Convolutional Neural Network Design for Eye Detection under Low-illumination

Duy-Linh Nguyen[0000−0001−6184−4133], Muhamad Dwisnanto Putro[0000−0002−1785−1018], Xuan-Thuy Vo[0000−0002−7411−0697], and Kang-Hyun Jo[0000−0002−4937−7082]

Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, South Korea
ndlinh301@mail.ulsan.ac.kr, dputro@mail.ulsan.ac.kr, xthuy@islab.ulsan.ac.kr, acejo@ulsan.ac.kr

Abstract. The eye is an important organ in the human body for sensing and communicating with the outside world. The development of human eye detectors is essential for applications in the computer vision field, especially under low illumination. This paper proposes a convolutional neural network to detect the position of the eye in the acquired image. This network architecture exploits the advantages of convolutional neural networks combined with the concatenated rectified linear unit (C.ReLU), inception module, and Bottleneck Attention Module (BAM) to extract feature maps. Then it uses two detectors to localize the eye area using bounding boxes. The experiment was trained, evaluated on the BioID Face and Yale Face Dataset B (YALEB) dataset. As a result, the network achieves 99.71% and 99.37% of Average Precision (AP) on YALEB and BioID Face datasets, respectively.

Keywords: Attention module · Convolutional neural network (CNN) · Concatenated rectified linear unit (C.ReLU) · Eye detection · Inception module.

1 Introduction

The computer vision field has been focusing on exploiting the structural features of the human body to develop application and support tools. In which, eye detection is a topic of extensive research interest. The location of the human eye and related components provides useful information for many fields such as psychological analysis, biomedical device development, medical diagnostics [12], assistance devices [14], and tracking technology [5]. However, the eye is a quite small organ with a complex structure, so detecting eye position faces many challenges. In particular, lighting conditions, distance, and camera placement can distort the acquired image area and make it difficult to detect. In addition, real-time applications require network architectures that work smoothly with mobile and embedded devices. Based on the above observations and inspired by the drowsiness warning system, this study proposes an eye detector based

on a convolutional neural network under low-illumination. Experiments were performed on BioID Face, YALEB proposed image datasets and tested on a real-time system. The main contributions of this paper are shown as follows:

1- Propose a convolutional neural network design for eye detection based on convolution layer, C.ReLU, inception, and Bottleneck Attention Module.

2- Built datasets for face detection under low-illumination conditions based on BioID Face and YALEB datasets.

The rest of the paper is organized as follows: Section 2 presents methods related to eye position detection. Section 3 describes the proposed method in detail. Section 4 discusses and analyzes the test results. Section 5 concludes the issue and direction development for future research.

2 Related work

This section introduces the methods used in eye detection and eye-related components. These methods can be considered in two aspects: the traditional-based and machine learning-based methods.

The traditional-based method mainly exploits the geometrical features of the eye to detect the eye area and its related components. The methods focus on extracting features of eyes and neighborhoods using classical geometric algorithms. A set of stochastic regressions was used by the authors in [17], the image gradients, and squared dot products in [4] to locate the pupil. The isophote curve is used in [23] to detect the position of the eye and the pupil. Later, several methods used pattern matching algorithms to replace classical algorithms. Specifically, [22] used the elliptical equation, [13] used the inner product detector to locate the eye. These traditional methods can achieve quite accurate results and are easy to deploy. However, their application may be limited by several conditions such as illumination, image resolution, and hardware devices.

The machine learning-based methods can be divided into two groups, traditional machine learning-based, and CNN-based methods. The traditional machine learning-based mainly uses classical image processing algorithms to extract eye and face features. With the advent and development of the OpenCV open-source library, the application of these algorithms has become even more convenient and easy. Some of these algorithms include self-similarity information and shape analysis [3], Haar Wavelet and Support Vector Machine (SVM) [19], Histogram of Oriented Gradients (HOG) features combine with Support Vector Machine [21].

