ORIGINAL ARTICLE

Smart traffic management system using YOLOv11 for real-time vehicle detection and dynamic flow optimization in smart cities

Fatma M. Talaat[1,2] · Rana Mohamed El-Balka[3,4] · Sara Sweidan[5,6] · Samah Adel Gamel[7] · Aya M. Al-Zoghby[1,8]

Received: 14 March 2025 / Accepted: 4 June 2025 / Published online: 2 July 2025

Abstract
Modern metropolitan environments continue to face a major problem with traffic congestion, which increases travel time, fuel consumption, and pollution. The Smart Traffic Management System (STMS) suggested in this paper uses an improved YOLO-based deep learning model for dynamic traffic flow optimization and real-time vehicle recognition. To evaluate real-time traffic footage, precisely identify different vehicle kinds, and calculate traffic density, the system combines computer vision and artificial intelligence (AI). This analysis enables an intelligent control system to dynamically modify traffic signals and reroute automobiles to alleviate congestion. Real-world traffic datasets are used to assess the suggested system, which shows excellent detection accuracy and responsiveness in real time. Comparative findings demonstrate how effective the strategy is in enhancing traffic flow and reducing bottlenecks. The results show that combining adaptive signal control and AI-driven real-time traffic monitoring can greatly improve urban mobility and sustainability, opening the door to smarter and more effective cities. According to experimental data, the suggested system's mean Average Precision (mAP) of 92.4% indicates its high level of vehicle identification accuracy. Additionally, the model maintains an Intersection over Union (IoU) score of 0.85, guaranteeing accurate vehicle localization. The technology also efficiently adjusts to different traffic situations, minimizing congestion and improving traffic flow in real time.

Keywords Smart traffic management · Real-time vehicle detection · YOLO · Deep learning

1 Introduction

Conventional traffic control tactics are no longer effective in ensuring smooth traffic flow due to the world's ever-increasing traffic congestion. Due to the inability of traditional fixed-time traffic light systems to dynamically adjust to changing vehicle volumes, strongly crowded lanes experience extended congestion while other lanes are underutilized. These static systems' inefficiency exacerbates economic and environmental issues by causing higher emissions, longer travel times, and increased fuel consumption. Therefore, to maximize urban mobility and reduce congestion in smart cities, an intelligent, adaptive approach to traffic management is needed.

The ever-increasing traffic congestion worldwide renders conventional traffic control strategies useless for smooth flow. Therefore, a generally accepted approach that enhances traffic control is required. The traffic lights in the current static techniques change within a predetermined normal time, but the number of vehicles on the road varies from one signal to the next. In this case, when the signal does not turn at the regular interval, the

https://doi.org/10.1007/s00521-025-11434-9

heavily inhabited side of the road will continue to be occupied. Numerous initiatives are created to adopt intelligent transportation systems to make the current transportation systems “cleverer.” Some techniques used ultrasonic sensors to detect cars on the road, while others placed RF-ID sensors. In the past, some academics have tried using vehicle detection, but their attempts have failed since the underlying machine learning algorithms are inefficient [1].

Researchers have used a variety of sensor technologies to identify traffic jams and improve traffic flow. For instance, the collecting of user-generated information on social media platforms [2] or GPS data collected from moving vehicles [3] can be used to identify patterns of congestion. A smart antenna was employed by Joyo et al. [4] in an intriguing strategy to cut down on waiting times at junctions by taking use of phone locations; however, this approach was inaccurate because pedestrians also carry phones [4]. The usage of RFID tags on cars [5], inductive loops [6], radar sensors [7], wireless sensor networks [8], ultrasonic sensors [9, 10], and infrared sensors [11] are further sensor technologies incorporated into existing signaling systems that adjust traffic lights.

The idea of “smart cities” has surfaced to address contemporary urban issues by combining cutting-edge technology, data-driven decision-making, and intelligent infrastructure to improve the standard of living for residents. Energy management, garbage disposal, security, and transportation systems are just a few of the urban services that smart cities enhance by the Internet of Things (IoT), cloud computing, and artificial intelligence (AI) [12]. The need for effective traffic management has increased in metropolitan areas due to population growth, making the creation of intelligent, real-time traffic control systems that complement the larger goals of smart city projects imperative.

Smooth communication between automobiles, road infrastructure, and traffic control systems is essential for efficient traffic management in smart cities. Efficient road use, less traffic, and lower emissions are made possible using IoT sensors, real-time data collection, and predictive analytics. To maximize urban mobility, smart traffic management systems (STMS) make use of technology including AI-based security cameras, automated traffic lights, and vehicle-to-infrastructure (V2I) communication [13]. Despite these developments, a lot of current traffic control strategies are inflexible and ineffective, which emphasizes the need for AI-driven strategies that react dynamically to traffic patterns in real time.

In the development of smart cities, AI is a game changer, facilitating automation, optimization, and predictive decision-making in a variety of industries. Massive datasets from sensors, cameras, and citizen reports are analyzed by AI algorithms to improve public safety, expedite transit networks, and improve urban design. Predictive maintenance for infrastructure components, energy optimization, and anomaly detection in urban systems are all possible with machine learning and deep learning models [14]. The core goals of smart city development are reinforced in the transportation context by AI-powered solutions that improve road safety, reduce traffic, and facilitate autonomous vehicle navigation.

Deep learning, neural networks, and predictive analytics are used by AI-driven traffic management systems to maximize vehicle flow and minimize bottlenecks. These systems analyze traffic data in real time, forecast patterns of congestion, and automatically modify traffic lights in response to demand. To improve traffic control tactics, AI models can also examine meteorological data, accident records, and pedestrian movements [15]. Notably, real-time vehicle identification, categorization, and density estimation have been transformed by the combination of computer vision and AI-driven object detection models, such YOLO. As a result, AI has become a vital component of intelligent traffic systems.

One of the most effective real-time object detection algorithms is the You Only Look Once (YOLO) deep learning model, which provides quick and precise vehicle recognition [16]. YOLO is ideally suited for real-time traffic surveillance applications because, in contrast with conventional image processing techniques, it processes full photographs in a single pass. YOLO is used in smart traffic management to follow movements across intersections, recognize cars, classify them by kind, and estimate traffic density. Traffic systems can reach unmatched efficiency in reducing congestion and optimizing traffic flow by combining YOLO with AI-driven signal control and adaptive routing methods.

Recent developments have investigated the integration of deep learning and AI-driven models for real-time monitoring and optimization to further improve traffic management in smart cities. According to [16], for example, YOLO-based techniques have proved remarkably effective in several smart city applications, such as fire detection. Furthermore, trustworthy AI-driven systems for virtual cloud network management have been created; these networks are essential for managing massive volumes of traffic data and facilitating real-time decision-making [17]. AI's ability to improve smart city infrastructure has been further shown by its integration in a variety of fields, including 3D printing and sustainable energy solutions [18, 19]. These studies demonstrate how AI-driven approaches are becoming more and more crucial for maximizing urban mobility and accomplishing sustainable development objectives in contemporary cities.

