Face Verification: Strategies for Employing Deep Models

Ricardo Barbosa Kloss, Artur Jordão, William Robson Schwartz
Smart Surveillance Interest Group, Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
rbk@dcc.ufmg.br, arturjordao@dcc.ufmg.br, william@dcc.ufmg.br

***Abstract*—Features extracted with deep learning have now achieved state-of-the-art results in many tasks. However, to reuse a learned deep model, transfer learning with fine-tuning needs to be employed, which requires to re-train the whole model or part of it to extract useful features in the new domain. This step is burdensome and requires heavy computing power. Therefore, this work investigates alternatives in transfer-learning that do not involve performing fine-tuning for a model with the new domain. Namely, we explore the correlation of depth and scale in deep models, and look for the layer/scale that yields the best results for the new domain, we also explore metrics for the verification task, using locally connected convolutions to learn distance metrics. Our experiments use a model pre-trained in face identification and adapt it to the face verification task with different data, but still on the face domain. We achieve 96.65% mean accuracy on the Labeled Faces in the Wild dataset and 93.12% mean accuracy on the Youtube Faces dataset comparable to the state-of-the-art.**

***Keywords*-Transfer Learning, Artificial Neural Networks, Face Verification, Metric Learning**

I. Introduction

Images are two dimensional representation of a scene in the world, and are a means for us to record this scene and extract information, such as detect objects, which could be associated with a tool used by a perpetrator of a crime, or recognize a face by verifying that it matches with a valid identity, allowing us to open an automated door, for instance. Machine learning and computer vision are usually employed to tackle these tasks, where neural networks with deep architectures have gained much popularity and have broken many state-of-the-art records.

Since the AlexNet model [1] won the ImageNet 2012 challenge [2], deep learning models [3] have been popularized and made breakthroughs in many areas, such as winning matches of a Go, a difficult game, against world champions [4] and achieving results comparable to humans in face verification [5]. Many architectures have been explored for image tasks since the work of Krizhevsky et al. [1], from Inception [6], which leaves the choice of window size to pick in a convolution layer to the model, to Densely Connected [7] and Resnet [8], which attempts to tackle the vanishing gradient problem by using early layer in the computation of latter ones.

Many models which have achieved great results on ImageNet competitions have also been successfully used in datasets or tasks different than the ones they were originally trained. This is done through *transfer learning*, which is the act of adapting a model learned on one data domain to another, e.g., different task or different labels. In our case, we perform transfer learning for different labels, same task.

The transfer learning usually consists of replacing some of the final layers of a learned model and reconditioning the weights to new data, a process called fine-tunning. The disadvantage of this process is that it requires robust hardware to perform the learning process and although the need of large datasets is diminished, it is still present. Motivated by these limitations, we explore transfer learning approaches that would require little to no fine-tuning. We hypothesize that features associated with each layer of a sequential model tend to be more specific to the learned environment as its depth increases, since a node of a neural network is being built by correlating patterns of patterns. This way, we believe that the penultimate layer might not be ideal to extract features if the new data are characteristically different than the data used to learn the model.

To evaluate our hypothesis, we extract features from different layers and test them with distance metrics to be employed for face verification, a largely explored research topic [5], [9]. The goal of face verification is to determine whether two face images belong to the same individual.

Early works on face verification employed low-level feature extractors to obtain feature representations for face images [10]–[14]. Nowadays, it is more common to learn features instead of engineering them [15], [16]. Ouamane et. al. [14] achieved great results in face verification by using an adaptation of Discriminant Analysis for the weakly labeled case of same/not-same pairs coupled with exponential kernel Regarding Deep learning methods, Sun et al. [15] proposed one of the first deep architecture for the task, it relied on an ensemble of crop specialized convolutional networks. In their work a pair of face images is fed as input to an ensemble of networks. Hu et al. [16] also used deep learning models on face recognition. They employed a siamese network with shared weights to learn a MDML. That is, their network model works as a space transformation where the transformed features are more discriminable.

