection count and average confidence per frame



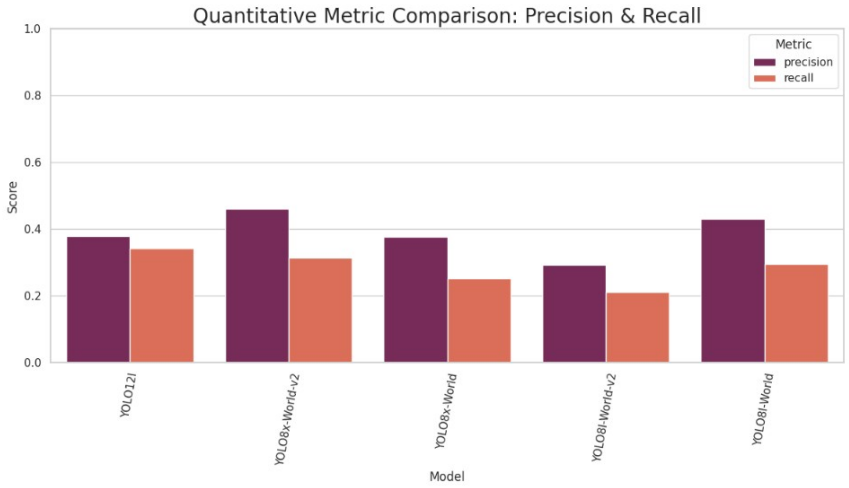
Statistics on the detection counts for top-performed models



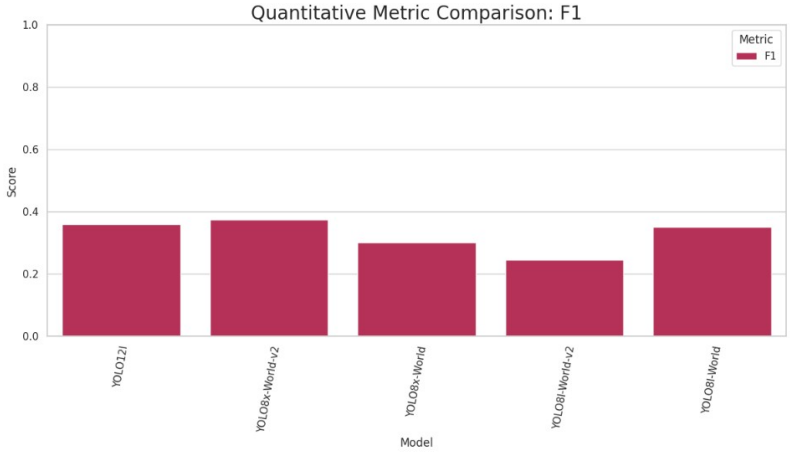
Statistics on the detection counts for top-performed models



