Multiple One-Shots for Utilizing Class Label Information

Yaniv Taigman[1,3]
yaniv@face.com
Lior Wolf[1]
wolf@cs.tau.ac.il
Tal Hassner[2]
hassner@openu.ac.il

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

Abstract

The One-Shot Similarity measure has recently been introduced as a means of boosting the performance of face recognition systems. Given two vectors, their One-Shot Similarity score reflects the likelihood of each vector belonging to the same class as the other vector and not in a class defined by a fixed set of "negative" examples. An appealing aspect of this approach is that it does not require class labeled training data. In this paper we explore how the One-Shot Similarity may nevertheless benefit from the availability of such labels. We make the following contributions: (a) we present a system utilizing subject and pose information to improve facial image pair-matching performance using multiple One-Shot scores; (b) we show how separating pose and identity may lead to better face recognition rates in unconstrained, "wild" facial images; (c) we explore how far we can get using a single descriptor with different similarity tests as opposed to the popular multiple descriptor approaches; and (d) we demonstrate the benefit of learned metrics for improved One-Shot performance. We test the performance of our system on the challenging Labeled Faces in the Wild unrestricted benchmark and present results that exceed by a large margin results reported on the restricted benchmark.

1 Introduction

In this paper we focus on the problem of image pair-matching (same/not-same) using unconstrained images of faces. Specifically, given two images, both containing faces, both taken under natural conditions (*i.e.* "in the wild"), our goal is to answer the following simple question: are these two images of the same person, or not? Recently, the Labeled Faces in the Wild (LFW) face image data set was released as a benchmark for this problem [15]. The LFW benchmark provides two testing protocols. In the first, called the "image restricted training", training is performed using pairs of images labeled as either "same" (they both portray the same person) or "not-same", but no additional class labels are available. A second protocol, the "unrestricted" protocol, also provides the subject identity for each image.

The restricted protocol is attractive, since many similarity learning algorithms are based on learning equivalence relations. At the same time, the limited amount of available information makes this protocol potentially more challenging. Results reported to date on the LFW

BMVC 2009 doi:10.5244/C.23.77

benchmarks employ the restricted protocol * . These results show a gradual improvement over time (*e.g.* [11, 14, 18, 24, 27], see current results in [6]). Here we focus on the second protocol where class labels are also available for each training image. We present a method which utilizes this extra information to improve classification scores well beyond the known state-of-the-art.

Our method is based on the One-Shot Similarity (OSS) score, originally introduced by [27]. OSS compares two vectors by considering an auxiliary set of examples, a "negative" set, containing vectors not belonging in the same class as the vectors being compared. The OSS score of two vectors is computed by first learning a model for each vector, discriminating it from this set of negative examples. These models are then used to determine if each vector shares the same label as its counterpart or belongs in the negative set. The average of these two prediction scores is the OSS score for the two vectors.

The OSS score presented in [27, 28] does not use label information, and thus cannot benefit from the extra labeling available in the unrestricted protocol. Here, we utilize these labels to compute multiple OSS scores for a pair of images. Each score out of the multitude of scores is associated with one individual in the training set, and uses the face images of that individual as the negative set. Furthermore, pose information obtained automatically through the use of a facial alignment algorithm is used as a second source of labels for generating more One Shot scores. This approach provides a number of benefits:

1. By combining multiple OSS scores we are able to substitute the use of multiple descriptors for each image, with a single descriptor and multiple similarities. We thus provide evidence to the power of a single descriptor as an image representation. By doing so, the solidity of a classification pipeline can be easily demonstrated by showing that it works regardless of the underlying image representation.

2. We argue that the LFW set, and indeed any similar set of facial images under unconstrained viewing conditions, contains a bias towards pose. Specifically, pose similarities outweigh subject identity similarities, leading to matching based on pose rather than identity. We show how by employing multiple OSS scores we are able to decouple pose and identity, and avoid this bias.