In contrast to the aforementioned works, we evaluate well-known handcrafted measures such as χ^2, L_1 and L_2 to compare two faces. Moreover, we also use convolutions to learn

978-1-5386-2335-0/18/$31.00

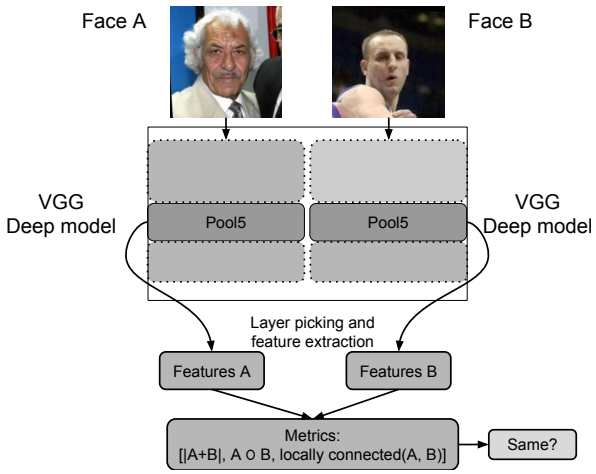

Figure 1. Illustration of our approach for face verification.

a weighted sum of the variables, i.e., the convolution would be able to identify and give higher weights for variables which are more discriminative. Our second hypothesis is that the locally connected convolution [5] would be more suitable for learning a weighted metric than the standard convolution because the variables in the deep features have a spatial independence that can be exploited by the locally connected convolution.

In our experiments, we extract a feature vector of each face with a VGG16 model, learned on the VGGFaces dataset [17]. The learned models are trained using approximately 2.6 million images and, although, additional training is usually employed, e.g., triplet loss embedding or top layer fine-tuning, we neither conduct additional learning nor use additional data, but we still achieve very competitive results.

We perform an experimental evaluation using the labeled faces in the wild dataset [18] and in the Youtube Faces Database [19], which are popular datasets in the literature having been used by many works such as [5], [17], [20]. Our results are comparable to those in the state-of-the-art scenario, with $96.65\% \pm 0.34$ mean accuracy and standard error on the Labeled Faces in the Wild (LFW) dataset and $93.12\% \pm 0.39$ mean accuracy and standard error on the Youtube Faces (YTF) dataset. Moreover, our approach is simple and require a small number of samples.

II. Proposed Approach

Our method, illustrated in Figure 1, can be roughly described as follows. First, we carefully choose a layer from a neural network extracting its output as feature vectors for each face image. Then, the relationship between the two feature vectors is captured according to some distance metric, resulting in a new vector. The metrics employed are listed in Table I, we call them handcrafted because they are not learned through the data. Finally, the resulting feature vector is presented to a classifier, which in our experiments we used the Multilayer Perceptron (MLP) [21].

Table I
Handcrafted distance metrics used. x and y are two vectors and all operations applied are element-wise, that is they also return a vector.

Name	Formula
χ^2	$\frac{(x-y)^2}{x+y}$
L_1	$\lvert x-y \rvert$
$Sign$	$x \times y$
L_2	$(x-y)^2$

To search for complementary information, we conducted experiments with different metrics: i) the χ^2, defined by the square of differences over the sum, suited for histogram comparison and used in the work of Taigman et al. [5] to compare different feature vectors extracted from deep models; ii) the square of differences (L_2), an alternative to taking the absolute value to make it indifferent to the order of the variables, which also accentuates very small and very large values, this last characteristic could convey complementary information to that of the simple absolute value of differences. An overview of the metrics is listed in Table I.

We use a VGG16 architecture [22] learned on the face identification domain to extract features from the LFW and YTF datasets. This architecture consists of 16 convolutional and fully connected ones layers. In general, the features are extracted from the penultimate fully connected layer [5]. However, we experiment with different layers (*pool5*, the last convolutional layer, *fc6*, the first fully connected layer and *fc7*, the penultimate layer), hypothesizing that there is an inverse correlation between generalization and layer depth.

