IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.DOI

Real-Time Eye Tracking for Bare and Sunglasses-wearing Faces for Augmented Reality 3D Head-Up Displays

DONGWOO KANG[1], AND LIN MA[2],
[1]Department of Electronic and Electrical Engineering, Hongik University, Seoul 04066, South Korea (e-mail: dkang@hongik.ac.kr)
[2]SAIT China Lab, SRC-Beijing, Samsung Electronics, Beijing 100007, China (e-mail: lin.ma@samsung.com)

Corresponding author: Lin Ma (e-mail: lin.ma@samsung.com).

This work was supported by 2021 Hongik University Research Fund.

ABSTRACT Eye pupil tracking is important for augmented reality (AR) three-dimensional (3D) head-up displays (HUDs). Accurate and fast eye tracking is still challenging due to multiple driving conditions with eye occlusions, such as wearing sunglasses. In this paper, we propose a system for commercial use that can handle practical driving conditions. Our system classifies human faces into bare faces and sunglasses faces, which are treated differently. For bare faces, our eye tracker regresses the pupil area in a coarse-to-fine manner based on a revised Supervised Descent Method based eye-nose alignment. For sunglasses faces, because the eyes are occluded, our eye tracker uses whole face alignment with a revised Practical Facial Landmark Detector for pupil center tracking. Furthermore, we propose a structural inference-based re-weight network to predict eye position from non-occluded areas, such as the nose and mouth. The proposed re-weight sub-network revises the importance of different feature map positions and predicts the occluded eye positions by non-occluded parts. The proposed eye tracker is robust via a tracker-checker and a small model size. Experiments show that our method achieves high accuracy and speed, approximately 1.5 and 6.5 mm error for bare and sunglasses faces, respectively, at less than 10 ms on a 2.0GHz CPU. The evaluation dataset was captured indoors and outdoors to reflect multiple sunlight conditions. Our proposed method, combined with AR 3D HUDs, shows promising results for commercialization with low crosstalk 3D images.

INDEX TERMS eye tracking, iris regression; eye position estimation; augmented reality (AR) display; autostereoscopic three-dimensional display; head-up displays (HUDs)

I. INTRODUCTION

AUGMENTED reality (AR) three-dimensional (3D) head-up displays (HUDs) are a promising technology for next-generation assistive driving systems. AR HUDs have been increasingly adopted in the automotive industry to show visual contents, such as line-of-sight navigation arrows via combiners or windshields [1], [2]. While two-dimensional (2D) HUDs can cause additional distractions and visual mismatches between real-world and virtual objects, AR 3D HUDs can overlap 3D visual information directly on the road after 3D depth adjustments [1], [2] (Figure 1a). In such systems, an autostereoscopic 3D display [3]– [5] is important to provide the user with a realistic sense of the image depth without the need of 3D eyeglasses. However, traditional autostereoscopic displays have some limitations, such as a limited best-viewing zone and that the user needs to stay in a restricted position to avoid 3D crosstalk artifacts. An eye-tracking-based autostereoscopic 3D method overcomes these limitations, allows a single-user seamless 3D experience, and provides higher 3D resolution contents [3]– [5]. A prototype AR 3D HUD with a 3D margin according to eye position is shown in Figure 1a. With more accurate and faster pupil tracking [6]– [8], 3D images with lower 3D crosstalk and higher resolution can be provided in real-time, even when the user's head movements are fast. However, accurate real-time pupil tracking while driving with an AR 3D HUD is particularly challenging due to multiple real-world driving conditions, such as varying light conditions, head pose changes, eyeglasses reflection, and eye occlusion caused by wearing sunglasses (Figure 1b). These challenges become even more difficult to overcome under limited vehicle system resources.

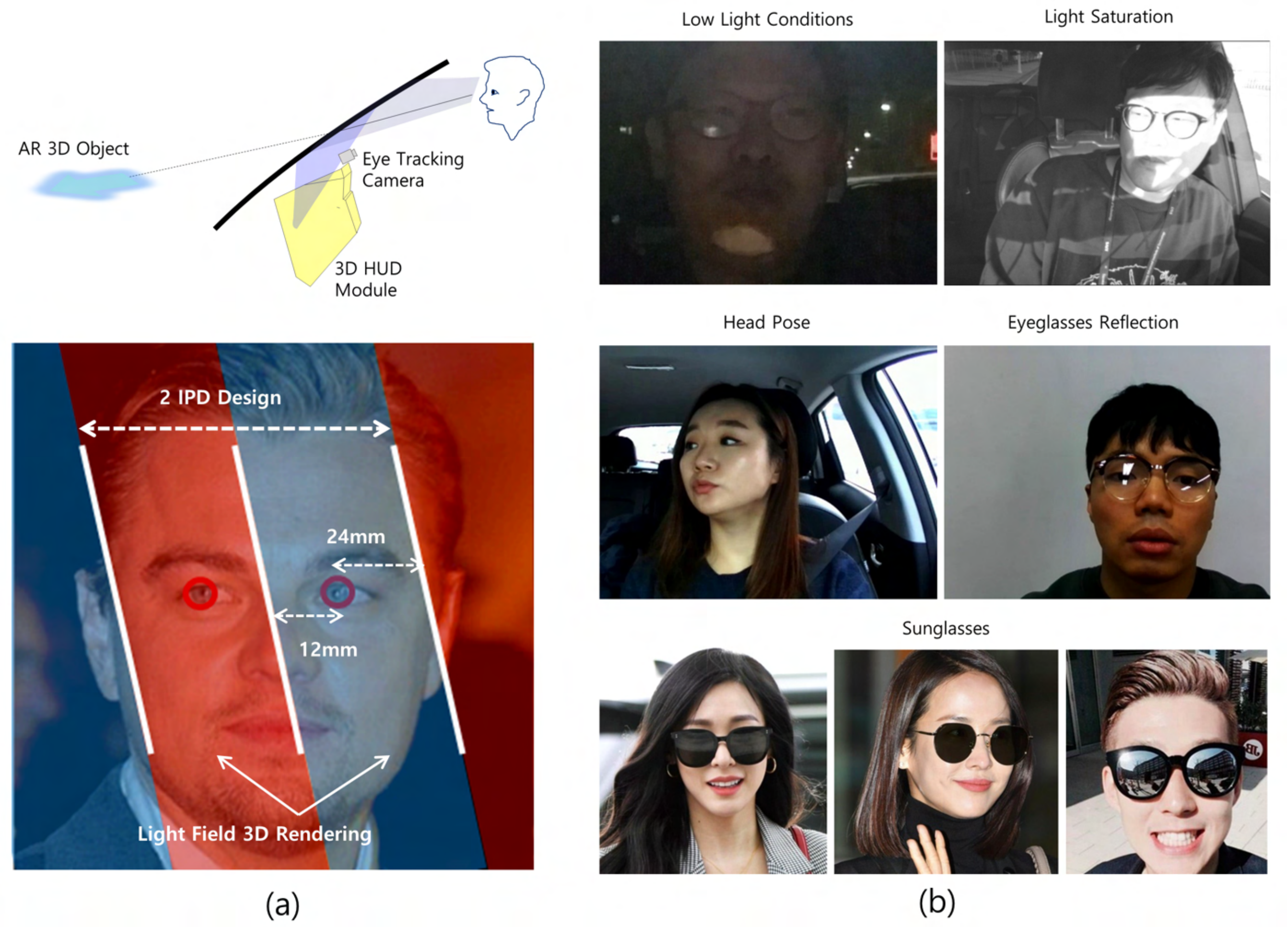


FIGURE 1. (a) Example of an AR 3D HUD system (upper panel) and eye-position margin in the x-direction of a 27-view autostereoscopic 3D display design. (b) Challenging conditions for eye pupil center tracking while driving.

In this paper, we focus on pupil tracking while the user is driving. The proposed system classifies human faces into bare faces and sunglasses faces. It consists of a face detector, eye–nose shape aligner, tracker-checker, and a switching design between two tracking modes. Depending on the user face conditions, corresponding shape aligners are applied based on image classification. For bare faces, we extended our previous studies [9], [10], which were based on the fast Supervised Descent Method (SDM) [11] algorithm with Scale-Invariant Feature Transform (SIFT) [12]. In this study, we further increased precision by adding an iris regression module, where pupil boundaries are regressed and the corresponding eye center positions are refined. Because our previous method [9], [10] is not able to handle cases when the user is wearing sunglasses, which are quite challenging due to occluded eyes and sunlight reflection, we propose a new method for sunglasses faces. To tackle such cases, we use non-occluded areas to infer the pupil center with a revised Practical Facial Landmark Detector (PFLD) network [13]. Furthermore, we propose a novel re-weight module design to improve the pupil tracking system performance when dealing with sunglasses occlusions. Our proposed method targets to implement pupil tracking on commercial AR 3D HUD systems. For commercial use, the pupil tracking module should be fast, accurate, robust, and have a small model size. Our system is designed to fulfill these four requirements with the following strategies.

- Fast. We designed an optimized processing procedure to ensure pupil center estimation is mainly performed in tracking mode in a local image area. In this way, face detection across the whole image in each frame, which is time consuming, is not needed. Additionally, for bare and sunglasses faces we utilize SDM [11] and PFLD [13], respectively, as the base method. SDM and PFLD are both high speed methods in terms of CPU time. Our proposed revisions on SDM and PFLD require little additional computational load while retaining their high speed characteristics. For bare faces, it takes only 4 ms per 640x480 image frame, and for sunglasses faces, it takes approximately 10 ms per 640x480 image frame, whereby only CPU calculations are required.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3110644, IEEE Access

These are desirable characteristics considering the limited system resources available in actual automobiles.

