The SVM-minus Similarity Score for Video Face Recognition

Lior Wolf Noga Levy
The Blavatnik School of Computer Science, Tel-Aviv University, Israel

Abstract

Face recognition in unconstrained videos requires specialized tools beyond those developed for still images: the fact that the confounding factors change state during the video sequence presents a unique challenge, but also an opportunity to eliminate spurious similarities. Luckily, a major source of confusion in visual similarity of faces is the 3D head orientation, for which image analysis tools provide an accurate estimation.

The method we propose belongs to a family of classifier-based similarity scores. We present an effective way to discount pose induced similarities within such a framework, which is based on a newly introduced classifier called SVM-minus. The presented method is shown to outperform existing techniques on the most challenging and realistic publicly available video face recognition benchmark, both by itself, and in concert with other methods.

1. Introduction

Face recognition applications for border control and photo-album tagging, which are based on recent image-based methods, have proved to be extremely useful. However, looking into future applications of face recognition, the role of video-based methods might become more and more dominant. The required technologies for video and images are obviously related, but video presents additional challenges that require a dedicated consideration.

In both images and video, the most significant challenge for real-world face recognition systems might be that of head pose. When the subjects are not required to collaborate with the system, the 3D orientation of the head can cause changes in appearance within the captured faces of the same person that are larger than changes among faces of different people. Even with advanced face alignment techniques, the practical implications of pose variations seem to suppress those of other factors such as expression, illumination, and image quality.

In this paper, we present a similarity score which specifically asks given two videos: how much is the face in one video sequence similar to that of the other, where this similarity is uncorrelated with the pose-induced similarity. The novel similarity score belongs to a family of classifier based similarities that were shown previously to be much more effective for face recognition in unconstrained video than all other methods in the literature, and pushes the performance envelope even further.

Within the novel similarity score we employ a new learning method called SVM$\ominus$ (reads SVM-minus), which learns to discriminate between positive and negative examples in a way that is uncorrelated with the discriminative function learned on an additional feature set. In our case, the appearance descriptors are the main features, and the additional information is based on estimated 3D head pose.

2. Previous work

Video face recognition is used for various tasks such as real-time face recognition [27], searching people in surveillance videos [26, 32], aligning subtitle information with faces [9, 29] and clustering by subject identity [24].

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 [26], subspaces [40], or more general manifolds [16, 25, 34]. Different choices of similarity measures are then used to compare sets [34, 35].

Algebraic methods that compare sets regard each video as a linear subspace, spanned by the vectors encoding the frames in the video. An accessible summary of a large number of such methods is provided in [35]. Many of the methods are based on the analysis of the principle angles between the two subspaces. Several distances can be defined based on these angles, including the CMSM method that uses the max correlation [40], the projection metric [7], and the Procrustes metric [6].

The Pyramid Match Kernel (PMK) [13] is a non-algebraic kernel for encoding similarities between sets of vectors, which was shown to be extremely effective in several object recognition tasks. The 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 em-

ploying hierarchical clustering to the data, and the similarity between histograms is captured by histogram intersection.

Following the success of comprehensive face image benchmarks taken under natural conditions, out of which 'Labeled Faces in the Wild' [15] might be the most prominent, the 'YouTube Faces DB' database of labeled videos of faces was presented and made available [1] . The recognition ability of a wide variety of video face recognition approaches was tested on this video dataset in [36], and compared to the Matched Background Similarity (MBGS) method suggested in that paper. The MBGS approach, which is described in detail in Sec. 3, differs from the methods mentioned above in that it employs a classifier that is trained to distinguish between the set being modeled and confusing samples from a preselected background set.

Learning with Side Information Incorporation of additional information within machine learning can be used is a supervised, semi-supervised or unsupervised manner. In the semi-supervised frameworks of domain adaptation [2] and co-training [3] knowledge from a labeled source domain is fused to a target domain containing little or no labeled data.

Side information is used to learn the relevant structures in the data by reducing irrelevant variability while amplifying relevant variability [28]. Both relevant and irrelevant additional information can be provided as in [12, 4], where relevant structures in the data are learned by maximizing the mutual information with relevant data and minimizing mutual information with irrelevant data.

