Learning Neural Bag-of-Features for Large Scale Image Retrieval

Nikolaos Passalis and Anastasios Tefas

Abstract—In this paper, the well-known Bag-of-Features (BoF) model is generalized and formulated as a neural network that is composed of three layers: a Radial Basis Function (RBF) layer, an accumulation layer, and a fully connected layer. This formulation allows for decoupling the representation size from the number of used codewords, as well as for better modeling the feature distribution using a separate trainable scaling parameter for each RBF neuron. The resulting network, called Retrieval-oriented Neural BoF (RN-BoF), is trained using regular back propagation and allows for fast extraction of compact image representations. It is demonstrated that the RN-BoF model is capable of a) increasing the object encoding and retrieval speed, b) reducing the extracted representation size, and c) increasing the retrieval precision. A symmetry-aware spatial segmentation technique is also proposed to further reduce the encoding time and the storage requirements and allows the method to efficiently scale to large datasets. The proposed method is evaluated and compared to other state-of-the-art techniques using five different image datasets, including the large scale YouTube Faces Database.

Index Terms—Information Retrieval, Neural Networks, Retrieval-oriented Optimization, Bag-of-Features Representation.

1 Introduction

Large scale content-based information retrieval (CBIR) has recently received a lot of attention due to the exponential growth of multimedia data available over the Internet [1]. Retrieval tasks range from face image retrieval [2], [3], scene retrieval [4], and trademark retrieval [5], to generic visual retrieval [6], and recommendation systems [7], [8]. The enormous amounts of data push the classical information retrieval techniques to their limits. This led to the development of various encoding and approximate nearest neighbor search techniques to efficiently tackle the task of large scale information retrieval [9], [10], [11].

Among the most widely used techniques for content-based information retrieval is the Bag-of-Features (BoF) model, also known as Bag-of-Visual-Words (BoVW) or Bag-of-Words (BoW) [12], [13], [14]. The pipeline of the BoF model involves the following steps:

1) First, multiple feature vectors, such as SIFT descriptors [15], are extracted from an object, such as an image. These vectors define the *feature space*, where each object is represented as a set of feature vectors.
2) Then, a set of representative feature vectors, known as *codewords* or simply *words*, are learned. This set of vectors is called *codebook* (or *dictionary*) and this learning process is called *codebook/dictionary learning*.
3) Finally, each object is encoded by quantizing its feature vectors using the learned dictionary and a constant-length histogram is extracted for each object. These histograms can be used for the retrieval tasks and define the *histogram space*.

The application of the BoF model is not restricted to image representation, e.g., extracting representations from scene images [13], face images [16], [17], [18], [19] or gesture images [20], [21]. Several types of objects, such as video [14], audio [22], and time-series [23], can be also represented using the BoF model. However, the main focus of this work is learning compact image representations for retrieval tasks.

The quality of the BoF representation is highly dependent on the used dictionary learning algorithm. The first approaches used unsupervised clustering techniques, such as the k-means algorithm, to learn a dictionary [12], [13]. In these approaches the feature vectors are clustered and the centroids of the clusters are used to form the codebook. Then, each feature vector is represented by its nearest codeword. The codewords (centroids) are chosen in such way to minimize the reconstruction loss of the feature vectors that belong to the corresponding cluster. These methods allow for exploiting the available unsupervised information. However, they usually require very large codebooks in order to perform well leading to very large representations that cannot be directly used for retrieval tasks [9].

Although the previous unsupervised approaches have been successfully applied to a wide range of problems, it was established that minimizing the reconstruction loss is not optimal with respect to the final task [24], [25], [26], [27], [28]. Supervised dictionary learning not only increases the discriminative power of the learned dictionaries, but also allows for using significantly smaller dictionaries. These algorithms can be used for a variety of retrieval tasks where the queries are usually expected to be from a set of known classes, e.g., celebrity face image retrieval [2]. However, most supervised dictionary learning approaches learn discriminative dictionaries that are unable to extract meaningful representations for objects outside of their training domain. The interested reader is referred to [28], where it is demonstrated that highly discriminative representations can severely harm the retrieval precision for out-of-domain retrieval. This problem was addressed in [28] by defining

Nikolaos Passalis and Anastasios Tefas are with the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece. email: passalis@csd.auth.gr, tefas@aiia.csd.auth.gr

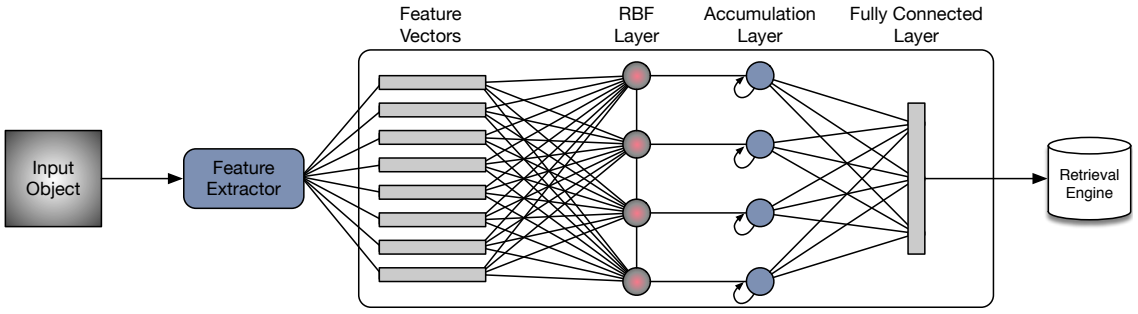

Fig. 1: The proposed RN-BoF model

an entropy-based loss function for optimizing the dictionary towards information retrieval instead of classification. Entropy loss follows the clustering hypothesis [29], which states that objects in the same cluster are likely to fulfill the same information need, and acts as a mild-discriminative criterion that increases the retrieval performing for both in-domain and (relevant) out-of-domain retrieval tasks.

Even these powerful task-oriented dictionary learning methods are not always enough to reduce the representation size to acceptable levels, especially when spatial segmentation techniques, such as Spatial Pyramid Matching [13], are used. The spatial segmentation techniques segment each image into a number of regions and use a separate dictionary for each region. Even for relative small dictionaries, e.g., 64 codewords, and pyramid level 2 that uses $8 + 4 + 1 = 13$ codebooks, the effective number of codewords builds up to $13 \times 64 = 832$. This limitation is intrinsic to the BoF model, since the size of the resulting representation depends on the number of used codewords, posing a barrier for scaling to large scale data.

In this paper, the BoF model is generalized and formulated as a neural network, that is composed of three layers (as shown in Figure 1): a *Radial Basis Function* (RBF) layer [30], [31], an accumulation layer, and a fully connected layer. This allows for overcoming the aforementioned limitation by decoupling the representation size from the number of used codewords. Also, each RBF neuron is equipped with a scaling parameter that allows for adjusting the shape of its Gaussian function to better fit the input distribution. This neural formulation, called Retrieval-oriented Neural BoF (RN-BoF), allows for significantly reducing the representation size, while still maintaining its representation power. To further reduce the encoding time and the storage requirements a symmetry-aware spatial segmentation technique is proposed. This technique exploits the fact that most images/visual objects are symmetric around their vertical axis and only uses horizontal segmentation. The RN-BoF model can be trained using the regular back-propagation technique and implemented using standard tools, such as the Theano library [32].

If trainable feature extractors are used, such as Convolutional Neural Networks (CNNs) [6], the gradients can also back-propagate, through the RN-BoF, to them (the extracted feature maps can be considered as slices of feature vectors that can be fed to the RN-BoF model). That way, the resulting deep architecture can be further fined-tuned towards extracting retrieval-oriented representations. Note that a similar setup is used in Section 4.4, where the proposed RN-BoF approach is evaluated using feature vectors extracted from a CNN. On the other hand, it should be stressed that the proposed approach can be also combined with fast hand-crafted feature extractors, such as SURF [33], to further increase the encoding speed. Furthermore, existing hashing and approximate nearest neighbor techniques can be combined with the proposed approach to increase the retrieval performance even more. The proposed RN-BoF formulation also allows for learning retrieval representations using Extreme Learning techniques, such as [34], and [35]. Finally, Relevance Feedback techniques [36], [37], [38], can be exploited to gather training data that can be used with the proposed method (especially when it is difficult to gather annotated data or concept drift issues exist).

The contributions of this paper are briefly summarized below. A neural network that learns to extract retrieval-oriented representations is proposed. This network generalizes the BoF model and allows for building powerful representation machines for image retrieval tasks that can rapidly encode the input images. The proposed technique is also combined with a symmetry-aware segmentation scheme to significantly reduce the encoding time and the representation size, allowing the method to efficiently scale to large datasets. The proposed technique is evaluated and compared to other state-of-the-art techniques using both small scale and large scale image datasets.

The rest of the paper is organized as follows. The related work is discussed in Section 2 and the proposed RN-BoF model is described in Section 3. The experimental evaluation of the proposed method is presented in Section 4. Finally, conclusions are drawn and possible extensions are discussed in Section 5.