• Accurate. Besides the inherited accuracy of the original SDM and PFLD demonstrated in previous studies, we provide further modifications to adapt them to address the challenges of actual driving scenarios. For bare faces, we utilize an SDM-based coarse-to-fine strategy that further improves accuracy with an iris boundary regression algorithm. Despite PFLD's top performance on classical datasets compared to other widely used methods [12], in our experiment PFLD gives different positions on the feature map the same confidence, which means occlusion influences it under certain critical conditions. To address this problem, for sunglasses faces we propose a new re-weight sub-network that revises the importance of different feature map positions and then the occluded part can be inferred by non-occluded parts. We conducted intensive experiments to demonstrate the validity of the proposed method. The mean error was < 1.5 mm for bare faces and 6.5 mm for sunglasses faces, which are below the 3D margins of our AR 3D HUD prototype.

• Robust. Our method includes a design to recover quickly from tracking failure in order to ensure a robust system. Specifically, we designed two fast tracker-checker modules. For bare faces, we perform Support Vector Machine (SVM) [14] with the extracted SIFT features around eyes and nose to act as tracker-checker. Experiments show that both SIFT and SVM are fast and robust. For sunglasses faces, because image features around the eyes are not available, we designed a new PFLD-based tracker-checker where we defined a cross entropy loss of whole face landmark points for the predicted tracking confidence. This tracker-checker shares feature maps with previous landmark detection steps and can characterize the representation ability of the current face image directly. Experimental results demonstrate the validity of the proposed tracker-checker modules. For bare and sunglasses faces, the tracker-checker took less than 1ms per frame. With the proposed tracker-checker designs, the pupil tracking can fulfill real-time requirements faster than 30 fps.

• Small model size. Because our design adds little additional space, the system model size is small. For bare faces, the model is approximately 500 KB, for sunglasses faces, the model size is less than 1300 KB.

The main contributions of this study can be summarized as follows:

1) We propose a new eye pupil tracking system that can handle both bare and sunglasses faces with two different strategies. The system optimizes the eye pupil tracking process with multiple collaborative modules, such as face detector, face classifier, facial landmark detector, and tracker-checker, which allows it to run in a fast, accurate, and robust manner. Additionally, we also propose an iris regression method, which refines eye center locations by regressing pupil centers and iris boundaries. By using a coarse-to-fine strategy, pupil centers can be detected and tracked fast and accurately.

2) For sunglasses faces, we infer the pupil in the sunglasses area with non-occluded areas, such as the nose and mouth, with a structural inference-based re-weight method and construct an end-to-end trainable network. When defining the weights, we consider both the feature for each specific landmark and the structural relationship between landmarks. Additionally, we add a small branch to the Convolutional Neural Network (CNN) backbone when treating sunglasses faces and use it as a tracker-checker. The tracker-checker shares features with the CNN-based landmark detector and can save computational time. Furthermore, we use the landmark prediction error as guidance to train the tracker-checker, which can represent the failure condition of the tracker more reasonably.

II. RELATED WORK

Facial Landmark Detection. Facial landmark localization has been researched for decades. Before the advent of deep learning methods, active shape [15], active appearance [16], and cascaded regression [17] models were widely used. Some researchers also utilized SDM for fast landmark detection [11], [18]. However, the hand-crafted features fail to represent multiple complex facial appearance conditions. Recently, deep learning-based methods have demonstrated more powerful facial landmark detection abilities.

Deep learning, especially CNNs, can model the facial appearances in multiple conditions by adopting a large number of learned parameters. TCDCN [19] utilizes auxiliary attributes to pre-train the CNN and then uses the pre-trained network to train the facial landmark detection model. Under certain conditions, the images are captured by different cameras and exhibit large style differences, which can cause a severe domain shift problem. To address this problem, some researchers integrate style transfer into facial landmark detection. As Generative Adversarial Networks (GANs) [20] are a powerful for transferring image styles [21], many previous works take advantage of GANs [22]. Previous methods first obtain several image styles via clustering, and then train the style transfer model with a GAN to obtain the aggregated style. During testing, the aggregated style image is combined with the original image and used as input to the network for performance improvement. Qian et al. [23] decoupled the shape and style of the image with an adversarial network and used this network to generate more training samples to introduce more sample variation. The loss function also greatly influences detection performance. Feng et al. [24] analyzed the influence of different prediction errors on loss computation and proposed a wing loss [24]. By amplifying small and medium errors, the loss function is more sensitive to errors around zero, then accuracy is improved. Wang et al. [25] further analyzed the influence of foreground and background on the loss computation in heat map-based methods and proposed an adaptive wing loss, which can adapt the shape to different ground truth heat map pixels.

Multi-Step Facial Landmark Detection. To improve the detection speed, PFLD [13] uses MobileNet [26], [27] as the backbone and achieves high speed on CPU mode. In

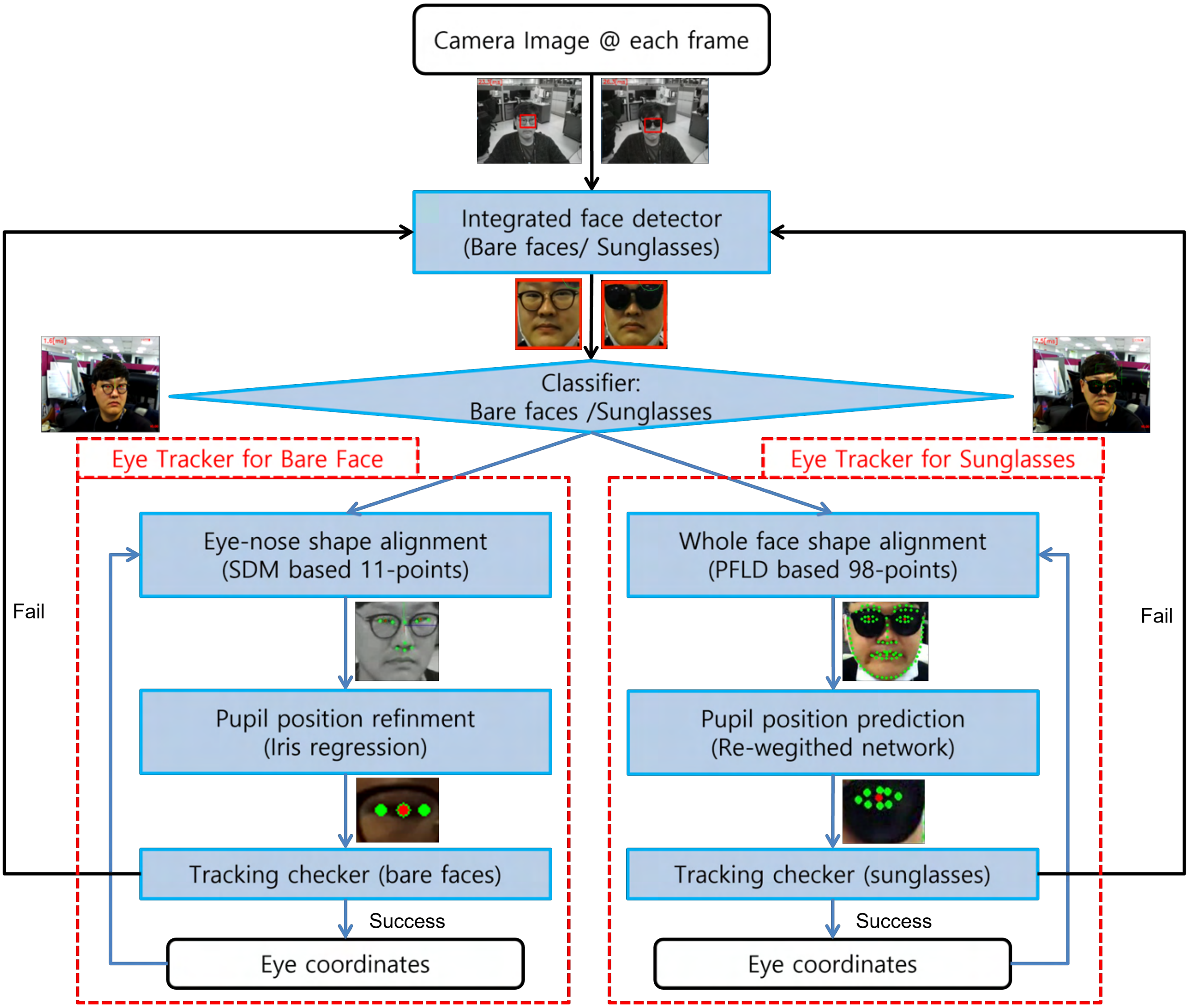

Camera Image @ each frame
Integrated face detector (Bare faces/ Sunglasses)
Classifier: Bare faces /Sunglasses
Eye Tracker for Bare Face
Eye Tracker for Sunglasses
Eye-nose shape alignment (SDM based 11-points)
Whole face shape alignment (PFLD based 98-points)
Fail
Fail
Pupil position refinment (Iris regression)
Pupil position prediction (Re-wegithed network)
Tracking checker (bare faces)
Tracking checker (sunglasses)
Success
Success
Eye coordinates
Eye coordinates

FIGURE 2. Overview of our proposed pupil tracking method. The algorithm works in two different modes for bare and sunglasses faces.

this paper, we take PFLD as the base architecture of our method and add a new stage to improve the model with a re-weight module. Other methods also utilize multiple stages for facial landmark detection [28]– [30], but generally they extract the features from the original image or image patch at each stage, which is time consuming. In contrast, the two stages in our method share the same feature maps, which significantly reduces time costs. Our method computes the weights of different feature map locations with the re-weight module. Wu et al. [31] also assessed the importance of different feature map locations. However, they predicted an edge map that requires a large time cost, whereas our method achieves higher speed by sharing the feature maps across the two stages.

Occlusion in Facial Landmark Detection. We propose a re-weight module to address occlusion problems, such as wearing sunglasses. Occlusion represents a significant challenge in facial landmark detection, both [32] and [33] utilized occlusion labels to perform occlusion inference. However, in many datasets occlusion labels do not exist, so this method is not suitable for certain conditions. Wu et al. [31] imported the information propagation in the edge map prediction to infer the occluded edge map, and used the recovered edge map to address the occlusion problem. As the information propagation is performed along a tree in a bi-directional manner, more time is consumed compared to our method. Our method infers confidence directly from the complete feature appearance and shape. The relations between different facial components that can be used to infer the occlusion condition are embedded in the network. Furthermore, the feature weight is also computed with an attention mode, which allows more discriminative areas to obtain larger weights, this, in turn, is useful for accuracy improvement.

