IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

Efficient Object Detection with YOLO: A Comprehensive Guide

Suvarna Patil, Soham Waghule, Siddhesh Waje, Prasad Pawar, Shreyash Domb
Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, Maharashtra, India

Abstract: *Object detection presents itself as a pivotal and complex challenge within the domain of computer vision. Over the past ten years, as deep learning techniques have advanced quickly, researchers have committed significant resources to utilising deep models as the basis to improve the performance of object identification systems and related tasks like segmentation, localization. Two-stage and single-stage detectors are the two basic categories into which object detectors can be roughly divided. Typically, two-stage detectors use complicated structures in conjunction with a selective region proposal technique to accomplish their goals. Conversely, single-stage detectors aim to detect objects across all spatial regions in one shot, employing relatively simpler architectures. Any object detector's inference time and detection accuracy are the main factors to consider while evaluating it. Single-stage detectors offer quicker inference times, but two-stage detectors frequently show better detection accuracy. But since the introduction of YOLO (You Only Look Once) and its architectural offspring, detection accuracy has significantly improved—sometimes even outperforming that of two-stage detectors. The adoption of YOLO in various applications is primarily driven by its faster inference times rather than its detection accuracy alone.*

Keywords: YOLO, Object Detection, Keras, Open CV, CNN, R-CNN, Tensor Flow, YOLO V3, Yolo NAS

I. INTRODUCTION

A fundamental component of computer vision, object detection propels advances by leveraging diverse Deep learning (DL) and Machine learning (ML) models. Two-stage object detectors have always been popular and effective in the field of object detection. But single-stage object identification methods and underlying algorithms have advanced recently, outperforming many conventional two-stage detectors in terms of performance. The introduction of YOLO (You Only Look Once) models has further revolutionized the landscape, with applications across various contexts showcasing remarkable performance compared to their two-stage counterparts. This has inspired the focus of this review, aiming to delve into the intricacies of YOLO and its architectural successors. Through a detailed exploration of their design nuances, optimizations, competitive edge over two-stage detectors, and other pertinent aspects, the goal of this paper is to offer a thorough grasp of the developing field of object detection. This section provides a brief introduction to computer vision and deep learning technology, explaining important terms, difficulties, phases, and their importance in applying object identification methods. It also discusses widely used datasets, the evolutionary history of different object identification algorithms, and the main goals and to help this review.

1.1 Motivation

There are several strong reasons to start a YOLOv3 object detection project. First off, the importance of these initiatives in the actual world cannot be emphasized. The uses are numerous and include everything from bettering surveillance systems to boosting autonomous car efficiency and easing medical imaging. Developers can explore the depths of neural networks and machine learning, staying at the forefront of computer vision innovation, by utilizing YOLOv3's state-of-the-art technology. This offers a great opportunity for learning, enabling people to improve their abilities while taking on challenging tasks. Furthermore, object detection models' adaptability allows them to be customized to fit a range of sectors and use cases, which promotes innovation and creativity. Interacting with the dynamic community centered around computer vision and deep learning presents chances for cooperation and information sharing, further

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

enhancing the experience. Being proficient in YOLOv3 and object detection can lead to exciting employment opportunities in computer vision engineering, data science, and artificial intelligence. Ultimately, the sense of accomplishment that arises from witnessing an accurate object identification and localization by a YOLOv3-based model is a potent motivator that propels enthusiasts to overcome obstacles and push the limits of computer vision technology.

1.2 Problem Statement

To meet the increasing demand for reliable computer vision solutions across multiple disciplines, create an accurate and efficient object detection system using YOLOv3, OpenCV, and TensorFlow. The aim is to develop a system that can identify and pinpoint objects of interest in either static photos or real-time video streams, while prioritizing speed, accuracy, and scalability. The system must to possess sufficient flexibility to handle various object classes and be able to adjust to varying lighting situations and surroundings. Furthermore, it ought to give precedence to optimization methods in order to guarantee peak performance for devices with limited resources, like edge devices or embedded systems. The suggested solution should have an intuitive user interface that makes it simple for developers and end users to deploy and utilize, and it should be able to integrate with current frameworks or apps. The overall objective is to utilize the advantages of YOLOv3, OpenCV, and TensorFlow to provide a cutting-edge object identification system that satisfies the requirements of contemporary computer vision applications.

