IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 20, 2020, accepted November 26, 2020, date of publication December 1, 2020,
date of current version December 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3041470

Boosting Network Weight Separability via Feed-Backward Reconstruction

JONGMIN YU AND HYEONTAEK OH, (Member, IEEE)
Institute for IT Convergence, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea

Corresponding author: Jongmin Yu (andrew.yu@kaist.ac.kr)

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government [Ministry of Science and ICT (MSIT)] (A Study on 5G based Intelligent IoT Trust Enabler) under Grant 2020-0-00833.

ABSTRACT This paper proposes a new evaluation metric and a boosting method for weight separability in neural network design. In contrast to general visual recognition methods designed to encourage both intra-class compactness and inter-class separability of latent features, we focus on estimating linear independence of column vectors in weight matrix and improving the separability of weight vectors. To this end, we propose an evaluation metric for weight separability based on semi-orthogonality of a matrix, Frobenius distance, and the feed-backward reconstruction loss, which explicitly encourages weight separability between the column vectors in the weight matrix. The experimental results on image classification and face recognition demonstrate that the weight separability boosting via minimization of feed-backward reconstruction loss can improve the visual recognition performance, hence universally boosting the performance on various visual recognition tasks.

INDEX TERMS Neural networks, feed-backward reconstruction, latent space.

I. INTRODUCTION

Representation learning based on deep learning methods has been achieved remarkable performances in various visual recognition studies such as image classification [1]–[3], object recognition [4]–[7], face recognition [8]–[11], and person re-identification [12]–[15]. A key of these successes is an effective feature extraction via non-linear and cascaded kernel structures of deep neural networks. However, in addition to extracting feature using locally connected and shared weight structure of a convolutional neural network, the neural networks' decision metrics based on Euclidean geometry have been demonstrating that embedded features on inner product space are sufficient to achieve superior recognition accuracies to the conventional discriminative approaches [16]–[19] based on hand-crafted features in various recognition tasks.

In recent years, not only studies to improve the representation learning capabilities of convolutional neural networks based on modifying structures of networks [3], [20] but also the discriminative embedding methods for latent features into Euclidean space have been actively studied [11], [21], [22]. Feature learning constrained on l_2-norm space [23] was proposed to improve the discriminative power of learned features by regularizing the vector scale of each data point. Angular cost function [24], Large-margin softmax function [21], and Sphereface [11] were proposed to improve the discriminative properties of learned features based on the understanding of the principle of cosine similarity. [22] presented the 'center loss' based clustering methodology and showed that even though the function is non-differential, it can improve the discriminative power of learned features during network training. Intuitively, these approaches were typically concentrated on the embedding latent features into some constrained space using restriction methodologies for the features by reinforcing of intra-class compactness and inter-class separability [21]. Even though these approaches have achieved remarkable performance in diverse visual recognition tasks, improving separability of learned weight kernels is one of the challenging issues. In recognition tasks by computing vector similarities between weight and latent features, inner product correlation between weight vectors can significantly affect the performance of the recognition models.

In this paper, we formulate the evaluation metric for weight separability and propose a method to boost the separability of a network weight in a last fully connected layer. Figure 1 shows the intuitive concepts of weight separability, inter-class separability, and intra-class compactness. Inter-class separability and intra-class compactness are computed based on the distribution of the extracted latent features, and these can be defined by the Euclidean distances between the class centroids and the expectation of the distance between the same class samples, respectively. Compared with these sort of feature-driven measurements, the weight separability is

The associate editor coordinating the review of this manuscript and approving it for publication was Fan Zhang.

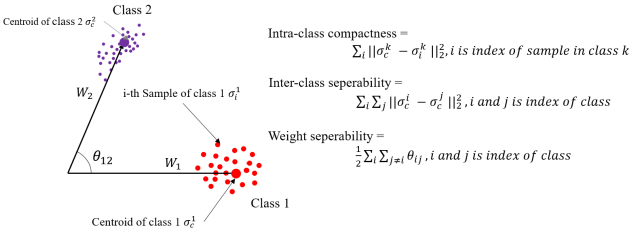

FIGURE 1. The intuitions of intra-class compactness, inter-class separability, and weight separability on neural networks. The dots denote the latent features extracted from the networks. σ_j^c and σ_j^k are the centroid and the k^{th} sample of the j^{th} class, respectively. θ_{ij} is the angle between the i^{th} and j^{th} weight vectors.

defined as an angle between the weight kernel for classifying the features.

Although one-hot encoded label vectors already induce the weight vectors of a last fully connected layer to be orthogonal in general approaches, there is a possibility for further improvement of the discriminative power of the learned features by revising loss functions or structural details [11], [21], [22]. Therefore, we focus on the semi-orthogonalization of a weight matrix, which is a process to find a set of orthogonal vectors that can span a specific subspace. The set of orthogonal vectors takes linear independence between elements. The orthogonalization of weight in neural network is considered as a regularization method to reduce the correlation between the detected features by networks [25]. Our main hypothesis is that the separability between vectors of a weight matrix is related to the recognition performances, and it can be evaluated by the linear independence of the weight matrix. The purpose of this paper, therefore, is to prove the hypothesis and apply this intuition to improving representation learning capability of deep neural networks for various visual recognition tasks.

Our key contributions are as follows. First, we define and demonstrate a quantitative evaluation metric for weight separability, which can be used for high-dimensional features without any dimension reduction method and visualization task. Second, we propose a straightforward method to boost the separability of the weight vectors explicitly during network learning. The experimental results show that the proposed method can improve the performance of image classification and face recognition tasks.

This paper is organized as follows. In Section II, we introduce the basic concepts of kernels' linearity and separability on neural networks. We explain how to evaluate weight separability of neural networks in Section III. Section IV contains the explanation for the approach to improve the weight separability explicitly, called feed-backward reconstruction. The explanation of the experimental setting and the analysis of the results are shown in Section V. In Section VI, we conclude this paper and provide our future works.