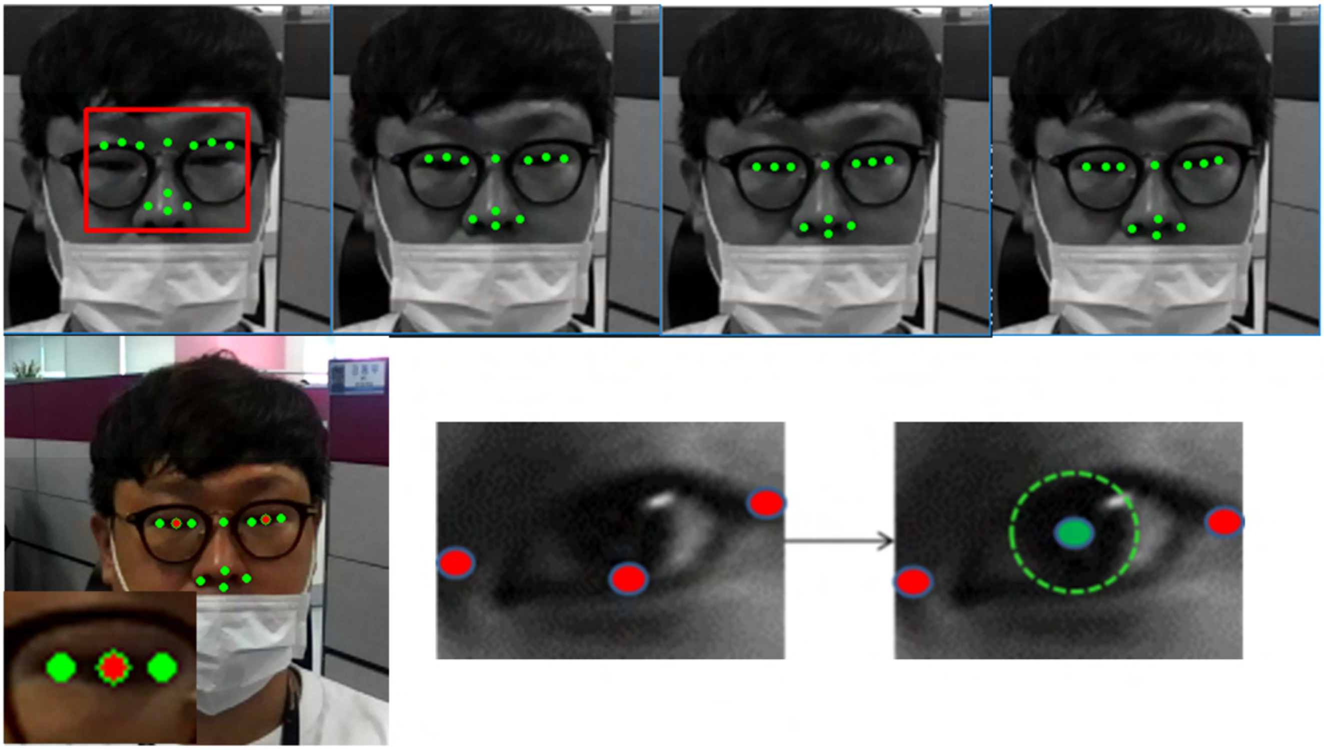


FIGURE 3. Pupil center positions refined with the proposed iris regression method for bare faces. The top row shows the 4-step SDM-based 11-point eye-nose regression. The bottom row shows further pupil position refinement via iris regression. The red dots in the center image of the bottom row indicate initial SDM results, the green dot in the bottom right image indicates the refined pupil center obtained via iris regression.

III. METHODS

Our proposed method aims to provide a pupil tracking tool for commercial assistive driving systems. In order to be suited for commercial use, it is required that the tool be fast, accurate, robust, and has a small model size. Our method satisfies these four requirements in terms of design and implementation. The basic components of our proposed pupil tracker can be divided into two main stages: (1) face detection from RGB webcam images and (2) position tracking of the pupil center from the detected face region. The system flowchart is shown in Figure 2. As Figure 2 shows, first, we perform face detection on a given RGB frame, then we perform pupil center detection on the detected face image. We use our previously developed error-based learning (EBL) method to detect face regions [9], [10], where the conventional cascaded Adaboost classifier [34], [35] with multiblock local binary pattern (LBP) features [36] were used. The EBL scheme trains only a small subset of detection training image DBs with a large size in much shorter training times, while accuracy is improved through three stages. Our proposed detector is simple and practical requiring only a CPU in AR 3D HUD systems, which adopt commercial vehicle-embedded computing boards with limited GPU resources. Details of the algorithms were published in [9], [10] and are not repeated here. If eye center detection is successful, the algorithm runs in tracking mode; if not, the algorithm repeats the face detection stage. In this way, we can achieve high speed with tracking mode while guaranteeing the algorithm's robustness with failure recovery. As bare faces are very different from sunglasses faces due to occlusion, corresponding eye trackers are applied based on image classification. Our system classifies human faces into bare faces and sunglasses faces and performs pupil tracking in two different ways accordingly.

A. FAST AND ACCURATE EYE TRACKING WITH IRIS REGRESSION ON BARE FACES

For bare faces, we use a coarse-to-fine strategy to infer the pupil center. First, we perform a SDM-based 11-point eye-nose landmark alignment, a process that is fully described in our previous study [9], [10]. In this paper, we propose an additional module for pupil center position refinement: an iris regression method that regresses both the pupil center and iris circle (Figure 3). From the initial eye positions from the SDM method, a regression-based pupil refinement algorithm is executed to obtain the pupil center and iris circle on the cropped eye areas. On the normalized eye images, the pupil center and iris circle regressions are performed simultaneously, where they share the SIFT features to achieve high speed. For each iteration, the SIFT features around the three eye corners and center landmarks are extracted first, which are concatenated to form a feature vector v. Then, we obtain the new landmark positions and iris radius by performing a regression on v. Assuming a regression matrix H, landmarks

P_i^k at iteration k for i = 0, 1, and 2, and an iris radius r, the new landmark and radius computation is defined as

$$\begin{bmatrix} P_0^k \\ P_1^k \\ P_2^k \\ r \end{bmatrix} = \begin{bmatrix} P_0^{k-1} \\ P_1^{k-1} \\ P_2^{k-1} \\ 0 \end{bmatrix} + Hv. \tag{1}$$

After experimenting with a different number of iterations, k = 2 was determined. With the regressed iris boundary, we can refine the pupil center by matching the estimated and real boundaries. Additionally, by segmenting the iris area with the regressed circle and center, we can utilize the iris information to provide more functions, such as personal identification for user-specific driving services.

B. RE-WEIGHTED NETWORK FOR EYE TRACKING ON SUNGLASSES FACES

Sunglasses faces need to be treated differently than bare faces as the features around the eyes are difficult to identify. Thus, we cannot infer the pupil center as detailed as in the bare faces case. To address sunglasses occlusion, we utilize the features of the whole face to infer the pupil center. In this paper, we utilize PFLD as the base architecture, which is a fast and robust landmark detection method with MobileNetv2 as its backbone, to detect 98 whole facial landmarks. Because eyes are occluded by sunglasses, we estimate the pupil positions from other shapes. There are tight relationships between different facial landmarks, so we can infer the pupil with non-occluded landmarks, such as the nose and mouth. With PFLD, we take the whole face image as input and use fully connection to regress the landmarks. In this way, the connections between different facial components are encoded into the feature maps.

However, with the original PFLD method, different facial areas have the same importance in the prediction process. As the non-occluded areas tend to provide more information for landmark localization, it is better to give different areas different weights. Moreover, because landmarks around the nose and mouth can also be occluded at times, it is difficult to know which facial components are occluded at any given time. To address this problem, we designed a new re-weight sub-network to automatically represent the importance of different pixels on the feature map.

As shown in Figure 4, the proposed re-weight method infers the pixel confidence on the feature map using both the landmark appearance and the graphical structure between landmarks, and the network can be trained end-to-end. The confidence maps w_{struc} and w_{appear} from the two types of information (structural and appearance) are combined via element-wise multiplication. Given feature map F, we obtain a new feature map F' by performing an element-wise multiplication on F and the combined confidence map. Then, the landmark position is obtained by performing a fully connected operation on F'. With the new landmark, we can compute the confidence map again in the next iteration. That is, take the newly predicted landmarks as input to the re-weight module and infer w_{struc} again. However, as more iterations consume more time, we only perform the iteration once in this paper. When some landmarks are occluded, the features of these landmarks are corrupted and tend to be predicted inaccurately. According to the graph formed by the computed landmarks, we give smaller confidence to inaccurately predicted landmarks. Moreover, different facial component appearances have different discriminative abilities. With this information, we can automatically select more discriminative areas with w_{appear} for the following regression process.

The re-weighted network contains two steps. Step 1 can be considered as the original PFLD. Here, we define loss 1 and loss 2 in the same manner as the original PFLD loss. When the head pose variation is not significant, pose estimation in the loss can be omitted. To guide the inference of w_{struc}, we add another loss to assign inaccurately predicted landmarks a smaller weight, which represents Step 2. Here, we define the ground truth confidence for landmark k in loss 3 as follows:

$$e_k = exp\left(-\beta\left\|\tilde{\alpha} - \alpha_k\right\|_2^2\right), k = 1, \dots, K, \tag{2}$$

where α_k is the predicted landmark in Step 1 (the original PFLD), $\tilde{\alpha}$ is the landmark ground truth, and β is a constant (β = 10 in this paper). Given the predictions in Step 1, we compute the error for each landmark and use it to define the landmark confidence (w_{struc}) for the re-weight module. For convenience, we perform fully connection directly on the concatenated 98 landmark locations, while the graphical structure between landmarks is embedded in the concatenated locations.

