IMPROVING OBJECT DETECTION PERFORMANCE ON HARD TO DETECT INSTANCES IN DOTA

Said Harb

Institute of Geodesy and Photogrammetry
Technische Universität Braunschweig

Tim Schmidt

Institute of Geodesy and Photogrammetry
Technische Universität Braunschweig

ABSTRACT

Research on object detection continues to evolve, revealing various challenges that persist within the field. One significant challenge is the detection of small instances. In this study, we employ strategies to augment the DOTA dataset to improve object detection performance on hard to detect labeled instances. In particular, we oversample the dataset with images containing hard to detect instances and we copy and paste hard to detect instances within the oversampled images. Our findings demonstrate that both methods increase the mean average precision on hard to detect instances by 42.2% and 21.37% respectively. In addition the performance on easy to detect instances rises simultaneously.

***Index Terms*—** Object Detection, DOTA, Small Object Detection, Data Augmentation

1. INTRODUCTION

Object detection in Computervision (CV) has a wide range of applications across various research domains, from analyzing aerial imagery in Remote Sensing (RS) to processing images from unmanned aerial vehicles (UAVs) [1, 2, 3].

In the field of Object Detection (OD) in RS data, applications span wildlife and plant protection to urban traffic planning [2]. While object detection algorithms have achieved impressive performance on benchmark datasets like Common Objects in Context (COCO), these results often do not translate directly to RS data such as the Dataset for Object Detection in Aerial Images (DOTA) [2, 4].

RS imagery presents unique challenges, as it is typically captured from a bird's-eye perspective at significant distances [4]. This results in images covering large areas of landscape that appear more planar compared to the three-dimensional scenes in datasets like COCO [5]. Furthermore, objects of interest in RS imagery tend to be small relative to the overall image size, posing additional difficulties for detection algorithms [3].

With this work we aim to contribute the following aspects:

- Scrutinize the effect of oversampling on the detection performance of hard to detect (HTD) objects in RS images.
- Analyse the effect of copying HTD instances within oversampled images on the detection performance of HTD objects.

This paper examines the background and challenges associated with detecting HTD objects in RS imagery. The related works section reviews recent research addressing the detection of small and HTD instances. We then conduct experiments using a You Only Look Once (YOLO)v5n model to improve its performance on instances labeled as HTD in the DOTA dataset. The methodology and experimental setup are detailed in Section 2 [6]. Finally, we present and analyze the results of our experiments, followed by conclusions and suggestions for future work in this domain. This research aims to contribute to the ongoing efforts to enhance object detection capabilities for small instances in RS applications.

2. BACKGROUND

As already stated in Section 1, the main challenge with RS data is that these images are taken from a far distance, which introduces some problems for OD. The authors of the DOTA dataset labeled the instances with a difficulty level, categorizing them as either HTD or not. Upon closer examination of the dataset, it becomes evident that the majority of instances labeled as HTD are primarily small objects. This relationship between object size and detection difficulty is illustrated in Figure 1.

Further, if we look at the instances per class in Figure 2 we can see that most of the HTD instances come from the class "small-vehicle", followed by "large-vehicle" and "ship".

As the small size is one of the primary reasons for many HTD labels, this property is the focus of this work. While considerable research has been conducted in the field of object detection, there are significantly fewer results pertaining specifically to the detection of small or HTD instances [2]. Despite the high importance of HTD instances in various applications, most models exhibit poor performance in detecting them [7]. A notable example is the Mask R-CNN model, which is recognized as the state-of-the-art for the COCO dataset. Although it performs well on the overall dataset, its ability to detect small instances is limited [7].

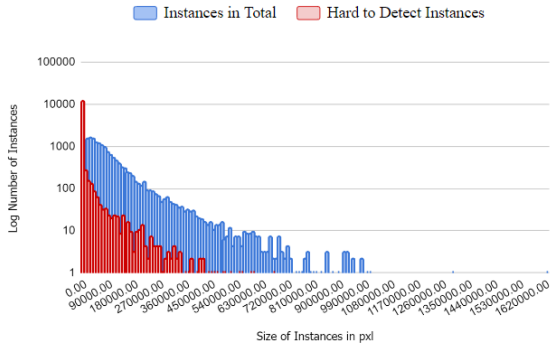


Fig. 1: **Histogram of DOTA instances size.** HTD instancs tend to be smaller than the average instance.

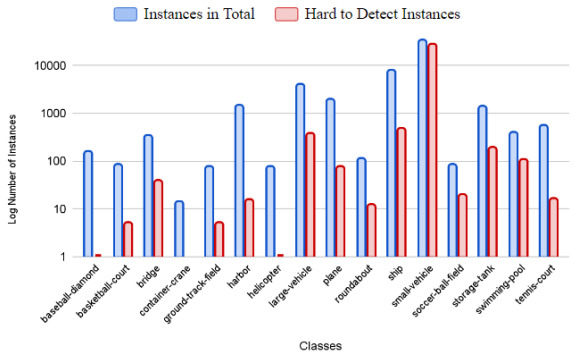


Fig. 2: **Number of instances per class.** Most HTD instances can be found for the class small-vehicle, followed by ship and large-vehicle. For the class container-crane no HTD instances exist, which is why it is excluded here.

Typically, an instance is labeled as "small" when it does not exceed 32×32 pixels [1]. With a low number of pixels as ground truth, several challenges may arise in detecting these instances.

Low Context. As small instances contain less information due to the limited number of pixels, they are often more challenging to detect [2][7][3]. To enhance detection capabilities, an object detection model can leverage additional contextual information from the surroundings of the instance. For instance, a harbor is typically bordered by water on at least one side, characterized by a specific pixel color, and has land on at least one other side, which also has a distinct color. Moreover, ships in the vicinity may exhibit a particular range of colors. This contextual information collectively suggests the presence of a harbor in the image. In contrast, a car serves as an example of low contextual information. Due to its relatively small size in the image, the model must rely on surrounding context to identify the instance as a car. However, if the car is parked on a road, the predominant contextual clue may be a grey pixel color, which is also true for pedestrians crossing the street, traffic lights, and various other objects. This situation results in a lack of sufficient information for accurately detecting this class [2][1].

Generating Bounding Boxes. Due to the small size of the instances, generating an appropriate bounding box can be quite challenging. Initially, at the beginning of the training process, the model must create one or more bounding boxes and evaluate the metrics to improve detection. When an instance occupies only a few pixels within a large image, it becomes difficult for the model to generate a reasonably sized bounding box that encompasses at least a portion of the instance. Furthermore, refining this initial guess poses additional challenges. When the model identifies a bounding box that includes part of the instance, shifting this box a few pixels in any direction may result in a bounding box that no longer contains any part of the instance. This situation can lead to significant fluctuations in the training metrics and confusion regarding the precise location of the instance [7].

Max-pooling and Stride in Convolutional Neural Network (CNN). The strategy behind a CNN also has certain limitations when it comes to detecting small instances. While the various layers within the network assist in compressing the image information into a manageable format, small details can be lost in the process. Initially, when an instance is only a few pixels wide, operations such as max pooling and the stride of the CNN kernel can result in the instance becoming unrecognizably small or, in the worst-case scenario, completely disappearing altogether [3].

Small Number of Instances. Another problem can be that the sample size of the small instances is not large enough. If this occurs, the learning process for detecting these instances can be adversely affected. The model may become biased, perceiving that there are only a limited number of HTD instances, either by class or by size. Consequently, it will learn to overlook these instances in most cases, focusing instead on larger objects or classes that are more frequently encountered [7].

3. RELATED WORK

As the detection of HTD instances is an important topic in today's research, there are already some different approaches to confronting the challenges [3].

In the paper by Liu et al. [8], the authors changed the architecture of the YOLO model and added a Feature Enhancement Block. This, in connection with an independent processing of the more shallow feature maps, led to better extraction of small features and therefore better detection of small instances, as stated in Section 2.

Another approach is worked out in the paper by Zhang et al. [9]. Here, they too work with a YOLOv5 based algorithm and introduce a space-to-depth conv module. Additionally, they added an attention mechanism to retain the lost details due to max pooling and stride.

Our work makes use of a strategy used in the paper by

Kisantal et al. [7]. Instead of changing the architecture of the YOLO model, the data that gets fed into the model is augmented here. The authors conduct many experiments from which we will replicate the most promising ones on our dataset. In particular these experiments are two approaches. In the first approach, the authors copied the images that contain HTD instances and thereby oversampled the dataset with images containing HTD instances. In the second approach, they copied copied and pasted all HTD instances within oversampled images. In both methods, they varied the number of copies to find a good performance on both small and larger instances. The result was that copying the images one time, as well as copy-pasting every HTD instance once on the image, yielded the best performance. To the best of the authors' knowledge, this kind of work was not done before to improve the performance on HTD instances in DOTA.

4. METHODOLOGY

The dataset utilized in DOTA consists of images of various sizes. Initially, these images undergo preprocessing to ensure they are suitable for training. The preprocessed training images are then processed in three different ways. In the first experiment, the preprocessed images remain unchanged. For the second and third experiments, we implement the techniques discussed in section 3, which will be described in greater detail in this section. Next, the preprocessed and modified images from both the training and validation sets must be prepared for YOLO[10] training. It is important to note that testing will take place outside of the YOLO framework, which is why the test images are excluded from this preparation process. Finally, three variants of YOLOv5n are trained on the three different versions of the training set and evaluated on the same test set. Figure 4 in the appendix section A provides an overview of the methodology.

4.1. Splits

Before initiating the training pipeline, we established the dataset splits to be used in this work. The authors of DOTA[6] provided the training and validation sets; however, the labels for the test set are not publicly available. Consequently, we opted to divide the validation set into 40% for validation and 60% for test images, using a seed of 42. The resulting number of images is presented in Table 1.

4.2. Preprocessing

To enable the YOLO model to process the images effectively, they must be divided into patches of uniform size. Leading papers that achieve state-of-the-art (SOTA) mean average precision (mAP) on the DOTA dataset benchmark[1] — such as those by Yu et al. [11], Feng et al. [12], and Li et al. [13] — split the DOTA images into patches measuring 1024×1024 pixels, with overlaps ranging from 200 to 500 pixels. In this work, we will also split the images into patches of 1024×1024 pixels, using an overlap of 256 pixels. Any additional space created by images that do not conform to this splitting format will be filled with zero padding. Consequently, the target bounding box information for each image will also be divided into corresponding targets based on the new image patches. The total number of patches (all and HTD) and the ratio of patches containing HTD instances after splitting the images and targets are presented in Table 2. The same overview regarding the instances can be found in Table 3.

Table 1: **Comparison of native DOTA and our DOTA image split.** The native training set from the authors could be used, but we had to split the 458 validation images into a new validation and test set, because no public test labels were available.

	Native DOTA	Our DOTA
Train images	1 411	1 411
Val images	458	183
Test images	937	275
Sum	2 806	1 869

Table 2: **HTD patches in the baseline (BL) split.** We aimed for a balanced distribution of patches containing HTD instances.

	Total patches	HTD patches	Ratio
Train	21 132	5 588	25.97%
Val	3 139	669	21.31%
Test	2 279	657	28.83%
Sum	26 650	6 914	25.94%

4.3. Oversampling

For our first experiment, we oversample patches that contain at least one HTD instance once in the BL training set. As a result, the oversampled training set includes all images containing hard-to-detect instances twice, while images without hard-to-detect instances appear only once. This approach increases the ratio of patches containing hard-to-detect instances from 25.97% to 41.23%. The resulting number of images in the training set is presented in Table 4.

[1] https://paperswithcode.com/sota/object-detection-in-aerial-images-on-dota-1(accessed on September 16, 2024)

Table 3: **HTD instances in the BL splits.** The ratio of HTD instances is more imbalanced than the ratio of patches containing HTD instances, but it is within an acceptable range.

	Total instances	HTD instances	Ratio
Train	357 576	160 157	44.79%
Val	34 204	12 647	36.98%
Test	52 115	28 656	54.99%
Sum	443 895	201 460	45.38%

Table 4: **HTD patches in the training set.** The ratio of patches containing HTD instances rises from the BL to experiment 1 and stays the same in experiment 2, as no further oversampling of patches is conducted here (OS = Oversampling).

Name	OS	Aug.	Total patches	HTD patches	HTD patches ratio
BL	—	—	21 132	5 488	25.97%
Exp. 1	✓	—	26 620	10 976	41.23%
Exp. 2	✓	✓	26 620	10 976	41.23%

4.4. Oversampling and Augmentation

In the second experiment, we replicate the procedure from the first experiment on the BL training set, meaning that the training set is oversampled once with patches containing HTD instances. The key distinction in this experiment is that the copied or oversampled images undergo augmentation. Following the recommendations outlined in section 3, all HTD instances in each patch are copied and pasted once within the respective patch. The algorithm for copying selects all pixels within the bounding box coordinates of the HTD instance and pastes them without any blur or overlap with existing bounding boxes at a random location within the image. This process increases the ratio of HTD instances further from experiment 1 to experiment 2 which can be seen in Table 5. It is important to note that 24 instances could not be copied because they were too large to be pasted without overlapping any other bounding boxes.

4.5. Preparing YOLO training

For the YOLO model to train effectively on the patches and labels, they must be organized into the correct directory structure. Additionally, the labels need to be converted to the YOLO standard. The bounding box coordinates are formatted as follows: $(\text{label}; x_{\text{center}}; y_{\text{center}}; w; h)$, where x_{center} and y_{center} represent the normalized center coordinates of the bounding box, and w and h denote the normalized width and height. Since YOLO models do not accept an additional input feature, such as the difficulty of the bounding box in our case, the evaluation of the YOLO models must be conducted manually. In this work, all instances not marked as HTD by the authors of DOTA will be noted as easy to detect (ETD).

Table 5: **HTD instances in the training set.** The ratio of HTD instances rises from BL to experiment 1 due to oversampling. The ratio increases further in experiment 2 due to augmenting the oversampled patches (OS = Oversampling).

Name	OS	Aug.	Total instances	HTD instances	HTD instances ratio
BL	—	—	357 576	160 157	44.79%
Exp. 1	✓	—	652 403	320 314	49.01%
Exp. 2	✓	✓	812 536	480 447	59.13%

5. EXPERIMENTAL PROTOCOL

In this work, we aim to investigate the effectiveness of the proposed methods. Consequently, the model selection is less targeted, as we want the methods to be applicable across different model architectures. We selected *YOLOv5n*[14] for all three experiments, which were trained for a maximum of 200 epochs on the *Phoenix-Cluster* at *TU-Braunschweig* using *Tesla P100-SXM2-16GB* Graphics Processing Unit (GPU) hardware.

