Final Report: Logo Detection Project

Course 20600 DEEP LEARNING FOR COMPUTER VISION, Bocconi University
Instructors: Gaia Rubera, Francesco Grossetti

Written by team Paco: Alessandro Caruso, Fabian Kraus, Silvia Juzova, Louis Lacombe, Riccardo Tordini, Steffen Brockmann

Content

1. Project Description

1.1 What is object detection?

Object detection is a technique from computer vision describing the training of neural networks models in order to teach them how to identify and locate objects in an image or video. With this kind of identification and localization, object detection can be used to count objects in a scene, determining and tracking their precise locations while accurately labeling them.

Object detection relies on a specific feature: bounding boxes, which describe the spatial region, through x and y coordinates, of an object. In fact, in object detection it's necessary that the trained models are able to predict rectangular boxes, with their relative coordinates, which identify objects and label them. The models used for this task are divided in 2 big classes: 2-stages and 1-stage detectors. The difference between them is that the first family generates region proposals and learns the features from each region, without any intermediates, while the second classifies each region. In order to understand what's in an image, the models will be fed through a standard convolutional network to build a rich feature representation of the original image. This part of the architecture will be referred to as the "backbone" network, which is usually pre-trained as an image classifier to learn in a cheaper way how to extract features from an image. Thus, the models will be trained on the labeled dataset in order to learn good feature representations.

1.2 The logo detection problem

This detection problem is based on popular brand logos. Based on a training and a noise set of 17 classes, with prespecified and labeled bounding boxes, a subset should be extrapolated from which the models will be trained. Obviously, some preprocessing steps have to be taken before starting the training. For example, data has to be augmented, in order to add a perturbed version of the same image in the dataset and increase the number of training data, for exposing our network to a wide range of variations of the same image. In the end, the data has to be resized and rescaled: the first is a rule for feeding models, while the second is helpful for reducing computational efforts.

Regarding the training phase, it's known that creating and training a model from scratch leads to poorer models: the better way to work on this problem is to adopt transfer learning. In fact, working with pre-trained and pre-designed models leads to better predictions in a shorter time.

An object detection model can be evaluated from different points of view. In the first place, the predicted bounding boxes can be true positives (TP), false positives (FP) or false negatives (FN). Starting from these metrics, we can carry further analysis: in fact, other indicators for the models are precision, recall and mAP, which synthesizes the goodness of the model through a score based on TP, FP and FN. In this case, the main metric for

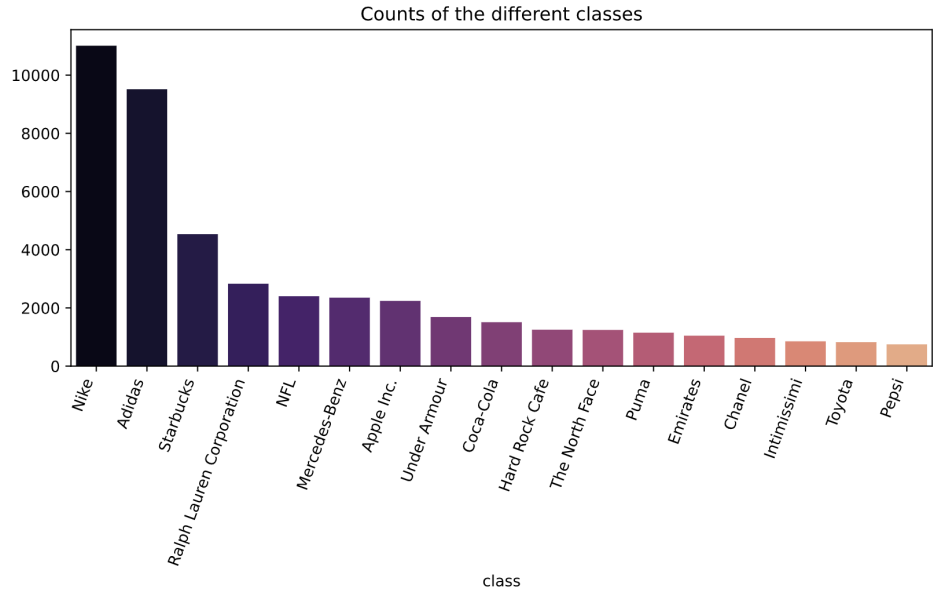
evaluating the models will be the Intersection over Union (IoU), which, given a true bounding box and a prediction, is defined as intersection of the 2 boxes divided by the union of them: the more accurate the prediction is, the closer the IoU will be to 1. Obviously, not every prediction of the models is considered, especially since 1-stage detectors produce many predictions. The focus is just in the logo detection. In this case, it will be stored just the highest IoU related to the prediction with the highest level of confidence.

2. Preliminary Data Analysis

The dataset provided consists of a set of 46,163 labeled instagram images as provided in the annot_train.csv with an additional 2,286 noise images as contained in the annot_noise.csv file. Before continuing to carry out several different preprocessing steps such as resizing and data augmentation, a preliminary analysis on the dataset was conducted to understand any specific areas of concern or issues with the data beforehand.

An important concern for any classification task is the issue of class imbalance. Underrepresented classes may lead to poor performance of the algorithm on the test set, particularly if such imbalance does not hold in the test case. Indeed, the number of images across the 17 different logos in the dataset varies quite considerably as shown in **Figure 1**.

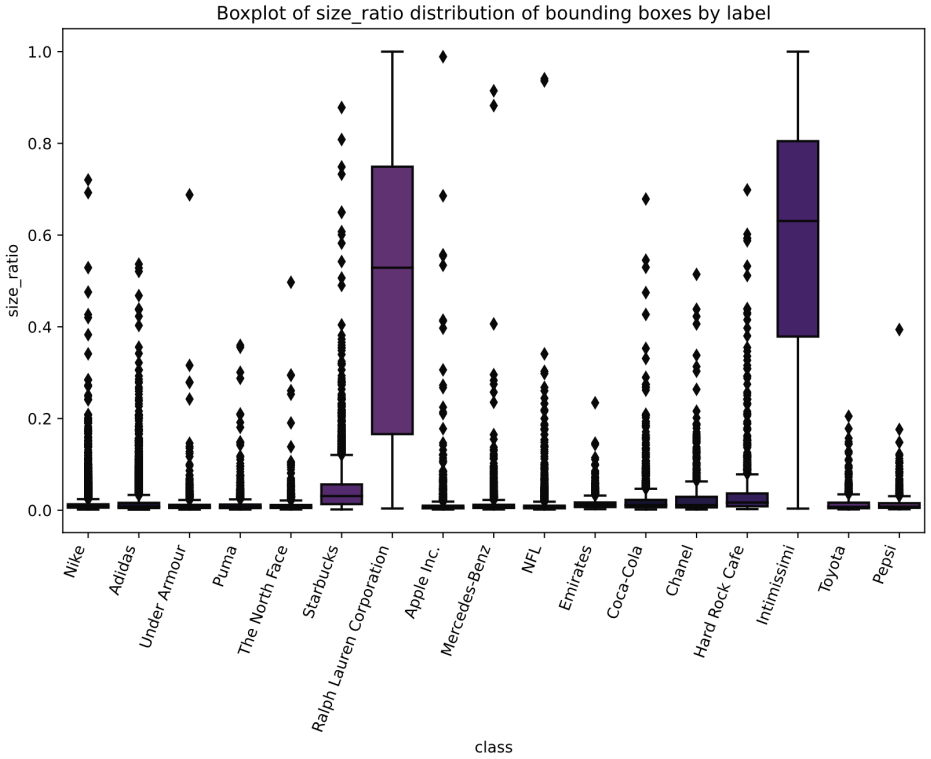
Figure 1: Counts of different classes in the original dataset



As it can be observed, there are significant differences in the class distribution, in particular Nike (11,011) and Adidas (9515) are severely overrepresented compared to the remainder of classes presented in the dataset. Of the required labels, Puma consists of the least data with just 1149 labels, only 1/10th of the images that include Nike as logo. Hence, some ways of dealing with these imbalances are discussed further in the preprocessing and model analysis sections.

Furthermore, the sizes of bounding boxes with respect to the overall image size were analysed. The intuition here is that the annotated images, being instagram posts, should not contain too large Bounding boxes, as people are taking pictures of themselves with the respective logos contained somewhere as part of such image, rather than an image of the logo only. Hence, the ratio of bounding box to image size was plotted, per class, to identify any anomalies, as shown in Figure 2.

Figure 2: Boxplot of bounding box size with respect to image size by label in original dataset



As shown in the Figure, Ralph Lauren Corporation and Intimissimi display a significantly different distribution of their bounding boxes with an average ratio of more than 0.5 compared to a close centering around 0 for the remaining logos. Indeed, a quick analysis of the images classed as Ralph Lauren or Intimissimi shows that the bounding boxes are wrong and neither of the two labels are actually contained in these images. Hence, the two labels are dropped from further consideration. As the images do not contain true logos in them, the images can be further utilised as noise dataset to ensure the correct prediction of background images. Furthermore, any bounding boxes with a ratio larger than 0.4 are deemed unlikely to be a viable bounding box and hence their annotations are removed. It should be noted that, while the distribution of Starbucks is also slightly higher than those of other logos, it is not considered to be a problem. The reason is that Starbucks images generally focus more around Starbucks, i.e. people posting pictures of drinking from Starbucks cups and hence the bounding boxes should be somewhat larger.

Due to the existing class imbalance it was further decided to remove some of the more infrequent labels. In particular, any of the logos with less images than Puma were removed from the dataset entirely. Hence, 11 classes remain for the analysis, excluding the background class.

