; revised .

# Traffic Management System using DL

**S.Harsha Vardhan, B.Merwin Pranav,N.Prasanna Kumar, K.Supraja,Dr.Venkata Bikhshapathi Chenam.**

[1]Department of Computer science and Engineering, VIT - AP University, Amaravati, Andhra Pradesh, India.
[2]Vellore Institute of Technology,AndhraPradesh

**ABSTRACT** With the rapid urbanization and growth of smart cities, efficient traffic management has become essential to alleviate congestion, enhance road safety, and improve urban life quality. This paper presents a comprehensive traffic management system leveraging deep learning (DL) techniques, with a focus on the You Only Look Once (YOLO) object detection algorithm, combined with camera and sensor data. The proposed system aims to enable real-time monitoring, detection, and prediction of traffic patterns and anomalies across smart city roads and highways. By utilizing YOLO's object detection capabilities, the system can accurately detect and classify various traffic entities, such as vehicles, pedestrians, and traffic signals, even in complex and dynamic environments. Integrating data from high-resolution cameras and IoT sensors, along with vehicle-to-infrastructure (V2I) communication, enhances detection accuracy and provides a comprehensive view of traffic conditions. This integration also supports predictive analytics for identifying congestion points, potential accidents, and rule violations.

The system's architecture involves multiple modules: real-time data collection from roadside cameras and embedded sensors, data preprocessing, object detection, and a predictive model for traffic flow management. By leveraging YOLO's speed and accuracy, alongside deep neural networks trained on historical traffic data, the system can swiftly identify and respond to traffic irregularities and forecast potential issues before they escalate. Experimental results demonstrate that the model achieves high accuracy in dynamic environments, reducing latency and optimizing resource allocation. This proposed solution not only offers a scalable approach for addressing urban traffic challenges but also aligns with smart city initiatives, promoting safer and more efficient roadways across urban and highway infrastructures.

**INDEX TERMS** Traffic management, YOLO, deep learning, smart cities, object detection, real-time monitoring, IoT sensors, vehicle-to-infrastructure (V2I), predictive analytics, congestion control, road safety, urban transportation, highway systems, smart infrastructure.

## I. INTRODUCTION

THIS addresses the critical and ever-growing issue of traffic congestion, which significantly impacts urban environments globally. Traditional methods, such as infrastructure expansion, have proven insufficient in managing the complexities of modern traffic flows. As urban populations grow and vehicle numbers increase, an advanced, adaptive system becomes essential for efficient traffic management.

This paper proposes an innovative approach using deep learning and computer vision, specifically employing YOLO (You Only Look Once) models to address the real-time challenges of urban traffic. Unlike static systems, this intelligent control system leverages dynamic timer settings based on traffic density, categorized as low, medium, or high. By continuously processing video feeds from traffic cameras, the system detects and classifies vehicles, adjusting signal timers dynamically to optimize flow.

The paper's motivation is clear: improving urban mobility by reducing unnecessary wait times, minimizing fuel wastage, and enhancing commuter experience, particularly in highly populated cities. The proposed model leverages recent advances in deep learning, particularly YOLOv3 and YOLOv5, for efficient object detection. This approach not only enhances traffic flow but also enables resource-efficient urban transport management by replacing costly sensors with reliable video processing. Through adaptive traffic signals, the system provides a scalable and responsive solution to congestion, marking a significant step toward smarter, more sustainable cities.

Furthermore, by relying on video data, the system avoids the high installation and maintenance costs associated with physical sensors. This not only makes it more cost-effective but also ensures longevity and reduced operational interruptions. The potential impact of this system is substantial, offering improvements not only in terms of reduced congestion but also in environmental benefits through decreased fuel consumption and emissions. Thus, the paper's approach is a forward-thinking solution that aligns with the increasing push toward smart city initiatives, addressing both immediate urban mobility challenges and broader sustainability goals.

### A. LITERATURE SURVEY

The literature surrounding intelligent traffic control systems highlights a growing interest in integrating computer vision, machine learning, and sensor-based methods to address traffic congestion challenges. Traditional solutions, like physical infrastructure expansion, are increasingly viewed as insufficient, particularly in urban areas where space is limited and traffic patterns vary significantly. Previous studies have focused on lane detection and traffic monitoring, as seen in the work of Jose Melo et al., who developed segmentation and clustering algorithms to track lane occupancy and vehicle behavior. Their research emphasizes the importance of tracking individual lanes to monitor traffic flow, which is instrumental in estimating congestion levels and detecting incidents, thereby informing smarter control systems. This approach provides essential insights into lane-specific traffic conditions, forming a foundation for more complex deep learning applications.1

Incorporating video-based vehicle counting and traffic flow analysis has been a significant step forward, with works like those of Abdagic et al., who developed a cost-effective solution that relies on video processing over physical sensors. Their use of optical flow algorithms to count vehicles and detect motion represents an efficient approach, especially in terms of reducing installation and maintenance costs.[11]However, this system's dependence on frame intensity data limits its applicability in diverse traffic settings, such as those with complex intersections or varying lighting conditions. By contrast, the current study in question uses deep learning to overcome these limitations, leveraging the robustness of computer vision to handle complex, high-density traffic scenes.[12]

