Face Recognition using FaceNet and MTCNN

Huynh Minh Tri TRI.HUYNH_TK15NBK@HCMUT.EDU.VN
Ho Chi Minh City University of Technology, Vietnam

Contents

List of Figures

List of Tables

Abstract

Deep learning applies multiple layers to learn more abstract and compact patterns that do not require hand-crafted features extraction This developing technology tackles existing method concerns caused by considerable intra-personal differences in face image in the wild, such as poses, illuminations, occlusions, and low resolutions, which cause great obstacles to face-related applications in the previous years. FaceNet uses a deep convolutional network trained to transform face images into an embedding space where the Euclidean distances of images are utilized to determine face similarity. FaceNet is able to approach up to 99.63% accuracy from dataset Labeled Faces in the Wild (LFW) and 95.12% accuracy on the YouTube Faces DB. MTCNN is a preprocessing approach for extracting and aligning human faces from multi-object images and unconstrained environments. MTCNN boosts up the model accuracy while keeping real-time performance by utilizing the intrinsic correlation between face detection and alignment. This report provides a deep learning framework based on MTCNN and FaceNet for face recognition method. Keras FaceNet provided by Hiroki Taniai pre-trained models is also used in testing which achieves up to 100% accuracy on the AT&T, YALE, 5-celebrity-faces-dataset and and 96.06% accuracy on the Extended Yale Face database B (YALE B+).

Keywords: face recognition, face detection, face alignment, deep convolutional network

1. Introduction

Face recognition (FR) has been the most widely used biometric technique for identity authentication, with applications in the military, finance, public security, and daily life. In the early 1990s, the study of FR became popular following the introduction of the historical Eigenface approach [1] which is a method called holistic learning. After a few decades, the FR community has proposed local-feature-based FR and learning-based local descriptors methods.

In general, these traditional methods attempted to recognize the human face using one or two layer representations. The research community worked hard, but FR accuracy only improved modestly as a result of these efforts. Most methods aimed to address one aspect of unconstrained facial changes only, such as lighting, pose, expression, or disguise. Despite repeated efforts over a decade, ”shallow” approaches only raised the LFW benchmark’s accuracy to around 95% [2].

Everything changed when AlexNet [3] won the ImageNet competition by deep learning that learns more abstract and compact patterns via multiple hidden layers without hand-crafted features extraction. So far so good, in 2014, DeepFace [4] and DeepID [5] utilized deep learning to outperform the face recognition capabilities of humans, which no previous method achieved. But before surpassing human performance on FR, people have proposed numerous feature-based FR over the past years are presented in Figure 1, in which the 4 main technical streams are highlighted [6]. The detail is presented in Section II. Inspired by AlexNet, DeepFace, and DeepID models, my report aims to exploit deep-learning-based approaches, which utilize Deep Convolutional Neural Networks (DCNNs) for FR tasks such as face verification, recognition, and clustering.

For accurate face recognition, I train two networks, MTCNN and FaceNet. MTCNN is used for face detection to get accurate face coordinates. Based on the results of the previous step, FaceNet is used for face recognition. FaceNet directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. Once this space has been produced, tasks such as face recognition, verification, and clustering can be easily implemented using standard techniques with FaceNet embeddings as feature vectors. To evaluate their performance, I select 4 public datasets: YALE, AT&T, 5-celebrity-faces-dataset, and YALE B+.

MTCNN is a preprocessing technique used for capturing and aligning the human face from multi-object images and unconstrained environments. MTCNN utilized the inherent correlation between face detection and alignment to boost up the model accuracy while keeping real-time performance.

FaceNet is one of the uses of face recognition based on DCNNs, which transforms face images into an embedding space where the Euclidean distances of images are used for face verification (is this the same person), recognition (who is this person), and clustering (find common people among these faces). In this report, I only focus on the face recognition task in the implementation and results section. FaceNet used two different architectures, namely The Zeiler&Fergus network method (ZFNet) and the latest Inception network method. Stochastic Gradient Descent (SGD) applying backpropagation and AdaGrad standards used to train DCNNs. FaceNet is able to approach up to 99.63% accuracy from dataset Labeled Faces in the Wild (LFW) and 95.12% accuracy on the YouTube Faces DB.

An overview of the rest of the report is as follows: in Section 2, I review the literature in this area; Section 3.1 gives the details about FaceNet and Section 3.2 describes MTCNN; in Section 4, I show the implementation and result. Finally, Section 5 is the conclusion of my report.

2. Related Work

Based on EigenFace in early 1990s, FR stydy became more popular which following by a vast corpus of face verification and recognition works over the past year. In this report, I briefly review some most famous methods that represent 4 major FR milestones: Holistic learning, Local handcraft, Shallow learning, and Deep learning [6]. See Figure 1.

2.1. Holistic learning (the 1900s and 2000s)

Certain distribution assumptions, such as linear sub space [7–9], manifold [10–12], and sparse representation [13–16], are used in holistic approaches to derive the low-dimensional representation. In the 1990s and 2000s, this concept dominated the FR community. However, a well-known issue is that these theoretically plausible holistic techniques fail to address uncontrolled facial changes that is out of their inductive bias.