1.3 Objective

This project aims to create a reliable and effective object identification system using TensorFlow, OpenCV, and YOLOv3. Our main goal is to prioritize speed and scalability while achieving high accuracy in real-time object identification within static photos or video streams. By putting the YOLOv3 architecture into practice, utilizing TensorFlow for model training on unique datasets, and including OpenCV for pre-, post-, and visualization processing, we hope to accomplish this. In order to guarantee real-time performance, we optimize to minimize latency and enhance throughput. Furthermore, we want to improve robustness against a range of scenarios, including background clutter, occlusions, lighting conditions and object sizes.The system will have an easy-to-use interface that allows for smooth integration with current frameworks, making deployment for developers and end users simple.

II. LITERATURE REVIEW

2.1 Literature Review of Research Papers

Following table gives the review of research papers we have undergone inorder to understand implementation and working of YOLO-

Sr. No	Title	Author	Drawback Year
1.	YOLO-based Object Detection Models: A Review and its Applications	Ajantha Vijayakumar & Subramaniyaswamy Vairavasundaram	2023
2.	Real Time Object Detection System with YOLO and CNN Models: A Review	Viswanatha V, Chandana R K, Ramachandra A.C.	2023
3.	A review of Yolo Algorithm Developments	Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, Bo Ma	2022
4	Object detection using YOLO: challenges, architectural successors, datasets and applications	Tausif Diwan,G.Anirudh,Jitendra V. Tembhurne	2022

2.2 Table of comparative Analysis

Following is comparative analysis between different models for different object detection models.

Aspect	Approach 1(YOLO)	Approach 2 (SSD)	Approach 3(Faster R-CNN)
Framework	Tensor Flow	Tensor Flow	Tensor Flow

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

Object Detection Model	YOLO V3	SSD	Faster R-CNN
Real Time Performance	High	Moderate	Low
Accuracy	High	Moderate	High
Training Time	Moderate	High	High
Inference Speed	Fast	Moderatre	Moderate
Model Size	Medium	Large	Large
Flexibility	Limited	Moderate	High
Implementation Ease	Moderate	Moderate	Low
Resource Consumption	Moderate	High	High

III. PROPOSED APPROACH

A suggested method for object detection that makes use of TensorFlow, OpenCV, and YOLOv3:

Understanding the Problem and Acquiring the Dataset: - Recognize the goals and specifications of the object detection project. Determine the kinds of items that need to be discovered as well as the environment in which they should be found. To train the object detection model, obtain or produce a suitable collection of annotated photos.

Preprocessing and Data Augmentation: Make sure that the image quality, size, and format are consistent throughout the dataset by performing preprocessing.

Use data augmentation methods to improve model generalization by increasing the diversity of training samples, such as rotation, scaling, flipping, and color jittering.

Model Selection:

Evaluate different object detection models such as YOLOv3, SSD, and Faster R-CNN based on their performance metrics, speed, and resource requirements.

Select YOLOv3 as the primary object detection model due to its balance between accuracy and real-time performance.

Training the YOLOv3 Model:

Set up the training environment with TensorFlow and GPU support for accelerated training.

Configure the YOLOv3 architecture for the specific object detection task and dataset.

Train the YOLOv3 model using the annotated dataset, optimizing hyperparameters such as learning rate, batch size, and number of epochs.

Model Evaluation and Optimization:

Evaluate the trained YOLOv3 model using validation data to assess its accuracy, precision, recall, and mean average precision (mAP).

Integration with OpenCV: Use OpenCV to integrate the trained YOLOv3 model for object identification and inference in real-time on static photos or video streams.

Create programs or scripts to preprocess incoming data, use the YOLOv3 model for inference, and use OpenCV to visualize objects that are discovered.

Testing and Validation: To guarantee robustness and dependability, test the integrated system in a variety of settings, situations, and datasets.

Verify the object detection system's accuracy and performance using benchmark datasets and ground truth annotations.

The entire development process, including dataset preparation, model training, OpenCV integration, and testing protocols, should be documented. This includes both deployment and documentation.

Provide documentation and user manuals so that the object detection system can be implemented in real-world settings.

