ABSTRACT

The biometric technique of iris recognition is considerably limited by the cost of optical devices and user inconvenience. Periocular-based methods are an alternative means of biometric authentication because they do not require expensive equipment. Moreover, the resulting data are suitable for biometrics because they include features such as eyelashes, eyebrows, and eyelids. However, conventional periocular-based biometric authentication methods use limited sets of features that are dependent on the selected feature extraction method, resulting in relatively poor performance. Therefore, we propose a deep-learning-based method that actively utilizes the various features contained in periocular images. The method maintains the mid-level features of the convolutional layers and selectively utilizes features useful for classification. We compared the proposed method with previous methods using public and self-collected databases. The experimental results show that the equal error rate is less than 1%, which is superior to the previous methods. In addition, we propose a new method to analyze whether mid-stage features have been utilized. As a result, it was confirmed that this approach, which utilizes the mid-level features, can effectively improve the feature extraction performance of the network.

INDEX TERMS

Convolutional neural networks, deep learning, mid-level features, periocular biometrics, periocular recognition.

I. INTRODUCTION

With the increasing popularity of virtual reality and head-mounted displays (HMDs), there is growing interest in the security and user authentication of HMDs [1], [2]. Particularly, biometric authentication is widely used in various fields because by definition it is portable and has little risk of forgetting or losing the means of authentication [3]. Fingerprints are a representative biometric authentication method. They are commonly used because of their convenience, uniqueness, and economy. However, recognition errors due to moist finger surfaces and the possibility of damage to fingerprints are often identified as problems [4]. In addition, the requirement of additional user cooperation of such a contact-type authentication causes inconvenience.

Face-based authentication is a non-contact, camera-based method that has less inconvenience. However, it offers significantly reduced performance because of external factors, such as changes in illumination and resolution, and it requires considerable computing resources [5]. Additionally, this is an unacceptable method in HMD environments where a display device must be worn on the face. Another non-contact authentication method is iris recognition, which is highly accurate, unlikely to be damaged, and suitable in HMD environments [6]. However, iris recognition has no universal usage because it requires an expensive high-resolution camera capable of capturing a fine iris pattern.

The biometric authentication method of periocular imaging has recently been considered as a means of overcoming these problems [7]. The periocular region is defined as the area including eyelashes, eyebrows, and eyelids, as well as the iris areas. This region contains features such as eye shape, eyelid shape, pores, skin texture, eyelash shape, and eyebrow shape [8], [9]. Compared to the fine pattern of the iris, these features can be used for identification because they do not require a high-resolution image and differ for each person [10].

In this paper, we propose a new deep learning architecture for periocular biometric. The paper has the following contributions. 1) We propose a network structure that can
improve classification performance by using mid-level features. 2) The performance of the proposed method is verified using public databases and self-collected data, and it is compared with state-of-the-art methods in the field. 3) In the neural network, a new analysis method is proposed to find a step that contributes significantly to the results, and it is used in the analysis of the proposed network.

II. RELATED WORKS

Karen et al. investigated what factors in the periocular region can serve as important features in identity recognition [11]. Consequently, identified factors such as eyelashes, tear ducts, eye shape, eyelids, eyebrows, the outer corner of the eyes, and the skin. Miller et al. analyzed the performance of a periocular-based authentication method by using the local binary pattern (LBP) descriptor and showed that eyelid and eye contour information are useful in people identification [12]. Bharadwaj et al. improved authentication performance by using a score-level fusion of the global matcher and circular LBP [13]. Woodard et al. fused LBP with a color histogram to exploit information such as skin texture, color distribution, and so on [14], [15]. Park et al. improved authentication performance by integrating LBP with the scale-invariant feature transform (SIFT) and histogram of oriented gradients, allowing the consideration of various factors, such as skin and eyebrows [7], [16].

In addition, there have been attempts to improve authentication performance using various periocular features [17]. Considering that the performance of these methods may be degraded due to distortion, pose, illumination, and so on, Padole and Proenca proposed the use of homomorphic filters and self-quotient images [18], [19]. Proenca and Briceno developed a globally coherent elastic graph matching algorithm to compensate for distortions caused by changes in expression, which improved the performance of previous methods [20].

