END-TO-END BINARY REPRESENTATION LEARNING VIA DIRECT BINARY EMBEDDING

Liu Liu, Alireza Rahimpour, Ali Taalimi, Hairong Qi

Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville

ABSTRACT

Learning binary representation is essential to large-scale computer vision tasks. Most existing algorithms require a separate quantization constraint to learn effective hashing functions. In this work, we present Direct Binary Embedding (DBE), a simple yet very effective algorithm to learn binary representation in an end-to-end fashion. By appending an ingeniously designed DBE layer to the deep convolutional neural network (DCNN), DBE learns binary code directly from the continuous DBE layer activation without quantization error. By employing the deep residual network (ResNet) as DCNN component, DBE captures rich semantics from images. Furthermore, in the effort of handling multilabel images, we design a joint cross entropy loss that includes both softmax cross entropy and weighted binary cross entropy in consideration of the correlation and independence of labels, respectively. Extensive experiments demonstrate the significant superiority of DBE over state-of-the-art methods on tasks of natural object recognition, image retrieval and image annotation.

Index Terms— Binary Representation, Deep Convolutional Neural Networks, Direct Binary Embedding, Image Hashing, Multilabel Classification

1. INTRODUCTION

Representation learning is key to computer vision tasks. Recently with the explosion of data availability, it is crucial for the representation to be computationally efficient as well [1, 2, 3]. Consequently learning high-quality binary representation is tempting due to its compactness and representation capacity.

Binary representation traditionally has been learned for image retrieval and similarity search purposes (image hashing). From the early works using hand-crafted visual features [4, 5, 6, 7] to recent end-to-end approaches [8, 9, 10] that take advantages of deep convolutional neural networks (DCNN), the core of image hashing is learning binary code for images by characterizing the similarity in a pre-defined neighborhood. Usually pairwise or triplet similarity are considered to capture such similarity among image pairs or triplets, respectively [6, 8, 9]. Albeit the high-quality of binary code, most image hashing algorithms do not consider learning discriminative binary representation. Recently this gap was filled by several hashing algorithms that learn binary representation via classification [10, 1, 2]. Not only does the learned binary code retrieves images effectively, it provides comparable or even superior performance for classification as well. Meanwhile, due to the discrete nature of binary code, it is usually impractical to optimize discrete hashing function directly. Most hashing approaches attempt solving it by a continuous relaxation and quantization loss [1, 9]. However, such optimization is usually not statistically stable [8] and thus leads to suboptimal hash code.

In this work, we propose to learn high-quality binary representation *directly* from deep convolutional neural networks (DCNNs). By appending a binary embedding layer directly into the state-of-the-art DCNN, deep residual network, we train the whole network as a hashing function via classification task in attempt to learning representation that approximates binary code without the need of using quantization error. Thus we name our approach Direct Binary Embedding (DBE). Furthermore, in order to learn high-quality binary representation for multilabel images, we propose a joint cross entropy that incorporates softmax cross entropy and weighted binary sigmoid cross entropy in consideration of the correlation and independence of labels, respectively. Extensive experiments on two large-scale datasets (CIFAR-10 and Microsoft COCO) show that the proposed DBE outperforms state-of-the-art hashing algorithms on object classification and retrieval tasks. Additionally, DBE provides a comparable performance on multilabel image annotation tasks where usually continuous representation is used.

2. DIRECT BINARY EMBEDDING

2.1. Direct Binary Embedding (DBE) Layer

We start the discussion of DBE layer by revisiting learning binary representation using classification. Let $\mathbf{I} = \{I_i\}_{i=1}^N$ be the image set with n samples, associated with label set $\mathbf{Y} = \{y_i\}_{i=1}^N$. We aim to learn binary representation $\mathbf{B} = \{\mathbf{b}_i\}_{i=1}^N \in \{0, +1\}^{N \times L}$ of $\mathbf{I}$ via the Direct Binary Embedding layer that is appended to DCNN. Following the paradigm of classification problem formulation in DCNN, we use a linear classifier $\mathbf{W}$ to classify the binary representation:

