IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# AI-BASED TRAFFIC MANAGEMENT SYSTEM

[1]Lonka Kushal Reddy, [2]Dr. K Shanthi Sree

[1]Student, [2]Professor

[1,2]Department of Information Technology,

[1,2]JNTUH University College of Engineering Science & Technology, Hyderabad, Telangana, India

***Abstract:*** The "AI-Based Traffic Management System" project introduces an innovative approach to optimize traffic flow and enhance intersection safety using computer vision and artificial intelligence (AI). Traditional traffic signal systems often struggle to efficiently handle varying traffic conditions, leading to congestion and delays. This project addresses these challenges by leveraging advanced technologies. The system utilizes the YOLOv3 deep learning model for real-time vehicle detection and classification from live video streams captured by traffic cameras. Vehicles, including bicycles, cars, buses, and trucks, are identified and counted as they approach a simulated intersection. This enables the system to dynamically assess traffic density from different directions (north, south, east, west) and make informed decisions to optimize traffic light control, thereby reducing congestion and improving overall traffic flow.

**Keywords:** AI-Based Traffic Management, Computer Vision, Artificial Intelligence (AI), YOLOv3.

## I. INTRODUCTION

One of the numerous issues we deal with in modern life is the increasing severity of traffic congestion every day. Determining the traffic flow can be crucial to learning more about them. Accurate records of traffic volume patterns and censorious flow time periods, such as the impact of big cars and specific parts on vehicular traffic flow, are produced using this data. When it comes to processing better traffic in terms of traffic signal timing, this recorded information is also helpful. There are numerous ways to gauge traffic flow by counting the number of cars that pass by in a given amount of time. Camera-based technologies are now a superior option for tracking down a vehicle's data. The unique firmware-based method for vehicle detection is the main emphasis of this study. This method finds the cars in the original picture and uses the pre-existing identifier for every car. Subsequently, it counts each vehicle separately after classifying each one according to its vehicle type group. The created method was applied to a firmware platform, leading to increased precision, increased dependability, and a decrease in errors. Traffic lights are an essential component of traffic regulation and control on a daily basis. After image processing, Python is employed to calculate road density. The microcontroller then uses this output to adjust the duration of the green light for each road.

## II. PROBLEM DEFINITION

Current traffic management systems are predominantly static, lacking the capability to adjust to real-time traffic conditions. Fixed-timing traffic signals do not account for variations in traffic volume, leading to inefficient use of road infrastructure. During peak hours, this can result in significant traffic jams, while off-peak hours might see unnecessary waiting times at red lights. The problem is further compounded by the diverse mix of vehicles, including bicycles, cars, buses, and trucks, each contributing differently to the traffic dynamics. The inability to differentiate and appropriately manage these various vehicle types exacerbates congestion and delays. The core problem this project addresses is the need for a dynamic and intelligent traffic control system that can monitor real-time traffic conditions, identify different vehicle types, and adjust traffic signals accordingly to optimize flow and reduce congestion.

## III. EXISTING SYSTEM

In India, traffic signals are essentially pre-programmed, meaning that the moment each lane receives a green light is set in advance. One lane at a time is given a green signal in a four-lane traffic signal. As a result, the traffic light permits sequential passing for all lanes of traffic. Therefore, even if a certain lane has the least amount of traffic, it still needs to wait a long time and unnecessarily causes other lanes to wait even longer until it gets the green signal.

The following drawbacks exist:
● Time-consuming
● Delay
● Reduced precision

## IV. PROPOSED SYSTEM

The proposed system aims to overcome the limitations of existing traffic management systems by integrating an advanced object detection framework with a dynamic traffic control algorithm. Using the YOLO (You Only Look Once) object detection model, the system will accurately identify and count different types of vehicles in real-time from video feeds. YOLO is chosen for its balance between speed and accuracy, making it suitable for real-time applications. The system will process the video data to dynamically adjust traffic signal timings based on the current traffic conditions, optimizing traffic flow and reducing congestion.

