Compact Global Descriptors for Visual Search

Vijay Chandrasekhar[1], Jie Lin[1] , Olivier Morère[1,2,3], Antoine Veillard[2,3], Hanlin Goh[1,3]
[1]*Institute for Infocomm Research, Singapore*
[2] *Université Pierre et Marie Curie, Paris, France*
[3] *Image and Pervasive Access Laboratory, UMI CNRS 2955, Singapore*

Abstract

The first step in an image retrieval pipeline consists of comparing global descriptors from a large database to find a short list of candidate matching images. The more compact the global descriptor, the faster the descriptors can be compared for matching. State-of-the-art global descriptors based on Fisher Vectors are represented with tens of thousands of floating point numbers. While there is significant work on compression of local descriptors, there is relatively little work on compression of high dimensional Fisher Vectors. We study the problem of global descriptor compression in the context of image retrieval, focusing on extremely compact binary representations: 64-1024 bits. Motivated by the remarkable success of deep neural networks in recent literature, we propose a compression scheme based on deeply stacked Restricted Boltzmann Machines (SRBM), which learn lower dimensional non-linear subspaces on which the data lie. We provide a thorough evaluation of several state-of-the-art compression schemes based on PCA, Locality Sensitive Hashing, Product Quantization and greedy bit selection, and show that the proposed compression scheme outperforms all existing schemes.

1. Introduction

For mobile visual search and augmented reality applications, the size of data sent over the network needs to be as small as possible to reduce latency and improve user experience. One approach to the problem is to transmit JPEG compressed images or MPEG compressed videos over the network, but this might be prohibitively expensive at low uplink speeds. An alternate approach to sending JPEG images or MPEG videos is to extract feature descriptors on the mobile device, compress the descriptors and transmit them over the network. Such an approach has been demonstrated to reduce the amount of transmission data by orders of magnitude for both visual search and augmented reality applications [1], [2]. To this end, MPEG is close to completion of a standard titled Compact Descriptors for Visual Search (CDVS) for descriptor extraction and compression, and is embarking on extending the CDVS standard to video sources, titled Compact Descriptors for Video Analysis (CDVA).

State-of-the-art content-based image retrieval pipelines consist of two blocks: (1) retrieving a subset of images from the database that are similar, and (2) using Geometric Consistency Checks (GCC) (e.g., based on RANSAC) for finding relevant database images with high precision. The GCC step is computationally complex and can only be performed on a small number of images (tens to hundreds). As a result, the first step of the pipeline is critical to achieving high recall. For the first step, state-of-the-art schemes are based on comparing global representations of images. The *global descriptor* of an image is represented by a single high dimensional vector with tens of thousands of dimensions. Examples of global descriptors include VLAD [3], Fisher Vector [4], Residual Enhanced Visual Vector(REVV) [5], the Scalable Fisher Compressed Vector

(SFCV) [6], and the recently proposed descriptor based on Convolutional Neural Networks [7]–[9]. Subsequently, *local descriptors* like SIFT, SURF, CHoG are used in the GCC step to check if a valid geometric transform exists between database and query images.

Over the course of the MPEG-CDVS standard, there was significant work on compression of *local* descriptors. Schemes based on Lattice coding [1], Transform Coding [10], Product Vector Quantization [11] and Locality Sensitive Hashing [12] were proposed. However, there was relatively little work on compression of *global* descriptors, and studying the trade-offs between descriptor size and retrieval performance.

The problem of *global* descriptor compression is an important one. The more compact the *global* descriptor, the faster the descriptors can be compared for matching. Further, it is highly desirable that the global descriptors be binary to enable fast matching through Hamming distances. With internet-scale image databases, like the recently released Yahoo 100M image database [13], compact global descriptors will be key to fast web-scale image-retrieval. Ideally, a 32-bit or 64-bit binary global descriptor is highly desirable, as it can be directly addressed in RAM. However, finding such a representation is extremely challenging, as uncompressed Fisher Vectors are stored as 8192 to 65536 floating point numbers.

2. Related Work and Contributions

The Fisher Vector (or its variants) are obtained by quantizing the set of local feature descriptors with a small codebook (64-512 centroids), and aggregating first and second order residual statistics for features quantized to each centroid. The residual statistics from each centroid are concatenated together to obtain the high-dimensional global descriptor representation, typically 8192 to 65536 dimensions. The performance increases as the dimensionality of the global descriptor increases, as shown in [4]. There is relatively little work on compression of high-dimensional Fisher Vectors.

