Exploring Features and Attributes in Deep Face Recognition Using Visualization Techniques

Yaoyao Zhong, Weihong Deng*
School of Information and Communication Engineering,
Beijing University of Posts and Telecommunications, Beijing, China
Email: {zhongyaoyao, whdeng}@bupt.edu.cn

***Abstract*— Deep convolutional neural networks (CNNs) currently have achieved state-of-the-art results on face recognition; yet, the understanding behind the success of the deep face models is still lacking. In particular, it is still unclear the inner workings of deep face models. What effective features do deep face models learn? What do these features represent and what are the sematic meanings of them? This work explores this problem by analyzing the classic network VGGFace using deep visualization techniques. We first explore features computed by neurons, investigating characters of features like diversity, invariance, discrimination. It's worth noting that the middle layer is the least robust to transform, which contradicts the conventional view that robustness to transform increases as the network going deeper. The most significant phenomenon we find is that high level features are correspond with complex face attributes which human could not describe using a few words. We present a quantitative analysis on these face attributes perceived by deep CNNs, understanding them and the complex relationships between them. Additionally, we also focus on the significant point, the pose invariance in face recognition. Our research is the first work to understand the inner works of deep face models, elucidating some particular phenomena in deep face recognition.**

I. INTRODUCTION

Deep convolutional neural networks (CNNs) have demonstrated dramatic improvements over traditional approaches in various visual recognition tasks, such as object classification [1] [2] and face recognition [3] [4].

With the processing units (GPUs), massive annotated data and advanced algorithms, face recognition develops rapidly. Today some face recognition techniques go out of laboratories and come into our daily life [26]. The recognition system may make correct decisions most of the time. However, the lack of interpretability could leave users confusing when systems make a few mistakes. This motivates us to make the face model more transparent and predictable.

It is of vital importance to understand why the system make a decision and which part of the image contributes most to the prediction. Besides, the internal explanation and the inner workings is another significant research point. That is, what do features in different layers of CNNs represent in input RGB space? What is the sematic meaning of them? How much information a neuron of different layers detects? Scientists have already make a few preliminary attempts to explain this "black-box" in some aspects, which will be discussed in section II.

978-1-7281-0089-0/19/$31.00

There are some preliminary insights into CNNs trained for classification tasks. Some methods have been used in explanation in applications of CNN-based classification tasks such as medical diagnosis [11] and expression recognition [12]. However, there is few works focusing on explanation of deep networks trained for face recognition.

The particularity of the face model should be considered mainly in two aspects. On one hand, the deep network trained for face recognition is more like a feature extractor as the trained class is limited compared with countless human in the word. Thus it would make no sense to analyze the classifier output as we do in object assignment. On the other hand, face recognition is a kind of fine-grained recognition. The network is trained in a distinct way using novel architectures and particular loss functions, for the reason that, in face recognition, intra-personal variations could be larger than inter-personal differences due to different poses, illuminations, expressions, ages, and occlusions.

We consider the interpretation of face model towards its particularity in two aspects accordingly. First, we concentrate more on semantic information of features of face model, other than classifier output as we do in object classification. In addition, despite robustness to pose variation has improved in some degree in the deep learning period compared with the handcrafted feature period, large pose variation still needs further research and exploration. We dedicate to understanding how deep face models deal with poses.

Towards better understanding of DCNNs trained for face recognition, Inspired by the mathematical model [19], we push visualization technique proposed by [5] one step further using t-sne [7] by taking diverse face attributes into consideration, analyzing semantic information of features. We discover the relations between features in different layers and face attributes, promoting comprehension of the diverse face attributes of a neuron.

To be specific, we start with the typical architecture VGGFace [3] and explore features which a neuron compute, investigating characters of features like diversity, invariance and discrimination. We also pay attention to the pose invariance of CNNs and design an experiment to understand and evaluate the robustness to pose variation. We then focus on the face attributes perceived by deep CNNs, understanding these complex face attributes and analyzing the relationship of different attributes. The code is available at https://github.com/zhongyy/interpretable-face.

II. RELATED WORK

It is quite difficult to understand CNNs exactly because we could not use functions to describe particular, trained CNNs due to the large number of interacting, nonlinear parts. Nevertheless, a number of previous works have explained CNNs predictions and interpreted concepts learned by CNNs in terms of object classification in a visual way. According to [8], "understanding" refers to explanation and interpretation.

An "explanation" is the collection of features of the interpretable domain, that have contributed for a given example to produce a decision. In terms of object classification, it is of vital importance to understand classification decisions. For a given input sample, what makes it representative for a specific concept encoded by CNNs, could be evaluated by the relevance/sensitivity/influence matrix of the same size as the input image, indicating that how much each pixel contributes to the score of the concept.

