

AN OPTIMIZED REAL-TIME VEHICLE DETECTION FRAMEWORK FOR VIDEO STREAMS USING YOLOv4

GOLLA. KHYATHI¹, PROF. G SREENIVASULU²

¹M. Tech Student, ²Professor,

Department of ECE, Sri Venkateswara University College of Engineering,

Tirupati, A.P., India.

Abstract

With the number of vehicles growing at an accelerated rate, efficient traffic monitoring and vehicle detection systems are needed. The current methods of traffic monitoring are largely manual, thus requiring lots of time and manpower. In this paper, a YOLOv4 object detection algorithm is used to develop an optimized real-time vehicle detection framework for video streams. The proposed system receives image and video inputs, and outputs vehicle detection results in the form of a bounding box, class label and confidence score. The system offers individual modules for detection of images and live stream vehicles. The object detection is performed by YOLOv4 for fast

and efficient detection. The developed framework can be adopted for traffic surveillance, vehicle monitoring and intelligent transportation applications. The experimental results show the vehicle

detection capabilities of the system from traffic images and live video streams.

keywords: Vehicle Detection, YOLOv4, Deep Learning, Object Detection, Computer Vision, Video Surveillance, Real-Time Detection.

I. INTRODUCTION

The growing population of vehicles on the roads has presented a number of road traffic management, road safety and transportation monitoring challenges. One key demand for an intelligent traffic control system is efficient vehicle detection. Traditional traffic monitoring techniques require constant human supervision, and can be challenging if there are many vehicles present.

Automatic vehicle detection can be effectively achieved using computer vision and deep learning techniques. These techniques can process images and video streams and recognize various objects

without relying on constant manual supervision. Using bounding boxes, an object detection algorithm can detect an object and determine the location of the object in an image.

You Only Look Once (YOLO) is a popular deep-learning-based object detection algorithm. Unlike the standard detection methods, YOLO only performs one-stage object detection which is suitable for the real time application. YOLOv4 achieves a good result on accuracy and speed in detection.

II. LITERATURE SURVEY:

Wang et al. (2023): The authors presented a new vehicle target detection method for real-time identification in adverse weather conditions, which is based on YOLOv4. The study was directed towards the decrease in detection accuracy and speed due to the adverse weather conditions. Vehicle detection in difficult situations was improved by making improvements to the YOLOv4 framework. The research showed that the improved YOLOv4 model is applicable to the real-time vehicle detection. But challenging weather and complex traffic conditions still pose an obstacle for the detection to be reliable. [1]

Kumeechai (2023): The author explored a real-time vehicle detection system in highway environments, and tested the performance of the YOLOv4 model

against Faster R-CNN and SSD. This study mainly aimed at the detection and classification of various types of vehicles in traffic video. The study emphasized the need for real time processing in traffic surveillance and/or intelligent transportation applications. The study demonstrated the ability of the detection system based on YOLO to be applied in the practical case of highway monitoring; however, the accuracy of the detection depends on the traffic situation and the quality of the input video. [2]

In this work, Mishra and Yadav (2023) explored the vehicle detection in high resolution traffic surveillance data using YOLOv5. The problem was one of identifying and tracking the vehicles in dense traffic. Various road objects like cars, trucks, trams and vans were studied, and the KITTI dataset was utilized for training and evaluation. The results showed a potential of YOLO-based techniques in traffic surveillance, and scenes with high number of vehicles are still difficult to process due to occlusion and overlapping. [3]

Bar et al. (2023): The authors introduced a real-time framework for vehicle counting and velocity estimation for deep learning. YOLOv4 has been integrated with Deep SORT to track and detect vehicles in traffic videos. Region-of-interest calibration

using road markings was used to estimate vehicle movement and speed. The study showed that the fusion of YOLOv4 and tracking could help to realize real-time traffic analysis. The authors also reported that lower resolution videos can also decrease the vehicle detection and tracking performance. [4]

