Driver Distracted Behavior Detection Technology with YOLO-Based Deep Learning Networks

Yen-Sok Poon, Ching-Yun Kao, Yen-Kai Wang, Chih-Chin Hsiao, Ming-Yu Hung, Yu-Ching Wang, and Chih-Peng Fan*
Department of Electrical Engineering, National Chung Hsing University, Taichung city, Taiwan (R.O.C.)
Email: cpfan@dragon.nchu.edu.tw*

2021 IEEE International Symposium on Product Compliance Engineering - Asia (ISPCE-ASIA) | 978-1-6654-4342-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISPCE-ASIA53453.2021.9652435

***Abstract*— In order to develop a non-contact driving behavior detection system for the improvement of driving safety, in this study, the YOLO-based deep learning technology is utilized by setting up a webcam on the dashboard to detect the driver's behaviors. By RGB-channel images as inputs, the YOLO-based deep learning models, including YOLOv3-tiny, YOLOv3-tiny-3l, YOLO-fastest, YOLO-fastest-xl are adopted and trained as the candidate detectors. The detected behaviors involve normal driving, distracted head turning, drowsiness, eating, talking on the phone, etc. The experimental results show that when the same parameters are set, the YOLO-fastest-xl has the best performance with multi-category datasets, and its F1-score, false negative rate (FNR), and mAP are 91.84%, 6.94%, and 95.81%, respectively. By the software implementation, the proposed design performs 30 frames per second (FPS) on the NVIDIA GPU-based embedded platform.**

Keywords—deep learning, object detection, YOLO, driver distracted behavior

I. INTRODUCTION

Since the automobile was invented, it had provided the advantages of high power and carrying capacity, and the automobile had become the most reliant means of transportation. However, the resulting traffic accidents have also emerged one after another. In addition to over-speed driving, the distractive behaviors, such as fatigue, dozing off, eating, and using mobile phones have also greatly increased the frequency of traffic accidents. Fig. 1 demonstrates some examples of driver's distracted behaviors

Fig. 1 Examples of driver distracted behaviors with/without wearing glasses (From left to right: closed-eyes, distraction, normal, and distraction)

The driver's behavior detection technology could be roughly divided into two categories, which included the "contact" and "non-contact" behavior detection systems [1-5, 7-9]. One of the contact-type driver distraction behavior detection technologies used electroencephalographic (EEG) signals to measure brain activity [1]. The specific method was to place electrodes on the scalp to record the voltage fluctuations generated by the ionic current of the brain neurons, and then the system determined whether the driver was distracted or not. However, the use of a contact system to detect driver behavior required the contact between the equipment and the driver to obtain measurement data, which caused discomfort to the driver for long-time uses. Different from the "contact" based designs, the non-contact driving behavior detection technology took into account the driver's comfort and the convenience of uses, and did not directly contact the driver during uses. The system obtained the driver's images through the camera, and the digital image processing was performed on the captured image, and then the behavior of the driver was analyzed and inferred. In [3], a driving fatigue detection system used an infrared camera to obtain driver's images. Fig. 2 shows the traditional design flow for the image-based driver drowsy detection [3]. The infrared camera provided active light sources, discard color information, and used only grayscale information for image processing. The system by using infrared camera was to avoid the image information to be affected by various light. The system in [3] was divided into two sub-systems, which included the nodding detection and drowsiness detection. The purpose of nodding detection was to avoid the situation that the driver could not find his eyes after wearing sun glasses. The method of drowsiness detection was to measure the length of time when the driver closed his eyes. If the system detected one of the drowsiness states, the system issued a warning to remind the driver. Fig. 3 depicts the image-based driver drowsy detection system.

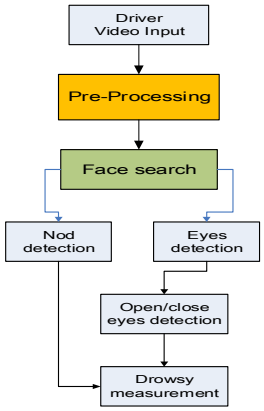

Fig. 2 The traditional design flow for driver drowsy detections [3]

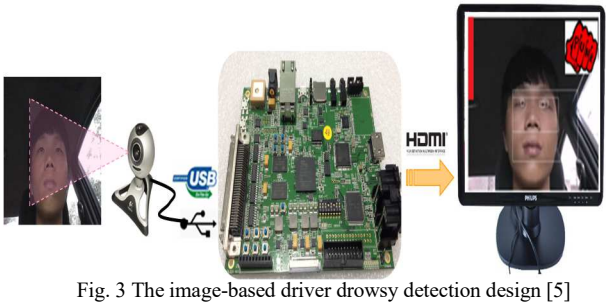

Fig. 3 The image-based driver drowsy detection design [5]