Moreover, the annot_train.csv file contains several more image file names that are not found in the image folder. Of the 46,163 annotations, only 38,914 images are actually provided in the folder. Finally, the remainder of images is split into subfolders of roughly 10,000 images that are subsequently loaded into Roboflow to carry out several preprocessing and augmentation steps.

3. Preprocessing

The preprocessing can be divided in mainly two parts: the first being a manual adjustment of all the bounding boxes provided and the second instead with the choice of augmentation steps (with the auxiliary of Roboflow functions).

3.1 Manual Adjustment

Given the peculiarity of the dataset provided and the increasing numbers of wrong examples detected while exploring the dataset, it has been chosen to manually correct each picture (45k images!) and adjust the bounding boxes coordinates, eliminating those images with wrong annotations, and adjust errors in labelling. Moreover the huge number of duplicates in the dataset has been eliminated, in concordance to the memory skills of the agent performing the cleaning.

In particular, in most of the cases, the truth bounding boxes were actually too large compared to the real logo, or inconsistent in capturing the right format. (e.g. for very similar pictures, some boxes were capturing only the logo, some cases only the company name, other cases both).

Moreover, sometimes the true bounding boxes of logos such as Apple, were actually pointing to completely other objects, such as Youtube Like buttons. This suggests that these data are not the result of manual labelling, but of a statistical learning procedure where only one logo is kept and the others are discarded..

Through manual adjustments, a boost in terms of predictive performance has been given to the statistical models, which can now deal with a more organized and appropriate training dataset, reflected in higher performance when computing the final IoU by class. However it has been acknowledged that it would have been optimal to not only adjust the already existent bounding boxes, but also manually add all the bounding boxes present in an image, resulting in a multi-object detection task (and not one-object detection).

The choice to only correct the already labeled ones has been made due to lack of time-resources and to not deform the task assigned, which was evaluated on a one-logo per

image detection basis. However, for the purpose of object detection, it is strongly recommended to entirely relabel the dataset for further research.

3.2 Augmentation Steps

Apart from resizing the image to the desired number of pixels, different augmentation steps have been taken. In particular, as common practice in object detection tasks, each image has been flipped both horizontally and vertically. In this way, the number of images where logos appear in non-standard situations can increase and enable the statistical procedure to generalize better on unseen images.

As a next step, both hue and saturation were applied. In practice, hue is the color in the image and saturation is the intensity, or richness, of that color. Varying these features enables the creation of different representations of the same image (encoded as RGB tensors), and also creates variations from the standard color of logos, enabling a more accurate extrapolation.

In addition, the exposure of the image inside the bounding box has been chosen to vary randomly between -40° and +40° to create the right balance in both training and test set, given that logos are in many cases emphasized in images, by being the object of interest of the camera that is taking the picture. The aim of this augmentation choice is to stabilize these cases and achieve a balanced level of penalization/emphasis for the role that logos play in an image.

In terms of which preprocessing steps have been avoided, it can be clearly stated that any form of partial rotation was excluded, since many logos were positioned at the corners of images. In those cases, if the rotation was applied, the logo would have just been excluded from the image entirely.

Finally, for logos that had a high frequency in the dataset, such as Adidas and Nike, a x2 augmentation has been decided. On the other hand, for the other logos, given their minority, a x3 augmentation has been instead decided.

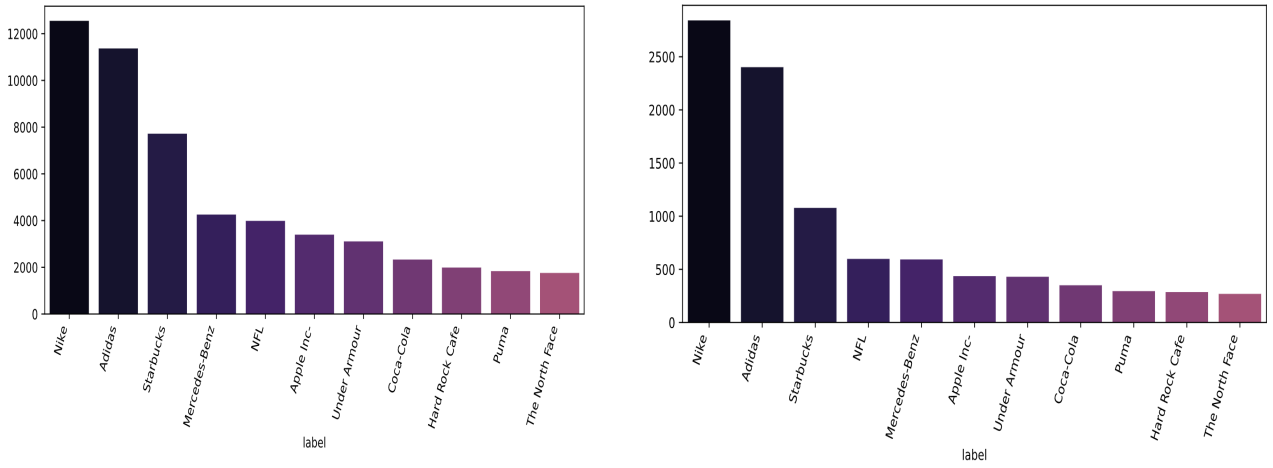
3.3 Post-Preprocessing Analysis

Due to the limit of 10,000 images per project in Roboflow, it was required to merge back the datasets together for the different models. In particular, two different dataset formats were generated, COCO and YOLO v5 PyTorch format respectively whose images and labeling were merged back together utilising the data_merger.py script. Subsequently, a quick post-preprocessing analysis was undertaken on the data in COCO format as a sanity check of the data and ensure its correctness.

The removals of underrepresented logos or incorrectly labelled images as well as the increase of sample size for the remaining logos, resulted in a training dataset consisting of 57,800 images, of which 94% are labelled and the remaining 6% are classified as background.

The testset on the other hand consists of 9337 images, with a similar level of labelled data points. The distribution of data by logo for both train and test is shown in Figure 3 below.

Figure 3: Processed train dataset (left) and test dataset (right) by label



As can be shown the distributions are almost identical apart from the fact that Nike and Adidas seem to have relatively less data compared to the other labels in the train set compared to the test set. This is a result of the data augmentation, which was carried out less for those two logos than the other to reduce the existing class imbalance in order to provide somewhat less of a class imbalance.

Furthermore, an analysis on the bounding box ratios highlights no difference compared to the preprocessed data, highlighting that there was no preprocessing step that significantly distorted the boxes in any way, which could potentially lead to lower prediction accuracy. The plot of the bounding box distributions is shown in Figure 3 in the appendix.

Moreover, the relabelling of the bounding boxes highlighted a rather high % of images containing more than one logo. Any good object detection algorithm should therefore predict all instances of logos observed in a given image. While this is not a problem per se, it does present some issues in scoring the algorithm's performance, as several predicted logos will be receiving an IoU score of 0, due to the fact that the labels contain no more than one prediction per image. This issue seems to be particularly strong with Starbucks, Nike and Adidas, where up to 30% of all the images can contain multiple logo instances. In subsequent sections, detailed ways of highlighting this issue and some statistics on the frequency of occurrence will be provided.

4. Models

In this section, various of the models that were utilised for the Object Detection (OD) task were presented. In particular, OD models can be broadly classified into two categories: Two-stage and one-stage detectors. Two-stage detectors, such as Faster R-CNN used to perform better but do have the disadvantage of longer execution times compared to one-stage detectors. Hence, the performances of two algorithms, Faster R-CNN which is a two-stage detector and YOLOv5 which is a one-stage detector will be presented and compared. It should be noted that, while generally one-stage detectors such as YOLO do perform worse, YOLOv5 presents the current state-of-the-art of its class whereas the Faster R-CNN implementation follows an older built-in pytorch implementation which is utilised mainly for its improved transparency of the full training pipeline. Hence, it is expected for YOLOv5 to outperform Faster-RCNN in terms of both predictive performance and execution.

The models are subsequently evaluated by calculating the best matching predicted IoU with the true IoU for any predicted box with a confidence score above λ% (lambda). It should be noted that such evaluation cannot be used for the prediction of a true and unlabelled test set, as it typically cannot be assumed to know the true bounding boxes. This method is used to combat the understated performance of the algorithms, resulting from the fact that only one logo per image is actually labeled in the datasets, whereas the models predict multiple correct logos per image. Hence, if selecting the predictions solely based on which bounding box has the highest confidence, the algorithm may have identified a different correct logo found on the image as the most accurate. A deeper insight on such instances will be provided to highlight that this is indeed the case. Consequently, the confidence score λ% is chosen in such a way that most bounding boxes with a confidence score higher than the threshold are actually identifying correct logos.

Finally, it is proposed that for testing on new data, all predicting boxes with a confidence higher than the threshold λ% are utilised. In order to highlight the correctness of this classification method, an additional 800 images where all instances of logos in an image are labelled are utilised and scored by mean IoU per class.

4.1 Two-Stage-Detectors (Faster R-CNN)

In the following subsection, Faster-RCNN will be explained in detail, following an analysis of the results of predictions utilising this approach.

The pytorch built-in Faster R-CNN with feature proposal network is utilised for showcasing the predictive performance of a two-stage detector. More advanced models such as Retinanet and EfficientDet were also used but not presented in this report due to the strong transparency inherent in the Faster RCNN implementation by pytorch. Faster RCNN, when released, provided significant improvements on the previous state-of-the-art model Fast RCNN, in terms of both performance and speed, due to its inclusion of the region

proposals in the training of the network via the Region Proposal Network (RPN). In particular, the used implementation implements a Resnet50 backbone, whose outputs are fed into a Feature Pyramid Network, aimed at reducing the input dimensionality and thereby increasing the speed in later parts of training. The outputs of the FPN are then inputted into the RPNs and two stages of double classification of the logos and bounding boxes are finally performed to arrive at the final output.

