

Design and Implementation of a Real-Time Object Detection System using YOLO and Deep Learning

Co-Author - **Mr. Y. DayanandKumar**

Assistant Professor

Department of Computer Science C Artificial intelligence Central University of Andhra Pradesh, Ananthapuramu,
India

Email : dayanandkumar@cuap.edu.in

Author –**Tuppala Lakshmanna**

Department of Computer Science C Artificial intelligence Central University of Andhra Pradesh, Ananthapuramu,
India

Email :lakshmannatuppala492@gmail.com

Abstract

In this paper, an object detection system implemented using a deep-learning algorithm is discussed. Specifically, a real-time object detection system is built using the YOLO algorithm to detect various objects present in an image or video feed.

To train the model, various object classes are used to ensure that detection works under changing scenarios involving changes in light, background, or size.

The proposed system employs the full-stack approach, whereby processing is done in the back-end through implementation of the YOLO algorithm on the image and videos, whereas detection results visualization is done in the front-end. In evaluating the performance of the system, the metrics used include precision, recall, mean Average Precision, and FPS.

Based on experimental results, the model developed balances efficiency with performance. Although there were some challenges in detecting small objects and/or those that are partially occluded, overall, the detection process was efficient.

Keywords: Object Detection, YOLO, Deep Learning, Computer Vision, Real-Time Detection

I. INTRODUCTION

Artificial Intelligence (AI) is one of the 21st century's most impactful technologies which heavily impacted system design and implementation in fields such as healthcare, finance, autonomous vehicle, smart city etc. Amongst AI, computer vision is playing a very important role since computer vision provides machines to understand and analyse the information in images and videos, a larger portion of which is real world information.

The focus of computer vision is enabling machine to understand image and video in a human like fashion. A specific task that is of great importance within computer vision is that of object detection where the goal is to detect objects and locate them in an image/video using bounding boxes and labels. Unlike image classification where the output for the image will just be a category of an object present within, in object detection, not only will the category of the object be recognized but also its location be given along with bounding box. This helps the machines to realize objects and thus has applications in many areas where

an object must be identified in real world, unlike simple image classification, where only the recognition is needed .

The application areas where object detection is getting widely used include autonomous driving, security and surveillance, medical images, analysis in retail store, etc. \cite{ref10}. There are some challenges which exist when implementing object detection. The main difficulties associated with object detection are: Variation in lighting conditions, different object sizes, occlusion and complex background . It is also imperative for objects to be detected in real time since failure to identify an object might be critical for time-sensitive applications, causing wrong decisions and possibly system failure.

Traditional object detection relied on machine learning like Support Vector Machine (SVM), which require manual selection of features but had limited generalization . The introduction of Deep learning has led to great increase in detection performance since the networks have the ability to automatically extract and learn features through hierarchal composition which are highly discriminative .

Today object detection can be primarily categorized into region-based and single-shot approaches. Region based approaches such as Faster R-CNN provide high accuracy but requires more computation and is time consuming. On the other hand, single shot object detection such as YOLO model is able to perform detection at faster pace, which is suitable for real-time applications.

II. LITERATURE REVIEW

Object detection has been a subject of research in computer vision for several decades, and approaches have progressed from handcrafted features to deep learning methods. Handcrafted features such as SIFT and HOG coupled with classical machine learning classifiers such as Support Vector Machines (SVM) were initially used to detect objects. While these methods achieved some initial success, they had many limitations in detecting objects in real-world scenarios where the scene is very complex. Deep learning methods enabled automatic feature learning from images and had huge impacts on object detection task. Models such as AlexNet and later ResNet were a success and provided very good feature representations of images which improved the detection accuracy tremendously[1].

Region-based methods of object detection represent an important milestone which made object detection more accurate. The R-CNN algorithm detected objects by using Selective Search to generate regions in the image, and classified these regions using a CNN. Although the detection accuracy was improved, R-CNN was very slow due to redundant computation on the proposal regions. By processing each proposal region only once after classifying the whole image, Faster R-CNN further reduced computation while achieving excellent accuracy by integrating the region proposal mechanism into the CNN itself (RPN) and eliminating the need for Selective Search[2].