mAP metrics for op-performed models



Precision and Recall for top-performed models



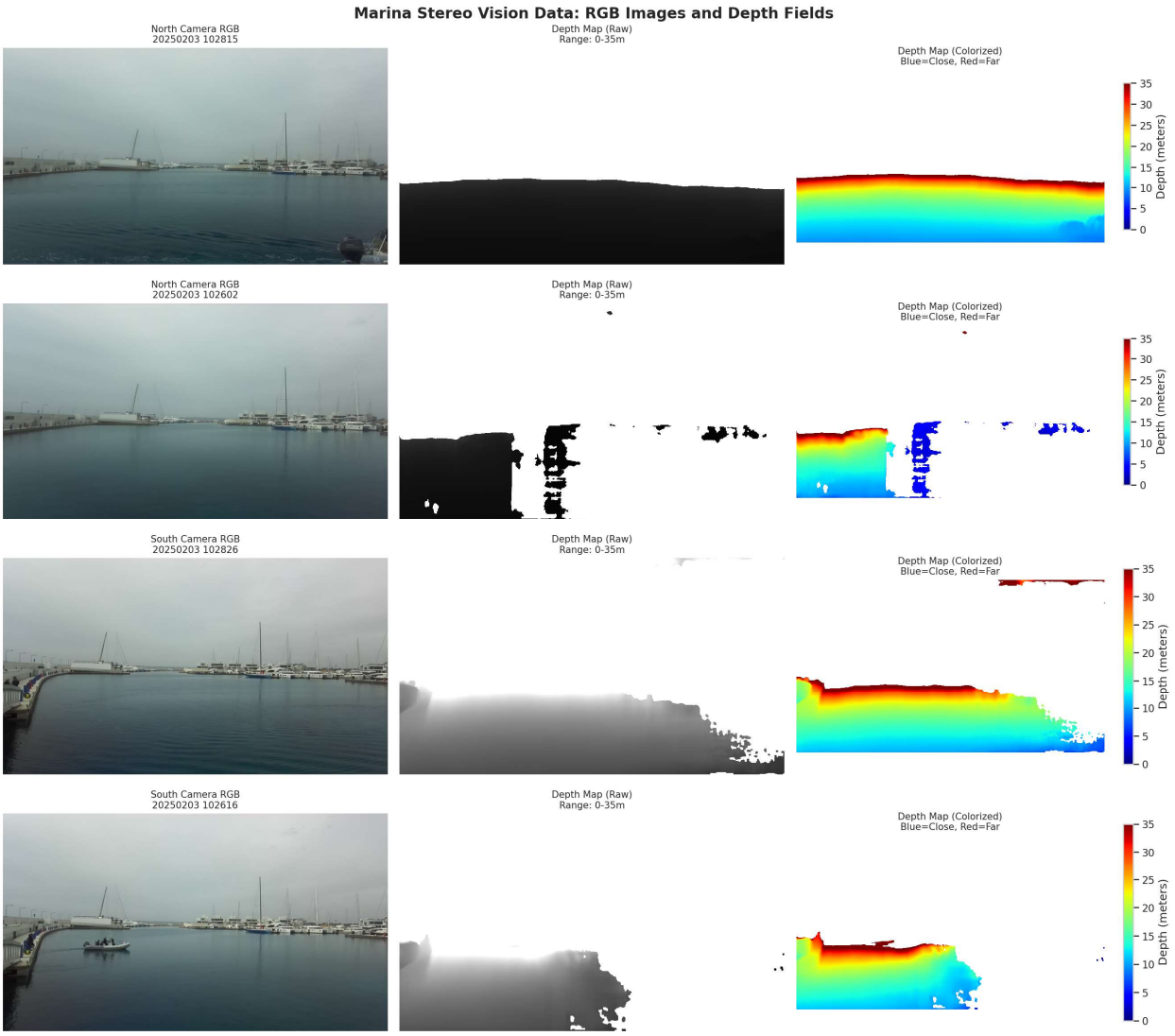
F1-score for top-performed models

YOLO8x-World-v2 model was used to generate the main output: two JSON files containing the detections from each camera (“detections_cam_South.json” and “detections_cam_North.json”).