Related works

978-1-6654-4342-5/21/$31.00 ©2021 IEEE

With the rise of artificial intelligence, the awareness of driving safety has also gradually risen. Many researchers who focused on artificial intelligence have participated in the development of related technologies. Various sensors were used in the driving system to detect and integrate multi-layer perceptions. By this way, many different levels of driving assistance functions and applications were planned and developed. Among these technologies, object detection through images was a very important part for the deep learning applications [6]. In [8], the system provided the functions to detect driving drowsiness through deep learning and computer vision. When the driver was not attentive, the warning bell sounded, and the image recognition was completed through the CNN model. The advanced CNN model was used to estimate the position of the face and eyes to obtain better detection results while reducing false positive and false negative rates. The driving behavior detection system was divided into three parts, which includes eye predict, face predict, and hand predict. Using a mobile phone while driving a vehicle will seriously threaten traffic safety. In [9], in order to prevent the driver from being distracted by using a mobile phone, it was extremely important to monitor the driver's behavior through real-time captured images. The previous methods were easily affected by factors, such as object obstruction, image rotation, and light and shadow changes. Therefore, this design proposed a driver's mobile phone detection method based on deep learning, and the method was divided into two stages. Firstly, the Progressive Calibration Network (PCN) was used to detect the face and face tracking to determine the detection area. Next, the CNN-based driver mobile phone detection method was used to detect mobile phones in the candidate area.

In this paper, the YOLO-based deep learning technology is utilized by setting up a webcam on the dashboard for detecting the driver's behaviors. By training a driver behavior detection based on YOLO based models, the trained model can analyze the driver's images taken by the camera in the car, and recognize the real-time driving behavior to judge whether the driver is distractive or not, and the proposed design achieves to reduce the frequency of traffic accidents.

II. YOLO-BASED MODELS

The YOLO series [10, 11] adopted a multi-scale input training strategy from the second generation, i.e. YOLOv2. Specifically, the input image size of the model was randomly changed after a certain interval of iterations during the training process, and the range of random resize fell within 320x320 pixels between 608x608 pixels and a multiple of 32. At the same time, the use of multi-scale training strategy made the model adapt to different sizes of images and improved the robustness of the model. The YOLOv3 [12] was the third-generation model of the YOLO series. The difference from the previous two versions was that the original single-label multi-class Softmax used for classification was changed to multi-label multi-class logistic. The purpose was to improve the complex scenario, and an object belonged to multiple categories, which led to the inapplicability of Softmax. The logistic regression layer used the Sigmoid function constrained the input within the range of 0 to 1.

YOLOv3-tiny was a lightweight version of YOLOv3, which had been reduced from the original 100-layer network to more than a dozen layers. Although the accuracy was slightly reduced, because the goal of this work was to train the model for real-time detections through an embedded platform, it was appropriate to use YOLOv3-tiny as the network architecture. YOLOv3-tiny_3l was an extended version of YOLOv3-tiny. It was mainly aimed at improving the detection of small objects, i.e. 16x16 pixels by adding a set of feature maps of different scales. To improve the detection accuracy of small objects, although there was a slight increase in the amount of calculations, if the used datasets had more small object targets for high accuracy requirements, the YOLOv3-tiny-3l model will be better than the YOLOv3-tiny model.

