

# FAST BINARY EMBEDDING OF DEEP LEARNING IMAGE FEATURES USING GOLAY-HADAMARD MATRICES

Chanattra Ammatmanee\*, Lu GAN\* and Hongqing Liu†

\* Dept. of Electronic and Electrical Engineering, Brunel University, UK

† School of Communications and Information Engineering,

Chongqing University of Posts and Telecommunications, Chongqing, China

Chanattra.Ammatmanee@brunel.ac.uk; lu.gan@brunel.ac.uk; hongqingliu@cqupt.edu.cn

## ABSTRACT

Convolutional neural networks (CNNs) have emerged as powerful tools for image retrieval and classification. In this paper, we study binary embedding of CNN-based image descriptors for resource-constrained devices. We propose two classes of fast computable and memory-efficient dimension reduction operators using Golay-Hadamard matrices (GHMs), which are constructed by multiplying the columns of a Hadamard matrix with a Golay sequence. Simulation results on CNN-based instance image retrieval and classification show that GHM-based operators can offer competitive performance to those of full random Gaussian matrices and Gaussian circulant matrices at much lower computational cost and storage space. This implies the potential of proposed approaches on devices with low memory, bandwidth and power restrictions.

**Index Terms**— Convolutional neural network, deep learning, structured matrix, binary embedding, Hadamard matrix, Golay sequence, dimensionality reduction.

## 1. INTRODUCTION

In today's society, the large volume of multimedia data poses great challenges for tasks of visual information processing. Over the past few years, there have been increased interests in the development of convolutional neural networks (CNNs) for multimedia applications, e.g., [1–3] and the references therein. Compared with conventional methods, CNN-based approaches yield higher accuracy for image retrieval and classification. Feature vectors extracted from CNN often have hundreds or thousands of dimensions. Consequently, for devices with power, buffer and bandwidth constraints, data transmission and storage could be challenging using these high dimensional data. To provide short feature vectors, dimensionality reduction is often applied to post-process the CNN feature vectors. One well-known method is principal component analysis (PCA), in which the projection matrix was first learnt from an independent data-set [1, 2]. However, the storage and computation cost in PCA is very expensive.

For example, in NetVLAD CNN architecture [1], the corresponding feature vector learnt from VGG neural network has a dimension of 32k. To reduce the dimension to 4k, a  $4k \times 32k$  projection matrix requires about 512 MB of memory, which is too large for mobile applications. Besides, the multiplication of such a projection matrix is time-consuming.

Various deep hashing methods [4–11] have also been investigated to generate binary codes directly from deep learning neural networks. Binary codes are attractive as they are more memory efficient. They also allow fast query by computing Hamming distance in binary space. However, most of the existing works focused on supervised hashing. Besides, binary codes produced by unsupervised hashing methods are often inferior to real-coefficient CNN descriptors.

In this paper, we propose the binary embedding of deep learning features through *fast* Johnson-Lindenstrauss transform (JLT) [12] followed by simple binary operations. In particular, the fast JLTs are constructed from Golay-Hadamard matrices (GHMs) by modulating columns of a Hadamard matrix with a Golay sequence. The development is based on two facts. First, we found a CNN-based image descriptor is often sparse (or compressible) in the wavelet domain. Second, previous work shows that a Golay sequence is effective in Hadamard-based compressive imaging [13]. Especially, two classes of fast JLT are proposed: (i) *GHM-Rand* is constructed by randomly selecting  $M$  rows of an  $N \times N$  GHM ( $N \geq M$ ); (ii) *GHM-Fix* picks up the first  $M$  rows of a cascade of two different GHMs. Compared with existing fast JLTs such as Gaussian circulant matrices (GCMs) [14], our proposed operators have reduced randomness and computation complexity. Simulation results show that they can produce comparable results to those of full random Gaussian matrices and GCMs in image retrieval and classification.

The rest of this paper is organized as follows. Section 2 briefly reviews binary embedding using full Gaussian matrices and GCMs. Section 3 presents the proposed dimensionality reduction operator using GHMs. Extensive simulation results on image retrieval and classification are given in Section 4, followed by conclusion and future work in Section 5.