The images for this model are resized to 800*800 pixels, representing the minimum requirement for the implementation at hand. The data was generated in COCO format with a 70/30%, train/test, split. It should be noted that due to the long execution times of about 3 hours per epoch, it was decided to not do any hyperparameter tuning by re-running the model and hence no validation set was generated. Instead, hyperparameters were tuned beforehand to suit well for logo classification based on previous research and an analysis on the aspect ratios of the bounding boxes in the set. Hence, the test set of 30% can be considered as the validation set to validate a good accuracy based on the metric provided, however the 600 extra generated images containing multiple logos should be considered as the real test set as resembling a model in deployment.

Faster R-CNN, when untuned, has the tendency to produce several dozen bounding boxes per image, as it is trained for object detection which can contain a large number of certain objects, such as people walking on sidewalks, etc. As the images at hand contain only a small number of logos and oftentimes even one, the parameters defining the outputted number of regions need to be tuned down to account for this. Specifically, the parameters *rpn_batch_size_per_image* and *box_batch_size_per_image* are tuned down to 36 and 24 respectively to directly reduce the number of bounding boxes found in each of the two regression rounds. Furthermore, several parameters that apply harsher Non Maximum Suppression (NMS), such as narrowing down the threshold for NMS consideration and the positive fraction selected. The resulting detection model should produce only logos it is able to reproduce with high surety and hence provide better scores for the task at hand.

Further important tuning parameters are the ones defining the sizes of the anchors which directly influence the sizes of the bounding boxes produced. In particular, the different aspect ratio distributions, calculated as height/width, of the bounding boxes per class are inspected to identify optimal ratios to input into the algorithm. An analysis on the aspect ratios of the original dataset showed a distribution between 0.2 and 1.5 with a tendency towards the left. Hence, a wide range of possible values in between those boundaries is selected with a stronger focus on ratios below 1. It should be noted that the inclusion of such a high number of aspect ratios does increase the execution time of the model.

The results of the model will be presented in conjunction with the YOLO results in Section 5.

4.2 One-Stage-Detectors (YOLOv5)

YOLO is an acronym for 'You Only Look Once' which already describes how this particular family of object detection models is different compared to R-CNNs. While 2-stage detectors like R-CNNs perform detection on different region proposals or utilize sliding windows, YOLO can be compared to a single neural network and predicts object classes and their corresponding bounding boxes globally for the full image in one run of the algorithm. In basic terms, YOLO first splits the image into cells (e.g a 16x16 grid) where each cell is responsible for predicting a fixed number of bounding boxes. For every cell YOLO determines the probability that it contains a certain class and subsequently the class with the maximum probability is chosen and assigned to the respective cell. Using a process called non-maximum suppression, bounding boxes that cover the same object and are close to each other are eliminated. Since YOLO treats all elements it wants to predict (center of bounding box, height and width of bounding box and object class) as a single regression problem, its loss function is designed to learn about all these parameters simultaneously. Thus this single network architecture is optimized end-to-end on detection performance.

A major advantage of YOLO is that the algorithm looks at the full image during training; it infers global contextual information about classes and their appearances. Another advantage of YOLO is that once trained, it is extremely fast and can detect objects in better-than-real-time.

Comparisons of YOLO (v3 and upwards) and Faster R-CNNs have shown that YOLO performs much better not only in speed but also accuracy. For example Srivastava, Shrey, et al. in "Comparative analysis of deep learning image detection algorithms." (Journal of Big Data 8.1 (2021): 1-27) compare the three major algorithm classes in object detection, sSingle Shot Detection (SSD), Faster Region based Convolutional Neural Networks (Faster R-CNN), and You Only Look Once (YOLO) . They find that SSD performance is much worse compared to Faster R-CNN for smaller objects, but that Faster R-CNN's accuracy comes at the cost of time complexity since it has to pass over a single image multiple times. Even though Faster R-CNN has improved over R-CNN and Fast R-CNN, the YOLOv3 model they use clearly has the highest accuracy and is even faster than SSD. In their paper "Comparison of Faster-R-CNN, YOLO, and SSD for Real-Time Vehicle Type Recognition" (2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), 2020, pp. 1-4,) Kim, Sung and Park compare the three models on a different object detection task and find that their YOLOv4 model is more accurate with better scores in precision and recall compared to the other two state of the art algorithms while being extremely fast.

For this project YOLOv5 which is the latest released version of the YOLO was used, utilising a 50/20/30 train/val/test split in the YOLOv5 PyTorch format, provided by Roboflow. The validation set is utilised for hyperparameter tuning, specifically with respect to the built in evolve option of yolo models.

YOLOv5 allows for additional image augmentation with some parameters. Examples for this are "degrees" which induce rotation in the training set of the images by x degrees or

"flipud" which will probabilistically horizontally flip the picture. Since many of these augmentations had already been performed, it was checked which additional ones were implemented by default in YOLOv5 to see if those would bring a lot of extra accuracy to the model. While some of the additional augmentation methods seemed to improve the performance when using the Nano models, the performance of utilising these on the superior X model actually did not improve, and hence it was decided not to use an additional augmentation beyond those outlined in Section 3.

Like many neural networks, YOLOv5 encompases a large set of training hyperparameters. For tuning them, it was decided to make use of the nano model, given its faster computational running time, and an overall similar architecture compared to the heavier models. Some of the hyperparameters that were extensively tuned include the learning rates or momentum, but also more image oriented parameters such as loss_bbox or anchor-multiple threshold. In particular, the former is a loss function for how tight the bounding boxes are compared to the true bounding box and the latter represents the min/max matching of anchor sizes to the box. Finally, it was found that, apart from removing the different augmentation steps, the default hyperparameter set provides among the strongest results with only negligible improvements by other specifications. Hence, a model with default parameters, but no additional augmentation is presented in Section 5.

5. Results

In this section, a complete analysis of the results of the two models, FasterRCNN and YOLO will be provided. In particular, an analysis on the optimal confidence threshold λ% is given. Subsequently, both the IoU by class using the prediction with the highest confidence score, as well as the best fitting above the threshold λ% are presented. To highlight that the difference in score results from predictions that represent other correct, but non labelled logos, we will manually go through all these instances and quantify the frequency of correct versus incorrect logos. Furthermore, a comparison of the scores between the models is provided and some of the shortcomings of each of the models are outlined. Finally the performance of the algorithms is evaluated on an additional test-set of X logos, including a total number of Y true logos, where all predictions with a confidence higher than the threshold λ will be chosen.

5.1 Analysis on λ%

In order to identify the optimal threshold for considering a prediction as valid, it is necessary to analyse the results in more detail. Particularly an optimal tradeoff between not losing correct predictions while not identifying too many logos needs to be found. Several different graphs are presented below to highlight this.

Firstly, the goal is to ensure that not too many of the predictions that represent the labelled logos are lost. For this reason, a boxplot of the confidence scores across different levels of IoU is presented in Figure 4 and Figure 5 below.

Figure 4: Distribution of predicted confidence over IoU thresholds >0.5, <0.5, 0 for Faster R-CNN

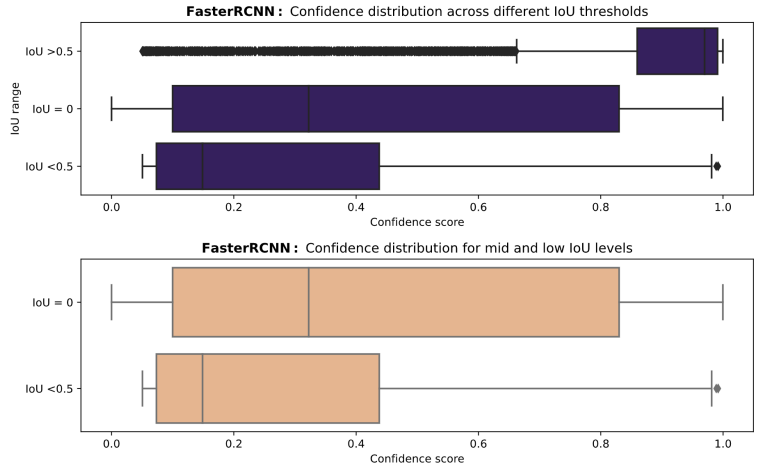
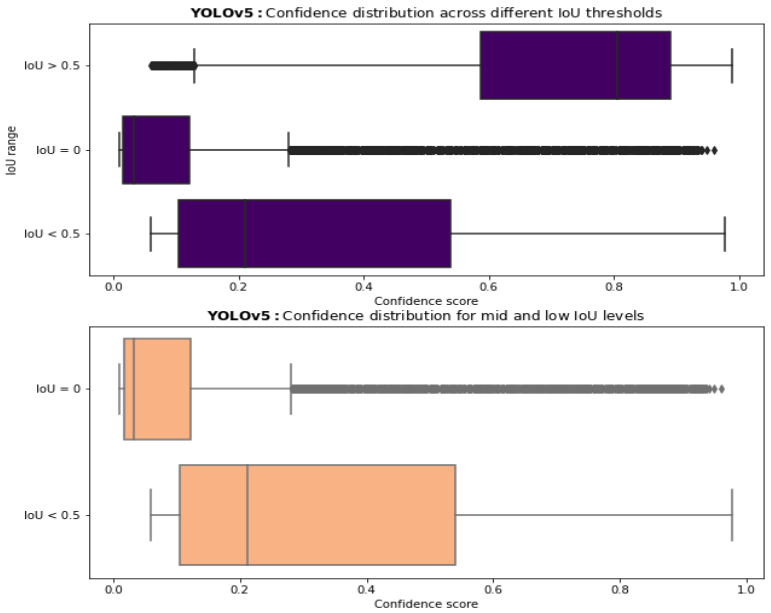