$$\min_{W,F} \frac{1}{N} \sum_{i=1}^{N} \Big(\mathcal{L}(\mathbf{W}^\top \mathbf{b}_i, y_i) + \lambda \|\mathbf{b}_i - F(I_i; \Omega)\|_2^2 \Big) \quad (1)$$
$$\text{s.t. } \mathbf{b}_i = thresold(F(I_i; \Omega), 0.5)$$

where $\mathcal{L}$ is an appropriate loss function; $\|\mathbf{b}_i - F(I_i; \Omega)\|_2^2$ measures the quantization error of between the DCNN activation $F(I; \Omega)$ and the binary code $\mathbf{b}_i$; λ is the coefficient controlling the quantization error; $threshold(v, t)$ is a thresholding function at t, and it equals to 1 if $v \geq t$, 0 otherwise; F is a composition of $n + 1$ non-linear projection functions parameterized by Ω:

$$F(I, \Omega) = f_{\text{DBE}}(f_n(\cdots f_2(f_1(I; \omega_1); \omega_2) \cdots ; \omega_n)\omega_{\text{DBE}}), \quad (2)$$

where the inner n nonlinear projections composition denotes the n-layer DCNN; $f_{\text{DBE}}(\cdot; \omega_{\text{DBE}})$ is the Direct Binary Embedding layer appended to the DCNN. The binary code $\mathbf{b}_i$ in Eq. 1 makes it difficult to optimize via regular DCNN inference. We relax Eq. 1 to the following form where stochastic gradient descent is feasible:

$$\min_{W,F} \frac{1}{N} \sum_{i=1}^{N} \Big(\mathcal{L}(\mathbf{W}^\top F(I_i; \Omega), y_i) + \lambda ||2F(I_i; \Omega) - \mathbf{1}| - \mathbf{1}|^2 \Big) \quad (3)$$

As proved by [8], the quantization loss $||2F(I_i;\Omega) - \mathbf{1}| - \mathbf{1}|^2$ in Eq. 3 is an upper bound of that in Eq. 1, making Eq. 3 an appropriate relaxation and much easier to optimize.

Several studies such as [8] share the similar idea of encouraging the fully-connected layer representation to be binary codes by using hyperbolic tangent (tanh) activation. Since it is desirable to learn binary code $\mathbf{B} = \{0, +1\}^{N\times L}$, we propose to concatenate the ReLU (rectified linear unit) nonlinearity with the tanh nonlinearity. Formally, we define DBE layer (shown in Figure 1):

$$\mathbf{Z} = f_{\text{DBE}}(\mathbf{X}) = \tanh(\text{ReLU}(\text{BN}(\mathbf{X}\mathbf{W}_{\text{DBE}} + b_{\text{DBE}}))) \tag{4}$$

where $\mathbf{X} = f_n(\cdots f_2(f_1(\mathbf{I};\omega_1);\omega_2)\cdots;\omega_n) \in \mathbb{R}^{N\times d}$ is the activation of n-layer DCNN; $\mathbf{Z} = f_{\text{DBE}}(\mathbf{X}) \in \mathbb{R}^{N\times L}$ is the binary-like activation of DBE layer; $\mathbf{T} = \text{BN}(\mathbf{X}\mathbf{W}_{\text{DBE}} + b_{\text{DBE}})$ is the activation after linear projection and batch normalization but prior to ReLU and tanh; $\mathbf{W}_{\text{DBE}} \in \mathbb{R}^{d\times L}$ is a linear projection, b_{DBE} is the bias; $\text{BN}(\cdot)$ is the batch normalization. And its activation is plotted in Figure 2a. The benefit of DBE layer approximating binary code is three-fold:

