applied sciences

MDPI

Article

Additive Orthant Loss for Deep Face Recognition

Younghun Seo and Nam Yul Yu *

School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, Korea

* Correspondence: nyyu@gist.ac.kr

Abstract: In this paper, we propose a novel loss function for deep face recognition, called the additive orthant loss (Orthant loss), which can be combined for softmax-based loss functions to improve the feature-discriminative capability. The Orthant loss makes features away from the origin using the rescaled softplus function and an additive margin. Additionally, the Orthant loss compresses feature spaces by mapping features to an orthant of each class using element-wise operation and 1-bit quantization. As a consequence, the Orthant loss improves the inter-class separabilty and the intra-class compactness. We empirically show that the ArcFace combined with the Orthant loss further compresses and moves the feature spaces farther away from the origin compared to the original ArcFace. Experimental results show that the new combined loss has the most improved accuracy on CFP-FP, AgeDB-30, and MegaFace testing datasets, among some of the latest loss functions.

Keywords: deep learning; discriminative feature learning; face recognition

1. Introduction

Face recognition (FR) [1,2] based on deep convolutional neural networks (DCNN) [3–7] is one of the fields in computer vision that found great success in the 2010s, and it has been actively studied as a biometric authentication [8–10] technique for face verification and identification. In particular, deep FR has better recognition accuracy for images with large variations such as pose, illumination, age, and expression compared to FR without using DCCNs. The good recognition accuracy of the deep FR comes from the fact that each DCNN layer is trained on a face image from various perspectives. It is important to design an efficient loss function that can distinguish identities well using a training dataset with a limited number of images and identities. To improve the recognition accuracy, various loss functions [11–17] as well as network architectures [18–21] have been studied.

Many loss functions for FR have been studied based on the softmax function [22] for image classification problems. In [23,24], the authors first trained a deep face verification model using the softmax function. Wen et al. [25] proposed the Center loss that uses both the softmax function and an additive term to improve the intra-class compactness. Liu et al. [26] proposed a multiplicative angular margin to increase the cosine similarity of each class. In [27], Wang et al. showed that the cosine similarity can be optimized using a new scaling constant replacing the magnitudes of the inner product between weight and feature vectors. Liu et al. [28] proposed the angular softmax function that moves to the hypersphere manifold through weight normalization. In [29], Gao et al. proposed the Margin loss that combines an additive term with the Center loss to enlarge the distances between centers of each class. Wu et al. [30] proposed the Center invariant loss combined with the center loss to solve a bias caused by imbalanced image data between identities. In [31,32], the authors proposed a loss function by introducing an additive angular margin to increase the cosine similarity of each class. Deng et al. [33] proposed the ArcFace function that performs feature and weight normalizations in addition to the use of an additive angular margin. In [34], Ou et al. proposed the LinCos-Softmax that replaces the cosine similarity between feature and weight vectors with an approximated linear angle

check for updates

Citation: Seo, Y.; Yu, N.Y. Additive Orthant Loss for Deep Face Recognition. *Appl. Sci.* **2022**, *12*, 8606. https://doi.org/10.3390/app12178606

Academic Editor: George A. Papakostas

Received: 11 July 2022
Accepted: 24 August 2022
Published: 28 August 2022

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

Copyright: © 2022 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 (https://creativecommons.org/licenses/by/4.0/).

to prevent the angle saturation caused by the nonlinearity of cosine similarity. In [35], Meng et al. proposed an adaptive loss function that efficiently learns the distributions of features using the magnitudes of features. Tao et al. [36] proposed the FCGFace loss that combines the ArcFace with two additive terms to compactly guide profile face features to each center of the frontal face features. In [37], Wang et al. proposed the RVFace that drops the noisy feature vectors and adaptively learn the distributions of features using semi-hard feature vectors to improve the feature discriminative capability. For surveys on more network structures and loss functions for deep FR, readers are referred to [38,39].

Many recent loss functions [31–34] only use the cosine similarity between weight and feature vectors in the hypersphere manifold without feature magnitudes. Then, even if feature vectors have high cosine similarities with weight vectors of their corresponding class, the feature magnitudes may be small. In this case, the feature vectors have a problem that the cosine similarity can be changed significantly when the signs of features generated from images with large variation are changed due to their small magnitudes. On the other hand, in the property of monotonicity of [35], it has been proven that feature magnitudes increase with the cosine similarity. It has also been shown that an optimal feature magnitude exists for a feature by the property of convergence of [35]. Based on the above properties, we can see that it is necessary to increase the feature magnitude to find an optimal one as well as to improve cosine similarity.

In this paper, we propose an additive orthant loss function, termed the Orthant loss, to improve the inter-class separability and the intra-class compactness. The Orthant loss improves the inter-class separability by increasing the distance between features and the origin using the rescaled softplus function and an additive margin. Additionally, the 1-bit quantization and element-wise operation of the Orthant loss simultaneously improve the intra-class compactness of each feature space by attempting to make the features of the same identity have the same signs. In this paper, the proposed loss function is termed as the Orthant loss, since we try to map features of the same identity to an orthant where the center of the corresponding class is located. The Orthant loss has two major differences from the Center loss [25]. First, the Center loss requires an additional fully connected layer to learn the center location of each class, whereas the Orthant loss does not require the layer. In addition, the Center loss learns the distributions of all features to gather at the center of each class without considering the feature magnitudes, while the Orthant loss learns the distributions to efficiently utilize the feature spaces by considering the feature magnitudes. Since our new loss function has no network modification, it is compatible with many of the latest softmax-based loss functions. Thus, we construct a final loss function by combining one of the lastest softmax-based loss functions [33,35] with the Orthant loss.