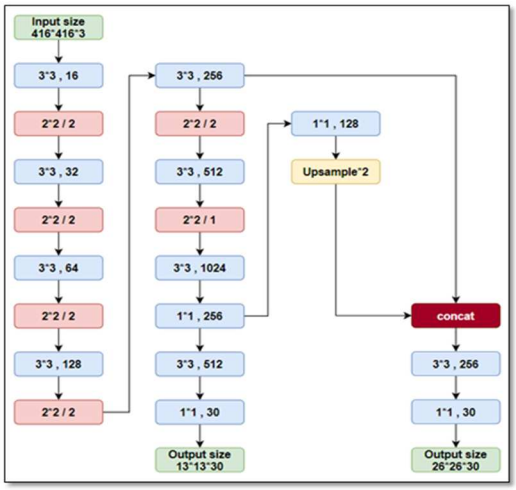

Fig. 4 Architecture of the YOLOv3-tiny model

The YOLO-fastest [13] model was an improved version of YOLO, and it was the lightest and fastest general object detection model in the YOLO series. The YOLO-fastest model integrated the model, i.e. EfficientNet-Lite [14] as shown in Fig. 5, which was jointly launched by Google and TensorFlow for image classifications, to be as the main architecture. The YOLO-fastest model performed quite well in terms of performances, and the amount of parameters and calculations used were very low. Although YOLO-fastest reduced the amount of parameters and calculations to a very low level, the memory exchange, that occurred when the model completed a forward pass, had increased a lot. In addition, YOLO-fastest-xl was an extended version of YOLO-fastest, and it doubled the number of filters in each layer of YOLO-fastest to increase the final detection accuracy.

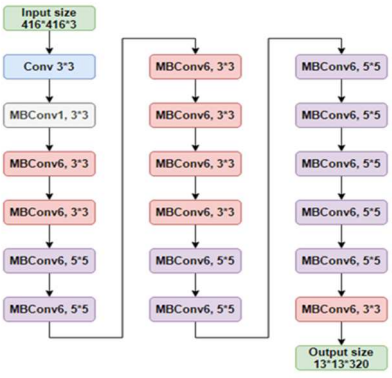

Fig. 5 Architecture of EfficientNet-lite [14]

III. PROPOSED YOLO-BASED DESIGN

Fig. 6 reveals the design flow of the proposed method. The databases used in this work are divided into two categories. One dataset for the driving behavior detection is collected by our laboratory, as shown in Fig. 7, and the others are selected from public image datasets, which are also re-labeled based on the driver's behavior categories. The used public datasets

are Pointing'04 database [15], FEI face database[16] and WIDER FACE database [17]. Table 1 lists the numbers of the used datasets.

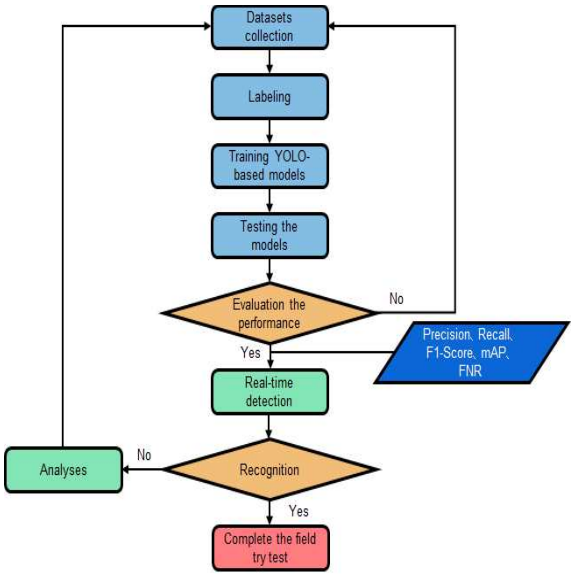

Fig. 6 The proposed design flow

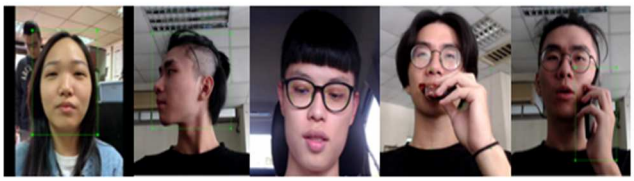

Fig. 7 Self-collected datasets for the driving distractive behaviors

Table 1 Numbers of the used datasets in this work

Datasets	Numbers of images
Self-collected driving distractive behavior images	32917
Pointing'04 [15]	2819
FEI face [16]	2695
WIDER FACE [17]	547
Total numbers	38978

The Pointing'04 database contains single face images of 384x288 pixels with 15 characters varying between the horizontal angle and the tilt angle between -90 and +90 degrees. In the dataset, 2819 images are selected to be used in this work. The FEI face database includes 640x480-pixel size single face images taken by 200 different people under a white uniform background. The vertical rotation angle is unchanged, and the horizontal rotation angle varies between 180 degree. In this dataset, 2695 images are selected to be used in this study.

In order to classify the driving behaviors, before generating the training data, we define the driving behaviors to the five corresponding categories. Table 2 defines the driving behaviors used in this work. The definition of driver's distracted behaviors includes "Warning", "Drowsy", "Eat", and "Phone" states. The purpose of defining driving behaviors is to correctly judge the distractive behaviors of drivers, reduce the occurrence of accidents, and improve the driving safety. Figs. 8 and 9 are classification diagrams of deriver's behavior definitions. Figs. 10 and 11 are classification diagrams by the public Pointing'04 and FEI face datasets, respectively.

