IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Real Time Vehicles Detection And Speed Measurement At Traffic Signals

D . Chandragowda [1], D . Hemantharaja[2], .B.S.Prithvi [3], Srujan A.C [4] , Sapna Patil [5]

[1 - 4] Students, Electronics and Communication Engineering, K. S. Institute of Technology, Bangalore, India.
[5]Assistant Professor, Electronics and Communication Engineering, K. S. Institute of Technology, Bangalore, India

Abstract: The number of vehicles on the road is increasing very fast, so we need smart systems to monitor traffic in real time. This paper explains a method for Real-time Vehicle Detection and Speed Estimation using machine vision. The system can automatically detect, track, and calculate the speed of vehicles without using costly sensors like IR sensors. The process has a few main steps: detecting the vehicle, tracking its movement, and then calculating its speed. The speed is found by measuring how far the vehicle moves in pixels over time and converting that into real distance using a calibration factor. This system is reliable, scalable, and low- cost, making it useful for traffic police, road safety, and intelligent transportation systems (ITS). It shows that machine vision can replace traditional sensors in real traffic monitoring.

Index Terms - Manual entry, Radio Frequency Identification (RFID), reliable, efficient, simple design, low cost.

1. INTRODUCTION :

Real-time vehicle detection and speed measurement play an important role in traffic monitoring and road safety. With the increase in vehicle movement on roads, it has become necessary to identify overspeeding vehicles and control accidents. This project presents a simple and effective system that uses two IR sensors and an Arduino Uno to measure the speed of vehicles as they pass through a fixed path. When a vehicle crosses the first sensor and then the second sensor, the system calculates the time taken and uses it to determine the speed. The measured speed is displayed on an LCD screen, and a buzzer alert is activated if the vehicle exceeds the preset speed limit. This model provides a basic yet reliable demonstration of how electronic sensors and microcontrollers can be used for traffic management and safety applications.

2. LITERATURE REVIEW

Wei Hou et al. [1] developed a vehicle detection and tracking system utilizing the OpenCV library with Gaussian Background Modeling. The background model helps identify moving vehicles by subtracting static scene components. The Continuously Adaptive Mean Shift (CAMSHIFT) algorithm was implemented to handle challenges such as object deformation and partial occlusion. The experiments indicated that a detection threshold of 20 produced the most accurate results. However, this approach remains sensitive to environmental variations, lighting fluctuations, and camera vibrations.

Dongyang Zhao et al. [2] introduced an aerial video-based system using YOLOv3, a state-of-the-art deep learning object detection network, to identify and track vehicles from drone footage. A tracking-by-detection method was used to estimate the vehicles' speeds by mapping pixel displacement to real-world distances via an exponential relationship. Although the system achieved high detection accuracy, the nonlinear pixel-distance mapping introduced noticeable estimation errors. The need for high-performance computing and camera calibration further limited its applicability for large-scale deployments.

Genyuan Cheng et al. [3] enhanced background modeling through a K-Nearest Neighbor (KNN) approach, which dynamically updates the background based on pixel intensity values. The authors applied centroid distance matching for tracking and speed estimation, yielding an approximate 5% relative error. This approach met real-time operational requirements but still suffered from issues such as shadow interference and perspective distortions on inclined camera angles.

S. M. Sunny et al. [4] integrated accident detection, vehicle tracking, and number plate recognition in a single framework. The system utilized the DenseNet architecture for accident recognition and OCR (Tesseract) for license plate extraction, achieving nearly 80% accuracy in accident detection scenarios. Their implementation demonstrated how a unified vision-based system could be used for both enforcement and emergency response applications.

Shengnan Lu et al. [5] proposed a highly accurate shadow-based speed estimation system that eliminates traditional feature-tracking dependence. Instead, it analyzes differences between consecutive frames to determine motion and uses vehicle shadows as reliable tracking references. By correlating the shadow displacement across frames, the system achieved an average error of 0.3 km/h with 99.4% of estimations within ±2 km/h. While the accuracy is remarkable, the method is highly dependent on consistent lighting conditions and clear shadow visibility, limiting performance during nighttime or low-light conditions.

C. Ranjeeth Kumar et al. [6] developed a hybrid classification framework combining Bat Optimization, Support Vector Machine (SVM), and Enhanced Convolutional Neural Network (ECNN). This model optimized the selection of Haar-like features, effectively reducing redundant data and enhancing vehicle classification accuracy. The hybrid ECNN-SVM approach achieved 93.63% accuracy, surpassing traditional CNN (84.88%) and CNN-SVM (89.08%) models. This highlights the advantage of combining machine learning and deep learning to improve robustness in complex traffic scenarios.