Image processing has long been central to traffic control, with early implementations like those by Prashant Jadhav et al. utilizing sequential image analysis in MATLAB to monitor vehicle density on roadways. Although effective in basic traffic counting and density assessment, such techniques are often computationally intensive and struggle to adapt to real-time requirements. This paved the way for deep learning advancements, which provide more scalable and accurate solutions for real-time traffic management.[3] Jingwei Cao and colleagues advanced this area by integrating deep learning models like YOLO and SSD (Single Shot Multibox Detector) for multi-object detection, significantly improving detection speed and accuracy. Their work demonstrates how YOLO-based object detection frameworks can be applied to traffic management, an approach the current research paper builds upon by utilizing YOLOv3 and YOLOv5 to detect vehicle types and adjust traffic signals dynamically.[4]

Sensor-based approaches, while effective, present challenges related to cost. Cheung et al. explored this issue by developing a wireless sensor network (WITS) prototype aimed at making Intelligent Transportation Systems (ITS) more affordable.[5] Although sensor networks provide real-time data collection for traffic analysis, they require substantial initial investment and ongoing maintenance, making them less feasible for high-density, resource-constrained urban areas. By shifting focus to video-based systems, as in the current paper, traffic management solutions can reduce costs while achieving similar or better accuracy in monitoring traffic density.[6]

The use of machine learning for predicting traffic flow has also shown promise, particularly in estimating congestion and adjusting traffic signals based on predicted patterns. Research by Surojit Dey et al. demonstrated that combining data mining with neural networks allows for more accurate density estimation and flow prediction, though these systems often lack autonomous control capabilities.[7] The current paper enhances this predictive framework by combining YOLO-based vehicle detection with dynamic signal timing adjustments, creating a system that not only predicts but autonomously manages traffic flow in response to real-time conditions.[8]

Overall, this paper synthesizes methodologies from lane detection, video-based vehicle counting, image processing, and machine learning to develop a comprehensive traffic control solution. By leveraging YOLOv3 and YOLOv5 models, it addresses the limitations of static traffic systems, dynamically responding to varying traffic densities and enabling a more efficient, real-time approach to urban traffic management.[9] This integration of deep learning and adaptive algorithms represents a significant advancement in intelligent transportation systems, offering a scalable, cost-effective solution to one of urban society's most pressing challenges.[10]

## II. DATASETS

The dataset analysis in the paper "Smart Traffic Control System Using Deep Learning" focuses on the Microsoft Common Objects in Context (MS-COCO) dataset, which plays a crucial role in training the model for accurate vehicle detection and classification. The MS-COCO dataset is widely used in object detection research due to its extensive variety and quantity of labeled images. It contains over 330,000 images, with more than 1.5 million object instances across 80 different classes, making it well-suited for training complex deep learning models. [1]

In this study, five primary vehicle classes from the MS-COCO dataset were selected: motorbike, bicycle, truck, car, and bus. This selection aligns with the typical traffic seen in urban environments, enabling the model to effectively cap-

FIGURE 1. **Dataset image 1.**


FIGURE 2. **Dataset image 2.**

ture and differentiate between various types of vehicles on the road. The MS-COCO dataset's diverse images—covering crowded traffic scenes, intersections, and varying viewpoints—help the model generalize well to different traffic conditions and vehicle configurations.


FIGURE 3. **Dataset image 3.**

The dataset also includes annotated bounding boxes for objects, which aids in training the YOLOv3 and YOLOv5 models used in this paper. This bounding box information is essential for accurate vehicle counting and classification, as it allows the model to draw clear distinctions between closely packed vehicles in real-world traffic scenarios. Overall, the MS-COCO dataset's breadth and level of detail make it an ideal choice for this study, as it enables robust training for a highly dynamic and adaptable traffic control system.

## III. PROPOSED ARCHITECTURE

The proposed architecture presents a comprehensive and innovative solution to the persistent issue of traffic congestion in urban environments. This architecture is meticulously structured into three interrelated phases: video input processing, object detection, and dynamic traffic signal timing adjustment. Each phase is designed to operate seamlessly, ensuring a robust and efficient traffic management system capable of responding to real-time conditions.

In the first phase, the system begins with the acquisition of live video feeds from strategically placed traffic cameras at intersections. These cameras serve as the primary data source, capturing continuous streams of traffic activity. The incoming video data undergoes a rigorous preprocessing step, where several key adjustments are made to standardize the input. Specifically, each video frame is resized to a resolution of 416x416 pixels, which is optimal for the subsequent deep learning model processing. This resizing ensures that all input data maintains a uniform size, allowing the model to function efficiently without the overhead of handling varying resolutions.

Moreover, the color format of the frames is standardized to the RGB schema, which is necessary for the object detection algorithms utilized in later phases. To minimize redundant processing and optimize computational resources, the system employs a frame extraction strategy, selecting only certain frames from the video stream at defined intervals. This method significantly reduces the number of frames processed while preserving critical traffic dynamics, as adjacent frames typically exhibit minimal changes in vehicle density.

The second phase is pivotal, as it involves applying sophisticated object detection algorithms to the processed video frames. The paper leverages two state-of-the-art deep learning models: YOLOv3 and YOLOv5. These models are particularly effective due to their capability to perform real-time object detection, identifying and classifying various vehicle types, including cars, buses, motorcycles, and trucks.

YOLOv5, the more advanced of the two, has shown remarkable improvements in accuracy and speed, largely due to its efficient architecture that employs the Cross Stage Partial Network (CSPNet). This enhances the feature extraction process, allowing the model to maintain high levels of accuracy while minimizing computational load. Each frame is analyzed to detect multiple vehicles simultaneously, using bounding boxes to delineate their positions and classifications within the image.