Liu et al. (2024): They introduced a real-time vehicle detection network, YOLOv8-FDD, to tackle the issues of missed detection, false detection, and deployment challenges within traffic scenarios. A Feature Sharing Detection Head was added to suppress redundant parameters and keep the same detection performance. The research was on optimization of the efficient and accurate detection of vehicles in traffic environments. [5]

Yu et al. (2024) presented an improved YOLOv7 model for real-time traffic surveillance. The method proposed a new model called Convolutional Block Attention Module (CBAM) to extract the key spatial and channel features. The study explored the differences between the sizes of vehicles and the changing traffic environment. The idea was to enhance adaptability of vehicle detection systems for large scale traffic monitoring. [6]

Li et al., (2024) Proposed the YOLOv8-DSAF for real-time object detection in smart-city traffic environment. The

research was focused on improving the accuracy of detection, real-time performance and adaptability of intelligent transportation applications. The modifications proposed were aimed to enhance the model's capability of detecting traffic objects in complex situations. [7]

Cui et al. (2025) suggested the RTL-Net, a lightweight real-time urban traffic object detection algorithm based on YOLOv8s. This study dealt with the problems of high computational complexity, limited real-time performance and detection accuracy in urban traffic scenes. The lightweight architecture was developed to enhance the applicability of YOLO-based detection for real-time traffic applications. [8]

The authors Hunain et al. (2025) proposed a vehicle detection, tracking and counting system utilizing YOLO with a Kalman Filter. The system was built to identify a variety of vehicles, monitor vehicle movement from frame to frame, and count vehicles in the presence of a detection line. Stable tracking was achieved during temporary detection interruptions or during occlusions with the use of the Kalman Filter. The results prove the effectiveness of the proposed detection and tracking system for real-time traffic video analysis. [9]

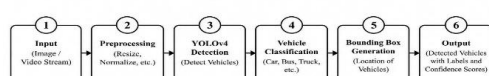
III. PROPOSED METHODOLOGY

The proposed methodology consists of several stages, including input acquisition, preprocessing, object detection, vehicle classification, and output visualization. The system accepts either an image or a live video stream as input.

First, the input image or video is provided to the system. In the case of video input, the video is processed frame by frame. The input data is prepared in a suitable format before being passed to the YOLOv4 detection model.

The YOLOv4 model analyzes each frame and identifies the objects present in the scene. When a vehicle is detected, the model generates a bounding box around the vehicle and provides its corresponding class label and confidence score. The detected results are then displayed through the developed user interface.

The overall process can be represented



as:

Fig.1.Vehicle Detection Pipeline

IV. SYSTEM ARCHITECTURE

The proposed system architecture as shown in the figure is composed of several stages which collaborate in order to detect objects in traffic images. The main goal of this architecture is to detect important

areas of a traffic scene and conduct detailed detection on those important areas. The architecture is a combination of initial image processing, region selection, zoom-in processing and fine detection, which can obtain accurate detection results. The initial step in the process is the Input, where a traffic image with vehicles and various other objects is supplied to the system. There can be several different objects of various sizes and locations in the input picture. A preliminary analysis is performed first and detailed detection is then carried out due to the fact that some objects may be small or hard to be detected directly from the whole image. The first step is to pass the input image through an R-Net. The input image is initially processed by R-Net and useful information about the entire traffic scene is extracted. Assists in identifying the salient characteristics and the likely positions of objects within the image. This is then passed on to the State Representation step. The information acquired from an input image in an appropriate format for further processing is called the State Representation stage. This representation enables the system to identify where to pay more attention in the picture. With this information, the Zoom-in Region Choose stage chooses the important part of the original traffic image. The chosen regions are labelled Region 1, Region 2, Region 3,

etc. Such areas can have vehicles or other objects that require detailed detection. The system only processes the selected regions repeatedly, instead of the entire image. This zoom in process will enhance the visibility and detectability of objects, particularly when objects are small or remote in the original image. Once the desired regions are selected, each region is fed to the Fine Detector. The Fine Detector carries out fine object detection on the selected regions. It detects the objects, and draws bounding boxes around the detected objects. The figure shows different examples of regions being processed by the Fine Detector. The locations of the detected objects are marked by bounding boxes. There is also an architecture element, namely an Update/Refine which allows to modify the architecture. The aim of this stage is to further enhance the selected regions with information gathered from the processing stage. If a region needs to be processed again, the system can update or refine the choice of regions to be processed and then process it again. This assists in making the overall process of detection better. The last step is to combine the results from the Fine Detector to give the final detection output. The output contains a list of objects detected and their respective bounding boxes. So

the whole process can be represented as:

