Journal of Computer Science

Research Article

FuzzyNET: A Fuzzy Logic-Based Framework for Enhancing Urban Efficiency in Smart Cities

Hemant Kumar, Safdar Tanweer, Parul Agarwal, Naseem Rao, Jawed Ahmed and Syed Sibtain Khalid

Department of Computer Science and Engineering, Jamia Hamdard, New Delhi, India

Article history
Received: 10-02-2026
Revised: 02-05-2026
Accepted: 22-05-2026

Corresponding Author:
Safdar Tanweer
Department of Computer Science and Engineering, Jamia Hamdard, New Delhi, India
Email: safdartanweer@jamiahamdard.ac.in

Abstract: The smart city paradigm emphasizes the strategic use of IoT, Big Data, and digital technologies to optimize urban operations and improve service delivery for citizens. Embedding digital technologies within operational workflows is driven by the objective of improving efficiency and long-term sustainability. The core transformation is Artificial Intelligence (AI) that revolutionizes management of cities through automation and instantaneous decisions based on data. Artificial intelligence can significantly reshape essential areas such as power infrastructure, waste services, and public safety. For urban mobility, deep learning has been used for advanced image classification tasks. AI-based traffic signal recognition also has the potential to facilitate optimal vehicle flow, resulting in less fuel consumption and less carbon emissions. Flexible traffic signals that respond to real-time traffic density can help cities significantly decrease idling and air pollution. To address these challenges, this paper proposes a novel two-phase framework: First, a hybrid CNN–Fuzzy model for robust traffic sign classification, and second, a dynamic traffic signal optimization system utilizing YOLOv8 for real-time vehicle density estimation. In the first phase, a hybrid CNN–Fuzzy is proposed in order to improve classification performance using the feature extraction capability of Convolution Neural Networks (CNNs) and fuzzy decision support. Strong model generalization was ensured. The overfitting was mitigated by rigorous preprocessing with data augmentation, normalization and some moderate dropout regularization applied at all levels. Quantitatively, the proposed hybrid method demonstrates clear superiority over recent state-of-the-art models; it achieved an impressive validation accuracy of 99.72% and a minimal validation loss of 0.0126, significantly outperforming the 96.12% validation accuracy and 10.95 validation loss reported by the baseline model. These performance criteria further support the effectiveness of fuzzy-aided deep learning in identifying such a crucial decision-making process. Lastly, the work offers an approach that is scalable to energy-efficient traffic control as a component of smart urban mobility in conjunction with environmental sustainability to lead towards CO2 reduction.

Keywords: Automation, CNN, Images, Fuzzy, Smart City

Introduction

A smart city makes use of technology, data analytics, and IoT (Internet of Things) to enhance the infrastructure and ensures the optimal use of resources for better living (Rout et al., 2020). A smart city aims to make the city sustainable, efficient, and livable by integrating digital innovations into public services. Artificial Intelligence (AI) has a role of change to bring about in smart cities. A Smart city provides automation, real-time decision, solutions to improve the different aspects of a city, such as energy efficiency, public safety, traffic management, waste management, and governance, to make the city smarter and more sustainable. Intelligent traffic management is one of the main aspects of AI in smart cities. AI adaptive traffic lights learn real-time traffic congestion patterns and optimize traffic to minimize bottlenecks and emissions. AI improves public transport through predictive analytics for better route planning and fleet management (de Oliveira et al., 2022). AI-based surveillance systems with facial recognition and behavior analysis help law enforcement detect anomalies, prevent crime, and ensure public safety. AI also transforms waste management with the help of intelligent sensors and machine learning algorithms. It reduces the garbage collection cycle and minimizes fuel consumption and costs. In energy management: AI-driven smart grids predict power consumption and optimize supply or fault detection in the power line for an increase of energy efficiency (Karyaningsih and Rizky, 2020). There is also environmental monitoring where artificial intelligence can monitor how clean the air and water around a city can be, but also measure up noise levels as well, so authorities are able to take action on those issues. AI promotes e-governance through automation into many processes such as chatbots for citizen requests answering and service delivery enhancement (Jafari et al., 2022).

AI-enabled disaster management systems help to analyze climatic conditions and also forecast natural disasters (Tomar et al., 2018). The use of mobile applications and AI powered virtual assistants in citizen participation that give real-time information about city services, schedule of public transport, warnings on emergencies, etc. AI also helps in the planning of cities so that they will be able to effectively use their land, grow infrastructure, and allocate resources. AI transforms smart cities to be more citizen-centric, efficient, and sustainable, towards a resilient, green, and secure urban environment. AI technologies will increasingly shape the character of urban life by making cities more responsive, adaptive, and smart in addressing the needs of urban dwellers. Fig. 1 illustrates the overall operations of smart cities.

The primary motivation of this research is to bridge the gap between static image classification and dynamic, real-time traffic management in smart cities. While existing intelligent transportation systems rely heavily on rigid signal timers or computationally prohibitive architectures, there is a pressing need for a lightweight, adaptive solution that can handle real-world uncertainties.

This research aims to design a comprehensive, energy-efficient model by combining the powerful feature-extraction functions of the deep learning with the decision-making ability of fuzzy logic. The rationale behind this is to go beyond theoretical accuracy and provide a practical, scalable AI system that proactively decongests urban intersection, reduces vehicle idling as well as carbon emission.

Problem Statement

Traditional traffic lights are programmed to follow a fixed program, wasting time at an intersection when few or no vehicles are going through.

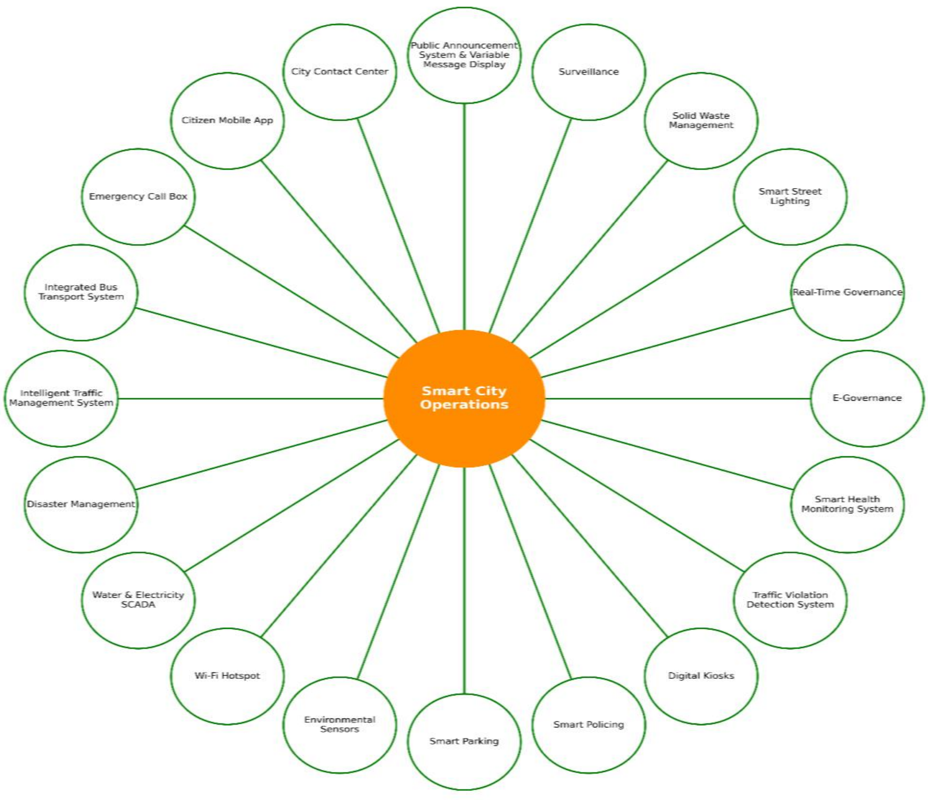

Fig. 1: Operations required for smart cities

This waste opens space to burn more fuel, a greater carbon footprint, and unwarranted delays. Inadequate green time during the rush hours causes traffic jam and excessive green time at low traffic usage, which translates to road way resource and energy waste. Such fixed signal control measures fail to embrace dynamic vehicle performance and heralds a high intelligence in management of traffic.