978-1-6654-3864-3/21/\$31.00 ©2021 IEEE

## 2. LITERATURE REVIEW

Consider a data set  $\mathcal{T} \subset \mathbb{R}^N$ . We want to embed each  $\mathbf{x}_i \in \mathcal{T}$  into  $M \leq N$  bits so that for any two points  $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{T}$ , their pairwise Euclidean (or angular) distances can be preserved. A typical way is to first multiply each  $\mathbf{x}_i$  with an  $M \times N$  data-independent projection matrix  $\mathbf{A}$ , and then map  $\mathbf{A}\mathbf{x}$  to binary bits  $\{1, -1\}^M$  using a given function. In what follows, we provide a brief review of binary embedding using independently identically distributed (i.i.d.) Gaussian random matrices and Gaussian circulant matrices (GCMs).

Recall that the classical Johnson-Lindenstrauss Lemma [12] states that if  $\mathbf{A}$  has independent Gaussian entries with zero-mean and unit variance, then with very high probability, the following inequality holds for  $\epsilon > 0$  and all pairs of points  $\mathbf{x}_i, \mathbf{x}_j$  in a finite data set  $\mathcal{T}$  [12]

$$\left| \left\| \frac{1}{\sqrt{M}} \mathbf{A}(\mathbf{x}_i - \mathbf{x}_j) \right\|_2^2 - \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \right| \leq \epsilon \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \quad (1)$$

provided that  $M \geq \mathcal{O}(\epsilon^{-2} \log Q)$ , in which  $Q$  denotes the total number of points in  $\mathcal{T}$ .

A simple binary operation is to take the sign of  $\mathbf{A}\mathbf{x}$ , as given below

$$f(\mathbf{x}) = \text{sign}(\mathbf{A}\mathbf{x}) \quad (2)$$

where the  $\text{sign}(\cdot)$  denotes the element-wise sign operation. If each data point is on the unit-sphere, i.e.,  $\|\mathbf{x}_i\|_2 = 1$  ( $1 \leq i \leq Q$ ), the angular distances of  $\mathbf{x}_i$  and  $\mathbf{x}_j$  can be evaluated or estimated using the normalized Hamming distances of  $f(\mathbf{x}_i)$  and  $f(\mathbf{x}_j)$ . That is, for  $\delta > 0$ , when  $\mathbf{A}$  is a standard i.i.d. Gaussian matrix with  $M \geq \mathcal{O}(\delta^{-2} \log(N/\eta))$ , the following holds with probability  $1 - \eta$  [15, 16]

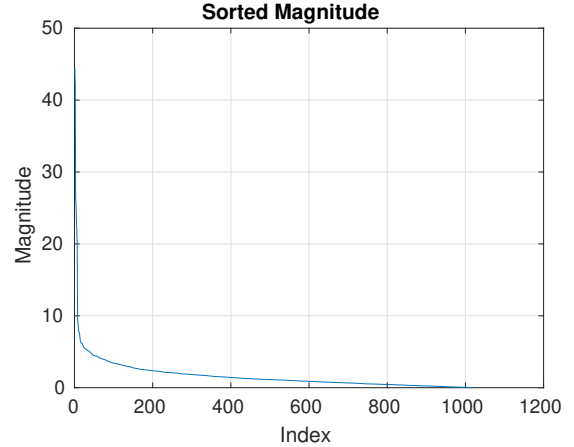
$$\left| \frac{1}{M} d_H(f(\mathbf{x}_i), f(\mathbf{x}_j)) - \frac{1}{\pi} \arccos(\langle \mathbf{x}_i, \mathbf{x}_j \rangle) \right| \leq \delta \quad (3)$$

It is also known that this bound is *optimal* in terms of bit complexity  $M$  [15]. It should be pointed out that the above result is for *any* finite data-set. The bound for  $M$  can be further improved if the data-sets have low-complexity structure. For example, when each  $\mathbf{x}_i$  is a sparse vector with  $s$  non-zero elements, (3) holds when  $M \geq \mathcal{O}(\delta^{-2} s \log(eN/s))$  even for the infinite set.

