Descriptor Based Methods in the Wild

Lior Wolf, Tal Hassner, Yaniv Taigman

Descriptor Based Methods in the Wild

Lior Wolf[1] Tal Hassner[2] Yaniv Taigman[1,3]

[1] The School of Computer Science, Tel-Aviv University, Israel
[2] Computer Science Division, The Open University of Israel
[3] face.com, Tel-Aviv, Israel

Abstract. Recent methods for learning similarity between images have presented impressive results in the problem of pair matching (same/not-same classification) of face images. In this paper we explore how well this performance carries over to the related task of multi-option face identification, specifically on the Labeled Faces in the Wild (LFW) image set. In addition, we seek to compare the performance of similarity learning methods to descriptor based methods. We present the following results: (1) Descriptor-Based approaches that efficiently encode the appearance of each face image as a vector outperform the leading similarity based method in the task of multi-option face identification. (2) Straightforward use of Euclidean distance on the descriptor vectors performs somewhat worse than the similarity learning methods on the task of pair matching. (3) Adding a learning stage, the performance of descriptor based methods matches and exceeds that of similarity methods on the pair matching task. (4) A novel patch based descriptor we propose is able to improve the performance of the successful Local Binary Pattern (LBP) descriptor in both multi-option identification and same/not-same classification.

1 Introduction

The Labeled Faces in the Wild (LFW) database [1] offers a unique collection of annotated faces captured from news articles on the web. The dataset is published with a specific benchmark, which focuses on the face recognition task of *pair matching*. In this task, given two face images, the goal is to decide whether the two pictures are of the same individual. This is a binary classification problem, in which the two possible outcomes are "same" or "not-same".

The simple binary structure of Same-Not-Same classification simplifies the design of benchmark experiments. However, in many face recognition applications the task is quite different, and can be defined as follows: given a *gallery* containing labeled face images of several individuals (one or more face images for each person), classify a new set of *probe* images. The classification label can be "unknown person" or that of one of the individuals in the gallery.

The existence of the "unknown person" label is crucial at the application level. However, it makes benchmarking and experimental design much more challenging. First, the design should address the distribution of unknown faces. This can be done for example by providing a training set containing images of individuals not present in the gallery. Second, the likelihood of encountering an

unknown individual should be defined and may greatly affect the results. For these reasons, in this work, as in many other face recognition reports, a more limited task is studied: the multi-option identification task, where the "unknown person" is excluded.

Methods developed for each of the three identification tasks discussed above can be adopted for any of the other tasks. For example, one can compute scores between all gallery and all probe images using pair matching techniques and then use them for identification based on a winner takes all approach.

Since the LFW benchmark focuses on the pair matching problem, it is important to understand whether the reported success of algorithms on this benchmark carries over to other identification tasks. In addition, we test how descriptor based techniques, which are often studied in multi-option recognition tests, perform in the pair matching task.

Finally, we develop novel image descriptors that are able to improve performance both in the multi-option task and in the pair matching task. These descriptors are based on patch statistics, and we suggest using them in combination with other features.

2 Existing methods

In the previous section we have focused on the importance of the specific face recognition task. We now turn our attention to relevant face recognition algorithms and techniques.

Modern image similarity learning techniques Recently, some effort has been devoted to the estimation of visual similarities between two unseen images [2,3,4,5], and such methods have been applied to determine whether two images belong to the same person. One method [5] that has shown good results for uncontrolled imaging conditions uses Randomized Decision Trees [6] and Support Vector Machines. In the first image of the pair, image patches (fragments of the image) are selected at random locations. For each patch the most similar patch in the second image is searched at a nearby image location. A decision tree is trained to distinguish between pairs arising from matching images and those arising from non-matching images. Given a pair of unseen images, a Support Vector Machine (SVM) classifier is used to determine if they match by aggregating the Decision Tree output of many image patches. More specifically, the leaf (terminal classification node) of the decision tree is recorded for each pair of patches, and the SVM input consists of a binary vector that indicates if patches arising from a particular pair of images have reached a certain leaf.

Descriptor based methods for face recognition Face Images can be most readily described by statistics derived from their intensities. Intensities have thus served in many template-based methods. The intensities were sometimes normalized and sometimes replaced by edge responses [7]. More recently [8,9,10], Gabor wavelets have been used to describe the image appearance.

