Robust Face Recognition by Computing Distances from Multiple Histograms of Oriented Gradients

Mahir Faik Karaaba, Olarik Surinta, Lambert Schomaker and Marco A. Wiering
Institute of Artificial Intelligence and Cognitive Engineering (ALICE), University of Groningen,
Nijenborgh 9, Groningen, The Netherlands
Email: {m.f.karaaba, o.surinta, l.r.b.schomaker, m.a.wiering}@rug.nl

***Abstract*—The Single Sample per Person Problem is a challenging problem for face recognition algorithms. Patch-based methods have obtained some promising results for this problem. In this paper, we propose a new face recognition algorithm that is based on a combination of different histograms of oriented gradients (HOG) which we call Multi-HOG. Each member of Multi-HOG is a HOG patch that belongs to a grid structure. To recognize faces, we create a vector of distances computed by comparing train and test face images. After this, a distance calculation method is employed to calculate the final distance value between a test and a reference image. We describe here two distance calculation methods: mean of minimum distances (MMD) and a multi-layer perceptron based distance (MLPD) method. To cope with aligning difficulties, we also propose another technique that finds the most similar regions for two compared images. We call it the most similar region selection algorithm (MSRS). The regions found by MSRS are given to the algorithms we proposed. Our results show that, while MMD and MLPD contribute to obtaining much higher accuracies than the use of a single histogram of oriented gradients, combining them with the most similar region selection algorithm results in state-of-the-art performances.**

I. Introduction

While easily performed by humans, recognizing a face is still a challenging task for computers. Face recognition has two different application fields. One is face identification, where the task is finding the real identity given a sample face image. The other one is face verification, where the task is deciding whether two faces belong to the same person. We focus in this paper on face identification, due to its demand and popularity.

In the last decade, there has been a significant advancement for solving the face recognition problem. Nevertheless, face recognition needs to work better to be widely employed in the real world. In many application fields, such as security and law enforcement applications, there are often not sufficient reference images to recognize a given test image of a person due to data collection difficulties. This is generally called the small sample size (SSS) problem [1]. In many cases, even more than one image is not available which is an extreme case of the SSS problem and is named as single sample per person (SSPP).

A known fact in a face recognition task is that the differences of the face images of the same person (intra-class) can be much bigger than differences of the face images of different subjects (inter-class) due to different poses and illumination conditions [2], [3]. For example, two photos of the same person taken in different poses or illumination conditions will have a higher geometrical distance than two photos of two different people whose pose and illumination conditions are the same. Due to this fact, if there are not enough training samples, a naive face recognition method which basically relies on raw image similarities will not perform well. To overcome such a problem, various methods have been proposed over the years [1], [4], [5], [6], [7]. Because of its importance, in this paper, we also seek a solution for the SSPP problem.

Related Work. The first proposed methods for the face recognition problem, which were proven effective at their time, are appearance (holistic) based methods. Eigenfaces [8] and Fisherfaces [9] are the simplest and most well-known methods of these. If there is a sufficient amount of well aligned training samples, these algorithms can work well. However, aligning a face automatically is usually error prone. Also, such methods are sensitive to illumination changes because they directly process pixels.

If pixel intensities are replaced with local feature outputs, better performances can be obtained. The Gabor filter is one of the oldest local feature extractors which is used in many computer vision applications. It has also been applied to face recognition as in [10]. The scale invariant feature transform (SIFT) [11] is reported to give promising results [12], and the histogram of oriented gradients (HOG) [13] has also been applied to the face recognition problem successfully [14], [15]. They are mostly invariant to illumination variations and, provided that there are enough properly aligned training data, they obtain good performances. Particularly HOG has been shown to get better performances than a Gabor filter if combined with an elastic matching method [14]. Nevertheless, they are not robust enough to handle the SSPP problem, since the pose variations and aligning errors skew the similarity of train and test distributions of face image data which is essential to obtain good performances.

As alignment is an important part of a face recognition application, active shape models (ASM) [16] and active appearance models (AAM) [17] have been proposed for robustly aligning a face. The basic idea behind the ASM is that face images of a subject can be modeled as a statistical shape model. Later, AAM was proposed on the bases that faces should not be modeled only by points but also by pixel intensities. These methods have also been extended and improved by adding texture information to the model [18], [19].

