IMPROVING THE EFFICIENCY OF IMAGE PROCESSING WITH DEEP LEARNING FOR VEHICLE DETECTION AND TRACKING

Abstract

Vehicle identification and tracking is an extremely important function of traffic surveillance systems that is necessary for efficient traffic management and the protection of drivers and passengers. Finding and following the path of vehicles is the primary goal of this research. The goal of this research is to develop methods for the automated identification of cars in digital photographs and moving pictures. One of the numerous uses for Deep Learning, which may include fuzzy logic, neural networks, and evolutionary algorithms, is in the detection and tracking of automobiles. The purpose of this project is to apply deep learning to the problem of vehicle recognition and tracking; the primary-stage target detection techniques will be YOLOv5 and Single Shot MultiBox Detector (SSD). This is the main topic of the article. The Single Shot MultiBox Detector (SSD) model architecture is then employed as the major foundation for vehicle detection. Focus loss, in addition to the standard SSD, is an optimization component that improves feature extraction speed. Therefore, the procedure begins with a series of training procedures on the photos included inside the publicly accessible road vehicle dataset. The vehicle recognition model is then trained using YOLOv5 and SSD algorithms; these two algorithms work together to show how effective they are at detecting vehicles. Comparing the models' detection rates on different cars is the key to locating it. The fundamental objective of this study is to develop an automated technique for detecting and tracking autos in both static and dynamic scenes. In the end,

Authors

Ankireddy Priyanka
Hindustan Institute of Technology and Science
Padur, Kelambaakkam
Chengalpattu, India.
priyasivakrishna99@gmail.com

Dr. V. Ceronmani Sharmila
Department of Information Technology
Hindustan Institute of Technology and Science
Padur, Kelambaakkam
Chengalpattu, India.
csharmila@hindustanuniv.ac.in

Dr.V. Lokeswara Reddy
Professor & HoD
Department of CSE
K.S.R.M. College of Engineering (Autonomous)
Krishnapuramu, Kadapa
Y.S.R, Andhra Pradesh, India.
vlreddy74@gmail.com

the trained network model is applied to the analysis of the vehicle camera video, and the detection performance is tested experimentally. The study's results show that the approach may enhance vehicle identification success to 97.65%. From video and picture inputs, it can reliably identify vehicles.

Keywords: Vehicle Detection, Image Processing, Vehicle Tracking, Deep Learning, Object Tracking.

I. INTRODUCTION

Both the vehicle information system and the intelligent traffic system make use of automatic vehicle data recognition. Academics have paid a lot of attention to it since the turn of the decade, thanks to developments in digital photography and processing power. The recognition of vehicles automatically is a cornerstone of many cutting-edge traffic management programs [1, 2]. Among them are automated vehicle accident detection, automatic traffic density estimates, lane departure warning systems, traffic signal controllers, and traffic response systems. The strain on those in charge of managing the population and its associated infrastructure grows with each passing year. The global population is expanding at a breathtaking pace. There was a subsequent increase in the manufacturing of automobiles and other mechanical equipment. But it's crucial to handle new problems like traffic, accidents, and other challenges with caution. In order for humanity to continue making progress toward their objectives, new discoveries and inventions have had to be developed and implemented. Congestion on main thoroughfares and in big cities is a prime example. Some of the solutions used to this issue include a traffic signal and a sign. These answers seem to be inadequate by themselves.

Decision-making-informing technologies like object identification and tracking are currently in development. The goal of this work is to facilitate the use of automated video surveillance solutions. These events have been used to solve many different sorts of problems. Object recognition and tracking are only two of the many subsystems that make up today's state-of-the-art Intelligent Transportation System (ITS). This technology is able to recognize a wide variety of vehicles, as well as lanes, traffic signals, and even individual models. It may also recognize distinct car manufacturers. Traffic and roads might benefit from vehicle identification and categorization, which could reduce the number of accidents and assist authorities maintain tabs on infractions. Vehicle recognition in both motion and still photographs is a natural ability for humans. Computer algorithms and programs rely heavily on the many forms of data. The lack or presence of certain components (such as favorable weather or appropriate illumination) might change the difficulty level. Meanwhile, there is a bewildering variety of automobile models and body designs available. Video objects might be of varying sizes and shapes, which presents a new difficulty when trying to recognize them in real time. Scientists have created a number of tools to help them find and follow moving things. In their numerous forms, these techniques make use of a wide range of algorithms, including fuzzy, neural network, evolutionary, Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN). Because of the corporate world's obsession with this system or Computer futurist, developments in this area occur often. In order to draw conclusions regarding the practical uses of these algorithms, this study compares and contrasts fuzzy algorithms, neural network algorithms, and evolutionary algorithms.

Increases in disposable money and standard of life have made private car ownership more accessible to the general public. As a result, more and more drivers approach their vehicles with fresh eyes and lofty goals. This has led to an increase in public unease over autonomous car technologies. More and more automobiles on the road have made travel easier and more difficult at the same time. It's becoming harder to get where you need to go as traffic rises, and this increased demand has piqued the interest of many in the idea of intelligent transportation. Recent years have seen explosive growth in computer vision thanks to the widespread adoption and use of deep learning. Autonomous cars, face recognition systems, and picture segmentation are just a few examples of the many real-world uses of computer vision technology [1-3]. Their ability to distinguish between vehicles is crucial to the success of autonomous cars and other kinds of intelligent transportation. As a first step toward completely autonomous and intelligent transportation systems, this research employs a deep neural network to recognize and track moving autos.

II. RELATED WORKS

Vehicle detection technology is analogous to target detection technology. Both vehicle detection and target detection aim to accomplish similar core goals, which may be broken down into the locations and types of targets. Historical data-based algorithms, deep learning-based algorithms, YOLO-based algorithms, and path-following algorithms are the four primary categories of vehicle identification algorithms.

1. **Traditional Vehicle Detection Algorithm:** Manual design of the target's attributes is essential to both conventional vehicle detection methods and conventional target detection approaches. The knowledge-based recognition algorithm is one such method; it is able to identify a vehicle by its distinct outline, as well as by its lines, shadows, and other edge elements. Therefore, we can determine the car's location in the picture by comparing the grayscale values of the shadow below it to those of the surrounding pixels that make up the vehicle body. That way, we can see vehicles and accomplish our mission. The algorithm then makes a final determination as to whether or not the automobile has been located. Despite its detectability, this method will be very light dependent. This is because the quality of the lighting will greatly affect the tonal range of its picture. It's important to remember that any object of a similar size and form will produce a similar shadow to that of the automobile. This imprecision renders the approach unsuitable for uses that need high precision. In addition to detecting brake lights, artificially constructed vehicle elements may aid in car identification. If you want to identify an automobile, you might try keeping an eye on its brake lights. However, the detection impact you'd get from doing so would be too little because of the substantial limits given by the influence of light. While the previous knowledge-based detection method accomplished its detection job by relying on the vehicle's inherent qualities, it struggled to satisfy the task's criteria owing to its limits and a poor detection recognition rate.

   Conventional techniques of vehicle detection also make use of a vehicle identification system predicated on a very simple application of machine learning. This system can identify vehicles based on their individual characteristics thanks to the integration of a vehicle-centric algorithm and a machine learning algorithm [4-6]. While the use of SVM and HOG features into shallow machine learning has the potential to improve detection accuracy, there are several downsides that must be considered. Even while this technique boosts detection accuracy by simple cascade, it also uses a larger model for detection, which requires more calculations overall. Even when using a shallow machine learning technique, feature selection is necessary to get the highest possible accuracy in vehicle recognition. The substantial modeling work required to simulate complex and ever-changing traffic circumstances is a primary cause for the delay in its development. Traditional techniques for identifying automobiles include frame difference approaches, streamer methods, and background modelling methods [7, 8]. These methods are among the most popular in use today. There are a number of other ways to detect vehicles. There are a variety of factors, such as lighting and weather, that might affect the accuracy of the identification results produced by such an algorithm [9, 10]. This makes it difficult to adapt to the ever-changing reality of traffic on the roadways and hampers the demands of real-time vehicle identification. It also makes it harder to avoid collisions with other cars. Therefore, it will be an extremely difficult task to satisfy all of these prerequisites simultaneously.

