RAPID VEHICLE SUPERVISION AND DETECTION IN A TERRITORY

Lakshmi T [1], Safiya Begam G [2], Asis Jovin [3]

[1, 3] *Department of AI & DS, Karpaga Vinayaga College of Engineering and Technology, Tamil Nadu, India*

[2] *Department of Computer Science, B.S.Abdur Rahman Crescent Institute of Science and Technology, Tamil Nadu, India*

ABSTRACT:

Deep learning algorithms may now be widely used to perform analysis of traffic video footage thanks to rapid recent advances in the processing capabilities of common computers. Basic components of traffic analysis include traffic flow estimates, anomaly detection, vehicle identification, and video vehicle tracking. One of the most important study subjects in recent years has been traffic flow forecasts or speed estimate. By better anticipating transit demand, good alternatives for this problem might help reduce traffic incidents and enhance road design. In this work, we provide various cutting-edge techniques to vehicle speed monitoring, alongside our remedy to the monitoring issue.

KEYWORDS: Deep Learning, Vehicle Tracking, Traffic, Speed Monitoring.

1. INTRODUCTION:

Technology growth has transformed many parts of our life, including communication and security. One of the most difficulties that authorities and organizations confront is effectively monitoring and controlling cars in a specific zone. Whether for road safety, law enforcement, border security, or critical infrastructure monitoring, the necessity for precise and timely vehicle supervision and detection has become vital.Rapid Vehicle Supervision and Detection is the process of monitoring, identifying, and tracking cars in a given region in real time utilizing modern technology and intelligent systems. This approach entails combining cutting-edge hardware such as cameras, sensors, and radars with sophisticated applications and machine-learning algorithms.Rapid vehicle supervision is built on modern technology such as machine learning, vision, and deep learning algorithms. To detect and identify cars effectively, these systems can handle massive volumes of data from many sources, such as CCTV cameras, satellite imaging, and other sensors.

Background modeling yields the highlighted portion of the video stream by subtracting consecutive frames. The exact location of the vehicle is detected through foreground detection. Centroids and a histogram of the removed picture are used to track the vehicle. This technique is also useful for determining vehicle velocity. The initial step in estimating vehicle speed is to accurately recognize the cars in the video.

For years, computer vision techniques have been used to study object detection. Traditional computer vision approaches typically evaluate the shape, contrast, and other properties of an item to detect it. However, the effective adoption of artificial intelligence technologies, particularly the deep convolution neural network (CNN), has enhanced object detection. The context removal and frame differencing of object identification approaches may be classed.

1.1 Analyzing the Threshold value:

In this case, we will evaluate the hue level of an item while excluding the backdrop sceneries. We shall be able to distinguish between vehicle and background by taking threshold intensities into account for an area. As a result, determining the appropriate threshold value becomes critical. If this threshold value is too low, we may miss the specific car. And if the threshold value is higher, there is a risk that the noise may be detected together with the car. Using this method, determining the minimum and maximum intensities is quite challenging. For object detection, gray-scale morphological techniques have been applied.

Background subtraction using a Gaussian mixture model is the best strategy for object detection. There are several approaches available, but the most effective is the Gaussian Mixture Model (GMM). The GMM acts at the pixel level and is most commonly used for still backgrounds. The difference between two successive frames is used to detect foreground items in the background using a subtraction approach.

To get the necessary results, frame transistors have to be eliminated from the needed video using the model developed by the GMM for background removal. It should respond to frame changes caused by the object's start and stop. Vehicle tracking from a video with automobiles in motion is a difficult undertaking. Many challenges can develop during vehicle tracking, such as a lack of hard vehicle structure, variations in appetite pattern, and occlusions connecting objects-to-scene. In general, tracking is the challenge of detecting an object's trajectory motion as it travels about in a scene. The object tracker labels all of the tracked cars here. In other words, a tracker can also offer centric-information, vehicle orientation, and vehicle shape.

So many techniques have been presented, but few of them are best suited to the requirement of employing best at night. As a result, you are leaving this section for future expansion. For accurate findings, we can develop a system that includes a calibrated camera with a high resolution that can assist us in low-light situations.

.

2. LITERATURE SURVEY:

RADAR systems were traditionally used to monitor vehicle speeds. The main disadvantage of RADAR is its high cost and lack of precision. The technology is highly costly, less precise, and requires uninterrupted line of sight connectivity between the RADAR and the cars. To circumvent the limitations of existing approaches, the suggested system employs the Gaussian methodology. As a result, numerous methods of image processing are applied to conquer the limitations of existing technologies. In general, tracking is the challenge of detecting an object's trajectory motion as it travels about in a scene.

Due to the creation of a bounding box using the KLT algorithm, it does not maintain the speed estimation. As a result, that algorithm lacks the consistency factor of speed estimation. The accuracy factor of creating bounding boxes while tracking vehicles is not working. It only accepts two frames as input. That algorithm also lacks the efficiency factor.

2.1. AUTOMATED TECHNIQUE DEVELOPMENT FOR MOTOR RATE PREDICTION AND MONITORING IN THE MEDIA STREAM.

In the past, inductive loops, RADAR guns, LASER guns, or manual counts were employed to determine vehicle speed. Inductive loops have significant maintenance and installation costs, and RADAR systems are not suitable for traffic investigation systems because to their high cost and low accuracy. The motion began in the estimate of the car's speed using the spherical projection equation. Previously suggested vehicle detection research employs a multi-camera technique in which items that cannot be caught with the first camera can be recorded by the succeeding cameras. The method occasionally employs up to 20 cameras to scan passing automobiles. Similarly, a new property known as "vehicle linearity" is defined, which categorizes the vehicle type. It was very useful for traffic control system in finding the images of high speed vehicle. It is very costly using the RADAR system, also the maintained of the system is very time taking.

2. 2. EVALUATION OF VEHICLE SPEED FROM SURVEILLANCE VIDEO INFORMATION.