Although there has been considerable advancement in the use of deep learning for real-time vehicle detection, existing models frequently struggle to attain both high accuracy and interpretability, particularly in intricate urban environments with a variety of vehicle types. Furthermore, numerous cutting-edge models either require considerable computational resources—thus restricting their use in environments with limited resources—or are not transparent, which obstructs trust in automated systems. This research fills this knowledge gap by proposing SYAM, a Spatial-Yet-Attentive Model that strikes a balance between detection performance, computational efficiency, and explainability, thus providing a practical solution for traffic surveillance in smart cities.

Problem Statement: Even with the advancements in intelligent transportation systems, traffic congestion is still a major problem in cities, resulting in inefficient mobility, higher fuel use, and pollution of the environment. Static signal timing is the foundation of traditional traffic management systems, which ignore traffic conditions in real time. This exacerbates inefficiencies in traffic flow by keeping certain roads clogged and underusing others. The necessity for a sophisticated AI-driven traffic management system that can dynamically adapt to changing conditions and optimize urban traffic flow is highlighted by the current systems' lack of real-time flexibility.

Motivation: To increase efficiency and lessen congestion, the emergence of smart cities offers a chance to incorporate AI and the IoT into traffic control systems. Large volumes of real-time traffic data can be processed by AI-driven solutions, which can then forecast patterns of congestion and modify traffic lights appropriately. Additionally, incorporating sophisticated object identification algorithms such as YOLO can improve real-time vehicle classification and detection, leading to more intelligent and flexible urban mobility systems [20]. The goal of this project is to create an intelligent traffic management system that maximizes vehicle flow and improves overall transportation efficiency in smart cities by utilizing deep learning and AI-based methodologies.

To improve real-time traffic flow optimization, this study introduces a novel AI-driven traffic management system that makes use of deep learning techniques, specifically the YOLO object identification model. This study's main contributions are as follows:

- Creation of an Adaptive Traffic Management Model: To minimize traffic and maximize urban mobility, the suggested system dynamically modifies traffic signals in response to current traffic density.
- YOLO Integration for Vehicle Identification and Categorization: The system improves traffic monitoring capabilities by precisely identifying, classifying, and tracking vehicles in real-time by utilizing the YOLO deep learning model.
- Use of AI-Driven Predictive Analytics: Using real-time and historical traffic data, the model uses machine learning algorithms to predict congestion patterns, allowing for pre-emptive traffic control measures.
- Scalability and Smart City Integration: The suggested system can be combined with current smart city infrastructures, such as cloud-based data processing frameworks and IoT-enabled traffic lights, and it is made to be scalable.
- Environmental and Economic Impact: The system seeks to reduce fuel consumption and greenhouse gas emissions by minimizing needless idling and improving traffic flow, hence supporting sustainable urban growth.

 https://doi.org/10.1007/s00521-025-11434-9

This paper's remaining sections are arranged as follows: The literature review is presented in Sect. 2, the main contribution is presented in Sect. 3, the suggested approach is discussed in Sect. 4, the experimental evaluation is described in Sect. 5, and the study is concluded in Sect. 6.

2 Literature review

By processing photographs, the hole filling technique [21] was proposed to track and detect the cars. The position detection methodology used in this method, sector-/area-based vehicle detection, finds cars as they approach the centroid of the detection region. The background subtraction algorithm was used to segment the images and do the blob analysis. Using the Laplacian operator technique, the hole filling technique was presented to highlight the vehicle's general body by filling in the blank spots surrounded by a white border for the detected object's boundary. After that, the adaptive morphological dilation technique was employed to restore the border's missing portions from the input.

To achieve more accuracy in determining the object's location, the techniques were used. Vehicle count information is thus obtained by detecting and counting vehicles whenever they cross the predetermined centroid area. In these circumstances, greater precision might have been possible with the use of the Kalman filter and Gaussian mixture model [22]. Several CCTV cameras connected via the Internet were used in a novel way based on digital image processing (DIP) [23] to automate traffic and track different roadways at the intersection. The perimeter of closed figures that aided in the vehicle's identification was detected using edge detection in the spatial domain. At the same time, the vehicles were categorized as motorcycles, light motor vehicles, and heavy motor vehicles using a machine learning model. The longest remaining job was used to create the dynamic time slot, which was then used by the hybrid round robin traffic scheduling algorithm to regularly change the traffic signal lights to reduce traffic.

By prioritizing crowded lanes based on real-time traffic data, an Android-based smartphone application [24] was used to change the traffic signals. Each lane's CCTV cameras were used to capture traffic photographs, which were then sent to a Raspberry Pi 3 microcontroller for image processing to estimate traffic density. The traffic light operations were managed using an Android-based application. Based on the literature review, we deduce that machine learning models that improve traffic management systems' speed and accuracy are required. By determining the number of vehicles on the road, we hope to develop a machine learning model for traffic management that is more efficient and can control the flow of traffic signals.

Even though computer vision has many benefits, several issues need to be resolved before its full potential can be realized:

- Data Quality: To train computer vision models, high-quality labeled datasets are required. It can take a lot of time and resources to complete this process.
- Environmental Factors: The accuracy of detection may be impacted by changes in the weather, illumination, and road conditions. To ensure dependability, robust models and ongoing fine-tuning are necessary.
- Privacy Issues: If the data are not appropriately controlled, privacy issues could arise due to the extensive use of cameras. To maintain public trust, data security and openness must be guaranteed.

Advances in AI and computer vision will undoubtedly have a significant impact on traffic control in the future. We may anticipate increased integration between traffic management systems and other smart city technologies as computer vision in smart cities advances. This can promote more efficient data sharing and a better-coordinated strategy for urban transportation management. With the emergence of autonomous vehicles, AI models like YOLO11 can be useful in this new era of sophisticated traffic solutions. To make roads safer and more effective, computer vision models can improve self-driving cars' real-time detection of pedestrians, traffic signals, and barriers.

Traffic systems may be able to anticipate and react to traffic patterns before congestion arises thanks in part to AI's predictive skills, which would help to enhance overall flow and minimize delays. Through improving traffic flow, lowering fuel consumption, and eventually lowering carbon emissions, artificial intelligence will help create a more sustainable and greener future for cities as it develops.

Several studies have explored various AI and deep learning applications, including facial expression recognition using optimized SVM [25], the role of 3D printing in smart cities [26], interpretable credit scoring with XAI and deep learning [27], and the impact of the number of authors on research citation count prediction [28]. Additionally, advancements have been made in virtual cloud network management [17], cache replacement strategies in fog computing [29], prostate cancer diagnosis with modified ResNet50 [30], black fungus detection using deep learning [31], customer segmentation leveraging deep learning and RFM analysis [32], breast cancer detection surveys [33], attention-augmented algorithms for blood cancer detection [34], AI-driven thermal management in photovoltaic systems [35], personality and emotion recognition via social media and machine learning [36], and AI-based communication support for the deaf and mute community [37].