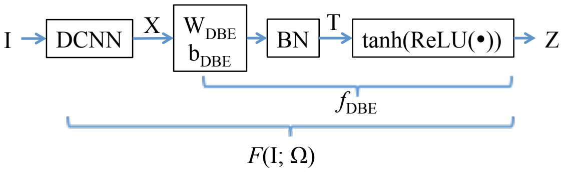
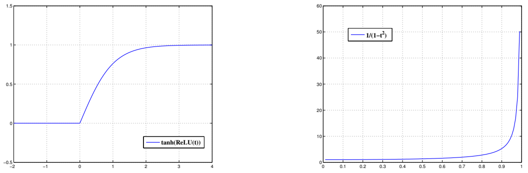


Fig. 1: The framework of DBE and outputs of different projections

1. batch normalization mitigates training with saturating nonlinearity such as tanh [11], and potentially promotes more effective binary representation.
2. ReLU activation is sparse [12] and learns bit '0' inherently.
3. tanh activation bounds the ramping of ReLU activation and learns bit '1' effectively without jeopardizing the sparsity of ReLU.



(a) activation

(b) The PDF for positive input

Fig. 2: $\tanh(\text{ReLU}(\cdot))$ activation and its PDF for positive input

Furthermore, DBE layer learns activation that approximates binary code statistically well. Consider random sampling t from $\mathbf{T}$, and assume it follows a distribution denoted by $p_T(t)$. Consequently the distribution of the DBE layer activation $z = f_{\text{DBE}}(t)$, and it follows distribution $p_Z(z)$, written as:

$$p_Z(z) = p_T(f_{\text{DBE}}^{-1}(z)) \left| \frac{1}{f'_{\text{DBE}}(f_{\text{DBE}}^{-1}(z))} \right| \tag{5}$$

Eq. 5 holds since f_{DBE} is a monotonic and differentiable function. Since it is also positive when z is positive, thus we have:

$$p_Z(z) = p_X(f_{\text{DBE}}^{-1}(z)) \frac{1}{1 - f_{\text{DBE}}^{-1}(z)^2}, \quad f_{\text{DBE}}^{-1}(z) = t > 0. \tag{6}$$

$p_T(f_{\text{DBE}}^{-1}(z))$ in Eq. 6 is equivalent to $p_T(t)$; $\frac{1}{1-f_{\text{DBE}}^{-1}(z)^2}$ grows sharply towards the discrete value $\{+1\}$ for any positive response z, as is plotted in Figure 2b. This suggests that the DBE layer enforces that the learned embedding z are assigned to $\{+1\}$ with large probability as long as z is positive. Conclusively DBE layer f_{DBE} can effectively approximate binary code. Eventually we choose to optimize Eq. 3 without the quantization error and replace the binary code $\mathbf{b}_i$ with DBE layer activation directly. Eq. 3 can thus be rewritten as:

$$\min_{\mathbf{W},F} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathbf{W}^\top F(I_i;\Omega), y_i) \tag{7}$$
$$\text{s.t. } F(\mathbf{I},\Omega) = f_{\text{DBE}}(f_n(\cdots f_2(f_1(\mathbf{I};\omega_1);\omega_2)\cdots;\omega_n)\omega_{\text{DBE}})$$

The inference of DBE is the same as canonical DCNN models via stochastic gradient descent (SGD).

2.2. Multiclass Image Classification

Majority of DCNNs are trained via multiclass classification using softmax cross entropy as the loss function. Following this paradigm, Eq. 7 can be instantiated as:

$$\min_{\mathbf{W},F} -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{C} \mathbb{1}(y_i) \log \frac{e^{\mathbf{w}_k^\top F(I_i;\Omega)}}{\sum_{j=1}^{C} e^{\mathbf{w}_j^\top F(I_i;\Omega)}} \tag{8}$$
$$\text{s.t. } F(\mathbf{I},\Omega) = f_{\text{DBE}}(f_n(\cdots f_2(f_1(\mathbf{I};\omega_1);\omega_2)\cdots;\omega_n)\omega_{\text{DBE}})$$

where C is the number of categories; $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_C]$ and $\mathbf{w}_k,\ k = 1, \ldots, C$ is the weight of the classifier for category k; y_i is the label for image sample I, and $\mathbb{1}(y_i)$ an indicator function representing the probability distribution for label y_i. Essentially Eq. 8 aims to minimize the difference between the probability distribution of ground truth label and prediction.