Our hypothesis comes from two clues. The first is the fact that as max-pooling occurs it has the effect of enlarging the receptive field and then capture larger and more complex patterns, usually associated with a higher semantic level. The second is that a node of a neural network layer can be interpreted as a type of correlation to detect patterns. Therefore, with stacked layers, one layer output is in a pattern domain and the next layer will be in a pattern of patterns domain, which we believe increases complexity of the patterns and decreases generalization. The feature vector of a face image extracted by a deep model has each of its variable as the result of a correlation of signals after applying a nonlinear activation. The variable, then, represents the confidence that a given pattern was located by a neural node. In the image space, a two dimensional signal, there is spatial independence of patterns, i.e., an object such as a ball, can be located at different positions, however, in the feature space the variables have more spatial dependency and it is reasonable that if a variable has large value on a sample and low value on the other this is correlated with them being different from one another.

Considering that some variables might be more important than others to discriminate a pair of samples, we experiment to use a weighted sum of the variables where the weights are learned from the data. This is done by means of a (2×1) locally connected convolutional (LCC) filter. These filters, different to standard convolutional filters, are not spatially shared, and for each variable of the pair of samples a different relationship in the form $x_i * w_{i,a} + y_i * w_{i,b}$ is learned, where x_i is the *i-th* variable from the first sample in the pair and y_i is the analogous for the second sample in the pair. We can note that the weights can therefore be learned to ignore a variable from either of the face of one of the pairs, suppose $w_{i,a} = 0$ or $w_{i,b} = 0$, to ignore the relationship altogether, $w_{i,a} = 0$ and $w_{i,b} = 0$, or to give it large importance, $w_{i,a} >> 0$ and $w_{i,b} >> 0$.

With the traditional convolution, the weights of a feature map are shared across different spatial positions. This implies that the weighting for one pair of variables is suitable to all the others. By using more convolutional filters, more weighting relationships are learned. This will generate an output with size $N \times F$, where *N* is the length of the feature vectors and *F* is the number of convolutional filters. In general, this process results in memory problems which can be circumvented with a trick also used in the Inception Module [6]. It consists of employing another convolution with a (1×1) sized filter to compress the channels dimension after performing the layer convolution, resulting in an output of size *N*, the original length of the input vector.

One problem of simply applying the weighted difference between variables is that the weighted sum of the convolutions would have different value depending on the order they are executed, since $a - b \neq b - a$. Thus, to avoid this, the difference is followed by an absolute value layer, making the relationship indifferent to the order that a pair is presented. However, by taking the absolute value the information of the sign of the variables is lost. In the product of variables, *sign* in Table I, the result of the operation will be positive if the operands have the same sign or negative if they have different signs. This apparent complementarity was the motivation that led us to test the sign metric and combine it with other metrics.

III. Experimental Results

In this section, we explain the protocol of our experiments, discuss the face verification task, the models and considered datasets. Then, we present and discuss our results.

A. Experimental Setup

We use the following two datasets in our face verification experiments, mainly for their ubiquity and ease of use. The first is the Labeled Faces in the Wild (LFW) [18], which is the de-facto academic test set for face verification. We follow the standard protocol for unrestricted, labeled outside data and report the mean classification accuracy, and the standard error of the mean. It consists of a 10-fold cross validation protocol, where each fold has 600 samples with balanced labels. The second is the Youtube Faces Database (YTF) [19], a dataset that has gained popularity in face recognition being used by [5] and [20]. The YTF setup is similar to LFW, but instead of verifying pairs of images, pairs of videos are used. It is also a 10-fold protocol, but each fold has 500 label balanced samples. Its images are usually of a smaller resolution than LFW.

Regarding implementation details, when comparing two faces in the verification task, this yields a similarity metric, which is a numeric value. It is then necessary to choose a threshold which is used to classify a pair of samples as belonging to the same identity or not, given this similarity. In our case, the similarity is constrained in the $[0, 1]$ domain, due to the output of a sigmoid neural node and we simply choose the median of the domain as the threshold, i.e., 0.5.

Table II
Accuracy obtained in Labeled Faces in the Wild when using the feature map obtained from layer *pool5* with different metrics. The + sign indicates concatenation of metrics.