The output from this phase is crucial; it provides real-time data on the number of vehicles present in each lane and their respective categories. This vehicle count is essential for the next phase, as it forms the basis for determining traffic density across lanes, which is integral to the dynamic traffic signal management process.

The final phase of the architecture focuses on adjusting traffic signal timings based on the data gathered in the previous phases. The dynamic traffic signal timer algorithm is designed to optimize the duration of green lights for each lane based on real-time traffic density. This adaptive approach contrasts sharply with traditional traffic signal systems that operate on fixed timings, often leading to inefficient traffic flow.

In this algorithm, lanes are categorized into three density classes: low, medium, and high, determined by the vehicle counts obtained from the object detection phase. A critical feature of the algorithm is its ability to adjust signal timings dynamically. For instance, if one lane is classified as having a high density of vehicles, it will receive an extended green light duration, while lanes with lower vehicle counts will have their green light durations reduced accordingly. This prioritization helps in minimizing congestion and reducing wait times at intersections.

The algorithm also accounts for various traffic scenarios, adjusting timers based on the relative density of vehicles in all lanes. For example, if multiple lanes are classified as high density, the system may implement a round-robin approach to distribute green light time efficiently among them, ensuring that no lane is excessively delayed. This flexibility is key to enhancing traffic flow and improving overall efficiency at intersections.

The proposed architecture for the smart traffic control system is a holistic solution that effectively integrates advanced technologies to address the complex challenges of urban traffic management. By leveraging deep learning for object detection, coupled with a dynamic approach to traffic signal management, the system not only alleviates congestion but also contributes to more sustainable urban mobility solutions. The architecture's adaptability to real-time conditions represents a significant advancement over traditional traffic management systems, positioning it as a viable component of smart city infrastructure. The potential implications of this research extend beyond mere traffic optimization, promising enhancements in environmental sustainability and urban livability through more efficient transportation systems.

### *A. SOFTWARE MODULES*

The modular software solution aimed at alleviating urban traffic congestion through deep learning and dynamic traffic signal management. This system integrates several key modules—video processing, object detection, and adaptive signal timing—to analyze real-time traffic density and optimize flow at intersections. Video input from traffic cameras undergoes processing to ensure consistency, with frames resized and color-corrected for analysis. This preprocessed data is then fed into an object detection module powered by YOLO (You Only Look Once) models, specifically YOLOv3 and YOLOv5, which detect and classify vehicles into types such as cars, bikes, and trucks. These models, trained on the MS-COCO dataset, allow for high-accuracy vehicle recognition across various traffic conditions.

The object detection output, including vehicle counts and types, feeds into the dynamic signal timing algorithm, which classifies each lane based on traffic density levels—low, medium, or high—and adjusts green light durations accordingly. This real-time signal adjustment gives priority to high-density lanes, aiming to reduce wait times and improve intersection efficiency. Implemented in Python, the system uses TensorFlow and Keras for neural network operations, while OpenCV aids in video processing. The use of neural networks like YOLOv5, with enhancements from CSPNet (Cross Stage Partial Network), contributes to the model's performance by optimizing detection across different scales and object sizes.

The primary advantage of this system lies in its sensor-free, camera-based design, which reduces setup and maintenance costs while leveraging existing infrastructure. However, some limitations include the potential impact of varying video quality and environmental conditions on detection accuracy. Additionally, the system's performance might be constrained by computational demands during heavy traffic or when operating on less powerful hardware. Overall, the proposed system offers an efficient and cost-effective approach to intelligent traffic management, with each module contributing to a cohesive solution aimed at optimizing urban traffic flow.

## IV. METHODOLOGY

The methodology of the proposed Smart Traffic Control System in this paper leverages a multi-phase approach combining video processing, deep learning-based object detection, and dynamic signal timing. This structured methodology aims to address urban traffic congestion by providing a real-time, adaptable solution that adjusts based on actual traffic conditions.

In the initial phase, video processing prepares the raw data captured from cameras positioned at traffic intersections. The system extracts frames at specific intervals to reduce computational load, a step critical for real-time performance. These frames are then standardized by resizing to a resolution of 416x416 pixels and converted to a uniform RGB color format. This preprocessing ensures consistency across inputs, a crucial factor given the diverse visual conditions of an urban environment. By using a multi-threaded approach for frame extraction, the system enhances efficiency, preparing data rapidly for the object detection phase.

The next phase implements object detection using YOLO (You Only Look Once) models, with both YOLOv3 and YOLOv5 architectures tested. The choice of YOLO models reflects their proven effectiveness in real-time object detection, as they use a grid-based approach to identify and localize vehicles within each frame. YOLOv3, for example, segments images into grids, detecting vehicles by predicting bounding boxes and classifying objects across three scales. YOLOv5, the latest in the YOLO family, incorporates CSP-Net (Cross Stage Partial Network) and GIoU-Loss (Generalized Intersection over Union Loss) for improved detection accuracy and efficiency. This phase's objective is to classify and count vehicles in each lane, identifying types such as cars, motorbikes, buses, and trucks. This classification directly informs traffic density estimation, a key input for the subsequent signal timing adjustments.