Most of the studies mentioned above do not extract only one feature from the periocular image, but extract and fuse several different characteristics. This fusion improves performance, demonstrating that various meaningful features exist in the periocular region. It is also notable that some features have a negative effect. However, among the several features distributed in the periocular region, only a few are currently used, and there has been no research on which features should be utilized and supplemented for optimal performance.

Recently, Kumari et al. [21] noted that researchers are moving toward a deep-learning-based approach to increase recognition performance by lowering dependence on these feature extraction methods. As neural networks can learn useful expressions from the data provided and achieve significant improvements over previous methods, convolutional neural networks (CNN) have also been used for biometric techniques, such as face recognition [22], [23].

Research on deep-learning-based periocular authentication in near-infrared (NIR) is at an early stage. Recently, Zhao et al. [24] proposed AttNet, which reportedly showed outstanding performance. Based on the assumption that important information for identification is concentrated in eyebrows and the eye, the authors used the method of detecting region of interest and weighting the features of the region. This enabled the achievement of state-of-the-art accuracy for various published datasets. However, as mentioned in [21], periocular authentication can still be improved through the development of feature extraction and learning methods. Therefore, we propose a CNN-based periocular authentication method, including a new feature extraction method, and we compared it with AttNet, a state-of-the-art technology, and hand-crafted feature-based methods (LBP, SIFT) [25].

III. PROPOSED METHOD

The proposed classification model for periocular authentication consists of a feature extraction network that expresses the features of an image as a vector, and a Siamese mechanism for learning the network. The feature extraction network has a structure in which intermediate features are not lost and can be used for model determination. In addition, the model contains a novel analysis method that analyzes whether the network uses the intermediate level features. Section III(A) describes the structure of the feature extraction network in detail, and Section III(B) describes this network is trained. Further, Section III(C) details the use of mid-level features based on gradients is analyzed.

A. FEATURE EXTRACTION NETWORK

According [26], neural networks tend to extract high-level features as they get deeper. Therefore, it is important to design a model with the optimal depth to extract features suitable for a specific task. However, it is difficult to find the optimal type of features and depth suitable for a given task. If a depth that is too shallow is selected, it would be difficult to utilize high-level features; if the depth is too deep, performance may deteriorate due to loss of low-level features. There have been successful attempts at improving performance by maintaining low-level features and high-level features extracted by the deep neural network [27]–[30].

Our key assumption in this study is that useful features for classification can be distributed not only in high-level features that are the output of a deep network, but also in mid-stage features. This assumption is based on [11], which reports that both shape of eye (high-level feature) and texture (low-level feature) are useful for classification. Therefore, we used a network structure that maintains intermediate features and can directly affect classification.

We developed a traditional CNN as shown in Fig. 1 to exploit the mid-level features. The activation map \( (a) \), which is the output of a specific step convolution, is not only input to the next convolution step, but also directly connected to the final feature vector. For this connection, the activation map is vectorized using global average pooling. As each channel of the activation map is generated by an already learned fixed weight kernel, this vector contains the global information of each channel. The pooled vector is connected to the fully
FIGURE 1. Structure of the proposed network that extracts features while maintaining intermediate-stage features.

connected layer to map to the same-dimension vector \(v\). Through this, the weight of each feature for classification is given, and the interactive effect between channels can be learned. The strategy of using the global feature of the activation map has been used to recalibrate the activation map in squeeze and excitation network (SENet), and it has been proven useful for improving the performance of existing networks [31]. In this present work, the extracted vectors are not used for recalibration, but the vectors of each step are concatenated and used as a feature vector for an input image. The process of extracting a feature vector from the activation map can be summarized as (1).

\[
v^m_n = \sum_{c=1}^{C} w^{c,m}_n \left( \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} a^c_n(i,j) \right)
\]  

where \(a^c_n\) means the activation map, which is the output of \(n\)-th layer of the network, and \(v^m_n\) is the feature vector generated from \(a^c_n\); \(H, W, C\) are height, width, and total number of channels of the activation map, respectively; \(a^c\) is \(c\)-th channel of the activation map, and \(i \text{ and } j\) of \(a^c(i,j)\) are the rows and columns, respectively; \(w^{c,m}_n\) means the weight for the fully connected layer performed on the global average pooling of \(a^c_n\); \(m\) denotes the element number of the feature vector, and \(w^{c,m}\) denotes the weight corresponding to the \(c\)-th channel of the activation map and \(m\)-th element of \(v\). The proposed method reduces dependence on the final feature extracted from a fixed-depth network through this vectorization mechanism and enables the intermediate-stage feature to be selectively used according to importance.

