Object detection using CNN

Prof. Nikitha K S, Amaan Ahiq, Minaam Reyaz Qureshi, Darshan Pasargi, Amar Bashir
Dept. of Computer Science & Engg., Bangalore Institute of Technology, Bangalore, India

***Abstract*— A YOLOv5 model-based object detection software is presented in this paper. A number of forms of media sources, such as pictures, videos, and real-time videos, can be used with the program. Users can choose the location of the model weights, the media to be analyzed, the size of the image processing to be done, and more. After identifying items in the media, the script outputs those objects together with the bounding boxes that belong to them. It uses a method known as Non-Max Suppression (NMS) to get rid of duplicate detections and provides options for saving the results in several formats or seeing them right on the screen. For possibly better detection accuracy over time, the script can optionally be set up to update the YOLOv5 model weights.**

Keywords—Object Detection, Convolutional Neural Networks (CNNs), YOLOv5, Bounding Boxes, Real-Time Performance

I. INTRODUCTION

1) *Background and context :* In the realm of computer vision, achieving real-time object detection with high accuracy remains a critical challenge. You Only Look Once (YOLO), particularly its latest iteration, YOLOv5, has emerged as a powerful tool for this task. Unlike traditional two-stage methods like R-CNN, YOLOv5 employs a single-stage approach, significantly accelerating processing speed. This approach builds upon the success of previous YOLO versions while introducing innovations like the Focus layer for improved efficiency. Furthermore, YOLOv5's integration with the PyTorch framework fosters user-friendliness and facilitates rapid development. This combination of speed, accuracy, and ease of use has positioned YOLOv5 as a valuable asset for various real-world applications, including autonomous vehicles, security surveillance, and medical image analysis. demanding fast and accurate object detection.
2) *Objective :* This research aims to assess the suitability of YOLOv5, a cutting-edge single-stage object detection model, for achieving real-time performance with high accuracy in computer vision tasks. We will evaluate its effectiveness in balancing these crucial aspects by analyzing its architecture, including the recently introduced Focus layer, and its integration with the PyTorch framework. By delving into these elements, we hope to gain a deeper understanding of YOLOv5's strengths and limitations for real-world applications aspects, we aim to understand YOLOv5's strengths and limitations, particularly in complex scenarios. Ultimately, this research seeks to explore optimization strategies for YOLOv5 to improve the speed-accuracy trade-off and its generalizability to new object classes.
3) *Research Gap :* YOLOv5's impressive performance in real-time object detection necessitates further exploration of its limitations in complex scenarios. A crucial research gap lies in its robustness under challenging conditions like poor lighting, occlusions, or the detection of small objects. These situations often lead to performance degradation in object detection models. A deeper understanding of YOLOv5's behavior in these settings is essential to determine its generalizability and effectiveness in real-world applications. Additionally, the trade-off between speed and accuracy requires further scrutiny. Can YOLOv5 be optimized to achieve even higher accuracy while maintaining its real-time capabilities? This research gap is particularly important for practical deployments where both speed and precision are critical.
4) *Current Scenario and Need for Innovation :* Current object detection models, despite impressive advancements, exhibit limitations in complex scenarios. Cutting-edge models like YOLOv5, while achieving high accuracy and speed, struggle with robustness in poor lighting, occlusions, and small object detection. This performance gap necessitates further research to enhance generalizability for real-world applications. Additionally, optimizing the speed-accuracy trade-off and improving model generalizability to new object classes remain crucial research frontiers.

5) *Scope of the paper :* This paper focuses on evaluating YOLOv5, a cutting-edge object detection model, for its suitability in real-world computer vision tasks. We will assess its ability to achieve high accuracy while maintaining real-time speeds. Our scope encompasses analyzing YOLOv5's architecture, including the Focus layer, and its PyTorch integration. By examining these
6) *Organization of the paper :* This paper unpacks YOLOv5's potential for real-world computer vision tasks that demand both speed and accuracy. We'll start by explaining why real-time object detection is crucial and how YOLOv5, with its innovative features, tackles real-world challenges.

Next, we'll peek at existing research to understand how YOLOv5 stacks up. Then, we'll delve into our experiment setup, including the data used, how we measured success, and any adjustments made to YOLOv5. Finally, we'll present the results and discuss what they reveal about YOLOv5's performance compared to others. This step-by-step approach ensures a clear understanding of YOLOv5's effectiveness in real-world applications.

II. CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural network (CNN) is a class of deep, feed-forward artificial neural network that has been utilized to produce an accurate performance in computer vision tasks, such as image classification and detection. CNNs are like traditional neural network, but with deeper layers. It has weights, biases and outputs through a nonlinear activation. The neurons of the CNN are arranged in a volumetric fashion such as, height, width and depth.

The CNN design, which consists of a convolutional layer, a pooling layer, and a fully connected layer, is seen in Fig. 1. The pooling layer and convolutional layer are usually alternated, and each filter's depth rises from left to right as the output size (height and width) decreases. The final step is called the fully connected layer, and it resembles the final layer of traditional neural networks.

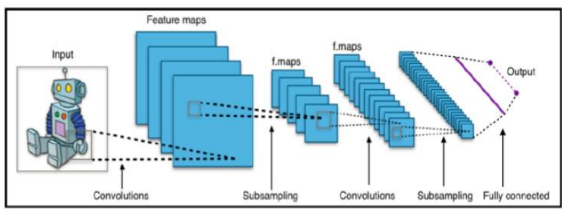

Fig. 1. CNN Architecture

A picture that contains pixel values is the input. It contains three dimensions, for example, [50 x 50 x 3], which are width, height, and depth (RGB channels). The output of neurons associated to specific local areas in the input will be calculated by the convolutional layer. The parameters of the layer consist of a collection of learnable filters, also known as kernels, that compute the dot product between the input and filter entries by varying the width, height, and depth of the input volume. This generates a two-dimensional map of the filter's activation, and as a consequence, the network learns filters that activate when it finds a certain kind of feature at a specific spatial location in the input. The function called Rectified Linear Unit (ReLU) layer will perform elementwise activation function. ReLU is defined in (1),

$$f(x) = \max(0, x) \quad (1)$$

This function is zero for negative values and grows linearly for positive values. This will not affect the volume size. The pooling layer outputs the maximum activation in a region. This down samples the spatial dimensions such as width and height. The output layer is the fully connected layer which is similar to the final layer of the neural network. This layer uses commonly used softmax activation to output probability distributions over the number of output classes.

III. TRANSFER LEARNING

Transfer learning is a potent deep learning technique that allows feature extraction and fine tuning from pre-trained models. This technique can be used for segmentation, object identification, and vehicle classification in images. The benefit of this method is that it requires less data to provide effective results and saves time by training the network from the beginning. Even without the graphics processing unit's (GPU) computing capacity, the training can be completed with a central processing unit (CPU). Pre-trained models can be utilized in place of starting from scratch with a new model. These models are trained from the large image databases like ImageNet and COCO dataset. Some of these models are AlexNet, VGG16, VGG19, MobileNet, InceptionV2, InceptionV3, Xception, DenseNet and MobileNet. This paper focuses on the model namely ResNet50.

IV. PyTorch OBJECT DETECTION MODELS

This research leverages PyTorch, a powerful open-source deep learning framework, to facilitate the development and deployment of the YOLOv5 object detection model.

PyTorch and ResNet-50 (A Tailored Backbone for Object Detection): This research employs ResNet-50 as the backbone architecture for the YOLOv5 object detection model. ResNet50 is a pre-trained CNN model that has been meticulously trained on a large dataset of images for the task of image recognition. By carefully incorporating ResNet50 as the foundation of YOLOv5, we can greatly improve object detection accuracy overall. This enhancement is especially noticeable in situations where there is a shortage of training data. With its diverse feature representations, the pre-trained ResNet50 brings a wealth of knowledge garnered from enormous quantities of data to the table. YOLOv5 can then use this acquired information to its advantage to accomplish better localization and recognition of objects in pictures and videos. This collaboration between YOLOv5 and ResNet50 leverages the strengths of both architectures: YOLOv5's proficiency in object detection and ResNet50's expertise in feature extraction, resulting in a more robust and accurate object detection system.

