# Comparison of YOLOv5 and Faster R-CNN for Object Detection

Rishabh Shah
*Artificial Intelligence*
*Duke University*
Durham, NC, USA
*rs659@duke.edu*

*Abstract*—**Object detection is a crucial task in computer vision with applications ranging from autonomous vehicles to surveillance systems. In this project, we conducted a comparative analysis of two state-of-the-art object detection models, YOLOv5 and Faster R-CNN, focusing on their performance metrics such as Mean Average Precision (mAP), speed, and accuracy. The models were evaluated on a dataset comprising images of utensils, laptops, and drinks, annotated using the Computer Vision Annotation Tool (CVAT).**

## I. Introduction

Object detection, a fundamental and challenging problem in computer vision, has seen significant advancements over the past two decades. It is a critical task in various applications, ranging from medical imaging to autonomous driving. This report presents a comparison of two state-of-the-art object detection models: YOLOv5 [1] and Faster R-CNN [2]. The models were trained and evaluated on a dataset comprising images of utensils, laptops, and drinks.

The task of object detection can be challenging for beginners to distinguish between different related computer vision tasks. For instance, image classification involves assigning a class label to an image, whereas object localization involves drawing a bounding box around one or more objects in an image. Object detection is more challenging and combines these two tasks and draws a bounding box around each object of interest in the image and assigns them a class label.

## II. History of Object Detection

The evolution of object detection can be broadly divided into two periods: the traditional object detection period (before 2014) and the deep learning-based detection period (after 2014). Early object detection algorithms were built based on handcrafted features, such as Viola-Jones Detectors, HOG Detector, and Deformable Part-based Model (DPM). With the advent of deep learning, models like YOLO and Faster R-CNN have revolutionized the field.

The era after 2014 marked the advent of deep learning-based detection methods. These methods leveraged the power of neural networks and large amounts of data to achieve remarkable improvements in object detection. This era witnessed the development of many milestone detectors, which have had a profound impact on the field of computer vision.

## III. Dataset and Annotation

The dataset utilized in this project comprises 993 annotations distributed among the images, averaging 3.3 annotations per image. These annotations are segmented into three distinct classes: Laptops, Utensils, and Drinks. Specifically, there are 131 annotations corresponding to laptops, 523 to utensils, and 339 to drinks. The annotation process was facilitated by the Computer Vision Annotation Tool (CVAT) [5] , a versatile platform offering a broad array of annotation tools tailored for various aspects of image and video labeling.

CVAT is a famous tool within the computer vision community, recognized for its robust annotation capabilities. It presents a diverse toolkit, with each tool addressing specific requirements in image and video annotation. For instance, the rectangle tool facilitates the delineation of bounding boxes around objects, aiding in precise labeling.

To annotate images using CVAT, we created a task and defining labels for the objects of interest . Subsequently, images were uploaded to CVAT, and the labeling process commences. Upon completing the annotations, they were saved and can be downloaded in the preferred format.

Annotation rules were implemented to maintain consistency and accuracy during labeling. These rules covered object coverage, bounding box consistency, labeling ambiguity, and object occlusion. Object coverage ensured annotations encompassed the entire object. Bounding box consistency maintained uniformity in size and shape. Labeling ambiguity guidelines minimized subjective interpretation. Object occlusion was addressed by accurately delineating partially obscured objects. These rules ensured a high-quality dataset for subsequent tasks such as machine learning model training.

## IV. Data Augmentation

Data augmentation is a critical component of training deep learning models. For this dataset, various data augmentation techniques were employed, including random cropping, image mirroring, and geometric transformations.

Data augmentation is a critical component of training deep learning models. It is especially important for object detection tasks due to the additional cost for annotating images. Traditional data augmentation techniques for object detection include rotation, scaling, flipping, and other manipulations of each image. These techniques encourage the model to learn more invariant features, thereby improving the robustness of the trained model.