### 4.1. Proposed Architecture:

This project uses the YOLO (You Only Look Once) object detection algorithm to detect vehicles in a video feed and control a traffic light system based on the vehicle count. The main components of the project include input handling, model loading, frame processing, object detection, post-processing, traffic light control, and output display.

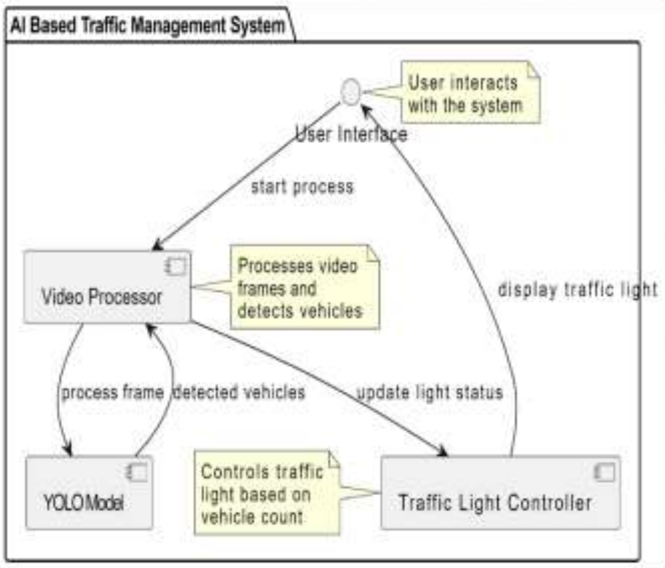


Fig 4.1: Proposed Architecture

### 4.2 Advantages of Proposed System:

- Real-Time Adaptation
- Improved Efficiency
- Enhanced Safety

### 4.3 YOLO Model

**How YOLO Works:**

**Input Processing:**

YOLO takes an input image and resizes it to a fixed size (e.g., 416x416 pixels).

**Feature Extraction:**

The input image is passed through a convolutional neural network (CNN) to extract features.

**Grid Cell Division:**

The input image is partitioned into a SxS grid by YOLO. The task of estimating bounding boxes and class probabilities falls to each grid cell.

**Prediction:**

For each grid cell, YOLO predicts bounding boxes and their associated class probabilities. Each bounding box is represented by a set of coordinates (x, y, width, height) and a confidence score indicating the likelihood that the box contains an object of any class.

**Non-Maximum Suppression (NMS):**

YOLO uses this to remove overlapping bounding boxes, maintaining only those that are the most confident ones.

**Output:**

YOLO produces a final set of bounding boxes along with their associated class labels and confidence scores.

**Applications:**

- **Autonomous Vehicles:** YOLO can be used for real-time detection of pedestrians, vehicles, and traffic signs.
- **Surveillance:** Efficiently monitor and detect objects in surveillance videos.
- **Medical Imaging:** Detect anomalies or specific objects in medical images.
- **General Object Detection:** Used in various applications requiring real-time object detection and localization.

YOLO models are typically trained on large datasets such as COCO (Common Objects in Context) to recognize a wide range of objects. Pre-trained YOLO models and their configurations (weights, configurations) are available, making it easier to implement and customize for specific applications.

## V. IMPLEMENTATION

### 1. Loading YOLO Model and Configurations

Start by loading the YOLO model and setup files. YOLO configures the network with pre-trained weights and a configuration file. Additionally, load the COCO class names to identify the various items observed by the model.

### 2. Loading Traffic Light Images and Video

Prepare the traffic light images representing green and red lights. These images will be displayed based on the vehicle count in the frame. Load the video file that contains the traffic feed.

### 3. Processing Video Frames

For each frame captured from the video:

- Resize the frame for consistent processing.
- Convert the frame to grayscale and apply binary thresholding for additional visualizations.

**4. Object Detection Using YOLO**

For each resized frame:

- Convert the frame to blob format, which is necessary for YOLO.
- Set the blob as the network's input and run a forward pass to obtain detections.
- Iteratively process the detections to obtain bounding boxes, class IDs, and confidence ratings.

**5. Filtering Detections**

Filter the detections to retain those with high confidence scores. Use Non-Maximum Suppression (NMS) to reduce overlapping bounding boxes and keep the best ones.