The explosion of convolutional neural networks in the computer vision field has spurred the development of applications for eye detection, eye segmentation, and eye classification. The CNN-based methods focus on studying the central position of the eye or pupil. In [10], the authors used a simple convolutional neural network to detect the central part of the eye. A set of convolutional neural networks is combined to locate the pupil [2]. In addition, several methods are also applied to detect the eye areas using twelve feature points [16] and then use transform operations [18] to localize the eyes. The CNN-based methods have

proved superior ability in feature extraction and flexibility compared to other methods. However, in order to increase performance precision, networks need to be designed with more depth and incorporate many other additional algorithms. This increases the computational memory which hinders the application to real-time systems. Especially, when the system is deployed in low light conditions such as operating at night.

3 Methodology

As depicted in Fig. 1, the eye detection network consists of two main modules: feature extraction and detection.

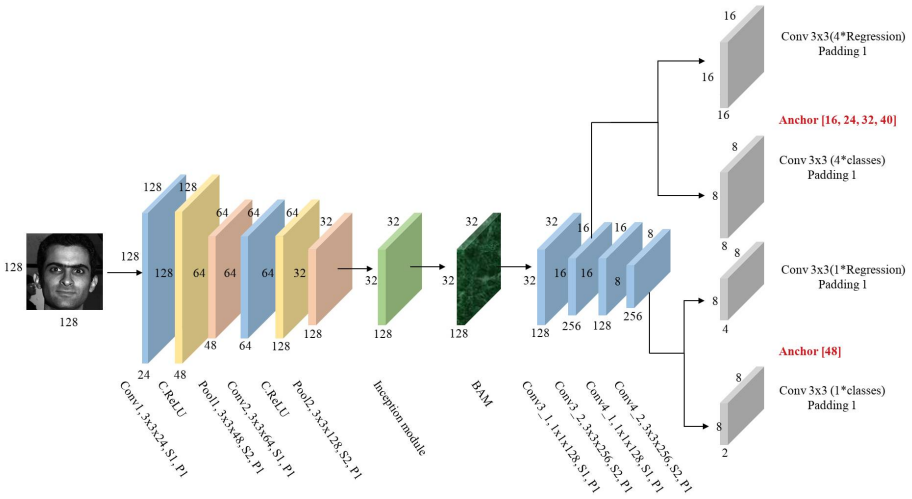


Fig. 1. The proposed eye detection network.

3.1 Feature extraction module

This module is designed based on the advantages of convolution layer, max pooling layer, C.ReLU module [6], Bottleneck Attention Module [15] to extract feature maps. In the first stage, the network extracts low-level feature maps using two convolutions with kernel size of 3×3 followed by each convolution layer a C.ReLU module and a max pooling layer. As shown in Fig. 1, Conv1, Pool1, Conv2 and Pool2 use strides 1, 2, 1 and 2 so it reduces the input image space size from 128×128 to 32×32. In this phase, the C.ReLU module plays the role of ensuring useful information for the feature maps. The architecture of the C.ReLU module is shown in Fig. 2 (a).

In the next stage, the inception module is used as a multi-scale block according to the width to enrich the receptive field of the network. This block is designed with four convolution branches, each of which uses convolutions with kernel size 1×1 or 3×3 and a number of kernels 24 or 32. The detailed description of the inception module is in Fig.2 (b). The feature map after going through this module continues to be enriched and maintains the size $32 \times 32 \times 128$.

