applied sciences

MDPI

Article

Low-Complexity Pupil Tracking for Sunglasses-Wearing Faces for Glasses-Free 3D HUDs

Dongwoo Kang [1,*] and Hyun Sung Chang [2]

[1] Department of Electronic and Electrical Engineering, Hongik University, Seoul 04066, Korea
[2] Multimedia Processing Lab, Samsung Advanced Institute of Technology, Suwon 16678, Korea; hyun.s.chang@samsung.com
* Correspondence: dkang@hongik.ac.kr

Abstract: This study proposes a pupil-tracking method applicable to drivers both with and without sunglasses on, which has greater compatibility with augmented reality (AR) three-dimensional (3D) head-up displays (HUDs). Performing real-time pupil localization and tracking is complicated by drivers wearing facial accessories such as masks, caps, or sunglasses. The proposed method fulfills two key requirements: low complexity and algorithm performance. Our system assesses both bare and sunglasses-wearing faces by first classifying images according to these modes and then assigning the appropriate eye tracker. For bare faces with unobstructed eyes, we applied our previous regression-algorithm-based method that uses scale-invariant feature transform features. For eyes occluded by sunglasses, we propose an eye position estimation method: our eye tracker uses nonoccluded face area tracking and a supervised regression-based pupil position estimation method to locate pupil centers. Experiments showed that the proposed method achieved high accuracy and speed, with a precision error of <10 mm in <5 ms for bare and sunglasses-wearing faces for both a 2.5 GHz CPU and a commercial 2.0 GHz CPU vehicle-embedded system. Coupled with its performance, the low CPU consumption (10%) demonstrated by the proposed algorithm highlights its promise for implementation in AR 3D HUD systems.

Keywords: eye tracking; eye detection; image occlusion; autostereoscopic 3D display; augmented reality

check for updates

Citation: Kang, D.; Chang, H.S. Low-Complexity Pupil Tracking for Sunglasses-Wearing Faces for Glasses-Free 3D HUDs. *Appl. Sci.* **2021**, *11*, 4366. https://doi.org/10.3390/app11104366

Academic Editor: Włodzimierz Kasprzak

Received: 19 April 2021
Accepted: 10 May 2021
Published: 11 May 2021

1. Introduction

Three-dimensional (3D) displays provide realistic visual experiences with an enhanced sense of image depth [1,2]. Recent developments of holographic optical elements (HOE) technologies have increased the possibilities for commercialization of augmented reality (AR) devices including wearable AR glasses [3]. Additionally, autostereoscopic 3D displays offer the full benefits of the 3D experience without requiring the observer to wear 3D glasses. This is enabled via the eye-tracking-based autostereoscopic 3D method, which overcomes 3D viewing zone limitations, thereby allowing users a seamless 3D experience with higher 3D-resolved content [1,2]. The eye-tracking-based autostereoscopic 3D method can be adopted in automobile head-up displays (HUDs) for drivers, which display realistic 3D navigation information about the road via combiners placed on the windshield [4–6]. Eye-tracking-based autostereoscopic 3D display systems require accurate and fast 3D measurements of the eye positions of the viewer to display 3D images with low 3D crosstalk and high 3D resolution. However, owing to the limited capabilities of automobile computing systems, recent deep-learning-based algorithms are not used because they typically require heavy graphical processing units (GPUs).

Fast and accurate eye-gaze-tracking technologies are required not only in glasses-free 3D HUDs, but also in many AR display systems [7]. Other applications such as antispoofing in face recognition [8] and interaction systems with virtual contents [9] also utilize eye-gaze-tracking technologies. To date, many studies of eye tracking have focused on eye-gaze-tracking for wearable devices such as head-mounted devices. These methods calculate the vector between the pupil center and the corneal reflection using near-infrared (NIR) light sources to estimate the viewer's looking direction. In most methods, the NIR camera and light sources must be placed in specific locations to achieve clear bright pupil and corneal reflection images [10–16]. Remote eye-tracking methods have adopted recent computer vision techniques to detect and track viewers' eyes at greater distances for various consumer electronics such as 3D televisions and monitors, gaming devices, smartphones, driver monitoring systems, and HUDs in automobiles [17–19]. These methods usually adopt NIR light sources to capture high-quality eye images under various light conditions [20–22].

The goal of gaze tracking is to estimate the viewer's gaze direction, while our eye position tracking aims to detect and track 3D eye positions using red–green–blue (RGB) web-cameras at remote distances. In previous studies, we published real-time computation-based eye-tracking methods for bare faces [23,24]. This accurate and fast pupil position tracking provided clear 3D images with high 3D resolution using limited embedded system resources, even when the head movements of the users were considered in real time. However, localization and tracking of the real-time pupil position of a driver can be affected by various occlusion factors, such as the driver wearing a mask, cap, or sunglasses. If the eye-tracking precision error is larger than the 3D margin of the HUD, users can see the 3D crosstalk and will feel 3D fatigue. As the 3D crosstalk margins of our AR 3D HUD prototype are 12 mm [2], the eye-tracking precision must be less than this threshold to allow users to enjoy 3D-crosstalk-free content (Figure 1). Furthermore, a user will experience 3D crosstalk when they move their head if the overall system latency, including eye tracking and 3D rendering, is high. Complex state-of-the-art deep-learning-based algorithms for detecting facial features [25–29] are very accurate, but incompatible with 3D autostereoscopic display systems, which have limited GPU capacity.