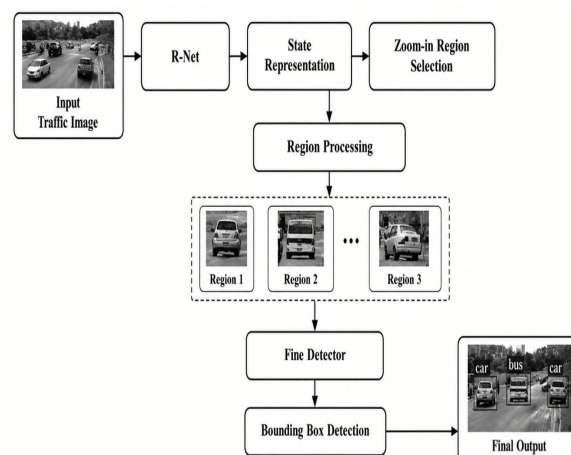


Fig.2.Overall Vehicle Detection Workflow

In general the system architecture is coarse to fine, as the whole picture of the traffic is first analyzed and the important areas are then analyzed in greater detail. This can be helpful in traffic surveillance and detection systems, where the system can use the computational resources it has for salient areas of the image.

V.DATASETAND PREPROCESSING

The vehicle detection system is proposed to be based on a pretrained vehicle object detection model, YOLOv4, to detect

vehicles from image streams and image. The YOLOv4 model is pre-trained on COCO (Common Objects in Context), a large scale object detection dataset that includes a large number of object categories in real world scenarios. Some of the object classes in the data set are related to traffic, like cars, buses, trucks, and motorcycles. These vehicle classes are considered important for the proposed traffic monitoring and vehicle detection application.

Using such a pretrained model lessens the need to train a deep learning model from scratch. The learned visual features already exist in the pretrained model YOLOv4 and allow it to recognize various categories of objects. The proposed system loads the pretrained weights with the YOLOv4 configuration file and class-name file. The structure and parameters of the YOLOv4 network are specified in the configuration file, and the weights file stores the learned parameters of the pretrained network. To connect the detected object identifiers with the labels of the objects, the class-name file is used.

The proposed system can be fed with a static image or a video stream. The video to be detected is first recorded frame by frame into individual frames using OpenCV library for video based detection. Each frame is considered as an individual

image and is processed separately by the detection system. This frame-by-frame processing enables the system to identify vehicles throughout the entire video footage.

For getting a proper input format, pre-processing is applied to frames before giving as input to the YOLOv4 network. The extracted frames are resized based on the frame size of the input model required by the YOLOv4 model. Resizing is used to ensure that the input size remains constant and the network can process frames efficiently. The image data is also transformed to the right format needed for the neural network. The pre-processing operations are useful to preserve the consistency in consecutive frames and for efficient object detection.

Each frame passes through the pre-processing network and then to the detection network, YOLOv4. The network looks at the visual characteristics of the frame and forecasts what is likely to be in the scene. The model outputs a confidence score for each detected object, a class label identifying the category of each detected object, and a bounding box that specifies the location of each detected object. Only detections with a confidence level that meets the specified criterion are outputted.

Non-Maximum Suppression (NMS) is used to minimize the number of detections of the same object. NMS will delete all duplicate detections and the most suitable bounding box. This helps to clarify and enhance the final detection results.