Some gradient-based methods calculate gradients to assign scores and generate heatmaps that reflect the influence of an individual pixel on the final classification. This technique is easy to implement for CNNs since the gradient can be computed by backpropagation. Simonyan et al. [9] back propagate a class score to the input data layer, visualizing image-specific class saliency map. Zeiler et al. [5] use Deconvolutional networks to visualize which patterns from the training set activate the feature map, which is essentially a modification of gradient-based method. In order to obtain a reconstruction conditioned on an input image from the network without pooling, guided Backpropagation [10] was proposed, making modifications to the raw gradient to generate influence map, which made significant improvement compared to Vanilla Backpropagation [9].

Apart from gradient-based methods, some methods localize discriminative regions for an output category to understand classification decisions. Zhou et al. [13] proposed a technique called Class Activation Mapping (CAM) that replace fully-connected layers with an average pooling layer that could help highlight task relevant regions. Grad-CAM [14] and Grad-CAM++ [15] improve CAM method by making modifications to the weight coefficient of features of the last convolutional layer. Bazzani et al. [16] propose a technique for self-taught object localization by analyzing the change in the recognition scores when artificially masking out different regions of the image.

An "interpretation" is the mapping of an abstract concept into a domain that the human can make sense of. The complex nonlinear mapping from the RGB input space to the output feature space makes neurons in high layer difficult for human to make sense of. This research focus is to build a prototype in the input space that is interpretable and representative of the abstract concept in high layers.

Activation maximization is an analysis framework to obtain prototypes of abstract concepts by searching for an input pattern that maximizes the response for a neuron. Activation maximization was proposed in [17] first to generate what a neuron computes in arbitrary layer in Stacked Denoising Autoencoders and Deep Belief Networks, trained on MNIST. Simonyan et al. [9] apply the method of [17] to the visualization of ImageNet classification ConvNets for the first time. [18] improve the recognizability of the generated images by incorporating natural image priors into optimization. Nguyen et al. [6] starts optimization from different centers of clustering natural images, revealing the different facets of each neuron and improve the quality of synthesized images.

The deep network trained for face recognition is particular. Previous works explore the peculiarity of a face [23] or the difference of a face pair [24], which both pay attention to the decision-making of the face model. Instead, we focus on the inner works of them, in which features in different layers, the corresponding face attributes and other particular phenomena of neurons in face recognition.

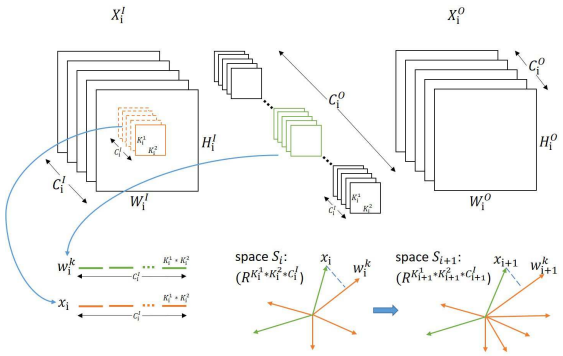

Fig. 1: The convolution process. Based on the analysis in the article, CNN could be considered as a progressively nonlinear space adaption from S^0 to S^n(n is the number of convolutional layers).

III. APPROACH

It is well known that signal convolution could also be seen as signal correlation or projection. For the convolutional layer i, let X_i^I denote the input matrix, the size of which is $W_i^I * H_i^I * C_i^I$. Let X_i^O be the output matrix, the size of which is $W_i^O * H_i^O * C_i^O$. The size of convolutional kernel W_i^I is $C_i^I * k_i^1 * k_i^2 * C_i^O$. In the process of convolutional computation, by using dot product, the correlation between a kernel vector $w_i^k \in R^{C_i^I * k_i^1 * k_i^2}$ (a kernel in channel k, layer i) and a input patch $x_i \in R^{C_i^I * k_i^1 * k_i^2}$ which is cropped from X_i^I. We name space $R^{C_i^I * k_i^1 * k_i^2}$ space S_i. The correlation value ${w_i^k}^T x_i$ is actually the result a neuron computes, that is to say, what the neuron k in layer i detects in space i is some vector who has high correlation with the kernel vector w_i^k.

Rethink about ${w_i^k}^T x_i$. See it as the projection from x_i to a new space-axis w_i^k, all the projection results are the input data to the next layer, where corresponding to a new space S_{i+1}, with new detectors w_{i+1}^k. Apart from the convolutional layer, ReLU adds nonlinear transformation to the space adaption and pooling layer is a kind of linear space reduction. So CNN could be considered as a progressively nonlinear space adaption from S_0 to S_n (n is the number of convolutional layers). The convolution process is shown in Fig. 1.