Keyou Guo et al. [7] proposed a multi-scale feature fusion technique to improve detection accuracy across varying object sizes. By incorporating adaptive learning rate adjustment, the authors replaced fixed learning rates during model training, improving convergence speed and overall model stability. The improved Single Shot MultiBox Detector (SSD) achieved a detection latency of 55.6 ms, outperforming conventional models and enabling real-time detection in high-traffic environments.

Suraiya Parveen et al. [8] presented a motion detection model designed using Python and OpenCV. Their framework allows users to specify a Region of Interest (ROI) for targeted detection and applies image differencing and thresholding for vehicle movement analysis. The system is efficient and cost-effective for small-scale monitoring but lacks robustness for multi-vehicle and high-speed environments.

Shriharsha S. Venia et al. [9] designed a video-based vehicle detection, counting, and classification system using OpenCV. The system achieved a classification accuracy of 93% and processed frames at 25 FPS, proving its efficiency for real-time deployment on surveillance cameras.

Valanukonda Lakshmi Padmini et al. [10] extended traffic monitoring research by developing a machine learning–based helmet detection system for two-wheeler riders. The model achieved 87.6% accuracy and contributes toward improving road safety compliance and enforcement in smart city applications.

3. SYSTEMS DESIGN AND IMPLEMENTATION

The Dynamic Wireless Charging System (DWCS) allows electric vehicles to receive power while moving. The system works on Inductive Power Transfer (IPT), where energy is transmitted from coils embedded under the road to a receiving coil installed beneath the vehicle. The goal is to deliver continuous charging during motion without requiring the vehicle to stop or plug in.

3.1 . BLOCK DIAGRAM

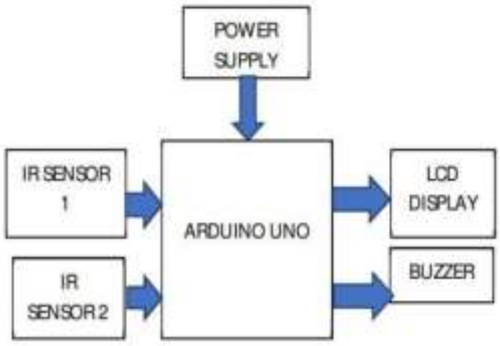

Figure 3 .1 .1. Block Diagram of Real Time Vehicles detection and Speed Measurement at Traffic Signals

- The real-time vehicle detection and speed monitoring system is built using an Arduino Uno, which acts as the main controller that connects and manages all other components.
- A regulated DC power supply is used to provide stable power to the Arduino and the connected modules, ensuring accurate sensor readings and smooth operation.
- The system uses two IR sensors placed at a fixed distance from each other to detect the movement of vehicles.
- The first IR sensor senses when a vehicle enters the detection area, and the second IR sensor detects when it crosses the next point. Based on the time taken between these two detection points, the Arduino calculates the vehicle's speed using the basic formula of distance divided by time.
- The calculated speed is then shown on the LCD display so that users can easily see the results in real time.
- If the vehicle crosses the set speed limit, the Arduino activates a buzzer that alerts about over speeding.
- This makes the system simple, effective, and useful for monitoring vehicle speed at traffic signals, road intersections, and other safety-critical areas.

3.2 FLOW CHART

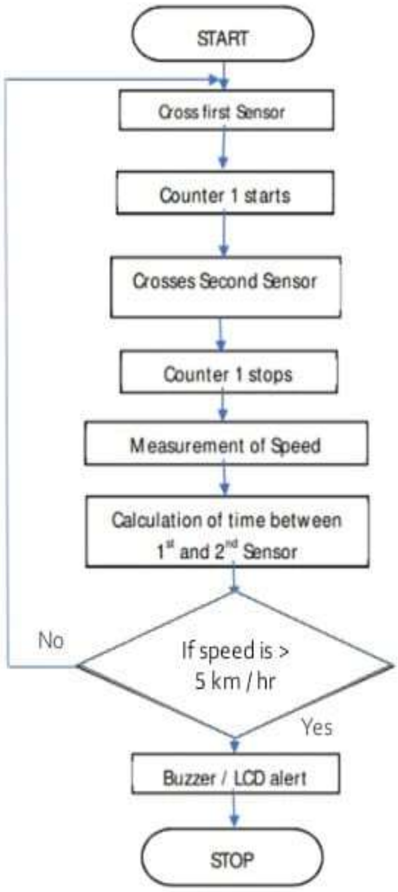

Figure 3.. 2. 2 Flow Chart Of Real Time Vehicles detection and Speed Measurement at Traffic Signals

