IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 13, Issue 4, April 2025

DOI: 10.17148/IJIREEICE.2025.13426

VEHICLE DETECTION AND TRACKING SYSTEM

K. ARUNTHATHI[1], Dr. R. SENTHIL KUMAR[2]

Student, Dept of Computer Science with Cognitive Systems, Dr. N.G.P Arts and Science College, Coimbatore, India[1]

Professor, Dept of Computer Science with Cognitive Systems, Dr. N.G.P Arts and Science College, Coimbatore, India[2]

INTRODUCTION

1.1 ABSTRACT

This project focuses on vehicle detection and counting for traffic control, a crucial task for highway regulation. Detecting and counting vehicles accurately is challenging due to their varying structures. The paper presents a video-based approach using OpenCV technologies, specifically background subtraction to identify moving vehicles. It also applies techniques like thresholding, adaptive morphology operations, and hole filling to improve detection accuracy. Vehicle counting is performed using virtual identification zones. The experimental results show that the proposed system achieves an accuracy of about 96%.

1.2 OBJECTIVES

This project focuses on vehicle detection, classification, tracking, and counting using deep learning techniques for traffic monitoring. Vehicles are detected in traffic video frames using Convolutional Neural Networks (CNN), which classify vehicles into categories like cars, trucks, buses, etc. To track multiple vehicles, the Hungarian algorithm is applied, linking detected vehicles across frames. The project also discusses the use of Transfer Learning with pre-trained models to overcome the challenge of limited datasets for vehicle classification, improving accuracy without requiring extensive training. The deep learning models, such as Faster RCNN and CNN, help in detecting, classifying, and tracking vehicles efficiently for traffic management.

2.SYSTEM STUDY

2.1 EXISITING SYSTEM

- Manual counting by human observers
- Labor-intensive and prone to errors
- Limited accuracy and scalability
- High operational costs
- Inability to process complex traffic scenarios

2.2 PROPOSEDSYSTEM

- Automated vehicle detection using deep learning algorithms
- Real-time processing of video streams
- Multi-object tracking and classification
- High accuracy (up to 96% detection rate
- Flexible deployment across various traffic scenarios

2.3 EXPECTED OUTPUT

- Precise vehicle count
- Vehicle type classification (cars, trucks, motorcycles)
- Real-time traffic flow analysis
- Directional movement tracking (up/down)
- Detailed traffic pattern visualization

DOI: 10.17148/IJIREEICE.2025.13426

3.SYSTEM SPECIFICATION

3.1 HARDWARE SPECIFICATIONS

- **Processor (CPU)**: Intel i5/i7 or AMD Ryzen 5/7
- **Graphics (GPU)**: NVIDIA GTX 1650+ (for deep learning)
- **RAM**: 8GB+ for optimal performance
- **Storage**: 256GB SSD for speed and reliability
- **Camera**: 1080p or higher for best detection accuracy

3.2 SOFTWARE REQUIRMENTS

- **platform**: Windows or macOS (Windows 10 or later)
- **Recommended Version**: Python 3.7 or higher
- **back- end**:Python

3.3 SOFTWARE DESCRIPTION

- **Programming Language** : Python
- **Computer Vision Library** : OpenCV
- **Detection Technique** : Object detection using YOLO (You Only Look Once) algorithm

Methodology Overview

1. Project Setup
2. Data Acquisition
3. Preprocessing
4. Vehicle Detection
5. Vehicle Counting
6. Analysis and Visualization
7. Testing and Validation

SYSTEM DESIGN

4.1 DATA FLOW DIAGRAM

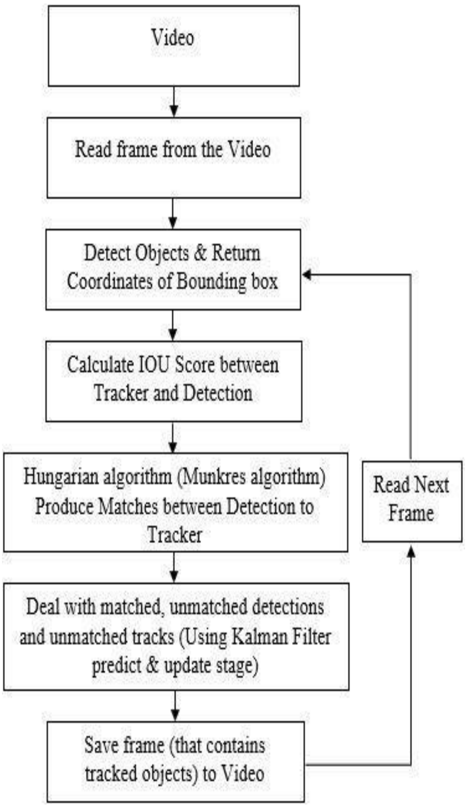

Fig:1

DOI: 10.17148/IJIREEICE.2025.13426

5. SYSTEM IMPLEMENTATION

The experiment includes the study of vehicle detection using a popular pre-trained model. Input images to these models are taken from the IITM-Hetra [6] dataset and AAU-Rainsnow[22] dataset contain 22 videos of the road with categories of vehicles such as a car, bus, truck, bicycle.

The experiment is implemented using Google Colab and Spyder with Tensor flow and Keras as the main deep learning library. The output obtained from models are in the form of, confidence values as score and class type and it returns only those class names and score values whose confidence score is greater than 70% in the final output.

Vehicle tracking and counting are implemented using Keras Tensorflow library on the python environment. The experiment setup uses a pre-trained model for vehicle detection, the Hungarian algorithm used for determining the association between detection to tracker, and CNN used for vehicle tracking. The vehicle tracking and counting will track both moving and stationary objects and if the vehicle is not detected properly then it results in false counting.