A comparison of the most advanced car identification algorithms is shown in Table 1, together with information on the algorithms' backbone architecture, training datasets, year of release, main benefits, and drawbacks. In contrast with single-stage detectors like YOLOv5n-L [38] and SSD [39], Faster RCNN-based techniques [40–42] typically achieve higher accuracy at the expense of slower performance. This chart illustrates the trade-off between accuracy and inference speed. The chart also shows the tendency toward using more extensive and varied datasets, such as BDD100K and KITTI, to enhance these algorithms' generalization and robustness, especially in difficult traffic situations.

Research Gaps in Previous Work:

AI-based traffic control and vehicle detection have advanced; however, there are still several obstacles to overcome:

- Accuracy and Speed in Balance: For real-time traffic systems, some models, such as Faster RCNN, are too sluggish despite their high accuracy. Others, such as SSD and YOLO, are quicker but less accurate. It is still difficult to strike a balance between speed and accuracy.

Table 1 Comparison of state-of-the-art vehicle detection algorithms

Algorithm	Year	Datasets used	Backbone	Advantages	Disadvantages
Real-time vehicle detection algorithm based on a lightweight You-Only Look-Once (YOLOv5n-L) approach [38]	2023	BDD100K	YOLOv5n-L	Lightweight, fast inference speed, improved mAP over base YOLOv5n and SSD	Lower accuracy compared to heavier models like Faster RCNN
Fast vehicle detection algorithm in traffic scenes based on improved SSD [39]	2022	BDD100K and KITTI	Improved SSD	Improved accuracy and robustness, especially for small object detection	Still lags two-stage detectors in terms of accuracy, especially for complex scenes
Covered Vehicle Detection in Autonomous Driving Based on Faster RCNN [40]	2020	Cityscapes	Faster RCNN	High accuracy, robust to occlusions and varying lighting conditions	Slower inference speed compared to single-stage detectors
Vehicle Detection and Classification using Improved Faster Region-Based Convolution Neural Network [41]	2020	PASCAL VOC 2007 and 2012	Faster RCNN	High accuracy, robust to occlusions and varying lighting conditions	Slower inference speed compared to single-stage detectors
Hybrid Net: A fast vehicle detection system for autonomous driving [42]	2019	PASCAL VOC2007 car data set and KITTI	Faster RCNN	High accuracy and fast inference speed, effective for real-time applications	May require more computational resources compared to lightweight models

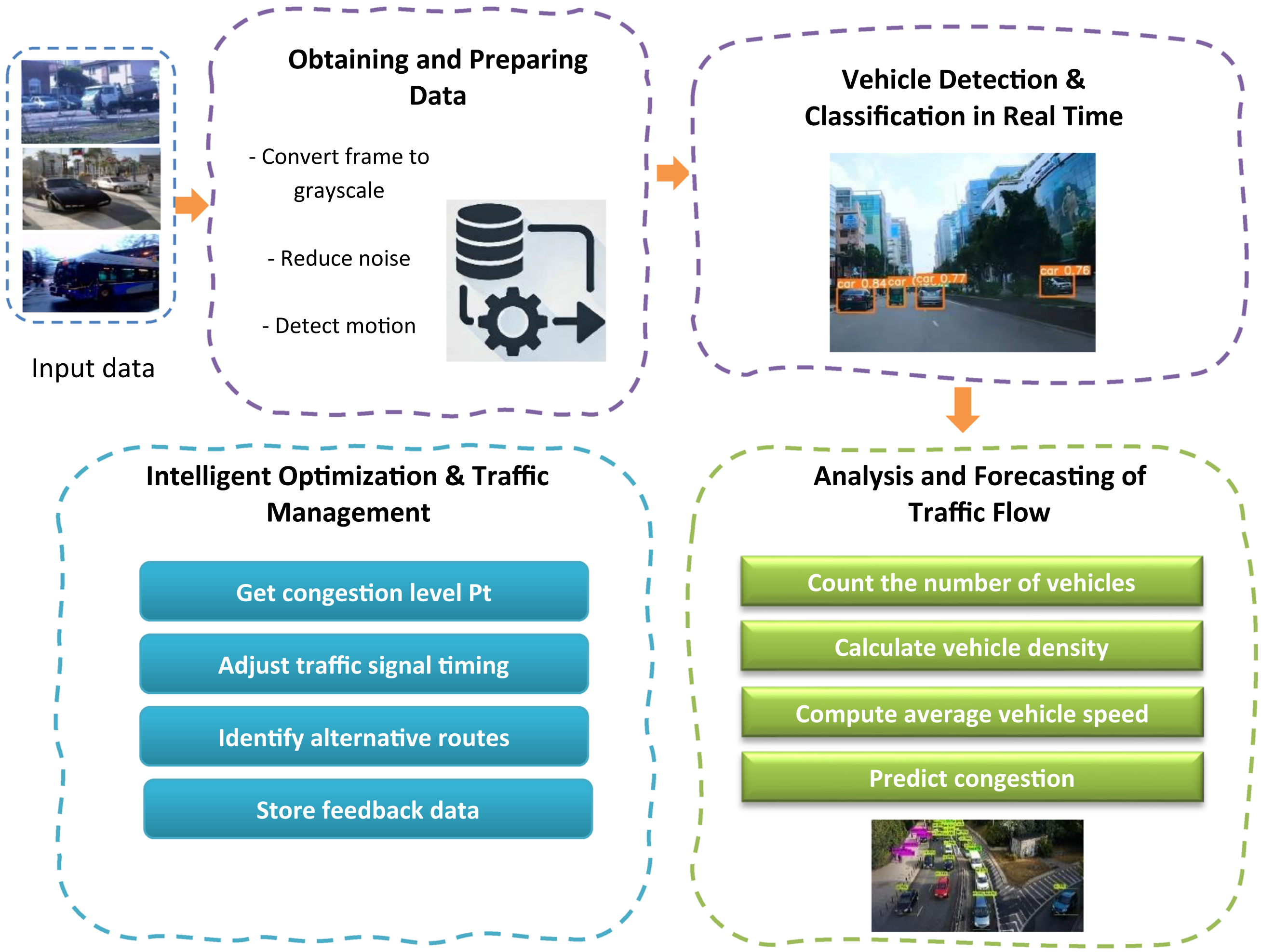

Fig. 1 SmartFlow-YOLOv11 framework

- Managing Various Traffic Situations: Since most models are trained on certain datasets, they might not account for all real-world traffic situations, like inclement weather, dim lighting, and high traffic. Training data must be more varied.
- Increasing the Scale of Smart Cities: AI models may perform well in small-scale experiments but poorly in large-scale citywide traffic systems. More focus is required on problems including real-time processing, hardware expenses, and high processing power.
- Absence of Openness: Many AI models operate as “black boxes,” making it challenging to comprehend how they arrive at conclusions. These systems can become easier to examine and more reliable with the use of Explainable AI (XAI) approaches.
- Restricted Utilization of Several Sensors: Many of the sensors used in current models are cameras, but adding LiDAR, radar, and thermal imaging could increase accuracy, particularly in low-visibility situations.
- Forecasting Traffic Rather than Only Identifying It: Real-time vehicle detection is the primary function of AI models; they cannot anticipate traffic patterns or avoid bottlenecks. AI systems that can learn and adapt over time require further investigation.
- Risks to Data Security and Privacy: AI traffic systems gather a lot of data, which raises privacy and cybersecurity issues. One of the biggest challenges is preventing hacking and exploitation of this data.