2. **Vehicle Detection Algorithm Based on Deep Learning:** Vehicle detection is effective in the same manner as target identification using deep learning is. Even though one- and single-stage target detection are more popular, two-stage detection might be considered if a suitable site suggestion cannot be given after the first. In a one-step procedure, a single network analyzes input pictures, detects targets, and returns both a bounding box and a categorization label. To correctly identify a target, we use a pair of networks: one that makes area recommendations based on the input pictures, and another that forwards those recommendations to a classifier for labeling.

R-CNN with slow, medium, and quick speeds R-CNN is a common technique for two-stage target identification [11, 12]. After producing a predetermined number of targets based on regional suggestion, the most notable feature of this technique is the use of a convolution neural network to deal with prospective targets. In order to accomplish sparse sampling, (ER-CNN first takes the original picture as input and uses candidate areas. Once potential areas have been located, a convolutional neural network (CNN) collects features, and a support vector machine (SVM) assigns labels. With R-CNN, detection accuracy has dramatically increased while the algorithm's bounds have become much more reasonable [13, 14], making it the de facto standard in target detection. Yet there are also several obstacles; the detection rate issue being the most pressing of them. To do this, the network extracts characteristics from an enormous set of candidate locations without any outside assistance. As a result, the network must use resources doing several redundant computations, which in turn raises calculation costs. Between the convolution layer and the full connection layer, the SPP-net inserts a spatial pyramid pooling mechanism. It's a method for training a neural network. By performing feature extraction from input photos just once, this method may provide photographs with consistent dimensions. Before the network can do any detection, it must first use region suggestion to choose suitable areas, then use CNN to extract Roi feature information, and lastly perform category determination and position adjustment. The feature map eliminates the need for convolution by providing a direct source for all Roi feature information, leading to a more effective network.

By further optimizing the SPP network, Fast R-CNN is able to replace the support vector machine (SVM) in classification with multi-task loss. That's why it's possible to train the network for both classification and frame regression. Since traditional approaches to constructing bounding boxes include the usage of a central processing unit (CPU), this limits the maximum running speed of the system. Prior to the development of the Faster R-CNN technique, full target identification was possible using the Faster R-CNN algorithm. In order to find these possible places, the system employs an RPN network. Both the freshly generated candidate regions and the CNN then employ the convolution layer for classification. Using region mapping and spatial pyramid pooling, faster R-CNN produces a large number of candidate regions for a target and then extracts features from those regions. Finally, CNN can recognize many distinct target features at once. Faster R-CNN [15, 16] is an improved version of Fast R-CNN that speeds up region formation by using the properties of an RPN network. To begin, it checks to verify whether the candidate box contains the proper characteristics for the detection aim using a multi-task loss function. If this is the case, the procedure moves onto the next step. This will allow us to assign a name to the detected item.

3. **Vehicle Detection Using YOLO (You Only Look Once):** Initially, YOLO [17] approached object identification as a regression issue inside a single neural network. The method's rapid adoption as the de facto standard in object detection is a direct result of its stellar performance. Consistent development since YOLO's inception has resulted in five generations of the architecture: YOLO [18], YOLOv2 [19], YOLOv4 [20], and YOLOv5 [21]. The original YOLOv1 combined the three processes of feature extraction, object localisation, and classification into a single operation. Even though it had a high mAP, this network was SOTA when measuring mean average precision. The foundation of the first incarnation of the YOLO architecture were layers of convolutional and then maxpool activation functions. The network is now adaptive to picture resolution thanks in large part to the elimination of the fully-connected layer that existed at the very end of YOLOv1. The third iteration, dubbed YOLOv3, builds upon the foundation laid by its predecessors. Two prior generations, ResNet [22] and the feature-pyramid network (FPN) [23], served as inspiration for this new generation's architecture. Fast models such as YOLOv3, Faster-RCNN [24], single shot multibox object detection (SSD) [25], and Center Net [26] may achieve comparable mAPs on the COCO-2017 dataset. Every one of these configurations works with YOLOv3 to get the mAPs. But it does its job 17 times quicker. For this reason, we looked at using both YOLOv3 and YOLOv5 as the basis for our methods. Even though YOLOv4 performed well, we opted to switch to YOLOv5 since it had the same architecture and had a smaller model. The second part of this article will go into research that has employed such designs while keeping tabs on moving cars. In this research, we focus on MOT tracking systems that can operate with a single camera and identify many targets in a single video frame. Any reader with a curiosity may discover these techniques in the writers' previous works. This form of tracking relies on precise detection and the lack of occlusion [27], since a single camera can only catch one side at a time. With the identification difficulty that these components generated, deep learning models that can recognize objects even when they have partial occlusion have improved. Even if the item is partly concealed by a bigger one, modern CNNs can still generate an accurate prediction of it. Many deep learning networks, such as Faster-RCNN [20], SSD, and YOLO, have been used in the context of real-time MOT. To facilitate the development of a real-time method for monitoring autos, this research compares the efficiency of YOLOv3 and YOLOv5 in order to handle multiple video streams on a single GPU [28].

4. **Vehicle Tracking:** In this research, we focus on MOT tracking systems that can operate with a single camera and identify many targets in a single video frame. Any reader with a curiosity may discover these techniques in the writers' previous works. This form of tracking relies on precise detection and the lack of occlusion [27], since a single camera can only catch one side at a time. As a result of the identification problem that these factors introduced, deep learning models that can recognize objects even when they have partial occlusion have improved. Even if the target item is partially obscured by another, modern CNNs can nevertheless provide an accurate forecast of its position. Examples of deep learning network deployment in the context of real-time MOT include YOLO, SSD, and Faster-RCNN [20].

In order to find a series of images that best matches an object, conventional tracking systems often begin by recognizing objects in the first frames and then scanning the surrounding environment for characteristics that correlate to those objects. Conventional detectors including contour-based target identification [29], the Harris corner detector [30], symmetric integral and fluctuating transform (SIFT), and feature point-based approaches [31,32] all suffered from the same problem of false detection. Better performance was achieved, however, by using DL models to identify the objects first, and then going to match features through the traditional tracking approaches. We use DeepSORT, a tracking methodology, in combination with low-confidence track filtering, to implement the strategies presented in [33] for tracking through detection. This meant that the default DeepSORT algorithm produced less false positives. Using 3-D constrained multiple kernels, [34] recently described a method for following objects recognized by a YOLOv3 network. The use of Kalman filters made this possible. The development of more sophisticated tracking algorithms has led to a notable improvement in object tracking accuracy in recent years. However, these methods need a large amount of computing resources to execute. In this study, we propose a straightforward approach to object-centroid tracking as a means of monitoring the detection efforts of YOLO-based DL networks across several lanes of traffic in real time. Furthermore, this study evaluates the differences between YOLOv3 and YOLOv5's performance in an attempt to develop a real-time system for monitoring cars that can handle several video streams on a single GPU by using multi-threading algorithms [35].

III. PROPOSED METHODOLOGY

This study's authors suggest investigating the vehicle-recognition and tracking technique using deep learning. In this study, we use first-stage target recognition methods such the Single Shot MultiBox Detector (SSD) and YOLOv5 algorithms. The Single Shot MultiBox Detector (SSD) model architecture is then employed as the major foundation for vehicle detection. The fundamental objective of this study is to develop an automated technique for detecting and tracking autos in both static and dynamic scenes. The suggested procedure consisted of three separate actions. To begin with, YOLOv5 takes N frames at set intervals to search for and locate vehicles. To gather and evaluate characteristics of objects, we next utilize K-means clustering and the KLT tracker to follow the corner points as they travel over N-frames. The article concludes by detailing a dependable method for assigning vehicle trajectories to each of the highlighted bounding boxes. This method ensures that the labels applied to the trajectories of individual vehicles are unique from one another. You can view a diagram of the suggested solution architecture in Figure 1. Next, we'll provide a high-level overview of the most important aspects of the proposed method for detecting, tracking, and counting the number of cars.

