Detection of Brands in South Korea using YOLO

PESARAN ZADEH FATEMEH
Department of Computer Science and Engineering, Seoul National University
Graduate Mentor: Prof. Bernhard Egger

INTRODUCTION

The main purpose of object detection is to identify and locate one or more effective targets from image or video data. In this project our main goal is detecting common brands in South Korea. And in the project, we focused on detecting different brands' names and their icons. For this purpose, we used one of object detection model. We trained the model with our primary dataset. However, the obtained result was lower than our expectation therefore, we finetuned the model with the second dataset and achieved much higher accuracy for the model.

METHODS & MATERIALS

01 13500 images from 723 different brands were used. This dataset is publicly available logos_in_the_wild dataset. Since YOLOv5 expects the specific format, we labeled the images and converted the annotations to the desired input format for the YOLOV5 model.

02 There are 4 four choices available for model architecture: yolov5s, yolov5m, yolov5l, yolov5x. The size and complexity of these models increases in the ascending order. After splitting the dataset to the train and valuation sets, by 90% and 10% respectively we trained our model on all the named architectures.

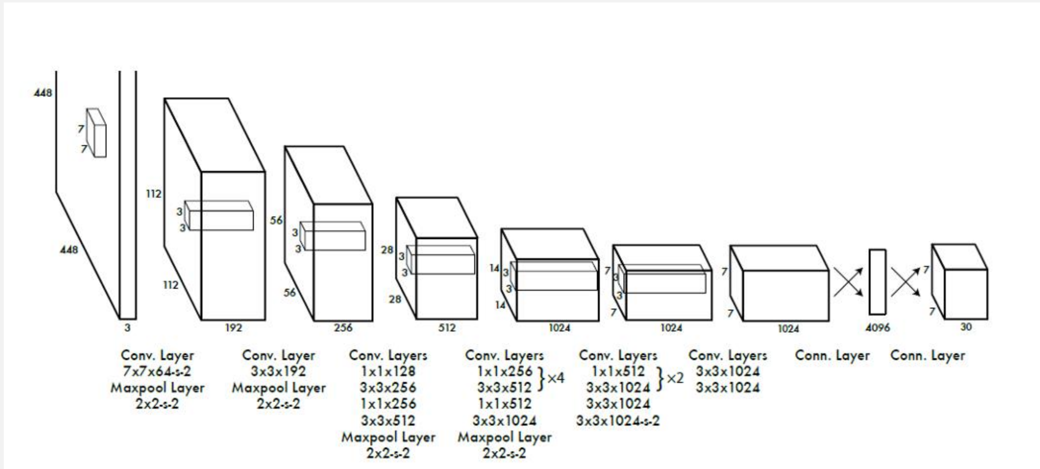

Fig1. Overall architecture of the YOLO algorithm

03 Obtained accuracy from the first training was not as we expected, and we decided to increase the accuracy by fine tuning the model with the new dataset. The second dataset contain more popular South Korean brands. Which were collected by us.

RESULTS

With our custom dataset and custom architectures files ready to go we are ready to train. During the training there are some important hyperparameters which need to be initialized.

- **batch**: determine the batch size
- **epochs**: define the number of training epochs which were set to 300 here.
- **weights**: initialized weight for the model.

In the training we set batch size as 32 due to our limited access to gpu. Bigger batch size would cause error.
The obtained results were highly different from the original results that were announced in the YOLOv5 github directory. The accuracy of trained model with our dataset were lower than what we were expecting. Therefore, we decided on fine tuning the model.
Since our purpose is to detect common brands in mostly South Korea, after testing the trained model we realized it is not as accurate as some famous brands such as Starbucks. And the second problem was there are so many brands in Korea which weren't included in the primary dataset and the model wasn't able to detect them and it would put them in the random already existing classes. Therefore, we trained the model with the second dataset which contains 8350 images with 115 extra classes compare to the first dataset.
After training the trained model with the second dataset the Mean Average Precision increased in the fine-tuned model.
Our obtained result showed in table1.