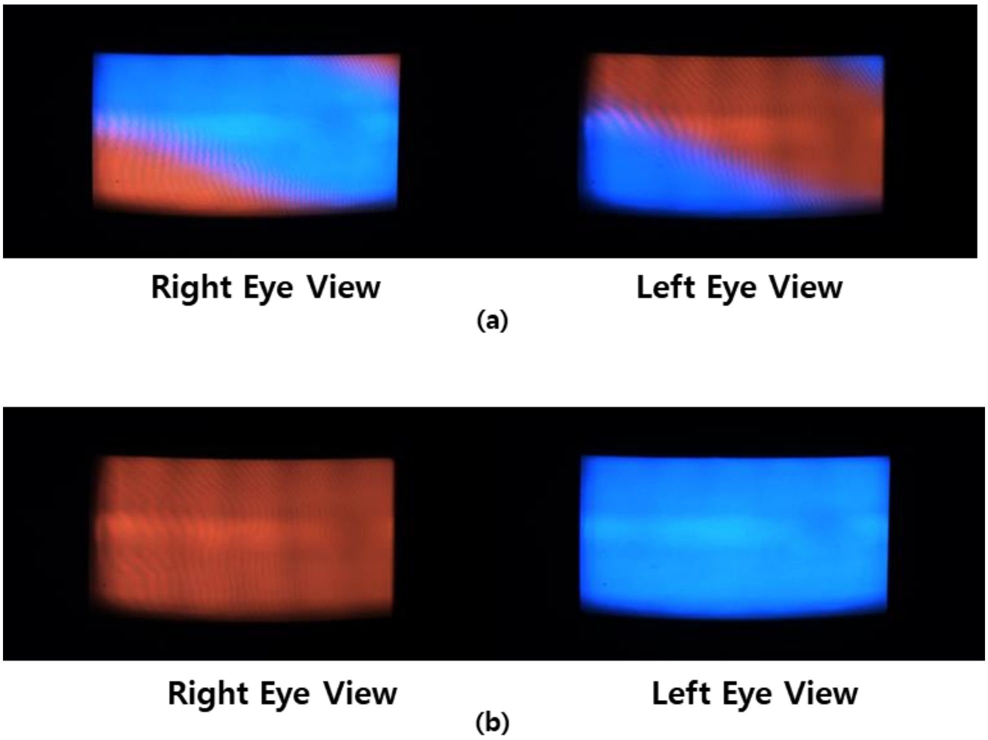

Figure 1. Eye-tracking-based 3D light field rendering results based on an input of a red right image and a blue left image. (**a**) 3D crosstalk occurs when the error associated with the eye-tracking accuracy exceeds the 3D crosstalk margin. (**b**) 3D rendered results corresponding to high-accuracy eye tracking with smaller error than the 3D crosstalk margin. The 3D crosstalk margins of the AR 3D HUD prototype are 12 mm, representing the eye-tracking precision threshold.

The purpose of this study was to realize pupil center localization for drivers wearing sunglasses. Our proposed system consists of face detection, eye–nose shape keypoint alignment, a tracker checker, and tracking mode switching. For scenarios such as the user wearing sunglasses, corresponding shape aligners are applied using an image classification approach for pupil center localization. Occlusion, which can be exacerbated by sunlight reflection, makes pupil localization quite challenging in such scenarios. To tackle this problem, we use nonoccluded areas to infer the pupil center. The contributions of our method can be summarized by the following points:

- We propose a pupil center localization system applicable to both bare faces and sunglasses-wearing faces. We classified facial images into these two categories and performed pupil tracking accordingly.
- For sunglasses-wearing faces, we inferred the eye position behind the sunglasses by applying a supervised regression method to the non-occluded areas.

2. Methods

The proposed eye-tracking method, which works for bare faces and sunglasses-wearing faces, comprises two different modes. Two different machine-learning-based eye trackers, which include facial shape alignment and tracker checker modules, are adopted to deduce the eye center positions. The eye-tracking mode selection is based on an eye-area classifier, which determines whether a person is wearing sunglasses or not.

Both eye-tracking modes use the face region as their input, which can be obtained via face detection or face region extraction using the frame-by-frame tracking of facial points. Depending on facial classification differentiators, such as whether a person is wearing sunglasses, corresponding shape aligners are applied to obtain the pupil center localization. In our previous work [23,24], we developed an eye-tracking method for bare faces that involves 11-point eye–nose shape tracking based on the supervised descent method (SDM) [30] (for details of the algorithms for bare-face eye tracking, see References [23,24]). For sunglasses-wearing faces, we propose an eye-position estimation method: the eye tracker uses nonoccluded face area tracking (nose–mouth–face boundary) combined with a supervised regression-based pupil position estimation method to identify the pupil centers. The basic components of our proposed eye-tracking method for people wearing sunglasses can be divided into two main stages: (1) nonoccluded face area tracking and (2) eye-center position estimation for occluded areas. A flowchart describing the proposed eye-tracking method for bare and sunglasses-wearing faces is presented in Figure 2.

2.1. Whole-Face Detection and Classification of Faces with Sunglasses

We utilized the error-based learning (EBL) method published in our previous work [23], which utilizes the AdaBoost classifier [31–33] and local binary pattern (LBP) features [31–33] for facial detection tasks. The EBL framework trains only a small fraction (less than 5%) of the detection training image DBs in much shorter training times, while it improves the detection rate through three stages [23]. We used a cascaded classifier with N-boosting substages for each stage in EBL [23]. Multiscale block LBP feature space was adopted for the cascaded AdaBoost classifier [23]. Unlike our previous method, which used an eye–nose detector, for this study we constructed an entire face detector to address the issue of eye occlusion by sunglasses. The cascade-AdaBoost algorithm is used extensively for facial detection tasks. Although deep-learning-based facial detection approaches such as SqueezeDet [26], region-based convolutional neural networks (R-CNN) [34], and multitask CNNs [27] have been developed recently, such techniques require considerable GPU resources and, therefore, require network lightening for integration with automobiles, for example as part of AR 3D HUD systems, which adopt commercial vehicle-embedded computing boards with limited GPU resources. Therefore, fast processing must be married with low complexity to maximize the performance of the limited computational resources of AR 3D HUD systems.

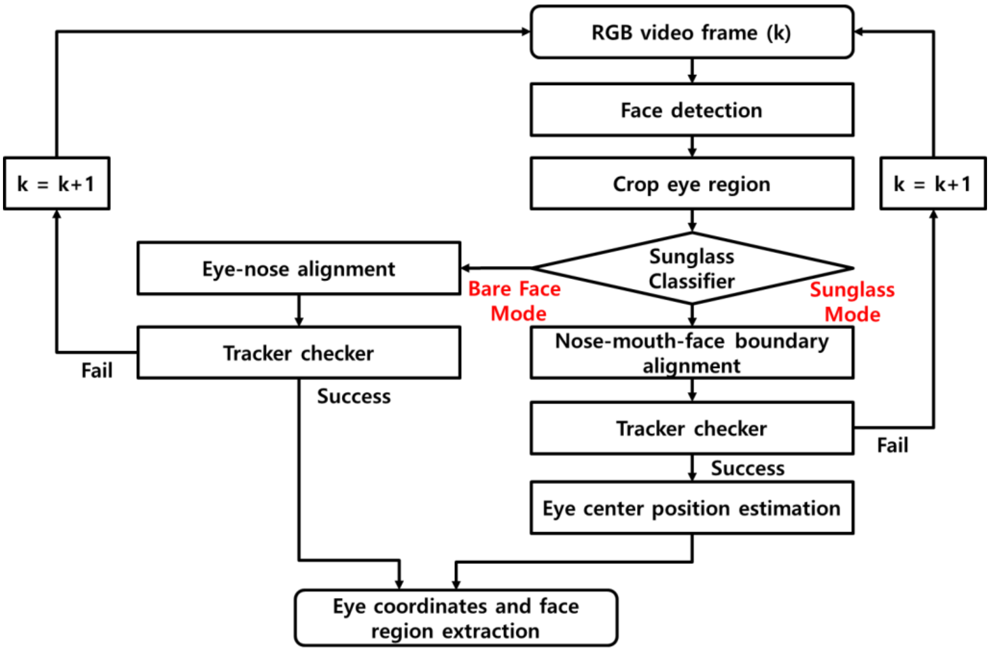

Figure 2. Flowchart of the proposed eye-tracking method for bare and sunglasses-wearing faces.