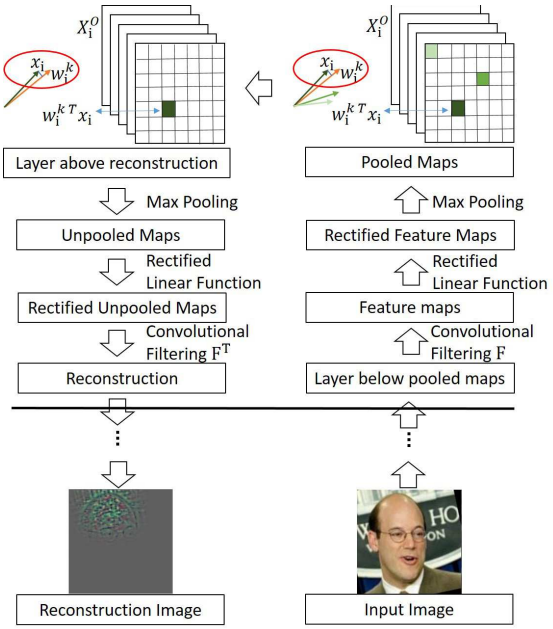

Fig. 2: To visualize a convnet, a deconvnet is attached to each of its layers, providing a continuous path back to the image pixels. To reconstruct a sample x^i which has big projection values on w_k^i, we set all other projection values to zero, pass the corresponding input image to the original convnet and pass the feature maps with this projection value $w_k^{i\,T} x^i$ to the attatched deconvnet layer.

We dedicate to finding a neuron detects (what w_i^k detects) in the input space that the human could make sense of. However, finding a neuron detects in space S_0 is not equal to finding it in space S_i. Because what w_i^k maps to S_0 is difficult to imagine after series of nonlinear space adaption. Nevertheless, the correlation function is unimodal so that we can find some samples in space S_i which has big projection values on w_i^k, and reconstruct these samples to the input space using the technique proposed by [5].

To visualize a convnet, a deconvnet is attached to each of its layers, providing a continuous path back to the image pixels. To reconstruct a sample x_i which has big projection values on w_i^k, we set all other projection values to zero, pass the corresponding input image to the original convnet and pass the feature maps with this projection value $w_i^{k\,T} x_i$ to the attatched deconvnet layer. The process is shown in Fig. 2. By doing so, we get a reconstruction which resembles a small piece of the original input image, with structures weighted according to their contribution to the projection value $w_i^{k\,T} x_i$. We search for samples with top n projection value (top n set) and reconstruct them to explore what a neuron find.

In object classification networks, research [6] reveals that neurons at all levels are multifaceted in object classification convnet. We also consider the possibility of multifaceted features in face model. Since various input patterns may converge into w_i^k after progressively nonlinear space adaption. We make an assumption that degree of dispersion of the samples in space S_i reflects the diversity of their reconstructions in some degree. So we conduct t-sne method [7] to the samples in space S_i. By running t-sne visualization on x_i of the top n set to produce a 2-D embedding, we can visualize the reconstructions for better understanding the diversity of a neuron.

IV. Experiment

A. Feature visualization

Considering the running time of searching, we pick 5000 face images of 500 individuals in VGGFace2 dataset [25] randomly and use this sub-dataset in the visualization experiment. We apply the visualization technique to the typical architecture VGGFace [3] and the result is shown in Fig. 3. For layer pool1-pool5, we show the top50 samples of a neuron in the dataset, projected down to pixel space using the deconvnet approach. The reconstructions are patterns from the dataset causing high activations in the given neuron. For each sample of a neuron, we show the corresponding image patch and the projection. Projections from each layer show the hierarchical nature of the features in the network. And the t-sne method guarantees that similar-looking samples of a neuron gather together in Fig. 3.

In Fig. 3, layer pool1 and pool2 responds to low level features like edge and color, which is similar to object classification networks. Layer pool3 has more complex invariances, capturing mid level features of similar shape (e.g. eyebrow (Row 1, Col 2); red lips (Row 2, Col 2)). Layer pool4 shows significant variation, a neuron captures some similar facial regions (e.g. ear region (Row 2, Col 2); nose region (Row 2, Col 3)). A neuron in layer 5 extracts high level features and covers most part of a face image, capturing some complex facial attributes (e.g. bald head (Row 1, Col 1); the black race (Row 1, Col 3); whisker (Row 1, Col 4)). This phenomenon is significant and particular in face model. We will present a quantitative analysis on these face attributes perceived by deep CNNs, understanding them and the complex relationships between them in section IV-B

Feature diversity: As we see in Fig. 3, high level features are more complex. Besides, the t-sne method is deployed to x^i so that similar input patterns are gathered together for observing feature diversity of a neuron. We find that neurons in higher layers may detect more than one face input pattern (e.g. eyebrow and nose (Row 1, Col 1 in layer pool4); eyebrow and teeth (Row 1, Col 2 in layer pool4); whisker and forehead (Row 1, Col 4 in layer pool5); bangs and bald (Row 2, Col 2 in layer pool5)). This experiment may reveal a phenomenon of neuron reuse.

