2023 Annual International Conference on Emerging Research Areas: International Conference on Intelligent Systems (AICERA/ICIS) | 979-8-3503-0345-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/AICERA/ICIS59538.2023.10420201

Enhancing Vehicle Analysis with Fine Estimation on Video Streams Using Image Processing Techniques

Radha R
Department of Computer Science and Engineering
Alliance University
Bangalore
radha.r@alliance.edu.in

Kotha Bhargavi Reddy
Department of Computer Science and Engineering
Alliance University
Bangalore
rebhargavibtech17@ced.alliance.edu.in

Yeshwanth R
Department of Computer Science and Engineering
Alliance University
Bangalore
ryeshwanthbtech16@ced.alliance.edu.in

***Abstract*—**Enhancing Vehicle Analysis with fine Estimation on video streams time object detection and tracking, and OCR for license plate recognition. By leveraging YOLO, the project extracts precise vehicle information from video streams, enabling comprehensive data collection for traffic analysis. Additionally, OCR algorithms automatically identify license plate numbers, facilitating law enforcement and fine estimation. This integrated approach enhances vehicle analysis, providing valuable insights for traffic optimization, infrastructure planning, and law enforcement strategies. The project aims to create safer, more efficient cities by harnessing the power of image processing techniques like YOLO and OCR in real-time vehicle analysis.

Keywords—vehicle analysis, image processing, object detection, YOLO, OCR

I. INTRODUCTION

Using precise tuning for vehicle analysis Utilizing computer vision and image processing methods to analyze video streams and compile comprehensive data about cars is known as estimation on video streams. Vehicle analysis is important in a number of areas, including traffic control, security, autonomous driving, and transportation planning. Traditional techniques for analyzing automobiles sometimes depend on simple object identification and tracking algorithms, which could not provide enough precision or information about the vehicles shown in the video. However, we can enhance the analysis of automobiles in video streams and get more accurate estimates by using cutting-edge image processing algorithms. These methods entail the use of advanced algorithms that can recognize and extract specific information from the video, enhancing analysis and improving knowledge of the traits and behaviors of the vehicles. The following are some of the crucial elements and methods:

A. Object Detection:

Object identification is the first and most important stage in improving vehicle analysis on video feeds. Deep learning-based techniques like YOLO (You Only Look Once)[1], Faster RCNN, and SSD (Single Shot MultiBox Detector) have recently gained popularity. These June 18, 2023 1 algorithmsare able to locate automobiles inside video frames and detect their presence. They are trained on large datasets that containlabelled vehicle photos in order to do this. They pick up distinguishing traits from this training that let them recognize automobiles with accuracy. Once the cars are located, they act as the starting point for additional analysis and estimate in the pipeline for vehicle analysis.

B. Vehicle Tracking:

The following phase includes using tracking algorithms to maintain track of each vehicle's identify across several frames when cars are properly detected in the video frames. The goal of these tracking algorithms[2] is to estimate the position, speed, and other time-related characteristics of each vehicle over time. Kalman filters, particle filters, and more contemporary deep learning-based trackers like Deep SORT (Simple Online and Real-time Tracking) are examples of well-known tracking algorithms. To precisely track vehicles, these algorithms use a few methods, including feature matching and motion modelling. They provide constant tracking and improve the analysis of vehicle behavior and motion patterns by creating linkages between cars in succeeding frames. As aresult, we can learn important things about how the moving and interacting cars in the video stream.

C. Fine Estimation:

Detailed information about automobiles may be extracted from video feeds using fine estimating techniques. These methods go beyond simple tracking and attempt to estimate further characteristics such vehicle size, direction, speed, acceleration, and even semantic data like vehicle kind and color. A mix of geometric and statistical methods is used to obtain precise estimates. For instance, by examining the perspective distortion brought on by the camera's viewpoint, one might estimate a vehicle's size. The vehicle's contour may be examined to estimate orientation, or more complex algorithms like the Hough Transform can be used. Analysis of the vehicle's motion over time can be used to estimate speed and acceleration. Deep learning techniques[3] may also beused to categorize different kinds of vehicles or recognize certain characteristics, including the identification of license plates or logos. These cutting-edge methods improve the analysis of cars in video streams and offer a more thorough comprehension of their traits and behaviors.

D. Speed Detection:

Using Haar cascades, it is possible to estimate the vehicle's speed while also detecting objects in vehicle analysis. Machine learning algorithms[4] such as Haar cascades may recognize certain items of interest in pictures or video frames. Real-time speed estimate is made possible using computationally effective Haar cascades for speed detection in vehicle analysis. Haar cascades can calculate the average speed values for cars in video streams by integrating object detection with feature tracking and distance computation. Although Haar cascades have certain limits when dealing with difficult situations, they provide a useful and approachable method for speed detection[4]. This method is used in several fields, including traffic control, law enforcement, and intelligent transportation systems. It permits the tracking and enforcement of speed limits, contributing to improved road safety and efficient traffic management.

E. License Plate Recognition:

The identification of license plates (LPR) is an essential component of vehicle analysis systems. It automatically extracts alphanumeric characters from license plates using optical character recognition (OCR) technology. Processes for vehicle identification and tracking are made effective by this extraction. We can automatically read the alphanumeric characters on license plates by using OCR techniques. This development makes vehicle recognition and tracking quicker and more precise. The accuracy and dependability of OCR-based license plate identification have substantially improved because to ongoing developments in deep learning and computer vision techniques. Applications in traffic management, law enforcement, and intelligent transportation systems are all made possible by integrating OCR with other vehicle analytic activities. It increases the effectiveness of jobs like locating stolen cars, enforcing parking laws, and observing traffic flow. OCR technology integration has transformed how we examine automobiles and helped create safer, more effective transportation networks.

