Multi-Emotion Features Extraction for Driver Distraction Detection

Sheng-Tao Chen[1], Chih-Hsien Hsia[2,3], Yo-Cheng He[2]*

[1] Department of Avionics Engineering, Republic of China Air Force Academy, Taiwan
[2] Department of Computer Science and Information Engineering, National Ilan University, Taiwan
[3] Department of Business Administration, Chaoyang University of Technology, Taiwan
iiccanffly@gmail.com, hsiach@niu.edu.tw, R1243005@ms.niu.edu.tw

Abstract

Two major factors contributing to human-induced traffic accidents are driver fatigue and distraction. To reduce the rate of occurrence of car accidents, some studies introduced systems that record biometric measurements, diver behavior, and driver's physical features while driving. Data obtained from these systems are used to predict the driver concentration states. However, these attention-detection systems face challenges in terms of applicability and accuracy in detecting negatively impacting driving behaviors. Contactless physical feature extraction and integration of deep learning are effective and applicable methods in this context. Therefore, this study proposes a method based on multi-feature processing in conjunction with you only look once (YOLO)-based object detection to classify driver attention. Experiments and validation were conducted using the open-source Yawn detection dataset (YawDD) and National Tsing Hua University drowsy driver detection dataset (NTHU-DDD). The proposed method not only outperforms those of certain studies and achieved high efficiency in detecting multi-emotion features of drivers and assisting driver attention.

Keywords: Deep learning, Object detection, Attention, Fatigue driving, Distracted driving

1 Introduction

In recent years, various measures to enhance driving safety performance have been continuously introduced. These include the global emphasis on autonomous vehicle automation and the development of a variety of driver-assistance systems designed to improve both driving comfort and safety. However, these initiatives aim to effectively reduce the occurrence rate of human-induced traffic accidents.

The national highway traffic safety administration (NHTSA) estimates that approximately 25% of accidents are related to driver inattention [1]. Table 1 illustrates the definitions of inattention and fatigue provided by NHTSA, the American automobile association foundation for traffic safety (AAA FTS), and the European traffic safety commission (ETSC) [2]. The classification of inattention by NHTSA encompasses factors such as driver distraction, drowsiness, mental lapses, and fatigue. Conversely, according to AAA FTS research, driver attentiveness is categorized into five states: attentive, distracted, looked but not seen, sleepy, and unknown. Within this classification, looked but not seen is considered a form of distraction, while sleepiness falls under the broader category of fatigue. Furthermore, the broader definition of fatigue sometimes includes symptoms of drowsiness and is associated with normal and significant factors such as physical exertion, emotional stress, or sleep deprivation. Fatigue detection has been extensively researched; however, there is still no unified definition. Therefore, ETSC [2] classified fatigue into four states: complete awake, moderate sleepiness, severe sleepiness, and asleep, to provide a more comprehensive categorization.

Table 1. Definition of driver status

Organization	Defined category	Driver status
NHTSA	Inattention	Driver distraction, drowsiness, mental lapses, fatigue
AAA FTS		Attentive, distracted, looked but not seen, sleepy, unknown
ETSC	Fatigue	Complete awake, moderate sleepiness, severe sleepiness, asleep

Table 2. Measurement methods for inattention or fatigue

Measurement	Illustration
Biometric measurements	Physiological signal measurement instruments are used to measure various biological information, including brain electrical activity in the form of electroencephalogram (EEG), to assess fatigue status. Additionally, measurements of heart rate, pulse wave frequency, respiratory rate, and other physiological parameters are also obtained.
Driver performance	Determination is made based on features such as steering direction deviation, changes in speed, and the average time-to-line crossing (the time it takes for a tire to move from leaving a lane boundary to contacting it again).
Physical characteristics	Monitoring the driver involves analyzing images, facial features (such as changes in the frequency of eye blinking or slower blink rates), yawning frequency, or head movements (such as nodding off).

When drivers experience distraction or fatigue, various observable characteristics may manifest, such as frequent yawning, impaired decision making, mood swings (e.g., sadness or irritability), slower reactions, difficulty in keeping eyes open, reduced ability to maintain focus, nodding or

*Corresponding Author: Chih-Hsien Hsia; E-mail: hsiach@niu.edu.tw
DOI: https://doi.org/10.70003/160792642024112506004

head bobbing, shallow breathing, and accelerated heart rate. The manifestation of these characteristics can vary between individuals. Therefore, finding specific and effective methods to measure fatigue levels is worth exploring. Table 2 outlines three measurement methods for assessing distraction or fatigue. Notably, in measuring driver behavior through physical characteristics, Liang *et al*. [3] found that distraction could be determined by monitoring the frequency and duration of a driver's gaze off the road. Furthermore, in a collaborative project between NHTSA and SAVE-IT, the duration of time when the driver's eyes are off the road and the time when the driver's head is oriented away from the road were identified as reliable indicators of the driver's visual distraction.