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.39

To tackle the SSPP problem, artificial data generation and using generic data are explored in this paper. Generating artificial face samples may be effective if these samples decrease the intra-class variance of training samples as well as increase the variance of the inter-class adequately. In [20], to exploit the asymmetric nature of face appearances, mirrored images created from original samples as artificial supplementary images were added to the original data and this was reported to perform better than only using original faces. To cope with alignment and pose problems where sufficient data are not available, a generic dataset may also be beneficial. In [4], generic data are used to learn a Fisher's linear discriminant model that is later adapted to the actual data.

Patch based methods have been popular in recent years in face recognition research, because of their successful results. In general, instead of using the whole face image as input, patch based methods divide an input image into several patches, in grid or sliding window fashion. In [6], faces are represented as manifolds which are composed of non-overlapping patches. Then, margins for each subject pair are optimized with a reconstruction-based discriminant learning method. This obtained better performances compared to using a single manifold which is based on a whole face image.

In [21], a random walk based similarity measure is proposed to compute face similarities. For this, an in-face and an out-face network are constructed. In the in-face network, several overlapping face patch samples together with 8 neighbouring patches are used to make the network. A vector of similarity values are calculated using this network. For the out-face network, the patch locations selected for the in-face network are collected for all the training face samples. Then, the final verification process is performed by these similarity vectors for each face patch-pair. In [22], a correlation-based filter bank is constructed to capture similarities of sample images of the same person and to capture differences of similar looking image samples of different people. There they use a grid-based partitioning to compute patches from images.

Another popular family of methods is based on neural networks with many layers which are called deep neural networks. Especially convolutional neural networks (CNN) for face verification are reported giving promising results. In [23], a CNN is adopted to learn if two faces are the same or not for face verification. A large amount of data is used (100K images of 3K subjects) to train the CNN. Besides, a 3D face alignment is employed additional to a 2D alignment before creating the data for the CNN. This alignment contributes to better performances.

In [24], the correlation among the amount of training data, distributions of the train and the test data and the accuracy is investigated regarding to using a CNN approach as the learning algorithm. According to this, increasing the training data is not helpful after some point and the imbalanced sample amount per subject distribution (also called *the long tail effect*) has a negative impact on the performance. The CNN is becoming more popular due to its very good performance potential in computer vision problems, though it requires a large amount of training data and long training times.

Contributions. In this paper, we propose two novel algorithms which work hierarchically to identify faces. When two input faces are given to the system, first the most similar regions are found by a distance-based search algorithm. The regions, for which the Euclidean distance computed by using HOG features is the smallest, are found by using a sliding window approach. After the best regions for two face images are located, the multi-HOG based algorithm is employed to create a vector of distances on these located regions. The vector of distances is then given to a distance computation function to obtain the real distance value before feeding it to a 1-nearest neighbour classifier (1-NN).

We propose two distance computation methods which map the list of distance values to a single distance value. These methods are the mean of minimum distances (MMD) and a multi-layer perceptron distance function (MLPD). To train the MLPD, we used a generic dataset, which is composed of the IMM and the MUCT face datasets. We have tested our algorithms on two face datasets, namely FERET and LFW. The results show that our methods give better or close performances compared to state-of-the-art face recognition algorithms.

Paper Outline. The rest of the paper is organized as follows: In Section II, the face recognition algorithm which is proposed is described. In Section III, experimental settings and the results are presented. In Section IV, the conclusion and future work is given.

II. Proposed Face Recognition Algorithm

A. Grid-Based Multi-HOG Technique

In this paper, we use a grid-based distance computation algorithm based on multi-HOG features.

1) Basic Definition of HOG: The histogram of oriented gradients was proposed in [13] for the application of pedestrian detection. HOG is a feature extraction technique that computes the oriented gradients of an image using gradient detectors. Because of its successful results, it has been used in many computer vision systems. For instance, it has been used for face [25] and on-road vehicle [26] detection applications. It has also been applied to face identification as well as emotion [27] and gesture recognition [28].