Contributions of the Study

This study proposes a two-phase model to improve urban mobility and address the limitations of conventional traffic management systems and existing AI traffic control methods. The principal contributions are:

Novel Two-Phase AI Architecture: We present an integrated, all-encompassing framework capable of connecting static environmental awareness with dynamic traffic control. In Phase 1, the traffic-sign classification module generates fuzzy confidence levels (Low, Medium, or High) representing the semantic certainty of detected regulatory conditions to address real-world ambiguity. Rather than operating in isolation, this contextual environmental awareness provides adaptive decision-support information to Phase 2, where a YOLOv8-driven fuzzy traffic controller optimizes green-time durations in real time based on live vehicle density and inferred traffic context. Together, these phases form a holistic intelligent transportation framework.

Hybrid CNN-Fuzzy Inference System (FuzzyNET): This research introduces a sophisticated hybrid system of traffic signs classification based on features extraction in three different Convolutional Neural Networks (Standard CNN, CNN + Dropout, and CNN + Augmentation). The model is able to handle uncertainty, overlapping class distribution, and overfitting is reduced to a minimum by passing these features through a fuzzy inference system making use of scaled Gaussian membership functions and 14 different rule sets.

Dynamic YOLOv8-Driven Signal Optimization: Our algorithm can optimize traffic signals in real-time and is based on a fine-tuned YOLOv8 model, which was trained on the Cityscapes dataset. This adaptive system is dynamically varied to change the duration of green lights based on real-time vehicle count of the road that effectively alleviated intersection congestion, idle time, and carbon emission.

Literature Review

The Artificial Intelligence (AI) is prevalently considered a transforming paradigm for the Smart City developments especially in Intelligent Transportation Systems (ITS), traffic congestion alleviation, and real-time urban mobility management. Different research studies have examined different methodologies, models, and applications of AI for traffic control optimization and smart city infrastructure optimization.

Englund et al. (2021) explored AI and IoT's application to break traffic congestion in smart cities. Their study highlights how sensor-activated objects within an IoT network collect and exchange data to facilitate autonomous traffic decision-making. An AI study proposed a low-complexity AI-based traffic control solution, although it had no experimental demonstration or implementation outline.

Soomro et al. (2018) came out with a detailed survey on AI, Machine Learning (ML), and Deep Reinforcement Learning (DRL) applications in a range of smart city applications. Their study covered traffic control besides cyber security, smart grids, Unmanned Aerial Vehicles (UAVs) for communications, and smart healthcare systems. They identified how AI-facilitated decision-making assumes a crucial position in urban planning and policy decision-making.

Agarwal et al. (2015) emphasized the use of AI for developing an Intelligent Transportation System (ITS) for smart cities. They saw problems such as congestion, an inefficient public transit system, a lack of parking, and increasingly frequent road accidents, and were proponents for AI-enabled solutions to improve urban mobility. Despite their study yielded rich knowledge with respect to AI application in IT, there was no actual case studies or concrete plans from them.

Tarawneh et al. (2023) investigated the Indian transport sector, indicating a requirement for AI- infused intelligent transport aids. The paper highlighted some of the most important issues in Indian cities traffic congestion, environmental pollution, and public transport inefficiency. What they proposed included AI-powered traffic flow optimization, smart parking, and predictive accident prevention. While their paper was used as a support for theoretical constructions. It lacked real-time AI implementation and test validation.

To solve the problem of real-time traffic control and accident detection of smart cities, (Ponnusamy et al., 2024) suggested an AI-IoT based architecture. They proposed to solve the problem by profiling drivers, checking their vehicle health and road safety by using IoT sensors and machine learning diagnostics. The results of the simulated work showed advantages in road safety, monitoring of the health of vehicles and traffic jams, but the article identified that extensive sensor networks and rapid data processing are necessary to implement the system.

The role of AI and ML in smart cities has been extensively studied by Navarathna and Malagi (2018) which emphasized policy making, decision making, and infrastructure development. Their article was a survey report for ICT advances in intelligent transport, energy and urban sustainability.

Chen and Zhang (2022) proposed a novel AI-TMS with Deep Belief Networks (DBN) for online congestion prediction and flow control. Their system integrated real-time data sensor processing with AI decision making to reduce vehicle halting, increase road utilization, and passenger mobility. They reported an increased degree of accuracy with their results about congestion prediction in comparison to conventional methods. Scaling may be challenging for the model due to its reliance on heavy data processing and high-level material calculations.

Kommineni and Baseer (2024) offered a detailed overview of IoT, Big Data, and AI enabled solutions for smart city infrastructure. Their research explained how AI enabled solutions for smart governance, public safety, and the efficient management of urban mobility. While their work attributed meanings to some of the considerations in AI-supported city plans, it did not describe how traffic management can be implemented.

Dash et al. (2018) has touched upon the challenge of traffic flow prediction through deep learning and Deep Belief Networks (DBN). They have used past traffic data to forecast and predict congestion trends. Their model has shown to be more precise in prediction and improved performance compared to other AI-powered traffic prediction algorithms like Neuro-Fuzzy C-Means (FCM) or conventional Convolutional Neural Networks (CNNs). Nonetheless, their model, when used in smaller towns, might not be realistic because of the significant data and computing needs.

Nikitas et al. (2020) examined adaptive traffic signal control models based on AI to control the timing of traffic signals depending on traffic congestion. Their research investigated in the timing of traffic lights that could decrease congestion, save fuel, and allow easier access to the cities. They also indicated that such adaptive systems need large scale network consisting of data collection service that is real time and a huge infrastructure investment. Table 1 presents an overview of prior studies in the smart city field.

Identification of Research Gap

Even though the level of AI-enabled smart cities technologies is incredibly advanced, the critical analysis of the literature indicates that there is a remarkable research gap.

Table 1: Previous work done in the domain of smart cities

Reference	Objective	Methodology	Advantages	Limitations
(Englund et al., 2021)	AI and IoT integration for traffic congestion reduction in smart cities	Literature review and AI-based solution proposal	Improved traffic flow and congestion management	Fails to address uncertainty in sensor data overlap
(Soomro et al., 2018)	AI, ML, and DRL for smart city applications	Survey of AI applications in ITS, cyber-security, and smart grids	Comprehensive overview of AI techniques in urban management	Does not focus on a specific AI model for traffic control
(Agarwal et al., 2015)	AI for Intelligent Transport Systems (ITS) in smart cities	Review and application discussion	Provides insights into AI's role in ITS development	Limited case studies and practical implementations
(Tarawneh et al., 2023)	AI for transport systems in Indian smart cities	Analysis of urban transport challenges and AI-based solutions	Addresses country-specific issues and policy recommendations	High latency in edge-computing scenarios
(Ponnusamy et al., 2024)	AI-driven traffic management for congestion reduction and safety	IoT and AI-based traffic analysis framework	Real-time driver monitoring and road condition analysis	Requires extensive sensor networks and infrastructure
(Navarathna and Malagi, 2018)	AI and ML applications in smart city development	ICT-based survey of AI's role in smart infrastructure	Highlights AI's contribution to various urban services	Lacks specific implementation details for ITS
(Chen and Zhang, 2022)	AI and Deep Belief Networks (DBN) for traffic congestion prediction	AI-TMS model integrating real-time traffic sensors	Reduces congestion and improves traffic efficiency	DBN architectures are computationally prohibitive for real-time video
(Kommineni and Baseer, 2024)	AI and IoT for smart city infrastructure	Review of AI, ML, IoT, and Big Data applications	Covers multiple aspects of smart city development	Broad focus with limited transport-specific solutions
(Dash et al., 2018)	Deep Learning for urban mobility and congestion prediction	Deep Belief Network trained on historical traffic data	High-accuracy congestion prediction	Requires large datasets and high computational power
(Nikitas et al., 2020)	Adaptive AI traffic signal control for congestion management	AI-based dynamic signal timing adjustments in real time	Improves traffic efficiency	Relies on rigid thresholds instead of continuous fuzzy adaptation

The main drawbacks of existing methodologies have been identified as either being more of a theoretical model rather than an experimental validation (Englund et al., 2021; Tarawneh et al., 2023) or being based on a complex architecture (Chen and Zhang, 2022; Dash et al., 2018). Moreover, existing literature typically considers the issues of static traffic sign classification and dynamic traffic flow optimization systematically, but lacks a framework that can efficiently tackle both of these concerns in concert with an object detector capable of operating on a distinctly limited scale (say, YOLOv8).

