Face Recognition in Unconstrained Videos with Matched Background Similarity

Lior Wolf[1] Tal Hassner[2] Itay Maoz[1]

[1] The Blavatnik School of Computer Science, Tel-Aviv University, Israel

[2] Computer Science Division, The Open University of Israel

Abstract

Recognizing faces in unconstrained videos is a task of mounting importance. While obviously related to face recognition in still images, it has its own unique characteristics and algorithmic requirements. Over the years several methods have been suggested for this problem, and a few benchmark data sets have been assembled to facilitate its study. However, there is a sizable gap between the actual application needs and the current state of the art. In this paper we make the following contributions. (a) We present a comprehensive database of labeled videos of faces in challenging, uncontrolled conditions (i.e., 'in the wild'), the 'YouTube Faces' database, along with benchmark, pair-matching tests[1]*. (b) We employ our benchmark to survey and compare the performance of a large variety of existing video face recognition techniques. Finally, (c) we describe a novel set-to-set similarity measure, the Matched Background Similarity (MBGS). This similarity is shown to considerably improve performance on the benchmark tests.*

1. Introduction

Although face recognition is one of the most well studied problems in Computer Vision, recognizing faces in on-line videos is a field very much in its infancy. Videos naturally provide far more information than single images [9]. Indeed, several existing methods have obtained impressive recognition performances by exploiting the simple fact that a single face may appear in a video in many consecutive frames (e.g., [3, 10]). These methods, however, were primarily developed and tested using either strictly controlled footage or high quality videos from motion-pictures and TV shows. People appearing in these videos are often collaborative, are shot under controlled lighting and viewing conditions, and the videos themselves are stored in high quality.

Videos found in on-line repositories are very different in nature. Many of these videos are produced by amateurs, typically under poor lighting conditions, difficult poses, and are often corrupted by motion blur. In addition, bandwidth and storage limitations may result in compression artifacts, making video analysis even harder.

Figure 1. Example frames from the spectrum of videos available in the YouTube Faces data set. The bottom row depict some of the challenges of this set, including amateur photography, occlusions, problematic lighting, pose, and motion blur.

Recently, the introduction of comprehensive databases and benchmarks of face images, in particular images 'in the wild' (e.g., [6]), has had a great impact on the development of face recognition techniques. In light of this success, we present a large-scale, database, the 'YouTube Faces' database, and accompanying benchmark for recognizing faces in challenging, unconstrained videos (see, e.g., Fig. 1). Following [6] our benchmark is a simple yet effective pair-matching benchmark, allowing for standard testing of similarity and recognition methods. We use this benchmark to survey and test existing state-of-the-art techniques for face recognition.

We further present a novel set-to-set similarity measure, the Matched Background Similarity (MBGS), used here to evaluate the similarity of face videos. This similarity is designed to utilize information from multiple frames while remaining robust to pose, lighting conditions and other misleading cues. We consequently demonstrate a significant

[1]The database, image encoding, benchmark tests, and the code of the baseline methods are available at www.cs.tau.ac.il/~wolf/ytfaces.

boost in accuracy over existing methods.

2. Previous work

Early work in video face recognition includes [11] which use manifolds to represent the time varying appearances of faces and [13] who focus on real-time face recognition in videos. More recently [3, 14] focus on the task of aligning subtitle information with faces appearing in TV shows. In [10] faces appearing in a TV show were clustered according to subject identity across 11 years of broadcast. Searching through people in surveillance videos [12, 16] is related to recognition from web videos.

Set-to-set similarities. Frames of a video showing the same face, are often represented as sets of vectors, one vector per frame. Thus, recognition becomes a problem of determining the similarity between vector sets, which can be modeled as distributions [12], subspaces [24, 26], or more general manifolds [7, 11, 19]. Different choices of similarity measures are then used to compare two sets [19, 20, 24]. The MBGS approach described in this paper differs in that it models a set by a combination of a classifier and the set itself. At its core, the similarity is asymmetric and uses the classifier of one set to determine whether the vector set of another set is more similar to the first set or to a preselected subset background set. It is thus related to a recent family of similarities [15, 22, 23] based on a background set of examples and which employ classifiers.