Feature invariance: We explore the invariance of features in different layers. Fig. 4 shows 5 sample images being vertical translated, rotated and scaled by varying degrees while observing the changes in the feature vectors from the top and bottom layers of the model, relative to the untransformed feature. In face verification, we usually use cosine distance of features to measure a face pair. And in this architecture, layer fc7 is usually used as the deep feature extractor.

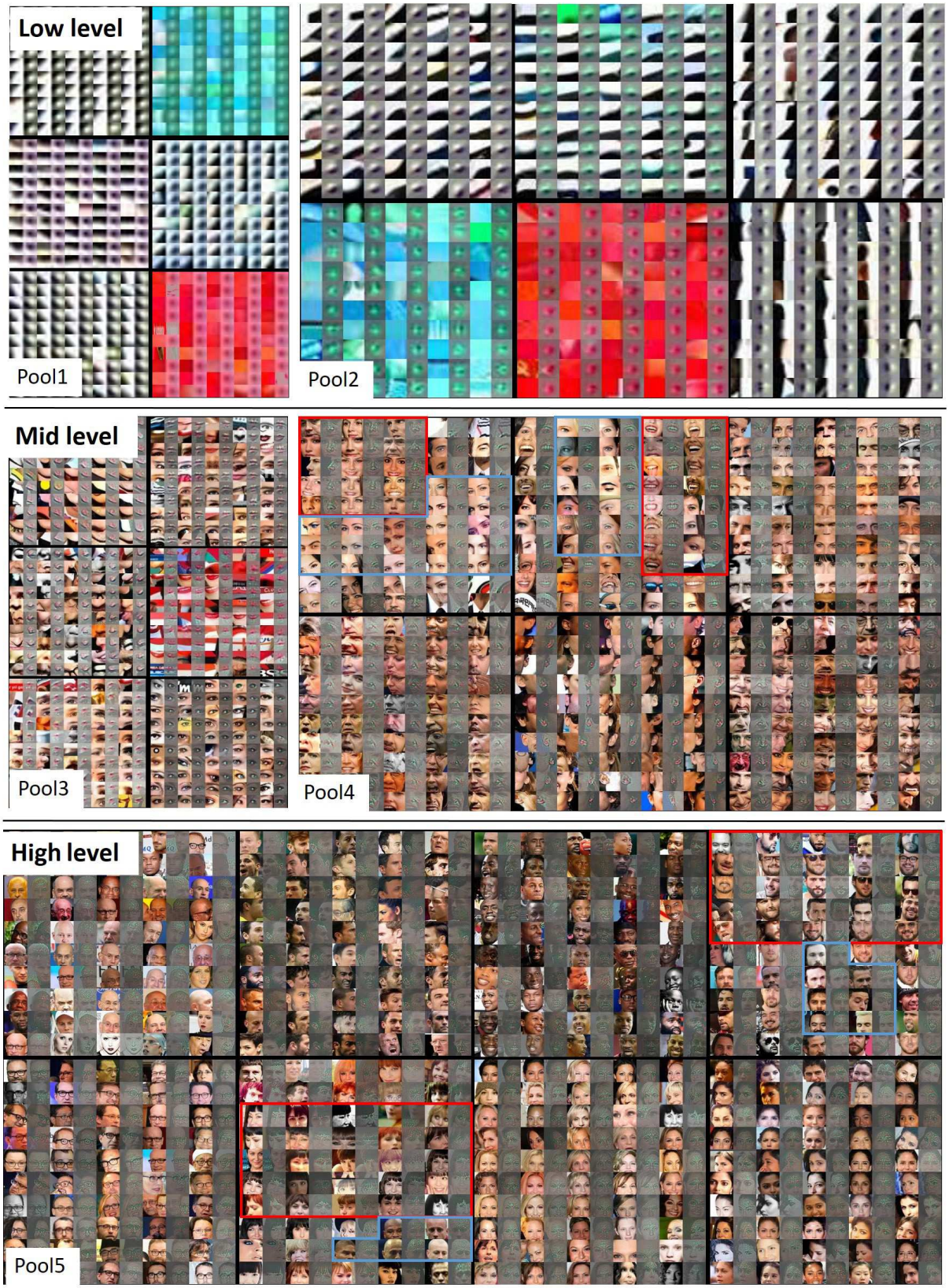

Fig. 3: Visualization of features in a fully trained model VGGFace. For layer pool1-pool5 we show the top 50 activations in the dataset we pick, projected down to pixel space using the deconvnet approach. The reconstructions are patterns from the dataset that cause high activations in the given neuron. For each neuron we also show the corresponding image patches. Best viewed in electronic form.

