International Journal of Sustainable Development Through AI, ML and IoT

Volume 4 | Issue 1 | 2025
ISSN(Online):2584-0827
https://ijsdai.com/index.php/IJSDAI/index

Vehicle Speed Estimation using Convolutional Neural Networks

[1] Lavika Khattar, [2] Ishika Kansal, [3] Cosmena Mahapatra, [4] Meenu Chopra
[1,2,3,4] Vivekananda Institute of Professional Studies – TC, India
[1] lavikakresearch@gmail.com, [2] ishikakansalresearch@gmail.com,
[4] drmeenuchopravips@gmail.com, [3] cosmenamahapatra1@gmail.com

ARTICLE INFO
Received: 01st June 2025
Revised: 17th June 2025
Accepted: 20th July 2025

ABSTRACT

Speed detection plays a crucial role in traffic monitoring, significantly contributing to the enhancement of vehicle efficiency and safety on roads. This paper highlights the importance of computer vision techniques to analyze video frames captured by cameras for detecting vehicles on roads and calculating their absolute speeds. Convolutional Neural networks (CNN) have demonstrated high accuracy in this domain. In this study, we propose a CNN-based approach to estimate vehicle speeds from video data. Specifically, we employ CNN-based object detection algorithms YOLO (You Only Look Once) and SSD (Single Shot Detection) to identify vehicles in each frame of the video. These approaches are tested using real-world scenarios, demonstrating their potential to enhance vehicle detection systems and reduce road accidents. We then compute the absolute speed of each vehicle based on its detection across frames. The results highlight and compare the performance of YOLO and SSD in both vehicle detection and speed estimation, highlighting their effectiveness for real-time traffic monitoring applications.

Keywords:
Vehicle speed detection, Convolutional neural networks, CNN, Object Detection YOLO, Speed detection system, SSD, Single shot Detection

1. Introduction

Vehicle speed monitoring technologies provide information or data that is important to improve the efficiency and safety of traffic on roads. Traditional speed-detection systems, such as radar-based solutions, typically rely on external sensors and video outputs for vehicle detection, speed monitoring, counting and tracking.

However, these sequential systems may occasionally produce inaccurate results due to limitations in detection precision, environmental interference, or occlusion. To address these challenges, vehicle speed detection employing cutting-edge technologies, such as convolutional neural networks (CNN), which provide more accurate and robust performance than traditional methods. This deep learning technology is inspired by the brain and can automatically extract features from the data without the need for manual feature extraction. For instance, information on the speed and density of vehicles can be used to reduce traffic jams and ensure the dynamic routing of vehicles [16]. CNN architecture is straightforward when it comes to recognizing real-time emergency signals and can predict the current vehicle velocity which is essential for quickly responding to hazardous situations. Beyond this, CNNs can also help in accident analyses. The detailed road scenes captured by cameras can serve as valuable insights for post-incident investigations.

This paper covers the significance of the use of CNN model, particularly YOLO and SSD to enhance road safety and driving autonomy with the help of the concept of object detection. Furthermore, this would enhance the idea of complex road scenes and diverse driving scenarios around the world.

1.1 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a type of Deep Learning neural network architecture that excel at tasks involving image and video analysis. They are particularly useful in computer vision applications like image classification, object detection, and facial recognition. Computer vision is a field of Artificial Intelligence that enables computers and machines to see the visual world by analyzing images and video data. CNNs are like smart helpers for computers to understand pictures and videos. They're highly effective at analyzing images or videos and figuring out what's in them. They help computers pick out things like faces, objects, or patterns, which is really handy for tasks like identifying animals in photos or finding specific items in videos. Recently, Computer Vision Techniques have been pushing the development of robust traffic monitoring systems. Such methods utilize images captured by video cameras to infer important traffic features, such as vehicle speed and traffic density.

CNN is a of deep learning model that is used for processing data that has a grid pattern, such as images, which are inspired by the organization of animal visual cortex and designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns [11]. A convolutional Neural Network is the extended version of an Artificial Neural Network that is primarily used to extract features from the grid-like matrix dataset. It comprises of many layers like the Convolutional layer, Activation layer, Pooling layer, and fully connected layer. Here are the layers of CNN:

Convolutional layers: The Convolutional layer is used to implement filters on the input image which help to extract its features. The convolutional layer is like a detective searching for clues in a picture. It uses small search patterns, called filters, to scan the image pixel by pixel. These filters help the detective find important features like edges, textures, and shapes hidden in the image.

