CO-SPARSITY REGULARIZED DEEP HASHING FOR IMAGE INSTANCE RETRIEVAL

Jie Lin[*,1], *Olivier Morère*[*,1,2], *Vijay Chandrasekhar*[1], *Antoine Veillard*[2], *Hanlin Goh*[1]

I2R[1], UPMC[2]

ABSTRACT

In this work, we tackle the problem of image instance retrieval with binary descriptors hashed from high-dimensional image representations. We present three main contributions: First, we propose Co-sparsity Regularized Hashing (CRH) to explicitly optimize the distribution of generated binary hash codes, which is formulated by adding a co-sparsity regularization term into the Restricted Boltzmann Machines (RBM) based hashing model. CRH is capable of balancing the variance of hash codes per image as well as the variance of each hash bit across images, resulting in maximum discriminability of hash codes that can effectively distinguish images at very low rates (down to 64 bits). Second, we extend the CRH into deep network structure by stacking multiple co-sparsity constrained RBMs, leading to further performance improvement. Finally, through a rigorous evaluation, we show that our model outperforms state-of-the-art at low rates (from 64 to 256 bits) across various datasets, regardless of the type of image representations used.

Index Terms— Image Instance Retrieval, Restricted Boltzmann Machines, Deep Hashing, Co-Sparsity

1. INTRODUCTION

Image instance retrieval regards the discovery of images from a database sharing same object/scene as the one depicted in query image. Most state-of-the-art image instance retrieval systems follow a two-stage pipeline: (1) retrieve a subset of candidate images from the database with high recall by comparing global descriptors of images, such as Fisher Vectors (FV) [1] and the recently proposed Deep Convolutional Neural Networks (DCNN) based descriptors like AlexNet [2] and OxfordNet [3], and (2) re-rank the candidate images with Geometric Consistency Check [4] for finding relevant database images with high precision. For the first stage, descriptor compression is usually applied to transform the high-dimensional global descriptors (4K to 64K) into compact codes, enabling fast matching and light storage on large scale database.

Descriptor compression techniques can be roughly grouped into two categories: (1) hashing, and (2) quantization. The goal of hashing is to compress raw descriptor into short binary vector with either data-independent hash like Locality Sensitive Hashing (LSH) [5] or data-dependent hash like Iterative Quantization (ITQ) [6] and Bilinear Projection Binary Codes (BPBC) [7]. For instance, ITQ first performs Principal Component Analysis (PCA) to reduce dimensionality of raw descriptor, then rotates the transformed PCA directions, finally binarizes each dimension according to its sign. The rotation operation is key to ITQ, as it balances the variance of PCA directions to ensure that each dimension carries comparable information before binarization.

Besides hashing, quantization based methods such as Product Quantization (PQ) [8] are alternative way for descriptor compression, where the raw descriptor is divided into smaller blocks and vector quantization is performed on each block. While this produces highly small descriptors composed of sub-quantizer indices, the final feature representation is non-binary and cannot be compared with ultra-fast Hamming distance computation.

In this work, we propose an unsupervised hashing scheme, termed Co-sparsity Regularized Deep Hashing (CRDH), for learning binary hash codes of high-dimensional descriptors. Our main contributions are three-fold:

- We propose Co-sparsity Regularized Hashing (CRH) to maximize the discriminability of hash codes. Specifically, CRH is formulated by adding a co-sparsity regularization term into the Restricted Boltzmann Machines (RBM) based hashing model, where each hash bit corresponds to a neural unit in latent layer. The generation of hash codes is directly optimized by co-sparsity constraints: (1) for a given image, half of the latent units are active (equal to 1) and (2) for each latent unit, there are half of the images on which it is fired (see Figure 1). Unlike existing hashing approaches, the co-sparsity constraints of CRH can balance both the variance of hash codes per image (i.e., with uniform sparsity 0.5) and the variance of each hash bit across images. This results in effective coding especially at extremely low rates (e.g., 64 bits) that hash codes spread in the limited binary space.
- We extend the CRH into deep network structure (i.e., CRDH) by stacking multiple RBMs with co-sparsity reg-

∗ Jie Lin and Olivier Morère contributed equally to this work.
1. Institute for Infocomm Research, A*STAR, Singapore.
2. Université Pierre et Marie Curie, Paris, France.

ularization within each latent layer. We experimentally find a tradeoff between network depth and co-sparsity constraint for further performance improvement.

- Through a thorough empirical evaluation on popular benchmark datasets with different image representations (FV and DCNN), we show that CRDH outperforms state-of-the-art unsupervised descriptor compression methods at low rates (e.g., 64 to 256 bits).

