Object Detection

Thao N. Le

***Abstract*— You Only Look Once (YOLO) model for object detection has been a popular model being one of the most balanced between detection accuracy and runtime. With older models relying on non-maximum suppression, YOLOv10 [1], the newest YOLO version, improves performance efficiency by introducing NMS-free training. However, in smaller models, YOLOv10 lacks accuracy compared to YOLOv9 by a margin of 1.0%AP to 0.5%AP. This paper aims to propose using YOLOv10 in combination with YOLOv9 to maximize training time by integrating the YOLOv9 model into YOLOv10 at a smaller number of parameters.**

***Index Terms*— Object Detection; Computer Vision; Neural Network**

I. Topic Introduction

Object Detection is a core task of Computer Vision, teaching machines to understand the visual world in the way humans do. This is a fast-growing sub-field with applications in medical, car automation, accessibility aid, text recognition, etc.

A. Background

In the early days of object detection, feature extraction used algorithms and image pre-processing to get information from images. Well-known traditional object detection algorithms for example, the Viola-Jones detector (used for face recognition), the Histogram of Oriented Grades detector (efficient object recognition and localization), and the Deformable Parts Model (detect human bodies, automobiles, etc.) bring forth many possibilities to the computer vision field at the time.

Though revolutionary, these methods became time-consuming and inefficient with the invention of Convolutional Neural Network (CNN). As CNN flourished in recent years, two main categories of object detection emerged: one-stage and two-stage object detection. Two-stage detection (R-CNN, Fast R-CNN, etc.) often box the object first and then performs classifications and localization afterward. Since performing two stages is slow, we developed one-stage models like YOLO detecting bounding boxes and classifying objects simultaneously [2, p. 1].

B. Applications

Object detection is used in various situations, one of them is face recognition. Face recognition starts from a simple algorithm involving edge detection, and geometric feature extraction, then evolves to Viola-Jones detector using Haar features and AdaBoost classifiers. Today, using models like Faster Region with Convolutional Neural Network (Faster R-CNN) or Single-Shot Multibox Detector (SSD), the face detection technology has expanded to face recognition, validation, and face tracking [2, p. 14-15]. Another noticeable application of object detection is text detection. With the rise of CNN, text detection went from edge detection-based (using Sobel, Canny, and geometric features) to Deep-learning-based methods (using CNN to extract image features) and improved rapidly over the last 10 years. Although this application is promising, it still faces many challenges with images having different text fonts, complicated backgrounds, multilingual paragraphs, etc [2, p. 16]. And there are many more applications not mentioned here i.e. pedestrians recognition, autonomous vehicles, medical imaging, etc.

II. Existing Methods

In this section, I will summarize two predominant models from two-stage and one-stage strategies in detecting objects. As said above, the two-stage strategy consists of screening for bounding boxes, and then classifying them, which yields high accuracy with the trade of time efficiency. The one-stage strategy uses only one network to detect, therefore, it will run faster but also the accuracy is high overall but not as high as R-CNN.

A. R-CNN Series

1) R-CNN: The Regional Convolutional Neural Network model was proposed in 2014, marking the first time that deep learning was used for Object Detection. It uses selective search to divide the candidate regions on the image, classify them, and regress on the convolutional network to achieve the result. At the time, the model was revolutionizing and achieved 58.5% on the VOC2007 test set. However, their runtime was reported to be inefficient and not suitable for real-time detection.

2) Fast R-CNN: [3] Proposed in 2015, fast R-CNN aims to fix the runtime problem in R-CNN previously. Fast R-CNN uses the spatial pyramid pooling network (SPPNet) to share the burden and merge classification and regression tasks in one network. The model, indeed, was improved in accuracy to 66.9% on VOC2007. However, due to the selective search algorithm, the model is still not suitable for real-life detection.

3) Faster R-CNN: In the same year, faster R-CNN was released replacing the selective search by Regional Proposal Network solving the runtime issue. The accuracy for the model on VOC2007 was improved to 69.9% with a much better run-time. Still, real-life detection is still a challenge due to long training and complex computational processes.

B. YOLO Series

1) YOLOv1: Also in 2015, the You Only Look Once model was released. By turning all tasks into one problem for one Neural Network (model from GoogLeNet [4]), the object detection problem was solved in a faster time. The algorithm goes through a grid, predicts bounding boxes, classifies objects, and then, runs them through CNN. Due to the prediction happening concurrently, the localization ability is weak. The model struggles to localize small objects and misses them in the images. Despite that, the model was able to run in real-time and detect objects quickly. [5]

2) YOLOv8: Proposed in 2023, YOLOv8 can perform tasks not only in detection but also in segmentation and classification. Combining the ideas of C3 and Efficient Layer Aggregation to design a new C2f model, which makes an even more lightweight model.

3) YOLOv10: Building on YOLOv8, YOLOv10 looks for improvement in non-maximum suppression. YOLOv10 provides the model with more accuracy and a lighter training weight. Achieving an even higher accuracy with more complex training weight than the model before but for lighter weight, the model is still lagging behind YOLOv9 by 0.5%-1%. [1]

III. Open Challenges

There are still challenges in images with complex backgrounds making it hard for existing models to detect objects. Objects with different poses are also an open problem since current models lack of ability to generalize especially with different shapes. These two problems make video detection a problem since, in the video, the background and object's pose change drastically. Data bias, computational efficiency, and small object detection are some of the other open challenges in the field.