II. LINEARITY AND SEPARABILITY

In commonly used deep learning structures for visual recognition tasks, a fully connected network is used to assign the label by calculating the confidence based on vectorial or probabilistic approaches. The column vectors of weight matrix in last fully connected neural network are used to decide recognition classes of inputs based on the vector similarity by the inner product: $w_i \cdot \alpha = \| w_i \| \| \alpha \| \cos\theta_i$, where w_i is the i^{th} column vector of weight matrix $W = [w_1, w_2, w_3, \ldots, w_n] \in R^{m \times n}$, where m and n are the row and column dimensionalities of weight matrix, and α and θ_i are a latent feature vector and the angle between w_i and α, respectively. In fully connected networks positioned at the last layer, the figures m and n indicate that the dimensionality of input feature and the number of classes. In recognition task using the fully connected layer, the class of a latent feature is assigned as the index of column vector which takes the largest value calculated by the inner product defined as follows:

$$ID = \text{argmax}_i f(\alpha \cdot w_i + b), \tag{1}$$

where i is the index of column vectors in a weight matrix, and α is a latent feature. f is an activation function in a network. w_i and b are i^{th} column vector in the weight matrix and a bias term, respectively. In further sections, the bias term can be deleted to simplify mathematical expressions and improve experimental efficiency.

In this paper, we argue that linear independence of the column vectors in a weight matrix has a relation to the separability of weight vectors which can influence performance of various recognition tasks based on vector similarities. To justify our argumentation, we conduct a simple experiment using MNIST dataset [1]. In this experiment, we use samples of classes: 0, 1, and 5 only. We compare two neural networks that have the same structure but trained in different ways. We also have employed LeNet [1] structure in our experiment. One network is trained by forcing with linearly dependent column vectors, and the other is composed of linearly independent column vectors in a final layer. We initially assign random real numbers between -1 to 1, and conduct QR decomposition to take the weight matrix composed of linearly independent column vectors. The formula for the above process is represented as follows:

$$W = \hat{W}R, \hat{W}\hat{W}^T = \hat{W}^T\hat{W} = I, \tag{2}$$

where $W \in R^{m \times n}$ is a randomly initialized weight matrix, $\hat{W} \in R^{m \times n}$ is an orthogonal matrix composed of linearly independent column vectors, and $R \in R^{n \times n}$ is an upper triangular matrix. We employ a square matrix ($W \in R^{10 \times 10}$) in this experiment even though QR decomposition is applied to $m \times n$ matrix with $m \geq n$. To maintain the linear independence to the weight vectors during learning, the parameter in the final weight matrix is not updated during training each model.

We have reduced the dimensionality of latent features as three (i.e., three dimensions) using principal component analysis (PCA) to visualize our results. The dimension reduction results would be improved when advanced PCAs such as Zhou *et al.* [26] and Chen *et al.* [27] are applied. However, the objective of this experiment is to monitor the effect of linearity of weight kernel on the performance of various recognition tasks. we have, therefore, applied basic PCA without

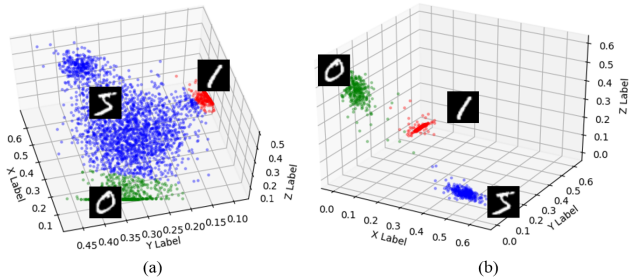

FIGURE 2. The comparison of the distributions of latent features between the normally trained network (a) and the network (b) which is forced to have the linear independence in their weight matrix. The green, red, and blue points are latent features extracted from input data of the 0, 1, and 5 classes, respectively.

any modification for this simple experiment. As visualization results in Figure 2, the weight matrix of a neural network composed of the column vectors, which take linear independence, shows better discriminative power in their distribution of latent features compared to the neural network that did not force the linear independence during network training.

III. WEIGHT SEPARABILITY EVALUATION

A. INTUITION

As the illustration in Fig 1 and the experimental results in Fig 2, the linearity of the column vectors in a weight matrix can influence recognition performances. We try to evaluate the weight separability using the orthogonality of a matrix. The property of orthogonal matrix is as follows: $QQ^T = Q^TQ = I$, where Q is a square matrix, and I is a corresponding identity matrix of Q. However, the dimensionality of the commonly used weight matrix W is not a square matrix, and also we can not guarantee that the weight matrix W is invertible in practical situations. Therefore, in this work, we employ the concept of a semi-orthogonal matrix. A non-square matrix A is semi-orthogonal if either $AA^T = I$ or $A^TA = I$, and it implies that A takes isometry property. With this notation, the linearity of a weight matrix $W \in R^{m\times n}$ is simply evaluated by calculating an error E defined as follows:

$$E(W, I) = W^TW - I_n, E(W, I) \in R^{n\times n}, \tag{3}$$

where W is a weight matrix, and I_n is the corresponded identity matrix of $n \times n$ dimension. The result of this subtraction operation is a matrix. When $E(W, I)$ are closer to a zero matrix, W can take stronger linearity. However, the matrix form is inappropriate to consider as a quantitative value to estimate the linearity. Moreover, in practice, above equation does not show the complete equivalence as mathematical semi-orthogonal. The cause of this inequivalence is a matrix structure of a neural network. To solve the inequivalence, the matrix notation for a final fully connected network is represented as follow:

$$\boldsymbol{\alpha} \cdot [w_1, w_2, w_3, \ldots, w_n] = o, \tag{4}$$

where $\alpha \in R^{1\times m}$ is the latent feature outputed from a previous layer which consisting of m of elements, $w_i \in R^{m\times 1}$ is i^{th} column vector in weight matrix W of the final layer, $o \in R^{1\times n}$ is the output of network, and n is the number of classes. In above notation, each output o_i $(i = 1, 2, 3, \ldots, n)$ is calculated as follows:

$$o_i = \alpha \cdot w_i = \sum_{j=1}^{m} \alpha^j w_{ij}, \tag{5}$$

where w_{ij} is j^{th} element of the i^{th} column vector w_i. In the above notations, the column vectors in weight matrix play a rule as a kernel to assign a specific class by computing vector similarity between the given feature α and each column vector w_i^f. In this work, we consider the separability of weight kernel so that we only consider the linear independent of column vectors of weight matrix W. Note that this principle can also be used for the network in which their row vector is used for the decision kernel.

