Metric Learning for Face Recognition in Unconstrained Environments: Enhancements in ResNet Architecture, Application of Thin Plate Spline Interpolation, and Comparative Analysis of Multiple Loss Functions

Xianzhe Fan
Tsinghua University
fxz21@mails.tsinghua.edu.cn

Zhaoxu Chen
Tsinghua University
Joeychen89@gmail.com

Junlin Luo
Tsinghua University
carrotmarkluo@gmail.com

Abstract

We present a comprehensive study on face recognition in unconstrained environments. By experimenting with the Labeled Faces in the Wild (LFW) dataset, we aim to address the challenge of accurately identifying whether two facial images represent the same individual, especially in conditions with unaligned faces and diverse backgrounds. Our contributions include modifications to the ResNet18 architecture, exploration of TPS interpolation for face alignment, and an indepth evaluation of loss functions like Contrastive Loss, Triplet Loss, and N-pair & Angular Loss in improving face recognition accuracy. Finally, we employ the stacking method from ensemble learning to enhance the overall predictive performance.

1. Introduction

Face recognition, a pivotal domain in computer vision and biometrics, aims to identify or verify a person's identity using their face. Historically, the focus has been on controlled scenarios with limited variability in facial orientation, expression, and illumination. However, real-world applications often encounter unconstrained environments where these factors vary widely, posing significant challenges to existing face recognition technologies. In unconstrained settings, face recognition systems must contend with diverse backgrounds, varying lighting conditions, unaligned faces, and a range of facial expressions and occlusions.

Deep Convolutional Neural Networks (CNNs) [9] have emerged as a powerful tool for image recognition tasks, including face recognition, due to their ability to learn complex, hierarchical feature representations from data. ResNet [6], introduced by He et al., is particularly notable for its skip blocks that address the degradation problem, enabling the training of networks with a much greater number of layers.

The current research landscape in face recognition is diverse [5, 12, 10, 15], encompassing improvements in network architectures and data pre-processing techniques, such as face alignment, and the development of more effective loss functions for training. These advancements are critical for improving the performance of face recognition systems in unconstrained environments, where traditional methods often fall short. Our study contributes to this work by enhancing the ResNet architecture, employing Thin Plate Spline (TPS) [4] interpolation for face alignment, and comparing various loss functions. These elements are crucial for improving the accuracy and robustness of face recognition systems in challenging real-world conditions.

Moreover, recognizing the potential of ensemble learning in boosting the accuracy of complex predictive models, our study also incorporates a stacking approach [16]. Ensemble learning, particularly stacking, combines the strengths of multiple models to achieve greater predictive performance.

2. Related Work

Initial efforts in the face recognition field relied heavily on geometric features and template-matching techniques [3]. Though effective in controlled settings, these methods were limited in handling the variability and complexity of unconstrained environments. With the emergence of deep learning, the focus shifted towards developing algorithms that could learn to recognize faces from data rather than relying on handcrafted features.

The introduction of CNNs marked a turning point in face recognition. Pioneering works like LeCun et al.'s introduction of CNNs for digit recognition [5] laid the groundwork for their application in more complex image recognition tasks, including face recognition. The development of deep CNN architectures, such as AlexNet [9], VGGNet [13], and GoogleNet [14], further advanced the field, demonstrating the effectiveness of deep networks in learning hierarchical feature representations.

ResNet, a deeper and more sophisticated architecture, addressed the limitations of previous CNNs by introducing residual learning. This innovation allowed for the training of networks with a significantly higher number of layers without degradation in performance, a critical advancement for complex tasks like face recognition. ResNet's ability to learn rich, deep representations made it particularly suited for face recognition tasks, where nuances in facial features are paramount.

Alongside advancements in network architecture, there has been a growing interest in the pre-processing of facial images, particularly in alignment. Face alignment techniques, such as TPS interpolation [2], have become essential in managing the variations in facial geometry that occur in unconstrained environments. These techniques adjust faces to a canonical pose, ensuring that the neural network receives well-aligned and standardized inputs.

Loss functions in neural networks are another area of significant research. Especially in metric learning for face recognition, traditional loss functions such as cross-entropy, have been replaced by Contrastive Loss [5], Triplet Loss [12], N-pair & Angular Loss (NL&AL) [15] and so on. These loss functions are designed to enhance the discriminative power of the network by encouraging it to learn embeddings that effectively differentiate between different individuals.

