Vehicle Detection And Counting Security System By Using Image Processing

Omkar Deshmukh[1], Sakshi Shejval[2], Ranjeet Sawant[3], Ganesh Solanki[4],

Prof. Nikhilesh Kumar Mishra[5]

[1,2,3,4]Student, of Information Technology, JSPM's Bhivarabai Sawant Institute Of Technology And Research, Wagholi, Maharashtra, India

[5]Assistant Professor, Dept. of Information Technology, JSPM's Bhivarabai Sawant Institute Of Technology And Research Wagholi, Maharashtra, India

Abstract: Vehicle detection and counting play a crucial role in enhancing security measures in various domains, including transportation, surveillance, and urban planning. This research paper provides a comprehensive overview of the state-of- the-art techniques, challenges, and applications related to vehicle detection and counting for security purposes. The paper explores various methodologies, such as computer vision, machine learning, and deep learning, employed in vehicle detection and counting systems. Additionally, it discusses real-world applications, including traffic management, border security, and smart city initiatives. The paper also delves into the challenges faced by existing systems and proposes potential solutions for improving accuracy, efficiency, and reliability.

Key words: Vechile Detection, Object Detection, Real Time Alerts And monitoring, Applications

1. INTRODUCTION

The provided document introduces a Real- time Moving Vehicle Detection, Tracking, and Counting System implemented with OpenCV, utilizing adaptive subtracted background technology, virtual detectors, and blob tracking techniques. Focused on enhancing traffic flow monitoring and control in Intelligent Transportation Systems (ITS), the paper emphasizes the significance of accurate vehicle detection, counting, and classification using image processing techniques. Addressing the challenges posed by the increasing number of vehicles on roads, the document explores methods such as background subtraction, thresholding, and morphological operations, achieving a remarkable accuracy rate of approximately 96%. It highlithe crucial role of real-time traffic information derived from computer vision technology, not only in detecting and counting vehicles but also in analyzing traffic patterns, enabling authorities to make informed decisions for efficient traffic management and road safety. The study underscores the importance of these advancements in managing traffic effectively.

2. LITREATURE REVIEW

The literature review presented in the paper provides a thorough analysis of existing research on vehicle detection and classification using image processing techniques. The authors have explored various methodologies, including machine learning-based classification, virtual loop- based vehicle counting, and automatic vehicle counting and classification using techniques such as background subtraction and morphological operations. These studies emphasize the significance of accurate data for efficient traffic management and highlight diverse methods, from machine learning models utilizing color, texture, and shape features to sophisticated deep learning techniques for vehicle classification.

The review also addresses challenges encountered in vehicle detection, such as low light conditions, and suggests potential solutions, including experimenting with different thresholds and parameters to enhance accuracy. While deep learning models have shown superior performance, the authors acknowledge the resource- intensive nature of these techniques. The comprehensive overview not only identifies the strengths and limitations of different approaches but also emphasizes the need for further research to refine existing methods and explore innovative solutions, ensuring advancements in the accuracy efficiency of vechicle detection and classification system.

3. EXISTING SYSTEM

3.1 Image Processing

- **Image Acquisition:**The process begins with capturing real-time traffic footage using CCTV cameras or video capturing devices strategically positioned along roads and intersections.These devices continuously acquire images or frames of the traffic flow, providing the raw data for analysis.

- **Image Analysis:**Background Subtraction: The acquired frames undergo background subtraction, a crucial step where stationary background elements are separated from moving objects (vehicles).

- **Region of Interest (ROI) Establishment:** Specific areas within the frames are designated as regions of interest, focusing the analysis on relevant parts of the image, such as lanes or intersections.

- **Thresholding**: A threshold is applied to distinguish vehicles from the background. Pixels with intensity values above the threshold are considered part of the vehicles, while the rest are treated as background noise.

- **Morphological Operations**: Techniques like erosion and dilation are employed to refine the binary image, filling gaps, removing noise, and enhancing object boundaries for accurate detection.

3.2 Object Detection:

- **Feature-Based Detection:** Object detection algorithms use distinctive features such as edges, corners, or color information to identify and locate vehicles within the defined regions of interest.

