Object Detection in Historical Film Shots

Giuseppe Tripodi[1]

***Abstract*— The analysis and understanding of historical images is a process that can enable us to gain more information about past events and increase our knowledge of them, in the past years the interest in this field increased a lot. In this paper, the execution of several Object Detection algorithms on historical data is presented, those methods will be used to detect different type of vehicles in the images. The aim of the work is to understand how certain models behave in situations in which the images have the classical defects of historical images. Well-proven models in the field of object detection such as YOLOR and YOLOX will be compared with a modern version of RetinaNet that replaces the traditional CNN with a new convolutional structure.**

I. INTRODUCTION

A. Motivation and Objectives

Object detection [8] is already a very complex task, the aim of which is to classify and recognize certain objects in particular images. When it comes to detecting objects in historical images, the task becomes more difficult: the images have many flaws and imperfections and the quality is rather low. However, the task is particularly interesting, the main reason is that understand which objects are present in the image can be a valuable factor in understanding what is happening in the scene, and in general also in the whole film from which the frames are taken. Moreover, on images of the Second World War, this process becomes even more important as it allows for a greater understanding of events.

B. Scope and objective of paper

The scope of this work is to work on the data so that it can be used to do object detection, and after that to test different algorithms on this data, compare the results and see which one gets better results. The objective of this investigation is to detect some vehicles in the images, as: tank; car; truck; boat; airplane; horse trailer; train; motorcycle.

C. Methods and Results

This paper proposes a comparison of the following state-of-the-art methods:

YOLOR [11] is a state-of-the-art deep learning algorithm for object detection, different from the rest of the YOLO family architecture, and model infrastructure. YOLOR stands for "You Only Learn One Representation", is a unified network to encode implicit and explicit knowledge together. YOLOX [3] is an anchor-free version of YOLO, that has show better performance than the other YOLO methods.
Retina Net [14] is a one-stage detector, that introduces a new loss function and a Feature Pyramid Network (FPN) to reach the accuracy of the most advanced two-stage method.

D. Organization of paper

The paper is organized as follows: after a brief description of the models used and their characteristics, the dataset used and its characteristics will be discussed. After that, the methods by which the techniques were compared will be described, and finally, the results obtained will be presented.

II. METHODOLOGY

The choice of methodologies was guided by the desire not only to test reliable methods that are widely used in practice, but also to try to use more recent techniques that, although they have some shortcomings, are an interesting starting point for future work.

A. YOLOR - You Only Learn One Representation

Human being have the ability to process data and information using different point of view and prospective, the connection of those different prospective can help the human to improve the understanding of the world. YOLOR [11] try to repeat this human ability and use different knowledge to do learning.

Two types of knowledge can be identified, *Implicit knowledge* refers to knowledge learned in a subconscious state, while *explicit knowledge* is knowledge learned in a higher level. YOLOR [11] was proposed to unify and integrate implicit and explicit knowledge. We can see in the Figure 1 below a generic YOLOR architecture that mix together implicit and explicit knowledge.

B. YOLOX

YOLOX was created as an improvement of the YOLO series, the basis was YOLOv3 [9], however in the creation of the model some training strategies were modified, in addition the network head was changed.
YOLOX decouples the YOLO sensing head into different feature channel. Each of the decoupled Head is for a different kind of output, in order we have: box co-ordination, regression and object classification. In the Figure 2 we can see the difference in the head between a YOLOv3 [9] model and YOLOX [3].