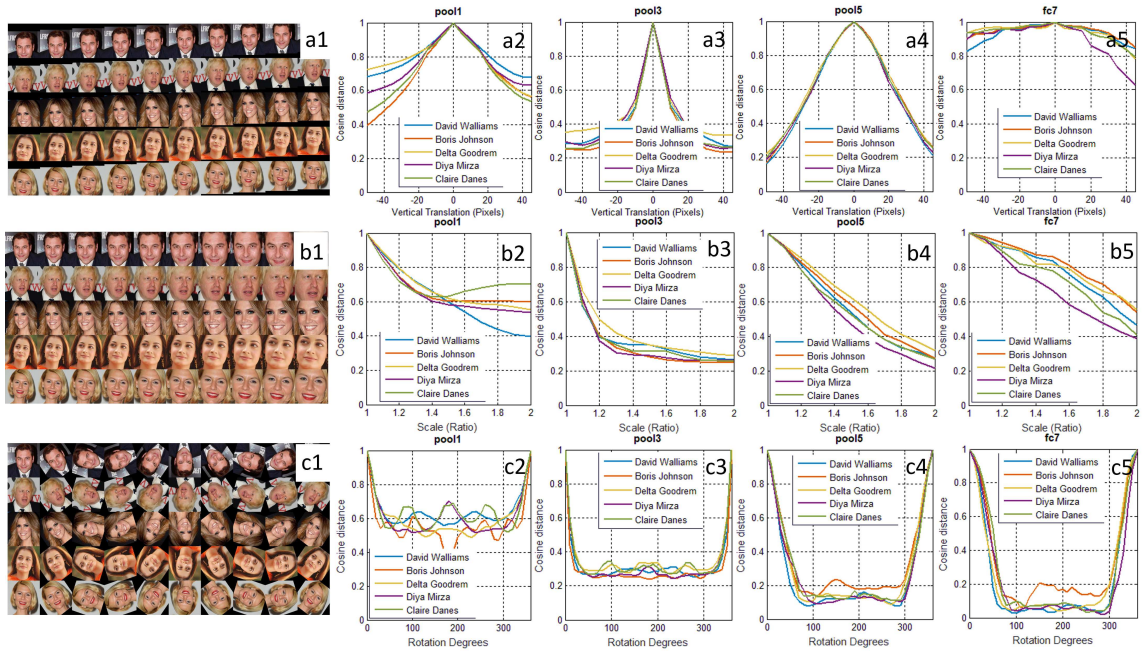

Fig. 4: Analysis of vertical translation, scale, and rotation invariance(rows a-c respectively). Col 1: 5 example images undergoing the transformations. Col 2- Col 4: Cosine distance between feature vectors from the original and transformed images in layer pool1, pool3 and pool5 respectively. Col 5: Cosine distance between feature vectors from the original and transformed images in layer fc7. Layer fc7 is usually used as the deep feature extractor in face verification.

From Fig. 4, we can see small transformations affect the middle layer (pool3) of the model dramatically, while they could have less influence on a lower and higher layer (pool1, pool5 and Fc7). In other words, there is the lowest robustness in the middle layer, which is quite different from the past view [5], [27], [28] that robustness to transform increases as the network going deeper. Besides, we find the network output (fc7) is stable to translations and scalings to a certain degree, while not invariant to rotation.

Feature discrimination: Apart from the complexity and diversity, we explore how discriminative features in each layer of our model are. Face verification computes one to one similarity to determine if two face images are of the same subject. Here we use the standard testbed database Labeled Faces in the Wild (LFW) [20] and Cross-Pose LFW (CPLFW) [21] to test the features of different layers in the model.

The result is shown in Table II. Accuracy on LFW and CPLFW increases as the layers ascend. The result reveals that as the feature hierarchies become deeper, they are more increasingly discriminative. Besides, lower accuracy on CPLFW reveals that cross-pose face recognition is still a challenge in face recognition.

Correspondence Analysis: It seems that deep face recognition differs from classic methods in that there exists no explicit mechanism for establishing correspondence between specific face parts in different images(e.g. LBP method depends highly on face alignment and parts segmentation). However, an intriguing possibility is that CNNs might be implicitly computing them in that some neurons concentrate on specific face parts as shown in Fig. 3.

We apply the method as [5] to evaluate the correspondence. First we take 15 identities with frontal images and systematically mask out the same part of the face in each image(e.g. all nose, as shown in Fig. 5). For each image i, the difference vector of features computed as $\varepsilon_i^l = x_i^l - \tilde{x}_i^l$, where x_i^l and $\tilde{x}_i^l$ are the feature vectors at layer l with/without occlusion respectively. Then the consistency of the difference vectors is measured by

$$\Delta_l = \sum_{i,j=1,i\neq j}^{m} \mathrm{H}(sign(\varepsilon_i^l), sign(\varepsilon_j^l)), \qquad (1)$$

where H is Hamming distance and m is the number of identities. A lower value of Δ_l indicates greater consistency in the change resulting from the occlusion, hence tighter correspondence between the same object parts in different images (i.e. occluding the left eye changes the feature representation in a consistent way).

