Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Automated Recognition and Tracking of Stationary and Moving Cars in Images and Videos: YOLOv5 and SSD Analysis

Priyanka Ankireddy[1,*], S. Gopalakrishnan[1], V. Lokeswara Reddy[2]

[1] Department of Information Technology, Hindustan Institute of Technology and Science, Chennai, Tamilnadu-603103, India
[2] Department of Computer science and Engineering, KSRM College of Engineering, Kadapa, Andhra Pradesh-516003, India

ARTICLE INFO

Article history:
Received 3 September 2023
Received in revised form 20 November 2023
Accepted 5 January 2024
Available online 28 February 2024

Keywords:

Vehicle detection; Image processing; Vehicle tracking; Deep learning; Object tracking

ABSTRACT

The detection and tracking of cars are a significant and useful aspect of traffic surveillance systems, which is essential for the efficient management of traffic and the security of drivers and passengers. The primary objectives of this investigation are vehicle detection and tracking. The automatic detection of cars in both still photos and video recordings are the main topic of this investigation. One of the many uses for Deep Learning, which combines fuzzy logic, neural networks, and evolutionary algorithms, is the detection and tracking of moving objects. In this work, the key object detection techniques YOLOv5 and SSD were used to analyse vehicle recognition and tracking with deep learning. The Single Shot Multi-Box Detector model architecture is then utilized as the main foundation for the detection of vehicles. The vehicle recognition model is then trained using the YOLOv5 and SSD algorithms, each of which contributes to the illustration of the detection effect. Comparing the detection rates obtained by both models on a range of cars is necessary to locate it. The objective of this work is to create an automated system for locating and tracking both moving and stationary cars in still images and moving movies. According to the research, using this method has improved the success rate of recognizing automobiles to 97.65%.

1. Introduction

Automatic vehicle data recognition has found use in several contexts, including the vehicle information system and the intelligent traffic system. Because of advancements in digital picture technology and advances in processing capability, it has drawn considerable attention from academics since the beginning of this decade. Automatic vehicle detection systems are a fundamental component of many modern traffic management applications [1,2]. As the population and the infrastructure that supports it continue to rise, so too does the pressure on those responsible for their management. The world's population is growing at an incredible rate. The result was a surge in the production of vehicles and other mechanical devices. However, the cautious management of emerging issues including traffic, accidents, and a wide range of other concerns is essential. It's

* *Corresponding author.*
E-mail address: priyasivakrishna99@gmail.com

https://doi.org/10.37934/araset.40.2.221241

challenging to keep up with them using the same methods that have always been used, thus fresh developments and innovations have been made to help humans achieve all of their goals. Inconvenient traffic in metropolitan areas and on major thoroughfares is one example. A traffic light and a sign are just two examples of the measures taken to address this problem. In and of themselves, it would appear that these solutions fall short.

Technologies like object detection and tracking are being created to generate useful information for decision-making. The purpose of developing these tools is to implement automatic video monitoring. Many different types of issues have been tackled using these occurrences. The state-of-the-art Intelligent Transport System (ITS) includes a wide variety of subsystems, including object identification and tracking. Vehicles, lanes, traffic signals, and even specific makes and models can all be identified by this technology. Additionally, it can identify specific automakers. Vehicle recognition and classification could improve traffic and roadways, lessen the likelihood of accidents, and help authorities keep track of violations. Humans have an innate talent for identifying cars, trucks, and other vehicles in moving or still images. The many kinds of data play a crucial role in computer algorithms and programs. The degree of difficulty could affect a variety of issues, namely the absence or presence of certain elements (such as favourable weather or sufficient lighting). Meanwhile, cars may be found in a dizzying array of types and styles. Further, the new challenge may lie in the fact that video objects vary in size and shape, making it harder to identify them in real-time. Researchers have developed a variety of devices to locate and track moving targets. Fuzzy, neural network, evolutionary, CNN (Convolutional Neural Network), RNN (Recurrent Neural Network), and many other algorithms are used in the many various implementations of these methods. There is constant evolution in this field because of the business world's fixation on this system or Computer futurist. This paper examines fuzzy algorithms, neural network algorithms, and evolutionary algorithms to compare their strengths and weaknesses and draw conclusions about their potential practical applications.

As people's incomes and level of living have improved, more of them have been able to afford personal automobiles. This, in turn, has increased the number of people who have both new and high expectations for their automobiles. As a result, public anxiety about autonomous vehicle technology is rising. With the advent of more and more cars on the road, travel has become both more convenient and more complicated. As traffic increases, it becomes more difficult to get where you need to go, and this rising demand has drawn a lot of people's attention to the concept of intelligent transportation. Deep learning's popularity and widespread deployment have spurred the rapid growth of computer vision in past years. Numerous practical uses of vision technology for computers, such as autonomous cars, face recognition systems, and picture segmentation, currently exist [1-3].

The purpose of vehicle detection and classification is to locate a vehicle in an image or video [4]. Vehicle tracking efficiency is an important step in traffic surveillance or surveillance. Therefore, autonomous vehicle detection methods must accurately detect traffic objects such as passenger cars, vehicles, police vans, and bicycles in real time. To achieve good governance and make the right decisions to ensure public safety [5].

With the development of DNNs, automated vehicle detection has made significant progress in recent years, for example in automated driving systems (ADS) and driver assistance systems related to traffic jams and driving safety issues [6]. Vehicle localization is an important aspect of the development of intelligent and autonomous systems such as self-driving, tracking, object detection, or tracking systems [7]. Self-driving is a new high-tech innovation based on the ability of vehicles to find themselves [8]. Traffic violations, traffic accidents, and theft caught on security cameras are common in major cities. Traffic monitoring system detectors must be fast, accurate, and reliable enough to detect vehicles in real-time. There have been many developments in traffic management

systems and surveillance technology. When evaluating vehicle detectors two main conditions are generally considered. It is capable of real-time detection and accuracy in detecting traffic objects in harsh weather conditions.

The development of DNNs has made it possible to reliably use ML-based models to detect various vehicles, but the training speed of deep learning networks is very slow in terms of CPU calculations. However, the training period is significantly reduced. Contribute to the development of technologies such as GPUs and TPUs. Compared to standard ML-based approaches, DNNs show significantly better performance in different scenarios. Self-driving and self-driving intelligent vehicles, intelligent observations, smart city-based apps, etc. Neural network-based DNNs are advanced classes of machine learning that can help solve complex modelling problems that are difficult to solve with simple statistics.

In addition, CNNs, a type of deep neural network, are widely used for image recognition and classification. These are algorithms that can recognize different objects such as license plates, cars, people, and all kinds of other objects. An important advantage of CNNs is that important features are extracted without human intervention after the training process. Various versions of CNN such as R-CNN, Fast-RCNN, and Faster-RCNN are the most popular and widely used CNN approaches. However, the computational load is still too high for devices with limited computing power and limited space for photo processing. D-based algorithms have long been recognized as powerful image recognition tools. CNN-based techniques are widely used in recent approaches to many vehicle detection algorithms and can be divided into region-based techniques and regression techniques.

YOLO is a new way to detect multiple vehicles in one step. YOLO treats the vehicle perception problem as a regression problem by classifying the images using a CNN used to achieve reliable vehicle detection. YOLO can determine the location, category, and reliability rating of objects, as well as improve detection speed and detect foggy vehicles in real-time. The regression-based YOLO technique is one of the most advanced methods for predicting constrained context and class probabilities directly on a single neural network. As a result, the YOLO model was created, which speeds up the process of object detection and localizes its position in an image. Use CNN to simultaneously detect multiple features in an image. Combine predictions from multiple feature maps at different resolutions to handle vehicles of different shapes and sizes. With the development of YOLO-based methods such as YOLOv3 and YOLOv5, YOLO continues to offer excellent performance in terms of processing time and accuracy.

2. Related work

The technology used to spot moving vehicles can be compared to that used to spot moving targets. The primary objectives of the task, which may be divided down into the location of targets and classification of targets, are similar for both vehicle detection and target detection. There are four main types of vehicle identification algorithms: those that use historical data, those that rely on deep learning, those that employ YOLO, and those that simply follow a vehicle's path.