2 Related Work

This work mainly concerns supervised dictionary learning for the BoF representation for which a rich literature exists. In [24], the proposed dictionary learning scheme tries to increase the mutual information between each codeword and the corresponding features labels. The CSMMI method [39], uses an information theory-based criterion for the optimization of the dictionary towards action and gesture recognition. In [25], multiple maximum margin hyperplanes are learned and at the same time the codebooks are adjusted to maximize the corresponding margins. This method requires

a quadratic number of codebooks with respect to the number of the training labels. This problem is addressed in later works, such as [40], where multi-class SVMs are used. A simple method for supervised codebook learning that incorporates both a traditional MLP layer and a codebook layer is proposed in [41], while a neural BoF formulation is provided in [42]. In [43], the optimization aims to minimize the logistic regression loss, while in [27], to minimize an LDA-based discriminative loss function. In [26], multiple dictionaries with complementary discriminative information are learned by adjusting the weights used during the clustering process using the predictions of a histogram-space classifier.

The aforementioned approaches focus on learning highly discriminative (classification-oriented) dictionaries that are not always optimal for retrieval tasks. This issue is discussed in detail in [28]. To this end, an entropy-based objective function was proposed for use either in the feature space [44], [45], or in the histogram space [19], [28]. These methods allow for optimizing the dictionary towards retrieval-oriented tasks, instead of learning a highly discriminative representation for classification tasks.

Although these retrieval-oriented dictionary learning approaches allow for using smaller representations [19], the extracted representation is still bound to the number of used codewords, leading to quite large representations (especially when combined with spatial segmentation techniques). To overcome this limitation, the method proposed in this paper is able to jointly learn the dictionary and a low-dimensional projection of the histogram representation allowing to significantly reduce the extracted representation size. Furthermore, the BoF model discards most of the spatial information contained in the original image, which can harm the retrieval precision. The BoF-based techniques for face recognition, e.g., [16], [17], [18], overcome this limitation by defining a grid over each image (or some regions of interest) and independently extracting a histogram from each cell of the grid. Object/scene recognition techniques also use similar spatial segmentation techniques, such as the Spatial Pyramid Matching (SPM) technique [13]. The method proposed in this paper exploits the symmetry that arises in most images to reduce the representation size and increase the retrieval precision. To the best of our knowledge, this is the first method that is able to jointly optimize multiple symmetry-aware codebooks and a final lower dimensional representation using a retrieval-oriented objective function.

Finally, it should be noted that hashing and approximate nearest neighbor search techniques have been also developed to tackle the task of large scale information retrieval [9], [10], [11]. The efficiency of these methods usually depends on the size of the extracted representation. Therefore, the proposed method can be also combined with the aforementioned methods, to further increase their performance and reduce the retrieval time.

3 Proposed Method

In this Section the proposed RN-BoF model is presented. First, the regular BoF model is briefly described. Then, the proposed neural extension of the BoF model, which is optimized towards information retrieval, and a symmetry-aware segmentation scheme are presented. Finally, a learning algorithm for the RN-BoF is proposed and the complexity of the proposed approach is discussed.

3.1 BoF Model

Let $\mathcal{X} = \{x_i\}_{i=1}^{N}$ be a set of N objects to be represented using the BoF model. As stated before, each object x_i consists of N_i feature vectors: $\mathbf{x}_{ij} \in \mathbb{R}^D$ $(j = 1...N_i)$, where D is the dimensionality of the extracted features. For example, in image retrieval each x_i would be an image and each $\mathbf{x}_{ij}$ a vector extracted from it, e.g., a SIFT feature vector. By quantizing the feature vectors of each object into a predefined number of histogram bins/codewords a fixed-length histogram can be extracted. When hard assignment is used each feature vector is quantized to its nearest codeword, while when soft assignment is utilized every feature contributes, by a different amount, to each histogram bin/codeword.

The set of all feature vectors, $\mathcal{S} = \{\mathbf{x}_{ij} | i = 1...N, j = 1...N_i\}$, is clustered into N_K clusters. Then, the 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. The codewords are used to quantize the feature vectors. In most cases only a subset of $\mathcal{S}$ is clustered, since this allows for reducing the training time without harming the retrieval precision. The dictionary/codebook is learned only once and then it can be used to represent any object.

To encode the i-th object the similarity between each feature vector $\mathbf{x}_{ij}$ and each codeword $\mathbf{v}_k$ is computed as:

$$[\mathbf{d}_{ij}]_k = exp(\frac{-||\mathbf{v}_k - \mathbf{x}_{ij}||_2}{\sigma}) \in \mathbb{R} \quad (1)$$

where the notation $[\mathbf{d}_{ij}]_k$ is used to denote the k-th element of the vector $\mathbf{d}_{ij}$. The parameter σ controls the quantization process: for harder assignment $\sigma << 1$ is used, while for softer assignment larger values are used. It is also common to use hard and non-continuous assignments [12], [13]:

$$[\mathbf{d}_{ij}]_k = \begin{cases} 1 & \text{if } k = \arg\min_{k'}(||\mathbf{v}_{k'} - \mathbf{x}_{ij}||_2) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The first definition in Equation (1) converges to the second in Equation (2) as $\sigma \to 0$. Many codebook learning algorithms, e.g., [23], [25], [27], [41], [44], [28], use soft-assignment techniques, similar to Equation (1), since this allows simple algorithms, such as gradient descent, to be used for the optimization. Then, the l^1 normalized membership vector of each feature vector $\mathbf{x}_{ij}$ is obtained:

$$\mathbf{u}_{ij} = \frac{\mathbf{d}_{ij}}{||\mathbf{d}_{ij}||_1} \in \mathbb{R}^{N_K} \quad (3)$$

This vector describes the similarity of the feature vector $\mathbf{x}_{ij}$ to each codeword. Finally, the histogram $\mathbf{s}_i$ is extracted for every object x_i:

$$\mathbf{s}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{u}_{ij} \in \mathbb{R}^{N_K} \quad (4)$$

The histogram $\mathbf{s}_i$ has unit l^1 norm, since $||\mathbf{u}_{ij}||_1 = 1$ for every j. These histograms describe each object and they can be used for the subsequent retrieval tasks. The training and the encoding process are fully unsupervised and no labeled data are required.

3.2 Retrieval optimized Neural BoF Model

3.2.1 Neural BoF

In this Section, a neural generalization of the BoF model is proposed. The proposed neural architecture is depicted in Figure 1. First, a feature extractor is used to extract multiple feature vectors from each object, as in the regular BoF model. The proposed network is composed of three layers: an RBF layer that measures the similarity of the input feature vectors to the RBF centers, an accumulation layer that builds a histogram of the input features and a fully connected layer that compiles the final representation for each object. The proposed architecture can be thought as a neural network that can be trained to extract object representations instead of performing classification tasks.

The output of the k-th RBF neuron is defined as:

$$[\boldsymbol{\phi}(\mathbf{x})]_k = \exp(-||(\mathbf{x}-\mathbf{v}_k)||_2/\sigma_k) \in \mathbb{R} \quad (5)$$

where $\mathbf{x}$ is a feature vector and $\mathbf{v}_k$ the center of the k-th RBF neuron. The RBF neurons behave similarly to the codewords in the BoF model since they are used to measure the similarity of the input vectors to a set of predefined vectors. Each RBF neuron is also equipped with a scaling factor $\sigma_k \in \mathbb{R}$ that adjusts the width of its Gaussian function and allows for better modeling the input feature distribution.

Similarly to the BoF model, which uses l^1 scaling in Equation (3), the used RBF architecture is normalized to ensure that the output of each RBF neuron is bounded. The output of the RBF neurons is re-defined as follows:

$$[\boldsymbol{\phi}(\mathbf{x})]_k = \frac{\exp(-||(\mathbf{x}-\mathbf{v}_k)||_2/\sigma_k)}{\sum_{m=1}^{N_K}\exp(-||(\mathbf{x}-\mathbf{v}_m)||_2/\sigma_m} \in \mathbb{R} \quad (6)$$

The output of the RBF neurons is accumulated in the next layer, similar to the BoF model (Equation (3):

$$\mathbf{s}_i = \frac{1}{N_i}\sum_{j=1}^{N_i}\boldsymbol{\phi}(\mathbf{x}_{ij}) \in \mathbb{R}^{N_K} \quad (7)$$

where $\boldsymbol{\phi}(\mathbf{x}) = ([\boldsymbol{\phi}(\mathbf{x})]_1, ..., [\boldsymbol{\phi}(\mathbf{x})]_{N_K})^T \in \mathbb{R}^{N_K}$ is the output vector of the RBF layer. Note that this behavior can be emulated in a neural network using a recurrent self loop and zeroing the memory of the neurons before feeding the feature vectors of a different object. The output of the accumulation layer remains normalized, i.e., it has unit l^1 norm.