In terms of market application, a monitoring system called Autopilot has been developed by the self-driving car benchmark company, Tesla, in recent years. Driver behavior is monitored by a cabin-style camera installed inside the rearview mirror of the vehicle. Researchers at Tesla used data from user in-car cameras and classified driver behaviors such as those shown in Table 2. In contrast, we employed an attention-assistance system that uses behavioral analysis and converts it into numerical values, allowing for a clear distinction of the current state of the driver and providing corresponding alerts.

Interference in biometric measurements can jeopardize driving safety, and inter-driver variations in driving habits can introduce measurement inaccuracies. Therefore, this study proposes a driving assistance system that uses in-car installed cameras to monitor and detect the physical characteristics of the driver. When the system detects instances of distracted driving, the images can be promptly analyzed through a model and alerts can be immediately issued to reduce the likelihood of human-induced traffic accidents.

2 Related Work

In the context of drivers, apart from performing feature extraction on the facial region of drivers, their emotions also play a crucial role as significant features. The facial emotions of drivers can provide important insights into the driving process, such as detecting fatigue and stress. In this regard, some studies have employed different models and methods to classify the emotional states of drivers. Feature extraction from the facial region of images can be achieved using discrete wavelet transformation (DWT) [4-5], which converts the original input image into four sub-bands to retain critical facial information. Alternatively, feature extraction is performed directly from the original input image, and entropy analysis is used to identify significant facial regions. Finally, an image is processed in a zigzag pattern using discrete cosine transformation (DCT) to extract features with higher variance, followed by facial emotion classification using these features.

Detection of eye closure has also been used as a basis for detecting fatigued drivers [6]. This method employs a model architecture involving face detection, feature extraction, and classification. Among the feature-extraction techniques compared, which included canny edge, local binary patterns, histogram of oriented gradient, Gabor filter, and normal gray image, it was demonstrated that the best performance (accuracy, F1-score) was achieved when using the histogram of oriented gradient features. In a study of eye movements and eyelids [7], it was acknowledged that closures could sometimes be inaccurately predicted due to environmental or emotional factors. Therefore, six parameters—percentage of eye closure, eye closure duration, blink frequency, nod frequency, facial position, and fixed gaze direction—were used by some researchers [8] to measure the level of fatigue and distraction using a fuzzy classifier. Subsequently, optical flow was employed to detect eye-blink features, and an alert was triggered after a period of no blinking information.

Regarding fatigue detection, an analysis of the relationship between stress and gaze spatial distribution (GazeDis), as well as average eye closure speed (AECS) for detecting and tracking eyes, was conducted [9]. The findings revealed that stress is positively correlated with GazeDis and percentage of large pupil dilation (PerLPD), and negatively correlated with AECS. Furthermore, fatigue was detected through head movements [10], which were considered as one of the indicators of early-stage fatigue. Nodding was determined by calculating the head pitch angle and exponentially weighted moving variance (EWVAR). Further, the calculated correlations between the results and drowsiness showed that the correlations were relatively insignificant compared to other features.

The correlation between sleepiness and other features was found to be relatively small compared to other characteristics. Concerning yawning detection [11], a threshold-based segmentation algorithm was employed for yawning detection, achieving an accuracy of 76%. Regarding driver behavior, Eskandarian *et al*. [12] identified the following features as highly correlated with fatigue:

1) Reflexive nodding when drivers check the rearview mirror.

2) Significant reduction in head movements.

3) A substantial increase in the frequency of drivers touching their faces, chin, head, ears, eyes, or thighs.

4) Noticeable increase in the frequency of eye blinking.

Driver distraction [3] can be determined by examining the frequency and duration of the driver's gaze being away from the road. In a collaborative project between NHTSA and SAVE-IT, it was noted that both the duration of time that the eyes leave the road and the orientation of the head towards non-road areas are reliable indicators of the driver's visual distraction.

Two algorithms were previously developed based on eye information tracking [13]. One algorithm calculates the percent road center of the eyes. If the percent road center falls below 58% and accumulates for more than 1 minute, it is determined that the driver is in a distracted state. The other algorithm uses a 3D model that encompasses areas such as the dashboard, windshield, and rearview mirror. It counts instances where the driver's gaze direction leaves the driving field for more than two seconds, classifying it as a distracted state. The results indicate that both methods effectively identify distracted driving states. A method for detecting drowsiness using eyelid closure and head nodding features was proposed [11], which triggers a warning of reduced

driving attention when the driver exhibits eyelid closure in 40 out of the recent 60 captured images. Eyelid closure and head nodding are predicted using two finite-state machines to assess the current driver condition.

