Submitted: 2025-04-18 | Revised: 2025-06-01 | Accepted: 2025-06-15

Keywords: seat belt detection, YOLOv7, object detection, traffic surveillance, deep learning

Robert Catur Edi WIDODO [1*], *Kusworo ADI* [1], *Priyono PRIYONO* [1], *Aji SETIAWAN* [2]

[1*] Diponegoro University, Indonesia, caturediwidodo@lecturer.undip.ac.id, kusworoadi@lecturer.undip.ac.id, priyono@lecturer.undip.ac.id
[2] Darma Persada University, Indonesia, aji_setiawan@ft.unsada.ac.id
[*] Corresponding author: caturediwidodo@lecturer.undip.ac.id

Real-time detection of seat belt usage in overhead traffic surveillance using YOLOv7

Abstract

Driving safety plays a critical role in minimizing traffic accidents, and seat belt usage is one of the most effective preventive measures. This study aims to implement the YOLOv7 object detection model to automatically detect seat belt usage in four-wheeled vehicles using overhead traffic surveillance images. The proposed method consists of three main stages: dataset preparation, model training, and model evaluation. Dataset preparation includes acquiring video footage from different locations and time conditions, extracting image frames, and annotating four object classes: car, windshield, passenger, and seat belt. The model is trained on a dataset consisting of images taken during both day and night conditions. During training, data augmentation and anchor box optimization are applied to improve model generalization. The trained model is evaluated on an unseen test dataset and achieves a Mean Average Precision at 50% Intersection over Union (mAP50) of 97.46% and an F1 score of 95.37% at the optimal confidence level. These results indicate high detection accuracy for all object classes, especially for the seat belt class with an AP of 93.40%. The proposed system offers a promising solution for real-time traffic enforcement, reducing the reliance on manual observation and potentially improving traffic safety monitoring.

1. INTRODUCTION

Driving safety is paramount and must be maintained by every driver. Consistently practicing safe driving behaviors can reduce traffic problems, including congestion and accidents. Driving safety is determined by three factors: the condition of the vehicle, the traffic infrastructure, and the condition of the driver (Castellà & Pérez, 2004). Ninety percent of accidents are primarily influenced by the negative attitudes of drivers (Lee, 2005). A common negative behavior among drivers in Indonesia is not wearing seat belts. Currently, efforts to ensure driver compliance with seat belt laws involve the deployment of patrol officers at various locations and the installation of cameras that are then manually analyzed. This method is highly inefficient and costly. The advancement of computer vision technology makes autonomous traffic violation monitoring through image identification inevitable. The application of computer vision to seat belt usage detection has been extensively researched in the past. Zhou (2017) integrated edge detection, salient gradient mapping, and radial basis function (RBF) techniques into a singular network architecture to detect the presence of a seat belt in an image, achieving an average accuracy of 84.3%. Guo (2011) used a comparable edge detection technique to identify seat belts from traffic surveillance cameras with an accuracy of 81%. Elihos (2018) presented a technique to detect seatbelts by first localizing the vehicle's windshield, and then identifying the passenger using a single shot detector (SSD). The localization method generates a region of interest (ROI) represented as a passenger image. Then, the passenger image is processed by an object recognition model using SSD along with various image classification models, specifically CNN-P, VGG16, and Fisher Vector, to determine seat belt usage. The experiment showed that the SSD model achieved the highest accuracy and precision, specifically 91.9% and 94.5%, respectively, compared to the three image classification methods. According to previous research results, the use of object detection is more effective in identifying seat belt usage than the use of image classification techniques. In the deep learning paradigm, feature extraction from images is performed automatically during the learning process (Deng & Yu, 2013).

Object detection or recognition is a challenge in computer vision that seeks to recognize and localize a semantic object of a specific class within an image (Dasiopoulou et al., 2005). In contrast to the classification process that assigns a single image to a certain class, the object recognition method can identify multiple classes of objects inside a single image (Mohialden et al., 2024). Object detection is typically performed with machine learning or deep learning techniques. The machine learning approach commences by delineating features by diverse strategies, including the utilization of the Histogram of Gradients (HOG) feature (Dalal & Triggs, 2005), the Haar feature (Viola & Jones, 2001), or the Scale-invariant feature transform (Lowe, 1999). Subsequently, classification is performed utilizing a machine learning model, such as the Support Vector Machine (SVM). Simultaneously, the deep learning approach employs artificial neural networks for object detection, eliminating the necessity for explicit feature specification (Girshick et al., 2014).

Deep learning object identification typically consists of two stages: localization and classification. The localization algorithm identifies segments of the image that may contain objects (region proposals), and then classifies each segment individually (Jiao et al., 2019). Girshick (2015) presented the RCNN detector model, which uses a selective search method to generate region recommendations. RCNN achieves an accuracy rate of 53.7% on the PASCAL VOC dataset. However, RCNN has a significant drawback: its lengthy recognition time. RCNN requires 47 seconds to process a single image, making it impractical for real-time applications such as video processing. As a result, the Fast-RCNN algorithm has been proposed, which achieves comparable accuracy to RCNN while reducing the detection time to 0.32 seconds per image. The Faster RCNN technique uses a streamlined network architecture to generate region proposals, enabling it to process up to 17 images per second and facilitating its application in real-time processing (Ren et al., 2017).