B. METRIC DEFINITION AND MATHEMATICS

Since a matrix format in equation (3) is not suitable to quantitatively evaluate the weight separability, we employ Frobenius distance that can convert the matrix form to real-number. We define the quantitative metric based on Frobenius distance to evaluate the linearity of column vectors in a weight matrix. The metric $\epsilon(W)$ for separability of a weight matrix $W \in R^{m\times n}, m > n$ is defined by

$$\epsilon(W) = \frac{1}{n} \parallel W^TW - I_n \parallel_F^2. \tag{6}$$

n is the number of column vectors in the weight matrix, and I_n is an identity matrix with $n \times n$ dimension. The proposed metric computes the weight separability using Frobenius distance and regularizes it by dividing with the number of classes. The reason for the regularization with the number of classes are to provide the generalized evaluation metric invariant to the number of classes and to prevent the fluctuatation of the evaluation values according to the problem domain. In equation (6), $W^TW - I_n$ is represented as follows:

$$\begin{bmatrix} w_{11} & \ldots & w_{1n} \\ w_{21} & \ldots & w_{2n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \ldots & w_{mn} \end{bmatrix}^T \begin{bmatrix} w_{11} & \ldots & w_{1n} \\ w_{21} & \ldots & w_{2n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \ldots & w_{mn} \end{bmatrix} - \begin{bmatrix} 1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1 \end{bmatrix}$$
$$= \begin{bmatrix} \sum_{i=1}^m w_{i1}^2 - 1 & \ldots & \sum_{i=1}^m w_{i1}w_{in} \\ \sum_{i=1}^m w_{i1}w_{i2} & \ldots & \sum_{i=1}^m w_{i2}w_{in} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^m w_{i1}w_{in} & \ldots & \sum_{i=1}^m w_{in}^2 - 1 \end{bmatrix} \in R^{n\times n}, \tag{7}$$

where w_{ij} is i^{th} row and j^{th} column element in a weight matrix. By the two properties of matrix transpose:

$$\begin{aligned} (A^T)^T &= A, \\ (A - B)^T &= A^T - B^T, \end{aligned} \tag{8}$$

the result $((W^TW) - I)^T = W^TW - I$ can be achieved. By this property, the metric in equation (6) can be represented as follows:

$$\begin{aligned} e(W) &= \frac{1}{n} Tr((W^TW - I_n)^T(W^TW - I_n)) \\ &= \frac{1}{n} Tr((W^TW - I_n)^2), \end{aligned} \tag{9}$$

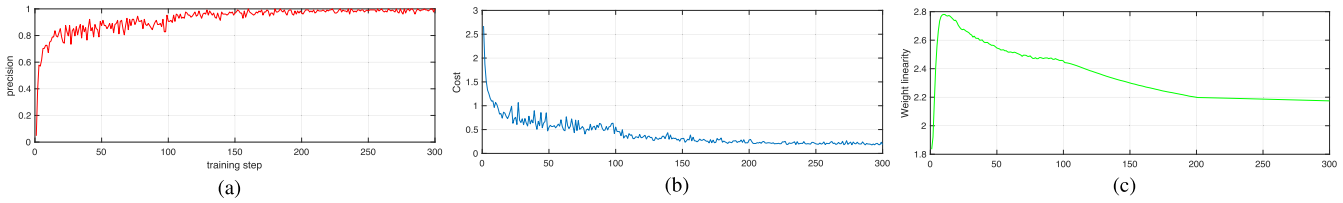

FIGURE 3. (a), (b), and (c) contain the classification accuracies, costs, and the kernel linearity ($\epsilon(W)$) on each training step, respectively. X-axis of each graph denote the training step. The baseline model is ResNet-34.

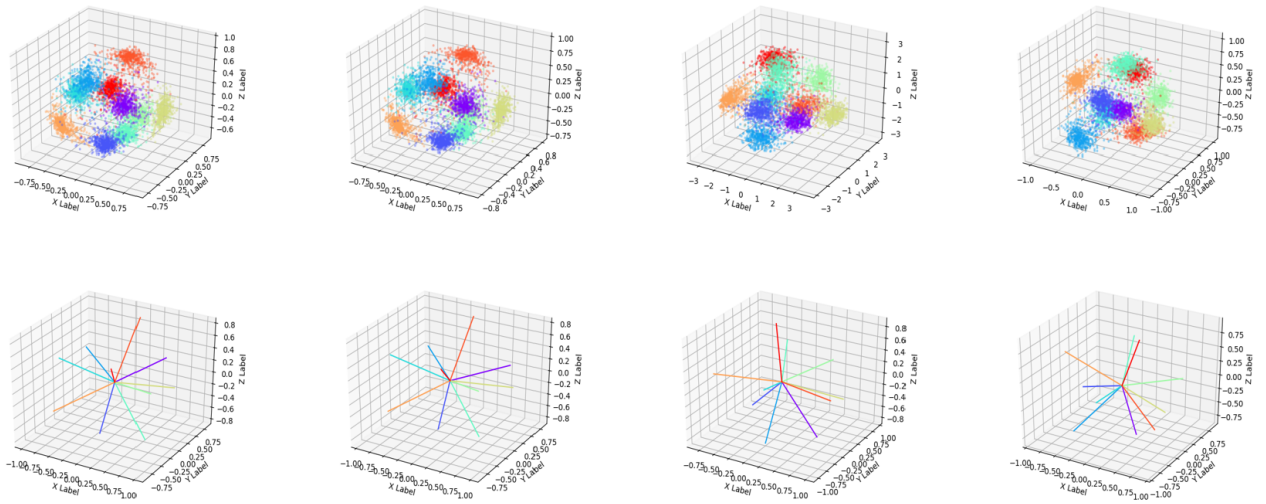

FIGURE 4. The visualization results for latent features and learned weight vectors of models trained by various loss functions. The graphs in the first row show the distribution of latent features in 3D space. The graphs in the second row represent the direction and magnitude of weight vectors. We employ the LeNet structure in this visualization, and we reduce the dimensionality of latent features as three using PCA. The figures under the name of loss functions show the classification accuracies and the results of weight separability evaluation using our metric, respectively. The noticeable thing is that the recognition performance and the results of weight separability are proportion even though the visualization results are difficult to correspond to the recognition performance.