Our study builds upon these developments, focusing on enhancing the ResNet architecture for face recognition, applying TPS interpolation for face alignment, and conducting a comparative analysis of various loss functions. By integrating these advancements, we aim to improve the face recognition performance in unconstrained environments.

Recent advancements in ensemble learning, specifically stacking methods [16], have also gained attention in image recognition tasks. Stacking involves training a meta-model to combine the outputs of several base models, enhancing overall performance. This approach has been effective in scenarios where a single model's perspective is insufficient, making it a valuable addition to our study of face recognition.

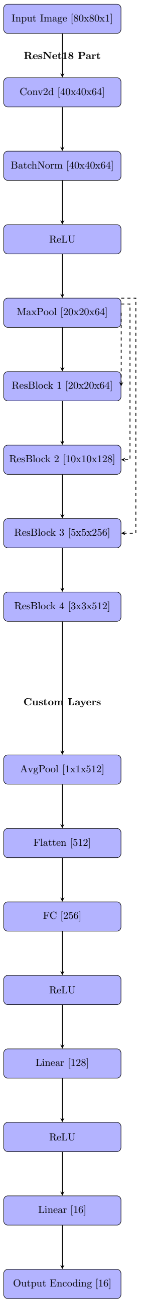


Figure 1: Model architecture.

3. Approach

3.1. ResNet18 Architecture Modification

We adapted the ResNet18 architecture for single-channel grayscale images and replaced its output layer with custom linear layers to produce 16-dimensional embedding vectors for face recognition (Figure 1). This model leverages the robust feature extraction capabilities of the ResNet architecture, and extends it to generate compact yet informative facial representations and ensure distinguishability between different individuals.

Specifically, we modified the initial convolutional layer of ResNet18, altering the input channel count to match single-channel grayscale inputs. Following the ResNet feature extraction, a Batch Normalization layer is introduced. This layer aids in stabilizing the learning process and normalizing the feature representation. The high-dimensional features obtained from ResNet are further processed through a sequence of layers, including ReLU activation and linear transformations. The final output is a 16-dimensional encoding vector for each face.

3.2. Thin Plate Spline (TPS) Interpolation

We utilized TPS, a spline-based data interpolation and smoothing technique, for face alignment. This approach helps manage complex facial image variations while maintaining the integrity of facial features.

3.2.1 Face Cropping and Alignment Based on Detection Boxes and Key Points

To reduce overfitting and improve computational efficiency, we resized the images in our training set from 250x250 pixels to 80x80 pixels and converted them to grayscale. Additionally, face cropping and alignment are crucial for enhancing recognition accuracy. As shown in Figure 3(a), we selected a standard face image with a resolution of 80×80. We utilized the dlib library [1] to determine the coordinates of 68 key points for the training samples and the standard face. In cases where multiple faces are detected, we focus only on the first 68 key points corresponding to the central face. By employing Thin Plate Spline (TPS) interpolation, we achieved an affine transformation between the key points of the standard face and those of the faces to be processed, aligning them accordingly.

3.2.2 TPS Algorithm

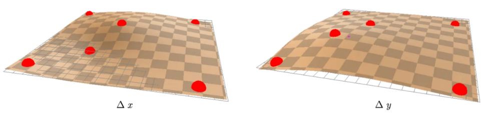


Figure 2: TPS algorithm.

The basic problem for the TPS algorithm is: Given a series of observation points $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$, with the generating function $Y = f(X)$ unknown, how to fit an expression approximating the actual generating function using these observations. This expression is known as the interpolation function.