(a) (b) (c)

Fig. 1. (a) The LBP image-texture descriptor is computed locally at each pixel location. It considers a small neighborhood of a pixel, and thresholds all values by the central pixel's value. The bits which represent the comparison results are then transformed into a binary number. The histogram of these numbers is used as a signature describing the texture of the image. (b-c) Present an example image from the LFW data set, and its LBP encoding (different intensities representing different codes.)

A texture descriptor called Local Binary Patterns (LBP) [11,12,13]has been shown to be extremely effective for face recognition [14]. The most simple form of LBP is created at a particular pixel location by threshholding the 3×3 neighborhood surrounding the pixel with the central pixel's intensity value, and treating the subsequent pattern of 8 bits as a binary number (Fig. 1). A histogram of these binary numbers in a predefined region is then used to encode the appearance of that region. Typically, a distinction is made between uniform binary patterns, which are those binary patterns that have at most 2 transition from 0 to 1, and the rest of the patterns. For example, 1000111 is a uniform binary pattern while 1001010 is not. The frequency of all uniform LBPs is estimated, while all non-uniform LBPs, which are typically around 10% of the patterns in an image, are treated as equivalent and given only one histogram bin. The LBP representation of a given face image is generated by dividing the image into a grid of windows and computing histograms of the LBP values within each window. The concatenation of all these histograms constitutes the image's signature.

In this work we propose a patch-based descriptor that has some similarities to a variant of LBP called Center-Symmetric LBP (CSLBP) [15]. In CSLBP, eight intensities around a central point are measured. These intensities are spread evenly at a circle every 45 degrees starting at 12 o'clock. The binary vector encoding the local appearence at the central point, consists of four bits which contain the comparison of intensities to intensities on the symmetric position (180 degrees/ 6 hours difference).

Multi-block LBP [16] is an LBP variant that replaces intensity values in the computation of LBP with the mean intensity value of image blocks. Despite the similarity in terms, this method is very much different from our own. Multi-block LBP is shown to be effective for face detection, and in our initial set of experiments does not perform well for face recognition.

Patch-based approaches in recognition As mentioned above, a patch based approach [5] provides state of the art capabilities in similarity learning of faces and of general images. Other successful object recognition systems based on patches include the hierarchical system of [17].

The ability to detect local texture properties by examining the cross correlation between a central patch and nearby patches on both sides have been demonstrated in the texture segmentation system of [18]. In [19] a central patch was compared to surrounding patches to create a descriptor which is an extension of the shape-context [20] descriptor. The resulting descriptor has been shown to be highly invariant to image style and local appearance.

Improving descriptors by learning A large body of literature exists on the proper way of learning classifiers and distances for face recognition. In this work, we build upon basic classification methods, such as Linear Discriminant Analysis and One-Vs-All Support Vector Machine [21], which is the simplest way to construct multi-class classifiers out of binary SVM.

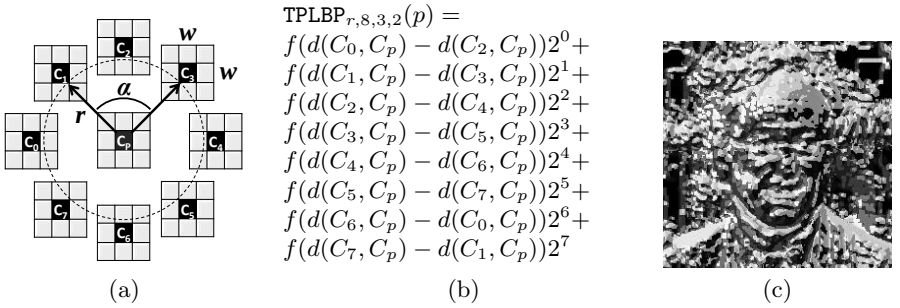
3 Novel patch based LBPs

The LBP descriptor and its variants use short binary strings to encode properties of the local micro-texture around each pixel. CSLBP [15], for example, encodes in each pixel the gradient signs at the pixel in four different angles. Here we propose a family of related descriptors each designed to encode additional types of local texture information. The design of these descriptors is inspired by the Self-Similarity descriptor of [19]. Specifically, we explore different ways of using bit strings to encode the similarities between neighboring patches of pixels, possibly capturing information which is complementary to that of pixel-based descriptors. Thus, employing patch based descriptors and pixel based ones in concert may improve the over-all accuracy of a classification system.