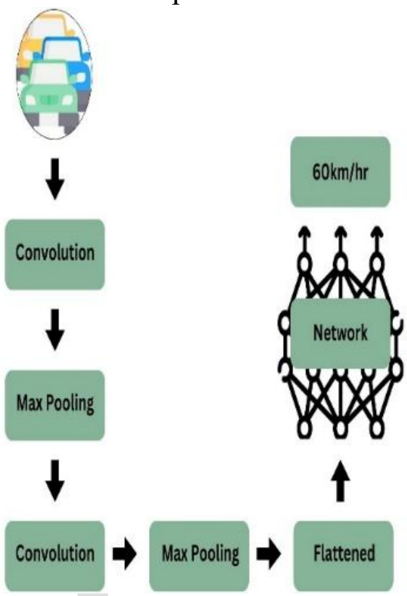

Figure 1: CNN architecture

Activation Layer: The Activation layer applies an element-wise activation function to produce output from the convolutional layer using RELU. The activation function, like ReLU (Rectified Linear Unit), is a set of rules or a formula that the activation layer uses to process the output from the convolutional layer. It goes through each element and applies the rules. For example, with ReLU, the rule is: if the information is positive (greater than zero), keep it unchanged; but if it's negative (less than zero), ignore it (treat it as zero). This way, the activation layer filters out the irrelevant information and keeps only the most crucial parts.

Pooling Layer: This layer is like a zoom-out button for the computer. It takes the detailed pictures created by the earlier layers and makes them smaller and simpler. This helps the computer work faster and use less energy. But it still keeps the important parts of the picture that help it recognize things like objects or patterns.

Fully Connected Layer: The Fully Connected Layers is also known as the dense layer; this layer connects every neuron in one layer to every neuron in the next layer. It takes the output from the previous layers and combines them to make predictions or classifications.

2.Literature Review

Numerous studies have focused on estimating and detecting vehicle speed on roadways, with video-based systems being the most commonly explored approach.

In one such study [1], researchers addressed the challenge of determining traffic flow based on video surveillance camera data by employing SORT tracker and an advanced modern Faster R-CNN detector, achieving 78% accuracy in estimating vehicle speed. The research highlights the importance of computer vision in analyzing video frames from cameras to determine the speed of vehicles on road. Similarly, the paper titled "Faster R-CNN and YOLO based vehicle detection: a survey" [5] presents a comprehensive review of existing deep neural networks for vehicle detection and tracking. The study classifies various vehicle detection methods based on Faster R-CNN and YOLO, highlighting their roles in enhancing road safety and driving autonomy with help of object detection. In contrast to traditional radar-based speed detection systems, the research in "Radar- based 2D Car Detection using Deep Neural Networks" [6] focuses on robust object detection in radar data. The study proposes an adaptation of the YOLOv3 neural network to operate on sparse radar point clouds, providing a novel approach for radar-based vehicle detection. Additionally, [7] develops a technique to enhance the efficiency of vehicle detection during low light and adverse weather conditions using a CNN approach. The CNN model, trained with 177 images and tested with 124 images, achieves significantly improved detection accuracy.

The paper [14] "An algorithm for highway vehicle detection based on Convolutional Neural network" builds upon existing research in the field of vehicle detection and traffic surveillance cameras. It proposes a novel approach that addresses the challenge of accurately detecting vehicles of different sizes. The results of this study indicate significant improvement over existing methods such as Faster R-CNN and SSD, highlighting the effectiveness of CNN-based framework for highway vehicle detection.

These studies emphasize the importance of deep learning techniques in traffic analysis and vehicle speed detection. By utilizing advanced computer vision technologies, researchers can enhance road safety and driving autonomy, paving the way for more safer traffic systems.

2.1 Vehicle speed detection technologies

In this study, we investigate how computer vision can analyze video frames from cameras to find out how fast vehicles are moving on the road. Emphasizing the significance of employing deep learning techniques in traffic analysis and speed estimation, the study sheds light on the limitations of traditional speed detection methods, paving the way for the integration of machine learning solutions.