2.1 Traditional Vehicle Detection Algorithm

Both traditional vehicle detection methods and traditional target detection techniques involve the manual design of the target's properties. Among these is the knowledge-based detection algorithm, which can recognize a car by its unique shape, as well as its lines, shadows, and other edge features. As a result, the position of the vehicle in the photograph by comparing the grey levels of the shadows under the car to the bounding box of the vehicle's body can be identified. Hence, the

purpose of detection and tracking will be met. Finally, the system determines that the automobile has been seen. While this technique can be used for detection, it will be highly sensitive to the illumination. This is because light will play a crucial role in determining the grayscale of its image. It's worth noting that anything else that shares the vehicle's size and shape will cast the same shadow as the car. Due to this inaccuracy, the method is unfit for applications requiring pinpoint accuracy. In addition to sensing brake lights, the recognition of automobiles can be achieved by the artificial construction of vehicle features. Watching a car's brake lights is one method of identifying it, however, the detection effect gained by this method is insufficient owing to the high limitations placed by the light's impact. Even though the prior knowledge-based detection technique was successful in completing the detection work by using the vehicle's properties, it has difficulties in meeting the requirements of the detection task due to its limitations and a low detection recognition rate.

A vehicle identification system based on a very basic application of machine learning is also used in conventional methods of vehicle detection. Combining a vehicle-centric algorithm with a machine learning algorithm [9-11] allows this system to recognize cars based on their unique properties. Shallow machine learning's introduction of SVM and HOG features has the potential to increase detection accuracy, but some drawbacks must be taken into account. One of the main reasons for the delay in its development is the extensive modelling effort needed to replicate complex and ever-changing traffic conditions. Frame difference approaches, streamer methods and backdrop modelling methods are some examples of traditional methods for detecting autos [12,13]. These techniques are among the most widely employed today. Vehicles can be spotted using a variety of alternative means. While this kind of algorithm can yield precise recognition results in lab conditions, it is nevertheless subject to several limitations that are influenced by things like lighting and weather [14,15]. This makes it tough to keep up with the needs of real-time vehicle detection and makes it impossible to adjust to the ever-changing realities of traffic on the roads. In addition, this makes it more challenging to dodge other vehicles. Therefore, fulfilling all of these requirements simultaneously will be a formidable challenge.

2.2 Deep Learning-Based Vehicle Detection Algorithm

Similarly, deep learning-based target detection works and vehicle detection works too. While one- and single-stage target detection are more commonly used, two-stage detection may be considered if a recommendation for a candidate location cannot be made after the first step. An individual network processes images, identifies targets, and then outputs a bounding box and a classification label in a single-stage target identification process. To successfully identify a target, two separate networks: the first network recommends regions based on the images provided as input, and the second network sends the recommended regions to a classifier for labelling was used.

Various R-CNN speeds, Rapid R-CNN, and Faster R-CNN is a common approach for two-stage target identification [16,17].The most significant aspect of this strategy is that it employs a convolution neural network to handle candidate targets after first generating a fixed number of targets based on regional recommendations. Starting with the original image as input, (ER-CNN performs sparse sampling using candidate regions. After identifying candidate regions, CNN extracts features, and SVM labels the data. R-CNN has significantly outperformed the standard gold target detection technique detection accuracy while making the boundaries of the algorithm substantially more manageable [18,19]. But there are also a lot of challenges, the most notable of which is the detection rate problem. The network can achieve this by extracting features from a wide pool of potential regions without any external help. This causes the network to perform numerous

unnecessary calculations, which in turn generates a huge quantity of unnecessary calculation redundancy and drives up calculation costs. The network is taught with this technique. With this technique, uniformly sized photographs are generated by extracting features from the input images only once. Specific system must first use region suggestions to choose candidate regions and then use CNN to extract the feature information, and finally conduct category decision and position modification before it can detect anything. With the feature map serving as a direct source of all Roi feature information, however, the need for convolution is much diminished, resulting in a more efficient network.

Fast R-CNN substitutes the support vector machine (SVM) in classification with multi-task loss employing further optimizing the SPP network. Due to this, the network may be used to train not only for classification but also for frame regression. Since conventional methods of developing bounding boxes require the use of a central processing unit (CPU), the system's greatest possible operating speed is limited by this requirement. An RPN network is used by the system to retrieve the potential locations. After then, the network's convolution layer is used by both the newly produced candidate areas and the CNN for classification. Faster R-CNN generates a large number of candidate areas for a target, and then uses region mapping and spatial pyramid pooling to extract features from those regions. Finally, CNN simultaneously identifies different characteristics of targets in many branches. By taking advantage of the characteristics of an RPN network, Faster R-CNN expedites the region generation process even more so than Fast R-CNN [20,21]. First, it uses a multi-task loss function to see if the candidate box has the right set of features for the detection goal. If so, it continues to the second process. By doing so, labelling the object of our detection can be done.

2.3 YOLO for Vehicle Detection (You Only Look Once)

In the beginning, YOLO [22] treated the task of object detection inside of a single neural network as a regression problem. The method's superior performance led to its swift elevation to the status of industry standard in the domain of object detection. The consistent growth of the YOLO architecture from its beginnings has yielded five generations: YOLO [23], YOLOv2 [24], YOLOv4 [25], and YOLOv5 [26]. All three steps—feature extraction, object localization, and classification—were rolled into one in YOLOv1, the initial version. When measured by Mean Average Precision (MAP), this network was SOTA despite its rapid detection speed. Convolutional layers followed by maxpool layers were the backbone of the original YOLO architecture. One of the main changes that the elimination of the complete layer enables the network to function regardless of picture resolution was present last in YOLOv1. The third generation, called YOLOv3, was developed by improving upon the work of the previous two generations. ResNet [27] and the FPN [28] were influential in the design of this new generation. Comparable Mean Average Precision may be attained on the COCO-2017 dataset with fast models like YOLOv3 [29], SSD [30], and Center Net [31]. All of these models can be used in combination with YOLOv3 to collect the mAPs. On the other hand, it works 17 times faster. To this end, the potential of employing both YOLOv3 and YOLOv5 as the foundation for our respective approaches was explored. Although YOLOv4 was found to be effective, we ultimately decided to adopt YOLOv5 instead due to its similar architecture and smaller model size. Studies that have used these architectures concerning tracking vehicles will be discussed in the next section. In this study, we examine MOT tracking systems that only require a single camera and can thereby detect several objects inside a single video frame. Any interested reader can find these methods in the authors' earlier papers. Since a single camera can only capture one side at a time, the success of this kind of tracking is dependent on the accuracy of the detection and the absence of occlusion [32]. To a certain extent, deep learning models that can detect objects even when they have partial occlusion have

improved with the identification challenge that these elements produced. Newer CNNs can make an accurate prediction of the object even if it is partially obscured by another, larger object. In the context of real-time MOT, numerous deep learning networks have been implemented, including Faster-RCNN [25], SSD, and YOLO. Also, the performance of YOLOv3 and YOLOv5 is compared in this study so that a real-time approach for tracking cars can be developed that uses multi-threading algorithms to process numerous video streams on a single GPU [33].

2.4 Vehicle Tracking

In this study, we examine MOT tracking systems that only require a single camera and can thereby detect several objects inside a single video frame. Any interested reader can find these methods in the authors' earlier papers. Since a single camera can only capture one side at a time, the success of this kind of tracking is dependent on the accuracy of the detection and the absence of occlusion. To a certain level, deep learning models that can detect objects even when they have partial occlusion have improved with the identification challenge that these elements created. The most up-to-date CNNs can still make a reliable prediction of the location of the target object even if it is hidden in some way by a larger object. Extensive work has been done to deploy deep learning networks in the context of real-time MOT, with examples being YOLO, SSD, and Faster-RCNN.

Traditional tracking methods work by first identifying items in the initial frames, and then searching the surrounding environment for features that correspond to those things to locate a picture sequence that best fits them. Traditional detectors, such as contour-based target tracking [34] and the Harris corner detection [35], produced false detection, SIFT (symmetric integral and fluctuating transform), and point-based approaches [36,37]. But, by first detecting the objects with DL models, and then proceeding to match features via the conventional tracking methodologies, better performance was attained. Regarding [38]'s methods for tracking via detection, we employ a tracking technique called Deep SORT in conjunction with low-confidence track filtering. As a result, fewer false positives were generated using the baseline Deep SORT algorithm. Tracking objects recognized by a YOLOv3 network has recently been presented by [39], who propose employing 3-D restricted multiple kernels. Kalman filtering allowed this to be accomplished. There has been a noticeable uptick in object tracking precision thanks to the invention of ever-more-complex tracking algorithms in recent years. However, running these algorithms requires a considerable amount of processing power. To keep tabs on the detection performed by YOLO-based DL networks over a wide variety of lanes of traffic in real-time, we offer a simple technique to object-centroid tracking in this research. Additionally, this work compares the performance of YOLOv3 and YOLOv5 to create a real-time system for tracking vehicles that can process several video streams on a single GPU by taking advantage of multi-threading strategies [40].