YOLOv5n is a lightweight and fast model, comprising 1.9 million parameters which does not require large amounts of memory. Early stopping was configured with a patience of 30 epochs, while the initial learning rate was set to 0.01, with a minimum learning rate of $0.000,01$. The input image size was established at 1024 pixels.

The batch size for the baseline model and the first experiment was set to 2, whereas the batch size for the third experiment was increased to 8, as it was conducted on 4 GPUs instead of one, as in the baseline and first experiment. This adjustment was necessary due to memory allocation constraints on the utilized cluster. Aside from these specifications, the standard settings for *YOLOv5n* training, as of September 11, 2024, were employed. All three experiments were executed for the full duration of 200 epochs. The Intersection over Union (IoU) threshold for bounding boxes was set to 0.5, and the confidence score for valid predictions, used in calculating precision, recall, and F1-score, was also set to 0.5.

5.1. Evaluation Metrics

In object detection, several metrics are used to evaluate model performance, including precision, recall, F1-score, Average Precision (AP), and mean Average Precision (mAP).

Precision is the ratio of correctly predicted positive instances (True Positives, TP) to all predicted positive instances, including both true positives and false positives (FP).

It is given by:

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

Precision measures how many of the predicted positive instances are actually correct.

Recall is the ratio of correctly predicted positive instances to all actual positive instances, which includes true positives and false negatives (FN). The formula is:

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

Recall measures how many of the actual positives are correctly identified by the model.

The **F1-score** is defined as:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-score is the harmonic mean between precision and recall and will reflect unbalanced metrics.

The **AP** is computed as the area under the **precision-recall curve (PR curve)**, the precision is plotted over the recall for all confidence thresholds. The AP can be defined as:

$$\text{AP} = \int_0^1 \text{Precision}(\text{Recall})\, d\text{Recall}$$

It summarizes the precision-recall curve across all recall levels.

Mean average precision(mAP) is the mean of the AP values across all object classes. It can be defined as:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i$$

where N is the number of object classes, and AP_i is the average precision for class i.

6. RESULTS

The mAP of the BL model, along with the models from both experiments, is summarized in Table 6 and visualized in Figure 3. The oversampling approach without augmentation from experiment 1 demonstrated the highest mAP across all difficulty levels. The oversampling and augmentation approach from experiment 2 performed only slightly worse than the aforementioned method. However, both approaches significantly enhance performance compared to the BL model, particularly for ETD instances. For HTD instances, the model from experiment 1 improved the mAP by 4.42 percentage points, representing a 42.18% relative increase over the BL model. In contrast, the model from experiment 2 improved the mAP of HTD instances by 2.24 percentage points, corresponding to a 21.37% relative increase compared to the BL model.

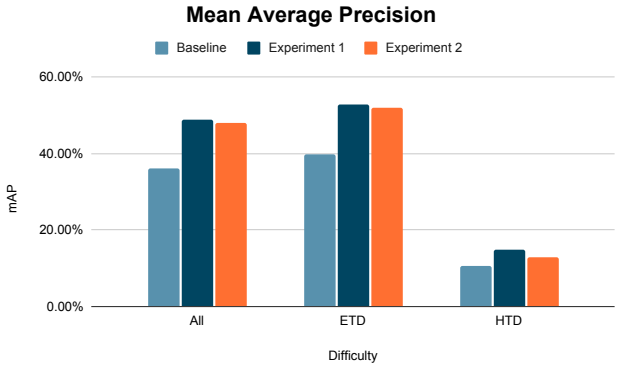


Fig. 3: **Mean average precision.** Both experiments improved the detection of instances in all difficulties. ETD instances have the highest mAP and HTD the lowest mAP. However, the approach in experiment 1 has a slightly better performance than the one in experiment 2.

Difficulty	BL (%)	Exp. 1 (%)	Exp. 2 (%)
All	36.14	**48.90**	47.79
ETD	39.64	**52.71**	51.91
HTD	10.48	**14.90**	12.72

Table 6: **Mean average precision for all models and difficulties.** The approach from experiment 1 improved the mAP the most across all difficulties.

For a more detailed analysis, Figure 8 in appendix section D illustrates the AP for all classes and difficulty levels across the three models, while Figure 9 presents the corresponding precision, recall, and F1-score. Figure 10 specifically highlights HTD instances by displaying only their AP. The concrete values for all difficulties and classes can be found section D in the appendix. Regarding the values for the precision, recall and F1-score at an IoU and confidence threshold of 0.5, refer to section B in the appendix.

In order to understand the differences between the models there are five examples of predictions of all models shown in section E in the appendix. The baseline model fails to detect many instances, especially HTD ones. The model from experiment 1 improves this, as on the examples noticable more objects were detected. The model from experiment 2 also performs better than the BL model, but it detects slightly less instances.

This section will primarily focus on HTD instances, although the performance on all instances and ETD instances will also be taken into account. The performance of all models on HTD instances can be categorized into three groups: classes with an AP above 20%, classes with an AP below 20% but greater than zero, and classes that were not detected at all. The first group will be considered as classes with good per-

formance, the second group as classes with bad performance and the last group as undetected classes.

6.1. Classes with good performance

In this group, the classes *ship*, *tennis-court*, *bridge*, *small-vehicle*, and *soccer-ball field* are included. All of these classes share the characteristic that at least one model achieved an average precision (AP) above 20% for HTD instances.

The class *tennis-court* exhibits the overall best performance, featuring the highest AP observed across all classes and difficulty levels. This can likely be attributed to the low variety of tennis court instances in the dataset, as noted by the authors during visual inspection. Tennis courts are predominantly rectangular and share similar color characteristics, which likely contributed to their higher detection rates. Additionally, tennis courts achieved the highest recall among all classes using the approach from experiment 2. Conversely, both experimental approaches resulted in a comparable improvement in AP over the BL model, with experiment 1 showing a slight advantage. Notably, the HTD instances experienced the most significant improvement compared to ETD instances or all instances combined.

The class *bridge* exhibits the second-best average precision (AP) from the model in experiment 1. There was a 14 percentage point increase compared to the BL model, demonstrating the effectiveness of oversampling in improving bridge detection. In contrast, augmenting the oversampled images resulted in only a marginal improvement over the BL model. Interestingly, the class *bridge* is the sole instance where the performance on HTD instances surpasses the performance on all instances and ETD instances, which is otherwise quite unusual. We speculate that the HTD bridges may not have been as challenging to detect as initially assumed. The class *bridge* is also unique in another aspect. For both experiments, the precision is 100%, which is remarkably high. However, the recall is below 12% for both experiments and 0% for the BL model. This suggests that in the BL model, no bridges were detected, and the model must have never been more confident than 50%, as the threshold for valid predictions is set to 0.5 for calculating precision, recall, and F1-score. Consequently, the AP value is derived solely from low-confidence predictions. We can infer that the model was often uncertain in detecting bridges, but it can reliably detect bridges with lower confidence thresholds relative to other classes.

The third-best performance is observed in the class *soccer-ball field*, where experiment 2 achieved an increase of 6.1 percentage points. However, the approach in experiment 1 more than doubled the AP for soccer-ball fields, adding 32.67 percentage points to the AP of the BL model. This substantial increase can be attributed to the significant rise in the sample size of soccer-ball fields due to oversampling. On the other hand, it was not always possible to copy soccer-ball fields to random locations within the image, as these fields are often large in size, making it challenging to copy them without overlap to other bounding boxes. Overall, the performance on soccer-ball fields benefited from both experimental approaches, with a more pronounced increase observed in experiment 1. Additionally, experiment 1 demonstrated greater precision in detecting soccer-ball fields, but both experiments identified only a small fraction of the soccer-ball fields, as indicated by the low recall values. Nevertheless, both experiments were able to detect soccer-ball fields with greater confidence than the BL model.

The fourth-best average precision (AP) for HTD instances is observed in the most abundant class in the training set: small vehicles. In this case, the oversampling method outperformed the oversampling and augmentation method, although both approaches resulted in an increase of AP compared to the BL model. Notably, hard-to-detect instances exhibited a significant improvement in AP. While ETD instances showed a maximum increase of only 2.37 percentage points in experiment 2, in contrast the maximum increase for HTD instances in experiment 1 is 7.98 percentage points. This suggests that the large sample size scaled most effectively with the oversampling method, making small vehicles even more prevalent in the training set. However, similar to other classes, most HTD instances were not detected, as indicated by the low recall values. On the other hand, precision remains relatively high. Interestingly, for a confidence threshold of 0.5, the precision of the BL model on HTD *small-vehicles* is the largest of all models.

The fifth-best performing class is *ship*, which, in contrast to the previous four classes, benefited the most from oversampling and augmentation in experiment 2. All difficulty levels improved with both experiments, but the class *ship* stands out by achieving the second highest average precision (AP) for both ETD and all instances. However, the recall remains very low, although it improved from the BL model to experiment 1 and again in experiment 2.

In summary, it is evident that the approach from experiment 1 — oversampling without augmentation — yielded the best improvements in four out of five cases. Nevertheless, the approach from the second experiment still outperformed the BL model. Another key aspect is that, in all cases, the performance for all instances also improved.

6.2. Classes with bad performance

This group includes all classes that have less than 20% average precision (AP) for hard-to-detect (HTD) instances, but not zero. Within this group, three classes benefited more from experiment 1, while two classes showed greater improvement from experiment 2. Two classes stand out as particularly noteworthy in this context. First, we will discuss the classes that benefited more from experiment 1, followed by those that showed greater improvement in experiment 2, and finally, we will highlight the special cases.

The classes *storage-tank*, *swimming-pool*, and *ground-track field* benefited more from oversampling without augmenting the oversampled images.

Notably, for *ground-track field*, which was not detected at all in the BL model, the improvement of 3.92 percentage points from experiment 1 is substantial. The recall for this class experienced a dramatic increase of 40 percentage points, making it the highest recall of all classes for HTD instances in experiment 1. However, *ground-track field* also has one of the lowest HTD precision scores, which is common for larger objects such as baseball fields, basketball courts, or soccer-ball fields. We hypothesize that these instances may be too large for the 1024×1024 image input size, making it more challenging for the model to consider the global context effectively.

HTD swimming pools exhibit a relatively low average precision (AP), low precision, and even lower recall. The model found it particularly challenging to detect swimming pools, although both experiments demonstrated a nearly identical increase in performance. In contrast ETD instances benefited more from the approaches employed in the experiments, with the approach from experiment 1 being more dominant.

The class *storage tank* has a high frequency in the training set. Its overall performance ranks fifth or sixth, depending on the model; however, the HTD performance is notably poor. This suggests that HTD storage tanks may possess unusual characteristics, as ETD instances benefited more from the approaches used in both experiments than their HTD counterparts. Additionally, storage tanks have one of the lowest recall rates among the classes.

The classes *harbors* and *roundabouts* benefited more from experiment 2. The best-performing class in this group is *harbor*, where HTD instances showed improvements from the approaches in both experiments, with a greater benefit observed in experiment 2. Notably, the overall performance increased by approximately 20 percentage points. However, both recall and precision remained relatively low compared to the classes in the first group.

In contrast, *roundabouts* were not detected by the BL model or by the model from experiment 1, but they experienced a significant increase in AP of 5.81 percentage points in the model from experiment 2. In this case, augmenting the oversampled images proved beneficial for detection performance, although this only applies to low confidence scores, as both precision and recall are zero for a confidence threshold of 0.5.

Two classes are exceptions to the observed trends: *planes* and *large vehicles*. For these classes, the experiments did not result in an increase or only showed a very small increase in HTD AP, while the overall performance increased. This suggests that the HTD instances for both classes possess such unique characteristics that it is not possible to substantially improve performance on them by oversampling or augmenting the oversampled images alone. Instead, refinements to the feature extraction process may be necessary. For *large vehicles*, the precision on HTD instances at a 0.5 confidence threshold decreases from the BL model to experiments 1 and 2.

In summary, while the approaches employed in the experiments help detect more instances, they are not a silver bullet for solving the HTD problem on their own. Considerations regarding model architecture must be made in order to improve performance on these challenging classes.

6.3. Non-detected classes

There are four classes for which no model was able to detect any HTD instances: *baseball fields*, *basketball courts*, *helicopters*, and *container cranes*. It is important to note that there are no instances of container cranes in the test set due to an already low sample size in the training set. Therefore, this class will not be considered further.

Helicopters are not only rare in the training set, but they also exhibit ambiguous appearances. Some helicopters are conventional, while others lack rotor blades, and occasionally, helipads without helicopters were labeled as helicopters by the DOTA authors.

Basketball-courts and *baseball-fields* are relatively large instances, and their HTD instances also possess unique characteristics, such as unusual shapes of the sports facilities or atypical colors due to seasonal changes, like yellow grass instead of green. To improve training performance for these classes, the native sample size must be increased, as even oversampling and augmenting the images did not yield any benefits in this case.

6.4. Precision-Recall Curves

In Figure 5, the PR curve for all classes and experiments are depicted for HTD instances. For all and ETD instances Figures 6 and 7 can be found in section C the appendix. These PR curve curves provide a confidence threshold-independent assessment of model performance and serve as the foundation for calculating average precision (AP) and mean average precision (mAP). The PR curve reinforces the findings of the previous subsections and highlights that the recall for many classes is relatively low. Additionally, they illustrate the significant improvements achieved in the experiments for classes such as *ship*, *tennis court*, *ground-track field*, *bridge*, and *soccer-ball field*.

7. DISCUSSION AND CONCLUSION

In this section, we present a detailed analysis of the results from our experiments, highlighting the key findings and drawing conclusions based on their implications. We examine the impact of oversampling and augmentation strategies on the

performance of our model, with particular focus on their effect on HTD instances. Following this, we outline potential factors that may have influenced the results and propose directions for future research.

7.1. Discussion

The first experiment, which focused on oversampling the training data with images containing hard-to-detect (HTD) instances, performed as expected. Since more training data generally leads to better model performance, the observed effects were anticipated. In contrast, the results from the second experiment were more surprising, as they underperformed relative to the results from the first experiment, despite showing a significant improvement over the BL model. The authors of the original paper [7] noted an improvement in performance with the approach of oversampling and augmentation compared to the approach of only oversampling, which was not the case in our study. In this section, we will discuss potential factors that could have influenced these outcomes.

First, the authors of the original paper perform their research on the COCO dataset [15] which contains images of common objects within their context. We on the other hand applied this approach to remote sensing images, which are of different context. Therefore it cannot be sure that the approach which worked on COCO will also work the same way on DOTA. In this paper we scrutinized this aspect.