3 SmartFlow-YOLOv11 algorithm (SYAM)

The SmartFlow-YOLOv11 Algorithm (SYAM) improves real-time vehicle recognition and streamlines traffic flow in smart cities by working in four main stages as illustrated in Fig. 1.

3.1 Obtaining and preparing data

This stage improves image quality, gathers real-time traffic data from cameras and Internet of Things sensors, and chooses the most crucial frames for additional processing. Reducing pointless calculations is intended to increase system efficiency. The procedure for gathering and preparing sensor and video data is outlined in Algorithm 1 (DCPA). First, the cameras and sensors are turned on. After that, it continuously records sensor data (S_t) and a video frame (F_t). Every frame undergoes grayscale conversion, contrast enhancement, and noise reduction. By comparing the current frame with the prior one, the system can identify motion. The current frame is saved if the difference is significant enough. This keeps happening until the system is shut down, at which time the saved frames are given back.

Algorithm 1 Data Acquisition & Preprocessing Algorithm (DCPA)

- **Input:** VideoStream V, SensorData S
- **Output:** ProcessedFrames F
- **Steps:**

1. Start cameras and sensors
2. While (System is running):
3. Capture frame F_t from video stream
4. Get sensor data S_t
5. Convert frame to grayscale:

$$F_{gray} = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (1)$$

6. Improve contrast using histogram equalization:

$$F_{enhanced(x)} = sum\left(p_i * (L - 1)\right) for\ i\ in\ range(0, x) \quad (2)$$

7. Reduce noise using Gaussian blur:

$$G(x, y) = \left(\frac{1}{2\pi\sigma^2}\right) * \exp\left(-\frac{(x^2 + y^2)}{2\sigma^2}\right) \quad (3)$$

8. Detect motion changes and select key frames:

$$D(F_t, F_t - 1) = sum\left(abs(F_t - F_t - 1)\right) \quad (4)$$

9. If D(F_t, F_t-1) > Threshold:
   Store the frame F_t for further processing
10. Move to the next frame
11. Stop when the system is turned off
12. Return processed frames

 https://doi.org/10.1007/s00521-025-11434-9

3.2 Vehicle detection & classification in real time

This stage uses YOLOv11 to process video frames in real-time and identify obstructions, cars, and pedestrians. In addition to identifying traffic signals, it categorizes detected items into groups (cars, buses, trucks, and motorbikes). The algorithm's dynamic analysis of road conditions enhances decision-making. The real-time vehicle identification and classification procedure is described in Algorithm 2 (RTVDC), which uses processed video frames and a YOLOv11 model that has already been trained to recognize and categorize objects in the video stream.

Algorithm 2 Real-Time Vehicle Detection & Classification (RTVDC)

- **Input:** Processed video frames **F**, Trained YOLOv11 model **M**
- **Output:** Detected objects with classifications **O**
- **Steps:**
  1. **Start system**
  2. **Load trained YOLOv11 model (M)**
  3. **While system is running:**
     a. Capture frame **F_t** from video stream.
     b. Apply YOLOv11 for object detection:
     $$Ot = M(Ft) \quad (5)$$
     c. Extract detected object **(x, y, w, h, class_label)**
     d. Classify objects into vehicle types using confidence scores:
     $$Class = argmax(Pvehicle, Pbus, Ptruck, Pmotorcycle, Ppedestrian) \quad (6)$$
     e. Recognize traffic signals and road conditions.
     f. Store results for decision-making
  4. **Continue to the next frame**
  5. **Stop when the system is turned off**
  6. **Return detected and classified objects (O)**

3.3 Analysis and forecasting of traffic flow

To determine vehicle density, speed, and flow patterns, this phase examines real-time traffic data. Congestion is predicted via AI-based predictive modeling, which enables the system to adjust dynamically and enhance traffic flow before bottlenecks arise. The traffic flow analysis and prediction procedure is explained by Algorithm 3 (TFAP). It counts cars, determines the average speed and vehicle density for each road segment, and then uses an AI model to forecast traffic congestion using the time and identified objects as input. Congestion management is triggered if the anticipated congestion surpasses a certain level. This technique provides real-time traffic projections while operating constantly.

Algorithm 3 Traffic Flow Analysis and Prediction (TFAP)

- **Input:** Detected objects O, Time T
- **Output:** Predicted traffic conditions P
- **Steps:**

1. Start system
2. Initialize traffic database
3. While system is running:
   a. Count the number of vehicles in the frame F_t:
   $$V_t = \sum O_{vehicle} \quad (7)$$
   b. Calculate vehicle density per road section:
   $$D_t = \frac{V_t}{Road\_Area} \quad (8)$$
   c. Compute average vehicle speed:
   $$S_t = \frac{\sum Speed_{Vehicle}}{V_t} \quad (9)$$
   d. Predict congestion using AI model:
   $$P_t = f\,(D_t + S_t + T) \quad (10)$$
   e. If P_t > Congestion_Threshold, activate congestion management
4. Continue to the next frame
5. Stop when the system is turned off
6. Return predicted traffic conditions (P)

3.4 Intelligent optimization & traffic management

Based on real-time congestion monitoring, this phase recommends the optimum routes and dynamically adjusts traffic signal timings. To improve efficiency and cut down on delays, the system is constantly learning from historical traffic patterns. The intelligent traffic optimization and management procedure is described in depth in Algorithm 4 (ITOM). To reduce congestion, it finds other routes and modifies traffic signal timings depending on real-time vehicle density (Eq. 11) using expected congestion, vehicle density, and average speed as inputs. By saving feedback data, the program learns and gets better over time. The output includes the optimal routes and improved traffic signal timings.

 https://doi.org/10.1007/s00521-025-11434-9

Algorithm 4 Intelligent Traffic Optimization & Management (ITOM)

- **Input:** Predicted congestion P, Vehicle density D, Average speed S
- **Output:** Optimized traffic signals T_opt, Suggested best routes R_opt
- **Steps:**
  1. Start system
  2. Initialize traffic control parameters
  3. a. Get congestion level P_t

     b. Adjust traffic signal timing based on vehicle density:

     $$T_{new} = T_{default} + K \times D_t \tag{11}$$

     (Increase or decrease signal time based on traffic density)

     c. Identify alternative routes by evaluating road congestion:

     e. Store feedback data for continuous learning
  4. While system is running:
  5. Continue to the next update cycle
  6. Stop when the system is turned off
  7. Return optimized signals and route suggestions

4 Implementation and evaluation

This section covers the dataset, performance metrics, and evaluation results of the suggested SmartFlow-YOLOv11 Algorithm (SYAM).