Lastly, detected vehicles are shown on the original video frames with bounding box, class label and confidence level. The resulting frames are continually displayed to provide real-time vehicle detection. The proposed pre-processing pipeline thus offers a uniform way of pre-processing the input data, and allows for the use of a pretrained YOLOv4 model for fast detection of vehicles in both live camera feeds and prerecorded video stream.

VI. YOLOv4 MODEL

YOLOv4 (You Only Look Once version 4) is a single-stage deep learning-based object detection algorithm for efficient and precise object detection. The major benefit of YOLOv4 is that it is able to detect multiple objects in an image or video frame because it has a single neural network that can be used for both object localization and classification. Due to its high detection speed and good accuracy, YOLOv4 is suitable for real-time applications such as traffic monitoring and vehicle detection.

YOLOv4 architecture consists of a backbone, neck and detection head. The backbone is responsible for extracting important features from the input image. YOLOv4 adopts the backbone network CSPDarknet53 to extract meaningful features in various regions of the input image. These extracted features will then be fed to the neck portion of the network where more features will be extracted and multi-scale features will be represented.

YOLOv4 introduces several modifications, including Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PANet), to enhance feature extraction and information propagation between layers of the networks. The model can find objects of various sizes with multi-scale feature processing. This is very helpful in traffic scenes where there can be a variety of The predictions are made using the extracted features in the detection head. YOLOv4 predicts a confidence score, object class, and a bounding box for each detected object. The area of the detected object is represented by the bounding box and the name of the class corresponds to the type of object. The confidence score is the chance of being correct when predicting whether the detection is correct distances and sizes of vehicles in one shot. YOLOv4 also employs cutting-edge methods like Mosaic data augmentation, Complete IoU

(CIoU) loss, and other optimizations for better detection accuracy. These techniques enable the model to acquire important visual properties and enhance its classification performance in other conditions.

One other key advantage of YOLOv4 is that it can make detection at several scales. This enables the model to detect large and small objects in the same image. For traffic-related applications, this feature can be helpful in identifying vehicles that are near the camera or further away. Non-Maximum Suppression (NMS) is used to eliminate duplication in the detected bounding boxes after object detection. If multiple bounding boxes are found for the same object, the one with the highest prediction score is kept and overlapping predictions are removed using NMS. This results in clear and reliable detection results.

The proposed system utilizes the YOLOv4 object detection model to identify vehicles in traffic videos. The model processes each video frame and determines if there are objects of a vehicle (from a vehicle type) in the scene. Vehicles detected are drawn as rectangles with labels and scores. Running the video frames in real time allows the model to perform real-time vehicle detection.

In this way, the YOLOv4 model has a good detection speed, feature extraction ability, and multi-scale object detection, which can be applied to the proposed real-time vehicle detection system.

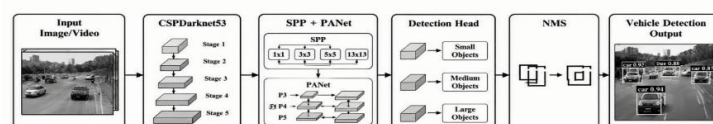


Fig. 3, YOLOv4 Model Architecture for Vehicle Detection

VII. RESULTS AND OUTPUT

The developed vehicle detection framework was tested using image and video inputs. The system provides a web-based interface through which the user can access the home page, upload an image, and perform live-stream detection.

For image detection, the user uploads an input image through the image upload page. The YOLOv4 model processes the image and identifies the vehicles present in the scene. The result is displayed with bounding boxes, vehicle labels, and confidence scores.

The prediction results demonstrate that multiple vehicles can be detected within a traffic scene. The system can identify vehicles at different positions and display their corresponding detection information.

The live-stream module processes video frames continuously. Vehicles detected in the live video are highlighted using bounding boxes and confidence scores. This demonstrates the capability of the proposed framework to perform vehicle detection from real-time video streams.