Materials and Methods

Research implementation is conducted in two interconnected phases. During the initial phase, a hybrid AI model is trained to categorize traffic signs using fuzzy logic and generate contextual traffic-awareness information. In the second phase, dynamic traffic signal optimization is performed using a YOLOv8-based fuzzy controller by considering real-time vehicle density along with this contextual information for adaptive signal-timing optimization. Together, these two phases contribute to reducing congestion, minimizing idle time, and improving traffic flow. Figure 2 presents an overview of the proposed two-phase FuzzyNET framework.

Dataset: Phase I

The German Traffic Sign Benchmark (GTSRB) is a single-image, multi-class classification dataset introduced at the International Joint Conference on Neural Networks (IJCNN) 2011.

The benchmark was designed to encourage broad participation from the research community, including researchers without access to domain-specific traffic datasets.

The dataset utilized for analyzing smart city traffic control is described in Table 2. The key characteristics of the benchmark are as follows:

- Single-image, multi-class classification problem
- Dataset comprising more than 40 distinct traffic sign classes
- Over 50,000 labeled images in total

Table 2 indicates that each image in the dataset is annotated with bounding box coordinates (Roi.X1, Roi.Y1, Roi.X2, Roi.Y2) that specify the area of interest corresponding to enclosing the traffic sign. Important attributes of the dataset are:

- Image width and height: Dimensions of original input images
- ROI coordinates: (X1, Y1) and (X2, Y2) which represent the bounding of the traffic sign
- ClassId: Integer identifier of the traffic sign category
- Path: File location of each image within the dataset

Phase 2: Cityscapes Dataset for Object Detection

Cityscapes Dataset is intended for the semantic interpretation of urban street environments, with polygon annotations, dense semantic segmentation, and instance segmentation for people and vehicles.

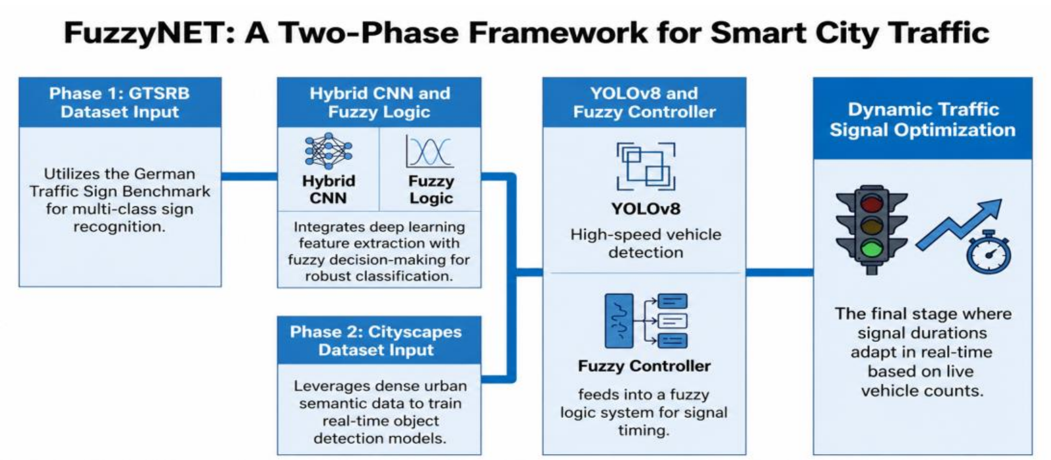

Fig. 2: Overview of the proposed two-phase connected FuzzyNET framework

Table 2: Key Attributes of the Traffic Sign Dataset

Height	Width	Roi.X1	Roi.Y1	Roi.X2	Roi.Y2	ClassId	Path
53	54	6	48	49	16	0	Test/00000.png
42	45	5	36	40	1	1	Test/00001.png
48	52	6	43	47	38	2	Test/00002.png
27	29	5	22	24	33	3	Test/00003.png
60	57	5	55	52	11	4	Test/00004.png

DOI: 10.3844/jcssp.2026.2189.2203

It has 30 unique classes with a diverse range of images from 50 cities during different seasons, shot in daytime with favorable to moderate weather conditions. The dataset has 5,000 highly annotated images and 20,000 roughly annotated images, along with metadata like GPS coordinates, ego-motion information, vehicle odometry, and ambient temperature. It also contains the previous and next video frames, so it can be used for sequence-based applications. The researchers have also added bounding box labels for pedestrians and image augmentations such as fog and rain. The dataset provides annotations suitable for pixel-level, instance-level, and panoptic semantic segmentation. One of the chief features of its labeling policy is that foreground objects are never holey; any visible background under an object, like tree leaves over the sky or clear car windows, is labeled as belonging to the foreground class so that annotations are consistent. In this work, the Cityscapes dataset is utilized to train and fine-tune the YOLOv8 (Jocher et al., 2023) model for vehicle and pedestrian detection in urban traffic scenes.

Pre-Processing

A series of preprocessing steps is applied to the corpus to make it suitable for training deep learning models with high accuracy and efficiency. These are the necessary steps to ensure image quality by normalizing pixel values into a good shape for model training. Fig. 3 shows the sample dataset records of different classes.

The primary pre-processing methods used are illustrated in Fig. 4 and are as follows:

Fig. 3: Sample dataset records from different traffic sign classes

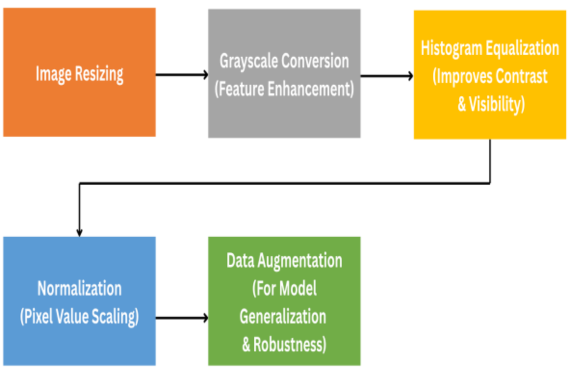

Fig. 4: Pre-processing steps

1. Image Resizing: Since the images in the dataset have different sizes, they are resized to (64,64,3). This is necessary for all input images to be the same shape as required by deep learning models like CNNs (Convolutional Neural Networks)
2. Grayscale Conversion (Feature Enhancement): For some models, converting from RGB to grayscale is a common practice for decreased computational complexity yet maintaining as much edge and shape information as possible. This is very effective in edge detection and contrast enhancement
3. Histogram Equalization (Improves Contrast & Visibility): In order to make the traffic signs visible for different illumination conditions, especially low light, the histogram equalization is applied on grayscale images. For color images, channel-wise adaptive histogram equalization (CLAHE) may be performed
4. Normalization (Pixel Value Scaling): Neural networks usually prefer that input pixel values are scaled to a standard range. The pixel values are normalized to [0,1] from original values in the range of 0–255 for faster convergence during training
5. Data Augmentation (For Model Generalization & Robustness): Data augmentation is used to achieve better performance of the model and to avoid overfitting This includes

- Rotation (e.g., ±15° to adjust for small changes in camera tilting)
- Flipping (horizontal/vertical flips to show perspectives)
- Brightness adjustments (simulating different lighting conditions)
- Zooming and cropping (for robustness over object detections)

The proposed model combines the CNN and fuzzy inference system to enhance image classification decisions. The model has a structured pipeline according to the flow diagram described in Fig. 5.