Another key aspect arising from the different datasets is that in COCO, instances not only have bounding boxes but also include segmentation masks for semantic segmentation tasks. This availability allows for a more refined process of copying and pasting hard-to-detect (HTD) instances in images, as the exact silhouette of the instances can be copied, rather than simply the content of the entire bounding box, as was done in our case.

We opted to copy the whole content of the bounding box for feasibility reasons, as manually cropping all HTD instances would be tedious for over $160,000$ instances. By copying the entire content of the bounding box, a portion of the context surrounding the instance is also included, making the copied instance more distinguishable due to the retained background. However, copying only the instance itself would likely enable the model to learn the characteristics of the instances more effectively, rather than just recognizing that something was copied.

Another aspect to consider is that when copying and pasting the instances, we did not account for areas that were zero-padded. As a result, it is possible for an instance to be copied into a zero-padded space, which removes a significant amount of context from the instance.

Additionally, the distribution of HTD instances within the training, validation, and test sets could be more stratified to mitigate potential biases in the model. In this work, the sets are not heavily imbalanced, but they are also not completely equal in terms of the ratio of images containing HTD instances or the ratio of HTD instances themselves.

The final aspect concerns the evaluation phase of this work. We trained on overlapping patches and validated on non-overlapping patches. Consequently, we also tested on non-overlapping patches, which leads to some instances being cut in half by the image partitions. This makes it more challenging for the model to learn the classes, even though we can envision that a more sophisticated model would enhance its generalization abilities by also training on these cut instances.

7.2. Conclusion

In this work, we demonstrated that oversampling the training data with images containing HTD instances and augmenting those images by copying and pasting HTD instances within the images is an effective method for improving detection performance on HTD objects. The first approach yielded better results, achieving an increase of 42.18% in mean average precision (mAP), while the second approach enhanced the mAP of the BL model by 21.37%.

It is not entirely clear which class benefits most from each approach, but we suggest that oversampling the training sets is beneficial for all classes. Subsequently, augmentation can be applied to selected classes to evaluate whether their performance improves through this method. If augmentation proves effective, it should be applied exclusively to those classes, while being omitted for classes that do not benefit. In this study, classes related to water contexts, such as ships and harbors, particularly benefited from augmentation. This may be due to reduced loss of context when copying an image from a water background to another water background. Conversely, copying a small vehicle onto a water background results in a greater loss of context.

In future work, more sophisticated models can be tested with the two approaches utilized in this study. Additionally, the augmentation process can be refined by employing segmentation masks to copy the exact silhouette of the instance, rather than the content of the bounding box. Combining this approach with modifications to the model architecture could also be of great interest, as it would integrate both external and internal aspects of detecting small or HTD objects.

Acronyms

AP Average Precision. 4–7, 27

BL baseline. 3–9, 14, 15, 22–25, 28, 32

CNN Convolutional Neural Network. 2

COCO Common Objects in Context. 1, 8

CV Computervision. 1

DOTA Dataset for Object Detection in Aerial Images. 1–4, 7–9

ETD easy to detect. 4–7, 9, 14, 17, 20, 23, 28–30

GPU Graphics Processing Unit. 4

HTD hard to detect. 1–9, 15, 18, 20, 22, 25–30, 35

IoU Intersection over Union. 4, 5

mAP mean Average Precision. 4, 5, 7, 8

OD Object Detection. 1

PR curve precision-recall curve. 5, 7

RS Remote Sensing. 1

YOLO You Only Look Once. 1–4

List of Tables

1 Comparison of native DOTA and our DOTA image split. 3
2 HTD patches in the BL split. 3
3 HTD instances in the BL splits. 4
4 HTD patches in the training set. 4
5 HTD instances in the training set. 4
6 Mean average precision for all models and difficulties. 5
7 Precision, Recall, F1-score for all instances in the BL model. 14
8 Precision, Recall, F1-score for ETD instances in the BL model. 14
9 Precision, Recall, F1-score for HTD instances in the BL model. 15
10 Precision, Recall, F1-score for all instances in the experiment 1 model 17
11 Precision, Recall, F1-score for ETD instances in the experiment 1 model 17
12 Precision, Recall, F1-score for HTD instances in the experiment 1 model 18
13 Precision, Recall, F1-score for all instances in the experiment 2 model 19
14 Precision, Recall, F1-score for ETD instances in the experiment 2 model 20
15 Precision, Recall, F1-score for HTD instances in the experiment 2 model 20
16 Average precision for the BL model. 28
17 Average precision for the experiment 1 model. 29
18 Average precision for the experiment 2 model. 30

List of Figures

1 Histogram of DOTA instances size. 2
2 Number of instances per class. 2
3 Mean average precision. 5
4 Methodology flow chart 12
5 Precision-Recall Curves for HTD instances. . 22
6 Precision-Recall Curves for ETD instances. . 23
7 Precision-Recall Curves for all instances. . . 24
8 Average precision for three models. 25
9 Precision, Recall, F1-Score for HTD instances for three models. 26
10 Average precision of HTD instances. 27
11 Prediction 1 32
12 Prediction 2 33
13 Prediction 3 34
14 Prediction 4 35
15 Prediction 5 36
16 Bounding box Loss on the Baseline model. . 38
17 Classification loss on the Baseline model. . . 39
18 Distribution Focal Loss on the Baseline model. 40
19 Bounding Box Loss on the Experiment 1 model. 42
20 Classification Loss on the experiment 1 model. 43
21 Distribution Focal Loss on the experiment 1 model. 44
22 Bounding Box Loss on the Experiment 2 model. 46
23 Classification Loss on the experiment 2 model. 47
24 Distribution Focal Loss on the experiment 2 model. 48
25 Mean Average Precision of all three models during training. 50
26 Learning rate of all three models. 52

8. REFERENCES

[1] Kang Tong, Yiquan Wu, and Fei Zhou, "Recent advances in small object detection based on deep learning:

A review," *Image and Vision Computing*, vol. 97, pp. 103910, 5 2020.

[2] Xuan Cao, Yanwei Zhang, Song Lang, and Yan Gong, "Swin-Transformer-Based YOLOv5 for Small-Object Detection in Remote Sensing Images," *Sensors*, vol. 23, no. 7, pp. 3634, 3 2023.

[3] Yang Liu, Peng Sun, Nickolas Wergeles, and Yi Shang, "A survey and performance evaluation of deep learning methods for small object detection," *Expert Systems with Applications*, vol. 172, pp. 114602, 6 2021.

[4] Gong Cheng, Chunbo Lang, Maoxiong Wu, Xingxing Xie, Xiwen Yao, and Junwei Han, "Feature Enhancement Network for Object Detection in Optical Remote Sensing Images," *Journal of Remote Sensing*, vol. 2021, 1 2021.

[5] Yang Long, Yiping Gong, Zhifeng Xiao, and Qing Liu, "Accurate object localization in remote sensing images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 5, pp. 2486–2498, 2017.

[6] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang, "Object detection in aerial images: A large-scale benchmark and challenges," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[7] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho, "Augmentation for small object detection," *arXiv (Cornell University)*, 1 2019.

[8] Zhen Liu, Xuehui Gao, Yu Wan, Jianhao Wang, and Hao Lyu, "An Improved YOLOv5 Method for Small Object Detection in UAV Capture Scenes," *IEEE Access*, vol. 11, pp. 14365–14374, 1 2023.

[9] Jian Zhang, Guoyang Wan, Ming Jiang, Guifu Lu, Xiuwen Tao, and Zhiyuan Huang, "Small object detection in UAV image based on improved YOLOv5," *Systems Science & Control Engineering*, vol. 11, no. 1, 8 2023.

[10] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.

[11] Hongtian Yu, Yunjie Tian, Qixiang Ye, and Yunfan Liu, "Spatial transform decoupling for oriented object detection," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, pp. 6782–6790, 3 2024.

[12] Mingkui Feng, Hancheng Yu, Xiaoyu Dang, and Ming Zhou, "Category-Aware Dynamic Label Assignment with High-Quality Oriented Proposal," *arXiv (Cornell University)*, 7 2024.

[13] Yuxuan Li, Qibin Hou, Zhaohui Zheng, Ming-Ming Cheng, Jian Yang, and Xiang Li, "Large Selective Kernel Network for Remote Sensing Object Detection," *arXiv (Cornell University)*, 1 2023.

[14] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu, changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020.

[15] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

A. METHODOLOGY

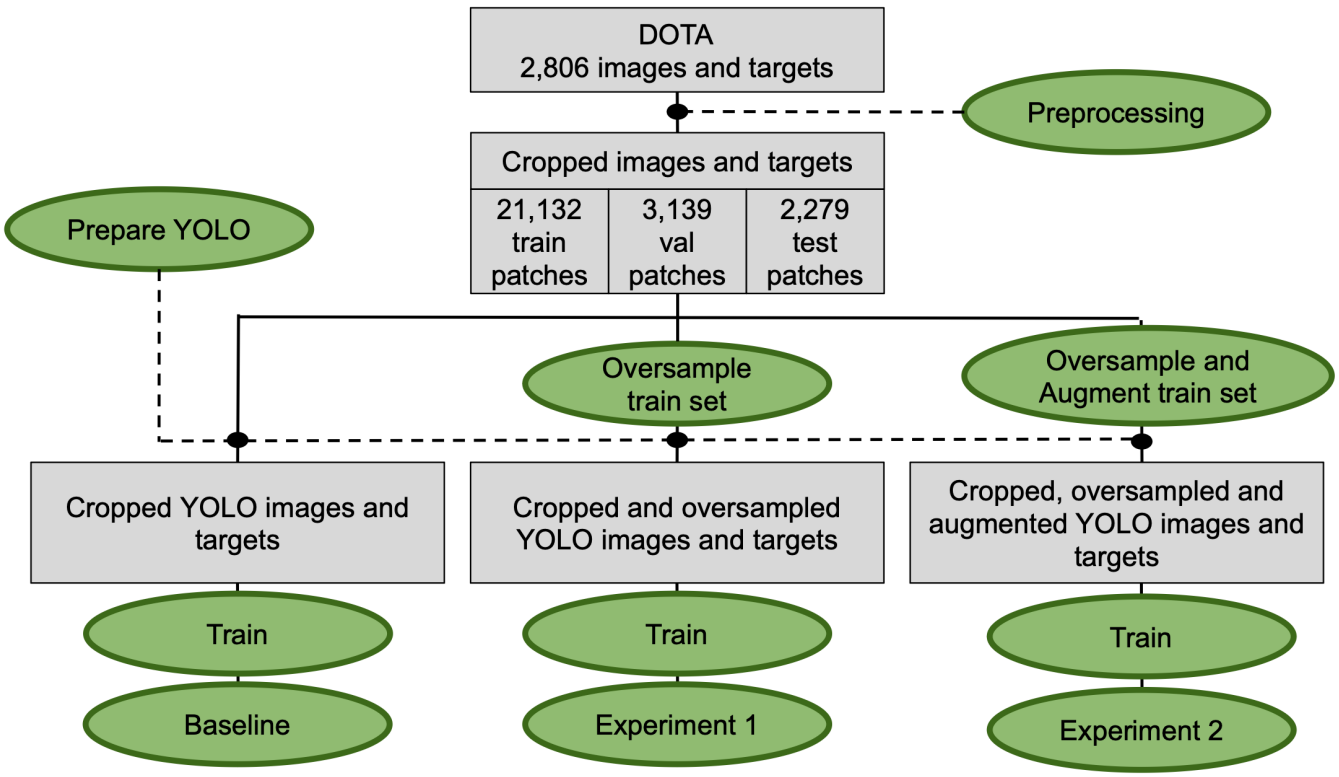


Fig. 4: **Flowchart of the methodology.** The images are split into patches of uniform size during preprocessing and are then handled in different ways according to the respective experiment.

B. RESULTS: PRECISION, RECALL, F1-SCORE

B.1. Baseline

class	iou threshold	confidence threshold	precision	recall	f1
ship	0.5	0.5	98.02%	56.66%	71.81%
storage-tank	0.5	0.5	98.87%	26.08%	41.27%
baseball-diamond	0.5	0.5	100.00%	1.96%	3.85%
tennis-court	0.5	0.5	98.82%	76.32%	86.13%
basketball-court	0.5	0.5	84.62%	13.58%	23.40%
ground-track-field	0.5	0.5	50.00%	13.16%	20.83%
bridge	0.5	0.5	88.89%	2.33%	4.55%
large-vehicle	0.5	0.5	94.89%	31.42%	47.21%
small-vehicle	0.5	0.5	89.71%	20.66%	33.58%
helicopter	0.5	0.5	0.00%	0.00%	0.00%
swimming-pool	0.5	0.5	91.67%	8.55%	15.64%
roundabout	0.5	0.5	85.71%	10.81%	19.20%
soccer-ball-field	0.5	0.5	91.67%	13.25%	23.16%
plane	0.5	0.5	98.20%	57.69%	72.68%
harbor	0.5	0.5	92.00%	11.11%	19.83%
container-crane	0.5	0.5	0.00%	0.00%	0.00%

Table 7: Precision, Recall, F1-score for all instances in the BL model.

class	iou threshold	confidence threshold	precision	recall	f1
ship	0.5	0.5	98.04%	59.73%	74.23%
storage-tank	0.5	0.5	99.15%	30.30%	46.42%
baseball-diamond	0.5	0.5	100.00%	1.97%	3.87%
tennis-court	0.5	0.5	99.52%	77.49%	87.13%
basketball-court	0.5	0.5	84.62%	14.47%	24.72%
ground-track-field	0.5	0.5	52.63%	14.08%	22.22%
bridge	0.5	0.5	88.89%	2.63%	5.11%
large-vehicle	0.5	0.5	95.18%	34.56%	50.70%
small-vehicle	0.5	0.5	81.27%	50.21%	62.07%
helicopter	0.5	0.5	0.00%	0.00%	0.00%
swimming-pool	0.5	0.5	94.29%	11.83%	21.02%
roundabout	0.5	0.5	85.71%	12.12%	21.24%
soccer-ball-field	0.5	0.5	100.00%	17.19%	29.33%
plane	0.5	0.5	98.82%	60.06%	74.71%
harbor	0.5	0.5	92.53%	11.23%	20.02%
container-crane	0.5	0.5	0.00%	0.00%	0.00%