Table 2 Definitions of the five driving behaviors

Five Categories		Definitions
Good		With the person's front face as the starting axis, the horizontal and vertical rotations are **within 30 degrees and the eyes are open**
Distracted behaviors	Warning	Taking the person's front face as the starting axis, rotate **more than 30 degrees horizontally and vertically**
	Drowsy	The eyeball is covered by the eyelid by more than 50%
	Eat	Holding food close to the driver's mouth
	Phone	Holding the smartphone close to driver's ears

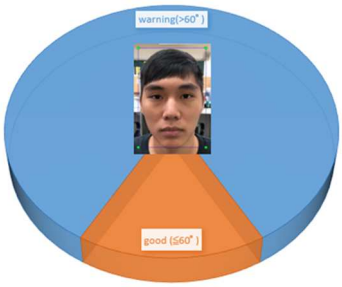

Fig. 8 Classification diagram for the "Good" status

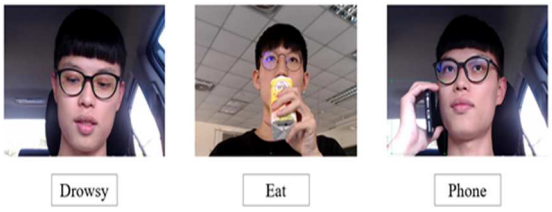

Fig. 9 Classification diagram for the "Drowsy", "Eat", and " Phone" states

Fig. 10 Classification diagram by Pointing' 04 database

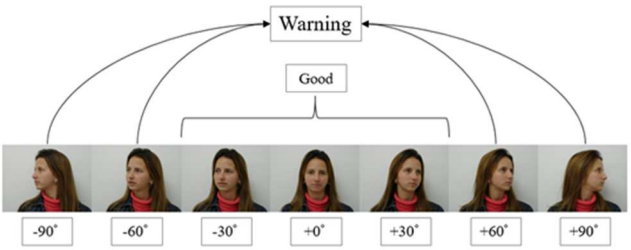

Fig. 11 Classification diagram by FEI face database

To generate the required driving behavior annotation file, we use an auxiliary program-LabelImg that provides the annotation function to select the bounding box, and the "LabelImg" tool generates the annotation file of driving inattentive behavior. In addition to saving the category information of the bounding box, the applied tool also saves the coordinate point information of the upper left and lower right of the bounding box to locate the position and size of the

bounding box. After the collection of datasets, the multi-category driving behavior definition, and the behavior category labeling, we divide the collected datasets into the training, validation, and testing parts. Table 3 lists the number of images used for training, validation, and testing modes.

Table 3 Number of images used for training, validation, and testing modes

Five Categories	Training	Validation	Testing
Good	1605	324	316
Warning	13397	1750	1872
Drowsy	5599	455	650
Eat	9706	913	1681
Phone	8506	1596	789

When the deep learning training is enabled, a large amount of data is needed to ensure that the overfitting does not occur during training. For data augmentation, the random adjustment of saturation is used, and the exposure within 1.5 times and the hue within ±0.1 are used to generate more augmentation images. Besides, we also use the data jitter skill, and also turn on random multi-scale training. Then the model will randomly scale the length and width of images between 320 and 680 pixels after every 10 iterations. In addition, the Mosaic method [18] is applied to enrich the background of the detected object for the training process. Through the above mentioned data augmentation process, more images can be generated to provide model learning, and it overcomes the overfitting problem that may cause by insufficient training images. This work is based on YOLO-based models to identify the driving distractive behaviors, and the YOLO-based models are optimized by the darknet [19] framework. Since the YOLO-based models are relatively lightweight, and the models support the real-time uses with CPU and GPU platforms.

To construct a YOLO-based model, firstly a "cfg" file of the model is created, and various layers of YOLO networks are generated by the "cfg" file. Then the framework forms a complete a network model, which will be the various structures. For example, the parameters of the batch, subdivisions, width, height, channels, momentum, decay, and learning rate as well as the convolution layer, max pooling layer, and YOLO layer, are required to be set for constructing the models. Moreover, the number, length, and width of the anchor box at the YOLO layer are also calculated by the K-means technology.

