Building Face Recognition System with Triplet-based Stacked Variational Denoising Autoencoder

Xuan Tuan Le
People's Security Academy
Hanoi, Vietnam
tuanlx.psa@gmail.com

ABSTRACT

Face recognition is a fundamental and critical topic in computer vision. In this work, a face recognition system based on stacked variational denoising autoencoders with triplet loss is proposed to overcome some existing challenges regard to face variations including poses, illumination, expression and low resolution with less training data. In our proposed system, a stacked variational denoising autoencoder is used to build a deep architecture for extracting salient and latent features from data. Together with that, by using a triplet loss function, we can preserve categorical similarity between faces, then improve the performance of the autoencoders in the clustering task. The proposed system is evaluated in some benchmark face datasets including ORL, Yale, Youtube Faces. Preliminary results demonstrate that the proposed system yields comparable results to other deep convolutional neural networks (CNN) and none-deep CNN based methods.

CCS CONCEPTS

• **Artificial intelligence → Computer vision**.

KEYWORDS

Face Recognition, Autoencoders, Triplet Loss, Denoising Variational Autoencoder

ACM Reference Format:
Xuan Tuan Le. 2019. Building Face Recognition System with Triplet-based Stacked Variational Denoising Autoencoder. In *The Tenth International Symposium on Information and Communication Technology (SoICT 2019), December 4–6, 2019, Hanoi - Ha Long Bay, Vietnam, Viet Nam.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3368926.3369707

1 INTRODUCTION

Face recognition plays an important role in identity authentication and has been widely used in many areas, such as military, public security, and daily life. In every face recognition system, each processing stage is designed to satisfy application requirements including high performance and accuracy. In face recognition, feature extraction and feature classification are considered as the key factors which directly affect the performance and accuracy of a face recognition system. However, these tasks are still challenging issues due to the effects of variations on images of faces including poses, illumination, expressions. Robust feature extraction is the one that can overcome the variation of data and acquire good feature representation. Recently, many deep neural network investigations have obtained fine results in representation learning. Autoencoder is a neural network (NN) which is an unsupervised learning algorithm that can learn efficiently non-linear features from high-dimensional and unlabelled datasets. A denoising autoencoder corrupts the input stochastically by setting some percentage of input elements to zero during the training phase. This is motivated by making embeddings more robust to small irrelevant changes in input. For the feature classification task in face recognition, a clustering algorithm is used. The goal of clustering is to categorize the feature representation of one person into one cluster by using some similarity measures. To improve the clustering performance, a triplet loss function is used with stacked variational denoising autoencoders. The triplet loss for face recognition has been used in [13]. The goal of this function is to make sure two things. The first is to ensure two examples with the same label have their embeddings close together in the embedding space. The second is to ensure that two examples with different labels have their embeddings far away from each other's. In this paper, a face recognition system based on a deep neural network is implemented. The architecture of the deep neural network is designed by triplet based denoising autoencoders which are stacked to get a more effective feature extraction. The deep neural network is evaluated on some public datasets including ORI, Yale, Yale-B with other deep and non-deep CNN based methods.
The rest of this paper is organized as follows: Section II presents some typical methods in the considered field. Section III explains the proposed face recognition system. Finally, Section IV demonstrates the performance and accuracy of the proposed method on the different datasets in comparison with other deep and non-deep CNN based methods.

2 RELATED WORKS

In recent years, along with the development of computer performance, face recognition has achieved remarkable results with various methods used, especially CNN and deep learning-based methods. For example, DeepFace which is originally introduced in [16] trained a CNN for a classification task. In [3], the siamese network is proposed that contains two or more identical subnetworks. Moreover, other typical CNN architectures such as VGG, GoogleNet, VGGFace, ResNet which are introduced in [5, 10, 15] are also considered as state of the art face recognition methods.
In addition to that, a new none CNN based approach called deep autoencoder network is proposed as a training algorithm [6]. The

SoICT 2019, December 4–6, 2019, Hanoi - Ha Long Bay, Vietnam, Viet Nam

ACM ISBN 978-1-4503-7245-9/19/12…$15.00
https://doi.org/10.1145/3368926.3369707

traditional autoencoder is then researched extensively and lead to a bunch of variations of it. Moreover, a progressive autoencoder is stacked and used for face recognition [8]. Zhang et al. [19] proposed Coarse-to-Fine Auto-Encoder Networks (CFAN) as a fine autoencoder for real-time face alignment. Gao et al. [4] used stack supervised autoencoders for face recognition. Even sparse autoencoders with softmax and autoencoder with dropout arer proposed in [8, 12, 14]. Although CNNs have accomplished great success in face recognition, there are several drawbacks to it. One of those is the generalization performance which poses a challenging issue with variations of the face including poses and expressions. Another issue is that most CNNs based methods require vast datasets while autoencoder and it's variations do not need too large ones. On the other hand, autoencoders can investigate unlabeled data and extract more robust features from face images.