B. NETWORK LEARNING METHOD

In a traditional CNN, as user information is stored in the model’s parameters through the training process, it is vulnerable to additions or changes in the user. A Siamese network is a neural network structure that improves the problems suffered by previous CNNs in image comparison [32]. This makes it possible to learn the differences in characteristics between two images that are meaningful to classification without having to learn the characteristics of each class individually. Therefore, it is unnecessary to relearn a property to modify the characteristics of a user, making it more efficient than a traditional CNN, which learns all the characteristics of the task.

As shown in the Fig. 2, a Siamese network is a structure that extracts each feature vector through the same network for two input images and calculates the distance between them. As the distance is proportional to the difference between features, it is possible to judge the identity of two images. To learn this network, it is required to select the appropriate loss function that can minimize the distance in the positive class and maximize the distance in the negative class. This can considerably affect the learning of the model. Hence, there have been various studies to improve the performance [24], [33], [34]. Among these, it has been experimentally verified that the recently proposed distance-driven sigmoid cross-entropy (DSC) loss shows superior performance compared to the previously used methods.

\[ p = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{d - d_0}{s} \right) \right) \]

\[ d = \sqrt{\sum_{i=1}^{H} \sum_{j=1}^{W} (v_1(i,j) - v_2(i,j))^2} \]

FIGURE 2. Structure of Siamese network for model training. Feature vector for a pair of images are extracted by one network, and the loss is calculated by the distance between the vectors.
belong to the same class.

\[ p = \frac{1}{1 + e^{-(b-ad^2)}} \]  

(2)

where \(a\) and \(b\) are pre-defined hyper parameters for the linear transformation of \(d\), and are set as 10 and 5 respectively as in [24].

\[ L_{DSC} = -\left(t \log p + (1-t) \log(1-p)\right) \]  

(3)

DSC loss is calculated using the cross-entropy function (see (3)), which has been widely used as a loss for probability and the calculated \(p\). \(t\) is the label of the input; a positive class at 1 and negative class at 0. It can contribute to better discrimination by improving the impact of hard margins (a problem with previous methods) and increasing learning efficiency. Therefore, we used DSC loss in training the model.

C. CONTRIBUTION OF BRANCH ANALYSIS METHOD

The proposed method does not depend on high-level features extracted from convolutional layers, and mid-level features can also be selectively used by adjusting the weight. However, the neural network is a complex structure, making it difficult to logically and clearly explain the basis of the results. Consequently, it is difficult to confirm whether the mid-level features significantly influence the results. Therefore, we propose a new analysis method to confirm the utilization of intermediate-stage features. Grad-CAM, which is a heatmap analysis method that visualizes meaningful features in network results, uses gradients to visualize pixels in the activation map with a high contribution to the model’s outputs [35]. Gradient refers to the amount of change in the result according to the change in pixel. It can be understood as the magnitude of influence of a pixel on the result.

As shown in Fig. 3, the proposed feature extraction method is divided into two branches: network branch where convolution is performed to further extract features and vectorization branch where pooling is performed to connect directly to the last step. Here, the gradient of the network branch \((G_u)\), gradient of the vectorization branch \((G_v)\) and unbranched gradient \((G_n)\) including the two branches can be obtained. As \(G_u\) contains the gradient of two branches, the gradient of the branch with the greater influence on the result also has a stronger influence on \(G_u\). For example, if the influence of \(G_v\) on the result is substantial and the influence of \(G_n\) is very small, then \(G_u\), which is the influence of the layer, is closer to \(G_v\). Therefore, the similarity between \(G_u\) and the gradient of each branch can be obtained and compared to identify branches with high influence in a specific layer. A large influence on the result can be considered to mean that features of the branch contribute highly and are useful to the result.

\[ s = \left( \frac{\sum_{i=1}^{n} G_1^i \times G_2^i + 1}{\sum_{i=1}^{n} (G_1^i)^2 \times \sum_{i=1}^{n} (G_2^i)^2} \right) \times \frac{1}{2} \]  

(4)

The similarity \((s)\) of the two gradients can be obtained as a cosine similarity robust to the scale of the gradient, and it is defined as (4). \(G_1\) and \(G_2\) are the two gradients to calculate similarity; \(n\) is the size of the activation map (height \(\times\) width \(\times\) channels); \(G^i\) is the \(i\)-th value of the gradient. The cosine similarity is a \(-1\) to 1 scale; the closer the two gradients are, the closer it is to 1. In this paper, to remove the sign, it is normalized to a value between 0 and 1. In this way, the similarity of \(G_u\) to \(G_v\) (\(s_v\) and similarity of \(G_u\) to \(G_n\) (\(s_n\)) can be obtained, and a branch having high contribution to the result can be identified by comparing them. The proposed method of analyzing the contribution is universally applicable to all networks in which branching occurs.