Construct executable files or libraries from the system to facilitate distribution and deployment.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

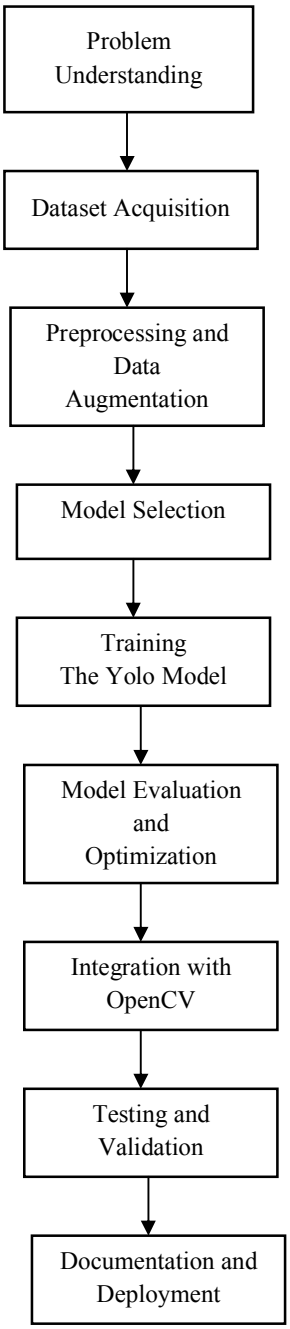

Fig 1. Flow of Proposed Chart

IV. YOLO VERSIONS AND AND THIR ARCHITECTURES

4.1 YOLO v1

The YOLO algorithm uses a straightforward method for object detection, using deep convolutional neural network to process images and detect objects within them. The YOLO's CNN model consists of 20 convolution layers, initially pretrained on the ImageNet dataset. This pretrained model undergoes a conversion process to facilitate object detection, a strategy found to enhance performance in previous studies. The final fully connected layer of YOLO is responsible for predicting class likeliness and bounding box positions.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

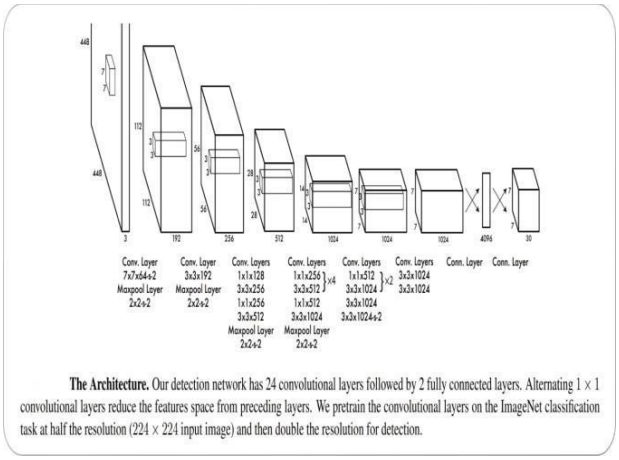

Fig 2. Yolo V1 architecture.

YOLO divides the input image into a S × S grid and gives each grid cell the job of identifying objects whose centers are within it. YOLO estimates B bounding boxes within every cell of the grid and provides confidence scores that represent the model's level of assurance regarding the existence of an item and the precision of the bounding box prediction. YOLO uses a technique that chooses the statistic with the greatest Intersection over Union with the underlying reality to guarantee that only one bounding box predictor is in charge of each item throughout training. Bounding box predictors that specialize improve their ability to foresee particular sizes, proportions, or item classes, which raises the recall score overall.

Non-maximum suppression is a crucial post-processing technique used in YOLO models to improve object identification efficiency and accuracy. By locating and eliminating unnecessary or inaccurate boxes, NMS resolves the problem of several bounding boxes being produced for a single object, eventually producing an unique bounding box for every item in the image.

Later iterations of YOLO have improved and refined the initial model, augmenting its capabilities and performance. These developments will be discussed in more detail in the debate that follows in order to clarify how YOLO has developed throughout time.

4.2 YOLO v2

As an improved version of the original YOLO algorithm, YOLO v2 debuted in 2016 with the goal of increasing detected item classes while improving speed and accuracy. This version used an altered version of the VGGNet architecture called Darknet19, which is a new convolutional neural network core that includes successive convolution and pooling layers.

The incorporation of anchor boxes, a preset set of bounding boxes with varying aspect ratios and scales, is a creative improvement in YOLO v2. When combined with the estimated offsets, these anchor boxes make it easier to determine the final bounding boxes, which allows the technique to work with objects of different dimensions and perspective ratios.

Furthermore, batch normalization was incorporated into YOLO v2 to improve model stability and accuracy. Using a multi-scale training approach improved the detection performance much more, especially for smaller items. Another significant development was the addition of a new loss function specifically designed for object detection tasks. This new function depends on the sum of squared errors between predicted and actual bounding boxes and class probabilities.

