Face verification using convolutional neural networks with Siamese architecture

Conference Paper · September 2017

DOI: 10.23919/ELMAR.2017.8124469

Dominik Sopiak
Slovak University of Technology in Bratislava
8 PUBLICATIONS **15** CITATIONS
SEE PROFILE

Jarmila Pavlovicova
Slovak University of Technology in Bratislava
46 PUBLICATIONS **213** CITATIONS
SEE PROFILE

Face Verification Using Convolutional Neural Networks with Siamese Architecture

Zuzana Bukovčiková, Dominik Sopiak, Miloš Oravec, Jarmila Pavlovičová
Faculty of Electrical Engineering and Information Technology/Slovak University of Technology in Bratislava,
Ilkovičova 3, 812 19, Bratislava, Slovak Republic
{xbukovcikova; dominik.sopiak; milos.oravec; jarmila.pavlovicova}@stuba.sk

***Abstract*—This paper evaluates the ability of convolutional networks to solve the problems arising with face classification in unconstrained environment. It contains design and implementation of Siamese architecture consisting of two convolutional networks used for face verification on sets of photographs. In the scope of the paper, training process is closely monitored and we evaluate several practices and parameters as well as their impact on the network learning.**

***Keywords*—Biometrics; Face Recognition; Machine Learning; Deep Convolutional Networks; Siamese Networks**

I. Introduction

Nowadays, thanks to significant progress in area of computer vision and image processing we can see various usage of these methods. There is great breakthrough in area of face recognition due to fact most of face recognition systems are based on methods of image processing and machine learning. This paper aims at proposing of face verification method based on Siamese architecture of convolutional neural network.

There are many papers that are focused on face recognition systems like [1], where local binary feature learning method was proposed. In [2] authors create surveys of sparse coding and dictionary learning algorithms used for face recognition systems. They do not only categorize existing dictionary learning algorithms but also show detail description of each of them. Authors in [3] or [4] focus on creating systems for mobile devices, where they need to solve the problem with limited resources.

Sparse coding is slowly replaced by other methods, where the most famous one is convolutional neural network (CNN). Usage of CNN for face recognition is mentioned in [5] and [6]. The main disadvantage of CNNs is the fact, that we need many diverse images per class to train this method. As solution to this problem, authors in [7], [8] propose CNN with Siamese architecture. Fusion of two parallel networks is usually made by cost function, whose main task is to classify features formed from networks. The aim of this paper is to replace this cost function, with simple Multi Layer Perceptron classifier.

The paper is organized as follows. Section II describes the proposed method of image verification and also image dataset, that was used for training and testing purposes. Section III describes our experimental pipeline. Results are presented in Section IV. All results and plans for future work are summarized in the end of this paper.

II. Proposed system

In this section we describe our proposed solution. At first we show used reference face database, that we use for training and testing [9]. Then we describe architecture of convolutional network that was used in Siamese architecture.

A. Input

CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than 200 celebrity images, each with 40 attribute annotations. The images in this dataset cover large pose variations and background clutter. CelebA has large diversities, large quantities, and rich annotations, including 10,177 number of subjects, 202,599 number of face images, and 5 landmark locations, 40 binary attributes annotations per image. Number of images per subject is between 1 and 35 [9].

Before training, all images are cropped by annotations, that describe position of face in images. Then images are resized to 73×53 pixels and organized into directories by subject ID. An example of different images of same person is shown in the Fig. 1. An example of cropped images is shown in Fig.

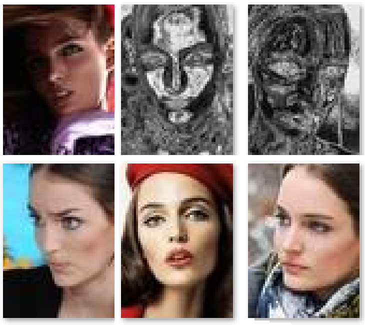

Figure 1: Example of various images of the same person [9].

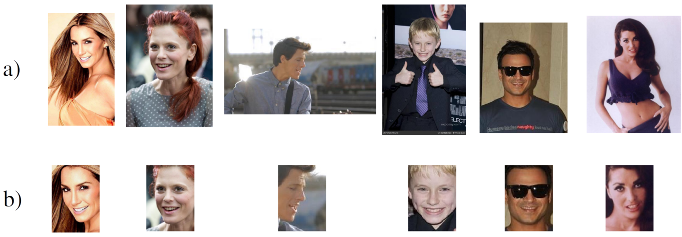

Figure 2: Example of CelebA database [9].