Table 8: Precision, Recall, F1-score for ETD instances in the BL model

class	iou threshold	confidence threshold	precision	recall	f1
ship	0.5	0.5	81.03%	9.83%	17.54%
storage-tank	0.5	0.5	0.00%	0.00%	0.00%
baseball-diamond	0.5	0.5	0.00%	0.00%	0.00%
tennis-court	0.5	0.5	66.67%	37.50%	48.00%
basketball-court	0.5	0.5	0.00%	0.00%	0.00%
ground-track-field	0.5	0.5	0.00%	0.00%	0.00%
bridge	0.5	0.5	0.00%	0.00%	0.00%
large-vehicle	0.5	0.5	20.00%	0.27%	0.54%
small-vehicle	0.5	0.5	93.54%	13.72%	23.92%
helicopter	0.5	0.5	0.00%	0.00%	0.00%
swimming-pool	0.5	0.5	0.00%	0.00%	0.00%
roundabout	0.5	0.5	0.00%	0.00%	0.00%
soccer-ball-field	0.5	0.5	0.00%	0.00%	0.00%
plane	0.5	0.5	8.33%	1.32%	2.27%
harbor	0.5	0.5	0.00%	0.00%	0.00%
container-crane	0.5	0.5	0.00%	0.00%	0.00%

Table 9: Precision, Recall, F1-score for HTD instances in the BL model

B.2. Experiment 1

class	iou threshold	confidence threshold	precision	recall	f1
ship	0.5	0.5	96.91%	64.87%	77.72%
storage-tank	0.5	0.5	98.99%	36.36%	53.19%
baseball-diamond	0.5	0.5	93.22%	35.95%	51.89%
tennis-court	0.5	0.5	99.35%	83.24%	90.58%
basketball-court	0.5	0.5	86.67%	32.10%	46.85%
ground-track-field	0.5	0.5	46.97%	40.79%	43.66%
bridge	0.5	0.5	54.12%	13.41%	21.50%
large-vehicle	0.5	0.5	92.29%	48.79%	63.84%
small-vehicle	0.5	0.5	85.75%	29.00%	43.34%
helicopter	0.5	0.5	88.24%	19.74%	32.26%
swimming-pool	0.5	0.5	93.42%	18.39%	30.74%
roundabout	0.5	0.5	94.12%	28.83%	44.14%
soccer-ball-field	0.5	0.5	94.29%	39.76%	55.93%
plane	0.5	0.5	98.59%	66.79%	79.63%
harbor	0.5	0.5	92.62%	38.10%	53.99%
container-crane	0.5	0.5	0.00%	0.00%	0.00%

Table 10: Precision, Recall, F1-score for all instances in the experiment 1 model

class	iou threshold	confidence threshold	precision	recall	f1
ship	0.5	0.5	97.02%	68.20%	80.10%
storage-tank	0.5	0.5	98.98%	41.82%	58.79%
baseball-diamond	0.5	0.5	94.83%	36.18%	52.38%
tennis-court	0.5	0.5	99.78%	84.62%	91.57%
basketball-court	0.5	0.5	100.00%	34.21%	50.98%
ground-track-field	0.5	0.5	47.54%	40.85%	43.94%
bridge	0.5	0.5	51.85%	13.82%	21.82%
large-vehicle	0.5	0.5	92.57%	53.57%	67.87%
small-vehicle	0.5	0.5	73.18%	63.79%	68.16%
helicopter	0.5	0.5	88.24%	20.00%	32.61%
swimming-pool	0.5	0.5	95.89%	25.09%	39.77%
roundabout	0.5	0.5	96.97%	32.32%	48.48%
soccer-ball-field	0.5	0.5	96.88%	48.44%	64.58%
plane	0.5	0.5	99.14%	69.53%	81.74%
harbor	0.5	0.5	93.06%	38.35%	54.32%
container-crane	0.5	0.5	0.00%	0.00%	0.00%

Table 11: Precision, Recall, F1-score for ETD instances in the experiment 1 model

class	iou threshold	confidence threshold	precision	recall	f1
ship	0.5	0.5	77.01%	14.02%	23.72%
storage-tank	0.5	0.5	83.33%	2.67%	5.18%
baseball-diamond	0.5	0.5	0.00%	0.00%	0.00%
tennis-court	0.5	0.5	75.00%	37.50%	50.00%
basketball-court	0.5	0.5	0.00%	0.00%	0.00%
ground-track-field	0.5	0.5	40.00%	40.00%	40.00%
bridge	0.5	0.5	100.00%	10.26%	18.60%
large-vehicle	0.5	0.5	16.67%	1.36%	2.51%
small-vehicle	0.5	0.5	90.79%	20.83%	33.88%
helicopter	0.5	0.5	0.00%	0.00%	0.00%
swimming-pool	0.5	0.5	25.00%	0.93%	1.80%
roundabout	0.5	0.5	0.00%	0.00%	0.00%
soccer-ball-field	0.5	0.5	66.67%	10.53%	18.18%
plane	0.5	0.5	9.09%	1.32%	2.30%
harbor	0.5	0.5	40.00%	13.33%	20.00%
container-crane	0.5	0.5	0.00%	0.00%	0.00%

Table 12: Precision, Recall, F1-score for HTD instances in the experiment 1 model

B.3. Experiment 2

class	iou threshold	confidence threshold	precision	recall	f1
ship	0.5	0.5	97.65%	64.06%	77.37%
storage-tank	0.5	0.5	99.10%	32.86%	49.36%
baseball-diamond	0.5	0.5	98.04%	32.68%	49.02%
tennis-court	0.5	0.5	99.11%	80.69%	88.96%
basketball-court	0.5	0.5	95.65%	27.16%	42.31%
ground-track-field	0.5	0.5	56.76%	27.63%	37.17%
bridge	0.5	0.5	68.42%	11.40%	19.55%
large-vehicle	0.5	0.5	93.54%	46.44%	62.07%
small-vehicle	0.5	0.5	85.78%	26.89%	40.95%
helicopter	0.5	0.5	85.71%	15.79%	26.67%
swimming-pool	0.5	0.5	85.29%	15.03%	25.55%
roundabout	0.5	0.5	100.00%	22.52%	36.76%
soccer-ball-field	0.5	0.5	87.50%	33.73%	48.70%
plane	0.5	0.5	98.72%	65.26%	78.57%
harbor	0.5	0.5	95.98%	34.60%	50.86%
container-crane	0.5	0.5	0.00%	0.00%	0.00%

Table 13: Precision, Recall, F1-score for all instances in the experiment 2 model

class	iou threshold	confidence threshold	precision	recall	f1
ship	0.5	0.5	97.69%	67.23%	79.65%
storage-tank	0.5	0.5	99.54%	37.66%	54.65%
baseball-diamond	0.5	0.5	98.04%	32.89%	49.26%
tennis-court	0.5	0.5	99.54%	81.80%	89.80%
basketball-court	0.5	0.5	100.00%	28.95%	44.90%
ground-track-field	0.5	0.5	58.82%	28.17%	38.10%
bridge	0.5	0.5	65.38%	11.22%	19.15%
large-vehicle	0.5	0.5	93.86%	51.10%	66.17%
small-vehicle	0.5	0.5	73.63%	61.30%	66.90%
helicopter	0.5	0.5	85.71%	16.00%	26.97%
swimming-pool	0.5	0.5	90.48%	20.43%	33.33%
roundabout	0.5	0.5	100.00%	25.25%	40.32%
soccer-ball-field	0.5	0.5	96.30%	40.63%	57.14%
plane	0.5	0.5	98.96%	67.93%	80.56%
harbor	0.5	0.5	96.15%	34.89%	51.20%
container-crane	0.5	0.5	0.00%	0.00%	0.00%

Table 14: Precision, Recall, F1-score for ETD instances in the experiment 2 model

class	iou threshold	confidence threshold	precision	recall	f1
ship	0.5	0.5	85.23%	15.69%	26.50%
storage-tank	0.5	0.5	75.00%	3.21%	6.15%
baseball-diamond	0.5	0.5	0.00%	0.00%	0.00%
tennis-court	0.5	0.5	77.78%	43.75%	56.00%
basketball-court	0.5	0.5	0.00%	0.00%	0.00%
ground-track-field	0.5	0.5	33.33%	20.00%	25.00%
bridge	0.5	0.5	100.00%	12.82%	22.73%
large-vehicle	0.5	0.5	4.76%	0.27%	0.51%
small-vehicle	0.5	0.5	90.54%	18.81%	31.14%
helicopter	0.5	0.5	0.00%	0.00%	0.00%
swimming-pool	0.5	0.5	20.00%	0.93%	1.79%
roundabout	0.5	0.5	0.00%	0.00%	0.00%
soccer-ball-field	0.5	0.5	33.33%	10.53%	16.00%
plane	0.5	0.5	16.67%	1.32%	2.44%
harbor	0.5	0.5	50.00%	6.67%	11.76%
container-crane	0.5	0.5	0.00%	0.00%	0.00%

Table 15: Precision, Recall, F1-score for HTD instances in the experiment 2 model

C. PRECISION-RECALL CURVES

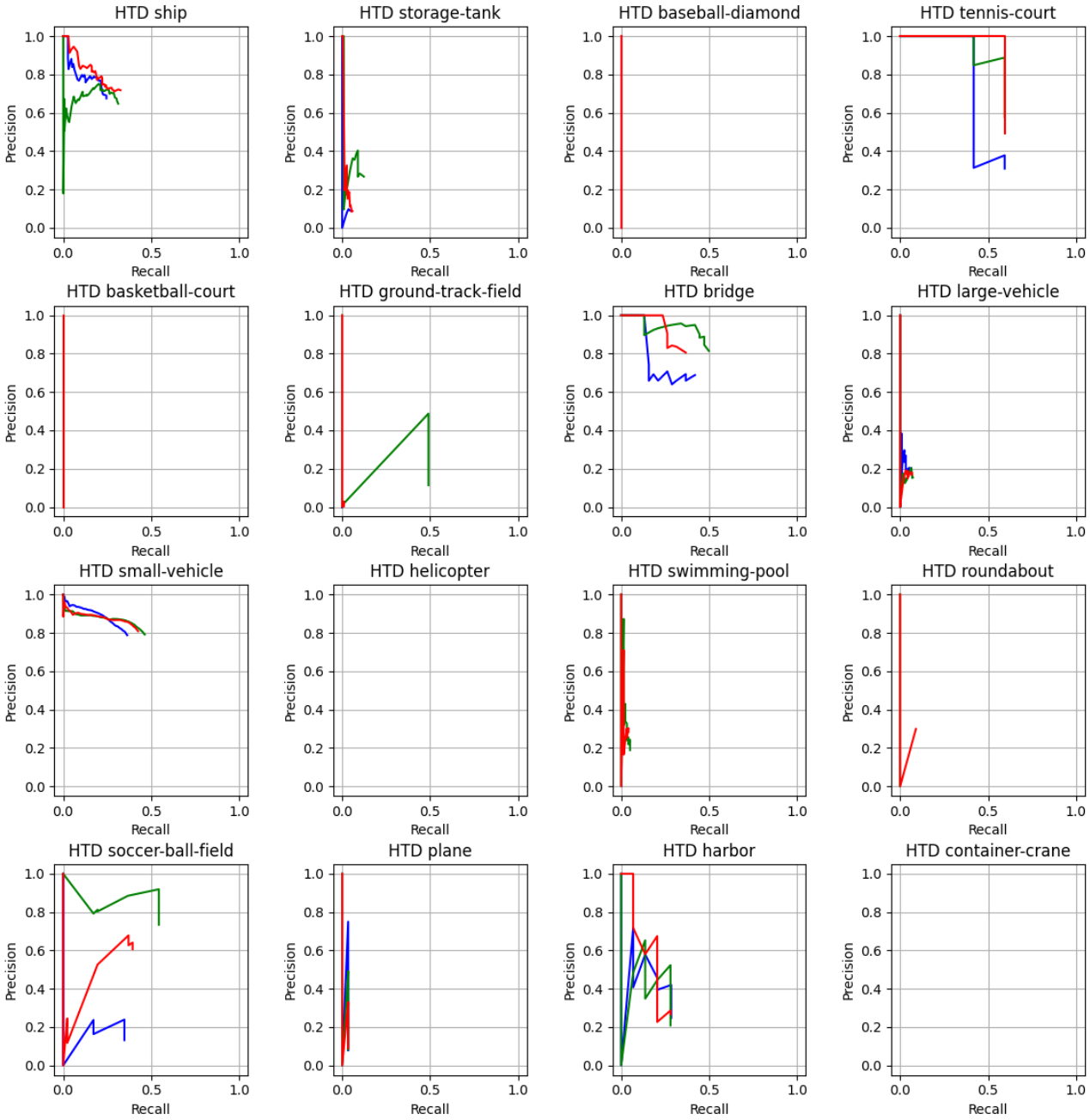


Fig. 5: **Precision-Recall Curves for HTD instances.** The blue line indicates the BL model, the green line the model from experiment 1 and the red line the model from experiment 2. The approach in experiment 1 improved the curves of the classes storage-tank, ground-track field, bridge, small-vehicle and soccer-ball-field. The approach from experiment 2 improved the performance on the classes ship, tennis-court, roundabout and harbor the most.