YOLO v2 showed significant improvements over its predecessor and contemporaneous models, highlighting its effectiveness in object detecting tasks.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

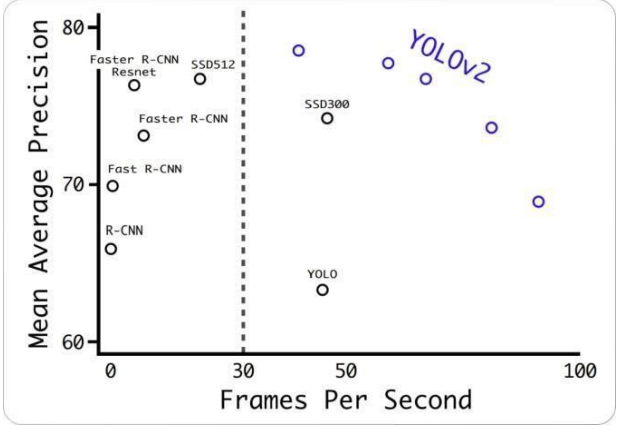

Fig 3. R-CNN comparison with Yolo and SSD300

4.3 YOLO v3

The third version of the YOLO object detection algorithm, known as YOLO v3, was released in 2018. Its goal was to improve speed and accuracy over YOLO v2's predecessor.

One of the main improvements in YOLO v3 is the use of a new CNN architecture called Darknet-53. With 53 convolutional layers, this ResNet-based architecture outperforms others in a variety of object detection benchmarks.

In addition, anchor boxes of different sizes and proportions are included in YOLO v3, which differs from YOLO v2's consistent anchor box size. This adaptability improves the algorithm's capacity to identify objects with different dimensions and forms.

The inclusion of feature pyramid networks , a CNN architecture capable of object detection at several scales, is another noteworthy addition to YOLO v3.

Feature-painting networks build a hierarchy of feature maps, where each tier serves objects of varying sizes. This improves detection performance, especially for smaller items.

With these improvements, YOLO v3 outperforms its predecessors in terms of accuracy, stability, and adaptability, securing its place as the industry-leading object detection algorithm.

4.4 YOLO v4

The primary advancement in YOLO v4 compared to YOLO v3 lies in its utilization of a novel CNN architecture known as CSPNet, denoting "Cross Stage Partial Network." This architecture is designed primarily for object detection tasks; it is an outgrowth of the ResNet framework. With just 54 convolutional layers, CSPNet has a rather shallow structure, but it performs remarkably well on a variety of object detection benchmarks.

To better align with the size and shape of detected objects, YOLO v3 and YOLO v4 use anchor boxes with different scales and aspect ratios. YOLO v4 presents a brand-new method for creating anchor boxes called "k-means clustering." In this method, ground truth bounding boxes are grouped into clusters using a clustering algorithm, and then the centroids of these clusters are used as anchor boxes. By ensuring a closer alignment between anchor boxes and identified objects, this improvement improves the accuracy of detection.

While the loss function used for model training is the same in both versions, YOLO v4 adds a new component called "GHM loss," which is a variation of the focal loss function designed to improve model performance on unbalanced datasets. Furthermore, the architecture of Feature Pyramid Networks is improved by YOLO v4.

4.5 YOLO v5

YOLO v5, released in 2020 and maintained by Ultralytics, marks a significant advancement within the YOLO series. Departing from its predecessors, YOLO v5 implements a more sophisticated architecture known as EfficientDet, which is rooted in the EfficientNet network design. This architectural refinement empowers YOLO v5 to achieve heightened precision and enhanced adaptability across a wider spectrum of object categories.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

When it comes to training data, YOLO v5 approaches it differently than its predecessors. Although the PASCAL VOC dataset, which included 20 item categories, was used to train the first YOLO model, YOLO v5 makes use of the D5, a larger and more varied dataset that includes 600 object categories.

A notable innovation introduced by YOLO v5 is the utilization of dynamic anchor boxes for anchor box generation. To do this, ground truth bounding boxes are grouped using a clustering method, and the centroids of these groups are then used as anchor boxes. This adaptation ensures a finer alignment between anchor boxes and the objects detected by the model.

Furthermore, this version integrates the concept of pyramid pooling, a specialized pooling layer aimed at improving detection performance on smaller objects by enabling the model to perceive objects across multiple scales. While SPP is also utilized in YOLO v4, YOLO v5 enhances the SPP architecture to achieve superior results.

While YOLO v4 and YOLO v5 use the same loss function to train their models, YOLO v5 adds a new element called "CIoU loss." Specifically designed to improve model performance on datasets with uneven class distributions, this version of the IoU loss function.