The experimental results show that a new loss function combining the ArcFace and the Orthant loss together, termed the ArcOrthFace, has the better accuracy compared to softmax-based loss functions, e.g., Center loss [25], SphereFace [28], ArcFace [33] and MagFace [35], respectively, on CFP-FP [40], AgeDB-30 [41] and MegaFace [42] testing datasets. Additionally, we confirm that the Orthant loss combined with the softmax-based functions [22,33,35] improves the recognition accuracy in most benchmarks. In conclusion, the Orthant loss, combined with the latest softmax-based functions, can improve the accuracy of deep face recognition by enhancing the inter-class separability and the intra-class compactness of feature spaces.

2. Background

2.1. Deep Face Recognition System

As shown in Figure 1, a deep FR system consists of training and testing phases. In the training phase, DCNNs such as AlexNet [3], VGGNet [4] and ResNet [6] are trained to obtain the discriminative features $\mathbf{x}$ of preprocessed face images. The goal of training is to learn a set of parameters $\mathcal{W}$ of the DCNN and the weight matrix $\mathbf{W} = (\mathbf{w}_1, \cdots, \mathbf{w}_M) \in \mathbb{R}^{l \times M}$ of the last fully connected layer, where l is the dimension of features and M is the number of identities of a training dataset. The optimization problem of the deep FR is formulated by:

$$\left(\widehat{\mathcal{W}}, \widehat{\mathbf{W}}\right) = \arg\min_{\mathcal{W},\mathbf{W}} \mathcal{L}(\mathbf{x}; \mathcal{W}, \mathbf{W}),$$

where $\mathcal{L}$ is a loss function, and $\widehat{\mathcal{W}}$ and $\widehat{\mathbf{W}}$ are learned parameters of $\mathcal{W}$ and $\mathbf{W}$, respectively. Once the model is trained, verification or identification is performed in the testing phase using a distance measure, e.g., cosine similarity or Euclidean distance, to determine whether the feature obtained from a test image is from a registered user of the system.

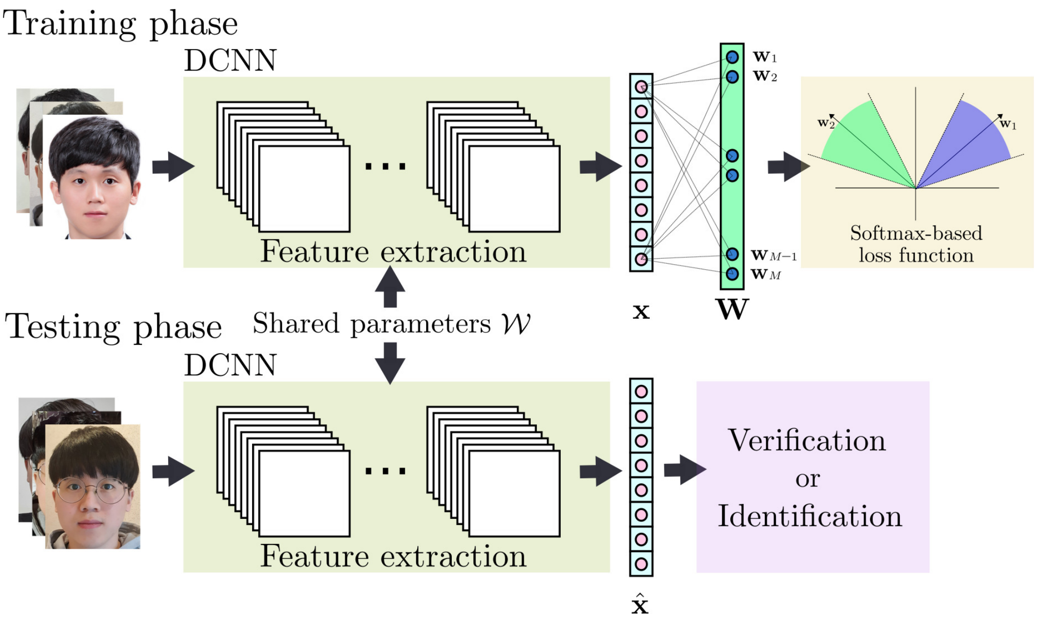


Figure 1. Deep face recognition (FR) system model.

2.2. Some Known Loss Functions

The softmax function [22] has been mainly used as a loss function, which is given by:

$$\begin{aligned}\mathcal{L}_{\text{Softmax}} &= -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\mathbf{w}_{y_i}^T\mathbf{x}_i}}{e^{\mathbf{w}_{y_i}^T\mathbf{x}_i} + \sum_{j\neq y_i} e^{\mathbf{w}_j^T\mathbf{x}_i}} \\ &= -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s_{y_i}\cos(\theta_{y_i})}}{e^{s_{y_i}\cos(\theta_{y_i})} + \sum_{j\neq y_i} e^{s_j\cos(\theta_j)}},\end{aligned} \tag{1}$$

where $\mathbf{w}_{y_i}$ and $\mathbf{w}_j$ are the y_ith and the jth column of $\mathbf{W}$, respectively, $\mathbf{x}_i$ is a feature obtained by inputting the ith image into DCNN, and N is the batch size. In (1), $\mathbf{w}_{y_i}^T\mathbf{x}_i = s_{y_i}\cos(\theta_{y_i})$ and $\mathbf{w}_j^T\mathbf{x}_i = s_j\cos(\theta_j)$. While $\mathcal{L}_{\text{Softmax}}$ is known in [23] to have good capability to improve the separability between classes (inter-class separability), it has insufficient capability to compress the feature space of each class (intra-class compactness).

In order to improve the intra-class compactness, many softmax-based loss functions have been studied. The Center loss [25] is given by

$$\mathcal{L}_{\text{Center}} = \mathcal{L}_{\text{Softmax}} + \frac{\lambda}{2}\sum_{i=1}^{N}\|\mathbf{x}_i - \mathbf{c}_{y_i}\|^2, \tag{2}$$

where $\mathbf{c}_{y_i}$ is the center position of the y_ith class, λ is a hyperparameter for balancing the two terms, and $\|\cdot\|$ is the l_2-norm of a vector. $\mathcal{L}_{\text{Center}}$ improves the intra-class compactness by penalizing the distances between features and their corresponding class centers.