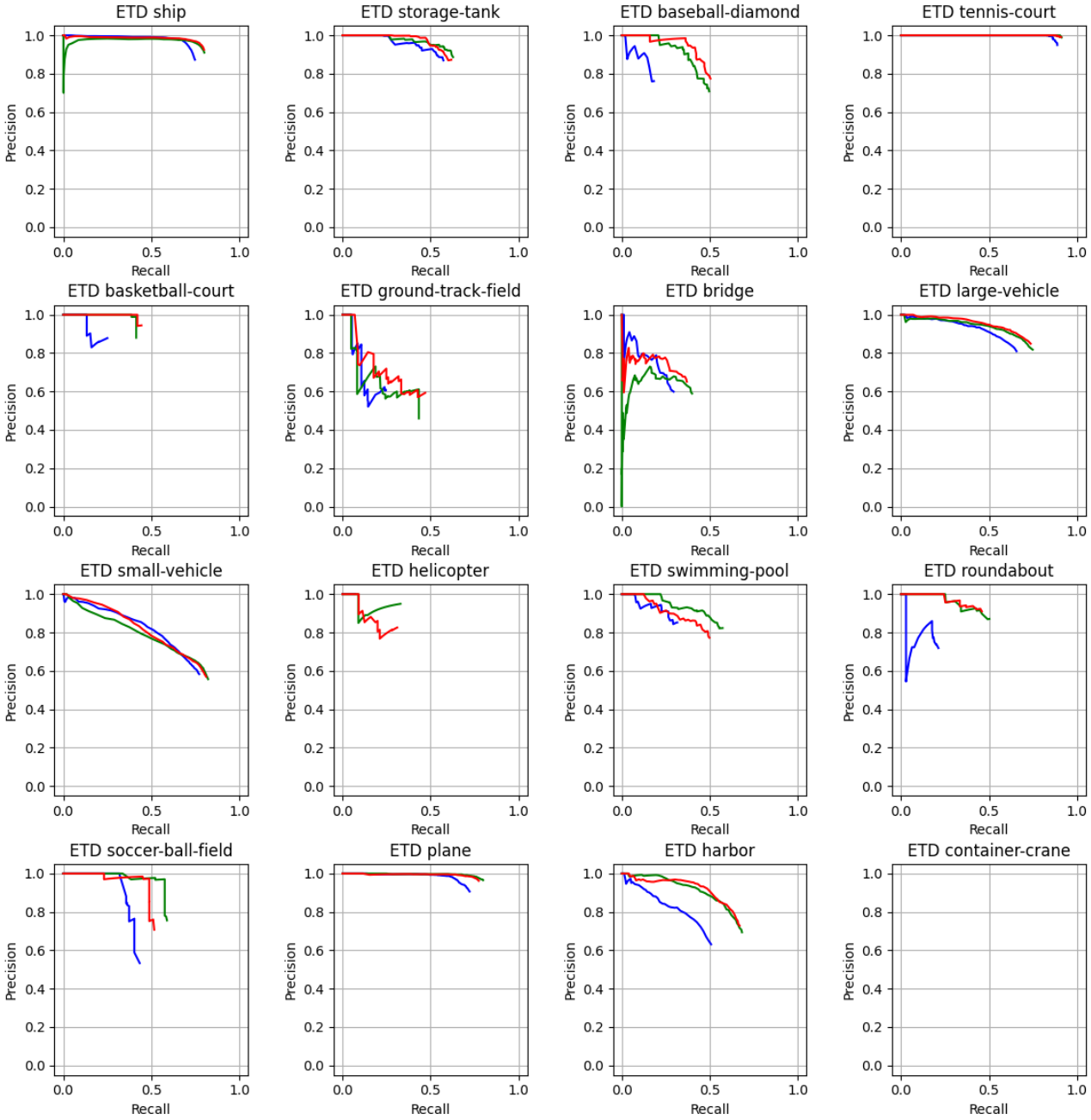


Fig. 6: **Precision-Recall Curves for ETD instances.** The blue line indicates the BL model, the green line the model from experiment 1 and the red line the model from experiment 2. Here it is visible that both approaches worked for ETD instances across all classes.

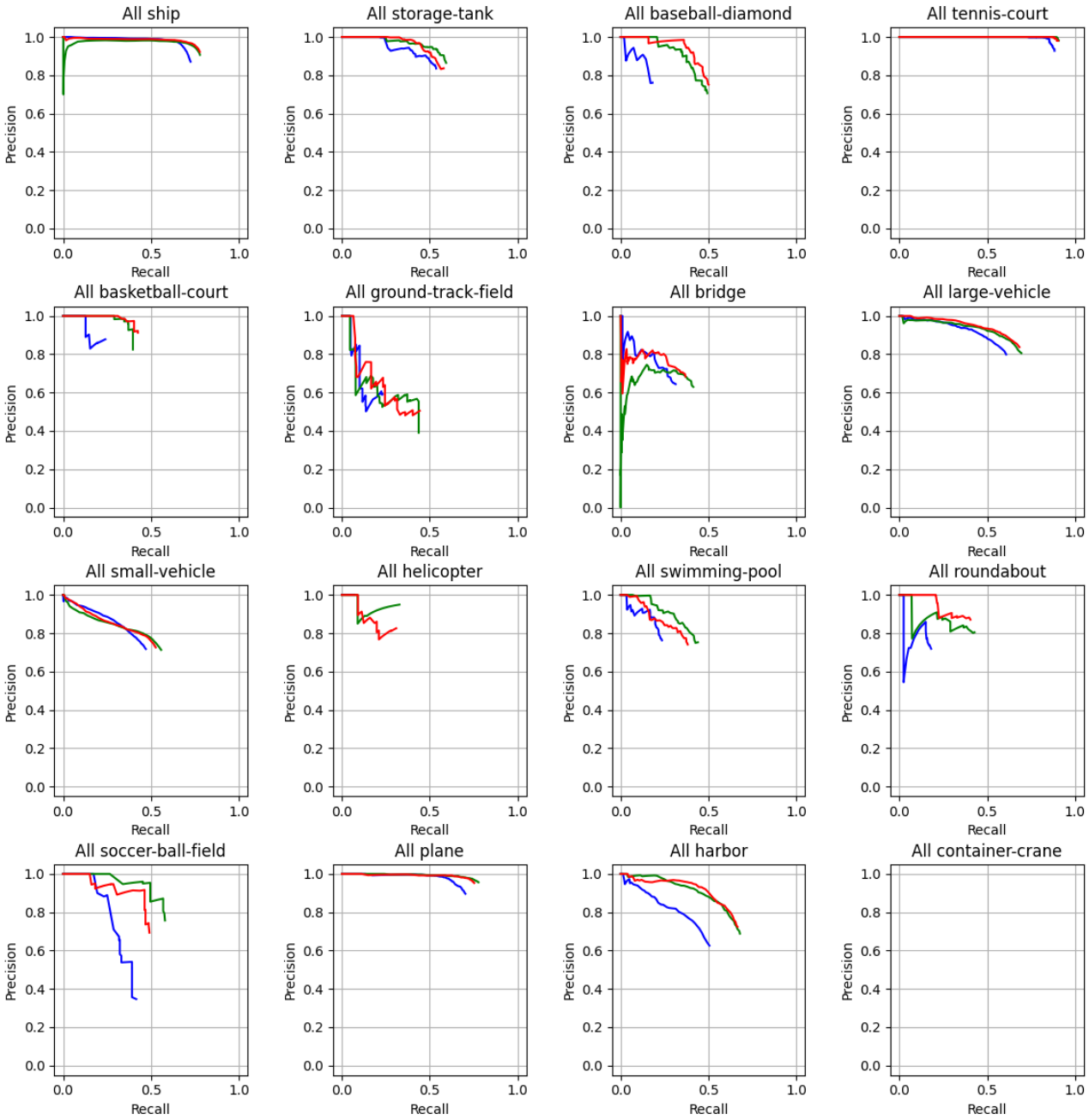
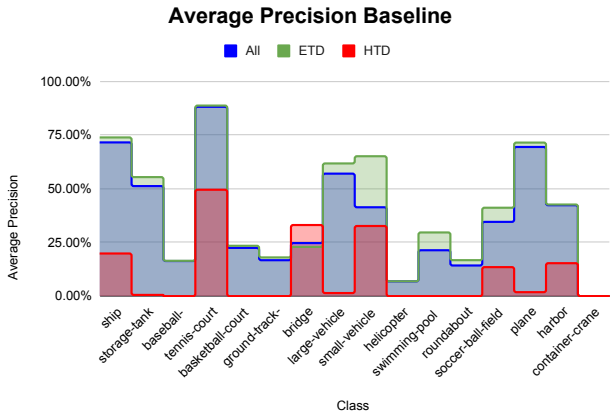
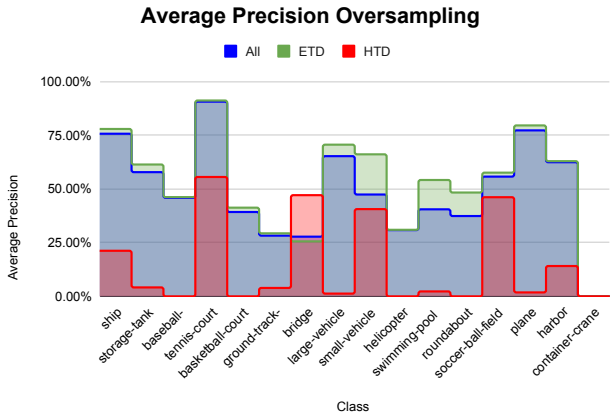


Fig. 7: **Precision-Recall Curves for all instances.** The blue line indicates the BL model, the green line the model from experiment 1 and the red line the model from experiment 2. Looking at all instances some classes benefit only from oversampling while other also benefit from the augmentation of the oversampled images.

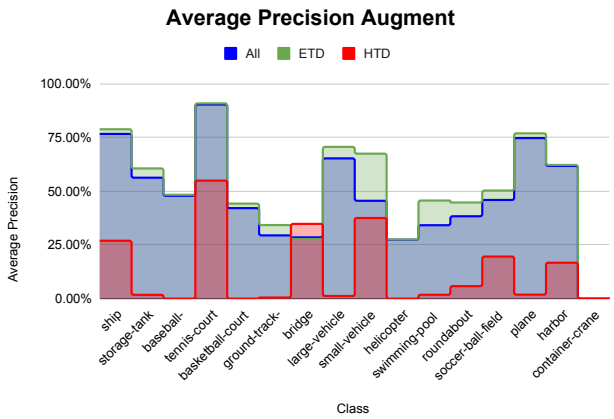
D. AVERAGE PRECISION



(a) **Average precision BL.** Here 7 classes were not detected for HTD instances. The best performance is on tennis-courts.

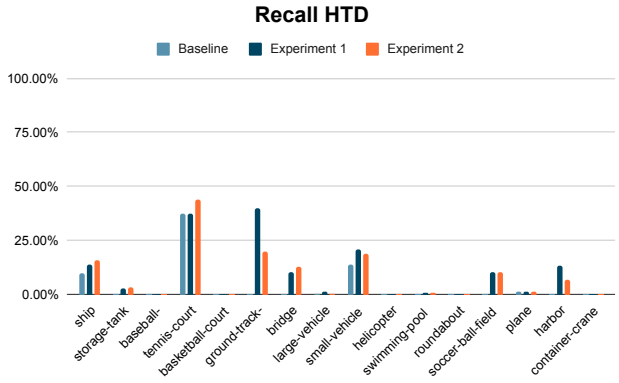


(b) **Average precision experiment 1.** Here 5 classes were not detected for HTD instances. The best performance is on tennis-courts.

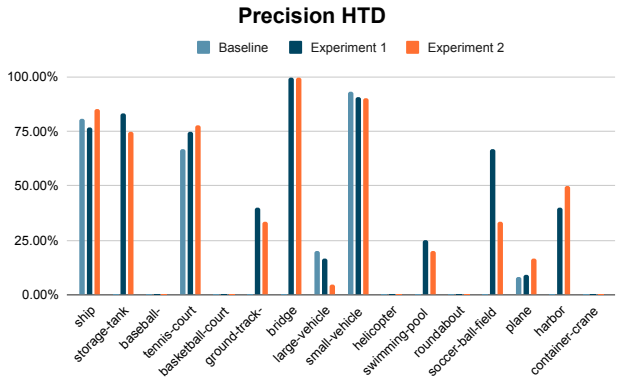


(c) **Average precision experiment 2.** Here 4 classes were not detected for HTD instances. The best performance is on tennis-courts.

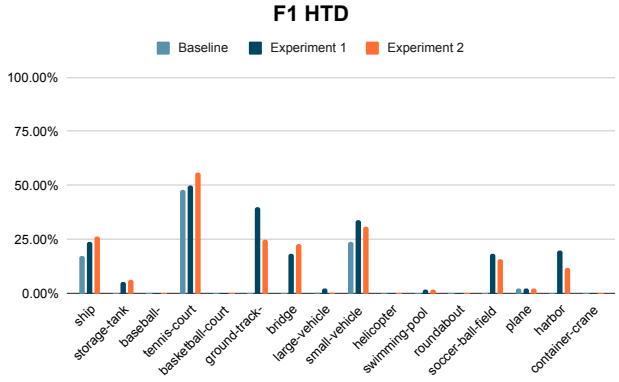
Fig. 8: Average precision for three models.



(a) **Recall HTD instances.** The highest recall is for tennis court, but overall the recall is low for all all models. Especially for ground track fields oversampling worked well.



(b) **Precision HTD instances.** The precision is visibly higher than the recall for all classes. Especially for soccer ball fields oversampling worked well.



(c) **F1 score for HTD instances.** Combining precision and recall, tennis courts and small vehicles are detected the most by all models. Major improvements are visible for the storage-tank, ground-track field, bridge, soccer-ball field and harbor class by the approaches in experiment 1 and 2.

Fig. 9: Precision, Recall, F1-Score for HTD instances for three models.

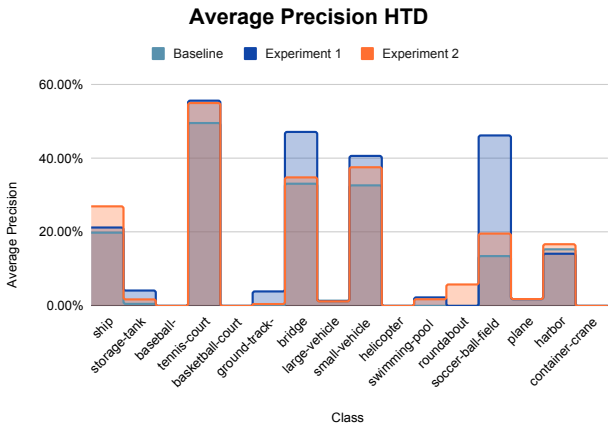


Fig. 10: **Average precision of HTD instances.** Comparing the AP across all three models it is evident that both approaches ine experiment 1 and 2 improved the detection of HTD instances. The model from experiment 2 detected one class more (roundabout) than the model from experiment 1, however the latter model shows higher AP in many classes.

Class	All	ETD	HTD
ship	71.55%	73.87%	19.84%
storage-tank	51.22%	55.40%	0.56%
baseball-diamond	16.33%	16.44%	0.00%
tennis-court	88.13%	88.70%	49.52%
basketball-court	22.45%	23.43%	0.00%
ground-track-field	16.71%	18.01%	0.00%
bridge	24.69%	22.91%	33.09%
large-vehicle	57.01%	61.76%	1.42%
small-vehicle	41.36%	65.12%	32.64%
helicopter	6.87%	6.90%	0.00%
swimming-pool	21.35%	29.63%	0.00%
roundabout	14.26%	16.73%	0.00%
soccer-ball-field	34.56%	41.15%	13.49%
plane	69.43%	71.47%	1.84%
harbor	42.31%	42.64%	15.32%
container-crane	0.00%	0.00%	0.00%

Table 16: Average precision for the BL model.

