FaceNet: A Unified Embedding for Face Recognition and Clustering

Weilun Song

Abstract

Despite significant recent advances in the field of face recognition, implementing face verification and recognition efficiently at scale presents serious challenges to current approaches. A system, called FaceNet, that 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.

The method uses a deep convolutional network trained to directly optimize the embedding itself, rather than an intermediate bottleneck layer as in previous deep learning approaches. To train, Facenet use triplets of face patches generated using a novel online triplet mining method.

1. Introduction

The method is based on learning a Euclidean embedding per image using a deep convolutional network. The network is trained such that the squared L2 distances in the embedding space directly correspond to face similarity: faces of the same person have small distances and faces of distinct people have large distances.

FaceNet directly trains its output using a triplet-based loss function. the triplets consist of two matching face thumbnails and a non-matching face thumbnail and the loss aims to separate the positive pair from the negative by a distance margin. Choosing which triplets to use is very important for achieving good performance. Facenet present a novel online negative exemplar mining strategy which ensures consistently increasing difficulty of triplets as the network trains. To improve clustering accuracy, it also explores hard-positive mining techniques which encourage spherical clusters for the embeddings of a single person.

Fig.1 This figure shows the output distances of FaceNet between pairs of faces of the same and a different person in different pose and illumination combinations. You can see that a threshold of 1.1 would classify every pair correctly.

2. Method

FaceNet uses a deep convolutional network. Google's 'Inception resnet v1' Given the model details, and treating it as a black box, the most important part of the approach lies in the end-to-end learning of the whole system. To this end FaceNet employ the triplet loss that directly reflects face verification, recognition and clustering.

It strives for an embedding f(x), from an image x into a feature space R^d, such that the squared distance between all faces, independent of imaging conditions, of the same identity is small, whereas the squared distance between a pair of face images from different identities is large.

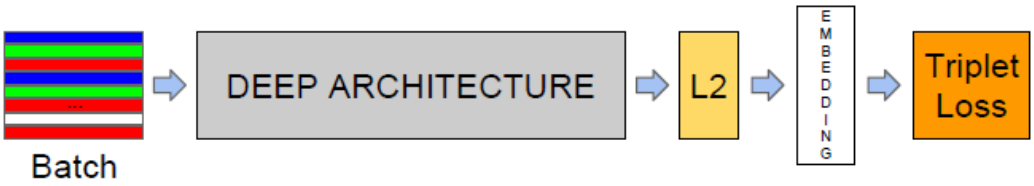


Fig.2 Facenet consists of a batch input layer and a deep CNN followed by L2 normalization, which results in the face embedding. This is followed by the triplet loss during training.

2.1. Triplet Loss

The embedding is represented by f(x) $\in \mathrm{R}^d$. It embeds an image x into a d-dimensional Euclidean space. Here we want to ensure that an image x_i^a (anchor) of a specific person is closer to all other images x_i^p (positive) of the same person than it is to any image x_i^n (negative) of any other person.

Thus,

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \,,$$

$$\forall \left(f(x_i^a), f(x_i^p), f(x_i^n)\right) \in \mathcal{T}$$

where α is a margin that is enforced between positive and negative pairs. T is the set of all possible triplets in the training set and has cardinality N.

The loss that is being minimized is then

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

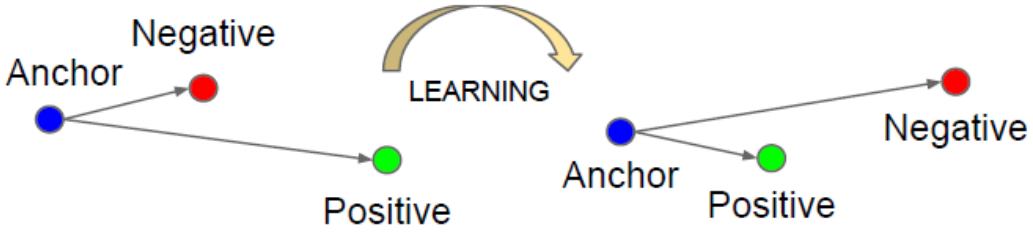


Fig.3 The Triplet Loss minimizes the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a

different identity.

2.2. Triplet Selection

To ensure fast convergence, it is crucial to select triplets that violate the triplet constraint. This means that, given x_i^a, we want to select an x_i^p (hard positive) such that

$$\text{argmax}_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$$

and similarly, x_i^n (hard negative) such that

$$\text{argmin}_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2.$$

It is infeasible to compute the argmin and argmax across the whole training set. Additionally, it might lead to poor training, as mislabeled and poorly imaged faces would dominate the hard positives and negatives. Generate triplets online can be done by selecting the hard positive/negative exemplars from within a mini-batch to solve issues describe above.

