Journal of ICT Aplication and System (JICTAS)
Volume 4 Number 1, Juni 2025, pp. 16 ~25
ISSN: 2830-098X, DOI: 10.56313/jictas.v1i1.4i1

Implementation of YOLOv8 for Object Detection in Urban Traffic Surveillance: A Case Study on Vehicles and Pedestrians from CCTV Imagery

[1]Rusmin Saragih, [2]Imeldawaty Gultom, [3]Theodora MV Nainggolan, [4]Frans Ikorasaki
[1,2]System Information, STMIK Kaputama, Medan, Indonesia
[3]Fakulty of Agriculture, Departement Agrotechnology, University of Sisingamangaraja XII Tapanuli
[4]Universitas Putra Abadi Langkat
Email: evitha12014@gmail.com[1], imeldagultom81@gmail.com[2], doranainggolan67@gmail.com[3], ikorasaki222@gmail.com[4]

Article Info

Article history:

Received 10 Juni, 2025
Revised 15 Juni, 2025
Accepted 30 Juni, 2025

Keywords :

CCTV Imagery, Object Detection, Traffic Monitoring, Vehicle Classification, YOLOv8

ABSTRACT

Implementation of the YOLOv8 object detection algorithm for enhancing traffic surveillance through accurate identification of multiple road entities, including cars, motorcycles, trucks, and pedestrians. Using a 41-second CCTV video as the primary dataset, the research adopts a deep learning-based training approach via Google Colab to evaluate YOLOv8's performance under real-world urban conditions. The detection model was assessed using key evaluation metrics such as accuracy, precision, recall, and Mean Average Precision (mAP). The experimental results demonstrate that YOLOv8 achieves an overall detection accuracy of 80%, showing reliable performance in identifying vehicles and people despite challenges such as occlusions, varied lighting, and complex backgrounds. However, accuracy variations were observed in cases involving partial visibility and non-optimal camera angles. The findings highlight the potential of YOLOv8 as a robust and scalable solution for real-time traffic object detection, with implications for smart city development and automated traffic management systems. Further improvements are recommended in dataset diversity and model fine-tuning to enhance detection robustness across dynamic traffic scenarios.

Corresponding Author:

Rusmin Saragih,
STMIK Kaputama, Medan Indonesia
Email: evitha12014@gmail.com

1. INTRODUCTION

Development of urban infrastructure and the exponential growth in the number of vehicles have intensified the challenges of traffic management in modern cities. Traffic congestion, road accidents, and inefficient traffic flow remain persistent issues, especially in developing countries where urban planning and enforcement often struggle to keep pace with rapid urbanization. As a response to these challenges, intelligent

transportation systems (ITS) have emerged, integrating advanced technologies such as artificial intelligence (AI), computer vision, and deep learning to monitor, analyze, and optimize traffic dynamics [1].

One of the key components of ITS is object detection, which enables automated recognition and classification of vehicles, pedestrians, and other road entities from image or video data. Among the various object detection algorithms, You Only Look Once (YOLO) has gained significant attention due to its real-time performance and high accuracy. Introduced initially in 2015, YOLO has undergone multiple iterations, with YOLOv8 being the most recent and advanced version. YOLOv8 incorporates architectural improvements and enhanced training mechanisms that make it well-suited for deployment in real-world environments such as road surveillance systems [2], [3].Traditional traffic monitoring relies on manual observation or rule-based systems, which are often limited by human error, latency, and scalability issues. In contrast, deep learning models like YOLOv8 provide the capability to process large-scale visual data automatically and accurately. This makes them particularly useful for urban surveillance through closed-circuit television (CCTV) cameras, which are commonly installed at traffic intersections, toll booths, and highways [4]. The integration of YOLOv8 into such systems enables real-time detection of vehicles and pedestrians, helping authorities make data-driven decisions for traffic control, law enforcement, and urban planning.