II. Literature Survey

Utilizing image processing methods on video feeds, there has been notable advancement in improving vehicle analysis in recent years. This area of studyfocuses on employing image processing and computer vision algorithms to extract specific details about automobiles from movies. A careful analysis of the literature has revealed a number of significant developments. Vehicle recognition inside video frames has significantly improved thanks to the use of deep learning-based object detection techniques as Faster R-CNN, YOLO, and SSD.

These algorithms make use of sizable datasets that have been annotated and learn various properties that allow for precise and effective vehicle recognition. The necessity of choosing the right deep learning models and tailoring them to vehicle identification tasks is emphasized by researchers. Once a vehicle is found, tracking techniques are used to keep track of it across several frames. Popular tracking algorithms like Kalman filters, particle filters, and Deep SORT employ feature matching methods and motion models to produce precise tracking. Consistent tracking[7] is made possible by establishing correspondence between cars in successive frames, which also offers insights into driving behavior and motion patterns. Researchers have looked into a number of ways to increase the robustness of tracking systems[5], including adding temporal information and handling occlusions. For obtaining specific information about automobiles from videofeeds, fine estimation approaches are essential. In addition tobasic tracking, these algorithms try to estimate semantic information about the vehicle, such as its kind and color, as well as its size, direction, speed, and acceleration.

To obtain precise estimate, geometric and statistical techniques are frequently utilized. For instance, perspective distortion analysis takes into account the camera's perspective to precisely estimate the vehicle's dimensions. Vehicle orientation[5] may be estimated with the use of contour analysis techniques and sophisticated algorithms like the Hough Transform. By tracking a vehicle's motion over time, motion analysis can estimate speed and acceleration. Deep learning techniques have also been investigated for tasks that are important for thorough vehicle analysis, such as identifying different types of vehicles, finding license plates, and recognizing logos.

Recent research has concentrated on applying deep learning algorithms and object identification approaches to estimate vehicle speed precisely in video streams. Speed detection is a crucial part of vehicle analysis. When used in conjunction with tracking algorithms, methods like YOLO allow for accurate speed estimate. Accurate speed prediction requires thorough training of deep learning models on a variety of datasets and reliable vehicle tracking. In order to increase accuracy and resilience, researchers have tackled problems with speed detection, such as how to handle various road and lighting situations as well as occlusions. Vehicle analysis systems must include license plate recognition (LPR) employing optical character recognition (OCR)[6].

Alphanumeric characters are automatically extracted from license plates using OCR algorithms, enabling accurate vehicle identification and tracking. To increase accuracy and dependability, researchers have looked at a variety of OCR methods, including conventional computer vision-based techniques and deep learning-based strategies. Applications in traffic management[8], law enforcement, and intelligent transportation systems have been made possible by integrating OCR with other vehicle analytic tasks, such object identification and tracking. The literature review concludes by highlighting important developments in improving vehicle analysis with precise estimates on video streams using image processing methods. There has been significant advancement in object identification, tracking algorithms, precise estimate methods, deep learning-based speed detection, and OCR-based license plate recognition.

The accuracy, effectiveness, and usefulness of vehicle analysis systems have considerably increased as a result of these developments, which has positive effects on several fields like traffic monitoring, surveillance, autonomous driving, and transportation management.

III. Proposed Method

The project uses image processing techniques to automate and enhance the analysis of automobiles in video feeds. Vehicle detection, license plate identification, speed estimation, violation rule formulation, and fine computation are important components. These methods allow the system to

recognise cars with accuracy, retrieve license plate data, gauge vehicle speed, identify breaches based on specified criteria, and calculate fine amounts. The combination of image processing and fine calculation improves traffic monitoring's overall effectiveness and accuracy, enabling automated enforcement of traffic laws and resulting in safer roadways.

A. Analysis for Speed Detection:

It is essential to complete tasks like accurate vehicle tracking and speed estimate since they are important in many different fields. One use for these approaches is in traffic monitoring. Authorities can locate places that need infrastructure upgrades or adjustments to traffic management techniques by analyzing traffic patterns and congestion hotspots. As a result, the efficiency of transportation systems is increased overall while traffic flow is optimized, and congestion is decreased. In order to enforce speed restrictions and maintain road safety, law enforcement authorities can also make use of vehicle tracking and speed estimation.

The right steps may be taken to prevent accidents and uphold law and order on the roadways by identifying drivers who violate the limits. The offered code can also be used by academics and analysts in the domains of computer vision and object tracking. It lays the groundwork for more research, algorithm development, and performance assessment, advancing the discipline. Despite the importance of tracking vehicles and estimating their speeds, accurate findings require overcoming several obstacles[9]. The accuracy of automobile recognition using the Haar cascade classifier is one of the main challenges. This classifier's performance is strongly influenced by variables including illumination, occlusions, and the calibrated classifier itself. For the trackingand speed calculation steps that follow, it is essential to have an accurate and reliable automobile detection system. Handling occlusion and object interactions in complicated traffic circumstances presents a substantial additional difficulty.

It is crucial to have strong algorithms that can precisely differentiate and monitor individual vehicles since automobiles regularly collide with one another or interact intimately. For precise and dependable vehicle tracking and speed estimate, these obstacles must be overcome. The offered code includes object tracking and speed estimate methods to handle the issues. The dlib correlation tracker is used to track objects because it uses a correlation filter to estimate the monitored object's position and updates the tracking depending on the tracked object's visual appearance. The effectiveness and durability of this tracker in handling object motion and appearance changes are well known. On the other hand, speed estimate is accomplished by measuring the distance travelled by a vehicle between two successive sites and dividing that distance by the passing time.

The pixel distances are transformed into real-world distances using an assumption of a constant frame rate and a pixels-per-meter (ppm) conversion factor, giving a rudimentary estimate of the vehicle's speed. Considerations should be made to increase the precision and dependability of vehicle tracking and speed estimate. The most important stage in ensuring accurate speed measurement in practical terms is camera calibration.