"Exploring Machine Learning-Based Solutions for Vehicle Speed Detection" critically examines conventional approaches to speed detection, identifying potential pitfalls and illustrating how machine learning can address these shortcomings. "Machine Learning for Vehicle Speed Detection" provides a foundational insight into machine learning principles and their applications in determining vehicle speed. Several computational models capable of performing vehicle speed estimation are discussed, alongside considerations of data quality and acquisition methods, with a focus on the inherent challenges of collecting reliable data from high-risk environments such as busy roads.

Moreover, the study categorizes different methods of vehicle detection based on CNN models particularly, SSD (Single shot Detection) and YOLO (You only look Once), showing how they improve road safety and drive autonomy through object detection. This brings up some unique challenges. We talk about problems like figuring out how fast vehicles are going and how many there are. We also explain how machine learning is used in the real world to estimate vehicle speed.

3. Methodology

This study leverages Convolutional Neural Networks (CNNs) to detect and estimate vehicle speed using video data. Pre-recorded road surveillance videos are processed to identify and track vehicles across frames, enabling speed estimation based on object displacement over time. Two models, YOLO and SSD, are used due to their real-time capabilities and accuracy in traffic scenarios. Video processing and visualization tasks are handled using OpenCV, which assists in frame-wise extraction and annotation.

The models are applied to preprocessed video frames, and their performance in detecting vehicles and estimating speed is compared. Below is brief architectural overview of both models along the implemented workflow.

3.1 YOLO algorithm

The Convolutional neural network (CNN) architecture known as YOLO, or “You only look once” is utilized for image categorization and object recognition applications. YOLO is an object detection algorithm capable of detecting multiple objects in an image in real time. It’s efficient in real-time object detection tasks, including vehicle detection, stems from its ability to detect objects in images in a single forward pass. YOLO does not apply the classifier to multiple regions of an image, instead it divides the image into grids and then predicts bounding boxes along with class labels for each grid cell. It directly outputs the position and category of the bounding box through the neural network [12].

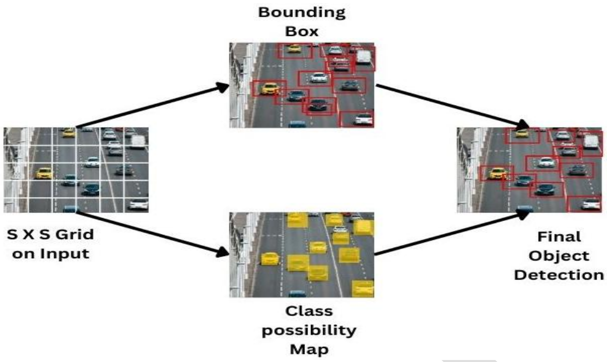

Figure 2: Object detection sequence of YOLO

YOLO is versatile than preceding CNN architectures as it can detect wide range of objects in various contexts. Its speed, efficiency and real -time capabilities make it well suited for application requiring fast object detection, such as autonomous vehicles, surveillance systems and robotics. It functions by dividing the image into a grid of cells and then for each grid cell, YOLO predicts bounding boxes that potentially contain objects. Along with bounding box prediction, YOLO calculates ‘object score’, that indicates the likelihood that the box contains a meaningful object and also predicts the class of the object detected. After predicting bounding boxes, object score and class probability, YOLO applies Non-Max Suppression (NMS) to filter out redundant bounding boxes. The final output of YOLO consists of the selected bounding boxes, along with their associated class labels.

For vehicle speed detection, YOLO’s speed and real-time capabilities are invaluable. It can simultaneously track and measure the speed of multiple vehicles, providing crucial information for traffic management and research studies on traffic patterns.

3.2 SSD algorithm

The SSD (Single Shot Detector) method is widely recognized for its effectiveness in object detection. It utilizes a deep CNN model that focuses on regression to address the challenge of target detection. This model is capable of swiftly and accurately detecting and tracking objects in both images and video frames, allowing for end-to-end training and enabling real-time detection with high precision.

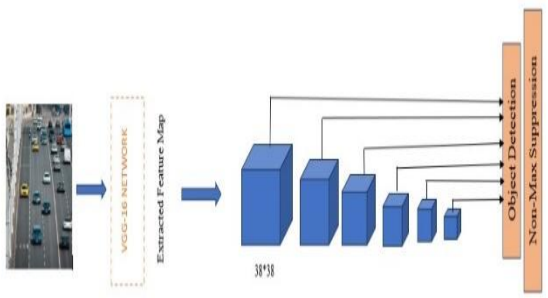

Figure 3: Object detection sequence of SSD

The concept behind SSD involves combining features from various networks like YOLO and R-CNN to achieve rapid detection speed while preserving high detection quality. Multiple bounding box predictions are generated by the last few layers of the network, each responsible for progressively smaller bounding boxes. The final prediction results from the union of all these predictions. Compared to other single stage methods, SSD demonstrates more uniform performance even with a smaller input image size [13].