There is also an object detector that requires only a single stage of processing, called a one-stage detector. This detector localizes and predicts object classes from a single image using a single CNN network, without the need for compiling region proposals. The entire computational process of this detector is performed by a single network. As a result, the single-stage detector has a simpler architectural model that allows for fast detection. An example of this model type is the single-shot detector (SSD), which employs the VGG16 architecture (Simonyan & Zisserman, 2015) as the backbone network, followed by multiple convolutional layers of decreasing size. SSD achieves a mean average precision (mAP) of 74.3% at a speed of 59 frames per second (W. Liu et al., 2016).

You Only Look Once (YOLO) is a single-step object detection technique using a convolutional neural network (CNN). YOLO is designed for real-time, end-to-end training while maintaining high accuracy. The first iteration of YOLO has a modified GoogleNet as its backbone network (Szegedy et al., 2015). YOLO can process at a rate of 45 frames per second with a mean average accuracy of 63.4% (Redmon et al., 2016). The YOLO architecture works by partitioning the input image into an S×S grid. The position of the object is identified based on the grid cell containing the center of its bounding box. The grid cell encompassing the center of the bounding box is tasked with object detection. Each grid cell predicts a bounding box characterized by parameters of C+5*B, where C is the number of classes and B is the number of projected bounding boxes. The B value is multiplied by 5 because it includes the location and dimensions of the bounding box, along with the confidence value (x, y, w, h, c) for each bounding box. Since an image contains S×S grid cells, the overall prediction of the model is represented as a tensor with dimensions S×S×(C+B*5). The first iteration of YOLO predicts the bounding box coordinates using a Fully Connected Layer (FCN) after processing the feature extractor. However, this method was later modified in the second version of YOLO (YOLO9000) by replacing the FCN with an anchor box. An anchor box is a preliminary bounding box with specified dimensions (p_w, p_h) distributed over the image region. This anchor box serves as a specialized predictor for objects with specific characteristics, such as dimensions, aspect ratio, and special placements. The bounding box of an object is established by predicting the variance from the size of the anchor box (Redmon & Farhadi, 2017).

The YOLO architecture consists of three components. The first segment consists of a backbone network, a CNN, that analyzes each pixel in the image to generate features with varying levels of detail. This backbone network is often trained on a classification dataset. The second component is the neck network, which is a feature pyramid network (FPN) that integrates the representations of the CNN layers prior to their processing by the prediction network. The final component is the head network, which acts as a prediction network to predict bounding boxes and object classes. The predictions generated by this network are driven by YOLO's three loss functions: class, box, and objectness. Figure 1 shows the YOLO architecture.

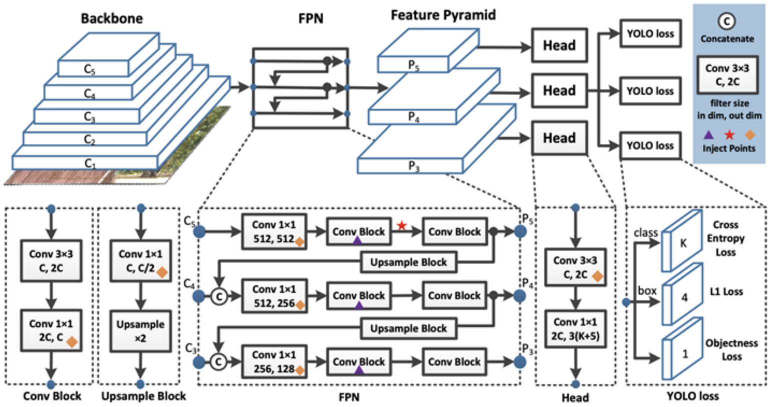


Fig. 1. YOLO network architecture (S. Liu et al., 2021)

YOLO remains the preeminent framework for real-time object recognition and continues to evolve. YOLOv7 represents the latest version of the YOLO series at the time of this writing and will be the model used in this investigation. YOLOv7 is designed to produce more accurate forecasts at a speed comparable to its predecessor. To achieve this goal, several modifications have been made to the YOLOv7 design, including the integration of E-ELAN (extended efficient layer aggregation network) into the backbone network, model scaling, re-parameterization, and the inclusion of auxiliary heads in the center of the network (Wang et al., 2023).

The latest version of YOLO (You Only Look Once), YOLOv12, is a significant advancement over YOLOv7. YOLOv7, introduced in mid-2002, is characterized by exceptional speed and high accuracy in real-time object detection. This version improves inference performance without excessive computational overhead by incorporating several advances, including E-ELAN, efficient head models, and re-parameterization approaches (Wang et al., 2023). Following YOLOv7, the community and Ultralytics developed YOLOv8 through YOLOv12, with incremental improvements in precision, attention-driven design, and multitasking versatility, including segmentation and posture estimation. In this study, YOLOv7 was used based on an assessment of available computational resources and appropriate data attributes. YOLOv7 provides an optimal balance between accuracy and efficiency, demonstrating sufficient stability and reliability for this research setting without requiring extensive infrastructure (Zhu & Miao, 2024).

This research aims to implement a CNN-based object detection system to detect seat belt usage in traffic surveillance images. This research uses a methodology similar to that of Elihos (2018), with modifications to the detection model employed specifically using the YOLOv7 model (Wang et al., 2023). This study involves three main mechanisms. The three stages include image dataset preparation, model training, and model evaluation. This research has used recorded video rather than performing recognition in a fully real-time environment. Although the system does not operate in true real-time, we expect that the use of video will approximate real-time conditions. However, due to limitations in processing resources, our existing configuration is not yet able to perform continuous real-time inference during live surveillance.