The SphereFace [28] is formulated as:

$$\mathcal{L}_{\text{Sphere}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{\|\mathbf{x}_i\|\psi(\theta_{y_i})}}{e^{\|\mathbf{x}_i\|\psi(\theta_{y_i})}+\sum_{j\neq y_i}e^{\|\mathbf{x}_i\|\cos(\theta_j)}},$$

where $\psi(\theta_{y_i}) = (-1)^k\cos(m_p\theta_{y_i}) - 2k$, $\theta_{y_i} \in \left[\frac{k\pi}{m_p}, \frac{(k+1)\pi}{m_p}\right]$ and $k \in [0, m_p - 1]$, in which m_p is a multiplicative angular margin. $\mathcal{L}_{\text{Sphere}}$ uses an multiplicative angular margin to improve the intra-class compactness on a hypersphere manifold.

The ArcFace [33] has been proposed by:

$$\mathcal{L}_{\text{ArcFace}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos(\theta_{y_i}+m_\theta)}}{e^{s\cos(\theta_{y_i}+m_\theta)}+\sum_{j\neq y_i}e^{s\cos(\theta_j)}},$$

where m_θ is an additive angular margin, and s is a scaling constant. $\mathcal{L}_{\text{ArcFace}}$ improves the intra-class compactness using an additive angular margin on a hypersphere manifold.

The MagFace [35] is given by:

$$\mathcal{L}_{\text{MagFace}} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathcal{L}_{\text{Mag}} + \lambda_g g(\|\mathbf{x}_i\|)\right),$$

where $\mathcal{L}_{\text{Mag}} = -\log\frac{e^{s\cos(\theta_{y_i}+m_g(\|\mathbf{x}_i\|))}}{e^{s\cos(\theta_{y_i}+m_g(\|\mathbf{x}_i\|))}+\sum_{j\neq y_i}e^{s\cos(\theta_j)}}$, in which $m_g(\|\mathbf{x}_i\|)$ is a strictly increasing function, $g(\|\mathbf{x}_i\|)$ is a strictly decreasing function and λ_g is a hyperparameter that controls the trade-off between $\mathcal{L}_{\text{Mag}}$ and $g(\|\mathbf{x}_i\|)$. $\mathcal{L}_{\text{MagFace}}$ uses an adaptive margin $m_g(\|\mathbf{x}_i\|)$ and regularization $g(\|\mathbf{x}_i\|)$ to improve the intra-class compactness.

3. Orthant Loss

In this paper, we propose an additive orthant loss, termed the Orthant loss, to improve the inter-class separability and the intra-class compactness. The Orthant loss learns the distributions of features away from the origin and compresses the feature spaces by attempting to make the features of the same identity have the same signs. The Orthant loss is proposed by:

$$\mathcal{L}_{\text{Orth}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{l} a\left[\frac{1}{r}\log\left(1+e^{-r(\bar{x}_{i,k}Q_1(w_{y_{i,k}})-m_c)}\right)\right]^2, \tag{3}$$

where $\bar{x}_{i,k}$ is the kth element of a normalized feature $\bar{\mathbf{x}}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}$, $w_{y_{i,k}}$ is the kth element of $\mathbf{w}_{y_i}$ and $Q_1(w)$ is the 1-bit quantization function, i.e., $Q_1(w) = +1$ if $w > 0$ and -1 otherwise. In addition, m_c is an additive margin, a is a hyperparamter to adjust the size of the loss, r is a hyperparameter to control the slope of the rescaled softplus function, l is the feature dimension and N is the batch size. The proposed algorithm of $\mathcal{L}_{\text{Orth}}$ is given in Algorithm 1. The reasoning behind the design of Orthant loss is to make each feature element have the same sign as the weight vector of each class by making $\bar{x}_{i,k}Q_1(w_{y_{i,k}}) > 0$ using 1-bit quantization and element-wise operation, which improves the intra-class compactness. In addition, the Orthant loss makes the magnitude of each feature element larger than m_c to improve the inter-class separability by using the additive margin and the rescaled softplus function. Since $\mathcal{L}_{\text{Orth}}$ only considers the feature and weight vectors of target identity y_i in (3), we combine $\mathcal{L}_{\text{Orth}}$ with a softmax-based loss $\mathcal{L}_{\text{Soft}}$, taking into account non-target identities $j \neq y_i$, which gives a final loss:

$$\mathcal{L}_{\text{Final}} = \mathcal{L}_{\text{Soft}} + \mathcal{L}_{\text{Orth}}. \tag{4}$$

Algorithm 1 The pseudo-code of Orthant loss $\mathcal{L}_{\text{Orth}}$

```
Input: Feature vectors $\mathbf{x}_i$, Weight matrix $\mathbf{W} = (\mathbf{w}_1, \cdots, \mathbf{w}_M)$, Ground-Truth ID $y_i$, Hyper-parameters $a, r$ and $m_c$.
    Normalize the feature vectors by $\bar{\mathbf{x}}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}$
    Clone and detach the weight matrix $\mathbf{W}$ to prevent gradient computation by
        $\mathbf{W} = \mathbf{W}$.clone().detach()
    Quantize the chosen weight vectors corresponding to $y_i$ by
        $\tilde{\mathbf{w}}_{y_i} = \left[Q_1(w_{y_i,1}), Q_1(w_{y_i,2}), \cdots, Q_1(w_{y_i,l})\right]$
    Calculate the element-wise magnitudes including margin $m_c$ by
        $\mathbf{u}_i = \tilde{\mathbf{w}}_{y_i} \odot \bar{\mathbf{x}}_i - m_c$, where $\odot$ is the element-wise product of two vectors.
    Calculate the proposed loss in (3) using the rescaled softplus function for a batch by
        $\mathcal{L}_{\text{Orth}} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{l} a \left[\frac{1}{r} \log\left(1 + e^{-r u_{i,k}}\right)\right]^2$
Output: Loss score $\mathcal{L}_{\text{Orth}}$
```

In what follows, we describe the features of the Orthant loss.

