VEHICLE DETECTION FROM VIDEO SEQUENCE USING DEEP LEARNING TECHNIQUE**.**

Rohini Chavan

[rohini.chavan@viit.ac.in](mailto:rohini.chavan@viit.ac.in)

Omkar Malwadkar, Omkar Kamathe, Shubham Krishnan

# Abstract

In the field of computer vision technology, deep learning-based classification and detection algorithms have proven to be a valuable tool for vehicle detection. However, the limited availability of high-quality labeled training samples makes it difficult for single vehicle detection methods to achieve satisfactory accuracy in detecting road vehicles. To address this issue, this experiment utilizes the YOLO-v5 architecture to detect and classify vehicles in publicly available datasets or pictures captured by CCTV. Transfer learning is employed by fine-tuning the weights of the pre-trained YOLO-v5 architecture using an extensive dataset of images and videos of congested traffic patterns collected by the researchers. This improves the YOLO-v5 structure's ability to detect difficult traffic patterns. The YOLO-v5 algorithm, which is well-known for its real-time problem-solving capabilities, is among the most popular object detection techniques. However, its performance heavily relies on the quality of the training dataset. The results of this experimental study indicate that YOLO-v5 can achieve a maximum accuracy of 88% when trained using only 31 images.

**Keywords:-**CCTV, YOLO(You Only Look Once), Deep Learning, Object Detection.

# INTRODUCTION

Detecting vehicles ahead and monitoring traffic conditions while driving are essential for safe driving, preventing accidents, and facilitating autonomous driving and tracking . Real-time perception is particularly crucial for the development of self-driving cars, and image processing can be employed for this purpose. Image processing, which is a widely discussed topic in photogrammetry, is a technique used to extract useful information from original images. With the help of advancing technology, various methods can be developed to extract useful information from original images, including object detection, identification, and classification. In the fields of computer vision and photogrammetry, detecting and classifying objects in image content have been one of the primary research topics.1]

AlexNet, proposed by Krizhevsky in 2012, outperformed all other competitors in the ImageNet Large Scale Visual Recognition Challenge that year. This marked the beginning of the development of deep learning for visual applications, leading to the proposal of numerous neural network architectures such as Vgg , ResNet . Deep learning has significantly improved the state-of-the-art performance in many fields, including natural language processing, computer vision, and recommender systems. In computer vision, images or videos are processed with fine-tuned algorithms to produce useful information for humans. Object detection is a crucial task in computer vision, and it can leverage the power of convolutional neural networks. In this experiment, several neural network architectures were utilized for vehicle detection.[2]

The Human Vision System (HVS) is capable of complex tasks, such as detecting, recognizing, and identifying diverse objects with minimal conscious attention. Recent advancements in Computer Vision (CV) and Machine Learning (ML), combined with the availability of massive datasets, faster GPUs, and better algorithms, have enabled computers to perform similar tasks with high accuracy. The goal of vehicle detection and classification is to accurately locate vehicles in images or videos. Efficiency of vehicle localization is a critical step in traffic monitoring or surveillance.

Figure 1 shows several detected vehicles from Pakistani traffic images that are achieved

using the machine learning algorithms. Therefore, autonomous vehicle detection methods

must exactly detect traffic objects, such as cars, vehicles, or police vans or bikes in real-time

to gain good control and make right decisions for the public safety.[3]

The YOLO method is a novel approach for detecting a wide range of vehicles in a single step. This method addresses the vehicle perception problem as a regression issue by using a convolutional neural network (CNN) to classify the image, enabling robust vehicle detection. The YOLO model can determine the object's position, category, and confidence score while enhancing detection speed and detecting motion-blurred vehicles in real-time. The regression-based YOLO technique is one of the most recent approaches that predicts bounding boxes and class probabilities directly within a single neural network. The YOLO model was designed to speed up the object detection and localization process in an image by using CNN to instantly identify multiple objects. To accommodate vehicles of various sizes and shapes, it integrates predictions from numerous feature maps with different resolutions. With improvements in YOLO-based methods, such as YOLOv3 and YOLOv5, the YOLO approach has continued to provide higher performance in terms of processing time and accuracy.[4]