3. Proposed Methodology

The authors of this study recommend examining the vehicle recognition and tracking procedure employing deep learning technology. In this research, we employ primary-stage target identification techniques, such as the YOLOv5 algorithm and the Single Shot MultiBox Detector algorithm. Then, as the primary basis for vehicle detection, the Single Shot MultiBox Detector model architecture is used. This study's major purpose is to give an automated method to locate the following moving, parked automobiles in still images and video. There were three distinct steps involved in the recommended process. At first, N frames are taken at regular intervals for YOLOv5 to identify and detect automobiles. Then, the K-means clustering and the KLT tracker are used to track the corner points as

they move across the N-frames, clustering them into groups from which the objects' features may be collected and assessed. Finally, a reliable strategy is described for ascribing vehicle trajectories to each of the identified bounding boxes. This technique guarantees that the labels assigned to each vehicle trajectory are completely distinct from one another. In Figure 1 we see a schematic of the proposed solution architecture. In the sections that follow, we will present a high-level summary of the most salient features of the suggested technique for detecting and tracking vehicles.

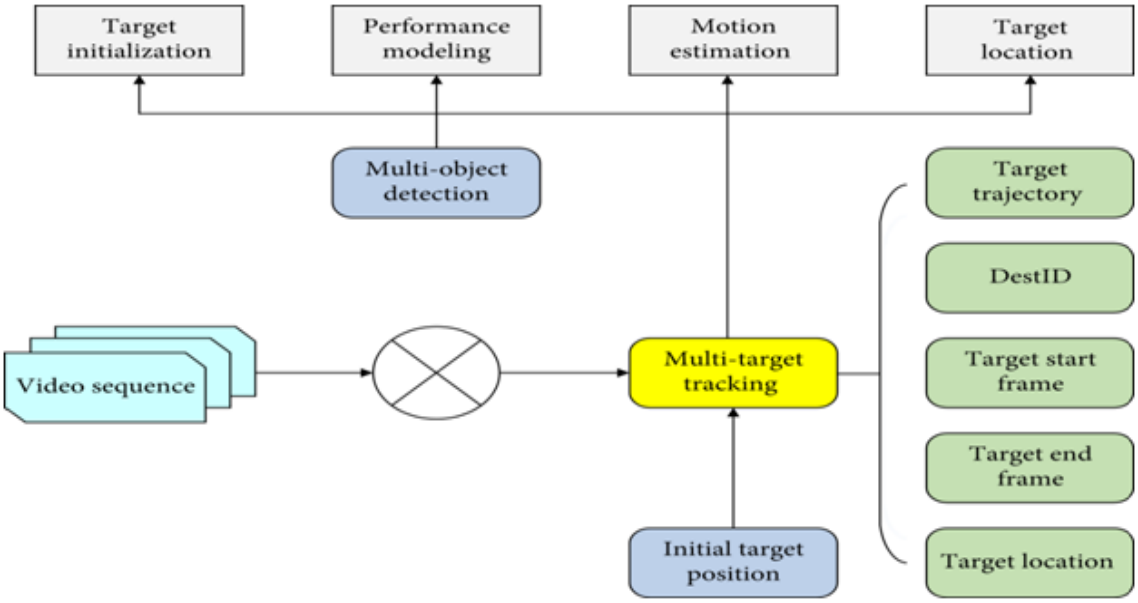

Fig. 1. Model architecture for vehicle identification and tracking

3.1 Vehicle Detection

To enhance the detection outcome, numerous deep learning-based strategies have been used recently. Such approaches include You Only Look Once v5 (YOLOv5) [30], Single Shot MultiBox Detector (SSD) [25], and several area suggestion methods [16,32]. In contrast, the authors of [19] obtained a promising result utilizing a background subtraction-based CNN and tracking data to help in detection. While using a CNN-based pixel classification system for detection, the waiting period needs to be lowered.YOLOv5 is a publicly available and free object detection system. Rapidly distinguishing features within a single image or video stream is one of its strongest features. With the Single-Shot Multi-box Detection (SSD) algorithm, it is possible to accurately identify items with only one forward pass of the computing feature map. It is feasible for it to work on video live streams, albeit accuracy may suffer. YOLOv5 and SSD are both top-tier systems; however, speed is an advantage for YOLOv5, while accuracy is an advantage for SSD. The YOLOv5 technique can be divided into three phases: the IoU phase, the residual blocks or gridding phase, and the bounding-box regression phase. In the first step, a grid called a residual block is used to map or divide the image. When compared to running CNN via loops for each item, this method checks each cell in a single forward pass. The complete picture can be covered in this way. An object can be recognized by the model if its centre of mass is located in a cell.

The model will yield a combined result matrix if a cell contains the centres of multiple object classes. Bounding-box regression can occur when there is an overlap between various bounding boxes. To check if the bounding boxes belong to the same objects, it looks at the object classes they include. IoU is used to determine if the boundaries of two object types intersect in this setting. Bounding boxes are deleted at a higher rate when there is an overlap of more than 50%. It will be the

spot in the middle of the average result. The bounding box does not remove based on the score; this phenomenon is known as non-max suppression. The SSD model employs deep learning to detect and pinpoint the positions of items. Like YOLOv5, it only requires a single forward pass to achieve full-picture object recognition. To put it simply, it is an effective method. In this respect, it differs from YOLOv5 because it uses bounding-box regression.

There have been a lot of talks lately about how YOLOv5 is one of the quickest CNN-based object identification systems [28]. Therefore, the work aims to look for the possibility of using YOLOv5 in the detection phase, alongside the tracking data to build a more successful technique of detection and counting. More than a million photos from the Image Net collection were utilized to train the deep learning approach used to create YOLOv5 [34]. Image Net contains 1.2 million images in total and is sorted into 1000 categories. Since auto-identification is our main focus, we use the YOLOv5 layers to the last fully connected one, where we reduce the number of classifications from a thousand to two.

In this research, we utilize the YOLOv5 architecture to speed up the detection process using transfer learning. Moreover, by incorporating the optical flow data into the counting strategy proposed in [14], we can make an accurate determination and further enhance detection performance. In the final layers, transfer learning is implemented by switching replace its SoftMax 1000 classes with the SoftMax 2 classes. In the convention training process, already-trained convolutional neural network models are used as a starting point for further training. To get to this level, these models required data. Our architecture's final, fully-connected layer is trained from the start using the vehicle dataset, whereas the preceding layers were developed using pre-trained models. With this completed layer, all of our efforts have paid off. A more in-depth explanation of the transfer learning method may be found in [27]. The Resnet-50 [21] was selected as the foundational model of the neural network for the YOLOv5 platform due to our decision to use transfer learning in this project. In Figure 2, a block diagram illustrating the YOLOv5 and SSD vehicle identification models may be seen. The inner workings of the models are depicted here in depth. Over a million images from the Image Net dataset were used to teach the convolutional neural network ResNet50 how to identify items in those images. There are almost 1.2 million photographs in this database, and each one has been tagged and organized into one of a thousand categories. After proposing a method for training YOLOv5, we performed three separate sessions using different collections of images to see how well it performed.

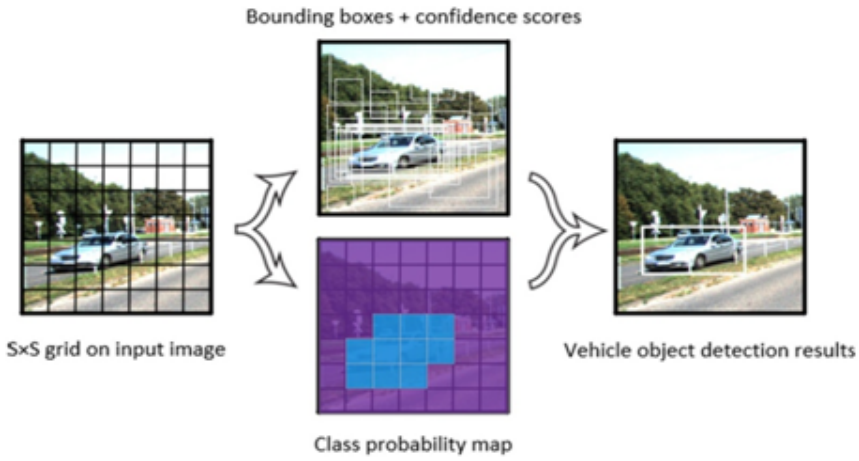

Fig. 2. Architecture of YOLOv5 and SSD vehicle Detection models

In the end, YOLOv5 was fine-tuned to an acceptable level of recall accuracy. Despite this, it is accurate in identifying some false positives, leading to poor accuracy overall. As will be shown in the

following sections, to eliminate these false positives, it will use K-means clustering and keep an eye on optical flow data. Therefore, it will be able to identify genuine dangers with greater precision. False-positive results acquired from the background regions show a substantial degree of diversity from one another when compared to the motion characteristics of the cars in the foreground. To ensure that certain feature points are not included in the final verdict, we employ the motion data that is linked to them. Clustering strategies are employed to organize the feature points into groups, and some are detailed in [12,36,37]. However, K-means clustering is sufficient for the needs of this task, as it yields a high degree of accuracy while keeping the computational complexity under control.

