A Multi-Stage Deep Learning Framework for Real-Time Seatbelt Detection in Traffic Surveillance Videos

Salah Alghyaline
Department of Computer Science, the World Islamic Sciences and Education University, Amman, Jordan

***Abstract*—Seatbelt non-compliance is a major road safety concern. Automated road traffic monitoring using surveillance cameras can support large-scale detection. This study builds a multi-stage deep learning framework for real-time seatbelt detection in roadside traffic surveillance videos. YOLOv8s detects vehicle windshields and defines the region of interest; then it detects occupants within that region. ByteTrack then tracks occupants across consecutive video frames. The ResNet18 model classifies each occupant as wearing or not wearing a seatbelt in each frame. To improve the reliability of frame-level predictions, confidence scores are temporally aggregated for each tracked occupant, and high-confidence occupant crops are selected using a top-k strategy. This study constructs a dataset that combines publicly available seatbelt-related datasets (eight sources) with 1,200 locally collected and annotated images from different areas in Jordan. The dataset includes 9,516 windshields, 5,058 occupants, 3,043 seatbelt-on, and 1,695 seatbelt-off instances. YOLOv8s achieved 98.7% mAP@50 and 96.8% mAP@50 for windshield and occupant detection, while ResNet18 achieved 96.73% classification accuracy. The complete temporal framework achieved 85.92% precision, 93.40% recall, and an F1-score of 89.51%. The core pipeline achieved an average processing rate of 34.04 frames per second. The complete application achieved 29.00 frames per second, including displaying the results on the screen and storing them on disk. These results demonstrate the feasibility of the proposed framework for real-time traffic surveillance applications.**

Keywords*—Deep learning; traffic surveillance; seatbelt detection; YOLO

I. Introduction

The number of cars has increased greatly since the 1950s, and the number of globally registered cars exceeded one billion in 2017 [1]. Road traffic crashes are a major cause of death worldwide, and a considerable proportion of vehicle fatalities is associated with non-use of seatbelts [2]. Wearing a seatbelt prevents the occupants from being thrown outside the vehicle in collisions, which greatly lowers the risk of severe injury or death. Many drivers and passengers do not wear seatbelts, which increases the risk of severe injuries in accident cases. According to some studies [3] on seatbelt prevalence during the years 2000 and 2020, the use of seatbelts was 43.94% for drivers and 38.47% among front-seat occupants. Moreover, the use of seatbelts is low in developing countries such as Africa, Asia, and the Middle East compared with developed countries such as the US and Europe. Implementing the seatbelt detection camera system improved road traffic accident outcomes. In terms of reducing mortality, reducing vehicle ejection, and improving injury-severity indicators [4].

To address this issue, automobile manufacturers developed mechanical or sensor-based technology to detect whether the occupants are wearing seatbelts or not. Recent research has focused on camera-based seatbelt detection inside vehicles [5][6], However, this is limited inside the vehicle and can be bypassed by the occupants; therefore, it is not applicable for large-scale road monitoring.

Deep learning has shown superior performance on real, complex images across many computer vision fields in terms of accuracy and speed [7][8][9]. These capabilities make deep learning a promising approach for automated seatbelt detection using roadside surveillance cameras. However, recognising seatbelts from roadside cameras is challenging because of variations in vehicle appearance, occlusion, background noise, sunlight effects, viewing angle, image quality, and weather conditions. Moreover, detection based on a single video frame may produce unstable predictions.

This study proposes a multi-stage deep learning framework for real-time seatbelt detection in traffic surveillance videos. The framework implements YOLOv8s for windshield and occupant detection, ByteTrack for tracking occupants across video frames, and ResNet18 for seatbelt classification. The framework uses a temporal, confidence-based strategy across consecutive frames to improve the stability of seatbelt detection decisions. In addition, the framework incorporates selective image storage to retain reliable occupant samples for potential dataset expansion. The study constructs a seatbelt detection dataset by combining images from eight different sources with locally collected images to build a large dataset.

II. Related Work

Chen et al. [10] extracted multi-scale features from the labelled windshield and seatbelt region using a CNN model. The CNN detection score, along with an SVM classifier, was used to produce the final detection result. They built a dataset of 10,000 images for vehicle and windshield detection. They trained the model on 4000 images for seat-on and seatbelt_off. They reported the results on 200 images. The model achieved an 87% seatbelt detection rate.