Region-based object detection systems typically run relatively slowly. So, some authors concentrated on developing single-stage methods. Single Shot MultiBox Detector (SSD) is one of such models which proposes that each part of the image can contain objects, and perform both localization and classification in a single pass of the CNN and achieve faster running speed with reasonable accuracy compared to region based detectors. However SSD model cannot detect small objects efficiently[7]. YOLO algorithm which stands for You Only Look Once has attracted a lot of interest. It proposes that the object detection task could be framed as a regression problem, thus, the system predict bounding boxes and class probabilities at the same time in a single forward pass which allows YOLO to have real- time performance compared to other region based detection methods[3]. Different versions of YOLO algorithm such as YOLOv2, YOLOv3, YOLOv4 and YOLOv7 also provide more advanced features that further increased the accuracy of the object detection model[6]. In addition, methods such as Segment Anything that combines segmentation to improve object detection, large scale datasets such as Pascal VOC, Microsoft COCO, and other similar databases have accelerated the progress in this area[10].

Therefore, we selected YOLO as the base model to develop an object detection system, as it gives both speed and reasonable accuracy and suitable for real-time applications.

III. PROPOSED METHODOLOGY

This section outlines the methodology adopted in this work in order to develop a real time object detection system utilizing YOLO and deep learning. A step-by-step approach has been adopted to construct efficient and reliable object detection system.

The proposed methodology is based on the principles of data driven learning approach and deep neural networks. The methodology includes various steps: data collection, preprocessing, model selection, training, evaluation and deployment; and each step contributes to the successful performance of the object detection system[9].

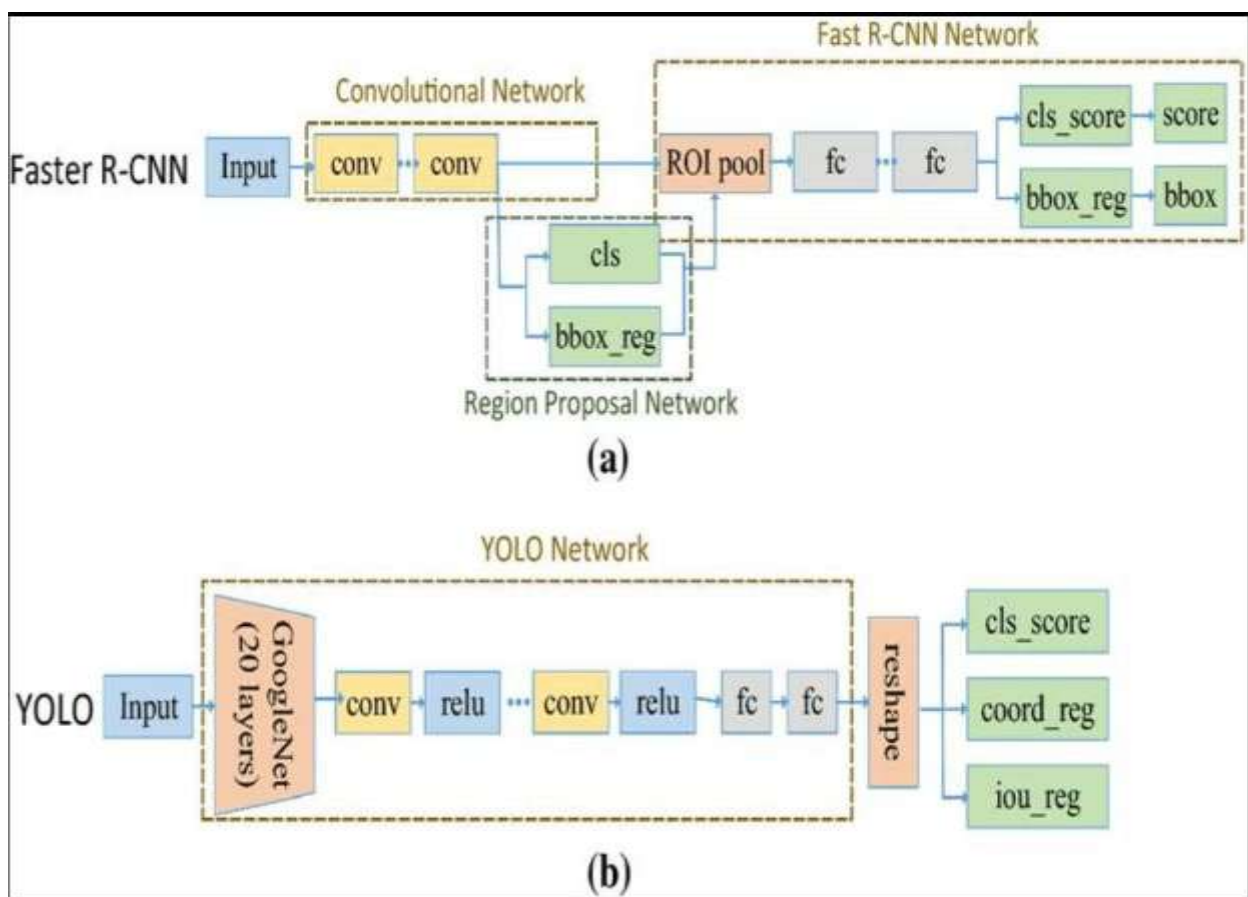
The initial stage includes data collection. An appropriate dataset is selected for training and testing purposes. The performance of the object detection system is directly related to the quality and variability of the dataset[10].

