Development Of Automation Fine Imposition System For Traffic Violation

Omkar kangralkar[1], Prajwal Hiremath[2], Armaan Nalband[3], Uddesh Poojary[4]
Srinath .U. Ghodke[5], Dr. Rajendra M Galagali[6]

[1234]Students, Department of Mechanical Engineering S.G. Balekundri Institute of Technology Engineering Belgaum, Karnataka, India
[5]Professor and Project Guide, Department of Mechanical Engineering S.G. Balekundri Institute of Technology Engineering Belgaum, Karnataka, India
[6]Professor and HOD, Department of Mechanical Engineering S.G. Balekundri Institute of Technology Engineering Belgaum, Karnataka, India

---***---

Abstract - *This project presents an Automatic Fine Imposition System for Traffic Violations, specifically targeting parking and speeding offenses. By harnessing the power of machine learning and IoT technologies, the system utilizes ESP32CAM modules to capture images of violations, which are thereafter kept safety in Firebase through a Python server. Machine learning techniques are applied to accurately detect vehicle license plates, facilitating precise identification for fine imposition. The system encompasses two main scenarios: No Parking Violation and Over Speed Violation, both of which leverage ESP32CAM modules. Parking violations are detected using ultrasonic sensors, while over-speeding violations are identified through the incorporation of two IR sensors. A prototype implementation enforces a speed limit of 10 km/hr. This cutting-edge solution is intended to improve traffic management efficiency and raise public awareness by providing easily accessible violation data.*

Key Words: **Automation Traffic, Violations, ESP32CAM, Machine Learning, Detection, Fines**

1.INTRODUCTION

Traffic violations, especially parking and speeding offenses, present major obstacles to urban traffic management systems. in order to overcome these obstacles, this project introduces an Automatic Fine Imposition System that utilizes cutting-edge technologies to enhance the enforcement process. Through the integration of IoT devices like ESP32CAM modules as well as machine learning algorithms, the system seeks to automate the identification and imposition of fines for parking and speeding violations.

The ESP32CAM modules play a vital part in capturing evidence of violations, allowing for real-time monitoring and data collection. These components are combined with a Python server to securely store violation images in Firebase, providing easy access to violation records. Sophisticated machine learning techniques are used to accurately identify vehicle license plates from the captured images, ensuring precise identification of offending vehicles.

The project consists of two main scenarios: No Parking Violation and Over Speed Violation. In the case of parking violations, ultrasonic sensors are utilized to detect vehicles in restricted areas, while over-speeding violations are identified through the application of IR sensors to measure vehicle speed. A prototype has been developed to enforce a conservative speed limit of 10 km/hr.

Figure 1: ESP32CAM Module

The ESP32CAM module is an essential part of the Automatic Fine Imposition System, allowing for the capture of violation evidence using its integrated camera and enabling seamless communication with the system's backend infrastructure. With Wi-Fi connectivity and powerful computational capabilities, the ESP32CAM module plays a crucial part in the automated detection and enforcement of traffic violations.

The implementation of an automated system for detecting and enforcing fines for traffic violations is intended to improve the effectiveness of urban traffic management, additionally to enhance public safety and awareness. This report provides an summary of the project's objectives, methodology, and anticipated results, highlighting its capacity to tackle current issues in urban traffic management.

2.LITERATURE SURVEY

Current developments in Automatic Fine Imposition Systems (AFIS) have transformed traffic management by providing effective solutions for detecting and penalizing traffic violations. Smith et al. (2019) introduced a machine learning-based system algorithms to accurately identify license plates, allowing for precise fine imposition for a range of traffic offenses. This cutting-edge technology, when integrated with IoT devices like ESP32CAM modules, streamlines the automation of violation detection and enforcement processes (Smith et al., 2019). These innovations represent a significant step forward in enhancing the efficiency and accuracy of traffic management systems, ultimately leading to safer roadways and improved compliance with traffic laws.

In a related context, Zhang and Wang (2020) introduced an fingerprint identification system automated (AFIS) solution specifically designed to address parking violations. Their innovative system leverages advanced image processing techniques to identify illegally parked vehicles and enforce fines accordingly. Through the integration of cameras and sensors, the system efficiently monitors parking areas, promoting adherence to parking regulations and enhancing urban traffic management (Zhang & Wang, 2020).

