
Article

A Methodology for Utilizing Vector Space to Improve the Performance of a Dog Face Identification Model

Bohan Yoon, Hyeonji So and Jongtae Rhee *

Department of Industrial and Systems Engineering, Dongguk University, Seoul 04620, Korea;
yi92run@dongguk.edu (B.Y.); hyeonji.so@dgu.ac.kr (H.S.)

* Correspondence: jtrhee@dongguk.edu; Tel.: +82-2-2264-8518

Abstract: Recent improvements in the performance of the human face recognition model have led to the development of relevant products and services. However, research in the similar field of animal face identification has remained relatively limited due to the greater diversity and complexity in shape and the lack of relevant data for animal faces such as dogs. In the face identification model using triplet loss, the length of the embedding vector is normalized by adding an L2-normalization (L2-norm) layer for using cosine-similarity-based learning. As a result, object identification depends only on the angle, and the distribution of the embedding vector is limited to the surface of a sphere with a radius of 1. This study proposes training the model from which the L2-norm layer is removed by using the triplet loss to utilize a wide vector space beyond the surface of a sphere with a radius of 1, for which a novel loss function and its two-stage learning method. The proposed method classifies the embedding vector within a space rather than on the surface, and the model's performance is also increased. The accuracy, one-shot identification performance, and distribution of the embedding vectors are compared between the existing learning method and the proposed learning method for verification. The verification was conducted using an open-set. The resulting accuracy of 97.33% for the proposed learning method is approximately 4% greater than that of the existing learning method.

Keywords: dog face identification; embedding vector; loss function; triplet loss

Citation: Yoon, B.; So, H.; Rhee, J. A Methodology for Utilizing Vector Space to Improve the Performance of a Dog Face Identification Model.

Appl. Sci. **2021**, *11*, 2074.

<https://doi.org/10.3390/app11052074>

Academic Editor: Wonjoon Kim

Received: 31 December 2020

Accepted: 23 February 2021

Published: 26 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recent years have seen an increase in the number of companion animals and also of abandoned or lost companion animals, leading to the proposal of several methods such as microchip implantation, nose recognition, and iris recognition for identification of companion animals [1,2]. However, the application of these methods is difficult owing to the perception that they may be harmful and might require specialized photographic equipment and techniques.

Both machine and deep learning techniques have been able to provide sufficient reliability for human face recognition [3], and many relevant products/services are being provided. The human face recognition model is trained based on metric learning. The metric learning [4,5] calculates the distance representing the similarity/difference of the object to be compared rather than the classification for a fixed class. For the classification of new classes that are not trained, models based on metric learning output embedding vectors, not classes [6–9]. In addition, the face recognition model based on metric learning adds an L2-norm layer to the end of the model for training. L2-normalization (L2-norm) constrain this embedding to live on the d-dimensional hypersphere [6]. These limitations allow the model to train to generate appropriate embedding vectors for classification. In various attempts, this approach has been applied to dog face identification, but at lower performance than that achieved for the human face identification due to the low normality

of dog faces and lack of related data. In order to overcome these limitations and improve the performance of dog face identification, this paper proposes a novel loss function and a two-stage learning method to utilize a wide vector space. The face identification model using the existing triplet loss was trained with cosine similarity by normalizing the embedding vector with the L2-norm layer, which facilitates training the face identification model, but limits the distribution of the embedding vector to the surface of a sphere with radius 1. This limitation makes it possible to learn a metric-learning-based model, but it is also limited that embedding vectors are used in an infinite vector space. If the model can be trained to generate an appropriate embedding vector in an infinite vector space, a much wider vector space can be utilized than the existing methodology. In addition, if the model is trained so that the embedding vector is aggregated while utilizing a wide vector space, the performance of the model based on metric learning can be improved.

This study aims to improve the model's performance by classifying the distribution of embedding vectors in a wide vector space beyond the surface of the sphere. Therefore, a novel loss function and a two-stage learning method are proposed to train the model with the L2-norm layer removed. In the previous work, the L2-norm layer uniformly transforms the length of the embedding vector into 1. These limitations allow the model to train based on metric learning. However, our newly proposed loss function trained the model to generate an embedding vector in a specific area without an L2-norm layer. This method trained the model to generate an embedding vector in the specific area, unlike the previous learning methods in which the range of the length of the embedding vector is infinite. In addition, we proposed the new learning method consisted of two stages. In stage 1, the proposed loss function adjusts the distribution of the embedding vector. In addition, the L2-norm layer and the existing loss function for metric learning were used together in stage 1. In other words, the model was trained with the classification by angle and the adjustment of the embedding vector distribution at the same time. As a result of stage 1, it is possible to learn the model using the triplet loss for metric learning without the L2-norm layer in stage 2. Training of the model with the L2-norm layer removed makes it possible to generate an appropriate embedding vector with the embedding vectors beyond the surface of the sphere and distributed in an infinite vector space. To verify the proposed methodology, the same architecture was trained with the existing learning method and the proposed learning method, and then the accuracy, one-shot identification performance, and embedding vector distribution of each model were compared. The evaluation was performed with images of dogs that have not been trained. The change in the distribution of the embedding vector according to the learning method was visualized as a 2-dimensional image in a toy experiment. In addition, to compare the distribution of multidimensional embedding vectors, the distance and the length of the embedding vector were calculated for each class.

The purpose of this study is to improve the performance of the dog face identification model, which is biometrics for dogs. In this paper, we proposed the learning method of the dog face identification model with improved performance compared to the previous training method for the return of lost dogs. We proposed the novel loss function and a two-stage learning method to improve the performance of the dog face identification model. As a result, we demonstrated that the model trained with the proposed learning method performs dog face identification with embedding vectors distributed in the vector space. In addition, it was demonstrated that the dog face identification performance of the model trained with the proposed learning method was improved compared to the model trained with the existing learning method. The proposed loss function and learning method can be used with various metric learning methods, and then the performance is improved. In addition, the study results proved that metric learning is possible to utilize infinite vector space beyond the surface of a sphere. The study results are expected to improve the performance of the dog face identification model and to help in finding lost dogs.

Studies on human and animal face identification are reviewed in Section 2. The novel loss function and learning method for utilizing a vector space are described in Section 3. The models are implemented based on the existing methodology and the proposed methodology in Section 4. The accuracy, one-shot identification performance, and embedding vector distribution are evaluated to compare the performance of the proposed methodology with that of the existing methodology in Section 5.

2. Related Studies

In the fields of deep learning and machine learning, studies for face identification were mainly performed on the human face. Therefore, the newly proposed model and loss function were evaluated with the human face dataset. Recently, as the number of companion animals increases, the number of abandoned or lost animals also has increased. To solve this problem, studies were conducted in which the deep learning models for human biometrics were applied to animals. In this paper, the previous studies on human face identification are described in Section 2.1. The previous studies on animal biometrics are described in Section 2.2. In particular, the previous studies related to dog face identification are described in detail to improve the performance of a dog face identification model.

2.1. Human Face Identification

Deep learning and machine learning models can be trained for human face recognition using large-scale datasets of human faces, leading to high-performance computing resources that have improved [10,11]. Metric learning has been widely used to train deep learning models for face identification [4,5,12,13]. The loss functions used in deep metric learning for the human face identification were contrastive loss [14] and triplet loss [6], which impose a Euclidean margin on the feature. One such model, deep face [15], used explicit 3D face modeling to revisit the alignment and expression stages in the pipeline of face recognition that consisted of four stages. Deep Face achieved accuracy levels of 96.75% and 92.4% for the labeled faces in the wild (LFW) and the YouTube faces (YTF) datasets, respectively. In addition, FaceNet [6], which directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity, was presented. In FaceNet, end-to-end training has shown that simplifying the setup and directly optimizing the loss associated with the task at hand improves performance. FaceNet achieved 99.63%, and 95.12% accuracy levels for the LFW and YouTube Faces DB datasets, respectively. Deep learning models for human face recognition such as SphereFace [7], CosFace [8], Ring loss [16] and ArcFace [9] with improved performances have also been proposed with new loss functions. In SphereFace, angular softmax, which can be viewed as imposing discriminative constraints on a hypersphere manifold, was proposed [7]. CosFace reformulated the softmax loss as a cosine loss by L2 normalizing both features and weight vectors to remove radial variations [8]. ArcFace proposed additive angular margin loss to simultaneously improve intraclass compactness and interclass discrepancy and obviously enforce a more evident gap between the nearest classes [9]. ArcFace achieved accuracy levels of 99.83% and 98.02% for the LFW and YTF datasets, respectively. In addition, global-local GCN [17], which removes noise from the human face dataset, and GroupFace [18], which learns latent groups, were proposed to improve the performance of human face recognition. Many studies related to deep learning have been conducted in various fields to improve the performance of human face identification.

2.2. Deep Learning on Animal Biometrics

Deep learning model research on animal biometrics has been mainly performed on cattle [19–22], horses [23], pigs [24] and endangered animals [25,26]. In cattle, studies on biometrics using muzzle points have been conducted [20–22]. In the case of horses, pigs

and endangered animals, studies on biometrics-based face recognition have been conducted [23–26]. However, dogs have mainly been studied for breed classification [27–30]. In [31], animal biometric identification was classified into four categories: muzzle point, iris pattern, retinal vascular, and face images. Although muzzle point identification is a reliable method, appropriate scanners are required for a shape extraction like human fingerprint identification. The same problem is further highlighted in iris pattern and retinal vascular identification. By contrast, dog faces can be taken with any camera. In addition, high-resolution dog faces can be easily found on the Internet. In the field of face identification, many existing studies have been conducted on the human face. However, rapid overfitting is found in the human face identification models for datasets with a low-level of normality, such as dog faces [31]. To solve this problem, studies on dog face identification have recently been conducted in this regard [1,31–33]. In [32], the authors used a support vector machine to classify the features of the Flickr dog dataset of 42 dog faces consisting of husky and pug images extracted using a convolutional neural network (CNN). In addition, the authors proposed shallow CNN and deep CNN models. The deep CNN model was built using the Overfeat [33] model. As a result, the authors achieved an accuracy of 67.6% on the Flickr-dog dataset. In [34], a dog face detection model was built by cropping dog faces from the Columbia dogs dataset and learning Faster RCNN [35]. Moreover, then, the dog face detection model cropped the dog faces in the Columbia dogs dataset and Stanford dogs dataset. Using the collected dog face images, breed classification was trained on a pre-trained GoogleNet [36]. In addition, the Flickr-dog dataset was trained only at the end of the model using a 10 k-fold cross-validation. In [34], as a result, the model achieved an accuracy of 83.94% on the Flickr-dog dataset. In [31], a dog face dataset was built by collecting and preprocessing a dog face dataset from the Internet, and a ResNet-like model suitable for the size of the collected dataset was built. From the collected dataset, 485 dogs were extracted by selecting dogs with more than 5 photos that are easy to learn. The ResNet-like model is mainly inspired by ResNet [37] because residual layers prevent the vanishing gradient problem. The ResNet-like model was trained and evaluated using the extracted dataset. As a result, an accuracy of 92% was achieved for the testing set.

