Data-specific Adaptive Threshold for Face Recognition and Authentication

Hsin-Rung Chou, Jia-Hong Lee, Yi-Ming Chan, and Chu-Song Chen
Institute of Information Science, Academia Sinica, Taipei, Taiwan
MOST Joint Research Center for AI Technology and All Vista Healthcare
$\{sherry18030, honghenry.lee, yiming, song\}@iis.sinica.edu.tw$

Abstract

Many face recognition systems boost the performance using deep learning models, but only a few researches go into the mechanisms for dealing with online registration. Although we can obtain discriminative facial features through the state-of-the-art deep model training, how to decide the best threshold for practical use remains a challenge. We develop a technique of adaptive threshold mechanism to improve the recognition accuracy. We also design a face recognition system along with the registering procedure to handle online registration. Furthermore, we introduce a new evaluation protocol to better evaluate the performance of an algorithm for real-world scenarios. Under our proposed protocol, our method can achieve a 22% accuracy improvement on the LFW dataset.

1. Introduction

Deep Convolutional Neural Networks (CNNs) have achieved great success for face recognition. Especially after the unified framework proposed by Schroff *et al.* [9]. In most works [1, 7, 9], researchers use a fixed threshold for the face verification tasks. The threshold is usually the optimal value that can separate different identities of the testing data. However, we argue that this method is deficient for the real-world scenarios, because the optimal threshold is usually case-specific, *i.e.* the best thresholds for different data sets are often different. As we do not have the testing data in practical applications, the optimal threshold is hard to find or even unobtainable. Furthermore, The content of a face database would change frequently, which also raise the need for threshold value tuning.

We propose the technique of feature-specific adaptive thresholding to improve the recognition accuracy. The adaptive threshold serves for two purposes: it performs the task of face verification, and it acts as a gatekeeper of the database. We also design a system to simulate the real-world scenario, which consists of a deep CNN and a database; we use the deep CNN to extract the embedded feature vector of a facial image and store it in the database along with the threshold and its identity.

To have a fair comparison with the results of using a fixed threshold, we introduce a new evaluation protocol that has the same registration flow as the proposed system. The experimental results show that our method outperforms the traditional one. It strengthens the robustness of the threshold and makes the selection process more tractable.

2. Related Work

Recently, deep CNNs play an important role in the research of face recognition. These works attempt to obtain a better deep learning model by either modifying the objective functions or redesigning the network structures. Their general goal is to utilize the deep learning model to produce discriminative facial features. For example, FaceNet [9] proposes triplet loss; NormFace [13] optimizes cosine similarity directly; A-Softmax loss [6] introduces the angular margin into the objective and AM-Softmax loss [12] improves A-Softmax to stabilize the training; VGGFace[7] combines VGG-net [10] with triplet loss and achieves a great success on the LFW benchmark; Recently, MobileFaceNets [1] designs a lightweight face recognition model with Global Depthwise Convolution.

Nevertheless, it is still challenging to choose the optimal threshold for differentiating different identities. In the benchmark of LFW [4], a canonical list of face verification pairs is provided and researchers are asked to evaluate the algorithm via 10-fold cross validation (CV). The best threshold for L_2-distance is then decided accordingly. The CV-based methods are sufficient for comparing the experimental results of different methods, but the thresholds selected usually do not suffice for practical applications. In this paper, instead of applying a fixed threshold to the entire database, we introduce an adaptive thresholding method that assigns a suitable threshold per each registered face in the database, and show that the approach performs more favorably on face recognition and authentication.

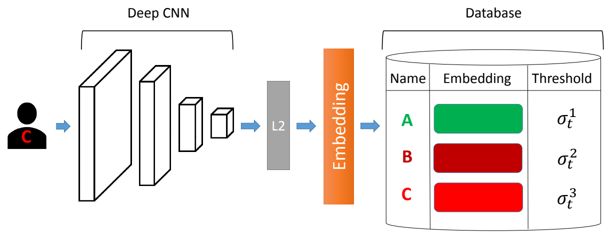

Figure 1. System Structure. It consists of a deep CNN with a L_2 normalization layer, and a database for storing feature embeddings.