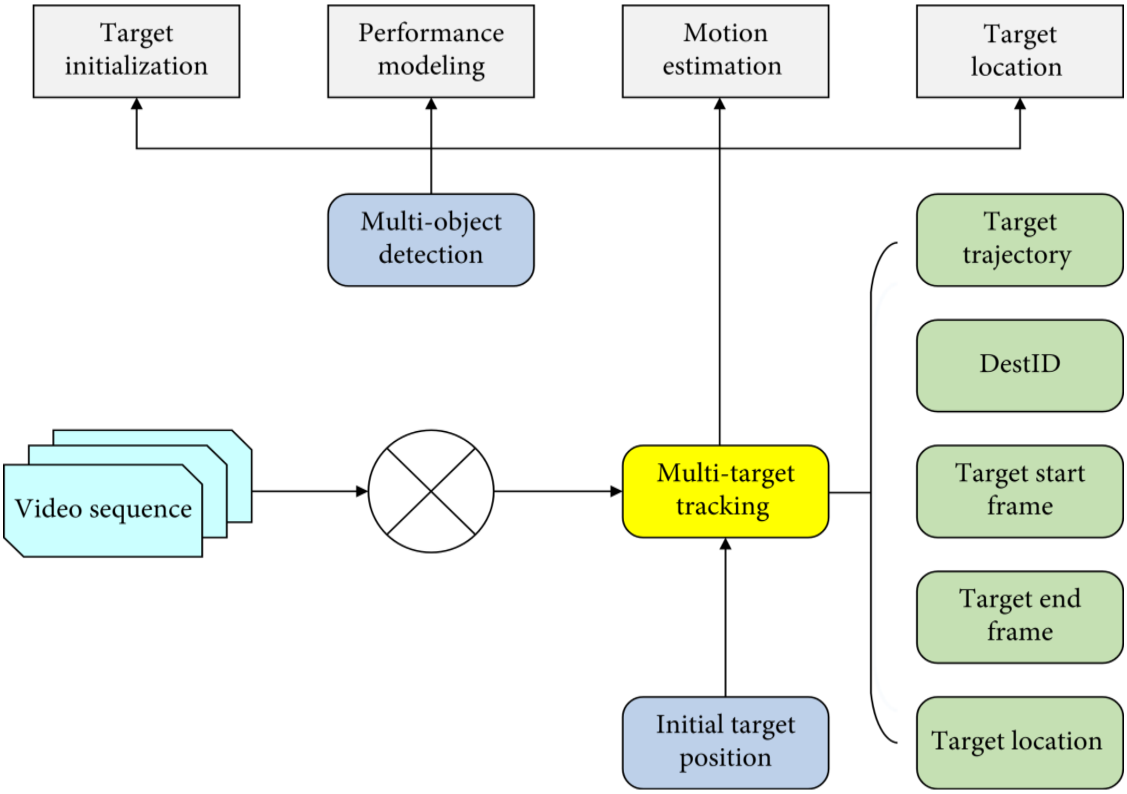


Figure 1: Architecture of vehicle detection and tracking model

1. **Vehicle Detection:** Recent years have seen a rise in the usage of deep learning-based tactics to improve detection results. Examples of such techniques are the You Only Look Once v5 (YOLOv5) [25], the Single Shot MultiBox Detector (SSD) [20], and the region proposal techniques [11, 27]. In contrast, the authors of [14] used tracking data to help with the identification and a background removal based CNN to reach a promising outcome. Despite using a convolutional neural network–based pixel classification technique for the detection, there is still room for improvement in terms of processing time. The YOLOv5 object identification technology is open-source and free to use. One of its most attractive qualities is how quickly it can differentiate between different parts of the same picture or video stream. With the Single-Shot Multi-box Detection (SSD) technique, a single forward pass of the calculating feature map is sufficient for accurate object identification. Although accuracy may degrade, it may theoretically function on real-time video feeds. Both YOLOv5 and SSD are top-tier devices; however YOLOv5 excels in speed while SSD excels in precision.

   There are three stages to the YOLOv5 method: the residual block or gridding stage, the bounding-box regression stage, and the IoU stage. The first stage involves mapping or dividing the picture using a grid of leftover blocks. Instead of executing CNN in loops for each item, this technique does a single forward pass over all cells. This approach allows for comprehensive coverage. The model can identify an item if its mass center falls inside one of its cells.

If a single cell includes the centers of many different types of objects, the model will provide a composite result matrix. When many bounding boxes intersect, a regression to the mean is conceivable. It examines the object classes included inside the enclosing boxes to see whether they represent the same objects. Using IoU, we may test whether or not the outlines of two object classes coincide. More than 50% overlap between bounding boxes causes a greater incidence of deletion. The location in the centre of the box where the average score is highest will be the winner. Non-max suppression occurs when the bounding box does not delete items based on the score.

The SSD model makes use of deep learning to perform object detection and localization. Similar to YOLOv5, it just takes a single forward pass to recognize all of the objects in a photo. It's an efficient strategy, to put it briefly. This is different from YOLOv5 since it employs bounding-box regression.

It has just come to light [23] that YOLOv5 is one of the fastest CNN-based object detection systems. Therefore, the goal of this study is to investigate the feasibility of incorporating tracking information and YOLOv5 into the detection phase to create a more effective detection and counting method. Training the deep learning algorithm used to develop YOLOv5 [29] required over a million images from the ImageNet database. ImageNet has 1.2 million unique photos over a thousand different topics. We employ the YOLOv5 layers all the way to the final fully linked one, where we limit the number of categories from a thousand to two, since auto-identification is our primary goal.

In this study, we use the YOLOv5 architecture to rapidly apply transfer learning to the detection process. In addition, we can make a precise assessment and substantially improve detection performance by combining the optical flow data into the counting approach suggested in [14]. In the last few layers, we use transfer learning by exchanging the softmax 1000 classes for the softmax 2 classes.

Transfer training makes use of pre-trained convolutional neural network models to expedite subsequent training. These models needed a lot of training data to become this good. After developing our architecture using pre-trained models up to the last, fully-connected layer, we train it from scratch on the vehicle dataset. Our hard work has finally paid off with the completion of this layer.

In [22], you can find a more thorough description of the transfer learning approach. [22] Using transfer learning, we settled on the Resnet-50 [16] as the primary neural network model for the YOLOv5 framework. Figure 2 is a block schematic of the YOLOv5 and SSD car-identification models.

This shows the model's inner workings in great detail. The convolutional neural network ResNet50 was trained using over a million photos from the ImageNet dataset. There are almost a thousand different types of tags used to organize the over 1.2 million pictures in this collection. We proposed a strategy for training YOLOv5, then tested it out over three independent sessions with varying sets of photos.

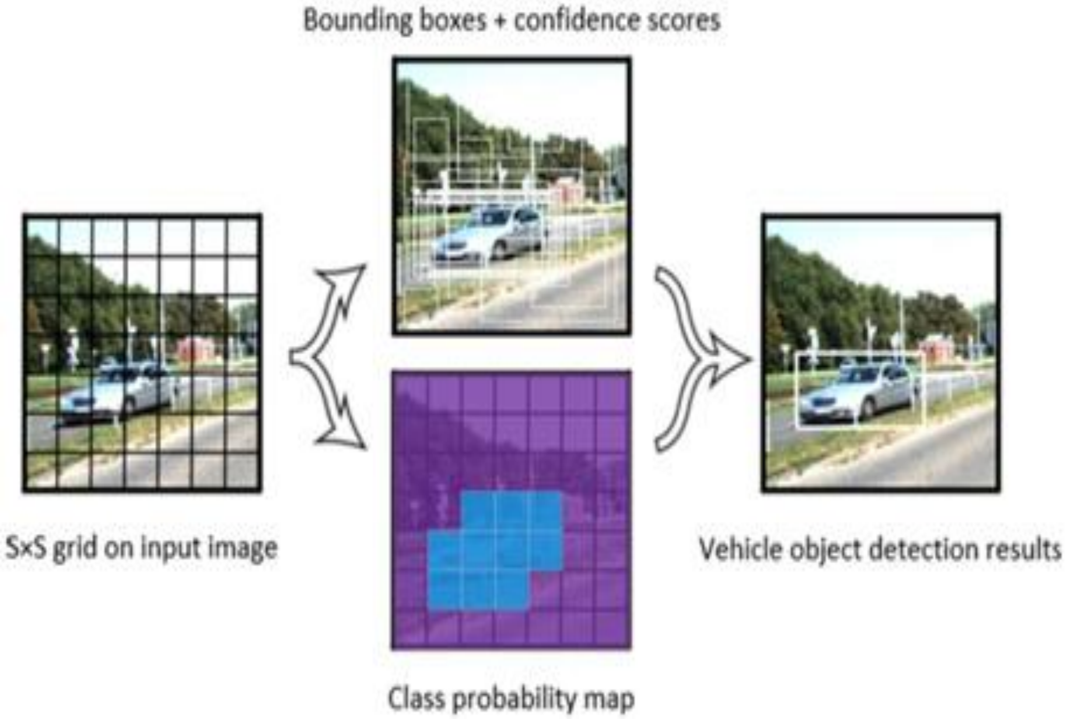

Bounding boxes + confidence scores
S×S grid on input image
Vehicle object detection results
Class probability map

Figure 2: Architecture of YOLOv5 and SSD vehicle Detection models

After some tweaking, YOLOv5 achieved a respectable degree of recall accuracy. Still, it has a low overall accuracy since it correctly identifies some false positives. In the next sections, we'll see how K-means clustering and monitoring optical flow data will be used to get rid of these spurious positives. As a result, it will be better equipped to spot serious threats. When compared to the foreground automobiles' motion characteristics, the false-positive data gathered from the background regions exhibit a large amount of variation. We use the motion data associated with particular feature points to exclude them from the verdict. Some of the methods used to cluster the feature points are described in [7, 31, and 32]. K-means clustering, on the other hand, is accurate enough for our purposes while being computationally manageable.