3. Methodology

In previous studies, (1) an L2-norm layer was added to the end of the model, (2) the embedding vector was normalized, and (3) cosine similarity-based learning was conducted to train the face recognition model with a triplet loss, as shown in Figure 1. In this study, the model with the L2-norm layer removed was trained with a triplet loss to locate the embedding vector within a wide space. However, when the L2-norm layer is removed, the triplet loss is not conducted for training. This paper proposes a two-stage learning method with a vector length loss, which is a novel loss function for the length of the embedding vector, to train a model with triplet loss without an L2-norm layer. The proposed loss function adjusts the length of the embedding vector and places it within a certain area in stage 1. As a result, the model can be trained with a triplet loss without an L2-norm layer in stage 2.

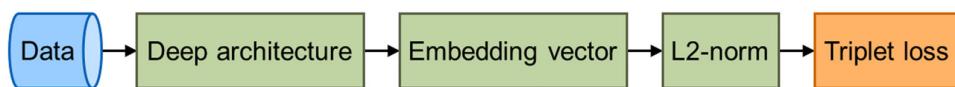


Figure 1. Existing triplet loss training model.

In this paper, a common architecture used to compare learning methodologies is described in Section 3.1. The existing loss functions for metric learning are described in Section 3.2. The vector length loss, a novel loss function for utilizing the infinite vector space, is described in Section 3.3. A new learning method for a model that generates an appropriate embedding vector in a vector space beyond the plane of the sphere using the

triplet loss and the newly proposed vector length loss consists of two-stage and is described in Section 3.4.

3.1. Architecture

Rapid overfitting is found in deep learning models such as FaceNet [6], CosFace [8], and ArcFace [9] for datasets with a low-level of normality, such as dog faces [31]. For this reason, in this paper, the ResNet-like model proposed in [31] was used as architecture for the implementation. The ResNet-like model is designed to prevent overfitting for training the dog face dataset. The ResNet-like model prevents the vanishing gradient problem using residual layers. In addition, the ResNet-like model contains a dropout layer to prevent overfitting. As a result, the ResNet-like model achieved the best performance in the field of dog face identification. Therefore, in this paper, the ResNet-like model is used as the architecture. In [31], the size of an image for learning is $(104 \times 104 \times 3)$, and the ResNet-like model takes the size as an input shape. The size of the original image is $(224 \times 224 \times 3)$. In this paper, an image of size $(224 \times 224 \times 3)$ is used to prevent loss of information in the process of reducing the image. Therefore, the input shape of the architecture is changed to $(224 \times 224 \times 3)$ and used. The ResNet-like model takes outputs an embedding vector of size 32 and used the same. Figure 2 shows the architecture used in this paper.

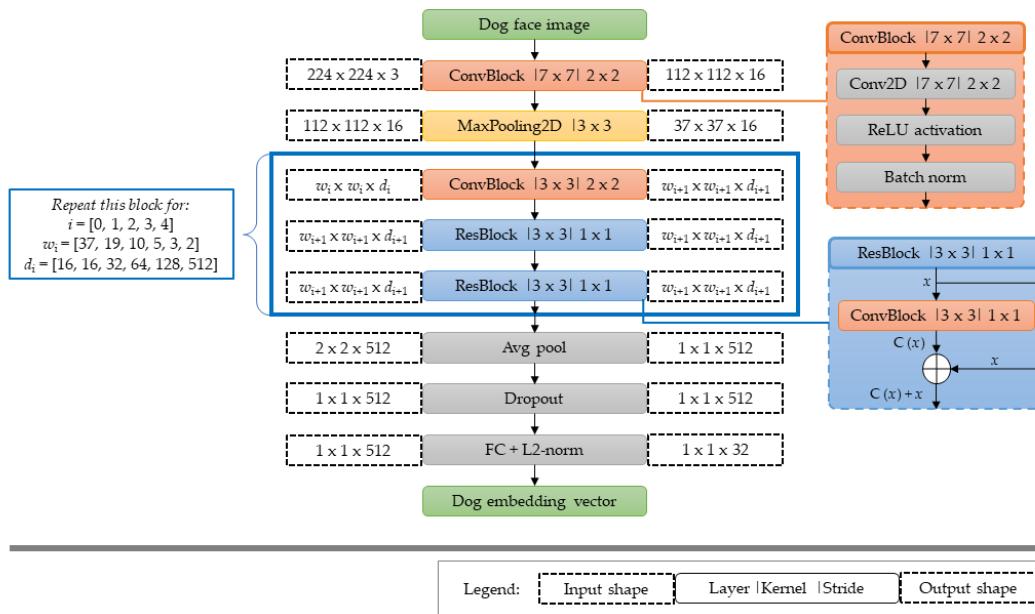


Figure 2. Architecture definition. The architecture takes as input a dog face image of size $(224 \times 224 \times 3)$ and outputs a 32-dimensional embedding vector for the input image. The repeated block is sequentially repeated 5 times. Descriptions of ConvBlock and ResBlock are shown on the right side of the figure.

3.2. The Existing Loss Functions

The loss function used in the field of facial identification based on deep neural networks has improved significantly over the past few years. In particular, metric learning [4,5], has been proposed, and a representative loss function includes a triplet loss proposed in FaceNet [6]:

$$\|f(x_a) - f(x_p)\|^2 + \alpha < \|f(x_a) - f(x_n)\|^2 \quad (1)$$

Training a deep neural network with triplet loss requires a triple dataset consisting of an anchor image (x_a), positive image (x_p), and negative image (x_n). The anchor image is an image of a randomly chosen class inside the dataset. The positive image is another

image of the same class as the anchor. The negative image is an image of class other than the anchor. The purpose of triplet loss is to ensure that the Euclidean distance between the anchor embedding vector $f(x_a)$ and the positive embedding vector $f(x_p)$ is lower than the Euclidean distance between the anchor embedding vector $f(x_a)$ and the negative embedding vector $f(x_n)$. In equations (1) and (2), f denotes a deep learning model. In addition, f generates an embedding vector for an image x and denotes it as $f(x)$. An appropriate constant (α) as a margin is added to increase the distance from other objects. The training of the model aims to minimize the following Equation (2), defined as the triplet loss in [6].

$$\text{Max} (||f(x_a) - f(x_p)||^2 - ||f(x_a) - f(x_n)||^2 + \alpha, 0) \quad (2)$$

Equation (2) shows that loss occurs when the sum of the margin and the Euclidean distance between the anchor embedding vector and the positive embedding vector is greater than the Euclidean distance between the anchor embedding vector and the negative embedding vector. Moreover, it is necessary to configure a dataset that is against the triplet condition for efficient learning. To this end, a hard triplet loss has been proposed. However, it may not be trained properly owing to the existence of several outliers if the entire training set is composed of a hard triplet dataset. To avoid this problem, an offline dataset reconfiguration method for each specific epoch and an online dataset configuration method for reconfiguring the dataset within a mini-batch have been proposed. Triplet loss is a Euclidean loss and is mapped to locate each class in an infinite vector space. However, the triplet loss is not conducted for training in an infinite vector space. FaceNet [6] adds an L2-norm layer to the end of the model to train the face recognition model with the triplet loss. Triplet loss was also used to train a model for dog face identification [31]. The L2-norm layer normalizes the length of the embedding vector to 1. The triplet loss here is the cosine similarity.

The loss function related to the recently proposed face recognition model is ArcFace loss [9]. ArcFace loss compares the angles between embedding vectors, unlike triplet loss compares Euclidean distances. Therefore, in ArcFace [9], the triplet loss is transformed as shown in Equation (3):

$$\arccos(f(x_a), f(x_p)) + \alpha < \arccos(f(x_a), f(x_n)) \quad (3)$$

In Equation (3), \arccos is the angle of the two input embedding vectors. In addition, the triplet loss-based ArcFace defined from Equation (3) is calculated as Equation (4):

$$\text{Max} (\arccos(f(x_a), f(x_p)) - \arccos(f(x_a), f(x_n)) + \alpha, 0) \quad (4)$$

Equation (4) shows that loss occurs when the sum of the margin and the angles between the anchor embedding vector and the positive embedding vector is greater than the angles between the anchor embedding vector and the negative embedding vector.

