A Comparison between HOG and Deep Learning Features in Face Recognition

Hui Shen Sam

20130036
hcyhs1@nottingham.ac.uk
School of Computer Science, University of Nottingham

Abstract—**In this paper, the comparison between two face recognition methods is conducted using features extracted with Histogram of Oriented Gradients (HOG) and a pre-trained deep network. As deep networks are hard to train and prone to overfitting when the dataset is limited, the pre-trained ResNet-50 network on the VGGFace2 dataset is used to extract deep features from faces. Both face recognition methods are evaluated on a provided face dataset. The method using HOG as the feature extractor achieves an accuracy of 39.51%, whereas the method using the pre-trained deep network as the feature extractor can achieve an accuracy of 91.15%. Both face recognition methods achieve higher accuracy than the baseline method (25.37%). The results show that with transfer learning, the features learned by the pre-trained deep network have a better representation of faces and offer a higher recognition accuracy compared to that of handcrafted HOG features.**

I. Introduction

Face recognition is the task of identifying people by the features of their face, where the images and identity of the people to be verified are available beforehand. Face recognition has proven to be useful in applications such as logging attendance of employees in a company, airport security, or even help to track down criminals. There are three steps in face recognition. The first step is to locate the position of the face in the image, and then isolate the face region from the background. This step is called face detection/acquisition. The second step is to extract unique or the most useful features that best represent the face image. This step is called feature extraction. After obtaining the face features, the next step is to compare the face image with other images in the database, and then verify the identity of the input face image. This is done in the classification step.

Feature extraction is an important step in face recognition as the effectiveness of the classifier mainly depends on the quality of the extracted feature [1]. The feature extractor takes in raw pixel values of an image and transforms the image into a feature vector, which can then be fed into a classifier or used to measure the similarity between different images. Image features can be roughly divided into 2 categories: handcrafted features and learned features. Handcrafted features are extracted according to a manually predefined algorithm based on expert knowledge [2]. Examples of handcrafted features used in face recognition are Local Binary Patterns Histogram (LBPH), Scale Invariant Feature Transform (SIFT), and Histogram of Oriented Gradients (HOG). Unlike handcrafted features, learned features are extracted through the training process on a set of images in order to fulfill a certain task. For instance, Deep Convolutional Neural Networks (DCNN) is capable to extract learned features. Current research in face recognition is focused on DCNN, and it also achieved the best performance on the Labelled Faces in the Wild (LFW) benchmark [3], [4]. The availability of high-capacity GPUs and large publicly available datasets are also factors that cause the advancement of DCNN in this field.

The advantage of using handcrafted features is due to the transparency that the algorithm provides. This makes it possible for the CV engineer to have insights into a specific task and able to tweak certain parameters in the feature extractor specifically for that task [5]. However, it requires expert knowledge and a long trial and error process to determine which features have a stronger ability to represent each image. On the other hand, since DCNN are trained rather than programmed, DCNN often requires less expert analysis and fine-tuning when dealing with a tremendous amount of image data [5]. The DCNN model is able to discover the underlying patterns of the images automatically, which breaks through the traditional method of using prior knowledge to manually design features [6]. Moreover, due to the strong ability of DCNN to learn non-linear features of the image, the features extracted by DCNN often have a better representation of the images compared to traditional handcrafted algorithms, which can improve the classification or recognition accuracy. The disadvantage of DCNN is due to its intensive convolution and complex non-linear operations, which results in high computational cost [7]. The performance of DCNN also relies heavily on the dataset that it is trained on. For instance, if the training dataset is limited, overfitting may occur and DCNN will likely fail to generalise. Transfer learning is used to solve this problem, where we apply the pre-trained model designed by experts to our task. DCNNs that are trained with large-scale datasets are capable of representing general features and can be used as a pre-trained model for small datasets [6]. Even though we do not have a huge amount of training data or resources that can train these deep networks, we can still take advantage of the robustness of the pre-trained models.

In this paper, a baseline method and two other face recognition methods are proposed. Each method uses a different feature extractor. The baseline method represents an image's feature vector by taking all the pixel values of the image. The first proposed method uses HOG as the feature extractor, whereas the second method uses the pre-trained ResNet-50 Convolutional Neural Network (CNN) on the VGGFace2 dataset [8] to extract the feature vector of an image. Then, similarity measures such as cross correlation and cosine similarity are used to classify a test image.