4.6 Yolo V9

The YOLOv9 algorithm represents a significant advancement in the realm of object detection, offering real-time performance with impressive accuracy. This paper provides a thorough examination of the YOLOv9 algorithm, delving into its architecture, training process, and performance metrics. Through a comprehensive review, this research aims to elucidate the key components of YOLOv9, its strengths, limitations, and potential applications in various domains.

A fundamental task in computer vision, object detection has many applications in fields such as autonomous driving and surveillance systems. In this area, YOLO (You Only Look Once) became a trailblazing method by providing a unified framework for concurrent object localization and classification.

The latest iteration, YOLOv9, builds upon its predecessors, incorporating enhancements to further improve accuracy and speed.

The architecture of YOLOv9 comprises multiple components, including backbone networks, feature pyramid networks, and detection heads.

Notable improvements in YOLOv9 include the integration of advanced backbone networks such as CSPDarknet53 and SPP (Spatial Pyramid Pooling) modules, facilitating richer feature representation and enhanced context perception. Additionally, the introduction of PANet (Path Aggregation Network) enables effective feature fusion across different scales, leading to improved localization accuracy.

Training YOLOv9 involves several stages, including data preparation, network initialization, and iterative optimization. One of the distinctive features of YOLOv9 is its utilization of a blended approach for training, using unsupervised and supervised techniques. By leveraging labeled data along with self-supervised learning objectives such as contrastive loss, YOLOv9 achieves superior generalization capability and robustness to variations in input data.

Evaluation of YOLOv9's performance encompasses metrics such as mean Average Precision, inference speed, and memory efficiency. Comparative analysis with previous YOLO versions and object detection algorithms demonstrates the efficacy of YOLOv9 in achieving a favorable balance between accuracy and speed. Real-world deployment scenarios further validate its suitability for applications requiring real-time object detection in resource-constrained environments.

The versatility of YOLOv9 opens up opportunities for deployment in diverse domains, including autonomous vehicles, surveillance systems, and industrial automation. Future research directions may involve exploring domain-specific adaptations of YOLOv9, incorporating additional modalities such as lidar or radar data for improved perception. Furthermore, advancements in hardware acceleration and model compression techniques hold promise for further enhancing the efficiency of YOLOv9 in embedded systems.

In conclusion, YOLOv9 represents a milestone in the evolution of object detection algorithms, offering a compelling combination of accuracy, speed, and versatility. Through a comprehensive review of its architecture, training process, and performance characteristics, this paper elucidates the underlying principles and potential applications of YOLOv9. Continued research and development efforts are poised to unlock further advancements, paving the way for transformative innovations in computer vision and beyond.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

V. APPLICATIONS

There are several real-world uses for the object detection project that makes use of YOLOv3, OpenCV, and TensorFlow in a variety of fields. Among the important applications are:

Surveillance Systems: Identify and follow items of interest in real-time video streams automatically to improve surveillance systems.

By spotting unlawful entries, suspicious activity, or abandoned items in prohibited areas, you can increase security.

Autonomous Vehicles: Facilitate the ability of autonomous vehicles to identify and categorize things in their environment, including bicycles, cars, pedestrians, and traffic signals.

By giving drivers real-time notice of potential roadblocks and hazards, you can increase safety.

Retail Analytics: Use object detection to analyze consumer behavior and preferences to improve your retail analytics.

To improve customer engagement and sales, track customer movements, pinpoint popular items or sections within stores, and optimize store layouts.

Help medical personnel interpret images from scans, including MRIs, CT scans, and X-rays.

Locate anomalies, cancers, or other illnesses to aid in early diagnosis and treatment planning.Automation of object detection tasks, such as spotting flaws in production processes or keeping an eye on inventory levels in warehouses, can increase productivity and safety in industrial environments.

Smart agriculture involves identifying and keeping an eye on crops, livestock, and equipment in order to optimize agricultural processes.

To improve crop management, enable precision agriculture tools including yield estimation, pest identification, and crop monitoring.