The TPS interpolation function, for $D = 2$, is as follows:

$$\phi(x) = c + a^T x + \omega^T s(x), \quad c \in \mathbb{R}^{1\times 1}, a \in \mathbb{R}^{D\times 1}, \omega \in \mathbb{R}^{N\times 1} \tag{1}$$

where

$$s(x) = (\sigma(x - x_1), \sigma(x - x_2), \cdots, \sigma(x - x_n))^T \tag{2}$$

$$\sigma(x) = ||x||_x^2 \log ||x||_x^2 \tag{3}$$

The output function $\phi(x)$ has N+D+1 parameters. Each observation point provides a constraint $y_k = \phi(x_k)$. Additional D+1 constraints are added:

$$\sum_{k=1}^{n} \omega_k = 0 \tag{4}$$

$$\sum_{k=1}^{n} \omega_k x_k^1 = 0, \cdots, \sum_{k=1}^{n} \omega_k x_k^D = 0 \tag{5}$$

Let

$$X = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^{\mathrm{D}} \\ x_2^1 & x_2^2 & \cdots & x_2^{\mathrm{D}} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\mathrm{N}}^1 & x_{\mathrm{N}}^2 & \cdots & x_{\mathrm{N}}^{\mathrm{D}} \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{\mathrm{N}} \end{bmatrix}$$

$$S = \begin{bmatrix} \sigma(x_1 - x_1) & \sigma(x_1 - x_2) & \cdots & \sigma(x_1 - x_{\mathrm{N}}) \\ \sigma(x_2 - x_1) & \sigma(x_2 - x_2) & \cdots & \sigma(x_2 - x_{\mathrm{N}}) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma(x_{\mathrm{N}} - x_1) & \sigma(x_{\mathrm{N}} - x_2) & \cdots & \sigma(x_{\mathrm{N}} - x_{\mathrm{N}}) \end{bmatrix} \tag{6}$$

The system of constraint equations is then

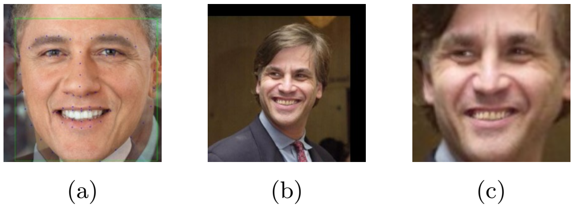
$$\begin{bmatrix} S & 1_N & X \\ 1_N^T & 0 & 0 \\ X^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega \\ c \\ a \end{bmatrix} = \Gamma \begin{bmatrix} \omega \\ c \\ a \end{bmatrix} = \begin{bmatrix} Y \\ 0 \\ 0 \end{bmatrix} \tag{7}$$

When Γ is non-singular, this system of equations has a unique solution. Therefore, the parameter matrix can be obtained as:

$$\begin{bmatrix} \omega \\ c \\ a \end{bmatrix} = \Gamma^{-1} \begin{bmatrix} Y \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

3.3. Loss Functions

We explored three special loss functions: Contrastive Loss, Triplet Loss, and N-pair & Angular Loss, to compute distances between facial feature embeddings.



(a) (b) (c)

Figure 3: (a) Standard face. (b) An image from the training set. (c) Processed image.

3.3.1 Contrastive Loss

Contrastive loss [5] is a loss function for metric learning. It is typically used in unsupervised and semi-supervised learning to ensure that similar points are closer in the feature space, while dissimilar points are farther apart beyond a certain margin.

The mathematical formulation of the contrastive loss is given by:

$$L = \frac{1}{2N} \sum_{i=1}^{N} y_i \cdot D_i^2 + (1 - y_i) \cdot \max(0, m - D_i)^2 \quad (9)$$

where L is the contrastive loss for a batch, N is the number of pairs in the batch, y_i is a binary indicator that equals 1 if the pair is similar and 0 if otherwise, D_i is the Euclidean distance between the features of the i-th pair, and m is a predefined margin.

3.3.2 Triplet Loss

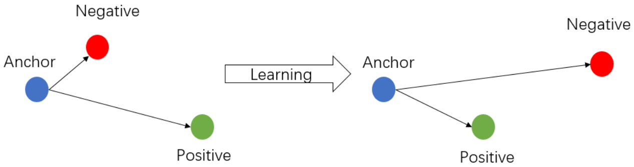


Figure 4: Triplet Loss.

For a given triplet (A, P, N) where A and P are different samples of the same class, and A and N are samples of different classes, Triplet Loss aims to learn a feature space where, in this space, the benchmark sample A of the same category is closer to the positive sample P and farther from the negative sample N. The distance expressions are:

$$d(A, P) = ||f(A) - f(P)||^2 \quad (10)$$

$$d(A, N) = ||f(A) - f(N)||^2 \quad (11)$$

To prevent overfitting, a hyperparameter α is added:

$$||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + \alpha \leq 0 \quad (12)$$

Therefore, the loss function is:

$$l(A, P, N) = \max(||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + \alpha, 0) \quad (13)$$

Using Triplet Loss to compute the loss clusters faces, requiring that the encoding distances of the same person and different people differ by at least a margin of 0.2 (α).