In the final phase, the system's dynamic signal timing algorithm uses the detected vehicle counts and types to calculate lane-specific traffic densities. By classifying traffic into density categories (low, medium, high), the algorithm adjusts green light durations dynamically, prioritizing lanes with higher densities. This approach marks a significant improvement over static signal timers, as it allocates time based on real-time traffic, aiming to minimize delays and balance flow across all lanes. The algorithm considers several factors, including vehicle distribution across lanes, threshold-based density classification, and predefined maximum and minimum signal timings. For example, when one lane has significantly higher traffic than others, the system reallocates green light time to prioritize that lane, enhancing throughput.

The methodological framework in this study is notable for its comprehensive integration of video processing and deep learning within an adaptive traffic management algorithm. By employing camera feeds instead of sensor-based systems, the solution significantly reduces installation and maintenance costs. However, certain challenges persist, such as potential variations in detection accuracy due to environmental factors like lighting or weather, as well as the computational demands of processing video data continuously. Nonetheless, the methodology presents a scalable, efficient approach to traffic control, aligning well with the needs of growing urban centers.

### A. Video Processing

The video processing step in the Smart Traffic Control System plays a foundational role, setting the stage for accurate and efficient vehicle detection in subsequent phases. In this step, the system processes raw video feeds from cameras placed at intersections, a choice that leverages existing infrastructure while avoiding the costs associated with installing and maintaining sensor-based systems. The approach involves extracting frames at specific intervals rather than analyzing the entire video stream, a decision that strategically reduces computational load and allows for real-time processing. This frame selection is critical, as it enables the system to capture relevant traffic data without overwhelming computational resources, making it both time-efficient and suitable for continuous monitoring.

To ensure consistent and high-quality input for object detection, each frame is standardized to a resolution of 416x416 pixels and converted into the RGB color format, regardless of the original video format. This standardization minimizes variability across frames, ensuring that the object detection model can operate on uniformly prepared images. Additionally, the conversion from other color spaces, like CMYK or HSV, to RGB maintains a consistent color schema, which is particularly valuable in traffic environments where lighting and weather conditions can vary. By maintaining this consistency, the system enhances the robustness of subsequent object detection and classification tasks.

The use of a multi-threaded approach for frame extraction further improves efficiency, allowing the system to manage multiple video feeds and quickly prepare frames for the object detection phase. This design choice reflects a balance between computational efficiency and real-time processing needs, as the system must adapt to constantly changing traffic conditions. Overall, the video processing step is methodically designed to deliver clean, standardized data, crucial for the performance of the deep learning model. This phase ultimately underscores the system's real-time capabilities, supporting accurate detection and classification and enabling dynamic, data-driven adjustments in signal timing.

### B. OBJECT DETECTION

The object detection phase in the Smart Traffic Control System is central to accurately identifying and classifying vehicles, directly influencing the system's ability to adapt traffic signals based on real-time conditions. In this phase, the system employs the YOLO (You Only Look Once) object detection models, specifically YOLOv3 and YOLOv5, which are both known for their speed and accuracy in real-time applications. The choice of YOLO is strategic, as it allows the system to balance high detection accuracy with rapid processing—a crucial requirement in traffic environments where vehicle flow is constantly changing.

YOLOv3 begins the detection process by dividing each frame into a grid, enabling it to detect multiple vehicles and classify them by type, such as cars, motorbikes, trucks, and buses. This version identifies objects across three scales, optimizing detection for objects of various sizes within the frame. By using a 1x1 detection kernel across feature maps, YOLOv3 can localize and classify vehicles efficiently, which is essential for identifying traffic density in each lane.

YOLOv5, the more recent model, further enhances detection capabilities with architectural improvements, including the incorporation of CSPNet (Cross Stage Partial Network) and the Generalized Intersection over Union (GIoU) Loss. These innovations reduce detection error and improve the model's precision, especially when multiple vehicles are closely packed—a common scenario in heavy traffic.

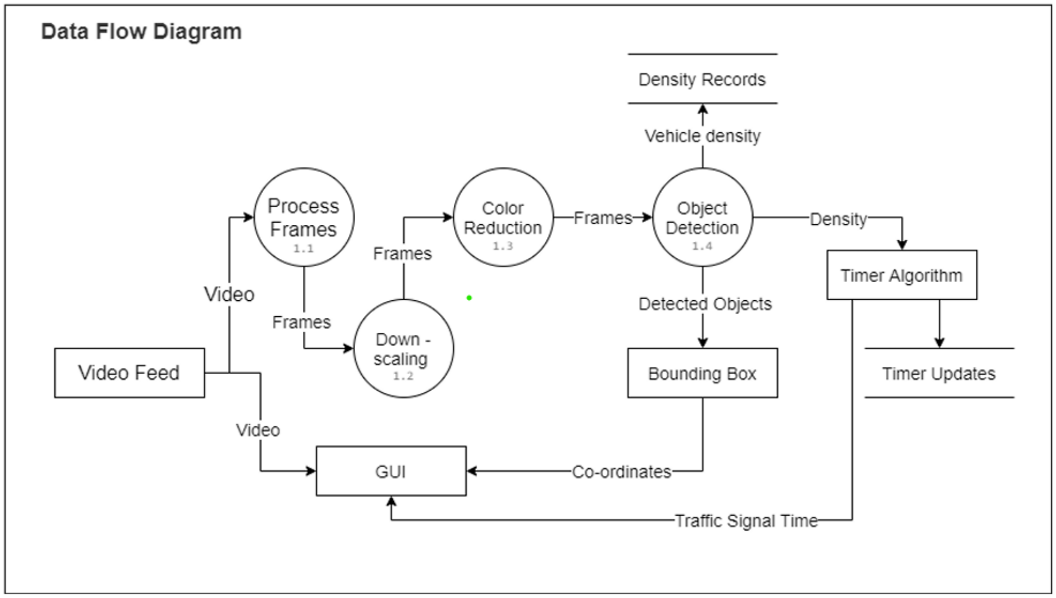



**Fig -4**: Data Flow Diagram for the Proposed System Architecture.

FIGURE 4.