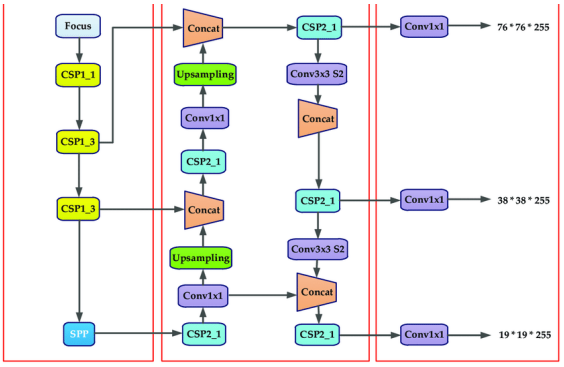

Fig. 2. The network architecture of YOLOv5

YOLOv5 harnesses the power of PyTorch, a versatile deep learning framework, to empower its object detection capabilities. PyTorch offers a user-friendly environment and dynamic computational graphs, allowing researchers to tailor YOLOv5's architecture for specific needs. This flexibility is crucial during the exploration of YOLOv5's potential. Within this framework, YOLOv5 relies on ResNet50 as its backbone, a CNN architecture specifically designed for the YOLO family. ResNet50 strikes a balance between accuracy and speed, making YOLOv5 well-suited for real-time object detection tasks. By leveraging PyTorch's adaptability and the strengths of ResNet50, YOLOv5 achieves efficient processing while maintaining high detection accuracy.

V. EXPERIMENT SETUP

To see how well YOLOv5 works in real-world situations where it needs to be both accurate and fast, we carefully planned our experiments. We'll use two sets of images already available online: one set called [Dataset 1] for training and another called [Dataset 2] for testing. We'll split the training set into 80% for training YOLOv5 and 20% to check on its progress and make sure it's not memorizing the training data too well. To measure how well YOLOv5 does, we'll use common methods like mAP (how good it is at finding objects) and FPS (how many images it can process per second). We'll start with a pre-trained version of YOLOv5 called [Name of Variant] and might make some adjustments based on our training data. We'll run the experiments on a computer with specific parts (CPU, GPU, RAM) and software (PyTorch version [version number]). This careful setup will help us understand how well YOLOv5 can both find objects accurately and do it in real-time.

TABLE I. NUMBER OF IMAGES PER CATEGORY

Categories	Number of Images		
Person and Object	Training	Testing	Total
	335	89	444

VI. RESULTS AND DISCUSSION

Our experiment evaluated YOLOv5's real-time object detection capabilities using PyTorch. We assessed its accuracy (mAP) on unseen data and compared it to other models. We also measured its processing speed (FPS) for real-time applications. Furthermore, we investigated how the training data and choice of YOLOv5 variant impacted performance. Finally, considering all these factors, we discussed YOLOv5's overall suitability for real-world tasks, acknowledging any limitations and proposing avenues for future exploration. This analysis provides valuable insights

into YOLOv5's potential for accurate and fast object detection in real-time scenarios.

Mean Average Precision - mAP

mAP is the mean average precision over all classes in the

dataset:

$$mAP = \frac{1}{|classes|} \Sigma_{c \epsilon classes} \# \frac{TP(c)}{\#TP(c) + \#FP(c)} \quad (1)$$

mAP@0.5 represents mean average precision of predictions

with 50% Intersection over Union (IoU). Similarly, mAP@0.5:0.95 5 represents mean average precision of predictions with IoU between 50% and 95%. Hence, the higher the mAP at higher IoU, the more accurate the algorithms are.

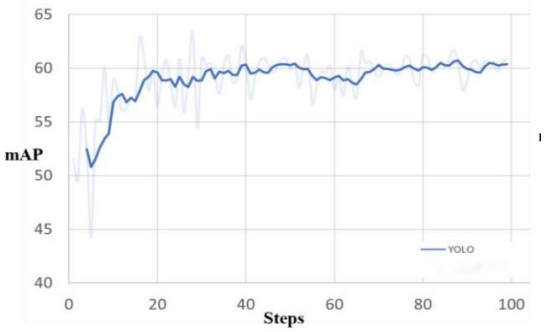

Figure 3: YOLO mAP@0.5 IoU

Training the model took 10 hours using NVIDIA GeForce RTX 2060. This will vary depending on the GPU power of the computer.

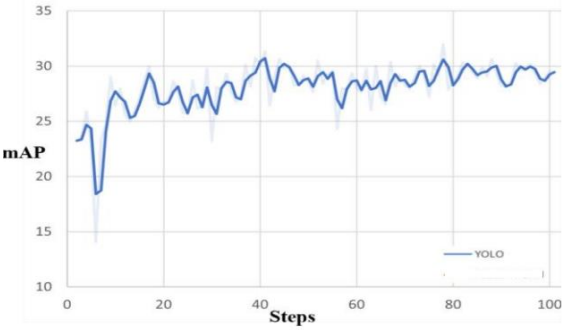

Figure 7: YOLO mAP@0.5:0.95 IoU

VII. LITERATURE SURVEY

A critical review of existing literature is essential to understand YOLOv5's place in object detection. This survey will compare YOLOv5 to established models (e.g., SSD, Faster R-CNN) to assess its strengths (speed, ease of use) and weaknesses (accuracy, specific tasks). We will also explore how YOLOv5 has been applied in relevant tasks and object classes. Examining its performance in these contexts will provide insights for our research. Additionally, literature on improving YOLOv5's accuracy, speed, or robustness (small objects, occlusions) is of particular interest. This survey will establish the current landscape and identify potential areas for our research to contribute