Rescaled softplus function. To learn the distributions of features to be well separated, we use the rescaled softplus function [43] $f_s(x) = \frac{1}{r} \log(1 + e^{-rx})$, where r controls the slope of $f_s(x)$. As r goes to infinity, $f_s(x)$ converges approximately into the hinge function [44] $f_h(x) = \max(0, -x)$. In $f_h(x)$, if $x \geq 0$, learning no longer proceeds as the gradient becomes 0. On the other hand, in $f_s(x)$, even if $x \geq 0$, learning is possible so that x is farther away from zero since the gradient of $f_s(x)$ is not zero. Therefore, we use $f_s(x)$ to improve the inter-class separability by distinguishing features around the origin.

Additive center separating margin. Figure 2a shows that a softmax-based loss function [28–35] learns the distribution of features centered on $\mathbf{w}_1$ and $\mathbf{w}_2$ to improve the intra-class compactness using an angular margin m_θ, respectively. However, since features with small magnitudes can easily move to the feature spaces of other classes, the softmax-based loss function using an angular margin has difficulty in distinguishing these features correctly. To overcome this problem, this paper uses an additive margin m_c to improve the inter-class separability by pushing the features away from the origin, as illustrated in Figure 2b. In the Center loss [25], which also uses an additive term, both ambiguous and discriminative features can be gathered at the center of each class without considering the feature magnitudes. On the other hand, both ambiguous and discriminative features move away from the origin in the Orthant loss, which improves the inter-class separability by increasing their magnitudes.

One-bit quantization and element-wise operation. To improve the intra-class compactness, we quantize each weight vector of the last fully connected layer and perform element-wise product on the normalized feature vectors. When the weight vectors are not quantized, the signs of some elements of normalized feature vectors can be easily changed if the magnitudes of the corresponding elements of the weight vectors are small. Therefore, we train the DCNN model so that the signs of feature and weight vectors are identical by multiplying the normalized feature by the 1-bit quantized weight vector. Then, we can improve the intra-class compactness through the element-wise sign consistency between feature and quantized weight vectors.

Despite the low computational cost, the 1-bit quantization may reduce the recognition accuracy due to the inaccurate gradient update [45]. Since 1-bit quantization has zero gradients on positive and negative sides, weight vectors of the last fully connected layer are updated in a wrong direction using a surrogate gradient function such as hard hyperbolic tangent (tanh) function. To maintain the accuracy while reducing the computational cost, we exclude the 1-bit quantized weight vectors from the gradient calculation. In other words, the weight vectors are updated using the softmax-based loss $\mathcal{L}_{\text{Soft}}$ without the Orthant loss $\mathcal{L}_{\text{Orth}}$ in the back-propagation process. Note that the Orthant loss $\mathcal{L}_{\text{Orth}}$ is combined to a softmax-based loss when updating the set of parameters $\mathcal{W}$ of the DCNN.