The first recent background similarity method to emerge is the One-Shot-Similarity (OSS) [21, 22]. Given two vectors $\mathbf{x_1}$ and $\mathbf{x_2}$, their OSS score is computed by considering a training set of background sample vectors $\mathbf{B}$. This set of vectors contains examples of items different from both $\mathbf{x_1}$ and $\mathbf{x_2}$, and which are otherwise unlabeled. First, a discriminative model is learned with $\mathbf{x_1}$ as a single positive example and $\mathbf{B}$ as a set of negative examples. This model is then applied to the second vector, $\mathbf{x_2}$, obtaining a classification score. In [21] an LDA classifier was used, and the score is the signed distance of $\mathbf{x_2}$ from the decision boundary learned using $\mathbf{x_1}$ (positive example) and $\mathbf{B}$ (negative examples). A second such score is then obtained by repeating the same process with the roles of $\mathbf{x_1}$ and $\mathbf{x_2}$ switched: this time, a model learned with $\mathbf{x_2}$ as the positive example is used to classify $\mathbf{x_1}$, thus obtaining a second classification score. The symmetric OSS is the sum of these two scores.

3. The Matched Background Similarity

A set-to-set similarity designed for comparing the frames of two face-videos, must determine if the faces appearing in the two sets are of the same subject, while ignoring similarities due to pose, lighting, and viewing conditions. In order to highlight similarities of identity, we train a discriminative classifier for the members of each video sequence. Doing so allows us the freedom to choose the particular type of classifier used, but more importantly, provides us with the opportunity to train a classifier using a 'negative' set selected to best represent misleading sources of variation. This negative set is selected from within a large set of background videos put aside for this purpose. For example, in our benchmark, the video sequences in the training splits are used as the background set (see Sec. 4).

```
Similarity = MBGS(X1, X2, B)
     B1 = Find_Nearest_Neighbors(X1,B)
     Model1 = train(X1, B1)
     Confidences1 = classify(X2, Model1)
     Sim1 = stat(Confidences1)

     B2 = Find_Nearest_Neighbors(X2, B)
     Model2 = train(X2, B2)
     Confidences2 = classify(X1, Model2)
     Sim2 = stat(Confidences2)

     Similarity = (Sim1+Sim2)/2
```

Figure 2. Computing the symmetric Matched Background Similarity for two sets, X_1 and X_2, given a set B of background samples. The function **stat** represents either the mean, median, minimum or maximum over the confidences.

Assume a set $B = \{b_1, \ldots, b_n\}$ of background samples $b_i \in \mathbb{R}^d$, containing a large sample of the frames in a 'background-videos' set, encoded using a feature transform (e.g., LBP [8]). Given two videos, X_1 and X_2, likewise represented as two sets of feature vectors in $\mathbb{R}^d$, we compute their MBGS as follows (Fig. 2). We begin by locating for each member of X_1, its nearest-neighbor in B. We aggregate all these matched frames discarding repeating ones. If the size of the resulting set of nearest frames is below a predetermined size C, we move on to the 2nd nearest neighbor and so on until that size is met, trimming the set of matches in the last iteration such that exactly C frames are collected. This set, call it $B_1 \subset B$, $|B_1| = C$, is the set of background samples matching the vectors in X_1. Provided that this set does not contain images of the same individual appearing in X_1, B_1 captures similarities to members of X_1 resulting from factors other than identity. We now train a discriminative classifier (e.g., SVM) to distinguish between the two sets X_1 and B_1. A key observation is that the resulting discriminative model is trained to distinguish between *similar* feature vectors representing *different* identities.