The accuracy of the measurements may be greatly increased by calculating the intrinsic and extrinsic camera characteristics and correcting lens distortion. Additionally, elements like frame rate and video quality directly affect how accurately speed is estimated. Better video quality and higher frame rates improve vision and tracking, which produces more precise results. As a result, thorough evaluation of these aspects helps to increase the tracking and speed estimate process' accuracy and dependability.

Several optimization techniques[10] can be used to improve the performance of the supplied code. Object recognition and tracking can be sped up by using parallel processing methods like multi-threading or parallel computing libraries. Performance can be enhanced by enabling concurrent task execution, which effectively distributes the computational burden over numerous threads or processes.

Another essential factor that has a big influence on performance is algorithm optimization. The total computational complexity can be decreased, resulting in quicker execution times, by removing superfluous computations, optimizing data structures, and using effective algorithms. A significant speed improvement may also be obtained through hardware acceleration approaches, such as the use of graphics processing units (GPUs) or the use of optimized libraries, such as OpenCV with CUDA support. These methods use the GPUs' parallel processing capability to speed up computationally demanding activities. The computational burden can be decreased while maintaining acceptable levels of accuracy by using data reduction strategies like frame skipping or extra filtering algorithms. Real-time or almost real-time vehicle tracking, and speed estimates are made possible by the combined effects of these optimization tactics, which increase performance and speed up processing times.

B. Analysis for License Plate Detection:

To automatically identify cars and their related data, license plate recognition is a critical part of the entire vehicle analysis system. It is essential for upholding traffic laws, tracking cars, and gathering data for law enforcement. There are several issues with number plate identification that need to be resolved. The variety of license plate designs is one of the difficulties. Font sizes, styles, and background patterns vary on license plates. To enable reliable recognition, the system must be able to handle these design differences successfully. Complex backdrops and difficult lighting in the collected video streams provide another difficulty. License plates may be photographed in low light, with shadows, or with reflections.

To accommodate these variances and yet achieve correct recognition, the system must be strong enough. The identification of license plates might also be complicated by noise and distortion. As a result of motion blur, camera shaking, or artefacts from video compression, license plates in video streams may be noisy or distorted. These difficulties should not prevent the system from achieving correct recognition. In video feeds, license plates are frequently obscured or only partially visible. Parts of automobiles or other items, such as license plates, may partially obscure license plates. Such occlusions should not be a problem for the system, and it should be able to identify partially viewable plates. To accommodate the constant influx of frames, real-

time processing is sometimes necessary for license plate identification in video streams.

To accomplish timely recognition, the algorithms and optimizations must be effective. This will allow the system to keep up with incoming frames and deliver results in real-time or very close to real-time. Several strategies for license plate identification can be used to address these issues. Using the bounding box coordinates discovered during the vehicle identification stage, the first step is extracting the area of interest (ROI) containing the license plate. This enables the recognition process to be concentrated on the important region. The clarity and quality of the license plate region can be increased by using pre-processing techniques such noise reduction, picture enhancement, and normalization. These methods boost image quality and reduce noise, which increases the identification accuracy. Algorithms and libraries for optical character recognition (OCR) can be used to decipher the license plate's alphanumeric characters. OCR methods examine the image and decipher the characters it contains.

Character segmentation algorithms can be used for license plates with several characters to enable individual recognition. Techniques like linked component analysis or contour detection may be used in this. In some circumstances, algorithms for template matching or pattern recognition can be employed to compare the recognized characters to a list of recognized characters or license plate templates. This lessens false positives and increases identification accuracy.

It is crucial to compile a broad and representative dataset of license plate photos, including a range of designs, backdrops, and lighting situations, to optimize the license plate identification module. The license plate recognition algorithm may be trained and tested using this dataset. Convolutional Neural Networks (CNNs), a type of deep learning technique, have demonstrated promising performance in picture classification tasks and may also be used to the recognition of license plates. The accuracy and robustness of the license plate recognition system can be further improved by investigating the usage of CNNs or other deep learning models. Integrating the system is essential. The vehicle analysis system's other modules, such those for vehicle identification and speed calculation, should effortlessly connect with the module for license plate recognition.

To accomplish accurate and effective analysis, smooth data flow and synchronization between various modules should be guaranteed. Finally, proper assessment measures should be devised to judge the reliability, durability, and effectiveness of the algorithm for recognizing license plates. It is important to test the algorithm's effectiveness in difficult situations, such as those with diverse license plate designs, poor illumination, and obstructions. The total vehicle analysis system can properly extract license plate information and enhance traffic monitoring, violation detection, and fine estimate by resolving the issues and putting in place efficient license plate recognition systems.

C. Analysis for Fine Estimation:

The vehicle analysis system's crucial component of fine estimation establishes the penalty or fine amount for traffic offences, particularly those involving speeding. By combining image processing methods and violation criteria, the procedure is intended to be automated. Several factors and procedures must be considered to attain precise precision estimation:

1. Rule-breaking guidelines: Based on regional traffic laws, the project should provide the parameters for violations. These regulations specify the circumstances under which a vehicle is deemed to be speeding. Speed restrictions, the type of roads used, and the severity of the violations are some examples of violation regulations.

2. Estimating the speed of the cars in the video streams requires the use of image processing algorithms. To examine the movement of vehicles between two frames, techniques like motion tracking, optical flow, or frame differencing can be used. The speed of the car may be measured by adding up the distance travelled and the passing time.

3. Fine calculation logic: The logic for calculating fines considers both the regulations for infringement and the predicted speed of the vehicle. The rationale should consider the seriousness of the infraction, the relevant infraction rule, and any other elements included in the local traffic laws. The penalties can be determined using a predetermined formula or lookup table, accounting for the seriousness of the offence and the applicable penalty structure.

4. Modification of violation rules: The system should provide modification of violation rules to consider various jurisdictions or traffic laws. This personalization guarantees the system's adaptation and flexibility to regional needs and makes it possible for it to be used in different places.