2. **Vehicle Features Refinement and Clustering:** At this point, we don't only group automobiles together, we also separate them from their backgrounds by wiping them clean. Processing speed and precision in matching features are both increased by optical flow tracking. As a result, we monitor the feature points between frames f and f + 1 using the optical flow Kanade-Lucas method [2]. Combining two successive images creates a new set of optical flow vectors, V, with elements Vi = (Mi, i), where S and are defined as follows:

$$M_i = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad (1)$$

$$\theta_i = Arctan\left(\frac{y_2-y_1}{x_2-x_1}\right) \qquad (2)$$

We use the notation X1 and Y1 for the X and Y coordinates of the previous frame, and X2 and Y2 for the X and Y coordinates of the next frame. In V, each number represents an individual corner point Pi captured between frames f and f+1, whereas M and denote the magnitude and angle of the corresponding displacement. Because of the erratic detections, many trackers end up failing. As noisy detections can only be followed for a limited length of time (9 consecutive frames), this study only considers a foreground detection to be a vehicle object if it has been tracked for a sufficient number of consecutive frames. After that, we use k-means clustering to group the remaining automobiles in the foreground of the picture. For further information, you may check out [36].

3. **Vehicle Counting:** After identifying the most trustworthy characteristics, we put them into their own categories for each car. We assign unique ID numbers to each of these vehicle components and track them until they are no longer visible. The allocation method starts with a region determined by the intersection of rectangular bounding boxes based on historical tracking and current detection. Two automobiles are deemed to be a perfect fit if they share a certain minimum percentage of space. The vehicle will get a new label if the junction area is either less than or does not exist at all. Below, we provide four different approaches to tally the N-frames.

- Since this is the first known instance of this sort of vehicle, the characteristics included inside its boundary are yet unnamed. After that, we'll assign these characteristics new names and increase the tally by one.
- In order for a vehicle's characteristics to be labeled, it must appear in the first frame and the prior frameset (N-frames). It would be "unmarked" if there were any traits that were not readily apparent.
- The vehicle's bounding box was visible in the previous frameset but is missing in the first frame. Since we've already settled on a label for these characteristics, that's what we'll call them going forward.
- We will refer to a vehicle as a "missed counted vehicle" if we are unable to find any instances of it across the whole set of video frames.

IV. IMPLEMENTATION OF VEHICLE DETECTION AND TRACKING USING YOLOv5

Small-sized target objects, certain size scaling in the process of continuous detection, the complexity of the vehicle environment, too many targets in a single image in the dataset, and overlapping of targets are just some of the difficulties encountered when attempting to detect the vehicle target using the vehicle dataset. The correct detection technique is very important since the success of a vehicle detection system is contingent on achieving a number of requirements. Despite the challenges, there are a few things about cars that are immediately clear. For instance, almost all car wheels are spherical. The design of an automobile's body lines is its most eye-catching aspect. In this article, we will discuss how we intend to use YOLOv5 with SSD, a deep learning technique, to recognize and track moving vehicles.

1. **Analysis of Algorithm Selection:** The algorithmic development process should prioritize speed and accuracy in target detection systems, since they are crucial in practical applications. Better item recognition is possible with larger detection accuracies, and faster detection rates make the technique applicable to more devices. Think about it In Table 1, we compare the detection rates and accuracy ratings of a number of widely used target identification techniques. Table 1 shows that Faster R-CNN has a higher detection accuracy than other one-stage detection networks, but a considerably different detection rate. This is because its use requires a greater depth of understanding. Faster R-CNN, a progressive learning approach, is responsible for this result. These single-stage detection networks clearly beat Faster R-CNN, a two-stage detection approach, with a higher detection rate. The accuracy of YOLO's detection is lower than that of competing approaches. In this work, we use YOLOv5 as the vehicle identification approach because it achieves high performance in terms of detection accuracy and detection rate and is especially well-suited for recognizing small-sized objects. YOLOv5's superiority in recognizing objects of smaller sizes was a major factor in the final selection, particularly in light of the problems we've just covered.

2. **Designing the Model**

- **Initialization Operation of Candidate Box:** The most important part of the detection process, network training, requires initialization of the network model's parameters. One of the first and most crucial steps is determining the appropriate size for the candidate box. There is a direct correlation between the quality of the first candidate box initialization and the amount of time and effort spent training and testing the network. To determine the appropriate proportions for each candidate box, YOLOv5 employs a K-means clustering technique. To measure how well k-means clustering performed, researchers calculated an assessment index using the Euclidean distance between clusters. Using the so-called Euclidean distance index, a lower figure for the distance between two objectives indicates a greater degree of similarity between them. This is because it is possible to use the Euclidean technique to determine how far apart two points are. K-means clustering begins with the selection of a set of nodes that will act as the cluster axis. Calculating the euclidean distance between any additional locations or targets and the discovered cluster centers is the next step. We may determine which cluster a target belongs to by calculating the Euclidean distance between it and the cluster's epicenter. Once we know where inside the cluster the target is hiding, we may shift the cluster's center to that point. Assuming the test run was successful, the next step is to execute the procedure again and again until the center no longer moves. The objective is to herd the targets into a compact group.

  To cluster bounding boxes with comparable contents, YOLOv5 use the K-means algorithm. Every training picture has several bounding boxes, which play a crucial role in the process. The initial step of bounding box clustering involves collecting all of the bounding boxes from the training images into a single set. Gathering the dimensions of the box is a prerequisite for doing bounding box clustering. This transition makes sense since the initial recording of those components was for locations on the top left and right corners of the box. This information is required for bounding box clustering. This takes place since the clustering method use the bounding box's width and height. It's clear why, especially in light of the previous statement. The K values of all the bounding boxes, also known as the beginning values of the boxes, must be chosen as the clustering values of the anchor box clusters once the data has been processed; the value of K chosen for this article is 9.

Table 1: Comparing the Detection-Rate and Accuracy-Rate Of A Number Of Popular Target-Detection Algorithms.

Training Set	**Algorithms**			
	FRCNN	**YOLO**	**SSD**	**YOLOv5**
VOC2007 + 2012	0.765	0.772	0.783	0.792
VOC2007 + 2012	0.958	0.963	0.972	0.987
VOC2007 + 2012	0.865	0.871	0.883	0.896
VOC2007 + 2012	0.784	0.789	0.793	0.798
Training Set	2010	2010	2010	2010
MAP	78.4	68.8	81.3	91.7
FPS	21	49	43	54

- **Detection Module of Network:** YOLOv5 is able to do feature extraction because to the support of DarkNet-53. The network is able to extract more feature information because of its deep-level structure. But as seen in Figure 3, there are also problems for the network at extremely deep levels. As a result, training a deep network will result in a decrease in the network's effectiveness while attempting to recognize small objects. Users of both the deep neural network and other one-stage detection algorithms have recently noticed these issues. DarkNet-53 is able to considerably enhance the learning capacity to image features since it is built on the concept of residual networks and makes use of residual connections. In addition, it compensates for the fact that it can't pick up on the tiniest of details.