Figure 5: Distribution of predicted confidence over IoU thresholds >0.5, <0.5, 0 for YOLOv5



As shown in the Figure, there are quite some differences between the two models. Considering Faster R-CNN, it can be observed that the prediction confidence scores for well predicting boxes, i.e. with an IoU bigger than 0.5, are indeed much higher in distribution that the badly predicting, <0.5 IoU and none predicting, IoU = 0. A result which may seem striking at first is that, for Faster R-CNN, the predictions with an IoU of 0 seem to have a larger amount of images with a high confidence score. However, an explanation can be found in the fact that the 0-IoU group contains almost exclusively true logos that have not

been labelled in the dataset and hence should also have high confidence scores, as will be shown in detail later in this section. As it is important not to downgrade too many of the IoU <0.5 images to zero, a score that represents the lower quantile (Q1) of that bounding box has been chosen as a starting point for a more manual analysis on the images itself. In the case of FasterRCNN this threshold is identified as **7.4%.**

YOLOv5 on the other hand tells a completely different story compared to FasterRCNN. It can be immediately noticed that the confidence distribution scores for boxes with an IoU = 0 is instead centered around very low levels of confidence, suggesting that YOLOv5 is producing a lot of predictions where it is uncertain. However, YOLOv5 and FasterRCNN share a similar confidence distribution for well predicting boxes. Likewise, as for YOLOv5, it has been decided a confidence score such that well predicted logos, present in the images with IoU < 0.5 but not labelled, are not entirely excluded from the analysis. The score has been computed, again, by taking the lower quantile (Q1) of the confidence score distribution boxplot for images with an IoU < 0.5, resulting in **8%.**

At those initial confidence thresholds, the observed average number of predictions per image is 1.25 and 1.3 for Faster-RCNN and YOLO respectively. Unsurprisingly, YOLO predicts quite a few more predictions per label than Faster R-CNN. Both numbers are furthermore, within the area of what has been estimated to be the correct average number of labels shown by image, supporting the hypothesis that, at the chosen threshold of 7.34% and 8% respectively, the models predict a large number of true logos.

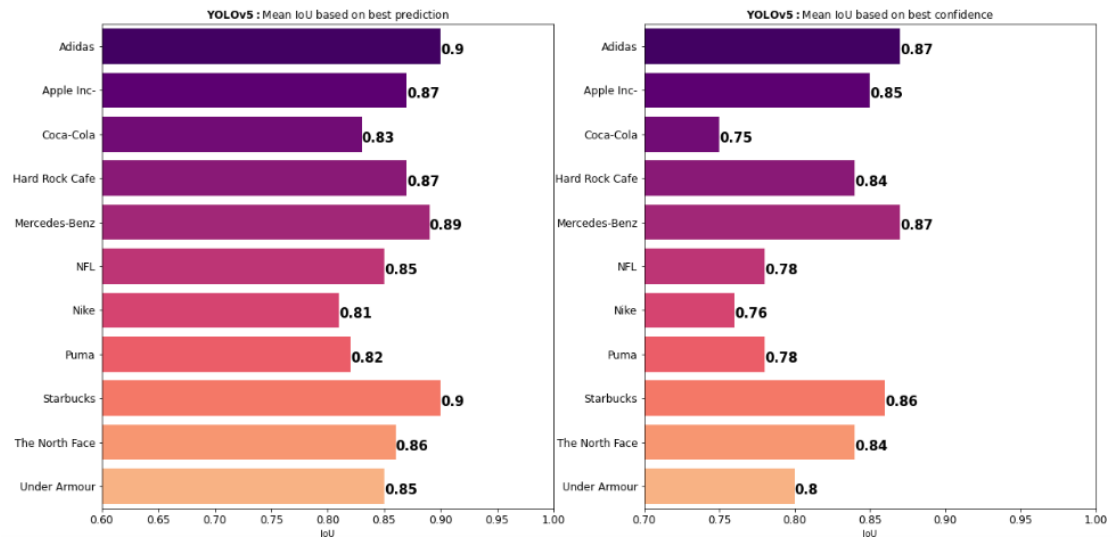
Finally, to confirm the two thresholds for Faster R-CNN and YOLO as a viable level for the parameter λ, 60 images for which there existed predictions with a threshold above 7.34% , but with an IoU of 0, were manually checked and the fraction of predictions with 0 IoU calculated. For FasterRCNN, 13.4% of the predictions represented such instances, whereas for YOLO the number stands at 18%. Given this relatively low number it has been decided that the chosen thresholds work well. Therefore, all predictions above the threshold λ = 7.34%, 8% for the respective models Faster R-CNN and YOLOv5 will henceforth be considered as valid bounding boxes and utilised in testing on unseen data.

5.2 Results Analysis

An analysis on both the scores when selecting the best IoU above λ% but also on the scores with the highest confidence per class will be undertaken to finally understate the reasoning for selecting the bounding boxes utilised in scoring. While the best IoU scoring method will be considered as the focus of the analysis, it will be shown that the largest fraction of the images, where a discrepancy between the prediction with highest confidence score and the highest IoU exists, are simply non labelled instances of correct logos. Hence, the graphs in Figure N show the performance of the performance of each of the models based on mean IoU per class in two cases:

1. Prediction based on highest confidence
2. Prediction based on best IoU predictions with confidence >λ%

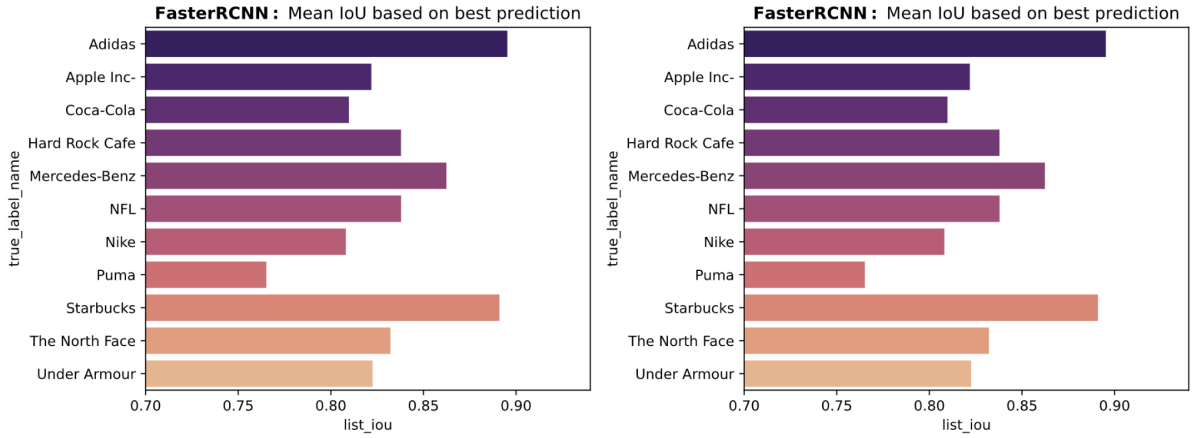
Figure 6: IoU based on best prediction with confidence >λ% and highest confidence by class for YOLOv5



As can be shown from Figures 6 & 7, both models perform quite strong with the scores of almost all the logos apart from Puma in FasterRCNN above an IoU of 0.8. Unsurprisingly the newer and generally considered better model of YOLOv5 is achieving higher IoU scores than Faster R-CNN, with each of the labels predicting between 2-5% better in the former case. Overall the results highlight strong performance across the different labels, in particular for Yolov5 with 8/11 logos above a mean IoU of 0.85 and the remainder above 0.8 (3/11 and 7/11 for Faster RCNN). This highlights the overall strong ability of the respective models in logo detection tasks.

At the same time, the logos Nike, Puma and Coca-Cola showcase significantly poorer performances compared to the remainder, with scores of 0.81,0.82 and 0.83 respectively. The lower performances of those labels is further shown when looking at the confidence score distributions across labels which are shown in Figure N in the appendix, whose distributions show a much stronger skew towards the left than the remaining labels.

Figure 7: IoU based on best prediction with confidence >λ% and highest confidence by class for Faster R-CNN



The low performance of Nike is somewhat surprising due to the large amount of training instances in the dataset. Typically overrepresented labels should perform higher as the algorithm has more opportunity to learn the unique features and is likely the reason for the great scores of Adidas and Starbucks. However, a closer look reveals that this issue can likely be explained by the fact that Nike is a shape with few unique features identifying the logo. Hence, the algorithm may confuse several other similar white shapes in images with Nike and consequently produce low confidences in the bounding boxes. Furthermore, specifically with the Faster R-CNN model, it results in a relatively high number of instances where the algorithm is not predicting the label at all, as it may be easily confusable with a simple line at the angle it is seen on the picture.

Furthermore, it can be observed from the mean results that there are considerable differences for both models in the predictions when taking the prediction with best IoU above the threshold λ% compared to the one with highest confidence. The difference is particularly high for some of the worse predicted logos, dropping as much as 7% in the case of Coca Cola. The subsequent analysis will show that this difference can however be explained almost entirely by instances of other correct logos which were not labelled in the images.