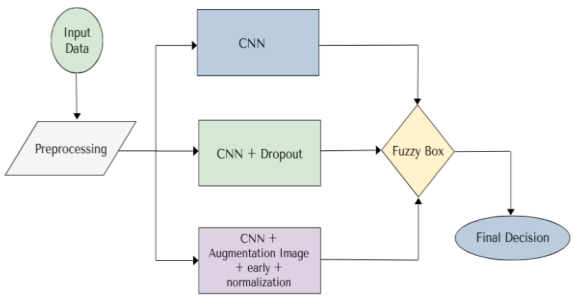

Fig. 5: Proposed model of CNN Fuzzy-Net. The central Fuzzy Box encapsulates the fuzzification of CNN confidence scores, evaluation against 14 rule sets, and centroid defuzzification to reach the final decision

Input Data and Preprocessing

The model receives the raw input signal, processes it to extract features, and ultimately classifies them. Preprocessing operations can include: Resizing and Normalization: Typical operations of normalizing input size and enhancing pixel intensity.

Data Augmentation: Increasing the amount of data to prevent overfitting by using synthetic data. Noise Suppression: Methods such as a Gaussian filter and adaptively thresholding to improve feature detection.

CNN-Based Feature Extraction

The model core is a sequence of CNN architectures, each contributing differently to feature extraction and generalization: Standard CNN: A deep CNN extracts hierarchical image features. CNN + Dropout: Dropout regularization is applied to mitigate overfitting by stochastically deactivating neurons during training. CNN + Augmentation + Early Stopping + Normalization: Augmentation: Improves model robustness using different versions of input images.

Early Stopping

Observes the validation performance and stops training when overfitting occurs. Normalization: Normalizes the distributions of features to help converge faster.

Integration of CNN Outputs With the Fuzzy Inference System

The transition of data from the deep learning architecture to the fuzzy logic controller is a crucial step in our proposed FuzzyNET framework. The interaction operates through the following structured pipeline.

CNN Probability Generation

First, each of the three distinct CNN architectures (Standard CNN, CNN + Dropout, and CNN + Augmentation) processes the input image and passes its extracted features through a final dense layer. Instead of a hard classification, each CNN utilizes a SoftMax activation function to output a probability vector. For n classes, this represents the confidence score P_i for the i^{th} class.

Fuzzification of Confidence Scores

The continuous probability scores (ranging from 0 to 1) from the three CNNs act as the crisp inputs (x) to the fuzzy layer. There are nodes everywhere $y = 1, 2, 3,., m$ for each input x. These crisp probabilities are mapped to the fuzzy linguistic variables (Low, Medium, High) using the scaled Gaussian membership function defined in Eq. 1 which guarantees membership values strictly bounded within [0,1]. The output is shaped by applying the variance (σ^2) and fixed centre parameters (μ) to these input features. Fig. 6 illustrates the Gaussian-based functions (gaussmf and gauss2mf) and the generalized bell membership function (gbellmf).

Fuzzy Layer Processing

The fuzzy layer consists of ab nodes, where a represents the number of fuzzy rules (14 in our proposed system), and b represents the number of input characteristics (the probability scores from the three CNN models). The fuzzified confidence scores from all three models are evaluated simultaneously against the 14 IF-THEN rules.

Aggregation and Final Classification: By evaluating the combined confidence of all three models simultaneously, the fuzzy system calculates an aggregated output weight for each class.

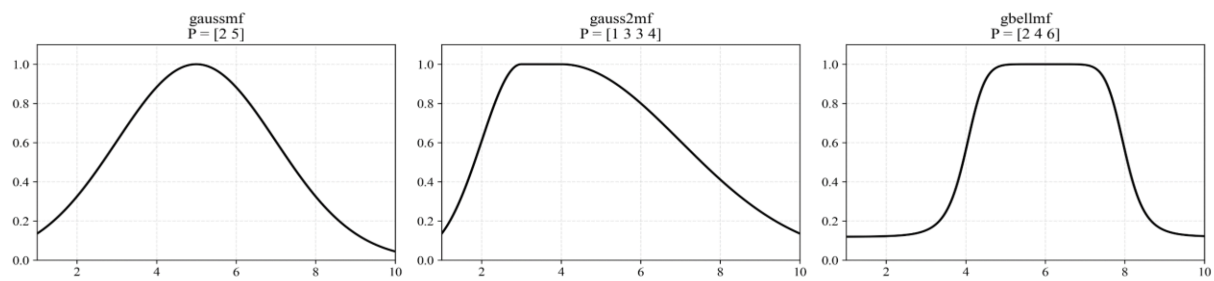

Fig. 6: Membership Function

The class with the highest defuzzified weight is selected as the final predicted label. If the fuzzy rules determine that all CNN models have "Low" confidence, the system flags the image as an "Uncertain Prediction" rather than forcing a potentially incorrect classification.

Fuzzy Inference System for Decision Making

The CNN modules output feature confidence scores, which are then fed into a Fuzzy Box that uses fuzzy decision rules for the final classification. The power of the fuzzy system lies in increasing decision robustness, making it ideal for dealing with uncertainty and overlapping class distributions. To formalize this process, the fuzzy inference system is structured into three distinct stages: Fuzzification, Rule Evaluation, and Defuzzification.

Fuzzification and Membership Functions

The crisp input values (the probability/confidence scores outputted by the Standard CNN, CNN + Dropout, and CNN + Augmentation modules) are first converted into fuzzy linguistic variables. The linguistic variables for the input confidence are categorized as Low, Medium, and High. The proposed system primarily makes use of the scaled Gaussian membership function to handle these variables. The scaled Gaussian distribution is evaluated using Eq. 1:

$$N(x) = e^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad (1)$$

Where μ is the fixed centre parameter controlling the location of the membership peak, and σ^2 is the fixed spread parameter controlling the width of the fuzzy region. These parameters define three linguistic regions Low, Medium, and High over the [0,1] confidence score domain and remain constant throughout inference.

Fuzzy Rule Base

The fuzzy inference system employs a comprehensive rule base to sharpen decision-making. Fourteen rule sets are produced in all, derived from the feature confidence scores of the three CNN models. Examples of these fuzzy rules include:

- IF (Feature Confidence from Standard CNN is High) AND (Feature Confidence from CNN + Dropout is High), THEN (Final Decision = Strong Prediction)
- IF (Feature Confidence from CNN + Augmentation is Medium) AND (CNN + Dropout is High), THEN (Final Decision = Probable Prediction)
- IF (Feature Confidence from CNN is Low) AND (CNN + Augmentation is Low), THEN (Final Decision = Uncertain Prediction; Request More Data)

Defuzzification and Final Decision

Once the 14 rules are evaluated, the resulting fuzzy output must be converted back into a crisp classification label. The proposed system employs the Centroid (Center of Gravity) defuzzification method. This technique aggregates the overlapping areas of the activated membership functions to find the center point, yielding a final crisp output. This hybrid scheme maximizes accuracy, reduces overfitting, and makes the model highly robust when deployed in real-world applications with noisy or uncertain environments.

Dynamic Traffic Signal Optimization Using Fuzzy Logic

The Smart Traffic Signal Optimization Algorithm changes how long traffic light stay on in real-time on the basis of vehicle counts and optimizes signal time. The system reduces congestion, minimizes idle time, and improves overall traffic flow, promotes energy efficiency, and urban sustainability.

The model pipeline shown in Fig. 7 is the process of an AI object detection system based on YOLO (You Only Look Once). The process starts with a dataset, which is image augmented and pre-processed to enhance model generalization and performance. The pre-processed images are then input into the AI model training process, where YOLOv8 is trained for object detection and classification tasks. After training, the model is used for detection and counting, detecting objects in real-time or batch processing applications. The system goes through a trained model evaluation process to determine its accuracy. and performance. Where needed, the results of the evaluation are fed back into the training process, fine-tuning the model for better detection accuracy. This cycle ensures ongoing improvement of the detection system.