The success of any detection network relies heavily on the design of its loss function. When training a model, the loss function is what ultimately matters. The loss function considers the target to be a positive sample if and only if the difference between the projected frame and the actual frame created during training has the largest possible IOU value. The loss function shouldn't consider the target to be a loss if the anchor frame's IOU isn't at its maximum. As a result, there is only one connected prediction frame for each previously gathered actual frame. When determining the severity of the damage, we take into account the loss of not just beliefs but also logical categories and anchor frame locations. It is usual practice to question whether or not a certain cell in the detection layer contains the center point of the target object when expressing confidence.

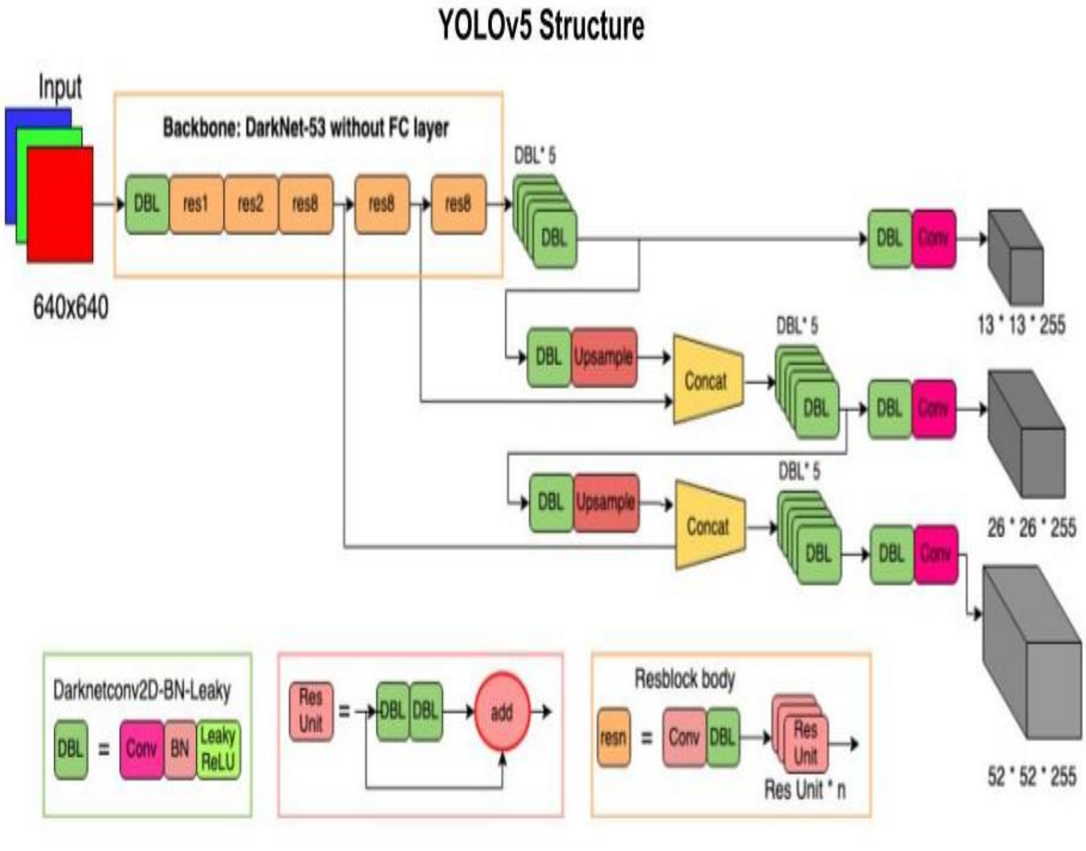


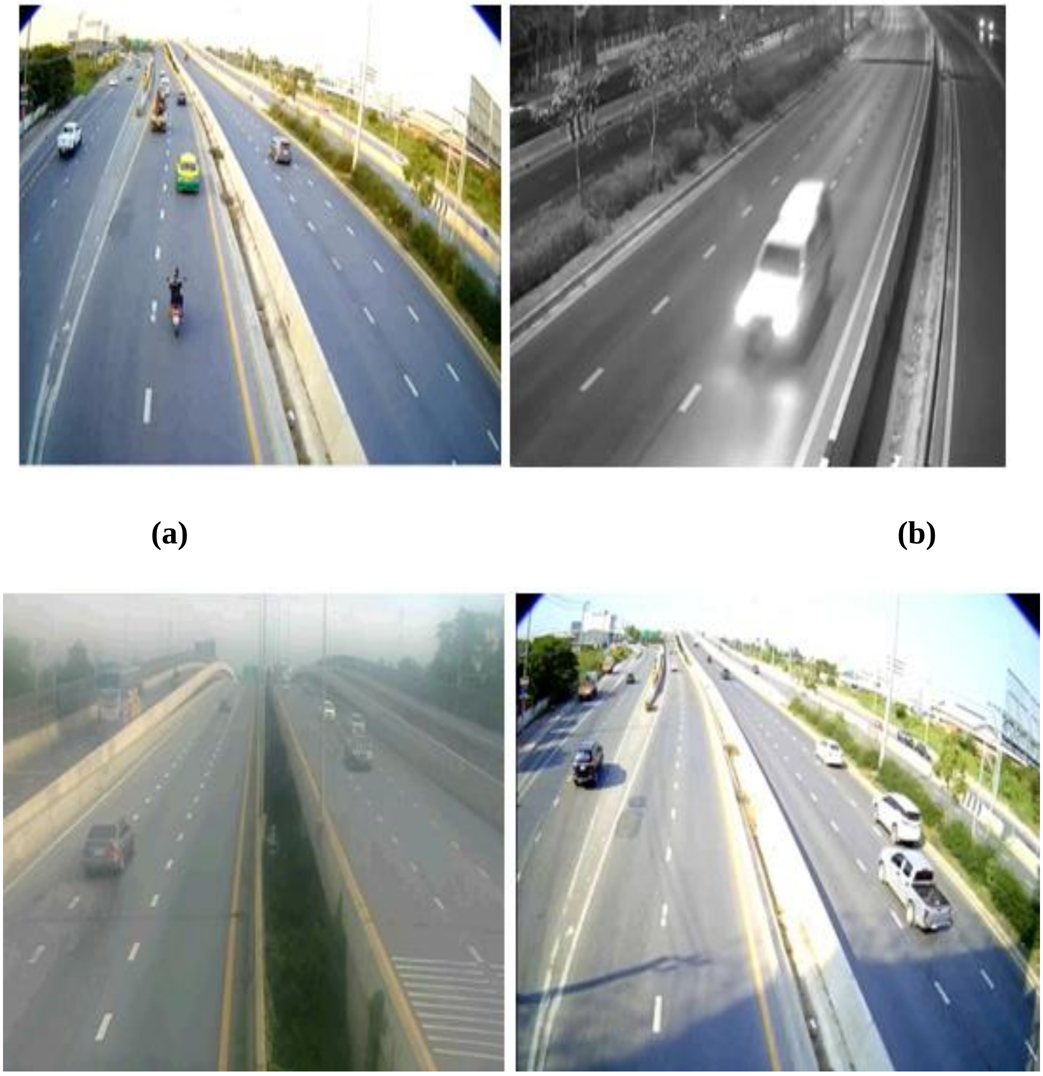
Figure 3: Network architecture of YOLOv5 with a backbone of DarkNet-53.

V. EXPERIMENTAL RESULTS AND ANALYSIS

This section primarily compares the speed and performance of the proposed vehicle recognition and tracking method against that of current high-performance CF trackers using publicly accessible datasets, and analyzes the results. In this subsection, we will compare the performance of the proposed algorithm for vehicle recognition and tracking to that of the high-performance CF trackers.

1. **Dataset Selection:** When developing a system for target identification using a deep learning technique, the selection of an appropriate dataset is a crucial and challenging step. In this study, we are developing a model and have chosen to train it using the BDD100K picture dataset. Figure 4 displays some representative data from the set. All of the images included in the offered data were taken from a moving vehicle on a public route. These pictures show a broad variety of vehicles, as well as human and non-human victims. We only shot at six of the 10 target categories available to us. Vehicles, buses, passengers, trucks, motors, and bicycles all count as means of transportation.

ii.