where $Tr(\cdot)$ is the trace operation of an square matrix defined by the sum of the elements on the main diagonal of the square matrix. Intuitively, when the value of $e(W)$ is converged to zero, the column vectors of weight matrix would be linearly independent and separability of the column vector can take maximum. We omit the bias in the fully connected layer because it just complicates our analysis via visualization and nearly does not influence the recognition accuracies [21]. Figure 3 shows the trend of the classification precision, cost function, and the kernel linearity evaluated by equation (9), based on ResNet-34 and Cifar-10 datasets. As shown in figure 3, the kernel linearity is gradually decreased and the classification precision increasing during the training.

Additionally, we conduct simple experiments using MNIST dataset to verify our metric. We train the LeNet using various loss functions including $l2$-norm softmax [23], center loss [22], and large-margin softmax [21], and we carry out the cross check for accuracy and weight separability about each model. Figure 4 illustrates the visualization results of the experiments. As shown in Figure 4, the experimental results show that the more accurate recognition performance can take the larger weight separability evaluated as our metric. One of the interesting observations is that the evaluation results for weight separability using our metric can be reflected the recognition performance even if it is difficult to figure out the superiority of recognition performance using visualization results.

IV. FEED-BACKWARD RECONSTRUCTION

A. MOTIVATION

Considering the commonly used optimization methods such as softmax-cross entropy, we have a latent feature α and corresponding annotation label o. If the latent feature α have to be classified into i^{th} class, the methods are concentrate on to encourage $w_i \cdot \alpha > w_j \cdot \alpha, j = 1, 2, 3, 4, \ldots, n$ $(j \neq i)$, where n is the number of classes, and w_i is i^{th} column vector in a weight matrix W. In this work, we want not only to improve intra-class compactness and inter-class separability but also to boost the separability between the column vectors in weight vector. Current loss functions such as softmax-cross entropy, l_2-distance loss, cosine angular loss, and large-margin softmax, do not consider the weight separability explicitly. Therefore, a new method is required to directly improve the weight separability.

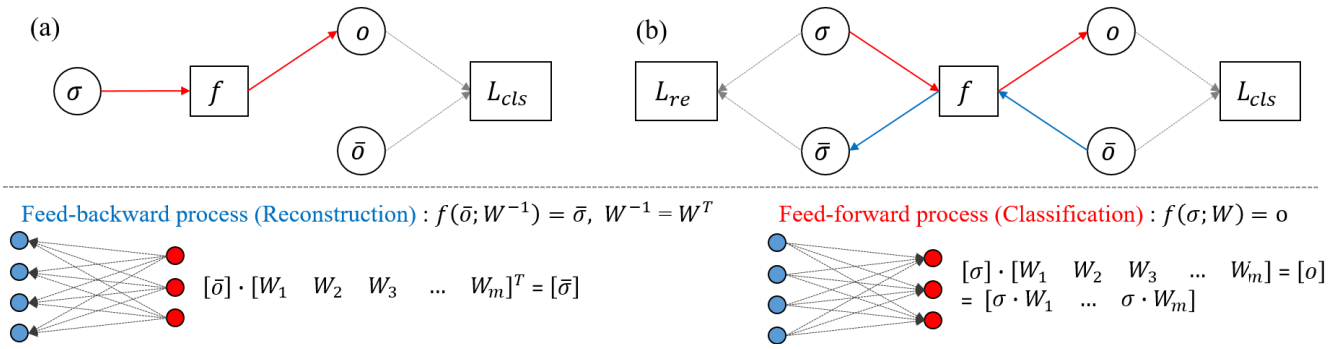

FIGURE 5. (a) The normally trained models contain a simple mapping pipeline for classification f and associated classification loss L_{cls}. (b) The models applied the proposed reconstruction loss contain two mapping pipelines: the classification f, reconstruction f^{-1}, and associated losses L_{cls} and L_{re} for each. o and $\hat{o}$ are the network output and corresponding annotation, respectively. α and $\hat{\alpha}$ are the latent feature and reconstructed latent feature from the given annotation $\hat{o}$, respectively. The red and blue arrows in the first row indicate the classification and reconstruction pipelines, respectively. The red and blue dots represent the activation units of output and previous layers.

B. FEED-BACKWARD RECONSTRUCTION LOSS

Following the notation for the weight separability evaluation in Section III, the weight separability would be maximum when $W^T W - I$ takes a zero matrix. In this case, basically, we assume that $W^T = W^{-1}$. However, using the proposed evaluation metric as an objective function is not suitable to train a model because of a problem for computing gradients as long as we use the back-propagation algorithm [28] to update network parameters. The evaluation metric is composed of the weight matrix of the final layer only; therefore, the gradient of the proposed metric for weight separability ($\frac{\partial \frac{1}{n} Tr((W^T W - I_n)^2)}{\partial w_{ij}}$) will vanish when the gradient for other layers are calculated. Consequently, it is necessary to develop an objective function that is suitable for applying the training procedure of networks.