The YOLOv8 model utilizes 125 epochs of fine-tuning and training with a 3-epoch warmup period using momentum of 0.8 to ensure a stable start, optimized for traffic. The batch size is set to 304 to allow for productive training while still accessing larger data sets for each iteration. Image size has been implemented at 640 in order to better manage memory, along with computational complexity.

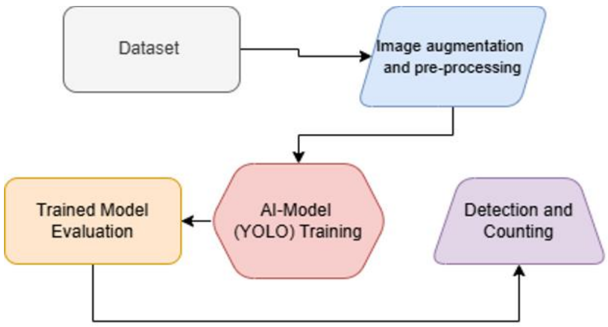

Fig. 7: Phase 2 AI model training using Cityscapes dataset

YOLOv8 is optimized with Stochastic Gradient Descent (SGD) and the learning rate is slowly decreased from 0.1 to 0.001. This allows a progressive decrease in learning rate for convergent stability. We have set the patience to 10 which is a period of time that we want to wait for to see if there is any improvement for this period of time, if there is no improvement then we want the model to stop training. The NMSIoU (Non-Maximum Suppression Intersection over Union) is 0.45 and is in the middle of the precision and recall in object detection. The model stabilizes the updates from training with a momentum of 0.937. Mask ratio = 0.1 showing masking methods are optimized. Last, the weight decay of 0.0005 is used to avoid overfitting and to continue to drive towards generalization. These settings aim to optimize YOLOv8's efficiency, accuracy, and stability during training. AI Model details related to training parameters are given in Table 3.

To clarify the integration of the proposed FuzzyNET framework, Phase 1 and Phase 2 operate as interconnected components rather than isolated workflows, as conceptually represented by the unified architecture in Figure 2. In practice, the multi-class traffic-sign recognition outcomes generated in Phase 1 (such as identifying pedestrian crossings or speed-regulated zones from the GTSRB dataset) provide contextual traffic-awareness information that supports the adaptive decision-making process in Phase 2. Algorithm 1 presents the baseline operational logic of the fuzzy traffic signal optimization framework using default boundary values for green-light duration. Within the proposed integrated architecture, the semantic understanding obtained from Phase 1 supports the contextual adaptation of traffic-signal policies, while the YOLOv8 model and fuzzy controller in Phase 2 utilize real-time vehicle density to determine the optimal green-light duration.

Table 3: AI Model details regarding training parameters

SNo.	Parameters	Values (YOLOv8)
1	Epochs	125
2	Warmup epochs	3
3	Warmup momentum	0.8
4	Batch size	304
5	Image size	640
6	Initial learning rate	0.1
7	Final learning rate	0.001
8	Patience	10
9	Optimizer	SGD
10	NMS IoU	0.45
11	Momentum	0.937
12	Mask ratio	0.1
13	Weight decay	0.0005

Algorithm 1: AI-Driven Fuzzy Traffic Signal Optimization

```
Initialize system settings:
min_green_time ← 10 ▷ Minimum green time (sec)
max_green_time ← 90 ▷ Maximum green time(sec)
function GET_VEHICLE_COUNT (YOLOv8 Output)
if camera sensor data available then
vehicle count ← YOLOv8_RealTime_Count ()
else
vehicle_count ← estimate_from_historical_data ()
end if
return vehicle_count
end function

function FUZZY_GREEN_TIME_CALC (vehicle_count)
    ▷ Step 1: Fuzzification
traffic_density ← Fuzzify(vehicle_count) into {Light,
 Moderate, Heavy}
    ▷ Step 2: Apply fuzzy rules
if traffic_density is Light then
    fuzzy_output ← Short Duration
else if traffic_density is Moderate then
    fuzzy_output ← Medium Duration
else if traffic_density is Heavy then
    fuzzy_output ← Long Duration
end if
    ▷ Step 3: Defuzzification (Centroid Method)
calculated_time ← Defuzzify(fuzzy_output)
    ▷ Step 4: Apply constraints
green time←max(min_green_time, min (calculated_time,
 max_green_time))
return green_time
end function

function UPDATE_TRAFFIC_SIGNAL ()
current count ← GET_VEHICLE_COUNT (YOLOv8
 Output)
optimized_green_time←FUZZY_GREEN_TIME_
 CALC (current_count)
APPLY optimized_green_time TO traffic_light
end function
while system_is_running do
UPDATE_TRAFFIC_SIGNAL ()
WAITS for next traffic cycle
end whiles
```

Experimental Setup and Implementation

All simulations and model training were done in Python 3 in order to rigorously test the proposed FuzzyNET framework. The two deep learning models, such as the one with combining CNN and YOLOv8, were developed with the help of the TensorFlow framework. Because of the complexity of the image processing and real-time object detection involved, the experiments were run more quickly on an NVIDIA RTX 3080 with 10GB VRAM, an Intel Xeon processor, and 32GB of system memory.

To achieve a high quality of model evaluation and avoid data leakage, the datasets were systematically split.

Phase 1 (GTSRB Dataset): The dataset, consisting of more than 50,000 labeled traffic sign images, is split into a 70% training, 20% validation, and 10% testing split. This separation enabled the stage-by-stage hyperparameter optimization and gave a totally unseen portion of data to assess the generalization potential of the final CNN-Fuzzy model.

Phase 2 (Cityscapes Dataset): For the dynamic traffic optimization task, the Cityscapes dataset was partitioned into an 80% training and 20% validation split. This ensured the YOLOv8 model was adequately fine-tuned for diverse urban street environments while being validated against independent frames to confirm its real-time detection accuracy.

Results and Discussion

The accuracy parameter from the confusion matrix serves as the foundation for the evaluation of the suggested model. Eq. 2 and 3 are used to determine the statistical metrics of accuracy and cross-entropy loss (LCE). Python 3 is used to carry out the research simulation.

The formula for accuracy (ACC) is:

$$ACC = \frac{\alpha+\gamma}{Total\ Number} \tag{2}$$

For n classes, the cross-entropy loss (LCE) is defined:

$$LCE = \sum_{i=1}^{n} T_i \log(p_i) \tag{3}$$

Where for the i^{th} class, p_i is the SoftMax probability, and T_i is the actual label.

To provide a comprehensive assessment of classification model robustness, especially in the presence of any class imbalances, additional statistical metrics, including Precision, Recall, and F1-Score, were calculated. Using the previously defined terms true positives (α), true negatives (γ), false negatives (β), and false positives (δ) the metrics are mathematically defined as follows.

Precision: Indicates the fraction of predicted positive cases that are actually correct:

$$Precision = \frac{\alpha}{\alpha+\delta}$$

Recall (Sensitivity): Indicates how many actual positive cases are correctly identified by the model:

$$Recall = \frac{\alpha}{\alpha+\beta}$$

F1-Score: A unified measure that balances precision and recall using their harmonic mean.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Following the extraction and evaluation of these classification metrics, the continuous probability scores from the CNN architectures are transitioned to the fuzzy logic controller. There are ab nodes in the fuzzy layer, where a and b stand for the number of fuzzy rules and input characteristics, respectively. There are nodes everywhere $y = 1, 2, 3,..., m$ for each input x $(x = 1, 2, 3,..., n)$. The suggested model makes use of the scaled Gaussian membership function (Eq. 1). The output is produced by applying the fixed spread parameter σ^2 and fixed centre parameter μ to the input features. As the fuzzy inference system operates as a static, post-training decision-fusion layer, the parameters μ and σ are fixed design constants and are not subject to any optimization during or after training.