5. System integration: The fine estimation module should work in unison with the other elements of the system for analyzing vehicles, such as the ones for detecting vehicles, reading license plates, and estimating speed. The output from these modules is used in the fine estimate procedure to precisely calculate the fine amount. Data synchronization[11] and seamless data flow between the various components are made possible via integration.

6. Results visualization: To clearly convey the infractions and associated fine amounts, the fine estimation results should be shown graphically. This might comprise showing recognized license plate numbers, estimated speeds, and computed fine amounts in addition to annotating the video frames with bounding boxes around identified cars. When providing information to traffic enforcement officers or creating proof for judicial proceedings, visualization is helpful.

7. Accuracy and dependability: The module for estimating fine quantities must be both accurate and trustworthy. Variations in vehicle speeds, various circumstances for violations, and potential causes of speed estimation mistake should all be taken into consideration. To guarantee the precision and dependability of the fine estimate process, solid algorithms and extensive testing should be carried out.

8. Updates and maintenance of the rules: The system should include mechanisms for updating and maintaining the violation rules in response to modifications to the law or changes to traffic regulations. This guarantees that the fine estimation algorithm is current and in accordance with the most recent traffic legislation. The fine estimation module improves the vehicle analysis system's capacity to

automatically calculate fine amounts for traffic offences by combining image processing algorithms, violation criteria, and precise speed estimation. This promotes safer road conditions by assisting in effective and efficient traffic monitoring and enforcement. We recommend the method listed below for the proposed system based on the analysis presented above.

IV. Design and Implementation

A. Algorithm for the proposed system:

1. Configure environment and dependencies, including OCR libraries: - Ensure that the programming environment is ready, including the installation of the essential libraries and dependencies for using the algorithm. - Install OCR libraries, such as Tesseract's or OpenCV's text recognition modules, that will be used to recognize license plates.

2. Obtain a YOLO model that has already been trained to recognize objects: - A well-known deep learning model for object recognition is called YOLO (You Only Look Once). - Obtain a YOLO model that has already been trained using a big dataset of different objects, including cars.

3. Load the pre-trained Yolo model into the deep learning framework: - Load the Yolo model into a deep learning framework like TensorFlow or PyTorch. This enables you to execute object detection on the input video frames using the architecture and weights of the model[12]. - Decide whether the algorithm will handle a video file or a live stream from a camera.

4. Prepare video input from file or live stream. - Set up the required video input pipeline so that it can read and handle each frame one at a time.

5. Analyze every video frame: - Repeat over all of the video input's frames. - Complete the following operations for each frame.

6. Apply YOLO to identify vehicles: - To detect objects, apply the loaded YOLO model to the current frame. - Pay special attention to locating automobiles in the picture. - For each item that is recognized, YOLO will offer bounding box coordinates and class names.

7. Extract the license plate region from each vehicle. - - Use the identified bounding box coordinates to extract the region of interest (ROI) containing the license plate for each vehicle. - The objective of this phase is to isolate the region of the picture containing the license plate.

8. Use OCR to identify license plate characters: - Use OCR methods to identify the characters contained on the extracted license plate region. - To process the image and extract the alphanumeric characters, OCR libraries and algorithms are used.

9. Calculate vehicle speeds using speed tracking algorithms: - Calculate each vehicle's speed using speed tracking techniques. - To follow a vehicle's movement between frames and determine its speed, this may utilize techniques like motion tracking or frame differencing.

10. Establish a set of violation criteria that specify when a vehicle is judged to be speeding or in violation of traffic laws, and then develop a logic for fee computation based on those rules. Local traffic laws, such as the top speeds allowed on various kinds of roads, may serve as the basis for these regulations.

11. Determine the fine amount depending on the law and the vehicle's speed: - Determine if a violation has occurred by applying the violation rules to the projected vehicle speeds.

12. Make findings visually appealing by using boundary boxes, license plate numbers, speed, and fine amounts: - Lastly, display the findings for each frame by: - Outlining the bounding boxes of any recognized cars. - Using the recognized license plate numbers as a display. - Displaying the guessed speeds of the vehicles. - Presenting the fine amounts that were computed. - An annotated movie or graphical user interface (GUI) that shows the processed frames and pertinent data might serve as this visualization's format

4.1. Formulation of solution, fabrication, and construction

1. Start: The project's goals and needs are established at the outset together with the process's initialization. This involves determining the necessity of enhancing the precision of vehicle analysis and fine estimate on video feeds. The project team thoroughly assesses the current difficulties in vehicle analysis and pinpoints the crucial areas that need improvement. Studying the constraints of present systems, comprehending the precise specifications of the intended application, and considering the available resources and technologies are all necessary steps in this process.
2. Custom Video Dataset: An original video dataset is produced to train and assess the solution. The dataset is essential for creating reliable and accurate vehicle analysis models. The project team chooses video streams carefully from a variety of sources, including