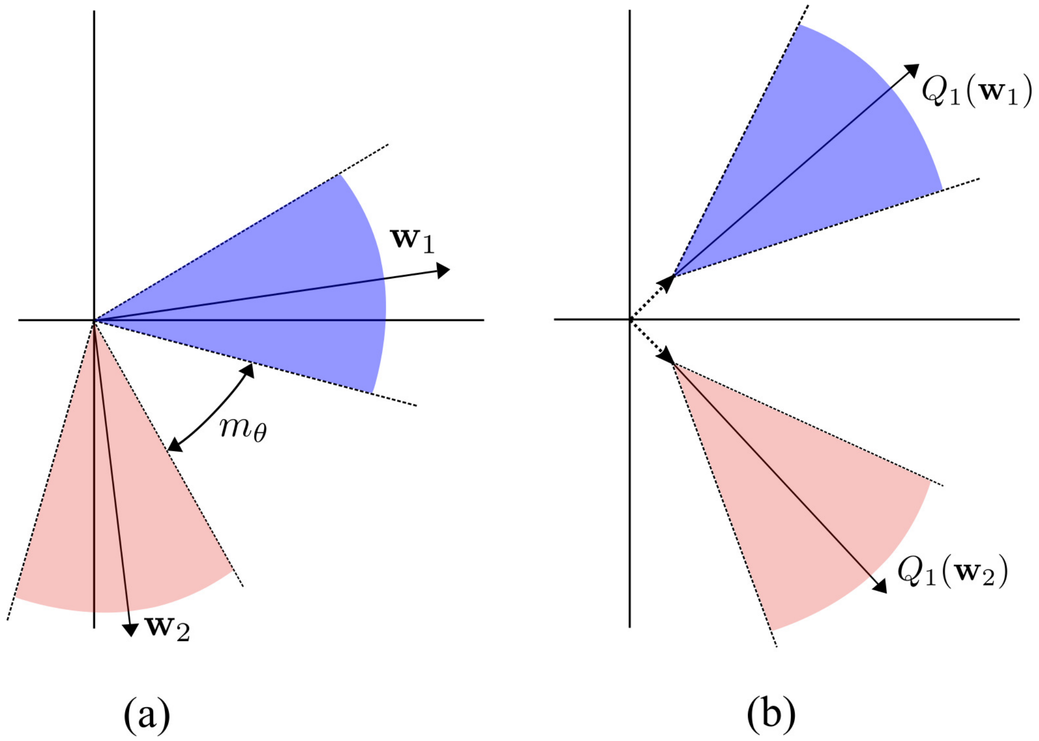
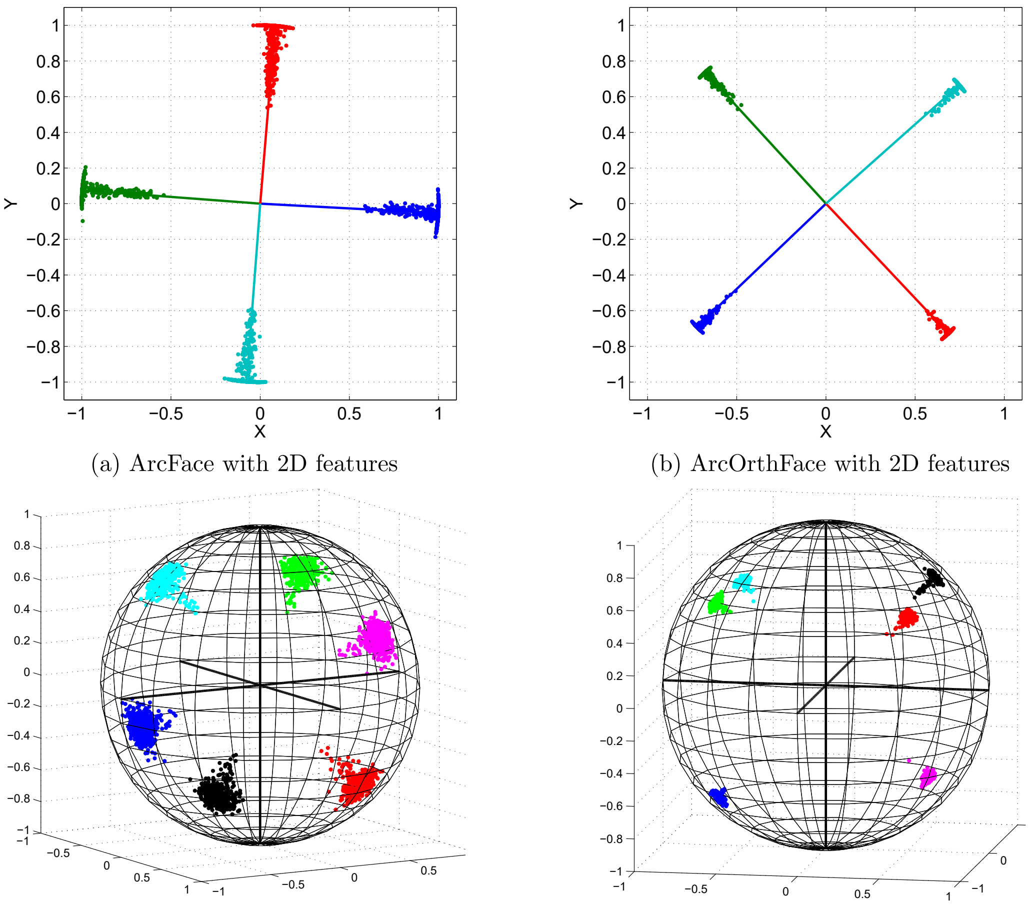


Figure 2. Geometric interpretation of feature space learned by a softmax-based loss function (**a**) without and (**b**) with Orthant loss, respectively.

Squared loss function. As r increases, the gradient for a misclassified feature in the rescaled softplus function $f_s(x)$ is approximately -1. Since $f_s(x)$ has the gradient of -1 for both easy and hard features to distinguish, features do not quickly converge. To improve the convergence speed, we introduce a squared loss to give a larger weight to a hard feature. In many studies [44,46], the squared hinge function has been used to give a larger weight to misclassified features. Inspired by this idea, we introduce a squared softplus function to expect fast convergence of misclassified features.

In the learning process, 1-bit quantization and element-wise operation of the Orthant loss make feature vectors have the same sign as the weight vectors of the corresponding class. However, feature vectors may converge to an orthant of the corresponding weight vector before the distances between weight vectors are sufficiently large, which fixes the sign of the weight vectors prematurely. In other words, since the speed at which feature vectors converge to the corresponding weight vector is faster than the speed at which the weight vectors move away from each other, the sign of the weight vectors may be fixed before the distance between the weight vectors increases. To avoid a premature decision of weight vectors, we apply the Orthant loss after the learning progresses sufficiently. In this paper, we apply the Orthant loss from 20,000th iteration.

Figure 3 displays 2D and 3D plots of features for the ArcFace and the ArcOrthFace, respectively. We use the LResNet18E-IR [33], where $a = 2$, $r = 30$, and $N = 128$. In addition, we use $s_{y_i} = s_j = 4$, $m_\theta = 0.1$ and $m_c = 1/\sqrt{2}$ for 2D features of Figure 3a,b and $s_{y_i} = s_j = 8$, $m_\theta = 0.2$ and $m_c = 1/\sqrt{3}$ for 3D features of Figure 3c,d, respectively. For training and testing, we use a total of 2907 images of top four classes and a total of 4186 images of the top six classes sorted by the number of images in the CASIA-Webface dataset [47] for 2D and 3D features, respectively. We assume that there are $d = 4$ and $d = 6$ identities for 2D and 3D features, respectively. Here, we choose the number of identities by $d \leq 2^l$, which is valid in practice. In Figure 3, we project features outside the unit sphere onto the sphere, where the features do not cause inter-class ambiguity due to the sufficiently high magnitude. In this example, Figure 3a,b have 2907 features, respectively, while Figure 3c,d have 4186 features, respectively.



(a) ArcFace with 2D features

(b) ArcOrthFace with 2D features

(c) ArcFace with 3D features

(d) ArcOrthFace with 3D features

Figure 3. Visualization of features learned by the ArcFace function (**a**,**c**) without and (**b**,**d**) with the Orthant loss, respectively. Each class has a different color and each point represents a feature.