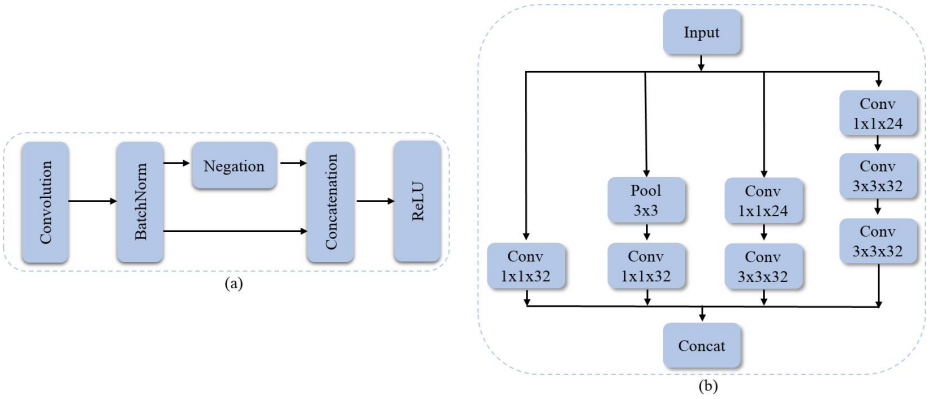


Fig. 2. The architecture of C.ReLu (a) and inception module (b).

After that, an attention mechanism is applied called Bottleneck Attention Module (BAM) [15] which is shown in Fig. 3. This module provides an attention map with separate channels and spatial branches.

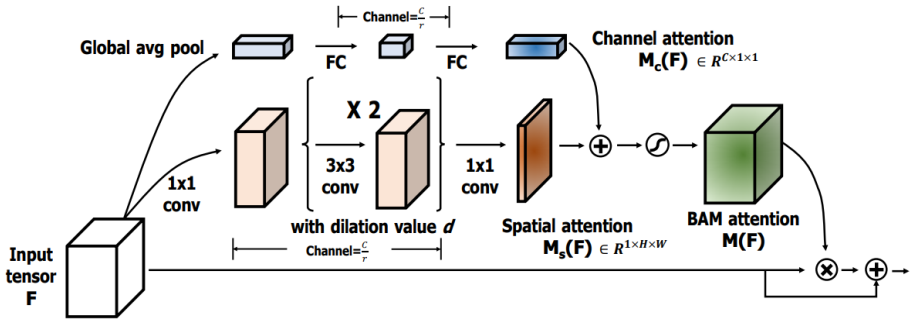


Fig. 3. The architecture of Bottleneck Attention Module [15].

The processing of BAM can be expressed as follows:

$$M(F) = \sigma(M_c(F) + M_c(F)), \tag{1}$$

where $M_c(F)$ is the channel attention, $M_s(F)$ is the spatial attention, is a sigmoid function.
The channel attention branch uses global average pooling on input feature map F and generates a channel vector F_c of size $C \times 1 \times 1$. Then, estimate attention across channels using a multi-layer perceptron (MLP) with one hidden layer. The hidden activation size is set to $\frac{c}{r} \times 1 \times 1$ with r is the reduction ratio. The end is a batch normalization (BN) layer.

$$M_c(F) = BN(MLP(AvgPool(F))), \tag{2}$$

The spatial attention branch uses 1×1 convolution (f_3) to compress the feature map F ($C \times H \times W$) across the channel dimension. Thus, the feature map F is reduced to $\frac{C}{r}$ (r is reduction ratio, it's the same as the channel attention branch). As a next step, this branch applies two 3×3 dilated convolutions (f_2, f_1) to take advantage of contextual information. Then the feature map was further reduced to $1 \times H \times W$ using 1×1 convolution (f_0) and a batch normalization layer was also used at the end of the branch.

$$M_s(F) = BN(f_3^{1\times1}(f_2^{3\times3}(f_1^{3\times3}(f_0^{1\times1}(F))))) \tag{3}$$

Finally, the two branches are extended to CxHxW and combined using element-wise summation operation. Furthermore, a sigmoid function is applied to produce the attention map. BAM only increases the learning capacity of the network and maintains the dimension of the feature map.
In the final stage of the extraction module, four conventional convolutions with 1×1 and 3×3 kernel sizes are used to further reduce the dimension and increase the number of channels of the feature map. Specifically, the first and third convolutions use 1×1 kernel size with 128 channels, the second and fourth convolutions use 3×3 kernel size with 256 channels. As a result, the final feature map is generated with size $8 \times 8 \times 256$ this shows that the feature map is reduced by four times, and the number of channels is increased by two times from 128 to 256.