Using the model, we now classify all members of X_2 as either belonging to X_1 or B_1. For each member of X_2 we thus obtain a measure of classification confidence. These confidence values can be, for example, the signed distances from the separating hyperplane when using an SVM classifier. Each such confidence value reflects the likelihood

that a member of X_2 represents the same person appearing in X_1. We take the mean (or alternatively, the median, the minimum, or the maximum) of all these values, obtained for all of the members of X_2 as the *one-sided MBGS*. This provides a global estimate for the likelihood that X_2 represents the same person appearing in X_1. The final, two-sided MBGS is obtained by repeating this process, this time reversing the roles of X_1 and X_2 and selecting a set $B_2 \subset B$, $|B_2| = C$ of background samples matching the members of X_2. The average of the two one sided similarities is the final MBGS score computed for the video pair.

4. The 'Youtube Faces' set and benchmark

In designing our video data set and benchmarks we follow the example of the 'Labeled Faces in the Wild' (LFW) image collection [6]. Specifically, our goal is to produce a large scale collection of videos along with labels indicating the identities of a person appearing in each video. In addition, we publish benchmark tests, intended to measure the performance of video pair-matching techniques on these videos. Finally, we provide descriptor encodings for the faces appearing in these videos, using well established descriptor methods. We next describe our database assembling process and associated benchmark tests.

Collection setup We begin by using the $5,749$ names of subjects included in the LFW data set [6] to search YouTube for videos of these same individuals. The top six results for each query were downloaded. We minimize the number of duplicate videos by considering two videos' names with edit distance less than 3 to be duplicates. Downloaded videos are then split to frames at 24fps. We detect faces in these videos using the VJ face detector [17]. Automatic screening was performed to eliminate detections of less than 48 consecutive frames, where detections were considered consecutive if the Euclidean distance between their detected centers was less than 10 pixels. This process ensures that the videos contain stable detections and are long enough to provide useful information for the various recognition algorithms. Finally, the remaining videos were manually verified to ensure that (a) the videos are correctly labeled by subject, (b) are not semi-static, still-image slide-shows, and (c) no identical videos are included in the database.

The screening process reduced the original set of videos from the $18,899$ originally downloaded ($3,345$ individuals) to $3,425$ videos of $1,595$ subjects. An average of 2.15 videos are available for each subject (See Table 1 for a distribution of videos per subject). The shortest clip duration is 48 frames, the longest clip is $6,070$ frames, and the average length of a video clip is 181.3 frames.

# videos	1	2	3	4	5	6
# people	591	471	307	167	51	8

Table 1. YouTube faces. Number of videos available per subject.

Database encodings All video frames are encoded using several well-established, face-image descriptors. Specifically, we consider the face detector output in each frame. The bounding box around the face is expanded by 2.2 of its original size and cropped from the frame. The result is then resized to standard dimensions of 200×200 pixels. We then crop the image again, leaving 100×100 pixels centered on the face. Following a conversion to grayscale, the images are aligned by fixing the coordinates of automatically detected facial feature points [2], and we apply the following descriptors: Local Binary Patterns (LBP) [8], Center-Symmetric LBP (CSLBP) [5] and Four-Patch LBP [21].

Benchmark tests Following the example of the LFW benchmark, we provide standard, ten-fold, cross validation, pair-matching ('same'/'not-same') tests. Specifically, we randomly collect $5,000$ video pairs from the database, half of which are pairs of videos of the same person, and half of different people. These pairs were divided into 10 *splits*. Each split containing 250 'same' and 250 'not-same' pairs. Pairs are divided ensuring that the splits remain subject-mutually exclusive; if videos of a subject appear in one split, no video of that subject is included in any other split. The goal of this benchmark is to determine, for each split, which are the same and which are the non-same pairs, by training on the pairs from the nine remaining splits. We note that this split design encourages classification techniques to learn what makes faces similar or different, rather than learn the appearance properties of particular individuals.

One may consider two test protocols. First, the *restricted* protocol limits the information available for training to the same/not-same labels in the training splits. This protocol is the one used in this paper. The *Unrestricted* protocol, on the other hand, allows training methods access to subject identity labels, which has been shown in the past to improve recognition results in the LFW benchmark [15].