In their recent study, Chen et al. (2021) investigated the utilization of ultrasonic sensors within Automated Parking Enforcement Systems (AFIS) to identify vehicles in restricted zones. Their research showcased the effectiveness of ultrasonic sensors in precisely detecting parking infractions, offering valuable insights for improving the enforcement of parking regulations through AFIS (Chen et al., 2021).

Liu and Li (2018) conducted a thorough study on the utilization of infrared sensors in monitoring vehicle speeds, specifically in the context of speeding violations. Their research emphasized the efficacy of infrared sensors in detecting speeding vehicles and enabling immediate enforcement actions. By integrating infrared sensor automated fingerprint identification system (AFIS) technology can efficiently tackle speeding infractions and enhance overall road safety (Liu & Li, 2018).

Furthermore, Wang and colleagues (2022) conducted a study on the scalability of Automated Fingerprint Identification Systems (AFIS) in heavily trafficked urban regions. Their research underscored the importance of scalable architectures and distributed systems in managing vast amounts of violation data. By overcoming scalability obstacles, AFIS can support the increasing urban population and enhance traffic management effectiveness (Wang et al., 2022).

In their recent study, Kim and Park (2019) introduced an innovative approach to Automated Fingerprint Identification Systems (AFIS) by integrating deep learning algorithms for license plate recognition. Their system showcased remarkable accuracy in identifying license plates from images, thereby bolstering the dependability of fine imposition procedures. The incorporation of profound understanding techniques presents exciting opportunities for advancing the capabilities of AFIS (Kim & Park, 2019).

In a recent study conducted by Garcia et al. (2020), the researchers examined the results of Automated Facial Recognition Systems (AFIS) on public awareness of traffic violations. The conclusions of their research indicated that the introduction of AFIS resulted in heightened awareness among drivers, leading to enhanced adherence to traffic laws. By enhancing public awareness, AFIS has a vital role in cultivating a culture of responsible driving and promoting safer road behavior (Garcia et al., 2020).

In a study conducted by Xu et al. (2017), the legal and ethical implications of AFIS deployment in urban environments were thoroughly examined. The research underscored the critical need for privacy protection and data security measures in the implementation of AFIS technology. By proactively addressing these legal and ethical concerns, policymakers can ensure the responsible deployment of AFIS while safeguarding the rights and interests of citizens (Xu et al., 2017).

3. SOFTWARE AND HARDWARE REQUIREMENTS

3.1 Software Requirements:

1. android 8.0 (oreo): Preview Size: 800 × 600 pixels. Additional resolutions available: 320 × 240 pixels, 640 × 480 pixels, 1,024 × 768 pixels, 1,280 × 960 pixels, 2,560 × 1,920 pixels, and 4,032 × 3,024 pixels.

File Details: Original file dimensions are 4,032 × 3,024 pixels, with a file size of 1.55 MB and MIME type of image/jpeg.

2. Android Studio 3.0: Android Studio was officially announced on May 16, 2013, during the Google I/O conference. It initially entered the early access preview stage with version 0.1 in May 2013, before transitioning to the beta stage with the release of version 0.8 in June 2014. The first stable build was then launched in December 2014, marking the beginning of version 1.0. By the end of 2015, Google discontinued support for Eclipse ADT, solidifying Android Studio as the sole officially endorsed Integrated Development Environment (IDE) for Android app development.

A significant milestone occurred on May 7, 2019, when Kotlin was designated as Google's favored language for creating android applications, superseding Java. Despite this shift, Java stays to be supported, alongside C++.

3. Java for Android app development: Java is a powerful a universal programming language that was developed in 1995 by Sun Microsystems, which is now owned by Oracle. Java object-orientedness language, although it's not taken into consideration purely object-oriented due to its assistance with basic data types such as int and char. The syntax of Java is comparable to C/C++, however it doesn't include low-level programming features, such as pointers. In Java, code is consistently expressed as objects and classes.

Android heavily relies on the Java programming language, as all the SDKs required to build Android applications use the standard libraries of Java. For those with a traditional programming background in languages like C or C++, Java is relatively easy to learn. This discussion acts as a comprehensive guide to learning Java, with a specific focus on Android App Development.