Our proposed detector is both practical and simple, requiring only a CPU. For the bare-face training dataset, we reutilized samples constructed for our previous study [23], which outlines the details of the EBL algorithm used for training involving the bare-face image datasets. For sunglasses-wearing faces, we constructed a new training image database containing 30,000 images of 37 people wearing sunglasses according to our own capturing and labeling annotations (Figure 3a). For the final face-detection training step, the 50,000 bare faces and 30,000 sunglasses-wearing faces from the training samples were used. To run two different eye-tracking modes according to the presence of sunglasses, we performed an image classification routine to separate the bare and sunglasses-wearing faces. The region surrounding the eyes was cropped using the statistical mean shape from our training dataset labeling (Figure 3b). In addition, the eye region classifications were assisted by the cascade-AdaBoost classifier and LBP features.

2.2. Nose–Mouth–Face Boundary Tracking: Alignment and Tracker Checker

After the detector identifies the eye–nose region, the tracking mode starts to extract the coordinates of the pupil centers based on eye–nose shape alignments using the SDM [30] and the scale-invariant feature transform (SIFT) [35]. The SDM mode trains a sequence of descent directions that minimize the mean of the nonlinear square functions from each landmark point [30]. SIFT extracts feature description from landmark points and is invariant to translations, rotations, and scaling transformations, and is widely used in face recognition technologies [35]. This non-CNN-based method has the advantages of low CPU consumption and fast speed compared to state-of-the-art CNN-based landmark point alignment methods.

To prevent erroneous detection or tracking, we propose a novel tracker checker idea. The proposed tracker checker guarantees that the aligned results contain the eyes. For each frame, after the nose–mouth–face boundary points are aligned, the proposed nose–mouth–face boundary tracker checker performs a final examination of the tracking results, irrespective of whether it tracks eyes or not. Thus, more efficient, faster eye-tracking system computations can be achieved.

A greedy search algorithm was performed to find the best nonoccluded area points for estimating the eye center positions. Among the 23 candidate points from nonoccluded areas, such as the nose–mouth–face boundary, we identified the 11 points that best suggested the eye position via the greedy algorithm with end-to-end eye position calculation (further details are provided in Section 2.3). Owing to the computational complexity of the SIFT feature extraction increasing according to the number of points involved, we only utilized 11 points to ensure reliability and manageable complexity. Among these points, the nose–mouth and nose–mouth–face boundary points were identified as the best two positions (Figure 4).

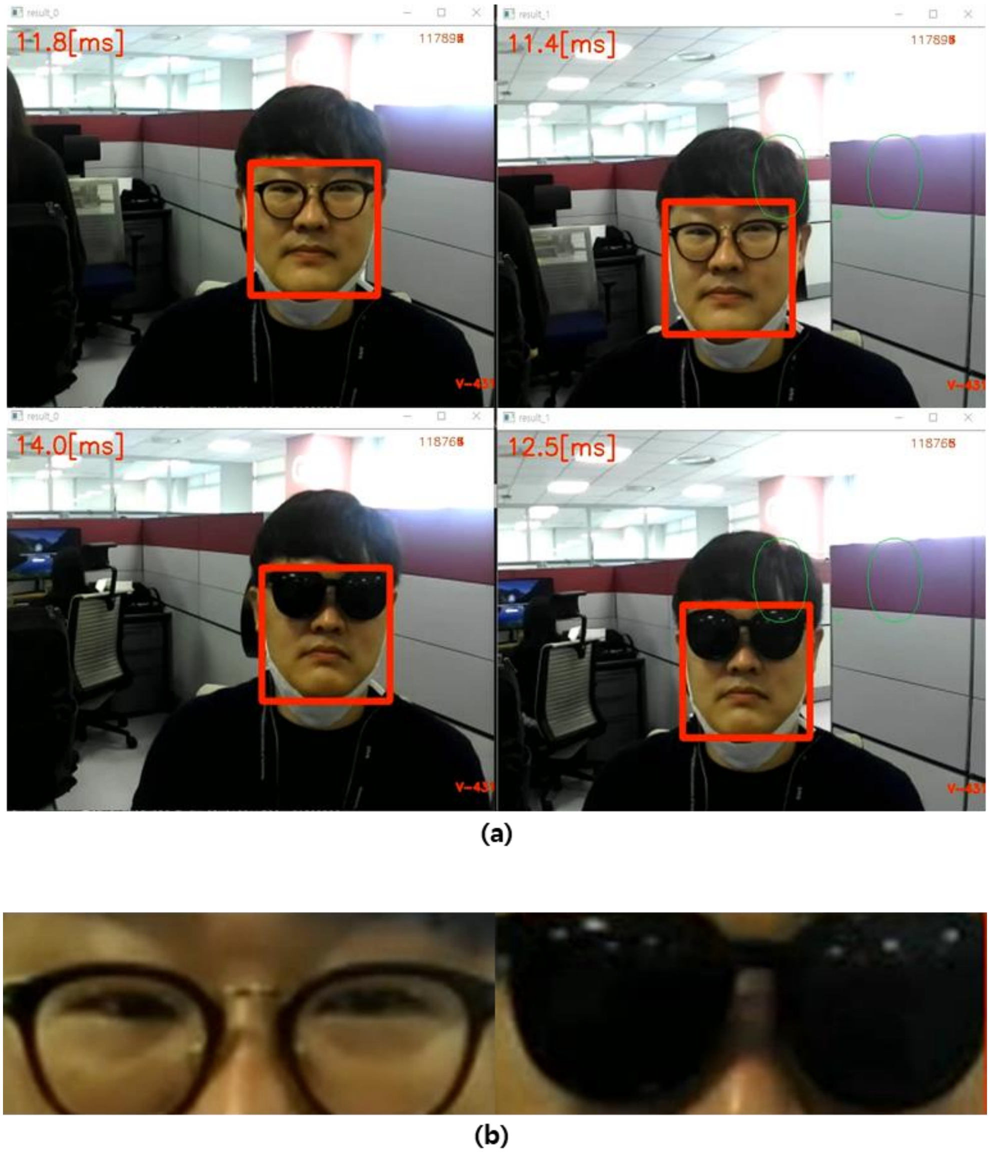

Figure 3. (**a**) Entire-face detector, which detects both bare faces (1st row) and sunglasses-wearing faces (2nd row). Each column shows the left and right image from the stereo camera. (**b**) Cropped eye regions for the classification of bare and sunglasses-wearing faces. Cropping was performed using the statistical mean shape determined according to training dataset labeling.