Additional information about the features in the form of meta-features can be integrated into SVM [18] efficiently, by deriving a linear transformation on the input and learning a standard SVM on the transformed input.

Latent information such as part locations in object detection and gesture recognition tasks can be learned based on local features, by maximizing [10] or marginalizing [23] all possible values. The side information is given through the structure of the hidden domain.

The learning using privileged information (LUPI) paradigm suggested in [31] utilizes privileged information supplied by the teacher during the training phase. The LUPI scheme can be applied in various machine learning contexts such as clustering [11] and boosting [5]. The SVM+ algorithm [22] is a LUPI classification method that is based on SVM, where the 'plus' sign refers to the additional discriminative power gained from the privileged information.

The algorithm we suggest in this work, SVM$\ominus$, is also intended to benefit from additional information that is exclusively available during training. However, in contrast to the SVM+ case, the data we regard does not give a better classification by itself. Instead, it describes a misleading factor, such as pose or lighting conditions in face images, which needs to be eliminated when considering the faces' identities. Hence, the 'minus' stands for the elimination of a factor that is irrelevant to the task at hand.

Building classifiers that minimize correlations with other classifiers have been studied before in the context of ensemble methods [20, 19] and dimensionality reduction [17] with no privileged or side information supplied. These methods measure correlation between consecutive models learned on the same data. The optimization problem proposed in [19] is the most similar to the one suggested in this work. However, the application is done in a completely different context; the details differ considerably, and a different optimization method is used.

3. The One-Shot Family of Similarities

The similarity methods described in this section build upon the common idea of finding the association between two objects using a background set of samples. The basic method is the One-Shot-Similarity (OSS) [37, 38] described in Fig. 1. 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 unlabeled examples of items different from both $\mathbf{x_1}$ and $\mathbf{x_2}$.

First, a discriminative model is learned with $\mathbf{x_1}$ as a single positive example and $\mathbf{B}$ as a set of background examples. This model is then applied to the second vector, $\mathbf{x_2}$, obtaining a classification score. In [37] 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 mean of these two scores.

The OSS score does not employ label 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 the label information is available, the OSS score does not benefit from it. The Multiple One-Shots method [30] employs label information by computing the One-Shot Score multiple times. Using this information, multiple background sets are considered, each such set reflecting either a different identity or a different pose. As described in Fig. 2, the OSS is then computed multiple times, where each time only one background subset is used. Finally, the multiple OSS scores are fed to a linear Support Vector Machine classifier, and the output is the final classification result.

The intuition guiding MSS is that a whole background set contains variability due to a multitude of factors including pose, identity and expression while the positive sample is an image of one person captured at one pose under a particular viewing condition. The trained classifier can distinguish based on any factor, not necessarily based on the

identity of the person. When the background set contains a single person or a single pose, the classifier is more likely to distinguish based on the approximately constant factor.

The Matched Background Similarity [36] (Fig. 3) is a set-to-set similarity designed for comparing the frames of two face-videos to determine if the faces appearing in the two sets are of the same person. In order to highlight similarities of identity, a discriminative classifier is trained for the frames of each video sequence vs. a subset of background frames that are selected to best represent misleading sources of variation such as pose, lighting, and viewing conditions. This subset is selected from within a large set of background videos put aside for this purpose.

Assume a set $B = \{b_1, \ldots, b_n\}$ of background samples $b_i \in \mathsf{R}^d$, containing a large sample of the frames in the 'background-videos' set. Given two videos, X_1 and X_2, likewise represented as two sets of feature vectors in R^d, their MBGS is computed as the mean of two one-side MBGS scores obtained via the *OneSideMBGS* method.

The *OneSideMBGS* method first constructs a subset of the background set B_1 matching the vectors in X_1. The nearest-neighbor of each member of X_1 is located in B, and all neighbors are aggregated discarding repeating ones. If the size of the resulting set of nearest frames is below a predetermined size C, the 2nd nearest neighbor is considered and so on until that size is met, trimming the set of matches in the last iteration to collect exactly C frames.

An SVM classifier is trained to distinguish between the two sets X_1 and B_1. Using the learned model, all members of X_2 are classified as either belonging to X_1 or B_1, and the confidence values for all of the members of X_2 are returned to the *MBGS* main function. Typically, a Linear SVM classifier is used, and the confidence values are signed distances from the separating hyperplane. These confidence values are averaged and produce a single score, which is related to 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, which requires the selection of B_2, a subset of the background set matching the vectors in X_2. The average of the two one sided similarities is the final MBGS score computed for the video pair.