3.2 Vehicle Features Refinement and Clustering

During this stage, we not only cluster cars but also extract them from their surroundings by erasing those areas. Tracking with optical flow improves processing speed and feature-matching accuracy. Because of this, we use the optical flow Kanade-Lucas technique [2] to keep an eye on the feature points between frames f and f + 1. When two consecutive frames are combined, an optical flow vector set V is formed, with elements Vi = (Mi, θi), where S and are defined as follows:

$$\text{Mi} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

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

Notation ally, the preceding frame's X and Y coordinates are denoted by X1 and Y1, respectively, whereas the following frame's X and Y coordinates are denoted by X2 and Y2.Each value in V represents a single corner point Pi that was observed between frames f and f+1, and displacement magnitudes and angles are represented by the vectors M and, respectively. Trackers often fail because of noisy detections. This work only considers a foreground detection to be a vehicle object if it has been tracked for a sufficient number of consecutive frames, as noisy detections can only be followed for a limited period (9 consecutive frames). After that, the remaining vehicles in the foreground of the image are clustered using k-means clustering. Additional details might be found at [41].

3.3 Vehicle Counting

After the most reliable features were determined, they were separated into their groups for each vehicle. All of these facets of the vehicle are allocated their ID numbers and followed until they disappear from the video. The technique of allocation begins with the space defined by the combination of the rectangular bounding boxes derived from the prior tracking and the most recent detection. If the proportion of shared space between the two vehicles is more than a threshold, the two cars are considered to be identical matches. If there is either no junction area or an area smaller than one, a new label will be applied to the vehicle. The N-frames can be counted in four distinct ways, as shown below.

Since this is the very first time a vehicle of this type has been discovered, the features within its bounding box do not yet have names. These traits will then be given a new name, and the counter will go up by one.

The features of a vehicle will be labelled if and only regardless of the bounding box vehicle appears in both the first and previous frameset (N-frames). If there were any unmarked characteristics, they would be categorized as "unmarked" in this case.

Although present in the previous frameset, the vehicle's bounding box is absent from the initial frame. Since a name has already been given to these particular traits, we will continue to refer to them by that name in the future.

We shall call a vehicle a "missed counted vehicle" if we are unable to locate it anywhere in the full sequence of video frames.

4. Implementation of Vehicle Detection and Tracking using Yolov5

Detecting the vehicle target using the vehicle dataset presents several challenges, including but not limited to small-sized target objects, the complexity of the vehicle environment, there being too many goals in a single view in the collection, and targets overlap. Furthermore, the success of a vehicle detection system depends on meeting several criteria, making it imperative that the right detection method be used. While there are obstacles to overcome, there are also certain obvious features of automobiles. For example, almost all vehicle wheels are round. The arrangement of the vehicle's body lines is the most striking feature of any automobile. This paper will cover our plans to utilize a deep learning technology called YOLOv5 with SSD to perform vehicle recognition and tracking.

4.1 Analysis of Algorithm Selection

Target detection system speed and accuracy should be prioritized throughout algorithm development, as these factors are crucial in real-world use cases. Faster detection rates mean the technology can be used in more practical devices, and higher detection accuracies mean better item identification. Consider this Table 1 summarizes the findings of numerous popular target identification algorithms, comparing their detection rates and accuracy scores. It is clear from Table 1 that Faster R-CNN outperforms other one-stage detection networks in terms of accuracy rate, but its identification rate is significantly below theirs. This is because it requires a higher level of knowledge before being used.

Table 1
Comparing the detection rate and accuracy rate of several 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
video Frames processed Per Second(FPS)	21	49	43	54

The use of a progressive learning method using Faster R-CNN is responsible for this outcome. The detection rate is not bad for these one-stage detection networks, and it is clear that they outperform the second-stage detection method Faster R-CNN. YOLO's methods. In this paper, YOLOv5 is used as the vehicle detection method since it performs well in aspects of accuracy rate and detection rate,

which makes it especially well-suited for recognizing tiny objects. This decision was made because YOLOv5 is superior at identifying objects of smaller sizes, which is especially important in light of the issues just discussed

4.2 Designing the Model

4.2.1 Candidate box initialization operation

The most crucial phase in the detection phase is network training, however, the network parameters of the model must first be specified. The most significant task at the beginning of the season is the length of the applicant box. Because the network's training and testing should be dependent on the introduction of the first candidate box, every introduction of an initial selection box will impact the time needed to learn as well as the impact of learning. The applicant box size in YOLOv5 is determined by the K-means clustering method. An evaluation index, which is the value produced by using the Euclidean distance, is used to evaluate the efficacy of k-means clustering. The reason for this is that the distance between two points may be calculated using the Euclidean method. The k-means clustering approach begins by identifying a group of nodes to act as cluster axes. It is next necessary to determine the Euclidean distance between any more points or targets and the cluster centres identified thus far. By measuring the Calculate the distance between the target and the cluster's centre, we may decide which group the target belongs to. As soon as this is known, the centre of clustering can be moved to the spot inside the group where the target is located. Assuming that the initial run of the method went well, the next step is to repeat it a set number of times until the centre point stopped shifting. The goal is to get the targets to bunch up together.

K-means is used in YOLOv5 to group similar bounding boxes. Bounding boxes are an integral part of the training process, and every image used in the process has many of them. In the first stage of bounding box clustering, the entire collection of bounding boxes from the test set is retrieved as well as assembled into a single set. To do bounding box clustering, it is necessary to acquire the box's width and height. Because those components were originally recorded as coordinates for two spots on the box its upper left and right corners this transformation makes sense. To bound box clustering, it is crucial to obtain these values. This occurs because, in the clustering procedure, the width and height of the bounding box are used. In light preceding phrase, it becomes easy to see why. When the data has been analysed, the K values for each of the bounding boxes should be chosen as the anchor box clustering values, also known as the box beginning values. The value of K employed in that article is 9.

4.2.2 Detection module of network

Backboned by DarkNet-53, YOLOv5 can extract features from input data. In a useful way, the network's deep-level structure allows it to extract additional feature information. However, as depicted in Figure 3, there are also difficulties for the network at excessively deep levels. Because of this, the network's performance while trying to detect little objects will suffer if it is trained on a deep network. These problems have been lately reported by users of both the deep-level network and alternative one-stage detection systems. It is based on the concept of residual network and makes use of residual connections, DarkNet-53 can greatly improve the learning capacity to picture characteristics. As a bonus, it makes up for limitations in detecting extremely minor details.