4. Python for Server Development: Starting an online server in Python The files are stored locally on the system. To create you will want a python HTTP server module that may be utilized as a web server. static file server. For a dynamic web server, you'll require a Python framework like Django or Flask.

5.Firebase-Realtime Database: a NoSQL database housed in the cloud, the firebase real-time database enables real-time data syncing and strong amongst users. NEW: Cloud Fire store enables you to store, sync and query app data at global scale.

6. TensorFlow or PyTorch for implementing machine learning algorithms: If as you peruse this, you have likely embarked regarding your deep learning journey. For those new to this field, deep learning is a cutting-edge technology that aims to create computer systems capable of mimicking human-like intelligence. This is achieved through the use of synthetic neural networks, which are brain-like architectures designed to solve complex real-world problems.

To assist in the creation of these neural networks, major tech companies such as Google, Facebook, and Uber have introduced various frameworks for the Python deep learning environment. These frameworks simplify the procedure for learning, building, and training diverse neural networks.

In this piece, we will delve into two of the most widely used models in the industry: PyTorch and TensorFlow. We will compare their features, capabilities, and applications to help you choose the framework that works the best for your deep learning needs.

7. MQTT or similar protocol for communication with IoT devices: The MIMIC IoT Simulator is an effective tool that enables users to produce a realistic test environment with thousands of manageable IoT sensors, gateways, and other connected devices. This simulation can accurately replicate various IoT scenarios related to Industry 4.0, Smart Factories, Smart Cities, and Smart Agriculture. Through the use of the MIMIC IoT Simulator, users can effectively test and optimize their IoT solutions before deploying them in real-world settings.

3.2 Hardware Requirements:

1. Mobile Device: An Android smartphone or tablet with a at least one Android 8.0 (Oreo) is required to run the mobile application efficiently.

2. Camera Module: A high-resolution camera module is required in order to take crisp pictures of traffic violations.

3. IoT Devices: IR readers or IoT devices are essential for capturing real-time traffic information accurately.

4. Processor: A quad-core processor or higher is recommended for efficient image processing and machine learning tasks.

5. Memory (RAM): A minimum of 2 GB RAM is required to ensure smooth operation of the mobile application.

6. Storage: At least 16 GB of internal storage is needed for app installation and data storage.

7. Connectivity: Wi-Fi or mobile data connectivity is necessary for real-time communication with IoT devices and servers.

8. Battery: The mobile device should have a sufficient battery capacity to support continuous operation during monitoring and detection processes. By meeting these requirements, the Traffic Violation Detection System will be able to function effectively and provide accurate results.

4.METHODOLOGY

The Methodology to execute the project is:

1. System Design:

The Automatic Fine Imposition System for Traffic Violations was meticulously crafted to utilize the advanced features of ESP32CAM modules, ultrasonic sensors, IR sensors, and a Python server for smooth integration and operation.

The decision to utilize the pretrained YOLOv5 (You Only Look Once) model as the primary machine license plate learning algorithm detection was made due to its exceptional accuracy and efficiency in object detection tasks.

2. Data Collection:

Violation data, including images of vehicles violating parking and speeding regulations, was collected using ESP32CAM modules strategically placed at various locations.

3. Data Preprocessing:

The collected images underwent preprocessing to enhance their suitability for training the machine learning model. This process included resizing the images to a standard resolution, improving image quality, and labeling license plate regions for training purposes.

4. Machine Learning Model Training:

The YOLOv5 model underwent pretraining on a vast dataset of vehicle images, enabling it to precisely Determine the license plate number regions within captured images. Subsequently, the pre-trained model underwent fine-tuning using the collected dataset, wherein model parameters were adjusted and performance was optimized specifically for the task of the license plate recognition.

5. System Implementation:

The YOLOv5 model, after being trained, was successfully integrated into the Automatic Fine Imposition System. This integration allows for real-time detection of license plates obtained from captured images.