Hosseini & Fathi [11] proposed an automatic seatbelt detection system. They used YOLOv5s to detect the windshield. The windshield is then divided into two parts: the left for the

occupant and the right for the driver. ResNet34 with Temporal Pyramid Pooling (TPP) and Power Mean Transformation (PMT) detects seatbelt violations for the driver and the occupant. The dataset used contains 3599 images for windshield, 1975 for Occupancy, and 1563 for seatbelt. The system accuracy is as follows: windshield detection (99.7%), Vehicle occupancy (98.9%), and driver's seatbelt (99.7%).

Qiu et al. [12] proposed a method based on YOLOv7 for automatic seatbelt detection; they used a modified YOLOv7 version called GM-YOLOv7 to increase the accuracy by 3.8% and increase the detection speed by 20%. An RST-Net Classifier is used for driver seatbelt classification. The dataset was collected from the traffic department: 4729 for windshield, 3983 for Seatbelt. The model obtained 98.49% precision and an F1-score of 98.25%.

Gu et al. [13] used a Gated Bi-LSTM with part-to-whole attention to detect seatbelts. The model uses a diagonal sampling strategy to detect fine seatbelt details. The dataset includes 14,936 surveillance camera images. The model achieved 98.1% Precision.

Catur et al. [14] built an automatic seatbelt detection system using YOLO7 to detect the car, windshield, occupant, and seatbelt. The dataset includes 1203 images: Car (4489), Windshield (4034), Occupant (4050), Seatbelt (2086). The model reached 93.4% Precision.

Singh et al. [15] used a web camera to capture images from the street. The camera was on the roadside, two meters high and 30 degrees to the street. YOLO3 was applied to detect the car. Then, template matching with a genetic algorithm was used to detect the presence of a seatbelt.

Nkuzo et al. [16] used an in-car camera to capture driver images. They collected 880 images. They used data augmentation to increase the number of images to 1240. YOLOv7 was trained to detect two classes: "buckled" and "unbuckled". They reported a mean average precision of 99.6%.

Suryanto et al [17] proposed a system that trained a YOLOv5 model on an embedded system (Raspberry Pi 4) to monitor seatbelt wearing inside the car and send SMS notifications. They collected a dataset of 1,617 images from the Roboflow website under different lighting conditions. The system achieved 99.1% accuracy.

Most previous studies relied on single-image detection, in-cabin cameras, small datasets, and direct image classification without temporal data aggregation across frames. In contrast, the proposed system is based on windshield region-of-interest (ROI) detection, occupant detection, ByteTrack occupant tracking, ResNet18 classification, and occupant confidence aggregation across frames. This makes the system more applicable for traffic surveillance videos captured from roadside or bridge-mounted cameras. Table I presents a comparison between the proposed seatbelt framework and existing methods.

TABLE I. COMPARISON OF EXISTING AND PROPOSED SEATBELT DETECTION METHODS

Ref.	Camera View	Method	Dataset	Temporal Tracking	Evaluation Unit	Primary Reported Metric	Key Characteristics
Chen et al. [10]	Roadside	CNN + SVM	Windshield: 8,000; Seatbelt: 4,000	No	Image-level	CIR: 87%	Image-level seatbelt detection
Hosseini & Fathi [11]	Roadside	YOLOv5s + ResNet34	Windshield: 3,599; Occupancy: 1,975; Seatbelt: 1,563	Not specified	Image-level	Accuracy: 99.7%	Image-level classification
Qiu et al. [12]	Traffic	GM-YOLOv7 + RST-Net Classifier	Windshield: 4,729; Seatbelt: 3,983	No	Image-level	F1-score: 98.25%	No temporal aggregation
Proposed Method	Bridge video	YOLOv8s + ByteTrack + Temporal Confidence Aggregation + ResNet18	Windshield: 9,516; Occupant: 5,058; Seatbelt: 4,738; 17 videos	Yes	Tracked occupant	Track-level F1-score: 89.51%	Temporal aggregation across tracked occupants

III. MATERIALS AND METHODS

The surveillance videos were captured from a bridge located in the middle of the street to monitor moving vehicles. YOLOv8s was first used to detect the windshields of vehicles. The detected windshield was cropped to define the region of interest (ROI), while other vehicle parts were ignored. YOLOv8s was then applied within the ROI to detect the occupants. Each detected occupant was cropped and classified as seatbelt-on or seatbelt-off using ResNet18. ByteTrack was used to track the detected occupants across multiple frames. Confidence scores from consecutive frames were accumulated to reduce noise and improve detection accuracy. Finally, occupant images with high confidence and good quality were automatically saved for each track using a top-*k* selection strategy. Fig. 1 presents the main stages of the proposed framework.