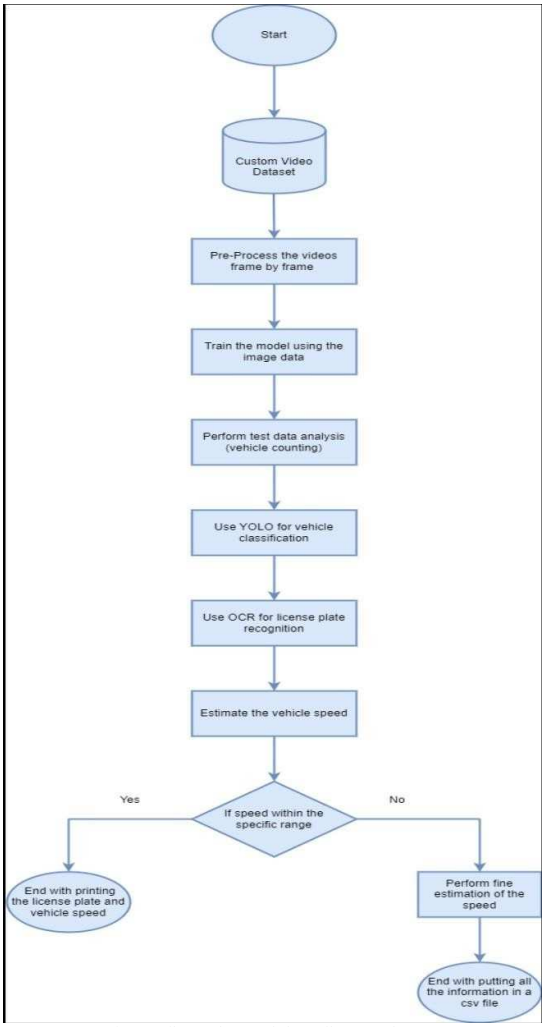

Fig. 1: flow chart of data flow and process

traffic cameras, surveillance cameras, and other pertinent sources. The dataset is intended to include a variety of scenarios, such as various weather conditions, varied lighting conditions, and various vehicle kinds and sizes. The team makes ensuring that there are enough labelled samples in the collection for the purposes of identifying vehicles, categorizing them, identifying license plates, and estimating speeds. The ground truth data, such as bounding boxes around cars and license plate numbers, are marked using an annotation procedure so that it may be used as training and assessment reference data.

3. Pre-process the Videos Frame by Frame: To get the data ready for analysis, pre-processing is applied to every video in the collection. To handle the different issues that frequently arise in video streams, pre-processing operations are carried out frame by frame. These duties consist of:

a) Frame Stabilization: Methods like picture stabilization are used to lessen the impact of camera motion and vibrations. As a result, noise is reduced and accuracy is increased during the subsequent analysis, which is done on stabilized frames.

b) Noise Reduction: To improve image quality and clarity, noise reduction techniques are used. Filters or algorithms are employed to lessen noise artefacts, such as sensor noise or compression artefacts, which may obstruct further analytic operations.

c) Image Enhancement: To increase visibility and improve the quality of the video frames, image enhancement techniques are used. To improve the details and make the photos more suited for precise vehicle analysis, techniques like contrast correction, histogram equalization, or adaptive filtering are used. The project team is working to improve the quality and consistency of the video frames through these pre-processing operations to increase the precision of the future analysis stages.

4. Train the Model using the Image Data: To train a model for vehicle analysis, pre-processed video frames are utilized. Convolutional neural networks (CNNs) are frequently used in deep learning approaches because of their capacity to recognize and extract valuable characteristics from pictures. As the foundational model for vehicle identification and tracking, the project team chooses an appropriate CNN architecture, such as Faster R-CNN, SSD (Single Shot MultiBox Detector), or YOLO. Using the previously constructed bespoke video dataset, the basic model is then adjusted. The model gains the ability to identify cars, group them into distinct categories, and precisely localize them inside the frames throughout training. To improve the performance of the model, the team uses the proper loss functions[13], optimization methods, and training techniques. Iterative tuning of the model's hyperparameters results in the required level of accuracy and generalization.
5. Perform Test Data Analysis (Vehicle Counting): After the model has been trained, test data analysis is carried out on several video streams to gauge its effectiveness. Vehicles are recognized and tallied using the learned model as it is applied to the video frames. This process enables evaluation of the model's precision and efficiency in properly classifying and counting automobiles. The project team assesses the model's accuracy, recall, and general performance by comparing the detected vehicle count with the ground truth annotations. To statistically evaluate the model's efficiency in counting vehicles, statistical metrics including accuracy, recall, and F1 score are computed. The findings are also visually examined for qualitative evaluation, and any differences between the model's predictions and the actual data are examined. The review process enables the group to see any flaws or areas for further improvement.
6. Use YOLO for Vehicle Classification: The project includes vehicle categorization utilizing the YOLO (You Only Look Once) algorithm or comparable methods in addition to vehicle detection and counting. To categorize identified

vehicles into distinct groups, such as automobiles, trucks, motorbikes, etc., the trained model is employed. This categorization information adds to our understanding of the make-up of the fleet of vehicles. The project team makes it possible for the system to properly distinguish between various vehicle kinds by using the properties discovered during the model training phase. The categorization findings can be used to identify certain vehicle types of interest, compile statistical data regarding the distribution of vehicles, or facilitate more focused study based on vehicle classifications.

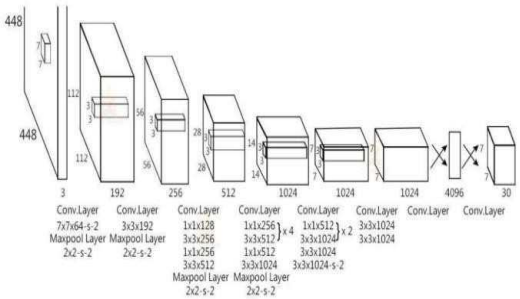

Fig. 2: YOLO architecture [Source: towardsdatascience.com]

7. Use OCR for License Plate Recognition:

The project includes vehicle categorization utilizing the YOLO (You Only Look Once) algorithm or comparable methods in addition to vehicle detection and counting. To categorize identified vehicles into distinct groups, such as automobiles, trucks, motorbikes, etc., the trained model is employed. This categorization information adds to our understanding of the make-up of the fleet of vehicles. The project team makes it possible for the system to properly distinguish between various vehicle kinds by using the properties discovered during the model training phase. The categorization findings can be used to identify certain vehicle types of interest, compile statistical data regarding the distribution of vehicles, or facilitate more focused study based on vehicle classifications.