2.2. Local handcraft (the early 2000s)

Local-feature-based Gabor [17] and LBP [18], as well as their multilevel and high-dimensional variants [2; 19; 20], were inspired by the holistic approach problem and obtained robust performance by exploiting some invariant aspects of local filtering. Handcrafted characteristics, however, lacked distinctiveness and compactness.

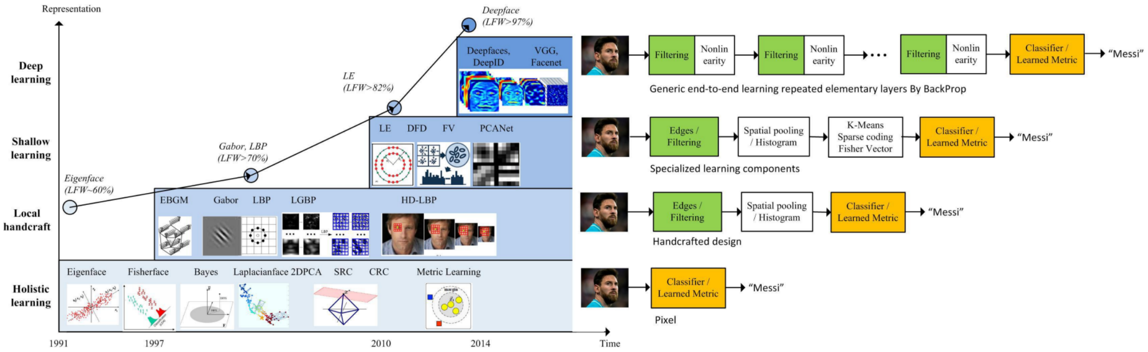


Figure 1: Milestones of face representation for recognition. The holistic approaches dominated the face recognition community in the 1990s. In the early 2000s, handcrafted local descriptors became popular, and the local feature learning approaches were introduced in the late 2000s. In 2014, DeepFace [4] and DeepID [5] achieved a breakthrough on state-of-the-art (SOTA) performance, and research focus has shifted to deep-learning-based approaches. As the representation pipeline becomes deeper and deeper, the LFW(Labeled face in-the-wild) performance steadily improves from around 60% to above 90%, while deep learning boosts the performance to 99.80% in just three years [6]

.

2.3. Shallow learning (the early 2010s)

With learning-based local descriptors [21–23], local filters are learned for better distinctiveness and the encoding codebook is learned for better compactness. Unfortunately, the limitation of these shallow representations against complicated nonlinear face appearance variations is unavoidable.

Generally, these traditional methods attempted to recognize the human face using one or two layer representations, such as filtering responses, histogram of the feature codes, or distribution of the dictionary atoms. The research community worked hard to separately improve the preprocessing, local descriptors, and feature transformation, but these approaches improved FR accuracy slowly. Most methods aimed to address one aspect of unconstrained facial changes only, such as lighting, pose, expression, or disguise. There was no integrated technique to address these unconstrained challenges integrally. For that reason, despite continuous efforts over a decade, "shallow" methods only improved the accuracy of the LFW benchmark to about 95% [2], indicating that "shallow" methods are insufficient to extract stable identity features invariant to real-world changes. Due to this technical insufficiency, FR systems were often reported with unstable performance or failures with innumerable false alarms in real-world applications.

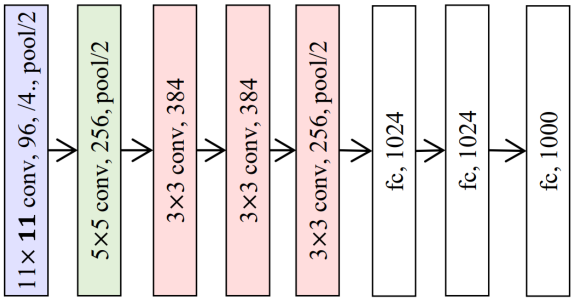


Figure 2: Alexnet model uses CNNs [3]

2.4. Deep learning (2012 - now)

In 2012, AlexNet won the ImageNet competition by a wide margin by employing a technique known as deep learning [3], which is considered one of the most influential papers published in computer vision. Deep learning methods, such as convolutional neural networks (CNNs) designs introduced by [24], use a cascade of multiple layers of processing units for feature extraction and transformation. They learn several representational levels that correspond to various levels of abstraction. Finally, the combination of higher level abstraction is able to represents facial identity with novel stability.

In 2014, DeepFace [4] approaches the SOTA accuracy on the LFW database [25], which is close to human performance on the unconstrained condition for the first time (DeepFace: 97.35% vs. Human: 97.53%). DeepFace trains a 9-layer model on 4 million facial images. Inspired by this work, research focus has shifted to deep-learning-based approaches and the accuracy was remarkably improved to above 99.80% in only 3 years after that. With the rapid development of deep learning, the FR research community generated many breakthrough mainstream network architectures, such as Deepface [4], DeepID series [5; 26–28], VGGFace [29], FaceNet [30], and VGGFace2 [31], as well as other architectures designed for FR. Many of them are successfully applied to many FR real-world applications due to their high performance and consistency.