Similarly to the OSS, the MBGS score does not employ label information. The Multiple OSS method cannot be directly used in video to eliminate the pose effect, since each video contains a multitude of poses and expressions. Using an idea similar to Multiple OSS applied to known identities is possible; However, it requires a labeled training set.

In Sec. 6 we suggest the SVM$\ominus$ similarity that uses additional information available during the similarity computation. In our case, this method discounts information that is correlated with pose information in order to eliminate this irrelevant factor that can be misleadingly discriminative.

```
Similarity = OSS(x1, x2, B)

Model1 = train(x1, B)
Sim1 = classify(x2, Model1)

Model2 = train(x2, B)
Sim2 = classify(x1, Model2)

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

Figure 1. One-Shot similarity computation for two vectors, x_1 and x_2, given a set B of background samples.

```
Similarity = MSS(x1, x2, {B1, B2,..., Bk})

for i = 1 ... k
   Sim(i) = OSS(x1, x2, Bi)
end
Similarity = classify(Sim, SVMmodel)
```

Figure 2. Multi-Shot Similarity score for two vectors, x_1 and x_2, using k background sets $B_1, \ldots, B_k$. SVMmodel is a stacking model learned on the training set.

```
Sim = OneSideMBGS(X1, X2, B)

B1 = Find_Nearest_Neighbors(X1,B)
Model1 = train(X1, B1)
Confidences = classify(X2, Model1)
Sim = mean(confidences)
```

```
Similarity = MBGS(X1, X2, B)