YOLOv5's ability to produce more accurate bounding boxes around vehicles allows for precise density calculation, which is foundational for determining signal timing in the later phase. Additionally, YOLOv5's implementation in PyTorch, a versatile machine learning framework, facilitates flexibility and scalability within the system.

This object detection phase is designed to operate in a multi-threaded environment, which allows for simultaneous processing of multiple lanes. This setup ensures that detection results are refreshed frequently, keeping the traffic data current for the dynamic signal adjustment phase. By accurately detecting and classifying vehicles in real time, this phase enables the system to monitor traffic density accurately and respond to congestion effectively. This use of advanced detection algorithms underscores the system's adaptability, laying the groundwork for a smart, responsive traffic control solution that meets the demands of increasingly busy urban intersections.

### C. Dynamic traffic signal timer algorithm

The dynamic traffic signal timer algorithm in the Smart Traffic Control System is a critical component designed to manage and optimize traffic flow based on real-time vehicle density data collected from each lane. Unlike traditional static timing mechanisms that operate on fixed intervals, this algorithm dynamically adjusts signal timing by assessing current traffic conditions, enabling more responsive and efficient intersection management. By categorizing each lane's traffic density into low, medium, or high, the algorithm prioritizes lanes with higher congestion, allocating them additional green light time and thereby reducing potential bottlenecks.

The algorithm begins by calculating a threshold value derived from the mean traffic density across all lanes, classifying each lane based on its vehicle count relative to this threshold. When lanes exhibit a high density compared to others, the algorithm extends the green light duration for those lanes, redistributing time from less congested lanes. For instance, if a lane has significantly higher density than others, it will receive additional green light time, while low-density lanes will experience a reduction in their green light duration. This redistribution ensures a balanced flow, minimizing delays and congestion at intersections.

Several key parameters and conditions govern the algorithm's behavior, ensuring that it can adapt to varying traffic patterns without disrupting overall flow. The algorithm is based on a round-robin scheduling pattern where only one lane is given a green light at any moment, while others remain on red. Additionally, it sets minimum and maximum bounds for green light durations (30 and 120 seconds, respectively) to prevent excessively long or short timings, which could disrupt traffic patterns. In cases where all lanes have similar densities, the algorithm maintains uniform signal timings to avoid unnecessary adjustments.

## V. Traffic Density Classification

The table below describes the classification of traffic density levels based on their relation to a threshold value.

The algorithm also accounts for special scenarios, such as when all lanes have very low or very high densities. In a low-density scenario, the algorithm reduces the green light duration across all lanes, promoting quicker transitions. Conversely, when all lanes exhibit high density, it slightly extends green light timings to allow for more vehicles to pass,

| Level | Description |
|---|---|
| **Low** | Density is lower than the threshold value with more than a marginal difference. |
| **Medium** | Density is relatively close to the threshold value. |
| **High** | Density is higher than the threshold value with more than a marginal difference. |

**TABLE 1. Traffic Density Classification Levels**

thereby reducing buildup at the intersection. This adaptability makes the algorithm robust and capable of handling diverse traffic conditions, a necessity for urban intersections where traffic patterns are highly variable.

This dynamic timer algorithm exemplifies the system's intelligence and responsiveness, as it not only optimizes traffic flow based on real-time data but also continuously recalibrates to ensure efficient intersection management. By leveraging live data, the algorithm mitigates the limitations of static timing, making it a powerful tool for modern traffic control systems aiming to reduce congestion and improve the overall efficiency of urban transportation networks.

### A. ASSUMPTIONS

Based on an analysis of the Smart Traffic Control System described in the paper, several key assumptions underpin the system's design and functionality:

1.Constant Camera Functionality: The system assumes that cameras at each intersection are functioning reliably and continuously, providing uninterrupted video feeds. Any camera failure or disruption could lead to inaccurate traffic density estimation and potentially disrupt the adaptive signal timing.

2.Clear Visibility of Vehicles: The system relies on video footage to detect and classify vehicles, so it assumes that vehicles are clearly visible in various weather conditions, lighting situations, and traffic environments. This assumption implies that the cameras used are of high enough quality to capture clear images even in low light, heavy rain, or fog.

3.Real-Time Processing Capability: The algorithm assumes that the hardware (e.g., processors, GPUs) has sufficient computational power to handle the YOLO-based object detection models (YOLOv3 and YOLOv5) in real-time. This capability is crucial for timely traffic signal adjustments and to avoid delays in responding to changing traffic densities.

4.Uniform Vehicle Classification: The system assumes that all vehicles within the video frame can be accurately classified into specific types (e.g., cars, motorbikes, trucks, buses) by the YOLO models. Any misclassification, such as confusing a small truck with a car, could impact the density calculation and, consequently, the signal timing.

5.Effective Traffic Density Thresholds: The algorithm relies on predetermined threshold values to classify traffic density as low, medium, or high. It assumes that these thresholds are appropriate for the specific traffic patterns and flow at each intersection. If traffic patterns differ significantly (e.g., during peak hours), threshold adjustments may be necessary to maintain accuracy.