In face recognition, the design of the loss function is a challenging task. The loss function is used not only to control the final goal of the optimization of the neural networks but also to affect the efficiency of the training model. CNN models can be trained using different approaches. One of them consists of treating the problem as a classification one. In this approach, each identity in the training set corresponds to a class. After training, the model can be used to recognize faces that are not presented in the training set by using the features of the previous layer as face representation. In the realm of deep learning, these features are commonly referred to as bottleneck features. Another common approach to learning face representation is to directly learn bottleneck features by optimizing a distance metric between pairs of faces, or triplets of faces. Triplet loss is used to help to improve the performance of the clustering task. The functionality of triplet loss in training a neural network is to preserve the similarity of samples that have the same labels and maximize the distance between two samples' embeds that have different labels.

Motivated by the advantages of autoencoders and triplet loss function, in this study, we investigated the performance of stacked variational denoising autoencoders with triplet loss training strategy. Specifically, we used a stacked variational denoising autoencoder to build a deep architecture. We also corrupted face images to deep architecture can learn more robust and latent features from images. To improve the accuracy of the face recognition system, we train the deep architecture with triplet loss. These tactics make our face recognition system more effective in real circumstances.

3 PROPOSED ALGORITHM

3.1 Stacked variational denoising autoencoders

A variational autoencoder (VAE) is composed of two networks. The first one is the encoder which is simply a bunch of layers that are fully connected layers or convolutional layers that are going to take the input and compress it to a smaller representation that has fewer dimensions than the input which is known as a bottleneck. From this bottleneck, it tries to reconstruct the input using full connected layers or convolutional layers. The basic idea behind a variational autoencoder is that instead of mapping an input to a fixed vector, the input is mapped to a distribution. The only difference between the autoencoder and variational autoencoder is the bottleneck vector is replaced with two different vectors, one representing the mean of the distribution and the other representing the standard deviation of the distribution. The VAE loss consists of two parts: the reconstruction loss and the KL-Divergence loss. The reconstruction loss L_{rec} is the negative expected log-likelihood of the observations in x, and the KL-Divergence loss L_{kl} characterizes the distance between the distribution $q(z|x)$ and the unit Gaussian distribution. VAE models are trained by optimizing the sum of the reconstruction loss and the KL-Divergence loss using gradient descent.

$$L_{vae} = L_{kl} + \sum_{i}^{l}(L_{rec}^{l})$$

A stacked variational denoising autoencoder corrupts the input stochastically by setting some percentage of input elements to zero during the training phase. This is motivated by making embeddings more robust to small irrelevant changes in input.

Figure 1 and Figure 2 show how to form a stacked denoising autoencoder. Firstly, a traditional denoising variational autoencoder is trained using input data and the learned data is acquired. Then, the learned data from the previous layer is used as an input for the next layer and this continues until the training is completed. Once all the hidden layers are trained, the backpropagation algorithm is used to minimize the cost function and weights are updated with the training set to achieve fine tuning.

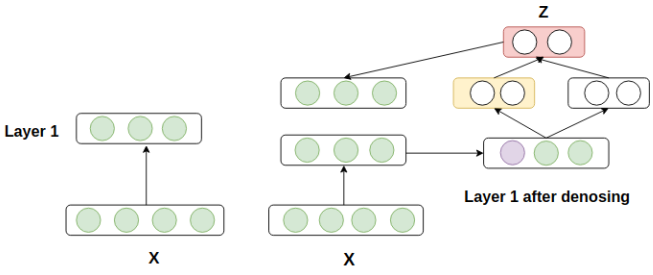

Figure 1: Stacking Denoising Variational Autoencoder.

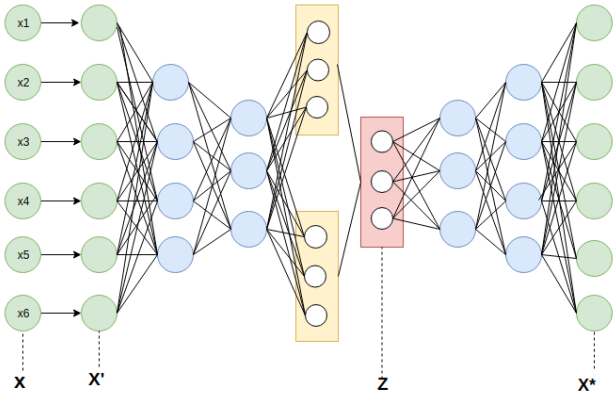

Figure 2: Stacked Denoising Variational Autoencoder.