The CNN network utilized in SSD is fully convolutional, with VGG-16 as the base network. It incorporates numerous auxiliary convolutional layers that decrease in size progressively. To effectively detect small objects, SSD employs shallower layers with higher resolution. For objects of various sizes, SSD conducts multi-scale detection by operating on multiple convolutional feature maps. Each of these maps predicts category scores and box offsets.

3.3 Workflow

This study explores the domain of vehicle recognition in videos, essential for applications such as monitoring and autonomous driving [9]. Leveraging, CNN models, particularly YOLO and SSD, renowned for their real time processing capability and high efficiency, we propose a method for vehicle speed estimation.The process of identifying and estimating the speed of vehicles in a video involves the following key steps:

1. Segmenting the video into frames: The video is partitioned into frames for analysis.

2. Analyzing each frame using a pre-trained CNN model

3. Detecting Vehicles in the video frame

4. Tracking vehicles across frames: Once vehicles are detected, a tracking mechanism is initiated to monitor their movement across frames.

5. Drawing boundary boxes for detecting vehicles: Boundary boxes are drawn around each detected vehicles, visually delineating their location and dimensions within the frame.

6. Assigning IDs to each detected vehicle: Unique identifiers are assigned to each detected vehicle.

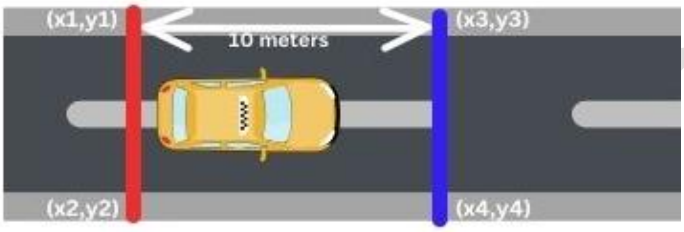

Figure 4: Prototype system for speed recognition

Through the systematic execution of these steps, the proposed method provides a practical and efficient solution for real-time vehicle detection and speed estimation in video frames.

Following the detection and tracking of vehicles, a speed estimation algorithm estimates the speed of vehicles by measuring the time taken to travel between two predefined reference lines in the video frame (as illustrated in Figure 4). In this study, the distance between these lines is assumed to be 10 meters. By combining this time interval with the known distance between the lines, the algorithm computes the speed using the standard formula:

$$\text{Speed} = \frac{\textbf{Distance}}{\textbf{Time}}$$

With each frame, the tracker recalculates the position of vehicles in the scene and returns the path taken by each of them in the last frames. The execution of the trace block is presented in pseudocode format in Figure 5.

While, the main loop efficiently processes each frame of the video, detecting vehicles, tracking their movement, calculating their speeds, and providing visual for analysis and monito monitoring purpose.

TRACKER

1. Initialize:
    a. Center_points={}
    b. Id_count=0
2. Function Update(objects_rect):
    a. Objects_bbs_ids=[]
    b. For each rect in objects_rect:
        i. (x,y,w,h) = rect
        ii. Cx = (x+x+w) // 2
        iii. Cy = (y+y+h) // 2
        iv. Same_objects= False
        v. For each id, pt in Center_points:
            1. Calculate distance between (Cx, Cy) and pt
            2. If distance < threshold:
            3. Update Center points
            4. Add bounding box with id to objects_bbs_ids
            5. Same_objects=true
            6. End Loop
        vi. If same_objects is False:
            1. Assign new ID
            2. Add bounding box with ID to objects_bbs_ids
            3. Increment id_count
        vii. Clean up Center_points dictionary to remove unused IDs
        viii. Return objects_bbs_ids

Figure 5: Vehicle tracking routine

Furthermore, Figure 6 provides the pseudocode outlining the main loop for processing the continuous video stream. This loop handles the initialization of the tracker, reading video frames, vehicles detection, tracker updates and the new bounding boxes, speed estimation, and display of the annotated frames for analysis and monitoring. The video stream can originate from various sources, such as surveillance cameras or traffic monitoring systems.