Data preprocessing comes after data collection. This phase comprises image resizing, normalization, labeling of the images by bounding box and data augmentation such as rotation, scaling and flipping. Augmentation increases variability of dataset and avoid overfitting and the model shows better performance in real environment

The model selection phase has been adopted in the next stage. In this system YOLO model has been chosen due to its ability to perform object detection at a single pass and hence it is highly suitable for real time detection[8].

In the subsequent step, the trained model undergoes training with a prepared data. During the training phase, the model learns to detect object and identify it accurately based on the labeled data.

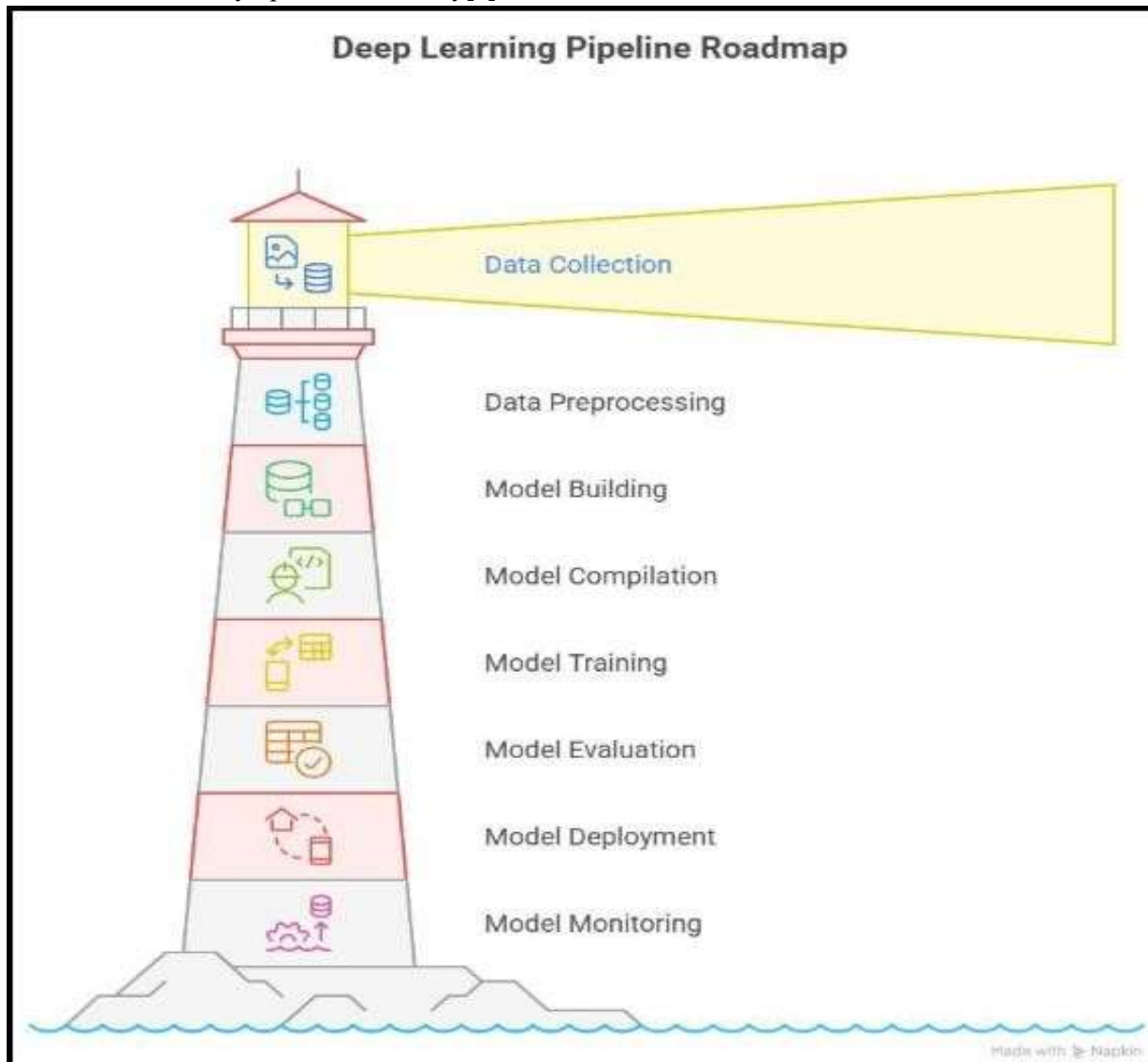
Once the model is trained, the model is evaluated with the test data set. The performance parameters, which includes precision, recall and mAP (mean Average Precision), is determined to define the system performance[11].