5. Experiments

We test the performance of several baseline face recognition methods and compare them to the MBGS described in this paper. Several types of methods were considered. One group consists of methods employing comparisons between pairs of face images taken from the two videos. Another group uses algebraic methods such as distances between projections. Such methods often appear in the literature as methods of comparing vector sets, particularly sets of face images. A third group includes the Pyramid Match Kernel and the Locality-constrained Linear Coding methods, which were proven extremely effective in comparing

sets of image descriptors. We also include the performance obtained by using the straightforward heuristic of detecting an approximately frontal pose in each sequence and using this image to represent the entire set when comparing two videos. We next expand on these methods, and relate specific experiments to the rows of Table 2.

1. All pairs comparisons. Each video is represented by a set of vectors, each one produced by encoding the video frames using one of a number of existing face descriptors. Let X_1 be the matrix whose columns are the encoding of the frames of one video, and let X_2 be the corresponding matrix for the other video. We compute a distance matrix D where $D_{ij} = ||X_1(:,i) - X_2(:,j)||$, $X_1(:,i)$ denotes the i-th column of matrix X_1. Four basic similarity measures are then computed using D: the minimum of D, the average distance, the median distance, and the maximal distance. In addition we also compute the 'meanmin' similarity in which for each image (of either set) we match the most similar image from the other set and consider the average of the distances between the matched pairs.

2. Pose based methods. Presumably, the easiest image to recognize in each image set is the one showing the face in a frontal pose. We therefore locate the frontal-most pose in each sequence by using the web API of face.com to obtain the three rotation angles of the head. Comparing two sequences then involves measuring the similarity between the descriptors of the representative frames of the two videos.

Another baseline method uses one face image from each sequence by considering pairs of images with the smallest head rotation angle between them. Rotation is estimated from each image as before and all frames from one sequence are compared to all frames from the other sequence.

3. Algebraic methods. Algebraic methods view each matrix X_1 or X_2 as a linear subspace that is spanned by the columns of the matrix. A recent work [20] provides an accessible summary of large number of such methods. Many of the methods are based on the analysis of the principle angles between the two subspaces.

Let $U_k, k = 1,2$ be any orthogonal basis for the subspace spanned by the columns of X_k. The SVD of $U_1^\top U_2 = WSV^\top$ provides the principle angles between the column subspaces of the two matrices X_1 and X_2. Specifically, the inverse cosine of the diagonal of S are the principle angles, i.e., $S = diag(\cos\theta)$, where Θ is the vector of principle angles of X_1 and X_2. Note that this vector is sorted from the least angle to the largest.

Several distances are defined based on these notations: The max correlation is defined by the minimal angle $\theta(1)$; The projection metricis given by $||U_1U_1^\top - U_2U_2^\top||_F = ||\sin\Theta||$; The norm $||U_1^\top U_2||_F$ seems to be relatively effective; Finally, the Procrustes metric [1] is computed from the vector-norm $||\sin(\Theta/2)||$ (here and above, the sin of a vector is taken element by element).

Care should be taken when the number of frames differs between the sequences or if the number of samples is larger than the dimensionality of the representation. It is a good practice to restrict U_1 (U_2) to be the first r singular vectors of the subspace spanned by the columns of X_k. This is justified by the fact that the projections $U_kU_k^\top$, $k = 1,2$ provide the closest possible projection by a rank r projection to the vectors of X_k. In our experiments we found the value of $r = 10$ to provide relatively good performance.

The last algebraic method we compare to is the CMSM method [27]. This method utilizes a training set and is essentially a max correlation method after the vectors have been projected to the subspace spanned by the smallest eigenvectors of the matrix that is the sum of all projection matrices of the training set. The projection is followed by a normalization step and an orthogonalization step. Lastly, the max correlation, sometimes called MSM, is computed as the score. Alternatively, as is done in the code made available by the authors [27], the average of the largest $t = 10$ canonical correlations can be used.

4. Non-algebraic Set methods. We next consider methods that have emerged as effective ways to represents sets, not necessarily in the context of computer vision.

The Pyramid Match Kernel (PMK) [4] is an effective kernel for encoding similarities between sets of vectors. PMK represents each set of vectors as a hierarchical structure ('pyramid') that captures the histogram of the vectors at various levels of coarseness. The cells of the histograms are constructed by employing hierarchical clustering to the data, and the similarity between histograms is captured by histogram intersection. In our experiments, we construct the bins by clustering the nine training splits of each test. This is then applied to the tenth split in order to measure the similarities between the videos in that set.