Although i.i.d. Gauss matrices are theoretically optimal, they require high computational and storage costs. Over the past decade, the design of fast JLT for binary embedding has been an active area. Several operators have been proposed such as fast binary embedding [16] and circulant binary embedding [14]. As we aim to build low-complexity operators for practical applications, we provide a brief review on the construction of  $\mathbf{A}$  from a GCM with the following form [14, 17]

$$\mathbf{A} = \mathbf{R}_{\mathcal{I}} \mathbf{C}_{\nu} \mathbf{D}_{\zeta} \quad (4)$$

where  $\mathcal{I}$  is a fixed subset of  $\{1, 2, \dots, N\}$  with a size of  $M$ , and  $\mathbf{R}_{\mathcal{I}}$  is a subsampling operator that restricts a vector to



**Fig. 1:** An example of sorted Haar-wavelet magnitudes of a feature vector extracted from Resnet-50.

its entries indexed by  $\mathcal{I}$ .  $\mathbf{C}_{\nu}$  is a circulant matrix generated by a standard Gaussian vector  $\nu$  and  $\mathbf{D}_{\zeta}$  is a diagonal matrix with random sign vector  $\zeta$  on its diagonal, i.e.,  $\Pr(\zeta_i = 1) = \Pr(\zeta_i = -1) = 0.5$ . This class of matrices is attractive as its matrix-vector multiplication can be computed with only  $\mathcal{O}(N \log N)$  operations [14].

Circulant binary embedding using (2) and (4) was investigated in [14] for vectors on the unit sphere. If the maximum magnitude of  $\mathbf{x}_i$  is small, i.e.,  $\|\mathbf{x}_i\|_{\infty} = \mathcal{O}(\log N / \sqrt{N})$ , the Hamming distance  $d_H(f(\mathbf{x}_i), f(\mathbf{x}_j))$  can be used to estimate their angular distances. For an arbitrary data-set, one can preprocess each data vector by multiplying it with a randomly modulated Hadamard matrix. That is,  $\mathbf{A}$  can be replaced by  $\mathbf{A} = \mathbf{R}_{\mathcal{I}} \mathbf{C}_{\nu} \mathbf{D}_{\zeta} \mathbf{H} \mathbf{D}_{\kappa}$ , where  $\mathbf{H}$  is an  $N \times N$  Hadamard matrix and  $\kappa$  is a random sign vector.

## 3. BINARY EMBEDDING USING GOLAY-HADAMARD MATRICES

### 3.1. Problem formulation

Our goal here is to design memory efficient and fast-computable binary embedding for CNN features. Note that for fast locality sensitivity hashing (LSH), it was shown empirically [18] that a full random matrix can be approximated by  $\mathbf{A} = \mathbf{R}_{\mathcal{I}} \mathbf{H} \mathbf{D}_{\zeta_2} \mathbf{H} \mathbf{D}_{\zeta_1} \mathbf{H} \mathbf{D}_{\zeta_0}$  without much performance degradation in approximate nearest neighbour search, where  $\zeta_i$  ( $i = 0, 1, 2$ ) are random sign vectors. This construction is for *arbitrary* finite data-set. When the data-set has intrinsic low-complexity structure, more efficient dimension operator can be used [19].

Interestingly, we found that CNN-based image descriptors are still “compressible” in the wavelet domain. As a quick demo, Figure 1 shows the sorted wavelet magnitudes for a CNN feature vector. Here, the off-the-shelf feature

**Table 1:** Comparison of number of random coefficients and computation costs for different operators

Operators	No. of Random Coefficients		No. of Operations	
	Floats	Binaries	Multiplications	Additions
i.i.d. Gauss	$\mathcal{O}(MN)$	0	$\mathcal{O}(MN)$	$\mathcal{O}(MN)$
GCM [14]	$N$	$N$	$\mathcal{O}(N \log N)$	$\mathcal{O}(N \log N)$
GHM-Rand	0	$N$	0	$\mathcal{O}(N \log M)$
GHM-Fix	0	0	0	$\mathcal{O}(N \log N)$

is extracted from the last fully-connected layer of Resnet-50 [20] with dimension length of  $N = 1000$ . 8-level Haar wavelet decomposition is then applied. As one can see, most of the signal’s energy concentrates only on a few large wavelet coefficients. Similar results were found for other CNN-architectures. Based on this observation, we aim to design  $\mathbf{A}$  using only 1 or 2 stages of  $\mathbf{HD}_{\zeta_i}$  with *deterministic* sign vectors  $\zeta_i$  ( $i = 0, 1$ ).

### 3.2. Golay-Hadamard matrices for binary embedding

According to [13], a partial Hamdard matrix modulated by a Golay sequence can offer near-optimal bound for the compressive sampling of sparse signals in the wavelet domain [13]. The definition of a Golay sequence is given below [21]:

**Definition 1** Consider two length- $N$  binary sequences  $\mathbf{a} = [a_0, a_1, \dots, a_{N-1}]$  and  $\mathbf{b} = [b_0, b_1, \dots, b_{N-1}]$ . Define two polynomials  $A(z) = \sum_{n=0}^{N-1} a_n z^n$  and  $B(z) = \sum_{n=0}^{N-1} b_n z^n$ .  $\mathbf{a}$  and  $\mathbf{b}$  are said to be a Golay complementary pair (GCP) if

$$|A(z)|^2 + |B(z)|^2 = 2N, \quad (5)$$

for all  $|z| = 1$ .  $\mathbf{a}$  (or  $\mathbf{b}$ ) is called as a Golay sequence [21].

