Spatial Bag of Features Learning for Large Scale Face Image Retrieval

Nikolaos Passalis(✉) and Anastasios Tefas

Aristotle University of Thessaloniki, Thessaloniki, Greece
passalis@csd.auth.gr, tefas@aiia.csd.auth.gr

Abstract. In this paper a supervised codebook learning technique for the Bag-of-Features representation that optimizes the learned codebooks towards face retrieval is proposed. This allows to use significantly smaller codebooks reducing both the storage requirements and the retrieval time allowing the proposed technique to efficiently scale to large datasets. The proposed method is also combined with a spatial image segmentation technique that exploits the natural symmetry of the human face to further reduce the size of the extracted representation. It is experimentally demonstrated using one large-scale face recognition dataset, the YouTube Faces Database, as well as two smaller datasets, that the proposed technique can increase the retrieval precision, while reducing the encoding time by almost two orders of magnitude.

1 Introduction

Large-scale face image retrieval has recently received a lot of attention, especially in the context of celebrity face image retrieval [2]. Face image retrieval is defined as the task of retrieving face images of a person given a query image that depicts the face of that person. An image must be retrieved even if the person has a different facial expression, pose, or different illumination conditions exist. Given the vast amount of face images available on the Internet, a large-scale face retrieval technique must be able to successfully handle large amounts of data. Facial image retrieval also poses challenges of high-dimensionality, velocity and variety.

A wide range of methods have been proposed for face recognition and retrieval [18]. Recently, a widely known technique for image retrieval, the Bag-of-Features (BoF) model [4], also known as Bag-of-Visual Words (BoVW) model, has been applied for face recognition/retrieval after being appropriately modified [13,16,17]. The typical pipeline of the BoF model can be summarized as follows. First, multiple features, such as SIFT descriptors [9], are extracted from each image (feature extraction). That way, the *feature space* is formed where each image is represented as an unordered set of features. Then, the extracted features are used to learn a *codebook* of representative features (also called *codewords*). This process is called *codebook learning*. Finally, each feature vector is represented using a codeword from the learned codebook and a histogram is

P. Angelov et al. (eds.), *Advances in Big Data*, Advances in Intelligent Systems and Computing 529, DOI 10.1007/978-3-319-47898-2_2

extracted for each image. That way, the *histogram space* is formed where each image is represented by a constant dimensionality histogram vector.

The BoF model discards most of the spatial information contained in the original image, which can severely harm the face recognition precision. To overcome this limitation, the BoF-based techniques for face recognition, e.g., [13,16,17], define a grid over each image (or some regions of interest) and independently extract a histogram from each cell of the grid. These techniques tend to be invariant to facial expression, pose, and illumination variations when used for face recognition using a trained classifier [13]. However, when the extracted representation is used for face image retrieval the problem becomes ill-defined, since the user's information need is sometimes ambiguous. For example, if the query image depicts a smiling person the system might retrieve face images from the same person, but it might also retrieve images from other persons that smile. Therefore, it is critical that the extracted representation is appropriately tuned for the given retrieval task.

The main contribution of this paper is the proposal of a supervised codebook learning technique that is able to learn face retrieval-oriented codebooks using an annotated training set of face images. This allows to use significantly smaller codebooks reducing both the storage requirements and the retrieval time by almost two orders of magnitude and allowing the technique to efficiently scale to large datasets. The proposed method is also combined with a spatial image segmentation technique that exploits the natural symmetry of the human face to further reduce the size of the extracted representation. Furthermore, the proposed approach is able to learn in incremental mode without requiring in-memory access to large amounts of data.