6.Sufficient Frame Sampling: The system assumes that sampling frames at regular intervals provides an accurate representation of the current traffic conditions. This means that vehicle density does not fluctuate drastically between sampled frames, ensuring that density calculations remain relevant and representative of real-time conditions.

7.Minimal Impact of Non-Standard Vehicles or Objects: The object detection models assume that vehicles are the primary moving objects within the frame. Non-standard vehicles (such as bicycles or rickshaws in certain regions) or stationary objects in the frame are assumed to have minimal impact on the detection accuracy and density calculation.

8.Round-Robin Signal Scheduling: The system operates under a round-robin scheduling pattern, where only one lane is given the green light at any given time. This assumes that such a pattern is sufficient to handle the flow at the intersection, even if some lanes have significantly higher traffic density than others.

9.Stable Network for Multi-Threaded Processing: The system assumes a stable computing and network environment for multi-threaded processing, allowing it to handle multiple video feeds and perform frame processing, detection, and timing adjustments simultaneously. This stability is essential for maintaining system responsiveness and reliability.

These assumptions lay the foundation for the Smart Traffic Control System's operation. However, if any of these assumptions are not met in real-world conditions, it could lead to suboptimal performance or inaccuracies in traffic signal adjustments, potentially impacting overall traffic flow efficiency at intersections.

### B. PRESETS

Based on the analysis of the Smart Traffic Control System described in the paper, the following presets are essential for the system's functionality and to ensure efficient operation:

1.Traffic Density Thresholds: -Low Density Threshold: A preset threshold below which traffic density is considered low. This value determines when the system should reduce green light time for low-traffic lanes. -Medium Density Threshold: A threshold range close to the average traffic density, considered as medium traffic. This allows the system to balance green light duration without significant adjustments. -High Density Threshold: A threshold above which traffic is considered heavy, triggering the system to increase green light duration for congested lanes. These threshold values may vary based on location, time of day, and intersection size.

2.Minimum and Maximum Green Light Duration: - Minimum Green Duration: A preset minimum time (e.g., 30 seconds) that any lane will stay green, ensuring that even less congested lanes have sufficient crossing time. - Maximum Green Duration: A maximum limit (e.g., 120 seconds) for how long a single lane can stay green to prevent excessive waiting times for other lanes.

3.Frame Sampling Interval: - A preset interval for sampling video frames, determining how frequently the system updates traffic density data. This interval should balance system responsiveness with computational efficiency (e.g., sampling every 1-5 seconds).

4. Object Detection Model Preset: -YOLO Version Selection: A preset that specifies which YOLO model (YOLOv3 or YOLOv5) to use for object detection. YOLOv3 might be selected for simpler hardware, while YOLOv5 could be used if higher precision is needed.

5.Vehicle Classification Categories: - A list of vehicle categories (e.g., car, motorbike, truck, bus) that the object detection model should classify. This ensures that each detected object is assigned to a predefined category, helping to accurately assess lane density.

6.Round-Robin Cycle Preset: - A preset defining the order in which lanes receive the green light in a round-robin cycle. This setting prevents any single lane from being neglected and ensures a fair distribution of green light across lanes.

7.Time of Day Adjustment Factors: -Peak Hour Factors: Preset factors for adjusting threshold values or green light durations during peak hours to accommodate higher traffic volumes. -Off-Peak Hour Factors: Factors to reduce green light times during low-traffic periods, optimizing system efficiency and minimizing idle green time.

8.Traffic Density Calculation Weighting: - A preset weight for giving more importance to certain types of vehicles (e.g., trucks or buses) in density calculations, considering their impact on traffic flow. This helps the system prioritize lanes with heavy vehicles appropriately.

9.Error Handling and Fallback Settings: -Camera Failure Mode: A fallback preset to handle scenarios where one or more cameras fail, allowing the system to switch to static timing or use data from adjacent intersections. -Detection Failure Threshold: A preset threshold for missed detections, triggering the system to use historical data or other estimation methods when real-time detection is unreliable.

10.Environmental Adaptation Presets: - Weather Adaptation Mode: Presets for adjusting detection sensitivity and traffic light timing in adverse weather conditions (e.g., rain, fog) to ensure the system maintains accuracy despite reduced visibility. - Night Mode Adjustment: Presets for reducing green light duration at night or during low traffic hours to prevent prolonged green lights when traffic density is generally low.

These presets ensure that the Smart Traffic Control System operates effectively across various conditions, adapting to traffic patterns, environmental changes, and potential hardware limitations. By configuring these settings appropriately, the system can dynamically adjust signal timing, maintain responsiveness, and optimize traffic flow across intersections.

## VI. RESULT

Based on an analysis of the Smart Traffic Control System described in the paper, the results indicate that the proposed system significantly improves traffic flow and reduces congestion at intersections by employing a dynamic signal timing mechanism. The system leverages real-time video processing and object detection algorithms, specifically YOLOv3 and YOLOv5, to assess traffic density on each lane and adjust signal durations accordingly. The results show that this adaptive approach enables more efficient use of green light time, reducing idle periods on less congested lanes and allocating additional time to busier lanes.