Sim1 = OneSideMBGS(X1, X2, B)
Sim2 = OneSideMBGS(X2, X1, B)
Similarity = (Sim1+Sim2)/2
```

Figure 3. Computing the symmetric Matched Background Similarity for two sets, X_1 and X_2, given a set B of background samples. The one-side similarity is taken as the mean of the calculated confidences, since this operator was shown in [36] to outperform the other operators tested: median, minimum, and maximum.

4. The SVM-minus Classifier

The SVM$\ominus$ similarity (reads SVM-minus similarity) is based on the SVM$\ominus$ (SVM-minus) classifier. This classification method takes as input a training set $\{x_i\}$, $i = 1..m$, a matching set of privileged information $\{x'_i\}$ and the corresponding binary labels $\{y_i\}$. Let X (X') be the matrices whose columns are the vectors $\{x_i\}$ ($\{x'_i\}$).

First, an auxiliary SVM classifier is trained on the privileged data X' using the labels y. Let c denote the confidences of X' predicted by the learned classifier. The term confidence refers here specifically to the signed distance of an example from the separating hyperplane. The optimization problem at the core of the SVM$\ominus$ classifier takes as input the training set X, the labels y and the con-

fidences c, and solves an SVM-like optimization problem with the additional constraint that the confidences of the second learned model are uncorrelated with c.

The additional constraint of low correlation is applied to the vectors labeled as positive ($y_i = +1$) and to the vectors labeled as negative ($y_i = -1$) separately. This partition to positive and negative classes is necessary since all accurate classifiers are expected to be correlated as they provide comparable labeling. However, classifiers which rely on independent information sources can differ considerably with regards to the confidences they assign to the examples within each class. To construct the SVM$\ominus$ optimization problem, X is split into matrices X_p and X_n containing the vectors labeled as positive and the vectors labeled as negative respectively. The rows of X_p (X_n) are normalized to mean 0, where each row contains the values of a single feature across all positive (negative) vectors. Similarly, the confidences vector c is split into two vectors, c_p and c_n. Let σ denote the standard deviation operator, c_p and c_n are separately normalized to mean 0 and $\sigma(c_p) = \sigma(c_n) = 1$.

Denote by w the sought after solution of the SVM$\ominus$ optimization problem, then the Pearson's sample correlation between c_p and the confidence values of the positive vectors $w^T X_p$ is $\frac{w^T X_p c_p}{\sigma(w^T X_p)}$. Omitting the denominator $\sigma(w^T X_p)$ to maintain convexity, $(w^T X_p c_p)^2$ is added to the objective function. The square is required in order to minimize the magnitude of the correlation regardless of its sign. Similarly, the correlation constraint between c_n and the confidence values of the negative vectors added to the objective function is $(w^T X_n c_n)^2$. The trade-off among $\|w\|^2$ and the added correlation expressions is controlled by trade-off parameters λ_p and λ_n, and the optimization problem becomes

$$\begin{aligned} \min_w \frac{1}{2}\|w\|^2 &+ \frac{\lambda_p}{2}\left(w^T (X_p c_p)(X_p c_p)^T w\right) \\ &+ \frac{\lambda_n}{2}\left(w^T (X_n c_n)(X_n c_n)^T w\right) + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & \forall i.\ y_i \langle w, x_i\rangle \geq 1 - \xi_i \ , \ \xi_i \geq 0. \end{aligned} \tag{1}$$

5. Efficient Computation

The standard soft-margin SVM optimization problem is formulated as

$$\begin{aligned} \min_w \frac{1}{2}\|w\|^2 &+ C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & \forall i.\ y_i \langle w, x_i\rangle \geq 1 - \xi_i \ , \ \xi_i \geq 0. \end{aligned} \tag{2}$$

Finding an efficient reduction from SVM$\ominus$ to standard SVM enables the use of off-the-shelf efficient SVM solvers for SVM$\ominus$. Such a reduction to SVM indeed exists, using a linear projection of the training set as shown in Lemma 5.1.

Lemma 5.1 *Given a set X, labels y and confidences c, a projection matrix L can be constructed such that solving the SVM$\ominus$ optimization problem of Eq. 1 over the training set X reduces to solving the SVM optimization problem of Eq. 2 over the training set LX.*

Proof Let A be the quadratic coefficients matrix,

$$A = I + \lambda_p (X_p c_p)(X_p c_p)^T + \lambda_n (X_n c_n)(X_n c_n)^T \ ,$$

where X_p and X_n are as above. Note that since by definition $\lambda_p \geq 0$ and $\lambda_n \geq 0$, the matrix A is positive-definite.

The objective function in Eq. 1 can be rewritten as $\frac{1}{2} w^T A w + C\sum_{i=1}^{m}\xi_i$. Denote by α the vector of dual variables of the margin constraints, and by α_y the vector α signed by the labels y element-wise. The primal variable w can be expressed in the dual space as $w = A^{-1} X \alpha_y$. Substituting w with $A^{-1} X \alpha_y$, Eq. 1 can be rephrased as

$$\begin{aligned} \min_\alpha \frac{1}{2}\alpha_y^T X^T A^{-1} X \alpha_y &+ C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & \forall i.\ \alpha_y^T X^T A^{-1} x_i \geq 1 - \xi_i \ , \ \xi_i \geq 0. \end{aligned} \tag{3}$$

Since A is positive-definite, its inverse matrix A^{-1} is also positive definite, $A^{-1} = LL^T$, and the square root matrix L can be computed using the Cholesky decomposition. Replacing A^{-1} by LL^T in Eq. 3, we get

$$\begin{aligned} \min_\alpha \frac{1}{2}\alpha_y^T (LX)^T (LX) \alpha_y &+ C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & \forall i.\ \alpha_y^T (LX)^T (L x_i) \geq 1 - \xi_i \ , \ \xi_i \geq 0. \end{aligned} \tag{4}$$

the SVM$\ominus$ optimization problem becomes the standard SVM problem (Eq. 2) over the training set LX, as stated.

6. The SVM-minus Similarity

The SVM$\ominus$ similarity between sets X_i and X_j is computed using the corresponding privileged information of the sets, X_i' and X_j', and a background set B with privileged information B'.

First, a background subset B_i is chosen from the background set B as described in Sec. 3, and a matching B_i' is taken from the privileged background set B'.

The SVM$\ominus$ classifier is trained on $[X_i, B_i]$ and the matching privileged information $[X_i', B_i']$, referring to X_i, X_i' as the positive sets, and to B_i, B_i' as the negative sets.

The learned SVM$\ominus$ classifier then classifies X_j, and the output confidences are combined by their mean, similarly to MBGS, to form a one-side SVM$\ominus$ similarity score.

The sets X_i and X_j then exchange roles and an SVM$\ominus$ classifier is trained on set X_j. The learned model classifies X_i, and the confidences are combined by their mean to a second one-side SVM$\ominus$ similarity score. The final SVM$\ominus$ similarity is the average of the two one-side similarities.