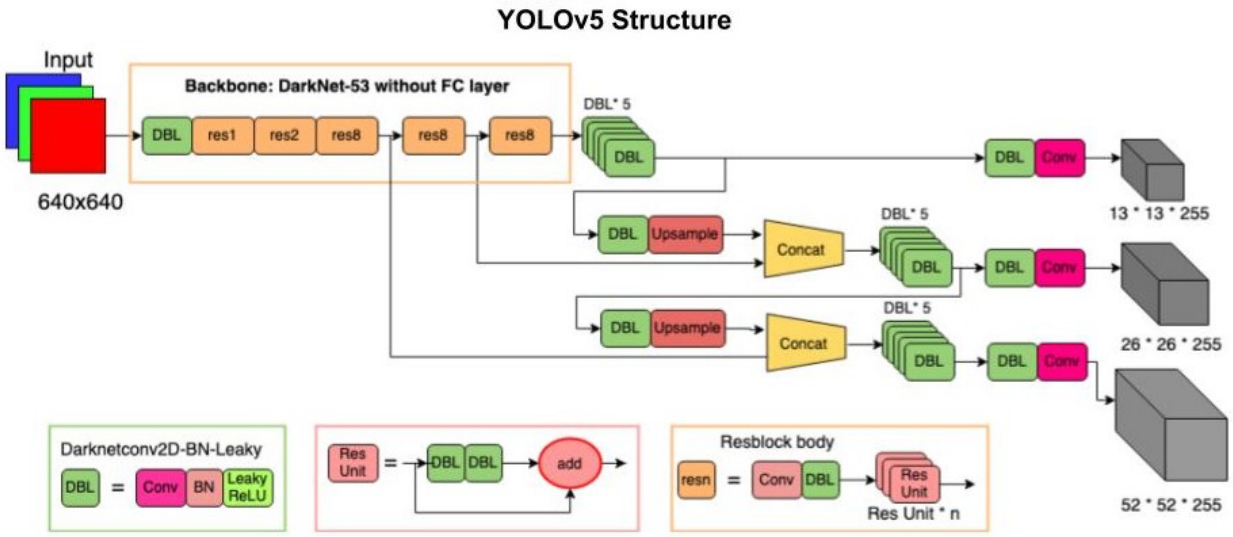

Fig. 3. Network architecture of YOLOv5 with a backbone of DarkNet-53

A loss function's design is essential for the effectiveness of any detection network. The loss function employed in training the model is ultimately what defines the model's quality. The target is deemed a positive sample for the loss function only when the IOU values among the projected frames and the actual frames created during training are maximum. If the anchor frame's IOU isn't at its maximum, the loss function shouldn't count the target as a loss. This means that for every real frame that has already been collected, there is only one linked prediction frame. We consider the collapse of not only beliefs but also logical categories and anchor frame coordinates when evaluating the degree of harm. When expressing confidence, it is common to practice inquiring as to whether or whether a particular cell in the detecting layer contains the target object's centre point.

5. Experimental Results and Analysis

This section evaluates and examines the vehicle dynamic detection and tracking method in terms of effectiveness and speed on publicly accessible datasets with current high CF trackers. Using high-performance CF trackers as a benchmark, this section will assess how well the proposed algorithm for vehicle detection and tracking performs in comparison.

5.1 Dataset Selection

The selection of a suitable dataset is an essential and difficult part of designing a system for target detection using the deep learning approach. For this research, we are creating a model and have decided to use the BDD100K image dataset as its training data. Sample data from the collection is shown in Figure 4.

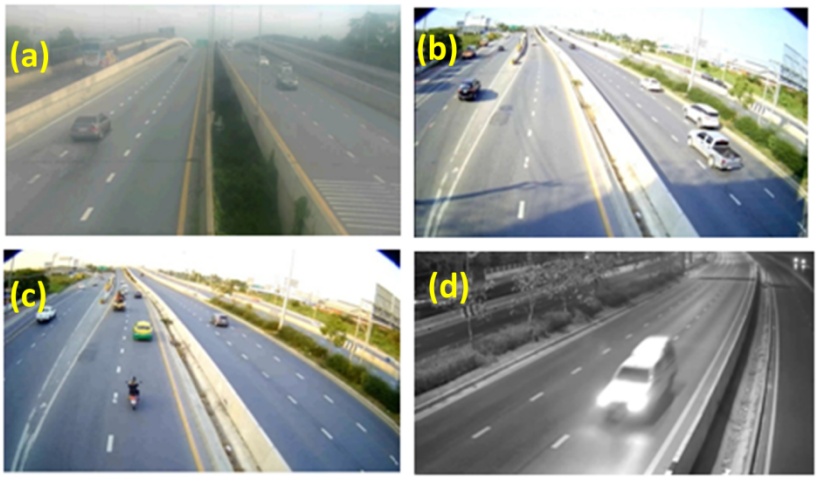

Fig. 4. Images of BDD100K data

All of the provided data photos were shot while riding in a moving car on a public road. A wide range of vehicles, as well as human and non-human targets, are depicted in these images. Out of a possible ten categories of target items, we only used six. Transport modes include cars, buses, people, trucks, engines, and bicycles.

5.2 Data Processing

The downloaded data consists of two sections, a "picture" section, and a "label" section. All of the newly collected data was indeed stored in JSON format, and each JSON file does indeed correspond to a picture with the same file name. This can be seen plainly in the namespace. It specifies the image file it is referring to, the category it falls under, and the name of that category. Location data specifies the exact location of the box down to the millimetre. The vertical as well as horizontal dimensions for the right-hand side are also supplied. These two landmarks can be used to determine the location of the box. Simultaneously, a single picture may feature many different containers. Files saved in JSON format can be cumbersome to use directly because of the extraneous information they contain. For this reason, a switch to this format is necessary to both streamline the file's data structure and eliminate any unnecessary information. It's essential to convert JSON files to XML before proceeding. The data in the converted JSON file is presented with a much more obvious structure, and the file as a whole is much smaller. Concurrently, the image's location and width/height measurements are saved accurately. The resulting XML file cannot be used as training input in its raw form; rather, it is analogous to the file produced by dedicated tagging software. However, training input data should be presented more straightforwardly, and unnecessary details should be removed from the file. Finally, only the image file's location, the box's relative position inside the image, and the box's contents type are stored. Last but not least, the information must be transformed into a text file ending in a .txt extension. In addition to the image's absolute location and name, this file should also specify the image's box's position inside the image and its category. In the final text file, we've incorporated all the data from the pictures we used for training. Pictures that need to be interacted with can be found by using the absolute path, and the contents of the images' input boxes come next. To get where you need to go, please follow this link. Images and data can be consolidated into a single text file for simpler sharing and uploading to the network.

5.2 Data Processing

The downloaded data consists of two sections, a "picture" section, and a "label" section. All of the newly collected data was indeed stored in JSON format, and each JSON file does indeed correspond to a picture with the same file name. This can be seen plainly in the namespace. It specifies the image file it is referring to, the category it falls under, and the name of that category. Location data specifies the exact location of the box down to the millimetre. The vertical as well as horizontal dimensions for the right-hand side are also supplied. These two landmarks can be used to determine the location of the box. Simultaneously, a single picture may feature many different containers. Files saved in JSON format can be cumbersome to use directly because of the extraneous information they contain. For this reason, a switch to this format is necessary to both streamline the file's data structure and eliminate any unnecessary information. It's essential to convert JSON files to XML before proceeding. The data in the converted JSON file is presented with a much more obvious structure, and the file as a whole is much smaller. Concurrently, the image's location and width/height measurements are saved accurately. The resulting XML file cannot be used as training input in its raw form; rather, it is analogous to the file produced by dedicated tagging software. However, training input data should be presented more straightforwardly, and unnecessary details should be removed from the file. Finally, only the image file's location, the box's relative position inside the image, and the box's contents type are stored. Last but not least, the information must be transformed into a text file ending in a .txt extension. In addition to the image's absolute location and name, this file should also specify the image's box's position inside the image and its category. In the final text file, we've incorporated all the data from the pictures we used for training. Pictures that need to be interacted with can be found by using the absolute path, and the contents of the images' input boxes come next. To get where you need to go, please follow this link. Images and data can be consolidated into a single text file for simpler sharing and uploading to the network.

5.3 Detection of Model

The final step in constructing a vehicle detection model is training the model, after which the trained weight may be loaded onto the model to perform the final detection task. In Table 2 we see examples of certain parameter sets that can be used in the training procedure. The most typical use of YOLOv5 is for target recognition across the three distinct levels of the backbone network, whether in training or detection. These are the foundation, medium, and upper levels of the backbone network.

Table 2
Parameter settings of the training process on various models

Models	Frame rate/ FPS	Memory size
+FRCNN	20	200.3
YOLOv3	30	243.6
SSD	35	15.7
YOLOv5	39	15.9
YOLOv5+SSD	39	31.6

5.4 Result Analysis of vehicle Detection and Tracking

Choosing a few parameters before training the detection network is essential for ensuring that the trained model is capable of having good performance. To assess the improved outcome after implementing the redesigned YOLOv5 algorithms, a performance comparison experiment using the same configuration environment and dataset with recognition techniques including R-CNN, YOLOv3, SSD, and YOLOv5 was conducted. This was done to ensure that the results we'd seen from our new and enhanced YOLOv5 algorithm were in fact due to it. NVIDIA's GTX1660Ti GPU is used to record the frame rate. The epoch can be retrained for a fixed number of times using the breakpoint continuation approach, assuming the model can be further optimized. If there is potential for improvement, this is a real possibility. After training is complete, the monetary worth of loss is shown in Figure 5.