IV. Experimental Results and Comparisons

In this section, the common evaluation indices for deep learning models, which include precision, recall, F1-Score, mAP (mean Average Precision), and FNR (False Negative Rate), are used for this study. The labeled datasets are divided into the training, validation, and testing sets as the ratio of 8:1:1, respectively, and then we apply different YOLO-based deep learning models to conduct the training, verification, testing, and inference processes. Table 4 lists the number of images for five categories for the testing mode. By the YOLO-based networks for inferences, the Non-Maximum Suppression (NMS) method is used to add the confidence threshold value (i.e. 0.4) and the IoU threshold value (i.e. 0.5). The training parameter settings for the YOLO-based deep learning models are listed in Table 5. Besides, the network models are unified for the input size as 416x416 pixels, and a total number of iterations is 40,200 during the training process, and then the training model is completed by the five-category detections. Table 6 lists the experimental results between YOLOv3-tiny and YOLOv3-tiny-3l models, and Table 7 describes the experimental results between YOLO-fastest and YOLO-fastest-xl models.

Table 4 Number of images for five categoies for the testing mode

Category	Good	Warning	Drowsy	Eat	Phone
Numbers of images	316	1872	650	1681	789

Table 5 Parameters setting for training the CNN models

Width*Height	416x416 pixels
Channels	3
Momentum	0.9
Decay	0.0005
Angel	0
Saturation	1.5
Exposure	1.5
Hue	0.1
Learning rate	0.001
Learning rate policy	steps
Steps	[32000, 36000]
Scales	[0.1, 0.1]
Max iterations	40200
GPU	PC2080

Table 6 Experimental results between YOLOv3-tiny and YOLOv3-tiny-3l models

Parameters setting	YOLOv3-tiny	YOLOv3-tiny-3l
IoU thresh	0.50	
Confidence thresh	0.40	
Evaluation indices		
Precision	92.14%	92.57%
Recall	87.78%	88.32%
F1-Score	89.91%	90.40%
FNR	12.22%	11.68%
Average Precision (AP)		
Good	96.53%	97.53%
Warning	98.31%	98.65%
Drowsy	94.77%	94.59%
Eat	91.88%	91.23%
Phone	94.03%	96.33%
mAP	95.10%	95.67%

Table 7 Experimental results between YOLO-fastest and YOLO-fastest-xl models

Parameters setting	YOLO-fastest	YOLO-fastest-xl
IoU thresh	0.50	
Confidence thresh	0.40	
Evaluation indices		
Precision	90.97%	90.66%
Recall	92.60%	93.06%
F1-Score	91.78%	91.84%
FNR	7.40%	6.94%
Average Precision (AP)		
Good	95.30%	96.19%
Warning	98.19%	98.44%
Drowsy	92.53%	93.97%
Eat	94.56%	94.39%
Phone	96.15%	96.07%
mAP	95.35%	95.81%

Table 8 lists the performance comparison of F1-Score, FNR, and mAP among the four YOLO-based models. Based on the same parameter settings, the YOLO-fastest-xl model has the best performance, and its F1-Score, FNR, and mAP are

91.84%, 6.94%, and 95.81%, respectively. Fig. 12 demonstrates the inference results by the proposed driver distracted behavior detection design.

Table 8 Comparison results of various CNN models with multi-category datasets

CNN Models	F1-Score	FNR	mAP
YOLOv3-tiny	89.91%	12.22%	95.10%
YOLO-v3tiny-3l	90.40%	11.68%	95.67%
YOLO-fastest	91.78%	7.40%	95.35%
YOLO-fastest-xl	**91.84%**	**6.94%**	**95.81%**

Fig. 12 The experimental results by the proposed driver distracted behavior detection (From left to right/top to down : normal, closed-eye drowsy, distraction, eating & distraction, and using the phone)

For the embedded software implementation, the Logitech C992 PRO HD webcam is used for image shooting during the test. The input image size is 1920x1080 pixels, and the output image size is set at 800x600 pixels. By the NVIDIA Xavier platform shown in Fig. 13, the proposed design performs up to 30 frame per second for the practical applications.

Fig. 13 The embedded software implementation with NVIDIA Xavier platform

V. Conclusions

In this work, to detect the driver's behaviors, the YOLO-based deep learning models, including YOLOv3-tiny, YOLOv3-tiny-3l, YOLO-fastest, YOLO-fastest-xl are applied to be the candidate detectors. The detected behaviors involve normal driving, distracted head turning, drowsiness, eating, and talking on the phone. The experimental results show that the YOLO-fastest-xl has the best performance with multi-category datasets, and its F1-Score, FNR, and mAP are 91.84%, 6.94%, and 95.81% respectively. By the software realization, the proposed design operates up to 30 FPS on the NVIDIA Xavier platform. In the future, the YOLOv4 series, the next-generation related versions, or other CNN architectures will be selected for the further studies. To increase accuracy, the performance of the overall model can be improved through the parameter adjustment and the adjustment of model's structure. For datasets, the richness of the training datasets will be enhanced by collecting more related images, and its functions can also be expanded by adding more categories. Then a more comprehensive driving distractive behavior detections can be achieved.