```
S = SVM-minus_Similarity(X1, X1, X2, X2, B, B ,C)

Model1 = One_Side_SVM-minus(X1, X1, B, B ,C)
Confidences1 = classify(X2, Model1)
Sim1 = mean(Confidences1)

Model2 = One_Side_SVM-minus(X2, X2, B, B ,C)
Confidences2 = classify(X1, Model2)
Sim2 = mean(Confidences2)

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

```
Model = One_Side_SVM-minus(X, X ,B, B ,C)

BX = Find_Nearest_Neighbors(X, B, C)
BX = privileged vectors matching BX.
m = number of columns of X (= that of X)
y = [1m followed by −1C]

Model = SVM-minus([X, BX], [X , BX], y)
```

```
Model = SVM-minus(X, X , y)

Model' = train(X , y)
Confidences' = classify(X ,Model')
Model =
  SVM-minus_optimization(X, y, Confidences')
```

Figure 4. Computing the SVM⊖ Similarity between two sets given X_1, X_2, a background B, privileged information X_1, X_2, B and the size of the background subsets C. The function *Find_Nearest_Neighbors* is defined in Sec. 3; The function *SVM-minus_optimization* optimizes Eq. 1 and is described in detail in Sec. 5. 1_d is a vector of 1s in R^d.

Note that in applications where recognition is to be performed on-line, one can rely on the one sided SVM⊖ similarity to compare all gallery image sets to the prob set, as the prob set manifests itself frame by frame. In this case the underlying SVM⊖ classifiers for the gallery sets can be constructed beforehand (they are independent of the probe set), and the confidences can be efficiently computed to each probe-frame as it is captured.

7. Experiments

Our experiments are conducted on the recent video dataset called 'YouTube Faces DB' [36], which was designed following the 'Labeled Faces in the Wild' (LFW) image collection [15]. The dataset contains a large collection of videos along with labels indicating the identity of a person appearing in each video. It also contains scripts and meta-data defining benchmark protocols for the task of video pair-matching, where given a pair of videos each tested method answers a binary same/not-same query.

The authors of [36] provide per-frame encoding of all video data using several well-established face-image descriptors. Encoding is done by considering the detected faces, expanding the bounding box around each detection to include more of the image, performing cropping, and resizing to an image of size 100×100 pixels. The images are then aligned by fixing the coordinates of a few detected facial feature points [8], and three descriptors are extracted: Local Binary Patterns (LBP) [21], Center-Symmetric LBP (CSLBP) [14] and Four-Patch LBP (FPLBP) [37]. In addition, every frame is provided with 3D head orientation data, which was estimated using the formerly-public API of face.com. These 3D vectors are taken as the privileged information in the SVM⊖ experiments.

Following the example of the LFW benchmark, 'YouTube Faces DB' follows a ten-fold, cross validation, pair-matching ('same'/'not-same') test. Specifically, $5,000$ video pairs from the database, half of which are pairs of videos of the same person, and half of different people were selected at random and divided into 10 splits. Each split contains 250 'same' and 250 'not-same' pairs. The splits were sampled to be subject mutually-exclusive; if videos of a subject appear in one split, no video of that subject is included in any other split. The task is to determine, for each split, which are the same and which are the not-same pairs, by training on the pairs from the nine remaining splits. We follow the restricted protocol that limits the information available for training to the same/not-same labels in the training splits. The subject identity labels are not used.

In [36], the performance of an extensive set of baseline video face recognition methods was evaluated and compared to the performance of the MBGS method. These include methods that are based on comparisons between pairs of face images selected from the two videos; Algebraic methods that currently dominate the video face recognition literature; Methods that are effective in comparing sets of local visual descriptors such as the Pyramid Match Kernel [13] and the Locality-constrained Linear Coding method (LLC) [33]. The MBGS method outperformed all of these other methods by a very significant gap.