To address this issue, we propose a feed-backward reconstruction loss that can improve the weight separability directly. The feed-backward reconstruction loss is defined as

$$L_{re}(\hat{o}; \alpha, W) = \sum_{i=1} P(\alpha_i) \log(\frac{P(\alpha_i)}{Q(\hat{o} w_i^T)}), \tag{10}$$

where α, w_i^T, and $\hat{o}$ are a latent feature, the i^{th} transposed column vector of the weight matrix W, and the corresponding label about the latent feature, respectively. P and Q are the distributions for the latent features and reconstruction results. The proposed loss functions mathematically equivalent to the Kullback-Leibler divergence, and literally this loss function defines the difference between the distributions of latent features and reconstruction results. Intuitively, if the proposed loss L_{re} is converged to zero, then it means $P(\alpha_i)$ is equivalent to $Q(\hat{o} w_i^T)$, and it is represented as $P(\alpha) log(\frac{P(\alpha)}{Q(\hat{o} W^T)}) = 0$.

In this situation, W^T can be regarded as W^{-1}, and it also can be regarded as a solution to maximize the weight separability. The reconstruction loss functions using l_1-norm or l_2-norm force to minimize the Euclidean distance even their angular difference is tiny. These approaches cannot be used with various activation functions since there is a probability that the Euclidean distance can be changed by an activation function. Therefore, we instead require some parameter-transformation invariant methods based on the results from computing a difference of probabilistic distributions.

When we apply the proposed loss to train a model, the proposed loss is added to ordinary loss functions L_{cls} such as softmax cross entropy, center loss [22], and large-margin softmax loss [21]. Therefore, the total loss function is defined as follows,

$$L_{total}(\hat{o}, o; \alpha, \theta) = L_{cls}(\hat{o}, o; \theta) + \lambda L_{re}(\hat{o}; \alpha, W), \tag{11}$$

where o and $\hat{o}$ are the output of models and corresponding labels. α is the output of previous layer that connected to the network for recognition tasks, and W is the weight of a final layer. θ is a set of network parameters, including W. λ is hyper-parameter to decide the weight of the proposed reconstruction loss in training task. In our experiments, the value of λ is set to 0.001, and this value is determined by the value with the best performance from several experiments.

C. INTERPRETATION

The model with the feed-backward reconstruction loss contains two mapping process: 1) Determination process $f : \alpha \rightarrow o$ and 2) Reconstruction process $f^{-1} : \hat{o} \rightarrow \hat{\alpha}$, and both processes share the weight parameter W. The determination process f encourages W to translate α into an encoded output o, and the reconstruction process f^{-1} forces W^T to recover $\hat{\alpha}$ from the given label $\hat{o}$. Figure 5 shows the comparison between a normal model and the model applying the feed-backward reconstruction process in a classification task. In optimization via these two processes, each process affects each other in achieving their objectives.

The objective of the determination process is to maximize the accuracy for visual recognition tasks by minimizing geometric or probabilistic difference between the output of a model $\alpha W = o$ and the given annotations $\hat{o}$. On the other hand, the reconstruction process aims to minimize the difference of distributions between the latent feature $P(\alpha)$ and the reconstruction results $Q(\hat{o} W^T)$. The reconstruction process can be optimized when the determination process takes highly accurate performance, and it is able to provide more accurate recognition performance when the weight separability become more advanced. Above cooperation between two

TABLE 1. Error rates (%) on CIFAR-10 and CIFAR-100 datasets. $+L_{re}$ denotes the model is trained with the proposed reconstruction error. + indicates that simple data augmentation is used. $e(W)_{avg}$ is the average result of weight separability evaluation between normally trained results and the results with the simple data augmentation corresponding to C10 and C100 dataset. *nan* represents the information is not provided from original paper. k is the growth rate in DenseNet. + indicates that the data augmentation based on simple image transformation is used. The marked value as red color is a change of performance after applying the proposed reconstruction loss. The bolded value is the best performance in our experiments.

Method	Depth	Params	C10	C10+	$e(W)^{avg}_{C10}$	C100	C100+	$e(W)^{avg}_{C100}$
Network in Network [29]	12	11.4M	13.76	11.2	8.56e-08	35.68	33.04	6.85e-08
Network in Network+L_{re}	12	11.4M	10.03(-3.73)	9.64(-1.56)	6.29e-08(-2.27e-08)	31.22(-4.46)	31.07(-1.97)	6.81e-08(-0.04e-08)
VGG-16 [30]	16	13.4M	10.48	10.26	7.32e-08	37.48	31.27	6.51e-08
VGG-16+L_{re}	16	13.4M	9.17(-1.31)	7.94(-2.32)	6.07e-08(-1.25e-08)	31.55(-5.93)	29.96(-1.31)	6.43e-08(-0.08e-08)
Highway Network [31]	12	11.8M	12.98	9.8	7.32e-08	39.51	33.07	5.12e-08
Highway Network+L_{re}	12	11.8M	9.13(-3.85)	7.72(-2.08)	6.73e-08 (-0.59e-08)	35.64(-3.07)	32.01(-1.06)	4.15e-08(-0.97e-08)
ResNet-34 [3]	36	1.7M	8.64	8.09	6.55e-08	32.18	31.37	4.62e-08
ResNet-34+L_{re}	36	1.7M	6.01(-2.63)	5.94(-2.15)	3.57e-08(-2.98e-08)	30.65(-1.03)	29.48(-1.89)	2.96e-08(-1.66e-08)
DenseNet-40 ($k = 12$) [20]	40	1.0M	9.42	6.17	3.45e-08	29.3	24.58	1.54e-08
DenseNet-40+L_{re} ($k = 12$)	40	1.0M	5.91(-3.51)	5.62(-0.55)	2.16e-08(-1.29e-08)	29.01(-0.29)	20.75(-3.83)	1.42e-08(-0.12e-08)
PyramidNet+ShakeDrop [32]	96	nan	3.1	2.4	3.17e-08	15.7	14.1	1.44e-08
PyramidNet+ShakeDrop+L_{re}	96	nan	**2.94**(-0.16)	**2.13**(-0.27)	2.95e-08(-0.22e-08)	**13.75**(-1.95)	**11.65**(-2.45)	1.29e-08(-0.15e-08)
Res2NeXt-29 [33]	29	36.9M	4.52	3.74	4.12e-08	18.10	17.40	1.68e-08
Res2NeXt-29+L_{re}	29	36.9M	4.31(-0.21)	3.17(-0.57)	3.07e-08(-1.05e-08)	15.42(-2.68)	15.39(-1.91)	1.35e-08(-0.33e-08)

processes is similar to the cycle consistency losses [34], [35]. Consequently, above processes not only can boost the weight separability but also can improve the cyclic consistency via dual minimization schemes for classification task and latent feature reconstruction.