For the proposed re-weighted sub-network, the three multi-scale feature maps from the original PFLD [13] were utilized, which have 1x1, 7x7, and 14x14 feature map sizes. Because the top two layers, i.e., 1x1 and 7x7, have a small feature map size, we only computed the combined weight on the third 14x14 layer. The top two layers are concatenated with the third layer directly. When computing w_{struc}, the predicted landmark locations can be concatenated with the feature map to act as the input. However, in order to reduce time consumption, the feature map is not used as input for w_{struc} in this paper.

C. TRACKER-CHECKER

At each frame, after eye center prediction finishes, the initial face region for the next frame tracking can be decided using the aligned face points without face detection, which takes additional time. The tracker-checker module assesses tracking correctness by filtering out the misaligned and non-facial areas. However, the proposed eye tracker restarts the face detection processes from the start only when the tracker-checker determines that final tracking has failed. To address this problem, we propose to add tracker-checker modules after the landmark prediction process. For bare faces, we extract SIFT features around 11 landmarks and perform SVM

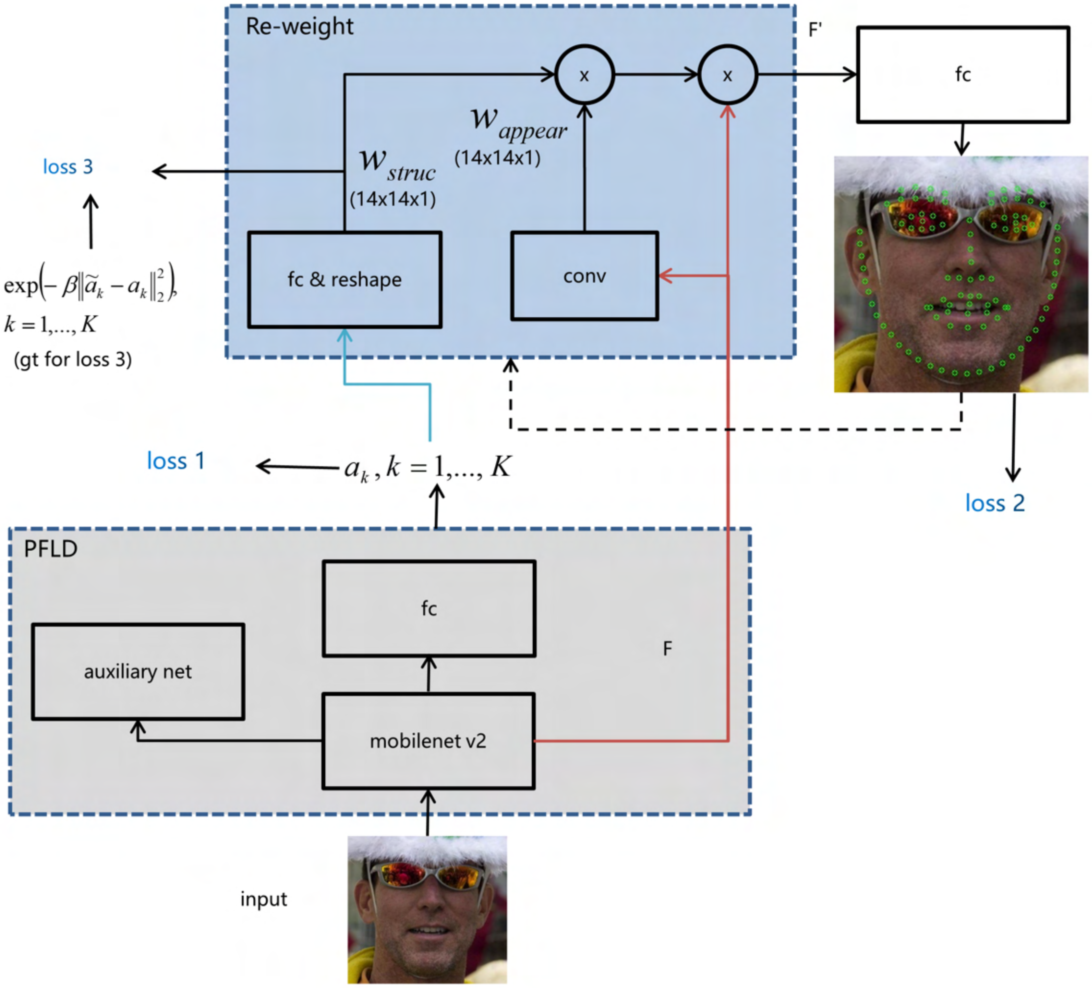


FIGURE 4. Pupil center positions are refined with the proposed re-weight method for sunglasses faces. From the original PFLD method results (the bottom raw), the re-weight method infers the pixel confidence on the feature map using both the landmark appearance and the graphical structure between landmarks as shown in the top row.

to check if tracking succeeds, this process was described in our previous study [9], [10].

For sunglasses faces, as the features around the eyes are corrupted, the tracker-checker needs to use the features of the whole face. The proposed tracker-checker utilizes the feature maps obtained in PFLD. From the last feature map for regressing the 98 landmarks points in PFLD, we directly perform convolution on this feature map to obtain the score representing the current tracking success confidence. In the training stage, we define a cross entropy loss for the predicted tracking confidence score. The error between the predicted landmarks and the ground truth is utilized as the ground truth for the defined cross entropy loss (Figure 5). In this way, if the landmark prediction is bad, the confidence score is small and the tracking is considered a failure. To train the tracker-checker, we sample a number of face images around the real face area as training samples. Because the tracker-checker shares feature maps with PFLD, time consumption can be significantly reduced. The time cost of the tracker-checker is small, less than 0.5 ms. Training the proposed sunglasses tracker-checker utilizes the trained re-weight model described in Section 3.2, and the re-weight model weights remain unchanged during the tracker-checker training process.

TABLE 1. Performance of the proposed method on various image DBs. The training sample number, testing sample number, pupil center mean error, and time consumption are shown.

	Training image DB	Testing image DB	Precision (mm)	Speed (ms)
Bare faces (RGB)	11,000	20,000	1.5	4
Sunglasses faces (RGB)	37,000	5,000	6.5	10

IV. EXPERIMENTAL RESULTS

The proposed algorithm was implemented with C++ and tested based only on CPU computations on both a Windows PC and a commercial embedded computing system running Linux. It yielded successful tracking results on both bare

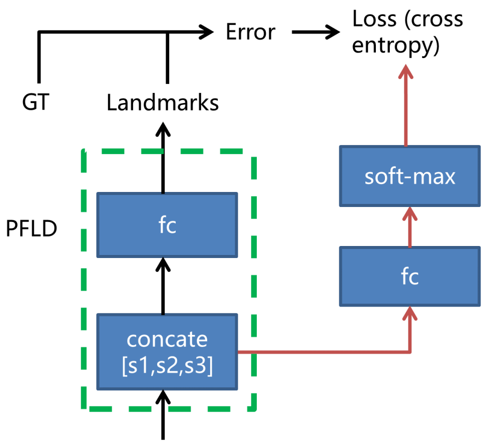


FIGURE 5. Flowchart of the proposed PFLD-based tracker-checker for sunglasses faces. The red arrows indicate the tracker-checker confidence calculation process. The cross entropy loss for training is calculated with the errors from the PFLD alignment.

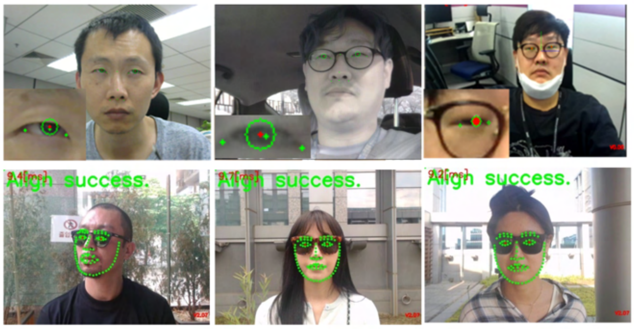


FIGURE 6. Experimental results on bare faces with 11 eye-nose landmark points with iris regression (top row), and sunglasses faces with 98 whole facial points with eye position refinement via feature map re-weighting (bottom row). For bare faces (top row), the greed dot inside the pupil circle indicates the initial eye position from SDM and the red dot indicates the refined eye center position using the proposed iris regression algorithm. For sunglasses faces (bottom row), the red dots indicate the estimated pupil position and the green dots indicate the remaining facial points.

(200 fps) and sunglasses (100 fps) faces on a 2.0 GHz CPU. We evaluate our algorithm precision by calculating the distances between the ground truth and the tracked pupil centers, whereby the pixel distances were converted to physical distances based on the assumption that the inter-pupil distance (IPD) was 65mm. In the experiment, our method achieves high accuracy and speed, approximately 1.5 mm error at less than 5 ms for bare faces and approximately 6.5 mm error at less than 10 ms for sunglasses faces on the 2.0 GHz CPU. These results are summarized in Table 1. The training set contains 10,000 face RGB images from the public Wider Facial Landmarks in-the-wild (WFLW) [31] DB and 27,000 captured RGB images from our own DB. All 25,000 images in the evaluation dataset (20,000 for bare faces and 5,000 for sunglasses faces) were captured both indoors and outdoors to reflect multiple lighting conditions (Figure 6).

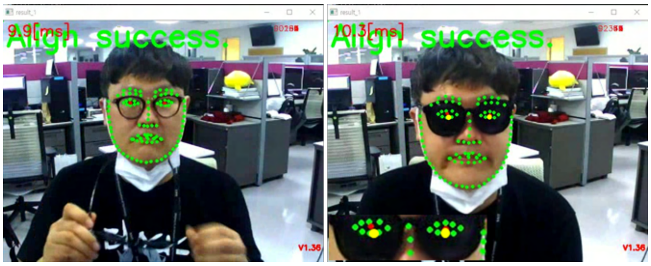


FIGURE 7. Examples of comparison of the predicted pupils from the proposed algorithm and the ground truth points on sunglasses faces. All the ground truths of eye positions were annotated as precisely as possible via comparison with the corresponding bare face images (left). Yellow points indicate the ground truth points of pupil center positions and red points indicate the predicted pupil center positions by the proposed algorithm (right).