3. We employ learned metrics to improve the discriminative performance of each OSS score. We show that combining OSS scores with learned metrics can significantly improve the performance of a pair-matching classifier.

1.1 Related work

The Labeled Faces in the Wild Benchmark. The LFW database [15] offers a unique collection of around 13,000 annotated faces automatically captured from news articles on the web. The images in this set are partitioned into 5,749 individuals each having anywhere from one to 150 images. These images are paired into 6,000 pairs of images, half labeled "same" and half "not same". These pairs are further divided between ten test splits mutually exclusive in the subject identities (a person appearing in one will not appear in the other nine). Although there is a growing number results published on this data set, using the restricted protocol (where training may be performed using only same/not-same labels [6]), we know of no published result on the unrestricted protocol employed here.

*First results are being published independently for the unrestricted protocol [11]. We compare with these results in section 5.

Similarity measures. The literature on similarity functions, their design and applications, is extensive. Some of the similarity measures proposed in the past have been hand crafted (e.g., [5, 30]). Alternatively, a growing number of authors have proposed tailoring the similarity measures to available training data by applying learning techniques (e.g., [7, 9, 12, 17, 25, 26, 29]). In all these methods testing is performed using models (or similarity measures) learned beforehand, whereas the OSS score used here learns discriminative models exclusive to the vectors being compared.

An independent contribution [18] (in publication) studies trait- or identity-based classifier-outputs as a feature for identification. Unlike our work, it encodes one vector per face image and not per pair of images.

As a part of our pipeline, we use a particular metric learning method called Information Theoretic Metric Learning (ITML) [16, 17]. ITML is a supervised metric learning technique for learning a Mahalanobis distance. It uses pairs of examples belonging to the same class (in our case, images of the same person) which are constrained to have similarities below a specified threshold. The similarities of pairs of points from different classes are constrained to have similarities above a second threshold. A regularization term ensures that the learned metric is similar to the original metric.

The ITML method was shown to be extremely potent in Computer Vision problems [16]. In our work we employ the implementation provided by the authors [17], where all parameters are set to default values. In order to speed up the method, we prune the constraints such that $10,000$ same constraints and $10,000$ not-same constraints are used. This is about four times the number of constraints available in each training split of the restricted LFW benchmark, where 2,700 same pairs and a similar amount of not-same pairs are given.

2 The One-Shot Similarity score

To compute a One-Shot similarity score, a set **A** of "negative" training examples is required. These are examples which have different labels (identities) from the ones we will later compare. This set is used to compare two vectors by asking if each vector is more likely to have the same label as the other, or alternatively belong in the negative set. Given two vectors **I** and **J** their *symmetric* One-Shot score is computed as follows (see also Fig. 1). First a discriminative model is learned by taking **A** as the negative example set and **I** to be a single positive example (hence the term "One-Shot"). Then, this model is used to obtain a classification prediction score for **J**, `Score1`. The nature of this prediction score depends on the classifier used. Using linear SVM as a classifier, for example, this value can be the signed distance from the separating hyperplane. Intuitively, this value gives us a measure of how likely **J** is to belong to the same class as **I**, given the one-shot training. The process is then repeated, this time switching the roles of **I** and **J** and obtaining `Score2`. The OSS score is defined to be the average of these two scores.

```
function One-Shot-Similarity(I, J,  A)
    Model1 = train(I, A)
    Score1 = classify(J, Model1)

    Model2 = train(J, A)
    Score2 = classify(I, Model2)

    return ½(Score1+Score2)
```

Figure 1: Computing the symmetric One-Shot Similarity score for two vectors, **I** and **J**, given a set **A** of negative examples.

A note regarding complexity. Two classifiers need to be trained each time two images are compared. Depending on the classifiers used and the particulars of the implementation this may lead to unreasonable computational costs. Here, similarly to [27] LDA is used. Due to the particular nature of the LDA problem solved in order to compute the scores, the within-class covariance matrix is constant and can be inverted once. The direction of the LDA projection is then obtained by a simple direction computation.