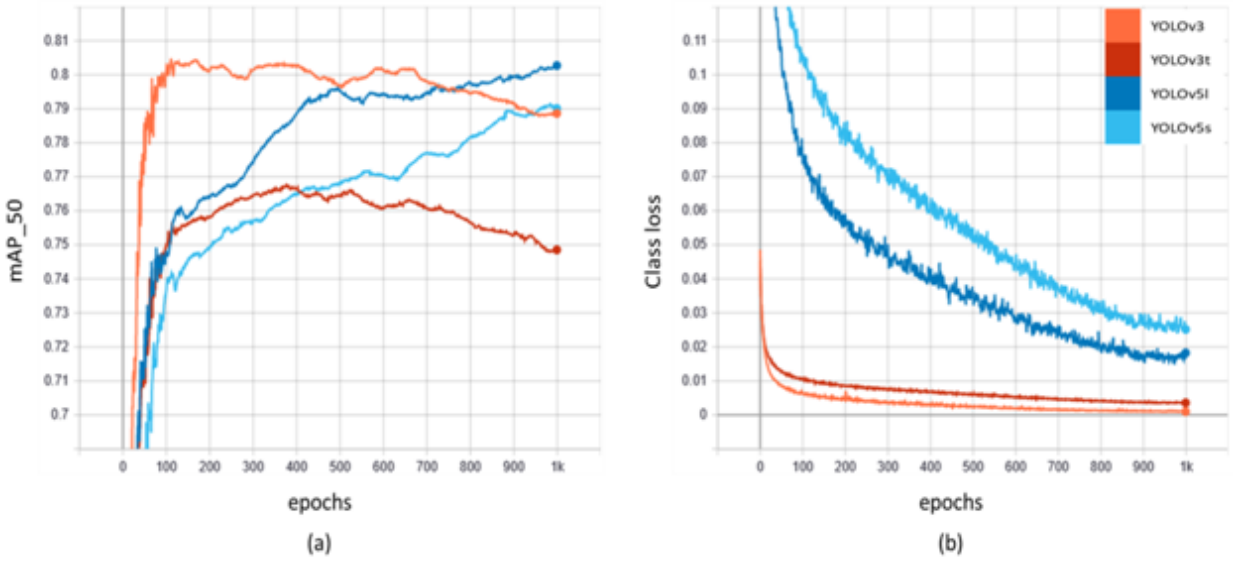

Fig. 5. (a) mAP at IoU = 50; (b) categorization loss, FRCNN, SSD, YOLOv3t, YOLOv5l, and YOLOv5s training outcomes on the BDD100K dataset

The orange curve in the illustration shows the test set's loss value, whereas the blue curve reflects the training set's loss value. The loss values are initially extremely large, but they soon begin to fall after a few training repetitions. The rate of fall eventually slows, and values begin to converge. Even though the loss value remains rather large after training, the analysis proves that the model is gradually nearing convergence. In general, as time goes on, the values of the training set loss decrease, then stabilize and finally oscillate within a narrow band. Even though the results won't be drastically different from the second part of Figure 6, training must be maintained by adjusting the learning rate in line with the breakpoint continuation approach.

To determine how effective a target detection model is, its value is measured by the map's quality. Each category's ap value represents the detection network's ability to identify that category. The MAP also displays the value of the model, 72.8, which was trained using this research. According to the data, its performance is below average. While cars continue to have a significant detection effect, buses and other motor vehicles now have significantly reduced detection performance. Figure 6 demonstrates that model is capable of accurately classifying the vast majority of vehicles, as indicated by the detection box scores, which are consistently maintained at or above 0.7. Trucks have

an average detection impact, bus types have a somewhat worse detection effect, and detection boxes have a usually low detection score.

Fig. 6. Detection of vehicles on the road from different camera views by day scenes and night scenes

Figure 7 shows the network's detection performance on two different categories: humans and bicycles. Figure 7 illustrates that the detection approach lacks severe influence on either the person or bicycle categories, but that the score for the human category is much higher than the score for the bicycle category, which is significantly lower. Another significant development is the improvement in "person" detection box accuracy and the expansion of available box options.

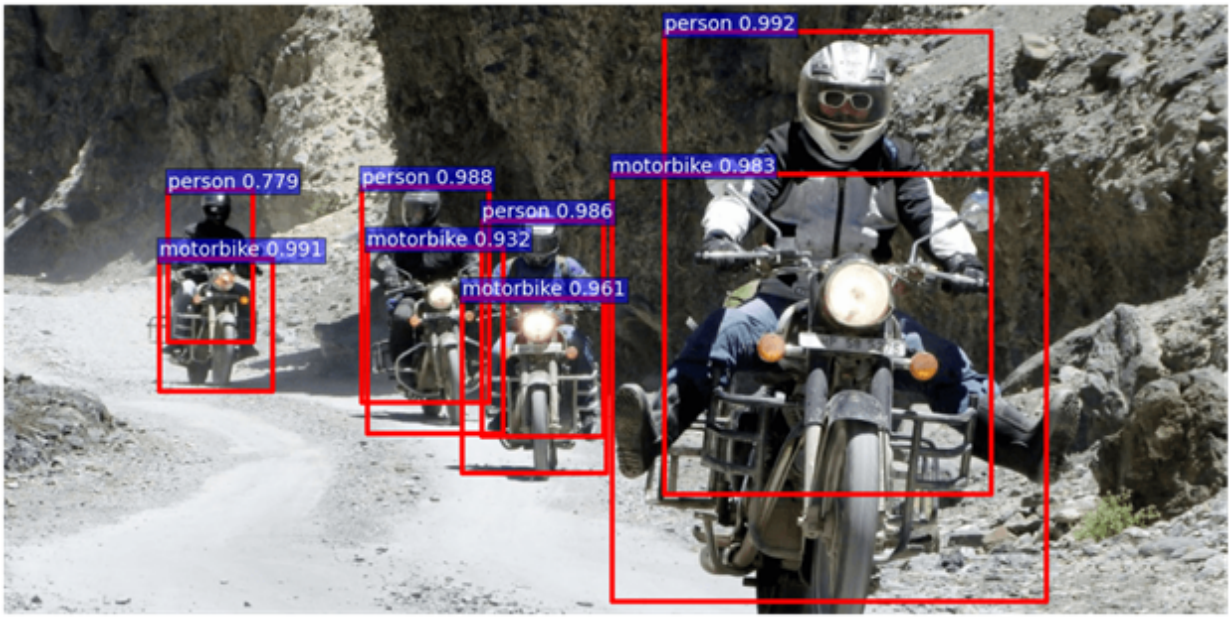

Fig. 7. Displaying the detection on the bike and human categories

Figure 8 of the detecting method's findings demonstrates that the detecting impact of the bus group is general. It's conceivable that this evidence will come to light throughout the inquiry. There are 3 buses displayed on the left, but only the first bus does have a greater score than all the others, therefore no categories can be assigned to any of them. Yet, as can be seen on the right, the container meant to stand in for the type of bus being used is surprisingly empty. A result of this is that the detection box does not include the entirety of the vehicle's body. The human category's detection effect is more effective than the motor categories, while the motor category's detection effect is just slightly worse. Overall, the model employed in this research achieves a respectable level of detection accuracy; the best detection effects are seen in the car and human categories. For both automobiles and people, this is true. While the vehicle category includes millions of targets, the bus category has more than 10,000, and the motor category has more than 4,000. Here are a few of the explanations for why.

Fig. 8. Vehicle detection for buses

Thus, the majority of the model's training data is related to the car category, the majority of the retrieved features are relevant to the vehicle classification, and the 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. As a result, even after all these years, vehicle classification detection is very accurate. Since there are already over 100,000 goals in the "person" category, it's safe to assume that a large number of people are taking advantage of this opportunity. Humans or the person category have seen a significant boost in their detection effect, which has been attributed in part to the fact that their feature information is quite different from that of the different categories of vehicles. The potential benefits of integrating deep learning-based techniques for improving the accuracy and efficiency of vehicle detection and tracking in image processing. The methodology involves leveraging neural networks, convolutional neural networks (CNNs), and other advanced algorithms to detect and track vehicles in real-time. The theme emphasizes the significance of this approach in various applications, such as traffic management, surveillance, and autonomous driving. It also underlines the need for continuous research and development to refine and enhance the performance of deep learning-based vehicle detection and tracking methodologies. The ultimate goal of this theme is to promote the adoption of advanced image processing techniques to achieve improved safety, efficiency, and performance in various industries.