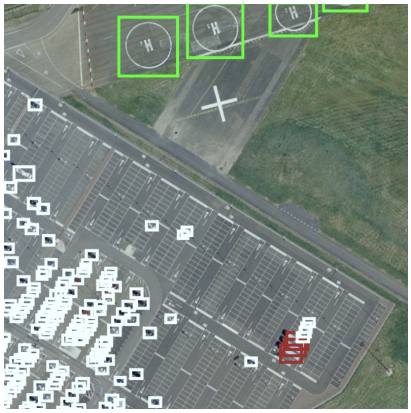
Class	All	ETD	HTD
ship	75.72%	77.93%	21.23%
storage-tank	57.83%	61.39%	4.17%
baseball-diamond	45.88%	46.21%	0.00%
tennis-court	90.65%	91.19%	55.58%
basketball-court	39.28%	41.30%	0.00%
ground-track-field	28.27%	29.35%	3.92%
bridge	27.79%	25.61%	47.12%
large-vehicle	65.28%	70.63%	1.29%
small-vehicle	47.42%	66.13%	40.62%
helicopter	30.85%	30.99%	0.00%
swimming-pool	40.53%	54.18%	2.30%
roundabout	37.40%	48.34%	0.00%
soccer-ball-field	55.75%	57.59%	46.16%
plane	77.28%	79.59%	1.84%
harbor	62.45%	62.95%	14.13%
container-crane	0.00%	0.00%	0.00%

Table 17: Average precision for the experiment 1 model.

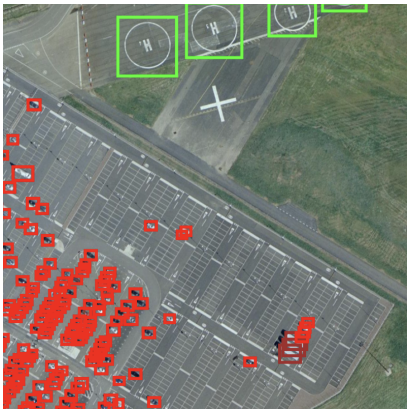
Class	All	ETD	HTD
ship	76.68%	78.85%	26.96%
storage-tank	56.30%	60.62%	1.76%
baseball-diamond	47.88%	48.35%	0.00%
tennis-court	90.37%	90.91%	54.94%
basketball-court	42.15%	44.23%	0.00%
ground-track-field	29.45%	34.26%	0.50%
bridge	28.53%	27.73%	34.80%
large-vehicle	65.30%	70.64%	1.23%
small-vehicle	45.55%	67.49%	37.55%
helicopter	27.41%	27.54%	0.00%
swimming-pool	34.18%	45.67%	1.80%
roundabout	38.35%	44.75%	5.81%
soccer-ball-field	45.94%	50.33%	19.59%
plane	74.76%	76.96%	1.85%
harbor	61.78%	62.23%	16.71%
container-crane	0.00%	0.00%	0.00%

Table 18: Average precision for the experiment 2 model.

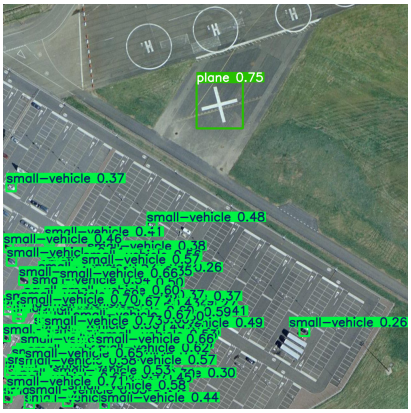
E. PREDICTION EXAMPLES



(a) Target



(b) HTD Instances in Red

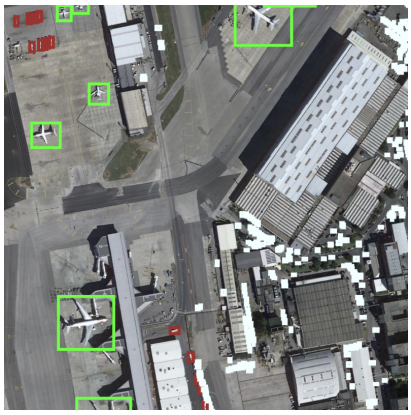


(c) Detections of the Baseline Model

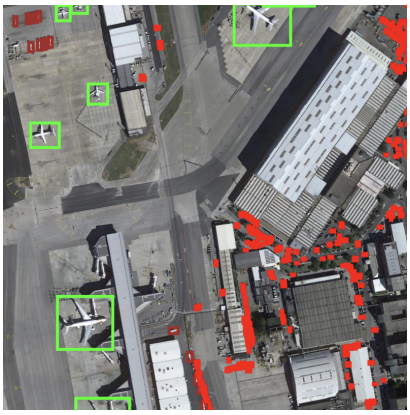
(d) Detections of the Model in Experiment 1

(e) Detections of the Model in Experiment 2

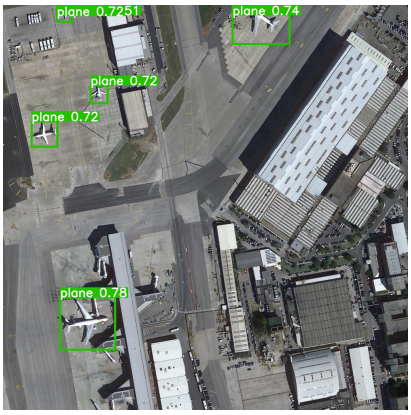
Fig. 11: Classes: White → small vehicle, green → helicopter, brown-red → large vehicle. Note that the white cross is a false positive in the BL model and in experiment 1.



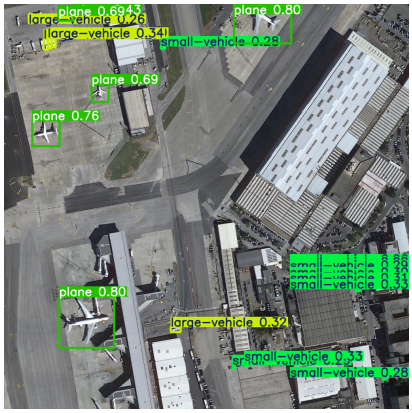
(a) Target



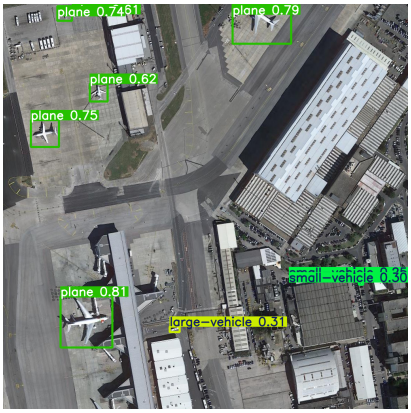
(b) HTD Instances in Red



(c) Detections of the Baseline Model

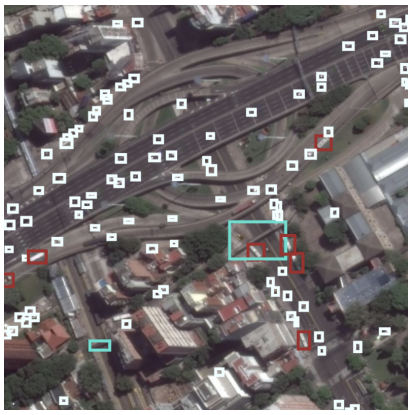


(d) Detections of the Model in Experiment 1



(e) Detections of the Model in Experiment 2

Fig. 12: Classes: Green → plane, brown-red → large-vehicle, white → small-vehicle.



(a) Target

(b) HTD Instances in Red

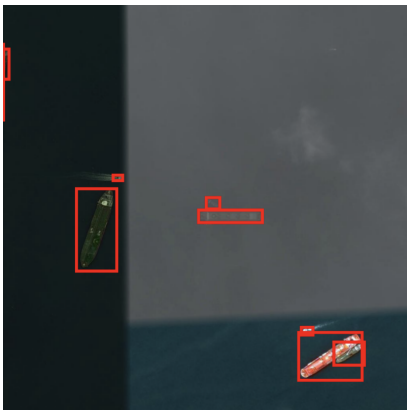
(c) Detections of the Baseline Model

(d) Detections of the Model in Experiment 1

(e) Detections of the Model in Experiment 2

Fig. 13: Classes: Light blue → bridge, brown-red → large-vehicle, white → small-vehicle.

(a) Target



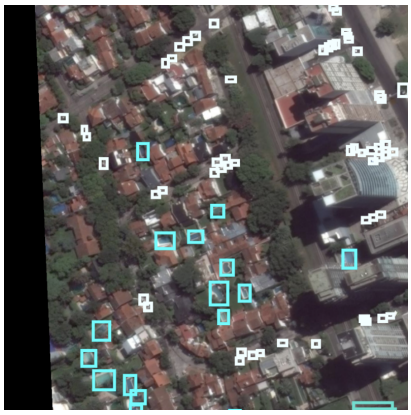
(b) HTD Instances in Red

(c) Detections of the Baseline Model

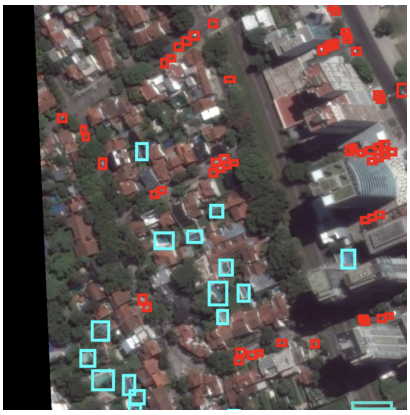
(d) Detections of the Model in Experiment 1

(e) Detections of the Model in Experiment 2

Fig. 14: Classes: Blue → ship. This is a good example for HTD instances due to different contrasts.

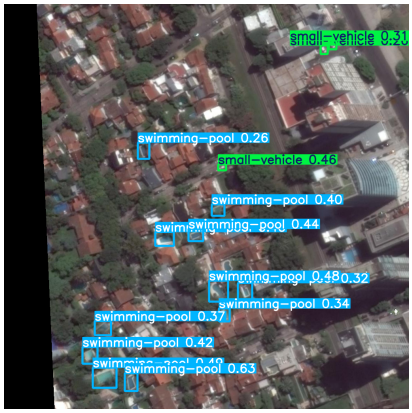


(a) Target

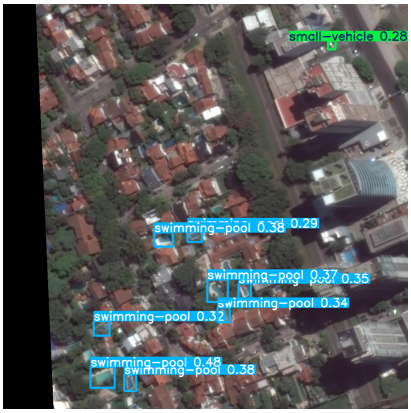


(b) HTD Instances in Red

(c) Detections of the Baseline Model



(d) Detections of the Model in Experiment 1



(e) Detections of the Model in Experiment 2

Fig. 15: Classes: Blue → swimming pool, white → small-vehicle.

F. TRAINING REPORT

F.1. Baseline

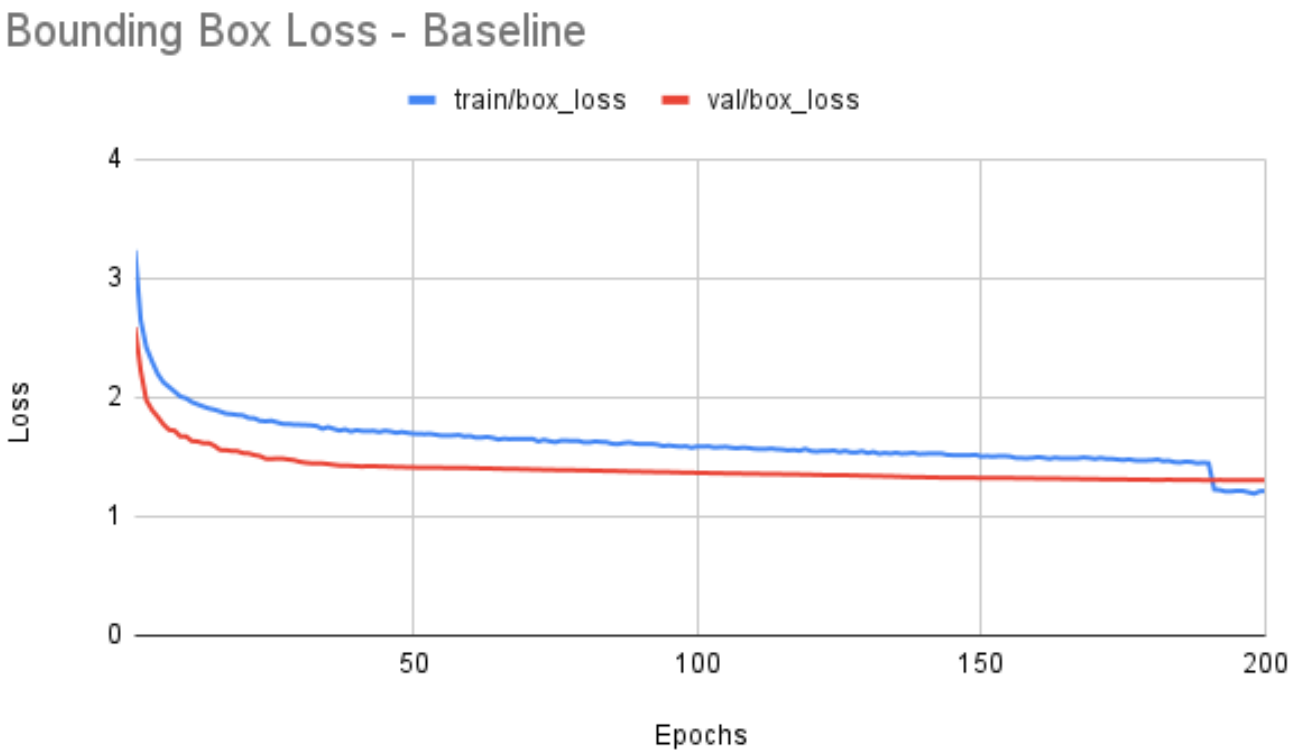


Fig. 16: **Bounding box Loss on the Baseline model.** The blue line indicates the training loss and the red line indicates the loss of the validation data.

