Face Recognition via Convolutional Neural Networks and Siamese Neural Networks

Wanxin Cui
School of Computer Science
Yangtze University
Jinzhou, Hubei, China

Wei Zhan *
School of Computer Science
Yangtze University
Jinzhou, Hubei, China
e-mail: zhanwei814@yangtzeu.edu.cn

Jingjing Yu
School of Computer Science
Yangtze University
Jinzhou, Hubei, China

Chenfan Sun
School of Computer Science
Yangtze University
Jinzhou, Hubei, China

Yangyang Zhang
School of Computer Science
Yangtze University
Jinzhou, Hubei, China

***Abstract*—Deep convolutional neural networks is playing very important role to solve computer vision task in these decades. In this research has shown implementation state of art face recognition methods and compared them. Advantages and disadvantages of Convolutional and Siamese neural networks is explored for the face recognition task. The novel face recognition system accuracy is checked on the widely used Labelled Faces in the Wild (LFW) dataset. In the study has introduced concepts of triplet loss function. Experiment results can be useful for the developers who is going to create AI applications such as Security system, Self-learning, Visitor analysis system, Face recognition system, Face verification system and many more.**

Keywords—face recognition, convolutional neural networks, Siamese neural networks, triplet loss

I. INTRODUCTION

There are two categories for the face recognition task. The first one is face verification task; it is a one to one matching problem. In example, when you unlock your phone using your face you use face verification, another example is in some airports, you should pass through system that scans your passport and your face to verify you are the correct person. The second one is face recognition task, to find answer the question who is this person. It is a one to many matching problem.

Since CNN based techniques are used the performance of some challenging tasks like face verification, face detection is highly improved.

Another technique to solve above tasks is called one-shot learning. This method of learning representations from an example. In the Siamese neural networks (Siamese NNs), we calculate encodings of taken input image, then, with the same network without doing any updates on network parameters we take an image as an input of different person and calculate its encoding. After these calculations, we can check if there is similarity between the two images.

The structure of the paper is as follows, discuss about related works in section 2, section 3 discuss details and advantages of CNNs and Siamese NNs, about datasets, which is used in the experiment in section 4. Section 5 discuss results of experiment on the LFW datasets. Section 6 conclusion of work.

II. RELATED WORKS

In the last decades, a large of image data have been collected by social networks, in that data includes a lot unconstrained materials such as scenes, objects and faces. The improvement of computational resources and amount of data have enabled the benefit of statistical models. The statistical models have increased the robustness of computer vision systems. However, deep convolutional neural networks have shown better performance and interest for them also increased. In this work, we explore two type of neural networks for the face detection task. The first one is deep CNN architecture based on Inception model of Szedgy, the second one is Siamese NNs.

The first works was done by Li Fei-Fei et al. back to the early 2000's on one-shot learning. In the study created variational Bayesian framework for one-shot image classification (Fei-Fei et al., 2003; Fei-Fei et al., 2006). Wu and Dennis address one-shot learning in the context of path planning algorithm for robotic actuation [1]. In some works have explored other modalities or transfer learning techniques. There are huge number of works on CNNs, let us brief review some recent studies related with face recognition and detection.

Principal component analysis (PCA) on the network output in conjunction with an ensemble of Support Vector Machines (SVM) is used for the face verification task. In other work, Multi stage technique is used for align faces to a general 3D shape model [2]. The authors has trained over four thousand identities to perform face recognition. In the work explored Siamese network and optimized the L1-distance between two face features. The research's best result on LFW dataset was 97.35%. The researchers Sun et al. suggested a simple and cheap to compute network [3]. In the study used an ensemble of twenty five network, each network operating on a different face patch. The final performance was 99.47% on LFW dataset. In that work, proposed method does not require explicit 2D/3D alignment. Using combination of classification and verification loss, the network is trained. The verification loss is similar to the

0-7695-6728-2/19/$31.00 ©2019 IEEE
DOI 10.1109/ICICAS48597.2019.00161

triplet loss that we discuss in the next section and deploy in our face recognition system.

III. Convolution Neural Networks And Siamese Neural

The section provides a brief description of CNNs and Siamese NNs. We concentrate to inspect their interior architectures to analysis strengths and weaknesses on the current face recognition task.

A. Convolution Neural Networks

Concepts of CNNs has introduced in 1995 by Yann LeCun and Yoshua Bengio. A CNN is a feedforward network that can extract topological properties, features from the input image.