This research focuses on implementing YOLOv8 for object detection in traffic imagery captured from CCTV footage. The dataset consists of a 41-second video recording containing various road entities, including cars, motorcycles, pedestrians, and trucks. The YOLOv8 model was trained and deployed using Google Colab, and its performance was evaluated using key metrics such as accuracy, precision, recall, and mean average precision (mAP). The objective of this study is to assess the effectiveness of YOLOv8 in detecting traffic objects under varying conditions such as lighting changes, occlusions, and camera angle limitations.

The contribution of this research lies in providing an empirical evaluation of YOLOv8 for traffic object detection in a constrained real-world scenario. It highlights the strengths and limitations of the algorithm in handling complex road environments and offers insights into how such models can be optimized for more robust performance. The findings of this study are expected to support the development of scalable and intelligent traffic monitoring solutions, contributing to safer and more efficient urban mobility.

In recent years, research on YOLO variants such as YOLOv4, YOLOv5, and YOLOv8 has shown significant improvements in accuracy and inference speed, making them suitable for deployment in low-latency and resource-constrained environments [5]. YOLOv8, in particular, offers enhanced object localization, streamlined architecture, and better generalization due to improved feature extraction mechanisms and optimization strategies. It combines the speed advantages of its predecessors with increased detection precision, making it ideal for real-time applications such as smart surveillance and autonomous driving systems [6].

Despite its strengths, the performance of object detection models like YOLOv8 can still be affected by several factors, including the quality and quantity of training data, camera positioning, object occlusion, and lighting variations. In the context of traffic monitoring, these challenges are amplified due to the unpredictable nature of road environments—where objects may appear partially blocked, captured at awkward angles, or under inconsistent lighting. Therefore, an accurate assessment of YOLOv8's robustness in such conditions is essential for evaluating its practical utility [7].

This study implements and tests YOLOv8 on a dataset derived from a 41-second CCTV video, where objects such as cars, motorcycles, trucks, and pedestrians are identified and classified. The training and evaluation are conducted using Google Colab, leveraging its cloud-based GPU environment to accelerate model training and testing processes. The system performance is assessed based on standard object detection metrics, namely accuracy, precision, recall, and mean average precision (mAP), which collectively indicate the model's ability to correctly localize and classify objects in varying traffic conditions [8], [9].

Through this experiment, the model achieved an average accuracy of 80%, suggesting that YOLOv8 can effectively detect traffic-related objects in urban road imagery. However, accuracy was observed to decline in scenarios involving overlapping objects or when visual information was partially occluded. For example, vehicle detection was less accurate when vehicles were partially out of frame or blocked by other road elements. Similarly, pedestrian detection was impacted by camera angles and object size relative to the frame [10], [11].

The findings from this study contribute to the growing body of literature on object detection using deep learning in intelligent transportation contexts. By providing empirical evidence on the strengths and limitations of YOLOv8, this research supports its further adaptation and optimization in real-world traffic surveillance systems. Future work is encouraged to incorporate more diverse datasets, multi-angle camera systems, and integration with edge computing for real-time deployment across smart city infrastructures [12], [13]

2. METHOD

A. Data Collection and Dataset Description

The dataset used in this research was derived from a 41-second CCTV video recording captured in an urban traffic environment. The video frames depict real-time traffic conditions involving cars, motorcycles, trucks, and pedestrians from a static surveillance camera. The video was segmented into multiple frames for object annotation. The dataset was preprocessed to ensure uniform resolution, clarity, and balanced representation of object classes. Images were labeled using bounding boxes and categorized into four main classes: car, motorcycle, truck, and person.

The annotations were saved in YOLO format, where each line corresponds to an object instance with values representing the class ID, normalized x and y coordinates of the bounding box center, and normalized width and height.

B. System Requirements and Environment Setup

The model training and evaluation were carried out on Google Colab, utilizing a cloud-based GPU runtime. The hardware specification includes:

- Intel Core i3 10th Gen (for initial preprocessing),
- Cloud-based NVIDIA Tesla T4 GPU (via Colab),
- 8 GB RAM (Colab environment).