- The working process of the system takes place in a clear and continuous sequence.
- When the system is switched on, the Arduino sets up both IR sensors and resets all timers and counters to zero. After initialization, the sensors start monitoring the area to detect any approaching vehicle.
- When a vehicle crosses the first IR sensor, the infrared beam is interrupted, and this immediately signals the Arduino to start an internal timer.
- This marks the exact moment the vehicle enters the detection zone.
- The timer keeps running as the vehicle moves forward. When the vehicle reaches and crosses the second IR sensor, the beam is broken again. This event tells the Arduino to stop the timer.
- The difference between the start time and stop time gives the total time taken by the vehicle to travel the fixed distance between the two sensors.
- Using this time, the Arduino calculates the speed of the vehicle using the simple formula: speed equals distance divided by time. After calculating the speed, the system compares it with the preset speed limit, such as 5 km/h.
- If the vehicle's speed is higher than the limit, the system marks it as an overspeed case. In such situations, the Arduino activates the buzzer to produce an audible warning and also displays the vehicle's speed along with an overspeed message on the LCD screen.
- If the speed is within the limit, no alert is triggered, and the system simply prepares for the next vehicle. Once the process is completed, the system resets itself automatically, clears all previous readings, and returns to its initial monitoring state. This allows the system to operate continuously without the need for manual control.

4. RESULTS

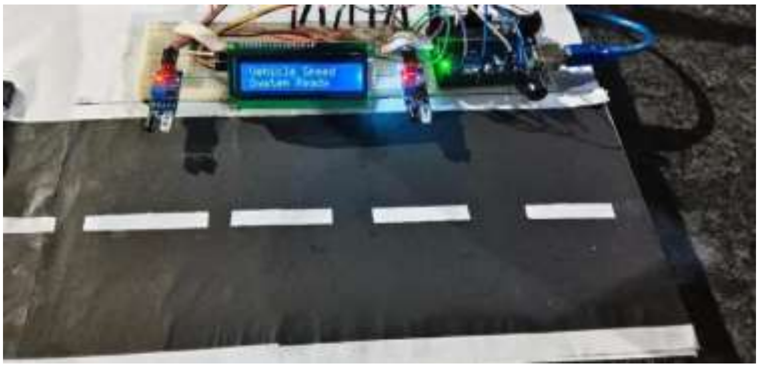

Figure 4.1 : Construction on the road part

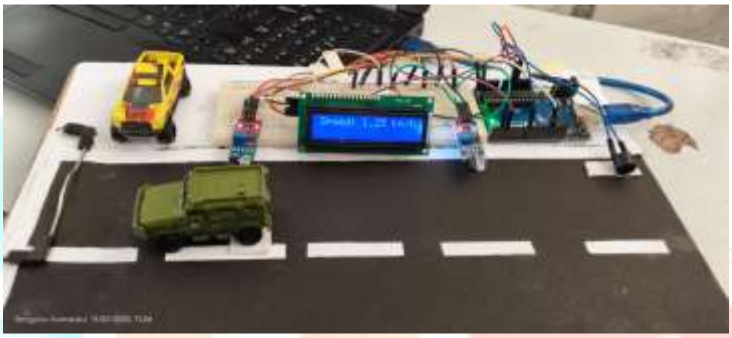

Figure 4 .2 : Vehicle detection of speed

The model shown in the picture successfully detects the vehicle and displays its speed on the LCD screen. As the toy car passes through the two IR sensors placed on the road setup, the Arduino measures the time taken and calculates the speed. The LCD display shows a reading of **"Speed: 1.25 km/h"**, which means the system is working correctly. All components, including the IR sensors, Arduino board, LCD, and buzzer, are properly connected on the breadboard. The setup clearly demonstrates real-time vehicle detection and speed measurement in a small working model.

5. CONCLUSION AND FUTURE SCOPE

The real-time vehicle detection and speed measurement system works effectively using two IR sensors and an Arduino Uno. The model accurately detects when a vehicle passes each sensor, measures the time difference, and calculates the speed with good reliability. The speed is displayed clearly on the LCD screen, and the buzzer provides an alert whenever the vehicle crosses the preset speed limit. This project demonstrates a low-cost and efficient method for monitoring vehicle speed, which can be useful at traffic signals, school zones, and road safety checkpoints. Overall, the system is simple to build, easy to operate, and successfully meets the objective of detecting vehicles and measuring their speed in real time.

1. Traffic Management Systems: The system can be deployed on highways and urban roads to monitor vehicle density, traffic congestion, and flow patterns, enabling intelligent traffic signal control and optimized road usage.
2. Law Enforcement and Speed Violation Detection: By estimating the speed of moving vehicles, the system can automatically detect over-speeding vehicles and record their number plates, assisting authorities in automated challan generation and enforcement of traffic regulations.
3. Accident Detection and Prevention: Integration with sensors and alert systems allows early detection of accidents or sudden stops, enabling emergency response systems to act promptly and reduce fatalities.
4. Smart City and Intelligent Transportation Systems (ITS): This technology can be an integral component of smart city infrastructures, contributing to real-time data analytics, predictive traffic management, and sustainable urban planning.