While, as outlined before, selecting the prediction with best IoU is not a valid method for predicting on an unlabeled test, it is utilised to highlight the performance of the models which would be understated through looking at the highest confidence. To showcase that the evaluation method is indeed the better indicator of the overall performance the following approach has been taken. By filtering all the instances where the confidence score of a prediction is higher than the confidence score of the prediction associated with the highest IoU, a set of images for the correct evaluation has been built. Upon manual inspection of all the images of this instance, the fraction of correctly predicted logos, even if not labelled, is calculated. An excerpt of 6 images randomly chosen from the set is provided in Figure 18 in the appendix.

The above filtering method leaves roughly 500 images for both Yolo and FasterRCNN and it was found that **91% and 89%** of the logos with a higher confidence were correctly identified logo instances for the models Faster R-CNN and YOLOv5 respectively, highlighting the fact that a majority of these predictions are true logos that have not been labelled. Hence, the method of selecting the prediction with a confidence higher than ƛ% that has the best IoU is considered the accurate method to showcase the mean IoU per class in the particular case in which the dataset contains only one of many labeled instances of logos. Nonetheless, to showcase the predictive power on a testset with multiple instances, the algorithms performances are subsequently tested on an additional testset, where all predictions above a confidence score of ƛ% will be considered as valid logos.

The slightly higher percentage observed for Faster R-CNN, while more inaccurate in overall performance due to not predicting some instances, does show higher confidence in the labels that it does predict and also has the tendency to predict less false bounding boxes. Hence, further research should aim to further understand whether the extra predictive performance provided by YOLOv5 in logo detection is worth the additional error in terms of badly predicted bounding boxes.

After analysing in detail the issue of predictions with 0 IoU, it is important to understand any further characteristics that display a systematic error in predictions of the bounding boxes for the respective algorithms. For instance, a common problem of the models are the bad predictions of small bounding boxes. Hence, Figure 8 highlights the differences in distribution of bounding box sizes across the different IoU scores. The class of IoU = 0 is not included due to the high frequency of true logo prediction included in the class, thereby not representative for bad logo predictions.

Figure 8: Correlation of bounding box ratio to image to IoU scores (left) and IoU > 0.5/ < 0.5 (right) for Faster R-CNN

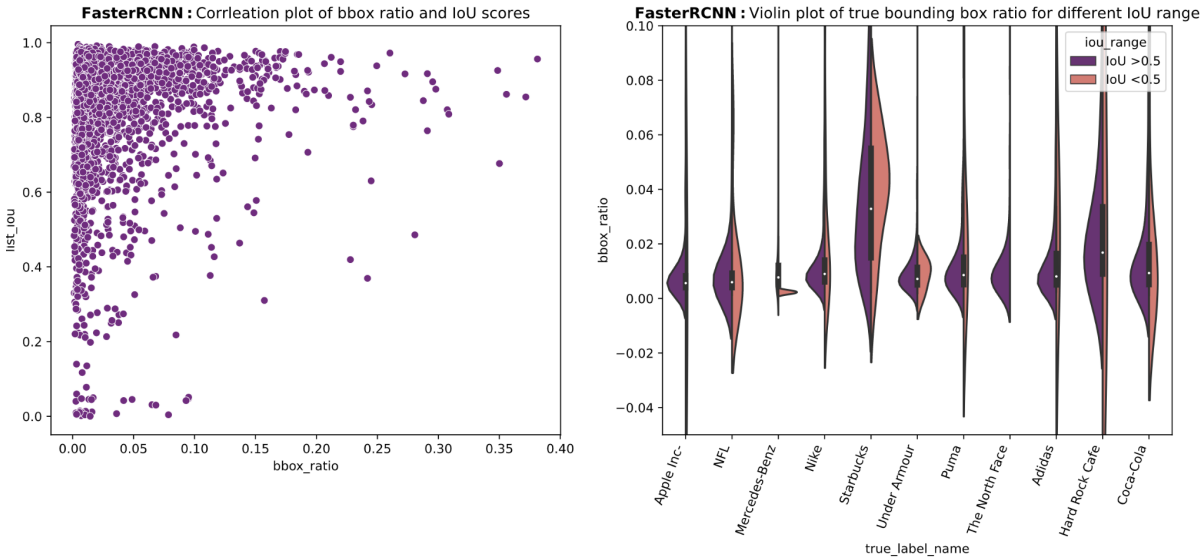
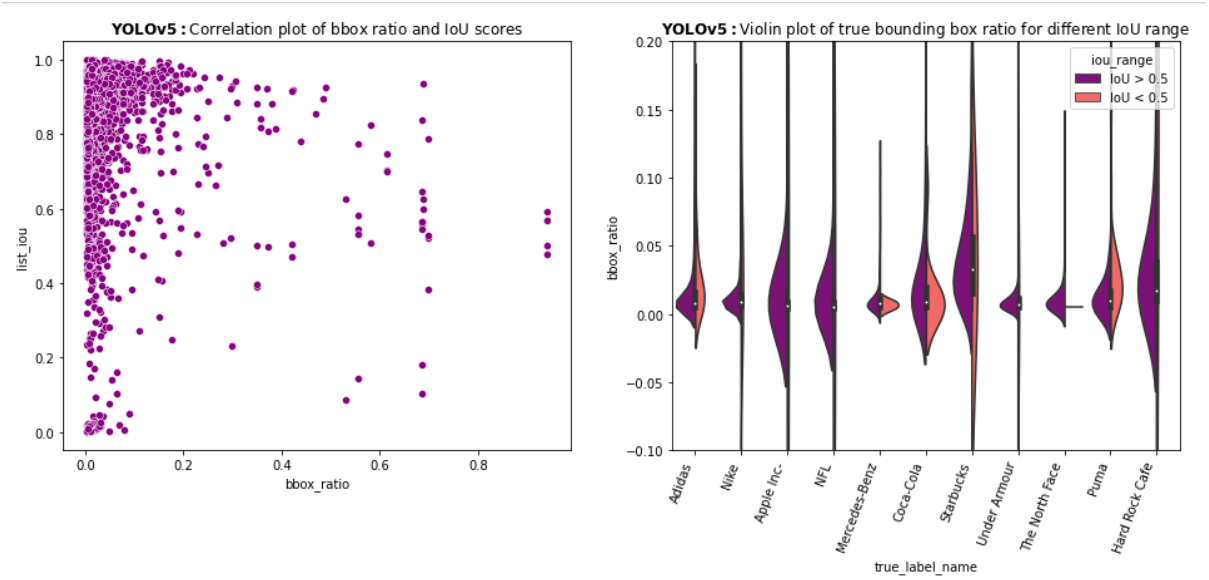


Figure 9: Correlation of bounding box ratio to image to IoU scores (left) and IoU > 0.5/ < 0.5 (right) for YOLOv5



As shown in the scatterplot of the left of the figures, it does indeed tend to be that there is indeed a strong correlation between the size of the bounding box and the prediction performance. At the same time, it does seem that specific logos exist that have more of an issue to predict the larger bounding boxes. It is particularly noticeable in the case of Starbucks, where the average of bounding boxes with IoU <0.5 is at 0.04, compared to 0.015 for well-predicted boxes. A similar relationship holds for the aspect ratios (height/width) shown in Figure 17 in the appendix, where an overall positive relationship between the IoU and aspect ratio has been found but some labels such as Mercedes-Benz, which are predicting quite badly only for very large ratios. This phenomena can at least partially be explained by the small number of instances seen of such ratios during training, thereby making it impossible for the algorithm to predict the feature accurately

Finally, it is important to ensure a high performance of the classification side of the models. Hence, the confusion matrices are provided for each model in Figure 10. As shown in the figures, the classifiers are having an extremely high performance with an above 97% classification score. It can be observed that Faster R-CNN however, has quite a few more wrong predictions assigned to the no value group, i.e. images where nothing was predicted. This further highlights the issue that Faster R-CNN seems to have in predicting the logos in some instances.