IV. Concept to Code

A. Method

To combine YOLOv9 and YOLOv10, I need to understand at what number of parameters YOLOv10 begins to outperform YOLOv9. According to the graph presented by Ao Wang [1, p. 1], at about less than 20, YOLOv10 begins to outperform or at least on par with YOLOv9. The number of Parameters is proportional to the complexity of the model so I tested on the smallest model of both and another close to the smallest model. In Figure 3, we can see that YOLOv10 performs worse with a mean Average Precision (mAP) of only around 60% for the training dataset. But for the close to the smallest model, Figures 1 and 2 show that they are both on par with mAP around 80% for 20 and beyond epochs training.

I proposed a pipeline in which we monitor the CPU/GPU activity and if the RAM usage is above 75%, we will drop the model YOLOv10 down to a lighter-weight model and when reaching the smallest, we will switch to YOLOv9t instead of YOLOv10n.

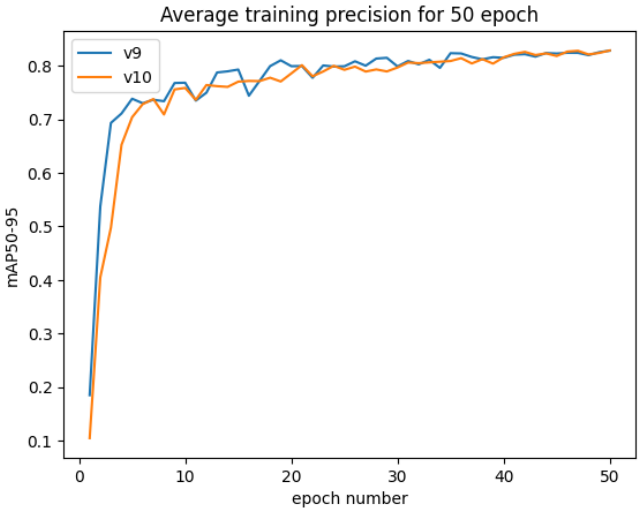

Fig. 1. The graph describes the mean Average Probability (mAP) of Intersection Over Union (IOU) 50-95 over the number of epochs of YOLOv9s and YOLOv10s. We can see that with 50 epochs, the two models provide similar accuracy over all epochs.

Epoch	YOLOv9	YOLOv10
20	0.803 32	0.791 39
50	0.796 13	0.793 62
100	0.819 88	0.826 65

TABLE I

The table show the nearly equal behavior between YOLOv9s and YOLOv10s in different epoch sizes.

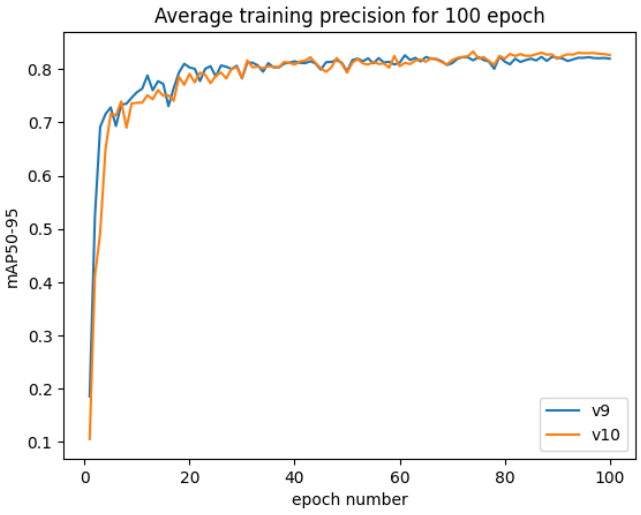

Fig. 2. The graph describes the mean Average Probability (mAP) of Intersection Over Union (IOU) 50-95 over the number of epochs of YOLOv9s and YOLOv10s. We can see that with 100 epochs, the two models provide similar accuracy over all epochs.

B. Dataset

The test dataset is the chess piece dataset on roboflow. e

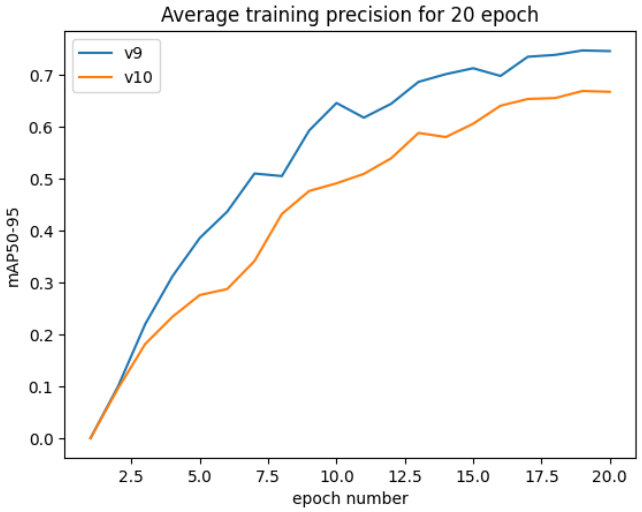

Fig. 3. Epoch20 on yolo9t and 10n

C. Instruction

Please see the details in the Python Notebook. The Notebook includes the instructions and also how the graphs were generated.

D. Future Work

For future work, I want to successfully have the multiprocess running on Google Colab. With that, I will be able to streamline the idea of changing the model to find the most optimal model for any system to run.

REFERENCES

[1] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "YOLOv10: Real-time end-to-end object detection." [Online]. Available: http://arxiv.org/abs/2405.14458

[2] Z. Li, Y. Dong, L. Shen, Y. Liu, Y. Pei, H. Yang, L. Zheng, and J. Ma, "Development and challenges of object detection: A survey," vol. 598, p. 128102. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231224008737

[3] R. Girshick, "Fast r-CNN." [Online]. Available: http://arxiv.org/abs/1504.08083

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions." [Online]. Available: http://arxiv.org/abs/1409.4842

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection." [Online]. Available: http://arxiv.org/abs/1506.02640