3.2 Detection module

This module consists of two detector modules to detect eyes at different scales. Each detector uses two 3×3 sibling convolutions for classification and bounding box regression. These two detectors are applied on the last two feature maps in the feature extraction module (at 16×16 and 8×8 feature maps). A set of square anchors of different sizes are used to localize the eyes in the input image. The proposed network uses five square anchors of which four (16, 24, 32, 40) are for small and medium eyes on 16×16 and one (48) for large eyes on 8×8 feature maps. The output of this module is a four-dimensional vector (x, y, w, h)

as the position offset and a two-dimensional vector (eye or non eye) as the label classification. Where (x, y) is the coordinate of the center point, w is the width and h is the height of the bounding box.

3.3 Loss function

The loss function in this paper comprises the loss from two tasks, classification and bounding box regression. In this case, the softmax loss function is used for classification and smooth $L1$ loss is used for regression. The loss function is defined as follows:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}\left(p_i, p_i^*\right) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}\left(t_i, t_i^*\right), \tag{4}$$

Where $L_{cls}\left(p_i, p_i^*\right)$ is the classification loss using the softmax loss defined as in Eq. (5), $L_{reg}\left(t_i, t_i^*\right)$ is regression loss, $L_{reg}\left(t_i, t_i^*\right) = H(t_i - t_i^*)$ and H is the smooth $L1$ loss defined as in Eq. (6), i is the index of an anchor bounding box, p_i is the predicted score of anchor i being an object, p_i^* is ground truth label and $p_i^* = 1$ (the anchor is positive) or $p_i^* = 0$ (the anchor is negative). t_i is the coordinates predicted bounding box and t_i^* is the coordinates of ground truth bounding box. N_{cls} is normalized by the mini-batch size, N_{reg} is normalized by the number of anchor locations, λ is balancing parameter and it is assigned by 10.
The softmax loss function:

$$L_{cls}\left(p_i, p_i^*\right) = -\sum_{i \in Pos} x_i^p log\left(p_i\right) - \sum_{i \in Neg} log\left(p_i^0\right), \tag{5}$$

Where $x_i^p = \{0, 1\}$ is indicator for matching the $i - th$ anchor bounding box to ground truth bounding box of category p, p_i^0 is the probability for non eye classification.
The smooth $L1$ loss:

$$H(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{6}$$

4 Experiments

4.1 Dataset preparation

Besides designing a CNN network architecture for eye detection, this work also proposes datasets for training and evaluating eye detection under low-illumination based on BioID Face [1] and Yale Face Dataset B [7] (YALEB) datasets. Specifically, 1,521 gray-scale images with 384 × 286 pixels resolution were selected from the BioID Face dataset. The images show the front view of 23 different people. From the eye's center coordinates, a square bounding box of size 36×36

is generated using Python code. The total number of labeled bounding boxes is 3,042. For the YALEB dataset, a set of 2,679 images with different lighting conditions and head poses were randomly selected and annotated manually using the LabelImg tool. The total number of labels assigned is 5,358 labels. The datasets are divided into 80% for training and 20% for evaluation. Table 1 shows detailed proposed datasets for eye location detection under low-illumination.

Table 1. The proposed datasets for eye location detection under low-illumination.