The rest of the paper is structured as follows. The related work is discussed in Sect. 2 and the proposed method is presented in Sect. 3. The experimental evaluation using one large-scale dataset, the YouTube Faces Database and two smaller datasets, the ORL Database of Faces and the Extended Yale Face Database B, is presented in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 Related Work

This work mainly concerns codebook learning for face image retrieval using the BoF model. A rich literature exists in the field of codebook learning for the BoF representation, ranging from supervised approaches [3,8,10], to unsupervised ones [11]. Despite the fact that supervised codebook learning is well established in general computer vision tasks, such as scene classification [8], or action recognition [3], little work has been done in the area of face image retrieval. One application of supervised codebook learning for face image retrieval is presented in [16], where a simple identity based codebook construction technique is proposed. This method is combined with hamming signatures and locality-sensitive hashing (LSH) in order to be applied to large datasets. In contrast to this, the method proposed in this paper reduces the size of the codebook, instead of relying on hashing and approximate nearest neighbor search techniques, allowing

the method to natively scale to large datasets. Nonetheless, these techniques can be still combined with the proposed method to further increase the retrieval performance.

All the proposed BoF-based image recognition/retrieval techniques either use a grid over each image [13,17], or define multiple grids over some points of interest [16], and process each cell independently using a separate codebook. This introduces spatial information to the extracted representation leading to the Spatial BoF (SBoF) model. In our approach each (aligned) face image is split into four horizontal strips instead of dividing each image using a grid. This allows to further reduce the length of the extracted representation, without significantly affecting the retrieval precision.

3 Proposed Method

3.1 Spatial Bag-of-Features Model

Before presenting the proposed codebook learning method, the BoF model and the SBoF model are briefly described. Let N be the number of face images that are to be encoded using the regular BoF model. The i-th image is described by N_i feature vectors: $\mathbf{x}_{ij} \in \mathbb{R}^D$ $(i = 1...N, j = 1...N_i)$, where D is the length of the extracted feature vectors. In this work, dense SIFT features [6], are extracted from 16×16 patches using a regular grid with spacing of 4 pixels. The BoF model represents each face image using a fixed-length histogram of its quantized feature vectors, where each histogram bin corresponds to a codeword. In hard assignment each feature vector is quantized to its nearest codeword/histogram bin, while in soft-assignment every feature vector contributes, by a different amount, to each codeword/bin.

In order to learn a codebook the set of all feature vectors $\mathscr{S} = \{\mathbf{x}_{ij} | i = 1...N, j = 1...N_i\}$ is clustered into N_K clusters and the corresponding centroids (codewords) $\mathbf{v}_k \in \mathbb{R}^D (k = 1...N_K)$ are used to form the codebook $\mathbf{V} \in \mathbb{R}^{D \times N_K}$, where each column of $\mathbf{V}$ is a centroid vector. These centroids are used to quantize the feature vectors. It is common to cluster only a subset of $\mathscr{S}$ since this can reduce the training time with little effect on the learned representation. The codebook is learned only once and then it can be used to encode any new face image.

To encode the i-th face image, the similarity between each feature vector $\mathbf{x}_{ij}$ and each codeword $\mathbf{v}_k$ is computed as: $d_{ijk} = exp(\frac{-||\mathbf{v}_k - \mathbf{x}_{ij}||_2}{g}) \in \mathbb{R}$. The parameter g controls the quantization process: for harder assignment a small value, i.e., $g < 0.01$, is used, while for softer assignment larger values, i.e., $g > 0.01$, are used. Then, the l^1 normalized membership vector of each feature vector $\mathbf{x}_{ij}$ is obtained by: $\mathbf{u}_{ij} = \frac{\mathbf{d}_{ij}}{||\mathbf{d}_{ij}||_1} \in \mathbb{R}^{N_K}$. This vector describes the similarity of feature vector $\mathbf{x}_{ij}$ to each codebook vector. Finally, the histogram $\mathbf{s}_i$ is extracted as $\mathbf{s}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{u}_{ij} \in \mathbb{R}^{N_K}$. The histogram $\mathbf{s}_i$ has unit l^1 norm, since $||\mathbf{u}_{ij}||_1 = 1$ for every j. These histograms describe each image and they can be used to retrieve

relevant face images. The training and the encoding process are unsupervised and no labeled data are required.