For sunglasses faces, all the ground truths of eye positions for evaluation were annotated as precisely as possible via comparison with the corresponding bare face images (Figure 7). Also different publicly available DB such as 300-W [37], AFLW [38], and COFW [32] can be adopted in our algorithm training and evaluation. The proposed method can detect and track the pupil center accurately in real-time on both bare and sunglasses faces from a RGB stereo camera. The RGB camera image resolution was 640x480 with a capturing speed of 60 fps, a 60x40° field-of-view, and the distance between the camera and users ranged between 70–400 mm. When considering the 12-mm 3D HUD crosstalk margin and the limited system environment in vehicles, the accuracy achieved by our system is determined to be quite suitable for commercialization. The proposed algorithm processing time for each frame gets correspondingly increased when using higher resolution and wider field-of-view cameras, which capture a wider area for detection and tracking.

A. EXPERIMENTAL RESULTS ON BARE FACES

We tested our method on various bare face datasets, including synthetic, near infrared, high resolution RGB, and normal resolution RGB eye images. Additionally, we also tested our method on video sequences. To test the accuracy of the proposed iris regression method, we added random noise to three eye landmarks (outer corner, pupil center, and inner corner). The noise conforms to a uniform distribution in the range of [-3, 3 mm] for both the x and y coordinate. The noisy landmarks were given as initial landmarks. For the videos, we utilized the complete process, including face detection, 11-landmark SDM, pupil segmentation, and tracker-checker.

1) Eye Image DB Evaluation for Iris Regression

Synthetic eye images. To test the performance of our method on synthetic eye images, we generated 5,000 eye images with the Unity Eye tool and separated them into 4,000 training samples and 1,000 testing samples. For synthetic eye images, the annotation is accurate, which is useful for improving the method's performance. These results are shown in Table 2 and Figure 8a, where we can see that the proposed iris

TABLE 2. Iris regression results on (a) synthetic eye images, (b) NIR eye images, (c) high resolution RGB images (1920x1080) at 1-m distance, and (d) normal RGB images (640x480) at 1-m distance.

Iteration	Good Rate (<3mm)		Mean Error (mm)		Median Error (mm)		Speed (ms)	
	Synthetic	NIR	Synthetic	NIR	Synthetic	NIR	Synthetic	NIR
4	100%	100%	0.31	0.57	0.27	0.54	3.16	3.17
3	100%	100%	0.31	0.55	0.27	0.53	2.78	2.65
2	100%	100%	0.34	0.54	0.31	0.51	2.09	2.09
1	100%	100%	0.67	0.74	0.62	0.68	1.54	1.58

regression method can achieve high accuracy on synthetic eye images, despite varying textures and reflection occurring within the pupil area. Furthermore, the regressed iris circle is in agreement with the actual iris circle. This is useful for other applications, such as iris recognition. Finally, our method achieves a mean error of 0.317 mm after 4 iterations.

NIR eye images. To test our method on NIR eye images, we constructed a NIR DB with 2,078 training samples and 500 testing samples. These results are shown in Table 2 and Figure 8b, where we can see that the detected pupil center stays accurately within the pupil area, we also obtain accurate regressed iris circles. In this case, the mean error decreases sharply after the initial two iteration steps and increases slightly later due to annotation errors. Thus, we set the iteration times to 2 in all our experiments when handling bare faces.

High resolution RGB eye images. In our experiment, we found that method performance is highly dependent on image resolution. To test the method accuracy, we used two RGB datasets, high and normal resolution RGB images. The high RGB DB includes 11,832 training samples and 20,000 testing samples. Results show that, although the iris area is occluded by eyelid and eyelash, our method is still able to regress both the pupil center and iris circle accurately (Figure 8c). After two iterations we obtain a mean error of 0.417 mm. Based on these results, we determined that our method is suitable for applications requiring high accuracy, such as autostereoscopic 3D displays.

Normal resolution RGB eye images. For the normal resolution RGB eye images we used the Samsung Advanced Institute of Technology (SAIT) DB. We collected 2,044 training samples and 500 testing samples. The mean error after two iterations is 1.03 mm, which means that accuracy is indeed influenced by image resolution. For the high-resolution eye images, IPD is approximately 604 pixels, while it is only 74 pixels for normal-resolution images. High IPD causes high annotation accuracy, and vice versa. Consequently, it is hard to annotate the pupil center accurately for normal-resolution eye images. For an IPD of 74, 1 mm is only 1 pixel. Nonetheless, our method achieves high accuracy under low-resolution conditions (Figure 8d).

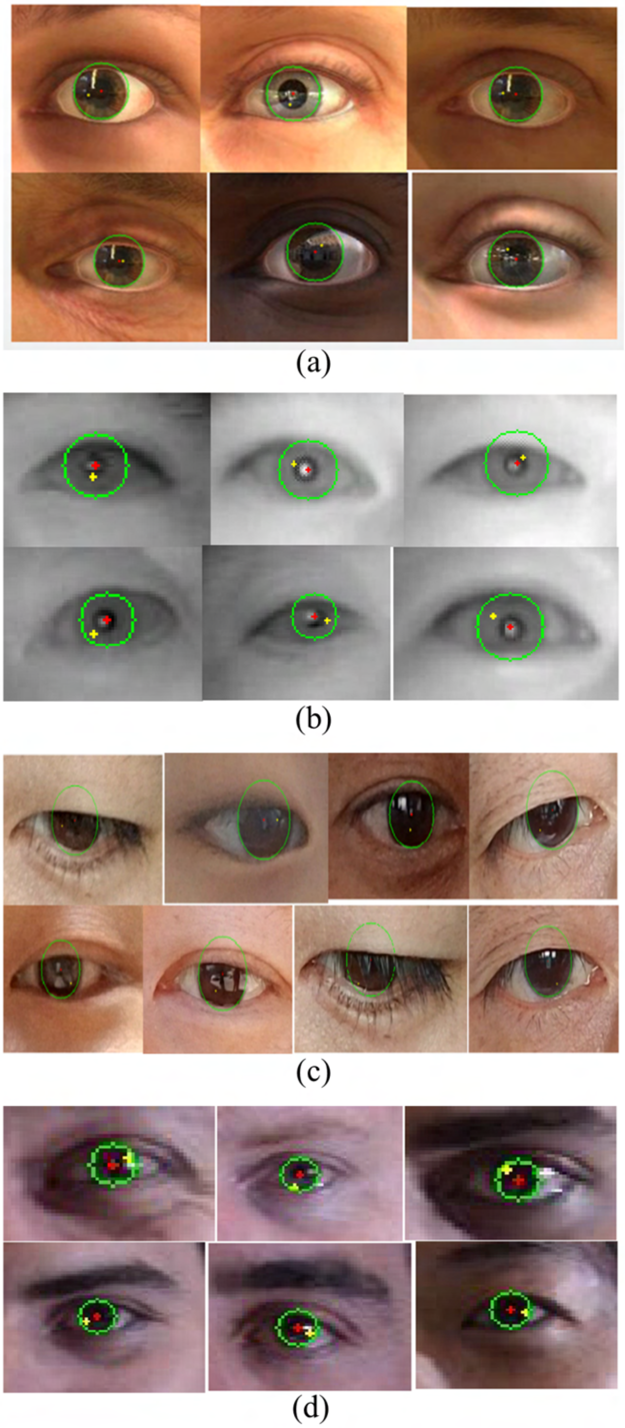


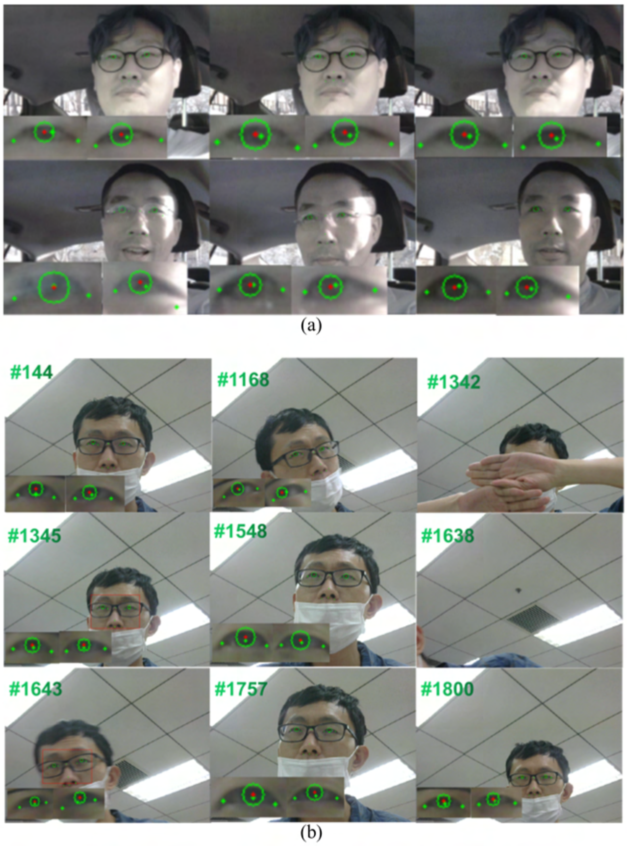
FIGURE 8. Iris regression results on (a) synthetic eye images, (b) NIR eye images, (c) high resolution RGB images (1920x1080) at 1-m distance, and (d) normal RGB images (640x480) at 1-m distance.

2) Face Video DB Evaluation for the Entire Pupil-Tracking Process

We tested the iris regression method on two videos taken while subjects were driving a vehicle. The results are shown in Table 3 and Figure 9a. The mean error is relatively larger

TABLE 3. Performance of the pupil-tracking s on two videos while driving on bare faces with proposed iris regression algorithm.