2.3. Multilabel Image Classification

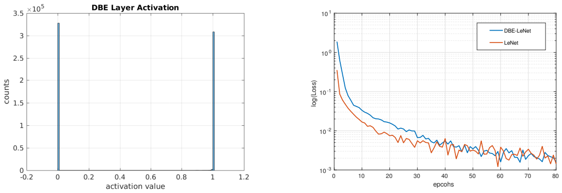
More often a real-world image is associated with multiple objects belonging to different categories. A natural formulation of optimization problem for multilabel classification is extending the multiclass softmax cross entropy in Eq. 8 to multilabel cross entropy. Indeed softmax cross entropy captures the co-occurrence dependencies among labels, one cannot ignore the independence of each individual labels. For instance, 'fork' and 'spoon' usually co-exist in an image as they are associated with super-concept 'dining'. But occasionally a 'laptop' can be placed randomly on the dining table where there are also 'fork' and 'spoon' in the image as well. Consequently, we propose to optimize a joint cross entropy by incorporating weighted binary sigmoid cross entropy, which models each label independently, to softmax cross entropy. Eq. 7 can therefore be instantiated as:

$$\min_{\mathbf{W},F} -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{c_+} \frac{1}{c_+} \log \frac{e^{\mathbf{w}_j^\top F(I_i;\Omega)}}{\sum_{p=1}^{C} e^{\mathbf{w}_p^\top F(I_i;\Omega)}} \tag{9}$$
$$-\nu \frac{1}{N} \sum_{i=1}^{N} \sum_{p=1}^{C} \Bigg[\rho \mathbb{1}(y_i) \log \frac{1}{1 + e^{\mathbf{w}_p^\top F(I_i;\Omega)}}$$
$$+ (1 - \mathbb{1}(y_i)) \log \frac{e^{\mathbf{w}_p^\top F(I_i;\Omega)}}{1 + e^{\mathbf{w}_p^\top F(I_i;\Omega)}} \Bigg]$$
$$\text{s.t. } F(\mathbf{I};\Omega) = f_{\text{DBE}}(f_n(\cdots f_2(f_1(\mathbf{I};\omega_1);\omega_2)\cdots;\omega_n)\omega_{\text{DBE}})$$

where c_+ is the number of positive labels for each image; ν is the coefficient controlling the numerical balance between softmax cross entropy and binary sigmoid cross entropy; ρ is the coefficient penalizing the loss for predicting positive labels incorrectly.

2.4. Toy Example: LeNet with MNIST Dataset

In order to demonstrate the effectiveness of DBE layer, we use LeNet as a simple example of DCNN. We add DBE layer to the last fully connected layer of LeNet and learn binary representation for MNIST dataset. **MNIST** dataset [13] contains 70K hand-written digits of 28×28 pixel size, ranging from '0' to '9'. The dataset is split into a 60K training set (including a 5K validation set) and a 10K test set[1]. We enhance the original LeNet with more convolutional kernels (16 kernels and 32 kernels on the first and second layer, respectively, all with size 3×3). We train the LeNet with DBE layer on the training set and evaluate the quality of learned binary representation on the test set. Figure 3a demonstrates the histogram of activation from DBE. Clearly DBE layer learns a representation approximating binary code effectively (51.1% of DBE activation less than 0.01, 48.6% greater than 0.99 and only 0.3% in between). We evaluate the quality of binary code learned by DBE qualitatively by comparing the classification accuracy on the test set with the state-of-the-art hashing algorithm. In order to demonstrate the effectiveness of DBE, we also compare with different λ in Eq. 3 for the purpose of showing that quantization error is not necessary anymore to learn high-quality binary representation. From Table 2 we can see that with the increase of λ in Eq. 3, the testing accuracy decreases. Due to the effectiveness of DBE layer, quantization error does not contribute to the binary code learning. Following the evaluation protocol of previous works [1], linear-SVM [14] is used as the classifier on all compared methods for fair comparison (including continuous LeNet representation). The classification accuracy on the test set is reported in Table 1.



