Fast Face Recognition by Using an Inverted Index

Christian Herrmann[a,b] and Jürgen Beyerer[b,a]

[a]Vision and Fusion Lab, Karlsruhe Institute of Technology KIT, Karlsruhe, Germany
[b]Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Karlsruhe, Germany

ABSTRACT

This contribution addresses the task of searching for faces in large video datasets. Despite vast progress in the field, face recognition remains a challenge for uncontrolled large scale applications like searching for persons in surveillance footage or internet videos. While current productive systems focus on the best shot approach, where only one representative frame from a given face track is selected, thus sacrificing recognition performance, systems achieving state-of-the-art recognition performance, like the recently published DeepFace, ignore recognition speed, which makes them impractical for large scale applications. We suggest a set of measures to address the problem. First, considering the feature location allows collecting the extracted features in according sets. Secondly, the inverted index approach, which became popular in the area of image retrieval, is applied to these feature sets. A face track is thus described by a set of local indexed visual words which enables a fast search. This way, all information from a face track is collected which allows better recognition performance than best shot approaches and the inverted index permits constantly high recognition speeds. Evaluation on a dataset of several thousand videos shows the validity of the proposed approach.

Keywords: face recognition, large scale, inverted index, video

1. INTRODUCTION

Besides the obviously vast collections of video portals like YouTube, large amounts of video footage are also present in surveillance scenarios or the increasing number of TV-channels. Finding specific persons in the data is a still existing challenge. For example, in the context of forensic analysis of surveillance footage, a typical challenge is to find all appearances of a specific person, probably a criminal, in the given data for crime reconstruction. Another example might be to find all YouTube videos containing a specific celebrity. While the latter situation offers further clues like video tags or titles, the former situation requires a solely image based analysis. In this contribution, we focus on a pure video based solution by analyzing the video content, thus covering all scenarios. The easiest way to identify persons in video data is by their face. The first necessary steps before face recognition are face detection, alignment and tracking. Because this is a whole field of research itself, we assume that a solution to these steps is available. The focus is to compare and match the extracted face tracks to a given query.

A high recognition speed combined with a decent recognition performance is achieved by a set of measures. First, collecting all local image features of a face track in a single feature set allows the application of classical image retrieval methods. Secondly, the recognition accuracy is enhanced by using the location of the image features and asserting that only features from the same location will be compared. Thirdly, the inverted index approach, which became popular in the area of image retrieval, is applied to that feature set. A face track is thus described by a set of indexed visual words and for each word a reference to this track is stored in the database index. Searching the database for a given person requires only an index lookup for the respective visual words of the face track. Because the index data structure can be pre-computed for database lookups, query time is low. Evaluation on two public datasets containing several thousand videos shows the validity of the proposed approach. This work addresses two areas of video face recognition:

Further author information:
C. Herrmann: christian.herrmann@iosb.fraunhofer.de
J. Beyerer: juergen.beyerer@iosb.fraunhofer.de