DB	Method	Good Rate (< 3 mm)	Mean Error (mm)	Median Error (mm)
DB #1	11-landmark SDM [9]	79%	2.19	2.21
	Proposed method (iris regression)	87%	1.70	1.40
DB #2	11-landmark SDM [9]	75%	2.17	2.13
	Proposed method (iris regression)	87%	1.50	1.48



(a)

(b)

FIGURE 9. Pupil-tracking results on bare faces videos (640x480 webcam) using the complete tracking system in a vehicle (a) and in a room (b). The green dots indicate the initial SDM eye alignment results and the red dots (and green circle) show refined pupil position results via iris regression.

than on the normal resolution RGB eye images. This is because the image styles differ greatly from the training set. These results could be improved by re-training the model on this image style. However, as Figure 9a shows, the pupil center is still located within the iris area considering the iris radius is approximately 3 mm (our method is 1.5 mm). This could meet the requirements of assistive driving. Furthermore, aided by the coarse-to-fine strategy, the proposed iris regression method can obtain higher accuracy than only using the 11-landmark SDM, as shown in Table 3. We also tested a video taken in a room and compared the proposed iris regression method with the 11 landmark points SDM (Figure 9b). In this case, the person moves his head to various poses.

TABLE 4. ION error comparison regarding the re-weight module on the WFLW DB.

Method	ION Mean Error
Original PFLD [13]	7.61
Proposed method with w_{struc}	7.52
Proposed method with w_{appear}	7.53
Proposed method with w_{struc} and w_{appear}	7.43

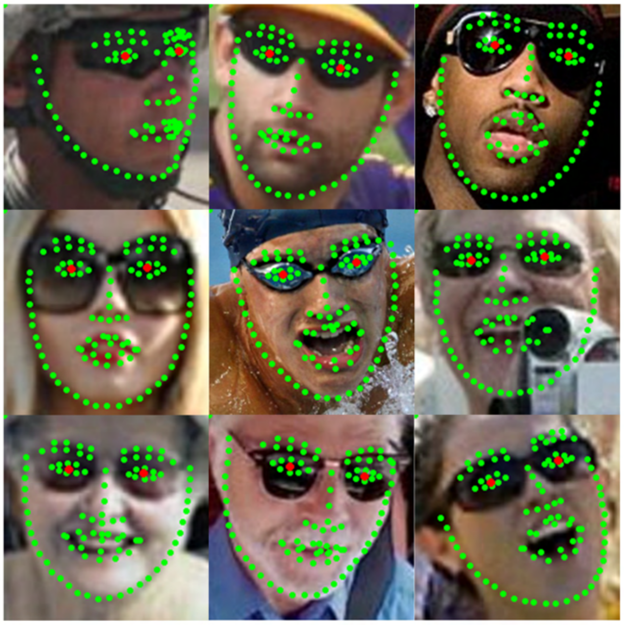


FIGURE 10. Proposed re-weighted PFLD algorithm results on sunglasses faces from a public image DB, WFLW.

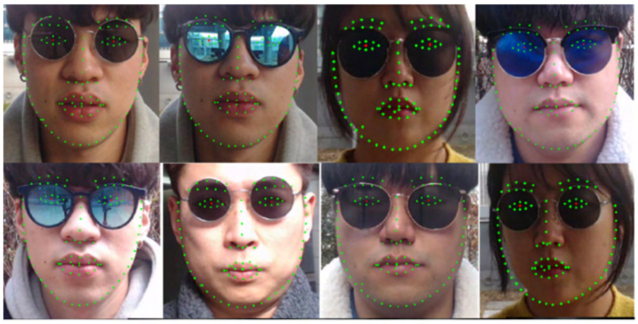


FIGURE 11. Proposed re-weighted PFLD algorithm results on sunglasses faces that were collected for this study.

The proposed iris regression method can regress the pupil center and iris circle accurately, while the 11-landmark SDM obtains the pupil center departing from the real one to a large extent. In the experiment, we also tested the tracker-checker. We can observe that our tracker recovers easily from tracking failure, e.g., frames 1342 and 1345 in Figure 9b.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3110644, IEEE Access

B. EXPERIMENTAL RESULTS ON SUNGLASSES FACESS

We also tested our method on sunglasses faces. Here, we infer the pupil centers with a revised PFLD method with the re-weight algorithm. First, we compare our method with the original PFLD to demonstrate the validity of the proposed re-weight sub-network, then the experimental results on the sunglasses faces image and video DBs are presented.

1) Validity of the Re-Weighted Network

We made an ablation experiment to test the validity of the proposed re-weighted network by testing a public DB, namely, WFLW. To improve the training efficiency, we first train the original PFLD for 300 epochs and then use it as a pre-trained model to train the re-weighted network for another 48 epochs. We tested the effect of using different re-weight strategies. These results are summarized in Table 4. We can see that using both w_{struc} and w_{appear} obtains the smallest Inter-Ocular Normalization (ION) mean error for the 98 points. With the re-weight module, our method assigns more discriminative and more confident areas larger weights and obtains better result. Using only either w_{struc} or w_{appear} also obtains better result than the original PFLD [13]. The weight w_{struc} defines the feature map weight according to the landmark prediction error. The occluded landmarks tend to have large prediction errors and are consequently given smaller weights according to the structural relationships between landmarks. Non-occluded landmarks tend to have accurate predictions, therefore, they are given larger weights. In this way, we can infer the occluded landmarks with non-occluded ones. The weight performs similar to an attention model. It finds the most discriminative areas in the feature map during the training process, which is useful to improve accuracy. By combining w_{struc} and w_{appear}, the proposed re-weight module takes advantage of both of them and can further improve the detection accuracy. Figure 10 shows some results of our method of handling sunglasses faces from the WFLW DB.

2) Results on the sunglasses faces image and video DBs

We also made experiments on image and video DBs that were collected by us to demonstrate the validity of the proposed network when handling sunglasses faces under various circumstances. Among our collected RGB face image DB, 27,000 images were utilized for training with 10,000 WFLW DB images. The remaining 5,000 images were used as the testing set (Figure 11). When using the original PFLD 0.25X algorithm trained on the WFLW DB, the mean pupil center error on the evaluation dataset was 10 mm. The proposed re-weighted method with training on both the WFLW DB and our own image DB achieves a mean pupil center error of 6.5 mm. When considering the 12-mm 3D HUD margin, our proposed method is determined suitable for commercial driving applications.

Our proposed method was also validated on a video DB taken with a live webcam while wearing sunglasses, as shown in Figure 12. The mean error is approximately 10 mm for a 640x480 stereo webcam. The eye positions from the stereo camera were utilized for calculating the 3D eye coordinates via stereo matching. The speed was less than 8 ms on a Windows PC with a 2.9 GHz CPU. We also implemented the algorithm on a Samsung Exynos-auto KITT with a 2.0 GHz CPU. As mentioned before, our algorithm only utilized CPU. The performance on this computational environment was the same 10-mm precision error with a speed of 10 ms for a 640x480 stereo webcam. When considering the 12-mm 3D HUD margin, our achieved accuracy and speed are quite suitable for commercial applications.

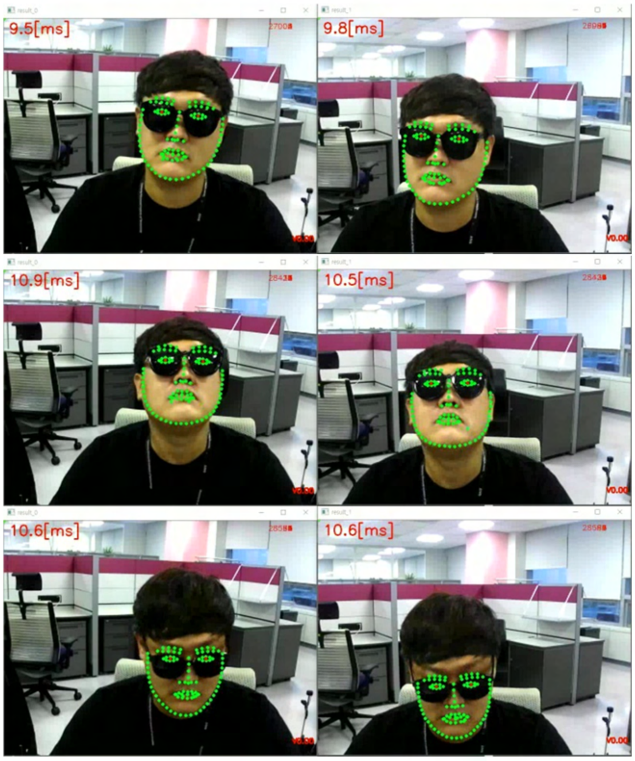


FIGURE 12. Eye-tracking results on sunglasses faces videos (640x480 stereo webcam). The left column indicates the left camera views and the right column shows the right camera views from the stereo webcam. The algorithm calculates the 3D pupil coordinates from these two pupil points from the stereo camera.

