On the Use of Pre-trained Neural Networks for Different Face Recognition Tasks

Leyanis López-Avila(✉), Yenisel Plasencia-Calaña, Yoanna Martínez-Díaz, and Heydi Méndez-Vázquez

Advanced Technologies Application Center, 7ma A # 21406, Playa, Havana, Cuba
{lelopez,yplasencia,ymartinez,hmendez}@cenatav.co.cu

Abstract. Deep Convolutional Neural Networks (DCNN) are the state-of-the-art in face recognition. In this paper, we study different representations obtained from a pre-trained DCNN, in order to determine the best way in which they can be used in different tasks. In particular, we evaluate the use of intermediate representations independently or combined with a Fisher Vector approach, or with a Bilinear model. From our study, we found that convolutional features may be more suitable than the features obtained from the last fully connected layers for different applications.

Keywords: Convolutional neural networks · Deep learning
Face recognition · Transfer learning

1 Introduction

Face recognition is one of the core problems in computer vision and it has been an active research topic in the last decades [14]. Most of existing methods have shown to work well on images or videos that are collected in controlled scenarios, but their performance often degrades significantly when there are large variations in pose, illumination, expression, aging, cosmetics, and occlusion, among others. To deal with this problem, different works have focused on learning invariant and discriminative representations from face images and videos [12].

In the last years, deep Convolutional Neural Networks (CNN) have demonstrated impressive performances on face recognition and are the state-of-the-art on this an other computer vision tasks [11,13]. CNNs are deep networks designed from the beginning to work with the large amounts of parameters that are fed into a network by an image, which exploit the structure and peculiarities of the input in order to learn discriminative features. To achieve this, they adopt a structure of alternating layers that achieves a reduction in characteristics (weights, parameters and features), and also decreases training times.

CNNs are hard to train because they have many hyperparameters (*e.g.* learning rate, momentum, different types of regularization, activation functions) and the layers of the networks can vary in type, number and width. The best network for a given problem is some combination of all those hyperparameters, so

M. Mendoza and S. Velastín (Eds.): CIARP 2017, LNCS 10657, pp. 356–364, 2018.
https://doi.org/10.1007/978-3-319-75193-1_43

researchers have to go through a vast space of possible combinations to find the best one. To perform all these combinations, great computational capacity is needed. However, once the model is learned, it can be used for other tasks without any further network training. This is beneficial for users who do not have the computer or data resources needed for training.

When using a pre-trained network, the standard choice is to compute the image descriptors at the end of the CNN. In this work, we aim at analyzing the behaviour of different representations obtained from a pre-trained network, in order to determine the most effective way in which the discriminative power of deep learning can be achieved without any training effort, and in different tasks. The adaptation of a trained method for a similar problem is called transfer learning, and the standard approach for CNN features is to retrain a new network starting from the first layers and weights learned by the trained network. Here we avoid retraining and analyze some of the different posibilities that exist for using an already trained network.

2 Using a Pre-trained Network for Face Recognition

There are different CNNs models publicly available in the literature. In this work, we choose the VGG-Face [11], because it is one of the most widely used for face recognition with outstanding results on several datasets, even very near to the best performing ones from commercial systems.

The VGG-Face network [11] has 11 blocks. The first eight blocks contain convolutional layers, while the last three blocks fully connected layers. The input to the network is a face image of size 224×224 with the average face image (computed from the training set) subtracted; and the output of the last convolutional layer, before the linear class predictor and the softmax layers is a 4096 vector descriptor. For every test image, ten 224×224 patches are obtained, by using the center and four corners, and applying horizontal flip to them. To enable multi-scale testing, the face image is first scaled to three different sizes of 256, 384 and 512 pixels, and the cropping procedure is repeated for each of them (see Fig. 1). In total 30 feature vectors are obtained which are averaged for obtaining the final face descriptor. This kind of net descriptor has been used in previous face recognition studies [8]. Besides, we propose to use other representations by using features extracted from not-fully connected blocks. They can be used like a tensor or a vector if we calculate the average from each descriptor in the tensor (see Fig. 1). We are going to refer to these descriptors as cube descriptor and average pooling descriptors respectively. A tensor can be read in different ways, but we use it as in [2], then a tensor is composed by several descriptors with size: $1 \times number\ of\ filters$ in the convolutional layer.