The advantages of this study [14] include addressing overfitting by reducing redundant information and employing the Pearson distance function to calculate classification loss, thereby effectively preventing overfitting. The results on the PASCAL VOC and FSOD datasets demonstrate outstanding performance and maintain stability across various scenarios. However, the study's reliance on specific datasets may limit the generalizability of the method, and it entails higher computational complexity as it primarily improves upon existing fine-tuning and metric learning methods. Kamruzzaman *et al*. [15] propose a blockchain as a service (BaaS) based deep facial feature extraction (DFFE) architecture for evaluating student attention in online education. The primary advantages of this approach are its ability to precisely assess student attention, emotions, and behaviors, coupled with high performance and efficiency. Nonetheless, the study's limitations include a restricted sample size and diversity, which affect the generalizability of the algorithm. Moreover, further enhancements are necessary to address specific challenges across different educational domains. In [16], a direct LL-mask band scheme was introduced for detecting and tracking moving objects using low-resolution images. Detecting moving objects in real environments is challenging due to noise from false motion, such as moving tree leaves. While many methods have been developed for controlled environments, DLLBS effectively reduces noise with low computing cost in both indoor and outdoor settings. Wang *et al*. [17] propose an optimized object detection technique called S2F-YOLO, specifically improved for fish classification. The model incorporates focal loss, effectively addressing sample imbalance issues, making it suitable for real-world applications requiring rapid detection. However, the model's size and computational resource demands may limit its use in resource-constrained environments. Although most electronic consumer terminals lack sufficient computing power for AI-based image matching algorithms, they possess identification capabilities that allow them to automatically recognize individuals or objects connected to the network. Additionally, they feature intelligent characteristics, enabling the network system to self-feedback and perform intelligent control.

Following the above methods and literature, wherein yawning frequency and eye-blink rate have been used as indicators of fatigue, this study also adopted yawning frequency and eye-blink rate as the standards for detecting fatigued driving.

3 Proposed Methodology

The focus-detection system of this study was primarily implemented using object detection [18], image-recognition techniques, and integrated multi-feature technology. We employed you only look once (YOLO)v7 to detect and label driver behaviors within the car environment, recognize facial expressions, and extract various facial features. These extracted features were used in the proposed decision-making mechanism to classify driver focus levels.

In contemporary computer vision, object detection frameworks like YOLO and region-based convolutional neural networks (R-CNN) are pivotal. YOLO is often chosen over R-CNN-based methods due to its superior speed and simplicity performance. A critical advantage of YOLO lies in its real-time performance. In contrast, R-CNN [19] involves multiple stages: region proposal, feature extraction, and classification, making it slower. Despite improvements in fast R-CNN and faster R-CNN, these methods remain slower than YOLO. YOLO's architecture is also significantly simpler and more integrated. It uses a unified approach, predicting bounding boxes and class probabilities in one evaluation, which simplifies both training and deployment. On the other hand, R-CNN requires separate training stages for different components, such as region proposals, bounding box regression, and classification. This multi-stage approach complicates implementation and tuning, making YOLO more straightforward.

In computational efficiency, YOLO is more resource-effective. It avoids the repetitive convolutions over proposed regions characteristic of R-CNN methods. Even with optimizations in faster R-CNN, these methods still demand substantial computational resources due to their multi-step process. While YOLO excels in speed and efficiency, it does involve a trade-off in accuracy, particularly in detecting smaller objects within dense scenes. However, newer versions like YOLOv7 have made significant strides in improving accuracy. Conversely, R-CNN methods are known for superior accuracy due to refined region proposals and detailed classification stages, making them preferable for accuracy-critical tasks such as medical imaging.

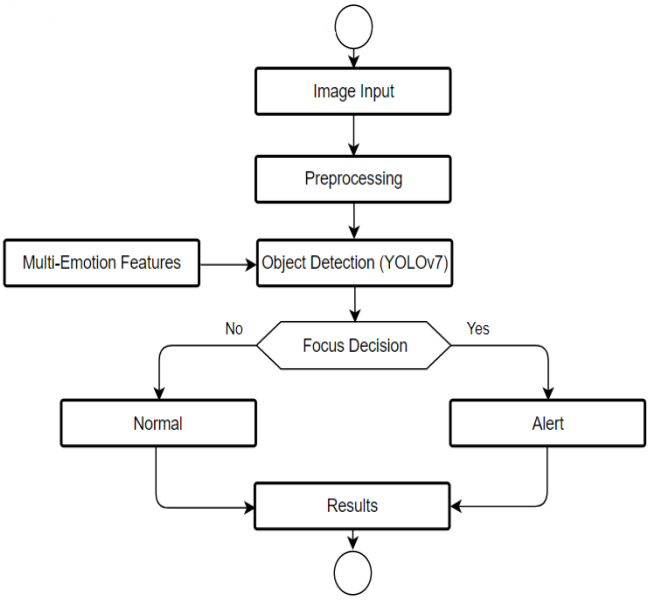


Figure 1. Focus-classification workflow

Figure 1 illustrates the focus-classification workflow proposed in this study. In this workflow, input image data are preprocessed and then fed into the multi-emotion features, and you only look once (YOLO)-based object-detection

method for focus decision making. Finally, the results are displayed, showing both normal and alert states as the outcome of focus classification. Detailed explanations of the multi-emotion feature and YOLO-based object-detection method is provided in the following sections. Through these two methods, the goal of focus-classification is achieved to enhance driver attention assistance.