To define the background set, in each of the ten cross validation rounds, the frames of the videos of one out of the nine training splits are used. There are four variants of MBGS presented in [36], each is based on a particular statistical operator to summarize the per-frame classification measurements (last statement of the method OneSideMBGS, Fig. 3): mean, median, min, and max. The mean operator provides the best results in [36] and is therefore used here too. The other parameters of MBGS are the size of the background set (C) and the regularization parameter of the underlying SVM classifier. These were set in [36] to 250 and 1 respectively, and we use these values without modification for both MBGS and the SVM⊖ similarity score. The latter has two additional parameter – the regu-

larization parameters of the SVM⊖ classifier λ_p, λ_n. These parameters, too, are set to 1. Note that following [36], all SVM classifiers employed in this work are linear.

Results are presented in Table 1. As mentioned, these results were obtained by repeating the classification process 10 times. Each time, nine sets are used for training, and the tenth is used for evaluation. 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.

In addition to MBGS and the proposed SVM⊖ similarity score, we present results for a selected subset of the methods for which results exist on the "YouTube Faces DB" dataset. These are selected due to their relative effectiveness compared to other methods of the same family, or due to their popularity. Shown are the simple heuristics: the minimal pairwise distance between the two sets of frames, the distance between the most frontal frames in each set, and the distance between the two frames that are most similar in pose; The algebraic methods: CMSM [40], the norm of the multiplication of the projection matrices of the two linear subspaces ($||U_1^\top U_2||_F$) [7], and the Procrustes distance [6].

The results support the effectiveness of the presented SVM⊖ similarity score. It outperforms all other methods, including MBGS, when considering the area under the ROC (AUC) and the equal error rate (EER). We note that with regards to recognition rate ('accuracy') SVM⊖ does not outperform MBGS. This score is computed by applying a Linear SVM classifier to the similarity scores treated as 1D feature vectors. Therefore, the SVM classifier simply selects a threshold for each similarity, and provides sub-optimal thresholds for the SVM⊖ similarity. Examining the similarity scores, the reason for this seems to be the existence of a few negative pairs which are given relatively high scores.

We also present results for combined scores, which include both MBGS and the SVM⊖ similarity. The combination is done through a technique called stacking [39]. In our experiments, a Linear SVM classifier is applied to the 2D vector which contains both scores to produce a combined one. In each of the 10 cross-validation rounds, this classifier is trained on the 8 training splits (leaving the split used for background frames aside), and applied to the 10th. As can be seen in Table 1, combining the two scores produces more accurate results than each method separately. The combined score is superior to MBGS for the FPLBP and LBP features in a statistically significant way (t-test p-value < 0.05).

The SVM⊖ classifier is used within the SVM⊖ similarity to produce similarity scores that differ from those of MBGS. To examine this effect we have computed the correlations between the similarity scores produced by each method on the $5,000$ benchmark pairs. The results are shown in Table 2. As can be seen, each similarity score is more similar to other similarities of the same type (MBGS or SVM⊖ similarities) than to those of the other type. As expected, among the similarities of the other type, the correlation to the similarity that is derived from the same face descriptors is the highest.

As a sanity check, we also tested the use of the entire background set (without matching and selecting). This seems to considerably diminish the resulting accuracy. For example, in the case of the LBP descriptors, the AUC of the SVM⊖ similarity drops from 83.6% to 79.9%. Weighing the positive class to increase its contribution to the loss function did not improve the obtained results.

As mentioned in Sec. 5, for on-line applications of the similarity score, one might be interested in a one-sided version: when the one-sided version is used, there is no need to retrain the underlying classifiers given the new video, and the score can be computed incrementally one frame at a time. We have therefore conducted similar experiments by employing the one-sided score. For MBGS, the resulting drop in AUC for the leading LBP features is from 82.6 to 81.2; for SVM⊖ the drop is from 83.6 to 81.9.

Finally, in order to examine which examples are most likely to benefit from the boost in performance obtained from the SVM⊖ similarity in comparison to MBGS, we have provided additional measurements to each video sequence and to each pair by examining the minimal measurement value of the two associated videos. These measurements include (1) the amount of variability in appearance, as captured by the norm of the covariance matrix of the descriptors of each video; (2) the area in squared pixels of the face region (a proxy for image quality); (3) the amount of translation of the face region in the video; (4) the mean value of each 3D head orientation angle; and finally, (5) the variance of each of these angles.