3. Method

3.1. FaceNet

Facenet proposed by [30] is a method that directly learns a mapping from face images to a compact Euclidean space (vector of 128 numbers) where distances directly correspond to a measure of face similarity. The 128-dimensional vector is called **embedding**. This method uses deep convolutional networks to optimize its embedding, compared to using intermediate bottleneck layers as a test of previous deep learning approaches. When this embedding space is created, face recognition, verification, and clustering utilizing FaceNet embeddings as feature vectors will be simple to solve. FaceNet consists of batch layers as input and deep architecture which is deep CNN followed by L2 normalization, which becomes the result of face embedding Figure 3. In the training process, FaceNet applies

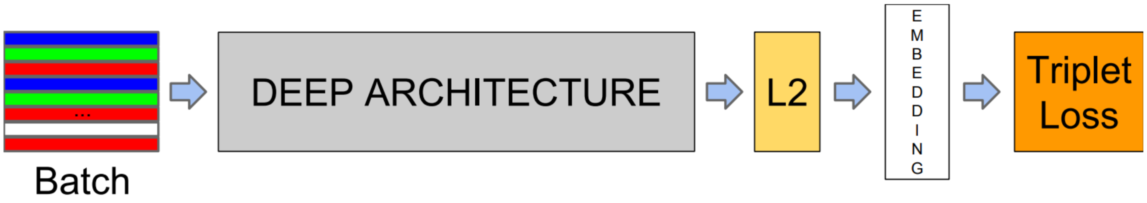


Figure 3: Our network consists of a batch input layer and a deep CNN followed by L2 normalization, which results in the face embedding. This is followed by the triplet loss during training [30]

triplet loss by matching face to face using the online triplet mining method for training more efficiently and doesn't require any offline mining. FaceNet's structural model is shown in Figure 3. FaceNet is made up of batch layers as input and a deep architecture that consists of a deep CNN followed by L2 normalization, which results in face embedding [8]. When the training process was completed, FaceNet was also pursued by the triplet loss, as shown in Figure 4

The triplet loss is motivated in [32] in the context of nearest-neighbor classification. Here, the Euclidean distance between each image with a person $x_i^a(anchor)$ to all other images $x_i^p(positive)$ of the same person is closer than that of any images $x_i^n(negative)$ of other person. This is embedded in formula Equation (1) and visualized in Figure 4. Thus we want,

$$\|x_i^a - x_i^p\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2, \forall \left(x_i^a, x_i^p, x_i^n\right) \in \mathcal{T} \tag{1}$$

where, α represents the margin enforced between positive and negative pairs. $\mathcal{T}$ is the set of all possible triplets in the training set and has cardinality N. The loss that is being minimized is then $L =$

$$\sum_i^N \left[\|f\left(x_i^a\right) - f\left(x_i^p\right)\|_2^2 - \|f\left(x_i^a\right) - f\left(x_i^n\right)\|_2^2 + \alpha\right]_+ \tag{2}$$

where, $f(x)$ is the embedding vector of image x. Training all possible triplets would result in many triplets that have no contribution to training and slow down the learning convergence. It is important to select hard triplets that are active and can therefore contribute to improving the model. The online triplet mining was introduced in [30] which want to select the top k hard triplets on each batch size ($k < |\beta|$) to generate the global loss, that ignores the simple triplets where the distance between anchor and positives are significantly closer than that between the anchor and negatives. Picking the hard losses enforced the learning to be more stable and converged slightly faster at the beginning of training. Furthermore,[30] do not select the hardest negatives due to local minimal problem, they select x_i^n such that:

$$\|f\left(x_i^a\right) - f\left(x_i^p\right)\|_2^2 < \|f\left(x_i^a\right) - f\left(x_i^n\right)\|_2^2 \tag{3}$$

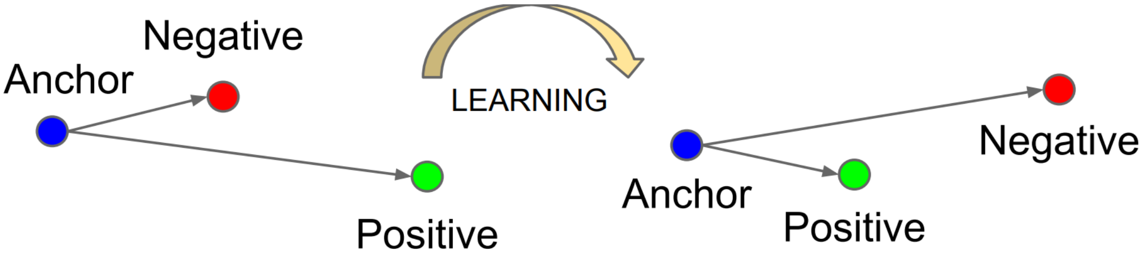


Figure 4: The Triplet Loss minimizes the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity [30]

Those negatives are called semi-hard negatives, since they are further away from the anchor than the positives, but still hard because the squared distance is close to the anchor positive distance. Those negatives lie inside the margin α.