**6. Counting Vehicles**

Classify detected objects into predefined categories: bicycles, cars, buses, and trucks. Count the occurrences of each vehicle type to determine the total vehicle count in the frame.

**7. Traffic Light Control Logic**

Based on the total vehicle count:

- Display the red light image if the vehicle count is below a specified threshold.
- Display the green light image if the vehicle count exceeds the threshold.

**8. Displaying Results**

Display the processed frame, including bounding boxes and labels for discovered items. In addition, display the appropriate traffic signal graphic based on the vehicle count. Create a loop that processes each frame until the movie stops or a termination condition is satisfied.

**9. Releasing Resources**

Once the processing is complete, release the video capture object and close all OpenCV windows.

This approach leverages YOLO's real-time object detection capabilities to create an intelligent traffic light control system that dynamically adapts to traffic conditions, potentially improving traffic flow and reducing congestion.

## VI. RESULT ANALYSIS

**Input Video:**

The input video stream was processed to detect vehicles using the YOLO algorithm.

Figure 6.1: Input Video

**Vehicle Detection:**

- The YOLO algorithm successfully detected various types of vehicles, including bicycles, cars, buses, and trucks, with high accuracy.
- The detection results were transmitted to the output window for further processing.

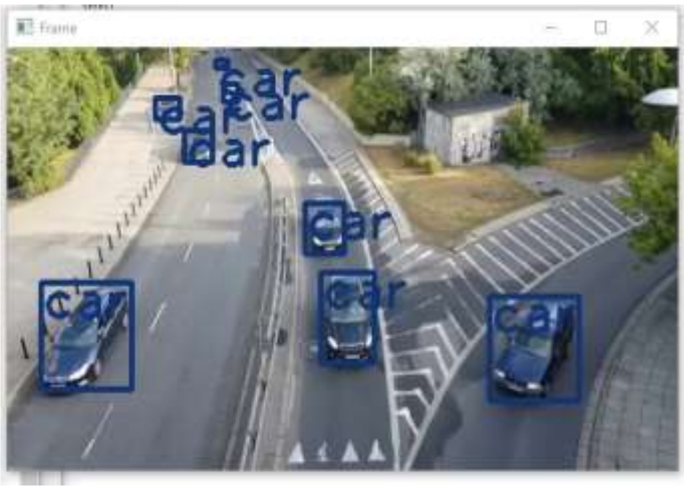


Figure 6.2: Vehicle Detection

**Traffic Signal Control:**

- The system counted the total number of detected vehicles and controlled the traffic signals accordingly.