For each of the three descriptors, each of the $5,000$ pairs was scored by the difference in their ranking among all pairs by MBGS and the ranking obtained by the SVM⊖ similarity. In other words, the pair with the highest LBP based SVM⊖ similarity was given a score of $5,000$ minus the ranking it obtained using LBP-based MBGS. The higher the difference-of-ranks is, the more a pair was influenced by the introduction of the SVM⊖ similarity. Fig. 5 depicts for each descriptor, the pair that was most affected by the shift from MBGS to SVM⊖. As can be seen at least one video in each pair contains considerable head motion.

Spearman correlations between these three scores and the five measurements described above were computed. The only correlations that were significant at a confidence level of 0.05 were the ones between the FPLBP ranking or the LBP ranking and the measured variance of the yaw head orientation angle (p-values of 0.05 and 0.04 respectively).

Method	CSLBP			FPLBP			LBP		
	Accuracy ± SE	AUC	EER	Accuracy ± SE	AUC	EER	Accuracy ± SE	AUC	EER
Min dist	62.9 ± 1.1	67.3	37.4	65.6 ± 1.8	70.0	35.6	65.7 ± 1.7	70.7	35.2
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
CMSM	61.2 ± 2.6	65.2	39.8	63.8 ± 2.0	68.4	37.1	62.9 ± 1.8	67.3	38.4
$\|\|U_1\ U_2\|\|_F$	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
MBGS	72.4 ± 2.0	78.9	28.7	72.6 ± 2.0	80.1	27.7	76.4 ± 1.8	82.6	25.3
SVM⊖	70.0 ± 2.7	79.4	28.4	71.1 ± 3.6	80.1	27.6	73.6 ± 2.5	83.6	24.7
MBGS + SVM⊖	72.6 ± 2.1	81.8	26.1	76.0 ± 1.7	83.7	24.9	78.9 ± 1.9	86.9	21.2

Table 1. Benchmark results obtained for various similarity measures and image descriptors. See text for the description of each method.

Figure 5. Each row contains example frames from one pair of videos, which was ranked highest by the magnitude of the difference between MBGS and the SVM⊖ similarity. The three rows correspond to the three face descriptors: CSLBP, FPLBP, and LBP.

		MBGS			SVM⊖		
		CSLBP	FPLBP	LBP	CSLBP	FPLBP	LBP
MBGS	CSLBP	1.0	0.78	0.92	0.68	0.49	0.47
	FPLBP	0.78	1.0	0.85	0.57	0.63	0.44
	LBP	0.92	0.85	1.0	0.64	0.52	0.52
SVM⊖	CSLBP	0.68	0.57	0.64	1.0	0.66	0.68
	FPLBP	0.49	0.63	0.52	0.66	1.0	0.65
	LBP	0.47	0.44	0.51	0.68	0.65	1.0

Table 2. Pairwise correlations among MBGS and SVM⊖ similarity scores on the 5,000 benchmark pairs.

8. Discussion and future work

Face recognition in video deserves attention not just because of its wide applicability, but also since the algorithmic challenges it raises are largely unresolved. First and foremost is the intuitive expectation that face recognition in video should be at least as accurate as image-based face recognition. While the inverted gap in performance could be partially explained by contemporary (past?) issues such as video resolution and compression artifacts, we believe that the additional information in video should be more than enough to compensate for these.

Initial approaches for face recognition in video were based on the linear subspace or manifold models. Such approaches are not robust enough for unconstrained video. More generally, the problem of comparing sets of vectors is a corner stone in modern object recognition, where PMK and LLC have been shown to provide excellent results when applied to sets of image descriptors. However, algorithms designed for large sets of local pieces of information are not effective for the problem at hand, which is characterized by smaller sets of very informative vectors containing a large amount of overlapping information.

Classifier based approaches such as those studied here, are more robust to the overlap in the frames' information, since classifiers are designed to be robust to uneven distributions of training example. Nevertheless, most classifiers are guaranteed to generalize well in cases where the train and test distributions are similar, which does not hold here. The effect of this issue should be further examined.