Moreover, the CNN, which is a form of deep neural network, is extensively used for image recognition and categorization. These are the algorithms that can identify various objects, such as license plates, cars, people, and a verity of others objects. A primary benefit of the CNN is that it extracts essential features without any human interaction after the training process. Different versions of the CNN, such as R-CNN, Fast-RCNN, and Faster-RCNN are the most popular and commonly utilized CNN approaches. However, the computational load is still to high for devices with limited computing power and space to process photos. The D-based algorithms have been frequently used in recent approaches among the many vehicle detection algorithms and are divided into region-based and regression-based methods.[5]

Moving object recognition involves two stages: object detection and object recognition. The objective of object recognition is to classify an object into predefined classes based on various features, while object detection is the initial stage that provides information about the background and foreground objects. However, to verify and validate the object, feature extraction-based object recognition is required. While video data was previously too compressed for video surveillance, we can now convert the video to images to detect the specified object.

This study combines vehicle detection with deep learning approaches. The vehicle detector is trained using the YOLO deep learning method on sample vehicle datasets and is successfully tested on the test data, yielding efficient results for the vehicle detection problem. This section will outline the proposed method in detail, and the study will conclude with a summary of the findings

# 2.LITERATURE SURVEY

**Comparing Model Performance for Real-Time Vehicle Detection in UAV Images"**

After comparing both models, it was determined that they both provided reliable estimates and accurate results, with one model in particular demonstrating high performance on terrestrial videos. For real-time vehicle detection, the YOLO algorithm's structure makes it a suitable choice. When applied to a targeted dataset, it can successfully identify vehicles in UAV images. To improve the accuracy, it's important to use appropriate training and verification procedures on the dataset that's being used as input. By doing so, more successful detections can be achieved while also aligning with the goals of the study. With stronger training, success rates can be increased, instilling greater confidence in the efficacy of deep learning methods.[1]

**Enhancing Vehicle Detection with Deep Learning Techniques**

The focus of this thesis is on introducing and enhancing fundamental components of deep learning that are crucial in vehicle detection. The implementations and results presented showcase the impressive performance of deep learning techniques in accurately detecting vehicles. The proposed model not only enhances the performance of this task, but also reduces testing time. By integrating an extended RPN, the model is able to handle the significant variation in vehicle scales, thereby improving its robustness.[2]

**Improving Vehicle Detection in High-Resolution UAV Images with an Adaptive Clipping Algorithm**

This paper proposes a vehicle detection method that overcomes the limitations of traditional object detection algorithms when dealing with high-resolution images captured by UAVs, particularly with small objects. We adopt the YOLOv5 object detection algorithm as a baseline and introduce an adaptive clipping algorithm during data preprocessing and detection to enhance the algorithm's ability to detect small object vehicles. We evaluate the algorithm's performance using precision, recall, and mAP as evaluation indices, and conduct comparison experiments to verify its effectiveness. Our findings reveal that improving the resolution of UAV aerial images can lead to better detection results.

To further improve the efficiency and real-time performance of the algorithm during UAV operations, our future research will focus on enhancing the framework detection speed. Additionally, we plan to optimize the single-scale object detection process and the network model structure, possibly through techniques such as model pruning, backbone structure optimization, and reparameterization. These improvements will pave the way for wider application of UAVs in intelligent traffic management.[3]

**Exploring the Effectiveness of a CNN-Based Vehicle Detection Method in Capturing Traffic Parameters**

In chapter two, we conducted a review, analysis, and evaluation of various vehicle detection methods, taking into consideration their respective strengths and weaknesses. Building on this knowledge, we proposed a new implementation based on a Convolutional Neural Network (CNN) to study its effectiveness in capturing traffic parameters, especially under illumination variance. We utilized the YOLOv5s architecture for vehicle detection, optimized with k-means for anchor boxes. To evaluate the system's performance under different traffic conditions, we measured metrics such as accuracy, IoU, and recall. We compared our proposed method with previous research in the field and also with the baseline YOLOv5s model to further validate its effectiveness.[4]