As previously mentioned, the BoF model discards most of the spatial information contained in the images during the encoding process. This can severely harm the face recognition precision, since most of the discriminant face features, e.g., eyes, nose, mouth, are expected to be found in the same position when the face is properly aligned. This allows to have highly specialized codebooks for each of these regions.

In this work the following segmentation technique is used for the SBoF model. Each image is segmented into N_r equally spaced horizontal stripes. Since the human face is symmetric to a great extent, no vertical segmentation is used. A separate codebook is learned for each strip and N_r histograms are extracted (one from each strip). These histograms are fused into the final histogram vector and renormalized. The length of this vector is $N_r \times N_K$. When four regions are used, i.e., $N_r = 4$, and the images are correctly aligned, each region roughly corresponds to one of the following parts of the human face: forehead, eyes, nose and mouth. Although the images can be further split to more regions (either horizontal either vertical or both), this provides little increase in retrieval precision. This fact is experimentally validated in Sect. 4. Note that the storage requirements and the retrieval time depend on the number of regions used to segment each image, since the storage required for each image is $N_r \times N_K \times B$ bytes. This technique also provides mirror-invariance, while still preserving the discriminant non-symmetric facial features that might appear.

3.2 Spatial Bag-of-Features Learning

The goal of the proposed optimized SBoF technique, abbreviated as O-SBoF, is to learn codebooks that minimize the entropy in the histogram space using a training set of face images, where the i-th image is annotated by a label $l_i \in \{1, ..., N_C\}$ and N_C is the number of individuals (classes) in the training set. Intuitively, the entropy in the histogram space is minimized when the histograms are gathered in pure clusters, i.e., each cluster contains face images of the same person.

To simplify the presentation of the proposed method, it is assumed that only one region exists in each image, i.e., the feature vectors are extracted from the whole image. This is without loss of generality, since the method can be independently applied to optimize the codebook of each region/strip by considering each region/strip as an image.

In order to measure the entropy in the histogram space, the vectors $\mathbf{s}_i$ are clustered into N_T clusters. The centroid of the k-th cluster is denoted by $\mathbf{c}_k$ ($k = 1...N_T$). Then, the entropy of the k-th cluster can be defined as:

$$E_k = -\sum_{j=1}^{N_C} p_{jk} \log p_{jk} \tag{1}$$

where p_{jk} is the probability that an image of the k-th cluster belongs to the class j. This probability is estimated as $p_{jk} = h_{jk}/n_k$, where n_k is the number of images in cluster k and h_{jk} is the number of images in cluster k that belong to class j.

Low-entropy clusters, i.e., clusters that contain mostly vectors from images of the same person, are preferable for retrieval tasks to high-entropy clusters, i.e., clusters that contain vectors from images that belong to several different persons. Therefore the aim is to learn a codebook $\mathbf{V}$ that minimize the total entropy of a cluster configuration, which is defined as:

$$E = \sum_{k=1}^{N_T} r_k E_k \tag{2}$$

where $r_k = n_k/N$ is the proportion of images in cluster k.

By substituting r_k and p_{jk} into the entropy definitions given in (1) and (2) the following objective function is obtained:

$$E = -\frac{1}{N} \sum_{k=1}^{N_T} \sum_{j=1}^{N_C} h_{jk} \log p_{jk} \tag{3}$$

However, it is not easy to directly optimize (3) as this function is not continuous with respect to $\mathbf{s}_i$. To this end, a continuous smooth approximation of the previously defined entropy is introduced. To distinguish the previous definition from the smooth entropy approximation, the former is called *hard entropy* and the latter is called *soft entropy*.

A smooth cluster membership vector $\mathbf{q}_i \in \mathbb{R}^{N_T}$ is defined for each histogram $\mathbf{s}_i$, where $q_{ik} = exp(\frac{-||\mathbf{s}_i - \mathbf{c}_k||_2}{m})$. The corresponding smooth l^1 normalized membership vector $\mathbf{w}_i$ is defined as $\mathbf{w}_i = \frac{\mathbf{q}_i}{||\mathbf{q}_i||_1} \in \mathbb{R}^{N_T}$. The parameter m controls the fuzziness of the assignment process: for $m \to 0$ each histogram is assigned to its nearest cluster, while larger values allow fuzzy membership.