Task 3.2

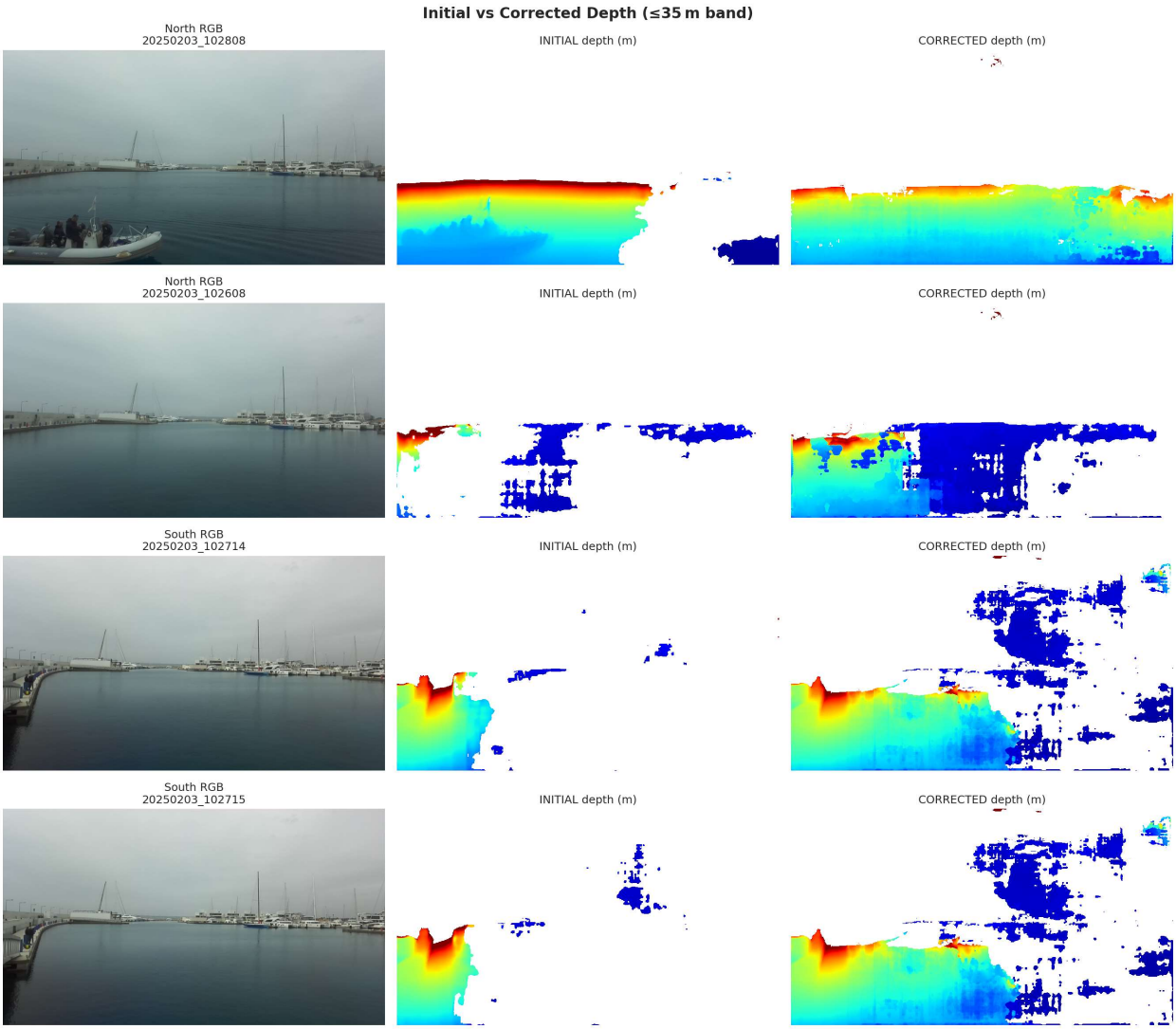
Restoring the depth field by inpainting and temporal smoothing

At first let us see on the depth filed provided by existing “npy” files.



RGB image and its depth filed provided by “npy” data

As it can be seen the regions in depth map with distances < 35 has a lot if NaN values (white regions). There can be a lot of reasons for such behavior: the surface of water has very smooth texture and it is really hard to calculate the distance to such surface, low-contrast regions or specular regions, etc. So the first task is to “repair” the “white” region in ground truth depth field (< 35 meters). It can be done by several approaches. One of the possible approaches comprises spatially inpainting each depth frame (using OpenCV’s Telea inpainting over a NaN mask) and temporal smoothing the resulting volume with an exponential moving average (because we have 167 frames). The results of post-processing depth field data are shown further.