**Accurate and Robust Vehicle Detection using YOLO-v5 Architecture with Transfer Learning**

Innovative YOLO-v5 based vehicle detection method for challenging scenarios This paper presents a new vehicle detection approach using the YOLO-v5 architecture with transfer learning. The method is tested on three public datasets, showing superior performance under challenging conditions such as night, rain, and snow. Future work includes detecting occluded and moving vehicles, exploring network structure changes, and integrating with deep learning methods for tracking and recognition.[5]

**Improving Infrared Vehicle Detection Accuracy Using Improved YOLOv5 Architecture**

Analysis of Infrared Vehicle Images with Improved YOLOv5 Network Using DenseBlock, Ghost Convolution, SE Module, and EIOU Modules

The article presents an analysis of infrared vehicle images with an improved YOLOv5 network using four modules: DenseBlock, Ghost Convolution, SE Module, and EIOU. The article evaluates the effectiveness of each module and their combinations in detecting small, occluded targets. The authors provide recommendations for adjusting the insertion-extraction module based on task requirements. Future work includes optimizing the network to increase confidence in detecting missed targets and addressing the challenge of extracting vehicle targets in complex interference environments.[6]

**Development of a Vehicle Detection and Tracking Method for Highway Surveillance Video Scenes using YOLOv3 Algorithm**

A Method for Vehicle Detection and Tracking in Highway Surveillance Videos"

This study proposes a method for detecting and tracking vehicles in highway surveillance videos using YOLOv3 algorithm and ORB feature extraction. The method extracts a more effective ROI area by defining the road surface area as a remote and proximal area, which is then used to detect the vehicles. The vehicle trajectories are analyzed to collect data, and abnormal parking events and traffic jam events can be detected. This method is low in cost and high in stability, making it a practical solution for highway surveillance. The presented methodology and results can also serve as important references for European transport studies.[7]

**Improving Object Detection with Deep Learning and Multi-Sensor Fusion**

Investigating Deep Learning for Moving Object Detection with Multi-Sensor Fusion

This paper explores the use of deep learning (CNN) object detection for improving the accuracy and robustness of moving object detection using multi-sensor fusion. The study provides promising results for object detection using only image sensor data and will be extended to include data from LIDAR/RADAR and ultrasonic distance sensors in the future. The research is being conducted at the Mechatronics Research Lab at IUPUI.[8]

# Methodology

We have used YOLO V5 version to detect the moving objects in frame or image or video. So here we started with What is YOLO? YOLO is term for ‘YOU ONLY LOOK ONCE'. YOLO V5 proposes using an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once. It differs from the approach taken by previous object detection algorithms, which repurposed classifiers to perform detection.

Following a fundamentally different approach to object detection, YOLO achieved state-of-the-art results, beating other real-time object detection algorithms by a large margin.