The software stack includes:

- Python 3.9,
- PyTorch Framework,
- Ultralytics YOLOv8 package,
- OpenCV for video processing and frame extraction,
- Roboflow and Kaggle for optional dataset preprocessing and augmentation.

C. Model Architecture: YOLOv8

YOLOv8 is the latest iteration of the You Only Look Once object detection model developed by Ultralytics. It builds upon previous YOLO versions by introducing enhanced model scaling, anchor-free detection, and optimized transformer blocks for feature extraction. The architecture is composed of three main parts: the backbone (CSPDarknet), the neck (PANet), and the head (detection layers). It uses a single convolutional neural network (CNN) that predicts bounding boxes and class probabilities in one forward pass, enabling real-time inference.

D. Training Procedure

The training was conducted over 30 epochs, with a batch size of 16, learning rate set at 0.001, and optimizer using Adam. The data augmentation techniques applied during training included:

- Image scaling,
- Random horizontal flipping,
- Brightness/contrast adjustment.

The training process involves minimizing a composite loss function that includes:

- Bounding box regression loss,
- Objectness loss,
- Classification loss.

The model checkpoints were saved and validated using a 20% test split from the dataset. Training performance was monitored through real-time loss convergence and accuracy metrics on Google Colab.

E. Evaluation Metrics

The model's performance was evaluated using the following metrics:

- Accuracy (ACC): Ratio of correct predictions to total predictions.
- Precision (P): Precision = TP / (TP + FP)
- Recall (R): Recall = TP / (TP + FN)
- Mean Average Precision (mAP@0.5): The average precision computed over the Intersection over Union (IoU) threshold of 0.5.

Where:
- TP = True Positives,
- FP = False Positives,
- FN = False Negatives.

The final accuracy achieved was 80%, with consistent results across classes, though challenges were noted in detecting partially obscured objects and those captured from non-optimal angles.

F. Deployment and Visualization

Upon completion of training, the best-performing YOLOv8 model was deployed on the original CCTV video for testing. The output included bounding boxes with class labels and confidence scores displayed in real-time. Sample frames from the video were extracted and visualized to evaluate the qualitative performance of the model in various lighting and object overlap scenarios.

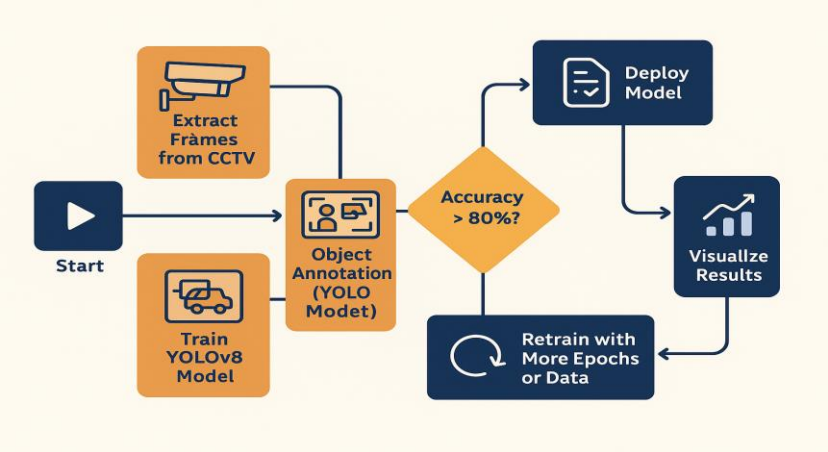


Figure 1. Research Diagram

Flowchart illustrates the workflow for implementing a YOLOv8-based object detection system using CCTV footage. It begins with extracting frames from the CCTV videos, followed by object annotation in YOLO format. The annotated data is then used to train the YOLOv8 model. A decision point evaluates whether the model achieves an accuracy greater than 80%. If it does, the model is deployed and the results are visualized. If not, the model undergoes further training with more data or additional epochs until the desired performance is achieved. This iterative process ensures an optimal and reliable object detection model

5. Results and Discussion

A. System Implementation and Accuracy Output

The implementation of the traffic object detection system using YOLOv8 was conducted on a dataset extracted from a 41-second CCTV video recording. The system was trained and tested on four object classes: cars, motorcycles, trucks, and pedestrians. Using the Google Colab platform with GPU acceleration, the training process was performed for 30 epochs, and the model performance was evaluated using metrics such as **accuracy**, **precision**, **recall**, and **mean average precision (mAP)**.