Apart from using a net descriptor directly, they can be used in combination with other methods. Here we will explore the use of Fisher Vector encoded convolutional features (FV-DCNN) [2] and Bilinear Models [3].

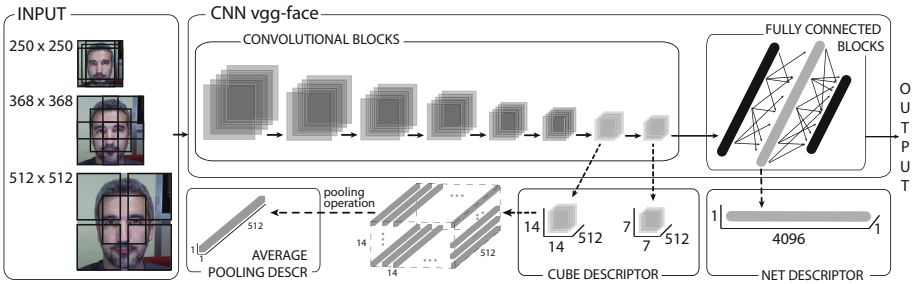

Fig. 1. Schematic view of the VGG-Face representations. The scales and crops of the face images used as input for the network are shown in the left and the convolutional blocks that were used for obtaining the intermediate descriptors are enlarged showing their dimensions.

2.1 FV-DCNN

The Fisher Vector (FV) method [12] has shown competitive performance on unconstrained face recognition. The original FV representation is obtained by densely computing local descriptors (e.g., SIFT), at different scales, which are aggregated into a high-dimensional vector by assuming a parametric generative model for the data and stacking the derivatives of its log-likelihood with respect to all its parameters. Usually, a Gaussian Mixture Model (GMM) with parameters $\theta = \{\pi_1, \mu_1, \sigma_1, ..., \pi_k, \mu_k, \sigma_k\}$ denoting the weight, mean vector and diagonal covariances of the K mixture components, is assumed as generative model. Let $S = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N\} \in \mathbb{R}^Q$ be a set of local descriptors. The FV representation of S based on the GMM is given by $F = \left(\phi_1^{(1)}, \phi_1^{(2)}, ..., \phi_K^{(1)}, \phi_K^{(2)}\right)$ where the entries $\phi_{ik}^{(1)}$ and $\phi_{ik}^{(2)}$, $i = 1, ..., Q$, of the vectors $\phi_k^{(1)}$, and $\phi_k^{(2)}$ are defined as follows:

$$\phi_{ik}^{(1)} = \frac{1}{N\sqrt{\pi_k}} \sum_{n=1}^{N} \alpha_k(\mathbf{s}_n) \left(\frac{s_{in} - \mu_{ik}}{\sigma_{ik}} \right), \tag{1}$$

$$\phi_{ik}^{(2)} = \frac{1}{N\sqrt{2\pi_k}} \sum_{n=1}^{N} \alpha_k(\mathbf{s}_n) \left[\left(\frac{s_{in} - \mu_{ik}}{\sigma_{ik}} \right)^2 - 1 \right], \tag{2}$$

where $\alpha_k(\mathbf{s}_n)$ is the soft assignment of $\mathbf{s}_n$ to component k. Finally, the FV is further improved by applying power normalization followed by l_2 normalization.

Recently, it was proposed in [2] to encode deep convolutional features with FV approach for unconstrained face verification. Based on this work, we extract the CNN descriptors from the last convolutional block of the VGG-Face network as described previously and encode them by applying FV. In order to meet the assumption of diagonal covariances for GMM, all descriptors are decorrelated by using PCA before feeding into FV encoding.

2.2 Bilinear Model

Another proposal in the literature for the use of trained networks is the bilinear model, initially proposed for image classification [7] and later used for face recognition [3]. Bilinear combination of two or more feature extractors allows considering extra information from pairwise interactions between initial features, which can contribute to increase discriminative ability in classification. This kind of mixing model combines the advantages of previous approaches for fine-grained recognition tasks, such as part-based and texture based models [7].