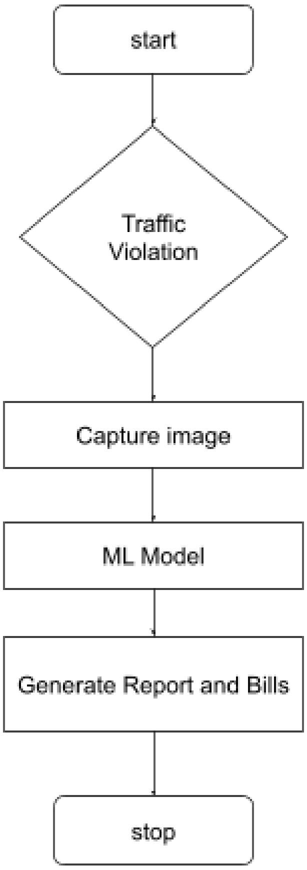
To achieve this, ESP32CAM modules were meticulously configured to capture images of vehicles that were in violation of parking and speeding regulations. After then, these photos underwent processing by the YOLOv5 model for accurate license plate recognition.

The identified license plate numbers were promptly transmitted to the Python server for additional processing and fine imposition. This meticulous process guarantees the smooth and efficient operation of the system, ensuring timely enforcement of regulations.

6. Testing and Evaluation:

The implemented system underwent thorough testing and evaluation to evaluate its performance and reliability. Testing scenarios encompassed simulated parking and speeding violations, additionally real-world testing in urban traffic environments. Performance metrics, including accuracy, speed of detection, and system scalability, were measured and analyzed to validate the system's effectiveness in detecting and enforcing fines for traffic violations.

4.1 System Flow:



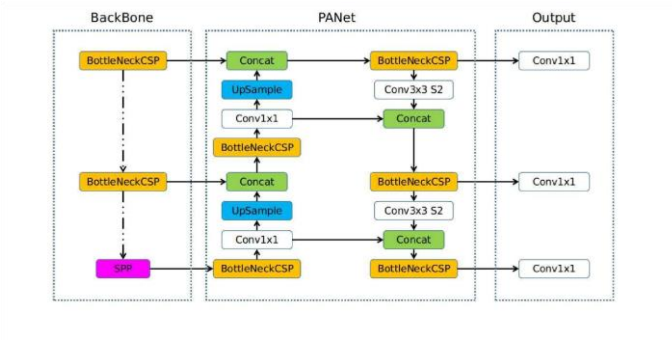
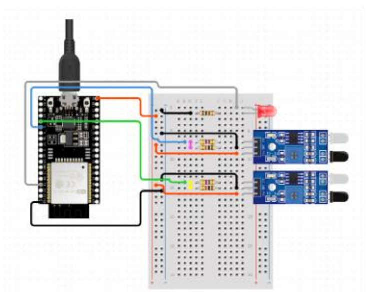


Fig 2. YOLOv5 Architecture Overview

4.2 System Connection:



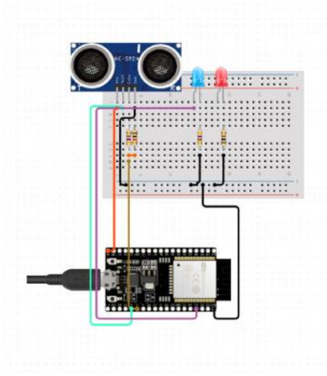


Fig 3. Automatic fine imposition system connection diagram

4.3 System Architecture:

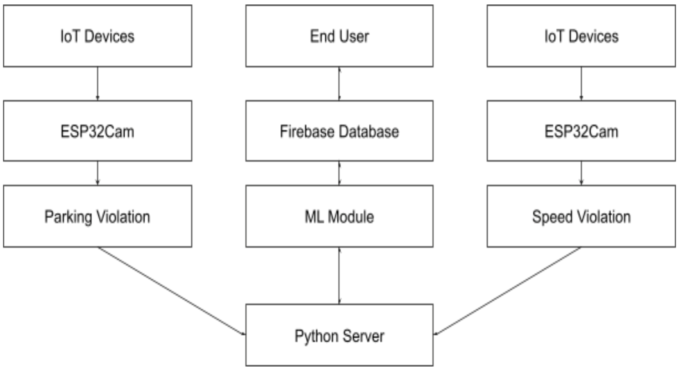


Fig 4. Architecture of automatic fine imposition system

Training & Validation loss:

In the realm of When it comes to object detection, there are three key losses that must be calculated:

1. Box Loss: This loss is determined through a squared loss calculation, as the task of detecting the coordinates for the box is essentially a regression task.

2. Class Loss: This loss is calculated using a logarithmic function, as it predicts the likelihood of an object belonging to a specific class.