From (5), it is clear that  $|A(z)| \leq \sqrt{2N}$  for all  $z = e^{j\omega}$  ( $0 \leq \omega < 2\pi$ ), which means that a Golay sequence is nearly flat in the spectrum domain, i.e., it is a binary pseudo-random sequence. A GCP can be constructed directly or recursively. When  $N = 2^n$ , the methods in [21] can produce  $N \cdot n!$  different Golay sequences. One popular way is through the Golay-Rudin-Shapiro recursion formula [21]:

$$\mathbf{a}^{(0)} = 1, \quad \mathbf{b}^{(0)} = -1 \quad (6)$$

$$\mathbf{a}^{(l)} = [\mathbf{a}^{(l-1)}, \mathbf{b}^{(l-1)}] \quad \mathbf{b}^{(l)} = [\mathbf{a}^{(l-1)}, -\mathbf{b}^{(l-1)}] \quad (7)$$

for  $l = 1, 2, \dots, n - 1$ . Unlike a random sign vector, each element  $a_i$  in a Golay sequence has an explicit form. Hence, it can be easily implemented in both software and hardware without storing the whole sequence. Next, we define Golay-Hadamard matrix (GHM):

**Definition 2** A Golay Hadamard matrix (GHM) is given by  $\mathbf{G} = \mathbf{HD}_{\zeta}$ , where  $\mathbf{H}$  is the Hadamard matrix and  $\mathbf{D}_{\zeta}$  is a diagonal matrix with Golay sequence  $\zeta$  on its diagonal.

It can be shown that each row in a GHM is still a Golay sequence [13, 21]. Thus,  $\mathbf{G}\mathbf{x}$  calculates the inner products of  $\mathbf{x}$  with  $N$  orthogonal binary pseudo-random sequences. Below, we propose two classes of  $M \times N$  dimensionality reduction operators  $\mathbf{A}$  constructed from GHMs:

- GHM-Rand:

$$\mathbf{A} = \frac{1}{\sqrt{M}} \mathbf{R}_{\Omega} \mathbf{G}_0, \quad (8)$$

- GHM-Fix:

$$\mathbf{A} = \frac{1}{\sqrt{NM}} \mathbf{R}_{\mathcal{I}} \mathbf{G}_1 \mathbf{G}_0 \quad (9)$$

where  $\mathbf{G}_i$  ( $i = 0, 1$ ) are GHMs,  $\mathbf{R}_{\Omega}$  and  $\mathbf{R}_{\mathcal{I}}$  represent the sub-sampling operator similar as that in (4). In particular,  $\Omega$  in (8) is a uniform random subset of  $\{1, 2, \dots, N\}$  with size of  $|\Omega| = M$ ; and in (9),  $\mathcal{I}$  is a fixed set with  $\mathcal{I} = \{1, 2, \dots, M\}$ .

Assume that the feature vectors  $\mathbf{x}_i$  are strictly sparse in the Haar-transform domain with only  $s$  non-zero coefficients each, i.e.,  $\|\mathbf{W}\mathbf{x}_i\|_0 \leq s$  for all  $\mathbf{x}_i \in \mathcal{T}$ , in which  $\mathbf{W}$  denotes the Haar wavelet transform matrix. For GHM-Rand in (8), when the Golay sequence is constructed from the Golay-Rudin-Shapiro recursion using (6) and (7), (1) holds with high probability when  $M \geq \mathcal{O}(\epsilon^{-2}s \log^3(2s) \log N)$  [13]. This implies that GHM-Rand can be used as a fast JLT to embed wavelet-domain sparse CNN feature vectors *without* any binarization. At this stage, we are not sure how to get rigorous proof with binary function  $f(\mathbf{x})$ . But numerical results in the next Section show they can offer excellent performance for image retrieval and classification.