Furthermore, by examining Equations 10 and 11, it becomes evident that the Triplet loss formulation does not encapsulate all three points (A, P, N) collectively within a single equation.

3.3.3 Angular Loss

Angular Loss [10] considers angular relationships as a measure of similarity. For a given triplet (A, P, N):

$$\frac{\partial \lambda_{\text{ang}}(T)}{\partial x_a} = 2(x_a - x_p) - 2\tan^2\alpha(x_a + x_p - 2x_n) \quad (14)$$

$$\frac{\partial \lambda_{\text{ang}}(T)}{\partial x_p} = 2(x_p - x_a) - 2\tan^2\alpha(x_a + x_p - 2x_n) \quad (15)$$

$$\frac{\partial \lambda_{\text{ang}}(T)}{\partial x_n} = 4\tan^2\alpha[(x_a + x_p) - 2x_n] \quad (16)$$

Nevertheless, during training, we encountered an issue where the loss values were excessively small, resulting in inefficient training progress or a complete halt in improvements. Consequently, we address this challenge by incorporating the N-pair & Angular Loss, a proposed improvement outlined in the work by Wang et al. [15]

3.3.4 N-pair Loss & Angular Loss (NL&AL)

N-pair loss reduces the computational burden by employing an efficient batch construction strategy. It constructs batches using only N pairs of examples instead of the (N+1)×N pairs that would be required for the same number of comparisons with triplet loss. The differential formula in Triplet Loss only considers pairwise relations, omitting the third point. In addition, due to significant variations in intra-class distances, using a single global margin in real-world tasks is inappropriate. Therefore, we can use NL&AL [15] to resolve this issue.

The joint of N-pair and angular loss learns more discriminative features, especially with large intra-class variance. Figure 7 shows our model's training accuracy and loss.

3.4. Ensemble Learning through Stacking

We integrate a stacking ensemble method to refine our face recognition framework further. This involves training multiple diverse models on our dataset and then using a meta-learner to learn the optimal way to combine their predictions. Stacking aims to provide a more robust and accurate face recognition system by capturing the strengths and mitigating the weaknesses of individual models. . The image below illustrates the fundamental structure of a Stacking Neural Network. We utilize four neural networks that we previously trained (Network 1,2,4,5 in Table 1). For two given photos, we calculate four distances using the aforementioned networks. These four distances are then fed as inputs into a fully connected neural network, which ultimately outputs whether the two samples represent the same person (label 0) or different individuals (label 1).

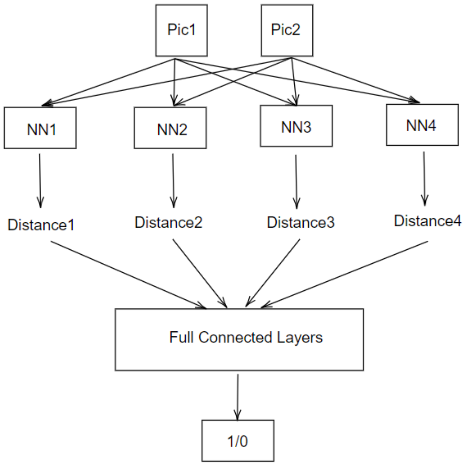


Figure 5: Ensemble learning network structure.

The loss function here is the Binary Cross Entropy Loss. For a single data, it is given by:

$$\text{BCELoss} = -(y \cdot \log(p) + (1-y) \cdot \log(1-p)) \quad (17)$$

Where: y is the actual label (0 or 1). p is the predicted probability (output of the model). log is the natural logarithm.

Since it aggregates the outputs of multiple models, the ensemble model is more robust to noise and variance in the data, leading to more stable predictions. Furthermore, ensembles can reduce the risk of overfitting because different models will likely overfit in different ways.