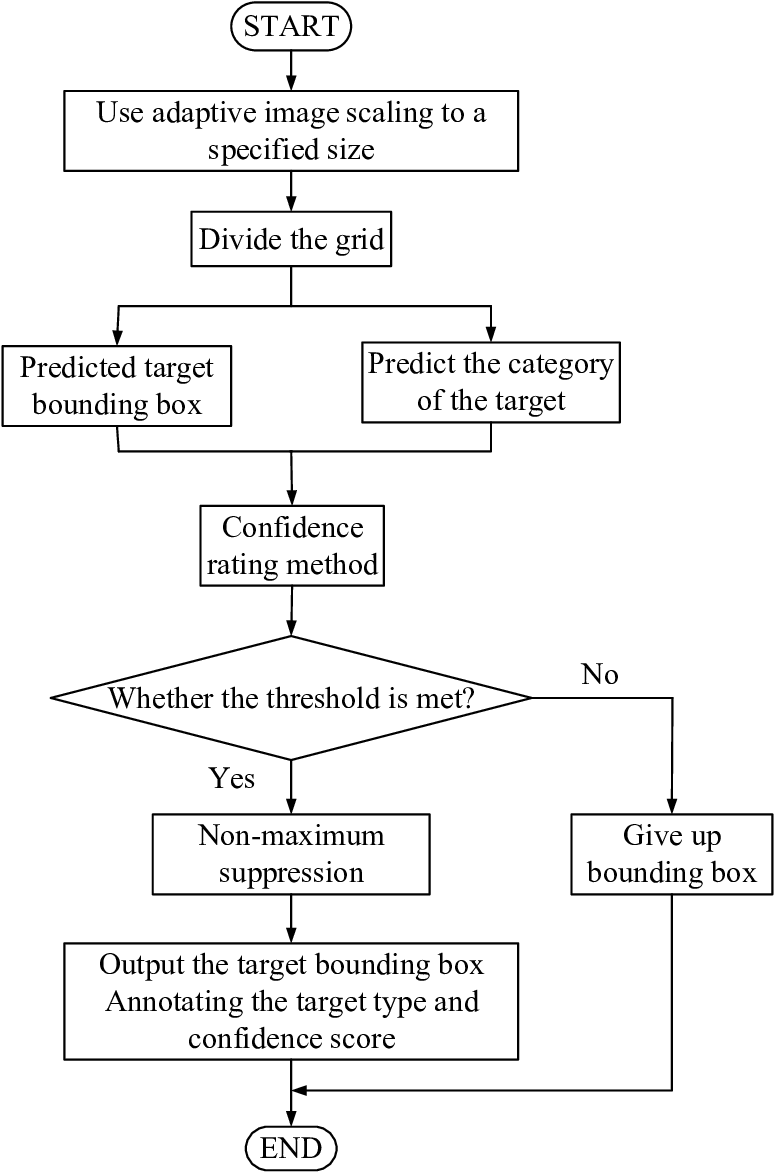
While algorithms like Faster RCNN work by detecting possible regions of interest using the Region Proposal Network and then performing recognition on those regions separately, YOLO performs all of its predictions with the help of a single fully connected layer.

Several new versions of the same model have been proposed since the initial release of YOLO in 2015, each building on and improving its predecessor. Here's a timeline showcasing YOLO's development in recent years.

YOLO v5 uses a new method for generating the anchor boxes, called "dynamic anchor boxes." It involves using a clustering algorithm to group the ground truth bounding boxes into clusters and then using the centroids of the clusters as the anchor boxes. This allows the anchor boxes to be more closely aligned with the detected objects' size and shape.

YOLO v5 also introduces the concept of "spatial pyramid pooling" (SPP), a type of pooling layer used to reduce the spatial resolution of the feature maps. SPP is used to improve the detection performance on small objects, as it allows the model to see the objects at multiple scales. YOLO v4 also uses SPP, but YOLO v5 includes several improvements to the SPP architecture that allow it to achieve better results.

YOLO v4 and YOLO v5 use a similar loss function to train the model. However, YOLO v5 introduces a new term called "CIoU loss," which is a variant of the IoU(intersection over union) loss function designed to improve the model's performance on imbalanced datasets.



# Figure 1. YOLOv5 target detection flow chart.

The given flow chart is explaining the working in the easiest and the simplest way.

CREATE CUSTOM IMAGE DATASET FOR TRAINING AND VALIDATION

CREATE IMAGE ANNOTATIONS

TRAIN YOLO V5 MODEL USING OUR CUSTOM DATASET

TEST MODEL BY GIVING DIFFERENT VIDEO INPUT

OBSERVE AND ANALYZE RESULTS

## Figure 2. steps involved in vehicle detection

**3.1. Step 1: creating custom image dataset**

The initial objective was to acquire images of various types of vehicles, including ambulances, cars, trucks, buses, and motorcycles. To accomplish this, we utilized a dataset of vehicle images from Open Images. Following this, we established a directory entitled "train data," within which we generated two additional directories named "images" and "labels." Inside the "images" directory, we established two more subdirectories called "train" and "Val." The vehicle images were partitioned for the purposes of training and validation, with a total of 170 images designated for training and 56 images for validation. These images were saved in their respective folders, "train" and "Val."