We are now giving the mathematical description of the HOG method: Let G_x and G_y be the horizontal and vertical components of the gradients, respectively. These are computed by using intensities of the pixels $I(x,y)$ at positions (x,y) as in the following equations:

$$G_x = I(x+1,y) - I(x-1,y) \tag{1}$$

$$G_y = I(x,y+1) - I(x,y-1) \tag{2}$$

The magnitude and angle of the gradients are computed as:

$$M(x,y) = \sqrt{G_x^2 + G_y^2} \tag{3}$$

and

$$\theta_{x,y} = \tan^{-1} \frac{G_y}{G_x} \tag{4}$$

where $M(x,y)$ is the magnitude of gradients, and $\theta_{x,y}$ is the angle of the gradient at the given location.

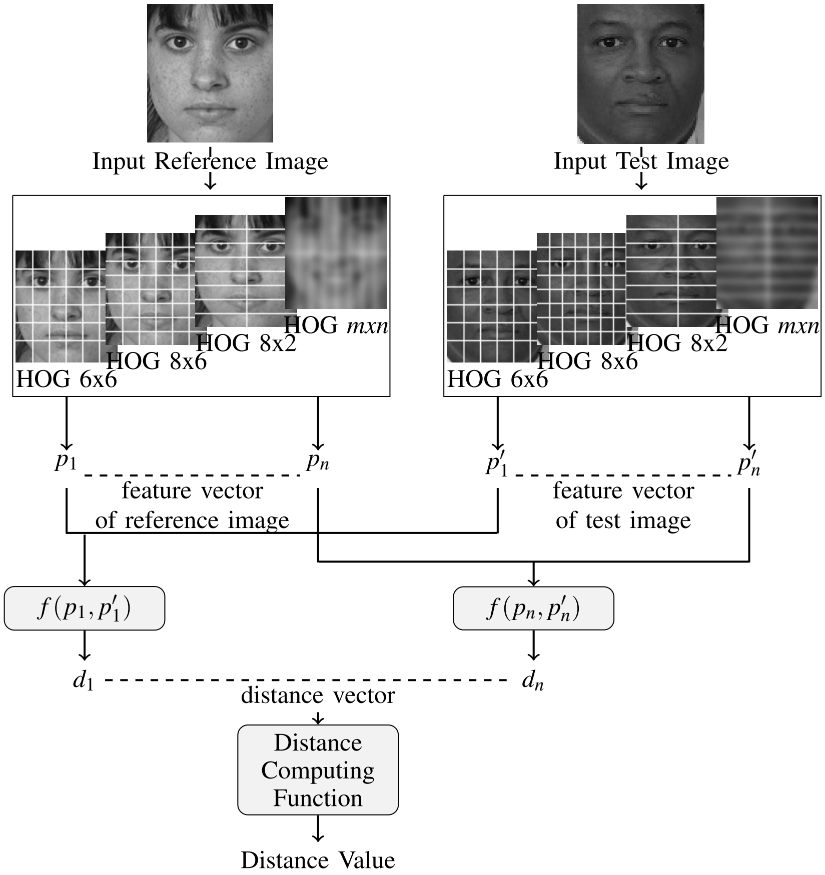


Fig. 1: Method of computing the distance between two faces where $f(,)$ is the Euclidean distance function.

Here, angles and magnitudes are computed from the gradients. The angles are selected according to *orientation bins*. Then, magnitudes are summed up for each angle. We compute the bin-values $V(b_\theta)$ as follows:

$$V(b_\theta) = \sum_{y=1}^{y_{max}} \sum_{x=1}^{x_{max}} M_{b_\theta}(x,y) \tag{5}$$

where

$$M_{b_\theta}(x,y) = \begin{cases} M(x,y) & \text{if } b_\theta = \lceil \frac{\theta_{x,y} B}{2\pi} \rceil \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

here b_θ is the bin for the angle θ, and B is the bin size.