VI. ADVANTAGES AND DISADVANTAGES

6.1 Advantages

1. Achievement in real time: Because of its quick inference speed, YOLOv3 is well-suited for real-time object identification applications including video analytics, autonomous cars, and surveillance.
2. Excellent Precision: In object detection tasks, YOLOv3 can achieve high accuracy, particularly when trained on vast and diverse datasets. For applications that require precise object localization, this accuracy is essential.
3. Adaptability: The project's integration of TensorFlow, OpenCV, and YOLOv3 offers a flexible framework for creating object identification solutions in a variety of industries, such as retail analytics, medical imaging, surveillance, and more.
4. Simplicity of Integration: Robust frameworks and APIs are available for image processing, deep learning, and computer vision tasks in OpenCV and TensorFlow. By integrating these libraries with YOLOv3, deployment can be done seamlessly and the development process is made simpler.
5. Assistive Community:There are sizable and vibrant developer, researcher, and enthusiast communities for YOLOv3, OpenCV, and TensorFlow. Faster development and debugging are made possible by the abundance of tools, tutorials, and pre-trained models that are made available.

6.2 Disadvantages

1. Implementation Difficulty: To create an object detection system with YOLOv3, OpenCV, and TensorFlow, one must have a firm grasp of computer vision methods, deep learning principles, and software development techniques. For inexperienced developers, the implementation's intricacy may provide difficulties.
2. Heavy on Resources: Deep learning models like YOLOv3 can be resource-intensive to train and implement, needing a lot of computing power and robust hardware like GPUs. This could restrict the project's scalability, particularly for apps installed on devices with limited resources.
3. Requirements for Data Annotation: To train a YOLOv3 model for object detection, a sizable and precise dataset annotation is required. Such datasets can be labor- and time-intensive to gather and annotate, particularly in specialized or niche sectors.
4. Optimization of the Model: Even though YOLOv3 has excellent accuracy and real-time speed, more work may be needed to optimize the model for particular applications and deployment scenarios. To lower the size of the

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

model and inference delay, methods including compression, quantization, and model pruning could be required.

5. Control of Dependencies: It can be difficult to manage dependencies and compatibility problems between OpenCV, TensorFlow, YOLOv3, and other libraries, especially when dealing with intricate software ecosystems. It could be necessary to pay close attention to ensuring stability and compatibility across various upgrades and versions.

VII. CONCLUSION

Object detection includes basic tasks including segmentation, localization, and object classification. These are separated into two groups: single-stage and two-stage methods. This study explores the significant uses of two-stage object detectors, such as RCNN, FastRCNN, and Faster-RCNN. We thoroughly examine the architectural improvements, underlying pretrained CNN architectures, and loss functions of single-stage object detectors, in particular YOLOs. This study provides in-depth explanations of the fundamental concepts while carefully examining various elements and optimizations applied in subsequent iterations of YOLOs. We also describe the difficulties and reasons behind the particular assessment of one-stage object detectors. Notably, YOLOs are an appealing option for object detection tasks since they beat their two-stage counterparts in terms of detection accuracy and inference time.