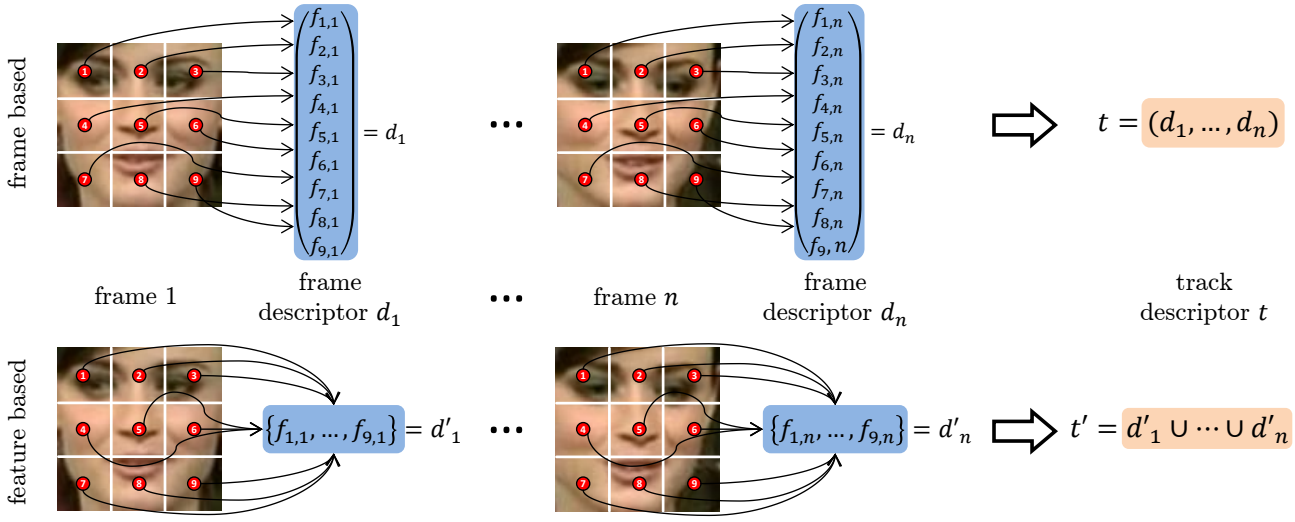

Figure 1. Illustration of the conventional frame based track description method (top) and the applied feature set based one (bottom). Local features are denoted by $f_{i,j}$.

Face track description. The usual way to build a face track descriptor from video data is a two step strategy. In the first step, each frame is represented by a frame descriptor. There is a large variety of descriptors available from image based face recognition: gray scale intensity, Eigenfaces[1], Fisherfaces[2], LBP[3], Gabor[4] to name only a few. In the second step, a track descriptor is derived from the sequence of frame descriptors, for example by taking the mean over all frames on image[5], feature[6,7] or decision level[8,9]. Further options include modeling the space of the frames by a linear model[10,11], a manifold[12,13] or performing a pairwise comparison of all[14], randomly selected[15] or the best-shot[14] frame descriptors and searching for the closest match. Pairwise comparison takes considerable time for larger numbers of involved frames, influenced by track length and percentage of selected frames. In the case of the currently best performing video based face recognition algorithm DeepFace[15], which compares randomly sampled frames, simulations with the reported feature dimensions and frame numbers indicate matching speeds of only 500 track to track comparisons per second, which is insufficient for large scale applications. Promising methods with respect to matching speed are the ones using small track descriptors and fast comparison strategies. Namely these are the mutual subspace method (MSM)[11] and the best-shot approach. Especially the best-shot approach where the best frame according to some criterion (e.g. most frontal pose or least blurred), is used to represent the whole track, is widely used in time critical applications.

Instead of the frame based strategy, we follow the suggestions of a few recently proposed approaches, which use a local feature based representation[16,17]. Local features are collected over all frames and put together into one feature set. We propose to use this feature set as base of a bag of visual words descriptor with an inverted index[18], which enables the construction of large databases and performing fast queries. One advantage of the bag of visual words descriptor is its independence of the track length, making comparison tasks independent thereof.

Spatial feature information. Augmenting the local features by the respective image coordinates proved useful for face recognition tasks[16,17]. In the previous contributions, augmentation is performed by concatenation of the feature vector and the 2D image coordinate vector. Instead of a concatenation, we propose a different way to use spatial information, constructing a separate feature set for each of a few fixed feature locations.

2. FRAME FEATURES

As argued before, instead of using descriptors based on whole frames, a different strategy is applied as illustrated by figure 1. For comparison, the conventional frame based method is shown at the top, which uses a face descriptor d_j for each frame j, built by the concatenation of several local features vectors $f_{i,j}$, where i denotes the feature location. The final track descriptor is derived from the sequence $(d_1, \dots, d_n)$ of all n frame descriptors. The feature based track description is shown at the bottom. In this case the face descriptor d'_j is only a mathematical utility, but has no meaning by itself.

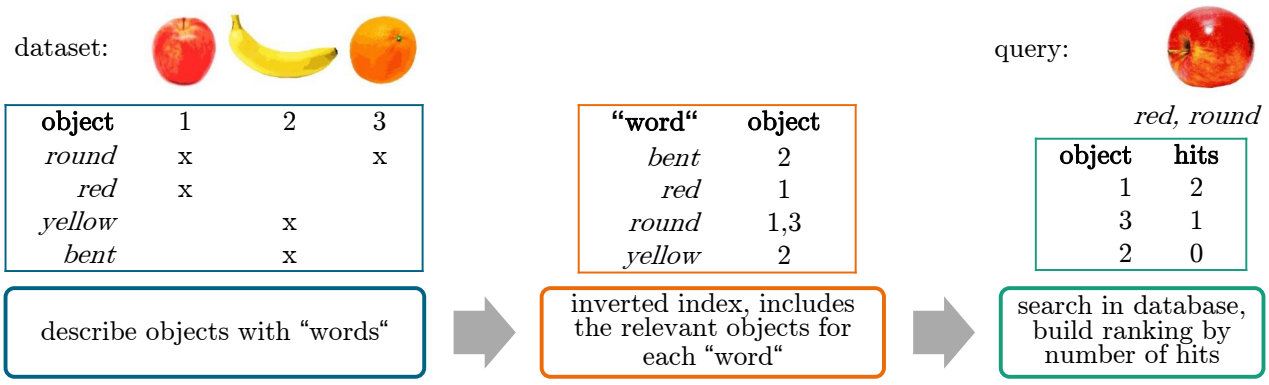

Figure 2. Illustration of the inverted index approach with a basic example using images of fruit instead of face tracks and textual words instead of visual ones.