3.3. Vector Length Loss

This paper proposes a vector length loss, a novel loss function that adjusts the length of the embedding vector to remove the L2-norm layer and use the triplet loss. This loss function is used with the triplet loss, including the L2-norm layer during stage 1, and the loss is calculated during the previous L2-norm layer. The loss for the length of the unnormalized embedding vector is calculated with the proposed loss function and minimized. The purpose of the vector length loss function is to adjust the length of the embedding vector by minimizing the length difference between the anchor and positive image in the triplet dataset. The vector length loss function only uses the anchor and positive image in the triplet dataset, not the negative image. In other words, the vector length loss aims to reduce the difference in the length of embedding vectors between the same class. This allows the model to be trained in stage 2 using triplet loss without the L2-norm layer. Since the vector length loss adjusts the length of the embedding vector, it can

be used simultaneously with the triplet loss that adjusts the direction of the embedding vectors.

$$g(x, z) = xz + \frac{1}{x(x+z)} \quad (5)$$

Equation (5) calculates the loss for the length of the embedding vector of the anchor and positive images in the dataset, which is composed of a triplet. In Equation (5), x is the smaller value of the two embedding vector lengths, and z is the absolute value of the difference in length of the two normalized embedding vectors (Appendix B). The x is the length of the embedding vector. Therefore, x is greater than 0. Consequently, Equation (5) avoids division by zero:

$$\frac{\delta^2 g(x, z)}{\delta x^2} = \frac{2z^2 + 6xz + 6x^2}{x^3z^3 + 3x^4z^2 + 3x^5z + x^6} \quad (6)$$

$$\frac{\delta^2 g(x, z)}{\delta z^2} = \frac{2}{xz^3 + 3x^2z^2 + 3x^3z + x^4} \quad (7)$$

Equations (6) and (7) are the second derivatives of x and z in Equation (5), respectively, which are always positive if x and z are greater than zero. In other words, vector length loss has one local point because the smaller embedding vector length of the anchor and positive images is always greater than zero, and the absolute value of the difference in length of two embedding vectors is greater than or equal to zero. The minimization of the loss calculated by this loss function converges z into 0 and diverges x . However, Equation (5) is too sensitive to the value of z , and the minimum loss cannot reach zero:

$$h(x, z) = xz + \frac{1}{x(x+z)} - \beta \quad (8)$$

The specific constant β is subtracted from Equation (5) in Equation (8), and the minimum value of the loss calculated by the equation can be adjusted to below zero, as shown in Figure 3a. In addition, the distribution of x and z can be appropriately adjusted by changing the β . As the β increases, the z , which can achieve the loss of zero, also increases. In addition, as the β increases, the range of x —which can achieve the loss of zero—increases. The purpose of this loss function is to minimize loss. In other words, the distribution of x and z means the distribution that makes the value of this loss function zero at a specific β . As a result, it is possible to adjust the degree of distribution of x and z without calculating the loss. However, Equation (8) is still sensitive to the z value, and the allowable range for x is extremely small, which interferes with the triplet loss training because it is extremely convex.

$$H(x, z) = z \ln(x+1) + \frac{1}{x(x+z)} - \beta \quad (9)$$

A natural logarithm was added to smoothen the equation and solve the problem of Equation (6) being too convex in Equation (9). The modified Equation (9) was less sensitive to the value of z , and the allowable range of x was wider. Adding 1 to the natural logarithmic value of x prevented rapid shortening of the length of the embedding vector.

$$H(x, z) - H(x, z) = z(x - \ln(x+1)) \quad (10)$$

Equation (10) is Equation (8) minus Equation (9), which always has a positive value if x and z are greater than zero. Therefore, Equation (9) consistently has a lower value than Equation (8).

$$\frac{\delta h(x, z)}{\delta x} - \frac{\delta H(x, z)}{\delta x} = \frac{xz}{x+1} \quad (11)$$

$$\frac{\delta h(x, z)}{\delta z} - \frac{\delta H(x, z)}{\delta z} = x - \ln(x+1) \quad (12)$$

Equations (11) and (12) are the first derivative of Equation (6) minus the first derivative of Equation (9) and are always positive if x and z are greater than zero. Therefore, Equation (8) is less convex than Equation (5), as shown in Figure 3b. In this paper, an experiment was conducted using $\beta = 0.3$ in Equation (9).

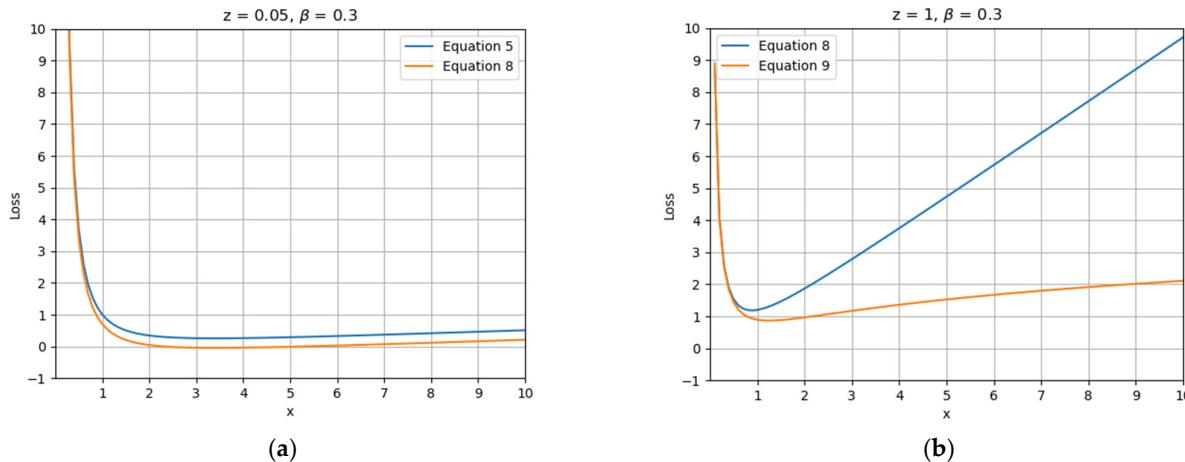


Figure 3. Comparison of Equations (5), (8), and (9) for z , which is the absolute value of the difference in length of the two normalized embedding vectors, and β , which is the specific constant. (a) Equations (5) and (8) for $z = 0.05$, $\beta = 0.3$; (b) Equations (8) and (9) for $z = 1$, $\beta = 0.3$.

3.4. Learning Method

The goal of this study is to learn a metric-learning-based model without an L2-norm layer so that the embedding vector is spread over an infinite vector space. We propose a two-stage learning method. In stage 1, the distribution of the embedding vector is located within a certain range by using the vector length loss. By adjusting the distribution of the embedding vector in stage 1, the model can be trained without the L2-norm layer. Stage 1 is described in Section 3.4.1. Moreover, then, in stage 2, training is performed by removing the L2-norm layer of the model trained in stage 1. Stage 2 is described in Section 3.4.2.

3.4.1. Stage 1 of Learning Method

This study has proposed a two-stage learning method for training in a wide vector space. In stage 1, the model, including the L2-norm layer, was trained equally as the existing triplet loss learning model. In this model, the triplet loss was applied to the result value of the L2-norm layer, and vector length loss was applied to the result value of the layer prior to L2-norm, as shown in Figure 4. Two loss functions were used to simultaneously train the angle and length of the embedding vector. The training was conducted to minimize the vector length loss and the triplet loss at the same time. By applying stage 1, the distribution of the embedding vector was located within a certain area. In stage 1, the accuracy was measured based on the Euclidean distance of the resulting value that passed the L2-norm layer.

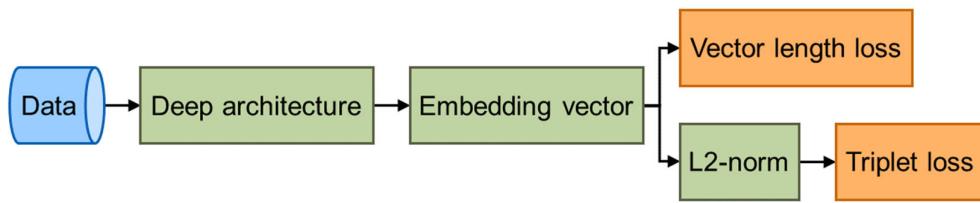


Figure 4. Stage 1 triplet loss and vector length loss learning model.

3.4.2. Stage 2 of Learning Method

In stage 2, the L2-norm layer was removed from the model, and the model was trained using only the triplet loss, as shown in Figure 5. The triplet loss for this model was trained based on the Euclidean distance rather than the cosine similarity because the L2-norm layer was removed. As a result, the embedding vector is trained in vector space, not on the surface of the sphere. The accuracy of the model was calculated based on the Euclidean distance in stage 1.

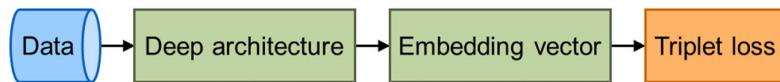


Figure 5. Stage 2 L2-normalization (L2-norm) removal and triplet loss learning model.

4. Implementation

In this paper, we trained the models to compare the novel loss function and the learning method proposed in Section 3 with the existing learning method. In Section 4.1, a toy experiment is performed with the modified national institute of standards and technology (MNIST) dataset to confirm that the proposed loss function and the learning method are generalized to other datasets using a simple CNN model. In addition, we compared the learning methodologies using a common architecture described in Section 3.1 except for the toy experiment in Section 4.1. In Section 4.2, we describe the dataset and model parameters for training the model. In Section 4.3, we compare the accuracy and loss, which are the results of the training models based on the proposed learning methodology and existing learning methodologies.

4.1. Toy Experiment

We performed a toy experiment to verify the methodology proposed in this paper. The MNIST dataset was used for the toy experiment. A simple CNN model that generates 2-D embedding vectors were trained using the proposed learning method and the existing triplet loss method. Figure 6 shows the visual image of the results of the trained model for the testing set of MNIST. In addition, Figure 6 shows the unnormalized embedding vector. The initial state in Figure 6a shows the result of extracting the embedding vector from the untrained model. The triplet loss in Figure 6b shows the result of extracting the embedding vector from the model trained by the existing triplet loss learning method, including the L2-norm layer. Stage 1 in Figure 6c is the result of extracting the embedding vector from the model trained in the first stage of the proposed learning method, and it shows a similar form to the existing triplet loss learning method. The maximum length of the embedding vector in stage 1 of the proposed learning method was less than ten. However, the maximum length of the embedding vector in the existing triplet loss method was over thousands. This means that the length of the embedding vector was adjusted by using the vector length loss in stage 1. Due to this, it became possible to perform stage 2 of the proposed learning method. As a result of stage 2 of the proposed learning method, the model classified classes based on the Euclidean distance, not the angle, as shown in Figure 6d.

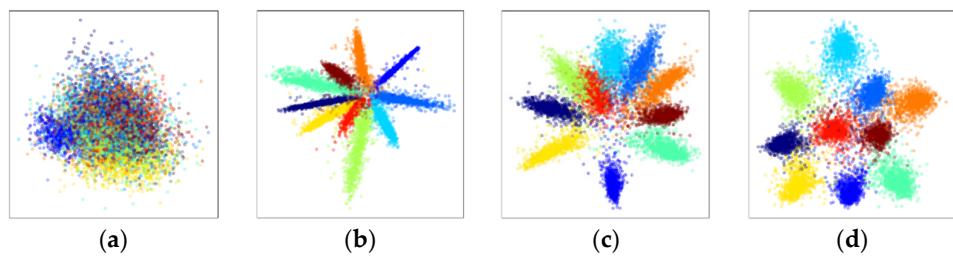


Figure 6. Results of the modified national institute of standards and technology (MNIST) values for changes in the distribution of embedding vectors of each learning method. (a) initial state; (b) triplet loss; (c) stage 1; (d) stage 2.