5. Toll Booth Automation: The system can assist in automatic vehicle identification and speed-based toll calculation, reducing manual intervention and improving toll booth efficiency.
6. Parking and Access Control Systems: Used in intelligent parking systems, it can help in monitoring vehicle entry and exit, improving security and automation in commercial and residential complexes.
7. Transportation Research and Analytics: Traffic and vehicle data collected through this system can be used for transport policy planning, infrastructure improvement, and research in vehicular behavior analysis.
8. Surveillance and Security Applications: In sensitive zones like border areas, airports, and defense installations, the system can be utilized for monitoring unauthorized vehicle movements and ensuring enhanced security.

6. REFERENCES

[1]. Alessandro D Alconzo, Idilio Drago, Andrea Morichetta, Marco Mellia, Pedro Casas, A survey on big data for network traffic monitoring and analysis, IEEE Trans. Netw. Serv. Manag. 16 (3) (2019) 800–813.
[2]. Chakchai So-In, A survey of network traffic monitoring and analysis tools, in: Cse 576m computer system analysis project, Washington University in St. Louis, 2009.
[3]. Alisha Cecil, A summary of network traffic monitoring and analysis techniques, Comput. Syst. Anal. (2006) 4–7.
[4]. Amin Shahraki, Hamed Taherzadeh, Øystein Haugen, Last significant trend change detection method for offline poisson distribution datasets, in: 2017 In ternational Symposium on Networks, Computers and Communications (ISNCC), IEEE, 2017, pp. 1–7.
[5]. Amin Shahraki, Amir Taherkordi, Øystein Haugen, Frank Eliassen, Clustering objectives in wireless sensor networks: A survey and research direction analysis, Comput. Netw. 180 (2020) 107376.
[6]. Jun Liu, Feng Liu, Nirwan Ansari, Monitoring and analyzing big traffic data of a large-scale cellular network with hadoop, IEEE Netw. 28 (4) (2014) 32–39.
[7]. Se-Hee Han, Myung-Sup Kim, Hong-Taek Ju, James Won-Ki Hong, The archi tecture of NG-MON: A passive network monitoring system for high-speed IP networks, in: International Workshop on Distributed Systems: Operations and Management, Springer, 2002, pp. 16–27.
[8]. Marco Ehrlich, Alexander Biendarra, Henning Trsek, Emanuel Wojtkowiak, Jürgen Jasperneite, Passive flow monitoring of hybrid network connections regarding quality of service parameters for the industrial automation, in: 8. Jahreskolloquium ‘‘Kommunikation in der Automation–KommA, 2017.
[9]. Raouf Boutaba, Mohammad A Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada-Solano, Oscar M Caicedo, A comprehensive survey on machine learning for networking: evolution, applications and research opportunities, J. Internet Serv. Appl. 9 (1) (2018) 16.
[10]. Alexandros Labrinidis, Hosagrahar V. Jagadish, Challenges and opportunities with big data, Proc. VLDB Endow. 5 (12) (2012) 2032–2033.
[11]. Uthayasankar Sivarajah, Muhammad Mustafa Kamal, Zahir Irani, Vishanth Weerakkody, Critical analysis of big data challenges and analytical methods, J. Bus. Res. 70 (2017) 263–286.
[12]. Amin Shahraki, Marius Geitle, Øystein Haugen, A comparative node evaluation model for highly heterogeneous massive- scale internet of things-mist networks, Trans. Emerg. Telecommun. Technol. 31 (12) (2020) e3924.
[13]. Amin Shahraki, Øystein Haugen, An outlier detection method to improve gathered datasets for network behavior analysis in IoT, Academy Publisher, 2019.
[14]. Amin Shahraki, Amir Taherkordi, Øystein Haugen, Frank Eliassen, A survey and future directions on clustering: From WSNs to IoT and modern networking paradigms, IEEE Trans. Netw. Serv. Manag. (2020).
[15]. Enrico Masala, Antonio Servetti, Simone Basso, Juan Carlos De Martin, Chal lenges and issues on collecting and analyzing large volumes of network data measurements, in: New Trends in Databases and Information Systems, Springer, 2014, pp. 203–212. [16]. Simone Basso, Antonio Servetti, Juan Carlos De Martin, Rationale, design, and implementation of the network neutrality bot, in: Congresso AICA 2010 (L'Aquila), 2010.
[17]. Srikanth Sundaresan, Walter De Donato, Nick Feamster, Renata Teixeira, Sam Crawford, Antonio Pescapè, Broadband internet performance: a view from the gateway, ACM SIGCOMM Comput. Commun. Rev. 41 (4) (2011) 134–145.