3.1 Three-Patch LBP Codes

As its name implies, the Three-Patch LBP (TPLBP) code is produced by comparing the values of three patches to produce a single bit value in the code assigned to each pixel. For each pixel in the image, we consider a $w \times w$ patch centered on the pixel, and S additional patches distributed uniformly in a ring of radius r around it (Fig. 2). For a parameter α, we take pairs of patches, α-patches apart along the circle, and compare their values with those of the central patch. The value of a single bit is set according to which of the two patches is more similar to the central patch. The resulting code has S bits per pixel. Specifically, we produce the Three-Patch LBP by applying the following formula to each pixel:

$$\texttt{TPLBP}_{r,S,w,\alpha}(p) = \sum_{i}^{S} f(d(C_i, C_p) - d(C_{i+\alpha \mod S}, C_p))2^i \qquad (1)$$



(a)

$$\begin{aligned}\texttt{TPLBP}_{r,8,3,2}(p) = \\ f(d(C_0,C_p)-d(C_2,C_p))2^0+ \\ f(d(C_1,C_p)-d(C_3,C_p))2^1+ \\ f(d(C_2,C_p)-d(C_4,C_p))2^2+ \\ f(d(C_3,C_p)-d(C_5,C_p))2^3+ \\ f(d(C_4,C_p)-d(C_6,C_p))2^4+ \\ f(d(C_5,C_p)-d(C_7,C_p))2^5+ \\ f(d(C_6,C_p)-d(C_0,C_p))2^6+ \\ f(d(C_7,C_p)-d(C_1,C_p))2^7\end{aligned}$$

(b)

(c)

Fig. 2. (a) The Three-Patch LBP code with $\alpha = 2$ and $S = 8$. (b) The TPLBP code computed with parameters $S = 8$, $w = 3$, and $\alpha = 2$. (c) Code image produced from the image in Fig. 1(b).

Where C_i and $C_{i+\alpha \mod S}$ are two patches along the ring and C_p is the central patch. The function $d(\cdot,\cdot)$ is any distance function between two patches (e.g., L_2 norm of their gray level differences) and f is defined as:

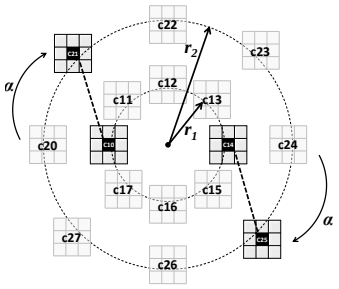
$$f(x) = \begin{cases} 1 \text{ if } x \geq \tau \\ 0 \text{ if } x < \tau \end{cases} \tag{2}$$

We use a value τ slightly larger than zero (e.g., $\tau = 0.01$) to provide some stability in uniform regions, similarly to [15]. In practice, we use nearest neighbor sampling to obtain the patches instead of interpolating their values, as this speeds up processing with little or no effect on performance.

Once encoded, an image's signature is produced similarly to that of the CSLBP descriptor [15]. The image is divided into a grid of none-overlapping regions and a histogram measuring the frequency of each binary code is computed for each such region. Each of these histograms are normalized to unit length, their values truncated at 0.2, and then once again normalized to unit length. An image is represented by these histograms concatenated to a single vector.

3.2 Four-Patch LBP Codes

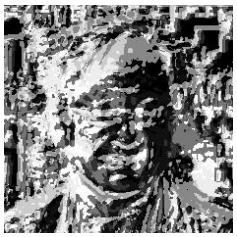
For every pixel in the image, we look at two rings of radii r_1 and r_2 centered on the pixel, and S patches of size $w \times w$ spread out evenly on each ring (Fig. 3). To produce the Four-Patch LBP (FPLBP) codes we compare two center symmetric patches in the inner ring with two center symmetric patches in the outer ring positioned α patches away along the circle (say, clockwise). One bit in each pixel's code is set according to which of the two pairs being compared is more similar. Thus, for S patches along each circle we have $S/2$ center symmetric pairs which is the length of the binary codes produced. The formal definition of the



(a)

$\texttt{FPLBP}_{r1,r2,8,3,1}(p) =$
$f(d(C_{10}, C_{21}) - d(C_{14}, C_{25}))2^0 +$
$f(d(C_{11}, C_{22}) - d(C_{15}, C_{26}))2^1 +$
$f(d(C_{12}, C_{23}) - d(C_{16}, C_{27}))2^2 +$
$f(d(C_{13}, C_{24}) - d(C_{17}, C_{28}))2^3$

(b)



(c)

Fig. 3. (a) The Four-Patch LBP code. Four patches involved in computing a single bit value with parameter $\alpha = 1$ are highlighted. (b) The FPLBP code computed with parameters $S = 8$, $w = 3$, and $\alpha = 1$. (c) Code image produced from the image in Fig. 1(b).