4.2. Dataset and Model Definition

The dog face dataset collected in [31] was used for the training set. The dog face dataset has 1393 classes of dogs, 8363 images. In [31], the author selected 485 dogs with more than 5 photos that are easy to learn from the dataset for training and testing. In addition, to prevent overfitting, the training set is augmented by zooming into the images (zoom range = 0.1), by rotating them (rotation range = 8°), and by shifting their channels (channel shift range = 0.1). In this paper, 1001 dogs, including all dogs with 5 or more images, were trained and tested as a class to verify the robustness even with low normality. A total of 7040 images of 1001 dogs were used. A total of 6460 images of 901 dogs, which was 90% of the selected dataset, were used as the training set, and the remaining 580 images of 100 dogs were used for the testing set. In addition, both the training set and the testing set are augmented by zooming, rotating, and shifting channels for data augmentation. Figure 7 shows an example of the dog face dataset.

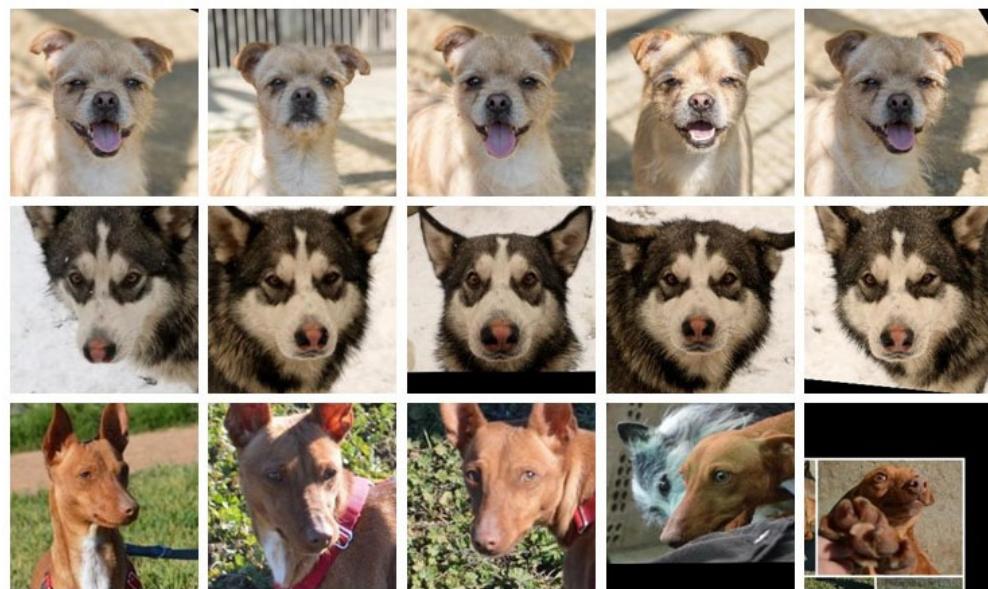


Figure 7. An example of the dog face dataset.

For the model, the ResNet-like model proposed in [31] was used by changing its input image size to $(224 \times 224 \times 3)$, as shown in Figure 2. Online adaptive hard triplet was used as a triplet dataset configuration method to sufficiently train the increased number of dogs. Unlike the existing hard triplet configuration, the adaptive hard triplet configuration method adjusted the ratio of the hard dataset according to the loss value of the model:

$$G(\text{Loss}) = \exp(-\text{Loss} \times \frac{C}{BS}) \quad (13)$$

Equation (13) calculates the hard triplet ratio to the loss and has a value between 0 and 1. BS indicates the batch size, and C indicates a constant. The sensitivity of the hard triplet ratio to loss is adjusted by the constant C . A batch size of 30 and C of 10 was used. Equation (13) was multiplied by 10 so that the maximum number of hard triplets was 10, which is 1/3 of the batch size. The number of hard triplets increased as the loss decreased and reached 10 when the loss was zero. The value in Equation (13) is rounded to an integer and used as the number of hard triplets. The same model was trained on both the base method and the proposed method for a performance evaluation and comparison. The online adaptive hard triplet was used as the dataset configuration method. The alpha of the triplet loss was 0.3. The model's compiler used Adam optimizer, and the learning rate was 1×10^{-4} in the entire epoch.

In this paper, the dataset and the ResNet-like model of [31] were used. In dog face identification based on deep learning, [31] achieved the state of the art. Therefore, based on the learning method presented in [31], the state-of-the-art (SOTA) model was trained and tested for the same dataset. In [31], offline hard triplet was used as a triplet dataset configuration method. The offline triplet dataset is regenerated every 3 epochs. The alpha of the triplet loss was 0.3. The model's compiler used Adam optimizer. The learning rate is scheduled as shown in Table 1. In this paper, a total of 3 models were compared.

Table 1. Learning rate schedule for the state-of-the-art model.

Epochs	Learning Rates
39	0.001
12	0.0005
12	0.0003
6	0.0001
Total	69
	-

For comparison, models were implemented by applying various learning methodologies to the architecture illustrated in Figure 2. We trained the model with SOTA, the existing triplet loss and the ArcFace, including the L2-norm layer as shown in Figure 1. In addition, we trained the model by combining the proposed loss function and learning method with triplet loss and the ArcFace.

4.3. Models Implementation

The SOTA model was trained using the existing triplet loss with the L2-norm layer. The minimum loss of the SOTA model for the testing set was 1.0612. The maximum accuracy of the SOTA model for the testing set was 65.00%. In this paper, the process of extracting data with a low learning difficulty was not performed. For this reason, the performance of the model based on the learning method proposed in [31] decreased. The triplet loss model was trained using the existing triplet loss with 1000 epochs. The training increased the number of hard triplets of the triplet loss model increased to 9, and the minimum loss of the triplet loss model for the testing set was 0.2069. The maximum accuracy of the triplet loss model for the testing set was 93.33%. In addition, the ArcFace model was trained using the triplet-loss-based ArcFace with 1000 epochs. The training increased the number of hard triplets of the ArcFace model increased to 9, and the minimum loss of the ArcFace model for the testing set was 0.3305. The maximum accuracy of the ArcFace model for the testing set was 84.00%.

We implemented a model that combines the proposed learning method and triplet loss. The proposed learning method was conducted in two training stages, with a total of 1200 epochs: 1000 in stage 1 and 200 in stage 2. In stage 1, both loss functions triplet loss

and vector length loss were used for training the model, and only the triplet loss was used as the loss for the hard triplet ratio. As a result of stage 1, the number of hard triplets increased to 8, and the minimum loss for the testing set was the lowest when the sum of the triplet loss of 0.2119 and the vector length loss of 0.05868 was 0.27058. The maximum accuracy for the testing set was 89%. Stage 2 was applied using the minimum loss model of stage 1. In stage 2, as shown in Figure 5, the L2-norm layer of the lowest loss model trained in stage 1 was removed, and the model was trained using only the triplet loss. As a result of stage 2, the number of hard triplets increased to 9, and the minimum loss for the testing set was 0.1833. The maximum accuracy for the testing set was 97.33%.

In addition, we implemented a model that combines the proposed learning method and ArcFace using the same epoch as the model combining triplet loss. Both loss functions, triplet-loss-based ArcFace and vector length loss, were used for training the model in stage 1. As a result of stage 1, the number of hard triplets increased to 8, and the minimum loss for the testing set was the lowest when the sum of the ArcFace loss of 0.2599 and the vector length loss of 0.0210 was 0.2809. The maximum accuracy for the testing set was 84.33%. In stage 2, our proposed learning method uses a Euclidean distance. Therefore, in stage 2 of the proposed learning method, the existing triplet loss was used. As a result of following stage 2, the number of hard triplets increased to 8, and the minimum loss for the testing set was 0.2566. The maximum accuracy for the testing set was 96.33%. Table 2 sums up the results of model implementation (Appendix A).

Table 2. Loss and accuracy of learning methods for the architecture described in Section 3.1. Vector length (VL) method indicates vector length loss.

Learning Methods	Loss	Accuracy
SOTA	1.0612	65.00%
Triplet	0.2069	93.33%
ArcFace	0.3305	84.00%
Triplet + VL stage 1	0.27058	89.00%
Triplet + VL stage 2	0.1833	97.33%
ArcFace + VL stage 1	0.2809	84.33%
ArcFace + VL stage 2	0.2566	96.33%

5. Evaluation

To compare the proposed learning method with the SOTA learning method, the triplet loss learning method and the ArcFace learning method, a performance evaluation was conducted by using the lowest loss models of Section 4.3. This evaluation used two methods of the testing set configuration: closed-set and open-set. A closed-set is the untrained dataset of the object used for training, which means the test for the object trained at the training process, whereas an open-set is a dataset of an untrained object, which means the test for the new untrained object. The test for an open-set is more difficult than that of a closed-set and is more closely related to real-world problems. The present evaluation was carried out with the open-set because the model must be able to correctly identify new objects for the return of newly occurring lost animals. The testing set for evaluation consists of 580 images of 100 untrained dogs. The model, which trained by the proposed loss function and learning method, was evaluated in the following three methods:

- Comparison of embedding vector distribution between the learning methods: The embedding vectors of the open set were extracted from the model trained by the base learning method, the SOTA learning method, and the proposed learning method, and their distributions were compared. The embedding vectors used values before the L2-norm layer. The distribution was compared by the mean and variance for the

length of the embedding vector and the distance between the center of the object's embedding vectors and the embedding vector for each object;

- Face verification: The model discerned whether a pair of images was the same object and calculated its accuracy. Specifically, the distance (d) between the two embedding vectors extracted by the model was compared to a specific threshold (t), and when $d < t$, the images were discerned as the same object. The accuracy of the discernment result is calculated. The receiver operating characteristic (ROC) curve and the optimum accuracy were compared between models while changing the threshold. This procedure was repeated 100 times to calculate the average of the ROC curves;
- Face identification: one image was selected for each object included in the open set to configure a set of 100 sub-training sets, and the remaining data comprised the sub-testing set. Rank 1 and rank 5 were extracted by comparing the distance between the embedding vectors of one sub-testing data and all sub-training data, their accuracy was calculated, and this procedure was repeated 1000 times for the entire sub-testing set to compare the mean, maximum, and minimum accuracy of rank 1 and rank 5.