The Δ_l score is computed for different parts in face images of 15 identities, from different layers in VGGFace, is shown in Table I. In Table I, compared to random parts occlusion, the lower value for face parts in low layer (pool3) reveals the model does establish some degree of correspondence of face parts in low layer. The degree of correspondence decreases in middle layer (pool4). While in high layer (fc7), the Δ_l value of random occlusion is lower than that of face parts, which reveals that the high layers trying to discriminate between identities so the random occlusions without face attributes could change deep feature little generating similar value of ε_i^l.

How to deal with pose: Since pose variation is widely regarded as a major challenge in the face recognition, we focus on comprehension of how CNNs deal with pose

TABLE I: Analysis of correspondence Δ_l for different parts in face images of 15 identities. We test the features from different layers in VGGFace.

Occlusion Location	Layer pool1	Layer pool4	Layer fc7
Left eye	0.094±0.029	0.191±0.028	0.225±0.031
Right eye	0.112±0.033	0.197±0.025	0.220±0.030
Nose	0.103±0.037	0.181±0.021	0.245±0.035
Mouth	0.101±0.036	0.181±0.024	0.212±0.033
Random	**0.214±0.044**	0.186±0.039	**0.178±0.046**

variance and the robustness to pose variation of different layers. We pick 10 identities each with approximately 30 frontal and 30 profile images and experiment on their features from different layers in CNNs to classify frontal/profile using kNN classification. The parameter k is learned via 10-fold cross validation and the best result is reported in Fig. 6.

Experiment results of VGGFace on both single people and all of ten people demonstrate the same trends: accuracy (1) increases from Layer pool1 to pool4, (2) decreases from layer pool4 to fc7, (3) increases a little in layer fc8. It's worth noting that the accuracy reveals that how much information of pose variance the network keeps. To begin, it is clear that the features at the early layers of the network are not very informative so the accuracy is relatively low. While it seems that decreased accuracy in high layers means a little better robustness to pose variation than middle layers. Meanwhile we observe that accuracy in high layers in VGGFace2 drops significantly than that in VGGFace, which is reasonable as VGGFace2 is well known for its robustness to pose variation. Perhaps this shows an early understanding of the robustness to pose variation.

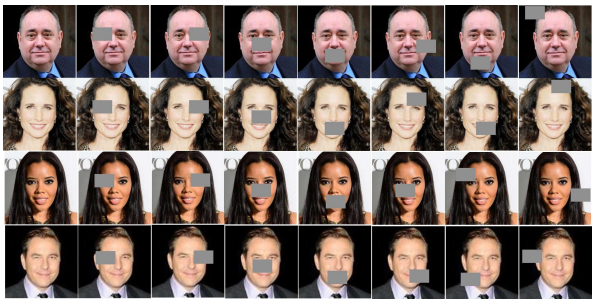

Fig. 5: Face Images for correspondence analysis. Col 1: Original image. Col 2,3,4,5: Occlusion of the right eye, left eye, nose and mouth respectively. Other columns show examples of random occlusions.

B. Face attributes of neurons

As we see in Fig. 3, a particular phenomenon in face networks is that features detected by neurons in high layers correspond to some face attributes (bald (Row 1, Col 1 in layer pool5); whisker (Row 1, Col 4 in layer pool5); eyeglasses (Row 2, Col 1 in layer pool5); bald or bangs (Row 2, Col 2 in layer pool5)), while features in high layers of object model may coincide with object types.

TABLE II: Analysis of Feature discrimination. We test the features from layer pool1-fc7 in VGGFace to on Labeled Faces in the Wild (LFW) and Cross-Pose LFW (CPLFW).

Layer	Acc on LFW (%)	Acc on CPLFW (%)
Layer pool1	62.93%	55.95%
Layer pool2	64.75%	55.83%
Layer pool3	71.27%	56.03%
Layer pool4	74.28%	55.90%
Layer pool5	93.82%	64.42%
Layer fc6	96.98%	77.62%
Layer fc7	96.73%	78.78%

Some attributes (e.g. eyeglasses, bald) are easy to define using a word which could be found in some annotated face attribute database [22]. However, some attributes are hard to define using a few words. In Fig. 3, we could observe that some reconstructions of a neuron are similar-looking yet we could not give them precise definitions. Maybe we could describe them with some approximate descriptions. For example, oblique and thin eyebrows upside the high nose in Row 2, Col 3 of layer pool5 and thick bushy eyebrows on flat and wide forehead in Row 2, Col 4 of layer pool5. However, these descriptions may be inaccurate and incomplete.