3.1 Multi-Emotion Features

A driver emotion detection system necessitates real-time collection and analysis of extensive data, including physiological signals, facial expressions, and voice patterns of the driver. Through network communication, this data is transmitted in real-time to edge and cloud computing platforms for more accurate emotion assessment, with the results then relayed back to the internet of vehicles (IoV) system [20]. This approach not only enhances the accuracy of emotion detection but also leverages long-term data accumulation for personalized optimization, thereby improving the driving experience. In the future, the system could also share driver emotion state data with other vehicles. For instance, if a driver's emotional state indicates fatigue or stress, nearby vehicles can proactively respond by taking evasive actions or issuing alerts, thereby enhancing road safety and achieving intelligent traffic management.

According to the NHTSA definition of distracted driving, this study defined a distracted driving state as when a driver cannot maintain normal driving behavior or fails to maintain a focused eye gaze on the road ahead. However, based on the definition of fatigue states by AAA FTS, a driver is classified as fatigued when yawning and eye closure frequency exceed predefined thresholds. Additionally, we categorized the distracted state into three subcategories: normal (no yawning or high eye closure frequency), alert (yawning or high eye closure frequency), and fatigue (both yawning and high eye closure frequency).

This study used facial changes during driving as multiple emotion features, such as yawning and eye closure behaviors. This approach involved detecting the driver's facial features and implementing them using physical measurements. Furthermore, to enhance fatigue-detection accuracy, multiple features were fused, and YOLOv7 was used for image recognition.

3.2 YOLO-based Object-detection Method

YOLO is a one-stage object-detection method. YOLO needs to perform only one pass of convolutional neural network on an image to determine the positions and classes of objects within the image, which significantly improves the speed of object recognition. The convolutional neural network architecture of YOLO is mainly based on the GoogleNet model and consists of 24 convolutional layers and 2 fully connected layers. However, what sets YOLO apart from GoogleNet is that it uses 1×1 convolutional layers before the 3×3 convolutional layers to reduce the number of filters. Figure 2 illustrates the YOLOv7 architecture.

In YOLOv7, when compared with state-of-the-art real-time object-detection models, reduces the parameter count by approximately 40% and the computational workload by about 50%. YOLOv7 primarily focuses on two aspects of optimization. First, it employs extended-scaling methods to optimize parameters and computational workload within the model architecture. Second, YOLOv7 uses reparameterization techniques to replace the original modules and employs a dynamic label-assignment strategy to optimize the training process. This strategy efficiently assigns labels to different output layers. On the left side of Figure 2, the backbone describes the process from input to feature extraction, including the repetitive execution of convolution, batch normalization, SiLU/ReLU (CBS), as well as edge-enhanced local attention network (ELAN), and multi-scale positive interaction-enhanced local attention network (MPI-ELAN) units. The head on the right side of the figure illustrates the feature extraction performed through concatenation (CAT), upsampling (UP), and maximum pooling (MP). The final output is generated through repeated ELAN head (ELAN-H) and re-parameterized convolution (REP-CONV) layers.

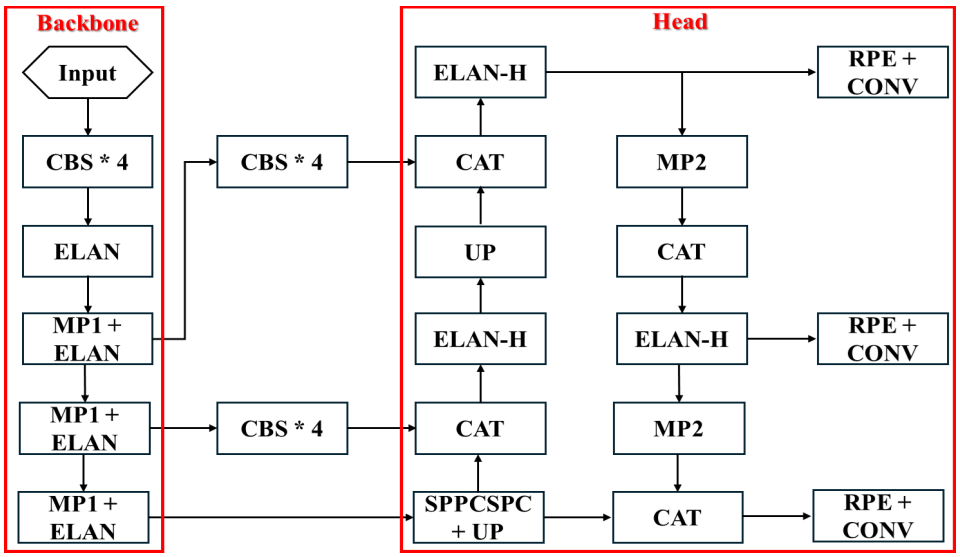


Figure 2. YOLOv7-based architecture

where ELAN denotes the edge-enhanced local attention network, MPI-ELAN is multi-scale positive interaction-enhanced local attention network, CAT is concatenation, UP is upsampling, MP: maximum pooling, ELAN-H is ELAN head, REP-CONV is re-parameterized convolution, CBS denotes the convolution, batch normalization, SiLU/ReL, SiLU is sigmoid linear unit, ReL is rectified linear unit.

The model training process based on the YOLOv7-based architecture is described in training algorithm 1, as shown in Figure 3.