5.1. Comparison of Embedding Vector Distribution between Learning Methods

The embedding vector distributions of the testing set extracted from the model trained with the SOTA learning method, the triplet loss learning method, the ArcFace learning method and the proposed learning method were compared. To compare the distributions, the mean and variance were calculated for the distance between the center of the object's embedding vectors and the embedding vector (D), as well as for the length of the embedding vector (L). In Figure 8, each point represents one object; in addition, the x-axis is D , and the y-axis is L . The mean and variance of D and L of the model trained by the SOTA learning method, the triplet loss learning method and the ArcFace learning method were much greater than the equivalent values for the model trained by the proposed learning method as shown in Figure 8. These results suggest that embeddings vectors extracted from models learned with the proposed learning method are well clustered among classes based on distance compared to embedding vectors extracted from other models. In addition, it means that learning of triplet loss using distances other than angles is possible.

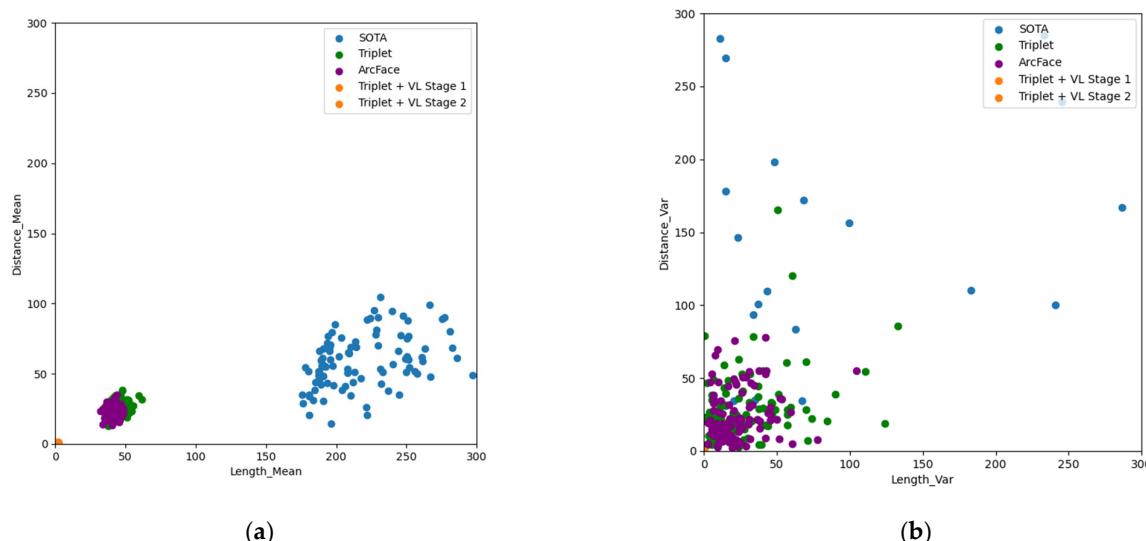


Figure 8. Mean and variance of D , which is the distance between the center of the object's embedding vectors and the embedding vector, and L , which is the length of the embedding vector, of models trained with the state-of-the-art (SOTA) learning method, the triplet loss learning method, the ArcFace learning method and models trained with the proposed learning method. (a) mean; (b) variance.

In stage 1, the variance of L was extremely small, and the mean remained at an almost similar level, as shown in Figure 9. Figure 9 shows that the variance of L of the model trained in stage 1 was extremely small, and the mean remained almost the same. This means that the length of the embedding vector is similar overall. Stage 1 of the proposed learning method is similar to the existing triplet loss learning method. Afterward, in stage 2, the variance of L increased but was extremely insignificant, and the variance of D decreased. The distribution of the mean of L was expanded, and the distribution of the mean of D was reduced. Therefore, the embedding vector was located in a wider area, and an aggregation of the vector for each object was conducted at the same time during the training. This result means that the model learned by stage 1 was learned based on the distance by performing stage 2 of the proposed learning method. As a result, we demonstrated that classification has been performed on vector spaces beyond the plane of a sphere with a radius of 1. In addition, the distribution of embedding vectors for each stage of the learning methods combined with the triplet loss and the ArcFace is similar, as shown in Figure 9. This means that the proposed learning method can be generalized even when combined with other loss functions of metric learning.

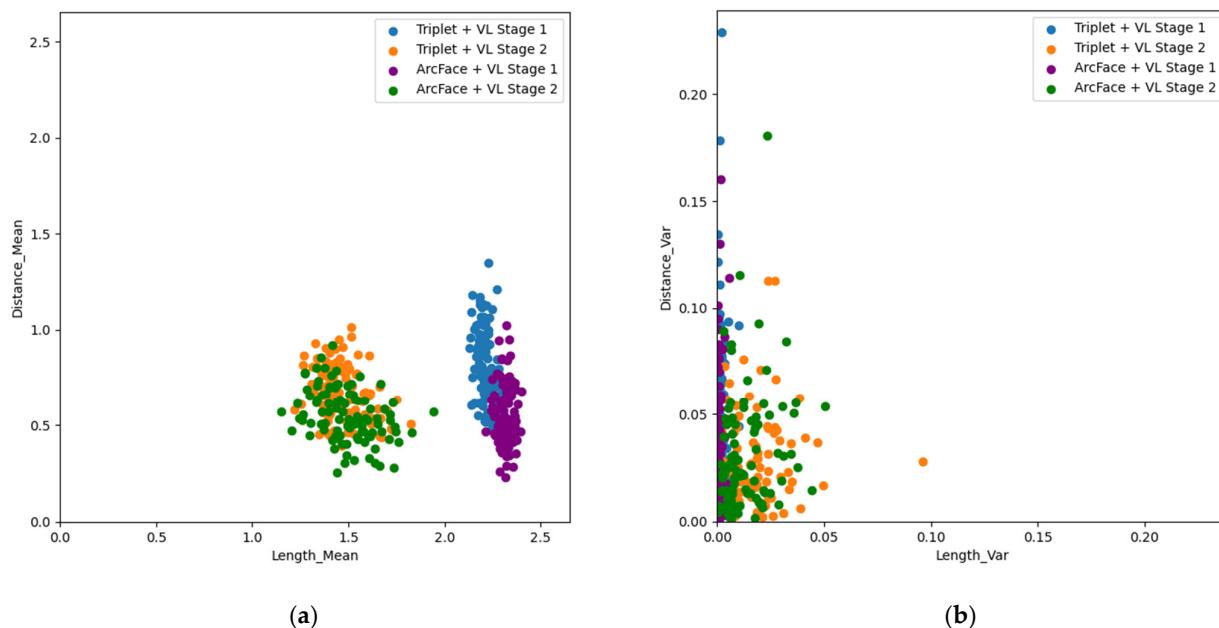


Figure 9. Mean and variance of D and L of models trained with the proposed learning method. (a) mean; (b) variance.

5.2. Face Verification

In Section 4.3, the minimum loss of the testing set was 0.2069 for the triplet loss learning method and 1.0612 for the SOTA learning method, but only 0.1833 for the proposed learning method combined with triplet loss. The optimum accuracy was improved by 4% from 93.33% for the triplet loss learning method to 97.33% for the proposed learning method combined with triplet loss. From the testing set, 2500 positive and 2500 negative pairs were generated and used to compare the performance of the three trained models. This procedure was repeated 100 times. The average ROC curves of models for face verification are shown in Figure 10. The five learning methods learned the same architecture introduced in Section 3.1. However, the triplet loss learning method, the ArcFace learning method and the proposed learning method used the online adaptive hard triplet dataset configuration method, but the SOTA learning method used the offline hard triplet dataset configuration method. For this reason, the results of other learning methods are relatively similar compared to the SOTA learning method. In addition, through the ROC curves of Figure 10, it was demonstrated in 100 repeated experiments

that the proposed learning method has better performance than the triplet loss learning method and the ArcFace learning method.

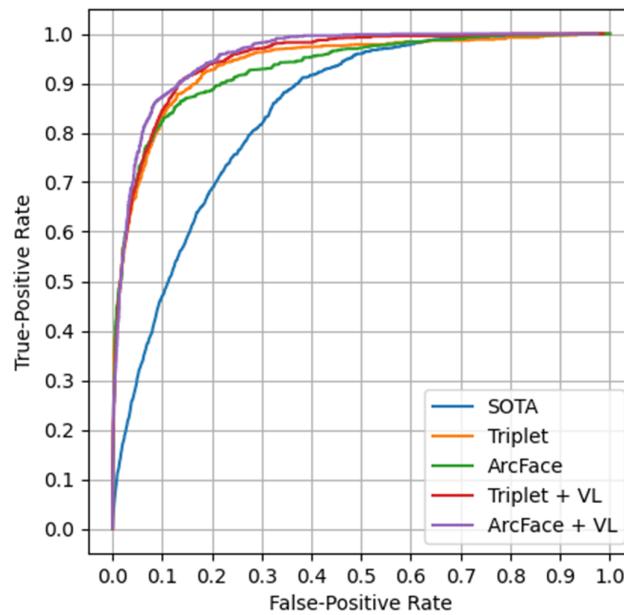


Figure 10. The average receiver operating characteristic (ROC) curves comparing the proposed learning method with the SOTA learning method, the triplet loss learning method and the ArcFace learning method.

Table 3 sums up the results of face verification. Table 3 shows the performance of the models using the best threshold of each model and the performance of the models using the value of α used for learning. The best accuracy and the threshold of each model were measured while changing the threshold. The mean of the best accuracy of the model trained with the SOTA learning method was 76.9%, and the mean of the threshold was 0.38. The mean of the best accuracy of the model trained with the triplet loss learning method was 87.0%, and the mean of the threshold was 1.29. The mean of the best accuracy of the model trained with the ArcFace learning method was 86.4%, and the mean of the threshold was 1.31. The best mean of the accuracy of the model trained with the proposed learning method combined with triplet loss was 88.4%, and the mean of the threshold was 2.49. The best mean of the accuracy of the model trained with the proposed learning method combined with ArcFace was 88.8%, and the mean of the threshold was 1.49. As a result, the accuracy of the proposed learning method with ArcFace was 1.8% higher than that of the triplet loss learning method. This means that the error rate is reduced by about 13.9 percent. Some examples of false-positive and false-negative pairs of the proposed learning method are presented, as shown in Figure 11. A false-negative error has occurred when the difference in lighting and angle of the same object is large. In addition, the error occurred when the degree of the face being covered by an object is large.