(a) (b)

(c) (d)

Figure 4: Data images of BDD100K

2. **Processing of Data:** Both a "picture" and a "label" portion make up the downloaded data. All the fresh information was actually saved as JSON files, and each JSON file does indeed map to an image with the same name. The naming conventions reveal this to be true. It includes details such as the image's filename, the category to which it belongs, the category's name, and the image's coordinates for the indicated box. The position data includes the horizontal and vertical coordinates of the top left corner and the bottom right corner, specifying the precise location of the box down to the millimeter. Below there, you'll see the horizontal and vertical coordinates for the bottom right corner. Using these two points of reference, you can determine where the box is. Many distinct types of storage units may appear simultaneously in a single image. The additional data included in JSON-formatted files might make their immediate usage difficult. That's why it's important to make the changeover to this format, which will simplify the file's data structure and get rid of any extraneous data. Converting JSON data to XML is a must for

moving on. Aside from its reduced size, the converted JSON file also benefits from a more intuitive presentation of its contents. The coordinates and width and height of the picture are recorded simultaneously. The generated xml file is not suitable for use as training input, but rather it is comparable to the output of specialized tagging programs. There has to be a simplification of the training input data's presentation, with extraneous information deleted. Finally, we just keep track of where the picture is located, where the box is in relation to the image, and what kind of information the box contains. The last step is to save the data as a text file with the.txt extension. This file should additionally provide the picture's box position inside the image and its category in addition to the image's absolute location and name. We have merged all the information from the training images into the final text file. Images with input boxes may be located using the absolute path, and the image's input contents follow next. Please click this link to go to the requested location. Consolidating many image or data files into one text file makes them far more manageable for online transfer and sharing.

3. **Detection of Model:** To complete the process of building a vehicle detection model, one must first train the model before loading the trained weight onto the model to carry out the actual detection. The training technique may make use of the parameter sets shown in Table 2. Target identification throughout the three backbone network layers is where YOLOv5 often shines in both training and detection. These are the lowest, middle, and highest tiers of the network's central infrastructure.

Table 2: Parameter Settings of the Training Process On Various Models

Models	**mAP/ %**	**Vehicle Detection and Tracking/FPS**	**Memory size**
FRCNN	92.4	20	200.3
YOLOv3	85.7	30	243.6
SSD	92.5	35	15.7
YOLOv5	97.6	39	15.9

4. **Result Analysis of Vehicle Detection and Tracking:** To guarantee a high-performing trained model, it's important to choose a few parameters before beginning the detection network training process. We ran a performance comparison experiment using state-of-the-art object identification methods, including Faster R-CNN, YOLOv3, SSD, and YOLOv5, utilizing the same setup environment and dataset to further analyze the improved outcome after applying our redesigned YOLOv5 methodology. Our goal in doing so was to verify that the improvements we had seen with the new YOLOv5 algorithm were really the consequence of those improvements. We capture FPS using an NVIDIA GTX1660Ti graphics processing unit. Table 4 displays the results for both the Track Maintenance dataset and the BDD100K dataset.

Our method's recognition accuracy is higher than that of Faster R-CNN, YOLOv3, SSD, and YOLOv5, as shown in Table 4. Our improved YOLOv5 model has a 5.1% higher mAP value (mAP at 0.5: 0.05: 0.97) than the previous version. Our enhanced model is 1.12 times faster than YOLOv5, 1.45 times faster than YOLOv4, and 2.9 times faster than Faster R-CNN in terms of detection speed, indicating that it can match the requirements of real-time detection. Compared to the YOLOv5 algorithm, our enhanced technique uses somewhat more memory, but still only about a seventeenth as much as YOLOv3 and about a thirteenth as much as Faster R-CNN. The breakpoint continuation method allows the learner to control the rate of progress. Initiating model training at a high learning rate and then dropping to a lower rate for optimization. There is now an epoch value of 50 in effect. Assuming the model can be optimized further, the epoch may be retrained a certain number of times using the breakpoint continuation method. This is a genuine option if there is room for development. Figure 5 depicts the monetary value of loss when training is complete.

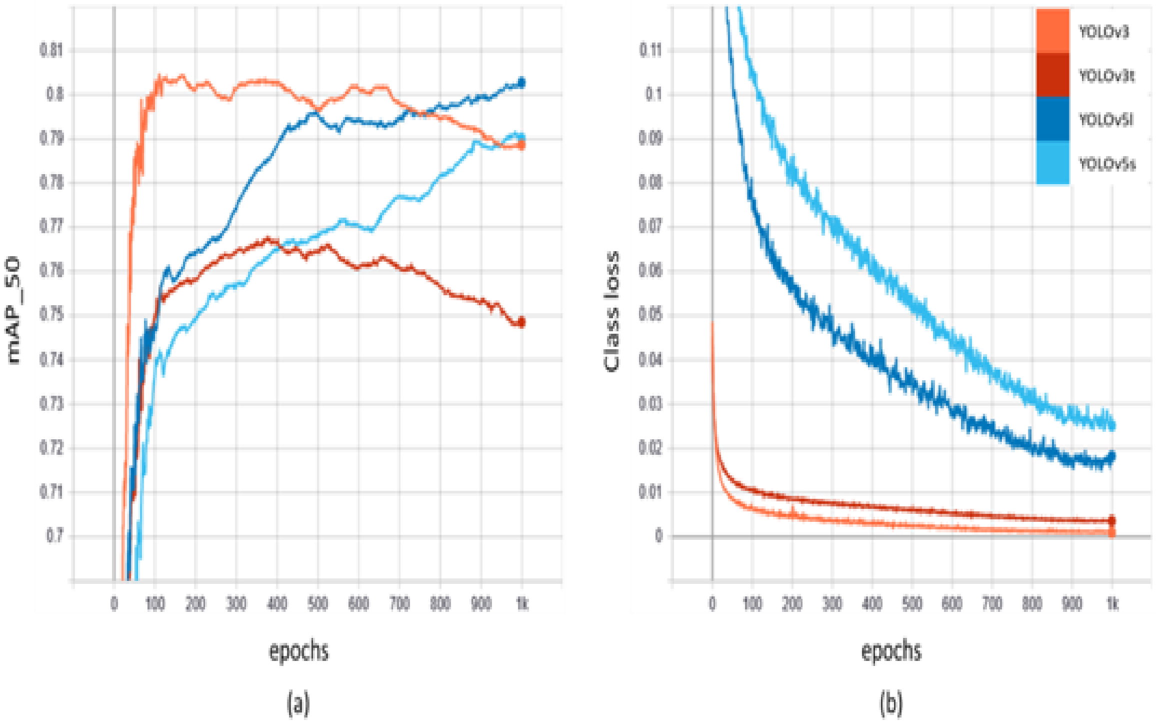


Figure 5: Training results of FRCNN, SSD, YOLOv3t, YOLOv5l, and YOLOv5s on the BDD100K dataset.
(a) mAP at IoU = 50 (b) loss of classification.

The image's orange curve shows the loss value for the test set, while the image's blue curve shows the loss value for the training set. After a few repetitions of training, the initial high loss numbers begin to decrease. Values start to converge and the rate of decrease slows. Comparing results after training and testing reveals that the model is steadily nearing convergence, even though the loss value is still rather large. The values of the training set loss tend to decline with time, stabilize, and then fluctuate within a small range. Adjusting the learning rate in accordance with the breakpoint continuation strategy is necessary to keep training going, even if the final results won't be that different from the second portion of Figure 6.

To evaluate the efficiency of a target detection model, we look at how well it maps to reality. The ap value for a given category indicates how well the detection network can distinguish between that category and all others; the map value is the aggregate of all ap values. Using this data, the map shows the trained model's value of 72.8. The numbers show that its efficiency is poor. The detection performance of buses and other motor vehicles has decreased dramatically in recent years, whereas that of automobiles remains high. The detection box scores in Figure 6 remain constantly above 0.7, indicating that the model can correctly categorize the great majority of cars. The detection impact of trucks is around average, that of bus types is a little lower, and that of detection boxes is typically minimal. In Figure 7, we can see how well the network can recognize both people and bicycles.

Figure 7 demonstrates that although the detection model does not substantially affect either the person or bicycle categories, the human category scores much higher than the bicycle category scores. The extension of available box possibilities and the subsequent increase in "person" detection box accuracy are other noteworthy developments. Figure 8 demonstrates the detection method's outcomes, providing evidence that the bus category detection effect is general. During the course of the inquiry, this evidence may become apparent. In the image on the left, there are three buses, but only the first bus has a score that is greater than the others, therefore it is impossible to classify them.