Algorithm 1 YOLOv7 Training Process

1: **Input:**
2: Training samples $X = \{x_1, x_2, \ldots, x_k\}$
3: Labels $Y = \{y_1, y_2, \ldots, y_k\}$
4: Number of epochs E
5: Batch size B
6: Learning rate α
7: **Output:** The well-trained YOLOv7 model
8:
9: **Stage 1: Model Initialization**
10: Construct the YOLOv7 model: $model \leftarrow YOLOv7()$
11: Initialize the parameters θ (weights and biases):
12: $optimizer \leftarrow$ torch.optim.SGD($model.parameters(), \mathrm{lr} = \alpha$)
13:
14: **Stage 2: Training Loop**
15: **for** epoch = 1 **to** E **do**
16: **for** batch in dataloader **do**
17: $X_b, Y_b \leftarrow$ batch ▷ Randomly select a batch of instances from X
18: $optimizer.zero\_grad()$ ▷ Clear the gradients
19: Forward pass the training samples through the YOLOv7 model:
20: $(Y_b)^{\hat{}} = model(X_b)$
21: Compute the training loss $L(\theta)$ by:
22: $L(\theta) = \mathrm{box\_loss} + \mathrm{object\_loss} + \mathrm{class\_loss}$
23: Backpropagate $L(\theta)$ through YOLOv7 and update the parameters:
24: $L(\theta).backward()$
25: $optimizer.step()$ ▷ with SGD
26: Print loss for current batch
27: **end for**
28: Print loss for current epoch
29: **end for**
30:
31: **Stage 3: Validation (Optional)**
32: Validate the model with validation dataset every few epochs.
33: Adjust hyperparameters or apply early stopping if validation loss does not improve.
34:
35: **Stage 4: Save Model**
36: Save the final model weights:
37: torch.save($model.state\_dict()$,"$yolov7\_final.pth$")

Figure 3. Algorithm of YOLOv7 training process

From Algorithm 1, the entire training procedure can be divided into four stages: model initialization, training loop, validation (optional), and save model. The descriptions are as follows:

1. In the model initialization stage (lines 9-12):

Once the YOLOv7 model is created, its parameters (such as weights and biases) are initialized, along with other parameters such as the learning rate set to α. These parameters are then updated using a stochastic gradient descent (SGD) optimizer provided by PyTorch.

2. In the training loop stage (lines 14-29):

Several training epochs are set up. In each epoch, the model retrieves training samples and labels in batches from the data loader. For each batch, forward propagation is performed to predict the results, and then training losses, including bounding box loss, object loss, and classification loss, are calculated. Finally, this training loss is used for backpropagation and the parameters are updated by using SGD optimizer.

3. In the validation (optional) stage (lines 31-33):

The validation dataset is used to evaluate the performance of the YOLOv7 model. If it is observed that the validation loss does not improve, hyperparameters will be adjusted or an early stopping strategy will be employed to prevent overfitting and ensure the model's generalization ability. This stage helps to assess and enhance the accuracy of the model.

4. In the save model stage (lines 35-37):

Upon the completion of the training process, the final model weights are saved using PyTorch's torch.save function, which stores the YOLOv7 model's state dictionary (state_dict) into a file named 'yolov7_final.pth'.

4 Experimental Results

4.1 Dataset and Evaluation Metrics

The yawning detection dataset (YawDD) [21] and National Tsing Hua university drowsy driver detection (NTHU-DDD) [22] datasets were used for evaluation in this study. YawDD consists of 2,000 images contributed by various drivers, capturing driver behaviors such as yawning and eye closing. It includes 723 yawning samples and 726 samples of open and closed eyes, which were used to train the driver-fatigue behavior in this study.

On the contrary, NTHU-DDD dataset was collected from 37 participants under various driving scenarios. NTHU-DDD uses active infrared (IR) illumination to capture infrared images at a resolution of 640x480 pixels in AVI format. The Night_BareFace and Night_Glasses scenarios were recorded at 15 frames per second, while the BareFace, Glasses, and Sunglasses scenarios were recorded at 30 frames per second. These datasets were divided into training, validation, and testing sets. The testing images were created by mixing images from different driving scenarios.

The evaluation metrics of mean average precision (mAP), precision, recall, and F1-score, which are three evaluation indicators used to assess the performance of the detection system, were employed in this study to assess the detection performance:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad \textbf{(1)}$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad \textbf{(2)}$$

$$\text{F1 score} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall}) \quad \textbf{(3)}$$

where, TP (true positive) represents predictions that were correctly identified as positive, FP (false positive) represents predictions that were incorrectly identified as positive, and FN (false negative) represents predictions that were incorrectly identified as negative.

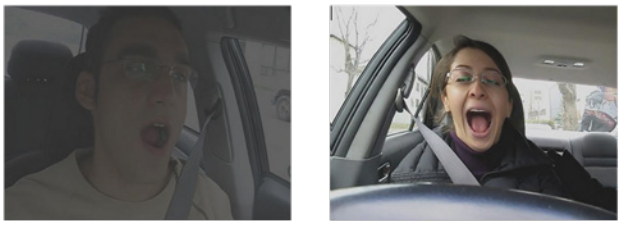
Additionally, a comparison was made with the methods used in other papers that used the same datasets. This allowed the determination of whether the approach used in this study offered advantages over existing methods on the datasets.