We also test sparsity based methods, and specifically methods based on locality constrained encoding. In such methods, a face image is represented as a linear combination of a small subset of faces taken from a large dictionary of face images. Sparse representation methods were shown to enable accurate recognition despite of large variations in illumination and even occlusions [25].

As far as we know, such methods were not previously used for multi-frame identification. However, similar methods have been used for visual recognition based on sets of visual descriptors. The emerging technique, which we adopt in our experiments, is to represent the set by the maximal coefficients (one coefficient per dictionary vector) over all set elements. In order to maintain a reasonable runtime, in our experiments we employ the LLC method [18], in which the sparse coefficients are computed based on the k-nearest dictionary vectors to each set element. The dictionary itself was constructed, as is often the case, by k-means clustering of the training frames.

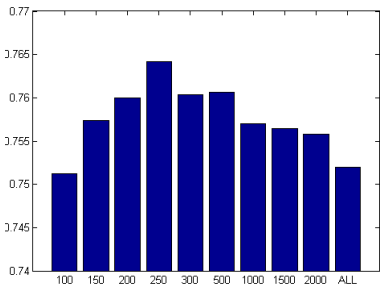

Figure 3. Success rate for MBGS using the LBP descriptor and the mean statistic as a function of the matched set size parameter C.

5. Matched Background Similarity. In MBGS samples from a large set B of background samples are matched to each set sample. In our experiments, random $1,000$ frames from each training split were used for this purpose. The matching itself is performed by the smallest L2 metric of the descriptor. Four methods are compared for combining the individual classification scores (there is one per frame in the second set) into one similarity value: mean, median, max and min. To combine the two asymmetric scores, the average of the two scores is always used. A value of $C = 250$ seems to provide good results. A graph showing the sensitivity of MBGS to this parameter is shown in Figure 3. As can be seen, plainly using the entire background set (without matching and selecting) is suboptimal.

Results are presented in Table 2. ROC curves of selected methods are presented in Figure 4. As detailed in Sec. 4, these results were obtained by repeating the classification process 10 times. Each time, we use nine sets for training, and evaluate the results on the tenth set. Results are reported by constructing an ROC curve for all splits together (the outcome value for each pair is computed when this pair is a testing pair), by computing statistics of the ROC curve (area under curve and equal error rate) and by recording average recognition rates $\pm$ standard errors for the 10 splits.

The results indicate that the newly proposed MBGS method outperforms the existing methods when considering all measures: recognition rate ('accuracy'), area under curve, and equal error rate. The simplest min distance method is a solid baseline. The pose based heuristics, while popular, are not entirely effective. The algebraic methods are not better than the min-distance method. PMK and our adaptation of LLC underperform, although more exploration of their parameter space might improve their results.

To gain further insight into the challenges of the benchmark and the limitations of the current methods, Figure 5 presents the most confident cases according the variant of the MBGS method based on L2-norm matching and the mean operator. The most confident same-person predictions are of video sequences that are captured at the same scene, and the 'easiest' not-same are where there are multiple discriminating factors. The method can sometimes be fooled by motion blur, hats, and variation in illumination.

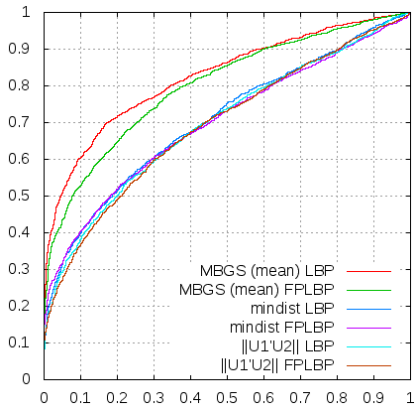

Figure 4. ROC curves averaged over 10 folds. The plots compare the results obtained for LBP and FPLBP using the baseline mindist and $||U1^{\top}U2||_F$ methods and the MBGS method, where the scores are combined by the mean operator.