Table II presents the computational performance of the proposed framework. Video 2 of 9748 frames and a length of 5:27 minutes was used for the evaluation. Windshield detection using YOLOv8s requires an average of 11.76 ms/frame, with the input video frame resized to 640 × 640. Occupant detection and tracking with ByteTrack require 13.61 ms per cropped occupant. Seatbelt classification using the ResNet18 model takes 4.98 ms, with an additional 0.15 ms for temporal aggregation.

The complete pipeline requires 34.49 ms/frame, including windshield detection, Occupant detection and tracking, seatbelt classification, temporal aggregation, visualisation of the cropped occupant on the screen, and saving up to k (k=20) high-quality cropped images for each predicted class. The full system without visualisation and disk saving requires 29.38 ms/frame. However, some frames or cropped ROIs contain no valid windshield or occupant, and some may contain multiple objects.

Fig. 1. Proposed temporal multi-stage framework for seatbelt detection.

TABLE II. STAGE-WISE LATENCY AND THROUGHPUT OF THE PROPOSED FRAMEWORK

Stage	Model	Input size	Mean latency	Processing rate
Windshield detection	YOLOv8s	640 × 640	11.76 ms/frame	85.02 frames/s
Occupant detection and tracking	YOLOv8s + ByteTrack	640 × 640	13.61 ms/ROI	73.48 ROIs/s
Seatbelt classification	ResNet18	224 × 224	4.98 ms/crop	200.87 crops/s
Temporal aggregation	Proposed temporal confirmation	—	0.15 ms/occupant	6,784.43 occupants/s
Core full system	Proposed framework	Original video frame	29.38 ms/frame	34.04 FPS
End-to-end application	Proposed framework + visualization and saving	Original video frame	34.49 ms/frame	29.00 FPS

A. Dataset Construction

The dataset used in this study was created from both locally gathered data and publicly available seatbelt-related datasets. Local data were gathered from 17 traffic surveillance videos taken with a smartphone camera in various locations in Jordan. We used four videos to evaluate the proposed framework, while the remaining 13 videos were used to create image samples for training the framework. The images collected in this study include 1,200 images with manually annotated objects, resulting in 2,655 windshield instances, 1,210 occupant instances, 377 seatbelt-off instances, and 575 seatbelt-on instances, as shown in Table IV. We captured these images with a smartphone camera from different areas in Jordan. After combining these images with publicly available datasets [11], [18]-[24], the final dataset contains 9,516 windshield instances, 5,058 occupant instances, 3,043 seatbelt-on instances, and 1,695 seatbelt-off instances, as shown in Table V.

TABLE III. SOURCES OF IMAGES USED TO CONSTRUCT THE COMBINED DATASET

Ref.	Source	Dataset / Project (Roboflow Universe)	Number of Images	Classes
[11]	Published research dataset	Seatbelt dataset used in Hosseini & Fathi (2023)	1563	with-seatbelt, without-seatbelt
[18]	Roboflow Universe	Seatbelt and Mobile Phone Detection	1116	mobile, seatbelt, windshield
[19]	Roboflow Universe	Only Seatbelt Object Detection	200	seatbelt, windshield
[20]	Roboflow Universe	Seatbelt-Vehicle Occupancy	2083	person, seatbelt
[21]	Roboflow Universe	Seatbelt Object Detection	1219	Windshield
[22]	Roboflow Universe	Seatbelt Detection	1776	seatbelt, person-seatbelt, person-noseatbelt
[23]	Roboflow Universe	BELT Object Detection	198	beltON, beltOFF
[24]	Roboflow Universe	Seatbelt Detection	477	vehicle, car windshield, seat-belt, un-seat-belt
	This study	Collected and annotated images	**1,200**	windshield, occupant, Seatbelt_on, Seatbelt_off

For the windshield detection, occupant detection, and seatbelt classification tasks, 80% of the images were used for training, while the remaining 20% were used for validation. Building a robust and accurate deep learning model requires a large and representative dataset; therefore, this study aimed to combine eight publicly available datasets with the collected dataset to build a large dataset. Table III shows the different

sources used to build the dataset. This study targets four classes (Windshield, Occupant, Seatbelt-On, Seatbelt-Off). As shown in the table, not all datasets include the four target classes, and some include additional classes that were not used in this study. The constructed dataset with images and annotations is available at: Dataset Repository[1].