IV. EXPERIMENTS AND ANALYSES

A. MODELS

In this study, we conducted experiments using two types of networks: traditional CNN (plain CNN) and simple network using shortcut of residual network (ResNet) [36]. In addition, to observe the change in performance by depth, experiments on deeper networks were added, and a total of four models (plain CNN, ResNet, deep plane CNN, and deep ResNet) were trained and tested respectively. The detailed structures of the networks are shown in Table 1.

Plain and ResNet models are identical, except for the shortcut structure, and deep networks have twice the number of
TABLE 1. Detailed structure of the four models used in the experiment.

| Unit  | # Layer | Type (plain) | Type (deep plain) | Type (ResNet) | Type (Deep ResNet) | # Output channels |
|-------|---------|--------------|-------------------|---------------|-------------------|------------------|
| Block1| 1       | Conv (7 × 7), stride=2 | Max pooling (3 × 3), stride=2 |               |                   | 32               |
|       | 2       |              |                   |               |                   |                  |
|       | 3       |              |                   |               |                   |                  |
|       | 4       |              |                   |               |                   |                  |
|       | 5       |              |                   |               |                   |                  |
| Block2| 6       |              |                   |               |                   | 32               |
|       | 7       |              |                   |               |                   |                  |
|       | 8       |              |                   |               |                   |                  |
| Block3| 9       |              |                   |               |                   |                  |
|       | 10      |              |                   |               |                   |                  |
|       | 11      |              |                   |               |                   | 64               |
|       | 12      |              |                   |               |                   |                  |
|       | 13      |              |                   |               |                   |                  |
|       | 14      |              |                   |               |                   |                  |
| Block4| 15      |              |                   |               |                   |                  |
|       | 16      |              |                   |               |                   |                  |
|       | 17      |              |                   |               |                   | 128              |
|       | 18      |              |                   |               |                   |                  |
|       | 19      |              |                   |               |                   |                  |
|       | 20      |              |                   |               |                   |                  |
| Block5| 21      |              |                   |               |                   |                  |
|       | 22      |              |                   |               |                   |                  |
|       | 23      |              |                   |               |                   | 256              |
|       | 24      |              |                   |               |                   |                  |
|       | 25      |              |                   |               |                   |                  |
|       | 26      |              |                   |               |                   |                  |
| Pool  | 27      | Vector (4)² | Vector (7)         | Vector (5)    | Vector (8)        | 32               |
|       | 28      | Vector (10)  | Vector (13)        | Vector (11)   | Vector (14)       | 64               |
|       | 29      | Vector (16)  | Vector (19)        | Vector (17)   | Vector (20)       | 128              |
|       | 30      | Vector (22)  | Vector (25)        | Vector (23)   | Vector (26)       | 256              |
| Concat| 31      | Concatenate (27, 28, 29, 30) |               |               |                   | 480              |

¹ Shortcut (x, y): This is a shortcut structure that adds the output of layer x to the output of layer y. If the channels of the two activation maps are different, perform 1 × 1 convolution on the output of layer x to match the channels.
² Vector (x): Global average pooling is performed on the output of layer x and vectorized as a fully connected layer.

The models were learned using images of 246 classes (4,908 images), that is, 60% of the 410 classes of CASIA-Iris-Lamp. The data were divided based on the class so that a subject used for learning was not used for testing. For validation, 82 classes (1,607 images), that is, 20% of the total, was used, and the remaining 20% (1,616 images) were used for the test. In addition, tests were conducted with our data to verify the generalized performance of the trained model. Through these cross-tests, we could determine not only the general classification performance of the model, but also its robustness to new characteristics (e.g., angle of shot, direction of eye, etc.) absent in the learning data.