C. RETINA-NET

RetinaNet [14] is a one-stage object detection model that uses a focal loss function to deal with class imbalance. The network is mainly composed of three parts: 1. blackbone network (ResNet) for feature extraction; 2. feature pyramid network (FPN) that is able to reorganize feature sub-networks, classification and regression to obtain the final detection results. A structure of a sample RetinaNet [14]

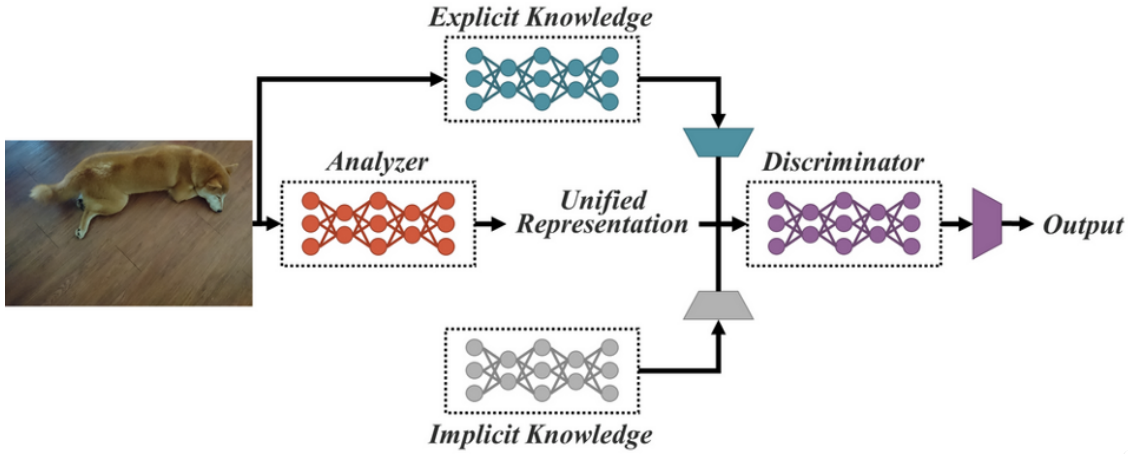


Fig. 1. YOLOR's unified network. The combination of implicit and explicit knowledge is used to generate a discriminator capable of classifying boxes.

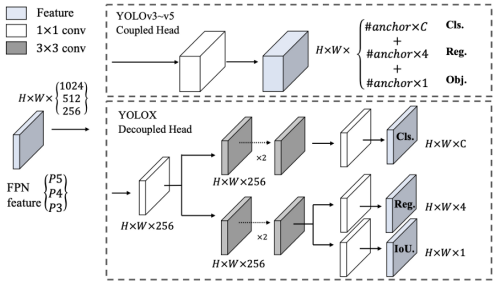


Fig. 2. YOLOX [3] architecture. It is clear how the head is separated to generate different outcome

network can be seen in the Figure 3.
However, the network used to improve the performance of a classical network makes some modifications to the classical structure.

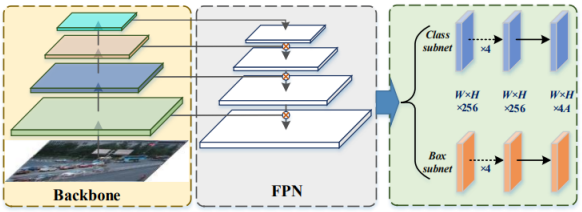


Fig. 3. Base Retina net architecture

In order to improve the performance of the backbone network, the traditional CNN was replaced by an octave convolution structure [2]. This convolution structure solves the problem of the classic CNN network and is able to consider both distribution of high frequency and low frequency. Moreover, instead of using a classical FPN, a weighted pyramid network (WPN) was proposed. This WPN enables the network to adaptively determine the main contribution point of the backbone feature map.

III. DATASET

The dataset was provided by the amount of data used for the Visual History of the Holocaust project VHH [10]. The dataset is part of a larger dataset in which the images are taken from footage relating to the liberation phase of the Nazi concentration camps, produced by the forces of: United States, Great Britain and Russia.
The images depict very different scenes, ranging from war-like images to formal meetings between politicians.

