International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 12 | December - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Vehicle Tracking and Counting through Image Processing

Varun Teja. M
B. Tech
School of Engineering
Computer Science-(AI&ML)
Malla Reddy University, India

Vasanth Kalyan. C
B. Tech
School of Engineering
Computer Science-(AI&ML)
Malla Reddy University, India

Bandi Veera Venkata Satyanarayana
B. Tech
School of Engineering
Computer Science-(AI&ML)
Malla Reddy University, India

Veera Venkata Laxman. A
B. Tech
School of Engineering
Computer Science-(AI&ML)
Malla Reddy University, India

Veerabhadra Sai Amarnath. A
B. Tech
School of Engineering
Computer Science-(AI&ML)
Malla Reddy University, India

Guide: *Prof. D. Manikkannan*
Assistant Professor
School of Engineering
Computer Science-(AI&ML)
Malla Reddy University, India

1. ABSTRACT

Vehicle Tracking and Counting through Image Processing is an efficient system for real-time vehicle detection, classification, and counting using the powerful combination of YOLO and OpenCV. The System uses the application of machine learning in the domain of image processing for vehicle tracking and counting. OpenCV in this system is used to process the video feed or images and provide real-time visualization of the vehicles for detection. The system uses advanced computer vision techniques such as contour detection, image segmentation, and feature extraction to accurately locate and recognize individual vehicles in a scene. The detected vehicles are classified into various categories, including cars, trucks, motorcycles, and bicycles (Two & four wheelers) using the trained YOLO model. In addition to detection and classification the count of vehicles is also noted down. The primary objective of this system is to track the vehicles, distinguish and count them. This system can be used in – Traffic Signal Management and Parking Lot Monitoring.
Keywords: **-** YOLO, OpenCV, Python, CVAT

2. INTRODUCTION

The "Vehicle Tracking and Counting through Image Processing" project addresses the need for an advanced, automated system to enhance vehicle monitoring and management in urban environments. The "Vehicle Tracking and Counting through Image Processing" system utilizes OpenCV to achieve detection, classification, and counting of vehicles. Employing the YOLO model for optimal object detection, the system leverages advanced computer vision techniques such as contour detection and image segmentation. OpenCV processes video feeds or images, providing real-time visualization of vehicles. The Yolo model classifies detected vehicles into categories, including cars, trucks, motorcycles, and bicycles. Additionally, the system records the count of vehicles, it is applicable for Traffic Signal Management.

3. LITRATURE REVIEW

This paper explores the application of yolo for the vehicle detection and counting using the object detection in yolo which can be employed in many systems.
In this research, the integration of YOLO (You Only Look Once) and OpenCV has been explored for real-time vehicle tracking and counting through image processing. This study responds to the growing need for advanced surveillance systems in urban settings. The application of contour detection, image segmentation, and feature extraction in image processing has been investigated to precisely locate and identify vehicles in dynamic scenes.

The YOLO model, known for its efficient object detection capabilities, is employed for swift and accurate classification of vehicles. OpenCV, as a versatile tool in computer vision, facilitates real-time visualization of detected vehicles. The proposed system finds practical applications in traffic signal management and parking lot monitoring, contributing to data-driven decision-making in urban planning and management. The literature underscores the significance of advanced tracking systems for optimizing traffic flow and enhancing parking space utilization. Overall, the integration of YOLO and OpenCV emerges as a pivotal advancement, demonstrating its potential in diverse applications within the domain of vehicle tracking and counting through image

In addition to its real-time capabilities, the proposed system stands out for its adaptability to various urban scenarios, offering a scalable solution for comprehensive vehicle monitoring. Studies highlight the system's effectiveness in accurately counting vehicles, providing valuable insights for traffic management strategies

3.1 Existing System:

- Manual Observation: In many cases, vehicle tracking and counting are performed manually by human operators.
- Simple Sensors: Some existing systems use basic sensors like inductive loops or infrared sensors to detect the presence of vehicles.
- Limited Automation: Existing system is typically basic and lack the sophistication needed for real-time, accurate vehicle tracking and classification.
- Lack of Object Classification: Traditional systems do not have the capability to classify vehicles into specific categories such as cars, trucks, motorcycles, and bicycles.