Figure 10: Confusion matrix actual label vs. predicted label for YOLOv5

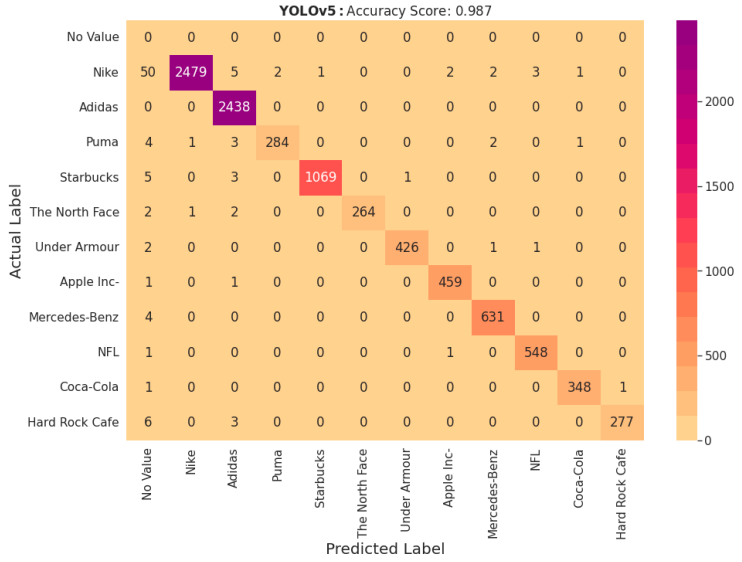
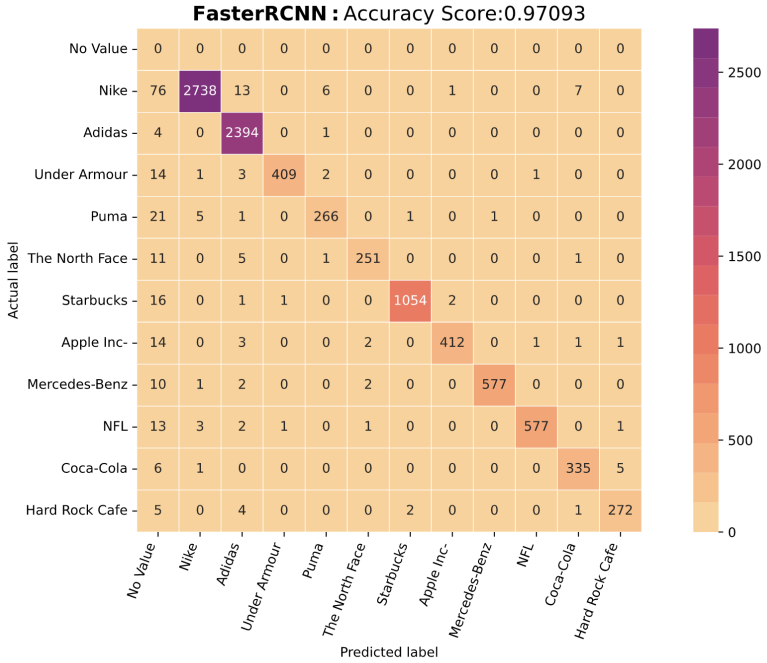


Figure 11: Confusion matrix actual label vs. predicted label for Faster RCNN



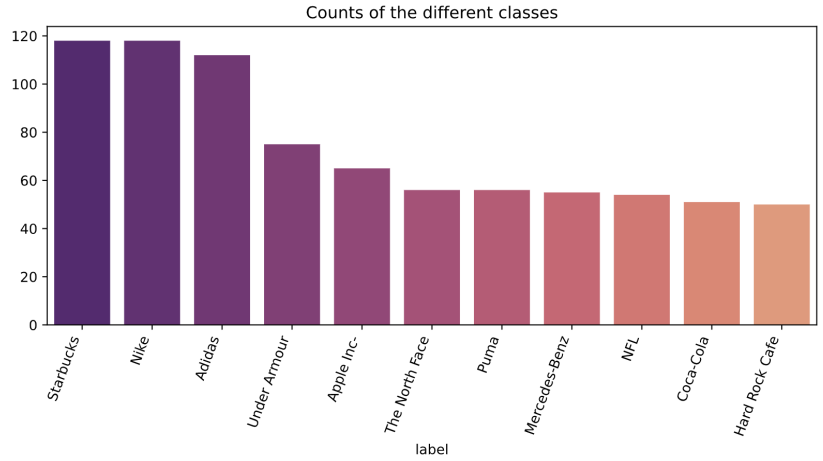
5.3 Extra Testset

Finally, to evaluate the real performance of our classifiers, an extra testset has been created where now multiple logos are allowed for each image. This results in a real multi-logo object detection task, which is essentially the true scope of the development of Yolov5 and Faster R-CNN models and produces a more realistic evaluation of predictive

performance. While the results have been run for both models, only the Yolo results are presented here due to negligible differences in terms of insights gained between the models.

For this scope, ~600 images were carefully scraped, according to the proportions already present in the train/test dataset (shown in figure X), containing the 11 selected logos. Subsequently, these images were manually labeled allowing for the presence of multi-logo (both of the same type and different logos). In Figure 12 below, the distribution of logo instances across the different logos is shown.

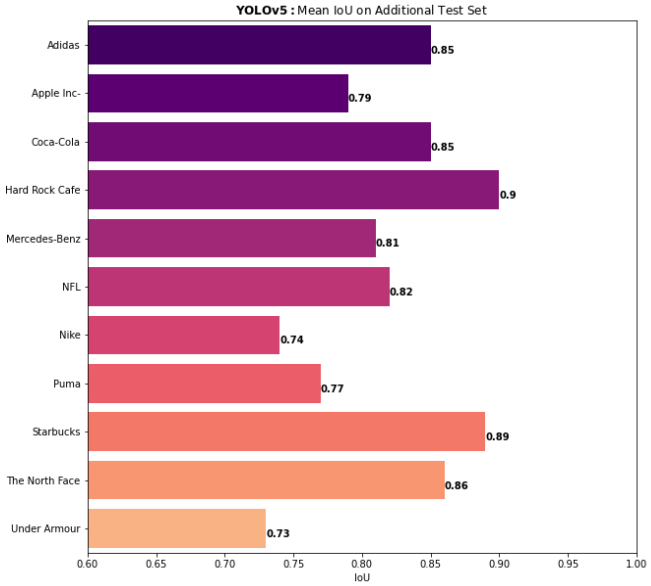
Figure 12: Additional testset counts of different classes



It can be observed that these instances are somewhat differently distributed, with several of the logos showcasing more samples compared to Nike and Adidas than in the original training set, thereby providing an opportunity to highlight the algorithm's performance on a differently sampled dataset.

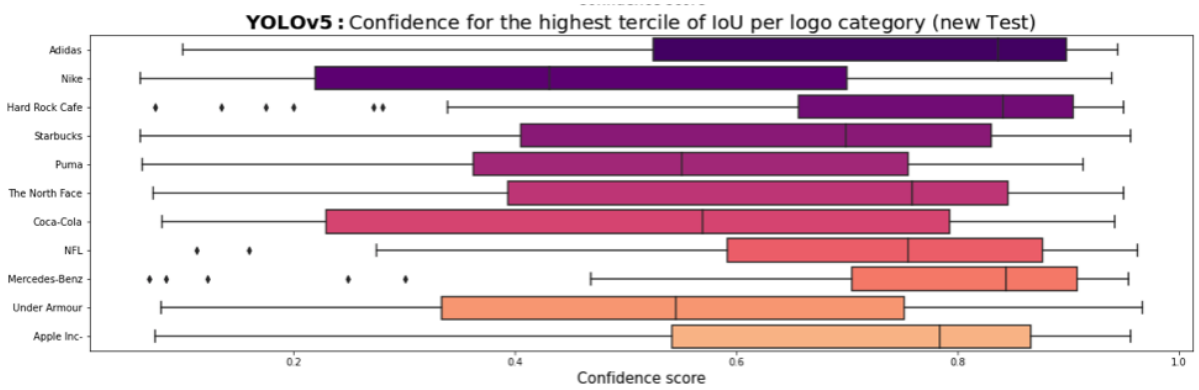
The results of the Yolov5 model based on mean IoU per class are shown in Figure 13. Overall, the YOLOv5 model performs well on the additional test with 7/11 logos achieving a mean IoU above 0.8. This highlights the performance of the model on a more representative dataset with multiple labeled instances, despite a training that was based on images with only one labelled instance. The logos Under Armour and Nike proved to be the most underperforming, with scores below 0.75.

Figure 13: IoU by class for the additional test set for YOLOv5



Comparing these results with the ones delivered on the true dataset, it can be seen that the statistical procedure has the same trend of performance predictions, with high numbers for Adidas, Starbucks and Hard Rock Cafè. It may be assumed that these higher values can be attributed to the nature of the images containing these logos. In fact, logos such as Starbucks and Hard Rock Cafè have unique design characteristics, compared to Nike or Puma which can be easily mistaken for basic shapes. This is confirmed by the confidence scores shown in Figure 14, that express how indeed confident is Yolov5 in predicting such logos, for IoU > 0.5.

Figure 14: Confidence for the highest tercile of IoU per class on new test dataset by YOLOv5

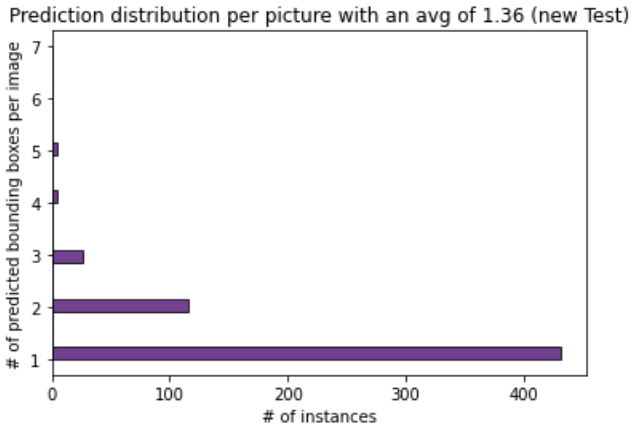


On the other hand, as stated before, for logos that are usually present in images with more instances, the model tends to miss those predictions at all (when filtered by the chosen confidence level of 8%) resulting in an IoU of zero and therefore, pushing the average value to smaller levels. This can be noticed in a set of images (in the Appendix) where it is shown

that for reasons that may be attributed to the training phase, the model tends to capture one logo when presented with two logos that are somehow identical.

This is due to the fact that during the training process, the weights are learnt in such a way that when the correct shape is identified, the search for other logos tends to end. From the figure 14, the model outputs 1.36 predicted bounding boxes per image for this new test set, compared to the 1.30 in the previous test set. Regardless of the higher amount in percentage compared to the previous test set in predicting more-than-one logos inside a picture, this is not enough, yolov5 still sticks at predicting single logos given that it has been trained on single bounding box images.