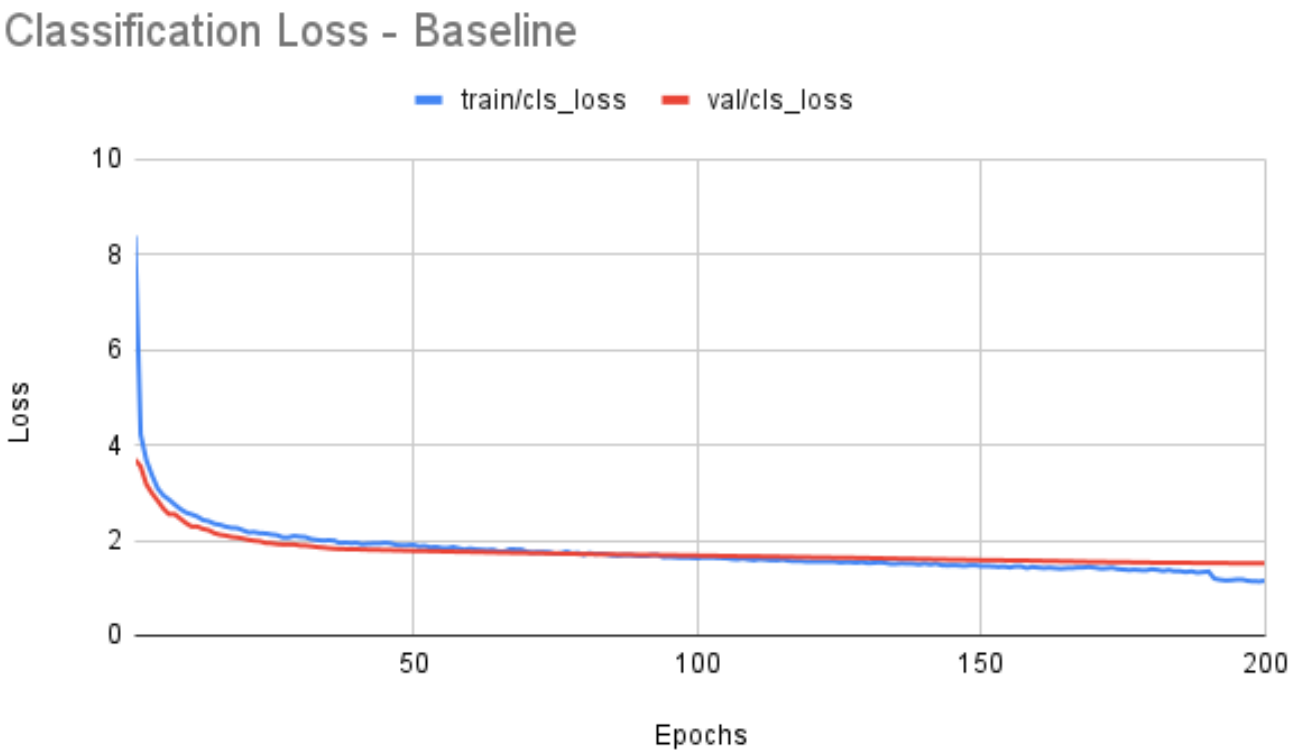


Fig. 17: **Classification loss on the Baseline model.** The blue line indicates the training loss and the red line indicates the loss of the validation data.

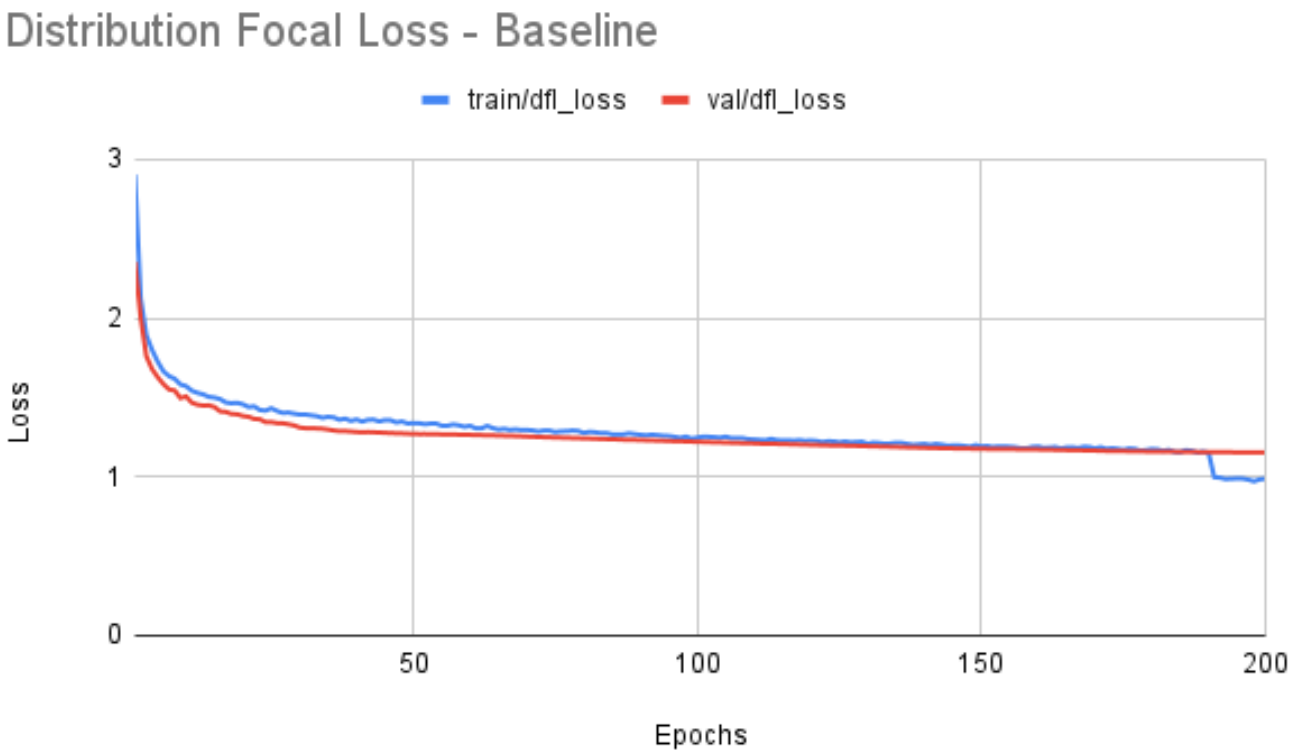


Fig. 18: **Distribution Focal Loss on the Baseline model.** The blue line indicates the training loss and the red line indicates the loss of the validation data.

F.2. Experiment 1

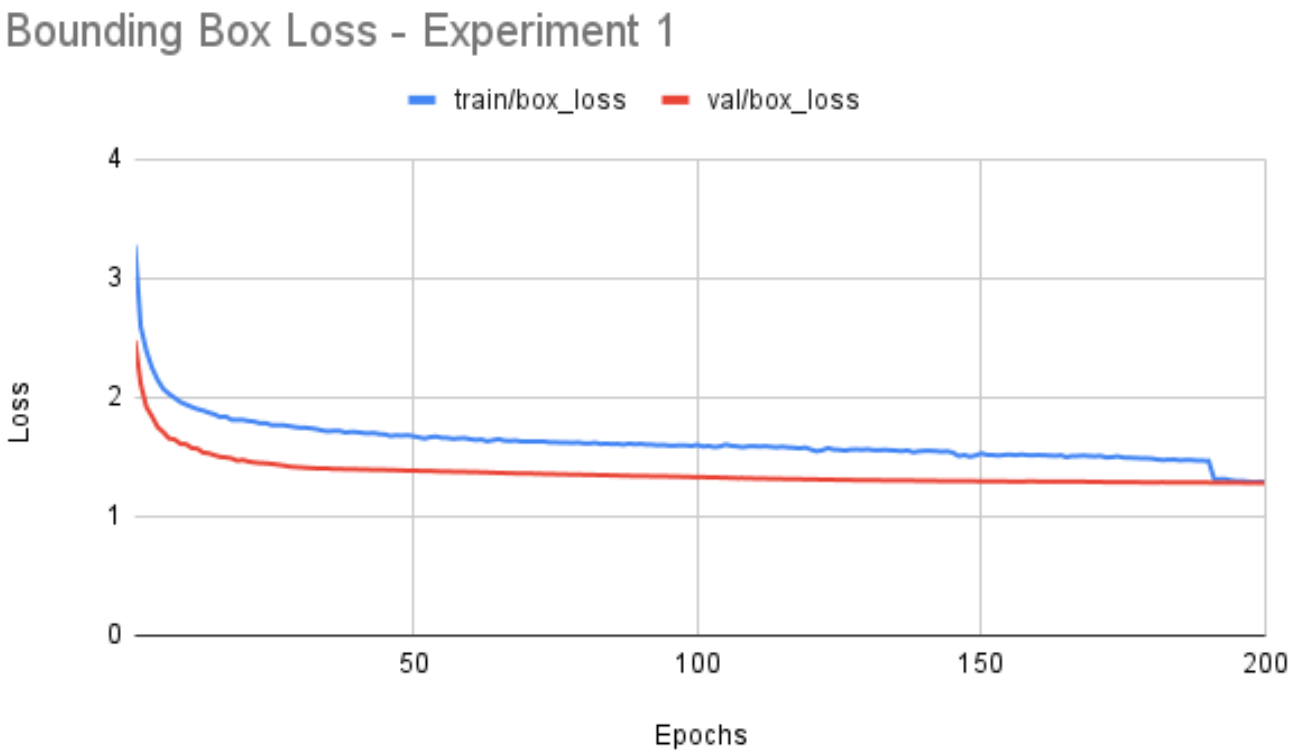


Fig. 19: **Bounding Box Loss on the Experiment 1 model.** The blue line indicates the training loss and the red line indicates the loss of the validation data.

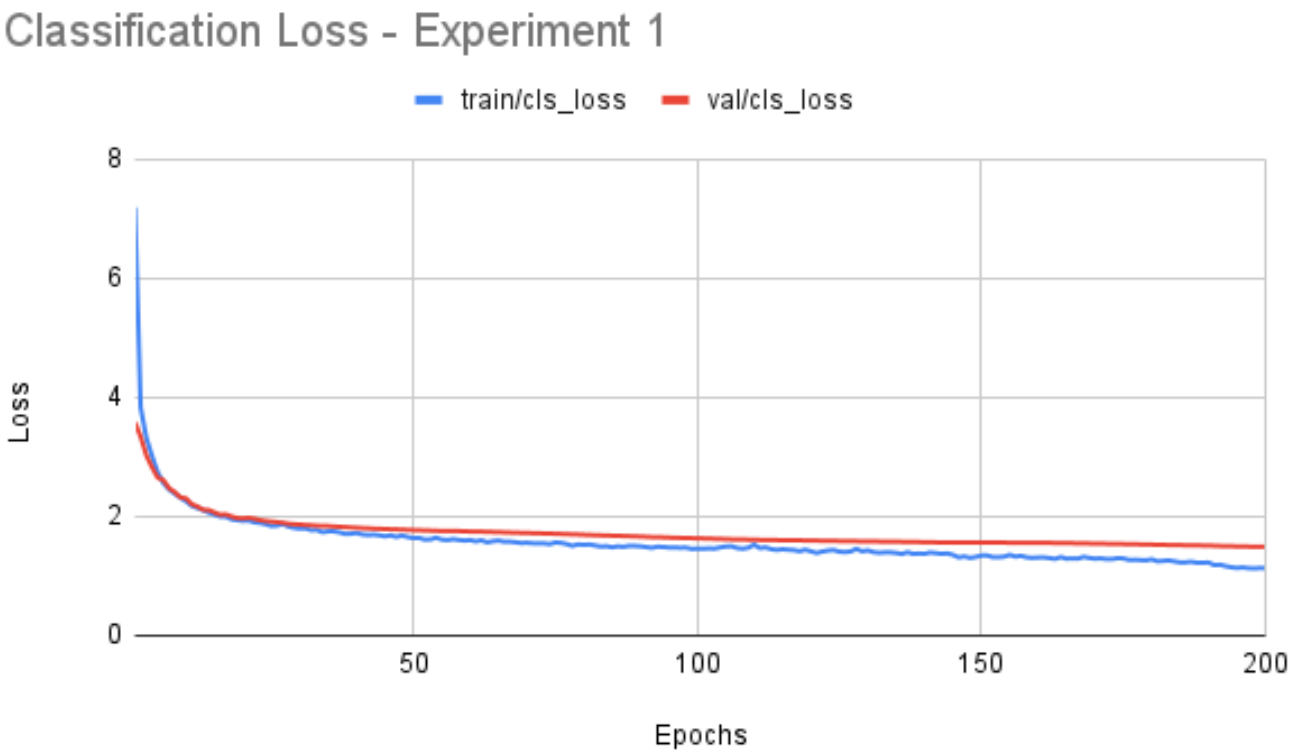


Fig. 20: **Classification Loss on the Experiment 1 model.** The blue line indicates the training loss and the red line indicates the loss of the validation data.

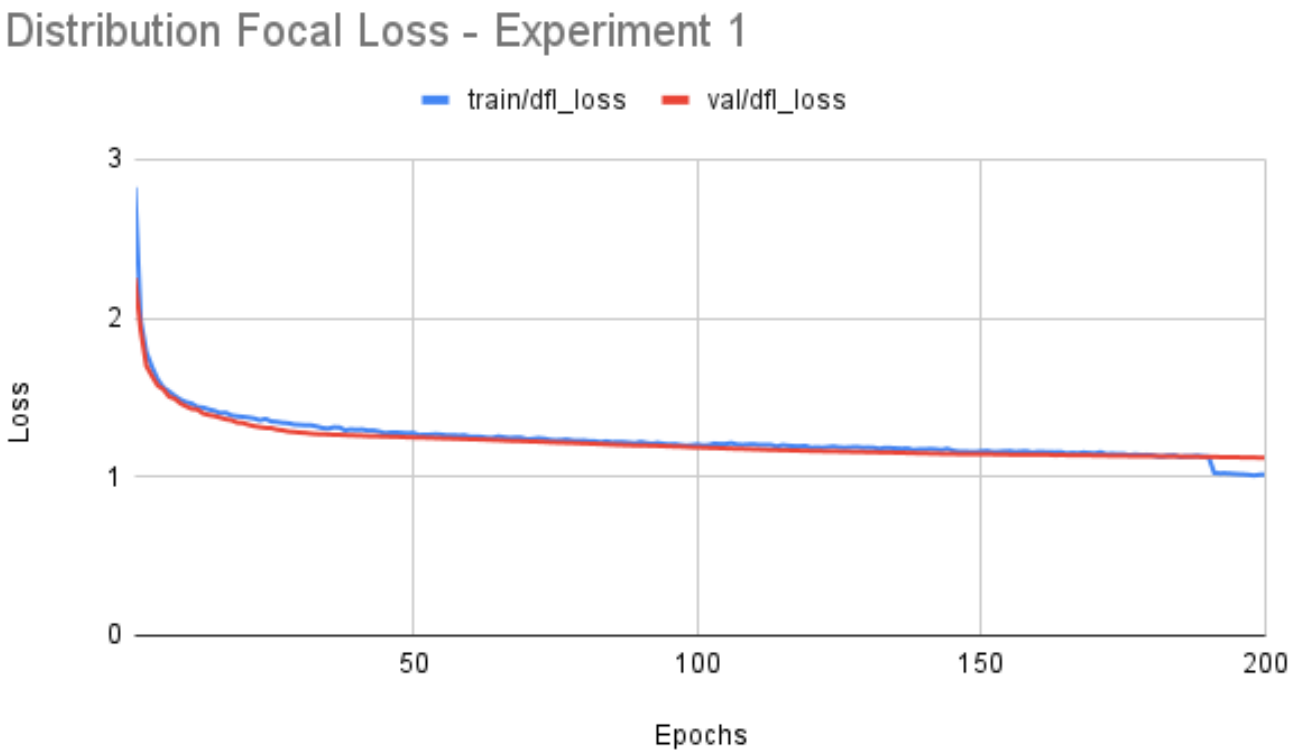


Fig. 21: **Distribution Focal Loss on the Experiment 1 model.** The blue line indicates the training loss and the red line indicates the loss of the validation data.