2. MATERIALS AND METHODS

2.1. Dataset

The dataset preparation procedure involves the acquisition of raw overhead video footage depicting traffic conditions, followed by image extraction, image annotation, windshield region of interest (ROI) extraction, ROI annotation, and separation of the images into training, validation, and test datasets. The photographs used in this research were taken with a digital camera positioned approximately 10 meters above the roadway, which has a 20°-30° slope. The camera is positioned directly above the roadway to ensure recording conditions as shown in Figure 2. The photos were taken at resolutions of 1920×1080 pixels and 3840×2160 pixels.

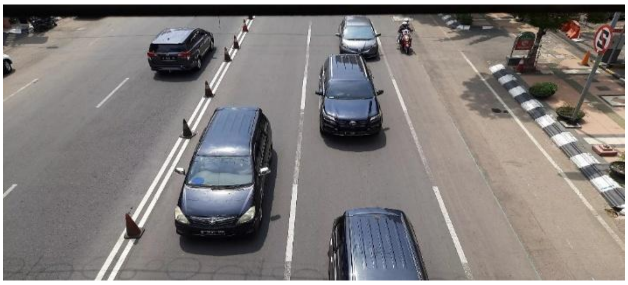


Fig. 2. Example of direct capture images

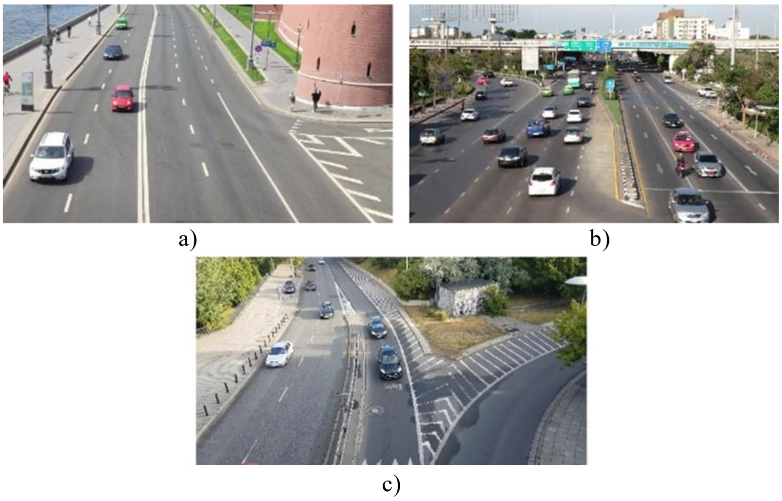
Additional image sources were obtained from the Internet to increase the diversity of the dataset. The first source is an aerial view of traffic on the Bolshoy Krasnokholmsky Bridge in Russia (DZ Computer Vision, 2021). The second source is an aerial view of traffic at an intersection in Thailand (Panasonic Connect Europe, 2015). The third source is an aerial view of traffic at an intersection in Poland (Majek, 2018). All three media were downloaded at a resolution of 3840×2160 pixels.

The next stage is to extract frames from each media file. Frames are extracted automatically at 10-second intervals. The extracted frame images are stored in a single directory. Table 1 provides details on the number of frame extraction images for each media source.

Tab. 1. Number of frame extraction images

Media	Frame Size	Amount
Indonesia	3840×2160	670
Russia	3840×2160	186
Thailand	3840×2160	116
Poland	3840×2160	231
Total	-	1203

Figure 3 shows examples of retrieved photos from each media outlet.



a) b)

c)

Fig. 3. Examples of images sourced from the internet located in (a) Russia, (b) Thailand, and (c) Poland

The next step is to annotate the image. The annotation procedure tries to provide information through class names and the spatial coordinates of objects through bounding boxes for recognition purposes. The frame image contains two categories of annotated objects: cars and car windshields, referred to as car and windshield classes, respectively. The annotation procedure is performed on each frame image containing both objects using the open-source software LabelMe and is stored in the YOLO (You Only Look Once) annotation format. The process of annotating a frame image is illustrated in Figure 4.

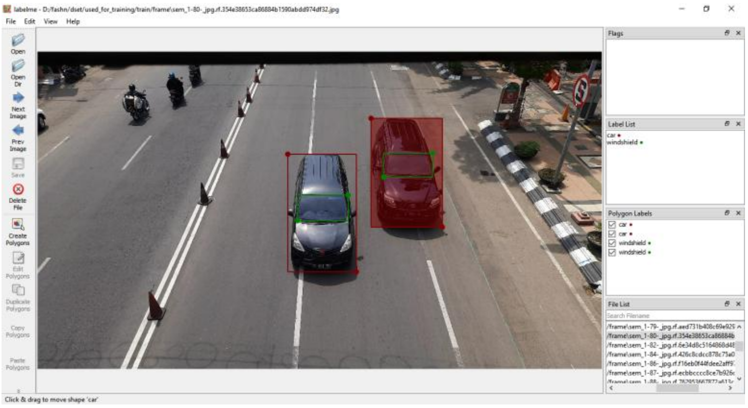

labelme - D:/fashn/dset/used_for_training/train/frame\sem_1-80-_jpg.rf.354e38653ca86884b1590abdd974df32.jpg
File
Edit
View
Help
Open
Open Dir
Next Image
Prev Image
Save
Delete File
Create Polygons
Edit Polygons
Duplicate Polygons
Copy Polygons
Paste Polygons
Flags
Label List
car
windshield
Polygon Labels
☑ car
☑ car
☑ windshield
☑ windshield
File List
Search Filename
/frame\sem_1-79-_jpg.rf.aed731b408c69e929
/frame\sem_1-80-_jpg.rf.354e38653ca86884b
/frame\sem_1-82-_jpg.rf.6e34d8c5164868d48
/frame\sem_1-84-_jpg.rf.426c8cdcc878c75a0
/frame\sem_1-86-_jpg.rf.f16eb0f44fdee2aff9
/frame\sem_1-87-_jpg.rf.ecbbcccc8ce7b926
Click & drag to move shape 'car'