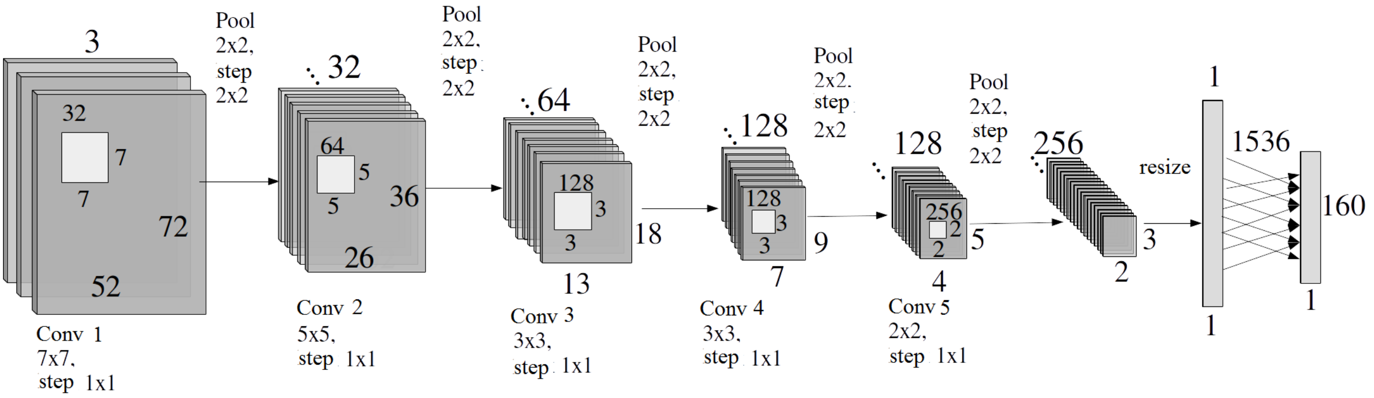

Figure 3: Proposed architecture of CNN.

2. In the first row (Fig.2 (a)) you can see original images and second row (Fig.2 (b)) shows cropped images.

B. *Convolutional network*

Our implemented convolutional network, is based on commonly used AlexNet [10] architecture, however we change the last fully connected layers as well as the size of kernels in convolutions.

Our network consists of five convolutional layers followed by fully-connected one. Since the input samples are quite large with size of $73 \times 53 \times 3$, we use bigger kernel in first layer (7×7). Size of kernels is gradually getting smaller, using 5×5 in second layer, 3×3 pixels in third and fourth layer, and finally only 2×2 in the fifth layer. The number of kernels in each layer is gradually increasing, starting at 3 in first and reaching 256 kernels in the last layer. The architecture is illustrated in Fig. 3. Input matrices to convolutions are all padded with zeros according to the same-padding principles of deep networks. Rectified Linear Unit (ReLU) is used as an activation function for all convolutional layers:

$$f(x) = max(0, x). \tag{1}$$

Every convolutional layer is followed by pooling layers with kernel size 2×2, to decrease dimension of input features. Pooling layers of CNN are all padded with zeros in the same manner as we do with convolutional layes. Output matrices of the last convolutional layer are transformed into vectors and sent as input into fully-connected layer. Hyperbolic tangent is used as activation function ((2)) for fully-connected output layer with 160 neurons.

$$f(x) = tanh(x). \tag{2}$$

CNN with 655 010 parameters has to be train, which requires 2.5MB of memory. If we need to store all partial results we shall need all together 571.5 MB of memory. But in the case of CNN we need to keep in each step only partial results between two adjacent convolutional layers which covers only 295 MB of memory.

C. *Siamese architecture*

Two face images are inputs for our system and system should decide if there is the same person on both images or there are two different peoples. Images of the same person are called genuine pairs. Images of different persons are called impostor pairs. For pair classification we use Siamese architecture, which was proposed in [11] and modified for face verification in [12].