A bilinear model is defined in [7] as a quadruple $B = (f_A, f_B, P, C)$. Here f_A and f_B are feature functions, P is a pooling function and C is a classification function. A feature function is a mapping $f : L \times I \rightarrow R^{c \times D}$ that takes an image I and a location l and outputs a feature of size $c \times D$. A bilinear combination, for I and l is given by $bilinear(l, I, f_A, f_B) = f_A(l, I)^T * f_B(l, I)$. In this work we are going to combine three feature functions: A pre-trained CNN VGG-Face truncated at the end of convolutional blocks (as is proposed in [3] for face recognition), and the same descriptors but projected with PCA and FV methods.

3 Using a Pre-trained Network for Heterogeneous Face Recognition

An interesting application where the use of pre-trained models might be beneficial is for heterogeneous face recognition. It encompasses several modalities, and also in this case, state-of-the-art performance is achieved by deep learning approaches [6,10] with complex networks and training strategies. Our intuition is that the information contained in the pre-trained model can be useful to tackle these problems without having to retrain it or use a transfer learning approach.

In our research we take face-sketch recognition as a case of study, but we believe the same can be applied for different heterogeneous problems. Recently, a transfer learning approach was proposed in [9] for this task, in which a deep believe network is re-trained with sketch-image pairs, using as the initial weights, the ones provided by the trained model. Instead, we may use less computationally expensive approach on top of the deep features, by carefully selecting the appropriate information to be extracted from the trained model. We start from the convolutional features obtained from the network as discussed in the previous section, and improve them by metric learning which is very fast to train.

Metric Learning: The main goal of metric learning is to learn the weights related to a Mahalanobis distance of the form: $(x - y)^t M(x - y)$, where the matrix M is positive-semidefinite, while x and y are objects from a dataset. This is learned in such a way that the discriminative information for the problem is emphasized. The main idea is to converge to weights that bring objects belonging to the same class closer while pulling objects from different classes apart. These weights can be incorporated directly into the vectorial representations and the Euclidean distance can be computed on top of them. Here we use Linear Discriminant Analysis (LDA) [1], which can be seen as a type of metric

learning method which learns a projection such that it maximizes the between or inter-class scatter over the within or intra-class scatter. This is solved using a closed-form expression based on a generalized eigenproblem.

4 Experimental Analysis

4.1 Performance Evaluation for Face Recognition

In order to evaluate and compare the different descriptors, we conducted experiments on the Labeled Face in the Wild (LFW) database [5]. It contains 13233 face images from 5749 people taken from Yahoo! News, with different variations in pose, scale, clothing, expression, focus, resolution among others. Since we aim at not training the network, we use the closed set protocol, where the dataset is divided in ten splits, with 300 genuines and 300 impostor comparisons for each of them. We used the aligned version of the images (LFW-a). The Euclidean distance and Cosine similarity were used as distance measures. We test the different descriptors that were described in Sect. 2. In particular we evaluate the original net descriptor from the 11th block of the network (b11) and the two different descriptors obtained from the third convolutional layer of the 8th block (b8): the concatenated cube descriptor (conc) and the average pooling descriptor (avg). We also test the FV-DCNN representation and the different bilinear combinations. In all cases we evaluate also the benefits of applying Principal Components Analysis (PCA) and L2 vector normalization.

Table 1 presents the obtained Equal Error Rates (EER) and the False Rejection Rate for a fixed False Acceptance Rate of 0.1% (FRR@0.1). The best results for every configuration is highlighted in bold. It can be seen that the original net descriptor (b11-norm) using Euclidean distance achieves an EER of 4.70% and all the different configurations evaluated outperform this result. This suggests that when using a pre-trained network for a given task, intermediate representations might achieve better results. It is also interesting to note that usually the feature vectors obtained from CNN networks are normalized and using intermediate representations we obtained better results without normalization. Besides, by applying PCA and using for classification vectors of only 20 or even 10 dimensions, results are equal to or better than the results found when using representations of higher dimensionalities. By using more complex representations such as FV-DCNN and Bilinear models good results are obtained but not better than simple intermediate representations from the network. As it was suggested in [7], for this combined configurations new training is needed.