2) Distance Vector Construction from Multi-HOG Features: In the typical HOG method, the image is divided into *sub-images* which are composed of pixels. For each sub-image a separate histogram is constructed after which all histograms are concatenated and normalized to form the feature vector. In our method, on the other hand, from the same input image we create several sub-image sets each of which contains different grid dimensions. Besides, each sub-image set is not fixed to the same bin size.

Then, all of these sub-images are used to construct a distance vector which is composed of distance values computed for each image-pair. The Euclidean distance is used to calculate these values. See Fig. 1 for a graphical explanation.

3) Distance Computation Function: The distance vector is then given to a distance computation function to be used for the computation of a single distance value between two images. After that a 1-nearest neighbor classifier (1-NN) classifier assigns the label of the reference face which has the closest distance to the test image. We used 2 distance computation functions in our experiments: MMD and MLPD functions.

The first method, the MMD, is computed as the mean value of the selected minimum distance values of HOG blocks. First many distances between different HOG features extracted from different patches are computed.

$$d_i = f(\mathbf{p}_{i_R}, \mathbf{p}_{i_T}), \quad i = 1 \ldots n, \quad \mathbf{p_i} \in \mathbb{R}^b \tag{7}$$

where f is the Euclidean distance function, d_i is Euclidean distance value for the ith patch, $\mathbf{p_i}$ is a patch vector obtained by HOG each of which has bin size b. n is the total number of patches. R and T represent reference and test (patch), respectively. To obtain the set of minimum distances, we use SelectMinimumDistances.

$$\mathbf{d_s} = \text{SelectMinimumDistances}(\mathbf{d}, k) \tag{8}$$

$$0 < k < n, \quad \mathbf{d} \in \mathbb{R}^n, \quad \mathbf{d_s} \in \mathbb{R}^k \tag{9}$$

where **d** is the distances vector produced by the multiple HOG features and $\mathbf{d_s}$ is the minimum distances vector. For the description of the *SelectMinimumDistances*, see Algorithm 1. We used this algorithm for eliminating the noise which results from occlusions, accessories (glasses, mustache and beard) as well as facial expressions. We noticed that this was effective for obtaining better performances than using all distances.

Algorithm 1 SelectMinimumDistances (**d**,k)

```
k is the number of minimum selected distances
Initialize md as minimum distance vector;
d is vector of main distances,
while $i \leq k$ do
    find the minimum distance value: $md_i := argmin(\mathbf{d})$
    add the minimum distance value to md: $\mathbf{md} \leftarrow md_i$
    remove that value from original distance vector d, i := i + 1
end while
Return md
```

Finally, the average distance value is calculated as:

$$\bar{d_s} = \frac{1}{k}\sum_{i=1}^{k} d_{s_i} \tag{10}$$

$\bar{d_s}$ is now the mean of the minimum selected distance values computed from a train and test image pair.

Let there be N reference samples in total. From this we employ a 1-NN to compute the final label belonging to a test image:

$$C = \arg\min_{c=1}^{N} \bar{d}_{s_c} \tag{11}$$

where C is the class label of the training sample, which is selected as the identity of the test face image.

In the second method, a multi-layer perceptron based neural network is employed. That multi-layer perceptron (MLP) is trained on the computed distance vectors with the same HOG features as before from two images as input and is trained to output if the two images are from the same person or not. For training the MLP a generic dataset is used. The inputs of the MLP are two distance vectors from two images and the target output is set to *0* if the distance vectors are constructed from the same person's face images, and *1*, otherwise. For this method, we are partly inspired by a face verification approach [29] where the classifier is expected to determine if an image pair is composed of the same person or not.

B. Adding Mirrored Faces

The face images appear usually in different poses rather than frontal. This presents sometimes serious problems for the performance of a face recognition algorithm. Non-frontal face images can also be considered non-symmetrical. This means that taking the mirrored image of such a picture will supply a novel face image. Due to this fact, we employed a mirrored version of each training face image sample as a supplementary sample, similarly as in [20], and the results show that this improves the recognition performance significantly.

C. Illumination Correction

Illumination usually presents a problem for a typical face recognition algorithm since it changes the appearance which creates additional noise. Despite that HOG features are generally robust to such changes, our illumination correction method still improves the performance slightly. In our correction algorithm, average brightness and contrast of the image are adjusted according to a fixed mean and standard deviation of pixel intensities.