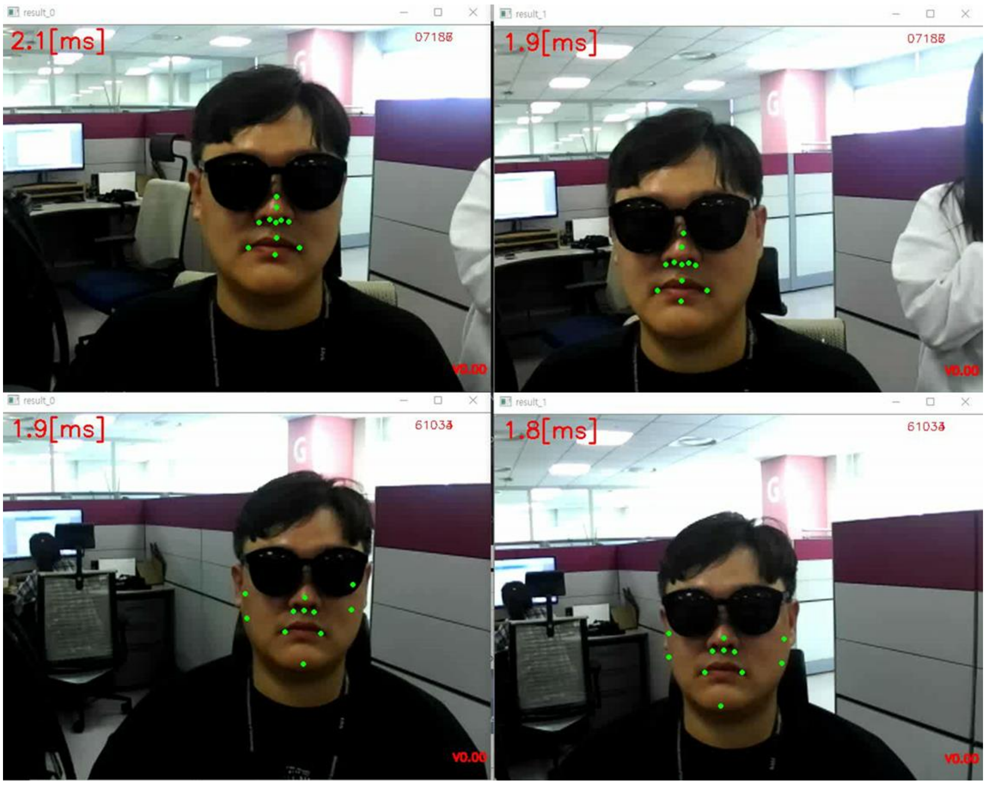

Figure 4. Nonoccluded area shape point candidates identified by a greedy search algorithm: nose–mouth points (**1st row**) and nose–mouth–face boundary points (**2nd row**). Each column shows the left and right images from the stereo camera.

2.3. Eye Center Position Estimation from Nose–Mouth–Face Boundary Points Using a Supervised Linear Regression Algorithm

Using the 11 points representing nonoccluded face areas identified in Section 2.2, we estimated the eye center positions using a supervised linear regression method, meaning that we constructed the training image database with the ground truth of eye position on sunglasses according to other points (Figure 5). All the ground truths of eye positions were annotated as precisely as possible via comparison with the corresponding bare-face images. Ultimately, 32,400 sunglasses-wearing images with a head pose less than 20° were constructed with eye position labeling and used for eye position estimations using a supervised linear regression.

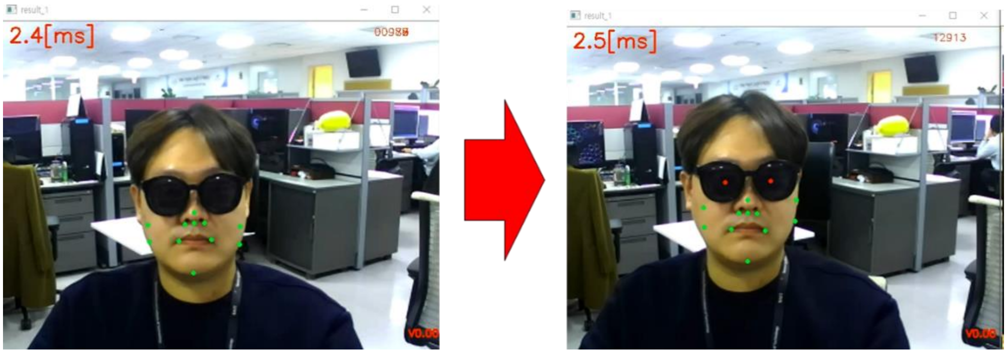

Figure 5. Eye position estimation by a supervised linear regression algorithm. Nonoccluded area shape tracking points (**left**, green dots), with eye positions inferred using these points (**right**, red dots).

In this paper, we set E_{left} as the left eye center position and let X represent the set of points corresponding to nonoccluded areas, i.e., the nose–mouth–face boundary points introduced in Section 2.1. If β_{left} is a linear regression matrix, then we can write

$$E_{left} = X\beta_{left}. \tag{1}$$

Similarly, the right eye position, E_{right}, can be formulated using β_{right}:

$$E_{right} = X\beta_{right}. \tag{2}$$

Solving Equations (1) and (2) leads to

$$\beta = X^{+}E_{left\ or\ right}, \tag{3}$$

where X^{+} is the pseudo-inverse of X when assuming a linear relationship between E and X. Collecting more data for X and E produces a more accurate value of β. This process of eye position estimation using a supervised linear regression method from n-samples of the 11 nose–mouth–face boundary (NMB) points is illustrated in detail below (Box 1):

Box 1. Detailed processes of eye position estimation using a supervised linear regression method.