B. DATASETS

We used the data we collected and CASIA-Iris-Lamp [37], a open database in NIR environment, to train and test the model. CASIA-Iris-Lamp is a set of images obtained by changing only the brightness of lighting in a fixed environment as shown in Fig. 4 (a). It consists of 16,212 images of 819 classes with left and right eyes taken for 411 subjects. The left eye data (410 classes, 8,131 images) alone were used in the experiment.

Our data was collected under the conditions of changing the brightness of the light, angle of shooting, and direction of the gaze of the NIR periocular image, as shown in Fig. 4 (b). This dataset is composed of a total of 500 observations from 20 left-eye images of 25 subjects.

The models were learned using images of 246 classes (4,908 images), that is, 60% of the 410 classes of CASIA-Iris-Lamp. The data were divided based on the class so that a subject used for learning was not used for testing. For validation, 82 classes (1,607 images), that is, 20% of the total, was used, and the remaining 20% (1,616 images) were used for the test. In addition, tests were conducted with our data to verify the generalized performance of the trained model. Through these cross-tests, we could determine not only the general classification performance of the model, but also its robustness to new characteristics (e.g., angle of shot, direction of eye, etc.) absent in the learning data.

C. PERFORMANCE OF MODELS

We trained AttNet and the proposed methods using the training set of CASIA-Iris-Lamp, and we tested them and hand-crafted feature-based methods with the same database test set.

Fig. 5 illustrates the results of the model evaluation using the receiver operating characteristic (ROC) curve. The performance shown in this figure indicates that the performance of learning-based methods is consistently superior to that of hand-crafted feature-based methods. In addition, it was
confirmed that the proposed methods showed higher overall performance than AttNet, which is a state-of-the-art technology in this field. The two methods performed best by using ResNet as the backbone, and both had equal error rates (EERs) of less than 1%. The performance of the proposed methods using a plain CNN as the backbone also showed better results than the previous studies. The EER of the deep plain CNN was 1.59%, which was slightly better than the shallow one.

The performance when the false acceptance rate (FAR) is low is a vital indicator in the field of biometrics and is considered as the ability to separate difficult samples. The false reject rate (FRR) when FAR of the models are less than 0.1% and 0.01%, respectively, and EER are shown in Table 2.

The results presented in Table 2 show that the models that use ResNet as backbones are consistently superior, even when the FAR is very low. Particularly, ResNet and deep ResNet have about 2.72% and 2.58% FRRs when FARs are less than 0.1%, respectively. This is a lower error rate than the EER of AttNet used in the experiment. Thus, it can be seen that the proposed method is robust against both challenging and easy-to-classify data.

The ROC curves when the trained models are tested with our dataset are shown in Fig. 6. The results confirm that the performance of all methods, including our method, deteriorated overall. This deterioration reflects the characteristics of learning-based methods that tend to be weaker on new data, which is also a reason why performance is relatively difficult to classify. Moreover, our database contains various factors (angle of shot, direction of eye, etc.), compared to CASIA-Iris-Lamp. This is supported by the fact that performance based on hand-crafted features unrelated to the learning data also degrades, compared to the previous experiment. Models using the proposed methods obtained relatively better results than the previous methods on the new database. This shows that the proposed method has a relatively good generalization performance. The FRR when FAR of the models are less than 1%, and EER from experiments using our data are shown in Table 3.
FIGURE 7. Samples of image pairs used for comparison and heatmap: (a) Imposter image pairs; (b) Genuine image pairs.
TABLE 3. Performance comparison of the methods used in the experiment using our dataset.

| Method      | EER (FAR < 1%) |
|-------------|----------------|
| LBP [25]    | 20.02%         |
| SIFT [25]   | 36.98%         |
| AttNet [24] | 21.10%         |
| Plain CNN   | 16.78%         |
| Deep plain CNN | 15.66%     |
| ResNet      | 13.83%         |
| Deep ResNet | 11.51%         |
|             | 50.53%         |
|             | 89.73%         |
|             | 57.18%         |
|             | 54.51%         |
|             | 52.16%         |
|             | 50.93%         |
|             | 45.54%         |

D. LEARNING ANALYSIS

Heatmap analysis was performed to visualize the importance of certain pixels of the input image to the classification. The heatmap results for some samples are shown in Fig. 7. The two images in the upper rows are samples of image pairs used for comparison, and the lower rows are the heatmap results for those images. The closer it is to red, the more important it is.