Fig. 4. Frame image annotation process

The final phase involves extracting the windshield region of interest from each frame image. The extraction procedure is performed using the crop operation on each bounding box of the windshield class and then saved as a new file. Table 2 shows the details of the number of image files for windshield ROI extraction.

Tab. 2. Number of windshield ROI extraction images

Media	ROI windshield
Indonesia	2341
Russia	139
Thailand	39
Poland	441
Total	2960

Then, annotation was performed on the ROI images for the passenger and seatbelt objects, called passenger and seatbelt classes, respectively. Figure 5 illustrates the annotation process for the extracted windshield ROI images.

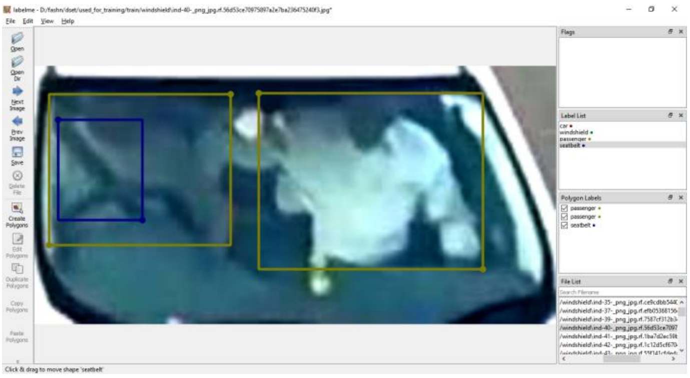

labelme - D:/fashn/dset/used_for_training/train/windshield\ind-40-_png_jpg.rf.56d53ce70975897a2e7ba236475240f3.jpg*
File
Edit
View
Help
Open
Open Dir
Next Image
Prev Image
Save
Delete File
Create Polygons
Edit Polygons
Duplicate Polygons
Copy Polygons
Paste Polygons
Flags
Label List
car
windshield
passenger
seatbelt
Polygon Labels
☑ passenger
☑ passenger
☑ seatbelt
File List
Search Filename
/windshield\ind-35-_png_jpg.rf.ce9cdbb544(
/windshield\ind-37-_png_jpg.rf.efb05388156
/windshield\ind-39-_png_jpg.rf.7587cf312b3
/windshield\ind-40-_png_jpg.rf.56d53ce7097
/windshield\ind-41-_png_jpg.rf.1ba7d2ac58t
/windshield\ind-42-_png_jpg.rf.1c12d5cf670
Click & drag to move shape 'seatbelt'

Fig. 5. Annotation process windshield image

The ROI windshield image is then merged with the frame image into a single directory. The final phase involves partitioning the dataset into a training dataset, a validation dataset, and a test dataset, with the specifics regarding the number of images in each dataset and the amount of annotations for each class detailed in Table 3.

Tab. 3. Number of images in the training, validation, and test datasets and the number of label annotations for each class

Dataset	Amount	Label			
		Car	Windshield	Passenger	Seatbelt
Training	2137	2981	2764	2434	1212
Validation	913	425	366	978	369
Testing	726	1083	904	638	505

2.2. Training model

The training model seeks to train the deep learning model by providing a dataset with annotations as input, enabling the model to identify the patterns and attributes of each class that underpin its predictive capabilities. This research uses the YOLOv7 architectural model (Wang et al., 2023). During the training process, the training image dataset is preprocessed by scaling the images to 640×640 pixels and using data augmentation through mosaics to mitigate overfitting. The anchor box is then optimized for the dataset using a genetic algorithm. The training uses the P5 model hyperparameter, yielding prediction outputs for P3 (step 8, small objects), P4 (step 16, medium objects), and P5 (step 32, large objects) within the Feature Pyramid Network (FPN). The model is trained for 100 epochs using a batch size of 16. During each epoch, the model is trained to minimize the loss function and is evaluated by computing the mean average precision (mAP) at an intersection over union (IOU) threshold of 50%, as well as the mean mAP over the IOU threshold range of 50% to 95%. Fitness is computed as a model evaluation metric represented by the average weighted value of the two mAP values, as shown in equation (1). The model from the era with the optimal fitness value is used in the next step. The entire training procedure is performed using the training program provided in the YOLOv7 repository (WongKinYiu, 2023)

$$fitness = 0.1 \times mAP_{50} + 0.9 \times mAP \tag{1}$$

Where: mAP – mean average precision.

2.3. Testing model

Object recognition model testing is used to assess the ability of the trained model to identify objects. The testing procedure uses a test dataset, which is a collection of photographs that the model has never seen before. The test dataset used in this study is divided into two time segments: daytime and nighttime. The result of this model evaluation phase is represented by the Mean Average Precision (mAP) value and the accuracy level of the model.