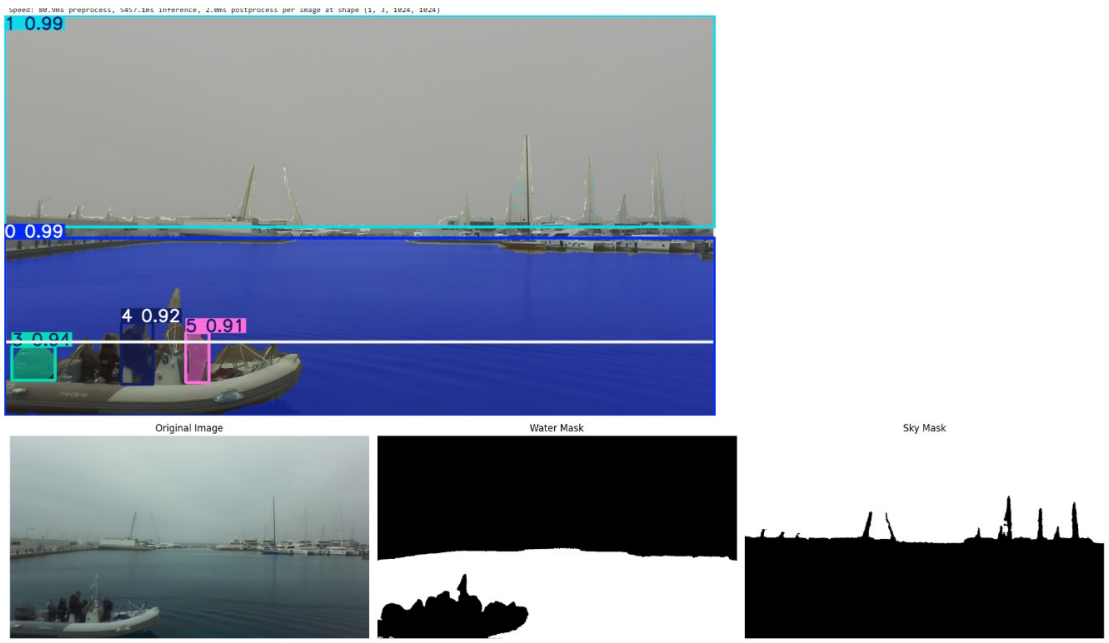


The depth field before post-processing (central) and after post-processing (on the right) using Telea inpainting and temporal smoothing

As it can be seen the depth fields now have less NaN regions.

Segmentation sky and water

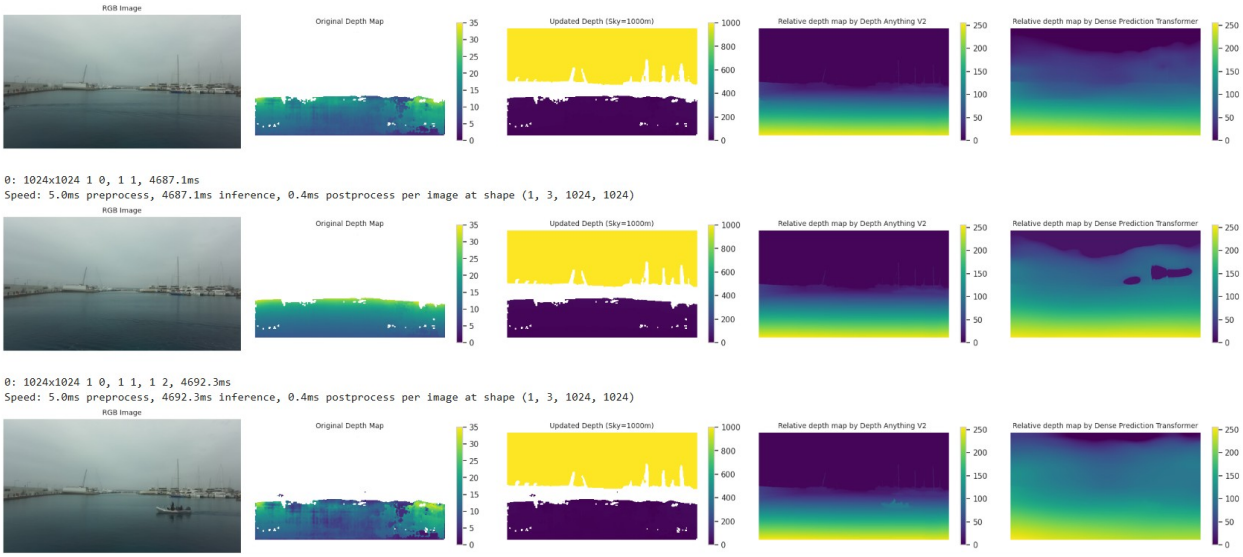
Before we go the next step of generation depth field let us note that there are two main regions on out images – sky and water. So calculation depth field for pixels related to sky region seems unvaluable because we can just assign to the maximum distance, for instance 1000 meters. So in order the reveal pixels related to sky region we can use segmentation. Let us see how it can be done.