Usually a concept comes after perceiving lots of samples and finding their generality. To conclude the generality of samples, we apply an activation maximization method [6] searching for a prototype to activate a neuron most.

In the experiment of [17], with a DBN and an SDAE both trained on MNIST dataset, most random initializations yields roughly the same prominent input pattern using activation maximization method. While in [6], experiment using AlexNet trained on ImageNet dataset shows that a neuron may correspond to various input patterns. Since the feature diversity has been discussed before and we focus on describing a particular attribute of a neuron, the initialization is used as the mean image of top 9 samples of a type of similar-looking samples. Other regularization method like total variation (TV) and jitter, is the same as [6].

Fig. 7 visualizes four attributes in layer pool5. Both neuron 134 and 267 concentrate on the upper part of a face while generated images intuitively reveals that face attributes of them are quite different.

To analyze the relationship between these complex face attributes, we propose a similarity analysis tool by defining a similarity matrix $\{S_{ij}\}$

$$S_{ij} = \frac{1}{2}\left(\frac{\sum_{x_p \in C_j} N_i(x_p)}{\sum_{x_k \in C_i} N_i(x_k)} + \frac{\sum_{x_p \in C_i} N_j(x_p)}{\sum_{x_k \in C_j} N_j(x_k)}\right), \quad (2)$$

where C_i denotes the set of top n activation samples of the neuron i and $N_j(x_k)$ denotes the activation value of sample x_k on neuron j. We set paramater n as 50 in the following experiment.

We apply the similarity analysis to layer pool5 and produce a graphical heatmap as shown in Fig. 8. The larger

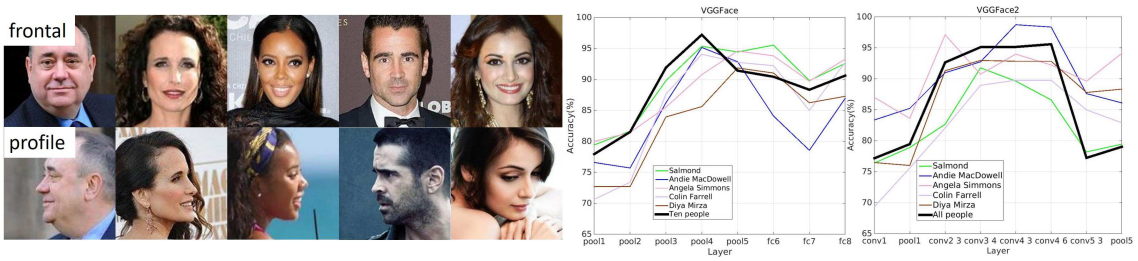

Fig. 6: Exploring how much information of pose variance the network learned by classifying frontal/profile using features from different layers in the model VGGFace/VGGFace2.

Fig. 7: Describing face attributes in layer pool5 using activation maximization. Both neuron 134 and 267 concentrate on the upper part of a face while generated images reveals intuitively that face attributes of them are quite different. Best viewed in electronic form.

intensity in Fig. 8 indicates higher similarity S_{ij} of two face attributes corresponding to neuron i and j. So overall, there exists a degree of redundancy among face attributes and the corresponding neurons. This result validates the former conclusion on redundancy of CNNs and would motivate the model compression of DNNs for face recognition.

We list the similarity of face attributes of shown in Table III, the corresponding neurons are shown in Fig. 8. We could observe that similarity value between any two of them is relatively low except for $S_{233,243}$, which is consistent with the intuitive visualization result in Fig. 7. Besides, the relationships between these four attributes miniature some of the low intensity similarity in Fig. 8, indicating that the network indeed perceives some independent complex face attributes. These complex face attributes are difficult for human to perceive when they try to distinguish face images. This is exactly why deep face models surpass human.

TABLE III: Similarity of face attributes shown in Fig. 7

	1	0.0285	0	0.0024
	0.0285	1	0.0051	0.0509
	0	0.0051	1	**0.1509**
	0.0024	0.0509	**0.1509**	1

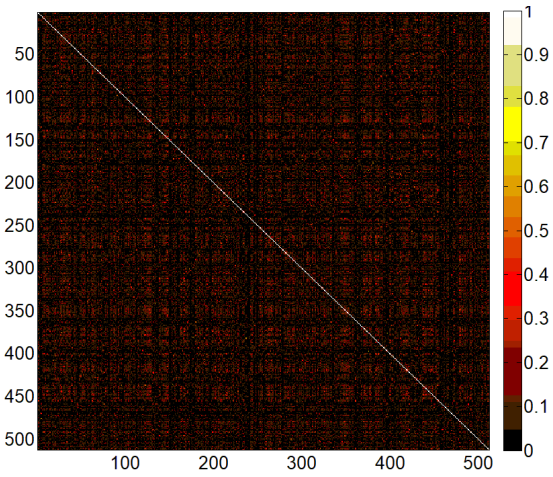

Fig. 8: Similarity analysis to layer pool5 in VGGFace. The black rows/cols indicate that the network indeed perceives some independent complex face attributes. The red rows/cols indicate that there are redundancy between these complex face attributes. Best viewed in electronic form.