Testing begins by feeding the test dataset into the model to obtain detection results, which include object classes, bounding boxes, and confidence levels for each object identified. In addition, each detection bounding box is compared to the ground truth bounding box, which serves as a label or annotation within the test dataset. The procedure calculates the Intersection over Union (IOU) with a threshold of 50%. A detection bounding box with an IOU value above the threshold is classified as a true positive (TP), while a detection bounding box with an IOU below the threshold is classified as a false positive (FP). An undetected ground truth bounding box is classified as a false negative (FN).

In the next phase, the recall and precision metrics are computed for each class at different confidence level thresholds τ(k). The average precision (AP) is calculated as the area under the precision-recall (PR) curve for each class. The calculation of mAP was then performed. The F1 score metric, which represents the harmonic mean of precision and recall, is used to evaluate model accuracy and is calculated using equation (2).

$$F1(\tau) = \frac{2TP(\tau)}{nbox_{det}(\tau) + nbox_{gt}} \tag{2}$$

Where: TP – true positive,

τ – threshold.

This metric is chosen because the size of the true negative is insignificant in the object detection domain. The negative signal in the dataset image is indicated by any pixel that is not bounded by a bounding box, resulting in numerous potential negative bounding box permutations within a single image (Lin et al., 2020). Furthermore, there is a disparity in the annotation of the used dataset, namely within the seatbelt class, which has a significantly lower amount of annotations compared to other classes, as shown in Table 3. Similar to other metrics, the F1 score is computed at different confidence levels τ(k), where TP(τ) represents the number of true positives, nboxdet(τ) denotes the number of detection bounding boxes, and nboxgt indicates the number of ground truth bounding boxes. The results of the calculation are presented as an F1-score curve plotted against the confidence level threshold, along with the maximum F1-score value.

3. RESULTS AND DISCUSSION

3.1. Results of the training model

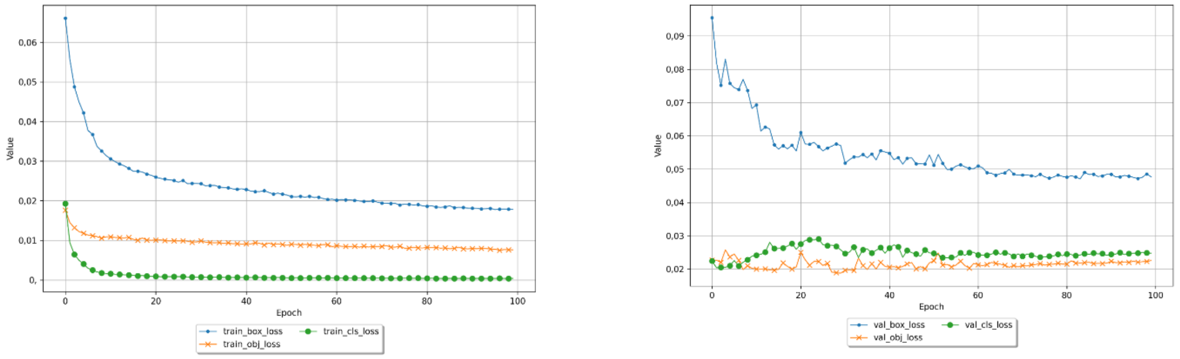
Model training was performed using the YOLOv7 model. The training and model testing process was performed using the Google Colaboratory service (https://colab.research.google.com) in the form of a virtual machine with specifications of an Intel(R) Xeon(R) CPU @ 2.20GHz processor, 12GB RAM, 16GB NVIDIA Tesla T4 graphics card, and Ubuntu 22.04.2 LTS 64bit as the operating system.

The model was trained on a 640×640 pixel image dataset for 100 epochs. Each epoch was evaluated using the mean average precision (mAP) measure at an intersection over union (IOU) threshold of 50% (mAP50) and the mean mAP over an IOU threshold range of 50% to 95% (mAP). The fitness value, represented by the average weight of the two metrics, is used to identify the epoch with optimal results. The training method yields an optimal model with mAP50 and mAP values of 94.59% and 67.17%, respectively, and a fitness value of 69.91%. The training outcome metrics for the optimal model are shown in Table 4.

Tab. 4. Best model training result metrics

No	Metrics	Value (%)
1	AP car class	96.93
2	AP windshield class	96.92
3	AP passenger class	96.35
4	AP seatbelt class	88.16
5	Precision	90.15
6	Recall	90.89
7	mAP50	94.59
8	mAP	67.17
9	Fitness	69.91