✓ 1-sample

: eye position from 11 NMB points

: $E_{left} = X\beta_{left}$

: $e_1 = a_1x_1 + a_2y_1 + \ldots + a_{11}y_{11}$

: $e_2 = b_1x_1 + b_2y_1 + \ldots + b_{11}y_{11}$

: $[e_1\ e_2] = [x_1\ y_1x_2y_2 \ldots x_{11}y_{11}] \begin{bmatrix} a_1b_1 \\ a_2b_2 \\ \ldots \\ a_{11}b_{11} \end{bmatrix}$

✓ n-samples

: n-samples' eye positions from n-samples' 11 NMB points by linear regres-sion

: $E_{left} = X\beta_{left}$

: $\begin{bmatrix} e_{11}\ e_{21} \\ e_{12}\ e_{22} \\ \ldots \\ e_{1n}\ e_{2n} \end{bmatrix} = \begin{bmatrix} x_{1,\,1}\ y_{1,1}x_{2,1}y_{2,1} \ldots x_{11,1}y_{11,1} \\ x_{1,\,2}\ y_{1,2}x_{2,2}y_{2,2} \ldots x_{11,2}y_{11,2} \\ \ldots \\ x_{1,\,n}\ y_{1,n}x_{2,n}y_{2,n} \ldots x_{11,n}y_{11,n} \end{bmatrix} \begin{bmatrix} a_1b_1 \\ a_2b_2 \\ \ldots \\ a_{11}b_{11} \end{bmatrix}$

➔ $\min_{\beta} ||E - X\beta||^2$

➔ $X^T(E - X\beta) = 0$

➔ $\beta = \left(X^TX\right)^{-1}X^TE$

➔ $\beta = X^{+}E$

3. Results

The proposed algorithm yielded successful real-time detection (~60 fps) and tracking (~200 fps) for a range of different environments, users, and system challenges based only on CPU computations. When the tracking mode was considered, the execution time was approximately 2–5 ms when using a standard PC with a 2.5 GHz CPU (consumption 10%, see Table 1) and running Windows 7. The algorithm was constructed in C++.Additionally, when tested using a commercial embedded computing board (Samsung Exynos-auto evt1 for 3D HUD), our proposed method achieved comparable eye-tracking speed and CPU usage results. Figure 6 shows some examples of real-time seamless pupil tracking using the proposed method with a stereo camera. The camera image resolution was 640 × 480 pixels with a recording speed of 60 fps and a field of view of 60° × 40°. This proved effective for capturing various images of sunglasses-wearing subjects, provided head movements had a speed below 250 mm/s. We tested the performance of the proposed algorithm for image and video database entries captured inside and outside an office environment of 10 people wearing different styles of sunglasses. The average aligner precision error was 2 mm for the bare-face images and 10 mm for the sunglasses-wearing images.

Table 1. Performance of the proposed eye-tracking method for bare and sunglasses-wearing faces.

	Bare-Face Eye Tracker	**Sunglasses-Wearing Face Eye Tracker**
Precision (mean error)	2 mm	10 mm
Distance between camera and user	~50–200 cm	~50–200 cm
Speed (ms/frame)	~2–5 ms (2.5 GHz CPU)	~2–5 ms (2.5 GHz CPU)
CPU consumption	10%	10%

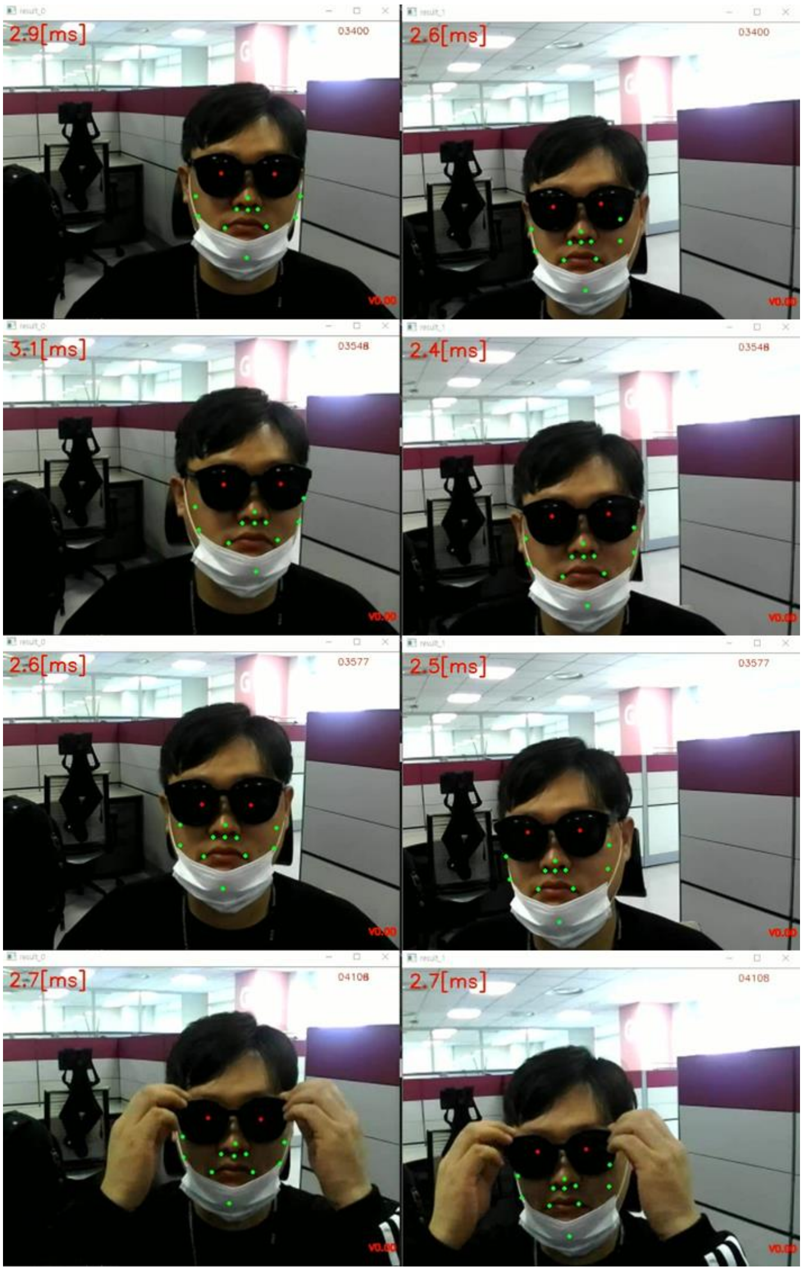

Figure 6. Seamless eye tracking for a user wearing sunglasses. A stereo RGB camera was used for 3D position conversion. The two columns show the left and right images from the stereo camera.