V. EXPERIMENTAL RESULTS

A. IMAGE CLASSIFICATION

We conducted experiments for image classification on the CIFAR-10 and CIFAR-100 datasets [40]. The CIFAR-10 dataset is composed of 50,000 training images and 10,000 test images in 10 classes. CIFAR-100 dataset consists of 100 classes, and each class contain 500 training images and 100 testing images. Our work is concentrated to demonstrate the efficiency of the feed-backward reconstruction loss, and not on encourage state-of-the-art performance. Therefore, our experiment is conducted based on the several baseline models intentionally and focuses on the comparison between the normally trained model and the trained model using the proposed feed-backward reconstruction loss.

The baseline models used in the experiment for image classification, are as follows: Network in Network [29], VGG-16 [30], Highway Network [31], Residual Network (ResNet) [3], Densely Connected Convolutional Neural Network (DenseNet) [20], PyramidNet+ShakeDrop [32], and Res2NeXt-29 [33]. To improve an experimental efficiency, we use the most shallow structure on ResNet and DenseNet, and the ResNet-34 and Densenet-40 structures are selected for our experiments. All networks are trained using stochastic gradient descent (SGD) [41]. We train all networks using 128 batch size for 300 epochs. During training networks, we employ a learning rate decay of 0.0001 and momentum of 0.9. The learning rate is initially set to 0.1, and divided by 10 in 100, 200, and 250 epochs.

The experimental results on CIFAR-10 and CIFAR-100 dataset are shown in Table 1. The PyramidNet+ShakeDrop [32] applying simple data augmentation and the proposed reconstruction loss achieved an error rate of 2.13% on CIFAR-10 dataset and 11.65% on CIAR-100 dataset. These figures are the best results in our experiment for image classification. The evaluation results of weight separability for above records are 2.95e-08 and 1.95e-08, respectively. The entire experimental results show that the trained model considering the feed-backward reconstruction loss outperformed the normally trained models. The most noticeable thing in our experiment is that the models trained for reflecting our loss achieve better performance whether the performance differences are small or large collectively.

B. FACE RECOGNITION

We also conduct additional experiments for face recognition to demonstrate the efficiency of the proposed method for improving weight separability. This experiment is conducted under the *unrestricted with labeled outside data* protocol, so that all models are trained only using CASIA-Webface dataset and are tested using Labeled Faces in the Wild (LFW) dataset [42] and the Youtube Faces (YTF) [43] dataset. CASIA-Webface dataset consists of 494,414 of face images labeled as 10,575 different identities, and the dataset also contains horizontally flipped images for data augmentation. The performance evaluation is carried out on 6000 of face pairs from LFW dataset, and 5000 of video pairs from YTF dataset.

The network model list used in this experiments as follows: DeepFace [23], Facenet [8], DeepID2+ [36], DDRL [37], ArcFace [38], CosFace [39], and the other methods proposed by Wen *et al.* [22] and Liu *et al.* [21]. These methods are initially trained via classification setting and conduct the evaluation using a verification scheme. We add the proposed feed-backward reconstruction loss to calculate the total loss when the models are trained. Table 2 shows the comparison results of the normally trained models and the models with the proposed loss.

The face recognition results usually show that the trained models with the proposed loss achieved better performance than the normally trained models. The highest recognition accuracies in LFW and YTF datasets are achieved by ArcFace [38] trained with the proposed reconstruction loss. The ArcFace trained by the proposed loss achieves 0.19% and 1.96% error rates on LFW and YTF datasets, respectively.

TABLE 2. Error rate (%) and the results of weight separability evaluation using our metric ($e(W)$) on LFW and YTF datasets. $+L_{re}$ denotes the model is trained with the proposed reconstruction error. + and − represent that the increase or decrease on recognition error rate after applying the proposed reconstruction loss. $e(W)_{LFW}$ and $e(W)_{YTF}$ indicate the evaluation result of the proposed metric for weight separability for each dataset. For a fair comparison, we implemented all models and loss functions directly and trained only using CASIA-Webface dataset. The bolded values represent the lowest error rate on LFW and YTF datasets.

Method	Data	LFW	$e(W)_{LFW}$	YTF	$e(W)_{YTF}$
DeepFace [23]	WebFace	3.65	11.67e-08	13.77	14.01e-08
DeepFace+L_{re}	WebFace	2.99(-0.66)	8.43e-08(-3.24e-08)	10.24(-3.53)	11.54e-08(-2.47e-08)
FaceNet [8]	WebFace	2.82	9.78e-08	6.21	9.76e-08
FaceNet+L_{re}	WebFace	2.60(-0.22)	8.96e-08(-0.82e-08)	7.87(-0.34)	9.13e-08(-0.63e-08)
DeepID [36]	WebFace	3.08	10.03e-08	7.35	10.83e-08
DeepID+L_{re}	WebFace	1.66(-1.42)	8.01e-08(-2.02e-08)	4.53(-2.82)	8.76e-08(-2.07e-08)
DDRL [37]	WebFace	0.99	6.81e-08	5.98	10.91e-08
DDRL+L_{re}	WebFace	0.87(-0.12)	6.30e-08(-0.51e-08)	7.15(+1.17)	7.38e-08(-3.54e-08)
L-Softmax [21]	WebFace	1.48	7.74e-08	6.21	9.65e-08
L-Softmax+L_{re}	WebFace	0.94(-0.54)	6.84e-08(-0.90e-08)	5.57(-0.64)	8.68e-08(-0.97e-08)
Softmax+Center Loss [22]	WebFace	1.22	10.24e-08	6.08	13.42e-08
Softmax+Center Loss+L_{re}	WebFace	1.47(+0.25)	9.53e-08(-0.71e-08)	6.03(-0.05)	11.97e-08(-1.45e-08)
ArcFace [38]	WebFace	0.22	8.21e-08	2.13	11.52e-08
ArcFace+L_{re}	WebFace	**0.19(-0.03)**	7.42e-08(-0.79e-08)	**1.96(-0.17)**	9.91e-08(-1.61e-08)
CosFace [39]	WebFace	0.41	7.41e-08	2.9	11.51e-08
CosFace+L_{re}	WebFace	0.25 (-0.16)	5.53e-08(-1.88e-08)	2.1(-0.8)	7.36e-08(-2.55e-08)