Algorithm 2 SearchMostSimilarRegion ($img_1, img_2, inc_x, inc_y, inc_w, inc_h$)

```
function fdist($img_1$ , $img_2$)
    $hog_1 = getHog(img_1)$
    $hog_2 = getHog(img_2)$
                ▷ getHog is histogram of gradients calculator
return get root mean square of $hog_1$ and $hog_2$
end function
Set $w$ and $h$ to initial values;
Set $x$ and $y$ to zero;
function Search($img_1, img_2, inc_x, inc_y, inc_w, inc_h$)
    while $x \leq max_x$ do
        while $y \leq max_y$ do
            while $w \leq max_w$ do
                while $h \leq max_h$ do
                    $subim_2 := subimage(img_2, x, y, w, h)$
                    $similarity := fdist(img_1, subim_2)$
                    $distances \leftarrow similarity$
                    increment $h$ value: $h := h + inc_h$
                end while
                increment $w$ value: $w := w + inc_w$
            end while
            increment $y$ value: $y := y + inc_y$
        end while
        increment $x$ value: $x := x + inc_x$
    end while
    $distance_{min} := argmax(distances)$
    Return $x$, $y$ and $w$ and $h$ with the minimum distance value
end function
$(x, y, w, h)@mindist := Search(img_1, img_2)$
$(x, y, w, h)@mindist := Search(img_2, img_1)$
if $dist_1 \leq dist_2$ then
    Return $img_2(x, y, w, h)$
else
    Return $img_1(x, y, w, h)$
end if
```

D. Maximum Similarity Based Region Selection

In general there are always some small errors in the face alignment process which can cause problems in the comparison stage. This is caused mainly by pose differences and ground truth errors. To handle this problem and improve the performance, we employ a search using the most similar region algorithm that finds the geometrically closest regions between the compared face pairs. This is done by computing Euclidean distances on extracted HOG features when different windows are used, and selecting the sub-images with the smallest distance. After the face regions are obtained for a

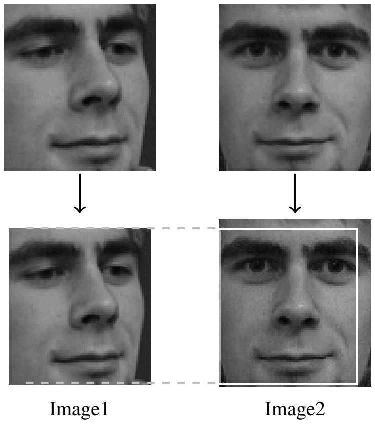


Fig. 2: The white rectangle in the second image is selected as the most similar region to the first image.

(a) (b)

Fig. 3: Sample aligned face images of one subject from the generic dataset. (a) the MUCT and (b) the IMM dataset.

face pair, these are given to the distance vector construction algorithm. As we will show, finding geometrically more similar facial regions than original ones improves the performance significantly. The pseudo-code of this algorithm is given in Algorithm 2. For the graphical illustration of the most similar regions found by the algorithm, see Fig. 2.

III. Experimental Setup and Results

A. Datasets

In our experiments, we make use of 4 face datasets: 2 of them (MUCT and IMM datasets) for MLP training, and the other datasets: FERET and Labeled Face in the Wild (LFW) are used for evaluating our methods.

1) Generic Training: The MUCT dataset was created in December 2008 at the University of Cape Town [30]. It is composed of totally 3,755 face images of 175 individuals. The dataset is divided into 5 categories for different pose angles at which the face pictures are shot. It also has annotations (76 for each photo) for alignment purposes created mainly for experiments of active appearance models [17].

The IMM face dataset was created by the Technical University of Denmark and contains 240 images with 40 individuals [31]. Like MUCT, it also provides annotations. Although it contains a lower amount of samples compared to MUCT, IMM has more pose variations than the former. Therefore we wanted to benefit from both datasets to train the MLP distance function in the proposed system. Sample photos of the MUCT and the IMM datasets are shown in Fig. 3.