2. CO-SPARSITY REGULARIZED DEEP HASHING

Towards optimal hash codes, the proposed CRDH is built up with deep network structure due to the remarkable success of deep learning in recent years. First, we briefly describe the Restricted Boltzmann machines (RBM), which is the base building block of CRDH. Then, we introduce how to add the co-sparsity regularisation term into the RBM. Finally, we present the CRDH by stacking multiple co-sparsity constrained RBMs.

Hashing with RBM. RBM is a bipartite Markov random field with the input layer $z^{l-1} \in R^I$ connected to a latent layer $z^l \in R^J$ via a set of undirected weights $W^l \in R^{IJ}$. The input units z_i^{l-1} and latent units z_j^l are also parameterised by their corresponding biases c_i^{l-1} and b_j^l, respectively. The input layer takes a high-dimensional image descriptor as input. Previous works [1, 9] have shown that binarization of FV and DCNN features results in negligible loss in performance. For this work, binarization is done by component-wise mean thresholding for the inputs. We use binary latent units with sigmoid activation function, because binary output bits are desired for our hash.

The units within a layer are conditionally independent pairwise. Therefore, the activation probabilities of one layer can be sampled by fixing the states of the other layer, and using distributions given by logistic functions for binary RBM:

$$P(z_j^l|z^{l-1}) = 1/(1+\exp(-w_j^l z^{l-1} - b_j^l)), \quad (1)$$

$$P(z_i^{l-1}|z^l) = 1/(1+\exp(-w_i^{l\top} z^l - c_i^{l-1})). \quad (2)$$

As a result, alternating Gibbs sampling can be performed between the two layers. The sampled states are used to update the parameters $\{W^l, b^l, c^{l-1}\}$ through minibatch gradient descent using the contrastive divergence algorithm [10] to approximate the maximum likelihood of the input distribution.

Given a trained RBM with fixed parameters and an input vector, a hash can be generated through a feedforward projection and thresholding Equation (1) at 0.5.

$$z_j^l = \begin{cases} 1, & \text{if } P(z_j^l|z^{l-1}) > 0.5 \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Co-sparsity Regularized RBM. The latent layer in RBM is trained to model the data distribution of the previous layer. It is, however, important for the RBM to project the data

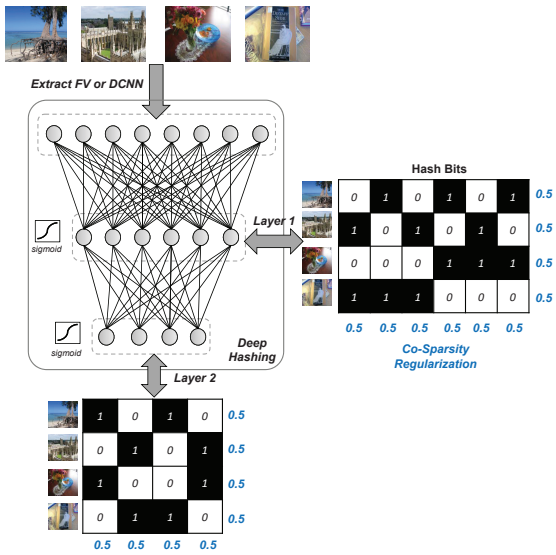


Fig. 1. Our proposed hashing and model training pipeline.

in a latent subspace that is suitable for hashing. One way to encourage the learning of suitable representations is to perform regularization, such as sparsity [11, 12, 13]. For classification, representations are encouraged to be very sparse to improve separability. For hashing, however, it is desirable to encourage the representation to make efficient use of the limited latent subspace.

For a given l and a minibatch of input instances z_α^{l-1}, we add a co-sparsity regularization term to the RBM optimization problem to encourage (a) half the latent units (hash codes) to be active for a given image, and (b) each hash bit to be equiprobable across images:

$$\underset{\{W^l,b^l,c^{l-1}\}}{\arg\min} -\sum_\alpha \log\Bigg(\sum_{z_\alpha^l \in E_\alpha} P(z_\alpha^{l-1}, z_\alpha^l) + \lambda h(E_\alpha)\Bigg), \quad (4)$$

where E_α is the minibatch of sampled latent units for layer l and λ is the regularization constant.