3.1.1. Triplet loss

3.1.2. CNN Architectures

FaceNet uses 2 types of CNNs, namely Zeiler & Fergus architecture and GoogLeNet style Inception model

Zeiler&Fergus architecture [33] The ZFNet architecture is used to visualize the training process of a CNN in order to better comprehend its internal workings. This architecture offered a novel visualization technique that provides insight into the operation of intermediary layers and classifiers. By visualizing the convolutional network layer by layer, ZFNet successfully reduces error rates and adjusting layer hyperparameters such as filter size or stride of AlexNet [3]. The architecture used in the FaceNet research paper by ZFNet is shown in Table 1.

This model has 140 million parameters and 1.6 billion FLOPS (Floating point operations per second) per image.

Inception Model [34] The main idea behind Inception network architecture is that of using multiple filters of different sizes on the same level instead of choosing a filter of size 3×3, 5×5, etc. Combining multiple filters in that way is inside an inception module. The outputs of a inception module are concatenated and sent to the next inception module. Figure 5(a) is a “naive inception module”. It performs convolution on an input, with 3 different sizes of filters (1×1, 3×3 and 5×5), as well as a max pooling layer give outputs in filter concatenation. However, the computation of this very expensive. To make it cheaper, the authors reduce the number of input channels by adding an extra 1×1 convolution before the 3×3 and 5×5 convolutions. The new module is called “inception module with dimension reductions” as shown in Figure 5(b).

GoogLeNet has 9 such inception modules stacked linearly. It is 22 layers deep (27, including the pooling layers). This Inception model architecture used in the FaceNet research

Table 1: The structure of our Zeiler&Fergus [33]

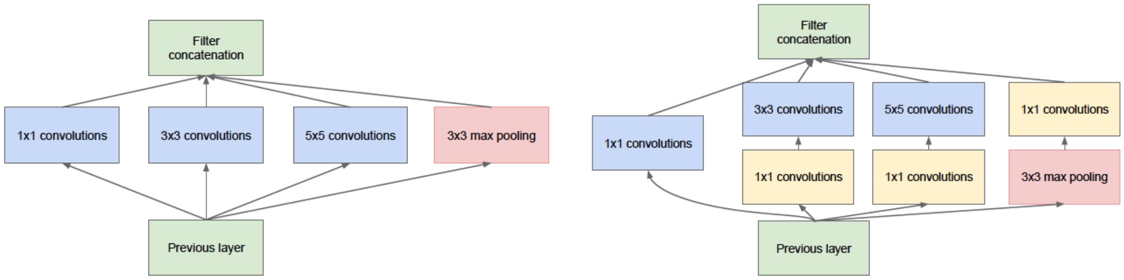
layer	**size-in**	**size-out**	**kernel**	**param**	**FLPS**
conv 1	220x220x3	110x110x64	7x7x3,2	9K	115M
pool1	110x110x64	55x55x64	3x3x64, 2	0	
rnorm1	55x55x64	55x55x64		0	
conv2a	55x55x64	55x55x64	1x1x64, 1	4K	13M
conv2	55x55x64	55x55x192	3x3x64, 1	111K	335M
rnorm2	55x55x192	55x55x192		0	
pool2	55x55x192	28x28x192	3x3x192,2	0	
conv 3a	28x28x192	28x28x192	1x1x192, 1	37K	29M
conv3	28x28x192	28x28x384	3x3x192, 1	664K	521M
pool3	28x28x384	14x14x384	3x3x384,2	0	
conv4a	14x14x384	14x14x384	1x1x384, 1	148K	29M
conv4	14x14x384	14x14x256	3x3x384, 1	885K	173M
conv5a	14x14x256	14x14x256	1x1x256, 1	66K	13M
conv5	14x14x256	14x14x256	3x3x256, 1	590K	116M
conv6a	14x14x256	14x14x256	1x1x256,1	66K	13M
conv6	14x14x256	14x14x256	3x3x256, 1	590K	116M
pool4	14x14x256	7x7x256	3x3x256, 2	0	
concat	7x7x256	7x7x256		0	
fc1	7x7x256	1x32x128	maxout p=2	103M	103M
fc2	1x32x128	1x32x128	maxout p=2	34M	34M
fc7128	1x32x128	1x1x128		524K	0.5M
L2	1x1x128	1x1x128		0	
total				140M	1.6B

paper has 6.6M - 7.5M parameters and around 500M - 1.6 B FLOPS. Various variations of the Inception model are used in FaceNet which have comparatively less parameters and filters.

Finally, FaceNet input should be a human face and fixed size. It is crucial to resize all images into fixed-size ones before the training process. Moreover, the images not only contain the human face but also many non-human faces, thus, the human face detection tasks need to be carried out in the preprocessing stage to improve the performance. The next section talks about the face detection and alignment approach that I use to crop the human face.

3.2. Multi-task Cascaded Convolutional Neural Networks (MTCNN)

MTCNN is a Joint alignment-recognition network [35] that was proposed to jointly train FR with 2 tasks (face detection and alignment) together. The network exploits the inherent correlation between them to enhance the performance. Regarding application, MTCNN is able to detect real-time with fairly high accuracy. The proposed MTCNN's purpose is to construct an advanced structure and use it as material for multi-task knowledge to predict



(a) Inception module, naïve version

(b) Inception module with dimension reductions

Figure 5: Inception network [34]

the location of the face in a coarse-to-fine way. This proposed architecture cascades 3 CNNs networks. The overall pipeline of our approach is shown in Figure 7.