3.2 Proposed System:

- Real-Time Object Detection: The proposed system employs YOLO, a powerful object detection model, for vehicle detection. This eliminates the need for manual observation and enables the system to process video feeds or images rapidly.
- Advanced Computer Vision Techniques: The system utilizes advanced computer vision techniques, including contour detection, image segmentation, and feature extraction, to accurately locate and recognize individual vehicles in a scene.
- Object Classification: The proposed system goes beyond basic detection and introduces vehicle classification. It can distinguish between various types of vehicles, including cars, trucks, motorcycles, and bicycles, providing more detailed insights into traffic composition.

4. PROBLEM STATEMENT

The urban landscape faces a pressing challenge in effective and automated vehicle monitoring, tracking, and counting. Current surveillance systems lack the sophistication needed to handle the complexities of dynamic traffic scenarios, leading to inefficiencies in urban planning and management.

4.1 Data Set Descriptions:

For our "Vehicle Tracking and Counting through Image Processing" project, we dedicated significant effort to building a robust dataset, crucial for effective vehicle detection and classification. Our dataset comprises a substantial number of vehicle images, each annotated to optimize training outcomes. These annotations, specific to various vehicles, streamline the classification process.

a) Training Data: This has diverse images, each tagged with vehicle bounding boxes that is annotated with CVAT. It's like teaching our model to recognize cars, trucks, and bikes in all kinds of settings – day, night, heavy traffic. We grabbed these snapshots from public places on the internet, making sure our model gets a taste of real-life scenarios.

b) Testing Data: We handpicked tricky images it had never seen before – like a surprise test for our model. Some were from public datasets, and we even threw in some tricky situations to see if it could handle the pressure.

c) Validation Data: It's a carefully chosen subset from our dataset, like giving our model a final polish. Used this to trin the model more efficiently

5. METHODOLOGY

For the Vehicle tracking and counting through Image Processing the yolo and the OpenCV are used to make the tracking and counting possible. With this many advancements in the traffic management can be done and the data about a particular areas traffic count can be gathered easily.

5.1 Yolo model:

YOLO (You Only Look Once) is a state-of-the-art object detection algorithm renowned for its efficiency and speed. It revolutionizes object detection by dividing an image into a grid and predicting bounding boxes and class probabilities within each grid cell simultaneously. This single-pass approach enables YOLO to deliver real-time object detection with remarkable accuracy. In the context of the proposed vehicle tracking and counting system, YOLO plays a pivotal role. Utilizing its rapid object detection capabilities, YOLO precisely classifies vehicles into distinct categories, including cars, trucks, motorcycles, and bicycles. This integration enhances the system's ability to provide timely and accurate data for vehicle tracking and counting in urban environments, contributing to optimized traffic signal management and parking lot monitoring. Yolo can also be used for various other purposes as we can train it to detect almost everything this was the main reason for selecting yolo as it is fast and processes the data efficiently. Open cv combined with yolo is a perfect combination for the object detection.

5.2 Modules

- Video Input Module: This module handles the input source, which can be a live video feed or pre-recorded video files. Using the open cv we take the Input and then process the video to separate the frames.

- Training Module: In this module the images are inputted into the training model which is yolo and the training is done here the output of this module is a trained model which was utilized in the program for object detection

- Image Processing Module: The Image Processing Module is the heart of the system, where OpenCV is employed for various tasks such as image segmentation, contour detection, and feature extraction.

- Object Detection Module: Utilizing the YOLO model, the Object Detection Module is responsible for real-time identification and localization of vehicles within the processed images or video frames. Here the images extracted from the video are processed to detect the objects using the trained model.