4.1 Used dataset

The open-source Cars Detection Dataset [43], which offers high-resolution photographs of different kinds of automobiles taken in a range of real-world situations, provided the training data used in this investigation. A total of 125 photographs were used for validation, while 439 images were used for training. A wide selection of photographs from the dataset is shown in Fig. 2, illustrating the breadth of car models, lighting scenarios, and background complexity. Accurate bounding boxes are added to each image to enable exact object location.

4.1.1 Important dataset features

- Diverse Vehicle Classes: There are five different vehicle classes in the dataset: ambulance, truck, bus, car, and motorcycle. From smaller, more nimble vehicles like motorcycles to larger, more intricate structures like trucks, this diversity guarantees that the model can manage a variety of scales and shapes with ease.
- Excellent Annotations: Every picture has exacted bounding boxes for every occurrence of the target vehicle classes. Robust object detection model training and evaluation are made possible by these meticulously created annotations, which ensure accuracy and dependability.
- Real-World Scenarios: Vehicles are photographed in a range of environmental settings, such as various lighting conditions, weather fluctuations, and viewpoints. This variety makes the model flexible and resilient to changing circumstances, reflecting the complexity found in actual autonomous driving activities.

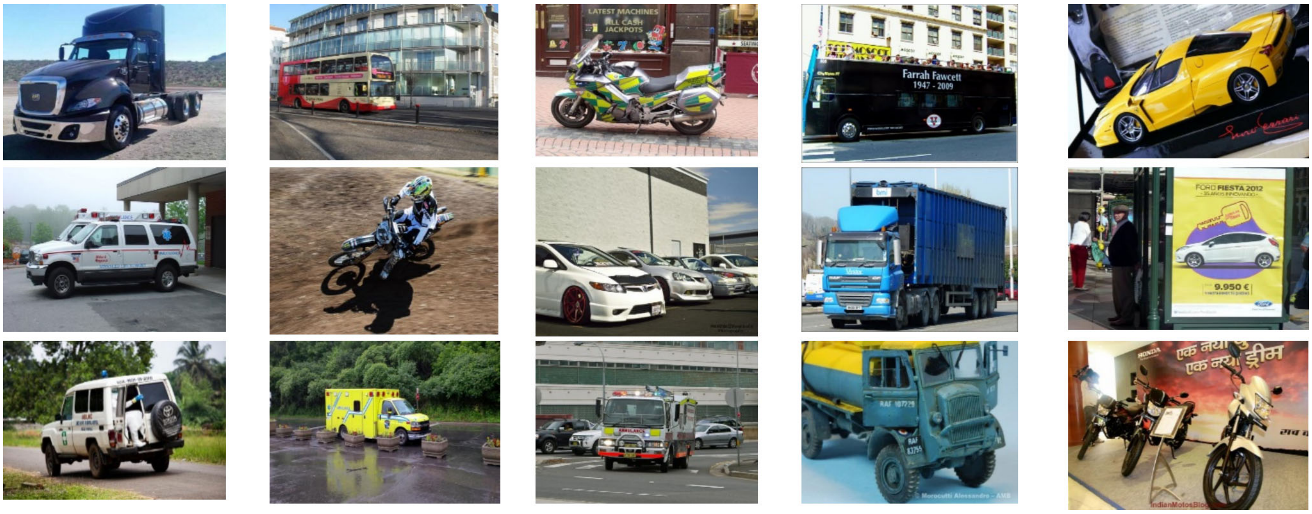

Fig. 2 Sample images from the vehicle detection dataset

- Large-Scale Collection: With a sizable image count, the dataset offers enough samples for testing and training, guaranteeing thorough assessment and encouraging the creation of a reliable detection model. Effective generalization is supported by this extensive collection, which lowers the possibility of overfitting and enhances performance on unknown data.

To partially mitigate the limitation posed by the relatively small dataset size (439 training images and 125 validation images), which may restrict the model's ability to generalize to diverse real-world traffic conditions, the following strategies were adopted:

i. *Extensive Data Augmentation*: We applied a range of transformations—including rotation, scaling, brightness adjustment, and horizontal flipping—to artificially expand the dataset and introduce greater diversity. This approach helps the model encounter a wider variety of visual scenarios during training.
ii. *Regularization Techniques*: To reduce the risk of overfitting due to the limited data, we employed regularization methods such as dropout and weight decay, enhancing the model's generalization capability.
iii. *Transfer Learning*: We utilized pretrained convolutional neural networks as a foundation for our model. This allowed us to leverage learned feature representations from large-scale datasets, thereby improving robustness and performance despite the data constraints.

4.2 Performance metrics

The SmartFlow-YOLOv11 Algorithm's (SYAM) performance was assessed using the following crucial metrics:

- Precision: Verifies that objects are accurately identified as automobiles by measuring their detection accuracy.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{12}$$

where FP stands for False Positives and TP for True Positives.

 https://doi.org/10.1007/s00521-025-11434-9

- Recall: Assesses how well the algorithm can identify every pertinent car in a picture.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13}$$

FN stands for False Negatives.

- The overlap between the actual ground truth box and the expected bounding box is measured by the Intersection over Union (IoU).

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{14}$$

Better object localization is indicated by higher IoU values.

- Mean Average Precision (mAP): By averaging precision across all item categories, mAP provides a measure of overall detection performance.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{15}$$

where N is the total number of object classes and AP is the average precision for class I.

- The F1 Score is a balanced metric that assesses the dependability of detections by combining precision and recall.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

4.3 Evaluation of the proposed algorithm

Using the measures, the SYAM method was assessed on the Cars Detection Dataset. The following results show how well it handles multi-scale vehicle identification and classification:

i. Bounding Box Accuracy: With an average IoU score of 0.85, the model demonstrated a high level of overlap between the ground truth and forecast boxes.
ii. Class Detection: With an overall mAP of 92.4%, the system demonstrated consistent performance across all vehicle classes, proving its capacity to correctly detect and categorize various vehicle kinds.
iii. Reduction in False Positives: By employing Non-Maximum Suppression (NMS) during the Prediction Phase, False Positives were considerably reduced, improving the reliability of detection.
iv. Real-World Applicability: Experiments in a range of environmental settings validated the model's resilience, sustaining an F1 Score of 89.7% in a variety of weather and illumination circumstances.