Note that GHM-Rand in (8) requires only one stage of a partial GHM. According to [22], multiplication of such a matrix requires only  $\mathcal{O}(N \log M)$  additions, which is computationally efficient. It is also memory-efficient as the randomness only comes from  $\Omega$ . In the particular case when  $M = N$ , GHM-Rand becomes a deterministic operator as well. GHM-Fix in (9) is completely deterministic, which includes a cascade of 2 different GHMs requiring  $\mathcal{O}(N \log N)$  operations. Table 1 compares our proposed GHM-Rand and GHM-Fix operators with i.i.d. Gauss and GCMs for memory requirement and computational cost. As can be seen here, our proposed systems have reduced randomness and lower computational complexity.

## 4. SIMULATION RESULTS

To evaluate the performance of the proposed operators, simulations were carried out for different CNN-architectures and different image data-sets in image retrieval and classification.

### 4.1. Results on unsupervised image retrieval

For unsupervised instance-level image retrieval, the test data-sets are the standard Oxford 5k [23] and Paris 6k [24] along with their revised versions ROxford 5k and RParis 6k [25]. The CNN architecture is based on the Resnet101-AP-GeM [2, 26] trained from Resnet101 [20] on Landmarks-full data-set [27]. The authors' source codes on GitHub [28] were used for evaluation. In the original setting [28], each CNN feature has a length of  $N = 2048$  with each float coefficient stored in 32-bits, i.e.,  $2048 \times 32 = 65536$  bits altogether. Binary embedding using i.i.d. Gauss, GCM [14], GHM-Rand and GHM-Fix are applied for the original CNN feature vectors. Hamming distance is then calculated for approximate nearest neighbour search. For comparison purpose, results of dimension reduction using PCA with whitening ( $PCA_w$ ) are also included, where the whitening operator was also learnt from the Landmarks-full data-set [2]. To produce  $M = 32d$ -bits, the  $d \times N$   $PCA_w$  operator is used with whitening power of 0.25. The performance is measured via standard mean average precision (mAP) [2]. Following the convention, only the annotated region of interests are used. The results are shown in Table 2. As one can see from here, the performance of  $PCA_w$  drops quickly with the decrease of  $M$ . Despite their low-complexity, our proposed operators offer the best performances in nearly all cases for a given  $M$  except for a couple of cases, in which GHM-Rand and GHM-Fix are slightly worse than those of the i.i.d. Gauss matrix. In fact, when  $M = 2048$  bits, GHM-Rand becomes a fixed operator and its performances are still very to those of the original one with 65536 bits (2048 floats). Although GHM-Fix is a *data oblivious and deterministic* operator, it offers very similar performance to those of full i.i.d. Gauss matrix and GCMs.

Table 3 further compares the results of our proposed systems with different state-of-the-art binary codes obtained from CNN networks. Specifically, we compare with supervised methods such as supervised semantics-preserving deep hashing (SSDH) [9], hierarchical deep hashing (HDH) [7] and un-supervised ones such as Deepbit [8], pixels to binary (P2B) codes [10] and embedding and aggregation on selective convolution features (EASC) [11]. One can observe the proposed GHM-Rand operator offers the best performance except for Paris 6k at 256 bits. The performance of GHM-Fix is also very close to that of GHM-Rand. These benchmark methods often require much more complicated training and/or post-processing to produce binary codes. On the hand, our proposed system only requires simple multiplication of CNN descriptors with GHM(s) followed by a sign operation.

### 4.2. Results on image classification

The classification performance was tested on two publicly available databases, Caltech101 (9,146 images) with 101 object categories [29] and Caltech256 (30,607 images) with 256 object categories [30]. The off-the-shelf feature vectors were extracted from the last pooling layer (pool5 –  $7 \times 7 - s1$ ) of Googlenet [31]. 70% of images were used during the model training process and the remaining 30% were for testing. Matlab 2020a's default linear support vector machine (SVM) classifier was used to train and classify on binary codes directly. The results are documented in Table 4. Here, 50 trials were performed for each dimensionality reduction operator with random coefficients. As one can see, our proposed GHM-Rand and GHM-Fix achieve similar performance on image classification for all bit length  $M$  on both data-sets.