In the last stage, the trained model is implemented in the real time system, which means, it operates over the live stream video or

image and output the results in the form of bounding box and labels. It ensures the usability in real world scenario[2].

In short, it offers a systematic framework to establish a real-time object detection system which provide an optimum trade-off between efficiency, speed and accuracy[4].



1. YOLO Functionality

YOLO will predict bounding boxes for detected objects using four variables. (x,y,w,h)

x and y give the center of the bounding box while w and h give the width and height of the bounding box. These parameters are scaled to the image size so that they will function on different images.

Additionally, to each bounding box we are assigning a confidence score:

$$\text{Confidence} = P(\text{Object}) \times \text{IoU}$$

This score states how confident is the detector on both the presence and location of an object.

2. Training Procedure

The extent of the overlap of the predicted bounding box with the ground truth is measured by IoU:

$$\text{IoU} = \text{Area of Overlap} / \text{Area of Union}$$

Higher the IoU value, greater is the overlap and more accurate is the detection. The YOLO network is trained by optimizing a loss function comprised of various components:

$$Loss = Localization Loss + Confidence Loss + Classification Loss$$

Localization loss is an error term to quantify the bounding box error. Confidence loss is to measure how well the confidence of the detected object corresponds to the prediction of whether it is present or not. Classification loss is the classification error.

3. Performance Metrics

The model's performance is judged by a standard set of metrics. Precision is defined as:

$$Precision = \frac{TP}{TP+FP}$$

Recall is defined as:

$$Recall = \frac{TP}{TP+FN}$$

The F1-score attempts to combine the two above:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The mAP is used as an aggregate measure of performance:

$$mAP = \frac{1}{N} \sum AP_i$$

4. Model Training

The preprocessed dataset is then used for training the YOLO model on over many epochs. While training, the model is learning to identify the objects, it aims at minimizing the loss function. The parameter update happens via methods such as gradient descent

5. Model Evaluation

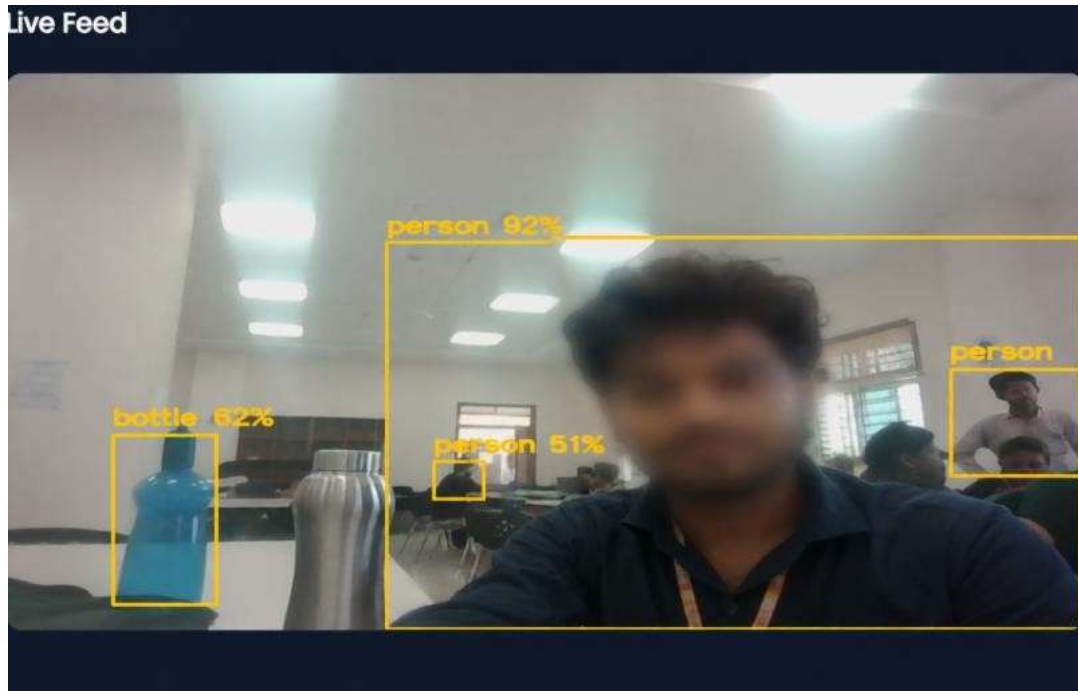
The model is tested after training and accuracy/reliability of the detection system is determined by different performance metrics, namely, precision, recall, mean Average Precision (mAP).