Metric Name	Mean Accuracy	Confidence Interval$_{90\%}$
LCC	95.15	[94.51, 95.79]
L_1	90.83	[90.34, 91.32]
$Sign$	90.75	[89.92, 91.58]
$L_1 + Sign$	96.32	[95.78, 96.85]
χ^2	96.39	[95.70, 97.08]
$L1 + Sign +$ LCC	96.48	[95.91, 97.05]

B. Results and Discussion

Table II shows the performance of the different metrics. According to the results, the χ^2 is able to achieve good results when employed alone. However, experimentally we note that its combination with other metrics did not have a positive effect. In comparison, the L_1 and $Sign$ metrics, although showing a lower accuracy, when combined showed an reasonable improvement, indicating that the information of each metric is indeed, complementary to one another. The Locally Connected (LCC) metric also obtained a good result alone. Finally, the combination of $L1 + Sign +$ LCC yielded the best result of the metrics with 96.48% mean accuracy. Other combinations were experimented, omitted due to space limitations.

Regarding our hypothesis that previous layers can provide better results, Tables III and IV reinforce that the complexity and specialization of the model increases with depth, and that is the reason we observe a worse result in the final layer *fc7* on both datasets. For the LFW, we note that the best result was with the *pool5* layer and the result was gradually degrading, this layer is probably the one with most suitable patterns for the LFW and the successive layers are too specific to the VGGFaces domain. This correlation between depth and generalization/specification is probably a

Table III
ACCURACY OBTAINED WHEN USING DIFFERENT FEATURES (LABELED FACES IN THE WILD). $L_1 + Sign$ METRIC USED.

Layer Name	Mean Accuracy	Confidence Interval$_{90\%}$	Rank
pool5	96.32	$[95.78, 96.85]$	1
fc6	94.63	$[93.95, 95.32]$	2
fc7	93.13	$[91.74, 93.69]$	3

Table IV
ACCURACY OBTAINED WHEN USING DIFFERENT FEATURES (YOUTUBE FACES). $L_1 + Sign$ METRIC USED.

Layer Name	Mean Accuracy	Confidence Interval$_{90\%}$	Rank
pool5	91.00	$[89.55, 92.45]$	3
fc6	93.12	$[92.45, 93.79]$	1
fc7	91.54	$[90.67, 92.41]$	2

Table V
STATE-OF-THE-ART COMPARISON (LABELED FACES IN THE WILD).

Method Name	Mean Accuracy	Confidence Interval$_{90\%}$
FaceNet [20]	99.63 ± 0.09	$[99.58, 99.68]$
DeepFace [5]	97.35 ± 0.25	$[96.90, 97.81]$
Ours	96.65 ± 0.34	$[96.03, 97.27]$

Table VI
STATE-OF-THE-ART COMPARISON (YOUTUBE FACES).

Method Name	Mean Accuracy	100% - EER	Confidence Interval$_{90\%}$
FaceNet [20]	95.12 ± 0.39	-	$[94.89, 95.35]$
DeepFace [5]	91.40 ± 1.1	91.4	$[90.76, 92.04]$
VGG [17]	91.6	92.8	-
Ours	93.12 ± 0.34	93.22	$[92.45, 93.79]$

characteristic of purely sequential models such as VGG16 and we believe this corroborates with the hypothesis of Huan et al. [7] that connecting a layer to multiple successive layers is also a form of regularization. We hypothesizes it as being a distribution of complexity across layers of all depths.

Our best result in LFW is presented in Table V. It was obtained by using the metric $L1+Sign+$LCC together with the concatenation of two feature vectors, one obtained from LBP [23] and the other from *pool5* as defined previously. Our result was a mean accuracy of 96.65, and although this number does not outperform the one from the work of Taigman et al. [5], a unpaired T-Test [24] with 90% Confidence shows that the confidence interval of the difference of these two means contains the zero, therefore, they are not statistically different. To our advantage, our results was obtained with a simpler approach, as we do not use of their 3D alignment called frontalization and we also use a model for deep features that was trained on a smaller number of samples. Regarding Schroff et al. [20], our result is statistically inferior, they employ a Triplet Loss network that learns a projection in which same pairs are closer regarding the second order minkowski distance and not-same pairs are far apart. On the other hand, our deep feature models use much fewer samples than them and also do not require a careful selection of the triplet to be presented to the model. Therefore, with much less complexity and computation, careful selection of a layer feature map and simple variable metrics, our approach was statistically equivalent.