TABLE IV. CLASS DISTRIBUTION OF ANNOTATED INSTANCES IN THE IMAGES COLLECTED IN THIS STUDY (1,200 IMAGES)

Class	Number of Instances
Windshield	2,655
Occupant	1,210
Seatbelt_Off	377
Seatbelt_On	575

TABLE V. NUMBER OF ANNOTATED INSTANCES IN THE FINAL COMBINED DATASET

Class	Number of Instances
Windshield	9,516
Occupant	5,058
Seatbelt_On	3,043
Seatbelt_Off	1,695

B. Windshield and Occupant Detection

Table VI shows the performance of YOLOv8n, YOLOv8s, and YOLOv8m models on vehicle windshield detection. YOLOv8s achieved the best Precision, Recall, and mAP@50 with Precision (96.5%), Recall (96.5%), and mAP@50 (98.7%), while YOLOv8m achieved the highest mAP@50–95 (74.9%).

Table VII presents the performance of the three YOLOv8 models on vehicle occupant detection. YOLOv8s achieves the best Precision (91.7%), Recall (94.9%), mAP@50 (96.8%), and mAP@50-95(62.2%).

TABLE VI. PERFORMANCE COMPARISON OF YOLOV8 MODELS FOR VEHICLE WINDSHIELD DETECTION

Model	Precision	Recall	mAP@50	mAP@50-95
YOLOv8n	0.96	0.96	0.986	0.73
YOLOv8s	**0.965**	**0.965**	**0.987**	0.745
YOLOv8m	0.96	0.959	0.985	**0.749**

TABLE VII. PERFORMANCE COMPARISON OF YOLOV8 MODELS FOR VEHICLE OCCUPANT DETECTION

Model	Precision	Recall	mAP@50	mAP@50-95
YOLOv8n	0.882	0.931	0.95	0.551
YOLOv8s	**0.917**	**0.949**	**0.968**	**0.622**
YOLOv8m	0.896	0.92	0.955	0.571

Table VIII compares the three versions of YOLOv8 in terms of number of layers, parameters, and GFLOPs (Giga Floating Point Operations). GFLOPs measures the amount of computation a neural network performs to process one image (forward pass). The GFLOPs for YOLOv8m is approximately 9.7 times higher than YOLOv8n.

TABLE VIII. YOLOV8 ARCHITECTURE AND COMPUTATIONAL COST

Model	Layers	Parameters (Millions)	GFLOPs
YOLOv8n	134	2.99	8.1
YOLOv8s	134	11.05	28.2
YOLOv8m	174	25.9	78.4

The confusion matrices in Fig. 2 and 3 show that the YOLOv8s model achieved high detection rates for both the windshield and occupant classes. A total of 1872 samples were correctly classified as Windshield, while only 29 samples were misclassified as background. Similarly, for occupant detection, 989 samples were correctly classified, with only 27 samples misclassified. However, both models produced some false positives, where background samples were incorrectly classified as target objects (166 for windshield and 114 for occupants).

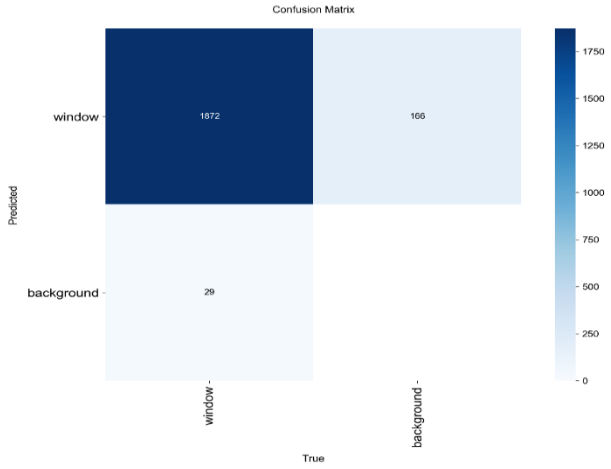

Fig. 2. Confusion matrix for Windshield detection with YOLOv8s.