Basically, all local features $f_{i,j}$ are combined to one feature set, which is used as track descriptor t'. There are three advantages for this method: First, the dimension D' of the vectors in t' (feature vectors f) is lower than that of the vectors in t (frame descriptors d). Thus, further processing can be performed faster, because basically all matching approaches scale at least linearly with D'. Secondly, this representation ignores temporal information. While loosing information is generally a bad idea, it is the opposite in this case, because temporal information includes no clues about a persons identity. For example, the fact that the head rotates in the face track includes no information about who is rotating his head. Thirdly, the feature based representation is widely used in object or image retrieval tasks, which means according approaches can be applied to face retrieval too.

We employ local binary patterns (LBP)[3] as local features and combine several scales by summation of the LBP histograms[19] in each local region. Each face image is split into a grid of $k \times k$ regions, where the region center denotes the location of the local feature. The LBP histogram is built over all LBP patterns inside of a local region. Overall, the proposed strategy results in a set S of $L = k^2 n$ local features for a track with n frames.

3. BAG OF WORDS AND INVERTED INDEX

Generally speaking, a retrieval scenario involves a database D of N objects and a query object Q. The task is to find all matching objects to the query object in the database. When targeting large scale retrieval applications, the inverted index method is a well-known approach. Basically, this includes three steps, shown also by figure 2:

Description of objects with visual words. Each object, in our case each face track, is described by a set of predefined visual words. Possible visual words are defined by a codebook (dictionary) which is constructed by clustering all the object features of the database in K classes and using the cluster centers $C_1, \ldots, C_K$ as visual words. This way, the codebook consists of domain specific visual words. For each object, the feature set S from the previous section is computed and the matching words are found by assigning each feature to the nearest visual word. The set of occurring words represents the object.

Building an inverted index for the whole database of objects. To avoid linear search for the best matches in the database, an inverted index is used. This means an index with the visual words from the codebook is constructed and for each visual word a list of objects including this word is maintained.

Database query by index search for visual word in query object. Performing a search for a query object requires first to find the visual words for the query object. Then, for each visual word of the query object the matching database objects are looked up in the index. Finally, counting the number of hits for the matching database objects results in a ranking. Note that in large scale applications it is common that a significant part of the database objects has no hits at all.

Using the inverted index strategy without adaptation has a serious drawback. Because all visual features are put together into a single feature set, their location in the face is lost. While this behavior is usually desired in image retrieval, because it