In the Youtube Faces, our best result achieved a mean accuracy of 93.12%, as presented in Table VI. Our method is statistically superior to that of Taigman et al. [5], even though we use fewer samples and a simpler face alignment technique. This is a valid example that careful consideration of which layer to use in another dataset can be a valid transfer learning technique.

In Table VI, in the third row, we see that our method, through this careful layer selection, outperformed the results (without triplet-loss embedding), of the model originally trained on the VGGFaces dataset, which we use to extract deep features. That is, by picking a different layer of the model trained on the VGGFaces we obtained better results.

IV. CONCLUSIONS

We presented accurate results on the task of face verification with a simple method of transfer learning, consisting of a careful choice of which layer of a neural network to extract feature maps. Our result outperformed Taigman et al. [5] in one benchmark (YTF) and was statistically equivalent in another (LFW). Their work was the first work to have human comparable results on this task, and they relied on a 3D alignment, frontalization. Our simpler model does no frontalization and also uses fewer samples to learn deep features. Finally, we believe that model fine-tuning, although able to yield improved results is not a hard requirement for transfer learning. In addition, we have showed experimental evidence that it is possible to adapt a model trained on one dataset to another through simple layer selection and with small additional computing cost.

ACKNOWLEDGMENTS

The authors would like to thank the Brazilian National Research Council – CNPq (Grant 311053/2016-5), the Minas Gerais Research Foundation – FAPEMIG (Grants APQ-00567-14, RED-00042-16 and PPM-00540-17) and the Coordination for the Improvement of Higher Education Personnel – CAPES (DeepEyes Project). The authors acknowledge the support of NVIDIA Corporation with the donation of the GeForce Titan X GPU used for this research.

REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[5] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior 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.

[6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[7] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger, "Densely connected convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] R. Chellappa, P. Sinha, and P.J. Phillips, "Face Recognition by Computers and Humans," *Computer*, vol. 43, no. 2, pp. 46–55, 2010.

[10] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-Statistical Model for Face Representation and Recognition," in *IEEE Intl. Conference on Computer Vision*, 2005, pp. 786–791.

[11] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma, "Robust face recognition via sparse representation," *Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[12] L. Wolf, T. Hassner, and I. Maoz, "Face Recognition in Unconstrained Videos with Matched Background Similarity," in *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, 2011, pp. 529–534.

[13] Andrew Wagner, John Wright, Arvind Ganesh, Zihan Zhou, Hossein Mobahi, and Yi Ma, "Toward a practical face recognition system: Robust alignment and illumination by sparse representation," *Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[14] Abdelmalik Ouamane, Messaoud Bengherabi, Abdenour Hadid, and Mohamed Cheriet, "Side-information based exponential discriminant analysis for face verification in the wild," in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*. IEEE, 2015, vol. 2, pp. 1–6.

[15] Yi Sun, Xiaogang Wang, and Xiaoou Tang, "Hybrid deep learning for face verification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1489–1496.

[16] Junlin Hu, Jiwen Lu, and Yap-Peng Tan, "Discriminative deep metric learning for face verification in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1875–1882.

[17] O. M. "Parkhi, A. Vedaldi, and A." Zisserman, ""deep face recognition"," in *"British Machine Vision Conference"*, "2015".

[18] Gary B. Huang, Marwan Mattar, Honglak Lee, and Erik Learned-Miller, "Learning to align from scratch," in *NIPS*, 2012.

[19] Lior Wolf, Tal Hassner, and Itay Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 529–534.

[20] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[21] Simon S Haykin, *Neural networks: a comprehensive foundation*, Tsinghua University Press, 2001.

[22] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[23] Timo Ojala, Matti Pietikainen, and David Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*. IEEE, 1994, vol. 1, pp. 582–585.

[24] Raj Jain, *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*, John Wiley & Sons, 1990.