FPLBP code is as follows:

$$\texttt{FPLBP}_{r_1,r_2,S,w,\alpha}(p) = \sum_{i}^{S/2} f(\, d(C_{1i}, C_{2,i+\alpha \mod S}) - d(C_{1,i+S/2}, C_{2,i+S/2+\alpha \mod S}))2^i \tag{3}$$

The final image signature is produced by using the same two-step normalization procedure described in Section 3.1.

4 Face same-not-same classification

The Labeled Faces in the Wild (LFW) dataset has two versions: the original version and the funneled version, in which images are automatically aligned using the method of [22]. In all of our experiments we use the funneled version only. We plan to add the results on the original images in the future. There are also two proposed pair matching benchmarks. We report results on the benchmark protocol called "image restricted training", for which public results are available for the algorithm of [5] on the LFW web-site (`http://vis-www.cs.umass.edu/lfw/results.html`).

The image restricted pair matching benchmark is a challenging one. It consists of 6000 pairs, half of matching subjects and half not, which are divided into 10 equally sized sets. The benchmark experiment is repeated 10 times. In each repetition one set is used for testing and nine others are used for training. The goal of the tested method is to predict which of the testing pairs are matching, using only the training data (the decision is done one pair at a time, without using information from the other testing pairs).

We test the performance of descriptor based methods on this benchmark, and focus on two questions: (1) How well do these methods perform compared to the

average recognition rate of 0.7333 ± 0.006 obtained by the binaries of [5]; (2) How can learning be applied to descriptor based methods in the pair matching (same-not-same binary classification) setting.

4.1 Distance thresholding for pair matching

The most straightforward approach for pair matching using image descriptors is to consider the distance between the vectors which encode the appearance of the two images. i.e., given two face images I_1 and I_2 which are encoded using some image descriptor g as $g(I_1)$, $g(I_2)$, the pairs are considered to match if $d(g(I_1), g(I_2)) < T$, where d is a distance function and T is threshold.

The distance d can vary. We use the Euclidean distance and the Euclidean distance applied to the square roots of the values in $g(I_1)$. The motivation for the second distance is that our descriptor vectors consist mostly of histograms, and applying square root prior to the distance calculations corresponds to the Hellinger distance between probabilities [23].

In order to learn the threshold one can choose the threshold that gives the highest recognition score on the 5400 examples of the training set. A method that gives similar performance, and can be generalized to more than one distance score per pair is to employ a binary Linear SVM. In the single distance case, we train a SVM classifier on the $5,400$ one-dimensional vectors each containing the distance between the two images of a pair. Then, this classifier is used to predict whether the 600 test pairs are matching or not using similar 1D input vectors. This experiment is repeated for the 10 train/test splits, and we record mean recognition rate as well as the standard deviation of it.

A simple way to combine multiple image descriptors and multiple distances is to create a vector of distances and run Linear SVM on this vector. Indeed, we find that combining multiple distances together improves results considerably. In Table 1 below, we report the recognition rate for the two distances (Euclidean, Hellinger) and for each of the four descriptors: LBP, Gabor (C1), Three Patch LBP (TPLBP) and Four Patch LBP (FPLBP). The Gabor descriptor is the $C1$ descriptor of [24] used for face recognition in [9]. The parameters of the other descriptors are given in the appendix. Note that other descriptors such as CSLBP [15] and direct use of gray values did not produce good results and are omitted. Table 1 also contains the combined result obtained by classifying the vector of 8 distances at once using a SVM classifier.

As can be seen, a direct application of the distance function perform somewhat worse than the 0.7333 ± 0.0060 recognition rate achieved by the much slower random-tree based similarity prediction [5]. The LBP based descriptions perform similarly to one another, and the choice of distance does not change the results significantly. The combination of all descriptors and distances improves results beyond the best combination of a descriptor and a distance function.