2.3. Deep Convolutional Networks

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Fig.4 Google inception resnet v1

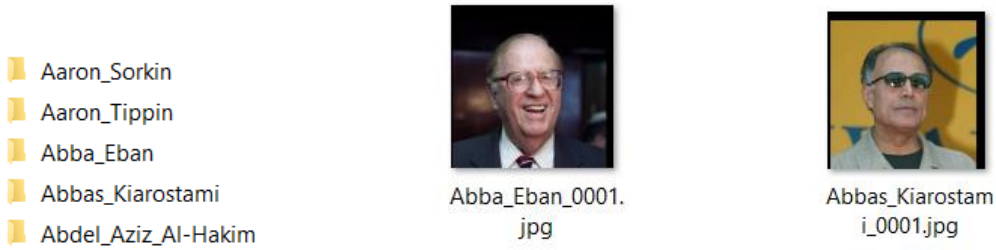
3. Datasets and Evaluation

Training dataset:

CASIA-WebFace: 10k+ people with about 500k faces

Test dataset:

LFW dataset:5.7k+ people with 13k+ faces



4. Experiments

4.1. image alignment

Google's 'inception resnet v1' need 160*160-pixel input. First apply a MTCNN face detector to run on each image and a tight bounding box around each face is generated.

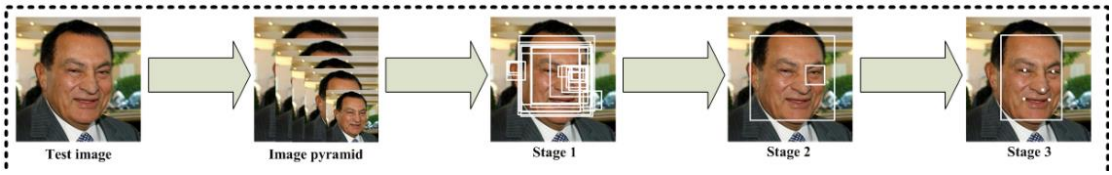


Fig.5 Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks

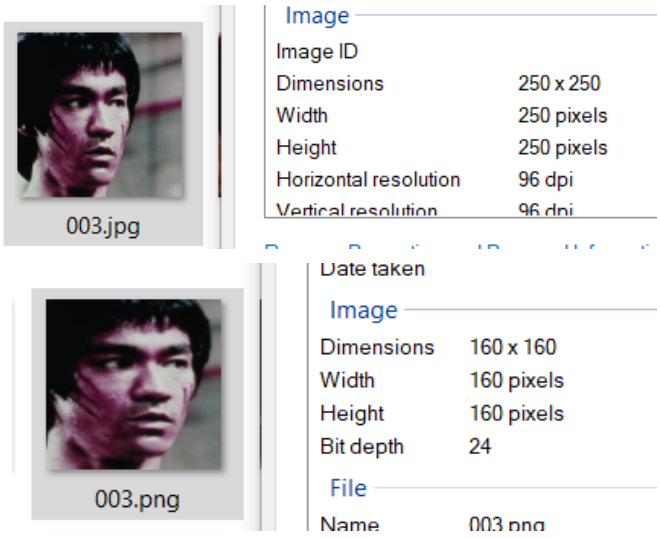


Fig.6 Face bounding and alignment

4.2. Amount of Training Data

Table shows the impact of large amounts of training data. Due to time constraints this evaluation was run 20 hours. The effect may be even larger on larger models. Using hundreds of thousands of exemplars results in a clear boost of accuracy on personal photo test set. Using another order of magnitude more images (hundreds of thousands) still gives a small boost, but the improvement tapers off.

# of training image	Accuracy	Validation rate
5K	65.1%	3.5%±1%
50K	71.1%	4.3%±1%
100K	72.3%	5.1%±1%
500K	84.8%	13.9%±2%

4.3. Effect of CNN Model

This table compares the performance of model architectures on the hold out test set. Reported is the accuracy and validation rate. It can be found that Google's inception resnet v1 gives a high accuracy and validation rate than squeezenet.

Architecture	Accuracy	Validation rate
Inception resnet v1	84.8%	13.9%±2%
squeezenet	74.6%	5.6%±1%

5. Summary

Facenet provide a method to directly learn an embedding into a Euclidean space for face verification. This sets it apart from other methods which use the CNN bottleneck layer or require additional post-processing such as concatenation of multiple models and PCA, as well as SVM classification.

The end-to-end training both simplifies the setup and shows that directly optimizing a loss relevant to the task at hand improves performance.

Another strength of the model is that it only requires minimal alignment (tight crop around the face area).

5.1. Future

Future work will focus on better understanding of the error cases, further improving the model, and reduce model size and reducing CPU requirements. And investigate ways of improving the currently extremely long training times, e.g. variations of the curriculum learning with smaller batch sizes and offline as well as online positive and negative mining.

Reference:

Florian Schroff, Dmitry Kalenichenko, James Philbin, ***FaceNet: A Unified Embedding for Face Recognition and Clustering***