Then, the quantities n_k and h_{jk} are redefined as: $n_k = \sum_{i=1}^{N} w_{ik}$ and $h_{jk} = \sum_{i=1}^{N} w_{ik}\pi_{ij}$, where π_{ij} is 1 if the i-th image belongs to class j and 0 otherwise. These modifications lead to a smooth entropy approximation that converges to hard entropy as $m \to 0$.

The derivative of E with respect to $\mathbf{v}_m$ can be calculated as the product of two other partial derivatives: $\frac{\partial E}{\partial \mathbf{v}_m} = \sum_{l=1}^{N} \sum_{\kappa=1}^{N_K} \frac{\partial E}{\partial s_{l\kappa}} \frac{\partial s_{l\kappa}}{\partial \mathbf{v}_m}$. In order to reduce the entropy in the histogram space the histograms $\mathbf{s}_l$ that, in turn, depend on the codebook $\mathbf{V}$, must be shifted. The partial derivative $\frac{\partial E}{\partial \mathbf{s}_l}$ provides the direction in which the histogram $\mathbf{s}_l$ must be moved. Since each codeword $\mathbf{v}_m$ lies in the feature space, the derivative $\frac{\partial s_{l\kappa}}{\partial \mathbf{v}_m}$ projects the previous direction into the codebook.

The calculation of these derivatives is straightforward and it is omitted due to space constraints. Note that the histogram space derivative does not exist when a histogram vector and a centroid vector coincide. The same holds for the feature space derivative when a codebook center and a feature vector also coincide. When that happens, the corresponding derivatives are set to 0.

During the optimization process the image histograms are updated. This might invalidate the initial choice of the centers $\mathbf{c}_k$ that should be also updated during the optimization. Therefore, the derivative of the objective function with respect to each $\mathbf{c}_k$, i.e., $\frac{\partial E}{\partial \mathbf{c}_m}$, is also calculated.

The codebook and the histogram centers can be optimized using gradient descent, i.e., $\Delta \mathbf{V} = -\eta \frac{\partial E}{\partial \mathbf{V}}$ and $\Delta \mathbf{c}_m = -\eta_c \frac{\partial E}{\partial \mathbf{c}_m}$, where η and η_c are the learning rates. In this work, the adaptive moment estimation algorithm, also known as Adam [5], is used instead of the simple gradient descent for the optimization, since it provides faster and more stable convergence. To avoid simply overfitting the histogram space centers, instead of learning the codebook, a smaller learning rate is used for the histogram space centers. For all the conducted experiments the optimization process runs for 100 iterations and the following learning rates are used: $\eta = 0.01$ and $\eta_c = 0.001$. The default parameters are used for the Adam algorithm ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$) [5]. Also, to reduce the training time, 100 features vectors are randomly sampled from each region during the training process. This reduces the training time, without affecting the quality of the learned codebook. The proposed approach can be also adapted to work in incremental mode by using small mini-batches of, e.g., 100 face images. However, this is not always necessary since in facial image retrieval the number of annotated training images is usually significantly smaller than the number of images that are to be encoded.

Regarding the initialization of the codebook and the histogram space centers several choices exist. In this work, the codebook is initialized using the k-means algorithm, as in the regular BoF model. For the histogram space centers, N_C centers are used (one for each person) and each center is initialized using the mean histogram of each person.

Finally, the softness parameters m and g must be selected. The parameter g controls the softness of the quantization process, while m controls the histogram's membership fuzziness and should be set to small enough value in order to closely approximate the hard entropy. It was experimentally established that the method is relatively stable with regard to the selected parameters: $m = 0.01$ and $g = 0.01$ is used for all the conducted experiments.