- Counting Module: The Counting Module automatically tallies the number of vehicles identified and classified. For this count we added a line to the video if the object touches the line the count of the vehicle images through this, we have achieved the counting

5.3 Architecture:

The architecture of the Vehicle Tracking and Counting through Image Processing

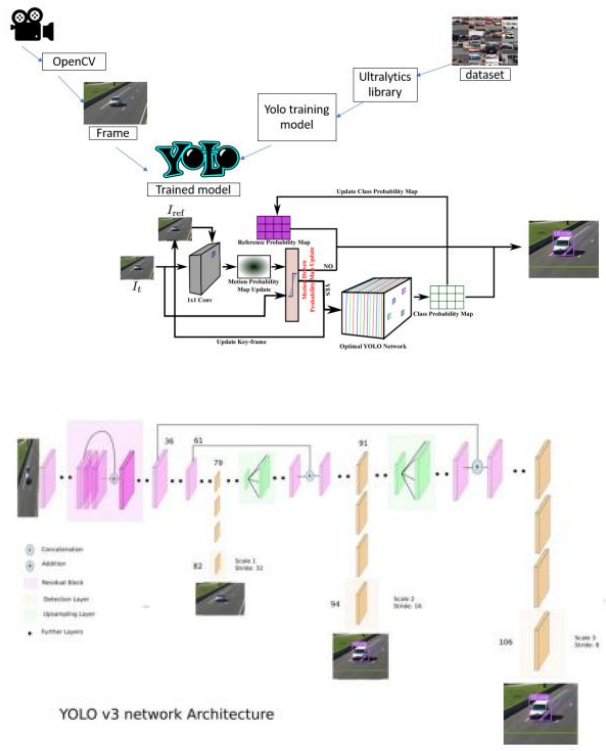

Figure 1: Architecture diagram of YOLO (Components Overview)

5.4 Methods used:

a) Object Detection using YOLO: We've adopted a real powerhouse for object detection – YOLO, or You Only Look Once. It's like the speed demon of our system. YOLO processes entire images in a single pass, zipping through them without the need for complicated multistep detection processes. We used the pre-trained weights and configurations to integrate it into Yolo and feed the input to the model we used the command model_path_obj = os.path.join(".", "runs", "detect", "train", "weights", "best.pt") to input the trained model into the program.

b) Advanced Computer Vision Techniques: We've used OpenCV's library to implement advanced computer vision techniques, using techniques such as contour detection, image segmentation, and feature extraction. These contribute to better training of the data by yolo in ultralytcs library through which it can easily detect vehicles within a scene. We used CV to get the Frames from the video and give it to the trained model and after that we also used it to join the segments together. CV was also used for preprocessing tasks – Image resizing, Normalization using min max scaling.

c) Automatic Vehicle Counting: The spotted vehicles are shown with bounding boxes now we have integrated a line in the frames using python open cv whenever the bounding box of the vehicle touches the line created by the program the count of the particular vehicle class is increased and displayed on the screen. The secret sauce here is a customized counting module. It's seamlessly integrated into the system architecture, ensuring not just accuracy but also efficiency in tallying up the vehicles.

5.5 Data Preprocessing Techniques:

To optimize our dataset for effective model training, we employ the following preprocessing techniques:

a) Image Resizing: We standardized image dimensions using OpenCV, ensuring computational efficiency by maintaining a consistent resolution across the dataset. As the data set has many images with different dimensions, we resized them to have same size.

b) Bounding Box Normalization: To precisely interpret object locations, we applied bounding box normalization, adjusting coordinates to a standardized format during preprocessing. Annotated using CVAT.

c) Color Normalization: To reduce sensitivity to lighting variations, we standardized color channels across images using color normalization techniques. We did this using min max scaling in Open CV min = np.min(images, axis=(0, 1, 2)) max= np.max(images, axis=(0, 1, 2))

d) Data Splitting: For effective evaluation and training, we strategically split the dataset into training, testing, and validation sets, ensuring diverse representation in each subset.