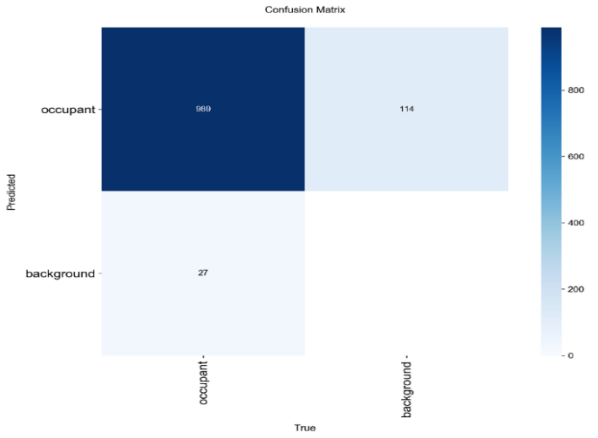

Fig. 3. Confusion matrix for occupant detection with YOLOv8s.

1 https://drive.google.com/drive/folders/1X125z4idio1u4FamkKfUvcbThO8r42Kg?usp=sharing

It is clear from Fig. 4 and 5 that the training and validation curves for the YOLOv8s model show stable convergence in both the windshield and occupant detection stages. The training losses include box loss, classification loss, and distribution focal loss (DFL). They consistently decreased across epochs, indicating effective learning and optimization. Similarly, the validation losses showed a gradual reduction, suggesting good generalization performance and limited overfitting. In addition, the precision, recall, and mAP metrics improved consistently throughout training. The windshield detection model achieved approximately 0.99 mAP@50 and 0.75 mAP@50–95, while the occupant detection model achieved around 0.97 mAP@50 and 0.62 mAP@50–95. These results demonstrated the validation for both windshield and occupant objects.

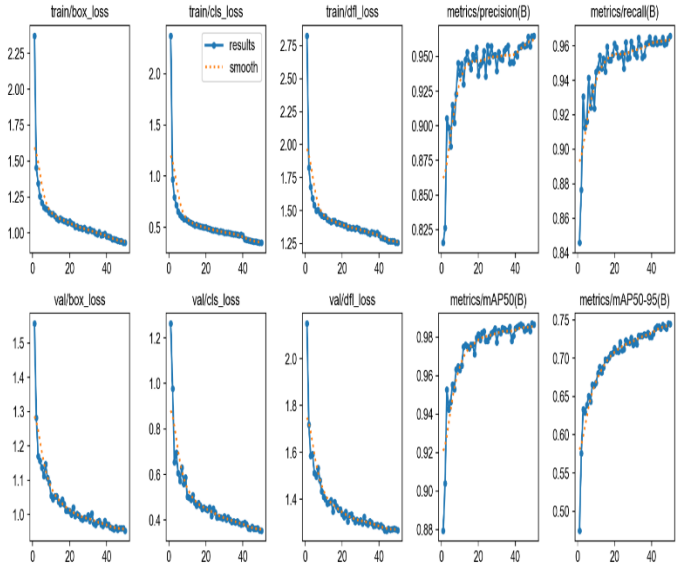

Fig. 4. Precision-recall curve of the best model windshield YOLOv8s.

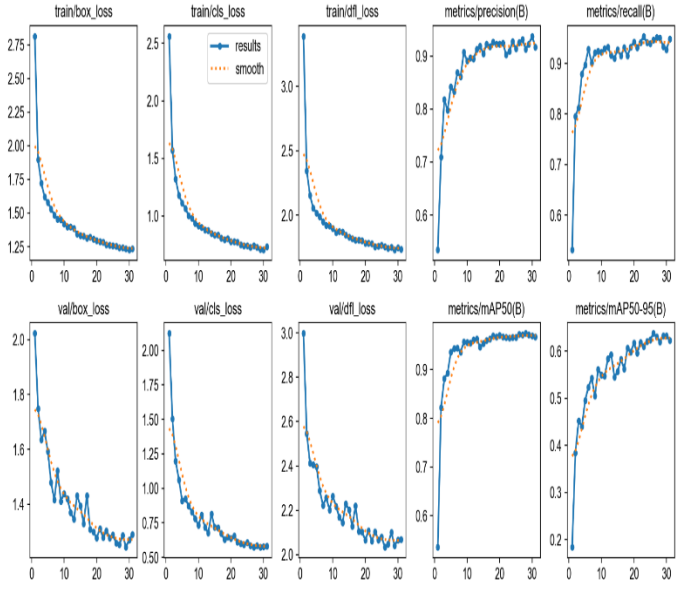

Fig. 5. Precision-recall curve of the best model for occupant YOLOv8s.