Traffic data is essential in Intelligent Transportation Systems (ITS) research and implementations. Current approaches, such as deploying roadside radar sensors upon roads electromagnetic loop detectors at junctions, GPS data acquired from probe fleets, and so on, have made a wealth in information available for roadway monitoring. Aside from such technologies, surveillance cameras are frequently utilized but have not yet been employed to collect traffic data. Changing the camera's usage from manual inspection to routine surveillance would be a worthwhile challenge. Surveillance cameras might be used by agencies not only for visual confirmation of occurrences, but also for speed data gathering and traffic flow assessment. Some benefits includes a) multi-object detection using Faster R-CNN, b) multi-object tracking based on histogram comparison, and c) speed conversion using warping with linear perspective transformation. Changing the camera's usage from manual inspection to continuous surveillance would be a positive challenge.

2.3. MOTOR SPEED IDENTIFICATION ALGORITHM ANALYSIS IN CAMERA SURVEILLANCE

Automobile recognition is the most fundamental and crucial component of intelligent transportation. Infrared detection, inducement line detection, ultrasonic detection, acoustic array detection, radars, video image detection systems, and other traditional detection methods are included in this research. Various novel vehicle recognition algorithms in video surveillance have been discovered during the previous few decades. The detection of vehicles methods centered on video may collect a wealth of information from video picture sequences, such as vehicle speed, flow, vehicle type, and so on, at a cheap cost and with great efficiency.

The research investigated the vehicle speed identification technique in video based on shifting target detection. The primary tasks are motion detection, vehicle tracking, and vehicle speed estimate. The vehicle's approximation velocity is computed by examining the centroids location of the moving target and the mapping connection. The approach has high resilience and practicability, however there is some mistake. The next step is to integrate the speed computation with more detailed information about visual motion.

3. PROPOSED WORK:

The suggested method estimates the speed of the vehicle approaching the camera by following its motion over a succession of photos. The suggested system comprises mostly of five components. The collected video is transformed into frames during preprocessing, and noise is eliminated using the Median Filter approach. The background Subtraction (Mixture of Gaussian) approach is utilized in moving vehicle identification. We are analyzing films collected by stationary cameras and using an optical-flow based approach to recognize several cars. Multiple pictures are detected at various moments using this approach, and the optical-flow-based technique identifies barriers indirectly by evaluating the velocity field.The identified foreground is cleared using morphological techniques such as opening. To detect each item, the Connected Component Method is employed. The ShiTomasi approach is used to detect corners in feature extraction. Vehicles are tracked using the optical flow approach in vehicle tracking. The centroids of each vehicle are computed during speed estimate. Centroids are monitored throughout a number of frames. The time and distance traveled by the vehicle are used to compute velocity.

3.1 STEPS INVOLVED IN PROPOSED SYSTEM:

1. The vehicle's speed is measured via tracking, and an image sequence is acquired.
2. The collected video is transformed into images and the unwanted information is eliminated using the Median Filter method during preprocessing.
3. The CNN algorithm, or Convolution Neural Network, is employed for this.
4. Vehicle velocity is computed using the distance traveled and the time consumed.
5. Pixels are deemed foreground when the separation between the backdrop and the present picture is large enough.

3.2 VEHICLE TRACKING:

Vehicle detection from collected video may be accomplished using a variety of ways. Accurately detecting moving vehicles in video is a difficult challenge. The moving vehicle detection classifies each of the pixels in the movie's frames as either salient or background data. Frame differencing and the background removal technique are two ways for vehicle detection. The frame differencing approach can be used to exclude static items (those that are not moving) and consider moving things. The fundamental disadvantage of this strategy is that slow moving items are not identified, necessitating improvisation.

Backdrop subtraction is the process of removing the backdrop or static scene from a video picture. Because the camera is stationary, each pixel in the picture has an associated number that is essentially constant across time. The goal of subtracting backgrounds is to determine the background value of each picture point. We utilized the Gaussian Mixture approach to detect the vehicle's motion since it overcomes the limitations of the other methodologies. The best method for eliminating the backdrop and considering the car from the video is background subtraction, in which each frame of the footage is compared with a reference or backgrounds model, and the pixels where divergence from the background is noticed are deemed to be the vehicles.

Accuracy of creating bounding boxes is better than existing System. Speed estimation is also taken a lesser time while compare to KLT algorithm. Efficiency of tracking a vehicle from captured video is also high in that proposed techniques. More often using object detection technique in vehicle tracking is better than using RADAR techniques for traffic signal monitoring.

Estimating vehicle speed is frequently beneficial in system as well as other processing of images application. For better results, the video quality is initially enhanced by applying noise reduction techniques such as the Median filter to the input footage. The next step is to remove the backdrop or static scene, leaving only the foreground. There are several strategies available here, and the particular one we utilized is a Gaussian mixture. The resulting foreground is cleaned up using morphological approaches such as opening. Optical flow is used to track vehicles in the next phase. And the axis centers of all objects are determined at each occurrence to anticipate their future motion. Finally, the speed of moving vehicles is approximated based on the distance traveled.