Table 1. Mean (± standard error) recognition rates on the funneled pair matching benchmark of LFW (Image-Restricted Training, "view 1"). Each column represents a different distance measure. Each row corresponds to an image descriptor. The last row ("Combined") corresponds to training a SVM on the vector containing the 8 distances.

Image Descriptor	Euclidian Distance	Hellinger Distance
LBP	0.6767 ± 0.0068	0.6782 ± 0.0063
Gabor (C1)	0.6293 ± 0.0047	0.6287 ± 0.0046
TPLBP	0.6875 ± 0.0044	0.6890 ± 0.0040
FPLBP	0.6865 ± 0.0056	0.6820 ± 0.0055
Combined	0.7062 ± 0.0057	

4.2 Model learning for pair matching

In the distance based experiments, the training data was used in an improvised manner to classify very small vectors **after** the distances were taken. How can we use the training data to learn a matching score between pairs of descriptor vectors? There is a vast literature on supervised learning of similarity or of distance functions, e.g., [25,26,27]. We tried several of these methods using the DistLearnKit Matlab toolbox (`http://www.cse.msu.edu/~yangliu1/distlearn.htm`) and were unable to improve the classification accuracy.

An alternative learning framework is to consider the pair matching problem as related to the problem of learning from one example ("one-shot learning"; see [1] for a comparison of this problem and the unseen pair matching problem). We learn a model of the person in the first image of the pair and try to classify the second image. We then replace the roles of the images and repeat. The average prediction of the two classifications is taken as the matching score.

Our procedure required two training sets: a set A containing face images that is used for the negative examples during the one-shot learning, and a second set B of matching and non-matching pairs, which is used to learn the decision threshold as before. Given a pair of images, we vectorize them using an image descriptor. We then train two classifiers by using either one of the two images as the positive example, and the set A as the negative examples. Afterwards, we apply each classifier on the other image of the pair and obtain a score. These two scores are averaged to obtain a similarity score.

This process is repeated to all training image pairs of set B, and a threshold on the obtained scores is learned using a SVM. Given a new test pair, we build a one-shot classifier to each of the images using the same procedure and average the two classification scores. We then apply the SVM threshold on the average to obtain a prediction.

Since two classifiers need to be trained per training pair in set B, and since the number of "negative" images (set A) may be large, using SVM to learn the underlying one-shot classifiers may be computationally demanding if proper care is not taken. Instead, we suggest using a Linear Discriminant Analysis (LDA) classifier [28]. Note that due to the special structure of the problem, the LDA

computation can be performed very efficiently. The within-class covariance matrix is constant, depends only on the set A, and can be inverted once. Moreover, the direction of the LDA projection can be obtained by extracting the leading eigenvector of a 4×4 matrix regardless of the dimensionality of the image descriptors.

For the LFW image matching benchmark, we use in each repetition, out of the nine training sets, one to construct the set A and eight for B. The single negative image partition contains a total of 1,200 images. Note that none of the subjects in the 1,200 images appears in the test set, since the LFW benchmark is constructed to have the persons in the training and test splits mutually exclusive [1].

The results of the experiments are described in Table 2 Below. The descriptors used are the same as in the direct distance learning experiments. Here again we use either the original descriptor vectors, or their square root. In the latter case, instead of using the vector $g(I)$ we use $\sqrt{g(I)}$.

The 8 descriptor/mode scores in the table are obtained by training SVM on $4,800$ (8 sets) 1D vectors containing the average of the two: the LDA projection of the second image obtained using one-shot model learning of the first image and the LDA projection of the first image obtained from the second model. The "Combined" classification is based on learning and classifying the 8D vectors which are the concatenations of the eight 1D vectors.

Table 2. Mean (± standard error) recognition rates on the funneled pair matching benchmark of LFW (Image-Restricted Training, "view 1") using the per image model learning method. Each row corresponds to one image descriptor, and the columns represent the use of the original descriptor or of its square root. The last row ("Combined") corresponds to training a SVM on the vector containing the 8 prediction scores.

Image Descriptor	Original	Square Root
LBP	0.7343 ±0.0064	0.7463 ± 0.0048
Gabor (C1)	0.7112 ±0.0078	0.7157 ± 0.0076
TPLBP	0.7163 ±0.0082	0.7226 ± 0.0080
FPLBP	0.7175 ±0.0079	0.7145 ± 0.0078
Combined	0.7653 ± 0.0054	

The results of LBP alone, using the model learning framework are similar to the results obtained by the method of [5], which are the best results on the benchmark known to us. The results of combining all 8 scores outperform previous results significantly.