Figure 15: Prediction distribution per image



Therefore, the right way for the chosen statistical procedures to actually produce consistent and evaluable results would have been to train the models on a dataset where ALL the existent logos have been manually labeled. In this way, the weights are learnt and the parameters validated to induce the model to perform an exhaustive search of logos inside each image, and only then produce its predictions. This would have had massive improvements on the predictions, eliminating any underestimation of the metric of interest, and transforming the average IoU into a fair metric for evaluation.

6. Conclusions and Recommendations

Overall, the models utilised showed a strong performance of over 80% average IoU across all classes for the logo detection task, with YOLOv5 outperforming Faster R-CNN by a few percentage points That was predictable, since the first one is a detection model of newer generation, nevertheless the second had an overall good performance. At the same time, the tendency for Yolo to predict more logos (including more bad ones), was bypassed carrying deep investigations about the confidence threshold, thus providing its best predictions, which were sharper and more accurate. Nonetheless, further investigations on this tendency may lead to interesting results to deeper study: a profound comparison between YOLO, which is a 1-stage detector, and Faster R-CNN which is a 2-stage one, may

highlight and link structural differences to their relative performances, and may help understanding whether the higher predictive performance is worth the additional bad predictions .

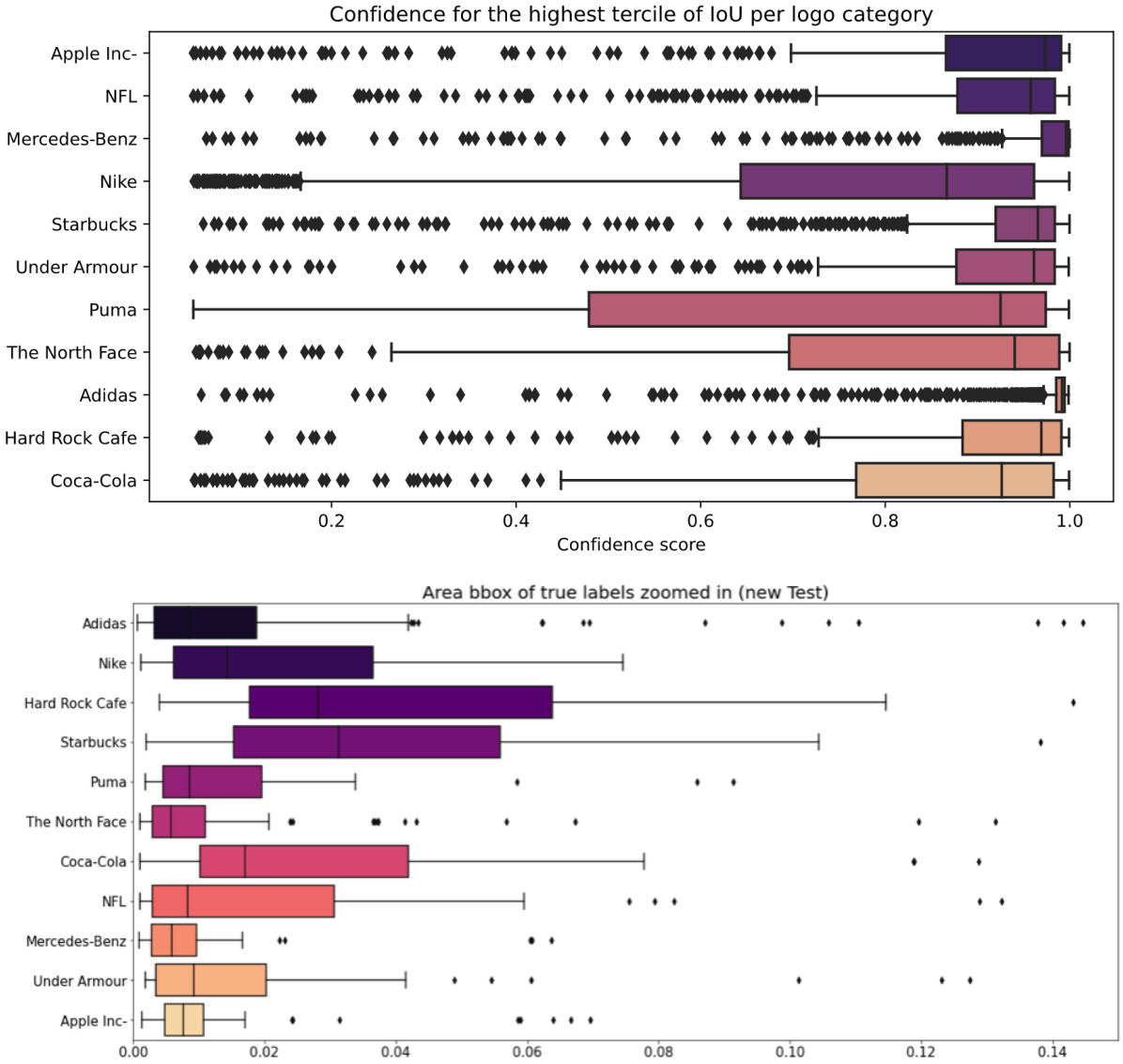
When deep diving on specific logos, a most interesting result was the low performance of Nike despite its overrepresentation in the training data. The reason behind this result is the fact that Nike could be easily misunderstood with a smile or any other swoosh-shaped figure. Furthermore, both algorithms showed strong difficulties in predicting small or blurry bounding boxes. Hence, it is vital to provide further data for such instances, as well as to improve the models at hand to better accommodate these instances.

Moreover, the dataset provided contains a large number of ethically questionable images. Many pictures, taken from Instagram, have a racist background or immortilize inappropriate scenes, and could, then, discredit the professionality of the work. For example, in a photo, the smile of a presumably afro-american guy was classified as “Nike”, and some people may find this result not pertinent for the study carried. Further research should aim to understand any structural reason for such algorithmic biases, as well as ensure no unethical image instances remain in the training dataset of the algorithms.

Finally, the most important improvement that is recommended is to relabel the whole training set specifying and providing annotations for multiple logos, since, for every image, it is provided a single one with the relative bounding box. That is an essential improvement since, providing all the bounding boxes for all the logos that appear in a single picture, provides a more accurate representation of real world Instagram images and hence provides an improvement to the scores of main OD metrics, such as IoUs and mAP, used for evaluating computer vision algorithms.

7. Appendix

Figure 16: Analysis on confidence and bounding box area of true and predicted instances on a new test set for YOLO