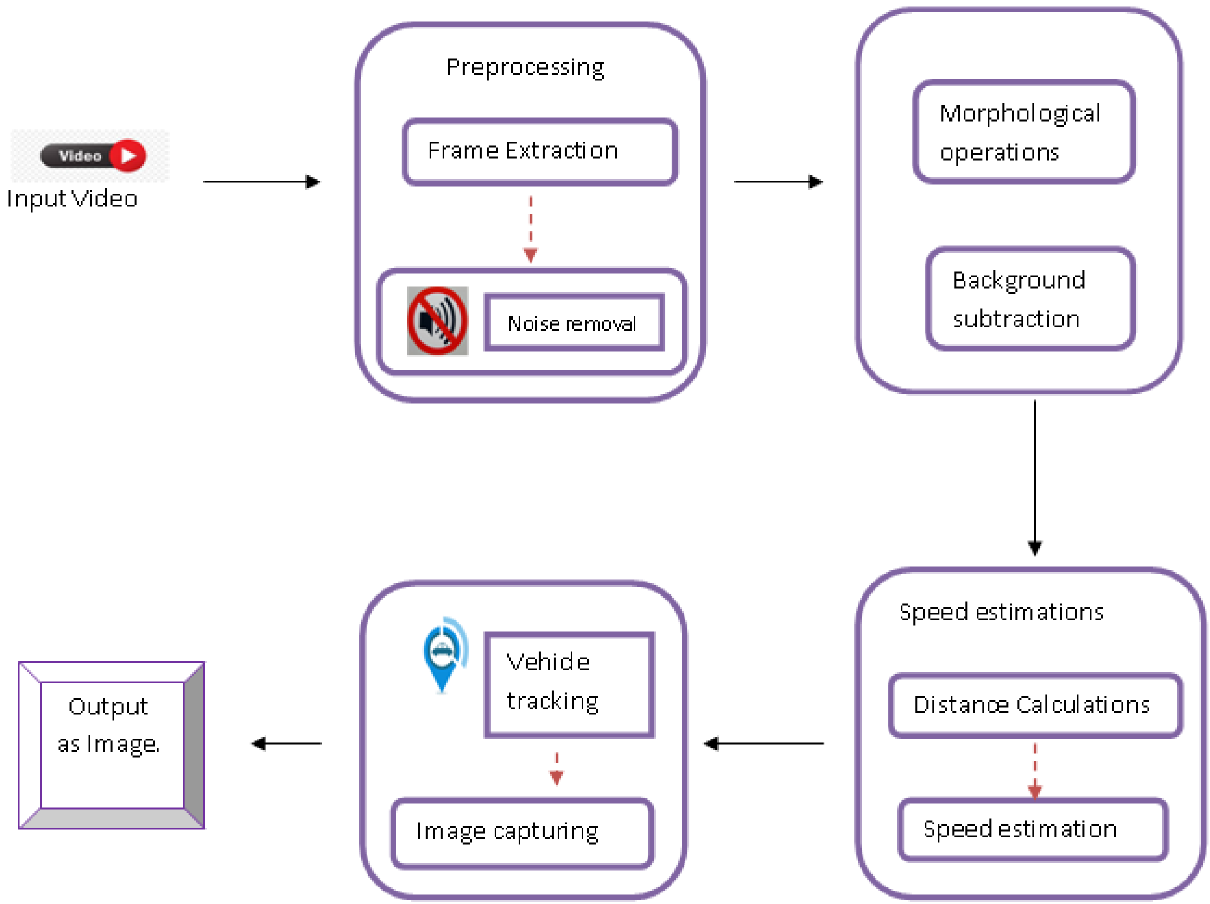

Figure 1: SYSTEM ARCHITECTURE

3.2.1. Preprocessing:

The first stage is video preparation, which includes frame segmentation and video enhancement. The provided video is transformed into n frames and the frame rate is retrieved during preprocessing. In the following stage, the video quality is increased by minimizing background noise. The median filter method is used for this. This is a filtering nonlinear technique that is frequently used to eliminate noise from the background and so produce a better foreground. This approach is frequently used as an initial process to increase the accuracy for further processing. Although there are other approaches, we favor the median filtered since it keeps the edges while reducing noise.

3.2.2. Detection of Moving Vehicles:

Background subtraction is the next stage. It is a method in the realm of computational imaging that extracts the foreground of a picture. It is also known as Background Detection. Vehicles in the foreground are the image's regions of interest. This approach is frequently used in object localization. It is assumed here that the intensity values of each pixel in the video may be described using a model consisting of a Gaussian mixture (MOG). A general rule defines which intensities belong to the background and which belong to the foreground. Foreground pixels are those that are incompatible with any of these. The pixels in the foreground are categorized using 8 connectivity linked component analysis.

3.2.3. Extraction of Characteristics:

Corner identification is a popular method for extracting specific types of features. It is mostly used to determine the content of a picture. There are certain corner selection criteria in the Harris corner detector. A score is assigned to each pixel in the pictorial representation, and if the score exceeds a certain threshold, the pixel is designated as a corner; otherwise, it is not.

3.2.4. Vehicle Monitoring:

The practice of tracing a moving item in relation to frames is referred to as vehicle tracking. Tracking is accomplished by extracting features (Shi-tomasi technique) from elements in a frame then discovering the items in order. We calculate the position relative to the moving item through looking at the relative position values of the object in each frame. It is the pattern of movement of picture objects between two successive frames produced by camera or object movement. It is a 2D vector field, with each vector representing the progression of points from one frame to subsequent.

3.2.5. Speed Prediction:

The previous and next frame's centroids are used to determine distance. The distance determined by Euclid is used to calculate distance. As the car enters the ROI, time is computed. The speed is initially determined using the formula. Distance travelled= (x2-x1)2+ (y2-y1)2 Speed=Distance travelled/Time taken if (x1, y1) are the centroids of the vehicle in the first frame and (x2, y2) are the centroids of the second frame.

3.3 TEST CASE DIAGRAMS:

A test case is a document that contains a set of test information, preconditions, expected outcomes, and post conditions that have been produced for a specific test scenario in order to check compliance against a given requirement.

3.3.1. USE CASE DIAGRAM:

A use case diagram is a depiction of a user's engagement with the system that depicts the interaction between the user and the many use cases in which that user is involved.