Table 3. Results of the mean of the optimum accuracy and the threshold for the learning methods in face verification.

Learning Methods	Threshold	Accuracy	Threshold	Accuracy
SOTA	$\alpha = 0.3$	76.0%	<i>best</i> = 0.38	76.9%
Triplet	$\alpha = 0.3$	55.9%	<i>best</i> = 1.29	87.0%
ArcFace	$\alpha = 0.3$	55.2%	<i>best</i> = 1.31	86.4%
Triplet + VL	$\alpha = 0.3$	54.3%	<i>best</i> = 2.49	88.4%
ArcFace + VL	$\alpha = 0.3$	60.2%	<i>best</i> = 1.49	88.8%



Figure 11. Example of face verification of the proposed learning method. (a) false-positive pairs; (b) false-negative pairs.

5.3. Face Identification

The accuracy of the one-shot identification at rank 1 and rank 5 was evaluated using the minimum loss models in Section 4.3. Models trained with the SOTA learning method, the triplet loss learning method and the ArcFace learning method included the L2-norm layer at the end, and models trained with the proposed learning method did not. From the 580 testing set for 100 dogs, one image was randomly selected for each object to configure 100 sub-training sets, and the remaining 480 images were configured as sub-testing sets. Rank 1 and rank 5 were extracted by comparing the distance between the embedding vectors of one sub-testing data and all sub-training data. The accuracy of the extracted rank 1 and rank 5 was calculated and compared. The experiment was repeated 1000 times. Table 4 represents the experiment results. As a result, the mean of the rank 1 and rank 5 of the model trained with the SOTA learning method was 10.96% and 32.58%, respectively. The mean of the rank 1 and rank 5 of the model trained with the triplet loss learning method was 37.52% and 65.84%, respectively, and this accuracy was improved by 2.22% and 2.96% to 39.74% and 68.80%, respectively, with the model trained using the proposed learning method combined with triplet loss. The model trained with the proposed learning method combined with triplet loss obtained the highest mean, maximum, and minimum accuracy for both rank 1 and rank 5.

Table 4. Results of mean, maximum, and minimum accuracy of rank 1 and rank 5 for the learning methods in face identification.

Learning Methods	Rank 1			Rank 5		
	Mean	Maximum	Minimum	Mean	Maximum	Minimum
SOTA	10.96%	14.79%	6.67%	32.58%	38.54%	26.25%
Triplet	37.52%	45.21%	31.04%	65.84%	72.50%	59.58%
ArcFace	38.41%	44.38%	32.92%	65.92%	70.83%	60.83%
Triplet + VL	39.74%	47.08%	32.92%	68.80%	73.96%	63.12%
ArcFace + VL	34.92%	41.67%	28.96%	67.57%	73.12%	61.25%

6. Conclusions

Although many studies have been conducted on animal identification, the performance improvement has been limited to the application of the existing face identification methodology owing to the low normality of dog faces and the lack of related data. This paper proposed to overcome these limitations with a novel loss function and learning method that utilize a wide vector space and thereby improves the performance of the dog face identification model. The triplet loss used in the existing face identification model was trained based on cosine similarity using the L2-norm layer. This paper proposes the training of a model without an L2-norm layer using vector length loss, which is a novel loss function for the length of the embedding vector, and its two-stage learning method. The embedding vector was distributed over the surface of the sphere into a wider vector space, which improved the performance of the dog face identification model by 4%.

The distribution of embedding vectors is limited compared to the infinite vector space, but this paper proposes the novel loss function and learning method to utilize the vector space wider than the surface of the sphere. The proposed methodology improves the model's performance compared to the existing methodology for the same model. As a result, this paper overcomes the limitation of triplet loss in face identification and improves the performance of the dog face identification model, which is expected to help return lost dogs. Furthermore, the model's performance will be further improved by using the proposed loss function and its learning method for other models and datasets using the existing triplet loss and L2-norm layer.

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

Funding: This work was carried out with the support of “Cooperative Research Program of Center for Companion Animal Research (Project No. PJ0139862020)” Rural Development Administration, Korea.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: <https://github.com/GuillaumeMougeot/DogFaceNet/releases/>.

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

Appendix A

In this paper, for comparison, we trained the model with SOTA, the existing triplet loss and the ArcFace, including the L2-norm layer. In addition, we trained the model by combining the proposed loss function and learning method with triplet loss and the

ArcFace. The SOTA model was trained using the existing triplet loss learning method with 69 epochs, as shown in Figure A1.

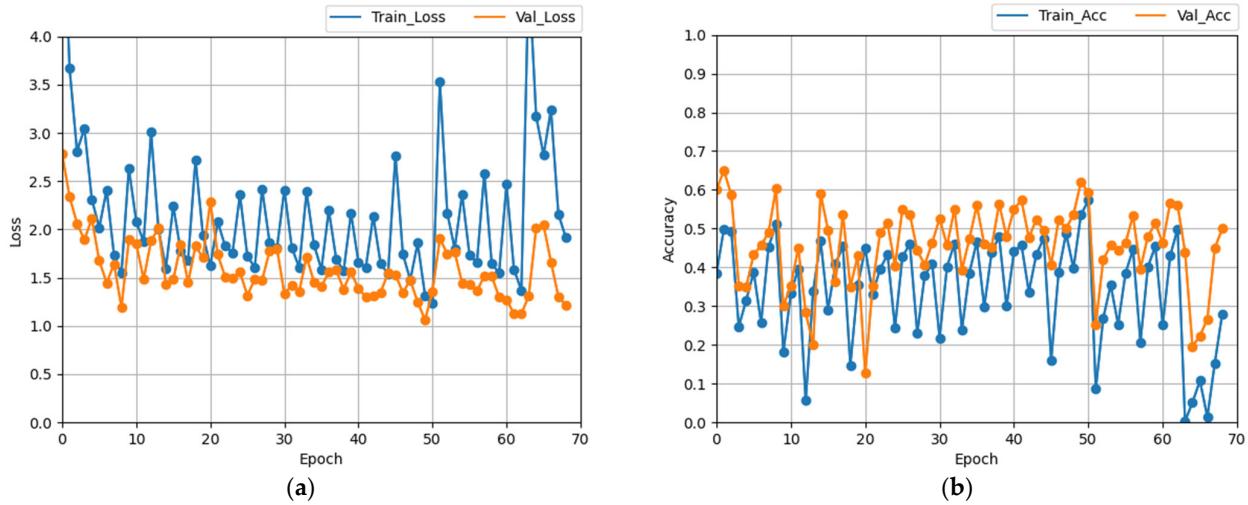


Figure A1. Training result for the state-of-the-art (SOTA) learning method. (a) loss; (b) accuracy.

The triplet loss model was trained using the existing triplet loss learning method with 1000 epochs. Figure A2 illustrates the training results.

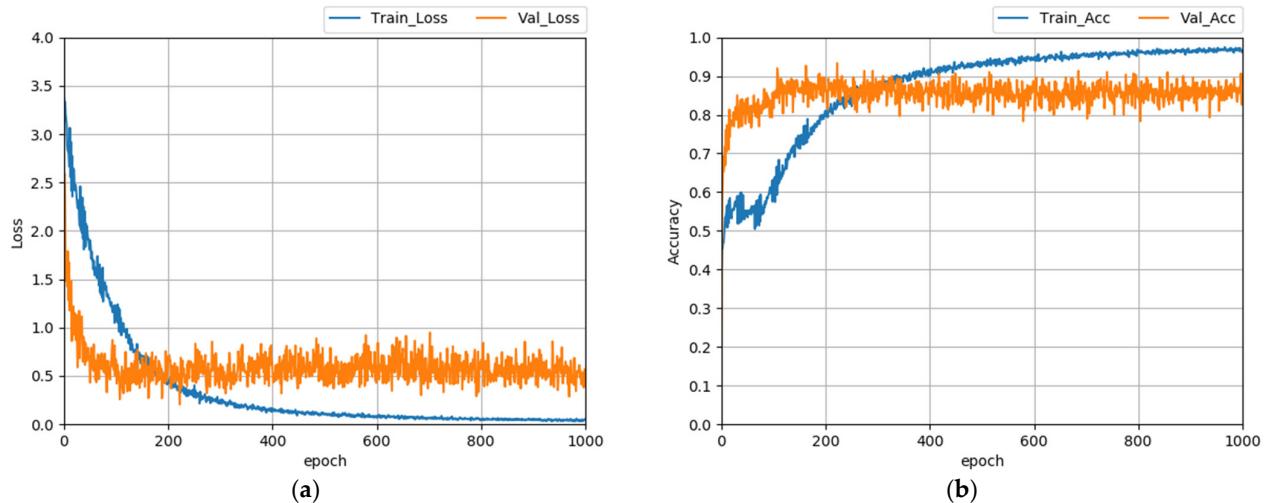


Figure A2. Training result for the triplet loss learning method. (a) loss; (b) accuracy.

In addition, the ArcFace model was trained using the ArcFace learning method with 1000 epochs. Figure A3 illustrates the training results.