To better understand how the proposed method works, a toy example of the optimization process is provided. Four persons are chosen from the YouTube Faces dataset, which is described in Sect. 4, 30 images are selected for each person and 64 codewords are used for each of the four strips. The histogram space is visualized during the optimization process in Fig. 1. Since the histograms lie in a space with 4×64 dimensions, the PCA technique is used to visualize the resulting histograms. The expression and illumination variations in the face images lead to a representation where two or more subclasses exist for each person. Nonetheless, the optimization of the representation using the proposed O-SBoF technique successfully reduces the overlapping of histograms that belong to different classes and brings the histograms that belong to the same person significantly closer.

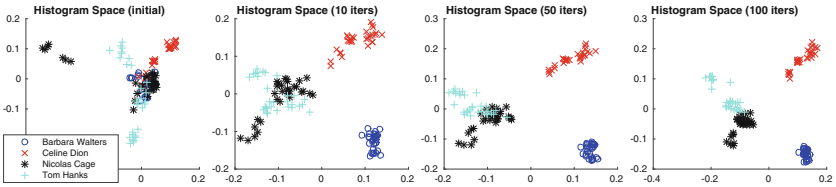

Fig. 1. Histogram space during the optimization using the proposed O-SBoF method

4 Experiments

4.1 Evaluation Setup

Two small-scale face recognition datasets, the ORL Database of Faces (ORL) [12], and the cropped variant of the Extended Yale Face Database B (Yale B) [1,7], and one large-scale face dataset, the YouTube Faces Database [14], were used for the evaluation of the proposed method.

The ORL dataset contains 400 images from 40 different persons (10 images per person) under varying pose and facial expression. The cropped Extended Yale Face Database B contains 2432 images from 38 individuals. The images were taken under greatly varying lighting conditions. The YouTube Faces dataset contains 621,126 frames of 3,425 videos and a total number of 1,595 individuals. The aligned version of this dataset is used, i.e., the face is already aligned in each image using face detection and alignment techniques. Before extracting the feature vectors each image is cropped by removing 25 % of its margins.

For the small-scale experiments each dataset is randomly split to two sets using half of the images of each person as the train set and the rest of them as the test set. The train set is used to build the database and train the O-SBoF model. The retrieval performance is evaluated using the images contained in the test set as queries. The training and the retrieval evaluation process are repeated five times and the mean and the standard deviation of the evaluated metrics are reported.

For the large-scale experiments a different evaluation strategy, similar to those of celebrity face image retrieval tasks [2], is used. The training set is formed by randomly selecting 100 images from the most popular persons that appear in the videos (5,900 training images are collected from the videos of the 59 most popular persons). A person is considered popular if it appears in at least 5 videos. To build the database, the images of persons that appear in at least 3 videos are used (the database contains 356,588 images from 226 persons). To evaluate the retrieval performance 100 queries from the popular persons (celebrities) are randomly selected. The evaluation process is repeated five times and the mean and the standard deviation of the evaluated metrics are reported.

Throughout this paper, two evaluation metrics are used: the interpolated precision (also abbreviated as 'prec.'), and the mean average precision (mAP). The mean average precision (AP) for a given query is computed at eleven equally

spaced recall points (0, 0.1, ..., 0.9, 1). A more detailed definition of the used evaluation metrics can be found in [10]. Also, for all the evaluated representations 16-bit floating numbers are used, since this can reduce the storage requirements without harming the retrieval precision.

4.2 Experimental Evaluation

First, the proposed method is evaluated using the ORL and the Yale B datasets. The proposed method is also compared to two other well-established face recognition features, the FPLBP and the TPLBP features (using the code provided by the authors of [15]), and two other SBoF representations with significantly larger codebooks. The results are shown in Table 1. The proposed method can greatly increase the precision over the baseline SBoF representations. For example, in the Yale B Dataset, the mAP and the top-5 precision increase by more than 20 % over the baseline. This allows to match the precision of the other representation techniques using 24 to 64 times smaller representation size. If a slightly larger codebook is used (64 codewords instead of 16), the proposed method greatly outperforms all the other evaluated methods, while still using smaller representation size than the other methods.