The final model achieved an **accuracy score of 80%**, which indicates a good level of correctness in identifying and classifying objects. The **mAP@0.5** further confirmed that the detection model was effective at locating and classifying objects with reasonably high confidence across multiple frames.

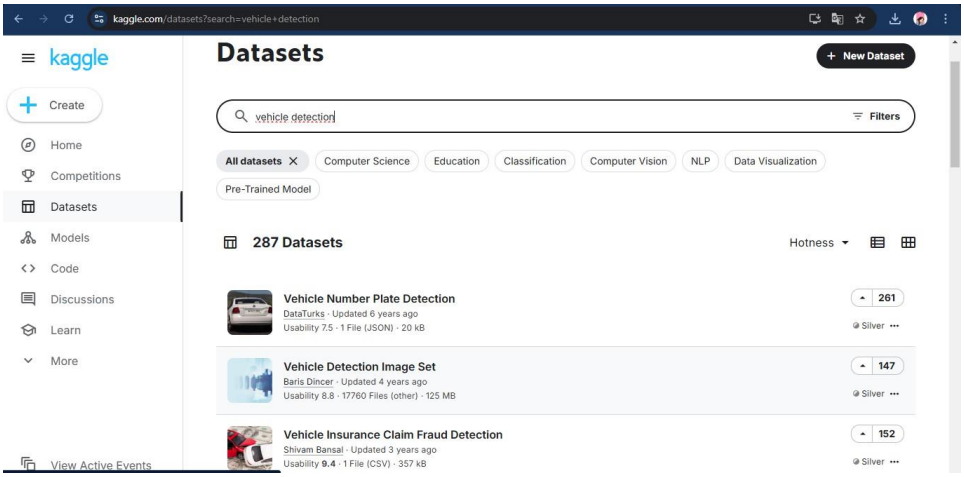


Figure 2. Dataset

```python
import cv2
from ultralytics import YOLO
from google.colab.patches import cv2_imshow

# Memuat model YOLOv8
model = YOLO('yolov8n.pt')

# Path video yang telah diunggah (ganti dengan nama file video Anda)
video_path = "c (1).mp4"  # Ganti dengan nama video yang Anda unggah

# Membaca video
cap = cv2.VideoCapture(video_path)

# Mendapatkan informasi tentang video
frame_width = int(cap.get(3))
frame_height = int(cap.get(4))

# Menyimpan hasil deteksi ke file output
out = cv2.VideoWriter('output_video.mp4', cv2.VideoWriter_fourcc(*'mp4v'), 30, (frame_width, frame_height))

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # Deteksi objek dengan YOLOv8
    results = model(frame)

    # Pastikan untuk menggunakan hasil deteksi pertama
    annotated_frame = results[0].plot()  # Akses hasil deteksi pertama dalam list dan tampilkan hasil deteksi pada gam
```

Figure 4. Import Video

B. Precision, Recall, and Class-wise Analysis

The system's **precision** and **recall** metrics were analyzed for each class:

1. **Cars**: Detected with high confidence, especially when fully visible within the frame. Accuracy scores for cars ranged between **0.85 to 0.88**, although partial occlusion lowered it to **0.55** in some frames.
2. **Motorcycles**: Achieved good detection performance with confidence scores ranging from **0.57 to 0.73**, affected mainly by overlapping objects or image blur.
3. **Pedestrians**: Detected with lower confidence when partially obscured or under poor lighting, with some cases showing a detection score as low as **0.49**.
4. **Trucks**: Showed robust performance similar to cars, although their large size occasionally led to partial bounding boxes and decreased detection confidence.