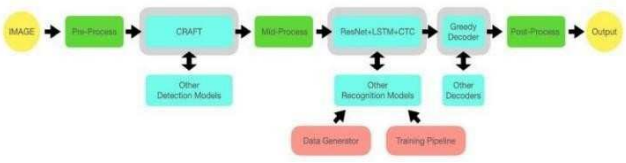

Fig. 3: EasyOCR Framework [Source: analyticsvidhya.com]

8. Estimate the Vehicle Speed:

An important step in the vehicle analysis process is determining the speed of observed cars. The research uses methods for image processing to combine speed estimating approaches. The process for estimating a vehicle's speed is outlined in the stages below: The time interval between each frame in the video stream is calculated using the following method:

a) Calculate Time Interval. The timestamps connected to the frames or the video's frame rate can be used to determine this interval.

b) Carry out motion analysis: The displacement of each vehicle may be calculated by tracking the moving objects across several frames. The information required to estimate speed is provided by the change in location over time. The project team uses methods for tracking motion analysis, such as feature tracking or optical flow to track the vehicles accurately.

c) Speed Calculation: The speed of each vehicle is determined using the displacement and the associated time period. You may calculate the speed in units like kilometers per hour or miles per hour by dividing the displacement by the time period.

d) The computed speed is then contrasted with a predetermined range of allowable speeds. The project concludes by publishing the number plate number and the accompanying vehicle speed if the speed is within this range. This demonstrates that the car is travelling at the desired pace.

e) Carry out Speed Estimation: Additional analysis is carried out to estimate the speed precisely if the determined speed is beyond the defined range. To increase the precision of speed estimation, this can include using additional methods such feature tracking, optical flow analysis, or sophisticated algorithms. The objective is to determine the vehicle's speed with great accuracy.

f) Save Data to CSV File: Following the determination of the vehicle's speed, number plate number, and any additional pertinent qualities, all relevant data is collected and stored in a CSV (Comma-Separated Values) file. The analyzed data is stored and organized in a structured format called a CSV file that makes it simple to access, retrieve, and conduct further research.

The initiative seeks to offer thorough information for vehicle analysis by including the calculation of vehicle speed and its correlation with license plate numbers. This contributes to better decision-making processes by allowing applications like traffic monitoring, surveillance, and law enforcement to have a fuller grasp of vehicle behaviors and trends.

4.2. Implementation of speed detection:

- Importing Libraries: The code imports the required libraries, including math, time, threading, OpenCV (cv2), and dlib (dlib).
- Initializing Variables: The code initializes a few variables, including the path to the

video file (cars.mp4), the width and height of the video frame, and the path to the Haar cascade XML file (myhaar.xml).

- Speed Estimation Function: The estimated Speed function determines a car's speed based on the distance and passing time between two sites.
- Object Tracking Function: The core section of the script where object identification, tracking, and speed estimation occur is the track Multiple Objects function. It sets up the variables needed to monitor autos, such as speed, position, and trackers.
- Using the dlib correlation tracker, it reads frames from the video and carries out automobile tracking. It produces new trackers for newly discovered autos and updates old ones for current vehicles.
- To identify autos in the frame, it employs a Haar cascade classifier (carCascade) every tenth frame. To prevent double tracking, it then compares the discovered automobiles with already installed trackers.
- Based on their past and present positions, it calculates the speeds of the monitored automobiles by drawing rectangles around them in the picture. Using the putText function in OpenCV, the speed is shown on the frame. The final frame, which includes the tracked automobiles and their speeds, is shown in a window and may be saved as an output video file.
- The loop won't stop until the user hits the Esc key or the video stops.
- Main Function: If the script is run as the main programme, it runs the track Multiple Objects function.
- The programme appears to be made to analyze a video file containing footage of vehicles, identify the vehicles using object identification and tracking methods, and calculate their speeds. It's important to keep in mind that the performance and accuracy of the speed estimation may differ based on the video quality, the object detection's precision, and other things.

V. RESULTS

To use the system created for the project "Enhancing Vehicle Analysis with Fine Estimation on Video Streams Using Image Processing Techniques," follow the guidelines below:

Setup Environment:

a. Verify that you have a suitable programming environment, such as Python and its dependencies, installed on your computer.
b. Make a project directory where you maykeep all the system's files and scripts.

Dataset Preparation:

a. Gather video feeds from a variety of sources, including traffic cameras, surveillance cameras, and pertinent databases.
b. Choose a variety of movies that depict various situations, such as various weather and lighting conditions, as well as a wide range of vehicle kinds and sizes.
c. Annotate the films with the relevant real-world data, such as bounding boxes around cars and license plate numbers. This annotated data will be used as a teaching and assessment tool[14].

Pre-processing:

a. Implement the jobs for pre-processing the video frames to improve them for analysis.
b. Use frame stabilization techniques to reduce camera vibrations and movement, resulting in stabilized frames for further analysis.
c. Use algorithms or filters that reduce noise artefacts to improve image quality.
d. To increase visibility and details in the video frames, use image enhancement techniques like contrast modification, histogram equalization, or adaptive filtering.

Model Training:

a. Install the necessary libraries after selecting a suitable deep learning framework, such as TensorFlow or PyTorch.
b. As the foundational model for vehicle recognition and tracking, choose an appropriate convolutional neural network (CNN) architecture, such as Faster R-CNN, SSD, or YOLO.
c. Use the previously created labelled video dataset to fine-tune the chosen base model.
d. Create suitable loss functions, optimization techniques, and training schemes to enhance the performance of the model.
e. Adjust the model's hyperparameters iteratively until the required degree of precision and generalization is reached.

Test Data Analysis:

a. Use the trained model to analyze and count vehicles in distinct video streams.
b. Compare the identified vehicle count with the ground truth annotations to assess the model's performance.
c. Determine statistical metrics like accuracy, recall, and F1 score to evaluate the model's performance in terms of numbers.
d. Carry out a qualitative analysis of the results by visually examining them and examining any differences between the model's predictions and the facts.

Vehicle Classification and License Plate Recognition:

a. Sort the identified vehicles into distinct groups, such as automobiles, trucks, motorbikes, etc., using the trained model.
b. Use OCR methods to identify and gather license plate data from the observed vehicles.

c. To increase recognition accuracy, fine-tune the OCR engine using a sizable collection of labelled licence plate photos.