We adapt the fine-grained regularization proposed in [13] to suit our hashing problem. For each instance z_α^l, the regularization term for binary units penalises each unit $z_{j\alpha}^l$ with the cross entropy loss with respect to a target activation $t_{j\alpha}^l$ based on a predefined distribution,

$$h(E_\alpha) = -\sum_{z_\alpha^l \in E_\alpha}\sum_j t_{j\alpha}^l \log z_{j\alpha}^l + (1-t_{j\alpha}^l)\log(1-z_{j\alpha}^l). \quad (5)$$

Unlike [13], we choose the $t_{j\alpha}^l$ such that each $\{t_{j\alpha}^l\}_j$ for fixed α and each $\{t_{j\alpha}^l\}_\alpha$ for fixed j is distributed according to $U(0,1)$. The uniform distribution is suitable for hashing high-dimensional vectors because the regularizer encourages each latent unit to be active with a mean of 0.5, while avoiding activation saturation. The result is a space-filling effect in the latent subspace, where data is efficiently represented.

After RBM training, we further enforce space utilization by substituting the learned RBM bias by the data set mean $\langle w_j z^{l-1} \rangle$ of the linear projection preceding the logistic. Equation (3) is modified such that the final hash is centered around 0.5:

$$z_j^l = \begin{cases} 1, & \text{if } w_j z^{l-1} - \langle w_j z^{l-1} \rangle > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Stacked Co-sparsity Regularized RBMs. The set of raw image representations lie in a complex manifold in a very high-dimensional feature space. Deeper networks have the potential to discover more complex nonlinear hash functions and improve image instance retrieval performance. Following [14], we stack multiple RBMs by training one layer at a time to create multi-layer deep networks.

Each layer models the activation distribution of the previous layer and captures higher-order correlations between those neurons (units). For hashing problem, we are interested in low bitrate operating points of 64, 256 and 1024 bits. We progressively decrease the dimension of latent layers by a factor of 2^n per layer, where n is a tuneable parameter. For our final models, n is empirically selected for each layer resulting in variable network depth.

3. EXPERIMENTS

3.1. Datasets

We use 2 widely used benchmark datasets for small scale experiments: INRIA Holidays (500 queries, 991 database images) [15], University of Kentucky Benchmark (UKbench) (10200 queries, 10200 database images) [16]. To evaluate large-scale retrieval, we present results on Holidays and UKbench data sets, combining with the 1-million distractor image dataset MIRFLICKR [17] respectively.

Most schemes, including our proposed scheme, require a training step. We train on a random 150K images subset of the *ImageNet* training set, which consists of 1.2 million images from 1000 different categories [18]. This training set is independent from the query and database images described above.

3.2. Experimental Setup

Image Descriptors. We start with global descriptor representations based on both Fisher Vectors (FV) and Deep Convolutional Neural Networks (DCNN). For the FV, we extract SIFT [19] descriptors obtained from Difference-of-Gaussian (DoG) detector. PCA is adopted to reduce dimension of SIFT descriptor from 128 to 64, which has shown to improve performance [8]. We use a Gaussian Mixture Model (GMM) with 128 centroids, resulting in 8192-dimensional FV with the first order statistics. Finally, the FV is power normalized, followed by L_2-normalization.

DCNN features are extracted using the open-source software Caffe [20] with AlexNet reference model proposed by Alex Krizhevsky et al. for 2012 *ImageNet* classification task [2]. We find that layer *fc6* (before softmax) performs the best for image retrieval, similarly to results recently reported in [21]. We refer to this 4096-dimensional fc6 as the DCNN feature from here-on.

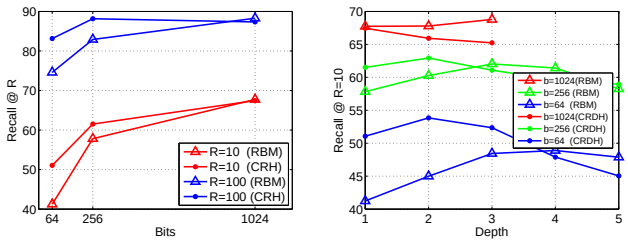
Baselines. We compare our approach with state-of-the-art compression algorithms on both FV and DCNN features. (1) *LSH* [5]. LSH is performed by random unit-norm projections of the raw descriptors, followed by signed binarization. (2) *ITQ* [6]. ITQ applies signed binarization after two transforms of raw descriptors: first the PCA to reduce dimensionality, followed by a learned rotation. (3) *BPBC* [7]. BPBC applies bi-linear projections to transform the raw data, which require far less memory than ITQ [6]. (4) *PQ* [8]. For FV, we consider blocks with dimensions $D = 64, 256$ and 1024, for each block we train a codebook with $K = 256$ centroids, resulting in $b = 128, 256$ and 64 bit descriptors respectively. For DCNN, we consider blocks of dimensions $D = 32, 128$ and 512, with $K = 256$ centroids, resulting in the same bitrates. For small retrieval experiments, we also show the performance of the uncompressed descriptors as a baseline. L_2 norm is used for *PQ* and uncompressed descriptors, while hamming distances are adopted for all binary descriptors.