The two databases were trained using the SIFT and SDM algorithms. Moreover, both bare-face and sunglasses-wearing-face image databases were used to inform the eye position estimation algorithm based on the supervised linear regression algorithm. Our method demonstrated high accuracy and speed, estimating the pupil center position with less than 10 mm error at a speed of 9 ms for sunglasses-wearing faces.

4. Discussion

This study proposes a low complexity technique for the detection and tracking of bare and sunglasses-wearing faces. The algorithm results showed high accuracy and fast speed even when eyes were occluded by sunglasses, while the CPU consumption was only 10%. In previous studies, we reported on the performance of a bare-face eye tracker [23]. Although this tracker identified and tracked eyes with high accuracy and speed, its performance deteriorated for users wearing sunglasses. The algorithms proposed herein for eye position estimation using nonoccluded shape points on sunglasses-wearing faces address this issue and showed a marked improvement. Furthermore, the proposed algorithm retains the advantages of low complexity and high speed demonstrated by our algorithms for analyzing bare faces [23]. Moreover, when compared to state-of-the-art deep-learning-based methods such as the practical facial landmark detector (PFLD) [36], which has a precision of 8 mm, our proposed method offers comparable precision (10 mm), reduced CPU consumption (10% down from 25%), and is faster (4 ms down from 12 ms). Figure 7 presents a performance comparison between our proposed algorithm and those reported in other studies.

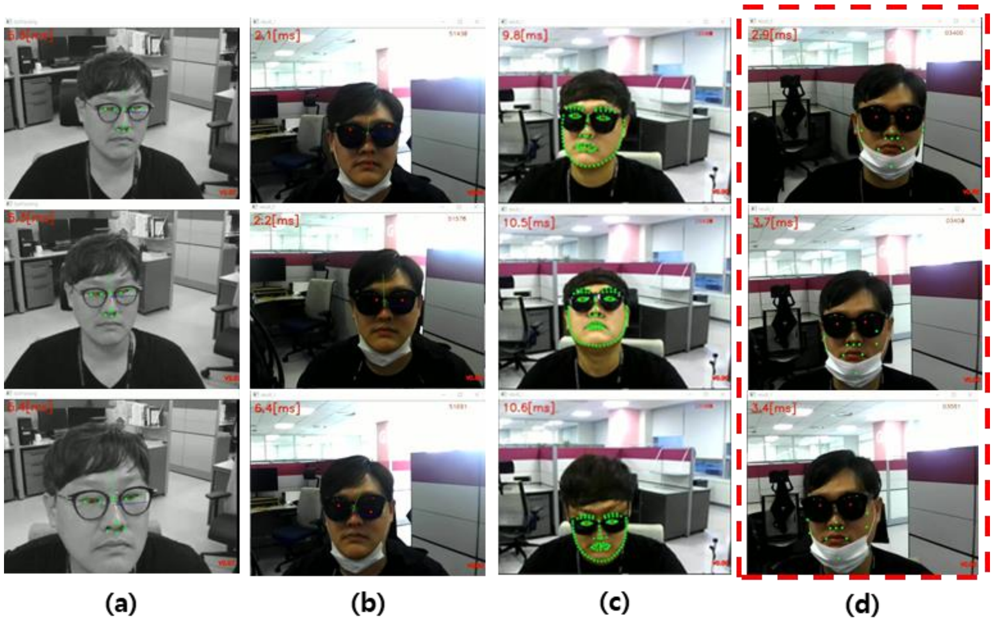

Figure 7. Method comparison. (**a**) Direct eye tracking for bare faces using the SDM+SIFT algorithm [23]. It showed 2 mm precision with 10% CPU consumption at a speed of 2–5 ms per frame. (**b**) Direct eye tracking for sunglasses-wearing faces using the SDM+SIFT algorithm [23]. It showed 25 mm precision with 10% CPU consumption at a speed of 2–5 ms per frame (**c**) Direct eye tracking using a CNN-based PFLD algorithm [36]. It showed 8 mm precision with 25% CPU consumption at a speed of 12 ms per frame. (**d**) Proposed indirect pupil position estimation method, which includes an indirect eye-tracking algorithm that estimates pupil location based on other facial feature points. The proposed method showed 10 mm precision with 10% CPU consumption at a speed of 2–5 ms per frame.

One of the main advances of our method compared with previously published works is its low complexity, with only low CPU computation required to provide real-time eye tracking. This is particularly valuable when considering the limited system resources of current vehicles. Additionally, our method offers different mode-switching systems according to user conditions. In that sense, our proposed system structures are highly compatible with other algorithms, including deep neural algorithms.

We also conducted experiments involving 20 test subjects to investigate the effects of 3D crosstalk when viewing 3D content generated by the AR 3D HUD prototype while wearing sunglasses (Table 2). None of the users reported experiencing static 3D crosstalk (i.e., user head movements < 250 mm/s) while wearing sunglasses. However, when user head movements exceeded 250 mm/s (i.e., dynamic 3D crosstalk), 11 of the 20 participants reported experiencing 3D crosstalk while wearing sunglasses, while only 2 of the 20 participants reported experiencing dynamic 3D crosstalk while not wearing sunglasses. Considering that the 3D margin of our AR 3D HUD prototype is 12 mm, the 10 mm precision error associated with the proposed method is near the 3D margin boundaries, which could cause delayed left and right 3D view conversion owing to limited system latency. Therefore, further improvements to the eye-tracking precision are required in the future to eliminate the potential for dynamic 3D crosstalk. To validate 3D crosstalk with a large eye-tracking error of 10 mm, we added a random eye position error of <8 mm to the bare-face eye tracker, which had an average mean error of 2 mm (Figure 8). The addition of this random error to our bare-face eye tracker did not lead to users reporting static 3D crosstalk when the eye-tracking precision was <10 mm.

Table 2. Experimental results for 3D crosstalk with the proposed eye-tracking algorithms for bare and sunglasses-wearing faces. All experiments were performed using a commercial embedded system with an AR 3D HUD prototype.