4.2 Experimental Environment and Settings

This study was conducted in a Windows environment using the Anaconda software suite. Microsoft Visual Studio Code served as the development platform, and the programming language used was Python. Additionally, PyTorch was chosen as the deep-learning framework. The proposed methods were implemented within the PyTorch framework specifically using Torch version 1.2.0, and OpenCV version 3.4.1 was employed for various computer vision tasks. Training was conducted using the open-source YOLOv7 code available on GitHub. This combination of software libraries and frameworks facilitated the development and training of the models for the study.

To better align with the input requirements of the model, image sizes were set to 640x640. During the training task, the Adam optimization was employed with an initial learning rate of 10^2, and the training process lasted for 300 epochs with mosaic augmentation. Furthermore, all training procedures were conducted on an NVIDIA RTX 2070 GPU and an Intel i7-8750H CPU.

The data augmentation toolkit, Augmentor [23], was used to augment the dataset. Figure 4 illustrates the data augmentation that is a data analysis technique based on existing data that involves minor adjustments or synthesis of new data to prevent overfitting, particularly when dealing with small datasets. This technique involves applying operations such as affine transformations, rotation, brightness, contrast adjustments, and more to the existing data. The primary goal is to increase the diversity of the dataset and improve model accuracy.



(a) (b)

(a) Contrast method (b) Affine transformation method

Figure 4. Data augmentation using augmenter [21]

4.3 Comparison

Figure 5 and Figure 6 illustrate the overall accuracy of the model evaluated using mAP. From Figure 5 and Figure 6, it is evident that the mAP achieved 91.75% for the YawDD dataset and 96.90% for the NTHU-DDD dataset. Furthermore, the average precision for detecting yawning and eye closure behaviors was found to be above 0.85 in both datasets.

Table 3 and Table 4 show the comparisons with other methods on YawDD and NTHU-DDD dataset, respectively. Table 3 shows that, compared to references [24-25], our proposed method consistently achieved better results in terms of recall, precision, and F1 Score on the same YawDD dataset. However, Table 4 shows that, in studies conducted using the NTHU-DDD dataset for training, our proposed method achieved higher precision compared to those in references [26-27]. The results show that our proposed combination of multi-emotion features with YOLOv7 achieved high accuracy.

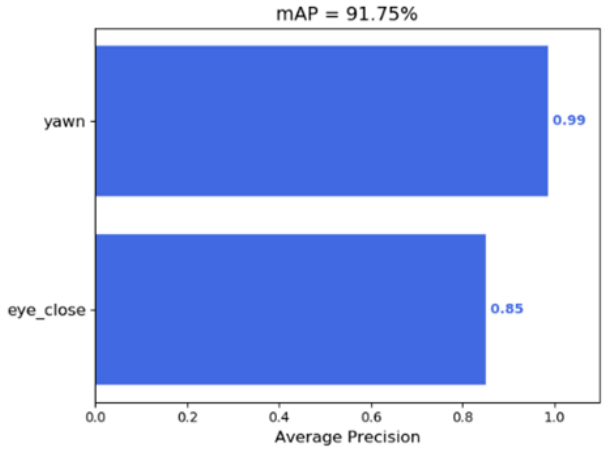


Figure 5. Evaluation of mAP on the YawDD dataset

Figure 7 illustrates the actual in-vehicle detection results of the driver's mental state by the proposed model. The system can label the distracted or fatigued behavior of the driver inside the car. However, the system may not accurately determine the driver behavior or may miss object labeling in some situations such as occlusions, lighting effects, and object angles.