The developed system therefore provides both static-image-based and live-stream-based vehicle detection functionality

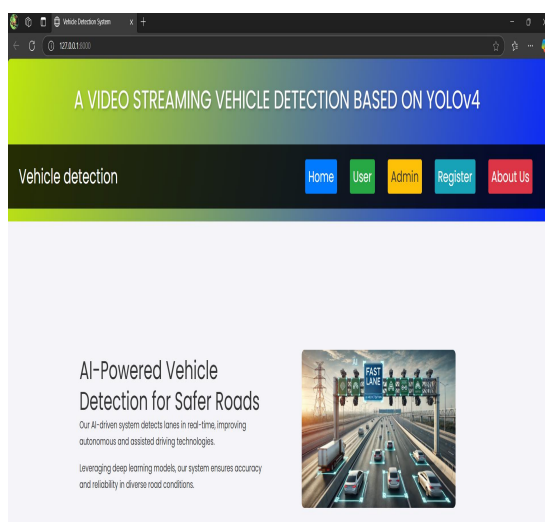


Fig. 4. Main Page

Welcome to the Vehicle Detection System that is powered by YOLOv4. Real time vehicle detection is accomplished using video streaming and image analysis on this platform.

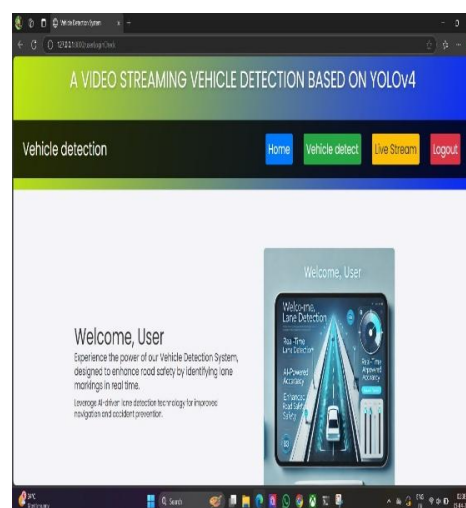


Fig.5.UserHomePage

The dashboard to manage detections. View video streams, upload images and view detection results, all in one place. Keep abreast of real-time information.

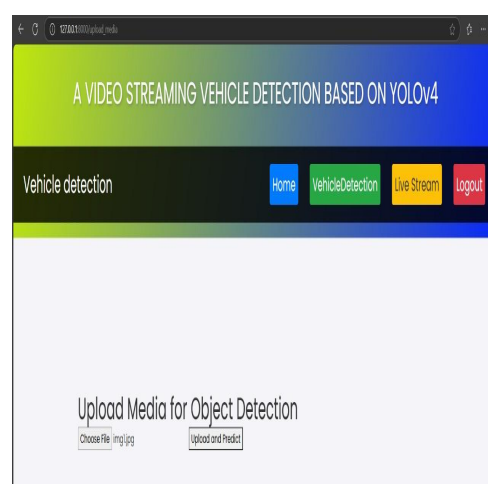


Fig. 6. Image Upload Page

Upload an image to recognize and classify vehicles with our algorithm

that is based on YOLOv4. Result will display the vehicles detected with bounding box and labels.

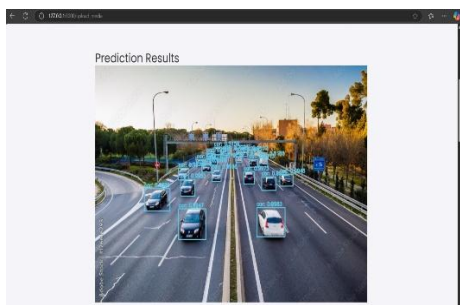


Fig. 7. Prediction Results with Detected Vehicles

See detailed detection results with identified vehicle types, count and confidence scores. Analyse image outputs (static or streamed).