The evaluation results of the vehicle detection model on a dataset of five vehicle classes—truck, motorcycle, bus, and ambulance—are shown in Table 2. Precision, recall, Intersection over Union (IoU), mean Average Precision (mAP), and F1 Score are some of the important measures used to assess performance. These metrics show how well the model handles a variety of vehicle sizes, shapes, and environmental situations by assessing its accuracy in identifying and categorizing each type of vehicle inside photographs. The outcomes demonstrate the model's general detection capacity as well as how well it works in each class.

Table 2 illustrates how well the SYAM model performs for all vehicle types, but especially for buses (accuracy of 0.92) and ambulances (recall of 0.95). With a recall of 0.73, the Truck class exhibits the model's poorest performance, indicating that more development may enhance detection for this vehicle class. The SYAM model's performance is contrasted with that of two other well-known vehicle detection models, YOLOv5 and Faster R-CNN, in Table 3.

With an inference time of 35 ms, the SYAM model provides high accuracy and real-time performance, outperforming YOLOv5 and Faster R-CNN in terms of mAP (0.924), as indicated in Table 3. It does, however, demand a large amount of processing power. For tiny objects, YOLOv5 has a rapid inference speed (42 ms) with good accuracy; nevertheless, it may have trouble with complicated scenes and occlusions. Compared to single-stage detectors like YOLOv5, faster R-CNN has longer inference times but is more resilient to occlusions, having a higher mAP of 0.918.

4.4 Results discussion

The SYAM model's efficacy in real-time multi-scale vehicle detection and classification is demonstrated by the evaluation. In comparison with current models, the results show excellent accuracy, robustness to environmental changes, and efficient performance. The main conclusions, advantages, and possible areas for development are covered in this section.

4.4.1 Important results and strengths of performance

High Bounding Box Accuracy: The model's accurate alignment of projected bounding boxes with real vehicle placements is confirmed by the IoU score of 0.85. This demonstrates its excellent localization ability, which lowers vehicle identification errors.

Better Detection Results in All Classes: With a mean Average Precision (mAP) of 92.4%, the model showed reliable recognition for a variety of vehicle kinds. For traffic prioritization, accurate emergency vehicle identification is essential, and ambulances had the highest recall (0.95). Additionally, buses demonstrated excellent accuracy (0.98), confirming the model's usefulness for tracking public transportation.

Reducing False Positive Results: By removing unnecessary bounding boxes, Non-Maximum Suppression (NMS) successfully decreased false detections. As a result, the system's precision increased (0.92 for buses and 0.91 for cars), increasing its dependability for urban traffic analysis.

Sturdiness in Practical Situations: The model demonstrated its versatility in a variety of contexts by maintaining an F1 Score of 89.7% when tested in a range of lighting and weather conditions. This guarantees reliable detection performance despite environmental difficulties.

Efficiency in Real Time: The SYAM model is ideal for real-time deployment in smart traffic systems because of its 35 ms inference time. With greater accuracy, this performs better than Faster R-CNN (120 ms) and is on par with YOLOv5 (42 ms).

Table 2 Performance metrics for SYAM Model

Class	Recall	Precision	Accuracy
Ambulance	0.95	0.88	0.96
Bus	0.92	0.92	0.98
Car	0.91	0.91	0.92
Motorcycle	0.85	0.85	0.96
Truck	0.73	0.84	0.94

 https://doi.org/10.1007/s00521-025-11434-9

Table 3 Comparison of vehicle detection models

Model	mAP	Inference time (ms)	Pros	Cons
SYAM	0.924	35	High accuracy, robust to various conditions, real-time performance	Requires significant computational resources
YOLOv5	0.902	42	Fast inference speed, good accuracy for small objects	May struggle with complex scenes and occlusions
Faster R-CNN	0.918	120	High accuracy, robust to occlusions	Slower inference speed compared to single-stage detectors

4.4.2 Addressing class-specific performance: truck detection

During model evaluation, we observed that the recall for the *truck* category was notably lower (0.73) compared to other vehicle classes. This performance gap highlights a challenge in reliably detecting larger vehicles, which play a vital role in traffic monitoring and integrated traffic management systems.

Upon further analysis, we attribute this limitation primarily to *class imbalance*, as the dataset contained significantly fewer truck images relative to other categories. To partially mitigate this issue, we implemented the following strategies during model training:

- *Class-Aware Data Augmentation:* Augmentation techniques such as random rotations, flips, brightness changes, and zooming were selectively applied to truck images to synthetically increase their diversity and frequency in the training process.
- *Class-Weighted Loss Function*: A class-weighted cross-entropy loss was adopted to assign higher penalties to misclassified truck instances, improving the model's sensitivity to underrepresented classes.
- *Error Analysis and Model Tuning*: We conducted a detailed error analysis of truck misclassifications, allowing us to adjust hyperparameters and refine the training process. The analysis revealed that misclassifications often occurred in scenarios involving occlusion or poor lighting, which will be a focus in future data collection efforts.

While these steps led to modest improvements, the recall for the truck category still indicates room for enhancement. In future work, we plan to enrich the dataset with a broader and more balanced set of truck images captured under varying real-world conditions to further improve detection performance for this critical class.

4.5 Interpretability, privacy, and ethical considerations

As smart city applications increasingly rely on computer vision systems for surveillance and traffic monitoring, addressing data privacy and ethical implications is essential for public acceptance and responsible deployment. In this work, several measures are taken to uphold privacy:

- *Data Anonymization*: The system avoids collecting or processing personally identifiable information (PII), and no facial data is used.
- *Edge Processing*: Where feasible, data processing is performed locally on edge devices to reduce risks associated with cloud-based surveillance.
- *Limited Retention*: Captured video frames are not stored permanently; only aggregate traffic metrics are retained for analysis.
- *Transparency and Oversight*: The development and deployment of the system are aligned with ethical AI principles and respect regulatory frameworks such as the GDPR.
- Future work will further explore the integration of privacy-preserving machine learning techniques and community-driven governance models to ensure ethical alignment in large-scale smart city deployments.

5 Conclusion

Real-time vehicle detection and classification have been successfully accomplished by the SYAM algorithm, which offers strong spatial recognition, high accuracy, and quick processing speeds. With an F1 score of 89.7%, an IoU score of 0.85, and a mAP of 92.4%, the model performs well in a range of vehicle kinds and environmental circumstances. Using YOLOv11 and Non-Maximum Suppression (NMS), SYAM reduces false positives while improving detection accuracy. Nevertheless, issues like a poor truck detection recall (0.73) and high computing requirements point to areas that need more work. Future developments will concentrate on incorporating AI-driven traffic management, enhancing occlusion handling, and creating lightweight models for edge devices. All things considered, SYAM raises the bar for intelligent transportation systems by providing a flexible and scalable solution for applications such as smart city management, traffic monitoring, and congestion control.