5.5 Comparative Analysis

The proposed YoLov5+SSD model is evaluated based on performance metrics like Mean Average Precision (MAP), Accuracy, Recall, Specificity, andF1-Score. Table 3 shows that compared to Faster R-CNN, YOLOv3, SSD, and YOLOv5, the YOLOV5+SSD technique provides better performance in terms of MAP, Accuracy, Recall, Specificity, and F1-Score.

Table 3
Comparison and Tabular Interpretation of YOLOv5+SSD with other models in terms of MAP

Models	Frame rate/FPS	MAP	Accuracy	Recall	Specificity	F1-Score
FRCNN [20]	20	92.4	91.5	93.5	95.2	95.4
YOLOv3 [6]	30	85.7	86.2	94.8	94.8	93.2
SSD [30]	35	92.5	91.2	93.6	93.8	91.5
YOLOv5 [26]	39	94.3	93.5	93.8	93.6	95.5
YOLOv5+SSD (proposed)	39	97.6	94.8	95.9	95.5	96.0

Figure 9 shows the graphical interpretation of YOLOv5+SSD performance when compared to Faster R-CNN, YOLOv3, SSD, and YOLOv5 in terms of MAP.

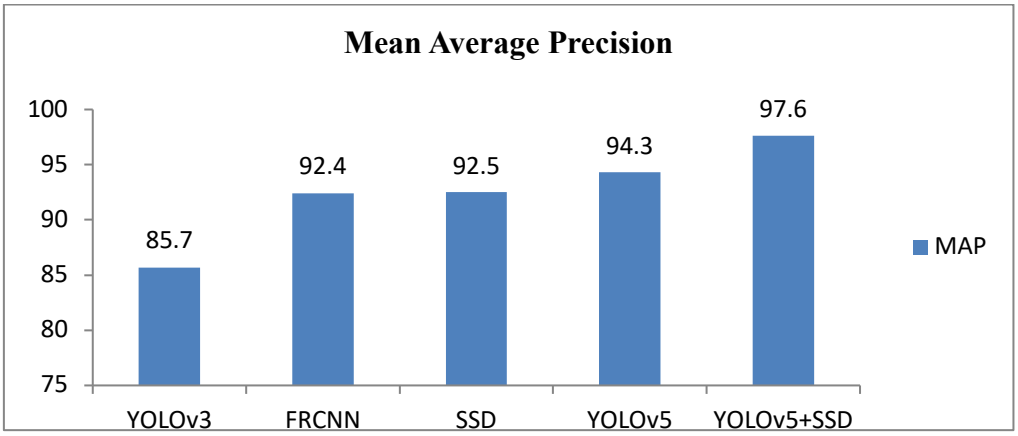

Fig. 9. Graphical Interpretation of YOLOv5+SSD with other models in terms of MAP

Figure 10 shows the graphical interpretation of YoLov5+SSD performance when compared to Faster R-CNN, YOLOv3, SSD, and YOLOv5 in terms of Accuracy, Recall, Specificity, and F1-Score. When comparing our YOLOv5+SSDwith other models, we find that the mAP value (mAP at 0.5: 0.05: 0.97) rises by 5.1%. Our proposed model's video surveillance is around 39 frames per second—1.12, 1.45, and 2.9 times faster than YOLOv5, YOLOv4, and Faster R-CNN, including both that it can match the demands of real-time identification. Although our improved method requires somewhat more memory than the YOLOv5 algorithm, it requires only about a seventeenth as much as YOLOv3 and around a thirteenth as much as Faster R-CNN. Altering the learning pace is accomplished via the breakpoint continuation technique. A high learning rate is used during initial model training, followed by a lower rate during optimization. A new epoch value of 50 has been implemented.

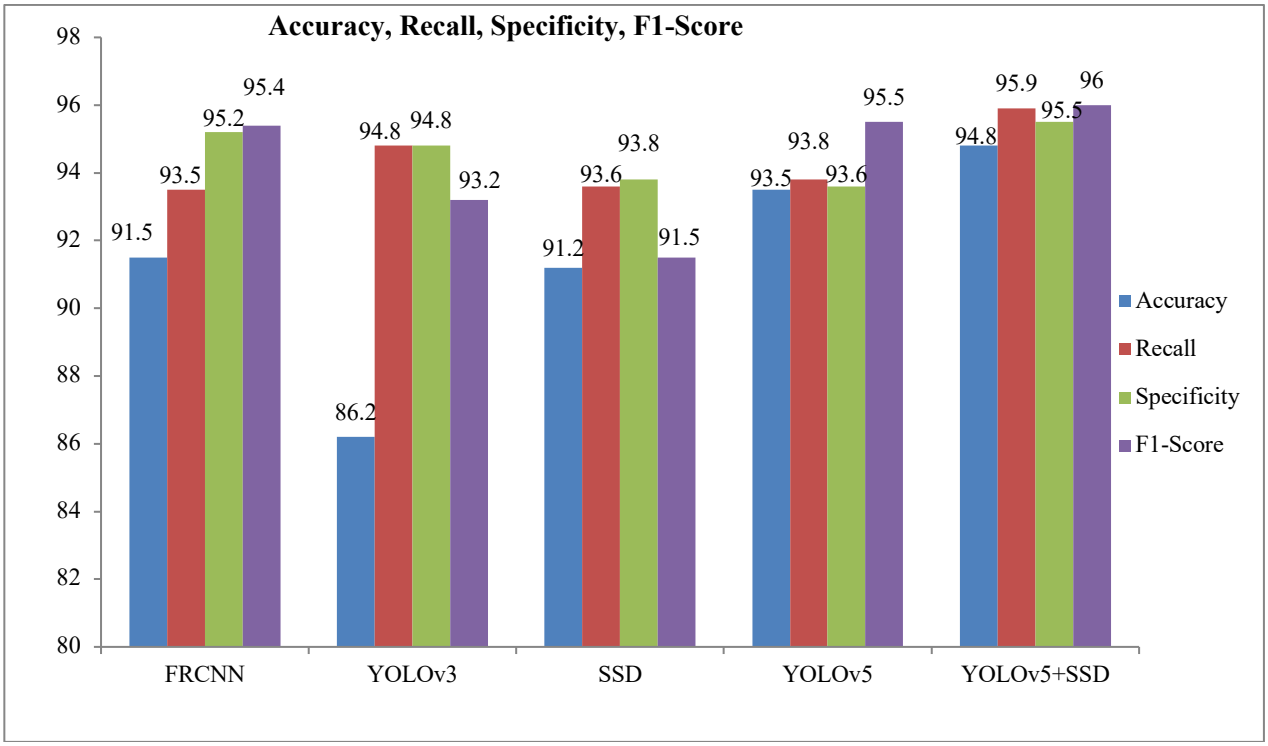

Fig. 10. Graphical Interpretation of YOLOv5+SSD with other models in terms of Accuracy, Recall, Specificity, and F1-Score

6. Conclusion

We implement learning to the problem of establishing tracks in this study. We also propose an improved version of the YOLOv5 and SSD algorithms for identifying persons and equipment on the track. Through extensive use of the YOLOv5 and SSD algorithms, we significantly improve the loss of classification and MAP at IoU for the various training results. With the newer YOLOv5 and SSD algorithms in place, convergence can be achieved more quickly, it is also possible to increase the detection rate of obstructed vehicle objects and tiny vehicle objects. Both of these advantages add up over time. A high level of robustness is shown by the experimental results for the newly created YOLOv5 and SSD method. Applying these algorithms allows us to effectively inspect the construction workers and tools, overcoming the problem of poor accuracy rate for complex picture concerns such as obstructed vehicle items and tiny vehicle objects, and eventually fulfilling all practical criteria for vehicle identification in the context of rail construction safety. This paper's results provide proof to the technical application of intelligent detection equipment and support in the in-depth study and

development of track safety vehicle detection technology. Based on the evaluated metrics, it was concluded that the YOLOv5 and SSD algorithm combination is the most effective in terms of both vehicle detection and tracking accuracy.

Acknowledgement
This research was not funded by any grant.