Perronnin et al. [4] propose ternary quantization of Fisher Vectors, where component dimensions are quantized to -1 or 1 based on their sign, and codewords that are not visited are quantized to 0. The authors show that this representation results in little loss in performance - however, this results in descriptor size of thousands of bits. Perronnin et al. also explore Locality Sensitive Hashing [12] and Spectral Hashing [14] for compressing Fisher Vectors. Spectral Hashing performs poorly at high rates, and LSH needs thousands of bits to achieve good performance. Yunchao et al. propose the popular Iterative Quantization Scheme (ITQ) scheme [15]. The authors first perform PCA and then learn a rotation to minimize the quantization error of mapping the transformed data to the vertices of a zero-centered binary hypercube. Yunchao et al. in [16] show how random bilinear projections can be used to create binary hashes, using significantly less memory than PCA projection matrices. However, the Bilinear Projection-based Binary Codes (BPBC) scheme proposed in [16] requires tens of thousands of bits to match the performance of the uncompressed descriptor. Jegou et al. propose PCA followed by random rotations and Product Quantization (PQ) for obtaining compact representations [17]. While this results in highly compact descriptors, the resulting representation is not binary and cannot be compared with Hamming distances. The MPEG-CDVS standard adopted the Scalable Fisher Compressed Vector [6], which was based on binarization of high-dimensional Fisher Vectors. The size of the compressed descriptor in the MPEG-CDVS standard ranges from 256 bytes to several thousand bytes per image, based on the operating point. Finally, we note that there is plenty of work on similarity preserving binary codes [18] - most of it is focused on descriptors like SIFT or GIST [19], while the data in consideration in this work are two to three orders of magnitude higher in dimensionality, making the problem much more challenging.

In this paper, we study the problem of *global* descriptor compression, focusing on the trade-off between descriptor size and retrieval performance. Our starting representation is the uncompressed

Fisher Vector representation adopted in the MPEG-CDVS standard [6]. Motivated by the remarkable success of neural networks for large-scale image classification in recent literature [9], [20], we propose a scheme based on stacked Restricted Boltzmann Machines (RBM) for computing binary representations. We propose training a multi-layer neural network to learn low dimensional non-linear subspaces on which the data lie. We perform a thorough survey and evaluation of popular hashing and compression schemes for high dimensional Fisher vectors, and show that the proposed scheme outperforms several state-of-the-art schemes.

The outline of the paper is as follows. In Section 3, we describe the evaluation framework used in the paper. In Section 4, we describe the propose scheme based on RBMs. Finally, in Section 5, we present detailed pairwise matching and image retrieval experimental results in the CDVS framework.

3. Evaluation Framework

For evaluation, we perform both pairwise matching and retrieval experiments. For pairwise matching experiments, our evaluation of compressed global descriptors is similar to [5]. We use the full set of matching image pairs from all 5 CDVS data sets: *Graphics*, *Paintings*, *Video Frames*, *Buildings* and *Common Objects*. In total, there are 16,319 matching image pairs, and generate random non-matching pairs. We present full Receiver Operating Characteristic (ROC) curves, and Area Under Curve (AUC) for global descriptors of different size. For large-scale retrieval experiments, we use the *CDVS Buildings*, *CDVS Objects*, and the *Holidays* data sets, combined with the 1 million MIR-FLICKR distractor data set [21].

Most schemes, including our proposed scheme, require a training step. We use the ImageNet dataset for training, which consists of 1 million images from a 1000 different image categories [22]. We choose a completely different data set for training to ensure that there is no overfitting while testing.

For the global descriptor, we extract a Fisher Vector (FV), starting from SIFT features obtained from Difference-of-Gaussian (DoG) interest points. We use PCA to reduce dimensionality of the SIFT descriptor from 128 to 64 dimensions, which has shown to improve performance [3]. We use a Gaussian Mixture Model (GMM) with 128 centroids, resulting in 8192 dimensions each for first and second order statistics. Only the first-order statistics are retained in the global descriptor representation, as second-order statistics only results in a small improvement in performance [6]. The Fisher vector is L_2 normalized to unit-norm, after signed power normalization.