2) Test Datasets: The datasets which are used to show final performances are the FERET [32] and the LFW [33] datasets. The FERET dataset was made by the defense advanced research projects agency (DARPA) and the national institute of standards and technology (NIST) for evaluation of face recognition algorithms. It is a huge dataset, thus we selected a subset of this dataset to use in our experiments, which contains 196 subjects with 7 samples of each subject. The subset we chose includes roughly 3 challenging features which can worsen the performance of a face recognition system: illumination changes (dark and bright images), pose (left, right and frontal poses) and expressions (smiles). For example face photos of the FERET dataset, see Fig. 4.

The LFW dataset is created for testing computer vision algorithms under unconstrained environments. It includes approximately 13,000 images of around 6,000 subjects. We selected from this dataset 150 subjects each of which contains at least 7 samples. For example face photos of the LFW dataset, see Fig. 5. We selected these dataset configurations similarly as in [22].

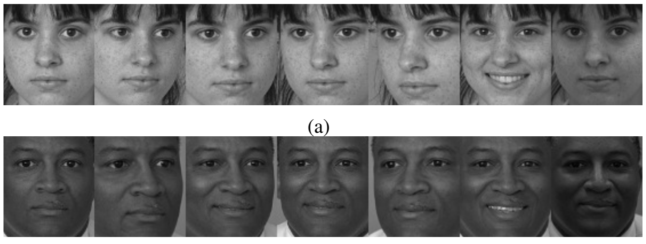
For all datasets we aligned the face images by using eye coordinates as ground truth. To obtain the eye centers, we used the manual crop information provided in the dataset folder, except for the FERET dataset, from which we cropped the face images automatically by our eye and eye-pair detector (since FERET does not provide sufficient ground truth information for each image) and replaced badly cropped ones with manual crops (around 5% of them). After obtaining eye-coordinates, we followed the aligning method as presented in [34].

B. Parameter Tuning

For the MMD algorithm, we choose the minimum 50% of the distances which worked best in preliminary experiments. To create data for the MLPD algorithm, we used 100 subjects from MUCT and IMM as a mixture, yielding about 750 sample pictures of faces with at least 6 samples per subject. We also added a mirrored version of each face image which accounts for 1,500 face images. The distance vector inputs that are given to the MLP are made of combinations of sample pairs. It means that the distance vector amount finally becomes $\binom{1500}{2}$ ≈1,100K. As hidden unit (*hu*) size, $hu = 15$ worked the best in our system.

To test the model's performance, we used two validation sets each of which is for a corresponding test dataset. These validation datasets are collected from unused parts of the training and test datasets. We trained the MLP with 10 epochs and saved the model after each epoch. Subsequently, the models which resulted in the best accuracy on the validation datasets are selected for testing with the actual dataset. The reason we used separate validation datasets is to handle the differences of the datasets, namely for FERET and LFW.

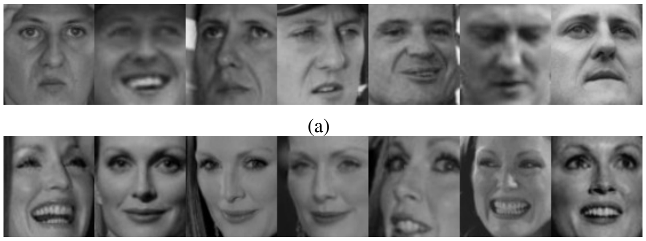
HOG Parameters We used 80×88 as resolution $width \times height$ of the images and extracted sub-images (by converting them to this standard size).. We use the notation of (w,h,b) for a HOG parameter where w is the number of columns, h is the number of rows and b is the number of bins. While the single HOG parameters are chosen as (8,8,24), the combination of HOG parameters (multi-HOG) which worked best in our experiments is (8,8,24), (6,6,24), (2,8,24), (1,11,21),



(a)

(b)

Fig. 4: Sample aligned face images of two subjects from the FERET dataset.



(a)

(b)

Fig. 5: Sample aligned face images of two subjects from the LFW dataset.