F.3. Experiment 2

Fig. 22: **Bounding Box Loss on the Experiment 2 model.** The blue line indicates the training loss and the red line indicates the loss of the validation data.

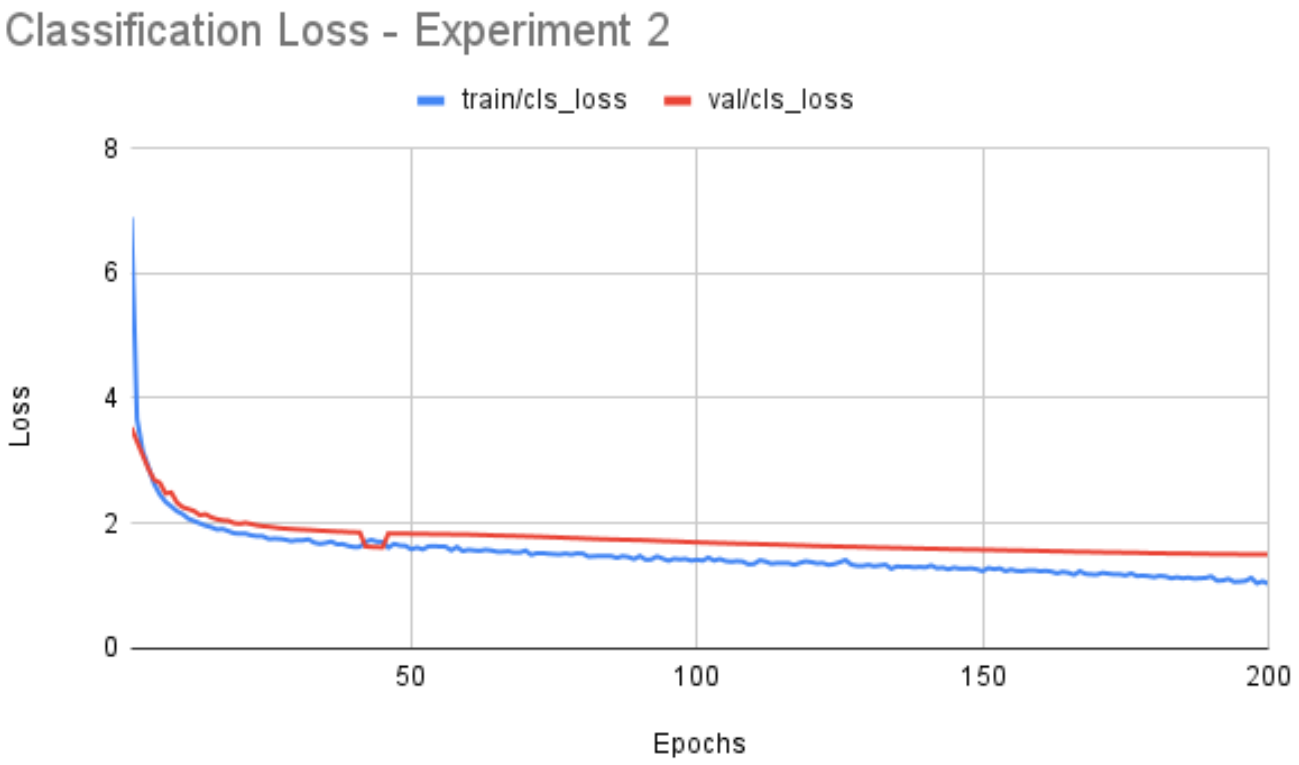


Fig. 23: **Classification Loss on the Experiment 2 model.** The blue line indicates the training loss and the red line indicates the loss of the validation data.

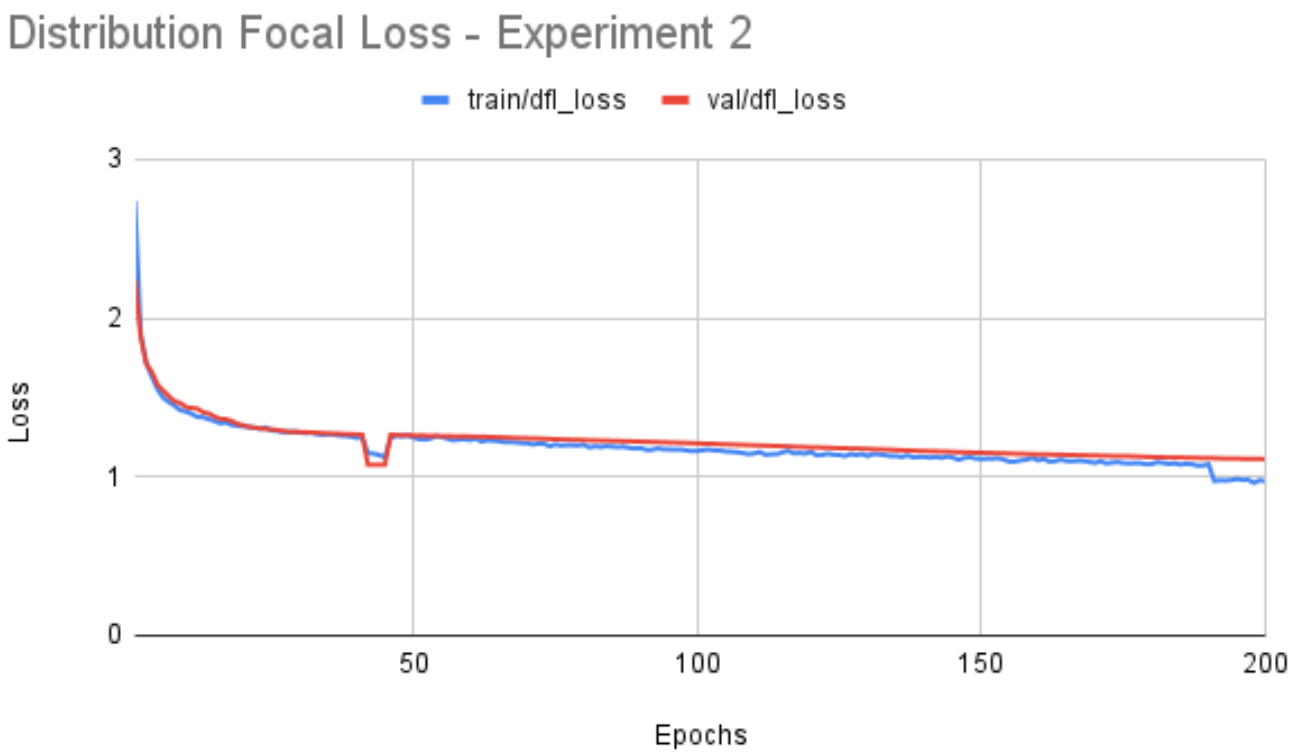


Fig. 24: **Distribution Focal Loss on the Experiment 2 model.** The blue line indicates the training loss and the red line indicates the loss of the validation data.

F.4. Mean Average Precission

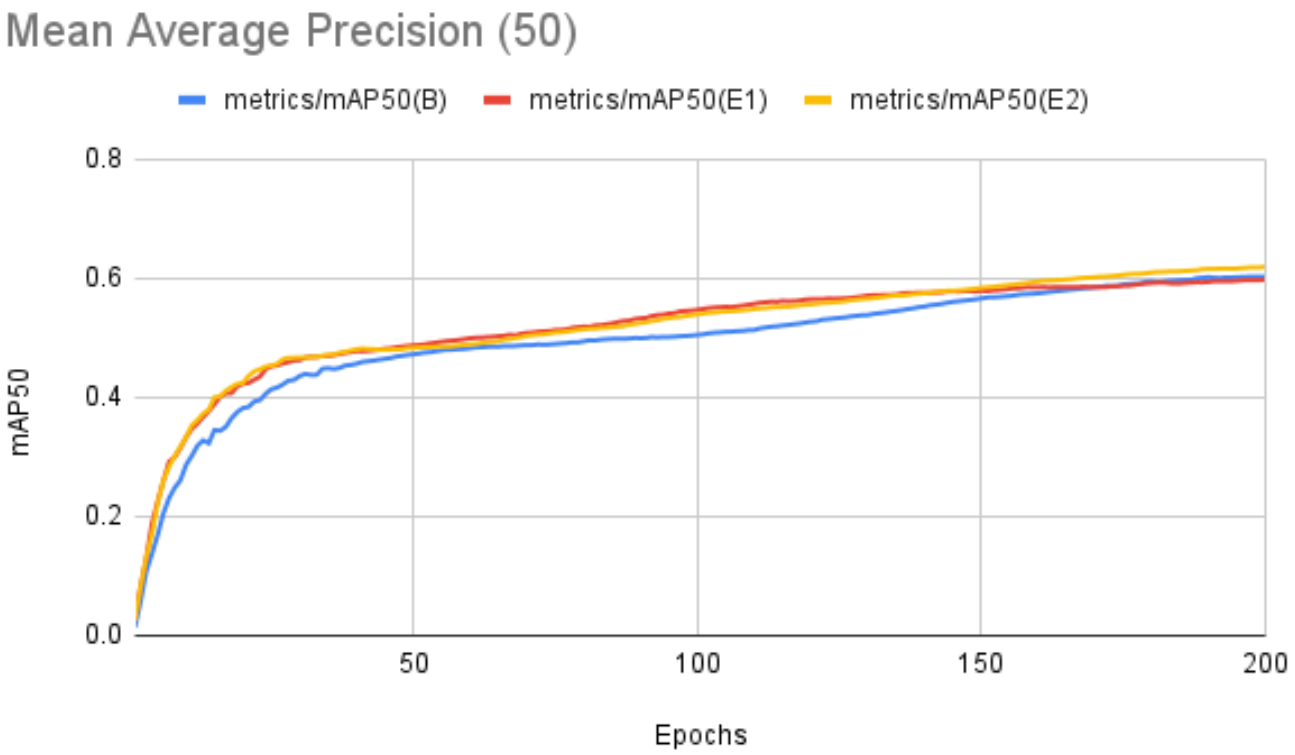


Fig. 25: **Mean Average Precision of all three models.** The blue line indicates the baseline, the red line indicates the first experiemtn and the yellow line represents the second experiment.

F.5. Learnig Rate

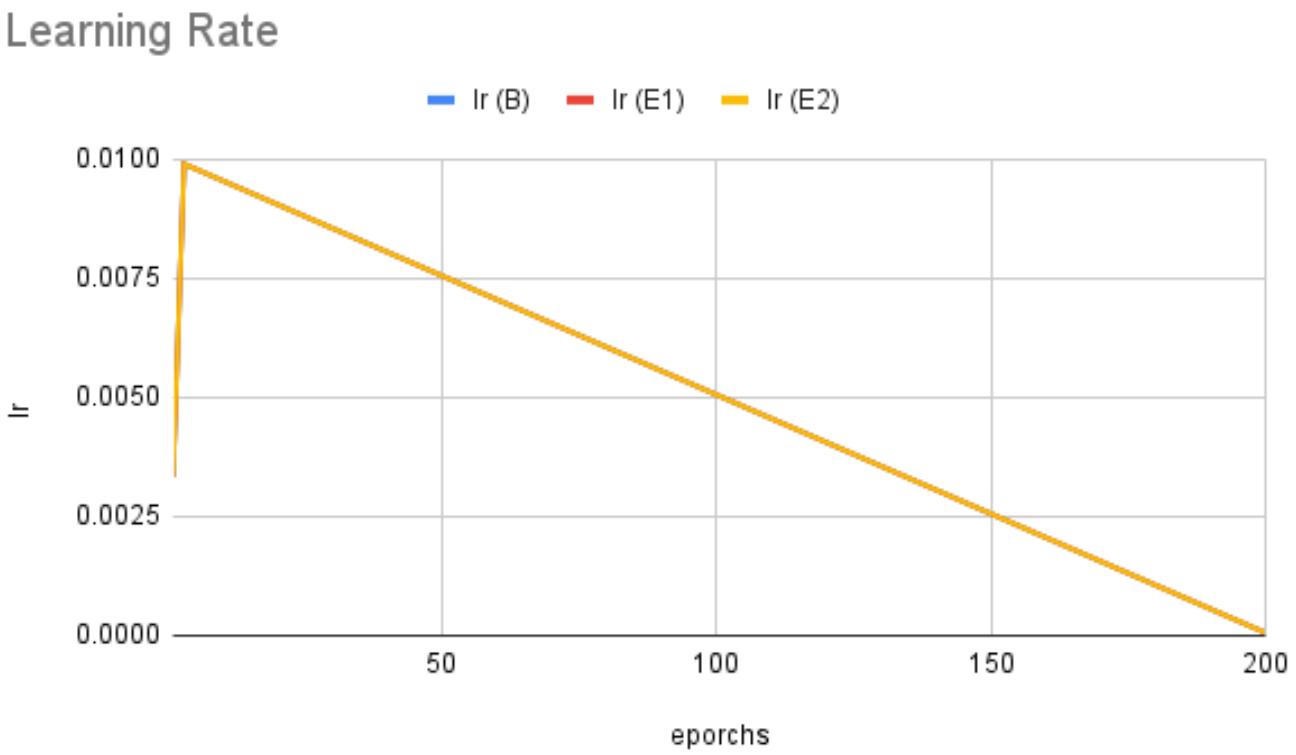


Fig. 26: **Learning rate of all three models.** The blue line indicates the baseline, the red line indicates the first experiemtn and the yellow line represents the second experiment.