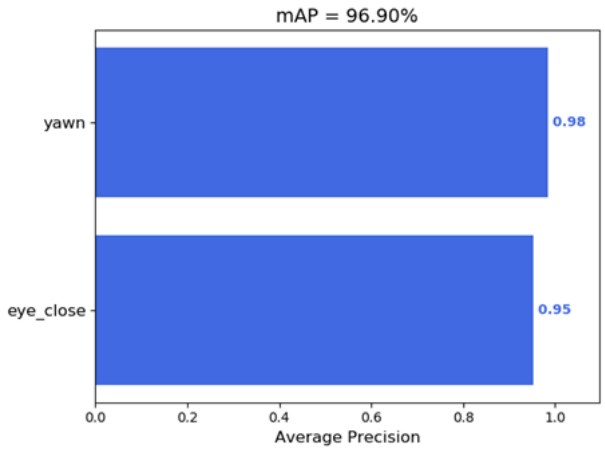


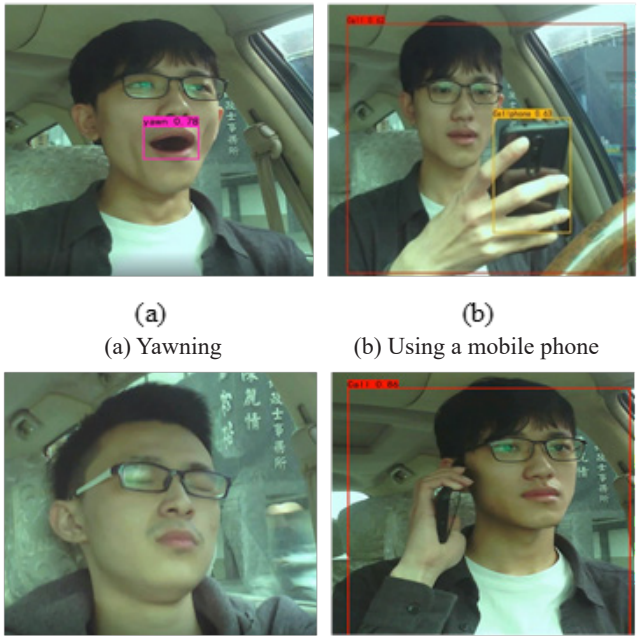
Figure 6. Evaluation of mAP on the NTHU-DDD dataset

Table 3. Comparison with other methods on the YawDD dataset

Methods	Recall	Precision	F1 Score
YOLO + MLP [24]	0.58	0.54	65.80
YOLOv3 [25]	0.80	0.86	70.85
CN + DSST [26]	0.85	0.96	63.00
This work	**0.97**	**0.96**	**96.50**

Table 4. Comparison with other methods on the NTHU-DDD dataset

Methods	Precision
DDD Network [12]	0.73
3D-DCNN [27]	0.92
This work	**0.93**



(a) Yawning (b) Using a mobile phone

(c) Detection affected by lighting (d) Detection affected by the angle of the mobile phone

Figure 7. Actual in-vehicle state detection results

5 Conclusion

Currently, most driver attentiveness-detection methods primarily focus on detecting driver fatigue, with limited consideration for distracted driving. An approach based on multi-feature processing and YOLOv7 object detection was introduced in this study to effectively detect various driver states, including distracted driving and fatigue. Experimental results demonstrated that the proposed approach achieved recall, precision, F1 Score, and mean average precision of 0.97, 0.96, 96.50, and 91.75%, respectively, on the YawDD, and precision and mean average precision of 0.93 and 96.9%, respectively, on the NTHU-DDD dataset. The method proposed in this study exhibited higher accuracy compared to other approaches. It achieved good results in detecting distracted driving and effectively recognized driver behavior in actual driving scenarios.

Acknowledgments

The authors would like to thank the anonymous reviewers of their paper for the many helpful suggestions. This work was partially supported by the National Science and Technology Council (NSTC), Taiwan, under grant number 113-2222-E-013 -001.

References

[1] T. A. Ranney, E. Mazzae, R. Garrott, M. J. Goodman, *NHTSA driver distraction research: past, present, and future*, 2000.

[2] H. D. Croo, M. Bandmann, G. M. Mackay, K. Rumar, P. Vollenhoven, *The role of driver fatigue in commercial road transport crashes*, European Transportation Safety Council, 2001.

[3] Y. Liang, J. D. Lee, Combining cognitive and visual distraction: less than the sum of its parts, *Accident Analysis and Prevention*, Vol. 42, No. 3, pp. 881-890, May, 2010.

[4] S. A. Khan, S. Hussain, S. Xiaoming, S. Yang, An effective framework for driver fatigue recognition based on intelligent facial expressions analysis, *IEEE Access*, Vol. 6, pp. 67459-67468, October, 2018.

[5] J.-S. Chiang, C.-H. Hsia, H.-J. Chen, T.-J. Lo, VLSI architecture of low memory and high speed 2-D lifting-based discrete wavelet transform for JPEG2000 applications, *IEEE International Symposium on Circuits and Systems*, Kobe, Japan, 2005, pp. 4554-4557.

[6] S. Panda, M. Kolhekar, Feature selection for driver drowsiness detection, in: N. Chaki, N. Devarakonda, A. Sarkar, N. Debnath (Eds.), *Lecture Notes on Data Engineering and Communications Technologies*, Vol. 28, Springer, Singapore, 2019, pp. 127-140.

[7] J. Solaz, J. Laparra-Hernández, D. Bande, N. Rodríguez, S. Veleff, J. Gerpe, E. Medina, Drowsiness detection based on the analysis of breathing rate obtained from real-time image recognition, *Transportation Research Procedia*, Vol. 14, pp. 3867-3876, 2016.

[8] L. Bergasa, J. Nuevo, M. Sotelo, R. Barea, M. E. Lopez, Real-time system for monitoring driver vigilance, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 1, pp. 63-77, March, 2006.

[9] W. Liao, W. Zhang, Z. Zhu, Q. Ji, A real-time human stress monitoring system using dynamic Bayesian network, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, San Diego, CA, USA, 2005, pp. 1-8.

[10] F. Friedrichs, B. Yang, Camera-based drowsiness reference for driver state classification under real driving conditions, *IEEE Intelligent Vehicles Symposium*, La Jolla, CA, USA, 2010, pp. 101-106.