C. Seatbelt Classification

Five CNN models: ResNet18, ResNet34, MobileNetV2, DenseNet121, and EfficientNet-B0, were trained to detect the seatbelt. The models classify images into Seatbelt_on and seatbelt_off. The results in Table IX show that ResNet18 and ResNet34 achieved the highest accuracy, precision, recall, and F1-score. ResNet34 achieved a slightly higher Macro F1-score (96.44%) than ResNet18 (96.43%). Therefore, increasing model depth from ResNet18 to ResNet34 did not improve the main classification metrics. Lightweight models such as MobileNetV2 achieved the lowest accuracy (91.67%), while DenseNet121 and EfficientNet-B0 achieved 96.2% and 95.68%, respectively.

TABLE IX. COMPARATIVE EVALUATION OF SEATBELT DETECTION USING FIVE CNN MODELS

Model	Accuracy	Precision	Recall	F1-score	Macro F1	Params (Millions)
ResNet18	**0.9673**	**0.9673**	**0.9673**	**0.9673**	0.9643	11.7M
ResNet34	0.9673	0.9673	0.9673	0.9673	**0.9644**	21.8M
DenseNet121	0.9620	0.9621	0.9620	0.9621	0.9589	8.0M
EfficientNet-B0	0.9568	0.9567	0.9568	0.9567	0.9528	5.3M
MobileNetV2	0.9167	0.9168	0.9167	0.9167	0.9095	3.5M

D. Temporal Occupant Tracking and Decision Aggregation

Instead of using a single frame to make seat-belt decisions, the proposed system uses temporal information from consecutive frames. After detecting occupants within the windshield ROI, ByteTrack tracks the same occupant across frames. For each tracked occupant, the ResNet18 classifier produces two confidence scores corresponding to seatbelt-on and seatbelt-off. We evaluate frame-level predictions sequentially and count only high-confidence predictions for each track. In the initial experiments, we used five predictions with a confidence threshold of 0.80 for both classes. To reduce false-positive seatbelt-off detections, we subsequently increased the threshold and confirmation count for the seatbelt-off class to 0.95 and 15, respectively. The final configuration therefore requires five high-confidence seatbelt-on predictions at 0.80 or fifteen high-confidence seatbelt-off predictions at 0.95 to confirm the corresponding status. This temporal confirmation strategy helps reduce motion blur, partial occlusion, scale variation, illumination changes, and unstable frame-level predictions. After the final decision is obtained, the system saves only high-confidence and high-quality occupant crops for each track using a *top-k* selection strategy. This avoids saving blurred or uncertain samples and supports automatic dataset expansion for future model improvement.

Four videos were used to evaluate the performance of the proposed system and were completely excluded from the training datasets. The tracked occupants are classified as either seatbelt-on or seatbelt-off. Table X presents the length, FPS, and the classification result for each video. Table XI shows that the proposed video-level seatbelt detection model achieved overall precision, recall, and F1-score of 85.92%, 93.40%, and 89.51%,

respectively. The Seatbelt-on class demonstrated excellent performance, indicating that videos containing occupants wearing seatbelts were detected with high reliability, with a precision of 95.83%, recall of 92.93%, and an F1-score of 94.36%. In contrast, the Seatbelt-off class showed that the model successfully identified most videos in which occupants were not wearing seatbelts, achieving a high recall of 95.12%. However, because of the larger number of false positives (46), its precision was comparatively lower (62.90%), resulting in an F1-score of 75.71%. This suggests that while the model rarely misses actual seatbelt violations, it tends to incorrectly classify some seatbelt-on videos as seatbelt-off. Overall, the results indicate that the proposed method is highly effective at detecting seatbelt usage at the video level, making it suitable for traffic safety monitoring applications.

Fig. 6 and 7 present samples of correctly classified images extracted from the surveillance videos by the proposed framework. The videos were captured under different challenging conditions, including lighting, clutter, occlusion, and noisy backgrounds. Despite these challenging conditions, the proposed system detected the occupants and successfully classified their seatbelt status. Fig. 8 Presents examples of misclassified images.