5.6 Model development:

The process of developing and training the models for our "Vehicle Tracking and Counting through Image Processing".

a) Image Preprocessing: We need to get our images ready. We take the images from our dataset and annotate them

using a tool called CVAT. It's where we add annotations to help the model understand and learn.

b) Training Process: We take our YOLO model, put it through its paces using a diverse dataset, making it a versatile vehicle identifier. We validate its performance using precision, recall, Confusion matrix –it checks how well the model learned to recognize vehicles in different scenarios.

c)Yolo Model Development: We developed a training module using the ultralytics library. Here, we use those annotated images we prepared earlier. It teaches YOLO to recognize vehicles by showing it what they look like. We feed those annotated images, and the magic begins – it starts learning to identify vehicles. We set the number of times the model trains on the images which is called epochs the more the number the better the result it is some time taking process.

d)Integration Of OpenCV Techniques: OpenCV was used in our project to enhance YOLO's abilities. OpenCV uses contour detection, image segmentation, and feature extraction to make sure YOLO is seeing things crystal clear. OpenCV takes frames from our input video, sending them for processing.

e) Customized Counting Module: The frames we get from OpenCV are snapshots of our traffic scene. The counting module steps in here. It takes those frames, uses the trained YOLO model, and starts detecting vehicles. Whenever the detected vehicle touches the pre placed line it increases the count.

6. EXPERIMENTAL RESULTS

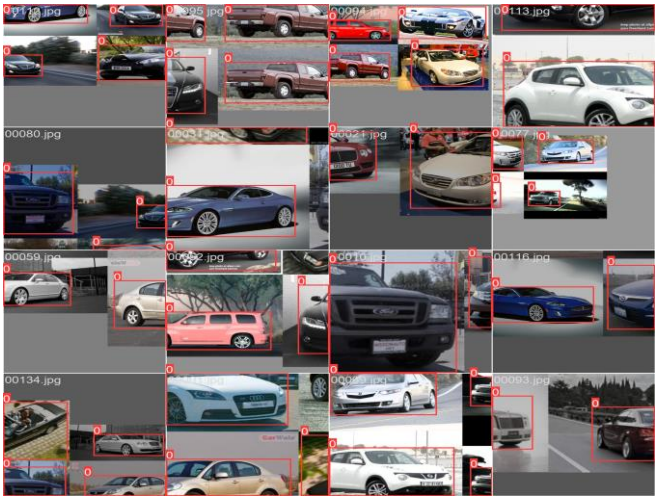

Figure 2: Train Batches

The above is an example of one in many train batches which were used to train the yolo model. The yolo automatically selects the batches and trains the model on one batch at a time

Figure 3: Input

Footage of a road side cc camera was used to test the model and it was inputted into the model using the CV module through this we send the data to the processing and the output is given

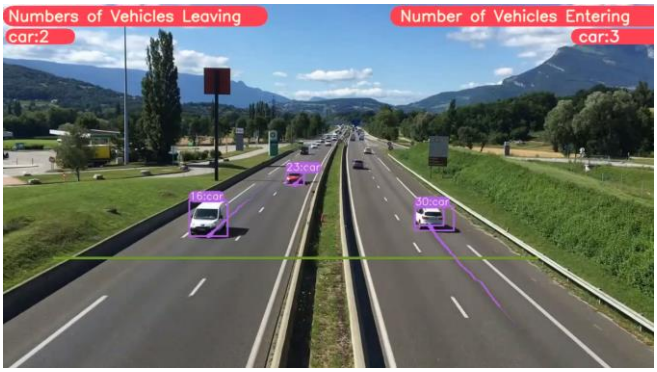

Figure 4: Output

This is the output of the model where the cars are detected and the number of vehicles is noted using the counter module

5.6 Model Evaluation Metrics:

a) Precision, Recall: Here the precision and recall of the model are checked for this model we got a precision recall of 61% It's a delicate balance, making our model both accurate and comprehensive.