TABLE I
Configuration for Data Augmentation

| Parameter | Value | Definition |
|---|---|---|
| lr0 | 0.01 | Initial learning rate (SGD=1E-2, Adam=1E-3) |
| lrf | 0.2 | Final OneCycleLR learning rate (lr0 * lrf) |
| momentum | 0.937 | SGD momentum/Adam beta1 |
| weight_decay | 0.0005 | Optimizer weight decay 5e-4 |
| warmup_epochs | 3.0 | Warmup epochs (fractions okay) |
| warmup_momentum | 0.8 | Warmup initial momentum |
| warmup_bias_lr | 0.1 | Warmup initial bias learning rate |
| box | 0.05 | Box loss gain |
| cls | 0.5 | Cls loss gain |
| cls_pw | 1.0 | Cls BCELoss positive weight |
| obj | 1.0 | Obj loss gain (scale with pixels) |
| obj_pw | 1.0 | Obj BCELoss positive weight |
| iou_t | 0.20 | IoU training threshold |
| anchor_t | 4.0 | Anchor-multiple threshold |
| fl_gamma | 0.0 | Focal loss gamma (efficientDet default gamma=1.5) |
| hsv_h | 0.015 | Image HSV-Hue augmentation (fraction) |
| hsv_s | 0.7 | Image HSV-Saturation augmentation (fraction) |
| hsv_v<br>heightdegrees | 0.4<br>0.0 | Image HSV-Value augmentation (fraction)<br>Image rotation (+/- degrees) |
| translate | 0.1 | Image translation (+/- fraction) |
| scale | 0.5 | Image scale (+/- gain) |
| shear | 0.1 | Image shear (+/- degrees) |
| perspective | 0.0 | Image perspective (+/- fraction), range 0-0.001 |
| flipud | 0.0 | Image flip up-down (probability) |
| fliplr | 0.5 | Image flip left-right (probability) |
| mosaic | 1.0 | Image mosaic (probability) |
| mixup | 0.2 | Image mixup (probability) |
| copy_paste | 0.0 | Segment copy-paste (probability) |

## V. Model Comparison

YOLOv5 and Faster R-CNN were compared based on their Mean Average Precision (mAP), speed, and size. YOLOv5 achieved an mAP@50 of 76.5 with a GPU latency of 3.9ms, while Faster R-CNN achieved an mAP@50 of 71.39 with a GPU latency of 76.024ms.

Object detection has seen significant advancements with the introduction of deep learning models such as YOLOv5 and Faster R-CNN . These models have been widely used for real-time object detection tasks due to their high performance in terms of speed and accuracy .

Faster R-CNN, a region-based convolutional neural network, is known for its precision in object detection. It employs a region proposal network (RPN) to generate potential bounding boxes and uses a separate network to classify these proposed regions . However, its complex architecture can lead to slower inference times. For instance, Faster R-CNN achieved an mAP@50 of 71.39 with a GPU latency of 76.024ms .

On the other hand, YOLOv5, part of the ”You Only Look Once” family, is designed for speed and real-time use . It treats object detection as a single regression problem, directly predicting bounding box coordinates and class probabilities from full images in one evaluation . Despite its speed, it has shown superior accuracy with an mAP@50 of 76.5 and a GPU latency of 3.9ms .

The choice between YOLOv5 and Faster R-CNN depends on the specific requirements of the task. Faster R-CNN may be preferred for tasks where precision is paramount, while YOLOv5 could be more suitable for real-time applications where speed is crucial.

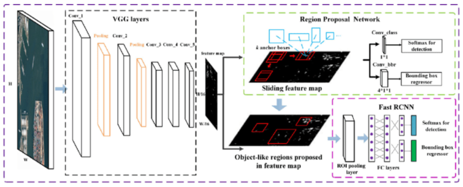

Fig. 1. Architecture of Faster R-CNN

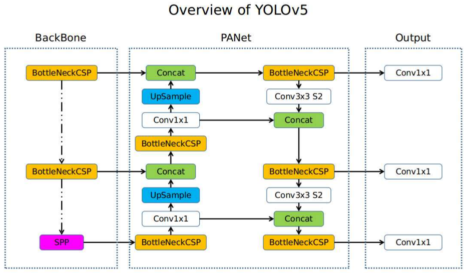

Fig. 2. Architecture of YOLOv5

TABLE II
Model Architecture Comparison

| Model | Architecture | Key Features |
|---|---|---|
| YOLOv5 | Single-stage object detection | Utilizes a unified neural network to directly predict bounding boxes and class probabilities without region proposal networks. Employs anchor-based predictions. |
| Faster R-CNN | Two-stage object detection | Comprises a region proposal network (RPN) to generate region proposals followed by a region-based convolutional neural network (R-CNN) for object detection. Typically achieves higher accuracy but with slower inference compared to single-stage detectors. |

## VI. Model Training

Training a deep learning model involves feeding it data and adjusting its parameters so that it can make accurate predictions. In this project, two different models were trained: Faster R-CNN using Detectron2 [3] and YOLOv5 using Ultralytics [4].