3 The Multiple One-Shot Similarity score

The OSS score does not employ labeling information. It can therefore be applied to a variety of vision problems where collecting unlabeled data is much easier than the collection of labeled data. However, when labeled information is available, the OSS score does not benefit from it. Here, we suggest employing label information by computing the One-Shot Score multiple times. Using the label information we split the set $\mathbf{A}$ of examples to n sets, $\mathbf{A_i} \subset \mathbf{A}, i = 1..n$, each one containing examples from a single class. The OSS is then computed multiple times, where each time only one subset $\mathbf{A_i}$ is used.

The rational for the split is as follows. The set $\mathbf{A}$ contains variability due to a multitude of factors including pose, identity and expression. During the computation of the (regular) OSS one tries to judge whether $\mathbf{J}$ is more likely to belong to the set containing just the point $\mathbf{I}$ or to the set $\mathbf{A}$. $\mathbf{I}$ contains one person captured at one pose under a particular viewing condition. The classifier trained to distinguish between the two sets can distinguish based on any factor, not necessarily based on the identity of the person.

Now consider the case where the OSS score is applied to a set $\mathbf{A_i}$ which contains a single person, possibly at multiple poses and conditions. In this case the classifier is more likely to distinguish based on identity since all other factors vary within the set $\mathbf{A_i}$. Thus, the score better reflects the desired property of discriminating based on the person in the photograph.

The separation between identity and other factors can be further enhanced by considering OSS scores based on sets which have one of these factors approximately constant. For example, if the set $\mathbf{A_i}$ contains people viewed in a certain pose, which is different than the one in $\mathbf{I}$, the resulting score would discriminate based on pose. This by itself is not what we seek. However, when combined with other scores to create a multitude of scores, a high pose-based OSS score can indicate that the visual similarity is not necessarily based on identity. Conversely, a low pose-based score indicates that an overall low similarity does not rule the same label. Note that pose-based OSS scores behave similarly to the regular OSS when $\mathbf{I}$ and $\mathbf{J}$ are of a pose similar to the images that are in $\mathbf{A_i}$.

The profile of similarities obtained by the vector of multiple OSS scores is passed during training to a classifier which extracts these relations. Figure 2 demonstrates the various OSS scores for pairs of similar/non-similar identities with similar and non similar poses.

4 The proposed system

Figure 3 presents a schematic description of our face image pair-matching system. It describes the following stages, each detailed in subsequent sections. Given two images, we first start by (a) aligning the two images (Section 4.1) and (b) computing a descriptor representation for each one (Section 4.2). (c) Applying the Information Theoretic Metric Learning technique (Section 4.3). (d) Multiple OSS scores are then computed using different nega-