Results of segmentation sky and water by YOLO SAM 2: Segment Anything Model 2 (https://docs.ultralytics.com/models/sam-2/)

Prediction relative depth map

So since we can segment sky pixel let us go to the step of prediction the relative filed. We compare two models. The first model is Dense Prediction Transformer (DPT) model trained on 1.4 million images for monocular depth estimation. It was introduced in the paper Vision Transformers for Dense Prediction by Ranftl et al. (2021). DPT uses the Vision Transformer (ViT) as backbone and adds a neck and head on top for monocular depth estimation. The second model is Depth Anything V2 which is trained from 595K synthetic labeled images and more than 62M real unlabeled images. Depth Anything V2 was introduced in the paper of the same name by Lihe Yang et al. in 2024. It uses the same architecture as the original Depth Anything release, but uses synthetic data and a larger capacity teacher model to achieve much finer and robust depth predictions. The original Depth Anything model was introduced in the paper Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data by Lihe Yang et al. Depth Anything V2 leverages the DPT architecture with a DINOv2 backbone. The results of relative depth field prediction are show further.



From left to right: RGM image, restored depth field with Telea inpainting and temporal smoothing, depth field with added mask for sky pixels, predicted by Depth Anything V2 relative depth map, predicted by Dense Prediction Transformer relative depth map

It is clear the Depth Anything V2 work a bit better, so we will use this model.

Calibration of Monocular Depth Estimation using Stereo Ground Truth

The next step is to calibrate provided by Depth Anything V2 model the relative depth map using ground truth data. Monocular Depth Estimation (MDE) models, like Depth Anything V2, are powerful at predicting the relative depth structure of a scene from a single image. However, such models suffer from a fundamental ambiguity: they predict depth in an arbitrary, uncalibrated scale. They can only tell that object A is twice as far as object B, but they cannot tell the absolute distance in meters, millimeters or another units like inches..

The goal of calibration is to resolve this ambiguity by mapping the model's relative depth values to real-world metric units (meters in our case). This is achieved by anchoring the model output to a source of "ground truth" metric depth.

The standard method assumes a simple linear relationship (scale and shift) between the relative depth (d_relative) and the true metric depth (d_metric):

d_metric = **s** * **d_relative** + **t**

where:

s is the scale factor.

t is the shift or offset.

So the task is to find the optimal **s** and **t** that best align the MDE predictions with our ground truth data.

It should be noted that there are several ways to process the calibration procedure. While the linear scale-and-shift method is most common, other strategies exist:

- *Scale-Only Calibration:* A simpler model that assumes only a scaling factor is needed (d_metric = s * d_relative). This is often sufficient but can be less accurate if there's a consistent bias in the MDE model.
- *Non-Linear Calibration:* For some models, the relationship might not be perfectly linear. One could fit a more complex function, like a polynomial or a spline, but this risks overfitting and is often unnecessary for modern, well-behaved MDE models.
- *Per-Image Calibration:* As implemented in the provided script, s and t are calculated for each individual frame. This is robust to frame-to-frame variations in lighting or exposure that might affect the MDE's output scale.
- *Global Calibration:* An alternative is to compute a single, global s and t across the entire dataset. This assumes the MDE model's scale is consistent across all frames and can be more stable if the ground truth in any single frame is sparse or noisy.

For this project, the per-image linear scale-and-shift calibration and global calibration were examined and evaluated.

The per-image linear scale-and-shift calibration

The main steps of the per-image linear scale-and-shift calibration:

- Step 1: Data Loading and Preprocessing

An RGB image and its corresponding ground-truth stereo depth map are loaded for a given frame. The ground-truth depth map, originally in millimeters, is converted to meters to match standard metric units.

- Step 2: Scene-Level Semantic Segmentation (Sky Detection)

The Segment Anything Model (SAM) is employed to perform semantic segmentation on the RGB image. The mask corresponding to the sky is extracted. This is a critical step, as both stereo and monocular depth methods fail for the sky (which has no texture and is effectively at infinite distance). A large, constant distance value (e.g., 250 meters) is assigned to all sky pixels in the depth map. This provides a semantically meaningful and realistic value for these regions.

- Step 3: Relative Depth Prediction

The Depth Anything V2 model is used to predict a high-resolution, relative depth map from the single RGB image. This map accurately captures the geometric structure of the scene but in an arbitrary, non-metric scale.