Fig. 7 (a) illustrates the heatmap when comparing image pairs of different classes, and (b) illustrates the heatmap when comparing image pairs of the same class. In both cases, it can be seen that pixels in various areas, such as the boundary of the eye, eyelids, tear ducts, eyelashes, eyebrows, and skin areas, are activated. This is clearly shown to be related to the structure information of the eye, and it is consistent with the common properties found in periocular. Additionally, in the heatmap of the two compared images, it can be seen that pixels corresponding to the same portion of the image are activated together. This can be confirmed not only in (b), which is an overall similar type of the eye image, but also in (a), which shows a completely different eye shape. This means comparing the same features for classification. While most of the strong features are concentrated on the eyes and eyebrows as mentioned in AttNet, it can be seen that skin information is also useful in some image pairs (when the skin texture or curvature is significantly different or the shape of the eyes is similar). This shows the distinction between AttNet and the proposed methods, and the strong performance of our models shows that other areas beside the eye and eyebrow can also be useful for classification. These results prove that the model is well trained to extract reasonable features for classification and that the results can be derived by correctly comparing the features.

Samples of images misclassified by the proposed model are shown in Fig. 8. (a) is a case of incorrectly classifying image pairs of the same class. The intensity of the overall feature appears to be weakened because the left image is slightly out of focus; hence, relatively clear reflected light among them is utilized as a distinct feature. (b) is a case where samples of different classes are classified incorrectly. The misclassification is likely due to the characteristic similarities of the eyelids and eyelashes, which are noticeable features even though there are some recognizable features. This shows that our model can rely on more powerful features than more detailed features in few cases. Our study clearly shows superior performance, but this is a persistent limitation of deep-learning-based technology that requires attention.

E. GRADIENT SIMILARITY ANALYSIS

Gradient analysis was performed on the four models used in the experiment using the analytical methods mentioned in Section III(C).

Fig. 9 illustrates how $s_v$ and $s_n$ of the four models used in the experiment changed with network depth. Overall, $s_v$ increases as the network deepens, but $s_n$ tends to decrease. This is because the vectorization branch contains only the gradient of the feature vectors extracted from the layer, but the network branch contains the gradient of all feature vectors extracted later. Here, Block 4 of the models in (b) and (d) is noteworthy. In these two cases, it can be seen that the vectorization branch contributes more to the results than the network branch. As the network branch of Block 4 contains the gradient of the feature vector of Block 5, this result shows that the feature vector extracted from Block 4 is more useful in the results than the feature vector extracted from Block 5. Through this analysis, plain CNN-based models can be seen that the activation maps in Block 5 contribute the most to the results, and ResNet-based models show that the activation maps in Block 4 contribute the most to the results. This shows that the feature vectors extracted by deep networks may not necessarily be the most useful for the results, and the excellent performance of our model, which exploits intermediate features, can be a strong support. In addition, the characteristics...
of the irrelevant classification steps can be adjusted in weight to prevent them from influencing the results. Nevertheless, the tendency for the difference between the two similarities to become narrower shows that mid-level features can be useful to some extent in (a) and (c).

V. CONCLUSION

The periocular biometric is considered to use the lower frequency components of the image relative to the iris biometric. Therefore, using only the final feature of the convolutional layer has a problem in that characteristics of a specific frequency band can be ignored. Therefore, by reflecting the characteristics of the mid-level layer, by presenting a method in which all of the various frequency bands of the image can be considered, a discriminative feature not limited to a specific frequency was produced. Consequently, it was possible to derive better performance than the previous methods. And through heatmap analysis, it shows that the proposed method can learn reasonable features for classification.

In this paper, we proposed a novel CNN architecture for periocular authentication and an analysis method. The proposed structure includes a feature extraction network that prevents the loss of mid-level features and selectively utilizes features that are important for classification. The proposed method was compared with previous studies using a large-scale public database, CASIA-Iris-Lamp, and a self-collected database. It was confirmed that our method performs better than existing methods with less than 1% EER. In addition, features that are useful for classification were identified through heatmap analysis, and the proposed analysis method was used to verify that the model uses intermediate-level characteristics. These results are the basis for our assumption that features useful for classification are distributed in the middle of the convolutional layers. However, we confirmed in a few cases that misclassification may occur from strong features that do not aid classification. This is a persistent problem of deep-learning-based methods; thus, we will improve the proposed method to utilize weak but useful features in the future. In addition, future studies may expand this study through improvements in network structure and learning methods.