- **Blob Detection:** Connected components analysis is often applied to detect blobs (groups of adjacent pixels) representing vehicles, providing precise spatial information about each vehicle in the frame.

- **Counting:** Tracking Movement: The system tracks the movement of identified vehicles over consecutive frames, monitoring their trajectories.

- **Virtual Loops or Lines:** Virtual detection zones, represented as loops or lines, are established in specific areas of the frame. Each time a vehicle crosses these zones, a counter is incremented, indicating the passage of a vehicle.

- **Data Storage:** The counted data, including vehicle types and counts, is stored for further analysis and reporting.

3.3 Classification:

Further analysis of vehicle features, such as size, shape, and aspect ratio, allows the system to classify vehicles into predefined categories like bicycles, motorcycles, cars, buses, and trucks.

- **Machine LearningModels:** In more advanced systems, machine learning models trained on extensive datasets can be employed to improve the accuracy of vehicle.

- **Displaying Results:** Graphical Interface: The results of vehicle detection, counting, and classification are often displayed through user-friendly graphical interfaces. Graphs, charts, and real-time visuals provide immediate insights into traffic patterns.

4. PROPOSED SYSTEM

The proposed system aims to enhance security measures by implementing a real- time vehicle detection and counting system. Utilizing advanced computer vision techniques, this system will accurately identify and count vehicles, providing valuable insights for security monitoring, traffic management, and law enforcement.

4.1 System Architecture:

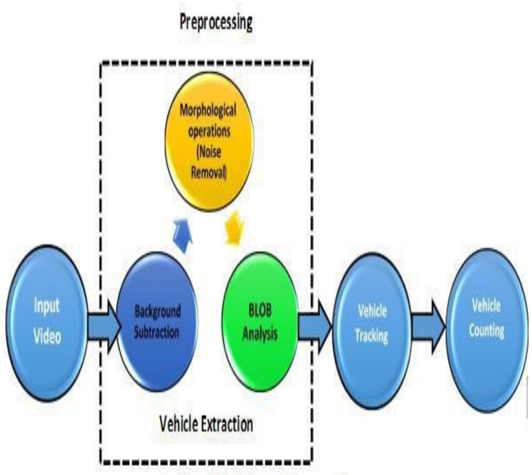

Fig.1 System Architecture

Resolution CCTV cameras at strategic locations, connected to a central processing Camera Network: Deploy high-unit.

A.Image Processing Unit:

- Process video feeds from cameras in real-time using **GPUs** and dedicated processors.
- Data Storage and Analysis: Store processed data in a secure cloud-based database for historical analysis, pattern recognition, and anomaly detection.

B.Vehicle Detection and Recognition:

- Object Detection: Employ deep learning models (e.g., YOLO, Faster R-CNN) for accurate vehicle detection in video frames.
- License Plate Recognition (LPR): Implement OCR-based LPR systems to read license plates, enabling vehicle identification.
- Color and Make Recognition: Develop algorithms for vehicle color and recognition for further make categorization.

C.Counting and Classification:
Vehicle Counting: Use virtual detection zones and line-crossing algorithms to count vehicles entering or exiting specific areas.
Vehicle Classification: Classify vehicles into categories (e.g., cars, trucks, motorcycles) using machine learning models trained on diverse datasets.

D.Real-time Alerts and Monitoring:
Anomaly Detection: Detect unusual patterns (e.g., wrong-way movement) and trigger immediate alerts to security personnel.
Live Monitoring: Provide a user-friendly dashboard for real-time monitoring, including live feeds, vehicle counts, and classifications.

E.Integration with Security Systems:
Access Control Integration: Integrate with access control systems to manage entry and exit based on vehicle recognition.
Alarm System Integration: Trigger alarms for unauthorized vehicles or suspicious activities detected by the system.

F. Data Analysis and Reporting:
Historical Data Analysis: Analyze historical data to identify traffic patterns and enhance security strategies.
Reporting: Generate detailed reports on vehicle counts, classifications, and security incidents for decision-making.