[11] V. K. Reddy, K. S. Swathi, Investigation of effectiveness of simple thresholding for accurate yawn detection, in: S. Bhatia, K. Mishra, S. Tiwari, V. Singh (Eds.), *Advances in Intelligent Systems and Computing*, vol. 553, Springer, Singapore, 2017, pp. 81-89.

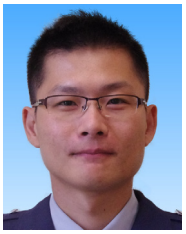
[12] A. Eskandarian, R. Sayed, P. Delaigue, J. Blum, A. Mortazavi, *Advanced driver fatigue research, U.S. Department Transportation, Federal Motor Carrier Safety Administration*, Washington, DC, Technical Report, Report FMCSARRR-07-001, April, 2007, pp. 16-23.

[13] K. Kircher, C. Ahlstrom, A. Kircher, Comparison of two eye-gaze based real-time driver distraction detection algorithms in a small-scale field operational test, *International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*, Big Sky, MT, USA, 2009, pp. 16-23.

[14] D. Zhang, H. Pu, F. Li, R. S. Sherratt, S.-J. Lim, Few shot object detection via a generalized feature extraction net, *Journal of Internet Technology*, Vol. 24, No. 2, pp. 305-312, March, 2023.

[15] M. M. Kamruzzaman, S. A. Alanazi, M. Alruwaili, Y. Alhwaiti, A. Alsayat, Blockchain as a services based deep facial feature extraction architecture for student attention evaluation in online education, *Journal of Internet Technology*, Vol. 24, No. 3, pp. 745-757, May, 2023.
[16] C.-H. Hsia, J.-S. Chiang, Real-time multiple moving objects detection and tracking with direct LL-mask band scheme, *International Journal of Innovative Computing, Information and Control*, Vol. 8, No. 7(A), pp. 4451-4468, July, 2012.
[17] F. Wang, J. Zheng, J. Zeng, X. Zhong, Z. Li, S2F-YOLO: an optimized object detection technique for improving fish classification, *Journal of Internet Technology*, Vol. 24, No. 6, pp. 1211-1220, November, 2023.
[18] C.-H. Hsia, J.-M. Guo, Efficient modified directional lifting-based discrete wavelet transform for moving object detection, *Signal Processing*, Vol. 96, Part B, pp. 138-152, March, 2014.
[19] M. Kuo, H.-T. Chan, C.-H. Hsia, Study on mask R-CNN with data augmentation for retail product detection, *IEEE International Symposium on Intelligent Signal Processing and Communication Systems*, Hualien City, Taiwan, 2021, pp. 1-2.
[20] C.-H. Hsia, C.-H. Liu, New hierarchical finger-vein feature extraction method for iVehicles, *IEEE Sensors Journal*, Vol. 22, No. 13, pp. 13612-13621, July, 2022.
[21] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, B. Hariri, *YawDD: Yawning Detection Dataset*, IEEE Dataport, 2020.
[22] NTHU-DDD video dataset. Available online: https://cv.cs.nthu.edu.tw/php/callforpaper/2016_ACCVworkshop/ (accessed on 20 July 2022).
[23] Augmentor. Available online: https://github.com/mdbloice/Augmentor (accessed on 20 July 2022).
[24] L. Li, B. Zhong, C. Hutmacher Jr, Y. Liang, Y. Liang, X. Xu, Detection of driver manual distraction via image-based hand and ear recognition, *Accident Analysis & Prevention*, Vol. 137, pp. 1-10, March, 2020.
[25] P. Mao, K. Zhang, D. Liang, Driver distraction behavior detection method based on deep learning, *IOP Conference Series Materials Science and Engineering*, Vol. 782, pp. 1-8, 2020.
[26] S. Park, F. Pan, S. Kang, C. D. Yoo, Driver drowsiness detection system based on feature representation learning using various deep networks, *Asian Conference on Computer Vision Workshops*, Taipei, Taiwan, 2016, pp. 154-164.
[27] J. Yu, S. Park, S. Lee, M. Jeon, Driver drowsiness detection using condition-adaptive representation learning framework, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 11, pp. 4206-4218, November, 2019.

Biographies



Sheng-Tao Chen received his Ph.D. degree from the Department of Electrical and Electronic Engineering of Chung Cheng Institute of Technology, National Defense University, Taiwan, in 2020. He was an Assistant Professor in the Republic of China Air Force Academy, Taiwan, in 2020. He current research interests include AIoT and Computer Vision.

Chih-Hsien Hsia received the Ph.D. degree in Electrical and Computer Engineering from Tamkang University, and the second Ph.D. degree from National Cheng Kung University, Taiwan, respectively. He currently is a Distinguished Professor and a Chairperson with the Department of Computer Science and Information Engineering, NIU. His research interests include DSP IC Design, GenAI/AI in Multimedia, and Cognitive Engineering.

Yo-Cheng He received the B.S. and M.S. degrees in Computer Science and Information Engineering from National Ilan University, Taiwan, in 2024. His research interests include deep learning, computer vision, and AI applications for autonomous vehicles.