A. Dataset Preprocessing

However, considering that the aim of the work is to recognize only certain objects within the images, namely: tank; car; truck; boat; airplane; horse trailer; train; motorcycle.
Therefore, of the almost 7,000 initial images, after an initial skimming phase in which images not containing objects of interest were removed, approximately 700 remained. Images not containing any objects were filtered, but also images with very low quality.

Some examples of images that have been considered are show in Figure 4. While some of the images removed are showed in the Figure 5.

After the screaming phase, the labelling phase of all images was performed. Image labelling was performed using the software VoTT (Visual Object Tagging Tool). Figure 6 shows an example of a labelled image.

B. Dataset considerations

The labelling phase highlighted certain characteristics of the dataset in question. The number of labels assigned is 1079 against a number of images of 634 (see Figure I). As can be seen from Figure 7, the labels are not evenly distributed. For example, the greatest number of labels are those relating to the classes 'airplane', 'car' and 'tank', while the classes 'motorbike' and 'train' are poorly represented in the data.

This unequal distribution obviously also affects the results, as will be shown later on there are classes that are better

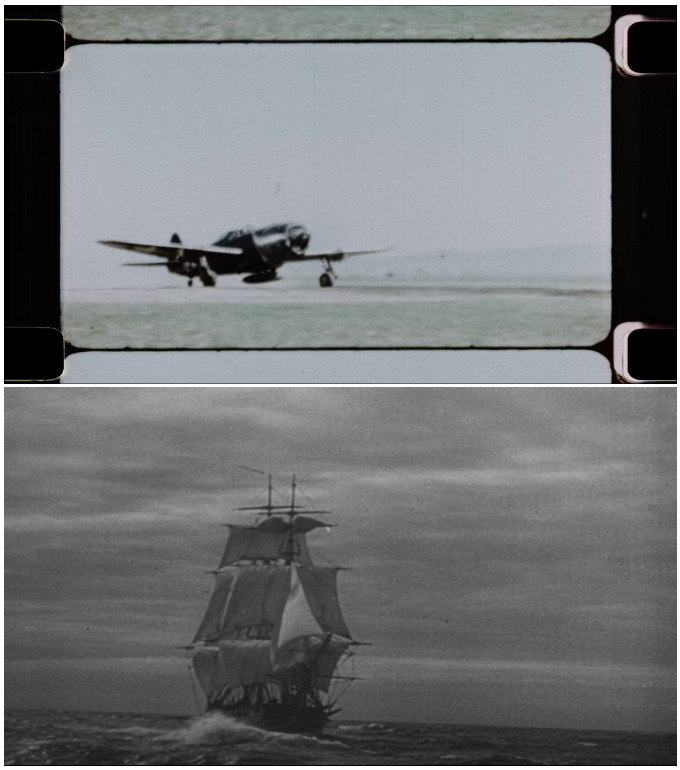
[b]  [b]

Fig. 4. The figure shows some examples of interested and usable images for our object detection task.

Fig. 5. The figure shows some immagin that cannot be used for our object detction task. The top figure contains no vehicles of interest, while the bottom figure shows nothing.

Fig. 6. Example of labelled image

TABLE I
TOTAL NUMBER OF IMAGES AND LABELS

Images	Labels
634	1079

recognized by the models while others are more difficult to predict.

C. Dataset Distribution

The dataset has been divided into three set: **training**, **validation** and **test**. The distribution of the sample in each dataset can be seen in the Table II:

IV. EVALUATION

The techniques used to compare the different models will be highlighted in this chapter. The main concepts on which the models were compared are essentially the following ones: **Mean Average Precision** [5], **Training Time**.