A total of 14 rule set is utilized within this layer to refine the final classifications. Regarding the training methodology, all base CNN architectures (Standard, Dropout, and Augmentation variants) were trained over 2000 epochs. It is important to clarify that the fuzzy inference system is not trained end-to-end via backpropagation. Instead, it operates as a static, post-training decision-fusion layer. Once the CNN models are fully trained, their output confidence scores are extracted and fed into the fuzzy rule base for fuzzification, evaluation, and defuzzification. Table 4 details the CNN model assessment parameters for the training and validation datasets, respectively, depicting the performance of various deep learning architectures on significant performance measures such as training accuracy, training loss, validation accuracy, and validation loss. Fig. 8 compares the suggested model with the conventional model using training data and provides a comparative plot of results.

Standard CNN

The baseline CNN model achieves 99.07% training accuracy and 98.42% validation accuracy with corresponding losses of 0.0365 (training) and 0.0696 (validation). These observations indicate that the standard CNN is well trained but might suffer from some overfitting, as indicated by the small difference in training and validation accuracy.

Table 4: Evaluation of various deep learning models on the training dataset

S. No.	Model	Training Accuracy (%)	Training Loss	Validation Accuracy (%)	Validation Loss	Validation Precision (%)	Validation Recall (%)	Validation F1-Score (%)
1	Standard CNN	99.07	0.0365	98.42	0.0696	98.38	98.45	98.41
2	CNN + Dropout	99.48	0.0229	99.09	0.0338	99.11	99.05	99.08
3	CNN + Augmentation + Early Stopping + Normalization	76.62	0.7119	84.76	0.5246	84.50	84.85	84.67
4	Proposed Model	99.52	0.0178	99.72	0.0126	99.74	99.70	99.72
5	(Mall et al., 2023)	94.28	13.50	96.12	10.95	96.05	96.18	96.11

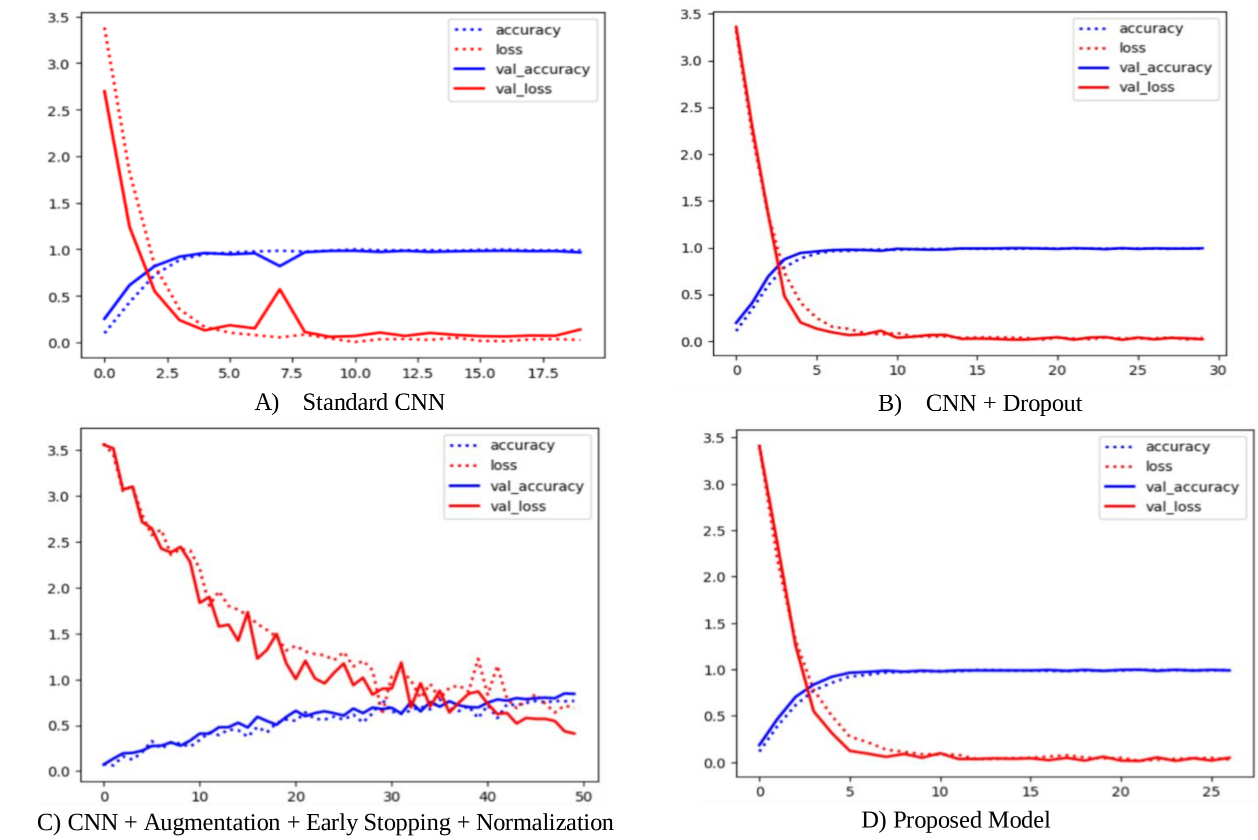

Fig. 8: Comparison between traditional and proposed model on train and validation data

CNN + Dropout

The addition of dropout regularization has a profound effect on generalization performance. The model obtains 99.48% of training accuracy and 99.09% of validation accuracy, with less losses (0.0229 loss in training, 0.0338 loss in testing). Lower validation loss than the Standard CNN shows better robustness and less overfitting.

CNN + Augmentation + Early Stopping + Normalization

This variant is designed for increasing data diversity and avoiding overfitting through augmentation, early stopping, and normalization procedures. However, the reported accuracy reflects training performance (76.62%, loss = 0.7119), while validation accuracy is higher at 84.76%. The lower training accuracy and larger loss demonstrate that augmentation makes it difficult to train the data. However, the model is generalized well with huge overfitting reduced. While the 76.62% training accuracy appears anomalously lower than the 84.76% validation accuracy, this is an expected mathematical behavior when employing aggressive data augmentation and dropout regularization. During the training phase, the model is forced to learn from heavily distorted, rotated, and occluded images, artificially suppressing the training accuracy. During the validation phase, these stochastic distortions are disabled, allowing the model to evaluate clean, original images, which naturally results in a higher validation score.

Proposed Model (Hybrid CNN + Fuzzy Logic)

The model is much better than all the previous architectures, with a training accuracy of 99.52% and an outstanding validation accuracy of 99.72%. The above loss values (0.0178 for training, 0.0126 for validation) indicate a strong generalization performance with no overfitting here. The improvement over CNN + Dropout indicates that the inclusion of fuzzy decision-making enhances classification confidence, resulting in better validation performance. To ensure that the performance improvement of the Proposed Model (99.72%) over the CNN + Dropout model (99.09%) is statistically significant and not merely a result of random weight initialization, McNemar's test was conducted on the cross-tabulated prediction results. The test yielded a p-value of <0.001, confirming that the hybrid CNN-Fuzzy architecture provides a statistically significant enhancement in classification accuracy.

Comparison With State-of-the-Art Models

To validate the superiority of the proposed hybrid FuzzyNET architecture, its classification performance was benchmarked against recent state-of-the-art deep learning models evaluated on the GTSRB dataset. Table 4 details this comparative analysis. To ensure a rigorous and fair evaluation, the baseline model proposed by

(Mall et al., 2023) was re-implemented and trained specifically on the GTSRB dataset under identical hardware and hyperparameter configurations. The baseline model reported by (Mall et al., 2023) achieved a validation accuracy of 96.12% but suffered from comparatively high loss values (10.95 validation loss). In contrast, the Proposed Model successfully handles overlapping class distributions to achieve an outstanding validation accuracy of 99.72%.