REFERENCES

[1] C. E. Rogers, A. W. Witt, A. D. Solomon, and K. K. Venkatasubramanian, “An approach for user identification for head-mounted displays,” in Proc. ACM Int. Symp. Wearable Comput. ISWC, 2015, pp. 143–146, doi: 10.1145/2802083.2808391.

[2] C. George, M. Khamis, E. von Zezschwitz, H. Schmidt, M. Burger, F. Alt, and H. Hussmann, “Seamless and secure VR: Adapting and evaluating established authentication systems for virtual reality,” in Proc. Workshop Usable Secur., 2017, pp. 1–12, doi: 10.14722/usec.2017.23028.

[3] S. Prabhakar, S. Pankanti, and A. K. Jain, “Biometric recognition: Security and privacy concerns,” IEEE Secur. Privacy, vol. 1, no. 2, pp. 33–42, Mar. 2003, doi: 10.1109/MSP.2003.1193209.

[4] M. Alsadi, “Physiological biometric authentication systems, advantages, disadvantages and future development: A review,” Int. J. Sci. Technol. Res., vol. 4, pp. 285–289, Oct. 2015.
H. Proença and J. C. Briceño, “Periocular biometrics: Constraining the elastic graph matching algorithm to biologically plausible distortions,” IET Biometrics, vol. 3, no. 4, pp. 167–175, Dec. 2014, doi: 10.1049/iet-bmt.2013.0039.

P. Kumari and K. R. Seeta, “Periocular biometrics: A survey,” J. King Saud Univ.–Comput. Inf. Sci., to be published. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157818313302, doi: 10.1016/j.jsui.2019.06.003.

Y. Taigman, M. Yang, R. Ranzato, and L. Wolf, “DeepFace: Closing the gap to human-level performance in face verification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1701–1708, doi: 10.1109/CVPR.2014.220.

Y. Sun, X. Wang, and X. Tang, “Deep learning face representation from predicting 10,000 classes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1891–1898, doi: 10.1109/CVPR.2014.244.

Z. Zhao and A. Kumar, “Improving periocular recognition by explicit attention to critical regions in deep neural network,” IEEE Trans. Inf. Forensics Security, vol. 13, no. 12, pp. 2937–2952, Dec. 2018, doi: 10.1109/TIFS.2018.2833018.

S. Kim and E. C. Lee, “Periocular biometric authentication methods in head mounted display device,” Int. J. Eng. Technol., vol. 7, no. 3, 2018. [Online]. Available: https://www.sciencepubco.com/index.php/ijet/article/view/22522, doi: 10.14419/ijet.v7i3.24.22522.

M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in Proc. ECCV, Zurich, Switzerland, 2014, pp. 818–833, doi: 10.1007/978-3-319-10590-1_53.

L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proc. ECCV, Munich, Germany, Sep. 2018, pp. 801–818, doi: 10.1007/978-3-030-01234-2_49.

W. Tian, Z. Wang, H. Shen, W. Deng, Y. Meng, B. Chen, X. Zhang, Y. Zhao, and X. Huang, “Learning better features for face detection with feature fusion and segmentation supervision,” 2018, arXiv:1811.08557. [Online]. Available: http://arxiv.org/abs/1811.08557.

Y. Zhang, J. Chu, L. Leng, and J. Miao, “Mask-refined R-CNN: A network for refining object details in instance segmentation,” Sensors, vol. 20, no. 4, p. 1010, Feb. 2020, doi: 10.3390/s20041010.

J. Chu, Z. Guo, and L. Leng, “Object detection based on multi-layer convolution feature fusion and online hard example mining,” IEEE Access, vol. 6, pp. 19959–19967, 2018, doi: 10.1109/ACCESS.2018.2815149.

J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, “Signature verification using a Siamese time delay neural network,” Int. J. Pattern Recognit. Artif. Intell., vol. 7, pp. 669–687, 1993, doi: 10.1142/S0218001493000339.

S. Chopra, R. Hadsell, and Y. LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2005, pp. 539–546, doi: 10.1109/CVPR.2005.202.

K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” J. Mach. Learn. Res., vol. 10, pp. 207–244, Feb. 2009.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

Biometrics Ideal Test. Accessed: May 27, 2020. [Online]. Available: http://biometrics.idealtest.org/