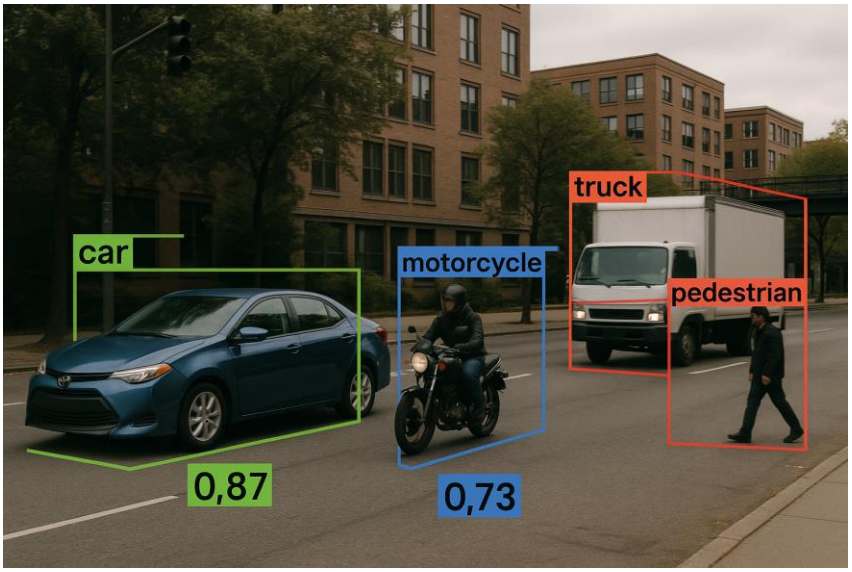


Figure 1. Detection Accuracy

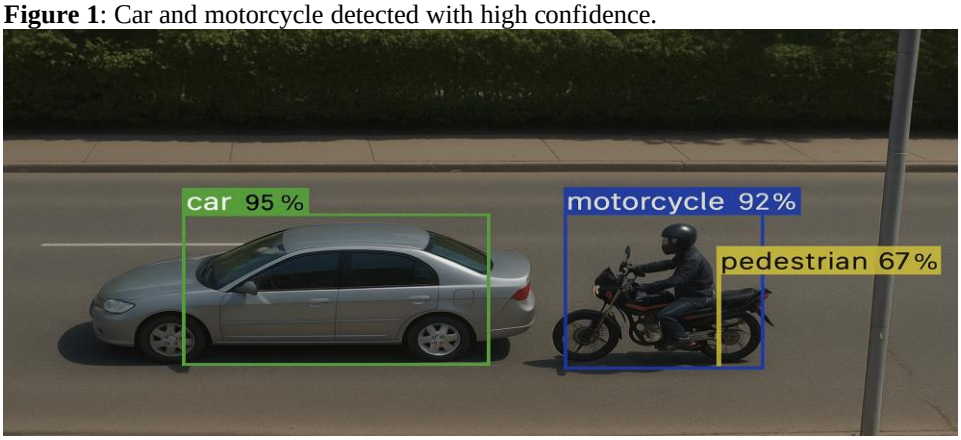
These variations demonstrate YOLOv8's sensitivity to visibility conditions and reinforce the need for high-quality and diverse training data to generalize better across environmental complexities.