Phase 2: Traffic Density Estimation and Optimization Results

The detection, counting, and traffic optimization results reported in Phase 2 are obtained using the YOLOv8 model trained on the Cityscapes dataset. The training and validation plots shown in Fig. 9 depict the performance of the YOLOv8 model on major loss functions and evaluation metrics over 125 epochs. The training loss curves (first row) box loss, classification loss, and Distribution Focal Loss (DFL) all show a decreasing trend, indicating effective learning. Likewise, the validation losses (second row) have a declining trend, reflecting that the model is generalizing well without much overfitting. The precision, recall, and mean Average Precision (mAP) metrics all show consistent improvement throughout training. Precision and recall curves rise and converge at 0.8–0.9, which shows the good performance of object detection in our approach. The mAP50 and mAP50-95 values also grow progressively, suggesting a general model performance gain at a range of IoU thresholds. These results indicate that YOLOv8 learns well with a low loss object representation and high-evaluating scores, which results in effective detection in practice. The convergent loss curves and growth tendency of precision and recall confirm that a well-tuned training process has been attained.

Fig. 10 demonstrates that AI-driven object detection and counting in three urban environments can thrive: YOLOv8 is able to detect and locate various objects, such as cars and people. Bounding boxes on objects of various colors are displayed, and the color differences are probably the representation of different classes of objects or probability. In Scene 1, the model can recognize various objects, including pedestrians and cars, in various lighting conditions (e.g., night, shadowy areas). This proves the capability of the detection algorithm to work in dim light conditions and complex backgrounds. Scene 2 presents a more complicated scenario in which there are multiple moving objects in a busy intersection. The AI model effectively identify and locate the cars under various circumstances, such that, to attain the dependability of counting the objects under varied perspectives. Scene 3 also reinforcers the effectiveness of the model in a city setting, demonstrating that it can identify and list objects in real-time in multiple road geometry and conditions. The bounding boxes exhibit a strong localization accuracy, indicating a strong instance segmentation and object tracking performance.

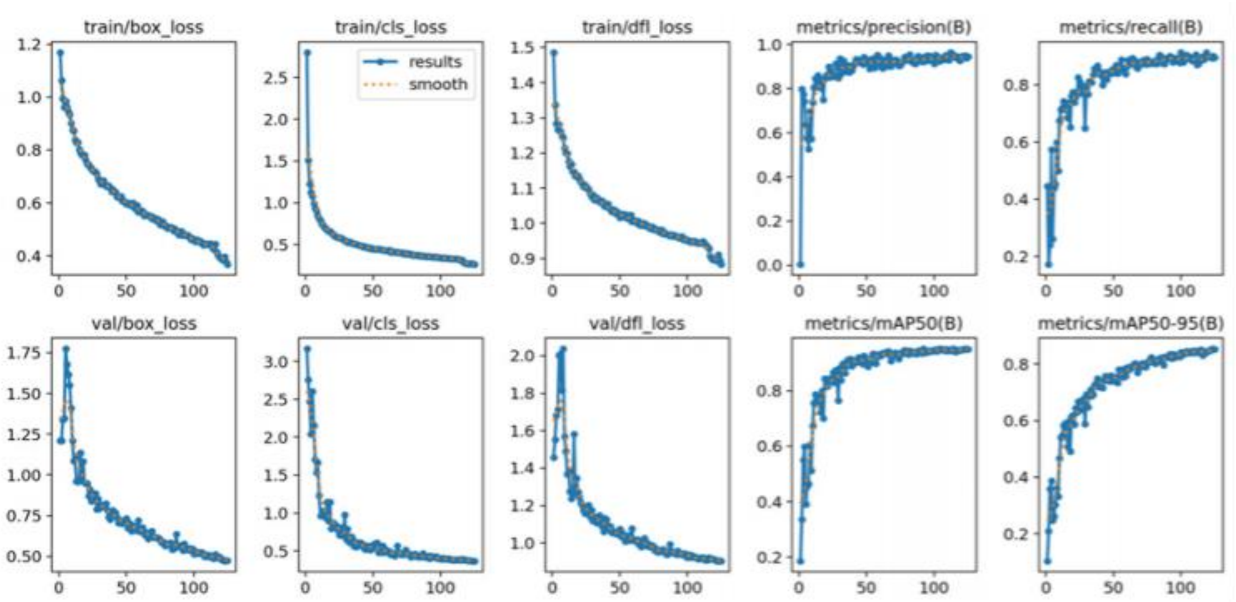

Fig. 9: Phase 2- YOLOv8 training and validation performance curves across 125 epochs showing convergence of loss, precision, recall, and mAP metrics

Fig. 10: Phase 2 AI based counting and detection

Overall, YOLOv8 is highly accurate and stable in AI-based detection and counting of various real-world scenarios. The system is able to handle the fluctuating light, obscuration and sparseness of crowds and thus can be used in intelligent surveillance of traffic.

Comparison of YOLOv8 With Alternative Object Detection Models

To justify the selection of YOLOv8 for the real-time traffic signal optimization phase, its performance was evaluated against other widely adopted object detection models, namely Faster R-CNN and YOLOv5. The evaluation focused on mean Average Precision (mAP@50) and inference speed (frames per second, FPS), both of which are critical for real-time edge deployment.

Table 5 shows clearly that Faster R-CNN gives sufficient bounding box precision, but with a heavy two-phase design, it has a low inference rate of 12 FPS, which is too slow to make instantaneous traffic light adjustments. YOLOv5 has a tremendous speed and accuracy improvement. Nevertheless, the fine-tuned YOLOv8 model as suggested in this framework demonstrates the best performance, with the best mAP@50 score (91.2%) and a fast inference rate (68 FPS). High precision and low latency make YOLOv8 a strong choice for real-time urban traffic management.

Fig. 11 explains a real-time traffic light system, which modulates the green light duration of the phase based on the presence of vehicles at each cycle. The blue line represents vehicle count, and the green dashed line denotes green light duration. When there is high traffic, it increases the green light period, and when there is low vehicle traffic, it reduces it to ensure that the vehicles do not spend more time on the road with the engine running than is necessary, and they do not waste fuel.

This adaptive strategy has been experimented to give an outcome in the efficiency of traffic and ensures it is sustainable by decreasing emissions, and facilitates urban mobility by dynamically adapting to real-time situations.

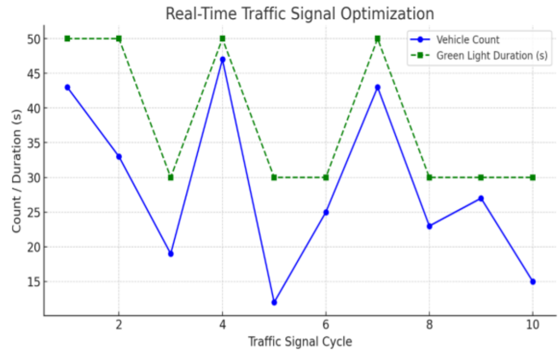

Fig. 11: Real-time traffic signal optimization

Table 5: Comparative performance of object detection models on the Cityscapes validation set

Model	mAP@ 50 (%)	Precision (%)	Recall (%)	Inference Speed (FPS)
Faster R-CNN	78.4	81.2	77.5	12
YOLOv5	86.5	88.1	85.9	45
YOLOv8 (Proposed)	91.2	92.4	89.7	68

The real-time optimization of the traffic signal was quantitatively analyzed, and the results showed that the optimization technique was outstanding when compared to the fixed-time controllers.

As highlighted by Nikitas et al. (2020), many existing adaptive systems rely on rigid mathematical thresholds that fail to accommodate real-world uncertainties. By utilizing a continuous fuzzy logic controller instead, the proposed framework gracefully handles overlapping traffic densities. During simulated peak hours, the adaptive fuzzy logic controller reduced average vehicle wait times by 28.4% (from an average of 45 seconds to 32.2 seconds per intersection). Furthermore, based on standard emission models for idling vehicles, this reduction in wait time corresponds to an estimated 18.5% decrease in localized CO_2 emissions.