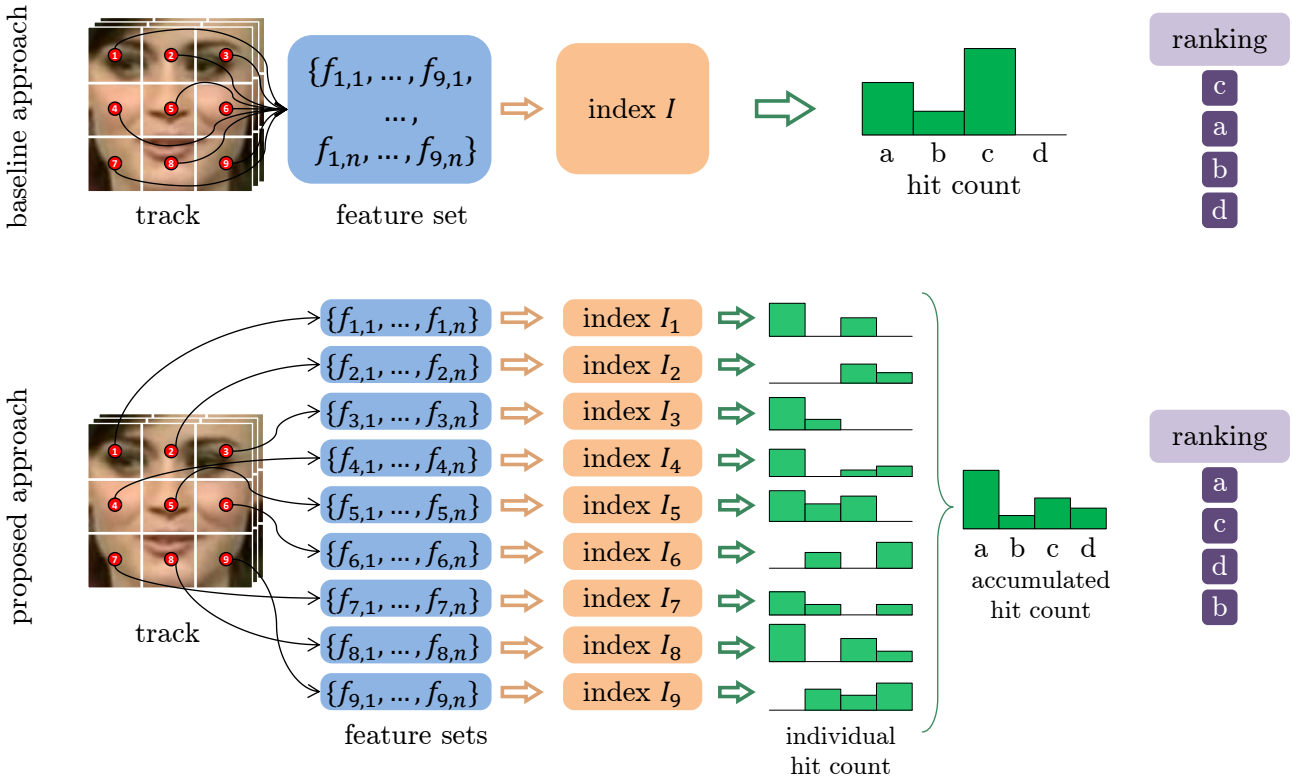

Figure 3. Comparison of basic (top) and proposed (bottom) strategy. Illustration shows a query process for a small sample database whose face tracks are called a, b, c and d.

guarantees invariance to rotation, scaling and shifting, it is counterproductive in face retrieval. Face detections are always aligned, thus they have a known and fixed rotation, scaling and are shifted equally. This way, the feature location is meaningful in contradiction to image retrieval and it can be used to improve the results. Because the nose is always in the middle, eyes at the top, mouth at the bottom and so on, comparing features from different locations is unnecessary. It has no meaning if the nose of one person shows the same feature as the eye of another one. Thus, instead of using one single index for all features, we suggest to use separate local indices as shown by figure 3 at the bottom. Each feature location in the grid, is handled individually and results are combined at the end by accumulating the hit counts from the different index searches.

Practical issues: The inverted index method includes to basic problems where fast algorithms are necessary. First, the clustering of a large dataset to build the index and secondly nearest neighbor search to assign the corresponding visual word to a feature. The VLFeat library[20] is used in both cases because it uses efficient algorithms based on KD-trees.

4. EXPERIMENTS

To show the benefits of the proposed method, evaluation is performed on the largest publicly available datasets YouTube Faces Database (YTF)[14] and Face in Action Database (FiA)[21]. While YTF is an in the wild dataset with 3,425 face videos originating from YouTube containing celebrities, FiA data was recorded in the lab in a controlled environment with fixed camera positions and predefined head movements resulting in 3,110 face videos (only indoor sequences are used).

4.1 Protocol

The retrieval evaluation protocol is a 10 fold strategy: the dataset is divided into 10 splits and each one will be used one after another as query split, while the remaining 9 splits build the database. Each track in a query split is used to query the database, which results, over all 10 splits, in M queries, where M is the dataset size.

Case	Algorithm	$\bar{K}$	Feature	Local indices	map	rand. test	t_q in s	t_m in s
1	inv. index	64000	LBP	no	0.013		0.062	$4.50 \cdot 10^3$
2	inv. index	64000	LBP	yes	**0.052**	1	0.050	$0.51 \cdot 10^3$
3	inv. index	64000	Intensity	yes	0.046	5	0.037	$0.86 \cdot 10^3$
4	inv. index	64000	LBP	yes	**0.052**	3,5	0.050	$0.51 \cdot 10^3$
5	inv. index	64000	LDP	yes	0.022		0.087	$0.52 \cdot 10^3$
6	inv. index	64000	LBP	yes	0.052		0.050	$0.51 \cdot 10^3$
7	inv. index	128000	LBP	yes	0.058	6	0.067	$0.91 \cdot 10^3$
8	inv. index	256000	LBP	yes	0.061	6,7	0.116	$1.78 \cdot 10^3$
9	inv. index	512000	LBP	yes	**0.067**	6,7,8	0.123	$3.66 \cdot 10^3$

Table 1. Evaluation of different features and parameters. Randomization test column denotes case numbers which yielded significantly worse results.