3. Methodology

We have two operations in our system: registration and recognition. In the operation of registration, a feature vector (or embedding) is extracted form an input face image by using a deep network. We assume that one face image is registered on the system at a time. The face could belong either to some person already registered on the system, or a new person not registered before. We assign a threshold to the registered face during each registration, and the thresholds of the other registered faces will be modified accordingly.

For recognition, given a query image, we extract its feature embedding and compute the similarity scores between it and all of the other stored embeddings. Then we use the similarity scores to determine the identity of the query image. The system structure is illustrated in Figure 1.

In the following sections, we first review the procedure of extracting the facial features by using state-of-the-art face detection and recognition methods; then we describe the details of registering and recognizing an image with adaptive threshold in sections 3.2 and 3.3.

3.1. Deep Convolutional Neural Network

Given a query image, we first utilize the Multi-task Cascaded Convolutional Networks (MTCNN)[14] to detect and align the face. Next, we utilize a well-trained face recognition model trained on Inception-ResNet-v1[11] with a L_2 normalization layer as proposed by FaceNet[9]. In the inference phase, we extract the output of the L_2 normalization layer as the unified facial feature embedding. For the following section, we use *embedding* as a shorthand for the extracted facial features as introduced by FaceNet. As the embeddings are L_2-normalized, we use the inner product between two embeddings to compute their similarity.

3.2. Registration with Adaptive Threshold

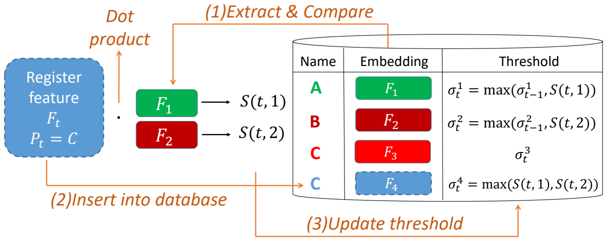

Figure 2. Registration Flowchart. F_t is the embedding of a registering image with identity C. (1) Compute the similarity scores of F_t and all embeddings other than C. (2) Store F_t and its name in the database. (3) Update the thresholds of all the embeddings accordingly.

Given a sequence of face images $\mathbf{I} = \{I_1 \cdots I_t \cdots I_T\}$, the identity labels $\mathbf{P} = \{P_1 \cdots P_t \cdots P_T\}$ and the embeddings $\mathbf{F} = \{F_1 \cdots F_t \cdots F_T\}$ extracted by the deep CNN, we register the embeddings into the database one-by-one as illustrated in Figure 2. At each registration t, we insert the feature embedding F_t and its identity P_t into the database; then we update the threshold σ_t^τ accordingly. σ_t^τ denotes the threshold of registered feature embedding F_τ when F_t is registering into the database ($\tau = 1, \cdots, t$).

We assign a different threshold for each facial embedding in the database. For threshold σ_t^τ (associated with the τ-th image at the t-th time), we first compute the similarity score between the embeddings F_τ and F_v in the database ($v = 1 \cdots t$):

$$S(\tau, v) = F_\tau \cdot F_v \tag{1}$$

Then we compute σ_t^τ as the maximum value among all facial embeddings not belonging to the same person at the current time:

$$\sigma_t^\tau = \max(S(\tau, v)),\ v = 1 \cdots t,\ \text{where } P_\tau \neq P_v \tag{2}$$

For the implementation of updating the thresholds, as the face images are registered one at a time, we can take advantage of the recursion of σ_{t-1}^τ and σ_t^τ for an efficient computation during registration.

3.3. Recognition and authentication

Given a query facial image I_λ without its identity label, we first extract the embedding F_λ by the deep CNN. Then we compute the similarity scores with all the embeddings that are already stored in the database as shown in Figure 3. We extract the one which has the highest similarity score with F_λ and denote its index as u:

$$u = \arg\max_v(S(\lambda, v)),\ \text{for } v = 1 \cdots t \tag{3}$$

Once the most similar embedding F_u is found, the system compare the associated threshold σ_t^u with the similarity score $S(\lambda, u)$. If $S(\lambda, u) \geq \sigma_t^u$, then the image I_λ will be