After the histogram is compiled it is fed to the fully connected layer to obtain the lower dimensional final representation:

$$\mathbf{t}_i = relu(\mathbf{W}^T\mathbf{s}_i) \quad (8)$$

where $\mathbf{W} \in \mathbb{W}^{N_K \times N_R}$ is the weight matrix of fully connected layer, N_R is the length of the final representation and $relu(x) = \max(0, x)$ is the rectifier activation function which is applied element-wise. Note that the relu activation function also ensures that the final representation will be non-negative.

3.2.2 Retrieval Optimization

The centers for the RBF neurons can be initialized using the k-means algorithm (as in the regular BoF model), while the final fully connected layer can be randomly initialized (performing random projections of the histogram representation [46]). However, the proposed model can be further trained to extract representations oriented towards specific information retrieval tasks. To this end, a retrieval-oriented loss function is used: the supervised entropy loss [28].

Let $l_i \in \{1, ..., N_C\}$ be the label of the i-th annotated object used for learning the representation, where N_C is the number of training classes. To measure the entropy in the representation space, the $\mathbf{t}_i$ vectors 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:

$$E_k = -\sum_{j=1}^{N_C} p_{jk}\log p_{jk} \quad (9)$$

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

Each centroid can be considered as a representative query for which the representation is optimized. According to the cluster hypothesis low-entropy clusters, i.e., clusters that contain mostly vectors from objects of the same class, are preferable for retrieval tasks to high-entropy clusters, i.e., clusters that contain vectors from objects that belong to several different classes. Therefore, the network is optimized to minimize the total entropy of a cluster configuration, which is defined as:

$$E = \sum_{k=1}^{N_T} r_k E_k \quad (10)$$

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

By substituting r_k and p_{jk} into the entropy definition given in (10) the following objective function is obtained:

$$E = -\frac{1}{N}\sum_{k=1}^{N_T}\sum_{j=1}^{N_C} h_{jk}\log\frac{h_{jk}}{n_k} \quad (11)$$

Note that this objective function is not continuous with respect to $\mathbf{t}_i$. Therefore, it cannot be directly used for learning the parameters of the network. To this end, a smooth cluster membership vector $\mathbf{q}_i \in \mathbb{R}^{N_T}$ is defined for each vectors $\mathbf{t}_i$, where $[\mathbf{q}_i]_k = exp(\frac{-||\mathbf{t}_i - \mathbf{c}_k||_2}{m})$. The corresponding smooth l^1 normalized membership vector is defined as $\mathbf{w}_i$ 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 for fuzzy membership.

Then, the quantities n_k and h_{jk} are redefined as $n_k = \sum_{i=1}^{N}[\mathbf{w}_i]_k$ and $h_{jk} = \sum_{i=1}^{N}[\mathbf{w}_i]_k\pi_{ij}$, where π_{ij} is 1 if the i-th object belongs to class j and 0 otherwise. Substituting these values into the objective function (11) leads to a smooth entropy approximation that converges to (hard) entropy as $m \to 0$.

The parameters of the network can be learned using simple gradient descent:

$$\Delta(\mathbf{V}, \boldsymbol{\sigma}, \mathbf{W}) = -\eta(\frac{\partial E}{\partial \mathbf{V}}, \frac{\partial E}{\partial \boldsymbol{\sigma}}, \frac{\partial E}{\partial \mathbf{W}}) \quad (12)$$

where η is the learning rate and $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, ..., \sigma_{N_K})$. Instead of using simple gradient descent, the Adam algorithm is utilized [47]. The Adam algorithm computes adaptive learning rates for each of the optimization parameters using estimates of the first and second moments of the gradient. The default parameters for the decay rate of the first and second order estimates, i.e., $\beta_1 = 0.9$ and $\beta_2 = 0.999$, are used and a small value $\epsilon = 10^{-8}$ is utilized to ensure numerical stability. Also, note that as the network learns the final representation the initial choice for the representative queries $\mathbf{c}_k$ might be no longer valid. Thus, to allow the loss function to "follow" the representation, the entropy centers are also adjusted during the learning process:

$$\Delta\mathbf{C} = -\eta_C \frac{\partial E}{\partial \mathbf{C}} \quad (13)$$

where $\mathbf{C} = [\mathbf{c}_1\mathbf{c}_2...\mathbf{c}_{N_T}]$. Typically, the learning rate for the entropy centers η_C is set to a lower value than the main learning rate η, since the focus of the method is learning an optimal representation, instead of learning the optimal entropy centers. The gradients $\frac{\partial E}{\partial \mathbf{V}}$, $\frac{\partial E}{\partial \boldsymbol{\sigma}}$, $\frac{\partial E}{\partial \mathbf{W}}$ and $\frac{\partial E}{\partial \mathbf{C}}$ are derived in detail in Appendix A. Finally, note that even though calculating the entropy requires the whole training set to be used, the network can be also trained using mini-batches and calculating the entropy for each mini-batch.

3.2.3 Spatial RBoF

Spatial segmentation schemes, such as Spatial Pyramid Matching [13], are usually used with the BoF model to further increase the image recognition accuracy. The proposed Neural BoF model can be also combined with a spatial segmentation scheme. A separate set of RBF neurons and accumulation neurons are used to encode the features extracted from each separate region of the image. To further reduce the storage requirements and the encoding time a symmetry-aware spatial segmentation scheme is proposed. Instead of using a simple grid over each image, N_S horizontal strips are used, as shown in Figure 2. The total number of RBF neurons used in the spatial RN-BoF model is $N_S \times N_K$, since N_K RBF neurons are used for each region. This exploits the symmetry around the vertical axis that is usually present in most visual objects and scenes. In Section 4, it is demonstrated that this scheme improves both the retrieval accuracy and the encoding time/storage requirements over the more computational demanding segmentation schemes that are usually used.

3.3 Learning with RBoF

The complete learning algorithm is presented in Figure 3. First, the RBF centers are initialized using k-means over the training features (line 1). If Spatial BoF is used, the RBF centers are initialized using feature vectors from the corresponding regions. The weights $\mathbf{W}$ are randomly initialized using a uniform distribution over [-1, 1], while the scaling factors σ_k are initially set to 0.1. The parameter m controls the entropy fuzziness and should be set to small enough

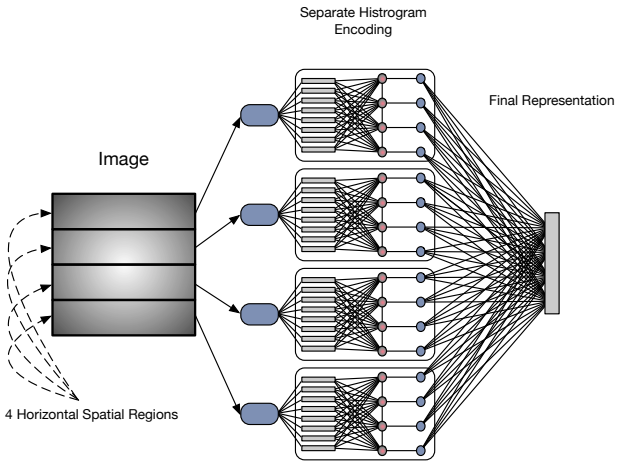

Fig. 2: Symmetry-aware spatial segmentation technique for the RN-BoF

value in order to closely approximate the hard entropy. A value of $m = 0.01$ is used for all the experiments in this paper. Next, the objects are encoded (line 3) and the entropy centers are chosen (lines 4-7). Each class must be represented by at least one cluster, i.e., at least N_C centers must be used. In this paper the class mean vector is used as the corresponding entropy center. If the distribution of some classes is multi-modal, more centers may be selected by running k-means over each class. Finally, the parameters of the network are learned using the Adam algorithm (lines 8-9) using the following learning rates: $\eta = 0.01$ and $\eta_C = 0.001$. For all conducted experiments $N_{iters} = 100$ optimization iterations are used. To accelerate the learning process a random subsample of 100 feature vectors are used for each iteration instead of feeding all the feature vectors extracted from an object. This allows for decreasing the time needed for learning the parameters of the network with little effect on quality of the learned representation.

To better understand how the proposed method works a simple toy example using data from the 15-scene dataset is provided [13]. Four classes (suburb, store, office and forest) are used and 50 images are randomly sampled from each class. A spatial segmentation scheme with 4 horizontal strips is used, and $N_k = 32$ RBF neurons are used for each strip. The length of the final representation is set to $N_R = 64$. The resulting histograms are projected into a 2-d space using PCA. Figure 4 shows the first two principal components of the extracted vectors during the optimization process. It is evident that the proposed RN-BoF technique successfully lowers the entropy in the final space by learning a representation that gathers the objects in pure clusters. Note that the method prevents collapsing the representation into a few distinct points, maintaining its representation ability.