**3.2.Step 2: Labelling images and creating classes**

Annotation plays a crucial role in constructing a customized object detection model. For our custom dataset, we employed a tool known as "makesense.ai" to generate annotations. Within this tool, we identified five categories: Ambulance, Bus, Car, Motorcycle, and Truck. We manually defined rectangular boxes for each image and then downloaded the corresponding label file, which was subsequently placed in the labels folder. Annotations assist in the training of our model using the image dataset

**3.3.Step 3 :Training of the yolo v5 model**

Step 3 involves the training of the yolo v5 model, which was executed using Google Colab's GPU runtime type. The subsequent step was to determine the appropriate batch size and number of epochs.

A) The batch size pertains to the number of samples that are processed simultaneously by the model during training. For instance, if the batch size is set to 32 and the model is training on a dataset of 1000 images, it would require 32 iterations (one for each image in the batch) to complete a single epoch. This implies that the model would have seen all 1000 images in the dataset before completing one epoch.

The selection of batch size is typically based on the available memory and computational resources. A larger batch size may result in faster training, but it may necessitate more memory and could not generalize as effectively as a model trained with a smaller batch size.

The use of mini-batches (i.e., a batch size smaller than the full dataset) is beneficial since it allows the model to update its weights more frequently, resulting in faster convergence. Additionally, it enables the model to learn from a diverse set of samples instead of seeing the same samples repeatedly in each epoch.

In our case, we utilized a batch size of 8 to train the yolo v5 model.

**Figure 3. Training Batch**



Epochs: An epoch refers to one pass through the entire dataset. Epochs are used to control the number of times the model sees the entire dataset during training. The number of epochs is a hyperparameter that you can tune to control the model's performance. Increasing the number of epochs can often improve the model's accuracy, but it can also increase the training time. Can lead our model to face problem of overfitting whereas too less epochs can lead to problem of underfitting. To train our model we have used 160 epochs.

# 4.RESULT AND ANALYESE

To evaluate object detection models like R-CNN and Yolo, the **mean average precision (mAP)** is used. The mAP compares the ground-truth bounding box to the detected box and returns a score. The higher the score, the more accurate the model is in its detections.

To measure mean average precision we have created **precision-recall curve**. And then measured mean average precision. The precision is calculated as the ratio between the number of Positive samples correctly classified to the total number of samples classified as Positive (either correctly or incorrectly). The precision measures the model's accuracy in classifying a sample as positive.

𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜n=

True positive

True positive + False positive

When the model makes many incorrect Positive classifications, or few correct Positive classifications, this increases the denominator and makes the precision small.

Recall: The recall is calculated as the ratio between the number of Positive samples correctly classified as Positive to the total number of *Positive* samples. The recall measures the model's ability to detect Positive samples. The higher the recall, the more positive samples detected.