MAIN LOOP

1. Open the video stream
2. Initialize the tracker
3. While True:
    a. Read frame from video stream
    b. If end of video reached
        i. End Loop
    c. Detect vehicles in the frame using pre-trained CNN model
    d. Update tracker with detected bounding boxes
    e. objects_bbs_ids = tracker.Update(objects_rect)
    f. For each bbox_id in objects_bbs_ids:
        i. Extract bounding box coordinates and object ID
        ii. Calculate center point (Cx, Cy)
        iii. Draw bounding box around detected vehicle
        iv. Assign a unique ID to each vehicle
        v. Perform speed detection
    g. Display frame with bounding boxes
    h. Save frame with annotation
4. Release video file

Figure 6: Vehicle detection and tracking routine

4. Result

To detect vehicles within video frames, pretrained YOLO (You Only Look Once) and SSD (single shot Detection) models from the COCO (Common Objects in Context) datasets were utilized. These models were selected for their established performance in object detection tasks and their pretraining on various range of objects, including vehicles.

Upon evaluation of these pretrained models on real-world traffic scenario, both the models exhibited effective performance in detecting vehicles. YOLO achieved an average detection accuracy of 92.85%, while SSD achieved an average accuracy of 42.42%. Table 1 below summarizes the results of vehicle detection using pretrained YOLO and SSD models on the COCO dataset.

	CORRECT DETECTIONS	INCORRECT DETECTIONS
TOTAL VEHICLES	14	N/A
YOLO	13	N/A
SSD	14	19

Table 1: Comparative analysis

Table 1 illustrates the total number of vehicles present in the video, alongside the number of vehicles detected by the YOLO and SSD models. The accuracy of both the detection models is typically calculated based on the number of correct detections versus the total number of vehicles present in the scene. The formula for accuracy is:

$$\text{Accuracy} = \frac{\textbf{Number of Correct detection}}{\textbf{Total number of vehicles detected}} \times \mathbf{100\%}$$

In the context of the above data:

For YOLO:

$$\text{Accuracy} = \frac{\mathbf{13}}{\mathbf{14}} \times \mathbf{100\%} \approx \mathbf{92.85\%}$$

For SSD:

$$\text{Accuracy} = \frac{\mathbf{14}}{\mathbf{33}} \times \mathbf{100\%} \approx \mathbf{42.42\%}$$

Thus, YOLO model demonstrates substantially better performance in accurately detecting vehicles compared to SSD in the given scenario. With an accuracy of 92.85%, YOLO is able to correctly identify vehicles at a much higher rate than SSD, which has an accuracy of 42.42%. Figure 7 depicts the results obtained using the YOLO model. The analysis of the annotated frame reveals that 6 vehicles were leaving the scene and 7 were entering, totaling 13 detected vehicles.

Figure 8 displays the results using the SSD model with VGG16 as the backbone architecture. Upon analyzing this screenshot, 15 vehicles were entering and 18 were leaving, totaling 33 detections, out of which 19 were incorrect.

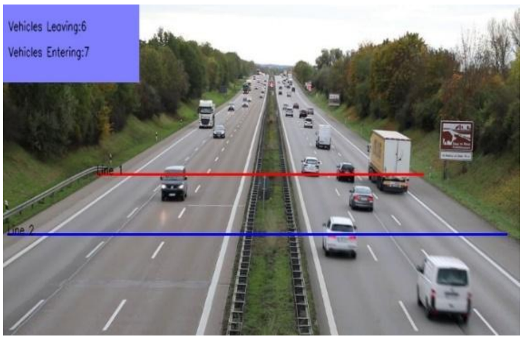

Figure 7: YOLO detection results

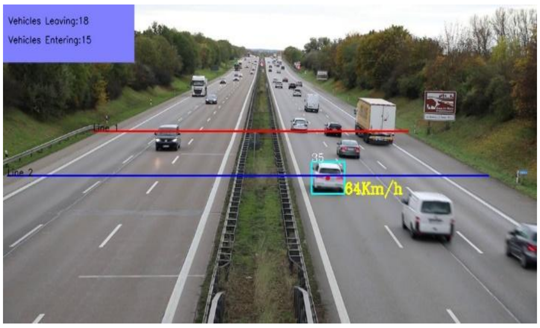

Figure 8: SSD detection results

Based on this comparative analysis, the YOLO model demonstrates substantially better performance in accurate vehicle detection, with a higher detection accuracy and significantly fewer false positives than SSD. Although SSD detected more vehicles overall, its higher number of incorrect detections lowered its precision, making YOLO a more reliable choice for real-world traffic monitoring scenarios.