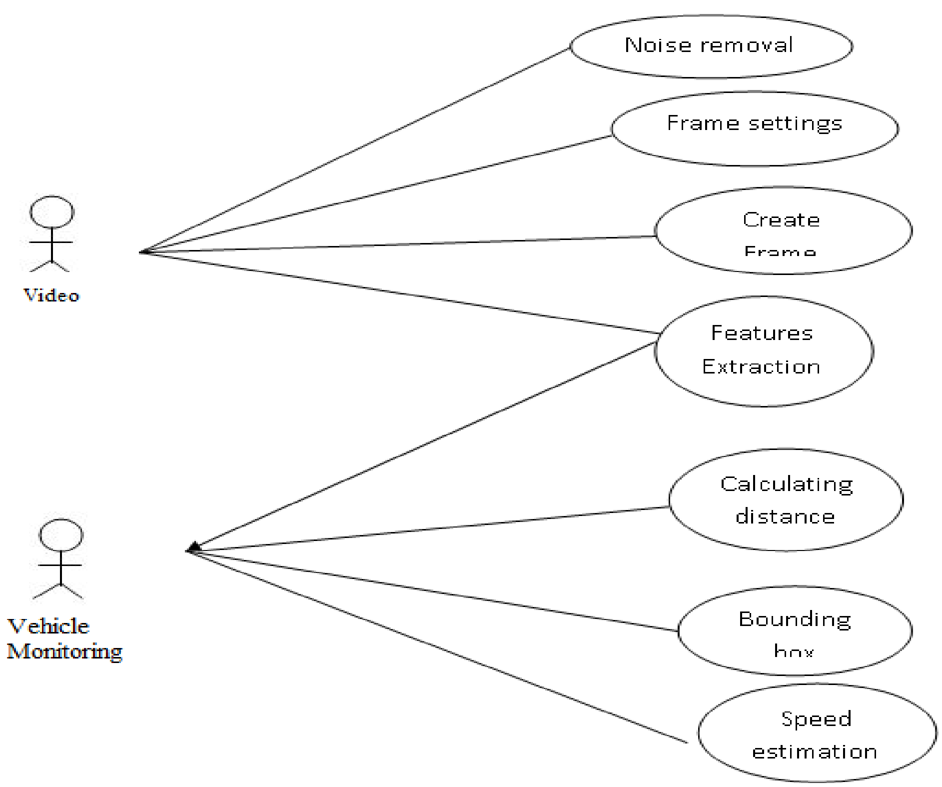

Figure 2: Use Case Diagram

3.3.2. SEQUENCE DIAGRAM:

A sequence diagram depicts object interactions in temporal order. It illustrates the classes and objects that involved in the scenario, as well as the ordered collection of messages sent between the objects required to carry out the scenario's functionality.

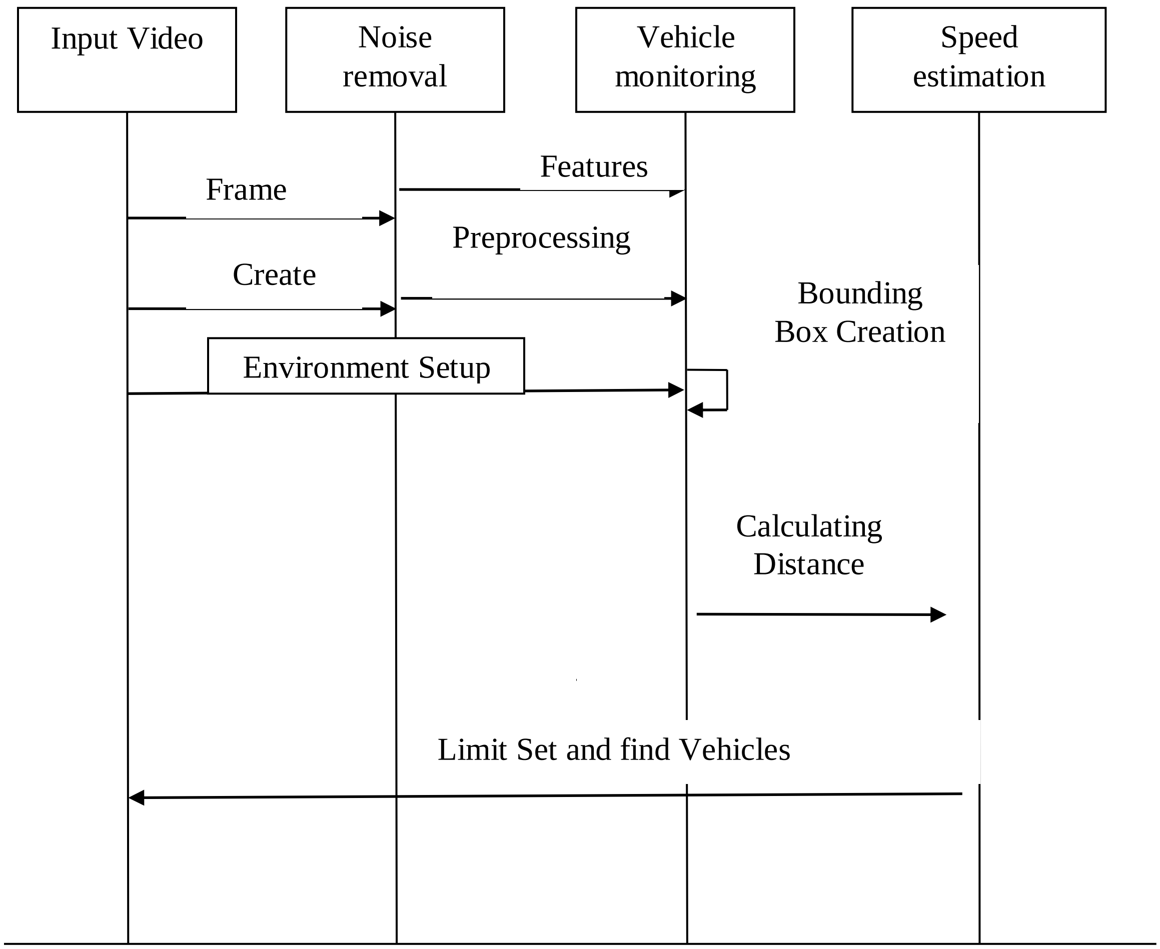

Figure 3: Sequence Diagram

3.3.3.COLLABORATION DIAGRAM

A conversation diagram in the unified model language 2.0 is a reduced version of a collaboration diagram in UML