V. DISCUSSION

The proposed method demonstrated high accuracy and fast speed regarding eye center position tracking when users were with bare faces or wore sunglasses under various environments. Note that our system does not save images but extract eye center coordinates from RGB cameras. In this way, the proposed eye tracking method prevents privacy leakage. The algorithm was evaluated on various image and video datasets, including public datasets and our own captured datasets. The indoor environment datasets target personal computer monitors and tablets at home, as well as actual driving conditions targeting AR 3D HUDs in vehicles. With a coarse-

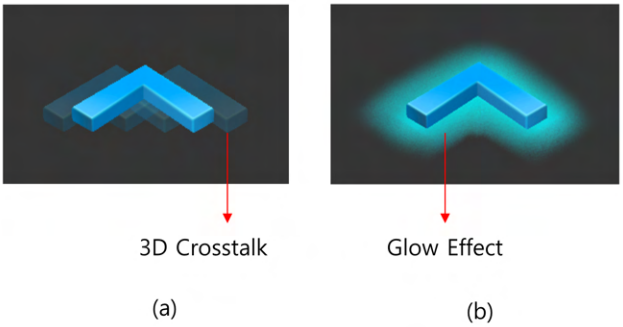


FIGURE 13. (a) A 3D content example with 3D crosstalk, and (b) the glow effects added in the 3D content which decrease 3D crosstalk feeling.

to-fine strategy on bare faces, the proposed iris regression algorithm refined its precision to a 1-mm mean error from the 3-mm mean error of the initial SDM-based eye position alignment. When the performances of our eye tracking method for bare faces (1.5 mm mean error) and sunglasses faces (6.5 mm mean error) are compared, the eye-tracker aligner mean error precision for sunglasses faces decreased from 1.5 to 6.5 mm. When the 3D margin (12 mm) of our AR 3D HUD prototype was considered, the results generated for both bare and sunglasses faces were still determined to be reasonable for commercial applications. Additionally, in our AR 3D HUD prototype, different graphic contents were applied depending on whether the user was wearing sunglasses or not. Because the sunglasses eye-tracking mode has lower precision according to the smaller 3D margin than the bare faces eye-tracking mode, head movements by the user can cause 3D crosstalk (Figure 13a). To handle this dynamic 3D crosstalk, glow effects were applied to the 3D contents for the sunglasses eye-tracking mode, which lower the 3D crosstalk effect, as shown in Figure 13b. Table 5 gives a detailed specification of our AR 3D HUD prototype and the related 3D crosstalk experiments. When compared to bare faces, pupil tracking for sunglasses faces showed similar 3D crosstalk user experiences, i.e., almost no 3D crosstalk feeling, when head movement speed is less than 250 mm/s. However, when head movements are greater than or equal to 250 mm/s, users wearing sunglasses felt 3D crosstalk when seeing the 3D content (Figure 13a). When the glow effects were added to the 3D contents (Figure 13b and Contents 2 in Table 5), users wearing sunglasses did not feel the 3D crosstalk experience. These results highlight the possibilities for AR 3D HUD commercialization, with providing low crosstalk 3D image experience even when the head movements of the users are greater than 250 mm/s.

A. COMPARISON WITH EXISTING APPROACHES

Compared to previously published works, the main development of our method is user-condition aware, whereby different eye trackers were applied according to user conditions, such as bare faces and sunglasses faces. The proposed method can be extended to various face occlusion cases, such as wearing masks and hats if such cases are properly trained. Additionally, to prevent erroneous tracking, a novel tracker-checker idea was proposed. Different tracker-checkers were utilized under different user conditions. For bare faces, SVM is utilized as tracker-checker, which acts on extracted SIFT features around predicted landmarks [9]. For sunglasses faces, we defined a new cross entropy loss of 98 whole face landmark points for the predicted tracking confidence. Experimental results demonstrated the validity of the proposed tracker-checkers for bare and sunglasses faces, which were faster than 1 ms per frame. With the proposed tracker-checker methods, the eye tracking system runs in real-time faster than 100 fps (10 ms) without performing face detection on every frame.

Many studies attempted automatic facial landmark point localization and tracking with state-of-the-art deep learning techniques and achieved high accuracy. Even though state-of-the-art deep neural network-based methods achieved significant improvements recently, they suffer from increased computational resource consumption, including Graphics Processing Unit (GPU) and lower speed, especially in limited resource systems, such as mobile devices and automobile systems. This study is also based on facial landmark point alignment techniques, but with explicit priority on pupil center position accuracy. Previous methods achieved practical pupil center position localization and tracking, whereby our proposed methods are based on both classical and recent deep learning-based methods. The proposed methods are switchable depending on user conditions: classical SDM for bare faces and deep learning-based PFLD for sunglass faces. Both methods utilize only CPU computations, whereby the bare and sunglasses faces modes require 10% and 20% CPU usage, respectively. The performance comparison between our proposed algorithm and state-of-the-art deep learning techniques, specifically ESR [17], CFSS [39], DVLN [40] and LAB [31], are listed in Table 6. For comparison purposes, the original results in the papers were used. Our user-conditional eye tracking algorithm achieved higher speed for both bare and sunglasses faces with only CPU computations. While LAB [31] showed higher precision than our method, it requires 2.6 s with CPU or 60 ms with GPU for its accurate landmark alignment algorithm. When considering the increasing GPU computations of AR algorithms, our proposed CPU method is important and highly beneficial for AR 3D HUD systems even with the enhanced system hardware resources in the future.

Figure 14 displays additional results obtained with a public database, 300-W [37]. The faces include bare faces and sunglasses faces under different poses, lighting conditions, and expressions.

VI. CONCLUSION

In this paper, we propose a new pupil center tracking system that handles both bare faces and sunglasses faces. For bare faces, we propose a SDM-based iris regression method and utilize a coarse-to-fine strategy. The iris center and iris

TABLE 5. AR 3D HUD prototype specification and the 3D crosstalk experiments with the proposed eye-tracking algorithms on bare and sunglasses faces.

	Bare Face Eye-tracker (eye-nose)	Sunglasses Face Eye-tracker (whole face)
Tracked Shape Points	11 Eye Nose Points	98 Whole Face Points
Pupil Precision (mm)	1.5	6.5
Distance between camera and users (cm)	80 to 120	80 to 120
Speed (ms)	4	10
HUD HW	AR 3D HUD prototype	
System	Samsung Exynos-auto (2.0 GHz CPU)	
Camera	S640x480 @100 fps Stereo-camera	
Contents 1	3D Arrow Contents (no Glow Effects)	
when head movements $<$ 250 mm/s	3D crosstalk (x)	3D crosstalk (x)
when head movements $\geq$ 250 mm/s	3D crosstalk (x)	3D crosstalk (o)
Contents 2	3D Arrow Contents with Glow Effects	
user head movements $<$ 250 mm/s	3D crosstalk (x)	3D crosstalk (x)
user head movements $\geq$ 250 mm/s	3D crosstalk (x)	3D crosstalk (x)

TABLE 6. Performance comparison between previous studies and the proposed algorithm on the WFLW test set (98 landmarks). For the proposed method for bare faces, we used 11 landmarks for evaluation.

Method	Precision (ION Mean Error)	Speed (per frame)
ESR [17]	11.13	15 ms (CPU)
CFSS [39]	9.07	40 ms (CPU)
DVLN [40]	10.84	15 ms (CPU)
LAB [31]	5.27	2.6 s (CPU)
	5.27	60 ms (GPU)
Proposed method (sunglasses faces)	7.43	10 ms (CPU)
Proposed method (bare faces)	1.71	4 ms (CPU)

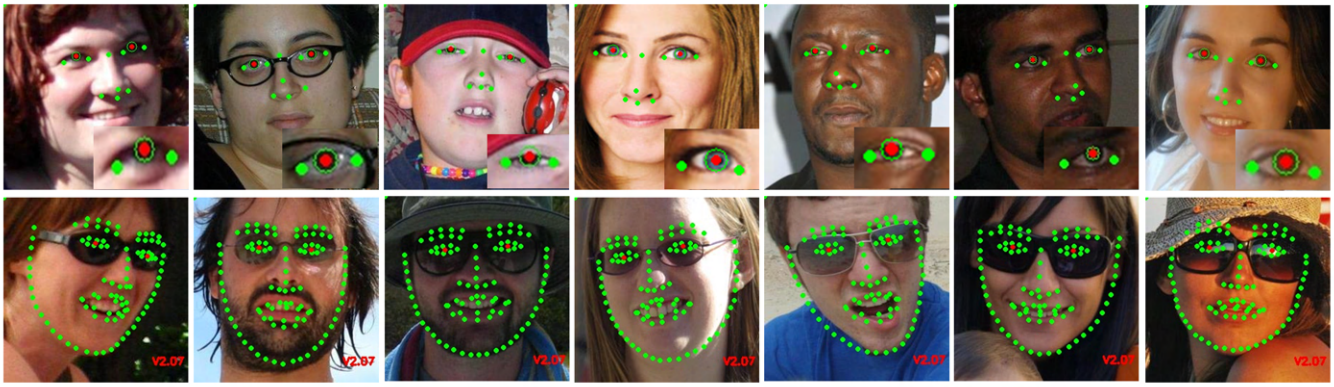


FIGURE 14. Qualitative results on several challenging faces on a public DB, 300-W, obtained with our proposed user-conditional eye-tracker. The red dots indicate the eye centers and the green dots indicate other shape points. The top row shows the bare faces results and the bottom row shows the sunglasses face results.

circle are regressed at the same time with our method. For sunglasses faces, we added a re-weight module to the original PFLD network and can identify the pixel importance on the feature map effectively. This is useful for dealing with sunglasses occlusion. A new cross entropy loss-based tracker-checker is also provided, which is robust and fast. Besides these contributions, we also construct an eye tracking system with face detection, face type classification, facial landmark detection, and tracker-checker modules. Our system is fast, accurate, robust, and has a small model size, which are all requirements for AR 3D HUD commercialization.

Despite its many advantages, our study has a few limitations. While our proposed method can handle cases where the pupil's shape is obstructed, such as when the user is wearing sunglasses, our method requires two different eye-tracking models, which exhibit different performances. Specifically, the sunglasses face eye-tracker has decreased eye center tracking precision and speed when compared to the bare faces eye-tracker. The sunglasses faces eye-tracker speed could be increased by using the deep neural network compression method or by pruning and quantization [41], [42], among other methods. Additionally, wearing sunglasses yielded limited precision in eye alignments because pupil locations were estimated with other shapes owing to the invisibility of the pupils. In our study, different types of sunglasses such as polarized lenses and anti-reflective lenses were included in the training and testing database. However, the algorithm was not examined on various types of sunglasses. Further study on the eye tracking approaches for various sunglasses is needed. To increase the instances at which the sunglasses faces eye-tracker can be used with satisfactory results, personalized pupil tracking technologies will be studied in future work. Also, privacy-preserving crowdsensing in vehicular networks scheme [43]– [46] can be combined with our algorithm in eye tracker model training update way, by aggregating various challenging cases in real driving. By outsourcing various challenging cases from each driver through a privacy-preserving network, the bare and sunglasses face eye tracker can handle various challenging cases by retraining the models with the additional image datasets in the platform layers. Finally, a deep study on parameter optimization in deep learning-based methods may provide a more robust study for eye tracking.

