POVa: Face verification using CNNs

JánProfant[1]

[1]BUT FIT, Brno

xprofa00@stud.fit.vutbr.cz

1. Introduction

Face recognition in unconstrained images is very actual topic and can be nowadays found even in smart phones used as biometric tool, even instead of fingerprint.

In this paper, we analyze performance of Convolutional Neural Networks, especially ResNet [1] on task of face verification. First, we train our ResNet using cross entropy to recognize person from CASIA WebFace database [1] and then we used embedding, so called face-print for task of face verification. We analyzed two different architectures of CNNs and also analyzed impact of data augmentation.

2. Data Description

We used CASIA-WebFace database containing images from 10,576 people, making it roughly 500,000 images in total. For training, we used only those training examples, which contained at least 15 images per class (person), 5 of them were used for validation and the rest for raining of convolutional neural network, respectively. For face verification we used all data, which were not in training nor validation - classes, which had less than 15 images.

We also used LFW database [2] for testing. As a benchmark for comparison, we report results on cross validation using splits that were randomly generated. [2]

3. Data Preprocessing

We used openface [3] framework developed at Carnegie Mellon University written in python for data preprocessing. We used demo code [3] as example for our data preprocessing. First, we found face's bounding boxes and then we used face alignment from same source. Example of alignment is show in Figure 1.

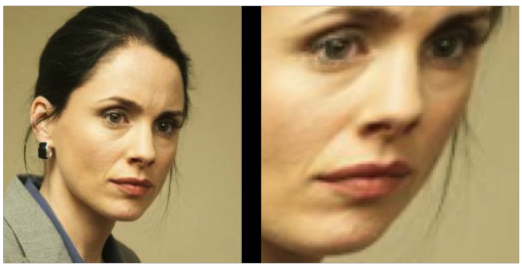

Figure 1: *Image before (left) and after boundary box detecting and alignment (right).*

4. Models Description

We used two variations of Convolutional Neural Networks. Both networks used crossentropy as objective function and ReLU activations. First one (denoted for the rest of the paper as basic) [4] had 11,096,216 parameters with together 24 layers - five blocks with two convolutional layers followed by max pooling layer and dropout - training of model is summarized in Figure 2. Second network was based on ResNet [1] with depth of 83 (280 layers) containing 21,196,920 parameters and 15,584 non-trainable parameters.

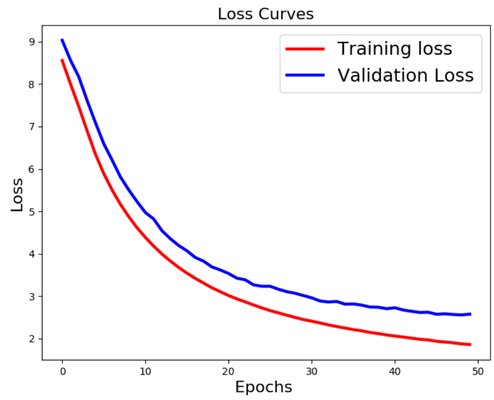

Figure 2: *Loss function (crossentropy) value in 50 epochs of training.*

5. Data Augmentation

One of the major reasons for overfitting is lack of data to train convolutional neural network. Apart from regularization, another very effective way to counter overfitting is data augmentation. It is the process of artificially creating more images from the training set - changing the size, orientation or other properties of the image. Impact of data augmentation is shown in Figure 3. We can see, that data augmentation helped in terms of loss function value on our validation data. We used ImageDataGenerator [5] from Keras for generating augmented examples.

[1] http://www.cbsr.ia.ac.cn/english/CASIA-WebFace-Database.html

[2] http://vis-www.cs.umass.edu/lfw/pairs.txt

[3] https://github.com/cmusatyalab/openface/blob/master/demos/classifier.py

[4] https://www.learnopencv.com/image-classification-using-convolutional-neural-networks-in-keras/

[5] https://keras.io/preprocessing/image/

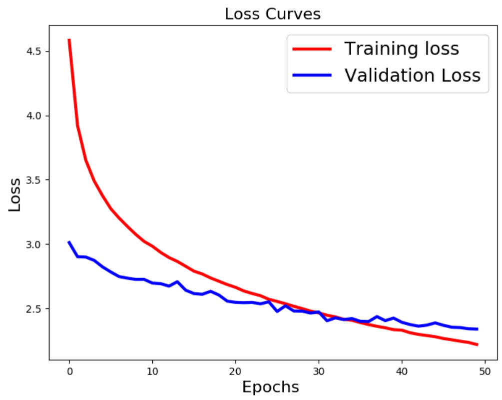

Figure 3: *Loss function (crossentropy) value in 50 epochs of training using augmented data. Model was initialized with pre-trained model which was trained on clean data.*

6. Face Verification

For face verification we used embedding from appropriate layer in convolutional neural network. On top of embedding, we used l2 normalization and cosine similarity for scoring. We processed every pair of embeddings to generate corresponding target and non-target scores.