The initial CNN network in this suggested architecture is the Proposal Network (P-Net) – the shallow CNN, which operates to obtain the face area and provide some boundary boxes for the face. The second network is the more complex CNN called Refine Network (R-Net), which removes several bounding boxes from the face by calibrating them and leaving only an accurate bounding box. The final network is the more powerful CNN called Output Network (O-Net). This stage is similar to the second stage, but in this stage there are more details in the face that are detected (five facial landmarks' positions). The cascaded CNN architectures are shown in Figure 6.

The MTCNN network trains 3 tasks: face/non-face classification, bounding box regression, and facial landmark localization.

3.2.1. Face classification

For each sample , they applied the cross-entropy loss:

$$L_i^{\text{det}} = -\left(y_i^{\text{det}} \log\left(p_i\right) + \left(1 - y_i^{\text{det}}\right)\left(1 - \log\left(p_i\right)\right)\right) \tag{4}$$

Where p_i is the probability produced by the network that indicates a sample being a face. The notation $y_i^{det} \in \{0, 1\}$ denotes the ground-truth label.

3.2.2. Bounding box regression

The Euclidean loss for each sample:

$$L_i^{box} = \left\| \hat{y}_i^{box} - y_i^{box} \right\|_2^2 \tag{5}$$

where $\hat{y}_i{}^{box}$ predicted from the network and y_i^{box} is the ground-truth coordinate. There are four coordinates, including left top, height and width, and thus $y_i^{box} \in \mathbb{R}^4$

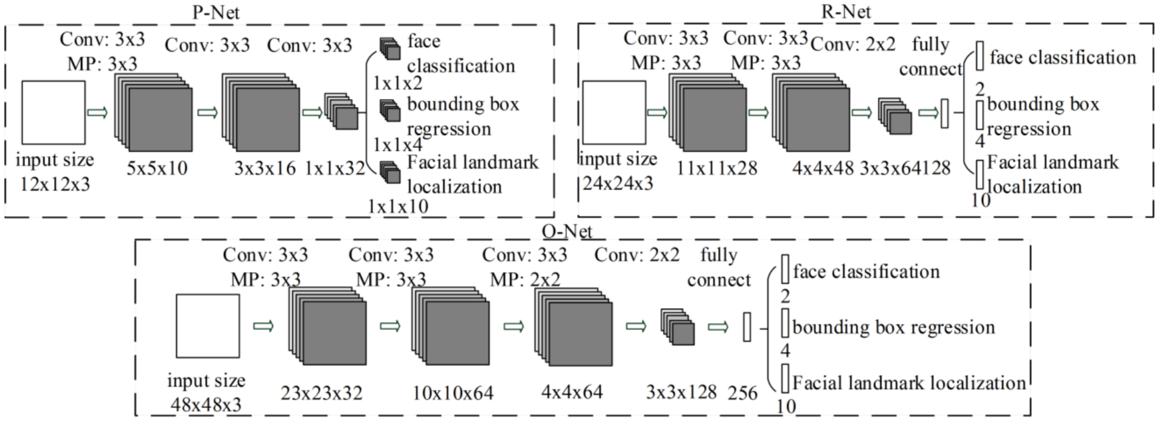


Figure 6: The architecture of P-Net, R-Net, and O-Net. Where "MP" means max pooling and "Conv" means convolution. The step size in convolution and pooling is 1 and 2, respectively [35]

3.2.3. Facial landmark localization

We minimize the Euclidean loss:

$$L_i^{\text{landmark}} = \left\| \hat{y}_i^{\text{landmark}} - y_i^{\text{landmark}} \right\|_2^2 \tag{6}$$

where $\hat{y_i}^{landmark}$ is the facial landmark's coordinate obtained from the network and $y_i^{landmark}$ is the ground-truth coordinate. There are five facial landmarks, including left eye, right eye, nose, left mouth corner, and right mouth corner, and thus $y_i^{landmark} \in \mathbb{R}^{10}$.

3.2.4. Multi-source training

Since this network is a multi-task CNNs, then the overall learning target can be formulated as:

$$\min \sum_{i=1}^{N} \sum_{j \in \{\text{ det }, \text{ box }, \text{ landmark }\}} \alpha_j \beta_i^j L_i^j \tag{7}$$

where N is the number of training samples. α_j denotes the task importance. We use $(\alpha_{det} = 1, \alpha_{box} = 0.5, \alpha_{landmark} = 0.5)$ in P-Net and R-Net, while $((\alpha_{det} = 1, \alpha_{box} = 0.5, \alpha_{landmark} = 1))$ in O-Net for more accurate facial landmarks localization. $\beta_i^j \in 0, 1$ is the sample type indicator.