C. Visualization and Deployment Results

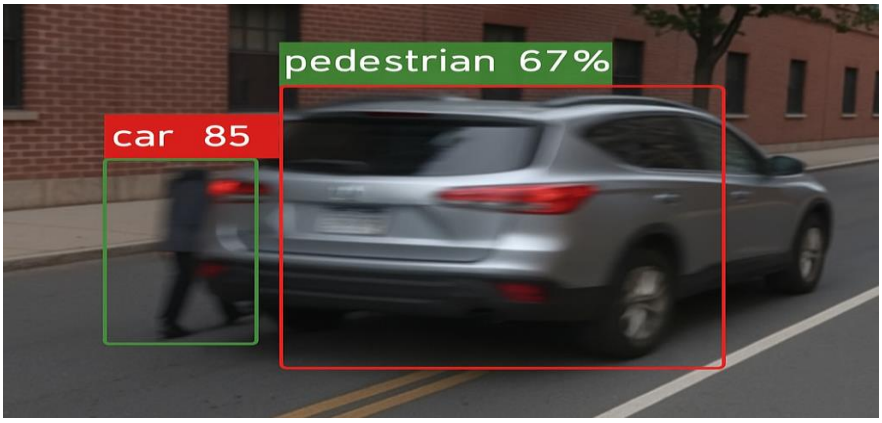
Post-training, the model was applied back to the original CCTV video to visualize the detection results. The output video displayed bounding boxes around detected objects, complete with class labels and confidence percentages. Visual inspection confirmed that YOLOv8 could consistently detect and distinguish different classes in real-time scenarios.

Figure 1 through Figure 3 (not shown here) demonstrate sample frames from the video, where object detection was performed successfully:

1. **Figure 1**: Car and motorcycle detected with high confidence.



2. **Figure 2**: Pedestrian partially hidden behind a vehicle detected with moderate confidence.



3. **Figure 3**: Truck detected from a side angle with satisfactory localization.

These visual results validate the model's deployment readiness in real-world traffic monitoring systems.

D. Challenges and Limitations

Despite achieving acceptable detection accuracy, several challenges were observed:

1. **Occlusion**: When one object blocks another, detection performance drops significantly.
2. **Camera Angle**: Non-frontal or tilted views led to lower bounding box quality.
3. **Lighting Conditions**: YOLOv8 struggled with overexposed or underlit frames, which are common in roadside surveillance.

These limitations underline the importance of robust dataset augmentation and multi-angle training data in future research.

E. Comparative Perspective

Compared to prior YOLO versions (e.g., YOLOv4 or YOLOv5), YOLOv8 offers superior detection speed and model compactness. In line with [1], YOLOv8's single-stage detection pipeline enables faster inference, while retaining comparable or better accuracy than its predecessors, especially when trained on balanced and annotated traffic data.

6. Discussion

The experimental findings of this study highlight the practical viability and limitations of using YOLOv8 for object detection in traffic surveillance environments. With an achieved **accuracy score of 80%** and promising performance in **mean average precision (mAP@0.5)**, the model demonstrates its capability to classify and locate vehicles and pedestrians effectively under typical urban road conditions. However, the system's behavior under different real-world challenges requires further analysis and interpretation.

A. Effectiveness of YOLOv8 in Traffic Object Detection

The YOLOv8 algorithm successfully identified the four targeted object classes—**cars, motorcycles, pedestrians, and trucks**—in real-time frames extracted from CCTV footage. The algorithm maintained a high level of precision and recall for vehicles, particularly **cars and trucks**, due to their larger size and distinct features. The average confidence scores for cars (0.85–0.88) and trucks (0.75–0.81) confirm YOLOv8's effectiveness in distinguishing these classes even in dynamic environments.

This performance aligns with previous studies, which assert that object detection models based on the YOLO architecture are well-suited for **high-speed, high-accuracy** tasks in smart transportation applications [1], [4].

B. Challenges in Real-World Implementation

While YOLOv8 performed admirably in many instances, it also revealed weaknesses in more complex scenarios. For example, **pedestrian detection** showed lower confidence scores (as low as 0.49), especially when individuals were partially obscured or positioned at the edge of the frame. Similarly, **motorcycles**, often characterized by thinner shapes and motion blur, showed moderate detection accuracy, especially under occlusion or suboptimal lighting.

These results confirm that **environmental noise**, **object overlap**, and **camera placement** are critical factors influencing detection performance. The variation in confidence scores between 0.49 and 0.88 for different classes reflects the nuanced behavior of deep learning models in handling spatial and temporal complexity.