Dataset	Image size	Format	Images	Annotations
BioID Face	384×286	PGM	1,521	3,042
YALEB	256×256	JPG	2,679	5,358

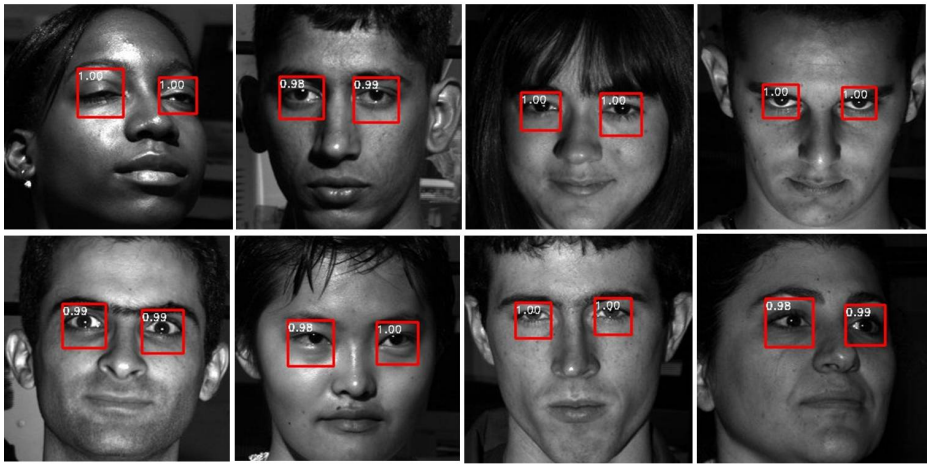
4.2 Experimental setup

The proposed network is implemented by the Pytorch framework. The network was trained with 300 epochs and evaluated on a GeForce GTX 1080Ti GPU, 32 GB of RAM. It applies some training configuration like a batch size = 16, weight decay = 5.10^4, momentum = 0.9. The learning rate is initially set to 10^{-3}. The stochastic Gradient Descent technique is used to optimize the weight during the back-propagation stage. The Non Maximum Suppression algorithm was applied with a threshold of 0.5.

4.3 Experimental result

The eye detection network is trained and evaluated on two datasets, BioID Face and YALEB. In order to make a fair comparison with other CNN network architectures, this work also implemented similar experiments with FaceBoxes [24], SSD [9] and several variants of the proposed eye detection network. As a result, this network achieved Average Precision (AP) up to 99.71% and 99.37% on YALEB and BioID Face datasets, respectively. This result shows that the proposed network outperforms other mobile networks under the same experimental conditions with only 780,383 network parameters. This study also replaces different attention modules to evaluate the eye detection ability of the network. The proposed method with Bottleneck Attention Module (BAM) also achieved the highest performance compared with network variants using Squeeze-and-Excitation (SE) [8], Convolutional Block Attention Module (CBAM) [20], and Triple Attention Module (TAM) [11] with a network parameter that is insignificantly larger. Table 2 shows the comparison result of eye detection network on two datasets with different CNN architectures. The several qualitative results of the eye detection network on BioID Face and YALEB datasets are shown in Fig. 4. However, under low light or uneven illumination conditions, the network may incorrectly detect several surrounding objects (pictures, shirt button) or the part on the face (mouth, lips) because they are similar in shape and color to the eyes, as shown in Fig. 5.

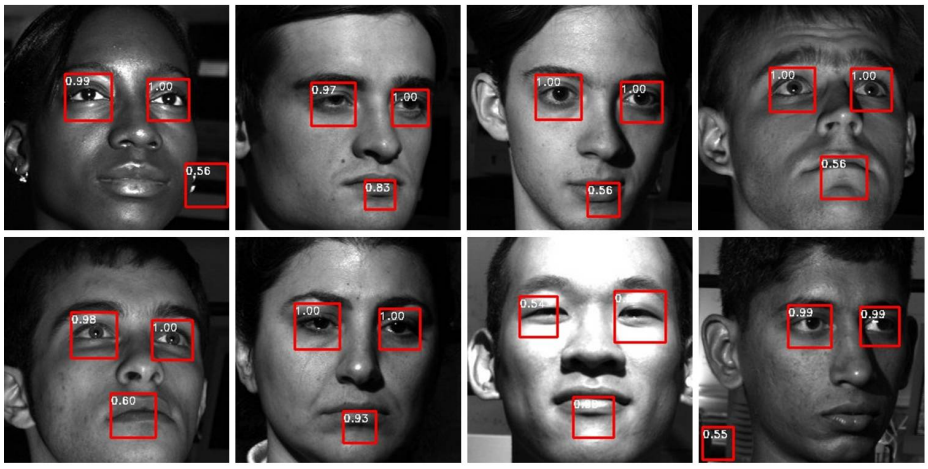
BioID Face dataset



YALEB dataset

Fig. 4. The qualitative results of the eye detection network on the BioID Face and YALEB datasets. The first two rows are the BioID Face dataset and the second two rows are the YALEB dataset.