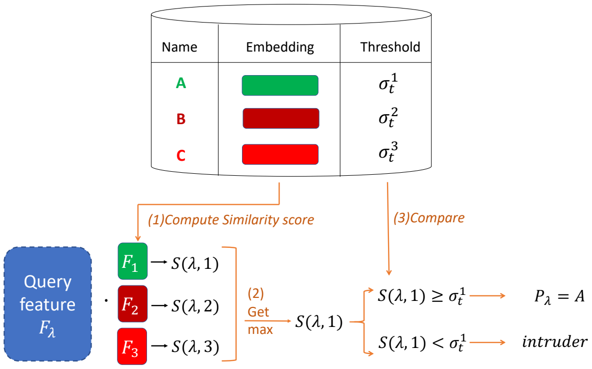

Figure 3. Recognizing Flowchart. F_λ is the embedding of a query image I_λ. (1) Compute the similarity score of F_λ with all embeddings. (2) Get the maximal similarity score. (3) Compare the score with the stored threshold to determine whether the query is an intruder or a registered identity.

classified as identity P_u; else it dissociates from any registered identities, in this case, we call it an *intruder* and reject the authentication request:

$$P_\lambda = \begin{cases} P_u, \text{ if } S(\lambda, u) \geq \sigma_t^u \\ intruder, \text{ if } S(\lambda, u) < \sigma_t^u \end{cases} \tag{4}$$

4. Evaluation and Experiment

Current open-set identification protocol such as [5] evaluates the algorithm under determinate probe (or query) and gallery (or database) sets. This setting standardizes the evaluation and makes the result measurable, but it does not consider the intruders nor changes in the gallery, which often occurs in industrial applications.

We introduce *timeline* in our evaluation protocol. At t-th time, we present an image to the system and check the correctness; we move the face image from the probe to the gallery set one at a time to simulate the real registration rocess and the changes in the gallery.

4.1. Evaluation protocol for singly registering and recognizing

In our evaluation protocol, we have T images in the testing data, the gallery begins with an empty set. When a probe feature embedding F_t is presented to the system, it will be used to calculate the similarity scores with all the gallery feature embeddings $F_\tau, \tau = 1 \ldots t$. The gallery feature embedding that gets the highest similarity score with F_t is denoted as F_u. The threshold function is denoted as $\Phi(t, u)$. For fixed threshold, $\Phi(t, u)$ always returns a constant value; for adaptive threshold, $\Phi(t, u) = \sigma_t^u$. We use 10-fold CV to compute the fixed threshold can compare the results with those that obtained using our adative thresholding method.

When F_t is presented to the system, if $S(t, u) \geq \Phi(t, u)$, and $P_t = P_u$, then we define this case as *true accept*:

$$\text{TA}(t) = \{S(t, u) \geq \Phi(t, u), \text{and } P_t = P_u\} \tag{5}$$

If P_t is an identity in the gallery but $S(t, u) < \Phi(t, u)$, then no matter whether $P_t = P_u$, we define these cases as *false reject*:

$$\text{FR}(t) = \{S(t, u) < \Phi(t, u), \text{and } P_t \in \mathbf{P}\} \tag{6}$$

If P_t is not contained in the gallery, then we call it an *intruder*. Thus, we define *false accept* and *true reject* as:

$$\text{FA}(t) = \{S(t, u) \geq \Phi(t, u), \text{and } P_t \notin \mathbf{P}\} \tag{7}$$

$$\text{TR}(t) = \{S(t, u) < \Phi(t, u), \text{and } P_t \notin \mathbf{P}\} \tag{8}$$

For the last possible case, if $S(t, u) \geq \Phi(t, u)$ and P_t is someone in the gallery, but P_t and P_u are different identities, then we define this case as *identification error*:

$$\begin{aligned} \text{IE}(t) = \{S(t, u) \geq \Phi(t, u), \ &\text{where } P_t \neq P_u \\ &, \text{and } P_t \in \mathbf{P}\} \end{aligned} \tag{9}$$

We add F_t to the gallery set iteratively to simulate the registration operation ($t = 1 \cdots T$); we extract F_{t+1} from the probe set and repeat the process until all of the T images are examined. The final accuracy, ACC, is then defined as the average correctness of all the T sessions:

$$\text{ACC} = \frac{\sum_{t=1}^{T} |\text{TA}(t)| + |\text{TR}(t)|}{T} \tag{10}$$