	mAP^{val} @0.5:0.95			mAP^{val} @0.5		
	COCO	Ours	Fine-tuned	COCO	Ours	Fine-tuned
YOLOv5s	37.4	27.318	66.0	56.8	39.696	66.2
YOLOv5m	45.4	32.2	73.9	64.1	44.9	73.8
YOLOv5l	49.0	35.3	76.5	67.3	48.0	76.5
YOLOv5x	50.7	38.2	75.2	68.9	50.19	75.3

Table1. Comparison of Mean Average Precision of original data on COCO dataset and our custom dataset and fine-tuned model on a second dataset

Figure2 shows the precision-recall curve of the YOLOv5x model on the first dataset. For a good model, precision and recall stays high even when confidence score is varied. In Figure2, the curve for all classes has been showed. And it is highlighted for a single class. During testing we evaluate the area under the curve as average precision, AP. The curve should ideally go from P=1, R=0 on the top left towards P=0, R=1 at the bottom right to capture the full AP (area under the curve).

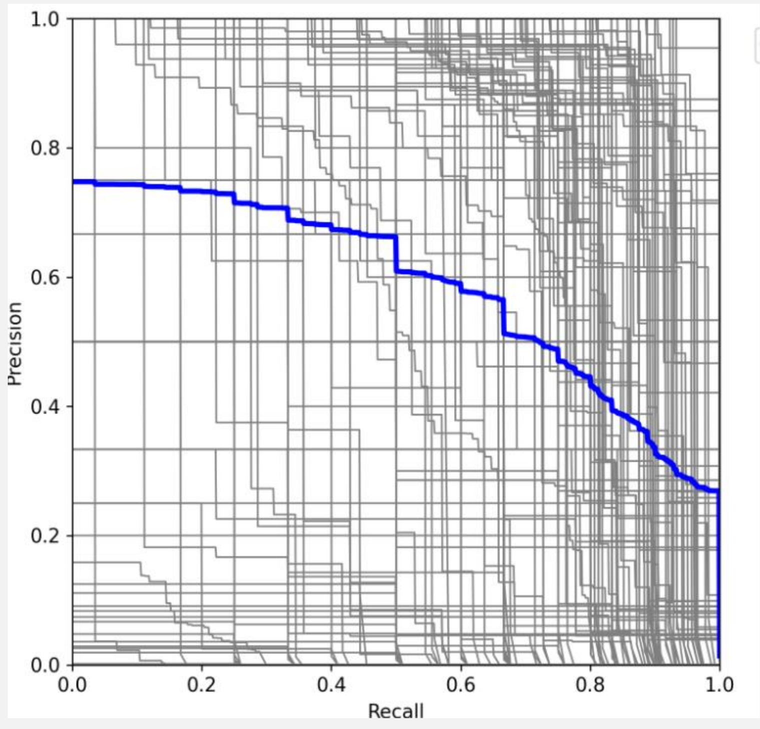

Fig2. precision-recall (PR) curve of YOLOv5x trained with the first dataset.

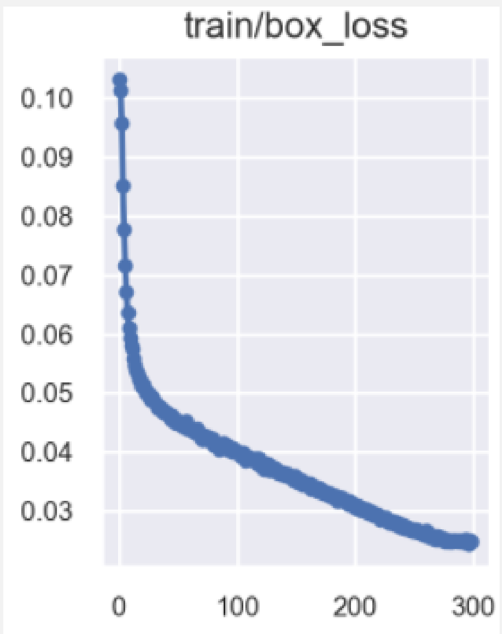

Figure3. loss function of YOLOv5x trained with first dataset in the 300 epochs.