References

[1] Farid, Annam, Farhan Hussain, Khurram Khan, Mohsin Shahzad, Uzair Khan, and Zahid Mahmood. "A Fast and Accurate Real-Time Vehicle Detection Method Using Deep Learning for Unconstrained Environments." *Applied Sciences* 13, no. 5 (2023): 3059. https://doi.org/10.3390/app13053059

[2] Song, Heng, Junwu Zhu, and Yi Jiang. "Two-stage merging network for describing traffic scenes in intelligent vehicle driving system." *IEEE Transactions on Intelligent Transportation Systems* 23, no. 12 (2021): 25509-25520. https://doi.org/10.1109/TITS.2021.3083656

[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)

[4] Han, Xiaohong, Jun Chang, and Kaiyuan Wang. "Real-time object detection based on YOLO-v2 for tiny vehicle object." *Procedia Computer Science* 183 (2021): 61-72. https://doi.org/10.1016/j.procs.2021.02.031

[5] Rani, Esther. "LittleYOLO-SPP: A delicate real-time vehicle detection algorithm." *Optik* 225 (2021): 165818. https://doi.org/10.1016/j.ijleo.2020.165818

[6] Taheri Tajar, Alireza, Abbas Ramazani, and Muharram Mansoorizadeh. "A lightweight Tiny-YOLOv3 vehicle detection approach." *Journal of Real-Time Image Processing* 18, no. 6 (2021): 2389-2401. https://doi.org/10.1007/s11554-021-01131-w

[7] Mahmood, Zahid, Nargis Bibi, Muhammad Usman, Uzair Khan, and Nazeer Muhammad. "Mobile cloud based-framework for sports applications." *Multidimensional Systems and Signal Processing* 30 (2019): 1991-2019. https://doi.org/10.1007/s11045-019-00639-6

[8] Wang, Jiandong, Yahui Dong, Shuangrui Zhao, and Zhiwei Zhang. "A high-precision vehicle detection and tracking method based on the attention mechanism." *Sensors* 23, no. 2 (2023): 724. https://doi.org/10.3390/s23020724

[9] Chen, Yuhua, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. "Domain adaptive faster r-cnn for object detection in the wild." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3339-3348. 2018. https://doi.org/10.1109/CVPR.2018.00352

[10] Papakis, Ioannis, Abhijit Sarkar, Andrei Svetovidov, Jeffrey S. Hickman, and A. Lynn Abbott. "Convolutional neural network-based In-vehicle occupant detection and classification method using second strategic highway research program cabin images." *Transportation research record* 2675, no. 8 (2021): 443-457. https://doi.org/10.1177/0361198121998698

[11] Goedert, Guenter, Dietmar Jungen, B. E. C. K. Jorg, and Gianluca Favalli. "Weight-responsive vehicle seat occupant detection and classification method and system." U.S. Patent 10,377,266, issued August 13, 2019.

[12] Al-qaness, Mohammed AA, Aaqif Afzaal Abbasi, Hong Fan, Rehab Ali Ibrahim, Saeed H. Alsamhi, and Ammar Hawbani. "An improved YOLO-based road traffic monitoring system." *Computing* 103 (2021): 211-230. https://doi.org/10.1007/s00607-020-00869-8

[13] Yin, H., Bo Chen, C. Yi, and Y. D. Liu. "Overview of target detection and tracking based on vision." *Acta Automatica Sinica* 42, no. 10 (2016): 1466-1489.

[14] Qi, Meibin, Yan Pan, and Yin-xia Zhang. "Preceding moving vehicle detection based on shadow of chassis." *Journal of Electronic Measurement and Instrument* 26, no. 1 (2012): 54-59. https://doi.org/10.3724/SP.J.1187.2012.00054

[15] Humayun, Mamoona, Farzeen Ashfaq, Noor Zaman Jhanjhi, and Marwah Khalid Alsadun. "Traffic management: Multi-scale vehicle detection in varying weather conditions using yolov4 and spatial pyramid pooling network." *Electronics* 11, no. 17 (2022): 2748. https://doi.org/10.3390/electronics11172748

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

[17] Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra 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. 2014. https://doi.org/10.1109/CVPR.2014.81

[18] Wang, Ye, Weiwen Deng, Zhenyi Liu, and Jinsong Wang. "Deep learning-based vehicle detection with synthetic image data." *IET Intelligent Transport Systems* 13, no. 7 (2019): 1097-1105. https://doi.org/10.1049/iet-its.2018.5365

[19] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).
[20] Kai, Han, Zhang Hongying, Wang Yuan, and Xu Min. "A Vehicle Detection Method Based on Faster R-CNN." *Journal of Southwest University of Science and Technology (Natural Science Edition)* 32, no. 4 (2017): 65.
[21] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).
[22] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788. 2016. https://doi.org/10.1109/CVPR.2016.91
[23] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263-7271. 2017. https://doi.org/10.1109/CVPR.2017.690
[24] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).
[25] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." *arXiv preprint arXiv:2004.10934* (2020).
[26] Jocher, Glenn, K. Nishimura, T. Minerva, and R. Vilariño. "YOLOv5 https://github. com/ultralytics/yolov5." *Accessed March* 7 (2020): 2021.
[27] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016. https://doi.org/10.1109/CVPR.2016.90
[28] Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature pyramid networks for object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117-2125. 2017. https://doi.org/10.1109/CVPR.2017.106
[29] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).
[30] Ma, Rujin, Zhen Zhang, Yiqing Dong, and Yue Pan. "Deep learning based vehicle detection and classification methodology using strain sensors under bridge deck." *Sensors* 20, no. 18 (2020): 5051. https://doi.org/10.3390/s20185051
[31] Duan, Kaiwen, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. "Centernet: Keypoint triplets for object detection." In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6569-6578. 2019. https://doi.org/10.1109/ICCV.2019.00667
[32] Kutlimuratov, Alpamis, Jamshid Khamzaev, Temur Kuchkorov, Muhammad Shahid Anwar, and Ahyoung Choi. "Applying Enhanced Real-Time Monitoring and Counting Method for Effective Traffic Management in Tashkent." *Sensors* 23, no. 11 (2023): 5007. https://doi.org/10.3390/s23115007
[33] Neupane, Bipul, Teerayut Horanont, and Jagannath Aryal. "Real-time vehicle classification and tracking using a transfer learning-improved deep learning network." *Sensors* 22, no. 10 (2022): 3813. https://doi.org/10.3390/s22103813
[34] Abiyev, Rahib, and Murat Arslan. "Vehicle detection systems for intelligent driving using deep convolutional neural networks." *Discover Artificial Intelligence* 3, no. 1 (2023): 16. https://doi.org/10.1007/s44163-023-00062-8
[35] Henriques, João F., Rui Caseiro, Pedro Martins, and Jorge Batista. "High-speed tracking with kernelized correlation filters." *IEEE transactions on pattern analysis and machine intelligence* 37, no. 3 (2014): 583-596. https://doi.org/10.1109/TPAMI.2014.2345390
[36] Krump, Michael, and Peter Stütz. "Deep Learning Based Vehicle Detection on Real and Synthetic Aerial Images: Training Data Composition and Statistical Influence Analysis." *Sensors* 23, no. 7 (2023): 3769. https://doi.org/10.3390/s23073769
[37] Liu, Tao, and Yong Liu. "Deformable model-based vehicle tracking and recognition using 3-D constrained multiple-Kernels and Kalman filter." *IEEE Access* 9 (2021): 90346-90357. https://doi.org/10.1109/ACCESS.2021.3091871
[38] Berwo, Michael Abebe, Asad Khan, Yong Fang, Hamza Fahim, Shumaila Javaid, Jabar Mahmood, Zain Ul Abideen, and Syam MS. "Deep Learning Techniques for Vehicle Detection and Classification from Images/Videos: A Survey." *Sensors* 23, no. 10 (2023): 4832. https://doi.org/10.3390/s23104832
[39] Wende, Florian, Frank Cordes, and Thomas Steinke. "On improving the performance of multi-threaded CUDA applications with concurrent kernel execution by kernel reordering." In *2012 Symposium on Application Accelerators in High Performance Computing*, pp. 74-83. IEEE, 2012. https://doi.org/10.1109/SAAHPC.2012.12
[40] Gomaa, Ahmed, Moataz M. Abdelwahab, Mohammed Abo-Zahhad, Tsubasa Minematsu, and Rin-ichiro Taniguchi. "Robust vehicle detection and counting algorithm employing a convolution neural network and optical flow." *Sensors* 19, no. 20 (2019): 4588. https://doi.org/10.3390/s19204588