4.2 Performance Evaluation for Face Sketch Recognition

From the analysis presented in the previous section, we consider that the average pooling descriptor from the third convolutional layer in block 8 (b8-avg) provides the best compromise between accuracy and efficiency; therefore we will use it

Table 1. EER (%) and FRR@0.1 (%) for different descriptors in LFW database.

Representation	Dim	Euclidean		Cosine	
		EER	FRR	EER	FRR
VGG-FACE					
b11	4096	15.3	54.2	5.10	27.7
b11-norm	**4096**	**4.70**	**25.7**	**4.60**	**25.7**
b11-PCA20	20	9.40	53.7	6.70	44.5
b11-PCA20-norm	20	6.30	43.1	59.3	44.2
b8-conc	$7 \times 7 \times 512$	0.80	8.00	4.30	43.4
b8-conc-norm	$7 \times 7 \times 512$	4.40	35.5	4.70	64.5
b8-PCA30	30×49	0.80	7.80	1.30	19.3
b8-PCA30-norm	30×49	4.10	34.7	4.10	36.7
(b8-PCA30)-conc-PCA20	**20**	**0.80**	**7.80**	**4.00**	**30.0**
(b8-PCA30-norm)-conc-PCA20	20	3.80	33.8	2.80	29.3
b8-avg	**512**	**0.80**	**7.80**	**4.30**	**46.5**
b8-avg-norm	512	4.20	35.1	4.20	56.4
b8-avg-PCA10	10	0.90	7.40	4.30	35.0
b8-avg-PCA10-norm	10	0.70	9.30	3.70	34.5
FV-DCNN					
fv-dcnn	3840	2.20	30.4	14.3	20.0
fv-dcnn-norm	3840	3.60	34.7	41.7	46.2
fv-dcnn-PCA30	**30**	**1.20**	**18.9**	**1.30**	**19.3**
fv-dcnn-norm-PCA30	30	63.8	100	64.8	100
BILINEAR					
bl-(b8)+(b8)	$\mathbf{512 \times 512}$	**3.30**	**61.2**	**4.30**	**39.9**
bl-(b8-norm)+(b8-norm)	512×512	4.20	35.1	4.20	70.5
bl-(b8-PCA20)+(b8-PCA20)	20×20	1.60	26.8	4.00	28.7
bl-(b8-PCA20-norm)+(b8-PCA20-norm)	20×20	5.70	39.4	2.80	29.2
bl-(fv-dcnn-PCA30)+(b8-PCA20)	$\mathbf{30 \times 20}$	**1.80**	**23.7**	**1.20**	**19.6**
bl-(fv-dcnn-norm-PCA30)+(b8-PCA20-norm)	30×20	4.30	34.3	2.90	30.7

for the experiments on face sketch recognition. Our goal here is to study the performance of these descriptors in a different problem for which the network was not trained for.

For the face sketch recognition evaluation we used the PRIP Viewed Software-Generated Composite (PRIP-VSGC) [4] dataset (see Fig. 2), which was created from photographs from 123 subjects from the AR database and three composites created for each subject using FACES (American and Asian users) and Identi-Kit softwares. Both mug-shots and sketches were normalized to the size required by

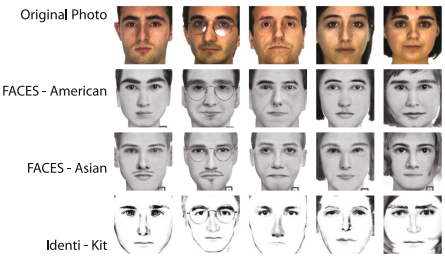

Fig. 2. Images from the PRIP-VSGC database.