	Bare-Face Eye Tracker	**Sunglasses-Wearing Eye Tracker**
3D crosstalk experimental system	AR 3D HUD prototype (3D margin 12 mm)	
3D content	3D arrows with glow effects	
Number of participants reporting 3D crosstalk (static)	0/20	0/20
Number of participants reporting 3D crosstalk (dynamic)	2/20	11/20

This study has a few limitations that we hope to address in subsequent work. Here, we assumed that only eyes are occluded by sunglasses, leaving the nose, mouth, and face boundaries unobstructed by other accessories such as masks or caps. To handle a greater variety of user face-shape occlusions, the proposed system structures require multiple different eye trackers, each trained using a separate image database, which requires enhanced multiclass classification capabilities. Additionally, the number of publicly available images of drivers wearing sunglasses for use in face occlusion training image datasets is limited, presenting an obstacle for state-of-the-art deep-learning-based algorithm training. Therefore, data augmentation is required to construct large face occlusion datasets for the designated eye trackers. Finally, to handle various user face occlusion cases, further study is necessary to develop and optimize our algorithm.

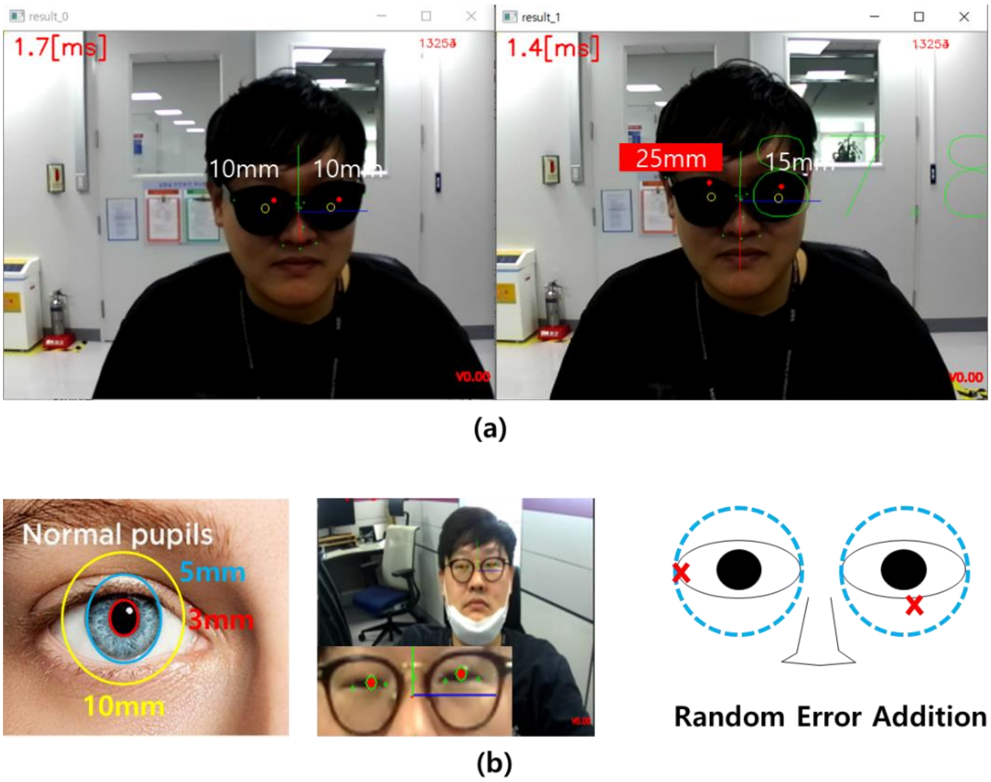

Figure 8. Participant-based experiments assessing the susceptibility of the proposed method to 3D crosstalk for users wearing sunglasses. (**a**) Examples of eye-tracking precision error of 10 mm (left), 15 mm, and 25 mm (right). (**b**) Eye-tracking precision error was calculated using the distance between the center of pupil center and tracked eye position (left). An example of a bare-face eye-tracking result with a precision of 2 mm (middle). To validate 3D crosstalk with a large eye-tracking error of 10 mm, we added a random eye-position error (right). Users experienced 3D crosstalk when the eye-tracking precision error was >15 mm. For the current 3D HUD prototype, the 3D crosstalk was low with an eye-tracking precision error of <10 mm.

5. Conclusions

In this paper, we propose a new pupil center detection system that is capable of analyzing both bare faces and those obstructed by sunglasses. The proposed method deploys different eye-tracking mode-switching systems according to the user conditions. For bare faces, we propose an SDM-based pupil segmentation method and utilize a coarse-to-fine strategy for multiclass shape detection. For sunglasses-wearing faces, pupil centers are located indirectly by estimating the pupil positions based on the nose–mouth–face boundary points using a supervised linear regression algorithm. The proposed system is fast, accurate, robust, and computationally undemanding compared to state-of-the-art CNN-based algorithms. Our system has the potential to accelerate AR 3D HUD commercialization, with simplicity and low processing power demands compatible with the limited computing resources of commercial embedded systems. Further study is necessary to extend our algorithm to handle various user face occlusion cases such as masks or caps with multiclass classification and training data augmentation.