However, as shown on the right, the container designed to represent the bus type being utilized is strangely devoid of any contents. Because of this, the detection box does not include the whole perimeter of the vehicle. The detection impact of the human category is clearly superior to that of the motor category, whereas the motor category is just slightly less effective. The "person" category likewise has much more elevated scores. While the detection area might be larger, the detection accuracy is fine for both the motor and bus categories. The model used in this study gets rather good results for detection overall; it excels most at identifying cars and people. This is true for both cars and humans. There are millions of potential victims in the vehicle category, but there are over 10,000 in the bus category and over 4,000 in the motor category. The following are many of the reasons why.

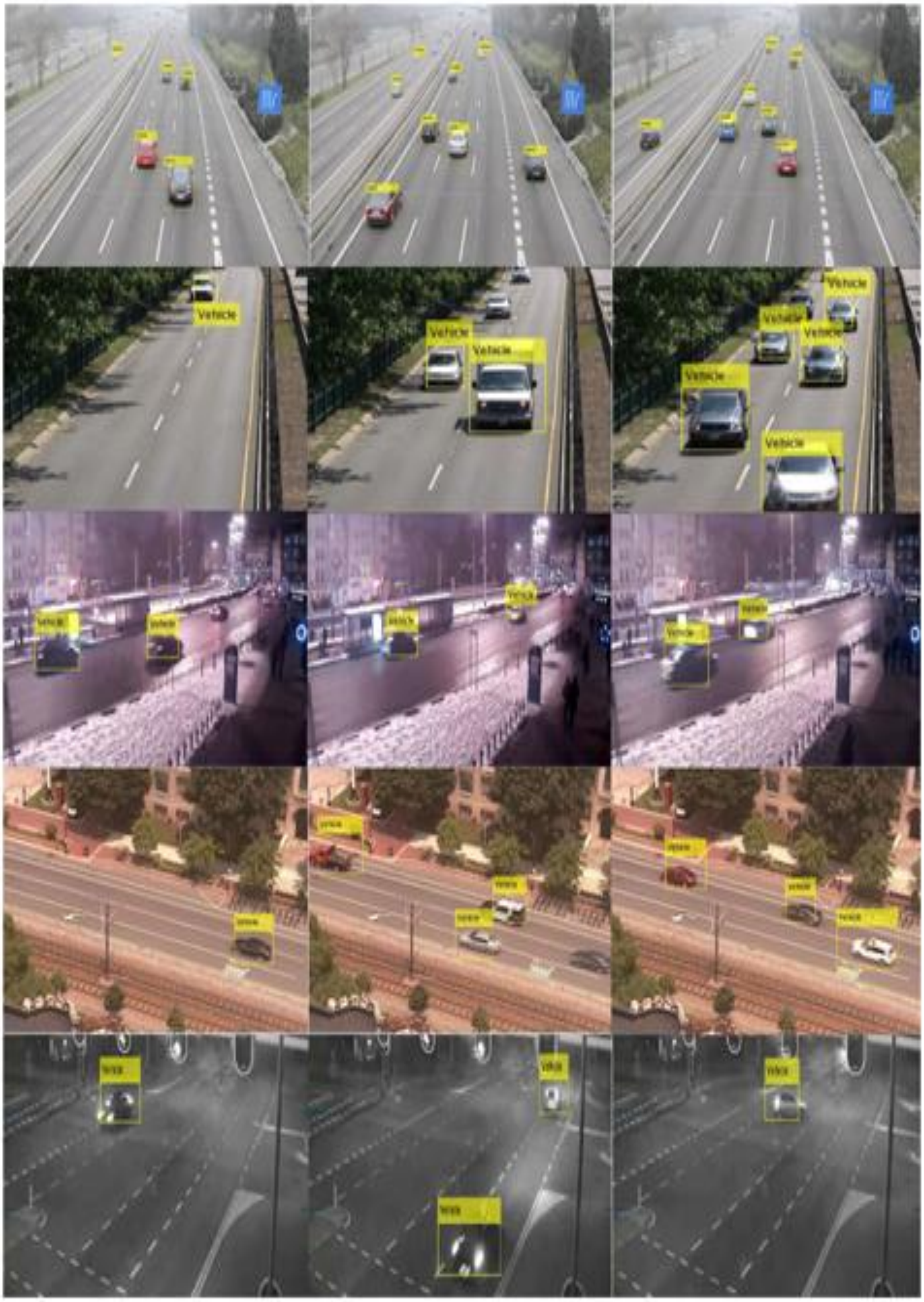


Figure 6: Detection of Vehicles on the Road from Different Camera Views By Day Scenes and Night Scenes