Performance is measured by the average precision a for each query and overall given by the mean average precision map. The average precision measures a the quality of the resulting ranking for a query by the recall r and precision p. The recall $r = \frac{TP}{TP+FN}$ denotes the percentage of the number of retrieved correct matches TP and the number of all possible correct matches $TP + FN$ in a database. TP, FP and FN are notations from classification tasks, meaning *true positives*, *false positives* and *false negatives*. This way, the precision $p = \frac{TP}{TP+FP}$ denotes the ratio of correct hits in the returned query. Let $r(k)$ denote the recall for retrieval results consisting of the ranks 1 to k and $p(k)$ the respective precision. Then the average precision for one ranking is given by $a = \sum_{k=1}^{K} \Delta r(k) \cdot p(k)$, which is a weighted average of the precision over all ranks. Finally, the mean average precision is the mean over all queries: $map = \frac{1}{M} \sum_{m=1}^{M} a_m$. The map ranges between 0 and 1, where a value of 1 signals a perfect result with all the correct matches at the top of the ranking.

Significant pairwise differences between measured values are determined by a randomization test[22], using an α-level of 0.05, which corresponds to a confidence of about 2 standard deviations. In comparison to simply giving the mean and standard deviation, it has the advantage to statistically exploit the large number of queries which are performed in this experimental setup. Thus, it is more accurate in showing significant differences between retrieval algorithms.

In contrast to a simple verification protocol, where 10-fold cross-validation provides only a shallow statistical base for proving significant differences between approaches, the retrieval protocol offers M samples. In consequence, significant pairwise differences between measured values are determined by a randomization test[22], using an α-level of 0.05, which corresponds to a confidence of about 2 standard deviations. In comparison to reporting only the mean and standard deviation, it has the advantage to statistically exploit the large number of queries which are performed in this experimental setup. Thus, it is more accurate in showing significant differences between retrieval algorithms.

4.2 Parameters

In the first set of experiments, parameter variations for the proposed method are evaluated on the YTF dataset and results are shown in table 1. Besides the map and the respective randomization tests, the mean time t_q for one query and the database construction time t_m are given.

Local indices. Using a separate local index for each spatial feature location enhances the retrieval results significantly (case 2). Thus, mixing together different features causes confusion in the recognition process and the separation solves this issue. For fair comparison between the baseline global inverted index and the local inverted indices, the sum $\bar{K}$ of the respective dictionary sizes is given. For the baseline (case 1) this means $\bar{K} = K$ is the size of the single dictionary, while in the case of local indices each dictionary has the size of $K = \frac{\bar{K}}{k^2}$, where k^2 is the number of local regions as in section 2.

Feature. Comparison to different features, namely raw pixel intensities (case 4) and local directional patterns (LDP),[23] indicates that the proposed usage of LBP from section 2 is justified. $k = 4$ local regions are used in all cases because it proved to be the best subdivision for resolutions on this level[19].

No.	Method	map YTF	FiA	comb.	query time t_q in s YTF	FiA	comb.
1	NN	0.145^{2}	0.351^{4}	0.255^{4}	12.31	10.37	30.51
2	MSM	0.084^{4}	0.237^{3}	0.170^{3}	0.281	0.150	0.428
3	best shot	0.056	0.147	0.103	0.074	0.069	0.134
4	inv. index	0.067^{3}	0.297^{2}	0.183^{2}	0.123	0.103	0.114

Table 2. Evaluation results on YTF and FiA public datasets, as well as a combination of both. Superscripts indicate results of randomization test: a method is significantly better than the one indicated by the superscript, including every worse one.

Index size. Retrieval results get better with an increasing index size (cases 6-9). Further increases are prevented by the limited memory of our test system. Thus for the further experiments in the next section we use the setting from case 9.