4.1 MAE Analysis on Custom Dataset

To further evaluate the models' effectiveness in detecting vehicles and estimating their speed, both YOLO and SSD were tested on a custom traffic dataset, we compared the two widely-used architectures: YOLO and SSD. Both models were employed with pre-trained weights from the COCO dataset, without any additional fine-tuning on the custom data.

The evaluation metric used was the Mean Absolute Error (MAE), calculated between the predicted and ground truth bounding box parameters: X-center, Y-center, width, and height. This metric provides a direct assessment of the models' ability to localize vehicles accurately.

MODEL	MAE (X-Center)	MAE (Y-Center)	MAE (Width)	MAE (Height)
YOLO	0.0008	0.0064	0.0021	0.0207
SSD	39.0093	35.19	204.5000	214.6481

Table 2: Mean Absolute Error (MAE) Comparison of YOLO and SSD on Custom Vehicle Dataset

As shown in Table 2, YOLO consistently outperformed SSD across all four MAE metrics. The lower MAE values indicate that YOLO provided more accurate bounding box predictions on the custom dataset, showcasing stronger generalization to new domains—even in the absence of task-specific fine-tuning. In contrast, SSD exhibited relatively higher error rates, suggesting it may benefit from fine-tuning when applied to unseen data distributions. These findings underscore the robustness of YOLO for vehicle detection tasks, especially in real-world, domain-specific scenarios like traffic monitoring.

The relatively higher MAE of SSD further confirms its susceptibility to false positives and inconsistent bounding box predictions, which negatively affect the reliability of speed estimation.

5. Conclusion

In conclusion, this study demonstrates the effectiveness of convolutional Neural network (CNN) models, specifically YOLO and SSD, in real-time vehicle detection and speed estimation from video data. Both models were employed with pre-trained weights on the COCO dataset and evaluated without additional fine-tuning, allowing for a fair comparison.

YOLO achieved a significantly higher detection accuracy of 92.85%, while SSD recorded an accuracy of 42.42%, indicating that YOLO is the more accurate model for vehicle detection tasks in real-world traffic videos. Additionally, the Mean Absolute error (MAE) evaluation on a custom vehicles' dataset revealed that YOLO consistently produced lower errors in bounding box prediction, further highlighting its superior localization performance and generalization capabilities.

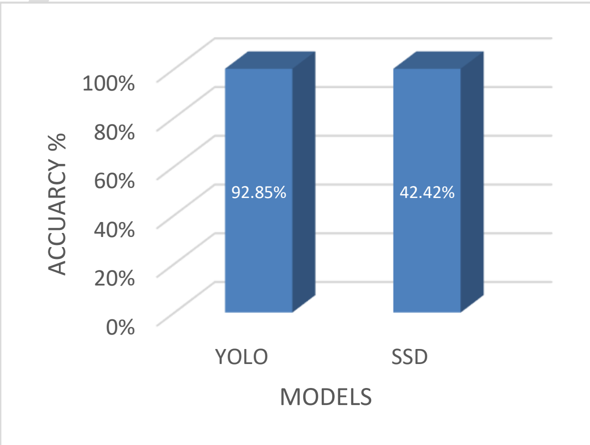

Figure 9: Accuracy comparison of the models

These findings suggest that YOLO is better suited for applications with limited resources for model retraining or domain-specific fine-tuning, offering robust performance out-of-the-box. In contrast, while SSD underperformed in accuracy and localization, its relatively faster processing speed may be beneficial for applications where computational efficiency is a priority, such as edge-device deployment or preliminary object screening.

Overall, both YOLO and SSD provide unique advantages depending on the application context. YOLO's precision and generalization make it ideal for traffic surveillance, smart traffic systems, and safety analytics, whereas SSD’s efficiency could be leveraged in scenarios requiring fast, large-scale processing, such as autonomous navigation or lightweight embedded systems. By integrating such CNN-based approaches, traffic monitoring systems can be significantly enhanced, contributing to smarter urban infrastructure and improved road safety.

Further, it is planned to improve the developed system to increase the accuracy of object recognition and speed estimation.