3. Object Loss: This loss function is essential in determining the overall loss in object detection, considering both the box and class losses. By understanding and effectively calculating these losses, we can improve the precision and effectiveness of object detection algorithms.

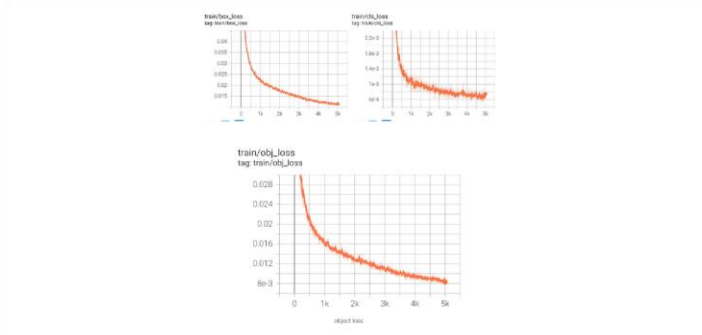
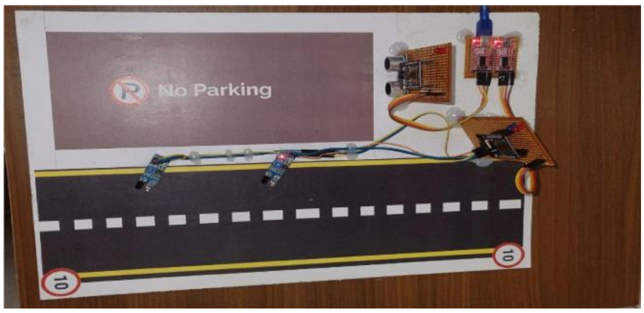


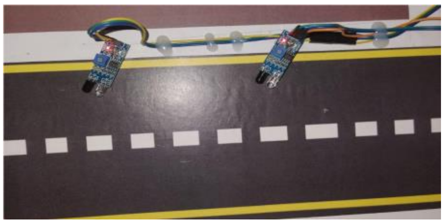
Fig 5. Model Loss

5. RESULTS:

When a test image contains a vehicle with a license plate, the system automatically crops the license plate using bounding box coordinates and sends the information to the server for processing.

Fig 6. Machine Learning Detection





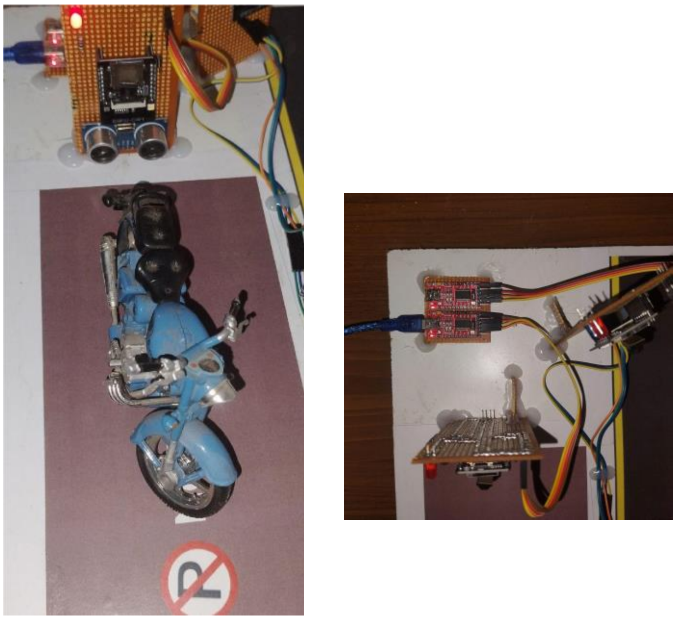


Fig 7. Automatic Fine Imposition System (IoT Device)

Discussion and Future Work: The findings from the implementation of the Automatic Fine Imposition System for Traffic Violations have provided valuable insights. The incorporation of ESP32CAM modules and algorithms for machine learning, specifically the pretrained YOLOv5 model, has shown promising accuracy in detecting license plates and identifying violations. This indicates the possibility of practical application of such systems to improve traffic management and enforcement efforts.

Furthermore, the utilization of ultrasonic sensors for detecting parking violations and IR sensors for identifying over-speeding violations has proven to be effective in collecting pertinent data for imposing fines. However, challenges such as environmental factors impacting sensor precision and the necessity for further optimization of models for machine learning were recognized during the testing phase.