To ensure shift, scale, and distortion invariance, convolutional neural networks use three architectural ideas, which are local receive fields, shared weights, and spatial or temporal subsampling [4]. CNNs is trained by back propagation, as well as other neural networks. The network layers equivalent between convolutional layers with feature map $C^{i}_{k,1}$ (Equation 1).

$$C^{i}_{k,l} = g\left(l^{i}_{k,l} \oplus W_{k,l} + B_{k,l}\right) \tag{1}$$

Also non-overlapping sub-sampling layers with feature map S^{i}_{k1} (Equation 2).

$$S^{i}_{k1} = g\left(l^{i}_{k1} \downarrow w_{k,1} + Eb_{k1}\right) \tag{2}$$

In the equations, $g(x) = \tanh(x)$ that means a S-type activation function, b and B both mean biases, w, W are weights, $I^{i}_{k,l}$ is i'th input and $\downarrow$ denotes down sampling. E is a matrix that elements are all one and $\oplus$ is a symbol of two-dimensional convolution. Lower case letters denotes scalars and upper case letters matrices.

A convolutional layer extracts features from local receptive fields in the previous layer. There are feature maps those responsible to detect specific features, it is a planes of neurons organized by convolutional layer.

GoogleNet designs a model called inception model that approximates a sparse CNN with a normal dense construction [5] (Fig. 1). In addition, inception model uses convolutions of varied sizes to capture details at different scales.

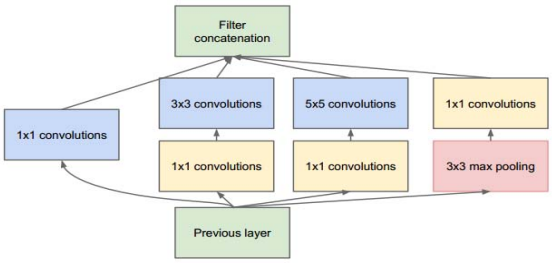

Fig. 1. Inception models

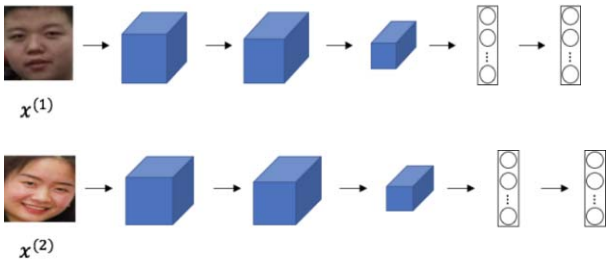

Fig. 2. Process calculating encodings of images.

B. Siamese Neural Networks

In the Siamese NNs, we calculate the encodings of input image, and then with the same network we do same work, calculate encoding image of different person. After calculations, we can compare two encodings to find out whether they are similar. The encodings of images act as a latent feature representations of them. The comparison of encodings shows that images have the same person.

Let us talk about how to train the network, we take anchor image and compare it with positive image example, and negative image example (Fig. 3). The distance between anchor-positive must be small, and between anchor-negative must be big.

$$L = \max(d(a,p)d(a,n) + \text{margin}, 0) \tag{3}$$

The above equation (Equation 3) called triplet loss function that we can use to calculate gradients. Where "a" is an anchor image, "n" a negative image and "p" is a positive image. There is another variable called margin. Margin shows how far should be distance of similarity. Let us see an example, if we take margin=0.3, and $d(a,p)$ then $d(a,n)$ must be at least to 0.8. It helps us better find out the input images. Triplet loss function helps to calculate the gradients, by using gradients we update parameters of the Siamese NNs.

Fig. 3. Anchor, positive, negative image examples.

IV. CNN Architecture And Training Process

In this chapter introduces our CNNs in the experiment as well as training.

A. Face recognition.

At the start, the network structure Φ are tagged by examining the task of classifying N different people, set up as a N-direction classification problem. The network connects to each training picture $l_t, t = 1, \ldots, T$ a label vector $x_t = W\varphi\left(l_t\right) + b \in R^{N}$ These labels are compared to the ground-truth label identity $c_t \in 1, \ldots, N$ by calculating the

empirical softmax log-loss (Equation 4)

$$E(\varphi) = -\sum_t \log\left(\frac{e^{(e_{c_t}, x_t)}}{\sum_{q=1\ldots N} e^{(e_{c_t}, x_t)}}\right) \quad (4)$$

After calculations, the classifier layer can be deleted and the label vectors can be used for face recognition using the Euclidean distance to match them. And also, if we use triplet loss training scheme for matching in Euclidean distance, result might slightly improve. At the end, we can get a good overall performance, and it makes training faster and easier.