3.4 Complexity Analysis

In Table 1 the encoding complexity and the storage requirements of the proposed RN-BoF method is compared to three other methods: the regular BoF model, the EO-BoW model [28], which is a retrieval-oriented dictionary

Fig. 3: RN-BoF Learning Algorithm

Input: A set $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ of N training objects and their class labels $\mathcal{L} = \{l_1, ..., l_N\}$
Parameters: N_K, N_S, N_R, N_{iters}
Output: The optimized network $(\mathbf{V}, \boldsymbol{\sigma}, \mathbf{W})$

procedure RN-BOF LEARNING
  Initialize V by running k-means on the extracted training feature vectors
  $T \leftarrow$ ENCODE(X)
  $C \leftarrow []$
  for $i \leftarrow 1; i \leq N_C; i++$ **do**
    Calculate and mean vector over the objects that belong to class i
    Append the new center to C
  for $i \leftarrow 1; i \leq N_{iters}; i++$ **do**
    Apply the Adam algorithm to update the parameters of the network using equations (12) and (13)
procedure ENCODE(X) **return** the object representation according to (8)

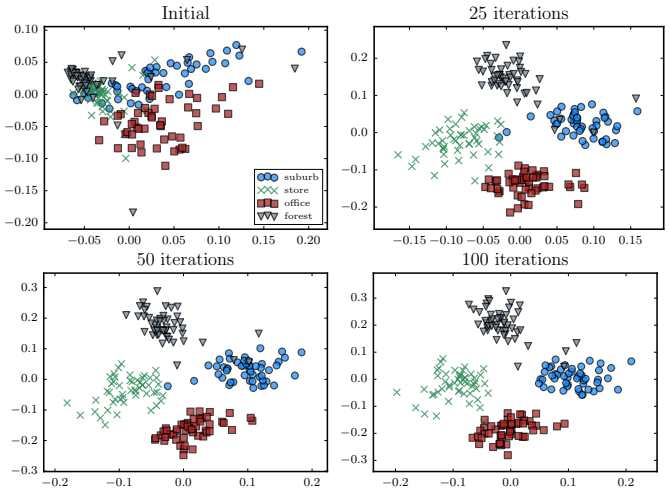

Fig. 4: Toy example of the learning process using 4 classes of the 15-scene dataset

learning method for the BoF model, and the VLAD model [48], which is a state-of-the-art representation technique. Note that the value of N_R can be independently adjusted to fit the available storage capabilities and does not depend on the number of used codewords. This allows the proposed RN-BoF method to use significantly smaller representations, while still performing better than the competitive technique, as shown in the next Section. On the other hand, the proposed method requires slightly more encoding time, since the extracted histogram must be projected into a lower dimensional space. However, in almost all the cases is expected that $N_R < D$, which means that all the methods have the same asymptotic complexity ($O(N_S N_K D)$) when the same number of codewords are used.

TABLE 1: Comparing representation size and encoding time for different methods

Method	BoF/EO-BoW	VLAD	RN-BoF
Repr. size	$O(N_S N_K)$	$O(N_S N_K D)$	$O(N_R)$
Enc. time	$O(N_S N_K D)$	$O(N_S N_K D)$	$O(N_S N_K (D + N_R))$

4 EXPERIMENTS

In this section, the proposed method is evaluated using five image datasets from various domains and it is compared to other state-of-the-art representation methods, including a recently proposed retrieval-oriented dictionary learning method for the BoF model. First, the used datasets, the evaluation metrics and the other evaluated methods are briefly described. In the next five subsections the methods are evaluated on five datasets. Finally, the effect of parameter selection on the proposed method is evaluated.

4.1 Evaluation Setup

4.1.1 Datasets

The proposed method is evaluated using five datasets, the ORL Database of Faces (ORL) [49], the cropped variant of the Extended Yale Face Database B (Yale B) [50], [51], the 15-scene dataset [13], the MIT indoor scene dataset (MIT67) [52], and the large scale YouTube Faces Database [53].

The ORL dataset [49], is a face image dataset that contains 400 images from 40 different persons. The dataset contains 10 images for each person under varying pose and facial expression. The cropped Extended Yale Face Database B is also a face image dataset that contains 2432 images, taken under greatly varying lighting conditions, of 38 different persons. From each image SIFT descriptors of 16 x 16 patches were sampled over a grid with spacing of 4 pixels (using the code supplied by the authors of [54]). For both datasets half of the images for each subject were used to build the database and optimize the network, while the rest of them were used to query the database and evaluate the retrieval accuracy. This process has been repeated five times and the mean and the standard deviation of the evaluated metrics are reported.

The 15-scene dataset [13], contains 15 different scene categories: *office*, *kitchen*, *living room*, *bedroom*, *store*, *industrial*, *tall building*, *inside city*, *street*, *highway*, *coast*, *open country*, *mountain*, *forest*, and *suburb*. The dataset contains 4,485 images and each category has 200 to 400 images. Each image is resized to 250x250 pixels and SIFT descriptors of 16 x 16 patches are densely sampled over a grid with spacing of 8 pixels. The standard evaluation procedure was used: 100 images were sampled from each class to build the database and optimize the network and the rest of them (2985 images) were used to evaluate the retrieval accuracy. Again, the experiments were repeated five times and the mean and the standard deviation of each metric is reported.

The MIT indoor scene dataset [52], is a larger scene recognition dataset that contains 67 different indoor scene categories, such as *living room*, *grocery store*, *church*, *library*, etc., and 15,620 images. In contrast to the 15-scene dataset, that contains both indoor and outdoor scenes, the MIT67 dataset contains only indoor scenes, with some of them

being characterized by their global spatial properties, while other by the objects they contain. The methods are evaluated on a more realistic scenario, where only a small subset of the database has been annotated. To this end, 14950 images are used to build the database, while only 50 images are annotated for each of the 67 classes (3350 images). The rest 670 images (10 from each class) are used to query the database. The training and the evaluation processes are repeated five times using random splits. Each image is resized to 250x250 pixels and SIFT descriptors (16x16 patches, 8 pixels grid spacing) are extracted.

The YouTube Faces dataset [53], contains frames from videos depicting 1,595 different individuals. The dataset is composed of 621,126 frames of 3,425 videos. The face is already aligned in each frame using face detection and alignment techniques, in the used cropped variant of the dataset. Before extracting the feature vectors each image is resized to 250x250 pixels and cropped by removing 25% of its margins. SIFT descriptors of 16 x 16 patches are densely sampled over a grid with spacing of 8 pixels. An evaluation strategy, similar to those of celebrity face image retrieval tasks [2], is used. The persons that appear in more than 5 videos are considered popular (celebrities). The training set is formed by randomly selecting 100 images for each of the most popular persons (5,900 training images are collected from the videos of the 59 most popular persons). The database contains the images of persons that appear in at least 4 videos, i.e., 197,557 images from 226 persons. The retrieval performance is evaluated using 100 randomly selected celebrity queries, i.e., frames that contains popular persons. The evaluation process is repeated five times.

4.1.2 Evaluation Metrics

Three retrieval evaluation metrics are used (similarly to [28]): precision, recall and mean average precision (mAP). To retrieve the relevant objects nearest neighbor search is used [36]. The precision is defined as $Pr(q,k) = \frac{rel(q,k)}{k}$, where k is the number of retrieved objects and $rel(q,k)$ is the number of retrieved objects that belong to the same class as the query q. The recall is similarly defined as $Rec(q,k) = \frac{rel(q,k)}{nclass(q)}$, where $nclass(q)$ is the total number of database objects that belong to the same class as q. The interpolated precision, $Pr_{interp}(q,k) = \max_{k',\, k' \geq k} Pr(q,k')$, is utilized, since it is preferred over the raw precision as it reduces the precision-recall curve fluctuation [36]. The average precision (AP) is computed for a given query at eleven equally spaced recall points (0, 0.1, ..., 0.9, 1) and the mean average precision is calculated as the mean of APs for all queries. Two types of curves are plotted: the precision-recall curve and the precision-scope curve. The scope refers to the number of objects returned to the user and the precision-scope curve allow us to evaluate the precision at lower recall levels. For storing the extracted feature vectors in the database half precision (16-bit) floating point numbers are used for all the conducted experiments.

4.1.3 Baseline and Competitive State-of-the-Art Methods

To learn a baseline BoF dictionary 50,000 feature vectors are randomly sampled and the k-means algorithm is used. The clustering process is repeated 5 times and the codebook that yields the lowest reconstruction error is kept. The proposed method is compared to the VLAD method, which is among the state-of-the-art representation techniques for image retrieval [48]. Furthermore, the proposed method is compared to the FPLBP and the TPLBP features (using the code provided by the authors of [55]) that are two other well-established global features. For the YouTube Faces dataset the precomputed CSLBP and FPLBP features were used [53]. Finally, the proposed method is compared to the EO-BoW method [28], which is a recently proposed method for retrieval optimization of the BoF representation ($m = 0.01$ and $\sigma = 0.1$ are used). Four horizontal spatial regions are used for the BoF method, the EO-BoW method and the proposed RN-BoF method. For both the EO-BoW and the RN-BoF methods the spatial dictionaries are jointly optimized.