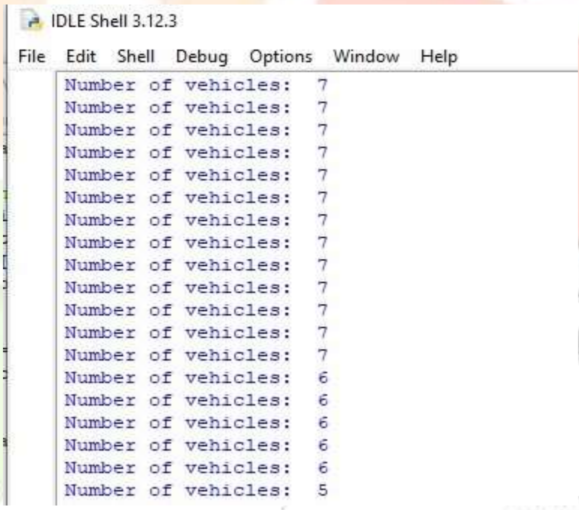


Figure 6.3: Output Screen

**Green Light:** When the number of detected vehicles was 7 or more.

Figure 6.4: Green Light

**Red Light:** When the number of detected vehicles was less than 7.

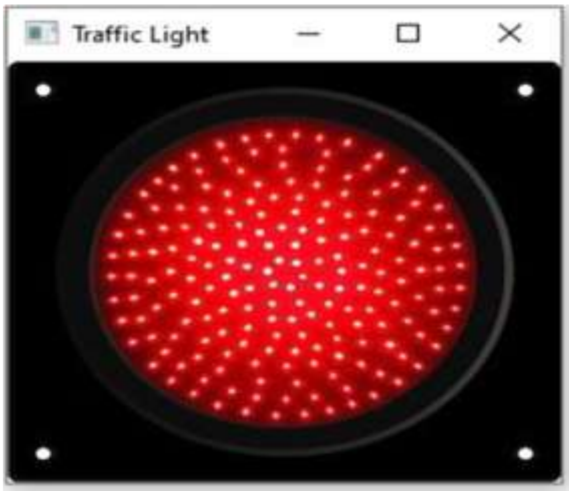


Figure 6.5: Red Light

Upon running the implemented code on various video datasets, the following observations were made:

- The vehicle detection algorithm successfully detected vehicles of different types (bicycle, car, bus, truck) with high accuracy.
- The traffic light control mechanism effectively switched between red and green lights based on the total count of vehicles detected.
- Performance metrics such as processing speed and resource utilization were within acceptable limits, allowing for real-time operation in most cases.
- Further optimization and fine-tuning may be required to handle challenging scenarios such as occlusions and varying lighting conditions.

Overall, the implemented system demonstrates promising results in vehicle detection and traffic light control, laying the foundation for future improvements and enhancements.

## VII. CONCLUSION

The "AI-based Traffic Light Management System" project leverages cutting-edge artificial intelligence and computer vision technologies to enhance traffic flow, safety, and efficiency at intersections. By dynamically adjusting traffic signals based on real-time data, the system optimizes traffic management, reduces congestion, and promises more sustainable and safer urban environments. This study demonstrates the feasibility and potential benefits of using AI for intelligent traffic management, paving the way for future urban traffic control systems advancements.

## VIII. FUTURE SCOPE

Future enhancements for the AI-based Traffic Light Management System include improving detection accuracy with advanced models like YOLOv4 and YOLOv5, expanding scalability for multi-camera feeds, and developing adaptive control algorithms for dynamic traffic prediction. Integrating the system with other smart city infrastructure, creating user interfaces for traffic managers, and ensuring robust, failsafe operations through extensive field testing are also key. These advancements will enhance system reliability, efficiency, and integration into broader urban traffic management strategies.

**REFERENCES:**


1) A. Ross and A. Jain, Information Fusion in Biometrics,in Pattern Recognition Letters, vol.24, pp.2115-2125,2003.
2) Ing. Ibrahim Nahhas, Ing. Filip Orsag, Ph.D “Real Time Human Detection And Tracking”, Bruno University of Technology.
3) Vikramaditya Dangi, Amol Parab, "Image Processing Based Intelligent Traffic Controller", Undergraduate Academic Research Journal (UARJ), ISSN : 2278 – 1129, Volume-1, Issue-1, 2012.
4) H. R. Babaei, O. Molalapata and A. A. Pandor, Face Recognition Application for Automatic Teller Machines (ATM), in ICIKM, 3rd ed. vol.45, pp.211-216, 2012.
5) Aru, O. Eze and I. Gozie, Facial Verification Technology for Use in ATM Transactions, in American Journal of Engineering Research (AJER), [Online] 2013,pp.188-193, Available: http://www.ajer.org/papers/v2(5)/Y02501880193.
6) Raoul de Charette and Fawzi Nashashibi, “Traffic light recognition using Image processing Compared to Learning Processes”.
7) Mriganka Panjwani, Nikhil Tyagi, Ms. D. Shalini, Prof. K Venkata Lakshmi Narayana, “Smart Traffic Control Using Image Processing”.
8) H. R. Babaei, O. Molalapata and A. A. Pandor, Face Recognition Application for Automatic Teller Machines (ATM), in ICIKM, 3rd ed. vol.45, pp.211-216, 2012.
9) Aru, O. Eze and I. Gozie, Facial Verification Technology for Use in ATM Transactions, in American Journal of Engineering Research (AJER), [Online] 2013,pp.188-193, Available: http://www.ajer.org/papers/v2(5)/Y02501880193.