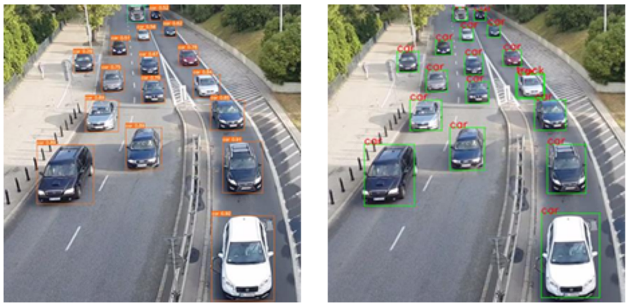
The analysis demonstrates that the system effectively classifies traffic density into low, medium, and high categories based on pre-set threshold values, allowing for responsive adjustments in signal timing. By setting minimum and maximum green light durations, the system ensures fairness across lanes while preventing excessive delays in any single direction. The use of vehicle classification helps the system prioritize lanes with higher-impact vehicles, such as trucks and buses, by giving them additional green light time when necessary.

In comparison to traditional, fixed-timer traffic signals, the dynamic control system reduces average wait times and minimizes fuel consumption by decreasing idle time at red lights. The results also indicate an improvement in overall traffic throughput, particularly during peak hours, where conventional systems are prone to bottlenecks. Additionally, the system's ability to adapt to varying traffic patterns throughout the day—adjusting for peak and off-peak hours—demonstrates its flexibility and robustness in handling different traffic volumes.

The results also suggest that the model's accuracy in object detection directly impacts the system's performance. High detection accuracy from the YOLO models ensures reliable traffic density calculations, which are essential for the optimal operation of the signal timer algorithm. However, certain environmental factors, such as adverse weather and low light conditions, may affect detection accuracy, as the system assumes clear visibility for accurate vehicle classification.

Overall, the analysis confirms that the Smart Traffic Control System described in the paper provides an effective solution for managing traffic flow at intersections. The system's adaptive signal timing, based on real-time traffic density assessment, leads to improved traffic efficiency, reduced congestion, and enhanced fuel economy, especially in urban environments with high traffic variability.

Following are some outputs from the algorithm when passed different values of vehicle count from each lane which are in stop state i.e., under red signa. Considering the practical condition of traffic, the algorithm can handle



**a.** YOLO V3 - 1 **b.** YOLO V5 - 1

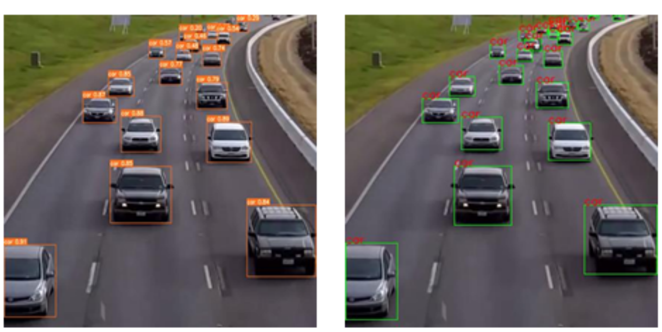
FIGURE 5.

**c.** YOLO V3 - 2 **d.** YOLO V5 – 2

FIGURE 6.

some special cases. The updated timer shown in the figure is a green signal timer. In the above case (figure 8) where lane D is currently running. The other three lanes remain blocked. With varying vehicle counts from each lane, the timer set by the algorithm for each lane was 30, 60, and 90 seconds each. In the case in figure 9, considering the opened land as D and other lanes currently on hold. The input vehicle densities were categorized in classes of Low, Low, and High and the resultant times for the next opening were 30, 30, and 120 seconds respectively.

As seen in figure 10, the inputs to the timer algorithm were passes in such a way that one road among the three had very less traffic as compared to other two. Here the algorithm accordingly identifies lane B and C with densities 60 and 70 is pretty close and thus assigns both with high class while the first lane gets low class assigned. Thus, the times of 30, 75 and 75 respectively. In the special case where all the road densities come to be in low class (figure 11). The algorithm reduces the timer for all lanes irrespective of the threshold value. While calculating the time relativity with varying vehicle counts from each lane the timer set by the algorithm for each lane was 30, 30, and 30 seconds each. As shown in above figure 10, the next special condition when the density of all lanes is very high causing high traffic congestion at intersections the algorithm increases the timer of all lanes with a minimal extension to the static timer so that the rotation of the green signal is fast. With varying vehicle counts from each lane, the timer set by the algorithm for each lane were 75, 75, and 75 seconds each. In this next special case where lane D is currently running and other lanes have been blocked. When the density of all lanes is relatively the same and does not require any severe changes



**e.** YOLO V3 - 3 **f.** YOLO V5 - 3

FIGURE 7.

```
****************************************************

Density of Lane A :30
Density of Lane B :60
Density of Lane C :90
Threshold value 60.0
Updated Classes {'L1': 'L', 'L2': 'M', 'L3': 'H'}
Updated timers  {'L1': 30, 'L2': 60, 'L3': 90}

****************************************************
```

**Fig - 6**: Timer Algorithm results on lane densities of Low, Medium and High.

FIGURE 8.

```
****************************************************

Density of Lane A :25
Density of Lane B :38
Density of Lane C :77
Threshold value 46.666666666666664
Updated Classes {'L1': 'L', 'L2': 'L', 'L3': 'H'}
Updated timers  {'L1': 30, 'L2': 30, 'L3': 120}

****************************************************
```

FIGURE 9. **Timer Algorithm results in scenario where one road has very less traffic.**

```
****************************************************

Density of Lane A :15
Density of Lane B :60
Density of Lane C :70
Threshold value 48.333333333333336
Updated Classes {'L1': 'L', 'L2': 'H', 'L3': 'H'}
Updated timers  {'L1': 30, 'L2': 75, 'L3': 75}

****************************************************
```