In this work we rely on the fact that the most prominent confounding factor – the 3D head orientation – is observable, and derive a new similarity score which discounts the spurious likeness that is induced by pose similarity. This novel similarity employs a new SVM variant called SVM⊖, which unlike SVM+, tries to "unlearn" the separation induced by pose. We note that in contrast to the conventional privileged knowledge scenario, the side information is available but unused even when the SVM⊖ model is applied as part of the SVM⊖ similarity score. The exploitation of this extra source of information is left for future research.

References

[1] YouTube Faces DB. www.cs.tau.ac.il/~wolf/ytfaces. 2

[2] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010. 2

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998. 2

[4] G. Chechik and N. Tishby. Extracting relevant structures with side information. In *NIPS*, 2002. 2

[5] J. Chen, X. Liu, and S. Lyu. Boosting with side information. In *ACCV*, 2012. 2

[6] Y. Chikuse. *Statistics on special manifolds, lecture notes in statistics, vol. 174*. New York: Springer, 2003. 1, 6

[7] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20:303–353, April 1999. 1, 6

[8] M. Everingham, J. Sivic, and A. Zisserman. "hello! my name is... buffy" - automatic naming of characters in tv video. In *BMVC*, 2006. 5

[9] 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. 1

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010. 2

[11] J. Feyereisl and U. Aickelin. Privileged information for data clustering. *Inf. Sci.*, 194:4–23, July 2012. 2

[12] A. Globerson, G. Chechik, and N. Tishby. Sufficient dimensionality reduction w/ irrelevance statistics. In *UAI*, '03. 2

[13] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005. 1, 5

[14] 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. 5

[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, TR 07-49, 2007. 2, 5

[16] T.-K. Kim, O. Arandjelovic, and R. Cipolla. Boosted manifold principal angles for image set-based recognition. *Pattern Recognition*, 40(9):2475–2484, 2007. 1

[17] A. Kocsor, K. Kovcs, and C. Szepesvri. Margin maximizing discriminant analysis. In *ECML*, 2004. 2

[18] E. Krupka et al. Incorporating prior knowledge on features into learning. In *AISTATS*. 2

[19] N. Levy and L. Wolf. Minimal correlation ensemble. In *ECCV*, 2012. 2

[20] Y. Liu and X. Yao. Simultaneous training of negatively correlated neural networks in an ensemble. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 29:716–725, 1999. 2

[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. 5

[22] D. Pechyony, R. Izmailov, A. Vashist, and V. Vapnik. Smo-style algorithms for learning using privileged information. In *DMIN*, 2010. 2

[23] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(10):1848–1852, 2007. 2

[24] D. Ramanan, S. Baker, and S. Kakade. Leveraging archival video for building face datasets. In *ICCV*, 2007. 1

[25] B. Raytchev and H. Murase. Unsupervised face recognition from image sequences based on clustering with attraction and repulsion. In *CVPR*, 2001. 1

[26] G. Shakhnarovich, J. Fisher, and T. Darrell. Face recognition from long-term observations. In *ECCV*, 2002. 1

[27] G. Shakhnarovich, P. Viola, and B. Moghaddam. A unified learning framework for real time face detection and classification. In *Auto. Face & Gesture Recognition*, 2002. 1

[28] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. In *ECCV*, 2002. 2

[29] J. Sivic, M. Everingham, and A. Zisserman. "Who are you?": Learning person specific classifiers from video. In *CVPR*, 2009. 1

[30] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *BMVC*, 2009. 2

[31] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Netw.*, 2009. 2

[32] D. Vaquero, R. Feris, D. Tran, L. Brown, A. Hampapur, and M. Turk. Attribute-based people search in surveillance environments. In *WACV*, 2009. 1

[33] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010. 5

[34] R. Wang, S. Shan, X. Chen, and W. Gao. Manifold-manifold distance with application to face recognition based on image set. In *CVPR*, 2008. 1

[35] 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. 1

[36] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, pages 529–534, 2011. 2, 3, 5, 6

[37] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *post-ECCV Faces in Real-Life Images Workshop*, 2008. 2, 5

[38] L. Wolf, T. Hassner, and Y. Taigman. The one-shot similarity kernel. In *ICCV*, 2009. 2

[39] D. H. Wolpert. Stacked generalization. *Neural Netw.*, 5(2):241–259, 1992. 6

[40] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. In *Automatic Face and Gesture Recognition*, 1998. 1, 6