- Step 4: Scale and Shift Calibration via Linear Regression

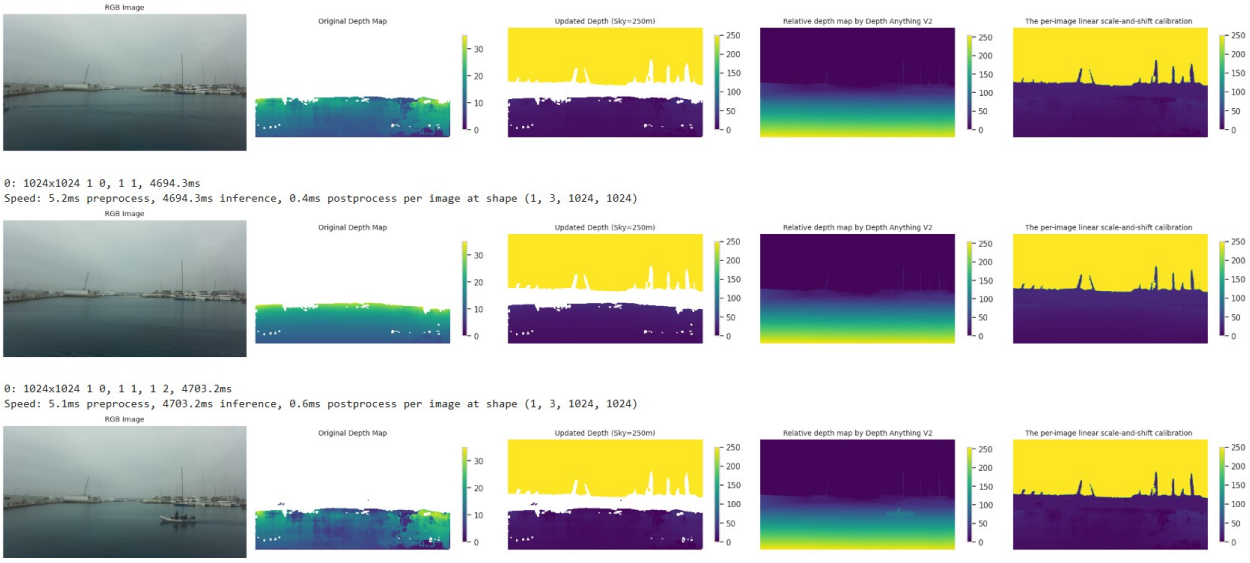
Create a High-Confidence Mask: A validation mask is created to select only the most reliable pixels from the ground-truth stereo depth map. The script includes pixels where the depth is between 2 and 35 meters and excludes: very close objects (<2m), which can be noisy as well as distant objects (>35m), where stereo depth accuracy degrades significantly and invalid (NaN)

pixels. Then linear regression is performed: Using the high-confidence pixels as anchor points, the *numpy.polyfit* function is used to solve for the best-fit linear relationship between the MDE predictions and the ground-truth stereo values. This calculates the optimal scale (s) and offset (t) parameters. Then the learned scale and offset are applied to the entire relative depth map from Depth Anything V2, transforming it from its arbitrary scale into metric meters.

- Step 5: Fusing ground truth and calibrated MDE

A final, robust depth map is generated by combining the two sources of information: ground truth and the calibrated MDE predictions. Then the sky mask is applied to ensure those regions retain their fixed far-distance value.

The outcomes of the per-image linear scale-and-shift calibration are show further.



The outcomes of the per-image linear scale-and-shift calibration: from left to right: RGB image, restored depth field with Telea inpainting and temporal smoothing, depth field with added mask for sky pixels, predicted by Depth Anything V2 relative depth map, predicted by Depth Anything V2 and calibrated with ground truth relative depth map

The global calibration

To ensure temporal consistency and robustness across the entire video sequence, a global calibration strategy was adopted. This approach computes a single, unified mapping from the monocular depth estimator's relative output to metric (meter) units. The process is executed in two main passes: a data collection pass and an application pass.