TABLE 2: ORL Evaluation Results

Method	N_K	Representation Size	mAP (%)
FPLBP	-	448 (896 bytes)	79.69 ± 0.95
TPLBP	-	7168 (14336 bytes)	80.74 ± 0.66
VLAD	16	2048 (4096 bytes)	88.59 ± 0.67
VLAD	64	8192 (16384 bytes)	91.28 ± 0.47
BoF	16	64 (128 bytes)	81.44 ± 1.00
BoF	64	256 (512 bytes)	88.94 ± 0.61
BoF	1024	4096 (8192 bytes)	93.15 ± 0.53
BoF	4096	16384 (32768 bytes)	93.45 ± 0.53
EO-BoW	16	64 (128 bytes)	95.63 ± 0.67
EO-BoW	32	128 (256 bytes)	96.57 ± 0.40
RN-BoF	8	8 (16 bytes)	80.00 ± 1.70
RN-BoF	16	16 (32 bytes)	93.48 ± 0.78
RN-BoF	64	32 (64 bytes)	$\mathbf{97.87 \pm 1.00}$

4.2 ORL Evaluation

The evaluation results for the ORL dataset are shown in Table 2. The proposed RN-BoF method achieves higher mAP than the simple BoF method for any number codewords, while reducing the representation size by three orders of magnitude. Also, the RN-BoF method performs better than the FPLBP/TPBLP and the VLAD methods increasing the retrieval precision and reducing the representation size. As it was already mentioned, the proposed RN-BoF technique allows for decoupling the representation size from the number of used codewords. Thus, significant smaller representations than the competitive EO-BoW method can be used. For this dataset the proposed method increases the retrieval precision, while using 4 times smaller representation than the competitive EO-BoW method.

It should be noted that the ORL dataset is relatively small and the supervised methods can easily overfit the representation. This is especially true for the proposed RN-BoF method, since it uses a set of scaling parameters to adjust the width of the RBFs as well as a trainable fully connected layer. Nonetheless, it manages to achieve better retrieval precision than the EO-BoW method. This can be also attributed to the choice of entropy as the objective function, since entropy is strongly regularized. That way, it prevents the collapse of the objects into a few points, allowing the method to maintain its representation ability.

The precision-recall and precision-scope curves for the best performing methods are shown in Figure 5a and 5e

TABLE 3: Yale B Evaluation Results

Method	N_K	Representation Size	mAP (%)
FPLBP	-	1536 (3072 bytes)	35.78 ± 0.37
TPLBP	-	24576 (49152 bytes)	31.88 ± 0.20
VLAD	16	2048 (4096 bytes)	17.24 ± 0.07
VLAD	64	8192 (16384 bytes)	21.89 ± 0.13
BoF	16	64 (128 bytes)	17.78 ± 0.21
BoF	64	256 (512 bytes)	20.68 ± 0.16
BoF	1024	4096 (8192 bytes)	25.49 ± 0.18
BoF	4096	16384 (32768 bytes)	28.25 ± 0.12
EO-BoW	16	64 (128 bytes)	36.88 ± 0.70
EO-BoW	32	128 (256 bytes)	36.48 ± 0.97
RN-BoF	8	8 (16 bytes)	36.93 ± 2.25
RN-BoF	16	16 (32 bytes)	52.04 ± 1.81
RN-BoF	64	32 (64 bytes)	$\mathbf{76.70 \pm 1.79}$

respectively. Again, the proposed RN-BoF method outperforms all the other evaluated methods, while using smaller representations.

4.3 Yale B Evaluation

The Yale B is evaluated using a setup similar to this of the ORL dataset. The results are shown in Table 3. Again, the RN-BoF significantly outperforms all the other evaluated methods increasing the mAP by more than 14% using a 16-dimensional (32-byte) representation. This improvement can be attributed to the greater discriminative capacity of the fully connected layer used in the RN-BoF. This can be also confirmed in the precision-scope curves of Figure 5f. Furthermore, the RN-BoF can match the retrieval precision of all the other methods using just 16 bytes for each image. The precision-recall and precision-scope curves are shown in Figures 5b and 5f respectively. Again, the RN-BoF method greatly outperform all the other evaluated methods.

4.4 15-scene Evaluation

The evaluation results for the 15-scene dataset are shown in Table 4. The retrieval precision for the BoF representation peaks when 1024 codewords per region are used leading to a mAP of 28.21%. Using more codewords slightly reduces the retrieval precision. The RN-BoF method outperforms all the other evaluated methods while using a 16-dimensional representation. This reduces the representation size from 8 times (when compared to the EO-BoW method) to 256 times (when compared to the BoF method). The same behavior is also observed in the precision-recall and precision-scope curves of Figures 5c and 5g respectively, since the proposed RN-BoF method leads to better retrieval precision at any recall/scope level, while using smaller representations.

As it was already mentioned in Section 1, the proposed approach can be combined with trainable feature extractors instead of using hand-crafted feature extractors (e.g., SIFT). The evaluation results using feature vectors extracted from the last convolutional layer of a pre-trained VGG-16 network [4], are shown in Table 5. The CNN was trained using the Places365 dataset [4]. As before, the RN-BoF method leads to significant precision improvements, while reducing the size of the extracted representation over all the other evaluated techniques. Also, using convolutional features allows for achieving higher precision than using SIFT features (Table 4). Fine-tuning the convolutional layers by back-propagating the gradients from the RN-BoF layer is expected to further increase the retrieval precision.

TABLE 4: 15-scene Evaluation Results

Method	N_K	Representation Size	mAP (%)
FPLBP	-	2816 (5632 bytes)	16.92 ± 0.13
TPLBP	-	45056 (90112 bytes)	29.71 ± 0.10
VLAD	16	2048 (4096 bytes)	27.60 ± 0.18
VLAD	64	8192 (16384 bytes)	28.07 ± 0.36
BoF	16	64 (128 bytes)	26.78 ± 0.29
BoF	64	256 (512 bytes)	27.85 ± 0.19
BoF	1024	4096 (8192 bytes)	28.21 ± 0.21
BoF	4096	16384 (32768 bytes)	27.26 ± 0.17
EO-BoW	16	64 (128 bytes)	35.79 ± 0.22
EO-BoW	32	128 (256 bytes)	36.83 ± 0.46
RN-BoF	8	8 (16 bytes)	35.63 ± 1.52
RN-BoF	16	16 (32 bytes)	41.32 ± 0.72
RN-BoF	64	32 (64 bytes)	$\mathbf{46.71 \pm 0.39}$

TABLE 5: 15-scene Evaluation Results (CNN features)

Method	N_K	Representation Size	mAP (%)
BoF	16	64 (128 bytes)	35.73 ± 0.91
BoF	1024	4096 (8192 bytes)	32.88 ± 0.67
EO-BoW	16	64 (128 bytes)	38.76 ± 0.63
RN-BoF	16	16 (32 bytes)	$\mathbf{65.12 \pm 0.79}$

4.5 MIT67 Evaluation

The evaluation results for the MIT67 are shown in Table 6. Similarly to the 15-scene dataset the mAP peaks for the BoF representation when 1024 codewords are used. The RN-BoF outperforms all the other evaluated methods using smaller representations. The representation size is reduced 4 times (when the method is compared to the EO-BoW) to 128 times (when compared to the BoF method). The same behavior is observed in the precision-recall and precision-scope curves shown in Figures 5d and 5h. Although the differences between the EO-BoW and the RN-BoF methods are clear in the precision-recall curves, they achieve almost the same precision for the top-20 results. However the RN-BoF method uses smaller representations than the EO-BoW method.