(a) (b)

Fig. 3: The qualitatively results of DBE-LeNet: (a)The histogram of DBE layer activation; (b)The convergence of the original LeNet and with DBE trained on MNIST

Method	LeNet [13]	DBE-LeNet	SDH [1]	FastHash [7]
testing acc(%)	99.34	99.34	99.14	98.62

Table 1: The comparison of the testing accuracy on MNIST. Code-length for all hashing algorithms is 64-bit. LeNet feature (1000-d continuous vectors) is used for SDH and FastHash.

λ	0	1e-4	1e-3	1e-2	1e-1
testing acc(%)	99.34	99.34	99.30	99.26	99.01

Table 2: The impact on quantization error coefficient λ

The convergence of training DBE-LeNet is reported in Figure 3b. Due to the saturating tanh activation, the gradient is slightly more difficult to propagate through the network. Eventually the convergence reaches the same level.

[1] http://yann.lecun.com/exdb/mnist/

3. EXPERIMENTS

We evaluate the proposed DBE layer with the deep residual network (ResNet). We choose to append DBE layer to the state-of-the-art DCNN, 50-layer Residual Network (ResNet-50) [15] to learn high-quality binary representation for image sets. For the multilabel experiments, we set $\nu = 2$ and $\rho = 5$ through extensive empirical study.

3.1. Dataset

CIFAR-10 dataset [16] contains 60K color images (size 32×32) with each image containing a natural object. There are 10 categories of objects in total, with each category containing 6K images. The dataset is randomly split into a 50K training set and a 10K testing set. For traditional image hashing algorithms, we provide 512-D GIST [17] feature; for end-to-end deep hashing algorithms, we use raw images as input directly. **Microsoft COCO 2014 (COCO)** [18] is a dataset for image recognition, segmentation and captioning. It contains a training set of 83K images with 605K annotations and a validation set of 40K images with 292K annotations. There are totally 80 categories of annotations. We treat annotations as labels for images. On average each image contains 7.3 labels. Since images in COCO are color images with various sizes, we resize them to 224×224.

3.2. Object Classification

To evaluate the capability of mulitclass object classification, we compare DBE with several state-of-the-art supervised approaches including FastHash [7], SDH [1], CCA-ITQ [4] and deep method DLBHC [19]. The ResNet-50 features are also included in the comparison. The code-length of binary code from all the hashing methods is 64 bits. We use linear-SVM to evaluate the all the approaches on the classification task.

Table 3 shows the classification accuracy on test sets for the two datasets. The accuracy achieved by DBE matches that of the original continuous ResNet-50 features. DBE improves the state-of-the-art traditional methods and end-to-end approaches by 28.6% and 5.6%, respectively. And it achieves the same performance as that of the original ResNet. This demonstrates 1) DBE's superior capability of preserving the rich semantic information extracted by ResNet, 2) there exists great redundancy in the original ResNet features.

Methods	Testing Accuracy (%)
CCA-ITQ [4]	56.34
FastHash [7]	57.82
SDH [1]	67.73
DLBHC [19]	86.73
ResNet [15]	**92.38**
DBE (ours)	92.35

Table 3: The testing accuracy of different methods on CIFAR-10 dataset. All binary representations have code-length of 64 bits.

Furthermore we also provide the classification accuracy on CIFAR-10 with respect to different code lengths in Table 4. From the table we can conclude that DBE learns high-quality binary representation consistently.

Code length (bits)	16	32	48	64	128
testing acc(%)	91.63	92.04	92.20	92.35	92.36

Table 4: Classification accuracy of DBE on CIFAR-10 dataset across different code lengths

3.3. Image Retrieval

3.3.1. Natural Object Retrieval

The CIFAR-10 dataset is used to evaluate the proposed DBE on natural object retrieval task. We choose to compare with state-of-the-art image hashing algorithms including both traditional hashing methods: CCA-ITQ [4], FastHash [7], and end-to-end deep hashing methods: DSH [20], DSRH [10]. For the experimental settings, we randomly select 100 images per category and obtain a query set with 1K images. Mean average precision (mAP) is used as the evaluation metric. The comparison is reported in Table 5. The proposed DBE outperforms state-of-the-art by around 3%. It confirms that DBE is capable of preserving rich semantics extracted by the ResNet from original images and learning high-quality binary code for retrieval purpose.