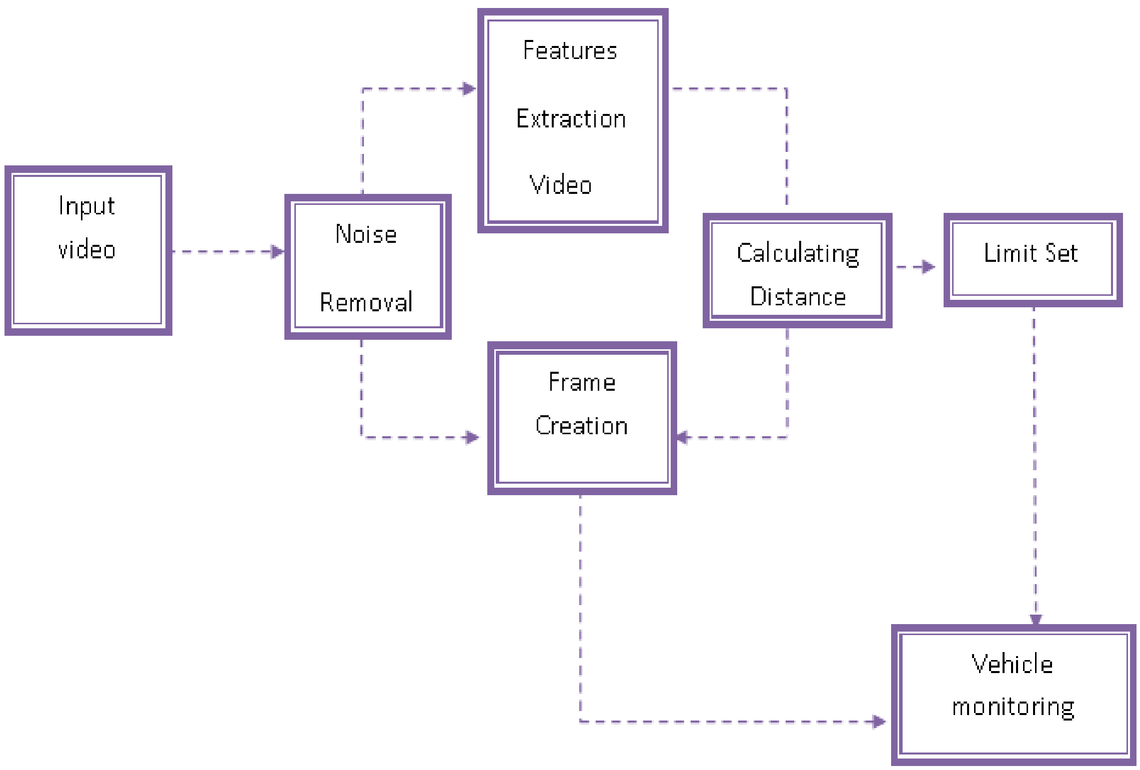

Figure 4: Collaboration Diagram

4. EXPERIMENTAL RESULTS:

Errors and prejudice can enter a system of monitoring at any point. Because records from surveillance are used for determining high-risk populations, focus treatments, and assess interventions, it is critical to understand the system's strengths and weaknesses. So far, the discussion of characteristics has focused on the information gathered for cases, although morbidity and death rates are estimated in many monitoring systems. The numbers used for these rate computations are frequently collected from a wholly independent data system administered by another organization, such as the Census Bureau. Consider the comparability of the categories (for example, race, age, and residency) on which the numerator and denominator of rate computations are based.

4.1 TESTING STRATEGIES:

Software testing is the process of comparing software to user requirements and system specifications. Testing occurs at the phase level of the software life cycle of development or at the module level of the program code. Validation and verification are components of software testing.

4.2 TESTING LEVELS:

1. TESTING OF UNITS
2. TESTING OF INTEGRATION
3. SYSTEM EVALUATION
4. EVALUATION OF FUNCTIONALITY
5. EVALUATION OF PERFORMANCE
6. SECURITIES AND PORTABILITY
7. TESTING FOR ACCEPTANCE
8. ALPHA TESTING
9. BETA TESTING
10. REGRESSION TESTING

1. TESTING OF UNITS

While coding, a programmer tests the unit of software to see if it is error-free. The white-box testing technique is used for testing. Unit testing allows developers to ensure that individual software units are operating properly and without errors.

2. TESTING OF INTEGRATION

Even if the software modules perform well independently, it is necessary to determine if the units will work correctly when integrated together. For instance, argument forwarding and data updation.

3. SYSTEM EVALUATION

The software is packaged as a product and then tested as a complete. This can be performed by the use of one or more of the following tests:

4. EVALUATION OF FUNCTIONALITY

Tests all functionalities of the software against the requirement.

5. EVALUATION OF PERFORMANCE

This test demonstrates how effective the program is. It evaluates the software's efficacy and average time to do the targeted task. Load and stress testing are methods of performance testing in which the program is subjected to significant user and data loads in a variety of environments.

6. SECURITIES AND PORTABILITY

These tests are performed when the product is intended to run on several platforms and be accessed by a large number of people.

7. TESTING FOR ACCEPTANCE

When the program is ready to be handed over to the client, it must go through the last stage of testing, which includes testing for user interaction and reaction. This is significant because even if the program meets all of the user's needs, if the user does not like the way it looks or functions, it may be rejected.

8. ALPHA TESTING

The developer team does alpha testing by utilizing the system as though it were in a working context. They attempt to determine how a user would react to a software operation and how the system should respond to inputs.

9. BETA TESTING

After the program has been internally tested, it is sent to users for testing purposes exclusively in their production environment. This is not the finished product. Developers anticipate that at this point, consumers will bring minor issues that were overlooked in order to attend.

10. REGRESSION TESTING

When a piece of software is updated with new code, a feature, or functionality, it is rigorously tested to see whether the extra code has a negative influence. This is referred to as regression testing.

4.3. BLACK-BOX TESTING:

It is done to test the program's functionality. It is also referred to as 'Behavioral' testing. In this scenario, the tester is given a set of input values and the intended outcomes. When given input, the program is checked 'ok' if the output matches the intended results, and problematic if it does not. The tester does not know the design or structure of the code in this testing approach, and the program is tested by testing engineers and end users.

4.3.1 BLACK-BOX TESTING TECHNIQUES:

1. Equivalence class - The information being provided is classified into related groups. If just one component of a class satisfies the test, the entire class is presumed to pass.

2. Boundary values - The input is separated into upper and lower bounds. If these numbers pass the test, it is presumed that all values between them will also pass.

3. Cause-effect graphing - Only a single input value is examined at a time in both prior techniques. A cause (input) - effect (output) testing approach tests every combination of input values in a systematic manner.

4. Pair-wise Testing - Multiple parameters influence software behavior. Multiple parameters are examined for their distinct values in pair wise testing.

5. State-based testing - When input is provided, the system's state changes. The states and input of these systems are used to test them.

4.2 WHITE-BOX TESTING:

It is used to put the software and its implementation to the test in order to enhance code efficiency or structure. It is also referred to as 'Structural' testing. The tester is aware of the design as well as structure of the code while using this testing approach. This test is carried out by code programmers.

4.2.1 WHITE-BOX TESTING TECHNIQUES:

1. Control-flow testing - The goal of control-flow testing is to create test cases that cover every single statement and branch situations. To ensure that all claims are addressed, the organization's conditions are verified for both true and false.

2. Data-flow testing - This testing approach focuses on covering all of the data variables in the software. It examines where variables were declared and defined, as well as where they were used or altered.