In Figure 3b,d, we can see that ArcOrthFace has fewer features in the unit sphere by attempting to push features away from the origin, compared to ArcFace. It suggests that ArcOrthFace has improved inter-class separability compared to ArcFace, since the fewer features cause less inter-class ambiguity around the origin. In specific, for 2D and 3D features, we can observe that ArcOrthFace reduces the number of features inside the unit sphere to 278 and 100, while 604 and 344 features remain by ArcFace, respectively. As a result, ArcOrthFace, which learns the distributions of features away from the origin, shows the improved inter-class separability, compared to ArcFace.

Figure 3b,d also show that ArcOrthFace has the better intra-class compactness than ArcFace by compressing feature spaces further at the center of each class through 1-bit quantization and element-wise operation. The feature spaces learned by ArcOrthFace are distributed around 1-bit quantized vectors, i.e., $\{-1, +1\}^l$, which demonstrates the improved intra-class compactness, compared to ArcFace.

4. Experimental Results

Preprocessing. We use the datasets given in Table 1 for training and testing several loss functions. All the preprocessed images used for training and testing are obtained by [33] for fair comparisons. For preprocessing of each face image, five landmarks are acquired using the multi-task cascaded convolutional networks (MTCNN) [48], alignment is performed using similarity transformation, and then cropped to the image with a size of 112 × 112. Each RGB image having a range of $[0, 255]$ is normalized to have a range of $[-1, 1]$ and exceptionally to have a range of $[0, 1]$ on the MagFace [35].

Table 1. Datasets for Training and Testing.

Datasets	**Number of Identities**	**Number of Images**	**Types**
CASIA-Webface [47]	10 K	0.5 M	Training
LFW [49]	5749	13,233	Validation
CFP-FP [40]	500	7000	Validation
AgeDB-30 [41]	568	16,488	Validation
MegaFace(pro.) [42]	530	3530	Testing
MegaFace(dis.) [42]	690 K	1 M	Testing

Training. Table 2 presents common training setting and hyperparameters. In order to fairly compare verification and identification accuracies that depend only on loss functions, we fix the network structure and training dataset. The recognition accuracies for the methods of comparison may be different from those in the original papers, which used their own training datasets and network structures. The learning rate starts at 0.1 and is changed to 0.01 at 20,000th iteration and 0.001 at 28,000th iteration, respectively. We use a horizontal random flip for data augmentation. For the hyperparameters of Center loss, SphereFace, ArcFace and MagFace not shown in Table 2, we use the recommended values of [25,28,33,35], respectively. In Tables 3 and 4, the loss function replacing the suffix 'max' or 'Face' with 'OrthFace', respectively, indicates a combined one with the Orthant loss. In addition, the loss function replacing the suffix 'Face' in ArcFace and MagFace with 'CentFace' indicates a combined one with the Center loss.

Table 2. Training Setting and Hyperparamters.

Network model	LResNet50E-IR [33]
Batch size N	512
Feature length l	512
Optimizer	SGD Optimizer
Momentum	0.9
Weight decay	0.0005
Total iteration	$32,000$
Scaling constant s	64
Margin size m_c	$1/\sqrt{l}$
Slope control factor r	30
Margin control factor a	2

Table 3. Verification Accuracy (%) on Several Benchmarks with Various Loss Functions.

Loss Function	**LFW**	**CFP-FP**	**AgeDB-30**
Softmax	99.27	94.67	92.90
SoftOrthFace	99.30	94.94	92.60
N-Softmax	98.47	89.81	89.12
N-SoftOrthFace	98.75	92.96	90.62
ArcFace [33]	99.43	95.57	94.95
ArcCentFace	**99.45**	95.41	95.13
ArcOrthFace	99.42	**95.73**	**95.18**
MagFace [35]	99.18	94.97	94.62
MagCentFace	99.32	95.33	94.88
MagOrthFace	99.42	94.80	94.67
Center loss [25]	99.28	94.84	92.05
SphereFace [28]	99.12	94.49	92.68

Note: The accuracies in bold represent the highest ones among the listed loss functions.

Table 4. Identification and Verification Results (%) on MegaFace with Various Loss Functions.

Loss Function	Rank1 Accuracy	Verification Accuracy
ArcFace	91.70	93.95
ArcCentFace	91.39	93.85
ArcOrthFace	**91.77**	**94.32**
MagFace	89.54	92.57
MagCentFace	91.22	93.96
MagOrthFace	89.79	92.83
Center loss	79.96	83.45
SphereFace	81.95	88.06

Note: The accuracies in bold represent the highest ones among the listed loss functions.

Testing. In the testing phase, features are extracted using the trained DCNN model, as shown in Figure 1. To make a feature from a testing image, we add the features obtained by original and flipped images and then normalize the combined feature. To evaluate the accuracy for LFW [49], CFP-FP [40] and AgeDB-30 [41], we measure the mean accuracy of 10 subsets of each dataset using the leave-one-out cross validation [50], following the unrestricted protocol with labeled outside data [49]. For the MegaFace dataset, we investigate the true positive rate (TPR) at 10^{-6} false positive rate (FPR) for face verification and rank-1 accuracy [3] for face identification with 10^{6} distractors, respectively.

Figure 4 shows the training loss of the new loss functions combined with the proposed Orthant loss function. We can observe that the new combined loss functions reduce the training loss to make a good-fitting model as training progresses. In particular, at the 20,000th and 28,000th iterations, we use a learning rate decay method to improve the training speed and model accuracy. In addition, it can be confirmed that the training loss does not diverge or converge to zero to avoid underfitting and overfitting, respectively, since we use dropout and weight decay techniques.