Enhancing Privacy and Compliance:
Privacy Measures: Implement privacy- conscious algorithms to anonymize personal information and comply with privacy regulations.
Data Encryption: Ensure data transmission and storage encryption to safeguard sensitive information.

G. Scalability and Future Expansion:

Scalability: Design the system to be scalable, allowing seamless integration of additional cameras and processing units as surveillance areas expand.

Research and Development: Continuously research emerging technologies to enhance accuracy and efficiency, incorporating edge computing and advanced machine learning model.

5. ALGORITHM

DEEP SORT:

The first step in Deep SORT involves extracting features from detected objects within each frame of a video sequence. This process, represented by the function $f(x)$, transforms each detected object x into a feature vector. Typically, this feature vector encodes relevant appearance information about the object, such as color histograms or features extracted from a convolutional neural network (CNN).

Deep SORT enhances the tracking process by introducing a learned metric to compute the similarity between pairs of detected objects based on their feature vectors. Denoted as $\text{sim}(a, b)$, this metric is learned using a deep neural network, often implemented as a siamese network. The similarity function $\text{sim}(a, b)$ computes the similarity score between two feature vectors a and b, effectively determining the likelihood that they represent the same object.

Following feature extraction and association metric computation, Deep SORT predicts the positions of objects in the current frame based on their positions in the previous frame. This prediction utilizes a state transition function derived from the Kalman filter framework. Mathematically, the predicted state of an object at time t, denoted as $\hat{x}_{t|t-1}$, is obtained using the equation $\hat{x}_{t|t-1} = F \cdot \hat{x}_{t-1}$, where F represents the state transition matrix.

Once predictions are made, Deep SORT performs a measurement update step to refine the state estimates based on the actual detections in the current frame. This update process employs a Kalman filter update equation to incorporate measurement information and correct state estimates. Specifically, the updated state estimate $\hat{x}_t$ at time t is computed using $\hat{x}_t = \hat{x}_{t|t-1} + K_t(z_t - H\hat{x}_{t|t-1})$, where K_t denotes the Kalman gain, z_t represents the measurement (i.e., the detected object position), and H is the measurement matrix.

These paragraphs elucidate the key components of the Deep SORT algorithm, delineating how feature extraction, association metrics, prediction, and measurement updates interact to facilitate robust multi-object tracking in video sequences.The accompanying mathematical expressions provide a formal representation of the underlying computations involved in each stage of the algorithm

6. RESULT

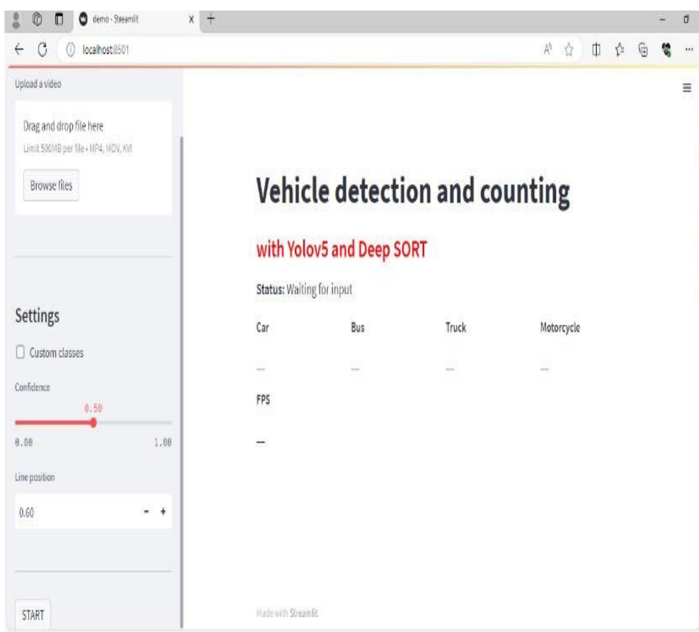

Fig.1 User Interface

The image is a screenshot of a graphical user interface showing an application related to Word processing. It seems to be a demo of an application called Streamlit. The interface includes options for uploading a video, vehicle detection and counting using Yolov5 and Deep SORT, settings for different vehicle classes, confidence level, frames per second, and line position. Additionally, the interface displays information about the software being used.