𝑅𝑒𝑐𝑎𝑙𝑙 =

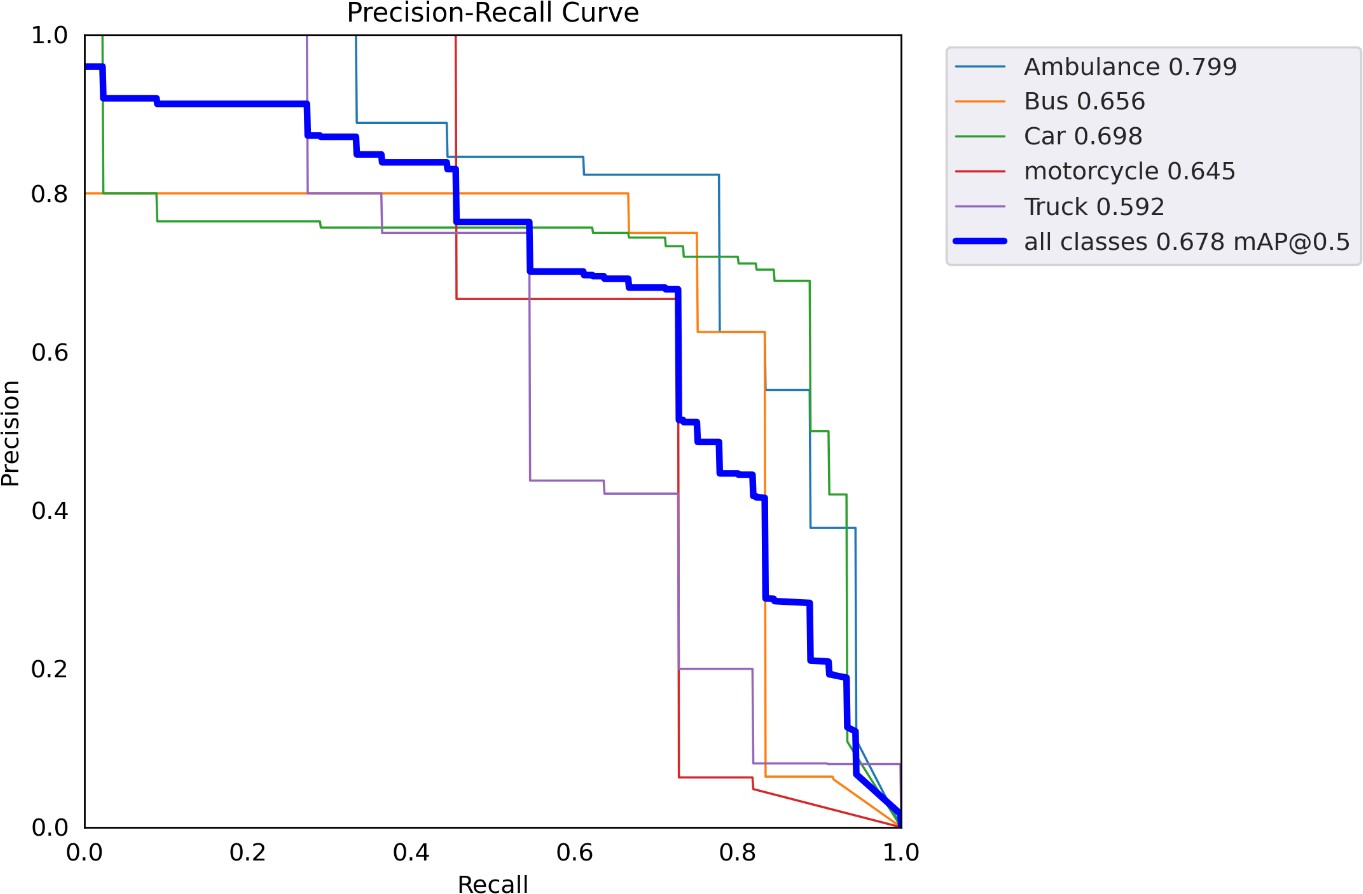
True positive

True positive + False negative

The recall cares only about how the positive samples are classified. This is independent of how the negative samples are classified, e.g., for the precision. When the model classifies all the positive samples as positive, then the recall will be 100% even if all the negative samples were incorrectly classified as positive.

Precision-recall curve: Due to the importance of both precision and recall, there is

a precision-recall curve the shows the tradeoff between the precision and recall values for different thresholds. This curve helps to select the best threshold to maximize both metrics.



**Figure 4. Precision Recall curve**

Mean average precision (mAP): Mean average precision (mAP) is calculated by calculating the Average Precision for each class. The mean of the Average Precision for all classes is the mAP. At 160 epochs we have achieved the mAP of 0.678

## Quantitative Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. of epochs | Box loss | Object loss | Class loss | mAP |
| 40 | 0.038314 | 0.021214 | 0.015635 | 0.49772 |
| 80 | 0.029473 | 0.020683 | 0.008619 | 0.59527 |
| 120 | 0.022811 | 0.016667 | 0.005495 | 0.60507 |
| 160 | 0.0199 | 0.016068 | 0.003863 | 0.67843 |