Table 2. Recognition rate (%) at Rank-10 in PRIP-VSGC for sketch vs. image

Representation	Asian	American	Identi-Kit
Mittal [9]	48.10	56.00	**52.08**
b8-avg	43.90	61.78	17.88
b8-avg+LDA	46.34	**67.47**	17.88
b8-c2-avg	27.64	51.21	13.00
b8-c2-avg+LDA	51.21	64.22	26.82
b7-c2-avg	21.95	33.33	16.26
b7-c2-avg+LDA	**52.84**	63.41	42.27

the VGG-Face network. With the aim of comparison, we used the experimental protocol in [9], which takes as gallery the 123 subjects mugshots and as probe the sketches. Since in this case the image modality is very different from the original training of the network, we also evaluate previous layers representations, i.e.: the second layer from the same 8th block (b8-c2-avg) and the second layer from the 7th block (b7-c2-avg). The obtained results for each descriptor with and without metric learning are depicted in Table 2. It can be seen in Table 2 that in general the performance is improved by using lower layers, except for the American users, in which the sketches are more similar to the original images. Due to this higher resemblance to real photos, this can be expected, since for higher layers the network is more specialized. When using lower layers, the improvement is particularly larger for the Identi-Kit software, where the sketches are more different from real face images, they seem more like a drawing. Therefore, the lower layers which are less specialized for face images than the upper layers, are able to represent better the different face modalities. In general it is corroborated that the use of a metric learning is of particular importance since it is able of emphasize the discriminative information from the descriptors. The results are comparable with the ones obtained in [9], where a more specialized approach specifically trained for face sketch recognition was used. Besides, it is highly convenient that the intrinsic dimensionality achieved by the selected convolutional features is very low, and therefore the final representations can be very compact with a low memory footprint.

5 Conclusions

In this paper we study different approaches to exploit the information provided by a pre-trained convolutional network as well as its application for other domains with some similarities to the one for which the network was trained for. From our analysis, we found that the best performing features from the network were obtained with an average pooling descriptor from the last convolutional block. This representation is more compact and by applying PCA a very low dimensional representation can be obtained maintaining a high discriminative power. This can be useful for specific applications such as large scale face recognition. Besides, we found that by using intermediate representations vector normalization does not provide good results. In the case of heterogeneous face recognition we found that the pre-trained network performs similar to state-of-the-art approaches, after learning a metric for the specific problem with the provided features. We also found that lower blocks are better for problems where the imaging modality is more different from the one used for training the network. The explanation for this is that in top blocks the network is more specialized for the problem that it was trained for.

References

1. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 711–720 (1997)
2. Chen, J.C., Zheng, J., Patel, V.M., Chellappa, R.: Fisher vector encoded deep convolutional features for unconstrained face verification. In: ICIP, pp. 2981–2985. IEEE (2016)
3. Chowdhury, A.R., Lin, T.Y., Maji, S., Learned-Miller, E.G.: One-to-many face recognition with bilinear CNNs. In: WACV, pp. 1–9. IEEE Computer Society (2016)
4. Han, H., Klare, B., Bonnen, K., Jain, A.K.: Matching composite sketches to face photos: a component-based approach. IEEE Trans. Inf. Forensics and Secur. **8**(1), 191–204 (2013)
5. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. Technical report, Technical Report 07–49, University of Massachusetts, Amherst (2007)
6. Jin, Y., Lu, J., Ruan, Q.: Coupled discriminative feature learning for heterogeneous face recognition. IEEE Trans. Inf. Forensics Secur. **10**(3), 640–652 (2015)
7. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear CNN models for fine-grained visual recognition. CoRR abs/1504.07889 (2015)
8. Mehdipour Ghazi, M., Kemal Ekenel, H.: A comprehensive analysis of deep learning based representation for face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 34–41 (2016)
9. Mittal, P., Vatsa, M., Singh, R.: Composite sketch recognition via deep network - a transfer learning approach. In: International Conference on Biometrics, ICB 2015, Phuket, Thailand, 19–22 May 2015, pp. 251–256 (2015)
10. Ouyang, S., Hospedales, T., Song, Y.Z., Li, X., Loy, C.C., Wang, X.: A survey on heterogeneous face recognition. Image Vis. Comput. **56**(C), 28–48 (2016)

11. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: BMVC, vol. 1, p. 6 (2015)
12. Simonyan, K., Parkhi, O.M., Vedaldi, A., Zisserman, A.: Fisher vector faces in the wild. In: BMVC, vol. 2, p. 4 (2013)
13. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–1708 (2014)
14. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: a literature survey. ACM Comput. Surv. **35**(4), 399–458 (2003)