The remaining of the paper is organised as follows: Section II discusses the background of HOG and the pre-trained ResNet-50 CNN on the VGGFace2 dataset; Section III describes in detail the methodology used to implement the face recognition methods; Section IV reports the results of each face recognition method on a provided test set; The conclusions are drawn in Section V.

II. Background

A. *Histogram of Oriented Gradients (HOG)*

The basic idea of HOG features is that the local object appearance and shape can often be characterized rather well by the distribution of the local intensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions [9]. In practice, the image is first divided into small cells and a local histogram of gradient orientations is computed for every cell. Then, larger spatial regions (blocks) that cover several cells are defined. The histogram entries of the cells within a block are concatenated to represent the HOG vector for that block. The purpose of overlapping spatial blocks is to normalise the contrast of local responses. To represent the HOG feature of the whole image, the HOG vector of each block is concatenated.

B. *Pre-trained ResNet-50 CNN on the VGGFace2 Dataset*

The VGGFace2 dataset is a large-scale face dataset that contains 3.31 million images of 9131 subjects. Cao et al. [8] evaluated the face recognition performance of the ResNet-50 CNN trained from scratch on three different datasets: VGGFace, MS1M (MS-Celeb-1M), and VGGFace2. The ResNet-50 CNN trained on VGGFace, MS1M, and VGGFace2 achieved a top-1 error of 10.6%, 5.6%, and 3.9% respectively when evaluating its recognition performance on the VGGFace2 test set. From the results, the ResNet-50 CNN trained on VGGFace2 achieved a better result as it has the lowest top-1 error compared to that of the model trained on VGGFace and MS1M. Moreover, the ResNet-50 CNN trained on VGGFace2 can achieve state-of-the-art performance on the IJB-A, IJB-B, and IJB-C benchmarks. As VGGFace2 is a high-quality dataset and it contains a large number of face images, the ResNet-50 CNN trained on it is effective in extracting generalised features from face images. Therefore, the pre-trained ResNet-50 CNN on the VGGFace2 dataset can be used to extract useful face features for the face recognition task in this paper.

III. Methodology

This section discusses the implementation of the baseline method and the proposed face recognition methods. These methods are implemented in Matlab. There are 100 training images and 1344 test images in the dataset. Each image is of size 600 × 600. It is verified that each image in this dataset has a face in it, therefore the face detection/acquisition step is skipped.

A. *Baseline Method*

In this paper, a simple baseline method is implemented and used as a benchmark for the other two proposed methods. The baseline method uses all pixel values of an image to represent its feature vector. The baseline method is implemented as follows:

1) The training and test images are converted to grayscale images to obtain images with a single channel.
2) The pixel values of the training and test images are normalised to the range between 0 and 1, simply by dividing each pixel value by 255. This is because the maximum pixel value of a grayscale image is 255.
3) The feature vector is obtained by flattening the pixel values of the image into a vector of size 360000.
4) The feature vector of each training and test image is normalised using the zero-mean normalisation.
5) To identify a test image, the dot product (cross correlation) of the test image's feature vector with each training image's feature vector is calculated. The label of the training image that corresponds to the largest cross correlation value will be the identity of the test image.

B. *Method 1 (HOG Features)*

This method uses the HOG feature extractor to extract the feature vector of the face image. The HOG feature extractor uses 8×8 cell size, 4×4 block size, 16 orientation bins, and the number of overlapping cells between adjacent blocks is 3×3. These parameters are tuned manually and evaluated its performance on the test set while ensuring the time needed to run this method will not take too long. The image is also resized to 160×130 such that the size of the HOG feature vector is not too large. The HOG feature of each image is a vector of size 56576. The face recognition method using HOG features is implemented as follows:

1) The training and test images are converted to grayscale images to obtain images with a single channel.
2) The training and test images are resized to 160×130.
3) Using the HOG parameters mentioned above, the HOG feature vector of each training and test image is obtained.
4) The feature vector of each training and test image is normalised using the L_2-norm.
5) To identify a test image, the dot product (cross correlation) of the test image's feature vector with each training image's feature vector is calculated. The label of the training image that corresponds to the largest cross correlation value will be the identity of the test image.

