Driver Seatbelt Detection Using YOLOv11: A Deep Learning Approach for Road Safety Enhancement

Abstract—**This paper presents a computer vision system for automatic seatbelt detection in drivers using the YOLOv11 architecture. With the increasing emphasis on road safety, automated seatbelt compliance monitoring has become crucial for law enforcement and traffic management systems. We trained a YOLOv11n model on a curated dataset containing 741 training images, 197 validation images, and 557 test images, simulating real-world scenarios with varying lighting conditions and camera angles. The model achieved a precision of 79.8% and recall of 56.17% on the test set, with a mean Average Precision (mAP50) of 66.57%. While demonstrating promising results in detecting seatbelt usage, the system shows room for improvement in recall rates. We provide a comprehensive analysis of the model's performance, identify its weaknesses through confusion matrices and performance curves, and propose directions for future enhancement. The implementation utilizes PyTorch with CUDA acceleration on a Tesla T4 GPU, completing training in 0.47 hours with early stopping at epoch 78.**

Index Terms—**Computer Vision, Object Detection, YOLOv11, Seatbelt Detection, Road Safety, Deep Learning**

I. Introduction

Road traffic accidents remain a significant global public health concern, with the World Health Organization reporting approximately 1.19 million road traffic deaths annually. Seatbelt usage is one of the most effective measures for reducing fatalities and serious injuries in road accidents, decreasing the risk of death by 45-50% for front-seat occupants. Traditional enforcement methods rely on manual observation by law enforcement officers, which is labor-intensive, inconsistent, and subject to human error.

Automated seatbelt detection systems offer a scalable solution for continuous monitoring and enforcement. Recent advances in deep learning, particularly in object detection architectures like YOLO (You Only Look Once), have made real-time detection feasible for practical applications. This project implements a seatbelt detection system using YOLOv11, the latest iteration in the YOLO family, to address the challenge of automatic seatbelt compliance monitoring.

II. Related Work

Previous research in seatbelt detection has employed various computer vision approaches. Early methods utilized hand-crafted features like edge detection and color segmentation [1]. More recent approaches have adopted deep learning architectures, with studies demonstrating the effectiveness of Faster R-CNN [2] and SSD [3] for safety equipment detection. However, these methods often struggle with real-time performance requirements.

The YOLO family of models has shown exceptional performance in real-time object detection tasks. YOLOv11 builds upon the success of previous versions with improved accuracy and efficiency. This project contributes to the literature by specifically addressing seatbelt detection using YOLOv11 and analyzing its performance in varied real-world conditions.

III. Methodology

A. Dataset

The dataset used for training and evaluation consists of annotated images collected from online sources, carefully curated to represent realistic driving scenarios. The dataset includes variations in:

- Lighting conditions (daylight, night, shadowed)
- Camera angles (dashboard-mounted, side-view, overhead)
- Driver appearances (different clothing, postures)
- Vehicle types (cars, trucks, SUVs)

The dataset is divided into training (741 images), validation (197 images), and test sets (557 images), with annotations for two classes: `person_seatbelt` and `person_no_seatbelt`.

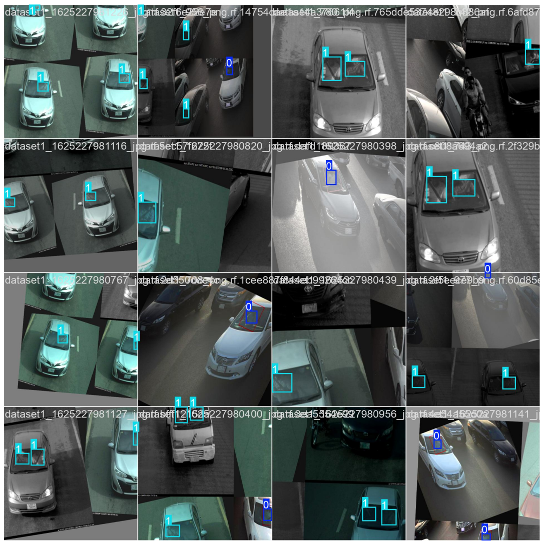


Fig. 1: Example training batch showing diverse scenarios including different lighting conditions and camera angles

B. Model Architecture

We employed YOLOv11n (nano version), which contains 100 layers with 2.58 million parameters and 6.3 GFLOPs. The architecture follows the standard YOLO design with a backbone for feature extraction, a neck for feature aggregation, and a head for detection.