3.2 Triplet-based Stacked Variational Denoising Autoencoder (TSVDA)

As mentioned in [13], rather than calculating loss based on two examples, triplet loss involves an anchor example and one positive or matching example (same class) and one negative or non-matching example (different class). The loss function penalizes the model such that the distance between the matching examples is reduced and the distance between the non-matching examples is increased. TSVDA is illustrated in Figure 3. In each iteration of training, the input triplet (x, x_p, x_n) is sampled from the training set in such a way that the anchor x is more similar to the positive (x_p) than the negative (x_n). Then the triplet of three images are fed into encoder network simultaneously to get their mean latent embedding $f(x)$, $(f(x_p)$ and $f(x_n))$. We then define a loss function $(L_{triplet})$ over triplets to model the similarity structure over the images. We use triplet loss in the same way as the one described in Wang et al [18]. The triplet loss can be expressed as:

$$Ł_{triplet}(x_a, x_p, x_n) = max\{0, D(x_a, x_p) - D(x_a, x_n) + m\}$$

where $D(x_i, x_j) = ||f(x_i) - f(x_j)||^2$ is the Euclidean distance between the mean latent vector of images x_i and x_j. Here m is a hyper-parameter that controls the distance margin in the latent embedding. This triplet loss function will produce a non-zero penalty of $D(x_a, x_p)D(x_a, x_n) + m$ if the Euclidean distance between x_a and x_n is not more than the Euclidean distance between x_a and x_p plus margin m in the latent space.

There are various sampling strategies are proposed such as random sampling, hard mining and semi-hard mining for training triplet loss based deep metric learning models. However, in our case, such approaches would alter the real distribution of the data and would negatively affect the training of VAE in general. Thus, in our work, we used random sampling for constructing training triplets. We performed this by first randomly sampling an anchor and a positive image from a class and then randomly sampling a negative image from a different class.

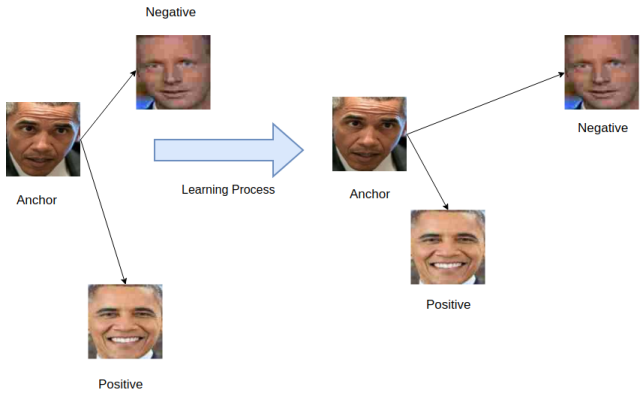

Figure 3: Learning triplet loss.

3.3 TSVDA network architecture and loss function

In this part, we apply the same network structure as proposed in [7], i.e. both encoder and decoder networks are based on deep CNN like AlexNet [9] and VGGNet [10] but we choose slightly different parameters' values. We also construct 4 convolutional layers in the encoder network but choose 3x3 size for kernels. To gain spatial downsampling instead of using deterministic spatial functions such as max-pooling, we utilize a fixed size stride of 2x2. Each convolutional layer is followed by a batch normalization layer and a LeakyReLU activation layer. Then two fully-connected output layers (for mean and variance) are added to the encoder and will be used to compute the KL-Divergence loss and sample latent variable z. For the decoder, we use 4 convolutional layers with 4x4 kernels and set stride to 2, and replace standard zero-padding with replication padding, i.e., feature map of input is padded with the replication of the input boundary. For upsampling, we use the nearest neighbor method by a scale of 2 instead of fractionally-strided convolutions used by other works. We also use batch normalization to help stabilize training and use LeakyReLU as the activation function. The details of autoencoder architecture are shown in Figure 4.

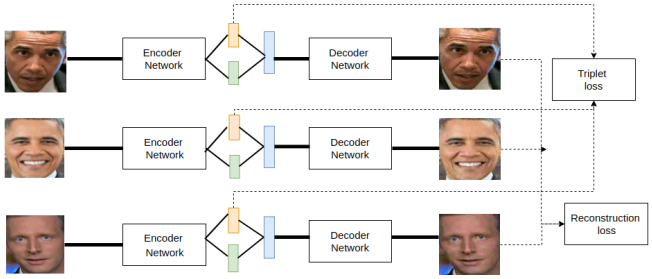

Figure 4: Triplet loss based variational autoencoder.