(2,11,21), (8,7,24), (8,6,24), (7,8,24), (5,8,24), (6,8,24) and (7,11,21).

SearchMostSimilarRegion Finally, the parameters used for Algorithm 2 are as follows:

For the *getHog* function, $8 \times 8 \times 24$ is used as HOG parameter. $inc_x, inc_y, inc_w, inc_h$ are all set to 2. While $w \times h$ (initial resolution) are initialized to 72×80, $max_w \times max_h$ (highest resolution) is set to 80×88.

TABLE I: Face recognition results on the LFW dataset.

Method	Mirrored	No Mirrored
HOG	17.87±0.6	17.61±0.5
Multi-HOG MMD	20.61±1.1	20.20±1.0
Multi-HOG MLPD	22.34±0.5	22.00±0.6
Multi-HOG MSRS-MMD	22.79±1.1	22.14±1.0
Multi-HOG MSRS-MLPD	**23.49**±1.2	**22.85**±0.9
DMMA [6]	-	22.17±2.8
MS-CFB (cos) [22]	-	21.15±2.9

TABLE II: Face recognition results on the FERET dataset.

Method	Mirrored	No Mirrored
HOG	46.52±1.2	39.50±1.3
Multi-HOG MMD	55.94±1.0	49.00±1.0
Multi-HOG MLPD	64.67±1.2	59.18±1.4
Multi-HOG MSRS-MMD	64.43±0.8	57.68±0.9
Multi-HOG MSRS-MLPD	**68.59**±1.0	64.68±1.3
DMMA [6]	-	65.24±2.0
MS-CFB (cos) [22]	-	**66.60**±2.1

C. Experiments and Results

In order to obtain statistically stable results, we used 10-fold cross validation in both learning (both of MMD and MLPD) and testing stages. In this way, we selected 1 example from each subject folder randomly as training samples and the rest is used as test samples.

Table I and Table II show the results (average accuracy and standard deviation). According to these results, if the methods are combined with the MSRS algorithm, our methods perform the best for both LFW and FERET. We also see that mirrored images generally improve the performance. When we use mirrored images together with our best distance functions, the results outperform the others. As easily seen from the table, for both datasets, Multi-HOG shows better results than single HOG. It suggests that using more than one HOG feature vector captures more information related to the class of the subject. If we use one fixed HOG vector, then pose variations cause increasing intra-class distance variations. For LFW, we have better results than the results of other state-of-the-art algorithms papers (DMMA [6] and MS-CFB [22]). It proves the efficiency of our method. For FERET, when we use mirrored versions together with the MLPD method, we obtained the best results. When not using mirrored images, we obtain comparable results. While good results are also obtained with MMD, they are worse than the results obtained with the MLPD function. From this, the usage of generic data is proven to improve accuracy considerably.

IV. Conclusion

In this paper, we have described three novel algorithms: the most similar region search (MSRS) algorithm, distance vector construction by multiple HOG features which takes multiple HOG based patches as input to return a distance vector, and the distance computation function which outputs the final distance value. We then introduced two kinds of distance computation functions: namely the mean of minimum distances (MMD) and the multi-layer perceptron based distance (MLPD) function.

Our results showed that using multiple HOG features together with MSRS combined with MLPD obtains the best results for the LFW dataset. For FERET, it gains very comparable results to state-of-the-art methods if no mirrored images are used. But, if the mirrored images are added, then the best results are obtained with our proposed technique. We should also point that MSRS-MLPD gives better results than MSRS-MMD, which proves the usefulness of using a generic dataset. Regarding to using mirrored images, while significant performance improvements can be seen on the FERET dataset, relatively smaller benefits are obtained on the LFW dataset.

In future work we want to improve these results further by using more distance values as well as a bag-of-words approach instead of grid-based fixed partitioning. We also consider using more layers for learning the distance function using a deep learning framework.

References

[1] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, "Face recognition from a single image per person: A survey," *Pattern Recognition*, vol. 39, no. 9, pp. 1725–1745, Sep. 2006.

[2] X. Zhang and Y. Gao, "Face recognition across pose: A review," *Pattern Recognition*, vol. 42, no. 11, pp. 2876 – 2896, 2009.