4. Stacked Restricted Boltzmann Machines (SRBM)

We propose a binary encoding scheme for high-dimensional vectors, based on Stacked Restricted Boltzmann Machines (SRBM) [23], [24]. In their seminal work, Hinton and Salakhutdinov [23] showed how high dimensional vectors can be converted to low-dimensional codes by training multi-layer neural networks, which can perform significantly better than PCA for dimensionality reduction.

In this work, we are motivated by the remarkable performance of deep neural network based schemes for large-scale image classification [9], [20], and large-scale image-retrieval [8] in recent literature. While neural networks have been around for several decades, their resurgence can be attributed to two key factors: availability of large training data sets, and large amounts of computing, which makes training large and deep networks possible. E.g., the neural networks in [9] and [20] have 7 and 16 layers respectively, and take multiple weeks to train with millions of images, and GPU clusters. The multi-layer or "deep" neural networks find compact low-dimensional non-linear subspaces on which the data lie and can be linearly classified. Also, we note that descriptor

components of SIFT and Fisher Vectors are known to have highly non-Gaussian statistics, and applying a single PCA transform can in-fact hurt compression performance at high rates, as shown in [10]. We start by discussing Restricted Boltzmann machines (RBMs), which are a generic framework for learning non-linear subspaces, and then describe how we adapt them to our problem.

An RBM is an undirected bipartite graphical model consisting of a layer of visible (input) units $\mathbf{v}$ and a layer of hidden (output) units $\mathbf{h}$. A set of symmetric weights $\mathbf{W}$ connects $\mathbf{v}$ and $\mathbf{h}$. For an RBM with binary visible and hidden units, the joint set of visible and hidden units has an energy function given by:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_i b_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i h_i w_{ij} \tag{1}$$

where v_i and h_j are the binary states of visible and hidden units i and j respectively, w_{ij} are the weights connecting the units, and b_i and b_j are their respective bias terms. Using the energy function in Equation (1), a probability can be assigned to $\mathbf{v}$ as follows:

$$p(\mathbf{v}) = \sum_{\mathbf{h}} \frac{\exp(-E(\mathbf{v}, \mathbf{h}))}{\sum_{\mathbf{u},\mathbf{g}} E(\mathbf{u}, \mathbf{g})} \tag{2}$$

The activation probabilities of units in one layer can be sampled by fixing the states of the other layer as follows:

$$p(h_j = 1|\mathbf{v}) = \sigma(b_j + \sum_i w_{ij} v_i) \tag{3}$$

$$p(v_i = 1|\mathbf{h}) = \sigma(b_i + \sum_j w_{ij} h_j) \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function. RBMs can be trained by minimizing the contrastive divergence objective [25], which approximates the maximum likelihood of the input distribution. Alternating Gibbs sampling based on Equations (3) and (4) is used to obtain the network states to update the parameters w_{ij}, b_i, b_j through gradient descent.

Similar to [23], we propose stacking multiple RBMs (SRBM) to create a deep network with several layers. Each layer captures higher order correlations between the units of the previous layer in the network. An SRBM model with three stacked RBMs is illustrated in Figure 1. With multi-layer networks, each bit in the output layer is obtained by a composite of several non-linear functions, thus, modeling highly complex non-linear subspaces in which the input data lie. We discuss the important details of how we adapt the model to our problem. There are several design choices in the training of SRBMs, such as the number of layers, the activation function, learning rates, frequency of parameter updates, types of visible and hidden units, etc.

- **Binary visible units.** The input dimensions of Fisher Vectors are represented as floating point numbers. We note that binary RBMs are a lot faster and easier to train than continuous RBMs [26]. In [4], the authors show that binarization of Fisher Vectors results in negligible loss in performance. As a result, we set both input and output units to be binary. We compute the mean of the Fisher Vectors from the train data and perform component-wise binarization using the pre-computed thresholds. Note that setting the threshold to 0 for binarization leads to a large drop in performance, due to the structure of the Fisher Vector data, as several sub-blocks within a vector are typically 0, corresponding to centroids that are not visited in the aggregation step.
- **Binary hidden units.** Since binary output bits are desired for our compression scheme, we use the sigmoid function as the activation function. The rectified linear activation function [27]