Vehicle Speed Estimation:

a. To calculate the time elapsed between each frame in the video stream, examine successive frames.
b. Use motion analysis methods to precisely follow the vehicles and calculate their displacement over time, such as feature tracking or optical flow.
c. By dividing the displacement by the appropriate time period, determine each vehicle's speed.
d. If further research or sophisticated algorithms are required to correctly predict vehicle speeds, do so.

Data Storage and Reporting:

a. Save all the analyzed data in a structured manner, such as a CSV file, containing the number of vehicles, categorization outcomes, number plate numbers, and predicted speeds.
b. Ensure that the data is correctly organized to facilitate retrieval, access, and subsequent analysis.

Execution:

a. Execute the main system script, which unifies all the produced modules and parts.
b. Provide the necessary information, such as the video stream to be examined or the location of the video file.
c. Keep track of the process's progress and go over the system's output, which should include the data that was analyzed and any reports or visualizations that were produced.

5.1. Output screenshots for license plate detection:

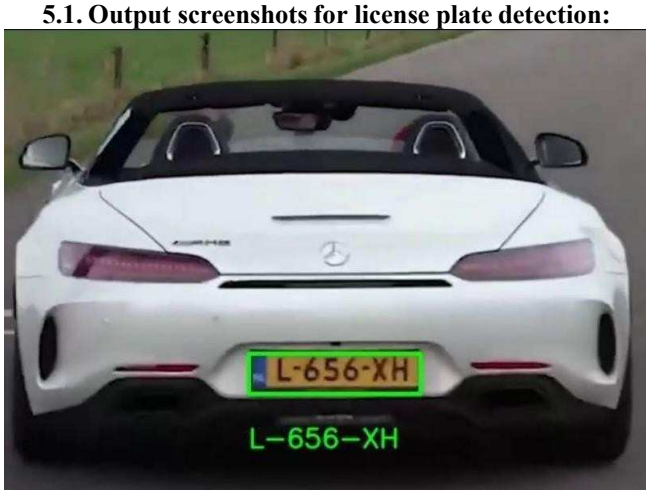

Fig. 4: License Plate Detecting frame by frame from the input video.

Fig 5: License Plate Detecting frame by frame from the input video

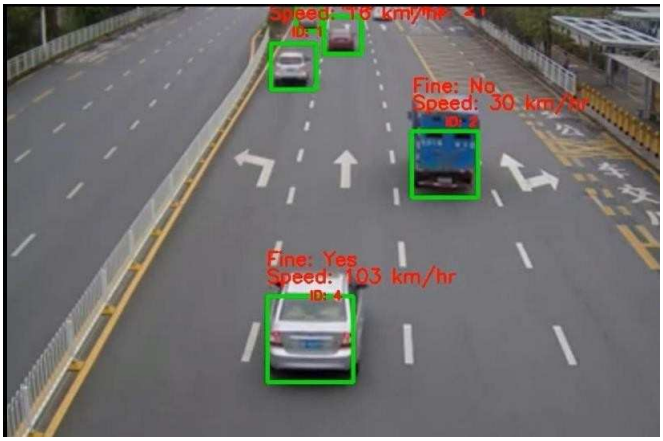

Fig 6: Speed Detecting in the input video along with tracking ID and fine estimation.

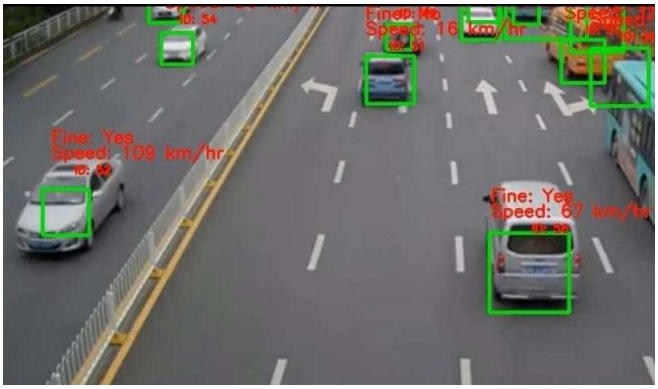

Fig 7: Speed Detecting in the input video along with tracking ID and fine estimation

	A	B	C	D	E
1	Tracking ID	Speed (km/hr)	Fine	Fine Range	
297	1	0	No	N/A	
298	2	0	No	N/A	
299	3	10.41375441	No	N/A	
300	4	14.72727273	No	N/A	
301	5	75.09465102	Yes	1000-2000	
302	0	0	No	N/A	
303	1	10.41375441	No	N/A	
304	2	0	No	N/A	
305	3	16.46559147	No	N/A	
306	4	16.46559147	No	N/A	
307	5	81.33402204	Yes	1000-2000	
308	0	7.363636364	No	N/A	
309	1	0	No	N/A	
310	2	0	No	N/A	
311	3	10.41375441	No	N/A	
312	4	23.28586277	No	N/A	
313	5	58.90909091	No	N/A	
314	0	7.363636364	No	N/A	
315	1	7.363636364	No	N/A	
316	2	10.41375441	No	N/A	
317	3	16.46559147	No	N/A	
318	4	20.82750883	No	N/A	
319	5	74.00362957	Yes	1000-2000	
320	0	7.363636364	No	N/A	
321	1	10.41375441	No	N/A	
322	2	7.363636364	No	N/A	
323	3	7.363636364	No	N/A	
324	4	14.72727273	No	N/A	
325	5	76.87862066	Yes	1000-2000	
326	0	7.363636364	No	N/A	

speeds

Fig 8: CSV file for tracking ID, speeds and fine estimations for vehicles

VI. Conclusion and Future Scope

The use of image processing techniques to improve vehicle analysis with fine estimates on video feeds has the potential to revolutionize several industries, including traffic management, surveillance, and autonomous driving systems. Image processing enables us to extract useful information from video feeds, enabling precise and thorough analysis of cars in real-time. This is done by utilizing cutting-edge algorithms and approaches. This method's main objective is to comprehend and analyze vehicle behavior, traits, and patterns more thoroughly. This improved analysis offers insightful information that may be used to better traffic control systems, increase traffic safety precautions, and help the development of autonomous cars.