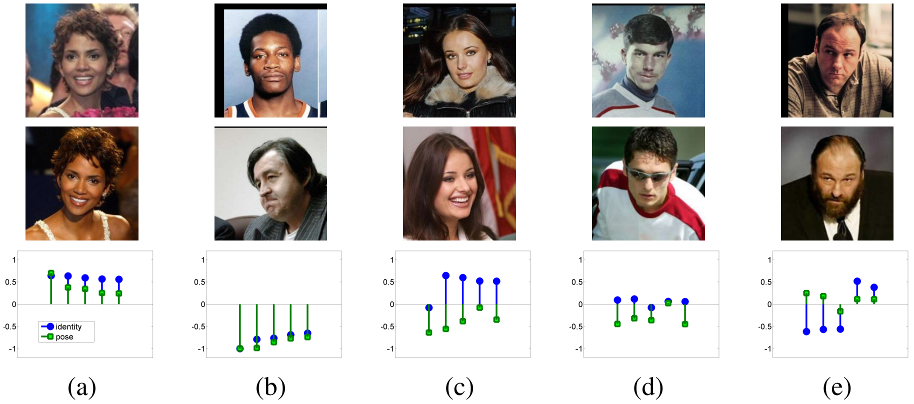


Figure 2: Each group contains two images and 10 sample multiple OSS scores. Identity based multiple OSS scores are plotted with circle markers and pose based are with squares. As can be seen the value of each type of OSS score is a good indication of the type of similarity between the images of the pair. (a) Same person, same pose. (b) Different persons and pose. (c) Same person, different pose. (d) Different persons, same pose. (e) Same person and pose, however, a mode of variability not modeled in the system is present.

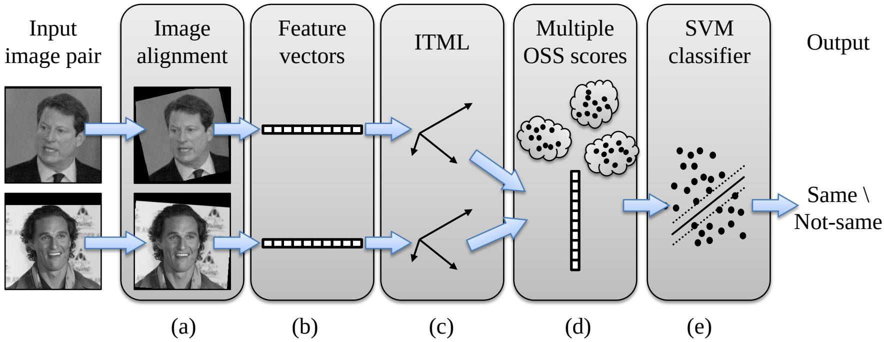


Figure 3: A schematic description of our system. Please see text for more details.

tive example sets, resulting in a single vector of similarity scores (Section 4.4). (e) This vector is then fed to a linear SVM classifier [8] to obtain a final same/not-same prediction (Section 4.5).

Note that in our tests we focused on a single descriptor. This is in contrast to the current trend of combining multiple descriptors in order to obtain good benchmark results [18, 24, 27]. Indeed, it is one of our goals to see "how far" one can get with a single descriptor.

4.1 Alignment

The LFW data set is available as both aligned and unaligned images. The aligned version was produced by using the "Funneling" technique of [13]. Results reported on these two sets have shown a small performance gain obtained by using the Funneled version (*e.g.* [20]). Here, we produced our own aligned version of the database, by using our own face alignment system. Such systems have been used before within the LFW benchmark, for example, [14] employs the MERL face alignment system and [11] uses facial feature point detectors. Note that the aligned images were not filtered and the alignment results are used even if this step fails.

The alignment system is based on localization of facial feature points in the spirit of [3], with several important differences which are beyond the scope of this work. In particular we compute a similarity transformation which attempts to bring seven specific fiducial points (corners of the eyes, mouth and the tip of the nose) to fixed locations. These points are automatically detected using a feature detector trained on a set of face images with manually marked fiducial points. This set of images does not intersect in images or in identity with the LFW data set. We omit further details as this was not the focus of this work.

4.2 Feature vector extraction

We examine a number of descriptors. (1) a SIFT [19] based face representation, in which the SIFT descriptor is produced over a 24-cell grid partition of the image. Each cell represented by a 128-dimensional histogram of gradient orientations, giving us a total descriptor size of 3072 values. (2) The LBP face descriptor [21, 22, 23]. (3) The Three-Patch and (4) the Four-Patch LBP [27]. All LBP descriptors (2-4) were constructed by concatenating histograms produced for 35 non-overlapping blocks of up to 23×18 binary codes. To produce the LBP descriptors the MATLAB source code available from [1] was used. Results are obtained with "uniform" LBP of radius 3 and considering eight samples. The patch based LBP descriptors were produced using code available from [2]. The parameters of these descriptors are $r_1 = 2$, $S = 8$, $w = 5$ for TPLBP, and $r_1 = 4$, $r_2 = 5$, $S = 3$, $w = 3$ for FPLBP.