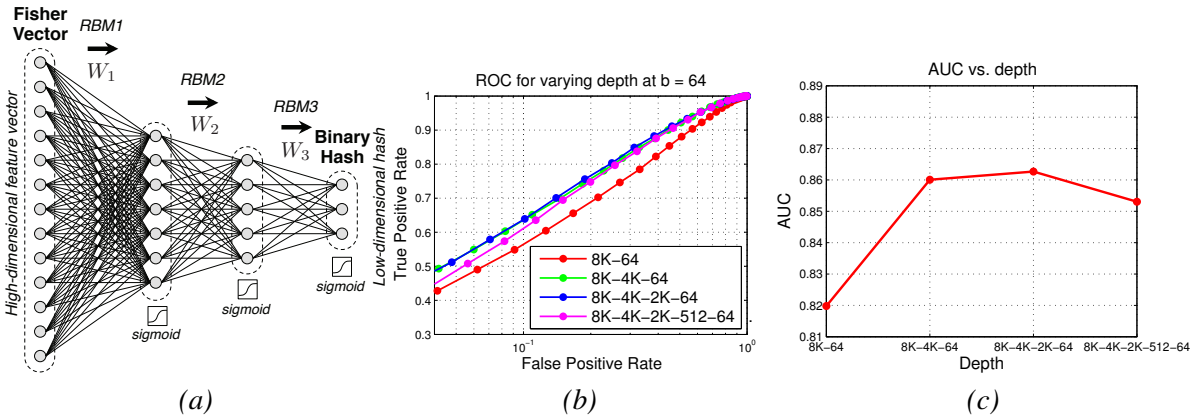


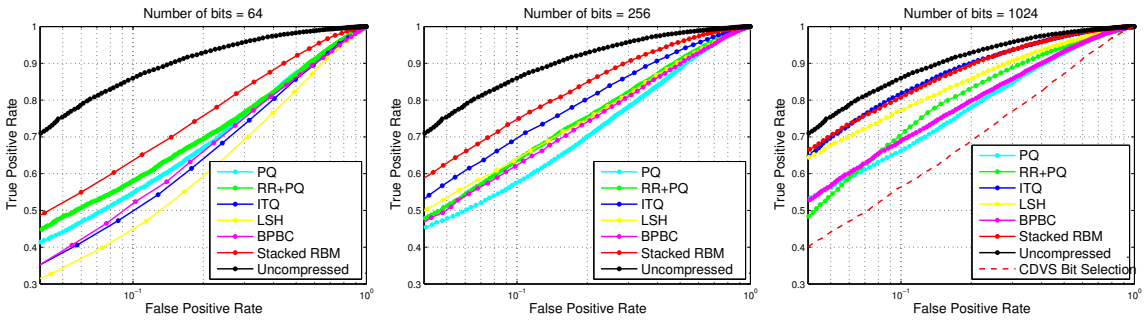
Figure 1. (a) Illustrated of Stacked RBM (SRBM). In (b) and (c), we show performance as depth is varied for a target bitrate of $b = 64$. We observe a sweet spot in the number of layers.

could also be chosen to train RBMs and deep networks, but they are not ideal for our purposes of binary compression.

- **Size of network.** The number of layers and the number of hidden units in each layer are parameters that are typically chosen experimentally [9], [20], [23]. Here, we progressively decrease the dimensionality of hidden layers by a factor of 2, and train several RBMs with varying number of hidden layers and output units to optimize parameters.
- **Sparsity.** We apply sparsity regularization to the network [28], with activation probability 0.5. Although it is typically desirable that the output vector be sparse with very few activated bits, in our case, we wish to build compact hashes from very high dimensional vectors, and hence we choose regularization parameters that push the network towards keeping half the bits active.
- **Greedy layer-wise unsupervised learning.** We perform greedy layer-by-layer training by fully training one RBM at a time using contrastive divergence. Each new RBM layer models the output layer of the previous layer. No supervised data is required for this training process. In this work, we explore and evaluate the performance of purely unsupervised representations.
- **Training data and parameters.** We use a random set of 150,000 images from the ImageNet data set [22]. This training set does not have any overlap with the query and database data used in the retrieval experiments. We set the learning rate to 0.001 for the weight and bias parameters, momentum to 0.9, and ran the training for a maximum 30 epochs.