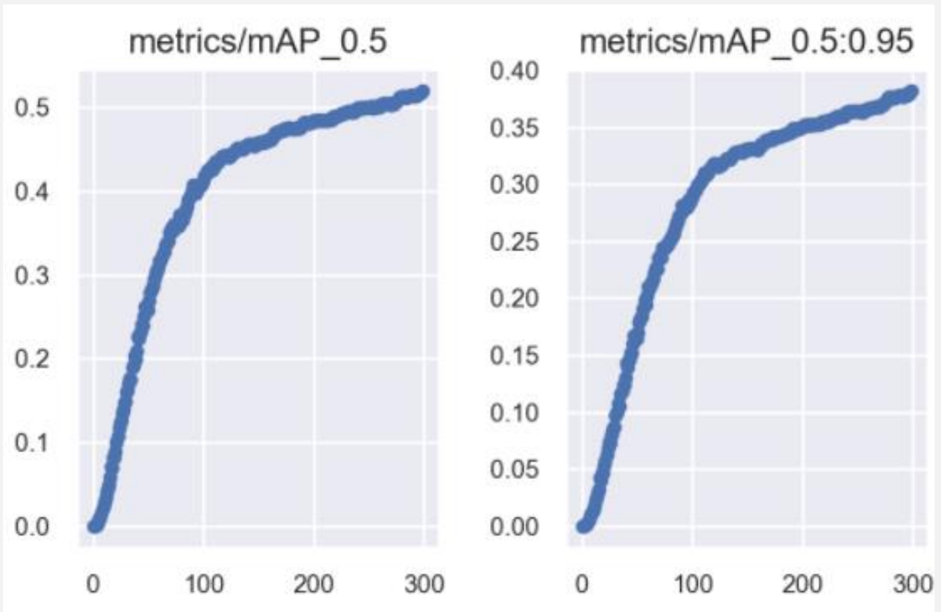

Figure4. mAP@0.5 and mAP@0.5:0.95 of YOLOv5x trained with first dataset in the 300 epochs.

Figure3 shows the box loss of YOLOv5x model trained with first dataset. The box loss represents how well the algorithm can locate the center of an object and how well the predicted bounding box covers an object. And it shows in Figure3 how box_loss is decreasing through 300 epochs of training the model.

Figure4 shows the mean average precision (mAP) to evaluate the object detection YLOv5x model. The mAP compares the ground-truth bounding box to the detected box and returns a score. The higher the score, the more accurate the model is in the detection. And it shows in the

DISCUSSION

As we mentioned the captured results after training model with the first data set were different from announced results in YOLOv5 github directory. The results were reduced by 30.11%, 29.95%, 28.67% and 27.15% for YOLO models s, m, l and x respectively.
Our second dataset did not have the ground truth and labels of images. Since the brands in the second dataset were cropped, for ground through locations we consider the whole image as the ground truth of the image. And for the labeling the image since the pictures file name is a substring of their class name, we tried t use this fact for labeling them. However, there were some mismatch labeling due to the similar substring which it affected our model accuracy.

CONCLUSION

There are many steps that still need to be taken to improve the detection of different brands in South Korea. However, we already showed that by using the fine tuning and increasing the number of images in the dataset and retrain the model we can improve the accuracy of a well-known model such as YOLOv5.
Also, through this experiment we realized that there is that model performance suffers with increased classes. It can be one of the reasons our accuracy was lower than what we were expecting.

References

1. You Only Look Once: Unified, Real-Time Object Detection, Joseph Redmon, University of Washington
2. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Shaoqing Ren
3. YOLOv4: Optimal Speed and Accuracy of Object Detection, Alexey Bochkovskiy
4. https://github.com/ultralytics/yolov5