TABLE 6: MIT67 Evaluation Results

Method	N_K	Representation Size	mAP (%)
FPLBP	-	2816 (5632 bytes)	3.14 ± 0.06
TPLBP	-	45056 (90112 bytes)	5.01 ± 0.08
VLAD	16	2048 (4096 bytes)	5.80 ± 0.03
VLAD	64	8192 (16384 bytes)	5.96 ± 0.09
BoF	16	64 (128 bytes)	4.89 ± 0.11
BoF	64	256 (512 bytes)	5.75 ± 0.14
BoF	1024	4096 (81 bytes)	6.04 ± 0.23
BoF	4096	16384 (32768 bytes)	5.49 ± 0.13
EO-BoW	16	64 (128 bytes)	6.46 ± 0.13
EO-BoW	32	128 (256 bytes)	6.63 ± 0.12
EO-BoW	64	256 (512 bytes)	6.64 ± 0.21
RN-BoF	64	32 (64 bytes)	6.42 ± 0.17
RN-BoF	64	64 (128 bytes)	6.80 ± 0.22
RN-BoF	64	128 (256 bytes)	$\mathbf{6.94 \pm 0.24}$

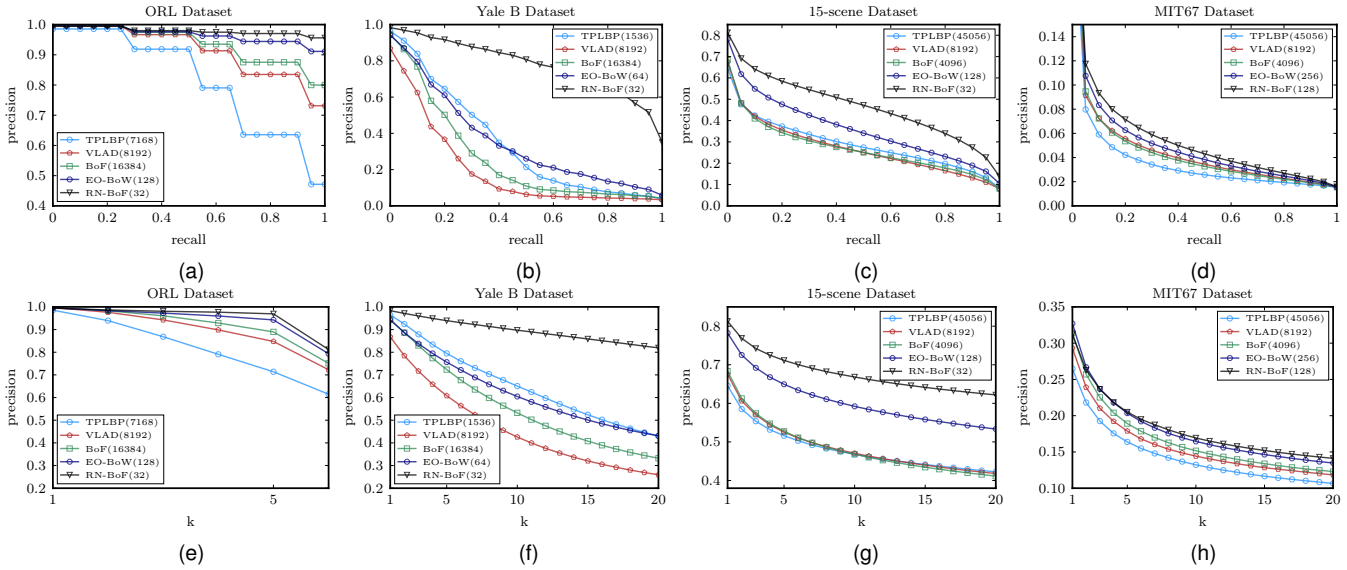

Fig. 5: Precision-recall and precision-scope curves for the ORL, Yale B, 15-scene and MIT67 datasets

4.6 YouTube Faces Evaluation

The evaluation results for the YouTube Faces dataset are shown in Table 7. The train time refers to the time needed for learning the dictionary using k-means and the time spent for the supervised optimization of the representation (only for the EO-BoW and the RN-BoF methods). The encoding time refers to the mean time needed for encoding one image using the learned dictionary, while the retrieval time to the mean time spend for querying the database. Since the CSLBP and the FPLBP descriptors are pre-calculated no train and encoding time are reported [53]. A six-core workstation was used for all the conducted experiments. Although the train/query times are reported only for this dataset they follow the same trend for the other datasets when the methods are used with the same setup, i.e., the same number of codewords and representation size.

Several interesting conclusions can be drawn from the results reported in Table 7. First, the training time is highly dependent on the number of used codewords. When more than 1024 codewords are used learning an unsupervised dictionary using k-means requires more time than both initializing and optimizing a smaller supervised dictionary that performs better than the larger unsupervised dictionary. Therefore, even though the supervised optimization requires an extra offline training step it can reduce the total training time by using smaller codebooks. The encoding time also depends on the size of the used codebook, although for codebooks smaller than 64 the differences are very small. The same is also true for the retrieval time when the representation size is kept under a certain threshold (1000 bytes). However, for larger codebooks and larger representations there is a measurable performance penalty. Note that for retrieving the relevant images simple sequential scan was used. All the methods can be combined with approximate nearest neighbors search and hashing techniques to further reduce the retrieval time.

Regarding the retrieval precision, the RN-BoF significantly outperforms all the other method, achieving a mAP of 46.58%, using 8 times smaller representation than the second best performing method (EO-BoW). Note that a relatively small number of SIFT feature vectors are extracted from each image due to resizing each image to 250x250 pixels and then cropping the 25% of its margins. This seems to affect only the BoF method that overfits the representation when more than 64x4=256 codewords are used. If more computational resources are available, then more feature vectors can be extracted from each image improving the performance of all the evaluated methods.

The precision-recall and the precision-scope curves are shown in Figure 6, where an interesting phenomenon is observed. Although the proposed RN-BoF method achieves the highest precision for any recall level greater than 0.15, there are other methods that achieve slightly higher precision for the first 200 results. This behavior can be explained if the nature of the used dataset is considered. The Youtube Faces dataset contains a large number of similar frames for each person, since multiple sequential frames were extracted from each video. If a method captures irrelevant frame information, e.g., the background, it will manage to achieve very high precision for the first results, since all the frames of the corresponding video will be retrieved. However, it will fail to retrieve frames from other videos where the same person appears, since it encoded only the information that relates each frame to its video. The RN-BoF method achieves slightly lower precision for the top results, but manages to retrieve different videos in which each person appears. This also highlights the differences between a retrieval-oriented and a classification-oriented representation (the top-1 precision is actually the 1-nn classification accuracy). Therefore, a representation might be able to achieve high classification accuracy, but fail to retrieve enough objects that belong to the same class.

4.7 Parameter Evaluation

The proposed method is relatively stable with regard to its hyperparameters. The scaling factors for the RBF neurons

TABLE 7: YouTube Faces Evaluation Results

Method	N_K	Representation Size	Train Time	Encoding Time	Retrieval Time	mAP (%)
CSLBP	-	480 (960 bytes)	N/A	N/A	9.7s	40.26 ± 1.06
FPLBP	-	560 (1120 bytes)	N/A	N/A	9.9 s	37.96 ± 0.69
VLAD	16	2048 (4096 bytes)	0.8 m	0.023 s	10.4 s	37.36 ± 1.14
VLAD	64	8192 (16384 bytes)	1.4 m	0.024 s	11.0 s	39.31 ± 1.24
BoF	16	64 (128 bytes)	2.3 m	0.017 s	9.8 s	37.31 ± 1.22
BoF	64	256 (512 bytes)	5.6 s	0.021 s	9.8 s	40.49 ± 1.00
BoF	1024	4096 (81 bytes)	58.0 m	0.059 s	10.3 s	39.09 ± 0.93
BoF	4096	16384 (32768 bytes)	136.6 m	0.223 s	12.0 s	38.45 ± 0.88
EO-BoW	16	64 (128 bytes)	2.4 m + 9.8 m	0.017 s	9.8 s	42.81 ± 1.07
EO-BoW	32	128 (256 bytes)	3.4 m + 15.7 m	0.018 s	9.8 s	42.57 ± 1.57
EO-BoW	64	256 (512 bytes)	5.5 m + 44.0 m	0.020 s	9.8 s	42.92 ± 0.99
RN-BoF	16	16 (32 bytes)	2.4 m + 8.5 m	0.017 s	9.8 s	37.05 ± 1.63
RN-BoF	64	32 (64 bytes)	5.5 m + 43.7 m	0.020 s	9.8 s	$\mathbf{46.58 \pm 2.21}$

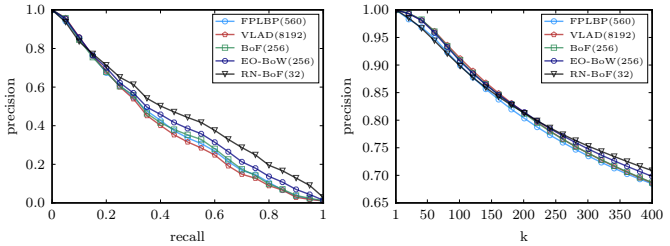

Fig. 6: Precision-recall and precision-scope curves for the YouTube Faces dataset

are learned using back-propagation, while the entropy parameter m is set to 0.01. For all the conducted experiments in this Subsection, three small-scale datasets (ORL, Yale B and 15-scene) are used. n Figure 8 the effect of the used spatial segmentation scheme is evaluated using the regular BoF model . The grid layout refers to the number of horizontal and vertical regions used for the segmentation of each image (as described in Section 3 and shown in Figure 2). Using schemes that are symmetrical around the vertical axis, i.e., 4x1 and 4x2, increases the retrieval precision over the corresponding schemes are symmetrical around the horizontal axis, i.e., 1x4 and 2x4. This is expected since the vertical axis carries more useful spatial information than the horizontal one. For example, the sky is almost always expected to be on the upper part of image, while a tree might be in any of the left/center/right parts of an image. Thus, not using vertical segmentation is expected to increase the spatial invariance of the method (e.g., a tree can be located in any vertical region). Indeed, for both the ORL and the 15-scene datasets, using only horizontal segmentation yields the best retrieval precision. For the Yale B dataset, the 4x4 grid leads to slightly better retrieval precision than the 4x1 grid. This can be attributed to the fact that non-symmetries in a face, e.g., a mole in the left part of the face, can sometimes help increase the retrieval precision.