4.3 Hybrid method

A last experiment was done by combining the direct distance method and the one-shot model-learning method. This was done by concatenating the 8 distances

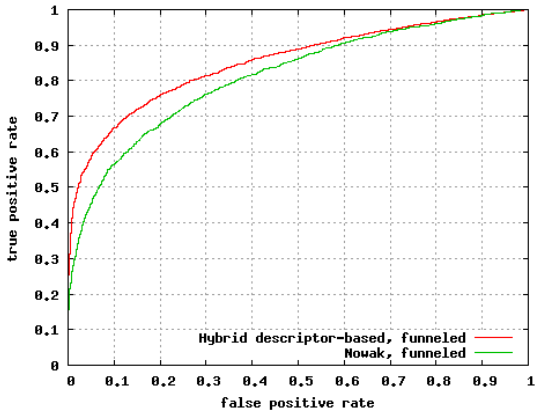


Fig. 4. ROC curves averaged over 10 folds of View 2 of the LFW data set. Each point on the curve represents the average over the 10 folds of (false positive rate, true positive rate) for a fixed threshold. The propose hybrid method is compare to the method of [5].

of the first method with the 8 scores obtained in the second method. The second method provides predictions for only 4800 of the training pairs, and the SVM classifier was trained on 4800 training $16-D$ vectors . A recognition rate of 0.7847 $\pm$ 0.0051 is obtained, which is significantly higher than each of the other methods, indicating that the distance base method and the model learning method employ different aspects of the data. The ROC plot of the hybrid method is depicted in Figure 4.

5 Face identification

Next, we evaluate the performance of the descriptor based methods and the similarity approach of Nowak and Jurie [5] on the task of image classification. To this end we use the LFW dataset, choosing only those subjects having enough images to contribute for both "probe" and "gallery" sets. In our experiments we use two images per person as probes and two as gallery. Thus, we employ a subset of the LFW image set which consists of the 610 subjects having at least four images. This subset contains a total of 6733 images.

The performance of the various methods as a function of the number of subjects (N) was compared. We perform 20 repetitions per experiment. In each, we select N random subjects and choose two random gallery images and a disjoint set of two random probes from each. Note that both N and the number of repetitions were limited by the computational demands of [5].

In order to learn the similarity function of [5], matching and non-matching pairs are required. We are not able to use the training pairs given with the LFW benchmark (either "view 1" or "view 2") since the images in those pairs and the images in our train/test splits overlap. Therefore, in order to create independent

training and testing sets we considered all subjects in the LFW database for which there are no more than three images per subject. There are 5139 such subjects (a total of 6500 images). From these images we extract the maximal number of matching pairs – 1652 pairs, and 2000 randomly selected non matching pairs. This training set is somewhat smaller than the data set of 5400 pairs (half matching, half non-matching) used in the same/not-same experiments. Once a similarity function is learned, given a probe image we compute its similarity to each of the gallery images. The probe image is then classified by a winner-take-all approach.

We test the following descriptors: LBP [11,12,13], CSLBP [15], C1 Gabor descriptors [24,9], Image intensities, TPLBP, and FPLBP. In some experiments, descriptors are combined by concatenating several descriptors into a single vector. Classification in the description based tests is performed by training a One-Vs-All multiclass Linear SVM on the gallery images and using it to classify the probe images one by one.

Fig. 5 (a) presents the performance of the three LBP variants (LBP, TPLBP, FPLBP), as well as the performance of the combination of the three. Also shown is the performance of the Gabor (C1) descriptor, and that of the method of [5]. To remove clutter, we omit the CS-LBP and the direct image intensities descriptors since they consistently under-perform the other descriptors. Adding Gabor to the LBP descriptors did not improve performance and those results are also omitted.

As can be seen, the similarity method of [5] outperforms the descriptor based methods and their combinations. This comes at a price, however, as the testing stage of the similarity based approach is more than an order of a magnitude slower than that required for descriptor based classification. Moreover, the similarity based approach uses a large data set for training, which was not utilized by the descriptor based methods. A natural question now arising is how to modify the learning stage of the descriptor based methods in order to make use of this training set?