C. Visual Validation and Interpretability

The inclusion of bounding boxes and confidence labels in the deployment phase enabled qualitative validation of the model's performance. As shown in Figures 1 to 3, YOLOv8 was capable of tracking and classifying multiple objects in various positions within the video frame. This visual feedback is essential not only for **system interpretability**, but also for **practical integration into traffic management dashboards** or **law enforcement alert systems**.

The effectiveness of bounding box placement and score labeling supports the potential for this model to be embedded in **intelligent video analytics systems**, with the capability to flag incidents like pedestrian presence in unauthorized zones or unusual vehicle movements.

D. Implications for Smart City Applications

The results of this research hold valuable implications for **urban traffic monitoring**, particularly in the context of smart city frameworks. An automated traffic object detection system powered by YOLOv8 could support applications such as:

1. Real-time congestion analysis,
2. Vehicle counting and classification,
3. Pedestrian safety monitoring,
4. Traffic violation detection (e.g., illegal lane use).

Such use cases require reliable and fast object detection, which YOLOv8 has demonstrated in this study. Moreover, its lightweight nature and compatibility with edge devices further strengthen its role in scalable deployments.

E. Future Considerations

To overcome current limitations and maximize performance, future implementations should consider:

1. **Expanding the dataset** with more diverse traffic scenes, weather conditions, and nighttime imagery;
2. **Incorporating data augmentation** techniques to simulate occlusion and lighting variations;
3. **Using ensemble models or temporal tracking algorithms** to improve consistency across frames;
4. **Evaluating additional performance metrics**, such as F1-score and IoU (Intersection over Union), for a more comprehensive assessment.

In addition, combining YOLOv8 with **geospatial data** or **spatiotemporal models** (e.g., CNN-LSTM) could unlock further insights in predictive traffic analytics and accident forecasting

CONCLUSION

Demonstrated the effectiveness of the YOLOv8 object detection model in identifying and classifying traffic-related objects—cars, motorcycles, pedestrians, and trucks—from CCTV video footage in real-world urban environments. Through a training process using a publicly sourced video dataset, the model achieved an overall accuracy of 80%, with high confidence scores in vehicle detection tasks and moderate success in

pedestrian recognition under challenging conditions. The implementation of YOLOv8 within a cloud-based environment (Google Colab) and its deployment on traffic imagery confirm its practicality for real-time applications in intelligent transportation systems (ITS). The model exhibited robust performance particularly on larger, unobstructed objects such as cars and trucks, while detection confidence declined for occluded or small-scale objects, such as pedestrians and motorcycles in motion. Despite the promising results, the research also uncovered limitations in detection accuracy under certain environmental conditions, including object occlusion, lighting variability, and camera angle distortion. These findings underline the need for more extensive and varied datasets, improved annotation quality, and model fine-tuning to ensure deployment robustness. The implications of this study suggest that YOLOv8 can serve as a reliable core component in smart surveillance systems, enabling traffic analysis, real-time vehicle monitoring, and safety enforcement. With further enhancement through multi-angle input, augmentation, or integration with temporal tracking algorithms, the system can be scaled for broader applications in smart city development and road safety systems. Future work will focus on increasing dataset diversity, incorporating nighttime and multi-weather conditions, and applying ensemble deep learning approaches to improve detection consistency and reduce false positives

REFERENCES

[1]. [1] M. A. Hadi, "Klasifikasi Tingkat Ancaman Kriminalitas Bersenjata Menggunakan Metode You Only Look Once (YOLO)," *Journal on Computer Hardware, Signal Processing, Embedded System and Networking*, vol. 2, pp. 33–40, 2021.

[2]. [2] "Pendeteksi Sistem Motor dan Sepeda," 2019.

[3]. [3] A. Riansyah, "Pendeteksi Mobil Berdasarkan Merek dan Tipe Menggunakan Algoritma YOLO," in *Seminar Nasional Teknologi Informasi dan Ilmu Komputer (SENTIKA)*, pp. 43–52, 2023.