BioID Face dataset



YALEB dataset

Fig. 5. The mistake results of the eye detection network on the BioID Face and YALEB datasets. The first two rows are the BioID Face dataset and the second two rows are the YALEB dataset.

Table 2. Comparison result of eye detection network on two datasets. SE presents the Squeeze-and-Excitation, BAM presents the Bottleneck Attention Module, CBAM presents the Convolutional Block Attention Module, and TAM presents the Triple Attention Module. The red color is the best competitor.

Model	**Number of paprameters**	**Average Precision (%)**	
		BioID Face	**YALEB**
Mobile architectures			
FaceBoxes	844,610	98.23	99.70
SSD300	23745908	90.90	90.90
SSD512	23745908	90.8	90.00
Our architecture			
Proposed	**780,383**	**99.37**	**99.71**
Our (SE)	778,110	96.31	99.56
Our (CBAM)	780,394	96.55	99.07
Our (TAM)	776,226	98.18	95.63

4.4 Ablation study

This study evaluates the effectiveness of the modules in the eye detection network by several ablation studies on the YALEB dataset shown in Table 3. This experiment removes the Inception Module and the Bottleneck Attention Module, respectively, and then compares the results with the proposed network. The results show that, when removing the Inception Module and using only the Bottleneck Attention Module, the network parameters decreased by 37,792 parameters and the AP decreased by 1.14%. When using only the Inception Module and removing Bottleneck Attention Module, the network parameters and AP only decreased by 4,457 parameters and 0.28%, respectively. Thus, the Bottleneck Attention Module plays an important role in improving the efficiency of eye position detection.

Table 3. Ablation studies on the YALEB dataset.

Module	**Network**		
Inception Module		✓	✓
Bottleneck Attention Module	✓		✓
Parameter	742,591	775,926	780,383
Average Precision (%)	98.57	99.43	99.71

5 Conclusion

This paper presents a convolutional neural network design for eye detection. The proposed network consists of two main modules: feature extraction and detection. Feature extraction module exploited the advantage of convolution, C.ReLu, max

pooling layers, inception, and Bottleneck Attention Modules to extract multi-scale feature maps. Based on the extracted feature maps, the detection module learns and locates the eyes using square anchor boxes. The network is trained and evaluated on two datasets YALEB and BioID Face with low or uneven illumination image sets. It scored 99.71% and 99.37% of AP on YALEB and BioID Face, respectively. In the future, this network will be developed and integrated into the drowsiness alert system for night operating conditions.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government. (MSIT)(2020R1A2C2008972)