4.3 VALIDATION AND VERIFICATION:

Software Validation:

This test Validation is the process of determining whether or not the program meets the needs of the user. It is completed at the conclusion of the SDLC. The program is verified if it meets the specifications for which it was created.

Validation ensures the product under development is as per the user requirements.

Software Verification:

Confirmation is the process of ensuring that software meets business needs and is produced in accordance with the appropriate specifications and techniques.

Verification verifies that the product being created meets the design criteria.

Verification provides a response to the inquiry, "Are we manufacturing this product while strictly adhering to all design specifications?"Design and system requirements are the focus of verifications.

Target of the test:

Errors - are genuine coding errors committed by developers. Furthermore, a disparity between the output generated by the program and the expected output is called an error.

Fault - A fault happens when there is a mistake. A fault, commonly referred to as a bug, is a mistake that can cause a system to fail.

Failure - is defined as the system's inability to fulfill the required job. When a problem exists in the system, failure happens.

4.4 FINDING THE VEHICLE AND ITS SPEED, ALONG WITH AN IMAGE:

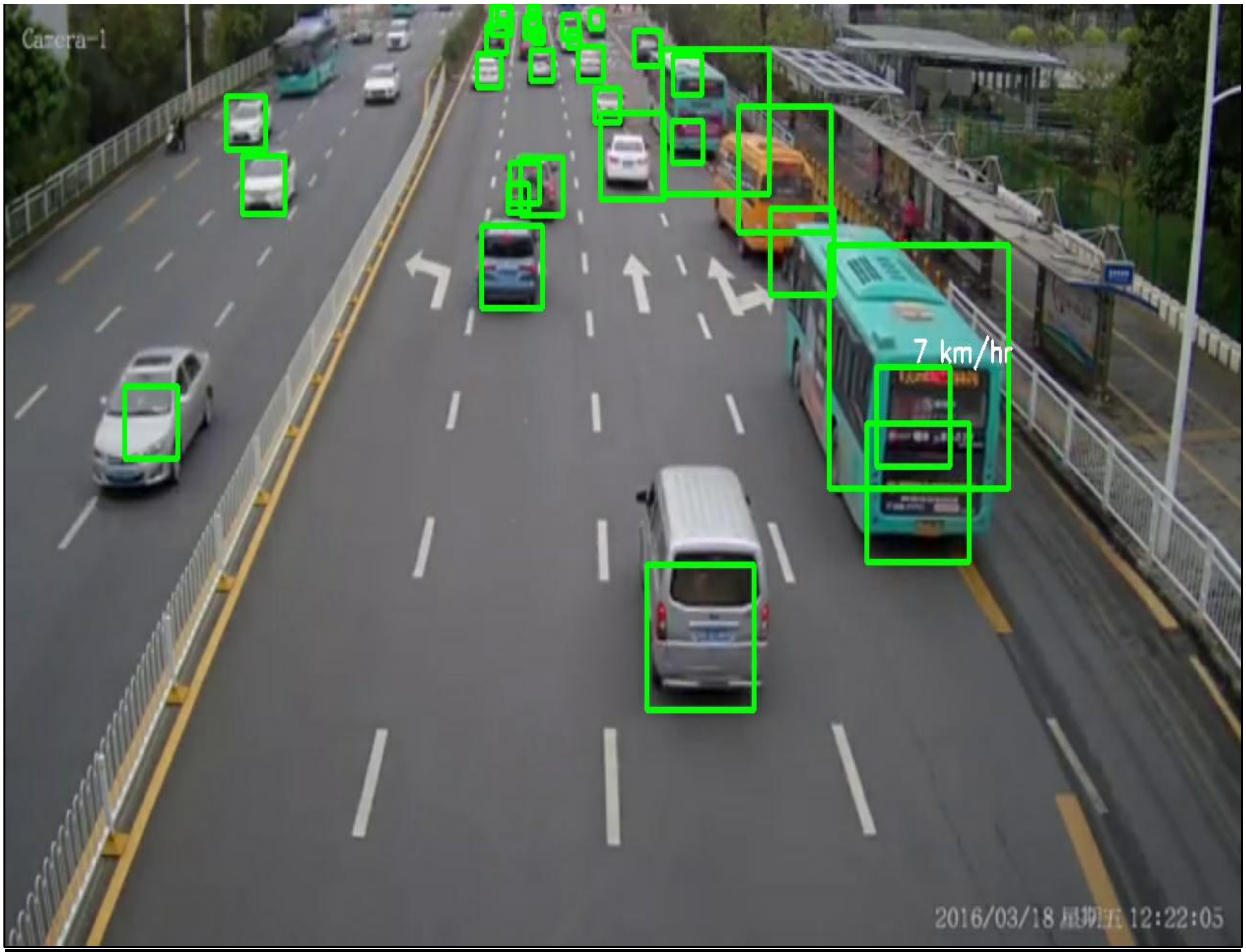

Figure 5: Vehicles and its speed

From figure 5 - The speed of the vehicle is estimated by tracking and the Sequence of images is captured. Speed estimation through image processing is efficient and economical than Conventional radar technology and takes benefit from its wide possibilities.