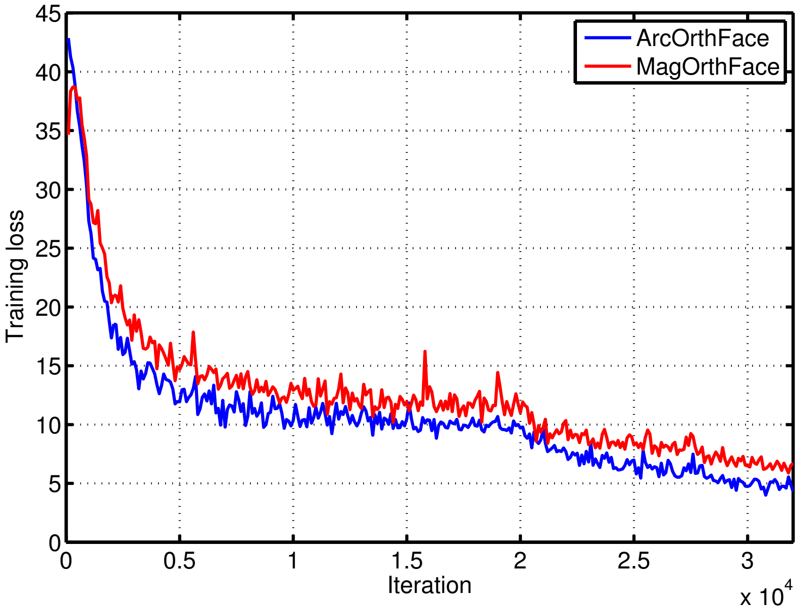


Figure 4. Training loss of the new loss functions combined with the Orthant loss function.

Table 3 shows the verification accuracy of various loss functions for LFW, CFP-FP and AgeDB-30, respectively. Table 3 demonstrates that the loss functions using Orthant loss improve the accuracy over the original ones in most benchmarks. This means that Orthant loss can improve the inter-class separability and the intra-class compactness by combining it with softmax-based loss functions. Additionally, ArcOrthFace, which combines ArcFace with Orthant loss, shows the highest accuracy on CFP-FP and AgeDB-30. In addition, it shows that SoftOrthFace has higher accuracy than Center loss, which demonstrates that Orthant loss has more discriminative capability than the additive term of Center loss. Although the accuracies of ArcCentFace and MagCentFace have increased compared to original ones, it can be seen that ArcOrthFace has the best accuracy in most benchmarks.

Table 4 shows identification and verification results of various loss functions for MegaFace. Similar to the results in Table 3, it can be seen that the loss functions using Orthant loss improve identification and verification accuracies. It demonstrates that Orthant loss makes feature spaces distinguishable by improving the inter-class separability and the intra-class compactness. In addition, it shows that MagOrthFace has higher accuracy than MagFace, which demonstrates that Orthant loss can improve the distinguishability of adaptively learned feature spaces of MagFace. In Table 4, it is shown that MagCentFace further improves the distinguishability of the adaptively learned feature spaces compared to MagOrthFace. However, ArcOrthFace shows the highest identification and verification accuracies compared to all the other loss functions.

5. Conclusions

In this paper, we proposed an additive Orthant loss to improve the inter-class separability and the intra-class compactness. The Orthant loss uses the rescaled softplus function and an additive margin to move features away from the origin. In addition, the Orthant loss uses 1-bit quantization and element-wise operation to map features to an orthant of each class. Experimental results demonstrated that loss functions combined with Orthant loss improve the recognition accuracy in most benchmarks compared to original loss functions. In particular, we showed that the loss function combining ArcFace with Orthant loss has the best accuracy on MegaFace, a challenging test benchmark containing more than 1 million distractors. In conclusion, Orthant loss, combined with a softmax-based loss function, can improve the accuracy of deep face recognition by improving the intra-class compactness and the inter-class separability. We believe that the proposed Orthant loss can be used as a loss function in recognition systems using various human characteristics, e.g., fingerprint, iris, and palmprint, which will require a future study for each recognition system.

Author Contributions: Conceptualization, Y.S.; methodology, Y.S.; software, Y.S.; investigation, Y.S.; writing—original draft preparation, Y.S.; writing—review and editing, N.Y.Y.; supervision, N.Y.Y.; project administration, Y.S. and N.Y.Y.; funding acquisition, N.Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by GIST Research Project grant funded by the GIST in 2022.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Jain, A.K.; Li, S.Z. *Handbook of Face Recognition*; Springer: New York, NY, USA, 2011.
2. Sun, Y.; Chen, Y.; Wang, X.; Tang, X. Deep learning face representation by joint identification-verification. In Proceedings of the Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, QC, Canada, 8–13 December 2014; pp. 1988–1996.
3. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
4. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2019**, arXiv:1409.1556.
5. 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 (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
6. 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 (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.

7. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
8. Shi, Y.; Jain, A.K. Probabilistic face embeddings. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 6902–6911.
9. Shi, Y.; Yu, X.; Sohn, K.; Chandraker, M.; Jain, A.K. Towards universal representation learning for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 6817–6826.
10. Ali, A.; Testa, M.; Bianchi, T.; Magli, E. Biometricnet: Deep unconstrained face verification through learning of metrics regularized onto gaussian distributions.In Proceedings of the European Conference on Computer Vision (ECCV). Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 133–149.
11. 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 (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823.
12. Sun, Y.; Wang, X.; Tang, X. Deeply learned face representations are sparse, selective, and robust. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 2892–2900.
13. Ding, C.; Tao, D. Robust face recognition via multimodal deep face representation. *IEEE Trans. Multimed.* **2015**, *17*, 2049–2058. [CrossRef]
14. Sankaranaratanan, S.; Alavi, A.; Castillo, C.D.; Chellappa, R. Triplet probabilistic embedding for face verification and clustering. In Proceedings of the 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS), Niagara Falls, NY, USA, 6–9 September 2016; pp. 1–8.
15. Liu, B.; Deng, W.; Zhong, Y.; Wang, M.; Hu, J.; Tao, X.; Huang, Y. Fair loss: Margin-aware reinforcement learning for deep face recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 10052–10061.
16. Sun, Y.; Cheng, C.; Zhang, Y.; Zhang, C.; Zheng, L.; Wang, Z.; Wei, Y. Circle loss: A unified perspective of pair similarity optimization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 6398–6407.
17. Wen, Y.; Liu, W.; Weller, A.; Raj, B.; Singh, R. Sphereface2: Binary classification is all you need for deep face recognition. *arXiv* **2021**, arXiv:2108.01513.
18. Sun, Y.; Wang, X.; Tang, X. Sparsifying neural network connections for face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4856–4864.
19. Wu, X.; He, R.; Sun, Z.; Tan, T. A light cnn for deep face representation with noisy labels. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2884–2896. [CrossRef]
20. Duong, C.N.; Quach, K.G.; Jalata, I.; Le, N.; Luu, K. Mobiface: A lightweight deep learning face recognition on mobile devices. In Proceedings of the 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS), Tampa, FL, USA, 23–26 September 2019; pp. 1–6.
21. Zhu, N.; Yu, Z.; Kou, C. A new deep neural architecture search pipeline for face recognition. *IEEE Access* **2020**, *8*, 91303–91310. [CrossRef]
22. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
23. 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 (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1701–1708.
24. Sun, Y.; Wang, X.; Tang, X. Deep learning face representaion from predicting 10,000 classes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1891–1898.
25. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 499–515.
26. Liu, W.; Wen, Y.; Yu, Z.; Yang, M. Large-margin softmax loss for convolutional neural networks. In Proceedings of the International Conference on Machine Learning (ICML), New York, NY, USA, 19–24 June 2016; pp. 507–516.
27. Wang, F.; Xiang, X.; Cheng, J.; Yuille, A.L. Normface: L2 hypersphere embedding for face verification. In Proceedings of the 25th ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 1041–1049.
28. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 212–220.
29. Gao, R.; Yang, F.; Yang, W.; Liao, Q. Margin loss: Making faces more separable. *IEEE Signal Process. Lett.* **2018**, *25*, 308–312. [CrossRef]
30. Wu, Y.; Liu, H.; Li, J.; Fu, Y. Improving face representation learning with center invariant loss. *Image Vis. Comput.* **2018**, *79*, 123–132. [CrossRef]
31. 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 (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 5265–5274.
32. Wang, F.; Cheng, J.; Liu, W.; Liu, H. Additive margin softmax for face verification. *IEEE Signal Process. Lett.* **2018**, *25*, 926–930. [CrossRef]

33. 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 (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4690–4699.
34. Ou, W.; Po, L.; Zhou, C.; Zhang, Y.; Feng, L.; Rehman, Y.A.U.; Zhao, A.Y. LinCos-softmax: Learning angle-discriminative face representations with linearity-enhanced cosine logits. *IEEE Access* **2020**, *8*, 109758–109769. [CrossRef]
35. Meng, Q.; Zhao, S.; Huang, Z.; Zhou, F. Magface: A universal representation for face recognition and quality assessment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 14225–14234.
36. Tao, Y.; Zheng, W.; Yang, W.; Wang, G.; Liao, Q. Frontal-centers guided face: Boosting face recognition by learning pose-invariant features. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2272–2283. [CrossRef]
37. Wang, X.; Wang, S.; Liang, Y.; Gu, L.; Lei, Z. Rvface: Reliable vector guided softmax loss for face recognition. *IEEE Trans. Image Process.* **2022**, *31*, 2337–2351. [CrossRef] [PubMed]
38. Wang, M.; Deng, W. Deep face recognition: A survey. *Neurocomputing* **2021**, *429*, 215–244. [CrossRef]
39. Fuad, M.T.H.; Fime, A.A.; Sikder, D.; Iftee, M.A.R.; Rabbi, J.; Al-rakhami, M.S.; Gumaei, A.; Sen, O.; Fuad, M.; Islam, M.N. Advances in deep learning techniques for face recognition. *IEEE Access* **2021**, *9*, 99112–99142. [CrossRef]
40. Sengupta, S.; Chen, J.-C.; Castillo, C.; Patel, V.M.; Chellappa, R.; Jacobs, D.W. Frontal to profile face verification in the wild. In Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, USA, 7–10 March 2016; pp. 1–9.
41. Moschoglou, S.; Papaioannou, A.; Sagonas, C.; Deng, J.; Kotsia, I.; Zafeiriou, S. AgeDB: The first manually collected, in-the-wild age database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 51–59.
42. Kemelmacher-Shlizerman, I.; Seitz, S.M.; Miller, D.; Brossard, E. The megaface benchmark: 1 million faces for recognition at scale. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4873–4882.
43. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS), Ft. Lauderdale, FL, USA, 11–13 April 2011.
44. Hastie, T.; Tibshirani, R.; Friedman, J.H. *The Elements of Statistical Learning*; Springer: Berlin/Heidelberg, Germany, 2003.
45. Liu, C.; Chen, P.; Zhuang, B.; Shen, C.; Zhang, B.; Ding, W. Sa-bnn: State-aware binary neural network. In Proceedings of the AAAI Conference on Artificial Intelligence, virtually, 2–9 February 2021; pp. 2091–2099.
46. Kartaphy, A.; Li, F.F.; Johnson, J. CS231n: Convolutional neural networks for visual recognition. *Neural Netw.* **2016**, 1. Available online: Http://cs231n.github.io (accessed on 1 July 2022).
47. Yi, D.; Lei, Z.; Liao, S.; Li, S.Z. Learning face representation from scratch. *arXiv* **2014**, arXiv:1411.7923.
48. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [CrossRef]
49. Huang, G.B.; Ramesh, M.; Berg, T.; Learned-Miller, E. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*; University of Massachusetts: Amherst, MA, USA, 2007; Technical Report 07-49, October 2007.
50. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.