Author Contribution. Lavika Khattar and Ishika Kansal wrote the code and manuscript of the research with inputs from all authors. Meenu Chopra and Cosmena Mahapatra were involved in overall planning and were involved throughout execution of the study.

References

[1] A. Grents, V. Varkentin, and N. Goryaev, “Determining vehicle speed based on video using convolutional neural network,” Transportation Research Procedia, vol. 50, pp. 192–200, 2020, doi: https://doi.org/10.1016/j.trpro.2020.10.024.

[2] “Speed Estimation from Vibrations Using a Deep Learning CNN Approach | IEEE Journals & Magazine | IEEE Xplore,” ieeexplore.ieee.org. https://ieeexplore.ieee.org/abstract/document/9340328 (accessed Feb. 26, 2024).

[3] S. Patira, “Over speed detection using Artificial Intelligence,” doi: https://doi.org/10.31979/etd.u8qc-9d6e.

[4] R. Itu and R. Danescu, “Fully Convolutional Neural Network for Vehicle Speed and Emergency-Brake Prediction,” Sensors, vol. 24, no. 1, p. 212, Jan. 2024, doi: https://doi.org/10.3390/s24010212.

[5] Hamam Mokayed, Amirhossein Nayebiastaneh, Lama Alkhaled, Stergios Sozos, Olle Hagner, Björn Backe, "Challenging YOLO and Faster RCNN in Snowy Conditions: UAV Nordic Vehicle Dataset (NVD) as an Example", 2024 2nd International Conference on Unmanned Vehicle Systems-Oman (UVS), pp.1-6, 2024.

[6] M. Dreher, E. Erçelik, T. Bänziger and A. Knoll, "Radar-based 2D Car Detection

Using Deep Neural Networks," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 2020, pp. 1-8, doi: 10.1109/ITSC45102.2020.9294546.

[7] I. A. Tarmizi and A. A. Aziz, "Vehicle Detection Using Convolutional Neural Network for Autonomous Vehicles," 2018 International Conference on Intelligent and Advanced System (ICIAS), Kuala Lumpur, Malaysia, 2018, pp. 1-5, doi: 10.1109/ICIAS.2018.8540563.

[8] Unsplash, "Photo by Keszthelyi Timi on Unsplash," unsplash.com, May 18, 2020. https://unsplash.com/photos/cars-on-road-during-daytime-P25wpQ8ndl8 (accessed Apr. 24, 2024).

[9] V. Barth, R. de Oliveira, M. de Oliveira and V. do Nascimento, "Vehicle Speed Monitoring using Convolutional Neural Networks," in IEEE Latin America Transactions, vol. 17, no. 06, pp. 1000-1008, June 2019, doi: 10.1109/TLA.2019.8896823.

[10] Kim, J., Sung, J.-Y., & Park, S. (2020). Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition. 2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia). doi:10.1109/icce-asia49877.2020.9277040

[11] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional Neural networks: an Overview and Application in Radiology," Insights into Imaging, vol. 9, no. 4, pp. 611–629, Jun. 2018, doi: https://doi.org/10.1007/s13244-018-0639-9

[12] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo Algorithm Developments," Procedia Computer Science, vol. 199, pp. 1066–1073, Jan. 2022, doi: https://doi.org/10.1016/j.procs.2022.01.135.

[13] W. Liu et al., "SSD: Single Shot MultiBox Detector," Computer Vision – ECCV 2016, vol. 9905, pp. 21–37, 2016, doi: https://doi.org/10.1007/978-3-319-46448-0_2.

[14] L. Chen, F. Ye, Y. Ruan, H. Fan, and Q. Chen, "An algorithm for highway vehicle detection based on convolutional neural network," EURASIP Journal on Image and Video Processing, vol. 2018, no. 1, Oct. 2018, doi: https://doi.org/10.1186/s13640-018-0350-2.

[15] T. Keerthi *et al*., "An Automatic Vehicle Speed Controlling based on Traffic Signs Recognition using Convolutional Neural Network," *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Delhi, India, 2023, pp. 1-9, doi: 10.1109/ICCCNT56998.2023.10306873.

[16] Shepelev, V., Aliukov, S., Nikolskaya, K. and Shabiev, S. (2020). The Capacity of the Road Network: Data Collection and Statistical Analysis of Traffic Characteristics. Energies, 13(7), p.1765. doi:https://doi.org/10.3390/en13071765.