If the driver's head swings up and down, the system will judge it as the "Warning" state; for the driver's face skew, because the training process does not collect data on the face skew state, the system will not be able to effectively detect the face skew. In future works, we will retrain and adjust the angle parameters by data augmentation before training, and the trained model will have a chance to detect the bounding box in the skew face.

Acknowledgment

This work was financially supported by the Ministry of Science and Technology (MOST) under Grant No. 109-2218-E-005-008.

References

[1] S. Wang, Y. Zhang, C. Wu, F. Darvas, and W. A. Chaovalitwongse, "Online prediction of driver distraction based on brain activity patterns", IEEE Trans. Intell. Transp. Syst., vol. 16, no. 1, pp. 136-150, 2015

[2] M. J. C. D. Castro, J. R. E. Medina, J. P. G. Lopez, J. C. d. Goma and M. Devaraj, "A Non-Intrusive Method for Detecting Visual Distraction Indicators of Transport Network Vehicle Service Drivers Using Computer Vision," 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2018, pp. 1-5.

[3] Wei-Liang Ou, Ming-Ho Shih, Chien-Wei Chang, Xue-Han Yu, and Chih-Peng Fan, "Intelligent Video-Based Drowsy Driver Detection System under Various Illuminations and Embedded Software Implementation," IEEE International Conference on Consumer Electronics - Taiwan, Taipei, Tawian, June 2015.

[4] Wei-Cheng Li, Wei-Liang Ou, Chih-Peng Fan, Chien-Hsiu Huang, Yi-Shian Shie, "Near-Infrared-Ray and Side-View Video Based Drowsy Driver Detection System: Whether or Not Wearing Glasses," 2016 IEEE Asia Pacific Conference on Circuits and Systems, Jeju, Korea, October 2016.

[5] Cyun-Yi Lin, Paul Chang, Alan Wang, and Chih-Peng Fan , "Machine Learning and Gradient Statistics Based Real-Time Driver Drowsiness Detection," IEEE International Conference on Consumer Electronics-Taiwan, Taichung city, Taiwan, pp.1-2, May 2018.

[6] "Convolution neural network" [Online]. Available: https://en.wikipedia.org/wiki/Convolutional_neural_network

[7] Luis M. Bergasa, Associate Member, IEEE, Jesús Nuevo,Miguel A. Sotelo, Member, IEEE, Rafael Barea, and María Elena Lopez, "Real-Time System for Monitoring Driver Vigilance," IEEE Transactions On Intelligent Transportation Systems,Vol. 7, No. 1, March 2006.

[8] S. Kusuma, J. Divya Udayan and A. Sachdeva, "Driver Distraction Detection using Deep Learning and Computer Vision," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2019, pp. 289-292.

[9] Q. Xiong, J. Lin, W. Yue, S. Liu, Y. Liu and C. Ding, "A Deep Learning Approach to Driver Distraction Detection of Using Mobile Phone," 2019 IEEE Vehicle Power and Propulsion Conference (VPPC), 2019, pp. 1-5.

[10] Redmon, J., Divvala, S., Girshick, R., "You Only Look Once-Unified Real-Time Object Detection," In Proceedings of the IEEE conference on computer vision and pattern recognition, pp.779-788, 2016.

[11] Redmon, J., & Farhadi, A., "YOLO9000: better, faster, stronger." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263-7271, 2017.

[12] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767, 2018.

[13] "github:YOLO-Fastest" [Online].Available: https://github.com/dog-qiuqiu/Yolo-Fastest

[14] "github:efficientnet-lite" [Online].Available: https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet/lite

[15] Pointing'04 ICPR Workshop Head Pose Image Database

[16] "FEI face datasets", [Online].Available: https://www.face-rec.org/databases/

[17] "WIDER FACE: A Face Detection Benchmark", [Online].Available: http://shuoyang1213.me/WIDERFACE/

[18] "Mosaic data augmentation" [Online].Available: https://reurl.cc/gW6Qjb

[19] "github:darknet" [Online].Available: https://github.com/AlexeyAB/darknet