**Table 2. Model loss and precision at different number of epochs**

## Graphical Analysis

Chart Title

0.8

0.7

0.6

0.5

0.4

0.3

0.2

0.1

0

40

80

120

160

Box loss Object loss Class loss mAP

**Figure 5. Graphical representation of model accuracy**

As we can see in table 2 at 160 epochs our model has achieved highest mean average precision 0.678 with minimum box loss and object loss

So finally, we have tested our module by giving videos as input and successfully detected the vehicles in that video. We have modified the code as per our requirement. Then, first of all we have download the code of YOLO v5 from official website then we open code file in google colab because its needs more GPU than system has. After that first we moved all data files to the drive so if we made any changes, it will save to drive directly (if we don’t move all data files to the drive and if google colab run-time disconnected we may be losing our changes and saved data too). Hence, it is necessary to move the data files to drive. Further we evaluated performance of our model by giving input videos.

## Table 3. Details about input data set

|  |  |  |
| --- | --- | --- |
| Video  Name | Video | Frames |
| Video1 |  | 238 |
| Video2 |  | 313 |
| Video3 |  | 500 |
| Video4 |  | 688 |
| Video5 |  | 702 |

**Dataset:-**

## 4.2 Implementation on Various Dataset:-

In this paper we provide the input and output frames. Using the data of video frames we can better understand the between difference between input frame and output frame of each video.

|  |  |  |
| --- | --- | --- |
| Video  Name | Input Video Frame | Output Video Frame |
| Video1 |  |  |
| Video2 |  |  |
| Video3 |  |  |
| Video4 |  |  |
| Video5 |  |  |

**5.CONCLUSION**

In this paper we have investigated the problem of detecting the movement and standing of the person is with more intelligently. Our system is successfully recognized about the object’s moving and standing activities. This is an essential step for making the legacy data useful for data mining and machine learning. We have used a pretrained YOLO neural network and used data generated with our own simulator to retrain the network to detect the components we are interested in. The experimentation is done with this process using data set for performance of evaluation of algorithm. We have used accuracy matrix for evaluating the performance of our algorithm and we got maximum 85% accuracy for different videos. We have represented the accuracy in term of quantitative and graphical.

**6.Reference**

**1**. Vehicle Detection Using Different Deep Learning Algorithms from Image Sequence

Sumeyye CEPNI, Muhammed Enes ATIK, Zaide DURAN

Istanbul Technical University, Faculty of Civil Engineering, Department of Geomatics,

Istanbul, Turkey

2. Vehicle Detection in Deep Learning Yao Xiao

3. YOLOv5-Based Vehicle Detection Method for High-Resolution UAV Images Ziwen Chen , Lijie Cao ,and Qihua Wang

4.Retracted: Vehicle Detection for Vision-Based Intelligent Transportation Systems Using Convolutional Neural Network Algorithm Journal of Advanced Transportation

Received 15 November 2022; Accepted 15 November 2022; Published 29 November 2022

5. A Fast and Accurate Real-Time Vehicle Detection Method Using Deep Learning for Unconstrained Environments Annam Farid 1,\*, Farhan Hussain 1,\*, Khurram Khan 2

, Mohsin Shahzad 3 , Uzair Khan 3 and Zahid Mahmood 3,\*

6. Application of Improved YOLOv5 in Aerial Photographing Infrared Vehicle Detection Youchen Fan 1,†, Qianlong Qiu 1,†, Shunhu Hou 1, Yuhai Li 1 , Jiaxuan Xie 1 , Mingyu Qin 2 and Feihuang Chu 1,\*

7.Vision-based vehicle detection and counting system using deep learning in highway scenes Huansheng Song, Haoxiang Liang\* , Huaiyu Li, Zhe Dai and Xu Yun

8.Object Detection from a Vehicle using Deep Learning Network and Future Integration with Multi-Sensor Fusion Algorithm Raja Sekhar Rao Dheekonda

Sampad K. Panda§ MD. Nazmuzzaman Khan§ Mohammad Al-Hasanv Sohel Anwar§