Table 1. Summary of the evaluation protocol

	$S(t, u) \geq \Phi(t, u)$	$S(t, u) < \Phi(t, u)$
$P_t = P_u$	True accept	False reject
$P_t \neq P_u$, $P_t \in \mathbf{P}$	Identification error	False reject
$P_t \neq P_u$, $P_t \notin \mathbf{P}$	False accept	True reject

4.2. Experiments on Facial Datasets

In all our experiments we use the same deep CNN trained on MS-Celeb-1M [3] dataset to extract the feature embedding. We evaluate our algorithm under the aforementioned protocol on three datasets: Labeled Faces in the Wild (LFW) [4], Adience [2] and Color FERET [8]. As some faces in the images are undetectable, we did not use all images in the datasets. The statistics of the images used in our experiments are show in Table 2. For each experiment,

Table 2. The number of aligned images, identities, and number of samples per identity (with the standard deviation) in facial datasets.

	#images	#classes	#images/class
LFW	13,233	5,749	2.3 ± 9.01
Adience	19,339	2,284	8.46 ± 23.13
Color FERET	11,285	994	11.35 ± 8.6

we randomly shuffle all of the images in the dataset, and then register the images one-by-one according to the random order. We perform the experiments 10 times on each dataset and average the accuracy. To fairly compare with the results of using a fixed threshold, we also conduct the fixed-thresholding experiments under our evaluation protocol with the same registration order.

As for the selection of fixed thresholds, following the verification benchmark of LFW, we use 6,000 image pairs randomly generated from the dataset. As done in the protocol of LFW for the verification performance evaluation, 10-fold CV is used and a threshold is selected for each fold-splitting. The 10 thresholds selected are then averaged to yield the fixed threshold used in our experiment. The same procedure is performed for the fixed thresholds selection of the color FERET and Adience datasets too. After that, the fixed thresholds also serve as the initial values ($t = 1$) of our adaptive thresholding procedure.

The accuracy (ACC defined in Eq. 10) is shown in Table 3. In the case of fewer samples per identity like LFW, the adaptive-threshold approach outperforms the fixed-threshold method by around 22.5% in accuracy. In Adience or Color FERET, the adaptive-threshold approach still consistently outperforms the fixed one. The temporary accuracy during each registration is shown in Figure 4. The adaptive-threshold approach converges earlier, showing it's promising in sample limited applications, e.g., small factory security systems.

5. Conclusion

We solve the long-term threshold-selection problem for face recognition and authentication, and also tackle the deficiency of current evaluation protocol. The introduced technique of adaptive thresholding can decide more favorable thresholds. We also design an on-line registration system for real-world scenarios, which can handle varied conditions for practical applications.

References

[1] S. Chen, Y. Liu, X. Gao, and Z. Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *CCBR*, 2018.

Table 3. Final Accuracy

	Adaptive	Fixed / Threshold
LFW	**76.46%**	53.97%/0.3779
Adience	**84.30%**	80.60%/0.2487
Color FERET	**83.79%**	80.72%/0.3968

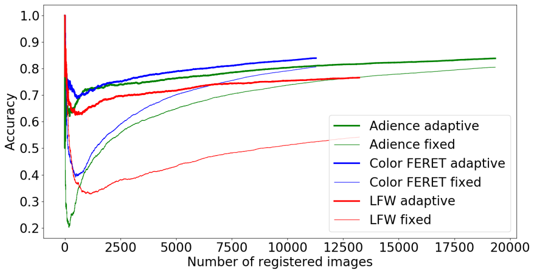

Figure 4. Comparisons of temporary accuracy during each registration.

[2] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *IEEE TIFS*, 2014.

[3] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *ECCV*, 2016.

[4] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *ECCV Workshop*, 2008.

[5] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *CVPR*, 2015.

[6] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *IEEE CVPR*, 2017.

[7] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, 2015.

[8] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE TPAMI*, 2000.

[9] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[10] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[11] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

[12] F. Wang, J. Cheng, W. Liu, and H. Liu. Additive margin softmax for face verification. *IEEE SPL*, 2018.

[13] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: l 2 hypersphere embedding for face verification. In *ACMMM*, 2017.

[14] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE SPL*, 2016.