Next, the proposed method is evaluated using the large-scale YouTube Faces dataset. The results are shown in Table 2. The precomputed CSLBP and FPLBP features are used [14]. Again, the proposed method outperforms all the other evaluated methods using significantly smaller representation size, leading to better retrieval precision, faster image encoding and retrieval, and lower storage requirements. The O-SBoF method increase the mAP over the best performing SBOF method by 6 %, while reducing the representation size by 16 to 64 times. Regarding the (offline) training time of the proposed O-SBoF method, less than 3 h were required for learning 4×64 codewords. However, the proposed method can significantly reduce the online encoding time. The histogram encoding time was reduced from 2 days when using 4096 codewords (12 h for 1024 codewords) to 20 min when using 16 codewords (less than 1 h for 64 codewords). For all the conducted experiments a workstation with two 10-core 2.8 GHz CPUs was used.

Table 1. Small-scale evaluation using the ORL and the Extended Yale B datasets

Method	# codewords	# bytes	ORL Dataset		Yale B Dataset	
			mAP	top-5 prec.	mAP	top-5 prec.
FPLBP	-	896 / 3072	79.56 ± 0.41	70.52 ± 0.65	35.96 ± 0.25	80.03 ± 0.88
TPLBP	-	14336 / 49152	80.78 ± 0.72	71.63 ± 0.81	32.15 ± 0.24	77.21 ± 1.04
SBoF	4×1024	8192	93.29 ± 0.70	88.69 ± 0.70	25.78 ± 0.26	68.76 ± 1.05
SBoF	4×4096	32768	93.70 ± 0.63	89.37 ± 0.42	28.65 ± 0.30	73.10 ± 0.95
SBoF	4×16	128	81.69 ± 0.65	71.95 ± 0.40	18.24 ± 0.28	51.75 ± 1.01
O-SBoF	4×16	128	$\mathbf{93.24 \pm 1.52}$	$\mathbf{88.92 \pm 1.12}$	$\mathbf{37.72 \pm 0.90}$	$\mathbf{79.05 \pm 1.08}$
SBoF	4×64	512	88.90 ± 0.74	81.60 ± 0.94	20.89 ± 0.25	59.43 ± 0.93
O-SBoF	4×64	512	$\mathbf{97.42 \pm 0.72}$	$\mathbf{95.47 \pm 1.00}$	$\mathbf{43.13 \pm 0.64}$	$\mathbf{83.69 \pm 1.14}$

Table 2. Large-scale evaluation using the YouTube Faces dataset

Method	# codewords	# bytes	mAP	top-20 prec.	top-50 prec.	top-100 prec.
CSLBP	-	960	37.06 ± 1.21	98.23 ± 1.21	94.47 ± 2.34	87.93 ± 2.56
FPLBP	-	1120	37.90 ± 1.34	99.15 ± 0.57	96.74 ± 1.61	90.99 ± 2.35
SBoF	4×1024	8192	41.19 ± 1.04	99.99 ± 0.02	99.05 ± 0.26	91.50 ± 1.41
SBoF	4×4096	32768	40.95 ± 1.07	99.90 ± 0.12	98.54 ± 0.51	90.99 ± 1.48
SBoF	4×16	128	32.13 ± 1.35	98.93 ± 0.46	94.72 ± 1.25	84.47 ± 2.25
O-SBoF	4×16	128	**40.89 ± 1.31**	**99.71 ± 0.28**	**97.45 ± 0.70**	**88.78 ± 1.77**
SBoF	4×64	512	38.60 ± 1.32	99.84 ± 0.17	98.18 ± 0.43	89.55 ± 1.31
O-SBoF	4×64	512	**47.19 ± 1.24**	**99.93 ± 0.13**	**99.41 ± 0.31**	**92.17 ± 1.29**

Finally, to justify the choice of the spatial segmentation into 4 horizontal strips a set of experiments using different grid layouts was performed. The results are shown in Table 3. The 4×1 grid, i.e., using 4 horizontal strips, achieves the best precision in the ORL Dataset, while only slightly decreases the precision over the 4×4 grid in the Yale B Dataset. However, the 4×4 grid uses 16 codebooks instead of 4, quadrupling the size of the extracted histograms. Therefore, a 4×1 grid was used for all the conducted experiments using the SBoF and the O-SBoF methods.