6. Deployment

In the last step, the model is put into use. In the system, we accept a live video stream from the webcam and detect objects in the stream. The results are plotted with bounding box, class label and the confidence level using a user interface.

IV. Experimental Result

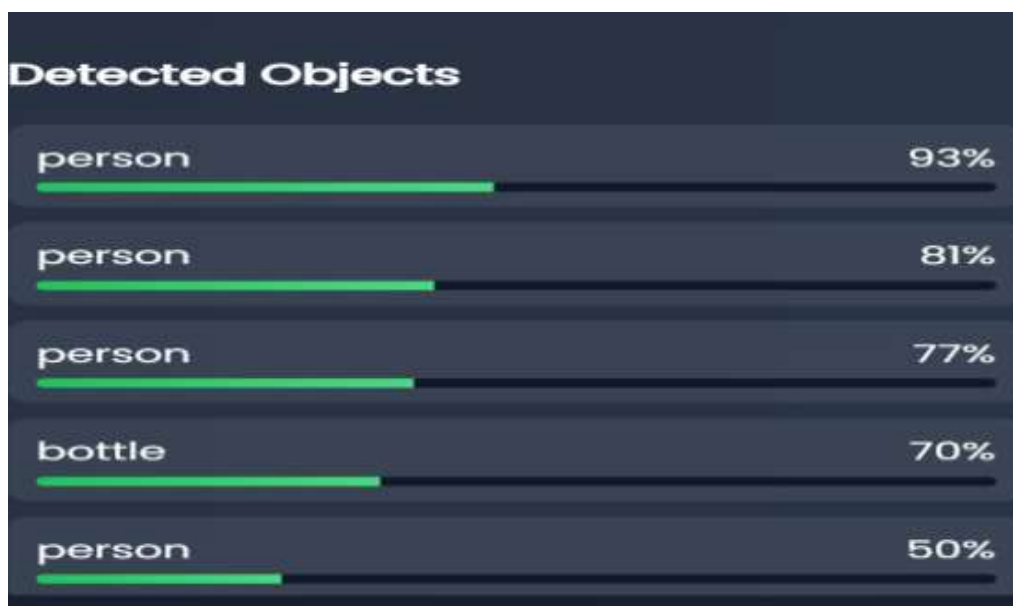
The quality and quantity assessment of the proposed YOLO based object detection system is explained below: The system is tested on real-time video sequence and static images.



The system correctly detects multiple objects within one frame and outputs bounding boxes along with class names and confidence scores. The system demonstrates its capability of detecting the objects under a challenging scene with multiple objects.

But there might be slightly lower detection accuracy when objects are extremely small or partially occluded. Nevertheless, the overall detection accuracy is quite well to achieve real time performance.

1. Accuracy Analysis



The accuracy plot shows how well the model performs throughout the training. It can be observed that accuracy is low when the model is learning the data for the first time and it then rises as the training continues. This clearly suggests that the model is learning the features as desired. Once accuracy reaches saturation point it means the model has converged to a result where the loss does not go any down and it hasn't overfit to the training data.



The accuracy graph should reach saturation/convergence stage after some epochs, where the accuracy is more or less same. This stage signifies that the system has learnt major features and would only show minor increase in accuracy and not lead to overfitting. The smooth increasing and converging trend of the accuracy is desirable.