The loss function is used in TSVDA is:

$$L_{total} = \alpha L_{kl} + \beta \sum_{i}^{l} (L_{rec}^{l}) + \gamma L_{triplet}$$

where α, β and γ are weighting parameters for KL-Divergence, image reconstruction and triplet loss.

4 EXPERIMENTAL RESULTS

In this part, we evaluate the recognition accuracy of our proposed method and compare it with other CNN or non-CNN based methods on benchmark datasets in the same field.

4.1 Training details

Our proposed model is trained on CelebFaces Attributes Dataset (CelebA) [11] which is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations. The images in this dataset cover large pose variations and background clutter. The CelebA has large diversities, large quantities, and rich annotations, including:

- 10,177 number of identities.
- 202,599 number of face images
- 5 landmark locations, 40 binary attributes annotations per image

We build the training dataset by cropping and scaling the aligned images to 64 x 64 pixels. We train our model with a batch size of

64 for 5 epochs over the training dataset and use Adam method for optimization [2] with an initial learning rate of 0.0005, which is decreased by a factor of 0.5 for the following epochs. In this section, we also perform experiments on face images to test our model. We test it on four known face recognition datasets, which are ORL, Yale, Youtube Faces databases. The followings are their details.

ORL: The ORL face database, by the Olivetti Research Laboratory in Cambridge, UK.

- A total of 400 images for 40 individuals.
- There are 10 different images for each individual.
- Contains different expressions, varying lighting, and occlusions (sunglasses).

Yale: The Yale Face Database.

- A total of 165 images for 40 individuals.
- There are 11 different images for each individual.
- Almost frontal, up-right and with a dark background.
- Contains different expressions, varying lighting, and occlusions (sunglasses).

YouTube Faces Database, a database of face videos designed for studying the problem of unconstrained face recognition in videos.

- The data set contains 3,425 videos of 1,595 different people
- All the videos were downloaded from YouTube. An average of 2.15 videos are available for each subject
- Almost frontal, with unconstrained backgrounds.
- The shortest clip duration is 48 frames, the longest clip is 6,070 frames, and the average length of a video clip is 181.3 frames.

4.2 Parameters setting

Face detection is to determine the presence of the face in a given image and returns the coordinates of a face when it is detected. In the proposed system, as a pre-processing step for feature extraction and face recognition, integrated face detectors of Viola and Jones, Haar Feature-based Cascade Classifiers [17] is applied to detect and locate a bounding box around the face.

Face pose, position, and illumination play an essential role in distinguishing between individuals. The goal of normalization is to minimize the difference of pose, position, and illumination for the same individual including the impact of non-facial details like background and clothing.

In this study, the first step of normalization was done by face rotating, scaling, warping and cropping. The center points of the eyes were obtained from detected landmarks in the previous step, the line and distance between eyes centers helped to determine the scale and the angle of rotation which was used to make an affine warping. After warping, the face was cropped so that the eyes were centered in all faces. The cropping area was determined by the left eye center, while the margin around the eye is a parameter that usually takes a range between 0.1 and 0.3. The width and the height of the cropping face are the parameters that can be determined manually. In experiments, 0.3 was chosen for margin and [64 x 64] pixels were selected for [width x height] respectively.

For denoising the variational autoencoder, instead of using the input x in the autoencoder, a partially corrupted copy of input ($\widetilde{x}$) is used to obtain a good representation. However, training still tries to reconstruct the original input. The corruption levels for denoising were 0.1 and 0.5. In here, we chose corruption of 0.3 according to practical experiments.

4.3 Computional results

Before diving into the computional results, we present some the definitions of Precision, Recall and F1 score.

$$\text{Precision} = \frac{TruePositive}{TruePostive + FalsePosiitive}$$

Precision talks about how precise or accurate your model is, that means out of those predicted positive, how many of them are actual positive. Precision is a good measure to determine when the costs of False Positive is high.

$$\text{Recall} = \frac{TruePositive}{TruePostive + FalseNegative}$$

Recall actually calculates how many of the actual positives our model capture through labeling it as Positive (True Positive). Applying the same understanding, we know that Recall shall be the model metric we use to select our best model when there is a high cost associated with False Negative.

$$\text{F1 Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

F1 Score is needed when we want to seek a balance between Precision and Recall determine, when the costs of False Positive is high. Table 1 shows performance metrics of the proposed method on the Yale, ORL, Youtube Faces using SVM classifer.