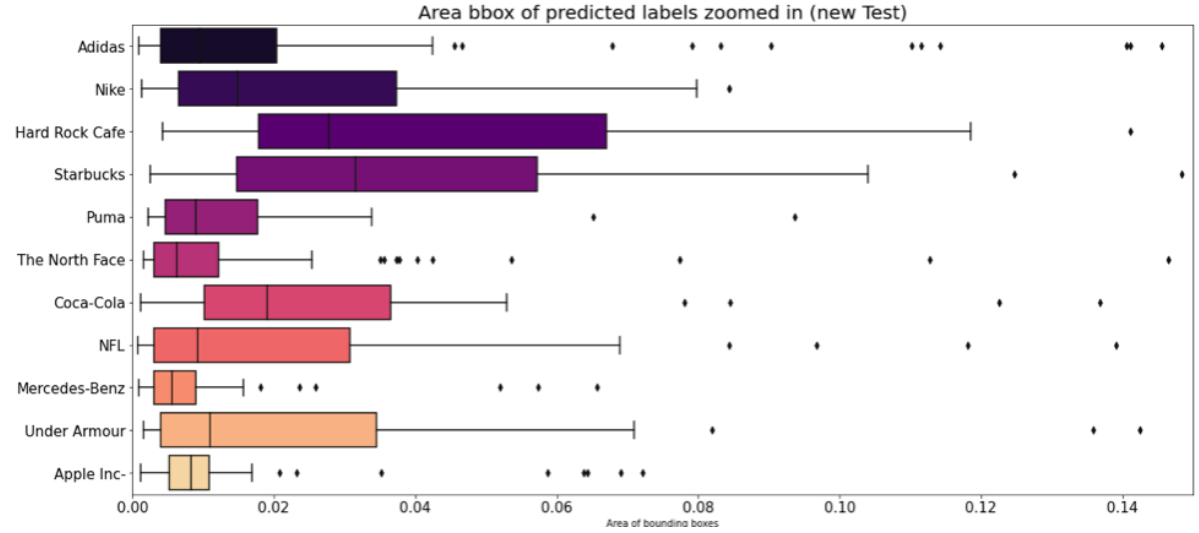
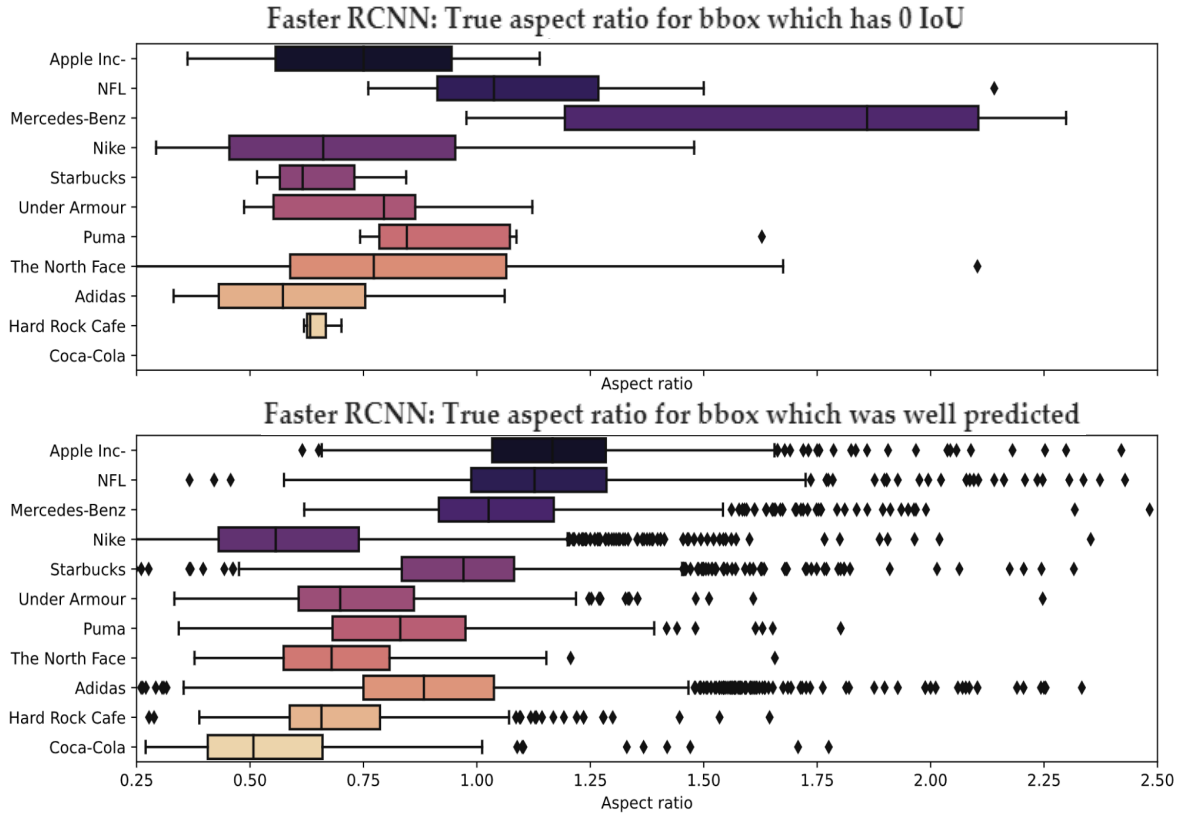
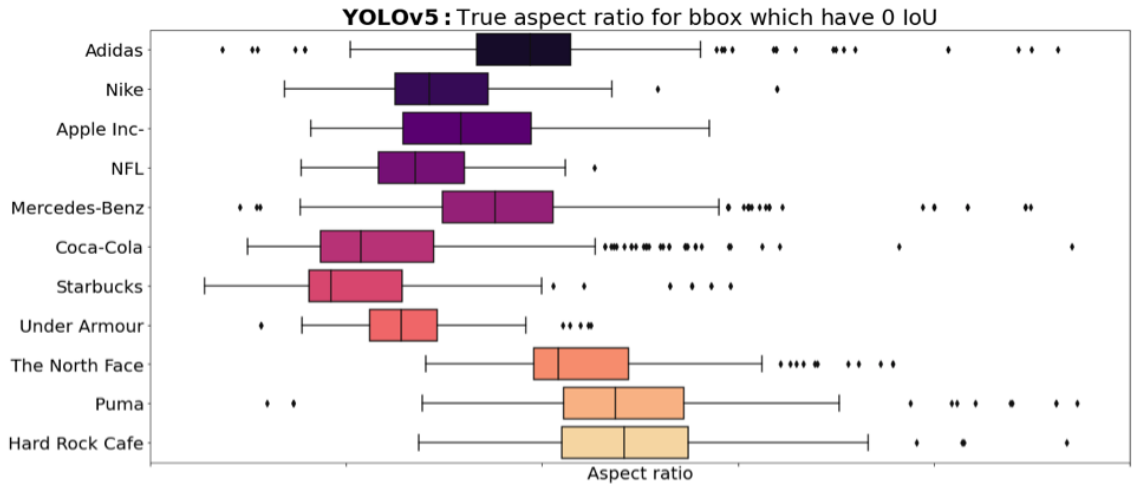


Figure 17: In-depth analysis of true aspect ratios of bounding boxes in new test set for both Faster RCNN and YOLO




YOLOv5 : True aspect ratio for bbox which have 0 IoU
Adidas
Nike
Apple Inc-
NFL
Mercedes-Benz
Coca-Cola
Starbucks
Under Armour
The North Face
Puma
Hard Rock Cafe
Aspect ratio

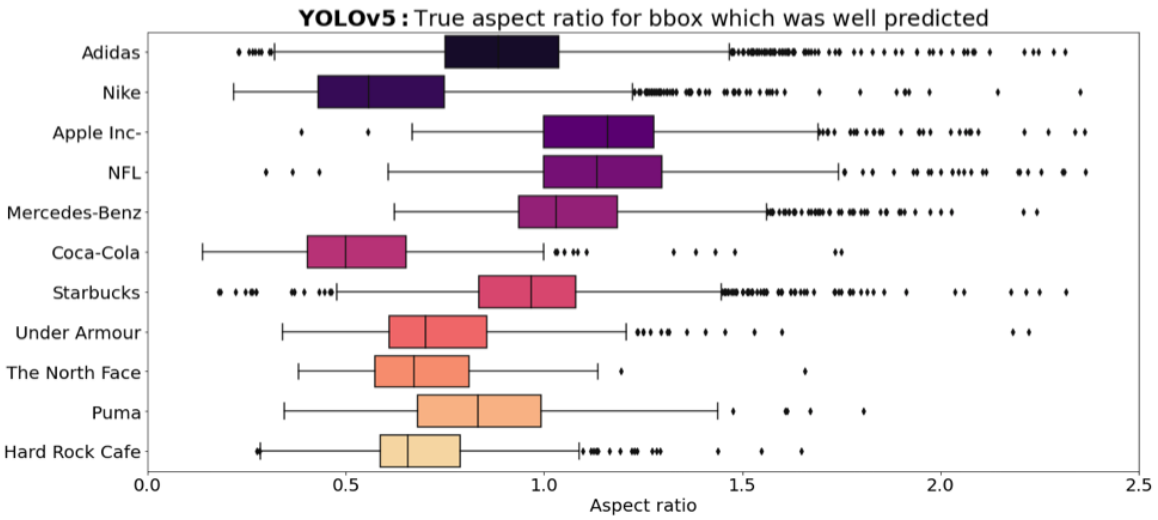
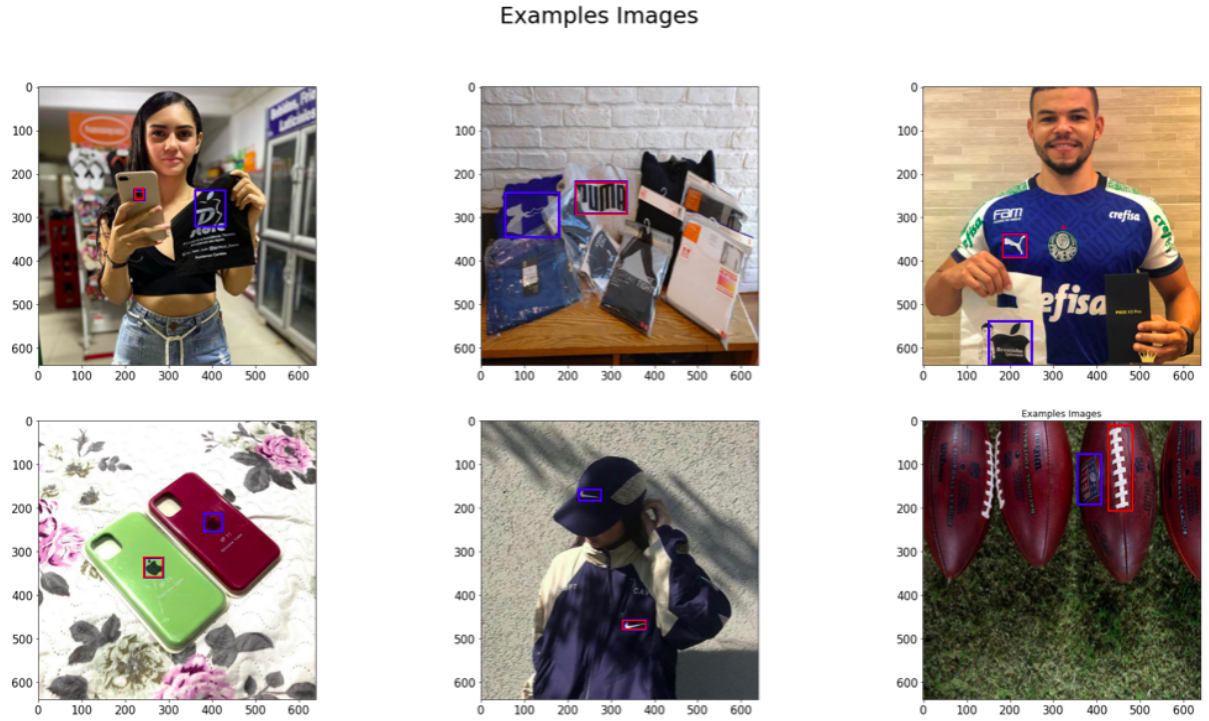

YOLOv5 : True aspect ratio for bbox which was well predicted
Adidas
Nike
Apple Inc-
NFL
Mercedes-Benz
Coca-Cola
Starbucks
Under Armour
The North Face
Puma
Hard Rock Cafe
0.0
0.5
1.0
1.5
2.0
2.5
Aspect ratio

Figure 18: Images where prediction with best IoU is not prediction with highest confidence

Figure 19: Images where prediction not all logos are captured from Yolov5 in additional test set



(Blue is true values, red is model predictions)