Table 3. Effect of varying the grid layout to the mAP

Grid layout	1×1	4×1	1×4	4×2	2×4	4×4
ORL dataset	79.58 ± 0.65	**88.90 ± 0.74**	76.17 ± 0.82	85.75 ± 0.51	78.73 ± 0.44	81.34 ± 0.40
Yale B dataset	13.30 ± 0.16	20.89 ± 0.25	15.50 ± 0.09	20.11 ± 0.18	18.53 ± 0.16	**22.14 ± 0.22**

5 Conclusions

In this paper, a supervised codebook learning technique for face retrieval was presented. It was demonstrated using one large-scale dataset and two smaller datasets that the proposed technique can improve the retrieval precision and, at the same time, reduce the storage requirements and the retrieval time by almost two orders of magnitude.

There are several future research directions. In this work it was assumed that the face images were already aligned using face detection and alignment techniques and a fixed grid was used for the O-SBoF technique. However, the proposed method can be combined with facial feature detectors to more accurately define the regions used for feature extraction and further improve the retrieval precision and reduce the representation size. Furthermore, using more robust feature extractors is expected to improve the face recognition precision even more.

References

1. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 643–660 (2001)
2. Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J.: Ms-celeb-1m: challenge of recognizing one million celebrities in the real world. In: IS&T International Symposium on Electronic, Imaging (2016)
3. Iosifidis, A., Tefas, A., Pitas, I.: Discriminant bag of words based representation for human action recognition. Pattern Recogn. Lett. **49**, 185–192 (2014)
4. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. Int. J. Comput. Vis. **87**(3), 316–336 (2010)
5. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. Computing Research Repository, abs/1412.6980 (2014)
6. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2169–2178 (2006)
7. Lee, K., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 684–698 (2005)
8. Lian, X.-C., Li, Z., Lu, B.-L., Zhang, L.: Max-margin dictionary learning for multiclass image categorization. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 157–170. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15561-1_12
9. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the 7th IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157 (1999)
10. Passalis, N., Tefas, A.: Entropy optimized feature-based bag-of-words representation for information retrieval. IEEE Trans. Knowl. Data Eng. **28**, 1664–1677 (2016)
11. Passalis, N., Tefas, A.: Spectral clustering using optimized bag-of-features. In: Proceedings of the 9th Hellenic Conference on Artificial Intelligence, p. 19 (2016)
12. Samaria, F.S., Harter, A.C.: Parameterisation of a stochastic model for human face identification. In: Proceedings of the Second IEEE Workshop on Applications of Computer Vision, pp. 138–142 (1994)
13. Wang, C., Wang, Y., Zhang, Z.: Patch-based bag of features for face recognition in videos. Biometric Recogn. **7701**, 1–8 (2012)
14. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 529–534 (2011)
15. Wolf, L., Hassner, T., Taigman., Y.: Descriptor based methods in the wild. In: Real-Life Images workshop at the European Conference on Computer Vision (ECCV) (2008)
16. Wu, Z., Ke, Q., Sun, J., Shum, H.-Y.: Scalable face image retrieval with identity-based quantization and multireference reranking. IEEE Trans. Pattern Anal. Mach. Intell. **33**(10), 1991–2001 (2011)
17. Yang, S., Bebis, G., Chu, Y., Zhao, L.: Effective face recognition using bag of features with additive kernels. J. Electron. Imaging **25**(1), 013025 (2016)
18. Zhang, X., Gao, Y.: Face recognition across pose: a review. Pattern Recogn. **42**(11), 2876–2896 (2009)