C. *Method 2 (Deep Features)*

This method uses the pre-trained ResNet-50 CNN on the VGGFace2 dataset to extract the feature vector of the face image. The pre-trained model together with its trained weights is available online[1]. For this method, the pre-trained model is used as a fixed feature extractor instead of re-training it on the given training images and then predict the test images. The deep features of the image are extracted from the flatten layer adjacent to the classification layer. The extracted feature vector has a size of 2048. The face recognition method using deep features is implemented as follows:

1) The pre-trained model saved in the .h5 file format is loaded into Matlab.
2) The training and test images are resized to 224×224 because the pre-trained model takes in an image of size 224×224 as input.
3) The training and test images are passed into the model, and the values of the flatten layer adjacent to the classification layer are extracted as the feature vectors.
4) All feature vectors in the training set are normalised as a whole using the L_2-norm. The same goes to the feature vectors in the test set.
5) To identify a test image, the cosine similarity between the test image's feature vector and each training image's feature vector is calculated. The label of the training image that corresponds to the largest cosine similarity value will be the identity of the test image.

IV. Results and Discussion

In this section, the performance of the two proposed methods is evaluated against the baseline method. For each method, the prediction on the test set is compared with the ground truth. The number of correct predictions made will be used as the evaluation metric (accuracy). Each method is run 5 times, and the average time and accuracy are recorded and shown in Table 1.

	Baseline	Method 1 (HOG Features)	Method 2 (Deep Features)
Avg. Accuracy (%)	25.37	39.51	91.15
Avg. Time (s)	82.01	214.58	97.32

TABLE I: Average accuracy and time of each face recognition method

The result shows that both proposed methods can achieve higher accuracy than the baseline method. The time taken for method 1 is the longest, followed by method 2 and then the baseline method. In method 2, the feature vector of an image is extracted using the pre-trained weights of the DCNN. The pre-trained DCNN is not retrained on the given dataset, which makes method 2 able to achieve a relatively short average time. All three methods are deterministic, therefore the accuracy will be the same given the same input, but only the time taken varies slightly.

While method 1 can achieve higher accuracy than the baseline method, the recognition accuracy is still lower than that of method 2 by a large margin. This shows that the features extracted by the pre-trained DCNN have a better face representation compared to the HOG features. This is because the model is pre-trained on the VGGFace2 dataset, where it can extract robust features for faces. However, the pre-trained model chosen to extract the image features has a large impact on the recognition performance in this method. For instance, using the pre-trained AlexNet to extract the face features in this dataset and then using the cosine similarity metric for classification only achieves an accuracy of 29.02%. This is because the AlexNet model is pre-trained on images used for object recognition, but not face recognition. While the model is capable of detecting general features like colours and edges, it could not extract features specifically for faces. This indicates that it is important to choose a pre-trained model that was trained on images of the same domain as the task at hand. Furthermore, with the limited amount of face images in this dataset, retraining the AlexNet model causes it to overfit the training set and performs poorly on the test set.

V. Conclusion

In conclusion, using the pre-trained ResNet-50 CNN on the VGGFace2 dataset as a feature extractor outperforms the baseline method and the HOG feature extractor. This shows that features learned by the pre-trained DCNN have a better representation than handcrafted HOG features. In practice, it is faster and more practical to use a pre-trained model to extract the feature vectors of the images compared to training a DCNN from scratch. Due to the limitation of resources and small dataset, this causes the model to overfit and does not perform well. With transfer learning, we can make use of well designed DCNN and its pre-trained weights to achieve good results, as shown in Table 1.

References

[1] A. J. Shepley, "Deep learning for face recognition: A critical analysis," *CoRR*, vol. abs/1907.12739, 2019.

[2] G. Antipov, S. A. Berrani, N. Ruchaud, and J.-L. Dugelay, "Learned vs. hand-crafted features for pedestrian gender recognition," 10 2015.

[3] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments," in *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, (Marseille, France), Erik Learned-Miller and Andras Ferencz and Frédéric Jurie, Oct. 2008.

[4] S. Balaban, "Deep learning and face recognition: the state of the art," *CoRR*, vol. abs/1902.03524, 2019.

[5] N. O. Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. A. Velasco-Hernández, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," *CoRR*, vol. abs/1910.13796, 2019.

[6] G. Lu, Q. Hao, K. Kong, J. Yan, H. Li, and X. Li, "Deep convolutional neural networks with transfer learning for neonatal pain expression recognition," in *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 251–256, 2018.

[7] S. Wu, M. Kan, Z. He, S. Shan, and X. Chen, "Funnel-structured cascade for multi-view face detection with alignment-awareness," *CoRR*, vol. abs/1609.07304, 2016.

[8] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," *CoRR*, vol. abs/1710.08092, 2017.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, 2005.

[1] https://github.com/rcmalli/keras-vggface