TABLE X. DETECTION OUTCOMES FOR THE FOUR EVALUATION VIDEOS

Video	length	FPS	Seatbelt-on (TP/FP/FN)	Seatbelt-off (TP/FP/FN)
Video 1	5:37	30	83 / 6 / 3	28 / 15 / 1
Video 2	5:27	30	77 / 2 / 1	30 / 23 / 0
Video 3	3:00	30	36 / 2 / 7	12 / 3 / 1
Video 4	5:51	30	80 / 2 / 10	8 / 5 / 2

TABLE XI. VIDEO-LEVEL SEATBELT DETECTION PERFORMANCE

Class	TP	FP	FN	Precision (%)	Recall (%)	F1-score (%)
Seatbelt_On	276	12	21	95.83	92.93	94.36
Seatbelt_Off	78	46	4	62.90	95.12	75.71
Overall	354	58	25	85.92	93.40	89.51

Fig. 6. Examples of correctly classified Seatbelt_on images extracted from the surveillance videos by the proposed model.

Fig. 7. Examples of correctly classified Seatbelt images extracted from the surveillance videos by the proposed model.

Fig. 8. Examples of misclassified images extracted from the surveillance videos by the proposed model.

IV. CONCLUSION

This study proposed a temporal multi-stage deep learning framework for automatic seatbelt detection and classification in roadside traffic surveillance videos. We used YOLOv8s for windshield and occupant detection, ByteTrack for occupant tracking, and ResNet18 for seatbelt classification. To improve decision stability across successive frames, the framework uses a temporal, confidence-based confirmation strategy: it counts

high-confidence frame-level predictions for each tracked occupant before assigning the final seatbelt status. To train and evaluate the proposed temporal multi-stage framework, we used a dataset of publicly available images from eight sources, along with locally collected images from Jordan. The experimental results demonstrated the feasibility of the proposed framework. YOLOv8s demonstrated robust performance for both windshield and occupant detection, achieving 98.7% and 96.8% mAP@50, respectively. ResNet18 achieved 96.73% accuracy for seatbelt classification (seatbelt-on and seatbelt-off) and provided the best classification performance among the evaluated CNN models. At the track level, the complete framework achieved an overall precision of 85.92%, recall of 93.40%, and F1-score of 89.51%. The high recall of 95.12% for the Seatbelt-off class indicates the system can identify most seatbelt violations, but its lower precision of 62.90% shows that false-positive detections remain a key limitation. The core pipeline achieved an average processing rate of 34.04 FPS for full pipeline processing, while the complete application, including visualization and image saving, operated at 29.0 FPS. Despite these promising results, further evaluation is required under more diverse traffic, illumination, camera-angle, and occlusion conditions. Future work should also investigate improved temporal decision strategies and methods to reduce false positives in the Seatbelt-off class, as well as evaluation on larger, independently collected video datasets. These improvements could further enhance the framework's reliability and generalizability for practical traffic surveillance applications.

REFERENCES

[1] R. Jean-Paul, "Vehicle use indicators, world 1950–2022," The Geography of Transport Systems. [Online]. Available: https://transportgeography.org/contents/chapter1/what-is-transport-geography/world-vehicle-use-indicators/

[2] D. Febres, S. García-Herrero, S. Herrera, M. Gutiérrez, R. López-García, and A. Miguel, "Influence of seat-belt use on the severity of injury in traffic accidents," European Transport Research Review, vol. 12, no. 9, pp. 1–12, 2020, doi: 10.1186/s12544-020-0401-5.

[3] S. Kargar, A.-M. Alireza, and A. Hossein, "The prevalence of seat belt use among drivers and passengers: a systematic review and meta-analysis," Journal of the Egyptian Public Health Association, vol. 98, no. 1, p. 14, 2023, doi: 10.1186/s42506-023-00139-3.

[4] I. Al Babtain, A. Alabdulkarim, G. Alquwaiee, S. Alsuwaid, and E. Alrushid, "Outcomes of Road Traffic Accidents Before and After the Implementation of a Seat Belt Detection System : A Comparative Retrospective Study in Riyadh," Cureus, vol. 14, no. 7, pp. 6–13, 2022, doi: 10.7759/cureus.27298.

[5] A. Kashevnik, A. Ali, I. Lashkov, and N. Shilov, "Seat Belt Fastness Detection Based on Image Analysis from Vehicle In-Cabin Camera," in 2020 26th Conference of Open Innovations Association (FRUCT), Yaroslavl, Russia, 2020, pp. 143–150. doi: 10.23919/FRUCT48808.2020.9087474.

[6] S. Chun, J. B. Choi, C. W. Schwarz, J. G. Gaspar, D. V Mcgehee, and S. S. Baek, "NADS-Net: A Nimble Architecture for Driver and Seat Belt Detection via Convolutional Neural Networks," in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea, 2019. doi: 10.48550/arXiv.1910.03695.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.48550/arXiv.1506.01497.