Utilizing image processing, we may gather useful data on automobiles, such as their make, model, speed, and trajectory, among other essential parameters.

Utilizing image processing[17] methods for vehicle analysis has several benefits, one of which is the capacity to analyze massive amounts of video data in real-time. Manual analysis techniques are labor-intensive and subject to human mistakes. The automated analysis of video feeds has become considerably more precise and efficient with the development of image processing algorithms, allowing for quicker decision-making and reaction times. Furthermore, using precise estimating techniques enables a more in-depth comprehension of vehicle features. For instance, recognizing particular characteristics[18] and identifying license plate numbers might help law enforcement locate stolen vehicles or identify potential suspects in criminal investigations. Additionally, fine estimates may be used to track traffic jams, find lane infractions, and examine parking trends, resulting in better traffic flow and better infrastructure utilization.

The efficiency of image processing methods for vehicle analysis has greatly benefited from developments in machine learning and deep learning. In tasks like object identification, tracking, and classification, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and other deep learning architectures have displayed astounding ability.

These algorithms[15] are capable of learning from complicated situations and adapting to them, delivering accurate and consistent results even under difficult circumstances like bad lighting, occlusions, or bad weather. Further improving the precision and robustness of vehicle analysis systems is the integration of image processing methods with additional technologies like LiDAR and radar.

It is feasible to develop a thorough awareness of the environment and increase the overall dependability of the analysis by merging different sensor inputs. These sensor fusion techniques enable the development of intelligent transportation systems that can adapt to dynamic scenarios. The profession has made considerable strides and has benefited much, yet there are still issues and constraints that must be resolved. Real-time video analysis might have significant computing needs, needing high-performance hardware or effective parallel processing architectures. Large volumes of labelled training data are also required, and any biases in these datasets might affect the analysis's precision and generalizability[19].

Moreover, while establishing vehicle analytic systems[16], privacy issues and ethical considerations must be considered. Concerns concerning privacy rights are raised by the usage of video feeds and the extraction of sensitive data from automobiles. To ensure responsible management of the acquired and processed data in compliance with legal and ethical frameworks, it is essential to have adequate protections and laws in place. In conclusion, applying image processing techniques to improve vehicle analysis with fine estimate on video streams opens a wide range of prospects for the development of transportation systems. We can accomplish precise and real-time vehicle analysis by utilizing cutting-edge algorithms, deep learning architectures, and sensor fusion techniques.

This will improve traffic management, increase road safety, and pave the way for the creation of intelligent transportation systems. To guarantee the appropriate and successful application of these approaches, it is crucial to meet computational needs, data constraints, and ethical issues. The future analysis and comprehension of automobiles will be fundamentally changed by ongoing image processing research and development.

VII. References

[1] PyImageSearch. (2018, November 12). YOLO object detection with OpenCV.

[2] Bewley, A. (n.d.). SORT: A Simple, Online and Realtime Tracker.

[3] Pham, D. T., Nguyen, V. C., & Nguyen, T. M. (2019). Vehicle Make and Model Recognition using Deep Learning and Image Processing Techniques.
[4] Zhang, Q., Zhang, X., & He, R. (2020). License Plate Recognition in Unconstrained Scenarios Using Deep Learning and Image Processing Techniques.
[5] Kalra, N., Sharma, M., & Bansal, A. (2018). Vehicle Detection and Tracking in Video
[6] Surveillance using Deep Learning and Image Processing Techniques.
[7] Saha, A., Chowdhury, S. S., & Islam, M. N. (2020). Real-Time Vehicle Detection and Tracking System for Traffic Analysis.
[8] Adhav, R. D., Jadhav, A. S., & Patil, A. B. (2018). Vehicle Speed Estimation using Image Processing Techniques.
[9] M Al-Smadi, K Abdulrahim, & R A Salam. (2016). Traffic surveillance: A review of visionbased vehicle detection, recognition, and tracking. International Journal of Applied Engineering Research ISSN, 11(1), 713–726.
[10] Erik Bochinski, Volker Eiselein, & Thomas Sikora. (2017). High-Speed tracking-by-detection without using image information. In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), (pp. 1–6). Lecce, Italy: IEEE.
[11] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, & Zheng Zhang. (2015). Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. CoRR, abs/1512.01274
[12] Gong Cheng, Peicheng Zhou, & Junwei Han. (2016). Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. IEEE Transactions on Geoscience and Remote Sensing, 54(12), 7405–7415.
[13] DeviceHive. (2013–2018). Open source IoT data platform. Retrieved from https://github.com/devicehive/devicehive-video-analysis.
[14] Jorge E Espinosa, Sergio A Velastin, & John W Branch. (2017). Vehicle Detection Using AlexNet and Faster R-CNN Deep Learning Models: A Comparative Study. In H. Badioze Zaman, P. Robinson, A. F Smeaton, T. K Shih, S. Velastin,
[15] T. Terutoshi, A. Jaafar, & N. M. Ali (Eds.), Advances in Visual Informatics, (pp. 3–15). Cham: Springer InternationalPublishing.
[16] Quanfu Fan, Lisa Brown, & John Smith. (2016). A closer look at Faster R-CNN for vehicle detection. In IEEE Intelligent.
[17] Vehicles Symposium, Proceedings, (Vol. 2016-Augus, pp. 124–129). Gothenburg, Sweden: IEEE.
[18] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, & Deva Ramanan. (2009). Object Detection with Discriminatively Trained Part-Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9), 1–20.
[19] Xiao-Feng Gu, Zi-Wei Chen, Ting-Song Ma, Fan Li, & Long Yan. (2016). Real-Time vehicle detection and tracking using deep neural networks. In 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), (pp. 167–170). Chengdu, China: IEEE.