4. Experiments and Results

Our facial recognition task is designed to determine whether two given facial images belong to the same person. In this task, we utilized the LFW dataset[7]. A characteristic of this dataset is that the target faces are centrally located against complex backgrounds and are not aligned. In section 4.1, we employed cross-validation methods, dividing the training data into training, validation, and test sets (comprising 12448 facial images from 5354 distinct individuals, each represented by one or more images). In section 4.2, we applied our trained models to a discrimination task involving 600 pairs of faces. All these images were sourced from the LFW dataset, and there is no overlap of images between sections 4.1 and 4.2.

In our experiments using the LFW dataset, we evaluated the combined effects of our modified ResNet18 architecture with different loss functions and the effectiveness of TPS interpolation for face alignment. Additionally, we explored the impact of using ImageNet [11] pre-trained ResNet18 model and ensemble learning on training effectiveness.

4.1. Cross-Validation

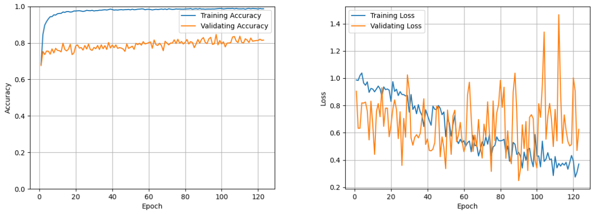
To ensure robustness and prevent overfitting, we use cross-validation, splitting the training data into training, evaluation, and test sets in an 8:1:1 ratio. We randomly select positive and negative samples to form unique triplets during each training epoch, using each sample three times. In our training configuration, we use Adam Optimizer [8], setting the learning rate to 0.0001 and including a weight decay of 0.003.

To address the twin problem, where the model encounters a large number of highly similar yet differently labeled faces during training (due to mislabeling, low image quality, or genuine resemblance), the model may mistakenly identify an unlabeled face of the same person as a different individual in subsequent encounters. The approach of FaceNet [12] is to calculate the closest match based on the local features of different individuals rather than the closest global features. Given that our dataset is smaller, we compute the closest match based on global features. Still, when selecting negative samples, we skip the first 20 images (sorted in ascending order of their encoding distances).

The facial features are represented as tensors, with encodings for the training set grouped by individual and test set pairs stored separately. Our model employs a multi-layer linear model to output these encodings and uses Euclidean or Angular Distance to measure the disparity between facial feature embeddings, applying Contrastive Loss, Triplet Loss, or NL&AL to enhance the discriminative power of the model. The accuracy and loss curves of different methods are presented in Figures 6, 7, 8, 9, 10.

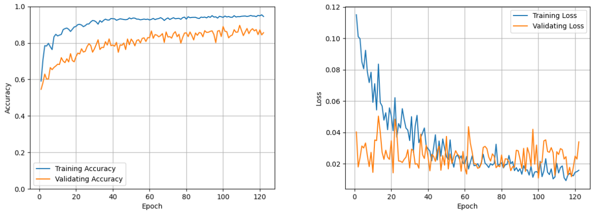
We allow for a 1% fluctuation in accuracy during model iterations to enhance the model's ability to find optimal solutions. If the accuracy of the validation set does not increase after 12 epochs, training is halted to prevent unnecessary computational expense and potential overfitting. The final results of the highest validating accuracy, test accuracy, and epoch counts are shown in Table 1.

From the figures and tables, it is evident that Contrastive Loss performs the worst, while there is no significant difference between the effects of NL&AL and Triplet Loss. The use of ImageNet pretrained models yields slightly better results than training from random weights. After employing TPS for face alignment, there is an improvement in recognition performance.



(a) Accuracy (b) Loss

Figure 6: Network 1 (ResNet + Contrastive Loss).



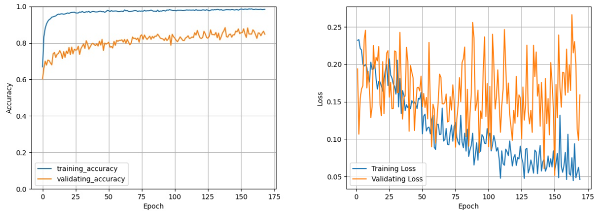
(a) Accuracy (b) Loss

Figure 7: Network 2 (ResNet + NL&AL).