4.3 Comparison

As already stated in the introduction, only few face recognition approaches are capable of fast matching for large scale face retrieval. Thus the number of possible baseline approaches remains limited for comparison. Namely, we employ MSM, pairwise-frame matching (NN) and best-shot on the same LBP features. The final results are shown in table 2 for evaluation on YTF, FiA and the combination of both datasets. Although, NN shows clearly the highest map, this is bought by heavy computational demands which are impractical for real applications. MSM and best-shot decrease the mean query time t_q significantly, however, both are a trade-off between speed and accuracy. The proposed inverted index method manages to break this trade-off in certain limits. While results for YTF fall in between the results of the best-shot and MSM method, they are significantly better for FiA. Comparing the results of the best-shot method with the inversed index results from the previous section (table 1, case 7) indicates that the proposed method has a better recognition performance per time ratio than the best-shot method. Finally, the combination of both datasets clearly shows the advantage of the inverted index method: the mean query times remains constant and independent of database size, thus making it the fastest retrieval method for larger datasets. The reason is the avoidance of a linear search in the database. Thus, increasing the database size by a combination of both datasets has only minor influences on the query time for the inverted index compared to the baseline approaches, where the query time roughly doubles with the doubled dataset size.

5. CONCLUSION

A face retrieval method based on local features and an inverted index is proposed. By using a separate local index for each spatial feature location instead of only one global index, the recognition accuracy for the inverted index approach can be increased significantly. This way, the widely used best-shot method is outperformed while showing smaller query times for large scale problems. The key benefit of the proposed system is its query time independence of the database size, which promises an increasing advance with growing datasets.

REFERENCES

[1] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience* **3**(1), pp. 71–86, 1991.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7), pp. 711–720, 1997.

[3] T. Ahonen, A. Hadid, and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," *Pattern Analysis and Machine Intelligence* **28**(12), pp. 2037–2041, 2006.

[4] J. Zou, Q. Ji, and G. Nagy, "A Comparative Study of Local Matching Approach for Face Recognition," *IEEE Transactions on Image Processing* **16**(10), pp. 2617–2628, 2007.

[5] R. Jenkins and A. Burton, "100% Accuracy In Automatic Face Recognition," *Science* **319**(5862), pp. 435–435, 2008.

[6] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Computer Vision and Pattern Recognition*, pp. 1875–1882, 2014.

[7] E. G. Ortiz, A. Wright, and M. Shah, "Face recognition in movie trailers via mean sequence sparse representation-based classification," in *Computer Vision and Pattern Recognition*, 2013.

[8] M. Tapaswi, M. Bauml, and R. Stiefelhagen, "Knock! Knock! Who is it? probabilistic person identification in TV-series," in *Computer Vision and Pattern Recognition*, pp. 2658–2665, 2012.

[9] L. Wolf and N. Levy, "The svm-minus similarity score for video face recognition," in *Computer Vision and Pattern Recognition*, 2013.

[10] H. Cevikalp and B. Triggs, "Face recognition based on image sets," in *Computer Vision and Pattern Recognition*, 2010.

[11] O. Yamaguchi, K. Fukui, and K. Maeda, "Face Recognition Using Temporal Image Sequence," in *Automatic Face and Gesture Recognition*, 1998.

[12] O. Arandjelović and R. Cipolla, "A pose-wise linear illumination manifold model for face recognition using video," *Computer vision and image understanding* **113**(1), pp. 113–125, 2009.

[13] K. Lee, J. Ho, M. Yang, and D. Kriegman, "Video-Based Face Recognition Using Probabilistic Appearance Manifolds," *Computer Vision and Pattern Recognition* **1**, pp. 313–320, 2003.

[14] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition*, 2011.

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

[16] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in *Computer Vision and Pattern Recognition*, 2013.

[17] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman, "A Compact and Discriminative Face Track Descriptor," in *Computer Vision and Pattern Recognition*, 2014.

[18] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *International Conference on Computer Vision*, pp. 1470–1477, 2003.

[19] C. Herrmann, "Extending a local matching face recognition approach to low-resolution video," in *Advanced Video and Signal Based Surveillance*, 2013.

[20] A. Vedaldi and B. Fulkerson, "VLFeat: An Open and Portable Library of Computer Vision Algorithms." `http://www.vlfeat.org/`, 2008.

[21] R. Goh, L. Liu, X. Liu, and T. Chen, "The CMU Face In Action (FIA) Database," *Analysis and Modelling of Faces and Gestures* , pp. 255–263, 2005.

[22] M. D. Smucker, J. Allan, and B. Carterette, "A comparison of statistical significance tests for information retrieval evaluation," in *Information and Knowledge Management*, 2007.

[23] T. Jabid, M. H. Kabir, and O. Chae, "Local directional pattern (LDP) for face recognition," in *Consumer Electronics*, pp. 329–330, 2010.