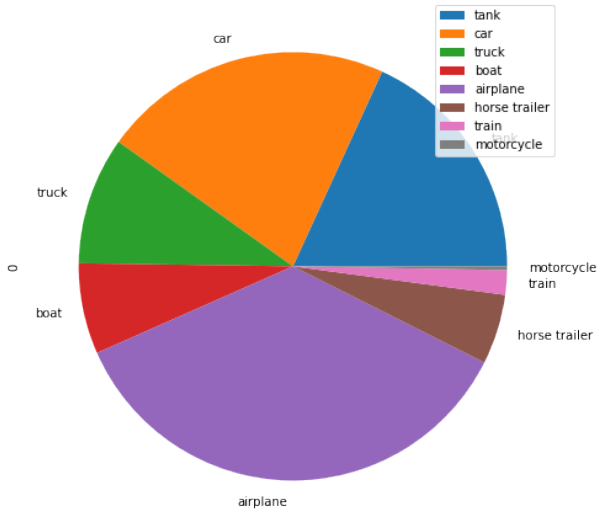


Fig. 7. class distribution

TABLE II
DATASET SPLITTING. THE DATASET WAS DIVIDED IN THREE DIFFERENT SET

	Dataset		
	Training Set	Validation Set	Test Set
Number of Images	448	167	180
Number of Labels	611	215	253

A. Evaluation Background

Before enumerating the metrics used, some background information will be provided to better understand the following metrics.

- **True Positive TP** is the outcome when the model *correctly* predict the positive class
- **False Positive FP** is the outcome when the model *incorrectly* predict the positive class
- **False Negative FN** is the outcome when the model *incorrectly* predict the negative class

These measures are used to calculate:

- **Precision** is the confidence of the algorithm to predict a positive value.

$$PRECISION = \frac{TP}{TP+FP}$$

- **Recall** is the percentage of the positive sample recognized by the method.

$$RECALL = \frac{TP}{TP+FN}$$

Another metrics to take into consideration is the **Intersection over Union (IoU)** [12]. This metrics measures the overlap between two boundaries. It is used to measure how much our predicted boundary overlaps with the ground truth. Sometimes is used a **IoU threshold** in classifying whether the prediction is a true positive or a false positive.

$$IoU = \frac{areaofoverlap}{areaofunion}$$

B. Metrics

Average Precision AP [13] is the area under the precision-recall curve:

$$AP = \int_0^1 p(r)dr \tag{1}$$

This measures combines recall and precision for ranked retrieval results.

Average Recall AR [1] describes the area doubled under the Recall $\times$ IoU curve. This curve plots recall results for each IoU threshold where IoU $\in [0.5, 1.0]$, with IoU thresholds on the x-axis and recall on the y-axis:

$$AR = 2\int_{0.5}^1 recall(IoU)\mathrm{d}IoU \tag{2}$$

Mean Average Precision mAP [6] is the average of AP for each class, is defined as:

$$mAP = \frac{1}{N}\sum_{i=1} AP_i \tag{3}$$

TABLE III

THE TABLE SHOW THE PARAMETER USED TO TRAIN THE MODELS.

Parameter			
	YOLOR	YOLOX	RetinaNet
Epochs	300	300	300
batch size	2	4	2
weight decay	0.0005	0.0005	0.0005
learning rate	0.01	0.05	0.001
momentum	0.937	0.9	0.9

where N is the total number of classes.

Average forward time [7] is the time it takes to propagate one batch of inputs from the input layer to the output layer. So for each batch of inputs is detected the timing to propagate the batch from to input nodes to the output nodes, and then the mean for the number of batches is computed.

$$AFT = \frac{1}{N}\sum_{i}^{N} \text{forward time}_i \tag{4}$$

Where N is the number of batches, and $forward\_time_i$ is the time to propagate the batch i from the input nodes to the output nodes.

Average inference time [7] is how long it takes for forwarding propagation during the testing. This metric is computed on the test set, so after the model was trained. Therefore, after we fixed the weight, we compute the time it takes to propagate one batch of the test set from the input to the output.

$$AIT = \frac{1}{N}\sum_{i}^{N} \text{inference time}_i \tag{5}$$

A **qualitative analysis** of the results obtained was also carried out in order to gain a broader view of the results of the methods.