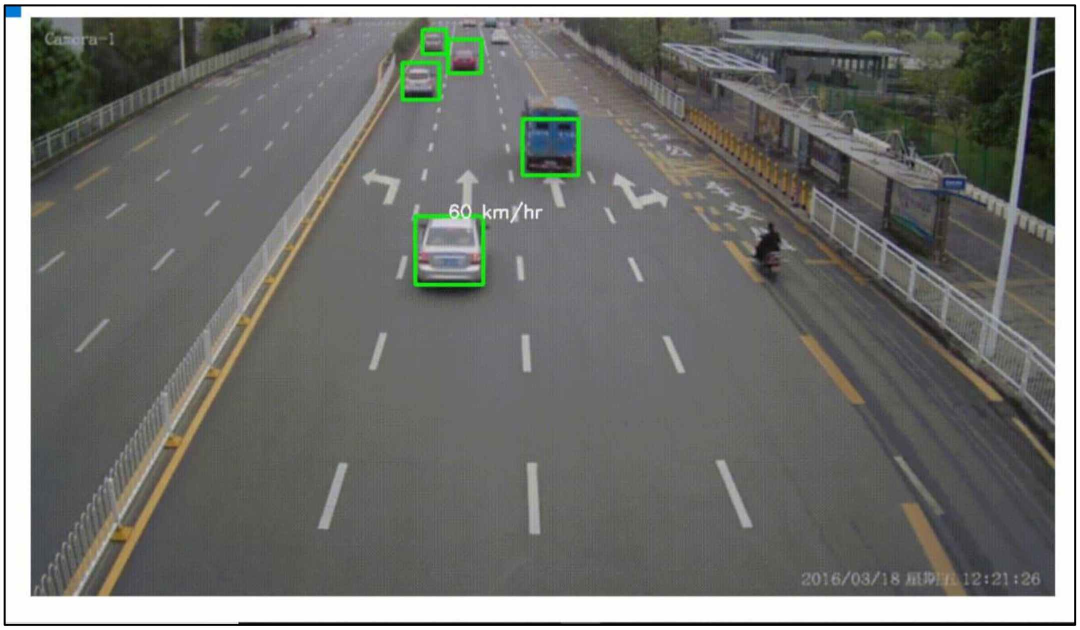

Figure 6: Finding Speed Vehicle

From Figure 6: Vehicle speed estimation using image processing from video data and Euclidean distance is used for distance calculation. Backdrop subtraction is the process of removing a video's backdrop or static scene. Because the camera is fixed, each pixel in the image is assigned a number that is essentially constant over time. Subtracting backgrounds is used to determine the background value of each picture point.

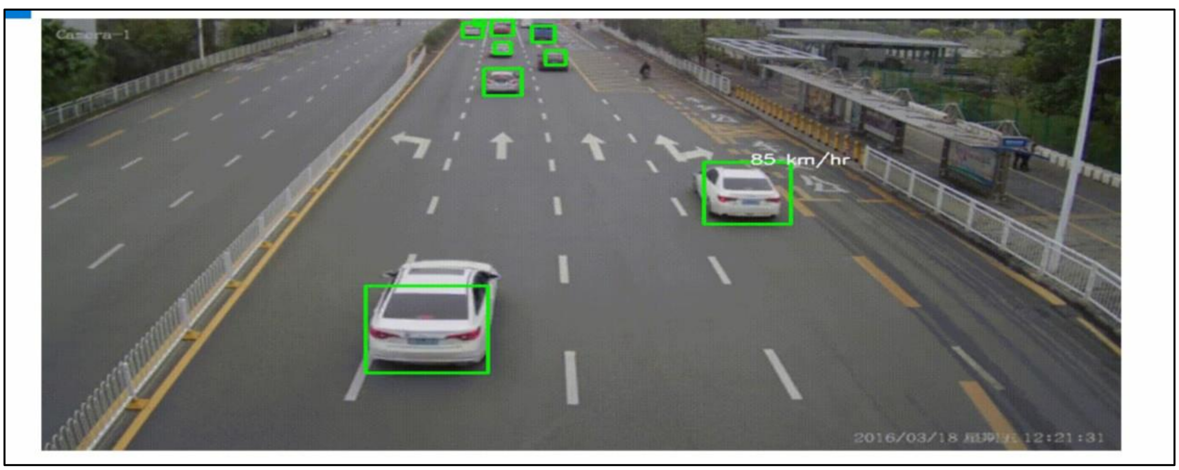

Figure 7: Speed vehicle Image Capture

From figure 7: Overcomes the limitations of the other methodologies, the Gaussian Mixture approach is used to detect vehicle speed and track the vehicle.