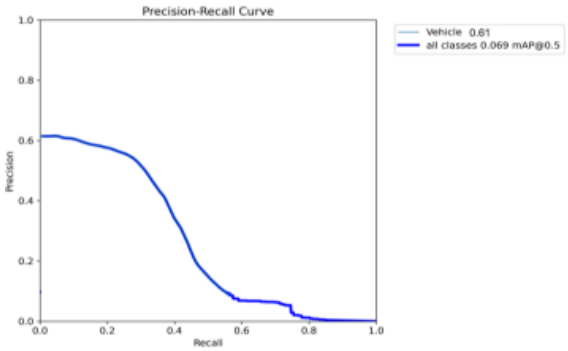

Figure 5: Precision-Recall Curve

b) IoU (Spatial Alignment): We measured the spatial alignment between predicted and actual

bounding boxes. It ensures our model has a keen eye for pinpointing objects accurately. These are all done during the training process.

c) ***mAP (Detection Summary):*** This was used to find the model's detection prowess across varying confidence thresholds. It's a quick overview, providing insights into precision under different confidence levels. It's the speedometer for our model's accuracy journey.

d) ***Confusion Matrix: The Detective's Breakdown:*** It's the detailed breakdown – true positives, true negatives, false positives, and false negatives. The output has shown how many times the model has given the correct answer and founded the object correctly it was correct 61% of the times. This comprehensive view helps in making us understand better about the model.

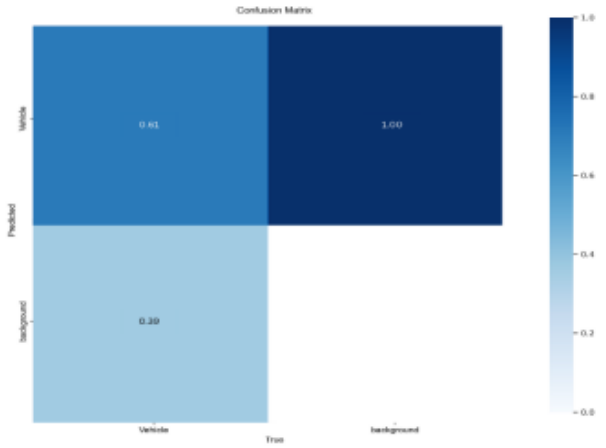

Figure 6: Confusion Matrix

7. CONCLUSIOON

In concluding our "Vehicle Tracking and Counting through Image Processing" project, we affirm the successful integration of advanced computer vision techniques and machine learning models, particularly YOLO, to achieve real-time vehicle detection, classification, and counting. The project's primary objectives of enhancing traffic monitoring, management, and analysis have been met through the seamless collaboration of these technologies. Our project showcases a robust and efficient system that addresses real-world challenges in traffic surveillance. By leveraging the YOLO model and OpenCV techniques, we have developed a solution capable of accurately identifying and counting vehicles in diverse scenarios. The integration of a customized counting module further enhances the system's utility for applications such as Traffic Signal Management.

8. FUTURE WORK

Vehicle classification algorithms: Further refinement of vehicle classification algorithms can provide more detailed insights into vehicle composition, improving the system's applicability.

Sensors and other data: Combining image processing with data from other sensors, such as radar or lidar, could enhance the system's accuracy and reliability, especially in challenging environmental conditions.

Scaling: The system can be scaled for deployment in larger urban environments, contributing to smart city initiatives for efficient traffic management.

9. REFERENCES

[1] YOLO Algorithm for Object Detection
https://www.analyticsvidhya.com/blog/2022/06/yolo-algorithm-for-custom-object-detection/

[2] OpenCV-python
https://pypi.org/project/opencv-python/

[3] Introduction to Image Annotation
https://www.superannotate.com/blog/introduction-to-image-annotatioon

[4] What is yolo algorithm?
https://www.youtube.com/watch?v=ag3DLKsl2vK

[5] Vehicle Dataset
https://www.kaggle.com/datasets/jessicali9530/stanford-cars-dataset/