[3] R. M. Makwana, "Illumination invariant face recognition: A survey of passive methods," *Procedia Computer Science*, vol. 2, no. 0, pp. 101 – 110, 2010, proceedings of the International Conference and Exhibition on Biometrics Technology.

[4] Y. Su, S. Shan, X. Chen, and W. Gao, "Adaptive generic learning for face recognition from a single sample per person," in *Computer Vision and Pattern Recognition, (CVPR) 2010 The Twenty-Third IEEE Conference on*, 2010, pp. 2699–2706.

[5] F. Hafiz, A. A. Shafie, and Y. M. Mustafah, "Face recognition from single sample per person by learning of generic discriminant vectors," *Procedia Engineering*, vol. 41, no. 0, pp. 465 – 472, 2012, international Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).

[6] J. Lu, Y.-P. Tan, and G. Wang, "Discriminative multi-manifold analysis for face recognition from a single training sample per person," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov 2011, pp. 1943–1950.

[7] B. Kveton and M. Valko, "Learning from a single labeled face and a stream of unlabeled data," in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, April 2013, pp. 1–8.

[8] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[9] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 711–720, Jul 1997.

[10] Y. B. Jemaa and S. Khanfir, "Automatic local Gabor features extraction for face recognition," *International Journal of Computer Science and Information Security (IJCSIS))*, vol. 3, no. 1, 2009.

[11] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[12] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, "On the use of SIFT features for face authentication," in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, June 2006, pp. 35–35.

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

[14] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol, "Face recognition using HOG-EBGM," *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1537 – 1543, 2008.

[15] O. Déniz, G. Bueno, J. Salido, and F. D. la Torre, "Face recognition using histograms of oriented gradients," *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1598–1603, 2011.

[16] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active Shape Models - Their Training and Application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

[17] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active Appearance Models," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Springer, 1998, pp. 484–498.

[18] P. Kittipanya-ngam and T. Cootes, "The effect of texture representations on AAM performance," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 2, 2006, pp. 328–331.

[19] M. Zhou, Y. Wang, X. Feng, and X. Wang, "A robust texture preprocessing for aam," in *Computer Science and Software Engineering, 2008 International Conference on*, vol. 2, Dec 2008, pp. 919–922.

[20] Y. Xu, X. Li, J. Yang, and D. Zhang, "Integrate the original face image and its mirror image for face recognition," *Neurocomputing*, vol. 131, no. 0, pp. 191 – 199, 2014.

[21] C. Lu, D. Zhao, and X. Tang, "Face recognition using face patch networks," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 3288–3295.

[22] Y. Yan, H. Wang, and D. Suter, "Multi-subregion based correlation filter bank for robust face recognition," *Pattern Recognition*, vol. 47, no. 11, pp. 3487 – 3501, 2014.

[23] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 1701–1708.

[24] E. Zhou, Z. Cao, and Q. Yin, "Naive-deep face recognition: Touching the limit of LFW benchmark or not?" *CoRR*, vol. abs/1501.04690, 2015.

[25] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 2879–2886.

[26] J. Arróspide, L. Salgado, and M. Camplani, "Image-based on-road vehicle detection using cost-effective histograms of oriented gradients," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 1182–1190, 2013.

[27] M. Dahmane and J. Meunier, "Emotion recognition using dynamic grid-based HoG features," in *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, March 2011, pp. 884–888.

[28] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 3169–3176.

[29] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 539–546.

[30] S. Milborrow, J. Morkel, and F. Nicolls, "The MUCT Landmarked Face Database," *Pattern Recognition Association of South Africa*, 2010.

[31] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B. Stegmann, "The IMM face database - an annotated dataset of 240 face images," Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Tech. Rep., may 2004.

[32] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.

[33] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[34] M. F. Karaaba, O. Surinta, L. R. B. Schomaker, and M. A. Wiering, "In-plane rotational alignment of faces by eye and eye-pair detection," in *Proceedings of the 10th International Conference on Computer Vision Theory and Applications*, 2015, pp. 392–399.