- Step 1: Aggregate Data Collection (Pass 1)

The first pass iterates through the entire dataset (167 frames) to build a set of corresponding data points. For each frame in the sequence a relative depth map is predicted using the Depth Anything V2 model. Then a high-confidence mask is created by selecting pixels from the ground-truth stereo depth map that are within a reliable range (2m to 35m). As a next action all

valid pairs of (monocular prediction, stereo ground truth) from this masked region are collected and appended to global lists.

- Step 2: Global Linear Regression (Pass 1)

After iterating through all frames, the collected data points are concatenated into two 1D arrays, representing all reliable monocular and stereo depth values from the entire dataset. A single linear regression (*numpy.polyfit*) is performed on these aggregated arrays to compute one global scale (s) and one global offset (t) that best fits the relationship $d\_metric = s * d\_relative + t$ across the whole sequence.

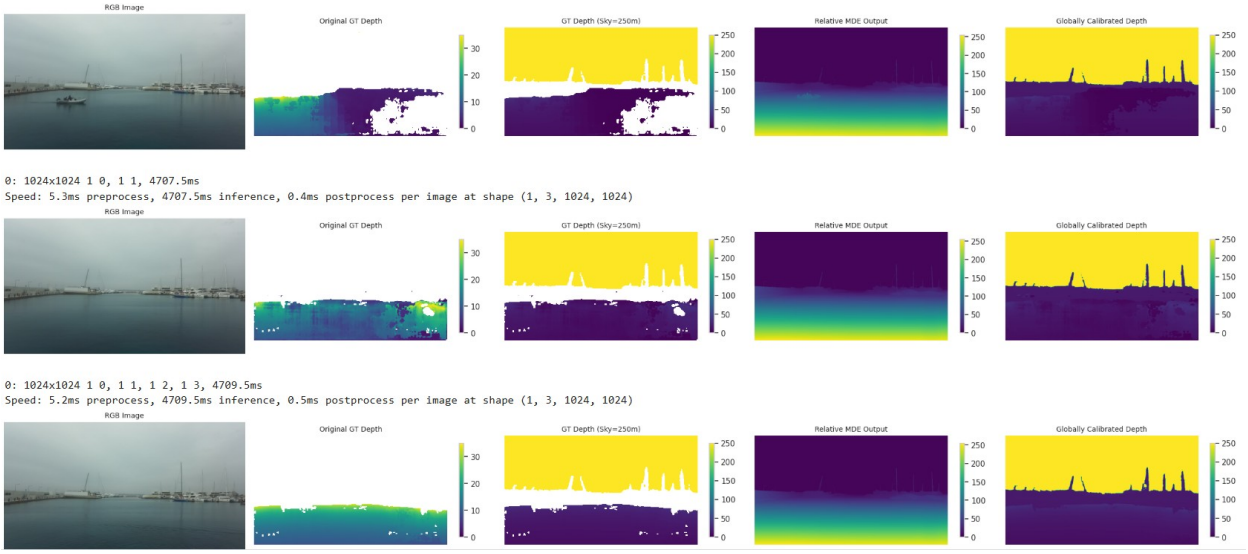
- Step 3: Application of Global Factors (Pass 2)

The second pass applies this globally consistent calibration to each frame individually. For each image a relative depth map is predicted using Depth Anything V2. Then the pre-computed global_scale and global_offset are applied to the entire relative depth map, transforming it into metric meters. There is no per-frame fitting.

- Step 4: Semantic Fusion for Final Depth Map

To generate the final depth map, the calibrated monocular depth is fused with the ground-truth stereo data and semantic information from a Segment Anything Model (SAM). The sky region, identified by SAM, is assigned a fixed far-distance value (250m). Then where the stereo ground truth is deemed reliable (2-35m), its accurate metric values are used. In all other regions (e.g., distances > 35m, reflective water surfaces), the globally calibrated and structurally coherent monocular depth prediction is used.

The outcomes of global calibration are presented further.



The outcomes of global calibration: from left to right: RGB image, restored depth field with Telea inpainting and temporal smoothing, depth field with added mask for sky pixels, predicted by Depth Anything V2 relative depth map, predicted by Depth Anything V2 and calibrated with ground truth relative depth map

Combining detections and depth field evaluation

Now we can combine predictions obtained in Task 3.1 with evaluation of depth field.