The learning algorithm we use is One-Vs-All SVM, in which for each subject a SVM is trained using the subject's gallery images as positive examples and all other gallery images as negative examples. We modify this method such that an extra set of negative training images is used for each binary classifier learned. These additional negative training images are randomly selected form the set of images used to train the similarity based method (see above). Hence the descriptor based classifiers use no more training information than it is used for the method of [5].

Figure 5(b) and Figure 5(c) present classification results with the additional negative examples. As can be seen, the performance of the descriptor based methods improves and several of the descriptor methods outperform the similarity based approach. Classification time, however, did not change and remains significantly lower than that required by the similarity based approach.

In order to estimate the effect of varying the number of gallery images per person on the recognition rates, we have varied this parameter. Note that the

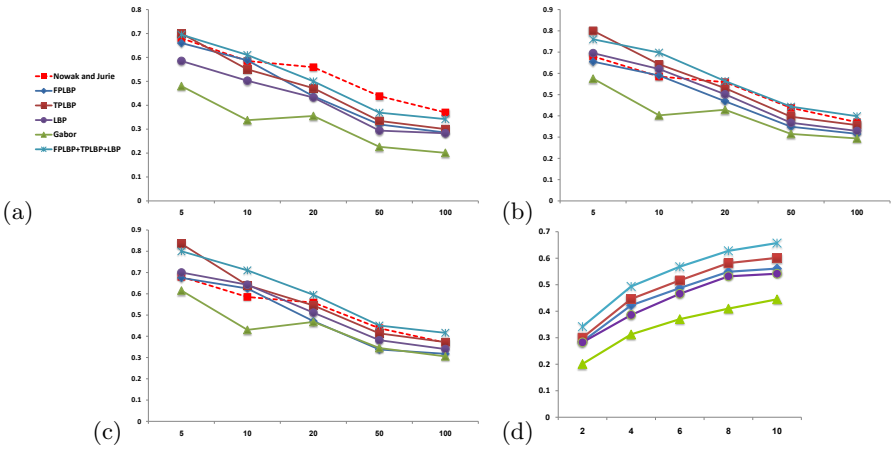


Fig. 5. Classification results in the multiple-option identification test. The X axis shows the number of classes (a-c) or the number of traninig examples per class (d). The Y axis shows the recognition rate. (a) Training of the descriptor based methods is done with no additional negative examples. (b) 100 additional negative examples are used; (c) 1000 additional negative examples are added to the gallery images during training. (d) performance, with no additional negative training images, as the number of gallery images per class increases (100 classes). The method of [5] is omitted from (d) since the running experiments did not finish in the time of submission. To avoid clutter we refrained from adding error bars. The standard deviations are of the magnitude of 0.05 in all cases. For example, the recognition rate of LBP for 50 classes and no extra negative examples 0.294 ± 0.0098

subset of the LFW dataset available becomes smaller as the number of gallery images increased. The results are reported in Figure 5(d). No extra set of negative images was used in this experiment.

6 Conclusions and future work

We evaluate the performance of a similarity-learning method in comparison to descriptor based methods. The similarity based method performs well on both pair matching and multiple option identification. Descriptor based methods, while performing worse than the similarity learning method when applied directly, can be combined with appropriate learning techniques in order to make better use of the training set and outperform the similarity based method.

There is still much work to be done. Some of the experiments are partial due to the computational complexity of the similarity based method; for the same reason, we were unable to conduct experiments on more than 100 classes. We also wish to study the significance of image alignment to the descriptor based methods. Finally, we are very interested in studying possible combinations of similarity based techniques with descriptor based techniques.

Acknowledgments

This research is supported by the Israel Science Foundation (grants No. 1440/06, 1214/06), the Colton Foundation, and a Raymond and Beverly Sackler Career Development Chair.