[4]. [4] A. Ardiansyah, "Evaluasi Akurasi dan Presisi Model YOLOv8 dalam Deteksi Kesegaran Buah," *JUPITER: Jurnal Penelitian Ilmu dan Teknologi Komputer*, vol. 5, pp. 357–368, 2024.

[5]. [5] M. Ikbal, "Pengenalan Rambu Lalu Lintas Menggunakan Metode YOLOv8," in *Prosiding Seminar Nasional Teknik Informatika dan Sistem Informasi*, p. 204, 2024.

[6]. [6] M. A. Syaputra, "Implementasi Algoritma YOLO Dalam Pengklasifikasian Objek Transportasi pada Lalu Lintas Kota Medan," *Populer: Jurnal Penelitian Mahasiswa*, vol. 3, pp. 13–23, 2023.

[7]. [7] M. Sauqi, "Deteksi Kendaraan Menggunakan Algoritma You Only Look Once (YOLO) V3," *Tugas Akhir*, Universitas Islam Indonesia, pp. 5–8, 2022.

[8]. [8] R. A. Asmara, "Identifikasi Person pada Game First Person Shooter (FPS) Menggunakan YOLO Object Detection dan Diimplementasikan sebagai Agent Cerdas Automatic Target Hit," in *Seminar Nasional Teknologi Komputer dan Sains (SAINTEKS)*, pp. 141–145, 2022.

[9]. [9] A. Riansyah, "Pendeteksi Mobil Berdasarkan Merek dan Tipe Menggunakan Algoritma YOLO," in *Seminar Nasional Teknologi Informasi dan Ilmu Komputer*, pp. 43–52, 2023.

[10]. M. A. Syaputra, "Implementasi Algoritma YOLO Dalam Pengklasifikasian Objek Transportasi pada Lalu Lintas Kota Medan," *Populer: Jurnal Penelitian Mahasiswa*, pp. 13–23, 2023.

[11]. I. H. Al Amin, "Implementasi Algoritma YOLO dan Tesseract OCR pada Sistem Deteksi Plat Nomor Otomatis," *Jurnal Teknologi dan Sistem Komputer*, p. 54, 2022.

[12]. T. Abuzairi, "Implementasi Convolutional Neural Network untuk Deteksi Nyeri Bayi melalui Citra Wajah dengan YOLO," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 3, pp. 624–630, 2021.

[13]. M. D. R. Pratama, "Deteksi Objek Kecelakaan pada Kendaraan Roda Empat," *Jurnal Informatika dan Rekayasa Perangkat Lunak*, vol. 4, no. 1, pp. 15–26, 2022.

[14]. M. Y. Efendi, "Implementasi Klasifikasi Jenis Kendaraan di Indonesia Menggunakan YOLO," in *Seminar Nasional Teknik Elektro, Sistem Informasi dan Teknik Informatika (SNESTIK)*, pp. 219–224, 2021.

[15]. "Implementasi Algoritma Deep Learning untuk Sistem Deteksi Kantuk pada Pengemudi Menggunakan YOLO," in *Seminar Nasional Sistem Informasi Indonesia*, pp. 399–405, 2021.

[16]. B. Yanto, B. Basorudin, J. Jufri, B. H. Hayadi, and N. S. T. E. Armita, "Indentifikasi Pola Aksara Arab Melayu Dengan Jaringan Syaraf Tiruan Convolutional Neural Network (CNN)," *Journal Scientific and Applied Informatics*, vol. 3, no. 3, pp. 106–114, 2020.

[17]. B. Yanto, L. Fimawahib, A. Supriyanto, B. H. Hayadi, and R. R. Pratama, "Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna dengan Metode Deep

Learning Convolutional Neural Network," *Jurnal Inovtek Polbeng Seri Informatika*, vol. 6, no. 2, pp. 259–268, 2021.

[18]. C. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.