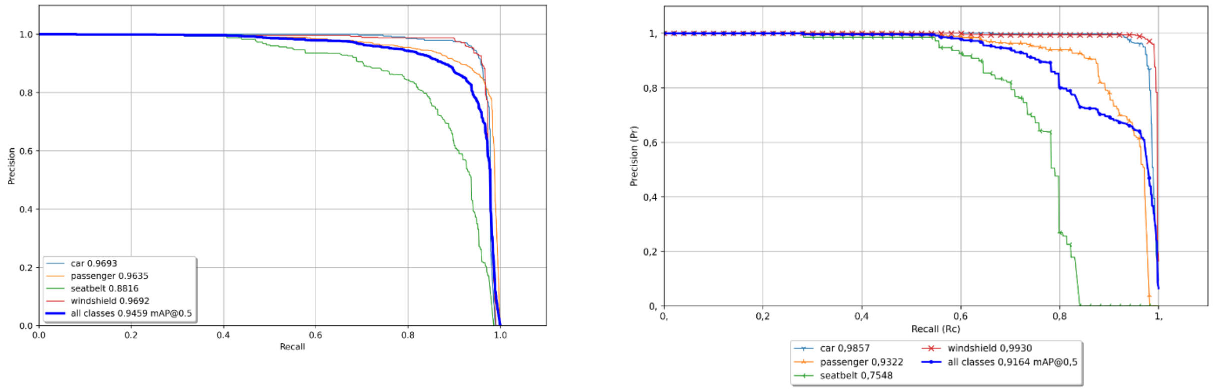
The analysis of Figures 6(a) and 6(b) shows that the model shows no signs of overfitting and has effectively converged to a minimum loss value. The first graph illustrates the loss throughout the training process and shows a steady reduction in the three loss components-train_box_loss, train_cls_loss, and train_obj_loss-as the number of epochs progresses. This reduction indicates that the model is successfully and consistently learning patterns from the training data. The second graph, showing the loss on the validation data, shows that the loss of the third component shows a very consistent trend without a significant increase, indicating that the model maintains effective generalization on untrained data. The model shows no signs of overfitting, which is typically characterized by an increase in the validation loss while the training loss continues to decrease. Moreover, in both the training and validation data sets, all loss components converge to relatively minimal values, indicating that the model has reached a stable state in the optimization process. This indicates that the model not only learns effectively from the training data, but also exhibits consistent performance on the validation data, a critical feature of a robust, non-overfitting model.



(a) **(b)**

Fig. 6. Training Loss (a) and Validation Loss (b) Graph

Figure 7 shows the precision-recall curve of the optimal daytime (a) and nighttime (b) model.



(a) **(b)**

Fig. 7. Precision-recall curve of the best model daytime (a) and nighttime (b)

The results of the training model, as shown in Figure 7(a) and (b), demonstrate that the model effectively detects all four object types in the validation dataset. The object classes identified by the model, ranked by the highest Average Precision (AP) values, are as follows: car class with an AP of 96.93%, windshield class with an AP of 96.92%, passenger class with an AP of 96.35%, and seat belt class with an AP of 88.16%.

3.2. Results of testing model

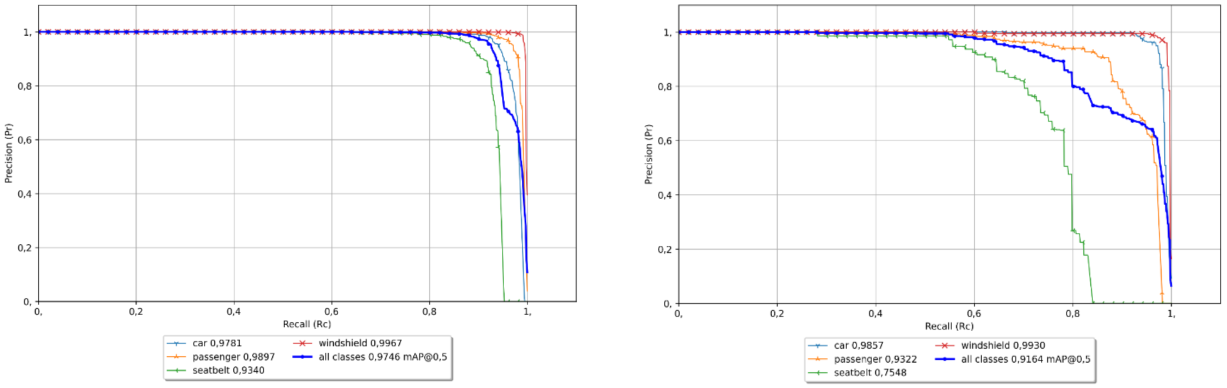
The trained model was evaluated on a test dataset containing images missing from the training and validation datasets. The evaluation was performed to test the model's ability to generalize object recognition. The model was evaluated on a test dataset of 726 images and 3130 label annotations, which served as ground truth for all classes. The model predicted 5,768 bounding boxes, including all four object classes, using the test dataset. The model processes a single image in 8.2 milliseconds, or 122 frames per second. The results of the precision, recall, and mAP50 calculations for daytime and nighttime conditions are shown in Table 5.

Tab. 5. Results of mAP50 calculations for daytime and nighttime conditions

Class	Number of True Labels	Number of Predictions	Pre (%)	Rec (%)	mAP50 (%)
Daytime					
All	3130	5768	97.37	93.58	97.46
car	1083	2172	98.50	91.51	97.81
windshield	904	1778	98.89	98.90	99.67
passenger	638	973	94.99	97.18	98.97
seatbelt	505	845	97.10	86.73	93.40
Nighttime					
All	1160	4819	92.43	86.15	91.64
car	384	1960	91.83	97.66	98.57
windshield	367	1615	94.46	99.18	99.30
passenger	285	663	92.68	84.45	93.22
seatbelt	124	581	90.75	63.31	75.48

Calculation of the evaluation metrics yields AP50 values for each class and mAP50 models with elevated values that exceed the validation threshold. The resulting mAP50 value is 97.46%. The windshield class has the highest AP50 at 99.67%, followed by the passenger class at 98.97%, the car class at 97.81%, and finally the seatbelt class at 93.40%. The AP50 and mAP50 measures took into account precision and recall at different confidence levels during the calculation process. Consequently, the results of this calculation indicate that the model has robust and consistent performance in object detection across different confidence levels.