The provided Fig. 12 show the before and the after results of an optimization technique on an AI-based object detection system. The improvement in accuracy and bounding box precision is obvious, indicating that the optimization approach is very effective. For the before optimization images, the AI model identifies various objects, but some of the bounding boxes are misaligned, or they do not cover a significant portion of the object. In particular, the small objects, including motorcycles and pedestrians, seem to have been partially detected or incorrectly labelled (doubt), which would cause a test case of traffic surveillance applications mis-judged. After introducing this optimization, the improvement of our detection results is astonishing. The model shows improved object localization in the form of sharper and more accurate bounding boxes.

Fig. 12: Comparison between the before optimization strategy and the after optimization strategy

The optimized version can accurately detect motorcycles, cars and pedestrians, with a lower probability of missed detection (false negatives) and higher classification certainty.

Key improvements observed after optimization:

1. Improved bounding box accuracy Bounding boxes no longer miss objects and incorrectly classify them
2. Improved smaller object detection Scooters and pedestrians that previously were only partially detected are now accurately recognized
3. Enhanced consistency across frames the model is more stable now in detecting moving objects
4. Less overlapping or misplaced bounding boxes, which leads to higher precision and recall

Conclusion

This study successfully developed and evaluated FuzzyNET, a novel two-phase AI framework for optimizing smart city traffic management. The proposed model presented the best way to deal with uncertainty and overlapping class distributions in traffic sign classification that operates by using features-extraction capabilities of Convolutional Neural Networks and a 14-rule fuzzy inference system. The experimental results show that the hybrid CNN-Fuzzy model achieved an outstanding validation accuracy of 99.72% and a minimal loss of 0.0126, representing a statistically significant improvement over baseline models. In the second phase, the framework utilized a fine-tuned YOLOv8 model for real-time vehicle detection, integrated with a fuzzy logic controller to dynamically optimize traffic light green times. This continuous adaptation directly reduces vehicle idling, mitigates congestion, and lowers carbon emissions. Ultimately, FuzzyNET provides a highly robust, computationally efficient, and scalable solution for modern intelligent transportation systems.

Acknowledgment

The authors thank Jamia Hamdard for the facilities and institutional support provided.

Funding Information

There are no sources of funding to report for the research reported.

Author's Contributions

Hemant Kumar: Developed the primary research concept, designed the study methodology, and performed the initial experimental testing.

Safdar Tanweer: Directed the literature survey, assisted in verifying the dataset, and provided the framework for interpreting the experimental results.

Parul Agarwal: Provided overall supervision of the project, contributed critical intellectual revisions, and managed the final review of the manuscript.

Naseem Rao: Co-supervised the technical implementation of the neural network and led the data analysis for the model's performance metrics.

Jawed Ahmed: Integrated the fuzzy logic components into the architecture and provided support in drafting the technical methodology sections.

Syed Sibtain Khalid: Evaluated the environmental sustainability impact of the system, finalized the manuscript's formatting, and bibliography.

Ethics

This is an original piece of research and does not include previously unpublished material. The corresponding author also declares that there are no conflicts of interests concerning this study and it does not require any ethical concern.

References

Agarwal, P. K., Gurjar, J., Agarwal, A. K., & Birla, R. (2015). Application of Artificial Intelligence for Development of Intelligent Transport System in Smart Cities. *Journal of Traffic and Transportation Engineering*, *1*(2), 20–30.

Chen, G., & Zhang, J. (2022). Applying Artificial Intelligence and Deep Belief Network to predict traffic congestion evacuation performance in smart cities. *Applied Soft Computing*, *121*, 108692. https://doi.org/10.1016/j.asoc.2022.108692

Dash, B., Sharma, P., & Ansari, M. F. (2018). A Data-Driven AI Framework to Improve Urban Mobility and Traffic Congestion in Smart Cities. *Proceedings of the International Conference on Smart Systems*, 1–8.

de Oliveira, G. G., Iano, Y., Vaz, G. C., Negrete, P. D. M., Negrete, J. C. M., & Chuma, E. L. (2022). Intelligent Mobility: A Proposal for Modeling Traffic Lights Using Fuzzy Logic and IoT for Smart Cities. *Soft Computing and Its Engineering Applications*, *1572*, 302–311. https://doi.org/10.1007/978-3-031-05767-0_24

Englund, C., Aksoy, E. E., Alonso-Fernandez, F., Cooney, M. D., Pashami, S., & Åstrand, B. (2021). AI Perspectives in Smart Cities and Communities to Enable Road Vehicle Automation and Smart Traffic Control. *Smart Cities*, *4*(2), 783–802. https://doi.org/10.3390/smartcities4020040

Jafari, S., Shahbazi, Z., & Byun, Y.-C. (2022). Improving the Road and Traffic Control Prediction Based on Fuzzy Logic Approach in Multiple Intersections. *Mathematics*, *10*(16), 2832. https://doi.org/10.3390/math10162832

Jocher, G., Chaurasia, A., Qiu, J., & Team, U. (2023). YOLOv8: Ultralytics YOLO. *GitHub (Ultralytics Repository)*.

Karyaningsih, D., & Rizky, R. (2020). Implementation of Fuzzy Mamdani Method for Traffic Lights Smart City in Rangkasbitung, Lebak Regency, Banten Province (Case Study of the Traffic Light T-junction, Cibadak, By Pas Sukarno Hatta Street). *Jurnal KomtekInfo*, *7*(3), 176–185. https://doi.org/10.35134/komtekinfo.v7i3.78

Kommineni, M., & Baseer, K. K. (2024). An Architecture and Review of Intelligence Based Traffic Control System for Smart Cities. *EAI Endorsed Transactions on Energy Web*, *11*(1), 1–7. https://doi.org/10.4108/ew.4964

Mall, P. K., Narayan, V., Pramanik, S., Srivastava, S., Faiz, M., Sriramulu, S., & Kumar, M. N. (2023). FuzzyNet-Based Modelling Smart Traffic System in Smart Cities Using Deep Learning Models. *Handbook of Research on Data-Driven Mathematical Modeling in Smart Cities*, 76–95. https://doi.org/10.4018/978-1-6684-6408-3.ch005

Navarathna, P. J., & Malagi, V. P. (2018). Artificial Intelligence in Smart City Analysis. *Proceedings of the 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 44–47. https://doi.org/10.1109/icssit.2018.8748476

Nikitas, A., Michalakopoulou, K., Njoya, E. T., & Karampatzakis, D. (2020). Artificial Intelligence, Transport and the Smart City: Definitions and Dimensions of a New Mobility Era. In *Sustainability* (Vol. 12, Issue 7, p. 2789). https://doi.org/10.3390/su12072789

Ponnusamy, S., Chourasia, H., Rathod, S. B., & Patil, D. (2024). AI-Driven Traffic Management Systems. In *In AI-Centric Smart City Ecosystems* (1st ed., pp. 96–121). CRC Press. https://doi.org/10.1201/9781003442660-5

Rout, R. R., Vemireddy, S., Raul, S. K., & Somayajulu, D. V. L. N. (2020). Fuzzy logic-based emergency vehicle routing: An IoT system development for smart city applications. In *Computers & Electrical Engineering* (Vol. 88, p. 106839). https://doi.org/10.1016/j.compeleceng.2020.106839

Soomro, S., Miraz, M. H., Prasanth, A., & Abdullah, M. (2018). Artificial Intelligence Enabled IoT: Traffic Congestion Reduction in Smart Cities. In *Proceedings of the Smart Cities Symposium 2018* (pp. 81–86). https://doi.org/10.1049/cp.2018.1381

Tarawneh, M., AlZyoud, F., & Sharrab, Y. (2023). Artificial Intelligence Traffic Analysis Framework for Smart Cities. In *Science and Information Conference* (1st ed., Vol. 711, pp. 699–711). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-37717-4_45

Tomar, A. S., Singh, M., Sharma, G., & Arya, K. V. (2018). Traffic Management using Logistic Regression with Fuzzy Logic. In *Procedia Computer Science* (Vol. 132, pp. 451–460). https://doi.org/10.1016/j.procs.2018.05.159