4.3 Information Theoretic Metric Learning

A popular means of utilizing the label information for classification problems is by using techniques for supervised learning of similarity or distance functions, e.g., [4, 26, 29]. In our tests we used the Information Theoretic Metric Learning (ITML) technique of Davis *et al*. [10]. We use the code made available by the authors at [10], setting the regularization term to the default value of 0.2, and choosing the lower and upper thresholds to be the default lower and upper tenth percentile.

4.4 Multiple OSS scores

Multiple OSS scores were produced for each pair of descriptors, by considering different negative training sets $\mathbf{A_i}$. Each such set reflecting either a different subject or a different pose.

To produce the negative set partitions based on subject identity, we use the unrestricted protocol to retrieve subject labels, and select 20 subjects having at least ten images each.

We improve the robustness of our system to pose changes by adding additional OSS scores computed with example sets representing different poses. We produce these sets automatically as follows. We use the coordinates of the seven fiducial points used for alignment (Section 4.1) after the alignment stage. Since the similarity transform can only align rotation and scale, faces of different poses (and shapes) differ in the aligned coordinates. We project the 14 dimensional vectors onto a one-dimensional line using standard PCA on the training set. This line is then partitioned into 10 bins of equal number of images. Each bin then represents a single pose. Figure 4(a) shows an example set of images, all clustered together as having the same pose. Figure 4(b) presents a single representative from each pose set, demonstrating the different pose sets automatically produced by this simple approach.

(a)

(b)

Figure 4: Partitioning into pose. (a) Images in the same pose bin. (b) One example from each pose bin ordered by value. In each subfigure the top row contains the original images and the bottom row contains the aligned versions.

4.5 Same/Not-same classification

The vectors of similarity values produced by computing multiple OSS scores are then fed to a linear binary Support Vector Machine classifier, previously trained on similar training vectors. We use a value $C = 1$ as a single parameter for the SVM algorithm. The value output by the classifier is our final classification result.

5 Results

We test the proposed system on the 10 folds of view 2 of the LFW dataset. Unlike other contributions we focus on the unrestricted protocol, in which the identity of the person in each image is known. 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. Testing data is never used for training, and the decision is done one pair at a time, without using information from the other testing pairs.

Throughout the experiments a default set of parameters is used, unless otherwise noted. Each descriptor is projected to 150 dimensions by means of PCA, which is trained using the training data only. The ITML transformation matrix G (the factor of the learned Mahalanobis matrix $S = G^\top G$) is learned for each split separately using $10,000$ same and $10,000$ not same pairs picked at random from among the training points. Note that while this subset is expected to intersect the restricted training set of $2,700$ pairs of each type, it is not strictly a superset of it.

Table 1: Recognition accuracy (± Standard Error) of various combinations of classifiers and descriptors. See text for details.

Image Descriptor	SIFT		LBP		TPLBP	FPLBP
→ version		SQRT		SQRT		
Euclidean distance	.7023±.0067	.7082±.0068	.6795±.0072	.7085±.0076	.6893	.6835
OSS	.7708±.0048	.7817±.0058	.7670± .0051	.7917± .0042	.7598	.7120
MultOSS ID	.7701±.0032	.7831±.0012	.7623± .0072	.7963± .0022	.7602	.7192
MultOSS pose	.7672±.0133	.7773±.0009	.7614± .0023	.7883 ±.0061	.7581	.7122
MultOSS ID + pose	.7741±.0012	.7891±.0021	.7723±.0012	.8001±.0032	.7682	.7222
ITML	.7960±.0097	.8063± .0077	.7665± .0030	.8167± .0054	.7793	.7223
ITML + OSS	.7990± .0063	.8113± .0070	.7867± .0050	.8175± .0055	.7803	.7160
ITML + MultOSS ID	.8320± .0077	.8397±.0070	.8173 ±.0051	.8517± .0061	.8055	.7465
ITML + MultOSS pose	.8153± .0081	.8238± .0082	.7998± .0054	.8340± .0071	.7828	.7325
ITML + MultOSS ID + pose	.8348± .0070	.8397± .0070	.8173± .0054	.8507± .0058	.8075	.7557