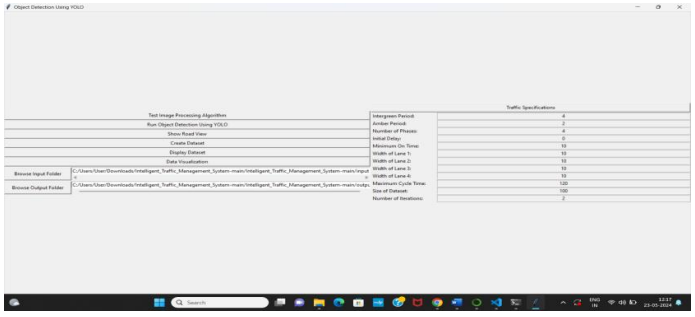

Fig 4-Project Dashboarde

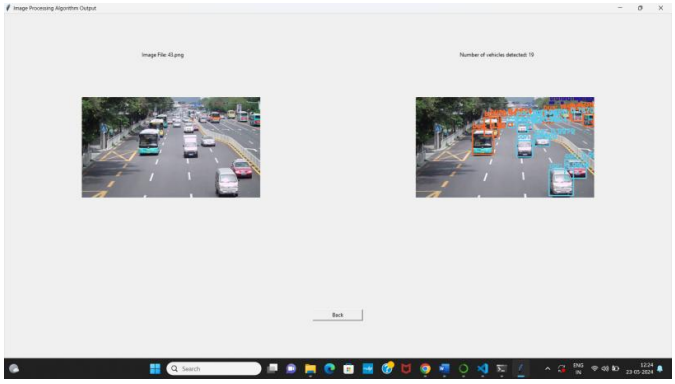

Fig 5-Traffic Image Processing Algorithm

Fig 6-Running Object Detection using YOLO

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

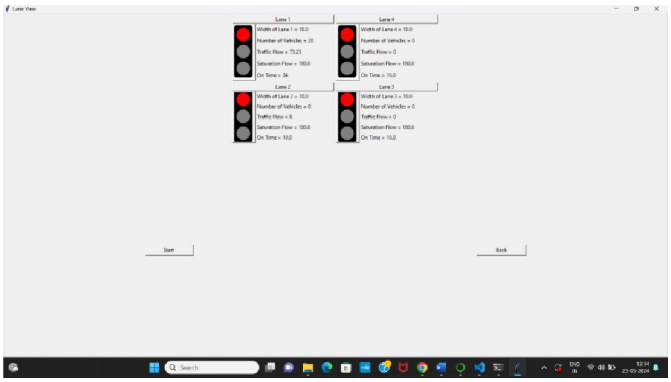

Fig 7-Signal Prediction as per Traffic in lane

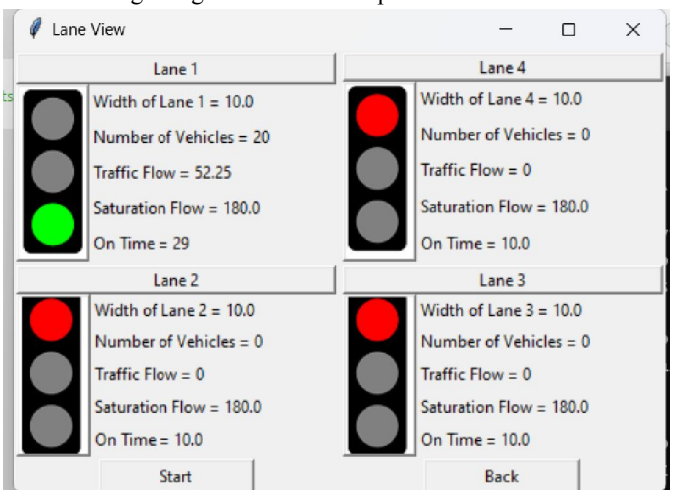

Fig 8-Signal Prediction as per Traffic in lane

Fig 9-Show Road View

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

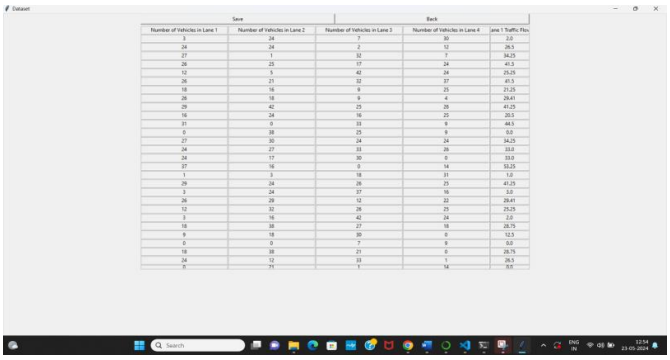

Fig 10-Create Dataset

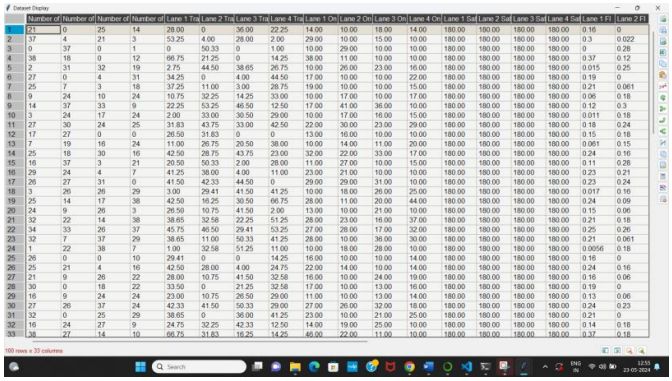

Fig 11-Display Dataset

Fig 12-Number of Vehicles in each lane

Fig 13-Traffic Flow in each lane

DOI: 10.48175/IJARSCT-18483

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

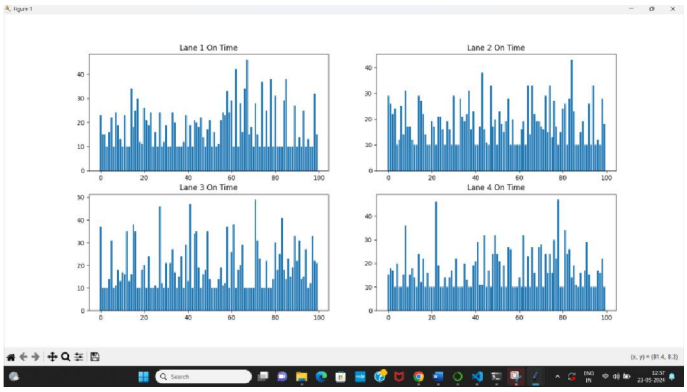

Fig 14-OnTime of Each Lane

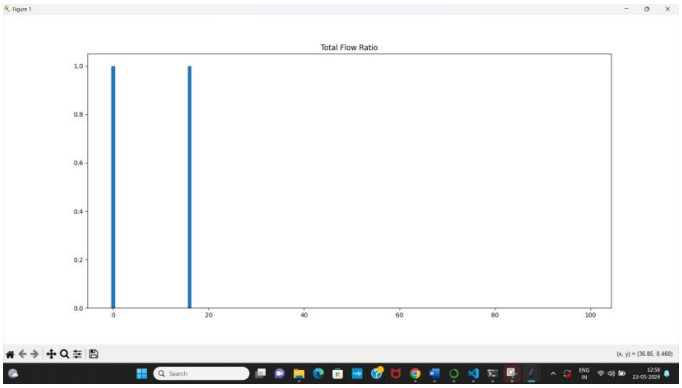

Fig 15-Total Traffic Flow Ratio

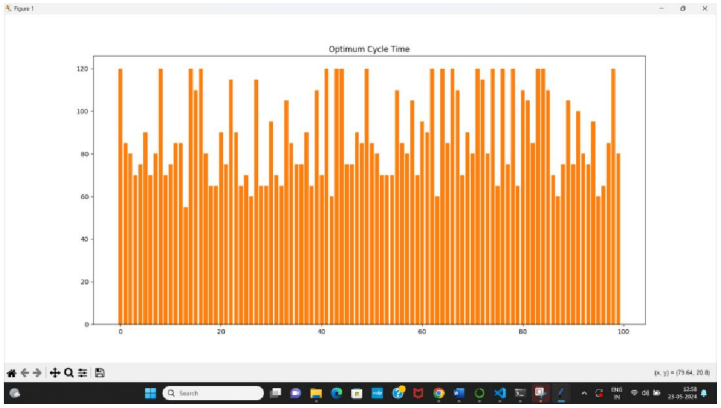

Fig 16-Optimum Cycle Time

REFERENCES

[1]. Ajantha Vijayakumar & Subramaniyaswamy Vairavasundaram (2024):YOLO-based Object Detection Models: A Review and its Applications

[2]. Marco Flores-Calero,César A. Astudillo ,Diego Guevara Jessica Maza,Bryan S. Lita ,Bryan Defaz ,Juan S. Ante ,David Zabala-Blanco andJosé María Armingol Moreno(2024):Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review