Code length (bits)	12	24	36	48
CCA-ITQ [4]	0.261	0.289	0.307	0.310
FastHash [7]	0.286	0.324	0.371	0.382
SDH [1]	0.342	0.397	0.411	0.435
DSH [20]	0.616	0.651	0.661	0.676
DSRH [10]	0.792	0.794	0.792	0.792
DLBHC [19]	0.892	0.895	0.897	0.897
DBE (ours)	**0.912**	**0.924**	**0.926**	**0.927**

Table 5: Comparison of mean average precision (mAP) on CIFAR-10

3.3.2. Multilabel Image Retrieval

COCO dataset is used for multilabel image retrieval task. Considering the large number of labels in COCO, we compare DBE with several cross modal hashing and quantization algorithms. Studies have shown that cross-modal hashing improves unimodal methods by leveraging semantic information of text/label modality [21, 22]. We choose to compare with CMFH [23] and CCA-ACQ [22]. Furthermore we also include traditional hashing method CCA-ITQ [4] and end-to-end approach DHN [8]. Following the experiment protocols in [22], 1000 images are randomly sampled from validation set for query and the training set is used for database for retrieval. And AlexNet [24] feature is used as input for algorithms that are not end-to-end, and raw images are used for end-to-end deep hashing algorithms. Due to the multilabel nature of COCO, we consider the true neighbors of a query image as the retrieved images sharing at least one labels with the query. Similar to natural object retrieval, mean average precision (mAP) is used as evaluation metric.

Code length (bits)	16	24	32	48	64
CCA-ITQ [4]	0.477	0.481	0.485	0.490	0.494
CMFH [23]	0.462	0.476	0.484	0.497	0.505
CCA-ACQ [22]	0.483	0.500	0.504	0.515	0.520
DHN [8]	0.507	0.539	0.550	0.559	0.570
DBE (ours)	**0.623**	**0.657**	**0.670**	**0.692**	**0.716**

Table 6: Comparison of mean average precision (mAP) on COCO

3.4. Multilabel Image Annotation

We generate prediction of labels for each image in validation set based on K highest ranked labels and compare to the ground truth labels. The overall precision (O-P), recall (O-C), and F1-score (O-F1) of the prediction are used as evaluation metrics. Formally they are defined as:

$$\text{O-P} = \frac{N_{CP}}{N_P},\ \text{O-R} = \frac{N_{CP}}{N_G},\ \text{O-F1} = 2\frac{\text{O-P} \cdot \text{O-R}}{\text{O-P} + \text{O-R}} \qquad (10)$$

where C is the number of annotations/labels; N_{CP} is the number of correctly predicted labels for validation set; N_P is the total number of predicted labels; N_G is the total number of ground truth labels for validation set.

We compare DBE with softmax, binary cross entropy and WARP [25], one of the state-of-the-art for multilabel image annotation. The performance comparison is summarized in Table 7 and we set $K = 3$ in the experiment. It can be observed that the binary representation learned by DBE achieves the best performance in terms of overall-F1 score. Due to its consideration of co-occurrence and independence of labels, DBE-joint cross entropy outperforms DBE-softmax and DBE-weighted binary cross entropy.

Method	O-P	O-R	O-F1
WARP [25]	**59.8**	61.4	60.6
DBE-Softmax	59.1	62.1	60.3
DBE-weighted binary cross entropy	57.1	60.8	58.9
DBE-joint cross entropy	59.5	**62.7**	**61.1**

Table 7: Performance comparison on COCO for $K = 3$. The code length for all the DBE methods is 64-bit.