V. RESULTS

A. Training Details

It should be noted that all models and tests carried out in this work were performed on Google Colab [1].

The parameter used to train the models will be defined in this chapter, and will be summarized in the Table III

As we can see, all the models were trained for the same number of epochs. Of relevance is the choice of batch size; the choice of this parameter was basically constrained by the machine on which the models were run. Due to a lack of enough space in the GPU, it was impossible to run models with larger batch sizes. However, it is thought that this factor still penalized and influenced the results obtained.

The Table IV shows the time needed to train the models on all images for 300 epochs.

It can be seen that in the machine used, the execution times are very large indeed. This is attributable both to the less than ideal performance of the machine used, but also to the volume of images with which the models were trained. However, it can be seen that in terms of execution times, the model that preforms better than the others is YOLOX [3].

TABLE IV

THE TABLE SHOW THE NEEDED TO TRAIN THE MODEL FOR 300 EPOCHS ON ALL IMAGES

	YOLOR	YOLOX	RetinaNet
hours	9	8	12

TABLE V

METRICS COMPARISON

Model	mAP@.5	mAP@.5:.95	AF time (ms)	AI time (ms)
YOLOR	0.473	0.388	46.4	45.1
YOLOX	-	-	47.09	47.68
RetinaNet	0.5037	- 0.5038	-	-

Both YOLOR [11] and YOLOX [3] were trained from pre-trained weights. For both models, in fact, this was the choice recommended by the respective authors, and *yolor-s* [2] weights were chosen as the weights to train the models.

B. Quantitiative Analysis

A comparison of the result can be see in the Table V and in the Table VI.

It can be seen from the outset that the various networks do not return the same metrics, this is due to implementation choices made by the authors. A cross-analysis was therefore carried out, considering several metrics.

Before analyzing the results, it is useful to specify the details by which the various metrics were calculated. In fact, the values in square brackets or after the metrics name (i.e. $[IoU = 0.50]$) correspond to the IoU threshold values that were used to classify an example as positive or negative.

Regarding timing (Average Inference Time (AIT) [7], Average forward time [7] (AFT)) can be seen from Table V as the fastest model is YOLOR [11]. Despite the missing data from RetinaNet [14], one can easily assume considering that the runtimes are much larger than the others that AIF and AFT are also higher than the other two models.

On the other hand, as far as mAP [6] values are concerned, it is considered to have achieved fairly good results. Suffice it to say that the mAP [6] obtained on the COCO [4] dataset is around 50%, values that were also achieved by the RetinaNet network [14]. Also in this case the result from YOLOX [3] are missing, however, considering the result of Table VI can be easily assumed that the mAP [6] are lower that the other two methods, since also the AP is much lower that the other two approaches.

The results obtained show quite clearly that the model that guarantees better results is RetinaNet [14], this network in fact guarantees very high AP [13] and AR [1] values, not only compared with the other two models but also with

[1] https://colab.research.google.com/?utm_source=scs-index

[2] https://drive.google.com/file/d/1WyzcN1-I0n8BoeRhi_xVt8C5msqdx_7k/view

TABLE VI
METRICS COMPARISON, PART 2

Model	AP [IoU=0.50:0.95]	AP [IoU=0.50]	AR [IoU=0.50:0.95]
YOLOR	-	-	-
YOLOX	0.108	0.281	0.195
RetinaNet	0.504	0.967	0.540

Fig. 8. This image shows the behavior of YOLOR on a batch. In this case, it manages to classify all vehicles correctly.