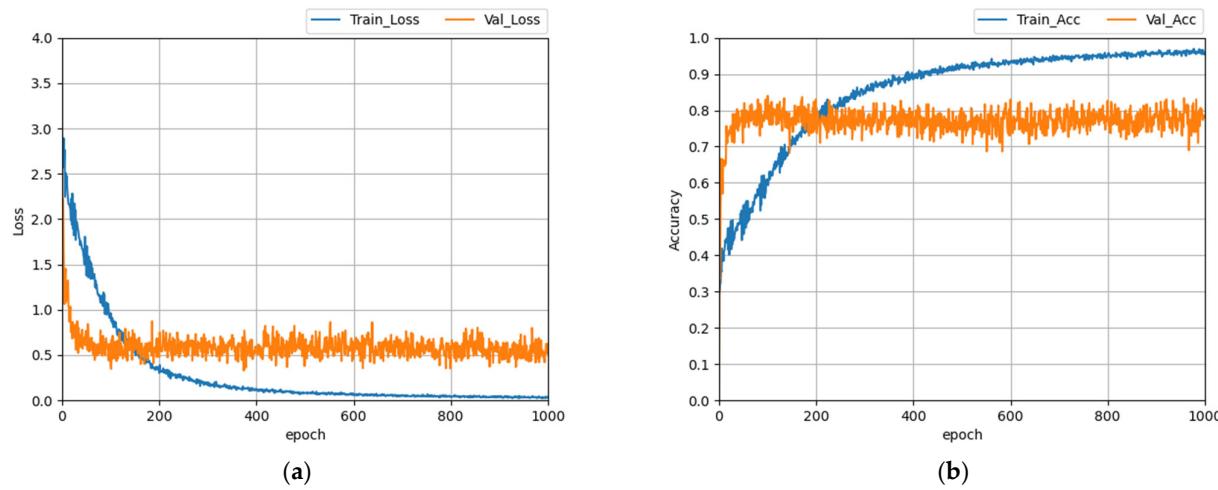
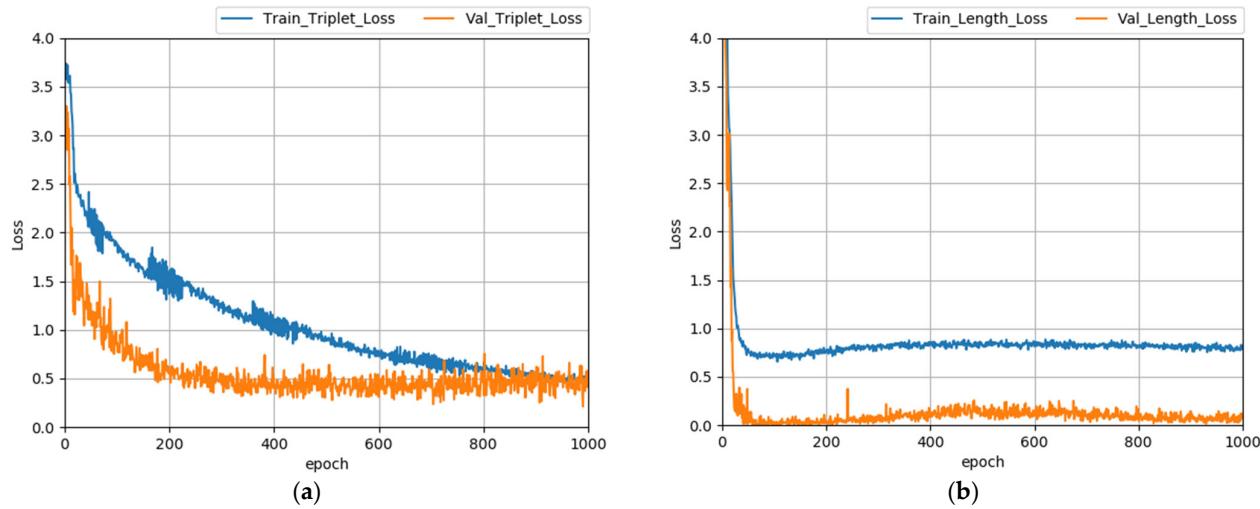


Figure A3. Training result for the ArcFace learning method. (a) loss; (b) accuracy.

The model was trained using the proposed learning method combined with the triplet loss and the ArcFace, respectively. First, the proposed model was trained with the triplet loss in stage 1. Figure A4 illustrates the training results.



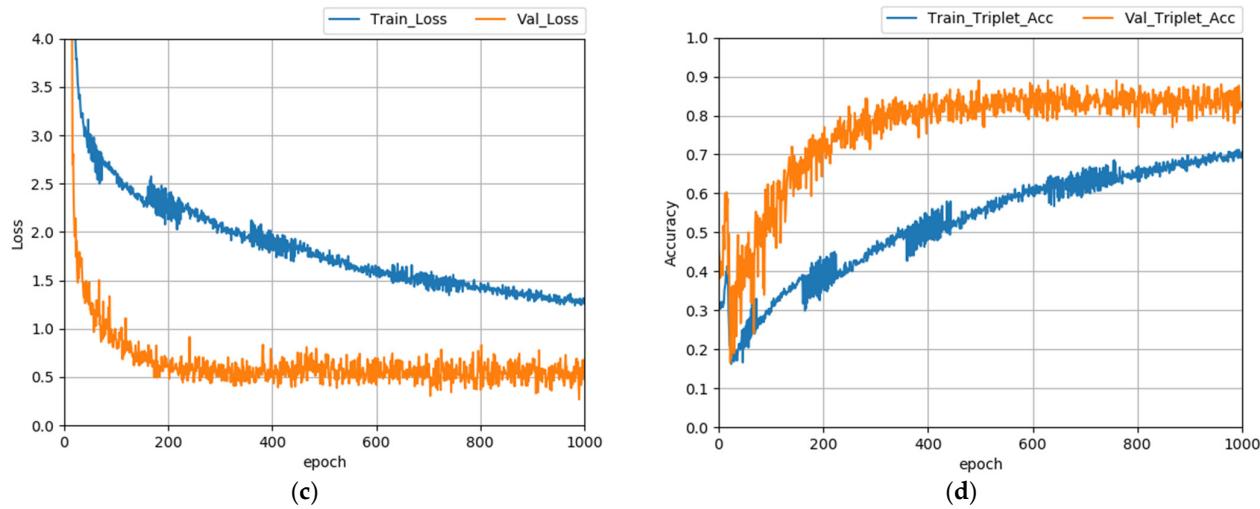


Figure A4. Stage 1 training result of the proposed learning method combined with the triplet loss. (a) triplet loss; (b) vector length loss; (c) total loss; (d) triplet accuracy.

In stage 2, the L2-norm layer of the lowest loss model trained in stage 1 was removed, and the model was trained using only the triplet loss. Figure A5 illustrates the training results of stage 2.

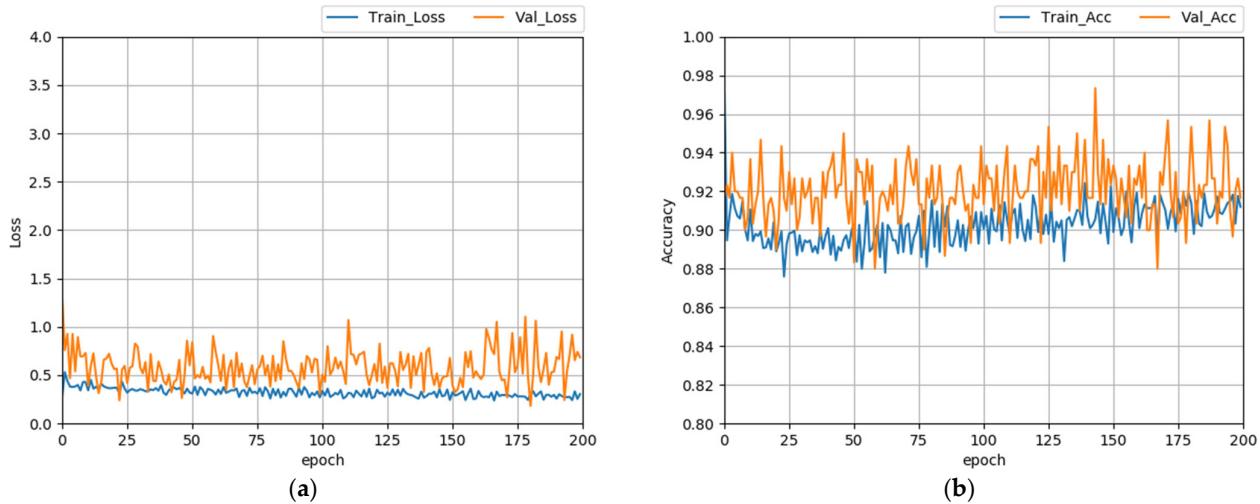


Figure A5. Stage 2 training result of the proposed learning method with the triplet loss. (a) triplet loss; (b) accuracy.

Second, the proposed model was trained with the ArcFace in stage 1. Figure A6 illustrates the training results.