[9] S. Alghyaline, "Real-time Jordanian license plate recognition using deep learning," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 6, pp. 2601–2609, 2022, doi: 10.1016/j.jksuci.2020.09.018.

[10] Y. Chen, G. Tao, H. Ren, X. Lin, and Z. Luming, "Accurate seat belt detection in road surveillance images based on CNN and SVM," Neurocomputing, vol. 274, pp. 80–87, 2018, doi: 10.1016/j.neucom.2016.06.098.

[11] S. Hosseini and A. Fathi, "Automatic detection of vehicle occupancy and driver ' s seat belt status using deep learning," Signal, Image and Video Processing, vol. 17, no. 2, pp. 491–499, 2023, doi: 10.1007/s11760-022-02244-w.

[12] L. Qiu, J. Rao, and X. Zhao, "Seatbelt Detection Algorithm Improved with Lightweight Approach and Attention Mechanism," Applied Sciences, vol. 14, no. 8, pp. 1–18, 2024, doi: 10.3390/app14083346.

[13] X. Gu, Z. Lu, J. Ren, and Q. Zhang, "Seat belt detection using gated Bi-LSTM with part-to-whole attention on diagonally sampled patches," Expert Systems With Applications, vol. 252, no. Part A, p. 123784, 2024, doi: 10.1016/j.eswa.2024.123784.

[14] R. Catur, E. Widodo, and K. Adi, "Real-time detection of seat belt usage in overhead traffic surveillance using YOLOv7," Applied Computer Science, vol. 21, no. 4, pp. 1–12, 2025, doi: 10.35784/acs_7594.

[15] L. Singh, D. Chaurasia, N. K. Tiwari, V. Upaddhyay, and A. Bajpai, "Paper Seat Belt Detection Using Genetic Algorithm-Based Template Matching," IEEJ Journal of Industry Applications, vol. 13, no. 1, pp. 91–97, 2024, doi: 10.1541/ieejjia.23006981.

[16] L. Nkuzo, M. Sibiya, and E. D. Markus, "A Comprehensive Analysis of Real-Time Car Safety Belt Detection Using the YOLOv7 Algorithm," Algorithms, vol. 16, no. 9, p. 400, 2023, doi: 10.3390/a16090400.

[17] A. Suryanto, D. H. Wicaksono, A. Mulwinda, M. Harlanu, and M. N. Syah, "Car seatbelt monitoring system using a real-time object detection algorithm under low-light and bright-light conditions," SINERGI, vol. 29, no. 3, pp. 771–778, 2025, doi: 10.22441/sinergi.2025.3.018.

[18] RoboFlow, "Seat Belt and Mobile Phone Detection Dataset," Roboflow Universe. Accessed: Dec. 15, 2025. [Online]. Available: https://universe.roboflow.com/aiactive20092009-gmail-com/seat_belt-and-mobile/dataset/2

[19] S. and W. Recognition, "Only Seatbelt Object Detection Dataset," Roboflow Universe. Accessed: Dec. 25, 2025. [Online]. Available: https://universe.roboflow.com/seatbelt-and-windshield-recognition/only-seatbelt

[20] Seatbeltdet, "Seatbelt-Vehicle Occupancy Dataset," Roboflow Universe. [Online]. Available: https://universe.roboflow.com/seatbeltdet/seatbelt-vehicleoccupancy/dataset/2

[21] Seg, "Seatbelt Object Detection Dataset," Roboflow Universe. Accessed: Dec. 20, 2025. [Online]. Available: https://universe.roboflow.com/seg-saicj/seatbelt-svdmb

[22] K. Panja, "Seat Belt Detection Dataset," Roboflow Universe. Accessed: Dec. 20, 2025. [Online]. Available: https://universe.roboflow.com/karan-panja/seat-belt-detection-uhqwa/dataset/1

[23] Optm, "BELT Object Detection Dataset," Roboflow Universe. Accessed: Dec. 20, 2025. [Online]. Available: https://universe.roboflow.com/optm/belt-dmrud/dataset/1

[24] Akaike, "Seatbelt Detection Dataset," Roboflow Universe. Accessed: Dec. 25, 2025. [Online]. Available: https://universe.roboflow.com/akaike/seatbelt-detection-vvyjz/dataset/1