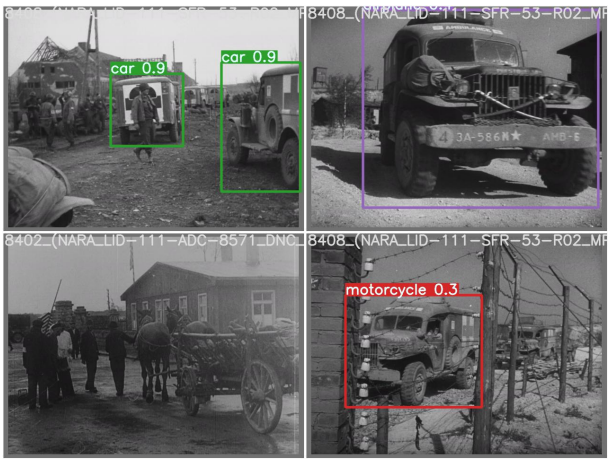


Fig. 9. The figure shows some of the problem of YOLOR, in fact the model usually miss the detection of the classes *car* and *horse trailer*

models trained on a much larger number of data (i.e. COCO [4]).

C. Qualitative Analysis

Considering that, all in all, the models perform relatively well, actually analyzing the behavior of the algorithms on test images makes it possible to assess their behavior in practice.

1) YOLOR: Despite its still satisfactory performance, this is the model that errs most often when classifying vehicles. In Figure 8 an example of correct prediction can be seen, while in Figure 9 the difficulties of YOLOR [11] are show.

In general, however, a certain frequency of the model in not correctly recognising or misclassifying horse trailers and motorbikes is noted. This characteristic, common to more or less all models, is probably due to the lack of these labels in the images.

2) YOLOX: This network together with RetinaNet is the one that returns the qualitatively best results. Although the recognized boxes are indeed larger than the object itself, the assigned class as can be seen in Figures 10, and 11 are correct.

This result is quite surprising considering the fact that the metrics for this model are rather low.

3) RetinaNet: Of the tested models, RetinaNet is definitely the one that returns better results, it can recognize all vehicles frequently. We can, for example, see in Figures 13, 12 how the model even manages to recognize the *horse trailer* and *truck* classes correctly. The only problem with this model is that it uses more anchor boxes than necessary to recognize the object, in fact we can see more boxes on the object.

Fig. 10. This Figure shows an example of YOLOX correct prediction of class *car*. This class is usually well recognized by the model.

VI. CONCLUSIONS

The object detection task on historical images has always been very complicated, particularly for this specific task as in most of the images the objects to be recognized were not clearly distinguishable, very often they were only half present in the images and sometimes there was also occlusion, with a person for example in front of the object to be recognized. Nevertheless, the results obtained are quite good, all three models performed quite well, and in most cases they easily managed to classify the objects in the images.

The model that guarantees the best results is undoubtedly **RetinaNet** [14], as this model not only has higher metrics

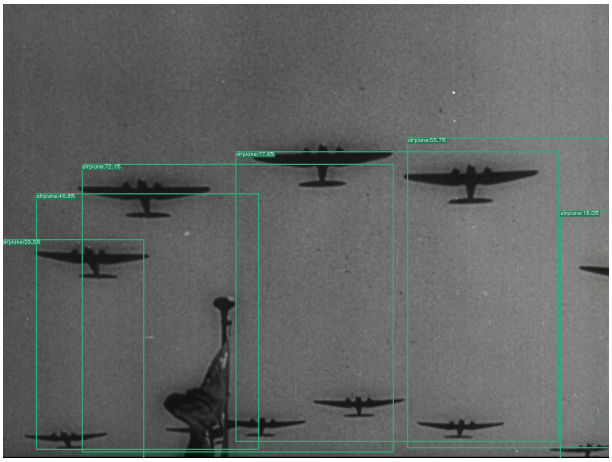


Fig. 11. This Figure shows an example of YOLOX correct prediction of class *plane*.

than the others but also better quality results. Almost all classes are recognized correctly, even those that are scarcely present in the dataset. The only disadvantage of this network is the unbelievably long execution times; this problem too, however, can be remedied by using more powerful machines. It must be added, however, that the images used to train the models were relatively few in any case; it is assumed that with a larger number of images and better labelling quality, even better results can be achieved.