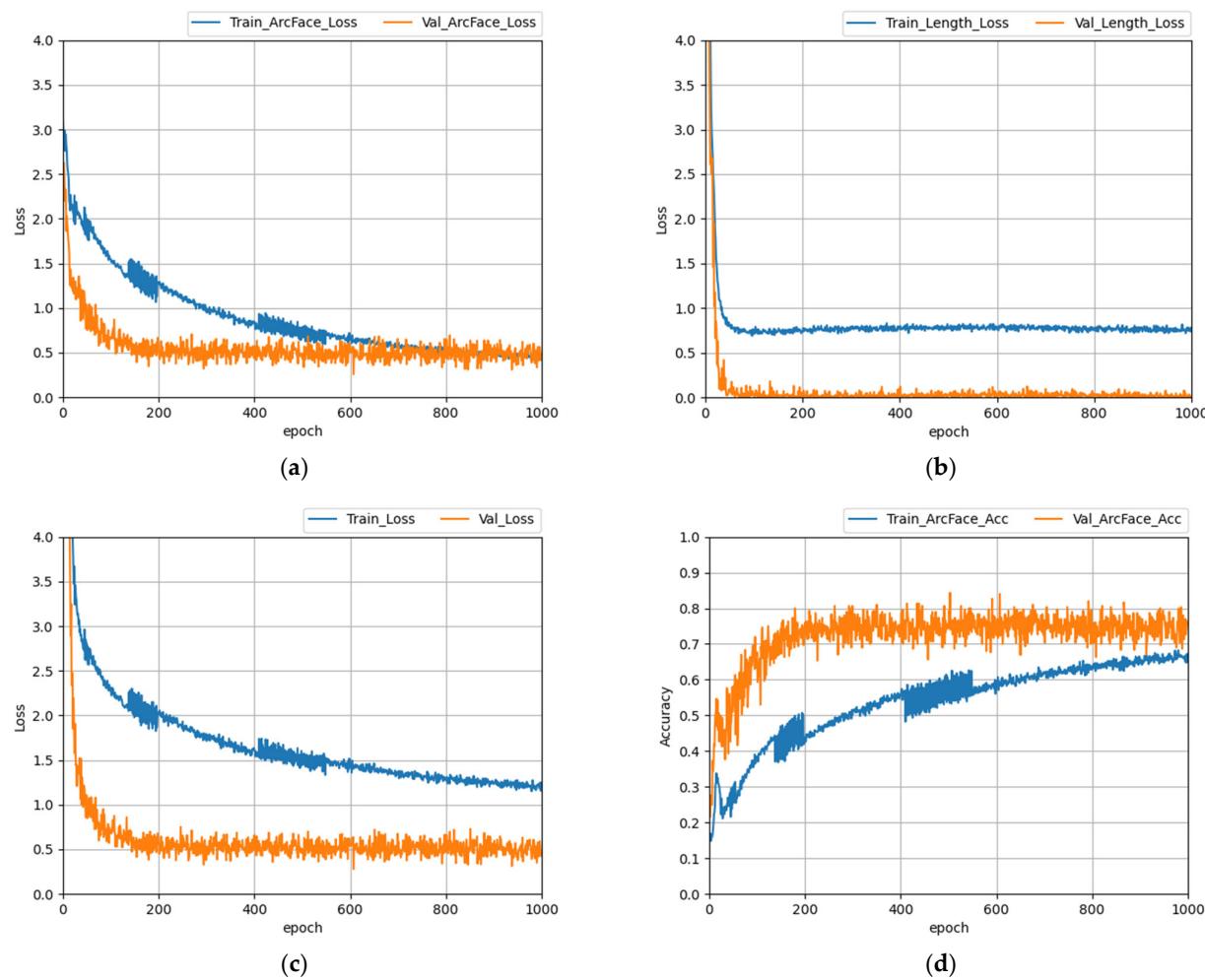


Figure A6. Stage 1 training result of the proposed learning method with the ArcFace. (a) ArcFace loss; (b) vector length loss; (c) total loss; (d) ArcFace accuracy.

In stage 2, the L2-norm layer of the lowest loss model trained in stage 1 was removed, and the model was trained using only the triplet loss. Figure A7 illustrates the training results of stage 2.

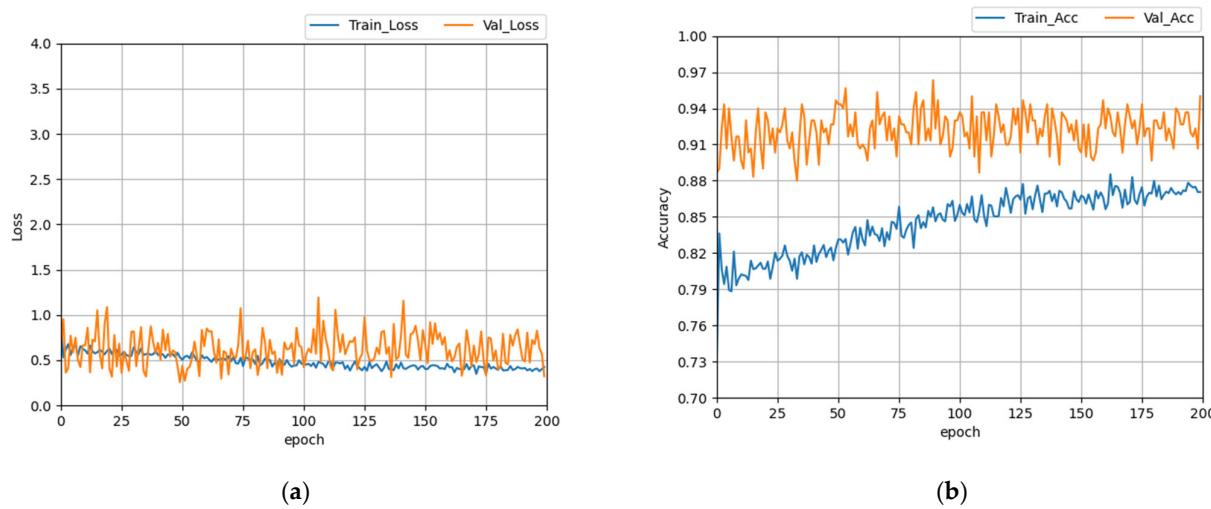


Figure A7. Stage 2 training result of the proposed learning method with the ArcFace. (a) loss; (b) accuracy.

Appendix B

In this paper, we proposed a new loss function named vector length loss. The vector length loss uses two variables, x and z . In vector length loss, x is the smaller value of the two embedding vector lengths, and z is the absolute value of the difference in length of the two normalized embedding vectors. The two variables, x and z , are equal to Equations (B1) and (B2), respectively:

$$x = \text{Min} (\|f(x_a)\|, \|f(x_p)\|) \quad (\text{B1})$$

$$z = \text{Abs} (\|f(x_a)\| - \|f(x_p)\|) \quad (\text{B2})$$

The vector length loss is calculated using only the anchor and positive in a dataset composed of a triplet.

References

1. Kumar, S.; Singh, S.K. Monitoring of pet animal in smart cities using animal biometrics. *Future Gener. Comput. Syst.* **2018**, *83*, 553–563.
2. Musgrave, C.; Cambier, J.L. System and Method of Animal Identification and Animal Transaction Authorization Using Iris Patterns. US Patent 6,424,727, 23 July 2002.
3. Trigueros, D.S.; Meng, L.; Hartnett, M. Face recognition: From traditional to deep learning methods. *arXiv* **2018**, arXiv:1811.00116.
4. Schultz, M.; Joachims, T. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems 16*; MIT Press: London, UK, 2004; pp. 41–48.
5. Weinberger, K.Q.; Blitzer, J.; Saul, L.K. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 18*; MIT Press: London, UK, 2006; pp. 1473–1480.
6. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
7. Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. Sphereface: Deep hypersphere embedding for face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 212–220.
8. Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; Liu, W. Cosface: Large margin cosine loss for deep face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5265–5274.
9. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4690–4699.
10. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3320–3328.
11. Sivic, J.; Everingham, M.; Zisserman, A. “Who are you?”-Learning person specific classifiers from video. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1145–1152.

12. Chen, D.; Cao, X.; Wang, L.; Wen, F.; Sun, J. Bayesian face revisited: A joint formulation. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 566–579.
13. Simonyan, K.; Parkhi, O.M.; Vedaldi, A.; Zisserman, A. Fisher vector faces in the wild. *BMVC* **2013**, *2*, 4. DOI: 10.5244/C.27.8
14. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 539–546.
15. Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1701–1708.
16. Zheng, Y.; Pal, D.K.; Savvides, M. Ring loss: Convex feature normalization for face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5089–5097.
17. Zhang, Y.; Deng, W.; Wang, M.; Hu, J.; Li, X.; Zhao, D.; Wen, D. Global-local gcn: Large-scale label noise cleansing for face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 7731–7740.
18. Kim, Y.; Park, W.; Roh, M.C.; Shin, J. Groupface: Learning latent groups and constructing group-based representations for face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 5621–5630.
19. Awad, A.I. From classical methods to animal biometrics: A review on cattle identification and tracking. *Comput. Electron. Agric.* **2016**, *123*, 423–435.
20. Kumar, S.; Singh, S.K.; Singh, R.S.; Singh, A.K.; Tiwari, S. Real-time recognition of cattle using animal biometrics. *J. Real-Time Image Process.* **2017**, *13*, 505–526.
21. Kumar, S.; Pandey, A.; Satwik, K.S.R.; Kumar, S.; Singh, S.K.; Singh, A.K.; Mohan, A. Deep learning framework for recognition of cattle using muzzle point image pattern. *Measurement* **2018**, *116*, 1–17.
22. Kumar, S.; Singh, S.K.; Abidi, A.I.; Datta, D.; Sangaiah, A.K. Group sparse representation approach for recognition of cattle on muzzle point images. *Int. J. Parallel Program.* **2018**, *46*, 812–837.
23. Jarraya, I.; Ouarda, W.; Alimi, A.M. A preliminary investigation on horses recognition using facial texture features. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong, China, 9–12 October 2015; pp. 2803–2808.
24. Hansen, M.F.; Smith, M.L.; Smith, L.N.; Salter, M.G.; Baxter, E.M.; Farish, M.; Grieve, B. Towards on-farm pig face recognition using convolutional neural networks. *Comput. Ind.* **2018**, *98*, 145–152.
25. Crouse, D.; Jacobs, R.L.; Richardson, Z.; Klum, S.; Jain, A.; Baden, A.L.; Tecot, S.R. LemurFaceID: A face recognition system to facilitate individual identification of lemurs. *BMC Zool.* **2017**, *2*, 2. DOI: 10.1186/s40850-016-0011-9
26. Deb, D.; Wiper, S.; Gong, S.; Shi, Y.; Tymoszek, C.; Fletcher, A.; Jain, A.K. Face recognition: Primates in the wild. In Proceedings of the IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS), Los Angeles, CA, USA, 22–25 October 2018; pp. 1–10.
27. Liu, J.; Kanazawa, A.; Jacobs, D.; Belhumeur, P. Dog breed classification using part localization. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 172–185.
28. Wang, X.; Ly, V.; Sorensen, S.; Kambhamettu, C. Dog breed classification via landmarks. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 5237–5241.
29. Hsu, D. Using Convolutional Neural Networks to Classify Dog Breeds. CS231n: Convolutional Neural Networks for Visual Recognition. 2015. Available online: http://cs231n.stanford.edu/reports/2015/pdfs/fcdh_FinalReport.pdf (accessed on 26 February 2021).
30. Ayanzadeh, A.; Vahidnia, S. Modified Deep Neural Networks for Dog Breeds Identification. *Preprints* **2018**, doi:10.20944/preprints201812.0232.v1.
31. Mougeot, G.; Li, D.; Jia, S. A Deep Learning Approach for Dog Face Verification and Recognition. In Proceedings of the Pacific Rim International Conference on Artificial Intelligence, Cuvu, Fiji, 26–30 August 2019; pp. 418–430.
32. Moreira, T.P.; Perez, M.L.; de Oliveira Werneck, R.; Valle, E. Where is my puppy? retrieving lost dogs by facial features. *Multimed. Tools Appl.* **2017**, *76*, 15325–15340.
33. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv* **2013**, arXiv:1312.6229.
34. Tu, X.; Lai, K.; Yanushkevich, S. Transfer learning on convolutional neural networks for dog identification. In Proceedings of the IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 23–25 November 2018; pp. 357–360.
35. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497.
36. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.