[3]. Mr. Mahesh Tiwari, Deepak Verma, Aryan Gupta(2024):DETECTION USING YOLO MODEL

[4]. Shubham Kamble, Vishal Ghaytadak, Aditya Nadgouda,Aditya Chavan, Prof. Wagh. D.B(2024):TIME OBJECT DETECTION USING YOLO TECHNOLOGY

[5]. Yan Zhou(2024):A YOLO-NL object detector for real-time detection

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 5, May 2024

[6]. U. Sirisha, S. Phani Praveen, Parvathaneni Naga Srinivasu, Paolo Barsocchi & Akash Kumar Bhoi (2023):Statistical Analysis of Design Aspects of Various YOLO-Based Deep Learning Models for Object Detection

[7]. Ajantha Vijayakumar & Subramaniyaswamy Vairavasundaram (2023)YOLO-based Object Detection Models: A Review and its Applications.

[8]. D. Akshaya, Charanappradhosh & J. Manikandan (2023):Social Distance Monitoring Framework Using YOLO V5 Deep Architecture

[9]. Ritayu Nagpal, Sam Long, Shahid Jahagirdar, Weiwei Liu, Scott Fazackerley, Ramon Lawrence, Amritpal Singh(2023):An Application of Deep Learning for Sweet Cherry Phenotyping using YOLO Object Detection

[10]. Mishuk Majumder andChester Wilmot(2023):Automated Vehicle Counting from Pre-Recorded Video Using You Only Look Once (YOLO) Object Detection Model

[11]. Tausif Diwan, G. Anirudh & Jitendra V. Tembhurne (2022):Object detection using YOLO: challenges, architectural successors, datasets and applications

[12]. Viswanatha V, Chandana R K, Ramachandra A.C.(2022)Real Time Object Detection