Evaluation Metrics. For instance retrieval, it is important for the relevant image to be present in the first step of the pipeline, matching global descriptors, so that a Geometric Consistency Check [4] step can find it subsequently. Thus, we present Recall @ typical operating points, R = {10,100} and R = 1000 for small and large experiments respectively. For *UKbench* small experiments, we plot 4× Recall @ $R = 4$, to be consistent with the literature.

3.3. Results

Co-sparsity Regularization. In Figure 2(a), we show the effect of applying co-sparsity regularization on a single layer RBM 8192-*b*, for $b = 64, 256, 1024$. The *Holidays* data set and FV are chosen. CRH improves performance significantly, $\sim$10% absolute Recall @ $R = 10$ at low-rate point $b = 64$. The performance gap increases as rate decreases. This is intuitive as the regularization pushes the network towards keeping half the bits alive and equiprobable (across hashes), with its effect being more pronounced at lower rates.

Depth. In Figure 2(b), we plot Recall @ $R = 10$ for the *Holidays* data set and *FV* features, as depth is increased for a given rate point b. For $b = 1024$, we consider configurations 8192-1024, 8192-4096-1024, and 8192-4096-2048-1024 corresponding to depth $1, 2, 3$ respectively. For rate points $b = 64$ and 256, similar configurations of varying depth are chosen. We observe that, with no co-sparsity regularization, recall improves as depth is increased for $b = 256$ and $b = 64$, with optimal depth of 3 and 4 respectively, be-



(a) Co-sparsity regularization (b) Depth

Fig. 2. Hashing FV for *Holidays*. (a) Co-sparsity regularization (CRH) improves performance significantly for single layer models 8192-b as b is decreased. (b) Recall improves as depth is increased for lower rate points $b = 64$ and $b = 256$. With co-sparsity regularization, we can achieve the same or better recall at lower depth.

yond which performance drops. At higher rates of $b = 1024$ and beyond, increasing depth does not improve as performance saturates.

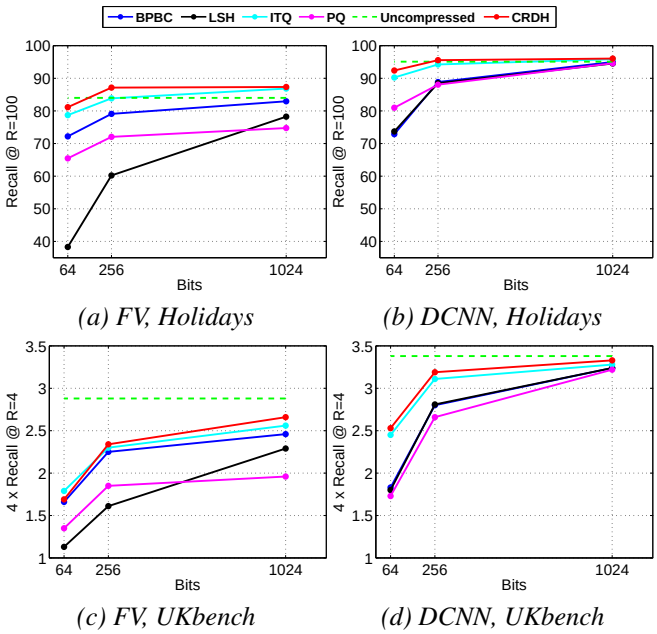
For hashing, depth parameter sweet spot varies with compression rate points. Similar trends are obtained for Recall @ $R = 100$. Importantly, we observe that with the proposed regularization, we can achieve the same performance with lower depth at each rate point. This is critical, as lower the depth, the faster the hash generation, and lower the memory requirements.

Comparison to state-of-the-art. The small scale retrieval results are shown in Figure 3. One can see that the proposed CRDH outperforms state-of-the-art at most rates on all data sets, for both DCNN and FV features. There is 2.4% improvement in absolute Recall @ $R = 100$ at $b = 64$ bits compared to the second performing scheme ITQ on Holidays for FV. Consistent trends are also obtained for the large-scale retrieval results in Figure 4.