Overall, the nighttime test resulted in a mAP50 value of 91.64%, which was lower than the validation mAP50 value and the mAP50 value of the daytime test results. The class with the highest AP50 was the windshield class at 99.30%, slightly lower than the daytime test results, followed by the car class at 98.97%, slightly higher than the daytime test results, then the passenger class at 93.22%, lower than the daytime test results, and finally the seat belt class at 75.48%, much lower than the daytime test results. Based on these results, it can be concluded that the model can still detect relevant objects at various confidence levels quite consistently in low-light conditions, although it experiences a decrease in performance in detecting the seatbelt class. The PR curve for day and night is shown in Figure 8(a) and 8(b).



(a) **(b)**

Fig. 8. Precision-recall curve of testing in daytime (a) and nighttime (b) conditions

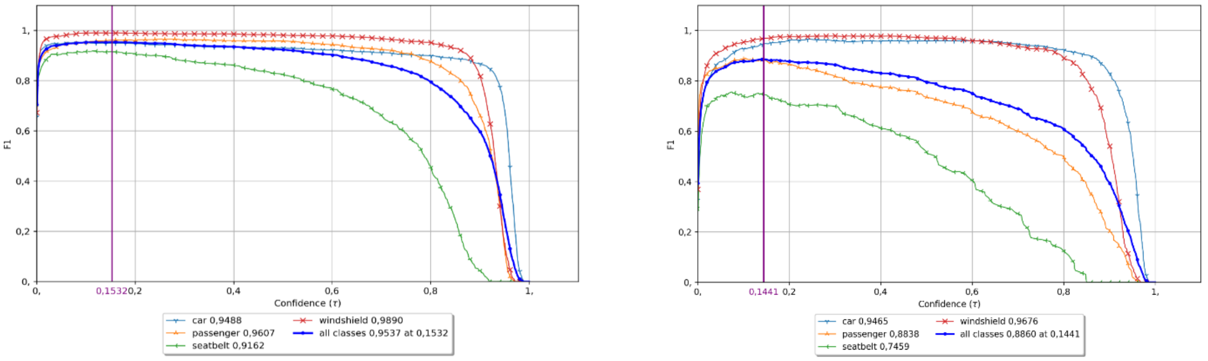
The F1 score is then derived from the precision and recall data to measure the accuracy of the model. The calculation yielded the optimal value at the confidence level threshold τ_best = 0.1532, equivalent to 95.37%, and τ_best = 0.1441, equivalent to 88.60%, as shown in Table 7, with the F1 score vs. confidence curve shown in Figure 9.

Tab. 6. Results of mAP50 calculations for daytime and nighttime conditions

Class	Number of True Labels	Number of Prediction (τ > τ_best)	TP	FP	FN	F1-score (%)
Daytime						
All	3130	3013	2943	70	187	95.37
Car	1083	1006	991	15	92	94.88
Windshield	904	904	894	10	10	98.90
Passenger	638	652	620	32	18	96.07
Seatbelt	505	451	438	13	67	91.62
Nighttime						
All	1160	1138	1057	81	103	88.60
Car	384	408	375	33	9	94.65
Windshield	367	385	364	21	3	96.76
Passenger	285	259	240	19	45	88.38
Seatbelt	124	86	78	8	46	74.59

Note : TP = True positive, FP = False Positive, FN = False Negative

The F1 score value for each class at the τ_best threshold in order from the largest to the smallest is the windshield class at 98.90%, the passenger class at 96.07%, the car class at 94.88%, and the seat belt class at 91.62%. In Table 6, it is mentioned that TP is a prediction that matches the class, FP is a non-class that is projected as a class, and FN is a class that is predicted as a non-class. Details of the accuracy calculation results are shown in Table 7, and the F1 score vs. confidence curve can be seen in Figure 9.



(a) **(b)**

Fig. 9. F1-score curve daytime (a) and nighttime (b) against confidence level

Using the resulting τ_best value, filtering is performed on the detection result bounding box with a confidence level less than τ_best. Then, the number of TP, FP and FN hits for each class with a confidence level threshold of τ_best is calculated and given in the form of a confusion matrix in Table 7. From these tables, the mAP is 97.46% in daytime and 91.64% in nighttime.

Tab. 7. Confusion matrix testing with τ_best=0.1532 (daytime) and τ_best=0.1441 (nighttime)

		True Label				
		car	**windshield**	**passenger**	**seatbelt**	**background (FP)**
τ_best=0.1532		Daytime				
Prediction	car	991	0	0	0	15
	windshield	0	894	0	0	10
	passenger	0	0	620	0	32
	seatbelt	0	0	0	438	13
	background (FN)	92	10	18	67	n/a
τ_best=0.1441		Nighttime				
Prediction	car	375	0	0	0	33
	windshield	0	364	0	0	21
	passenger	0	0	240	0	19
	seatbelt	0	0	0	78	8
	background (FN)	9	3	45	46	n/a

4. CONCLUSION

The research results indicate that the YOLOv7 model effectively detects seat belt use (seat belt class) in conjunction with other objects, including four-wheeled vehicles (car class), vehicle windshields (windshield class), and drivers and passengers (passenger class). It achieved a mAP5000 value of 97.46% during the day and 91.64% at night. The AP value for the belt category is 93.40%, dropping to 75.48% at night. The optimal accuracy, evaluated by the F1 score metric, is achieved at a confidence level threshold of τ_best = 0.1532, equivalent to 95.37%, and τ_best = 0.1441, equivalent to 88.60%.