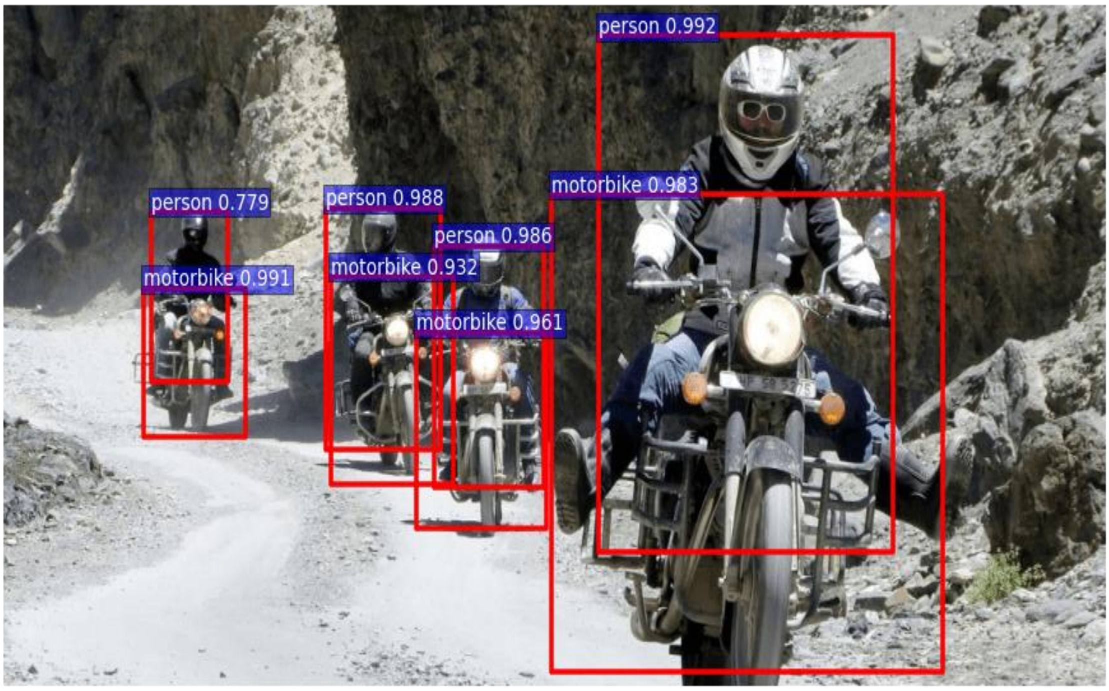


Figure 7: Illustrating the Detection on Bike Category And Person Category

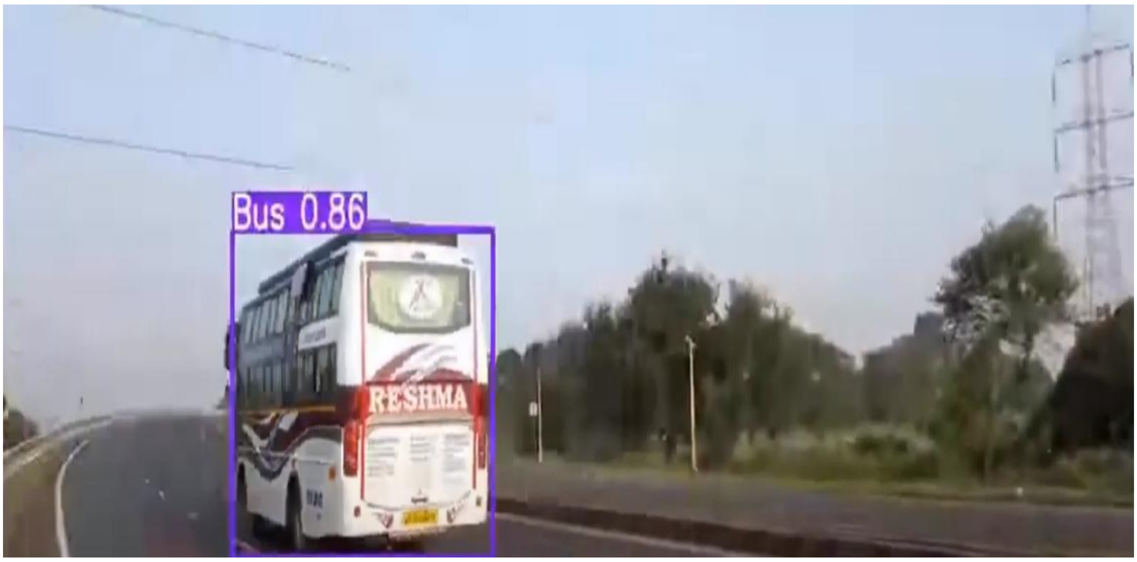


Figure 8: Vehicle Detection on Bus Category

Thus, the vast majority of the model's training data pertains to the car category, the vast majority of the retrieved features are pertinent to the vehicle classification, and the vast majority of the updated parameters appear to have been modified in a way that makes identifying the vehicle classification simpler as a result of the reverse-feedback process. Therefore, even after so many years, vehicle categorization detection remains very precise.

It's easy to believe that many individuals are making use of this option, given there are currently more than 100,000 objectives in the "person" category. Due in part to the fact that their feature information is considerably distinct from that of the many categories of vehicles, humans, or the person category, have witnessed a large jump in their detection impact.

VI. CONCLUSION

In this study, we use deep learning to automate the process of music production. For tracking down people and machines on the track, we also suggest updating the YOLOv5 and SSD algorithms. We make substantial use of the YOLOv5 and SSD algorithms to dramatically enhance the mean absolute performance (mAP) at IoU and the loss of classification for the different sets of training outcomes. Newer YOLOv5 and SSD algorithms allow for faster convergence and better recognition of obstructed vehicle objects and tiny vehicle objects. These benefits are cumulative. The testing findings demonstrate the excellent resilience of the newly developed YOLOv5 and SSD technique. By applying these algorithms, we are able to perform thorough inspections of construction workers and equipment, addressing the problem of low detection accuracy for complex scene issues like occluded vehicle objects and small vehicle objects, and meeting the practical requirements for vehicle detection in the context of track construction safety. The findings of this work provide credence to the practical use of intelligent detection tools and lend momentum to the thorough investigation and advancement of track safety vehicle detection technology. According to the measured KPIs, the combination of the YOLOv5 and SSD algorithms is the most successful in terms of both vehicle detection and tracking precision.

REFERENCES

[1] Liu, Z., Wang, S., Yao, L. et al. Online Multi-Object Tracking Under Moving Unmanned Aerial Vehicle Platform Based on Object Detection and Feature Extraction Network. J. Shanghai Jiaotong Univ. (Sci.) (2022). https://doi.org/10.1007/s12204-022-2540-4

[2] H. Song, J. Zhu, and Y. Jiang, “Two-stage merging network for describing traffic scenes in intelligent vehicle driving system,” IEEE Transactions on Intelligent Transportation Systems, no. 99, pp. 1–12, 2021.

[3] H. (omas and H. Michael, “Camera-based method for distance determination in a stationary vehicle, corresponding computer program and corresponding parking assistance system with camera,” 2020, https://patents.google.com/ patent/WO2012076400A1/en.

[4] Y. Chen, W. Li, C. Sakaridis, and D. Dai, “Domain adaptive faster R-CNN for object detection in the wild,” 2018 IEEE, https://arxiv.org/abs/1803.03243.

[5] I. Papakis, A. Sarkar, A. Svetovidov, and J. S. Hickman, “Convolutional neural network-based In-vehicle occupant detection and classification method using second strategic highway research program cabin images,” Transportation Research Record Journal of the Transportation Research Board, 2021.

[6] G. Goedert, D. Jungen, J. Beck et al., Weight -responsive Vehicle Seat Occupant Detection and Classification Method and System, US20180244172A1[P], 2018.

[7] M. A. A. Al-qaness, A. A. Abbasi, H. Fan, R. A. S. H. Ibrahim, and A. Hawbani, “An improved YOLO-based road traffic monitoring system,” Computing, vol. 103, no. 2, pp. 211–230, 2021.

[8] H. Yin, Bo Chen, C. Yi, and Y. D. Liu, “Overview of target detection and tracking based on vision,” Acta Automatica Sinica, vol. 42, no. 10, pp. 1466–1489, 2016.

[9] M. Qi, Y. Pan, and Y. Zhang, “Preceding moving vehicle detection based on shadow of chassis,” Journal of Electronic Measurement and Instrument, vol. 26, no. 1, pp. 54–59, 2012.

[10] T. Schamm, C. Von Carlowitz, and J. M. Zollner, “On-road vehicle detection during dusk and at night,” in Proceedings of the Intelligent Vehicles Symposium, pp. 418–423, IEEE, La Jolla, CA, USA, 21 June 2010.

[11] J. Hu, Research on Fast Tracking Algorithm of Moving Target Based on Optical Flow Method, Xidian University, Xi'An, City, 2014.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, IEEE, Columbus, OH, USA, 23 June 2014.

[13] R. Girshick, "Fast R-CNN," in Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448, IEEE, Santiago, Chile, 7 December 2015.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R- CNN: towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2015.

[15] K. Han, H. Zhang, Y. Wang et al., "A vehicle detection algorithm based on faster R-CNN," Journal of Southwest University of Science and Technology, vol. 32, no. 4, pp. 65–70, 2017.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN:towards real-time object detection with region proposal networks," in Proceedings of the Advances in neural information processing systems, pp. 91–99, Montreal, Quebec, Canada, 7 December 2015.

[17] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 Jun–1 July 2016; pp. 779–788.

[18] Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.

[19] Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv:1804.02767.

[20] Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. arXiv 2020, arXiv:2004.10934.

[21] Jocher, G.; Nishimura, K.; Mineeva, T.; Vilari no, R. Yolov5. 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 15 June 2021).

[22] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.

[23] Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

[24] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. 2015, 28, 91–99.

[25] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.

[26] Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6569–6578.

[27] Wu, C.W.; Zhong, M.T.; Tsao, Y.; Yang, S.W.; Chen, Y.K.; Chien, S.Y. Track-clustering error evaluation for track-based multi-camera tracking system employing human re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 1–9.

[28] Wende, F.; Cordes, F.; Steinke, T. On improving the performance of multi-threaded CUDA applications with concurrent kernel execution by kernel reordering. In Proceedings of the 2012 Symposium on Application Accelerators in High Performance Computing, Argonne, IL, USA, 10–11 July 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 74–83.

[29] Meyer, D.; Denzler, J.; Niemann, H. Model based extraction of articulated objects in image sequences for gait analysis. In Proceedings of International Conference on Image Processing, Santa Barbara, CA, USA, 26–29 October 1997; IEEE: Piscataway, NJ, USA, 1997; Volume 3, pp. 78–81.

[30] Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988.

[31] Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 2014, 37, 583–596.

[32] Cui, Z.; Xiao, S.; Feng, J.; Yan, S. Recurrently target-attending tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1449–1458.

[33] Hou, X.; Wang, Y.; Chau, L.P. Vehicle tracking using deep sort with low confidence track filtering. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.

[34] Liu, T.; Liu, Y. Deformable model-based vehicle tracking and recognition using 3-D constrained multiple-Kernels and Kalman filter. IEEE Access 2021, 9, 90346–90357. [CrossRef]

[35] Wende, F.; Cordes, F.; Steinke, T. On improving the performance of multi-threaded CUDA applications with concurrent kernel execution by kernel reordering. In Proceedings of the 2012 Symposium on Application Accelerators in High Performance Computing, Argonne, IL, USA, 10–11 July 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 74–83.

[36] Gomaa A, Abdelwahab MM, Abo-Zahhad M, Minematsu T, Taniguchi R (2019) Robust vehicle detection and counting algorithm employing a convolution neural network and optical flow. Sensors 19(20):4588