The evaluation results of the weight separability for these experiments are 7.42e-08 and 9.91e-08. However, in experiments using the DDRL [37] and the center loss [22], the accuracies are degraded even though the weight separabilities of these models are decreased. In the experiment using YTF dataset and DDRL, the 3.54e-08 of weight separability was reduced, but the DDRL applying the proposed reconstruction loss have achieved 7.15%, which is less accurate than the DDRL trained without the proposed loss (5.98%). Additionally, with the experiment using the center loss, the trained model with the proposed reconstruction loss achieves lower accuracies than the original model. These performance degradations may be interpreted as a collapse of gradients between loss terms, since the objective functions of the two studies [22], [37] are quite complicated compared with others. DDRL takes joint optimization process for face identification and verification in optimizing their model.

The overall experimental results on face recognition tasks show similar trend on the experimental results of image classification. Even though the experimental results in our experiment are slightly lower than the listed accuracies in their studies, these figures are comparable to the reported performance in the studies [8], [11], [36] and almost similar to the state-of-the-art methods only trained by CASIA-Webface dataset.

C. ANALYSIS

The experimental results show clear advantages over the current deep neural network models and a lot of compared baselines. Our interpretation of these performance improvements follows. In first, as we have mentioned in Section II and Section III, the weight separability can influence recognition performance in a model based on the neural network. We have tried to improve the weight separability via the feed-backward reconstruction loss that can encourage the linear independence between the column vectors in a weight matrix. In the learning procedure, the proposed

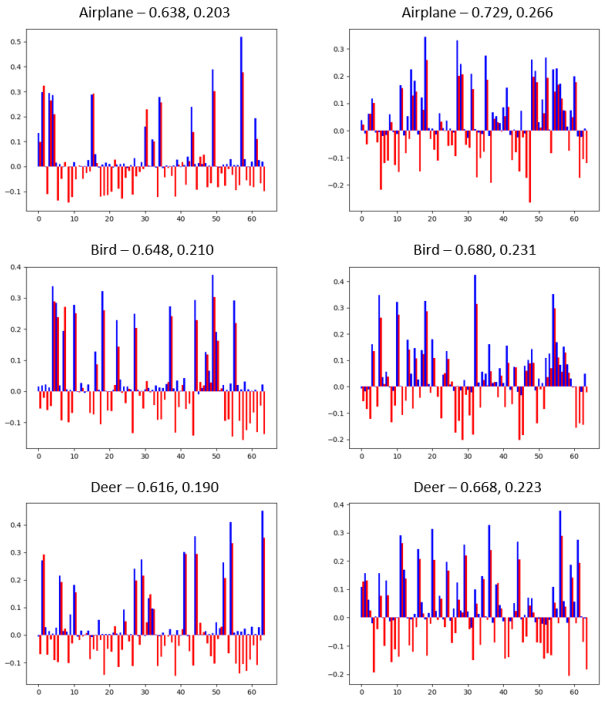

FIGURE 6. Pattern comparison of neuron activation and the corresponding weight vector on 'Airplane', 'Bird', and 'Deer' classes in CIFAR-10 dataset. X-axis shows the an index of each neuron, and Y-axis represents an activation output. The graphs in right-side are the pattern comparison for normally trained ResNet, and the graphs in left-side are the comparison on the ResNet applying the proposed reconstruction loss. The blue bar indicates the expectation of neuron activation, and the red bar represents the corresponding weight vector. The values beside of class name are vector similarities based on Euclidean distance and cosine similarities between the expectation value of neuron activation and the corresponding weight vector.

reconstruction loss plays an important role to improve the weight separability explicitly. The error rates and weight separability evaluation results in Table 1 show that the classification performance is probably proportional to the weight separability evaluation results. Experimental results not only

for image classification results but also for face recognition show similar circumstance.

In second, the feed-backward reconstruction can improve not only the weight separability but also intra-class compactness. Figure 6 represents the comparison of neuron activation pattern and the values of a corresponding column vector in a weight matrix in our classification experiment using ResNet. The figures on the top of each bar graph indicate that the Euclidean distance and cosine similarity between the neural activation and the corresponding column vector in a weight matrix, respectively.

These figures are regarded as that the similarities between neuron activation and the corresponding vectors. A common point of these figures is that the figures applying the proposed reconstruction loss are smaller than the normal ones. In figure 6, the Euclidean distance and cosine similarity of the model applying our reconstruction loss for 'Deer' class are 0.616 and 0.190, respectively. On the contrary, the corresponding Euclidean distance and cosine similarity of the normal model are 0.666 and 0.223, and these figures are bigger than the model applying the proposed reconstruction loss. In addition to the experimental results for 'Deer' class, other experimental results for 'Airplane' and 'Bird' classes show the same phenomenon. These results show that the proposed reconstruction loss can help to learn more discriminative representations.

VI. CONCLUSION

In this paper, we have explored the kernels of neural networks and have studied a way to improve the separability of networks' kernels, which can improve performances of various visual recognition tasks. We have defined the metric for weight separability evaluation and have proposed the feed-backward reconstruction loss to explicitly improve the weight separability. The evaluation metric have represented the linear independence property of column vectors in a weight matrix. With the proposed feed-backward reconstruction loss, the separability of column vectors in the weight matrix have been improved. We have demonstrated the efficiencies of the evaluation metric and the proposed reconstruction loss based on the experiments for image classification and face recognition. The experimental results show that the proposed feed-backward process and the loss function can contribute to performance improvement in recognition tasks.