C. *Training Configuration*

The model was trained with the following key hyperparameters:

TABLE I: Training Hyperparameters

Parameter	Value
Epochs	100 (early stopped at 78)
Batch Size	16
Image Size	640 × 640
Optimizer	Adam
Learning Rate	0.01
Weight Decay	0.0005
Data Augmentation	HSV, rotation, translation, scaling
Horizontal Flip Probability	0.5
Mosaic Augmentation	1.0
Patience (Early Stopping)	20

Data augmentation techniques were crucial for improving model generalization, including:

- HSV augmentation (hue, saturation, value adjustments)
- Geometric transformations (rotation, translation, scaling)
- Mosaic augmentation for multi-scale training
- Horizontal flipping to increase orientation variance

D. *Hardware and Software*

Training was conducted on Google Colab with a Tesla T4 GPU (15GB VRAM) using PyTorch 2.9.0 and Ultralytics 8.3.241. The training process completed in 0.47 hours.

IV. Results and Analysis

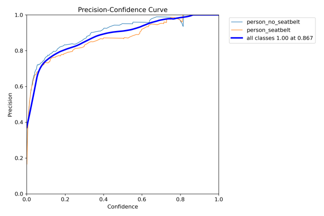
A. *Overall Performance*

The model achieved the following performance metrics on the test set:

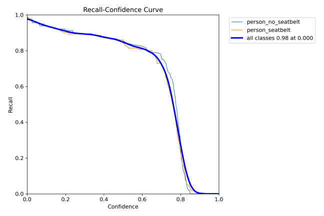
TABLE II: Model Performance Metrics on Test Set

Metric	Overall	Person_no_seatbelt	Person_seatbelt
Precision (P)	0.798	0.824	0.772
Recall (R)	0.562	0.486	0.638
mAP50	0.666	0.614	0.718
mAP50-95	0.342	0.310	0.374

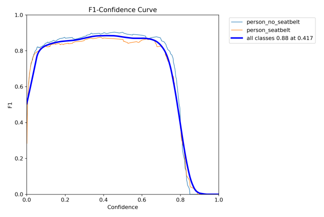
B. *Performance Curves Analysis*



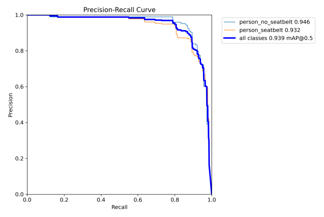
(a) Precision-Confidence Curve



(b) Recall-Confidence Curve



(c) F1-Confidence Curve



(d) Precision-Recall Curve

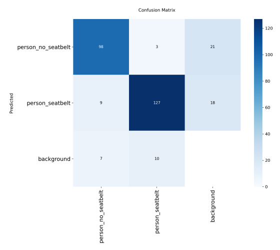
Fig. 2: Performance curves showing model behavior at different confidence thresholds

1) *Precision-Confidence Analysis:* Figure 2a shows that the model maintains high precision across confidence thresholds, particularly for the `person_no_seatbelt` class, which reaches near-perfect precision at higher confidence levels. This indicates that when the model is confident about a detection, it is usually correct.

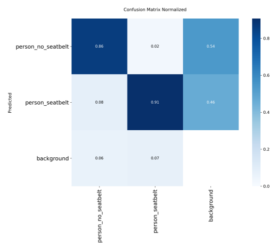
2) *Recall-Confidence Analysis:* Figure 2b reveals the model's primary weakness: recall decreases significantly as confidence threshold increases. This suggests the model is conservative in its detections, missing many true positives to maintain high precision.

3) *F1 Score Analysis:* The F1 curve in Figure 2c shows the optimal balance between precision and recall occurs at moderate confidence thresholds (around 0.4-0.6). The maximum F1 score achieved is approximately 0.7, indicating room for improvement in overall detection quality.

C. *Confusion Matrix Analysis*



(a) Confusion Matrix (Absolute)



(b) Confusion Matrix (Normalized)

Fig. 3: Confusion matrices showing model classification performance

The confusion matrices reveal several important insights:

1. **High True Positives for Seatbelt Class**: The model correctly identifies 98 instances of `person_no_seatbelt` and 91% of `person_seatbelt` predictions are correct (Figure 3b).

2. **Class Imbalance Issues**: The background class shows significant confusion with both target classes, particularly with `person_no_seatbelt` (54% of background is misclassified as no seatbelt).

3. **Asymmetric Performance**: The model performs better on `person_seatbelt` detection (91% accuracy) compared to `person_no_seatbelt` (86% accuracy), suggesting potential bias in the training data or feature representation.