REFERENCES

[1] Cho, Y.H. and Nam, D.K., "Content visualizing device and method," U.S. Patent 10573063B2, Feb. 25, 2019.

[2] GMartinez, L.A.V. and Orozoco, L.F.E., "Head-up display system using auto-stereoscopy 3d transparent electronic display," U.S. Patent 20160073098, March. 10, 2016.

[3] Nam, D., Lee, J., Cho, Y.H., Jeong, Y.J., Hwang, H. and Park, D.S., "Flat Panel Light-Field 3-D Display: Concept, Design, Rendering, and Calibration," *Proc. IEEE*, vol. 105, no. 5, pp. 876–891, Apr. 2017.

[4] Lee, S., Park, J., Heo, J., Kang, B., Kang, D., Hwang, H., Lee, J., Choi, Y., Choi, K. and Nam D., "Autostereoscopic 3D display using directional subpixel rendering," *Opt. Express*, vol. 26, no. 16, pp. 20233–20233, Aug. 2018.

[5] Dodgson, N. A., "Autostereoscopic 3D displays," *Computer*, vol. 38, no. 8, pp. 31–36, Aug. 2005.

[6] Meng Liu, Youfu Li, and Hai Liu, "3D Gaze Estimation for Head-Mounted Eye Tracking System With Auto-Calibration Method," *IEEE Access*, vol. 8, pp. 104207-104215, Jun. 2020.

[7] Andronicus A. Akinyelu and Pieter Blignaut, "Convolutional Neural Network-Based Methods for Eye Gaze Estimation: A Survey," *IEEE Access*, vol. 8, pp. 142581-142605, Jul. 2020.

[8] Wenyu Li *et al.*, "Training a camera to perform long-distance eye tracking by another eye-tracker," *IEEE Access*, vol. 7, pp. 155313-155324., Oct. 2019.

[9] Kang, D. and Heo, J., "Content-Aware Eye Tracking for Autostereoscopic 3D Display," *Sensors*, vol. 20, no. 17, pp. 4787, Aug. 2020.

[10] Kang, D., Heo, J. *et al.*, "Pupil detection and tracking for AR 3D under various circumstances," in *Electron. Imaging,* San Francisco, CA, USA, 2019, pp. 55-1-55-5.

[11] Xuehan X. and De la Torre, F., "Supervised descent method and its applications to face alignment," in *IEEE Conference on Computer Vision and Pattern Recognition,* Portland, OR, USA, 2013, pp. 532–539.

[12] Lowe, D.G., "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91-110, Aug. 2004.

[13] Guo, X., Li, S., Yu, J., Zhang, J., Ma, J., Ma, L., Liu, W. and Ling, H, "PFLD: A practical facial landmark detector," arXiv preprint arXiv:1902.10859, Feb. 2019.

[14] Cortes C. and Vapnik, V., "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[15] Cootes, T.F., Taylor, D., Cooper, D. and Graham, J., "Active shape models-their training and application," *Comput. Vis. Image Underst.*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

[16] Cootes, T., Edwards, G. and Taylor, C., " Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001.

[17] Cao, X., Wei, Y., Wen, F. and Sun, J., " Face Alignment by Explicit Shape Regression," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 177–190, Dec. 2014.

[18] Gou, C., Wu, Y., Wang, K., Wang F.Y. and Ji, Q., "Learning-by-Synthesis for Accurate Eye Detection," in *2016 23rd International Conference on Pattern Recognition (ICPR),* Cancun, 2016, pp. 3362-3367.

[19] CZhang, K. *et al.*, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.

[20] Goodfellow, I., Pouget-Abadie, J., Mirza, M.; Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., "Generative adversarial nets," in *Advances in Neural Information Processing Systems,* Montreal, Canada, 2014, pp. 2672-2680.

[21] Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W. and Webb, R., "Learning from simulated and unsupervised images through adversarial training," in *IEEE Conference on Computer Vision and Pattern Recognition,* Honolulu, HI, USA, 2017, pp. 2107-2116.

[22] Dong, X., Yan, Y., Ouyang, W. and Yang, Y., "Style aggregated network for facial landmark detection," in *IEEE Conference on Computer Vision and Pattern Recognition,* Salt Lake City, UT, USA, 2018, pp. 379–388.

[23] Qian, S., Sun, K., Wu, W., Qian, C. and Jia, J., "Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation," in *IEEE International Conference on Computer Vision,* Seoul, South Korea, 2019, pp. 10153–10163.

[24] Feng, Z.H., Kittler, J., Awais, M., Huber, P. and Wu, X.-J., "Wing loss for robust facial landmark localisation with convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition,* Salt Lake City, UT, USA, 2018, pp. 2235–2245.

[25] Wang, X., Bo, L. and Fuxin, L., "Adaptive wing loss for robust face alignment via heatmap regression," in *IEEE International Conference on Computer Vision,* Seoul, South Korea, 2019, pp. 6971–6981.

[26] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W. and Weyand, T., "Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861., 2017.

[27] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.-C., "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE Conference on Computer Vision and Pattern Recognition,* SSalt Lake City, UT, USA, 2018, pp. 4510-4520.

[28] Trigeorgis, G., Snape, P., Nicolaou, M.A., Antonakos, E. and Zafeiriou, S., "Mnemonic descent method: A recurrent process applied for end-to-end face alignment," in *IEEE Conference on Computer Vision and Pattern Recognition,* Las Vegas, NV, USA, 2016, pp. 4177-4187.

[29] Sun, Y., Wang, X. and Tang, X., "Deep convolutional network cascade for facial point detection," in *IEEE Conference on Computer Vision and Pattern Recognition,* Portland, OR, USA, 2013, pp. 3476-3483.

[30] Lv, J., Shao, X., Xing, J., Cheng, C. and Zhou, X., "A deep regression architecture with two-stage re-initialization for high performance facial landmark detection," in *IEEE Conference on Computer Vision and Pattern Recognition,* Honolulu, HI, USA, 2017, pp. 3317-3326.

[31] Wu, W., Qian, C., Yang, S., Wang, Q., Cai, Y. and Zhou, Q., "Look at boundary: A boundary-aware face alignment algorithm," in *IEEE Conference on Computer Vision and Pattern Recognition,* Salt Lake City, UT, USA, 2018, pp. 2129-2138.

[32] Burgos-Artizzu, X.P., Perona, P. and Dollár, P., "Robust face landmark estimation under occlusion," in *IEEE Conference on Computer Vision and Pattern Recognition,* Portland, OR, USA, 2013, pp. 1513-1520.

[33] Kumar, A., Marks, T.K., Mou, W., Wang, Y., Jones, M., Cherian, A., Koike-Akino, T., Liu, X. and Feng, C. LUVLi, "Face Alignment: Estimating Landmarks' Location, Uncertainty, and Visibility Likelihood," in *IEEE Conference on Computer Vision and Pattern Recognition,* San Francisco, CA, USA, 2010, pp. 8236-8246.

[34] Viola, P.; Jones, M.J. "Robust real-time face detection," *Int. J. Comput. Vis*, vol. 57, no. 2, pp. 137-154, May 2004.

[35] Viola, P.; Jones, M.J. "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition,* Kauai, HI, USA, 2001, pp. I-511-I-518.

[36] Zhang, L., Chu, R., Xiang, S., Liao, S., and Li, S. Z "Face detection based on multi-block lbp representation," in *International conference on biometrics,* Seoul, Korea, 2007, pp. 11-18.

[37] Sagonas, C., Tzimiropoulos, G., Zafeiriou, S. and Pantic, M., "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *IEEE Conference on Computer Vision and Pattern Recognition,* Portland, OR, USA, 2013, pp. 397-403.

[38] Kostinger, M., Wohlhart, P., Roth, P. M. and Bischof, H., "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *IEEE international conference on computer vision workshops,* Barcelona, Spain, 2011, pp. 2144-2151.

[39] Zhu, S., Li, C., Loy, C.C. and Tang, X., "Face alignment by coarse-to-fine shape searching," in *IEEE Conference on Computer Vision and Pattern Recognition,* Boston, MA, USA, 2015, pp. 4998–5006.

[40] Wu W. and Yang, S., "Leveraging intra and inter-dataset variations for robust face alignment," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop,* Honolulu, HI, USA, 2017, pp. 150–159.

[41] Han, S., Pool, J., Tran, J. and Dally, W., “Learning both weights and connections for efficient neural network,” in *Advances in Neural Information Processing Systems,* Montreal, Canada, 2015, pp. 1135–1143.

[42] Han, S., Mao, H. and Dally, W.J., “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding,” arXiv preprint arXiv:1510.00149., 2015.

[43] Y. Liu, H. Wang, M. Peng, J. Guan and Y. Wang, “An Incentive Mechanism for Privacy-Preserving Crowdsensing via Deep Reinforcement Learning,” *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8616-8631, 15 May, 2021.

[44] Y. Liu, T. Feng, M. Peng, J. Guan and Y. Wang, “DREAM: Online Control Mechanisms for Data Aggregation Error Minimization in Privacy-Preserving Crowdsensing,” *IEEE Transactions on Dependable and Secure Computing*, 2020, doi: 10.1109/TDSC.2020.3011679.

[45] W. Quan, N. Cheng, M. Qin, H. Zhang, H. A. Chan and X. Shen, “Adaptive Transmission Control for Software Defined Vehicular Networks,” *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 653-656, June 2019.

[46] W. Quan, Y. Liu, H. Zhang and S. Yu, “Enhancing Crowd Collaborations for Software Defined Vehicular Networks," in IEEE Communications Magazine,” *IEEE Communications Magazine*, vol. 55, no. 8, pp. 80-86, Aug. 2017.