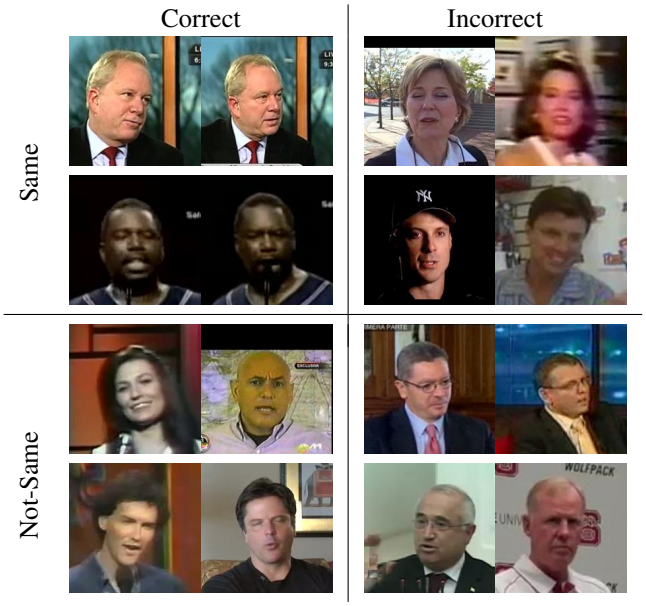

Figure 5. Most confident MBGS results (L2 norm, mean operator). The Same/Not-Same labels are the ground truth labels, and the Correct/Incorrect labels indicate whether the method predicted correctly. For example, the top right quadrant displays same-person pairs that were most confidently labeled as not-same.

References

[1] Y. Chikuse. *Statistics on special manifolds, lecture notes in statistics, vol. 174*. New York: Springer, 2003. 532

[2] M. Everingham, J. Sivic, and A. Zisserman. "hello! my name is... buffy" - automatic naming of characters in tv video. In *BMVC*, 2006. 531

Type	Method	CSLBP			FPLBP			LBP		
		Accuracy ± SE	AUC	EER	Accuracy ± SE	AUC	EER	Accuracy ± SE	AUC	EER
1	min dist	63.1 ± 1.1	67.3	37.4	65.6 ± 1.8	70.0	35.6	65.7 ± 1.7	70.7	35.2
	max dist	56.5 ± 2.3	58.8	43.8	55.7 ± 2.5	58.1	45.3	57.9 ± 1.8	61.1	42.6
	mean dist	61.1 ± 2.1	64.9	39.5	62.9 ± 1.4	67.0	38.2	63.7 ± 2.3	68.3	36.8
	median dist	60.8 ± 2.1	64.8	39.4	62.7 ± 1.5	66.8	38.4	63.5 ± 2.0	68.2	36.8
	"meanmin"	62.6 ± 1.5	66.5	38.3	65.5 ± 1.8	69.2	36.6	65.1 ± 1.8	70.0	35.8
2	most frontal	60.5 ± 2.0	63.6	40.4	61.5 ± 2.8	64.2	40.0	62.5 ± 2.6	66.5	38.7
	nearest pose	59.9 ± 1.8	63.2	40.3	60.8 ± 1.9	64.4	40.2	63.0 ± 1.9	66.9	37.9
3	Max corr	58.4 ± 2.1	64.3	39.8	61.2 ± 2.4	65.4	39.4	62.5 ± 1.5	67.4	37.8
	CMSM	61.2 ± 2.6	65.2	39.8	63.8 ± 2.0	68.4	37.1	63.1 ± 1.8	67.3	38.4
	projection	50.1 ± 0.2	45.7	53.1	50.0 ± 0.2	46.0	52.8	50.1 ± 0.3	45.0	52.4
	$\|\|U_1^\top U_2\|\|_F$	63.8 ± 1.8	67.7	37.4	64.3 ± 1.6	69.4	35.8	65.4 ± 2.0	69.8	36.0
	procrustes	62.8 ± 1.6	67.1	37.5	64.5 ± 1.9	68.3	36.9	64.3 ± 1.9	68.8	36.7
4	PMK	52.7 ± 2.2	53.1	47.0	52.7 ± 1.7	52.4	48.3	51.5 ± 1.8	51.8	48.0
	LLC	51.5 ± 2.1	53.4	48.1	51.1 ± 1.6	53.1	48.2	50.6 ± 1.7	53.4	47.8
5	mean	72.4 ± 2.0	78.9	28.7	72.6 ± 2.0	80.1	27.7	76.4 ± 1.8	82.6	25.3
	median	72.4 ± 1.9	78.9	28.5	72.3 ± 2.2	79.9	27.7	76.3 ± 1.5	82.6	25.4
	max	70.5 ± 1.5	77.1	29.9	71.3 ± 2.3	78.0	29.6	73.9 ± 2.0	80.9	26.4
	min	67.1 ± 2.6	73.9	33.1	68.0 ± 2.5	74.5	32.4	72.4 ± 2.8	79.3	28.5