5 Conclusions

In this paper the well-known BoF model was generalized and a neural extension, called RN-BoF, capable of working efficiently in a large scale setting, was proposed. The RN-BoF method is able to decouple the final representation size from the number of used codewords and allows for better modeling the input distribution using a separate trainable scaling factor for each RBF neuron. The RN-BoF can be trained with regular back-propagation and combined with retrieval-oriented loss functions, such as entropy [28]. Through extensive experiments on five datasets it was demonstrated that RN-BoF is able to a) increase the object encoding and retrieval speed, b) reduce the extracted representation size, and c) increase the retrieval precision over the competitive state-of-the-art methods. The proposed method is also combined with a symmetry-aware spatial segmentation scheme to further reduce the encoding time and the storage requirements.

The proposed method provides a neural framework that can be easily extended. First, the proposed approach can be combined with Convolutional Neural Networks (CNNs) [6]. That way, the gradients can back-propagate to the convolutional layers, allowing the resulting deep architecture to be further fined-tuned toward extracting retrieval-oriented representations. Also, existing hashing and approximate nearest neighbor techniques can be combined with the proposed approach to increase retrieval performance even more [9], [10]. Furthermore, a trainable hashing layer [11], can be used after the fully connected layer to jointly optimize both the RN-BoF and the hashing layer using the entropy objective. Finally, note that the proposed neural formulation allows the RN-BoF model to be combined with Extreme Learning techniques, e.g., [34], [35], to reduce the training time of the proposed algorithm.

Appendix A
RN-BoF Derivatives

The RN-BoF derivatives used for the gradient descent algorithm are derived in this Section. The required derivatives are calculated as follows:

$$\frac{\partial E}{\partial [\mathbf{W}]_{i\kappa}} = \sum_{l=1}^{N} \sum_{\kappa=1}^{N_R} \frac{\partial E}{\partial [\mathbf{t}_l]_\kappa} \frac{\partial [\mathbf{t}_l]_\kappa}{\partial [\mathbf{W}]_{i\kappa}} \tag{14}$$

$$\frac{\partial E}{\partial \mathbf{v}_m} = \sum_{l=1}^{N} \sum_{\kappa=1}^{N_R} \sum_{\mu=1}^{N_K} \frac{\partial E}{\partial [\mathbf{t}_l]_\kappa} \frac{\partial [\mathbf{t}_l]_\kappa}{\partial [\mathbf{s}_l]_\mu} \frac{\partial [\mathbf{s}_l]_\mu}{\partial \mathbf{v}_m} \tag{15}$$

$$\frac{\partial E}{\partial \sigma_m} = \sum_{l=1}^{N} \sum_{\kappa=1}^{N_R} \sum_{\mu=1}^{N_K} \frac{\partial E}{\partial [\mathbf{t}_l]_\kappa} \frac{\partial [\mathbf{t}_l]_\kappa}{\partial [\mathbf{s}_l]_\mu} \frac{\partial [\mathbf{s}_l]_\mu}{\partial \sigma_m} \tag{16}$$

TABLE 8: Grid Layout

Dataset	1 x 1 grid	4 x 1 grid	1 x 4 grid	4 x 2 grid	2 x 4 grid	2 x 2 grid	4 x 4 grid
ORL dataset	79.52 ± 1.27	$\mathbf{88.52 \pm 0.44}$	76.21 ± 0.82	85.02 ± 0.46	78.46 ± 0.60	82.58 ± 0.98	80.60 ± 0.85
Yale B dataset	13.03 ± 0.12	20.63 ± 0.10	15.16 ± 0.09	19.78 ± 0.16	18.23 ± 0.20	16.09 ± 0.12	$\mathbf{21.86 \pm 0.11}$
15-scene dataset	24.86 ± 0.24	$\mathbf{26.36 \pm 0.17}$	24.70 ± 0.19	25.43 ± 0.17	24.88 ± 0.21	24.85 ± 0.29	25.34 ± 0.19

Easily it can derived (see [28]) that $\frac{\partial E}{\partial [\mathbf{t}_l]_\kappa} = -\frac{1}{N}\sum_{k=1}^{N_T}\sum_{j=1}^{N_C} \log p_{jk}\pi_{lj}\frac{\partial [\mathbf{w}_l]_k}{\partial [\mathbf{t}_l]_\kappa}$, where $\frac{\partial [\mathbf{w}_l]_k}{\partial [\mathbf{t}_l]_\kappa} = -\frac{[\mathbf{w}_l]_k}{m}\left(\frac{[\mathbf{t}_l]_\kappa - [\mathbf{c}_k]_\kappa}{||\mathbf{t}_l - \mathbf{c}_k||_2} - \sum_{k'=1}^{N_T}[\mathbf{w}_l]_{k'}\frac{[\mathbf{t}_l]_\kappa - [\mathbf{c}_{k'}]_\kappa}{||\mathbf{t}_l - \mathbf{c}_{k'}||_2}\right)$.

Note that when a cluster center ($\mathbf{c}_k$) and an object ($\mathbf{t}_l$) coincide this derivative does not exist. When that happens, the corresponding derivatives are zeroed. This is also true the derivatives calculated in Equations (15) and (21). The rest of the derivatives are derived as:

$$\frac{\partial [\mathbf{t}_l]_\kappa}{\partial [\mathbf{W}]_{i\kappa}} = r([\mathbf{W}^T]_\kappa^T \mathbf{s}_l)[\mathbf{s}_l]_i, \; \frac{\partial [\mathbf{t}_l]_\kappa}{\partial [\mathbf{s}_l]_\mu} = r([\mathbf{W}^T]_\kappa^T \mathbf{s}_l)[\mathbf{W}]_{\mu\kappa} \tag{17}$$

where $r(x)$ is 0 if $x \leq 0$ and 1 otherwise.

Similar to the BoW definition, the following quantity is defined to simplify the calculations:

$$[\mathbf{d}_{ij}]_k = exp(-||(\mathbf{x}_{ij} - \mathbf{v}_k)||_2/\sigma_k) \tag{18}$$

Using equation (18), the output of the RBF layer is expressed as $[\boldsymbol{\phi}(\mathbf{x}_{ij})]_k = \frac{[\mathbf{d}_{ij}]_k}{||\mathbf{d}_{ij}||_1}$. The derivatives are calculated as: $\frac{\partial [\mathbf{s}_l]_\mu}{\partial \mathbf{v}_m} = \frac{1}{N_l}\sum_{j=1}^{N_l}\frac{\partial [\boldsymbol{\phi}(\mathbf{x}_{lj})]_\mu}{\partial \mathbf{v}_m} = \frac{1}{N_l}\sum_{j=1}^{N_l}\frac{\partial [\boldsymbol{\phi}(\mathbf{x}_{lj})]_\mu}{\partial [\mathbf{d}_{lj}]_m}\frac{\partial [\mathbf{d}_{lj}]_m}{\partial \mathbf{v}_m}$, where N_l is the number of feature vectors extracted from the l-th object, $\frac{\partial [\boldsymbol{\phi}(\mathbf{x}_{lj})]_\mu}{\partial [\mathbf{d}_{lj}]_m} = \frac{\delta_{\mu m}}{||\mathbf{d}_{lj}||_1} - \frac{[\mathbf{d}_{lj}]_\mu}{||\mathbf{d}_{lj}||_1^2}$ and $\frac{\partial [\mathbf{d}_{lj}]_m}{\partial \mathbf{v}_m} = \frac{[\mathbf{d}_{lj}]_m}{\sigma_m}\frac{\mathbf{x}_{lj} - \mathbf{v}_m}{||(\mathbf{x}_{lj} - \mathbf{v}_m)||_2}$. The Kronecker delta function used in the equations above is defined as $\delta_{km} = 1$, if $m = k$, and 0 otherwise.

Similarly, for the derivative of the scaling factors:

$$\frac{\partial [\mathbf{s}_l]_\mu}{\partial \sigma_m} = \frac{1}{N_l}\sum_{j=1}^{N_l}\frac{\partial [\boldsymbol{\phi}(\mathbf{x}_{lj})]_\mu}{\partial [\mathbf{d}_{lj}]_m}\frac{\partial [\mathbf{d}_{lj}]_m}{\partial \sigma_m} \tag{19}$$

where $\frac{\partial [\mathbf{d}_{lj}]_m}{\partial \sigma_m} = \frac{[\mathbf{d}_{lj}]_m}{\sigma_m^2}||(\mathbf{x}_{lj} - \mathbf{v}_m)||_2$. Finally, the derivative for adjusting the entropy centers is calculated as:

$$\frac{\partial E}{\partial \mathbf{c}_m} = -\frac{1}{N}\sum_{k=1}^{N_T}\sum_{j=1}^{N_C}\log\frac{h_{jk}}{n_k}\frac{\partial h_{jk}}{\partial \mathbf{c}_m} \tag{20}$$

where

$$\frac{\partial h_{jk}}{\partial \mathbf{c}_m} = \frac{1}{m}\sum_{i=1}^{N}\pi_{ij}[\mathbf{w}_i]_m(\delta_{km} - [\mathbf{w}_i]_k)\frac{\mathbf{t}_i - \mathbf{c}_m}{||\mathbf{t}_i - \mathbf{c}_m||_2} \tag{21}$$

REFERENCES

[1] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 2, no. 1, pp. 1–19, 2006.