3. Loss Analysis Graph

The loss graph depicts the discrepancy between the predicted output values and the ground truths. Initially, during the training process, the value for the loss is high as the model does not predict the actual output value accurately. This is because the model has not yet learnt any important features.



During the training process, the value for the loss starts to decrease and it becomes stable and the accuracy of prediction increases as the model learns the different features. A graph which depicts continuous reduction for loss means that it is effectively training the model

\cite{ref1}.

After few epochs the value of loss gets flatter. It implies that the model has started to behave the same at the final step and the learning process have completed. It implies that the model has achieved stable state for all its parameters and no further improvements can be done for this model. A continuous reduction graph with decrease indicates that it has been trained well.

If the loss gets fluctuating or starts to rise, it is possibly because of over fitting or poor tuning of parameters. For this system, loss is constantly decreasing, implying that it is a stable system \cite{ref2}.

V. future Scope

The suggested real-time object detection using YOLO can achieve good results but there are still several areas that could be enhanced or expanded. Further work could aim to develop the dataset by increasing the size and variety of images used so that varying illumination conditions, object orientations, and background scenes are included to improve generalisation. Enhancing detection of small and occluded objects through implementation of advanced techniques such as multi-scale feature extraction and appropriate network architecture can be explored. Model optimization through techniques such as pruning, quantisation, and knowledge distillation could decrease computation cost and be used for deployment on low power embedded devices. Use of advanced versions of YOLO with better performance will result in improvement in the detection accuracy and speed of the system. Addition of object tracking functionality enables the system to perform long term tracking, which is desirable in traffic or surveillance applications. Cloud based deployment will make the system highly scalable, remotely accessible and could support smart city and automated monitoring systems. The user interface can also be enhanced using the real time data analysis and visualization to support ease of usage. Extensions of the system to different data types such as thermal or medical images are another possibility to widen the range of applications. Deployment of the model on edge devices like mobile and embedded platforms will also allow high speed detection at the edge without having to rely on high-end hardware.

VI. Conclusion

The current research has achieved designing, implementing and testing a real-time object detection system by leveraging YOLO algorithm and deep learning techniques. Main goal in achieving faster and more accurate detection for images and real-time video streams has been successfully reached, it is able to detect multiple objects simultaneously while keeping a good trade-off between accuracy and processing speed suitable for real-time implementation.

The use of deep learning based backend and an interactive frontend, allow the system to display detections visually to the user and therefore increases its usability for real-time tasks. Detection results based on standard metrics (precision, recall, mAP, FPS) confirm that the model behaves correctly under several conditions and prove that YOLO-based approach shows a clear speed improvement in comparison with previous region-based methods.

There are few limitations in the developed system such as: loss of accuracy in small and partially occluded objects and dependence on hardware. However, the results and behavior are suitable for most of the real-world applications such as surveillance, traffic management, and intelligent environments.

In conclusion, this work shows a promising future of YOLO-based models for the implementation of efficient real-time object detection systems. The obtained results serve as a good base for further research and improvements in the field of computer vision.

Reference

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi,
“You Only Look Once: Unified, Real-Time Object Detection,”
Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016.
- [2] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” *Proc. IEEE CVPR*, 2017.
- [3] S. Ren, K. He, R. Girshick, and J. Sun,
“Faster R-CNN: Towards Real-Time Object Detection,”
IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.
- [4] J. Redmon and A. Farhadi,
“YOLOv3: An Incremental Improvement,”
2018.
- [5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao,
“YOLOv4: Optimal Speed and Accuracy of Object Detection,”
2020.
- [6] C.-Y. Wang *et al.*,
“YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art,” 2022.
- [7] W. Liu *et al.*,
“SSD: Single Shot MultiBox Detector,”
Proc. European Conference on Computer Vision (ECCV), 2016.
- [8] A. Krizhevsky, I. Sutskever, and G. Hinton,
“ImageNet Classification with Deep Convolutional Neural Networks,”
Proc. NIPS, 2012.
- 9] K. He, X. Zhang, S. Ren, and J. Sun,
“Deep Residual Learning for Image Recognition,”
Proc. CVPR, 2016.
- [10] T.-Y. Lin *et al.*,
“Microsoft COCO: Common Objects in Context,”
Proc. ECCV, 2014.
- [11] R. Al Amin and R. Obermaisser,
“Real-Time Object Detection and Classification using YOLO for Edge FPGAs,”
Proc. 7th International Symposium ELMAR, 2025.