Based on the completed research, the following recommendations can be implemented and further refined. Expand the dataset by integrating a larger number of images with different attributes, such as location, camera position and angle, lighting conditions, dimensions, and image resolution. Diversity in the dataset improves the model's ability to generalize object detection, reduces dataset bias, and avoids overfitting. Use near-infrared (NIR) cameras to capture images of the dataset in low-light conditions and to run detection systems. Light is an essential physical component in all computer vision systems, as RGB digital cameras are unable to collect the necessary information in low-light conditions, such as at night. The model can be augmented or combined with complementary detection systems, including road marking violation detection, mobile phone use while driving, helmet compliance for two-wheelers, and vehicle identification systems such as automatic license plate recognition.

Conflicts of Interest

The authors declare no conflict of interest.

Funding

This study was funded by APBN of the Faculty of Science and Mathematics, Diponegoro University (Grant No.: 25.VI.L/UN7.F8/PP/II/2024).

REFERENCES

Castellà, J., & Pérez, J. (2004). Sensitivity to punishment and sensitivity to reward and traffic violations. *Accident Analysis and Prevention*, *36*(6), 947-952. https://doi.org/10.1016/j.aap.2003.10.003

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (pp. 886-893). IEEE. https://doi.org/10.1109/CVPR.2005.177

Dasiopoulou, S., Mezaris, V., Kompatsiaris, I., Papastathis, V. K., & Strintzis, M. G. (2005). Knowledge-assisted semantic video object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *15*(10). https://doi.org/10.1109/TCSVT.2005.854238

Deng, L., & Yu, D. (2013). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, *7*(3–4), 197-387. http://dx.doi.org/10.1561/2000000039

DZ Computer Vision. (2021, August 15). Traffic count, monitoring with computer vision. 4K, UHD, HD [Video]. YouTube. https://youtu.be/2kYpqSMqrzg

Elihos, A., Alkan, B., Balci, B., & Artan, Y. (2018). Comparison of image classification and object detection for passenger seat belt violation detection using NIR RGB surveillance camera images. *2018 15th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)* (pp. 1-6). IEEE. https://doi.org/10.1109/AVSS.2018.8639447

Girshick, R. (2015). Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)* (pp. 1440–1448). IEEE. https://doi.org/10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 580-587). IEEE. https://doi.org/10.1109/CVPR.2014.81

Guo, H., Lin, H., Zhang, S., & Li, S. (2011). Image-based seat belt detection. *2011 IEEE International Conference on Vehicular Electronics and Safety* (pp. 161–164). IEEE. https://doi.org/10.1109/ICVES.2011.5983807

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., & Qu, R. (2019). A survey of deep learning-based object detection. *IEEE Access*, *7*, 128837-128868. https://doi.org/10.1109/ACCESS.2019.2939201

Lee, J. D. (2005). Driving safety. *Reviews of Human Factors and Ergonomics*, *1*(1), 172–218. https://doi.org/10.1518/155723405783703037

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(2). https://doi.org/10.1109/TPAMI.2018.2858826

Liu, S., Peng, Y., & Liu, L. (2021). A novel ship detection method in remote sensing images via effective and efficient PP-YOLO. *IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC 2021)* (pp. 234-239). IEEE. https://doi.org/10.1109/SDPC52933.2021.9563569

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot MultiBox detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016* (Vol. 9905, pp. 21–37). Springer International Publishing. https://doi.org/10.1007/978-3-319-46448-0_2

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *IEEE International Conference on Computer Vision* (pp. 1150-1157). IEEE. https://doi.org/10.1109/iccv.1999.790410

Majek, K. (2018, July 16). 4K Road traffic video for object detection and tracking - free download now! [Video]. YouTube. https://youtu.be/MNn9qKG2UFI

Mohialden, Y. M., Kadhim, R. W., Hussien, N. M., & Hussain, S. A. K. (2024). Top Python-based deep learning packages: A comprehensive review. *International Journal Papier Advance and Scientific Review*, *5*(1). https://doi.org/10.47667/ijpasr.v5i1.283

Panasonic Connect Europe. (2015, November 10). 5.5 4K Camera Road in Thailand No 2 [Video]. YouTube. https://youtu.be/F4bICvLY024

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)* (pp. 6517-6525). IEEE. https://doi.org/10.1109/CVPR.2017.690

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, real-time object detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 779-788). IEEE. https://doi.org/10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *ArXiv, abs/1409.1556*. https://doi.org/10.48550/arXiv.1409.1556

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1-9). IEEE. https://doi.org/10.1109/CVPR.2015.7298594

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)* (pp. 1-1). IEEE. https://doi.org/10.1109/cvpr.2001.990517

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7464-7475). IEEE. https://doi.org/10.1109/cvpr52729.2023.00721

WongKinYiu. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors [GitHub repository]. GitHub. https://github.com/WongKinYiu/yolov7

Zhou, B., Chen, L., Tian, J., & Peng, Z. (2017). Learning-based seat belt detection in image using salient gradient. *12th IEEE Conference on Industrial Electronics and Applications (ICIEA 2017)* (pp. 547-550). IEEE. https://doi.org/10.1109/ICIEA.2017.8282904

Zhu, S., & Miao, M. (2024). Lightweight high-precision SAR ship detection method based on YOLOv7-LDS. *PLoS ONE*, *19*(2), e0296992. https://doi.org/10.1371/journal.pone.0296992