3.2.5. Online Hard sample mining

In addition, in the learning process, [35] proposes a new online hard sample mining strategy that can improve the performance automatically without manual sample selection. The online hard sample mining aims to select the top $k\%$ hardest from all samples to compute

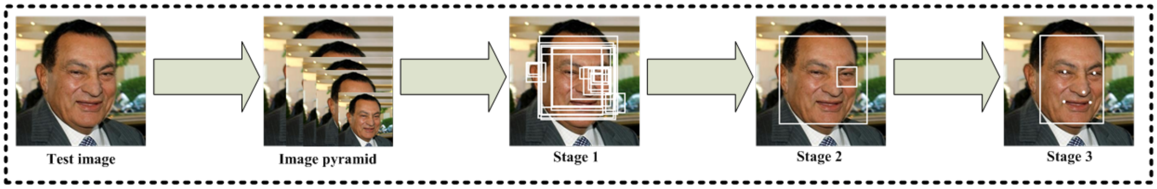


Figure 7: Pipeline of cascaded framework consists of 3 stages. Stage 1: candidate windows are produced through a fast Proposal Network (P-Net). Stage 2: refine these candidates through a Refinement Network (R-Net). Stage 3: The Output Network (O-Net) produces the final bounding box and facial landmarks position. [35]

gradient descent. [35] 's experiments illustrate that this approach boosts up the performance without manual sample selection.

4. Implementation and Results

There are 4 public datasets: YALE [1], AT&T [36], 5-celebrity-faces-dataset [2] and YALE B+ [37], which used to evaluate Facenet on the face recognition task. Utilizing standard face image dataset will be easy to compare with other methods that have been proposed in various previous studies. To do face recognition with FaceNet, the following steps are carried out:

4.1. Preprocessing

In the preprocessing stage, each image will be cropped such that the face in that image is detected and aligned correctly. This stage is carried out by MTCNN. After successfully applying MTCNN, the original image on the dataset with the size of x pixels × y pixels is done by cutting according to the detected face area with a size of 160 pixels × 160 pixels. See Figure 8. In the YALE B+ dataset, since the faces of a person are almost in a stable location, I apply MTCNN once for each individual and copy the location to all images of the same person. This strategy speeds up the preprocessing time from 9 to 4.5 hours on the CPU.

4.2. Training

After preprocessing all images of all datasets using MTCNN method, the model will then be trained. Since the FaceNet training process requires complex computing and a long time, in this report, I use a pre-trained model to evaluate the performance of face recognition tasks. There are a variety of pre-trained models, each one is trained on different architectures and different datasets. The one I choose is Keras FaceNet provided by Hiroki Taniai [3], which employ the Inception ResNet v1 model. It was trained on MS-Celeb-1M dataset [39] and

[1] YALE database: http://vision.ucsd.edu/content/yale-face-database
[2] 5-celebrity-faces-dataset : https://www.kaggle.com/dansbecker/5-celebrity-faces-dataset
[3] Keras FaceNet provided by Hiroki Taniai: https://github.com/nyoki-mtl/keras-facenet

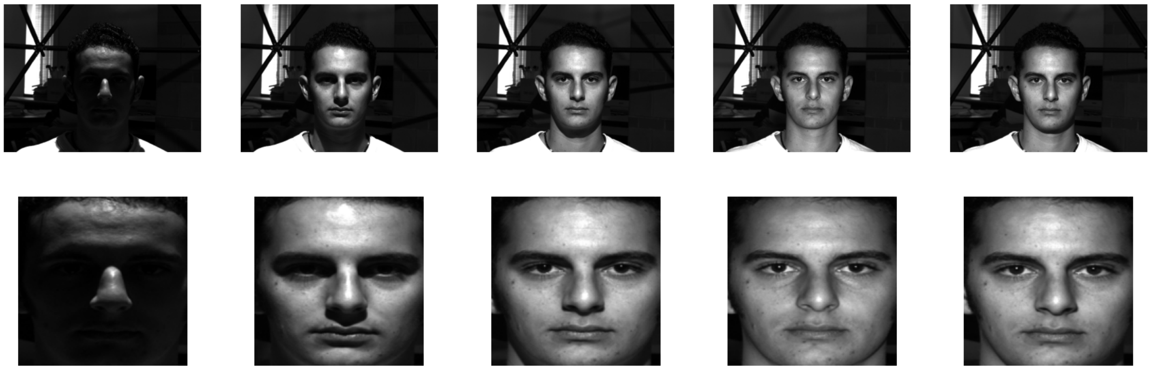


Figure 8: The top images are from Yale B+ datasets. The images below are cropped and resized into 160 pixels × 160 pixels using MTCNN.

expects input images to be color, to have their pixel values whitened (standardized across all three channels), and to have a square shape of 160 pixels × 160 pixels.

Then training the dataset of images that have been preprocessed and then collected in a folder called $< dataset\_name > .npz$ with labels according to their names and then trained using Facenet pre-trained model. The results of this process will be embedding vectors of images that will store in $< dataset\_name - embeddings.npz >$ format. Figure 9 illustrates the whole training process from an original image to an embedding vector corresponding to. This embedding has the nice property that a larger distance between two face embeddings means that the faces are likely not of the same person. This property makes clustering, similarity detection, and classification tasks easier than other face recognition techniques where the Euclidean distance between features is not meaningful [38].