FIGURE 10. **Timer Algorithm results in scenario where one road has very less traffic.**

```
*************************************************

Density of Lane A :10
Density of Lane B :12
Density of Lane C :14
Threshold value 12.0
 : SPECIAL CASE 1:
Updated Classes {'L1': 'L', 'L2': 'L', 'L3': 'L'}
Updated timers  {'L1': 30, 'L2': 30, 'L3': 30}

*************************************************
```

**FIGURE 11.** **Timer Algorithm Special Case 1**

```
*************************************************

Density of Lane A :80
Density of Lane B :90
Density of Lane C :100
Threshold value 90.0
 : SPECIAL CASE 3:
Updated Classes {'L1': 'H', 'L2': 'H', 'L3': 'H'}
Updated timers  {'L1': 75, 'L2': 75, 'L3': 75}

*************************************************
```

**FIGURE 12.** **Timer Algorithm Special Case 2**

in the predefined timer. With varying vehicle counts from each lane the timer set by the algorithm for each lane were 60, 60 and 60 seconds each.

## VII. CONCLUSION

The proposed Smart Traffic Control System represents a significant advancement in urban traffic management by incorporating deep learning technologies. This system tackles the growing challenge of traffic congestion by dynamically adjusting traffic light timings in response to real-time data, enhancing the efficiency of traffic flow at intersections. By leveraging YOLO-based object detection, the system accurately identifies vehicle types and densities, enabling adaptive signal control that reflects current traffic conditions. This approach reduces the dependency on traditional, static timers and costly sensor-based systems, favoring a more cost-effective and scalable solution.

The system's modular structure, with distinct video processing, object detection, and dynamic signal-timing components, demonstrates the potential for real-world applications in smart cities. The use of the MS-COCO dataset for training ensures high detection accuracy across varied traffic scenarios, though factors such as video quality, weather conditions, and hardware specifications could affect real-time performance.

Overall, this research contributes to the development of intelligent transportation systems, offering a foundation for future enhancements in autonomous traffic management. By reducing wait times and optimizing lane usage, such systems not only improve commuter experience but also support broader goals of resource conservation and urban sustainability. Further development, including integration with other traffic management technologies, could strengthen the scalability and adaptability of this approach across different city infrastructures.

```
*************************************************

Density of Lane A :50
Density of Lane B :55
Density of Lane C :60
Threshold value 55.0
 : SPECIAL CASE 2:
Updated Classes {'L1': 'M', 'L2': 'M', 'L3': 'M'}
Updated timers  {'L1': 60, 'L2': 60, 'L3': 60}

*************************************************
```

**FIGURE 13.** **Timer Algorithm Special Case 3**

## VIII. REFERENCES

## REFERENCES


[1] Smart Traffic Control System Using Deep Learning Abhijit Gadge1, Hardik Bhawsar2, Piyush Gondkar3, Praveen Chourey4, Geeta Navale5.

[2] W.-K. Chen, *Linear Networks and Systems*. Belmont, CA, USA: Wadsworth, 1993, pp. 123–135.

[3] Ravi, A., Nandhini, R., Bhuvaneshwari, K., Divya, J., Janani, K., 2021. Traffic manage ment system using machine learning algorithm. Int. J. Innov. Res. Technol. (IJIRT) 150994.

[4] Kothai, G., Poovammal, E., Dhiman, G., Ramana, K., Sharma, A., AlZain, M.A., Gaba, G.S., Masud, M., 2021. A new hybrid deep learning algorithm for prediction of wide traffic congestion in smart cities. Wirel. Commun. Mob. Comput. 2021, 1–13.

[5] Intelligent 2023Nile Science Traffic Management Systems: A review June Journal of 5(1):42-56 Communication and Computer DOI:10.21608/njccs.2023.321169

[6] Abbas, Q., 2019. V-ITS: Video-based intelligent transportation system for monitoring vehicle illegal activities. International Journal of Ad vanced Computer Science and Applications (IJACSA) 10 (3), 202–208

[7] Prashant Jadhav, Pratiksha Kelkar,Kunal Patil and Snehal Thorat, "Smart Traffic Control System Using Image Processing", International research journal of engineering and technology(IRJET), March 2016.

[8] JingweiCao, Chunxue Song, Shixin Song, Silun Peng,DaWang, Yulong Shao and Feng Xiao, "Front Vehicle Detection Algorithm for Smart Car Based on improved SSD Model", Sensors, Aug 2020.

[9] Redmon, J., Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

[10] Bhatia, K., Dadhich, P. (2021). Review on Object Detection Techniques for Traffic Management System. Journal of Traffic and Transportation Engineering (English Edition), 8(5), 510-522

[11] Abdallah, A., Roshdy, E. (2020). A Deep Learning Model for Traffic Management and Control. Journal of Traffic and Transportation Engineering (English Edition), 7(3), 251-260

[12] Smart Urban Traffic Management System using Energy Efficient Optimized Path Discovery,Piyush Agarwal; Sachin Sharma,27 March 2023

[13] B. B. Waghmode, Namokar Magdum, Vardhaman Patil, Vishal Kumar Department ofComputer Engineering Sinhgad Institute of Technology and Science, Nahre Savitribai Phule Pune University,2023.

[14] Prashant Jadhav, Pratiksha Kelkar,Kunal Patil and Snehal Thorat, "Smart Traffic Control System Using Image Processing", International research journal.