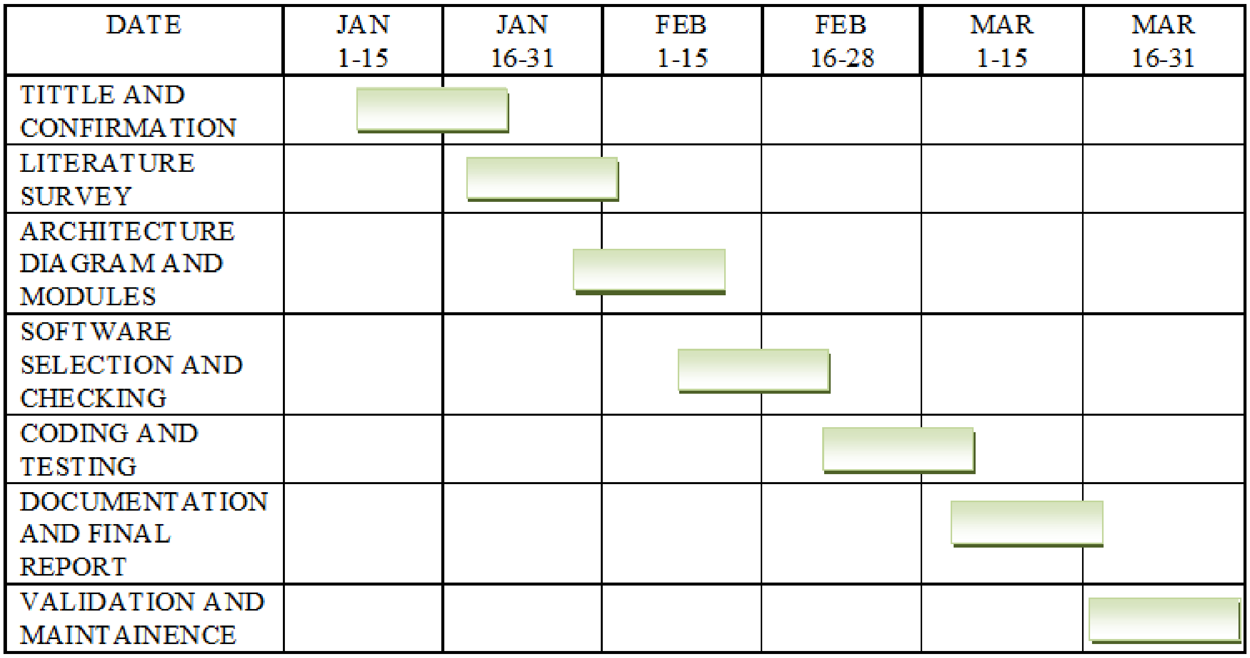

Figure 8: Gantt chart for proposed work

4.5 CONCLUSION:

Image processing-based speed estimate is more efficient and cost-effective than traditional radar technology, and it takes advantage of its extensive capabilities .Converting low-cost cameras into functional sensors is a worthwhile endeavor. This study attempts to extract vehicle speeds from surveillance video data using computer vision and deep learning techniques that are quickly developing. The next step is to incorporate more detailed visual motion information for speed computation. This software project will be expanded further in the future, with the assistance of suitable algorithms for future advancements. Additional applications for determining vehicle speed can also be created. Image processing-based speed estimate is more efficient and cost-effective than traditional radar technology, and it takes advantage of its extensive capabilities. Why because we utilized the Mat lab platform for programming, high-quality movies required significantly more processing time than standard video, resulting in a slow system. As a result, normal-sized videos are suggested for this project. The next step is to incorporate more detailed visual motion information for speed estimation. In the future, this software project will be expanded and enhanced with the aid of adequate algorithms. Additional applications for determining vehicle speed can also be created.

REFERENCES:

1. https://www.researchgate.net/publication/322514557_Development_of_automated_technique_for_vehicle_speed_estimation_and_tracking_in_video_stream
2. https://www.researchgate.net/publication/325768038_Traffic_Speed_Estimation_from_Surveillance_Video_Data
3. https://www.researchgate.net/publication/313542492_Research_of_vehicle_speed_detection_algorithm_in_video_surveillance
4. https://www.researchgate.net/publication/322511911_Development_of_prototype_for_vehicle_speed_measurement
5. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4883290/
6. Mahmood, Z.; Haneef, O.; Muhammad, N.; Khattak, S. Towards a Fully Automated Car Parking System. IET Intell. Transp. Syst. 2018, 13, 293–302.
7. Xiaohong, H.; Chang, J.; Wang, K. Real-time object detection based on YOLO-v2 for tiny vehicle object. Procedia Comput. Sci. 2021, 183, 61–72.
8. Rani, E. LittleYOLO-SPP: A delicate real-time vehicle detection algorithm. Optik 2021, 225, 165818. [CrossRef]
9. Tajar, T.; Ramazani, A.; Mansoorizadeh, M. A lightweight Tiny-YOLOv3 vehicle detection approach. J. Real-Time Image Process. 2021, 18, 2389–2401. [CrossRef]
10. Mahmood, Z.; Bibi, N.; Usman, M.; Khan, U.; Muhammad, N. Mobile Cloud based Framework for Sports Applications. Multidimens. Syst. Signal Process. 2019, 30, 1991–2019. [CrossRef]
11. Hamsa, S.; Panthakkan, A.; Al Mansoori, S.; Alahamed, H. Automatic Vehicle Detection from Aerial Images using Cascaded Support Vector Machine and Gaussian Mixture Model. In Proceedings of the 2018 International Conference on Signal Processing and Information Security (ICSPIS), Dubai, United Arab Emirates, 7–8 November 2018; pp. 1–4.
12. Mikaty, M.; Stathaki, T. Detection of Cars in HighResolution Aerial Images of Complex Urban Environments. IEEE Trans. Geosci. Remote Sens. 2017, 55, 5913–5924.
13. Arı, Ç.; Aksoy, S. Detection of Compound Structures Using a Gaussian Mixture Model With Spectral and Spatial Constraints. IEEE Trans. Geosci. Remote Sens. 2014, 52, 6627–6638. [CrossRef]
14. Hbaieb, A.; Rezgui, J.; Chaari, L. Pedestrian Detection for Autonomous Driving within Cooperative Communication System. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April

2019; pp. 1–6.
15. Xiong, L.; Yue, W.; Xu, Q.; Zhu, Z.; Chen, Z. High Speed Front-Vehicle Detection Based on Video Multi-feature Fusion. In Proceedings of the 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 17–19 July 2020; pp. 348–351.
16. Yawen, T.; Jinxu, G. Research on Vehicle Detection Technology Based on SIFT Feature. In Proceedings of the 8th International Conf on Electronics Info. and Emergency Communication (ICEIEC), Beijing, China, 15–17 June 2018; pp. 274–278.
17. Li, Y.; Wang, H.; Dang, L.M.; Nguyen, T.N.; Han, D.; Lee, A.; Jang, I.; Moon, H. A Deep Learning-Based Hybrid Framework for Object Detection and Recognition in Autonomous Driving. IEEE Access 2020, 8, 194228–194239. [CrossRef]
18. Li, Y.; Li, S.; Du, H.; Chen, L.; Zhang, D.; Li, Y. YOLO-ACN: Focusing on small target and occluded object detection. IEEE Access 2020, 8, 227288–227303. [CrossRef]
19. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into High Quality Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
20. Wang, C.; Wang, H.; Yu, F.; Xia, W. A High-Precision Fast Smoky Vehicle Detection Method Based on Improved Yolov5 Network. In Proceedings of the 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID), Guangzhou, China, 28–30 May 2021; pp. 255–259. [CrossRef]
21. Miao, Y.; Liu, F.; Hou, T.; Liu, L.; Liu, Y. A Nighttime Vehicle Detection Method Based on YOLO v3. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 6617–6621. [CrossRef]
22. Sarda, A.; Dixit, S.; Bhan, A. Object Detection for Autonomous Driving using YOLO [You Only Look Once] algorithm. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; pp. 1370–1374. [CrossRef]
23. Zhao, S.; You, F. Vehicle Detection Based on Improved Yolov3 Algorithm. In Proceedings of the 2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Vientiane, Laos, 11–12 January 2020; pp. 76–79. [CrossRef]
24. Corovi´c, A.; Ili´c, V.; Ðuri´c, S.; Marijan, M.; Pavkovi´c, B. The Real-Time Detection of Traffic Participants Using YOLO Algorithm. In ´ Proceedings of the 2018 26th Telecommunications Forum (TELFOR), Belgrade, Serbia, 20–21 November 2018; pp. 1–4. [CrossRef]

25. Lou, L.; Zhang, Q.; Liu, C.; Sheng, M.; Zheng, Y.; Liu, X. Vehicles Detection of Traffic Flow Video Using Deep Learning. In Proceedings of the 2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS), Dali, China, 24–27 May 2019; pp. 1012–1017. [CrossRef]