4.2. A Discrimination Task Involving 600 Pairs of Faces

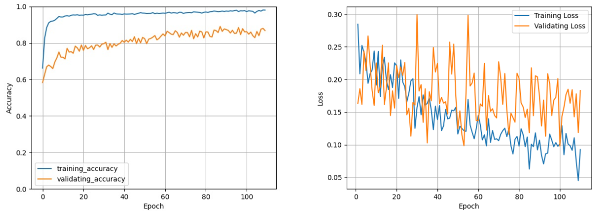
We tested the model trained with cross-validation on 600 pairs of facial images to determine if each pair of faces came from the same person. The accuracy rates of models corresponding to different methods for this task are shown in Table 2.

In ensemble learning, we utilized the four neural networks trained before (Network 1,2,4,5 in Table 1) and got different distances for 600 pairs of faces. We used the distances from 300 pairs of faces to be the training data of our ensemble model (fully connected neural network) and the other 300 pairs to be the testing data. Finally, the accuracy reached 82.67%.



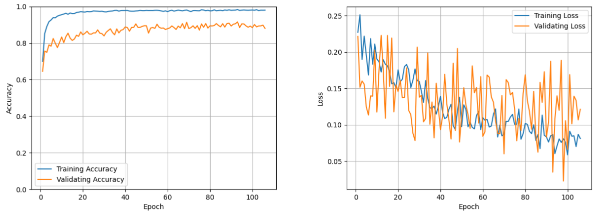
(a) Accuracy (b) Loss

Figure 8: Network 3 (ResNet + Triplet Loss).



(a) Accuracy (b) Loss

Figure 9: Network 4 (ResNet + Triplet Loss + Pretrain).



(a) Accuracy (b) Loss

Figure 10: Network 5 (ResNet + Triplet Loss + TPS).

5. Conclusion

This study demonstrates that targeted modifications to the ResNet architecture, effective face alignment using TPS interpolation, the careful selection of loss functions, and ensemble learning through stacking can significantly enhance face recognition accuracy in unconstrained environments.

Looking to the future, we can further explore the use of 3D modeling and transformations for face alignment, as well as investigate more efficient loss functions. Ad-

	Network 1 Contrastive Loss	Network 2 NL&AL	Network 3 Triplet Loss	Network 4 Triplet Loss + Pretrain	Network 5 Triplet Loss + TPS
Highest Validating Accuracy	84.63%	88.94%	88.40%	89.03%	91.53%
Test Accuracy	83.56%	89.27%	89.19%	88.90%	89.80%
Epoch Counts	122	122	169	110	106

Table 1: Performance comparison in cross-validation of various enhancements to the ResNet architecture.

	Network 1 Contrastive Loss	Network 2 NL&AL	Network 3 Triplet Loss	Network 4 Triplet Loss + Pretrain	Network 5 Triplet Loss + TPS
Test Accuracy	72.50%	72.30%	77.17%	81.00%	80.83%

Table 2: Performance comparison in 600-pairs task of various enhancements to the ResNet architecture.

ditionally, applying these technologies to other metric learning applications, such as multimodal biometric systems and intelligent surveillance systems, and even in virtual and augmented reality environments, may lead to new discoveries and enhance safety and user experience in practical scenarios.

References

[1] dlib: A toolkit for making real world machine learning and data analysis applications. https://pypi.org/project/dlib/. 3

[2] F.L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(6):567–585, 1989. 2

[3] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W. T. Freeman. Best-buddies similarity for robust template matching. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2021–2029, Los Alamitos, CA, USA, jun 2015. IEEE Computer Society. 1

[4] Jean Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In Walter Schempp and Karl Zeller, editors, Constructive Theory of Functions of Several Variables, pages 85–100, Berlin, Heidelberg, 1977. Springer Berlin Heidelberg. 1

[5] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, pages 1735–1742, 2006. 1, 2, 4

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016. 1

[7] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 5

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 5

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. Commun. ACM, 60(6):84–90, may 2017. 1, 2

[10] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6738–6746, 2017. 1, 4

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. International Journal of Computer Vision (IJCV), 115(3):211–252, 2015. 5

[12] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 815–823, 2015. 1, 2, 5

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 2

[14] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. 2

[15] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 2612–2620, 2017. 1, 2, 4

[16] David Wolpert. Stacked generalization. Neural Networks, 5:241–259, 12 1992. 1, 2