Table 2. Benchmark results obtained for various similarity measures and image descriptors. See text for the description of each method.

[3] M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automated naming of characters in TV video. *Image and Vision Computing*, 27(5):545–559, 2009. 529, 530

[4] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005. 532

[5] M. Heikkilä, M. Pietikäinen, and C. Schmid. Description of interest regions with center-symmetric local binary patterns. In *Computer Vision, Graphics and Image Processing, 5th Indian Conference*, pages 58–69, 2006. 531

[6] 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, TR 07-49, 2007. 529, 531

[7] T.-K. Kim, O. Arandjelovic, and R. Cipolla. Boosted manifold principal angles for image set-based recognition. *Pattern Recognition*, 40(9):2475–2484, 2007. 530

[8] T. Ojala, M. Pietikainen, and D. Harwood. A comparative-study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1), 1996. 530, 531

[9] A. J. O'Toole, P. J. Phillips, S. Weimer, D. A. Roark, J. Ayyad, R. Barwick, and J. Dunlop. Recognizing people from dynamic and static faces and bodies: Dissecting identity with a fusion approach. *Vision Research*, 2011. 529

[10] D. Ramanan, S. Baker, and S. Kakade. Leveraging archival video for building face datasets. In *ICCV*, 2007. 529, 530

[11] B. Raytchev and H. Murase. Unsupervised face recognition from image sequences based on clustering with attraction and repulsion. In *CVPR*, 2001. 530

[12] G. Shakhnarovich, J. Fisher, and T. Darrell. Face recognition from long-term observations. In *ECCV*, 2002. 530

[13] G. Shakhnarovich, P. Viola, and B. Moghaddam. A unified learning framework for real time face detection and classification. In *Auto. Face & Gesture Recognition*, 2002. 530

[14] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?": Learning person specific classifiers from video. In *CVPR*, 2009. 530

[15] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *BMVC*, 2009. 530, 531

[16] D. Vaquero, R. Feris, D. Tran, L. Brown, A. Hampapur, and M. Turk. Attribute-based people search in surveillance environments. In *WACV*, 2009. 530

[17] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2), 2004. 531

[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010. 532

[19] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *CVPR*, 2008. 530

[20] T. Wang and P. Shi. Kernel grassmannian distances and discriminant analysis for face recognition from image sets. *Pattern Recogn. Lett.*, 30(13):1161–1165, 2009. 530, 532

[21] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *post-ECCV Faces in Real-Life Images Workshop*, 2008. 530, 531

[22] L. Wolf, T. Hassner, and Y. Taigman. The one-shot similarity kernel. In *ICCV*, 2009. 530

[23] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *ACCV*, 2009. 530

[24] L. Wolf and A. Shashua. Learning over sets using kernel principal angles. *J. Mach. Learn. Res.*, 4, 2003. 530

[25] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, 2009. 532

[26] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. In *Automatic Face and Gesture Recognition*, 1998. 530

[27] O. Yamaguchi, K. Fukui, and K.-i. Maeda. In *Automatic Face and Gesture Recognition*, 1998. 532