B. Using Triplet-Loss For Face Embedding

It is a great performance for the Learning score vectors, and the method triplet loss training purposes on the final application.

Triplet-loss training scheme is used in this study, resembling in spirit to that of [6] The output $\varphi(l_t) \in R^D$ of the convolutional neural network, pre-trained as explained in Section 4.1, is l2-normalised and projected to a $L = D$ dimensional space using an affine projection $x_t = W' \varphi(l_t) / \| \varphi(l_t) \|_2, W' \in \mathrm{R}^{L \times D}$. The linear predictor studied earlier is very similar to this formula. however, it has two distinctions. To begin with, $L \neq D$ is not equal to the statistics of class identities, which is the (arbitrary) size of the descriptor embedding (we make L = 1,024). Secondly, the projection W is trained to minimize the empirical triplet loss (Equation 5)

$$E(W') = \sum_{(a,p,n) \in T} \max\left\{0, \alpha \leq \|x_a x_n\|_2^2 + \|x_a x_p\|_2^2\right\} \quad (5)$$

Where $\mathrm{x_i} = \mathrm{W}' \varphi(l_\mathrm{i}) / \| \varphi(l_\mathrm{i}) \|_2$

Mark that, it has learned no difference in Equation (5) would cancel it. Here $\alpha \geq 0$ is a fixed scalar that represents a learning margin and T is a assemblage of training triplets. An anchor face image and the anchor identities of a positive $p \neq a$ and negative n examples in the triplet (a, p, n).

C. Architecture

Table I shows detail of network architecture. It includes eleven blocks. Each block has a linier operator tracked by one or more non linarites, such as max pooling or ReLU. From the start eight convolutional blocks, other three blocks are fully connected layers. After each convolution layer , there is ReLU as in [4]; however, differently from [4] and similarly to [7], they do not include the Local Response Normalization operator.

Output of the first and second fully connected layers are 4096 dimensional. Loss functions for optimization determines the output of the last one is 2622 or 1024 dimensions. The resulting vector will pass to a softmax layer to calculate the class probabilities. Face image of size (224 x 224) is subtracted, equal to the input size (calculated same as training set) this is really important for the stability of the optimization algorithms.

TABLE I. NETWORK CONFIGURATION DETAILS OF THE FACE CNN CONFIGURATION

Name	Support	Filt dim	Num filts	Stride	pad
0 input -	-	-	-	-	-
1 conv conv1_1	3	3	64	1	1
2 relu relu1_1	1	-	-	1	0
3 conv conv1_2	3	64	64	1	1
4 relu relu1_2	1	-	-	1	0
5 mpool pool1	2	-	-	2	0
6 conv conv2_1	3	64	128	1	1
7 relu relu2_1	1	-	-	1	0
8 conv conv2_2	3	128	128	1	1
9 relu relu2_2	1	-	-	1	0
10 mpool pool2	2	-	-	2	0
11 conv conv3_1	3	128	256	1	1
12 relu relu3_1	1	-	-	1	0
13 conv conv3_2	3	256	256	1	1
14 relu relu3_2	1	-	-	1	0
15 conv conv3_3	3	256	256	1	1
16 relu relu3_3	1	-	-	1	0
17 mpool pool3	2	-	-	2	0
18 conv conv4_1	3	256	512	1	1
19 relu relu4_1	1	-	-	1	0
20 conv conv4_2	3	512	512	1	1
21 relu relu4_2	1	-	-	1	0
22 conv conv4_3	3	512	512	1	1
23 relu relu4_3	1		-	1	0
24 mpool pool4	2	-	-	2	0
25 conv conv5_1	3	512	512	1	1
26 relu relu5_1	1	-	-	1	0
27 conv conv5_2	3	512	512	1	1
28 relu relu5_2	1	-	-	1	0
29 conv conv5_3	3	512	512	1	1
30 relu relu5_3	1	-	-	1	0
31 mpool pool5	2	-	-	2	0
32 conv fc6	7	512	4096	1	0
33 relu relu6	1	-	-	1	0
34 conv fc7	1	4096	4096	1	0
35 relu relu7	1	-	-	1	0
36 conv fc8	1	4096	2622	1	0
37 softmax prob	1	-	-	1	0

D. Training

Studying the n way face classifier accompany the steps of [4] with the changes proposed by [7]. By using the way, the parameters of the neural network what we can discover minimize the prediction log-loss after softmax layer.