Instead of contrast loss function, that was used in papers [11], [12] for fusion of left and right side, we use another fully-connected layer as it is shown in Fig. 5.

$$d(x, y) = \|x - y\| \tag{3}$$

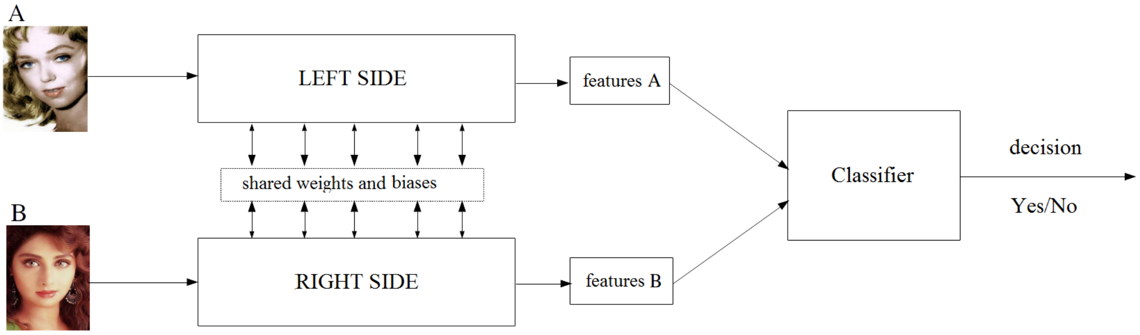

Figure 4: Siamese architecture.

For our fully-connected fusion layer we used two output neurons - one for genuine pairs and another for impostor pairs. Softmax function (4) is used as activation function.

$$L_i = -log\left(\frac{e^{f_{y_i}}}{\sum_{j=0}^{n} e^{f_j}}\right). \qquad (4)$$

The Siamese framework merges two identical network with one cost module. The pair of images and label is used as input to the system. One image is sent to the right net another one to the left net. Diagram of Siamese network is shown in Fig. 4. Both nets share their parameters and both of them have the same architecture. While training optimizer evaluates cost function only once and all weights are adjusted based on this decision.

For our purpose we decided to use ADAM (Adaptive momentum estimation) optimizer. It is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments [13]. For another training regulation we also use dropout. All biases were initialized with zero value and weights were initialized by Xavier initialization [14]. Learning rate value was set at 0.005.

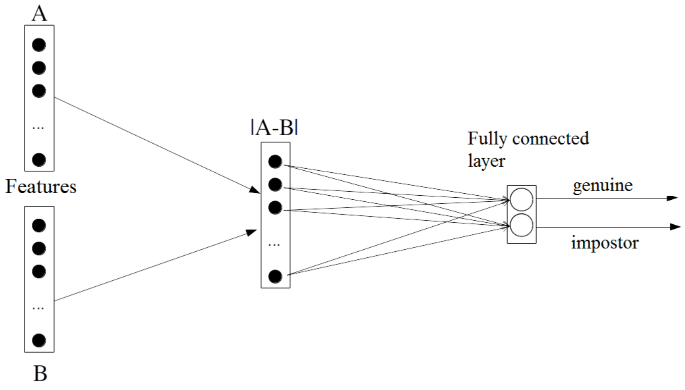

Figure 5: Classification network.

III. CNN Training

Training set consists of 198 656 images, that are combined into 99 328 pairs with balanced ratio. Ratio of impostor and genuine pairs was selected randomly between 45:55 and 55:45. Test and validation set consist of 1024 images each. It means 512 pairs are used for testing and validating, where half of them are impostor pairs and another half are genuine pairs.

As mentioned before, stochastic gradient descent was used for training, therefore we need to split data into minibatches, where each minibatch contains 256 pairs.

Our net is trained in 75 000 steps - it means that training process takes cca. 200 epochs. Success rate of training process is shown in Fig. 6.

IV. Verification using CNN

As we can see in Fig.7, value of cost function decreases significantly during 40 000 steps. After that success rate stagnate (Fig. 6). Also success rate of validation set is not improving after step No. 32 301, where we got the best result. Also we can notice, that our net did not overfit, because the increase of validation and training success rate are very similar. Thanks to CNN with Siamese architecture, we reached 85.74 % success rate.

V. Conclusion

Our results confirm our presumption, that Siamese convolutional network is effective for face recognition in uncontrolled conditions like various poses, illumination or overlapping of human faces with other objects. On the other hand, we need to partially prepare our data to processing by normalization or selection of region of interest.

Next improvement of our system can be expected by increasing the size of training dataset or number of training epochs. Another option to increase success rate is improvement of face detection algorithm. Also there are many ways how to change architecture of CNN. For example there is interesting option to use other types of convolutional layer like inception layer or experiment with deeper network. Our next research will aim at Triplets network, where we need to