In this work, we are interested in low rate points $b = 64$, $b = 256$ and $b = 1024$, typical operating points as discussed in Section 5. We present ROC results on the *CDVS* data sets, for output size $b = 64$ as the number of layers (depth) in the SRBM network is increased. In Figure 1(a), we show the full ROCs in log-scale for parameters $8K - 64$, $8K - 4K - 64$, $8K - 4K - 2K - 64$, $8K - 4K - 2K - 512 - 64$, corresponding to depth $1 \ldots 4$ respectively. Low FPR points are important as discussed in [29]. In Figure 1(b), we plot the Area Under the ROC curve (AUC) as a function of depth. AUC predicts retrieval performance well as shown in Section 3. For $b = 64$, we note that performance increases as the number of hidden layers increases from 1 to 3, and then starts decreasing. Similar trends are observed at other rate points too, where a sweet spot in performance for the depth parameter is observed.

For dimensionality reduction and hashing, there is a trade-off between performance and depth of the network. Having deeper networks is helpful as it allows discovery of more complex non-linear subspaces in which the data lie. On the other hand, given a target output bitrate, deeper



(a) $b = 64$ (b) $b = 256$ (c) $b = 1024$

Figure 2. Comparing ROC performance of difference schemes at varying bitrate b. We note that SRBM outperforms all other schemes.

networks can also result in worse performance as there is loss of information over the layers due to dimensionality reduction. We also observe that having more hidden layers at lower rates $b = 256$ is more beneficial, compared to higher rate points $b = 1024$. We decrease each hidden layers by a dimensionality factor of 2, train a set of RBMs, and choose the best performing RBM for each target rate. Each target setting requires several hours to train on a modern CPU. Next, we provide detailed comparisons against several other schemes.

5. Experimental Results

In this section, we compare pairwise matching and retrieval experiments for several state-of-the-art hashing and compression schemes. Some of these schemes have been proposed for lower dimensional vectors like SIFT and GIST, but we evaluate their performance on Fisher Vectors (FV).

- *PQ* [17]. Product Quantization (PQ) is applied to the original FV data. We consider groups of dimensions $D = 64, 256$ and 1024, and train $K = 256$ centroids for each block, resulting in $b = 64$, $b = 256$ and $b = 1024$ bit descriptors respectively.
- *RR+PQ* [17]. A random rotation matrix is applied to balance the variance across projected dimensions, followed by PQ [16]. We use the same parameters as above.
- *ITQ* [15] [15]. For the Iterative Quantization (ITQ) scheme, the authors propose signed binarization after applying two transforms: first the PCA matrix, followed by a rotation matrix, which minimizes the quantization error of mapping the PCA-transformed data to the vertices of a zero-centered binary hypercube.
- *BPBC* [16]. For high dimensional FV data, the PCA projection matrix might require GBs of data. Instead of the large projection matrices used in [15], the authors apply bilinear random projections, which require far less memory, to transform the data. This is followed by signed binarization to create binary hashes.
- *LSH* [12]. LSH is based on random unit-norm projections of the FV, followed by signed binarization.
- *CDVS Bit Selection* [6]. First, the Fisher Vector descriptor is binarized, similar to the Residual Enhanced Visual Vector [5]. For each image, a different set of 64 dimensional groups (corresponding to individual GMM centroids) are selected, based on a greedy bit-allocation scheme, which takes into account the variance of the GMM data sub-block. The first 128 bits for each image are mask bits, which indicate which of the GMM centroids are selected/active.