Here is description for the procedure for the CNN parameters. Optimization is done by SGD, mini batch 64, momentum coefficient of 0.9 [5]. In the study, we used dropout method after the 2 fully connected layer with rate 0.5. Initial learning rate set to 10-2, decreased by factor of 10 if the validation accuracy stops increasing.

Gaussian distribution is used while initializing weights of filters in the convolutional neural network. Images are rescaled to 256 width and heights. While training random 224 x 224 pixel patches cropped to feed the network. Data augmentation is used during the training by flipping the picture left to right with 50% probability. In the experiments, didn't perform any colour channel augmentation as described in [4] and [7].

Using triplet loss for face embedding described in section 4.2, except the last fully connected layer neural network is frozen. This layer is learnt for 10 epochs with SGD method, and the fixed learning rate is 0.25. Each epoch has all the viable positive pairs (a, p), here image is evaluated the anchor and p is the positive example for

pairing it.

At the time of test, the embedded descriptors $W'\varphi(l_t)$ were compared with Euclidean distance for the face verification. In verification, the goal is to determine whether two face images l_1 and l_2 have the same identity; this is obtained by testing whether the distance between $\left\|W'\varphi(l_t)W'\varphi(l_2)\right\|_2$ a threshold τ is bigger than embedded descriptors. In order to maximize the verification accuracy on suitable validation data, this threshold should be provided by learned separately.

V. Dataset

To estimate the performance of the neural network architecture, numerous experiments are done on widely used face database called LFW .

The database have more than 13,000 images of faces, each image has labelled with the name of individuals. In the LFW database, 1680 of characters has two or more different photos. Table II shows statistics of the face database.

TABLE II. LFW Database Statistics

Database	# of people	Total images
LFW	5749	13233
FRGC	>466	>50000
BioID	23	1521
FERET	1199	14126

Comparison of LFW, FRGC, BioID, and FERET datasets.

# of images /person	# of people %of people	# of images % of images
1	4069 (70.8)	4096 (30.1)
2-5	1369 (23.8)	3739 (28.3)
6-10	168 (2.92)	1251 (9.45)
11-20	86 (1.5)	1251 (9.45)
21-30	25 (0.43)	613 (4.63)
31-80	27 (0.47)	1170 (8.84)
>81	5 (0.09)	1140 (8.61)
Total	5749	13233

Distribution of LFW.

VI. Experiment Result

The Fig. 4 shows the experiment result.

In the Table III, number of parameters, average training time per epoch and train accuracy has shown.

TABLE III. Number of Parameters, Average Training Time Per Epoch and Train Accuracy

Number of parameters	Average training time per epoch (s)	Train accuracy (%)	Test accuracy
6.7977 mln	60	~97	~94

The average of all feature vectors is the final description of the face.

Method of detecting face is [8]. If face alignment is used, using the method of [9] to compute facial landmarks and mapping a face to a canonical position using a 2d similarity transformation.

Fig. 4. face detect result.

VII. Conclusion

In the research, CNNs and Siamese NNs method are explored for face recognition task with results on LFW dataset. Advantages of CNNs and Siamese NNs, with triplet loss function is investigated. The purposed method result can help to build face recognition system for organizations. In the future projects, we plan to explore fine tuning approach for improving face recognition accuracy.

References

[1] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts Amherst Tech. Rep., vol. 1, pp. 07-49, 2007.

[2] E. van der Spoel et al., "Siamese Neural Networks for One-Shot Image Recognition," Aging (Albany. NY)., vol. 7, no. 11, pp. 956–963, 2015.

[3] Y. Sun, X. Wang, and X. Tang, Deep Learning Face Representation by Joint Identification-Verification, 2014, pp. 1-9.

[4] H. Khalajzadeh, M. Mansouri, and M. Teshnehlab, "Face Recognition Using Convolutional Neural Network and Simple

Logistic Classifier," Soft Comput. Ind. Appl., pp. 197-207, 2014.

[5] Y. Bengio and P. Haffner, Gradient-Based Learning Applied to Document Recognition, vol. 86, no. 11, pp. 1-46, 1998.

[6] K. J. Russkovsky O, Deng J, "Imagenet large scale visual recognition challenge," IJCV, 2015.

[7] K. D. Schroff F, Facenet: A unified embedding for face recognition and clustering," Proc. CVPR, 2015.

[8] C. Szegedy et al., Going Deeper with Convolutions, pp. 1-9, 2014.

[9] P. M. Mathias M, Benenson R, Face detection withoutbells and whistles, Proc. ECCV, 2014.