Table 1: Performance metrics using SVM classifier

Dataset	Accuracy.%	Precision	Recall	F_1 score
Yale	98.37	0.9724	0.9838	0.9757
ORL	98.25	0.9826	0.9778	0.9810
Youtube Faces	93.24	0.9278	0.9325	0.9285

To evaluate the proposed TSVDA, we compared the results to some recent methods based on deep learning and non-deep learning methods. These methods which are presented in Table 2 include the deep sparse autoencoder by Zhang et al. [19] symbolized with SAE, spatial domain sparse representation with autoencoders - SDSRA [1]. The non-deep learning methods listed for comparison include systolic architecture based on principal component neural network symbolized - PCNN, principal component analysis - PCA, Gabor filters based method. These results show that the proposed TSVDA outperforms the other non-deep learning methods. We also conduct experiments on the Youtube Faces dataset to compare with FaceNet which is considered as state of the art method in this field. Despite being trained on a smaller dataset, our proposed method still gained an accuracy metric higher than 90 percent and reach 93.24 % after tuning.

Table 2: Comparision results

Method	Dataset	Accuracy.%
SAE	Yale	94.67
SDSRA	ORL	93.99
Facenet	Youtube Faces	95.18
SDSRA	ORL	93.99
VGG Face	Youtube Faces	91.60
PCNN	Yale	85.00
PCA	ORL	96.50
Gabor	Yale	93.33
TSVDA	Yale	**98.87**
TSVDA	Youtube Face	**93.24**
TSVDA	ORL	**98.70**

4.4 Discussion

In this study, we implemented the above-metioned TSVDA instead of using deep learning CNN. CNN is designed to handle supervised learning which depends on labeled data and it usually requires large data for training. For instance, DeepFace was trained with 4 million images and FaceNet was trained with 200 million images. It can be observed that the proposed method achieves results comparable to state-of-the-art methods such as Facenet while using much less data for training.

On the other hand, to extract more robust and discriminative features from variances of face images, we add a denoising process. That makes our proposed model can be generalized and overcome the overfitting problem and improve performance efficiency. Using triplet loss and composing the triplet loss with the VAE loss make the accuracy of our proposed model higher than the other compared ones. By using the advantages of triplet loss and a stacked variational denoising autoencoder, the TSVDA could be considered as a robust classifier due to powerful feature learning ability implemented.

4.5 Conclusion and future works

In this paper, a deep-stacked denoising variational autoencoder based on a triplet training strategy is proposed. The algorithm extracts robust and hidden feature by supervised learning through optimizing the triplet loss and the VAE loss. By using variational autoencoders, we can stack on the top of each other to form an effective feature extraction method while maintaining its simplicity. To generalize the model, we input the denoised face image before putting it into the network. Many aspects such as denoising, sparsity, weights initialization, overfitting problem, and more details are still the subject of research and development area to reach more robust results.

REFERENCES

[1] Suparna Biswas, Jaya Sil, and Santi P Maity. 2018. On prediction error compressive sensing image reconstruction for face recognition. *Computers & Electrical Engineering* 70 (2018), 722–735.

[2] Jason Brownlee. 2017. Gentle introduction to the adam optimization algorithm for deep learning. *Machine Learning Mastery* (2017).

[3] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. 2005. Learning a similarity metric discriminatively, with application to face verification. In *CVPR (1)*. 539–546.

[4] Shenghua Gao, Yuting Zhang, Kui Jia, Jiwen Lu, and Yingying Zhang. 2015. Single sample face recognition via learning deep supervised autoencoders. *IEEE Transactions on Information Forensics and Security* 10, 10 (2015), 2108–2118.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[6] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

[7] Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. 2017. Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1133–1141.

[8] Meina Kan, Shiguang Shan, Hong Chang, and Xilin Chen. 2014. Stacked progressive auto-encoders (spae) for face recognition across poses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1883–1890.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[10] Shuying Liu and Weihong Deng. 2015. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian conference on pattern recognition (ACPR)*. IEEE, 730–734.

[11] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[12] Mudassar Raza, Chen Zonghai, Saeed Ur Rehman, and Jamal Hussain Shah. 2017. Pedestrian classification by using stacked sparse autoencoders. In *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 37–42.

[13] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Stacked denoising auto encoder and dropout together to prevent overfitting in deep neural network. *Journal of Machine Learning Research* 15 (2014), 1929–1058.

[15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[16] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1701–1708.

[17] Paul Viola, Michael Jones, et al. 2001. Rapid object detection using a boosted cascade of simple features. *CVPR (1)* 1, 511-518 (2001), 3.

[18] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1386–1393.

[19] Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. 2014. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *European conference on computer vision*. Springer, 1–16.