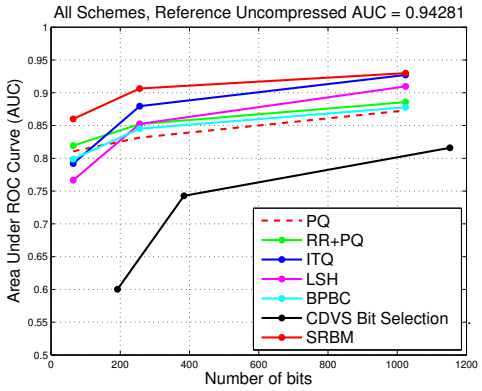


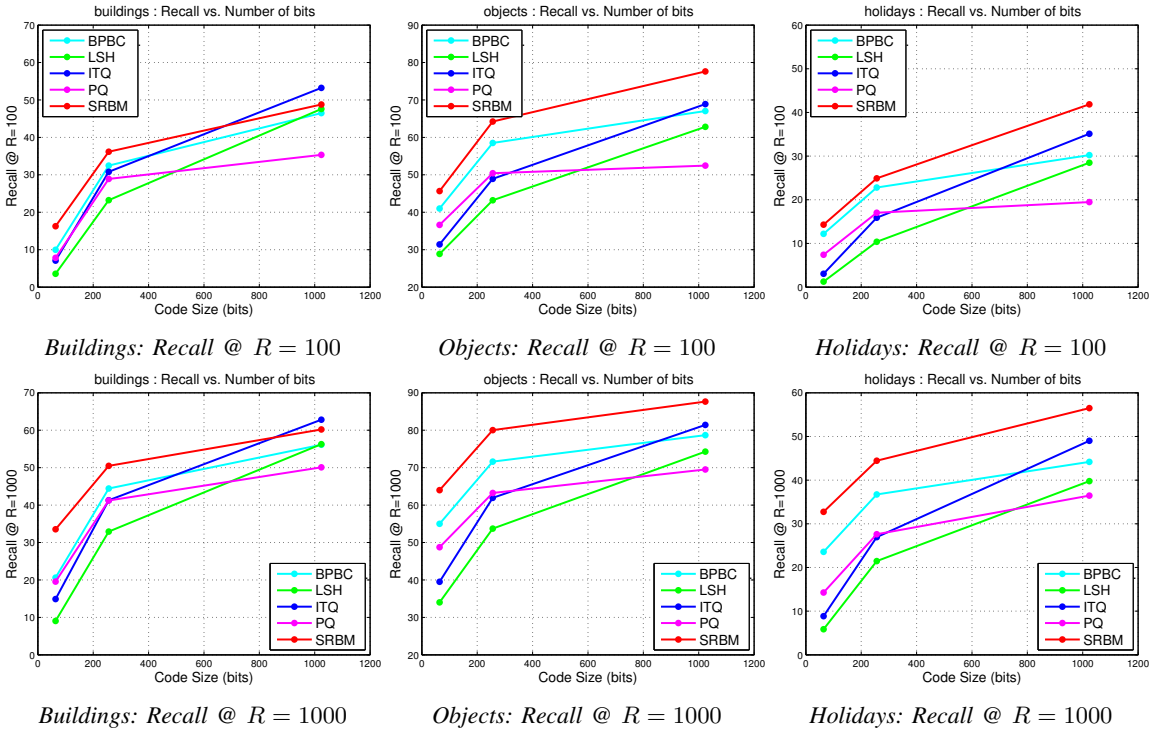
Figure 3. Comparing AUC performance of different schemes at varying b. We note that SRBM outperforms all schemes. Note that the AUC for the uncompressed FV is 0.94, and the *SRBM* scheme comes close to the performance of the uncompressed descriptor at $b = 1024$.

We report ROC and AUC results for $b = 64 + 128$, $b = 256 + 128$ and $b = 1024 + 128$ bits.

- *SRBM* For the proposed SRBM scheme, we use parameters $8K - 4K - 1K$, $8K - 4K - 256$ and $8K - 4K - 64$ for $b = 1024, 256, 64$ bits respectively. These parameters are chosen from the greedy optimization discussed in Section 4. At high rate points $b = 1024$ and above, performance quickly saturates, and adding hidden layers only provides a small improvement over a single layer RBM, and hence, a single layer can also be used.

Finally, as a baseline, we also show the performance of the uncompressed descriptors (floating point representation). L_2 norm is used for *PQ* schemes, while hamming distances are used for all binary hashing schemes. We present full ROC and AUC results in Figures 3 and 2 for the different schemes on the *CDVS* data set, at varying bitrates. We make the following observations.