6.CONCLUSION

The detection and classification of vehicles are performed using CNN deep learning algorithm. The different types of vehicles are easily detected and classified using the CNN algorithm. The Vehicle Tracking and Counting task are performed using the Hungarian algorithm and CNN algorithm. The Hungarian algorithm is used to determines whether the vehicle detected in the previous frame is the same as the vehicle detected in the current frame or not. The CNN is used to predict and update the state. The performance of the vehicle tracking and counting is tested using IITM-Hetra and AAU-Rainsnow dataset and found that it gives better results of tracking and counting.

Transfer Learning is also performed to detect the auto class. To improve the result of transfer learning, some experiments are performed by changing the batch size.
In the future, try to improve the model so it will able to track all the vehicles and count the vehicles accurately.

7.SCOPE FOR FUTURE ENHANCEMENT

1. Advanced Vehicle Detection Models

While Haar cascades are a good starting point for vehicle detection, they have limitations in terms of accuracy and performance in complex scenarios. You can improve detection by implementing more advanced deep learning-based models like:

YOLO (You Only Look Once)

- **YOLO** is a popular real-time object detection algorithm that is fast and accurate.
- It detects vehicles and other objects in real time, giving you bounding boxes around each detected vehicle.
  - **Implementation**: You can use the **YOLOv4** or **YOLOv5** models, which are optimized for better accuracy and speed.
  - **Benefits**: YOLO can detect various types of vehicles (cars, buses, trucks, motorcycles, etc.) with high accuracy and can be used for real-time vehicle counting.
  - **Benefits**: This model can be used when accuracy is a priority over speed, such as for post-processing videos with multiple frames.

2. Vehicle Classification

Enhance vehicle detection by also classifying vehicles based on their type (e.g., car, truck, bus, motorcycle). This can be useful in traffic analysis or parking systems.

2.1 Vehicle Type Classification using CNN (Convolutional Neural Networks)

- Implement a **Convolutional Neural Network (CNN)** to classify detected vehicles into categories (car, truck, motorcycle, etc.).
  - **Implementation**: Use a **pre-trained CNN** model like **MobileNet**, **ResNet**, or **VGG16**. Fine-tune the model with vehicle-specific datasets.
  - **Benefits**: Classifying vehicles allows for more detailed analysis, like differentiating between light and heavy vehicles in a traffic monitoring system.

2.2 License Plate Recognition (LPR)

- **License Plate Recognition (LPR)** can be added for applications like parking management, toll systems, or law enforcement. By detecting and reading license plates, you can keep track of individual vehicles.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2025

DOI: 10.17148/IJIREEICE.2025.13426

- ○ **Implementation**: Use an LPR algorithm like **OpenALPR** or train your own using deep learning models to extract license plate data.

REFERENCES

[1]. D. Mittal, A. Reddy, G. Ramadurai, K. Mitra, and B. Ravindran, “Training a deep learning architecture for vehicle detection using limited heterogeneous traffic data,” in *2018 10th International Conference on Communication Systems Networks (COMSNETS)*, Jan. 2018, pp. 589–294, doi: 10.1109/COMSNETS.2018.8328279.

[2]. Introduction to Object Detection Algorithms, *Analytics Vidhya*, Oct. 11, 2018. https://www.analyticsvidhya.com/blog/2018/10/a-step-by-step-introduction-to-the- basic-object-detection-algorithms-part-1/ (accessed Jul. 11, 2020).

[3]. S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[4]. K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Oct. 2017, pp. 2980– 2988, doi: 10.1109/ICCV.2017.322.

[5]. “The PASCAL Visual Object Classes Challenge 2012 (VOC2012).” http://host.robots.ox.ac.uk/pascal/VOC/voc2012/ (accessed Jun. 01, 2020).

[6]. Wu, Y., & Nevatia, R. (2006). *Detection and tracking of vehicles in highway and suburban environments*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1434-1441.

APPENDIX

A. SCREEN SHOTS

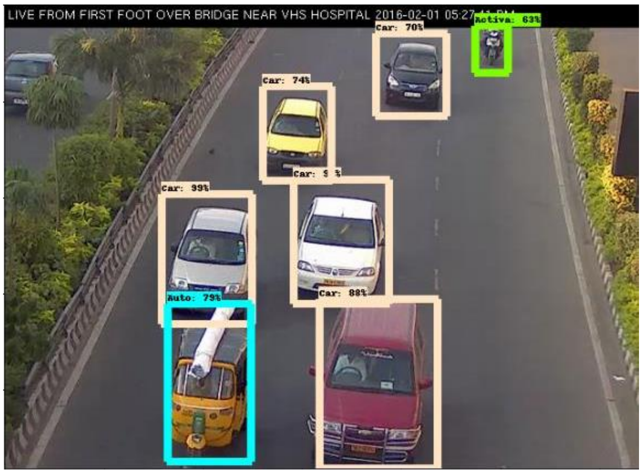

FIG:2

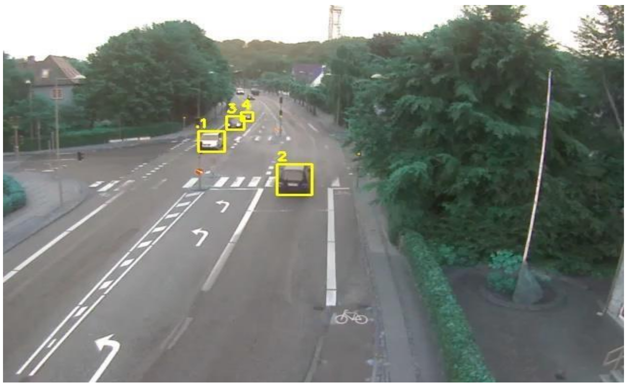

FIG:3