In the OSS experiments, the set **A** includes all training images (this number varies since the same image may appear in multiple pairs). This is in contrast to [27], where only images from one split were used for this purpose. In the Multiple OSS experiments we used persons with at least ten images in the training set (note that the identities of the training and the testing splits do not intersect), a number which has a minor effect on the results as shown below. 20 random classes (person identities) are used to build 20 identity based OSS scores.

The pose space is clustered into 10 different poses, and 10 pose based OSS scores are constructed. It seems that this value is not optimal, as twice as many scores provide better results. However, we chose it at the beginning of the experimental design and stick to it as the default value.

The results are presented on four different descriptors LBP, SIFT, TPLBP, and FPLBP as described in section 4.2. For LBP and SIFT, which are the more useful descriptors we provide results for both square root of their value and the original values. For TPLBP and FPLBP, which are somewhat weaker descriptors, only the non square root results were obtained to date. The square root preprocessing for histogram based descriptors such as the ones used in this work is often preferable since it mimics the Hellinger distance between probability distribution.

The results are described in Table 1. For each descriptor we provide results for the following methods, building from parts of the system in Figure 3 to the complete system: Euclidean distance (for sqrt descriptors this becomes the Hellinger distance), OSS scores, Multiple OSS based on identity, Multiple OSS based on pose, Multiple OSS both types combined, ITML, ITML followed by OSS, ITML followed by Multiple OSS based on identity, based on pose, and a combination of the two Multiple OSS sources. In all cases SVM is used to either learn the threshold, or to combine different parts of the score together, as describe above in Section 4.5.

As can be seen both OSS and ITML by themselves improve results considerably. We note that OSS, although unsupervised, provides a large portion of the benefit obtained from ITML. Moreover, the contributions of OSS and ITML accumulate. We also note that Multiple OSS of either type is not better than OSS on the original feature vectors, however, they provide a considerable boost after applying ITML. We attribute this to the fact that applying OSS with small sets of extra negatives ("*A*") is less effective when the underlying metric is not very good.

To evaluate the effect of the various parameters on the results, we run a second set of experiments in which we focus on the LBP descriptor (square root version), as depicted in Figure 5. We vary from left to right the number identities used within the Multiple OSS stage, the minimal number of images per identity used, the number of poses used within the Multiple OSS stage, and the PCA dimension. The PCA dimension results show final system performance, the other results show results of either Multiple OSS identity or Multiple OSS pose.

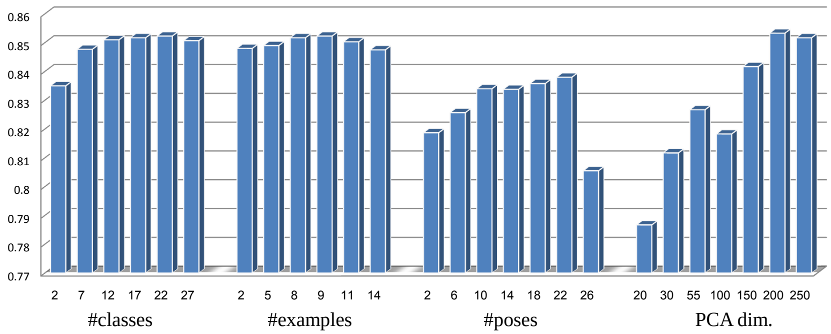


Figure 5: Testing the sensitivity to the parameters on the square-root LBP descriptor. As can be seen, the results are stable for a large range of parameters.