References

1. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, Technical Report 07-49 (2007)
2. Jain, V., Ferencz, A., Learned-Miller, E.: Discriminative training of hyper-feature models for object identification. In: British Machine Vision Conference, Edinburgh, UK (2006)
3. Ferencz, A., Learned-Miller, E.G., Malik, J.: Building a classification cascade for visual identification from one example. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. Volume 1. (2005) 286–293 Vol. 1
4. Ferencz, A., Learned-Miller, E., Malik, J.: Learning hyper-features for visual identification. In: Neural Information Processing Systems. Volume 18. (2004)
5. Nowak, E., Jurie, F.: Learning visual similarity measures for comparing never seen objects. In: IEEE Conference on Computer Vision and Pattern Recognition. (2007)
6. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning **36** (2006) 3–42
7. Brunelli, R., Poggio, T.: Face recognition: Features versus templates. IEEE Transactions on Pattern Analysis and Machine Intelligence **15** (1993) 1042–1052
8. Wiskott, L., Fellous, J.M., Krüger, N., von der Malsburg, C.: Face recognition by elastic bunch graph matching. PAMI **19** (1997) 775–779
9. Meyers, E., Wolf, L.: Using biologically inspired features for face processing. Internation Journal of Computer Vision **76** (2008) 93–104
10. Tan, X., Triggs, B.: Fusing gabor and lbp feature sets for kernel-based face recognition. In: Analysis and Modelling of Faces and Gestures. Volume 4778 of LNCS., Springer (2007) 235–249
11. Ojala, T., Pietikainen, M., Harwood, D.: A comparative-study of texture measures with classification based on feature distributions. Pattern Recognition **29** (1996) 51–59
12. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24** (2002) 971–987
13. Ojala, T., Pietikäinen, M., Mäenpää, T.: A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. In: ICAPR '01: Proceedings of the Second International Conference on Advances in Pattern Recognition, London, UK, Springer-Verlag (2001) 397–406
14. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006) 2037–2041
15. Heikkilä, M., Pietikäinen, M., Schmid, C.: Description of interest regions with center-symmetric local binary patterns. In: Computer Vision, Graphics and Image Processing, 5th Indian Conference. (2006) 58–69

16. Zhang, L., Chu, R., Xiang, S., Liao, S., Li, S.: Face detection based on multi-block lbp representation. In: IAPR/IEEE International Conference on Biometrics. (2007)
17. Ullman, S., Sali, E.: Object classification using a fragment-based representation. In: the First IEEE International Workshop on Biologically Motivated Computer Vision, London, UK, Springer-Verlag (2000) 73–87
18. Wolf, L., Huang, X., Martin, I., Metaxas, D.: Patch-based texture edges and segmentation. In: European Conference on Computer Vision. (2006) 481–493
19. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. CVPR (2007) 1–8
20. Belongie, S., Malik, J., Puzicha, J.: Shape context: A new descriptor for shape matching and object recognition. In Leen, T.K., Dietterich, T.G., Tresp, V., eds.: Advances in Neural Information Processing Systems 13, MIT Press (2001) 831–837
21. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. JMLR **1** (2001) 113–141
22. Huang, G., Jain, V., Learned-Miller, E.: Unsupervised joint alignment of complex images. In: IEEE International Conference on Computer Vision. (2007)
23. Pollard, D.E.: A user's guide to measure theoretic probability. Cambridge University Press (2002)
24. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. Nature Neuroscience **2** (1999) 1019–1025
25. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning with application to clustering with side-information. In Thrun, Obermayer, K., eds.: NIPS. MIT Press, Cambridge, MA (2003) 505–512
26. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relations. In: ICML. (2003)
27. Weinberger, K., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. NIPS **18** (2006) 1473–1480
28. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning. Springer (2001)
29. (`http://www.ee.oulu.fi/mvg/page/lbp_matlab`)

Appendix A: The parameters used in our experiments

Preprocessing For descriptor based methods, all LFW-funneled images used in our tests were cropped to 110×115 pixels around their center. Following [15] we further applied an adaptive noise-removal filter (Matlab's `weiner2` function) and normalized the images to saturate 1% of values at the low and high intensities. The similarity based method of [5] does not seem to benefit from the preprocessing stage, and we employ it on the original images.

Descriptor parameters Some parameter tuning was done on "view 1" of the LFW dataset, which is intended for such tests. Naive gray-level descriptors are produced by rescaling the cropped and normalized images to half their original size, and sampling all pixels. The image descriptors for all LBP variants are constructed by concatenating histograms produced for 35 non-overlapping blocks of up to 23×18 codes. To produce the LBP descriptors we use the MATLAB source code available from [29]. Results are obtained with "uniform" LBP of radius 3 and considering eight samples. The parameters of the patch based LBP descriptors are $r_1 = 2$, $S = 8$, $w = 5$ for TPLBP, and $r_1 = 4$, $r_2 = 5$, $S = 3$, $w = 3$ for FPLBP.