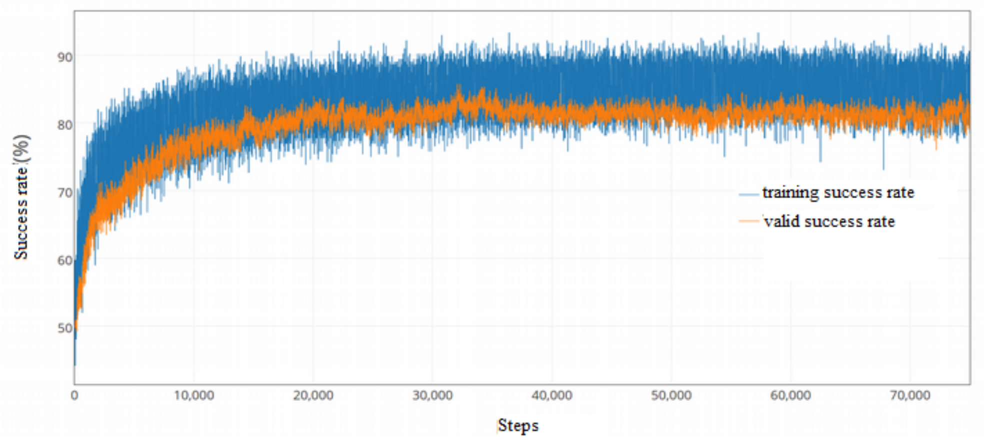

Figure 6: Success rate through training process.

train three identical convolutional networks and we will use three images as input. In this type of network, two images of the same subject (same class) are compared with another sample image from the different class.

Acknowledgment

The research described in the paper was done within the project No. 1/0867/17 of the Slovak Grant Agency VEGA.

References

[1] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Context-aware local binary feature learning for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[2] Y. Xu, Z. Li, J. Yang, and D. Zhang, "A survey of dictionary learning algorithms for face recognition," *IEEE Access*, vol. 5, pp. 8502–8514, 2017.

[3] G. Hassan and K. Elgazzar, "The case of face recognition on mobile devices," in *2016 IEEE Wireless Communications and Networking Conference*, pp. 1–6, April 2016.

[4] M. Oravec, D. Sopiak, V. Jirka, J. Pavlovičová, and M. Budiak, "Clustering algorithms for face recognition based on client-server architecture," in *2015 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 241–244, Sept 2015.

[5] C. Ding and D. Tao, "Trunk-branch ensemble convolutional neural networks for video-based face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[6] Y. H. Kim, H. Kim, S. W. Kim, H. Y. Kim, and S. J. Ko, "Illumination normalisation using convolutional neural network with application to face recognition," *Electronics Letters*, vol. 53, no. 6, pp. 399–401, 2017.

[7] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 378–383, Dec 2016.

[8] M. Khalil-Hani and L. S. Sung, "A convolutional neural network approach for face verification," in *2014 International Conference on High Performance Computing Simulation (HPCS)*, pp. 707–714, July 2014.

[9] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[11] J. Bromley, I. Guyon, Y. LeCun, E. Siickinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," 1994.

[12] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 539–546, IEEE, 2005.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[14] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks.," in *Aistats*, vol. 9, pp. 249–256, 2010.

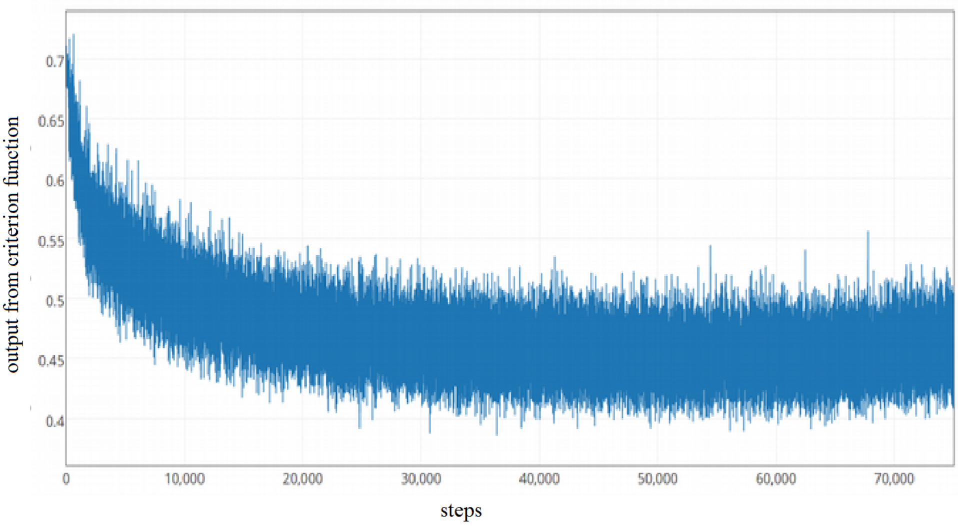

Figure 7: Output of cost function through training process.