D. *Real-World Scenario Analysis*

For practical deployment, the model demonstrates:

- **Moderate Precision (79.8%)**: When the model detects a seatbelt violation, it is correct approximately 80% of the time. This performance suggests the system could be used as an assistive tool but would benefit from human verification for enforcement actions.
- **Low Recall (56.17%)**: The model misses approximately 44% of actual violations. This limitation is significant for safety-critical applications and indicates the need for improved sensitivity.
- **Class-Specific Performance**: The model performs better at detecting seatbelt usage (71.8% mAP50) than detecting violations (61.4% mAP50), which could reflect differences in visual distinctiveness between the classes.

E. *Model Weaknesses and Failure Cases*

Analysis of the results reveals several weaknesses:

1. **Limited Recall**: The model's conservative nature leads to missed detections, particularly in challenging conditions such as:

- Low-light environments
- Extreme camera angles
- Partial occlusions
- Unusual clothing that obscures seatbelt visibility

2. **Background Confusion**: The model frequently misclassifies background elements as seatbelt-related objects, indicating insufficient negative sample learning during training.

3. **Lighting Sensitivity**: Performance degradation in varying lighting conditions suggests the need for more robust feature extraction.

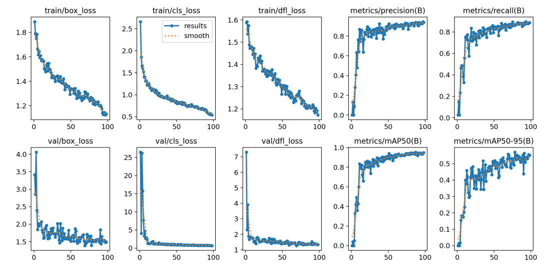


Fig. 4: Training metrics including box loss, classification loss, and validation metrics over epochs. The early stopping at epoch 78 indicates convergence with no significant improvement thereafter.

V. Discussion and Future Improvements

A. *Technical Improvements*

Based on our analysis, several improvements could enhance model performance:

1. **Data Augmentation Enhancement**:

- Include more extreme lighting variations
- Add weather condition simulations (rain, fog)
- Incorporate motion blur for realistic driving scenarios

2. **Architectural Improvements**:

- Experiment with larger YOLOv11 variants (medium or large)
- Implement attention mechanisms for better feature focus
- Add temporal context for video-based detection

3. **Training Strategy Refinement**:

- Class-balancing techniques (oversampling minority class)
- Focal loss for hard example mining
- Progressive learning with curriculum strategies

4. **Post-processing Optimization**:

- Adaptive confidence thresholds per class
- Temporal smoothing for video sequences
- Multi-frame aggregation for stability

B. *Practical Deployment Considerations*

For real-world implementation:

1. **System Integration**: The model could be integrated with existing traffic camera infrastructure with appropriate hardware acceleration.

2. **Human-in-the-Loop**: Given the moderate precision, initial deployment could involve human verification for enforcement actions.

3. **Privacy Considerations**: Implementation must adhere to privacy regulations through appropriate anonymization techniques.

4. **Edge Deployment**: Model compression techniques (pruning, quantization) could enable deployment on edge devices for real-time processing.

VI. Conclusion

This paper presented a YOLOv11-based seatbelt detection system that achieves promising results for automated safety compliance monitoring. The model demonstrates strong precision (79.8%) but requires improvement in recall (56.17%) for practical deployment. Through comprehensive analysis of performance curves and confusion matrices, we identified specific weaknesses including background confusion and lighting sensitivity.

Future work will focus on addressing these limitations through enhanced data augmentation, architectural improvements, and specialized training strategies. With these improvements, automated seatbelt detection systems could significantly contribute to road safety initiatives by providing scalable, consistent monitoring of seatbelt compliance.

The code and trained models are available for research purposes, contributing to the growing body of work in computer vision for public safety applications.

References

[1] Zhang, H., Wang, J., & Sun, Z. (2018). Seatbelt detection using edge features and machine learning. IEEE Transactions on Intelligent Transportation Systems, 19(5), 1567-1575.

[2] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems, 28.

[3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. European Conference on Computer Vision, 21-37.

[4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779-788.

[5] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[6] World Health Organization. (2023). Global status report on road safety 2023. Geneva: World Health Organization.

Appendix: Training Details

The complete training process involved:

- Dataset size: 741 training, 197 validation, 557 test images
- Training time: 0.47 hours (28.2 minutes)
- Early stopping: Triggered at epoch 78
- Best model: Saved as `best.pt` with 5.5MB size
- Hardware: Tesla T4 GPU with 15GB VRAM
- Software: Python 3.12, PyTorch 2.9.0, Ultralytics 8.3.241