[2] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," *arXiv e-prints*, vol. abs/1607.08221, 2016.

[3] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 529–534.

[4] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proceedings of the Advances in Neural Information Processing Systems, pages=487–495, year=2014.*

[5] F. M. Anuar, R. Setchi, and Y.-K. Lai, "Semantic retrieval of trademarks based on conceptual similarity," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 2, pp. 220–233, 2016.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[7] D. Rafailidis and P. Daras, "The tfc model: Tensor factorization and tag clustering for item recommendation in social tagging systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 3, pp. 673–688, 2013.

[8] N. Zheng, S. Song, and H. Bao, "A temporal-topic model for friend recommendations in chinese microblogging systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 9, pp. 1245–1253, 2015.

[9] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 316–336, 2010.

[10] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009, pp. 2130–2137.

[11] J. Wang, W. Liu, S. Kumar, and S.-F. Chang, "Learning to hash for indexing big data - a survey," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 34–57, 2016.

[12] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2003, pp. 1470–1477.

[13] S. Lazebnik, C. Schmid, and J. Ponce, "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, 2006, pp. 2169–2178.

[14] Y.-G. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proceedings of the ACM international conference on Image and Video Retrieval*, 2007, pp. 494–501.

[15] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[16] Z. Wu, Q. Ke, J. Sun, and H.-Y. Shum, "Scalable face image retrieval with identity-based quantization and multireference reranking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1991–2001, 2011.

[17] C. Wang, Y. Wang, and Z. Zhang, "Patch-based bag of features for face recognition in videos," in *Biometric Recognition*, 2012, vol. 7701, pp. 1–8.

[18] S. Yang, G. Bebis, Y. Chu, and L. Zhao, "Effective face recognition using bag of features with additive kernels," *Journal of Electronic Imaging*, vol. 25, no. 1, p. 013025, 2016.

[19] N. Passalis and A. Tefas, "Spatial bag of features learning for large scale face image retrieval," in *Proceedings of the INNS Conference on Big Data*, 2016.

[20] J. Wan, G. Guo, and S. Z. Li, "Explore efficient local features from RGB-D data for one-shot learning gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1626–1639, 2016.

[21] J. Wan, Q. Ruan, W. Li, and S. Deng, "One-shot learning gesture recognition from RGB-D data using bag of features." *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2549–2582, 2013.

[22] M. Riley, E. Heinen, and J. Ghosh, "A text retrieval approach to content-based audio retrieval," in *Proceedings of the International Conference on Music Information*, 2008, pp. 295–300.

[23] A. Iosifidis, A. Tefas, and I. Pitas, "Multidimensional sequence classification based on fuzzy distances and discriminant analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2564–2575, 2013.

[24] S. Lazebnik and M. Raginsky, "Supervised learning of quantizer codebooks by information loss minimization," *IEEE Transactions*

on Pattern Analysis and Machine Intelligence, vol. 31, no. 7, pp. 1294–1309, 2009.

[25] X.-C. Lian, Z. Li, B.-L. Lu, and L. Zhang, "Max-margin dictionary learning for multiclass image categorization," in *Proceedings of the European Conference on Computer Vision*, 2010, pp. 157–170.

[26] W. Zhang, A. Surve, X. Fern, and T. Dietterich, "Learning non-redundant codebooks for classifying complex objects," in *Proceedings of the Annual International Conference on Machine Learning*, 2009, pp. 1241–1248.

[27] A. Iosifidis, A. Tefas, and I. Pitas, "Discriminant bag of words based representation for human action recognition," *Pattern Recognition Letters*, vol. 49, pp. 185–192, 2014.

[28] N. Passalis and A. Tefas, "Entropy optimized feature-based bag-of-words representation for information retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1664–1677, 2016.

[29] E. M. Voorhees, "The cluster hypothesis revisited," in *Proceedings of the annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, 1985, pp. 188–196.

[30] T. Li, S. Duan, J. Liu, L. Wang, and T. Huang, "A spintronic memristor-based neural network with radial basis function for robotic manipulator control implementation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 4, pp. 582–588, 2016.

[31] K. Z. Mao and G. B. Huang, "Neuron selection for rbf neural network classifier based on data structure preserving criterion," *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1531–1540, 2005.

[32] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, 2016.

[33] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[34] G. B. Huang and C. K. Siew, "Extreme learning machine with randomly assigned rbf kernels," *International Journal of Information Technology*, vol. 11, no. 1, pp. 16–24.

[35] L. L. C. Kasun, Y. Yang, G. B. Huang, and Z. Zhang, "Dimension reduction with extreme learning machine," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3906–3918, Aug 2016.

[36] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2008, vol. 1.

[37] L. Zhang, Z. Wang, T. Mei, and D. D. Feng, "A scalable approach for content-based image retrieval in peer-to-peer networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 858–872, April 2016.

[38] M. Tzelepi and A. Tefas, "Relevance feedback in deep convolutional neural networks for content based image retrieval," in *Proceedings of the 9th Hellenic Conference on Artificial Intelligence, SETN 2016, Thessaloniki, Greece, May 18-20, 2016*, 2016, pp. 27:1–27:7.

[39] J. Wan, V. Athitsos, P. Jangyodsuk, H. J. Escalante, Q. Ruan, and I. Guyon, "CSMMI: class-specific maximization of mutual information for action and gesture recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 3152–3165, 2014.

[40] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu, "Max-margin multiple-instance dictionary learning," in *Proceedings of the International Conference on Machine Learning*, 2013, pp. 846–854.

[41] M. Jiu, C. Wolf, C. Garcia, and A. Baskurt, "Supervised learning and codebook optimization for bag-of-words models," *Cognitive Computation*, vol. 4, no. 4, pp. 409–419, 2012.

[42] N. Passalis and A. Tefas, "Neural bag-of-features learning," *Pattern Recognition*, vol. 64, no. C, pp. 277–294, Apr. 2017.

[43] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2559–2566.

[44] Y. Kuang, M. Byröd, and K. Åström, "Supervised feature quantization with entropy optimization," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 1386–1393.

[45] Y. Kuang, K. Åström, L. Kopp, M. Oskarsson, and M. Byröd, "Optimizing visual vocabularies using soft assignment entropies," in *Proceedings of the Asian Conference on Computer Vision*, 2011, pp. 255–268.

[46] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 245–250.

[47] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv e-prints*, vol. abs/1412.6980, 2014.

[48] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311.

[49] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 1994, pp. 138–142.

[50] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[51] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[52] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 413–420.

[53] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 529–534.

[54] Y. Jia and T. Darrell, "Heavy-tailed distances for gradient based image descriptors," in *Advances in Neural Information Processing Systems*, 2011, pp. 397–405.

[55] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *Real-Life Images workshop at the European Conference on Computer Vision*, 2008.

Nikolaos Passalis obtained his B.Sc. in informatics in 2013 and his M.Sc. in information systems in 2015 from Aristotle University of Thessaloniki, Greece. He is currently pursuing his Ph.D. studies in the Artificial Intelligence & Information Analysis Laboratory in the Department of Informatics at the University of Thessaloniki. His research interests include machine learning, computational intelligence and information retrieval.

Anastasios Tefas received the B.Sc. in informatics in 1997 and the Ph.D. degree in informatics in 2002, both from the Aristotle University of Thessaloniki, Greece. Since 2013 he has been an Assistant Professor at the Department of Informatics, Aristotle University of Thessaloniki. From 2008 to 2012, he was a Lecturer at the same University. From 2006 to 2008, he was an Assistant Professor at the Department of Information Management, Technological Institute of Kavala. From 2003 to 2004, he was a temporary lecturer in the Department of Informatics, University of Thessaloniki. From 1997 to 2002, he was a researcher and teaching assistant in the Department of Informatics, University of Thessaloniki. Dr. Tefas participated in 15 research projects financed by national and European funds. He has co-authored 69 journal papers, 177 papers in international conferences and contributed 8 chapters to edited books in his area of expertise. Over 3250 citations have been recorded to his publications and his H-index is 29 according to Google scholar. His current research interests include computational intelligence, pattern recognition, statistical machine learning, digital signal and image analysis and retrieval and computer vision.