However, it is worth mentioning a limitation of the proposed method. Although models applying the proposed reconstruction loss have achieved outstanding performances for visual recognition tasks, it forces to increase training time in learning those models compared with the models without the proposed reconstruction loss. The proposed reconstruction loss would not increase the number of parameters, thus it would not increase the complexity of models applying the proposed loss. The proposed approach, however, forces to conduct the feed-backward process, which is sort of extra task, to compute the reconstruction loss. As a result, applying the proposed approach increases time consumption for optimizing network models compared with normal ones. In future works, the primary goal would be to solve the aforementioned limitation. Also, we hope to explore the kernel space and latent features to improve various visual recognition tasks based on neural networks.

REFERENCES

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[3] 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.

[4] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 681–687.

[5] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 656–664.

[6] M. Zhang, M. Diao, and L. Guo, "Convolutional neural networks for automatic cognitive radio waveform recognition," *IEEE Access*, vol. 5, pp. 11074–11082, 2017.

[7] J. Yu, S. Park, S. Lee, and M. Jeon, "Driver drowsiness detection using condition-adaptive representation learning framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 11, pp. 4206–4218, Nov. 2019.

[8] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[9] 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.

[10] Y. Sun, D. Liang, X. Wang, and X. Tang, "DeepID3: Face recognition with very deep neural networks," 2015, *arXiv:1502.00873*. [Online]. Available: http://arxiv.org/abs/1502.00873

[11] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jul. 2017, p. 1.

[12] W. Li, R. Zhao, T. Xiao, and X. Wang, "DeepReID: Deep filter pairing neural network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 152–159.

[13] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," *Pattern Recognit.*, vol. 48, no. 10, pp. 2993–3003, Oct. 2015.

[14] S. Zhou, P. Luo, D. K. Jain, X. Lan, and Y. Zhang, "Double-domain imaging and adaption for person re-identification," *IEEE Access*, vol. 7, pp. 103336–103345, 2019.

[15] N. Perwaiz, M. M. Fraz, and M. Shahzad, "Person re-identification using hybrid representation reinforced by metric learning," *IEEE Access*, vol. 6, pp. 77334–77349, 2018.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893.

[17] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, p. 1150.

[18] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 2126–2136.

[19] L. Jia, Q. Zhang, Y. Shang, Y. Wang, Y. Liu, N. Wang, Z. Gui, and G. Yang, "Denoising for low-dose CT image by discriminative weighted nuclear norm minimization," *IEEE Access*, vol. 6, pp. 46179–46193, 2018.

[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, vol. 1, Jul. 2017, p. 3.

[21] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. ICML*, 2016, pp. 507–516.

[22] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.* Columbus, OH, USA: Springer, 2016, pp. 499–515.

[23] Y. Taigman, M. Yang, M. 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.
[24] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," 2018, *arXiv:1801.07698*. [Online]. Available: http://arxiv.org/abs/1801.07698
[25] P. Rodríguez, J. Gonzàlez, G. Cucurull, J. M. Gonfaus, and X. Roca, "Regularizing CNNs with locally constrained decorrelations," 2016, *arXiv:1611.01967*. [Online]. Available: http://arxiv.org/abs/1611.01967
[26] M. Zhao, Z. Jia, and D. Gong, "Improved two-dimensional quaternion principal component analysis," *IEEE Access*, vol. 7, pp. 79409–79417, 2019.
[27] X. Chen, Z. Jia, Y. Cai, and M. Zhao, "Relaxed 2-D principal component analysis by l p norm for face recognition," in *Proc. Int. Conf. Intell. Comput.* Nanchang, China: Springer, 2019, pp. 197–207.
[28] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural Networks for Perception*. Amsterdam, The Netherlands: Elsevier, 1992, pp. 65–93.
[29] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*. [Online]. Available: http://arxiv.org/abs/1312.4400
[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[31] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2377–2385.
[32] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation policies from data," 2018, *arXiv:1805.09501*. [Online]. Available: http://arxiv.org/abs/1805.09501
[33] S. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. H. S. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 30, 2019, doi: 10.1109/TPAMI.2019.2938758.
[34] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2017, *arXiv:1703.10593*. [Online]. Available: http://arxiv.org/abs/1703.10593
[35] J. Yu, K. C. Yow, and M. Jeon, "Joint representation learning of appearance and motion for abnormal event detection," *Mach. Vis. Appl.*, vol. 29, no. 7, pp. 1157–1170, Oct. 2018.
[36] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2892–2900.
[37] J. Yu, D. Ko, H. Moon, and M. Jeon, "Deep discriminative representation learning for face verification and person re-identification on unconstrained condition," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 1658–1662.
[38] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4690–4699.
[39] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.
[40] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 2009TR, 2009.
[41] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*. Paris, France: Springer, 2010, pp. 177–186.
[42] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Dept. Comput. Sci., Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 07-49, 2007.
[43] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. CVPR*, Jun. 2011, pp. 529–534.

JONGMIN YU received the Ph.D. degree from the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea. He was a Visiting Researcher with the School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Perth, WA, Australia. He is currently a Research Associative with the Institute for IT Convergence, Korea Advanced Institute of Science and Technology (KAIST). His research interests include artificial intelligence, machine learning, pattern recognition, and mathematical understanding of these.

HYEONTAEK OH (Member, IEEE) received the B.S. degree *(summa cum laude)* in computer science, the M.S. degree, and the Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), in 2012, 2014, and 2020, respectively. He is currently a Team Leader with Institute for IT Convergence, KAIST. He has actively participated in nationally-funded research projects for ICT environment. Since 2015, he has been contributing the International Telecommunication Union Telecommunication Standardization Sector Study Group 13/20 as contributors and editors. His research interests include ICT environments, personal data ecosystem, the Internet of Things (IoT), and web technologies.