References

1. The bioid face database. https://www.bioid.com/facedb, accessed: 2020-10-23
2. Araujo, G., Ribeiro, F., da Silva, E., Goldenstein, S.: Fast eye localization without a face model using inner product detectors. 2014 IEEE International Conference on Image Processing, ICIP 2014 pp. 1366–1370 (01 2015). https://doi.org/10.1109/ICIP.2014.7025273
3. Chen, S., Liu, C.: Eye detection using discriminatory haar features and a new efficient svm. Image Vision Comput. **33**(C), 68–77 (Jan 2015). https://doi.org/10.1016/j.imavis.2014.10.007, https://doi.org/10.1016/j.imavis.2014.10.007
4. Deng, J., Roussos, A., Chrysos, G., Ververas, E., Kotsia, I., Shen, J., Zafeiriou, S.: The menpo benchmark for multi-pose 2d and 3d facial landmark localisation and tracking. International Journal of Computer Vision **127**(6-7), 599–624 (2019)
5. Fu, H., Wei, Y., Camastra, F., Arico, P., Sheng, H.: Advances in eye tracking technology: theory, algorithms, and applications (2016)
6. Fuhl, W., Santini, T., Kasneci, G., Kasneci, E.: Pupilnet: Convolutional neural networks for robust pupil detection. CoRR **abs/1601.04902** (2016), http://arxiv.org/abs/1601.04902
7. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intelligence **23**(6), 643–660 (2001)
8. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. CoRR **abs/1709.01507** (2017), http://arxiv.org/abs/1709.01507
9. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. CoRR **abs/1512.02325** (2015), http://arxiv.org/abs/1512.02325
10. Markuš, N., Frljak, M., Pandžić, I., Ahlberg, J., Forchheimer, R.: Eye pupil localization with an ensemble randomized trees. Pattern Recognition **47**, 578–587 (02 2014). https://doi.org/10.1016/j.patcog.2013.08.008
11. Misra, D., Nalamada, T., Arasanipalai, A.U., Hou, Q.: Rotate to attend: Convolutional triplet attention module. CoRR **abs/2010.03045** (2020), https://arxiv.org/abs/2010.03045
12. Mosa, A.H., Ali, M., Kyamakya, K.: A computerized method to diagnose strabismus based on a novel method for pupil segmentation (2013)

13. Mosa, A.H., Ali, M., Kyamakya, K.: A computerized method to diagnose strabismus based on a novel method for pupil segmentation. In: Proceedings of the International Symposium on Theoretical Electrical Engineering (ISTET 2013) (2013)
14. Nguyen, D.L., Putro, M.D., Jo, K.H.: Eye state recognizer using light-weight architecture for drowsiness warning. In: Nguyen, N.T., Chittayasothorn, S., Niyato, D., Trawiński, B. (eds.) Intelligent Information and Database Systems. pp. 518–530. Springer International Publishing, Cham (2021)
15. Park, J., Woo, S., Lee, J.Y., Kweon, I.S.: Bam: Bottleneck attention module (2018)
16. Shang, W., Sohn, K., Almeida, D., Lee, H.: Understanding and improving convolutional neural networks via concatenated rectified linear units. CoRR **abs/1603.05201** (2016), http://arxiv.org/abs/1603.05201
17. Sharma, R., Savakis, A.: Lean histogram of oriented gradients features for effective eye detection. Journal of Electronic Imaging **24**, 063007 (11 2015). https://doi.org/10.1117/1.JEI.24.6.063007
18. Timm, F., Barth, E.: Accurate eye centre localisation by means of gradients. In: VISAPP (2011)
19. Valenti, R., Gevers, T.: Accurate eye center location through invariant isocentric patterns. IEEE transactions on pattern analysis and machine intelligence **34**, 1785–98 (09 2012). https://doi.org/10.1109/TPAMI.2011.251
20. Woo, S., Park, J., Lee, J., Kweon, I.S.: CBAM: convolutional block attention module. CoRR **abs/1807.06521** (2018), http://arxiv.org/abs/1807.06521
21. Wu, Y., Ji, Q.: Facial landmark detection: a literature survey. CoRR **abs/1805.05563** (2018), http://arxiv.org/abs/1805.05563
22. Xiao, S., Feng, J., Xing, J., Lai, H., Yan, S., Kassim, A.: Robust facial landmark detection via recurrent attentive-refinement networks. In: European conference on computer vision. pp. 57–72. Springer (2016)
23. Zadeh, A., Chong Lim, Y., Baltrusaitis, T., Morency, L.P.: Convolutional experts constrained local model for 3d facial landmark detection. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 2519–2528 (2017)
24. Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., Li, S.Z.: Faceboxes: A CPU real-time face detector with high accuracy. CoRR **abs/1708.05234** (2017), http://arxiv.org/abs/1708.05234