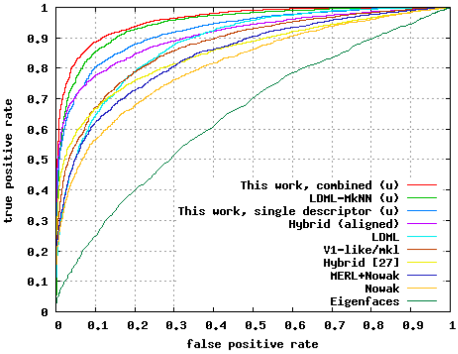
Finally, we present the results on the LFW benchmark compared to other contributions in Figure 6. Note that unlike all other methods, excluding the LDML-MkNN [11], we use the unrestricted protocol. We present results for our best method (using default parameters), the one based on the square root of the LBP descriptor (0.8517 ± 0.0061 S_E). Also, we further combined 16 Multiple OSS scores, that is 8 descriptors (SIFT, LBP, TPLBP, and FPLBP, as well as all four with square root applied) each trained separately using the ITML + Multiple OSS ID method and the same 8 but with the pose-based multiple shots, into one vector of $16D$. This vector was then classified using a linear SVM classifier (as in the Hybrid of [27]). The result we obtained for this combination was 0.8950 ± 0.0051 S_E, which is the best result reported so far.

As an additional baseline result, we provide also in Figure 6 our score obtained using the same Hybrid system of [27] on the restricted protocol, but this time using our own alignment technique (Section 4.1), instead of Funneling [13]. These results, unlike [27], do include the SIFT descriptor. Our mean classification accuracy for this result is 0.8398 ± 0.0035 S_E, which is the best result reported so far on the restricted protocol.

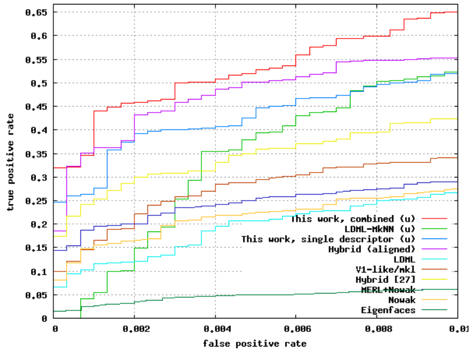
6 Discussion

Unsupervised and supervised dimensionality reduction and metric learning techniques are often combined with the goal of reducing the computational complexity of the more demanding supervised methods by a preprocessing stage. For example, before computing ITML, it is beneficial to reduce the dimensionality of the problem by PCA. An alternative would be to employ kernels (if the number of training examples is small), or to work in an online manner.

As practitioners, we seldom find a situation where unsupervised learning improves the *accuracy* of consequent supervised learning, although this is not unheard of, for example,



(a)



(b)

Figure 6: 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. (a) Full ROC curve. (b) A zoom-in onto the low false positive region. The proposed method (single and multiple descriptors) is compared to the best algorithms as reported on `http://vis-www.cs.umass.edu/lfw/results.html`. These algorithms include the combined Nowak+MERL system [14], the Nowak method [20], the hybrid method of [27] and as well using our alignment technique (Section 4.1), the V1-like/MKL method of [24] and the recent LMDL/MkNN methods of [11]. (u) indicates ROC curve is for the unrestricted setting.

the combination of Kernel PCA and boosting is particularly popular. More rare is a situation where an unsupervised technique follows a supervised one.

It is therefore interesting to note that the unlabeled One-Shot technique considerably improves over the ITML metric learning results. It seems that these two methods provide different benefits. ITML provides a global alignment of the coordinate system, while the OSS provides a very localized decision mechanism. The multiple OSS technique, which is a supervised method improves results even further, demonstrating that the label information is not fully utilized in the ITML method.