6 Limitations and future work

A significant drawback of the proposed SYAM model is its computational complexity, which could impede its practical use in low-resource settings or at a large scale within smart city infrastructures. Although the use of deep feature extraction layers and attention mechanisms in the model improves accuracy and interpretability, it also results in greater memory consumption and longer inference times. In order to tackle this issue, forthcoming research will investigate model optimization strategies like pruning, quantization, and knowledge distillation. The aim of these strategies is to lessen computational requirements without compromising performance. Additionally, integrating lightweight architectures or implementing hybrid edge–cloud frameworks could improve the feasibility of real-time deployment in embedded systems or resource-constrained platforms. These directions aim to enhance the model's scalability and adaptability across diverse operational contexts.

Although the proposed model shows promising results in recognizing traffic conditions, several limitations must be acknowledged. Firstly, the dataset used in this study comprises 439 training images and 125 validation images, which may not fully represent the complete variability and complexity of real-world traffic environments. This relatively small dataset could limit the model's ability to generalize, particularly when faced with unseen conditions such as rare weather patterns, changing lighting, and unusual road situations. To address this, data augmentation techniques—such as image rotation, flipping, scaling, and brightness variation—were utilized to artificially enhance dataset diversity and help improve model robustness. Moreover, transfer learning from pretrained convolutional neural networks was utilized to tap into knowledge from larger-scale datasets, thus partially offsetting the limitations posed by the small training set.

The SYAM model, while demonstrating robust performance in controlled experiments, has yet to be tested in the field within dynamic, real-world urban settings. This signifies a crucial restriction in completely evaluating its real-world applicability for integrated traffic monitoring and management. Future efforts will involve deploying the model in live traffic scenarios to assess its performance under different real-time conditions, including occlusions, changes in illumination, variations in weather, and shifts in traffic density. Such deployment will be essential for comprehending and enhancing the system's robustness and reliability in real smart city scenarios.

Even though SYAM was evaluated against two robust baselines—Faster R-CNN and YOLOv5—we acknowledge that there are no comparisons with more recent models like YOLOv8, YOLOv9, and transformer-based methods such as DETR. These models mark considerable progress in object detection, especially regarding the balance between speed and accuracy. These models were not incorporated into the current experimental phase because of resource constraints. Future research will include benchmarking SYAM against these emerging architectures to further confirm its effectiveness and efficiency in contemporary smart city environments.

In future work, we aim to greatly enhance the dataset by adding more images from a variety of sources that span different geographical areas, traffic densities, weather conditions, and times of day. With this expansion, model training will become more effective and real-world applicability will improve. Additionally, we intend to assess how well the system functions in live settings and investigate how the incorporation of real-time data streams could improve its adaptability and scalability. Finally, the ongoing efforts will target the creation of lightweight model versions that can be deployed on edge devices, as well as an exploration of explainable AI methods to enhance transparency and foster user trust.

Funding The authors received no specific funding for this study.

Data availability The datasets generated and analyzed during the current study are available in the Kaggle repository, https://www.kaggle.com/datasets/abdallahwagih/cars-detection

Declarations

Conflict of interest There is no conflict of interest.

Ethics approval There are no ethical conflicts.

References

1. Ashkanani M, AlAjmi A, Alhayyan A, Esmael Z, AlBedaiwi M, Nadeem M (2025) A self-adaptive traffic signal system integrating real-time vehicle detection and license plate recognition for enhanced traffic management. Inventions 10(1):14. https://doi.org/10.3390/inventions10010014
2. Zulfikar MT (2019) Detection traffic congestion based on twitter data using machine learning. Procedia Comput Sci 157:118–124
3. Kan Z, Tang L, Kwan M-P, Ren C, Liu D, Li Q (2019) Traffic congestion analysis at the turn level using Taxis' GPS trajectory data. Comput Environ Urban Syst 74:229–243
4. Joyo A, Yaqub R, Madamopoulos N (2020) Intelligent traffic-lights management by exploiting smart antenna technology (ITSAT). IEEE Intell Transp Syst Mag 13:154–163
5. Yadav S, Singh S, Chaurasiya VK (2022) Traffic light control using RFID and deep reinforcement learning. AI and IoT for smart city applications. Springer Nature, Berlin, pp 47–64
6. Singh MK, Mishra KD, Sahana S (2021) An intelligent real-time traffic control based on vehicle density. Int J Eng Technol Manag Sci 5:24–29
7. Yang B, Zhang H, Du M, Wang A, Xiong K (2023) Urban traffic congestion alleviation system based on millimeter wave radar and improved probabilistic neural network. IET Radar Sonar Navig 18:327–343
8. Rida N, Ouadoud M, Hasbi A (2020) Traffic signal control for a single intersection-based intelligent transportation system. Digital transformation and innovative services for business and learning. IGI Global, Hershey, pp 159–180
9. AlMulla K, Ashkanani S, Hassan F, AlMesbahi H, Alazmi M, Nadeem M (2024) IoT-based adaptive traffic signal controller to optimize the flow of traffic and reduce congestion. In: Proceedings of the 2024 mediterranean smart cities conference (MSCC), Tetuan, Morocco, 2–4 May 2024; IEEE, Piscataway, NJ, USA, pp. 1–6.
10. Chaware S, Chaware T (2019) Proposed algorithm for smart traffic control using ultrasonic sensors. Int J Eng Adv Technol 8:3912–3915
11. Zou Z, Chen K, Shi Z, Guo Y, Ye J (2023) Object detection in 20 years: a survey. Proc IEEE 111:257–276
12. Rehan H (2023) Internet of Things (IoT) in smart cities: enhancing urban living through technology. J Eng Technol 5(1):1–16
13. Elassy M, Al-Hattab M, Takruri M, Badawi S (2024) Intelligent transportation systems for sustainable smart cities. Transp Eng 16:100252. https://doi.org/10.1016/j.treng.2024.100252
14. Abbas A, and Duijster D (2025) Smart urban infrastructure: AI-powered solutions for sustainable transportation and construction project optimization. https://doi.org/10.13140/RG.2.2.25262.63046
15. Lawe S, Wang R (2016) Optimization of traffic signals using deep learning neural networks. Lect Notes Comput Sci 9992:403–415. https://doi.org/10.1007/978-3-319-50127-7_35
16. Talaat FM, ZainEldin H (2023) An improved fire detection approach based on YOLO-v8 for smart cities. Neural Comput Appl 35(28):20939–20954
17. Alshathri S, Talaat FM, Nasr AA (2022) A new reliable system for managing virtual cloud network. Comput, Mater Contin 73(3):5864–5885
18. Talaat FM, and Hassan E (2021) Artificial intelligence in 3D printing. In Enabling machine learning applications in data science: proceedings of arab conference for emerging technologies 2020, Springer Singapore, pp 77–88