3.5. The Impact of DCNN Structure

Similar to most deep hashing algorithms, DBE also preserves semantics from DCNN. Consequently the structure of DCNNs influences the quality of binary code significantly. We compare with the state-of-the-art DLBHC [19] and the DCNN it uses: AlexNet [24], which the upper bound in this comparison. Since DLBHC uses AlexNet, we also use AlexNet in our DBE. CIFAR-10 dataset is used. According to results reported in Table 8, DBE achieves higher accuracy than DLBHC, i.e., DBE learns more semantic and discriminative binary representation.

Method	AlexNet [24]	DBE-AlexNet	DLBHC [19]
testing acc(%)	89.20 (upper bound)	88.52	86.73

Table 8: The comparison of the classification accuracy on the test set of CIFAR-10. Code-length for all binary algorithms is 48-bit.

4. CONCLUSION

We proposed a novel approach to learn binary representation for images in an end-to-end fashion. By using a Direct Binary Embedding layer, we are able to approximate binary code directly in DCNN. Different from existing works, DBE learns high quality binary representation for images without quantization error as a regularization. Extensive experiments on two large-scale datasets demonstrate the effectiveness superiority of DBE over state-of-the-art on several computer vision tasks including object recognition, image retrieval and multilabel image annotation.

5. REFERENCES

[1] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen, "Supervised discrete hashing," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[2] Liu Liu and Hairong Qi, "Learning effective binary descriptors via cross entropy," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 1–8.

[3] Alireza Rahimpour, Ali Taalimi, and Hairong Qi, "Feature encoding in band-limited distributed surveillance systems," in *ICASSP 2017-IEEE International Conference on Acoustics, Speech, and Signal Proceeding*. IEEE, 2017.

[4] Yunchao Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 817–824.

[5] Yair Weiss, Antonio Torralba, and Rob Fergus, "Spectral hashing," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., pp. 1753–1760. Curran Associates, Inc., 2009.

[6] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang, "Supervised hashing with kernels.," in *CVPR*. 2012, pp. 2074–2081, IEEE Computer Society.

[7] Guosheng Lin, Chunhua Shen, Qinfeng Shi, Anton van den Hengel, and David Suter, "Fast supervised hashing with decision trees for high-dimensional data," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 2014, pp. 1971–1978.

[8] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao, "Deep hashing network for efficient similarity retrieval," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, 2016, pp. 2415–2421.

[9] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen, "Deep supervised hashing for fast image retrieval," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[10] Ting Yao, Fuchen Long, Tao Mei, and Yong Rui, "Deep semantic-preserving and ranking-based hashing for image retrieval," in *IJCAI*, August 2016.

[11] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015, pp. 448–456.

[12] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*, Geoffrey J. Gordon and David B. Dunson, Eds. 2011, vol. 15, pp. 315–323, Journal of Machine Learning Research - Workshop and Conference Proceedings.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.

[14] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[16] Alex Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[17] Aude Oliva and Antonio Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vision*, vol. 42, no. 3, pp. 145–175, May 2001.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick, *Microsoft COCO: Common Objects in Context*, pp. 740–755, Springer International Publishing, Cham, 2014.

[19] Kevin Lin, Huei-Fang Yang, Jen-Hao Hsiao, and Chu-Song Chen, "Deep learning of binary hash codes for fast image retrieval," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2015.

[20] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2064–2072.

[21] Mohammad Rastegari, Jonghyun Choi, Shobeir Fakhraei, Hal Daume III, and Larry Davis, "Predictable dual-view hashing," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. 2013, p. 13281336, JMLR.

[22] Go Irie, Hiroyuki Arai, and Yukinobu Taniguchi, "Alternating co-quantization for cross-modal hashing," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[23] Guiguang Ding, Yuchen Guo, and Jile Zhou, "Collective matrix factorization hashing for multimodal data," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, 2012, pp. 1106–1114.

[25] Yunchao Gong, Yangqing Jia, Thomas K. Leung, Alexander Toshev, and Sergey Ioffe, "Deep Convolutional Ranking for Multilabel Image Annotation," in *ICLR*, 2014.