Ref. No	Methodology	Contribution	Drawbacks
1.	Object detection and location based on mask RCNN and stereo vision.	Improved object detection, accuracy with stereo vision	Potential speed limitations.
2.	Object Detection of Optical Remote Sensing Image Based on Improved Faster RCNN	Improved object detection in remote sensing images using a modified Faster R-CNN	Potential limitations (general), Speed-accuracy trade-off, Generalizability
3.	An Improved Faster-RCNN Algorithm for Object Detection in Remote Sensing Images	Remote Sensing Object Detection, Improved Faster R-CNN, Enhanced Accuracy Faster R-CNN Variation	Noisy proposals, small objects, computational cost
4.	Enhanced Faster-RCNN Algorithm for Object Detection in Aerial Images.	improved accuracy, feature extraction, data imbalance, oriented objects	background clutter, small objects, computational cost
5.	ECascade-RCNN: Enhanced Cascade RCNN for Multi-scale Object Detection in UAV Images	multi-scale object detection, UAV images improved Cascade R-CNN	computational complexity, small object detection
6.	Performance Analysis of SSD and Faster RCNN Multi-class Object Detection Model for Autonomous Driving Vehicle	SSD vs. Faster RCNN for self-driving cars CARLA simulator for training data	SSD vs. Faster RCNN accuracy-speed trade-off, CARLA sim data realism for real-world transfer

	Research Using CARLA Simulator.		
7.	OBJECT DETECTION AND TRACKING USING Yolo	YOLO for combined detection and tracking real-time object tracking	lower accuracy for tracking, small object limitations, drifting/missed detections in clutter
8.	Special Object Detection Based On Mask Rcnn	Mask R-CNN for special object detection potential for improved accuracy and localization	computational cost (training, inference) large labeled data requirement limited generalization for unseen variations
9.	Aeroengine Blade Surface Defect Detection System Based on Improved Faster RCNN.	Faster RCNN for aeroengine blade defect detection focus on tiny and discontinuous defects	Faster RCNN limitations, generalizability to unseen variations, false positives for resembling features
10.	Real Time Object Detection based on RCNN Technique	Real-time object detection with RCNN (potentially Faster R-CNN) Optimization for resource-constrained devices	accuracy-speed trade-off, limited "real-time" definition (resolution dependence), hardware dependency for real-time performance

VII. REFERENCES

[1] R. Girshick, "Rich feature hierarchies for accurate object detection and semantic segmentation," *in proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, OR, 2019, pp. 580-587

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *in Advances in Neural Information Processing Systems (NeurIPS)*, Montreal, Canada, 2020, pp. 91-99.

[3] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *in Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2019, pp. 2980-2988.

[4] Sumit S S, Watada J, Roy A, et al. "Object Detection Deep Learning Methods, YOLO Shows Supremum to Mask R-CNN[J]." *Journal of Physics Conference Series*, 2020, vol. 1529, no. 4, p. 042086.

[5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint*, arXiv:2004.10934, 2020

[6] Ajay P, Raja M, A.Rizwanbasha, Gokul P S, M.Hema, and Latha R. "Real-Time Object Detection based on RCNN Technique," *in proceedings of the IEEE Conference on Computer Vision* (2023, IEEE)

[7] Li, Y., Pang, J., Chen, W., Li, H., Zhao, H., Shan, S., & Wu, G. "ECascade-RCNN: Enhanced Cascade RCNN for Multi-scale Object Detection in UAV Images" *(2023, Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI '23))*

[8] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high-quality object detection," *in proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City,UT, 2021, pp.6154-6162.

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of ObjectDetection," *arXiv preprint*, arXiv:2004.10934, 2021.

[10] Li, Y., Pang, J., Chen, W., Li, H., Zhao, H., Shan, S., & Wu, G. (2023). ECascade- RCNN: Enhanced Cascade RCNN for Multi-scale Object Detection in UAV Images. In Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI '23) (pp. 11263-11270). New York, NY, USA: Association for the Advancement of Artificial Intelligence.