V. CONCLUSION

We explore and understand CNNs trained for face recognition using some visualization methods. The experiments above all support the conclusion that higher-level features are more complex, more diverse, more discriminative and have more invariances than lower-level ones. The most significant phenomena we find is that high level features are correspond with complex face attributes which human could not describe using a few words. Some complex attributes are particular for CNNs which is difficult for human to perceive. This is exactly why deep face models surpass human. Besides, motivated by our experiment results, works on improvement on pose invariance and model compression remain to be done.

VI. ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China under Grant Nos. 61871052, 61573068, 61471048, and 61375031, and by the Beijing Nova Program under Grant No. Z161100004916088.

REFERENCES

[1] He, Kaiming, et al. "Deep residual learning for image recognition," *in CVPR*, 2016. 1

[2] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014. 1

[3] Parkhi, Omkar M., Andrea Vedaldi, and Andrew Zisserman. "Deep face recognition," *in BMVC*, 2015. 1, 3

[4] Liu, Weiyang, et al. "Sphereface: Deep hypersphere embedding for face recognition," *in CVPR*, 2017. 1

[5] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks," *in ECCV*, 2014. 1, 2, 3, 5

[6] Nguyen, Anh, Jason Yosinski, and Jeff Clune. "Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks," *ICML workshop*, 2016. 2, 3, 6

[7] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE," *in JMLR*, 2008. 1, 3

[8] Montavon, Grégoire, Wojciech Samek, and Klaus-Robert Müller. "Methods for interpreting and understanding deep neural networks," *in Digital Signal Processing*, 2017. 2

[9] Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps," *in ICLR Workshop*, 2014. 2

[10] Springenberg, Jost Tobias, et al. "Striving for simplicity: The all convolutional net," *arXiv:1412.6806*, 2014. 2

[11] Esteva, Andre, et al. "Corrigendum: Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, 2017. 1

[12] Khorrami, Pooya, Thomas Paine, and Thomas Huang. "Do deep neural networks learn facial action units when doing expression recognition?" *in CVPR Workshops*, 2015. 1

[13] Zhou, Bolei, et al. "Learning deep features for discriminative localization," *in CVPR*, 2016. 2

[14] Selvaraju, Ramprasaath R., et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *in ICCV*, 2017. 2

[15] Chattopadhyay, Aditya, et al. "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," *arXiv:1710.11063*, 2017. 2

[16] Bazzani, Loris, et al. "Self-taught object localization with deep networks," *in WACV*, 2016. 2

[17] Erhan, Dumitru, et al. "Visualizing higher-layer features of a deep network," *University of Montreal*, 2009. 2, 6

[18] Mahendran, Aravindh, and Andrea Vedaldi. "Visualizing deep convolutional neural networks using natural pre-images," *in IJCV*, 2016. 2

[19] Kuo, C-C. Jay. "Understanding convolutional neural networks with a mathematical model." *in JVCIR*, 2016. 1

[20] Huang, Gary B., et al. "Labeled faces in the wild: A database for studying face recognition in unconstrained environments." *Technical Report 07-49, University of Massachusetts, Amherst*, 2007. 5

[21] Zheng, Tianyue, and Deng, Weihong. "Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments," *Beijing University of Posts and Telecommunications, Tech. Rep*, 2018. 5

[22] Liu, Ziwei, et al. "Deep learning face attributes in the wild." *in ICCV*, 2015. 6

[23] G. Castanon and J. Byrne, "Visualizing and Quantifying Discriminative Features for Face Recognition," *in FG*, 2018. 2

[24] Yaoyao Zhong, Weihong Deng, "Deep Difference Analysis in Similar-looking Face recognition," *in ICPR*, 2018 2

[25] Q. Cao, L. Shen, W. Xie, O. M. Parkhi and A. Zisserman, "VGGFace2: A Dataset for Recognising Faces across Pose and Age," *in FG*, 2018. 3

[26] Wang Mei, Weihong Deng, "Deep Face Recognition: A Survey," *arXiv:1804.06655*, 2018. 1

[27] I. Goodfellow, H. Lee, Q. V. Le, A. Saxe, and A. Y. Ng. "Measuring invariances in deep networks," *in NIPS*, 2009. 5

[28] C. Bunne, L. Rahmann, and T. Wolf. "Studying invariances of trained convolutional neural networks," *arXiv:1803.05963*, 2018. 5