The last is to apply our favorite clustering or classification techniques to the features to complete the recognition task. In this experiment, I use support vector machines (SVMs) with the linear kernel using 128-dimensional face embeddings to classify and recognize the human face. About the evaluation, the ratio is 75% train and 25% test.

4.3. Results

At this stage, I implement an accuracy test for face recognition in each face database. The results of the accurate measurement of face recognition on each face image dataset are presented in Table 2. Each face is transformed into a 128-dimensional embedding vector after the FaceNet inference step. These embedding vectors are used as input to the SVMs in the measurement technique. Based on the results of FaceNet testing in Table 2, this shows that FaceNet accuracy is very accurate to use on 3 simple datasets: YALE, AT&T, 5-celebrity-faces-dataset. As shown in the table on YALE B+ dataset, the use of MTCNN yields 8.95% improvement over the FaceNet without MTCNN. However, the accuracy in the YALE B+ (uncropped) and YALE B+ (cropped) dataset is not optimal, this is possible because in this dataset there are several face labels that are in extremely low light condition, as shown in the first image of Figure 8. This poor condition affects the detection face

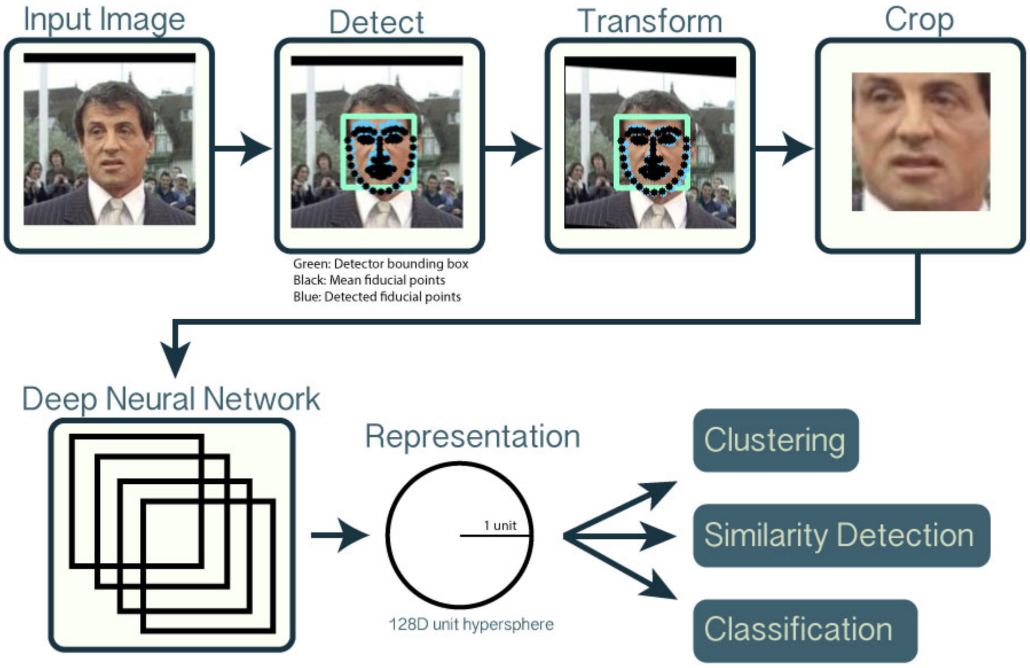


Figure 9: Pipeline for face recognition system contains two main stages: preprocessing and training. The preprocessing stage will detect and crop the face in all images using MTCNN. The output of the preprocessing step will continue to feed to the training stage, in which FaceNet transforms the processed images into 128D embeddings. By applying Euclidean distances between embedding images, it is simple to measure face similarity such as clustering, similarity detection, and classification [38]

performance of MTCNN, and then degrading FaceNet's accuracy. Furthermore, cropping only once for each person image due to minor location variation yields a reasonable but not perfect result, affecting the accuracy of face detection and alignment. Finally, the images were captured under different lighting conditions and various facial expressions also slightly affect the accuracy of using MTCNN and FaceNet.

5. Conclusion

The purpose of this study report is to conduct a Face recognition survey, propose a multi-task cascaded CNNs based framework (MTCNN) combined with a unified embedding for face detection and recognition (Facenet), and then evaluate the performance of these mixture networks on a variety public datasets. Although Facenet is relatively outdated nowadays, it was state-of-the-art in 2015 and became a very handy tool for real-time face recognition, verification, and clustering until now. Several public datasets, including YALE, AT & T, 5-celebrity-faces-dataset, and YALE B+ were tested. Keras FaceNet provided by Hiroki

Table 2: Face recognition results using FaceNet in each facial image database with 2 settings in preprocessing step. Uncropped: resize into 160x160 only. Cropped: employing MTCNN and then resizing into 160x160. N/A: Uncropped preprocessing can not apply in cropping images.

Dataset	**Total images**	**Cropped images**	**accuracy (%)**	
			train	**test**
YALE(cropped/uncropped)	165	165 (N/A)	100	100
AT&T(cropped/uncropped)	400	400 (N/A)	100	100
5-celebrity-faces-dataset (cropped/uncropped)	117	117 (N/A)	100	100
YALE B+ (uncropped)	16380	N/A	89.74	88.11
YALE B+ (cropped)	16380	12	96.77	96.06

Taniai pre-trained models is also used in testing. According to the test results, the exactness of the FaceNet algorithm is very good that can approach up to 100% accuracy. Compared to the result of the non-processing strategy (uncropped) on the YALE B+ dataset, cascading MTCNN and FaceNet strategy that detects and aligns face images before training is proven to have better performance.