The performance ordering of other schemes depends on the bitrate and type of feature, while CRDH is consistent across data sets. Compared to *ITQ* scheme which applies a single PCA transform, each output bit for CRDH is generated by a series of projections. The *PQ* scheme performs poorly at the low rates in consideration, as large blocks of the global descriptor are quantized with a small number of centroids, as previously observed in [7]. *LSH* performs poorly at low rates, but catches up given enough bits.

Comparing FV-CRDH and DCNN-CRDH. At a given rate point, DCNN-CRDH outperforms FV-CRDH for all data sets, as shown in Figure 3. At low rates, DCNN-CRDH improves performance by more than 10% on the small data sets. The reason may be DCNN features are able to capture more complex low level features and have a lower starting dimensionality compared to FV.

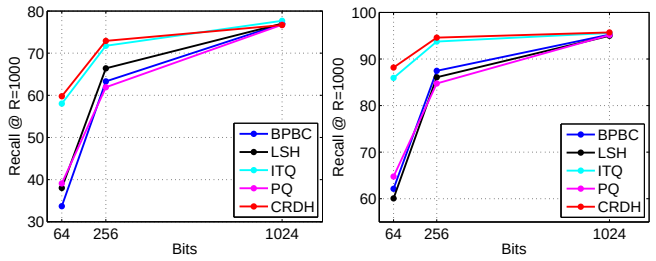
Comparison to Uncompressed Descriptors. We compare the results of CRDH to the uncompressed descriptor in Figure 3. At 256 bits for DCNN, we only observe a marginal drop (a few%) compared to the uncompressed descriptor for retrieval on all data sets. For FV, we can match the performance of the uncompressed descriptor with 1024 bits for *Holidays* and *UKbench*. The instance retrieval hashing problem becomes increasingly difficult as we move towards a 64-bit hash, with performance dropping steeply.



(a) FV, Holidays *(b) DCNN, Holidays*

(c) FV, UKbench *(d) DCNN, UKbench*

Fig. 3. Small-scale retrieval results. *CRDH* outperforms other schemes by a significant margin.



(a) DCNN, Holidays+1M *(b) DCNN, UKbench+1M*

Fig. 4. Large-scale retrieval results (with 1M distractor images) for different compression schemes. *CRDH* outperforms other schemes at most rate points and data sets.

4. CONCLUSIONS

A perfect image hashing scheme would convert a high-dimensional descriptor into a low-dimensional bit representation without losing retrieval performance. We believe that deep hashing, which focuses on achieving complex hash functions with deep learning, is a significant step in this direction. Our method is focused on a deep network which efficiently utilizes the binary subspace through co-sparsity regularization. Through a rigorous evaluation process, we show that our model performs well across various data sets, regardless of the type of image descriptors used.

5. REFERENCES

[1] 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.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[3] K Simonyan and A Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *arXiv preprint arXiv:1409.1556*, 2014, pp. 1–10.

[4] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[5] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni, "Locality-Sensitive Hashing Scheme based on p-stable Distributions," in *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, New York, USA, June 2004, pp. 253–262.

[6] Yunchao 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*, Washington, DC, USA, 2011, CVPR '11, pp. 817–824.

[7] Yunchao Gong, Sanjiv Kumar, Henry Rowley, and Svetlana Lazebnik, "Learning Binary Codes for High-Dimensional Data Using Bilinear Projections.," in *Proceedings of CVPR*, 2013, pp. 484–491.

[8] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, Sept. 2012.

[9] Pulkit Agrawal, Ross Girshick, and Jitendra Malik, "Analyzing the performance of multilayer neural networks for object recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

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

[11] Honglak Lee, Chaitanya Ekanadham, and Andrew Ng, "Sparse deep belief net model for visual area V2," in *Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 873–880.

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

[13] Hanlin Goh, Nicolas Thome, Matthieu Cord, and Joo-Hwee Lim, "Unsupervised and supervised visual codes with restricted Boltzmann machines," in *European Conference on Computer Vision (ECCV)*, 2012.

[14] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh, "A fast learning algorithm for deep belief networks," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[15] H. Jégou, M. Douze, and C. Schmid, "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search," in *Proceedings of European Conference on Computer Vision (ECCV)*, Berlin, Heidelberg, October 2008, pp. 304–317.

[16] D. Nistér and H. Stewénius, "Scalable Recognition with a Vocabulary Tree," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, New York, USA, June 2006, pp. 2161–2168.

[17] B. Thomee Mark J. Huiskes and Michael 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.

[18] 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.

[19] D. Lowe, "Distinctive Image Features from Scale-invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.

[20] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[21] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky, "Neural Codes for Image Retrieval," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.