REFERENCES

[1] B. Carterette, *Precision and Recall*. Boston, MA: Springer US, 2009, pp. 2126–2127. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_5050

[2] Y. Chen, H. Fan, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, S. Yan, and J. Feng, "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution," 2019. [Online]. Available: https://arxiv.org/abs/1904.05049

[3] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," 2021. [Online]. Available: https://arxiv.org/abs/2107.08430

[4] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014. [Online]. Available: https://arxiv.org/abs/1405.0312

[5] L. LIU and M. T. ÖZSU, Eds., *Mean Average Precision*. Boston, MA: Springer US, 2009, pp. 1703–1703. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_3032

[6] ——, *Mean Average Precision*. Boston, MA: Springer US, 2009, pp. 1703–1703. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_3032

[7] S. S. Ogden and T. Guo, "Characterizing the deep neural networks inference performance of mobile applications," 2019. [Online]. Available: https://arxiv.org/abs/1909.04783

[8] F. Porikli and A. Yilmaz, *Object Detection and Tracking*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 3–41. [Online]. Available: https://doi.org/10.1007/978-3-642-28598-1_1

[9] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

[10] E. Union. (2022) Visual history of the holocaust project. [Online]. Available: https://www.vhh-project.eu/

[11] C. Wang, I. Yeh, and H. M. Liao, "You only learn one representation: Unified network for multiple tasks," *CoRR*, vol. abs/2105.04206, 2021. [Online]. Available: https://arxiv.org/abs/2105.04206

[12] J. Xu, Y. Ma, S. He, and J. Zhu, "3d-giou: 3d generalized intersection over union for object detection in point cloud," *Sensors*, vol. 19, no. 19, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/19/4093

[13] E. Zhang and Y. Zhang, *Average Precision*. Boston, MA: Springer US, 2009, pp. 192–193. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_482

[14] L. Zhang, H. Wang, X. Wang, S. Chen, H. Wang, K. Zheng, and H. wang, "Vehicle object detection based on improved RetinaNet," *Journal of Physics: Conference Series*, vol. 1757, no. 1, p. 012070, jan 2021. [Online]. Available: https://doi.org/10.1088/1742-6596/1757/1/012070

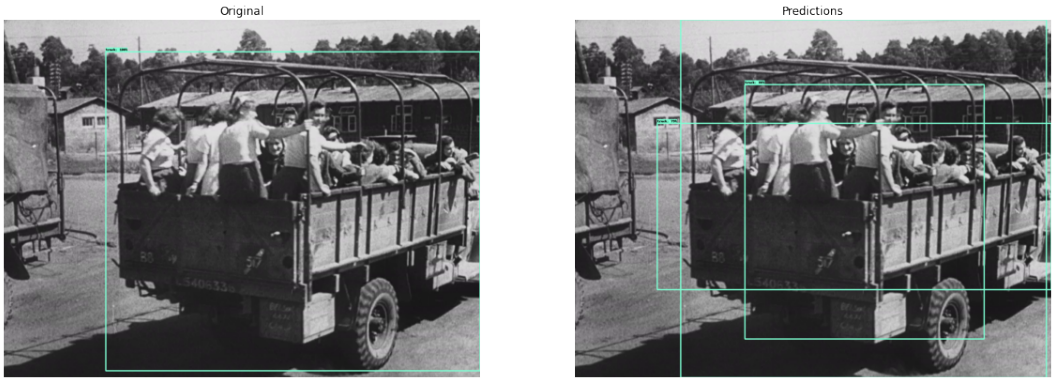


Fig. 12. The figure shows a RetinaNet correct prediction of the class *Truck*. On the left side we can see the original image with the right label, while on the right side we can see the prediction done by RetinaNet.

Fig. 13. The figure shows an example of RetinaNet correct prediction of the class horse trailer. On the left side we can see the initial image with the right label, while on the right side we can see the prediction done by RetinaNet. We can see that the object is well classified.