References

[1] M. Turk and A. Pentland. Eigenfaces for recognition. *undefined*, 3:71–86, 1991. ISSN 0898929X. doi: 10.1162/JOCN.1991.3.1.71.

[2] Dong Chen, Xudong Cao, Fang Wen, and Jian Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. 2013. doi: 10.1109/CVPR.2013.389.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[4] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. 2014. doi: 10.1109/CVPR.2014.220.

[5] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1891–1898, 2014.

[6] Mei Wang and Weihong Deng. Deep face recognition: A survey. 4 2018. doi: 10.1016/j.neucom.2020.10.081. URL http://arxiv.org/abs/1804.06655http://dx.doi.org/10.1016/j.neucom.2020.10.081.

[7] Peter N. Belhumeur, Joao P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[8] Baback Moghaddam, Wasiuddin Wahid, and Alex Pentland. Beyond eigenfaces: Probabilistic matching for face recognition. In *Proceedings third IEEE international conference on automatic face and gesture recognition*, pages 30–35. IEEE, 1998.

[9] Weihong Deng, Jiani Hu, Jiwen Lu, and Jun Guo. Transform-invariant pca: A unified approach to fully automatic facealignment, representation, and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1275–1284, 2013.

[10] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):328–340, 2005.

[11] Shuicheng Yan, Dong Xu, Benyu Zhang, and Hong-Jiang Zhang. Graph embedding: A general framework for dimensionality reduction. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 830–837. IEEE, 2005.

[12] Jian Yang, David Zhang, Jing-yu Yang, and Ben Niu. Globally maximizing, locally minimizing: unsupervised discriminant projection with applications to face and palm biometrics. *IEEE transactions on pattern analysis and machine intelligence*, 29(4): 650–664, 2007.

[13] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2008.

[14] Lei Zhang, Meng Yang, and Xiangchu Feng. Sparse representation or collaborative representation: Which helps face recognition? In *2011 International conference on computer vision*, pages 471–478. IEEE, 2011.

[15] Weihong Deng, Jiani Hu, and Jun Guo. Extended src: Undersampled face recognition via intraclass variant dictionary. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1864–1870, 2012.

[16] Weihong Deng, Jiani Hu, and Jun Guo. Face recognition via collaborative representation: Its discriminant nature and superposed representation. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2513–2521, 2017.

[17] Chengjun Liu and Harry Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image processing*, 11(4):467–476, 2002.

[18] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):2037–2041, 2006.

[19] Wenchao Zhang, Shiguang Shan, Wen Gao, Xilin Chen, and Hongming Zhang. Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 786–791. IEEE, 2005.

[20] Weihong Deng, Jiani Hu, and Jun Guo. Compressive binary patterns: Designing a robust binary face descriptor with random-field eigenfilters. *IEEE transactions on pattern analysis and machine intelligence*, 41(3):758–767, 2018.

[21] Zhimin Cao, Qi Yin, Xiaoou Tang, and Jian Sun. Face recognition with learning-based descriptor. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2707–2714, 2010.

[22] Zhen Lei, Matti Pietikäinen, and Stan Z Li. Learning discriminant face descriptor. *IEEE Transactions on pattern analysis and machine intelligence*, 36(2):289–302, 2013.

[23] Tsung-Han Chan, Kui Jia, Shenghua Gao, Jiwen Lu, Zinan Zeng, and Yi Ma. Pcanet: A simple deep learning baseline for image classification? *IEEE transactions on image processing*, 24(12):5017–5032, 2015.

[24] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, R. Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 2, 1989.

[25] 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.

[26] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2892–2900, 2015.

[27] Yi Sun. *Deep learning face representation by joint identification-verification*. The Chinese University of Hong Kong (Hong Kong), 2015.

[28] Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

[29] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 41.1–41.12. BMVA Press, September 2015. ISBN 1-901725-53-7. doi: 10.5244/C.29.41. URL https://dx.doi.org/10.5244/C.29.41.

[30] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June-2015:815–823, 3 2015. doi: 10.1109/CVPR.2015.7298682. URL http://arxiv.org/abs/1503.03832http://dx.doi.org/10.1109/CVPR.2015.7298682.

[31] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[32] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Advances in Neural Information Processing Systems*, 18, 2005.

[33] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8689 LNCS:818–833, 11 2013. ISSN 16113349. doi: 10.1007/978-3-319-10590-1_53. URL https://arxiv.org/abs/1311.2901v3.

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842.

[35] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(1):1499–1503, 10 2016. ISSN 10709908. doi: 10.1109/LSP.2016.2603342.

[36] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. 1994. doi: 10.1109/acv.1994.341300.

[37] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[38] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[39] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9907 LNCS:87–102, 7 2016. ISSN 16113349. doi: 10.1007/978-3-319-46487-9_6. URL https://arxiv.org/abs/1607.08221v1.