Faster R-CNN was trained using the Detectron2 framework. Detectron2 is a robust platform for object detection and segmentation developed by Facebook AI Research (FAIR). It provides a wide variety of pre-trained models and supports many state-of-the-art models. The training process involved setting up the configuration, loading the dataset, and then training the model. The configuration included the model type, the number of iterations, learning rate, and other parameters.

On the other hand, YOLOv5 was trained using the Ultralytics framework. Ultralytics is a Python-based machine learning (ML) framework that offers efficient and straightforward tools for ML research and development. It provides pre-trained models, training scripts, and easy-to-use CLI and Python interfaces. The training process involved loading the pre-trained model, setting up the configuration, and then training the model.

Both models were trained on a dataset of images containing utensils, laptops, and drinks. The dataset was annotated using the CVAT tool, and various data augmentation techniques were applied to increase the size of the dataset.

TABLE III
PERFORMANCE METRICS OF OBJECT DETECTION MODELS

| Metrics/Models | mAP@50 | mAP@75 | AP - Utensils | AP - Laptops | AP - Drinks | Speed (ms per image) | Size (Million) |
|---|---|---|---|---|---|---|---|
| YOLOv5n | 0.765 | 0.67 | 0.678 | 0.903 | 0.714 | 3.9 | 3.2 |
| Faster RCNN | 71.39 | 0.51 | 57.98 | 64.36 | 56.69 | 76.024 | 42 |

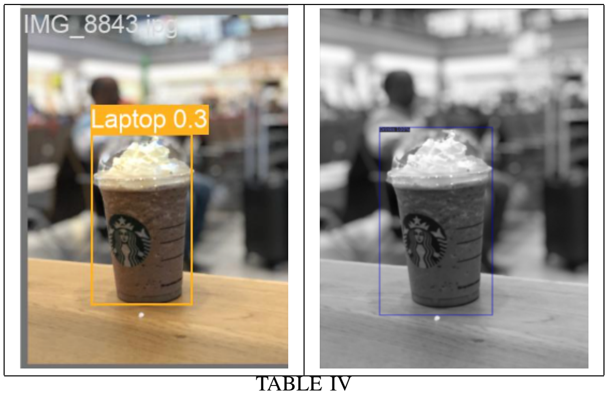

TABLE IV
AN EXAMPLE WHERE YOLOV5 PROVIDES INACCURATE RESULTS WHILE FASTER R-CNN DELIVERS CORRECT DETECTIONS

## VII. CONCLUSION

In conclusion, YOLOv5 emerges as the superior choice over Faster R-CNN due to its faster processing speed and comparable accuracy. While YOLOv5 excels in real-time object detection scenarios, Faster R-CNN offers higher precision at the expense of speed. The decision between the two models depends on the specific requirements of the task, with Faster R-CNN being preferable for applications where precision is paramount, while YOLOv5 shines in real-time settings where speed is crucial.

As for future work, there are several avenues for improvement. Firstly, increasing the dataset size can enhance the robustness and generalization ability of the models. Additionally, refining the annotations with better criteria can lead to more accurate and meaningful results. By addressing these aspects, we can further enhance the performance and applicability of object detection models in various real-world scenarios.

## REFERENCES

[1] Wang, G., Zhang, S., Ren, W., Sun, J., & Huang, K. (2020). YOLOv5: An Incremental Improvement. Zenodo. https://doi.org/10.5281/zenodo.3908559

[2] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28.

[3] Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). Detectron2. Retrieved from https://github.com/facebookresearch/detectron2

[4] Ultralytics. Retrieved from https://www.ultralytics.com/

[5] Boris Sekachev et al. (2020). opencv/cvat: v1.1.0. Zenodo. https://doi.org/10.5281/zenodo.4009388