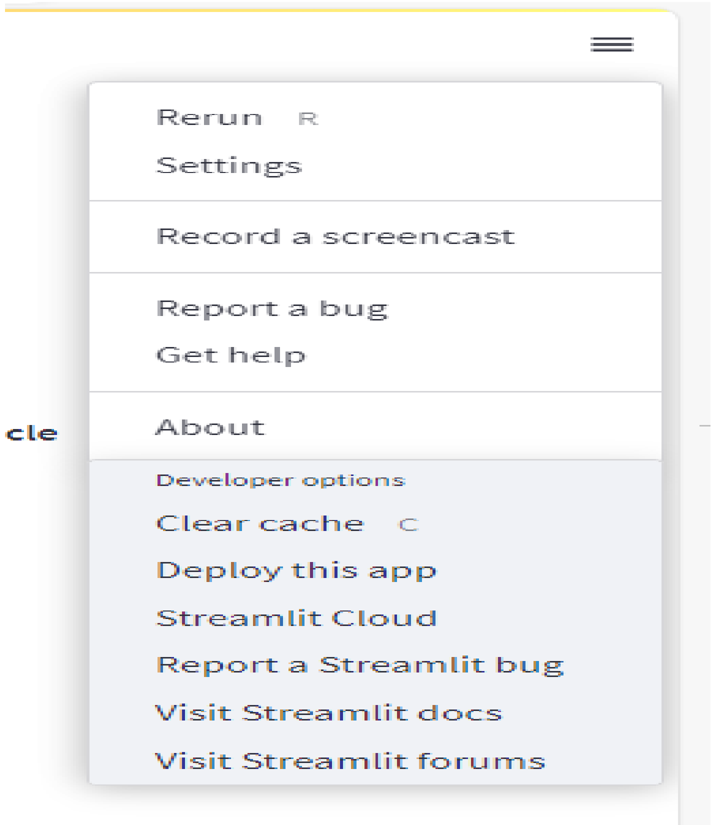

Fig.2 Tools

A graphical user interface displaying various options and settings for an application. The options include "Rerun R," "Settings," "Record a screencast," "Report a bug," "Get help," "About," "Developer options," "Clear cache C," "Deploy this app," "Streamlit Cloud," "Report a Streamlit bug," "Visit Streamlit docs," and "Visit Streamlit forums." The interface seems to be related to Streamlit based on the options provided.

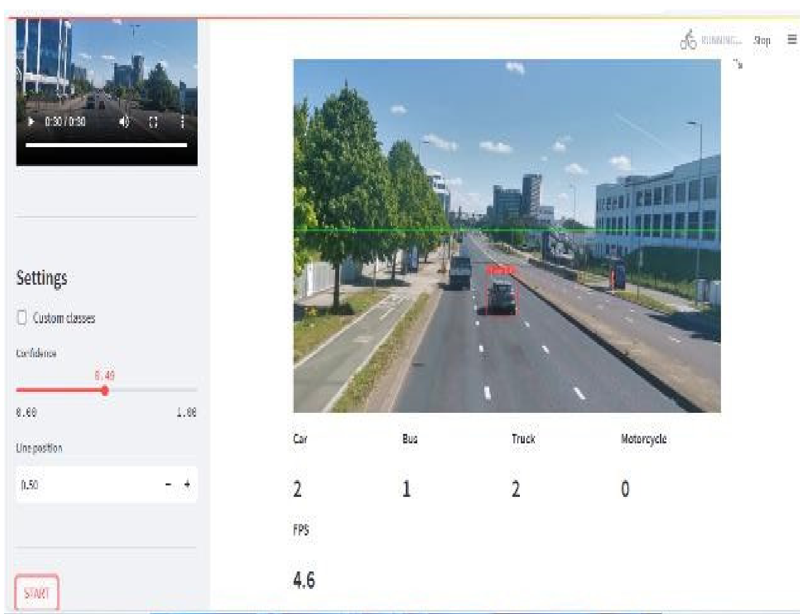

Fig.3 classification and counting of vehicle

The image includes information about running time, settings, custom classes, confidence levels for detecting objects like cars, motorcycles, buses, and trucks, line position, and frames per second. The tags associated with the image include text, screenshot, sky, outdoor, multimedia software, tree, software, road, thoroughfare, and street.