19. Talaat FM, Kabeel AE, Shaban WM (2024) The role of utilizing artificial intelligence and renewable energy in reaching sustainable development goals. Renew Energy 235:121311
20. Murendeni R, Mwanza A, Obagbuwa IC (2025) Using a YOLO deep learning algorithm to improve the accuracy of 3D object detection by autonomous vehicles. World Electr Veh J 16(1):9. https://doi.org/10.3390/wevj16010009
21. Roopa C et al. (2019) Vehicle detection and classification using image processing. In: 2019 International conference on emerging trends in science and engineering, ICESE 2019, Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICESE46178.2019.9194678
22. S. Maqbool et al. (2019) Vehicle detection, tracking and counting. In: 2018 IEEE 3rd International conference on signal and image processing, ICSIP 2018, Institute of Electrical and Electronics Engineers Inc., pp 126–132. https://doi.org/10.1109/SIPROCESS.2018.8600460
23. Krishnamoorthy R, Manickam S (2018) Automated traffic monitoring using image vision. In: Proceedings of the international conference on inventive communication and computationaltechnologies, ICICCT 2018, Institute of Electrical and Electronics Engineers Inc., pp 741–745. https://doi.org/10.1109/ICICCT.2018.8473086
24. Asha CS, Narasimhadhan AV (2018) Vehicle counting for traffic management system using YOLOand correlation filter. In: 2018 IEEE international conference on electronics, computingand communication technologies, CONECCT 2018, Institute of Electrical and ElectronicsEngineers Inc. https://doi.org/10.1109/CONECCT.2018.8482380
25. Alhussan AA, Talaat FM, El-kenawy ESM, Abdelhamid AA, Ibrahim A, Khafaga DS, Alnaggar M (2023) Facial expression recognition model depending on optimized support vector machine. Comput, Mater Contin 76(1):500–515
26. Prasittisopin L (2024) How 3D printing technology makes cities smarter: a review, thematic analysis, and perspectives. Smart Cities 7(6):3458–3488
27. Talaat FM, Aljadani A, Badawy M, Elhosseini M (2024) Toward interpretable credit scoring: integrating explainable artificial intelligence with deep learning for credit card default prediction. Neural Comput Appl 36(9):4847–4865
28. Talaat FM, Gamel SA (2023) Predicting the impact of no. of authors on no. of citations of research publications based on neural networks. J Ambient Intell Humaniz Comput 14(7):8499–8508
29. Talaat FM, Ali SH, Saleh AI, Ali HA (2020) Effective cache replacement strategy (ECRS) for real-time fog computing environment. Clust Comput 23(4):3309–3333
30. Talaat FM, El-Sappagh S, Alnowaiser K, Hassan E (2024) Improved prostate cancer diagnosis using a modified ResNet50-based deep learning architecture. BMC Med Inform Decis Mak 24(1):23
31. Hassan E, Talaat FM, Adel S, Abdelrazek S, Aziz A, Nam Y, El-Rashidy N (2023) Robust deep learning model for black fungus Detec6tion based on Gabor filter and transfer learning. Comput Syst Sci Eng 47(2):1507
32. Talaat FM, Aljadani A, Alharthi B, Farsi MA, Badawy M, Elhosseini M (2023) A mathematical model for customer segmentation leveraging deep learning, explainable AI, and RFM analysis in targeted marketing. Mathematics 11(18):3930
33. Hassan E, Talaat FM, Hassan Z, El-Rashidy N (2023) Breast cancer detection: a survey. Artificial intelligence for disease diagnosis and prognosis in smart healthcare. CRC Press, Boca Raton, pp 169–176
34. Talaat FM, Gamel SA (2023) A2M-LEUK: attention-augmented algorithm for blood cancer detection in children. Neural Comput Appl 35(24):18059–18071
35. Shaban WM, Kabeel AE, Attia MEH, Talaat FM (2024) Optimizing photovoltaic thermal solar systems efficiency through advanced artificial intelligence driven thermal management techniques. Appl Therm Eng 247:123029
36. Talaat FM, El-Gendy EM, Saafan MM, Gamel SA (2023) Utilizing social media and machine learning for personality and emotion recognition using PERS. Neural Comput Appl 35(33):23927–23941
37. ZainEldin H, Gamel SA, Talaat FM, Aljohani M, Baghdadi NA, Malki A, Badawy M, Elhosseini MA (2024) Silent no more: a comprehensive review of artificial intelligence, deep learning, and machine learning in facilitating deaf and mute communication. Artif Intell Rev 57(7):188
38. Chen Z, Wang C, Luo H, Wang H, Chen Y, Wen C, Yu Y, Cao L, Li J (2016) 'Vehicle detection in high-resolution aerial images based on fast sparse representation classification and multiorder feature.' IEEE Trans Intell Trans Syst 17(8):2296–2309
39. Chen Z, Guo H, Yang J, Jiao H, Feng Z, Chen L, Gao T (2022) Fastvehicle detection algorithm in traffic scene based on improved SSD. Measurement 201:111655. https://doi.org/10.1016/j.measurement.2022.111655
40. Dai X, Wei X (2018) HybridNet: a fast vehicle detection system for autonomous driving. Signal Process: Image Commun. https://doi.org/10.1016/j.image.2018.09.002
41. Alaba SY, Ball JE (2022) A survey on deep-learning-based LiDAR 3D object detection for autonomous driving. Sensors 22(24):9577. https://doi.org/10.3390/s22249577
42. Mittal U, Potnuru R and Chawla P (2020) Vehicle detection and classification using improved faster region based convolution neural network. In: 2020 8th international conference on reliability, infocom technologies and optimization (trends and future directions) (ICRITO), Noida, India, pp 511–514. https://doi.org/10.1109/ICRITO48877.2020.9197805
43. https://www.kaggle.com/datasets/abdallahwagih/cars-detection

 https://doi.org/10.1007/s00521-025-11434-9

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Fatma M. Talaat[1,2] · Rana Mohamed El-Balka[3,4] · Sara Sweidan[5,6] · Samah Adel Gamel[7] · Aya M. Al-Zoghby[1,8]

✉ Fatma M. Talaat
fatma.nada@ai.kfs.edu.eg

Rana Mohamed El-Balka
RannaMohamedElbalka@std.mans.edu.eg

Sara Sweidan
Sweidan_ds@fci.bu.edu.eg

Samah Adel Gamel
sgamel@horus.edu.eg

Aya M. Al-Zoghby
aya_el_zoghby@du.edu.eg; aya_el_zoghby@nmu.edu.eg

1 Faculty of Artificial Intelligence, Kafrelsheikh University, Kafrelsheikh 33516, Egypt

2 Faculty of Computer Science and Engineering, New Mansoura University, Gamasa 35712, Egypt

3 Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt

4 Electronic Engineering Department Higher Institute of Engineering and Technology, Manzala 35642, Egypt

5 Artificial Intelligence Department, Faculty of Computers and Artificial Intelligence, Benha University, Banha, Egypt

6 Faculty of Computer Science and Engineering, New Mansoura University, New Mansoura, Egypt

7 Faculty of Engineering, Horus University Egypt HUE, New Damietta 34517, Egypt

8 Computer Science Department, Faculty of Computers and Artificial Intelegence, Damietta University, Damietta, Egypt