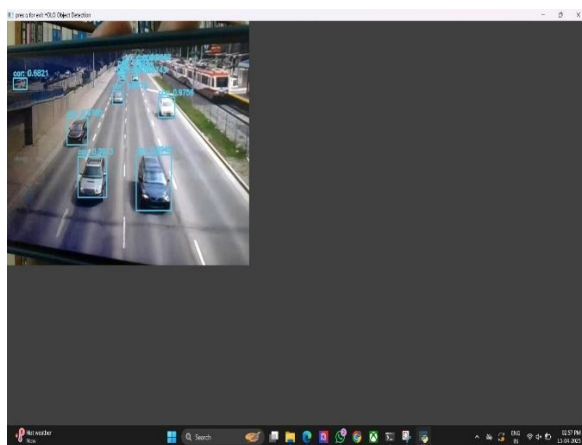


Fig. 8. Live Stream Vehicle Detection Output

Process live streamed video data to detect vehicles. See vehicles in real-time, straight through your video or camera feed.

YOLOv4 is able to process frames in real-time to detect and track vehicles efficient

VIII. CONCLUSION

A real-time vehicle detection framework optimized with the deep learning based YOLOv4 object detection algorithm was presented in this paper. The proposed system will detect vehicles from both static images and prerecorded videos as well as live video stream. Using a pretrained YOLOv4 model helps the system classify the commonly used vehicle types without starting to train the model again.

The proposed framework processes the input frames and identifies vehicles by drawing bounding boxes, labeling vehicle types and calculating vehicle confidence scores. The frames are processed continuously for video input, to generate results of vehicle detection in real-time. The system also offers independent image-based and live-stream detection, enabling the framework to be applied to various traffic monitoring needs.

The developed system is applied to vehicle detection in traffic scenes as an example of the practical application of YOLOv4. The output frames contain information about the location and class of detected vehicles which can be clearly visualized. The framework can then be applied to different applications like traffic surveillance,

vehicle monitoring, road safety, and intelligent transportation systems. Proposed system is efficient in real-time vehicle detection, but the detection performance is not guaranteed in the case of low light, high traffic density, vehicle occlusions, motion blur, complex background and bad weather. They can impact vehicle visibility and appearance and can decrease detection reliability.

In the future, there are opportunities to improve the accuracy of detection and processing speed for challenging environmental conditions. Vehicle tracking, vehicle counting, speed estimation, real-time traffic analysis can also be added to the system to be extended. The system can be deployed to edge devices and other resource-constrained devices to enhance system usage and utility in the field. In general, the proposed YOLOv4 based framework serves as a good basis for fast vehicle detection system and can be further improved for other advanced intelligent transportation and traffic surveillance applications

IX. REFERENCES

- [1] R. Wang et al., "Real-time vehicle target detection in inclement weather conditions based on YOLOv4," *Frontiers in Neurorobotics*, vol. 17, 2023.
- [2] P. Kumechai, "Real-time vehicle detection system on the highway," *Journal of Applied Research on Science and Technology*, 2023.
- [3] S. Mishra and D. K. Yadav, "Vehicle Detection in High Density Traffic Surveillance Data using YOLO.v5," 2023/2024.
- [4] "A framework for real-time vehicle counting and velocity estimation using deep learning," *Sustainable Computing: Informatics and Systems*, vol. 40, 2023.
- [5] X. Liu et al., "YOLOv8-FDD: A Real-Time Vehicle Detection Method Based on Improved YOLOv8," *IEEE Access*, vol. 12, 2024.
- [6] D. Yu et al., "Real-Time Monitoring Method for Traffic Surveillance Scenarios Based on Enhanced YOLOv7," *Applied Sciences*, vol. 14, 2024.
- [7] Y. Li, Y. Huang, and Q. Tao, "Improving real-time object detection in Internet-of-Things smart city traffic with YOLOv8-DSAF," *Scientific Reports*, vol. 14, 2024.
- [8] Z. Cui et al., "RTL-Net: Real-time lightweight Urban traffic object detection algorithm," *Complex & Intelligent Systems*, vol. 11, 2025.
- [9] A. Hunain, I. Indrabayu, and A. A. Ilham, "Vehicle Detection and Tracking

Techniques Using Yolo Detection and
Kalman Filter Optimization,” 2025.