First, we report results on our subset of CASIA-WebFace database using images from people not seen during training to recognize performance of our models. Results are described in Figure 4. We can see, that Basic architecture outperforms ResNet83 architecture. Data augmentation helped in both cases, but performance increase is more observable in case of ResNet83.

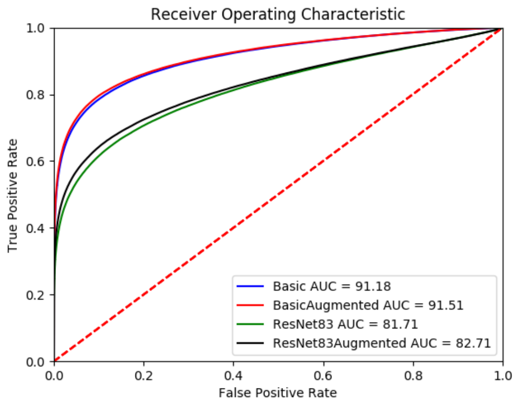

Figure 4

As mention in Section 3, we used face detection and alignment from openface [3], but this model can sometimes return more or less boundary boxes than expected - we always assume, that there is only single person in image. We analyzed impact of face detection and alignment.

In the first scenario, we skipped all images in which there were detected more or less than 1 face and only successful trials were used in evaluation - 4444 pairs were evaluated. Results of this scenario are shown in Figure 5.

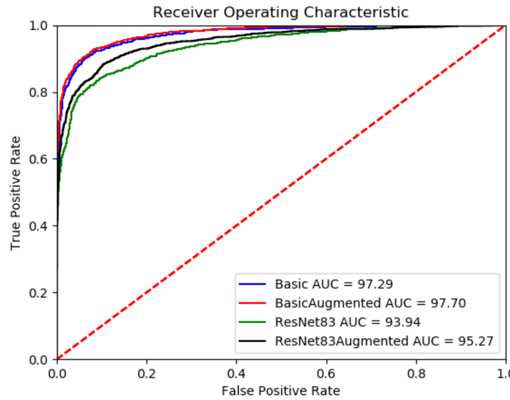

Figure 5: *ROC without problematic trials. Basic model outperforms ResNet83 architecture. Data augmentation helped both architectures to achieve better results.*

In the second scenario, in case of unsuccessful face detection we used whole image for alignment and processed all pairs. ROC curve for this scenario is shown in Figure 6. Based on the results, we can see that first scenario had same trend as second scenario and there is not much difference between these two ROC curves, even in the case, when we failed to process most probably problematic images.

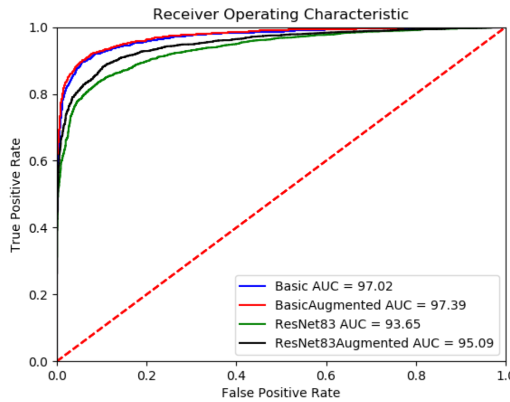

Figure 6

7. Conclusions

In this work, we analyzed performance of two architectures of convolutional neural networks using CASIA-WebFace database for training and LFW dataset for benchmarking. ResNet architecture we used seems to be more difficult to train and even computation expenses are much higher. We analyzed impact of data augmentation in Keras. Results shown, that our approach was not able to get competitive results comparing to approach used in [4]. Results could be improved using more data augmentation and more data in general, especially for ResNet architecture and of course more cautious training.

8. References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on com-*

puter vision and pattern recognition, 2016, pp. 770–778.

[2] 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," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[3] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.

[4] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.