- In Figure 2(a)-(c), we note that the performance ordering of schemes depends on the bitrate. The ordering of schemes is largely consistent between the full ROC curves and the AUC results. *SRBM* outperforms all schemes across the different rates. At $b = 64$, *RR+PQ* is second in performance, while the *ITQ* scheme comes second at $b = 256$ and $b = 1024$. The *PQ* schemes perform poorly at the low rates in consideration, as large blocks of the FV are quantized with a small number of centroids, as also observed in [16]. *SRBM* outperforms *ITQ*, as each output bit is generated by a series of non-linear projections which decorrelate the data.
- For the rest of the schemes, in Figure 3, we note that *PQ+RR* outperforms *PQ* by a small margin. Also, *BPBC* outperforms *ITQ* at $b = 64$ and $b = 256$, while the reverse is observed at $b = 1024$. *LSH* performs poorly at low rates, but given enough bits, *LSH* catches up in performance.
- We note that the *CDVS bit selection* scheme breaks down at low rates, as shown in Figure 3. However, this scheme is memory-efficient as it does not require storage of any large projection matrices, and was the basis of adoption in the MPEG-CDVS standard. In the standard, the global descriptor is stored as 2048-4096 bits: in that regime, most schemes work comparably.
- We note that there is a significant gap between the uncompressed descriptor and all the compression schemes at $b = 64$. In Figure 2(c), we note that the performance gap at 1024 bits is small between *SRBM* and uncompressed descriptors. In our experiments, most schemes match the performance of the uncompressed descriptor at 4096 bits.
- Finally, note that the retrieval performance of a scheme at a given database size depends



Buildings: Recall @ $R = 100$ *Objects: Recall @ $R = 100$* *Holidays: Recall @ $R = 100$*

Buildings: Recall @ $R = 1000$ *Objects: Recall @ $R = 1000$* *Holidays: Recall @ $R = 1000$*

Figure 4. Large-scale retrieval results (with 1M distractor images) for different compression schemes. *SRBM* outperforms other schemes at most rate points and datasets.

on the ROC curve at different TPR/FPR operating points, as shown in [29]. We note that the AUC results in Figure 3 predict well the performance ordering of different schemes for large-scale retrieval experiments, in Figures 4. Note that the large-scale retrieval experiments are performed on different datasets, but the AUC results generalize well, and can be used for quickly evaluating the performance of different algorithms.

Next, we present large-scale retrieval experiments in Figures 4 for three large data sets: *CDVS Buildings* (3499 queries), *CDVS Objects* (10200 queries) and *Holidays* data set (500 queries) with 1 million distractor images. For instance retrieval, the GCC step is computationally complex and can only be performed on a small number of images. As a result, it is important for the relevant image to be present in the short list, so that the GCC step can find it. Hence, we present recall at typical operating points, $R = 100$ and $R = 1000$ after the first step in the retrieval pipeline: matching of global descriptors. Best parameters for *SRBM* are chosen as described before. The retrieval results are largely consistent with the AUC results in Figure 3. We note that *SRBM* outperforms all other schemes at most rate points. The trends are consistent across $R = 100$ and $R = 1000$, with a larger gap between *SRBM* and other schemes at $R = 1000$.

Finally, we note there is a 20-30% drop in recall at low rate point $b = 64$, compared to $b = 1024$ for all schemes for large databases. Improving performance at such extremely low rates is an exciting direction for future work. While our starting global descriptor representation in this work was based on FV, we plan to explore low bitrate descriptors based on Convolutional Neural Networks (CNN) features in future work [8]. Also, the *SRBM* scheme can be further improved using weak supervision methods: e.g., taking into descriptor statistics of a large dataset of matching and non-matching pairs to find more compact and discriminative projections. Another promising direction would be to learn compact global descriptors for instance retrieval, directly

from image pixel data using CNNs [20].

6. Conclusion

In this work, we study the problem of global descriptor compression in the context of image retrieval, focusing on extremely compact binary representations: 64-1024 bits. We propose a compression scheme based on stacked Restricted Boltzmann Machines, which learn lower dimensional non-linear subspaces on which the data lie. With the proposed *SRBM* scheme, each output bit is generated by a series of non-linear projections which decorrelate the data. At 1000 bits, the proposed scheme comes close to matching the performance of the uncompressed descriptor. We provide a thorough evaluation of several state-of-the-art compression schemes based on PCA, Locality Sensitive Hashing, Product Quantization and greedy bit selection, and show that the proposed scheme outperforms all schemes.