It is interesting to note the role of pose in classification of “natural” news images. Alignment has been a part of the LFW main research dogma since the first results, with both unsupervised and supervised algorithms used for alignment. While face alignment surely improves results, it is not perfect. The residual pose information is actually useful in recognizing a person as many persons tend to be photographs at preferred poses. In fact, a classifier built on top of the face coordinates of the points used for alignment (a total of 14 coordinates) achieves a recognition rate of 0.6058 (SE 0.0082) on the LFW benchmark.

While the pose bias is helpful at first, it hurts performance at the high recognition levels. Errors returned by the best existing methods are mostly same person on different poses or different people with similar poses. It is therefore useful to factor the obtained similarity scores by identity and pose. In this work we propose a novel way of doing so. Multiple OSS by identity already provides a considerable improvement to this end, and so does the pose based Multiple OSS. Note however, that a combination of both is not that much better than either one separately.

Acknowledgments

Lior Wolf is supported by the Israel Science Foundation (grant No. 1440/06, 1214/06), the Colton Foundation, and The Ministry of Science and Technology Russia-Israel Scientific Research Cooperation.

References

[1] http://www.ee.oulu.fi/mvg/page/lbp_matlab.

[2] http://www.openu.ac.il/home/hassner/projects/Patchlbp/.

[3] M. Everingham J. and Sivicand and A. Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2006.

[4] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *International Conference on Machine Learning (ICML)*, 2003.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *The Neural Information Processing Systems (NIPS)*, 2001.

[6] LFW benchmark results. http://vis-www.cs.umass.edu/lfw/results.html.

[7] M. Bilenko, S. Basu, and R.J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *International Conference on Machine Learning (ICML)*, 2004.

[8] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[9] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor. On kernel-target alignment. In *The Neural Information Processing Systems (NIPS)*, 2002.

[10] J.V. Davis, B. Kulis, P. Jain, S. Sra, and I.S. Dhillon. Information-theoretic metric learning. In *International Conference on Machine Learning (ICML)*, pages 209–216, June 2007.

[11] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *International Conference on Computer Vision (ICCV)*, September 2009.

[12] T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin based distance functions for clustering. In *International Conference on Machine Learning (ICML)*, 2004.

[13] G.B. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *IEEE International Conference on Computer Vision*, 2007.

[14] G.B. Huang, M.J. Jones, and E. Learned-Miller. Lfw results using a combined nowak plus merl recognizer. In *Faces in Real-Life Images Workshop in ECCV*, 2008.

[15] 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, Technical Report 07-49, October, 2007, 2007.

[16] P. Jain, B. Kulis, and K. Grauman. Fast similarity search for learned metrics. In *University of Texas at Austin, Technical Report #TR-07-48*, September 2007.

[17] P. Jain, B. Kulis, and K. Grauman. Fast Image Search for Learned Metrics. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pages 1–8, 2008.

[18] N. Kumar, A. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2009.

[19] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[20] E. Nowak and F. Jurie. Learning visual similarity measures for comparing never seen objects. In *CVPR*, June 2007.

[21] T. Ojala, M. Pietikainen, and D. Harwood. A comparative-study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1), 1996.

[22] T. Ojala, M. Pietikäinen, and T. Mäenpää. A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. In *International Conference on Advances in Pattern Recognition (ICAPR)*, 2001.

[23] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987, 2002. ISSN 0162-8828.

[24] N. Pinto, J.J. DiCarlo, and D.D. Cox. How far can you get with a modern face recognition test set using only simple features? In *CVPR*, 2009.

[25] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. In *ECCV*, 2002.

[26] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. *The Neural Information Processing Systems (NIPS)*, 2006.

[27] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Faces in Real-Life Images Workshop in ECCV*, 2008.

[28] L. Wolf, T. Hassner, and Y. Taigman. The one-shot similarity kernel. In *International Conference on Computer Vision (ICCV)*, September 2009.

[29] E.P. Xing, A. Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *The Neural Information Processing Systems (NIPS)*, 2003.

[30] H. Zhang, A.C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.