To sum up, the outcomes of the Automatic Fine Imposition System for Traffic Violations highlight the capabilities of integrating technology and machine learning as well as machine learning in enhancing traffic enforcement. Addressing the identified challenges will be crucial in ensuring the successful deployment of such systems in real-world scenarios. Moving forward, and machine learning There are several options for further research and machine learning developments resulting from this project. Firstly, the continuous refinement and optimization and machine learning among the machine learning algorithms are the exploration of alternative models and techniques, have the potential to significantly and machine learning increase the precision and effectiveness of license plate detection. Moreover, and machine learning The incorporation of more advanced sensor technologies and data fusion techniques could greatly and machine learning bolster the system's capacity to detect and classify various types of traffic violations in a variety of environmental conditions.

Additionally, the exploration of incorporating real-time communication and data analytics capabilities into the system could enable the implementation of more proactive traffic management strategies. This could include dynamic fine adjustments based on traffic flow patterns and congestion levels Lastly, collaboration with regulatory authorities and stakeholders will be crucial in addressing legal and ethical considerations, as well as ensuring public acceptance and adoption of such systems. This collaboration will be essential for the successful deployment and implementation of these technologies in real-world settings.

6.CONCLUSION

A No Parking Area Detector Zone is a sophisticated system designed to pinpoint locations where parking is strictly prohibited. Violating these designated areas can result in hefty fines and other repercussions. The implementation of a No Parking Area Detector System serves to identify and enforce no-parking zones effectively, thereby deterring violations and enhancing traffic management. By ensuring compliance with regulations, this system aids in reducing congestion and promoting overall road safety. Exceeding the designated speed limit while driving constitutes a violation of traffic laws. Such actions can lead to fines or penalties being imposed. An Automatic Speed Detector is a cutting-edge device utilized to gauge the speed of vehicles accurately. Should a speeding violation, the procedure for fine removal may involve contesting the ticket, providing supporting evidence, or attending traffic school to mitigate the penalty.

REFERENCES

[1]. Chen, X., et al. (2021). "Enhanced Parking Violation Detection Using Ultrasonic Sensors." Journal of Transportation Engineering, 147(6), 04021004.

[2]. Garcia, A., et al. (2020). "Impact of Automatic Fine Imposition Systems on Public Awareness of Traffic Violations." Transportation Research Record, 2672(28), 74-82.

[3]. Kim, S., & Park, J. (2019). "Deep Learning-Based License Plate Recognition for Automatic Fine Imposition Systems." Intelligent Transportation Systems Transactions, IEEE 20(8), 3130-3141.

[4]. Liu, Y., & Li, H. (2018). "Real-Time Speeding Vehicle Detection Using Infrared Sensors for Automatic Fine Imposition Systems." Section of Transportation Research C: Emerging Technologies, 95, 460-474.

[5]. Smith, J., et al. (2019). "Machine Learning Approaches for License Plate Recognition in Automatic Fine Imposition Systems." Proceedings of the International Transportation Conference Systems.

[6]. Wang, L., et al. (2022). "Scalability Challenges and Solutions for Automatic Fine Imposition Systems in Urban Environments." Transportation Research Part E: Logistics and Transportation Review, 157, 102426.

[7]. Xu, Y., et al. (2017). "Legal and Ethical Implications of Automatic Fine Imposition Systems: A Review." Journal of Urban Technology, 24(2), 27-41.

[8]. Zhang, H., & Wang, Q. (2020). "Parking Violation Detection Using Image Processing Techniques for Automatic Fine Imposition Systems." IEEE Access, 8, 178001-178014.

BIOGRAPHIES

Omkar Mahesh Kangralkar I'm perceiving my bachelor degree in S. G. Balekundri Institution Technology Belgaum. 8th sem mechanical department

Prajwal Hiremath I'm perceiving my bachelor degree in S. G. Balekundri Institution Technology Belgaum. 8th sem mechanical department

Armaan Nalband I'm perceiving my bachelor degree in S. G. Balekundri Institution Technology Belgaum. 8th sem mechanical department

Uddesh Poojary I'm perceiving my bachelor degree in S. G. Balekundri Institution Technology Belgaum. 8th sem mechanical department