Author Contributions: Conceptualization: D.K. and H.S.C.; methodology: D.K. and H.S.C.; software: D.K.; validation, D.K.; formal analysis: D.K. and H.S.C.; investigation: D.K.; resources: D.K.; data curation: D.K.; writing—original draft preparation: D.K.; writing—review and editing: D.K. and H.S.C.; visualization: D.K.; supervision: D.K.; project administration: D.K.; funding acquisition: D.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Hongik University new faculty research support fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nam, D.; Lee, J.; Cho, Y.H.; Jeong, Y.J.; Hwang, H.; Park, D.S. Flat Panel Light-Field 3-D Display: Concept, Design, Rendering, and Calibration. *Proc. IEEE* **2017**, *105*, 876–891. [CrossRef]
2. Lee, S.; Park, J.; Heo, J.; Kang, B.; Kang, D.; Hwang, H.; Lee, J.; Choi, Y.; Choi, K.; Nam, D. Autostereoscopic 3D display using directional subpixel rendering. *Opt. Express* **2018**, *26*, 20233–20247. [CrossRef] [PubMed]
3. Xiong, J.; Yin, K.; Li, K.; Wu, S.-T. Holographic Optical Elements for Augmented Reality: Principles, Present Status, and Future Perspectives. *Adv. Photonics Res.* **2021**, *2*, 2000049. [CrossRef]
4. Cho, Y.H.; Nam, D.K. Content Visualizing Device and Method. U.S. Patent 20190139298, 9 May 2019.
5. Martinez, L.A.V.; Orozoco, L.F.E. Head-Up Display System Using Auto-Stereoscopy 3D Transparent Electronic Display. U.S. Patent 20160073098, 10 March 2016.
6. Lee, J.H.; Yanusik, I.; Choi, Y.; Kang, B.; Hwang, C.; Park, J.; Hong, S. Automotive augmented reality 3D head-up display based on light-field rendering with eye-tracking. *Opt. Express* **2020**, *28*, 29788–29804. [CrossRef]
7. Xiong, J.; Li, Y.; Li, K.; Wu, S.-T. Aberration-free pupil steerable Maxwellian display for augmented reality with cholesteric liquid crystal holographic lenses. *Opt. Lett.* **2021**, *46*, 1760–1763. [CrossRef]
8. Killioğlu, M.; Taşkiran, M.; Kahraman, N. Anti-spoofing in face recognition with liveness detection using pupil tracking. In Proceedings of the 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI), Herl'any, Slovakia, 26–28 January 2017; pp. 000087–000092.
9. Spicer, C.; Khwaounjoo, P.; Cakmak, Y.O. Human and Human-Interfaced AI Interactions: Modulation of Human Male Autonomic Nervous System via Pupil Mimicry. *Sensors* **2021**, *21*, 1028. [CrossRef]
10. Santini, T.; Fuhl, W.; Kasneci, E. PuRe: Robust pupil detection for real-time pervasive eye tracking. *Comput. Vis. Image Underst.* **2018**, *170*, 40–50. [CrossRef]
11. Mompeán, J.; Aragón, J.L.; Prieto, P.M.; Artal, P. Design of an accurate and high-speed binocular pupil tracking system based on GPGPUs. *J. Supercomput.* **2018**, *74*, 1836–1862. [CrossRef]
12. Ou, W.-L.; Kuo, T.-L.; Chang, C.-C.; Fan, C.-P. Deep-Learning-Based Pupil Center Detection and Tracking Technology for Visible-Light Wearable Gaze Tracking Devices. *Appl. Sci.* **2021**, *11*, 851. [CrossRef]
13. Bozomitu, R.G.; Păsărică, A.; Tărniceriu, D.; Rotariu, C. Development of an Eye Tracking-Based Human-Computer Interface for Real-Time Applications. *Sensors* **2019**, *19*, 3630. [CrossRef]
14. Li, B.; Fu, H.; Wen, D.; Lo, W. Etracker: A Mobile Gaze-Tracking System with Near-Eye Display Based on a Combined Gaze-Tracking Algorithm. *Sensors* **2018**, *18*, 1626. [CrossRef] [PubMed]
15. Wang, J.; Zhang, G.; Shi, J. Pupil and Glint Detection Using Wearable Camera Sensor and Near-Infrared LED Array. *Sensors* **2015**, *15*, 30126–30141. [CrossRef] [PubMed]
16. Lee, J.W.; Heo, H.; Park, K.R. A Novel Gaze Tracking Method Based on the Generation of Virtual Calibration Points. *Sensors* **2013**, *13*, 10802–10822. [CrossRef]
17. Kim, S.; Jeong, M.; Ko, B.C. Energy Efficient Pupil Tracking Based on Rule Distillation of Cascade Regression Forest. *Sensors* **2020**, *20*, 5141. [CrossRef] [PubMed]
18. Su, M.-C.; U, T.-M.; Hsieh, Y.-Z.; Yeh, Z.-F.; Lee, S.-F.; Lin, S.-S. An Eye-Tracking System based on Inner Corner-Pupil Center Vector and Deep Neural Network. *Sensors* **2020**, *20*, 25. [CrossRef]
19. Lopez-Basterretxea, A.; Mendez-Zorrilla, A.; Garcia-Zapirain, B. Eye/Head Tracking Technology to Improve HCI with iPad Applications. *Sensors* **2015**, *15*, 2244–2264. [CrossRef]
20. Brousseau, B.; Rose, J.; Eizenman, M. Hybrid Eye-Tracking on a Smartphone with CNN Feature Extraction and an Infrared 3D Model. *Sensors* **2020**, *20*, 543. [CrossRef]
21. Lee, D.E.; Yoon, H.S.; Hong, H.G.; Park, K.R. Fuzzy-System-Based Detection of Pupil Center and Corneal Specular Reflection for a Driver-Gaze Tracking System Based on the Symmetrical Characteristics of Face and Facial Feature Points. *Symmetry* **2017**, *9*, 267. [CrossRef]
22. Gwon, S.Y.; Cho, C.W.; Lee, H.C.; Lee, W.O.; Park, K.R. Gaze Tracking System for User Wearing Glasses. *Sensors* **2014**, *14*, 2110–2134. [CrossRef]
23. Kang, D.; Heo, J. Content-Aware Eye Tracking for Autostereoscopic 3D Display. *Sensors* **2020**, *20*, 4787. [CrossRef]
24. Kang, D.; Heo, J.; Kang, B.; Nam, D. Pupil detection and tracking for AR 3D under various circumstances. In Proceedings of the Electronic Imaging, Autonomous Vehicles and Machines Conference; Society for Imaging Science and Technology, San Francisco, CA, USA, 13 January 2019; pp. 55-1–55-5.
25. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep learning face attributes in the wild. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3730–3738.

26. Wu, B.; Iandola, F.; Jin, P.H.; Keutzer, K. SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 129–137.
27. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [CrossRef]
28. Wu, W.; Qian, C.; Yang, S.; Wang, Q.; Cai, Y.; Zhou, Q. Look at boundary: A boundary-aware face alignment algorithm. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2129–2138.
29. Dong, X.; Yan, Y.; Ouyang, W.; Yang, Y. Style aggregated network for facial landmark detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 379–388.
30. Xuehan, X.; De la Torre, F. Supervised descent method and its applications to face alignment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 532–539.
31. Viola, P.; Jones, M.J. Robust real-time face detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154. [CrossRef]
32. Viola, P.; Jones, M.J. Rapid object detection using a boosted cascade of simple features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; pp. 511–518.
33. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. In Proceedings of the International Conference on Machine Learning, Bari, Italy, 28 June–1 July 1996; pp. 148–156.
34. Ranjan, R.; Patel, V.M.; Chellappa, R. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 121–135. [CrossRef]
35. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision* **2004**, *60*, 91–110. [CrossRef]
36. Guo, X.; Li, S.; Yu, J.; Zhang, J.; Ma, J.; Ma, L.; Liu, W.; Ling, H. PFLD: A practical facial landmark detector. *arXiv* **2019**, arXiv:1902.10859.