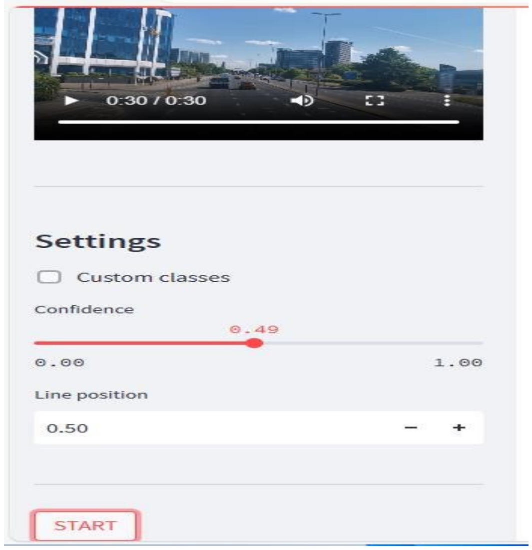

Fig.4 Frame settings

A graphical user interface displaying an application. It shows settings related to custom classes, confidence values, line position adjustment, and a start button. There is also a focus on the line position and confidence mechanism in the image.

7. CONCLUSION

The integration of advanced vehicle detection and counting systems offers significant advantages, including enhanced security, efficient traffic management, and data-driven urban planning. While privacy concerns, data security, high initial costs, and maintenance challenges are hurdles to overcome, the potential benefits outweigh these drawbacks. Responsible implementation, adherence to privacy regulations, and ongoing maintenance are crucial. With continuous advancements and responsible practices, these systems promise safer, more efficient, and secure environments for communities and cities.

REFERENCES

1. S.-Y. Cheung, and P.P. Varaiya, "Traffic surveillance by wireless sensor networks: Final report", PhD diss., University of California at Berkeley, 2006.
2. S. Oh, S. Ritchie, and C. Oh, "Real-time traffic measurement from single loop inductive signatures", Transportation Research Record: Journal of the Transportation Research Board, (1804), pp. 98-106, 2002.
3. B. Coifman, "Vehicle level evaluation of loop detectors and the remote traffic microwave sensor", Journal of transportation engineering, vol. 132, no.3, pp. 213-226, 2006.
4. M. Tursun, and G. Amrulla, "A video based real-time vehicle counting system using optimized virtual loop method", IEEE 8th International workshop on Systems Signal Processing and their Applications (WoSSPA), 2013.
5. M. Lei, D. Lefloch, P. Gouton, K. Madani, "A videobased real-time vehicle counting system using adaptive background method", IEEE International conference on Signal Image Technology and Internet Based Systems (SITIS'08), pp. 523-528, 2008.
6. N.C. Mithun, N.U. Rashid, and S.M. Rahman, "Detection and classification of vehicles from video using multiple time-spatial images", IEEE Transactions on Intelligent Transportation Systems, vol 13, no 3, p. 1215-1225, 2012.
7. G. Yang, "Video Vehicle Detection Based on SelfAdaptive Background Update", Journal of Nanjing Institute of Technology (Natural Science Edition), vol 2, p. 13, 2012.
8. M. Seki, H. Fujiwara, and K. Sumi, "A robust background subtraction method for changing background", Fifth IEEE Workshop on Applications of Computer Vision, 2000.
9. N. Friedman, and S. Russell, "Image segmentation in video sequences: A probabilistic approach", Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence, 1997, Morgan Kaufmann Publishers Inc.
10. C. Stauffer, and W.E.L. Grimson, "Learning patterns of activity using real-time tracking", IEEE Transactions on pattern analysis and machine intelligence, 2000. Vol 22, no 8, pp. 747-757, 2000.
11. C. Stauffer, and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking", IEEE Computer Society Conference Computer Vision and Pattern Recognition, 1999.