References

[1] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, R. Grzeszczuk, Y. Reznik, and B. Girod, "Compressed Histogram of Gradients: A Low Bitrate Descriptor," *International Journal of Computer Vision, Special Issue on Mobile Vision*, vol. 96, no. 3, pp. 384–399, January 2012.

[2] M. Makar, V. Chandrasekhar, S. S. Tsai, D. M. Chen, and B. Girod, "Interframe coding of feature descriptors for mobile augmented reality," *IEEE Transactions on Image Processing*, vol. 23, no. 8, August 2014.

[3] H. Jégou, M. Douze, C. Schmid, and P. Perez, "Aggregating Local Descriptors into a Compact Image Representation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, June 2010, pp. 3304–3311.

[4] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale Image Retrieval with Compressed Fisher Vectors," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, June 2010, pp. 3384–3391.

[5] D. M. Chen, S. S. Tsai, V. Chandrasekhar, G. Takacs, R. Vedantham, R. Grzeszczuk, and B. Girod, "Residual Enhanced Visual Vector as a Compact Signature for Mobile Visual Search," in *Signal Processing, Elsevier, In Press*, June 2012.

[6] J. Lin, L.-Y. Duan, T. Huang, and W. Gao, "Robust Fisher Codes for Large Scale Image Retrieval," in *Proceedings of International Conference on Acoustics and Signal Processing (ICASSP)*, 2013.

[7] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," *CoRR*, vol. abs/1403.6382, 2014. [Online]. Available: http://arxiv.org/abs/1403.6382

[8] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural Codes for Image Retrieval," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[10] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, and B. Girod, "Transform Coding of Image Feature Descriptors," in *Proceedings of Visual Communications and Image Processing Conference (VCIP)*, San Jose, California, January 2009.

[11] H. Jégou, M. Douze, and C. Schmid, "Product Quantization for Nearest Neighbor Search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, January 2011.

[12] C. Yeo, P. Ahammad, and K. Ramchandran, "Rate-efficient Visual Correspondences using Random Projections," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, San Diego, California, October 2008, pp. 217–220.

[13] *Yahoo! 100 million image data set*, http://webscope.sandbox.yahoo.com/catalog.php?datatype=idid=67.

[14] Y. Weiss, A. Torralba, and R. Fergus, "Spectral Hashing," in *Proceedings of Neural Information Processing Systems (NIPS)*, Vancouver, BC, Canada, December 2008, pp. 1753–1760.

[15] Y. Gong and S. Lazebnik, "Iterative Quantization: A Procrustean Approach to Learning Binary Codes," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11, Washington, DC, USA, 2011, pp. 817–824.

[16] Y. Gong, S. Kumar, H. Rowley, and S. Lazebnik, "Learning Binary Codes for High-Dimensional Data Using Bilinear Projections." in *Proceedings of CVPR*, 2013, pp. 484–491.

[17] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.

[18] K. Grauman and R. Fergus, "Learning Binary Hash Codes for Large-Scale Image Search," in *Machine Learning for Computer Vision*, 2013, vol. 411, pp. 49–87.

[19] A. Oliva and A. Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope," *International Journal on Computer Vision*, vol. 42, no. 3, pp. 145–175, May 2001.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[21] B. T. Mark J. Huiskes and M. S. Lew, "New Trends and Ideas in Visual Concept Detection: The MIR Flickr Retrieval Evaluation Initiative," in *Proceedings of the 2010 ACM International Conference on Multimedia Information Retrieval*, New York, NY, USA, 2010, pp. 527–536.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[23] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul 2006.

[24] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann Machines for Collaborative Filtering," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 791–798.

[25] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, p. 1771–1800, 2002.

[26] G. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines," in *Neural Networks: Tricks of the Trade*, ser. LNCS, vol. 7700. Springer Berlin Heidelberg, 2012, pp. 599–619.

[27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *International Conference on Machine Learning (ICML)*, 2010.

[28] V. Nair and G. Hinton, "3D Object Recognition with Deep Belief Nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2009, pp. 1339–1347.

[29] D. M. Chen, "Memory-Efficient Image Databases for Mobile Visual Search," *Ph.D. thesis, Department of Electrical Engineering, Stanford University*, April 2014.