## 5. CONCLUSION

In this paper, we study binary embedding of CNN feature vectors for low-powered, low-buffered devices, e.g., mobile or Internet of things (IoT) devices. Two fast dimensionality reduction operators were proposed based on Golay-Hadamard matrices (GHMs). In particular, to embed an  $N$ -dimensional feature vector into  $M$  bits, GHM-Rand requires  $N$  random binary bits and  $\mathcal{O}(M \log N)$  additions along with  $N$  sign flipping operations. GHM-Fix is completely deterministic with  $\mathcal{O}(N \log N)$  additions and  $2N$  sign flipping operations. To demonstrate the effectiveness of the proposed operators, simulation results were carried out for image instance retrieval and image classification on some popular image data-sets. Despite their low-complexity in both computation and storage, GHM-Rand and GHM-Fix offer competitive (or even better) performance to PCA, full random Gaussian matrices and Gaussian circulant operators. This indicates their promising potentials in practical mobile or IoT applications. In future, we aim to develop solid theoretical proofs and explore their use in video retrieval and classification.

## 6. REFERENCES

- [1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: CNN architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018.
- [2] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *Int. J. Comput. Vis.*, vol. 124, no. 2, pp. 237–254, 2017.
- [3] W. Min, S. Mei, Z. Li, and S. Jiang, "A two-stage triplet network training framework for image retrieval," *IEEE Transactions on Multimedia*, vol. 22, no. 12, pp. 3128–3138, 2020.

**Table 2:** Comparison of mean average precision (mAP) for different dimensionality reduction operators. For RParis 6k and ROxford 5k, “E”, “M” and “H” represent the easy, medium and hard subsets, respectively [25]. Bold values indicate the best results for a given  $M$  and data-set.

No. of Bits $M$	Method	Paris 6k	RParis 6k			Oxford 5k	ROxford 5k		
			E	M	H		E	M	H
65536	Original [2]	92.95	90.76	80.31	60.86	89.12	83.28	67.13	42.26
2048	PCA <sub>w</sub> [2]	88.36	86.38	75.94	54.15	79.88	71.16	56.30	27.34
	i.i.d. Gauss	91.28	<b>89.87</b>	77.34	55.93	87.05	81.72	64.31	38.56
	GCM [14]	91.87	89.70	77.51	56.70	84.94	79.30	62.82	36.78
	GHM-Rand	<b>92.20</b>	89.74	<b>78.65</b>	<b>58.46</b>	<b>87.95</b>	<b>81.95</b>	<b>65.64</b>	<b>40.73</b>
	GHM-Fix	91.81	89.78	78.49	58.01	85.89	80.83	63.41	37.61
1024	PCA <sub>w</sub> [2]	82.26	78.42	63.39	45.55	75.26	64.66	50.51	23.00
	i.i.d. Gauss	90.51	85.69	74.50	53.31	84.02	<b>78.65</b>	59.92	32.87
	GCM [14]	89.98	87.71	74.20	51.59	82.23	73.65	57.77	32.08
	GHM-Rand	<b>90.92</b>	<b>88.55</b>	<b>76.04</b>	<b>54.61</b>	<b>84.85</b>	77.75	<b>61.54</b>	35.29
	GHM-Fix	90.02	87.76	75.21	53.39	84.69	77.06	61.12	<b>35.45</b>
512	PCA <sub>w</sub> [2]	71.97	63.96	57.02	35.36	55.79	42.56	31.95	8.62
	i.i.d. Gauss	86.59	84.56	69.32	44.65	78.60	68.93	52.77	26.08
	GCM [14]	85.84	83.56	69.03	44.53	79.76	68.76	51.95	26.69
	GHM-Rand	<b>87.53</b>	<b>85.31</b>	<b>71.57</b>	46.68	<b>81.93</b>	<b>73.18</b>	<b>57.38</b>	30.42
	GHM-Fix	86.83	84.40	70.34	<b>46.74</b>	80.43	71.52	56.10	<b>31.64</b>

**Table 3:** A mAP comparison of binary code retrieval with supervised and unsupervised hashing methods on Paris 6k and Oxford 5k data-sets. Bold values indicate the best results for a given  $M$  and data-set.

Method		Paris 6K			Oxford 5K		
		1024 bits	512 bits	256 bits	1024 bits	512 bits	256 bits
Supervised Hashing	SSDH [9]	—	83.87	—	—	63.80	—
	HDH [7]	—	87.30	<b>85.20</b>	—	70.50	69.70
Unsupervised Hashing	Deepbit [8]	—	82.90	82.50	—	62.70	60.30
	P2B [10]	—	—	—	—	74.84	69.20
	EASC [11]	—	79.10	74.10	—	68.90	58.50
Proposed	GHM-Rand	<b>90.92</b>	<b>87.53</b>	78.94	<b>84.85</b>	<b>81.93</b>	<b>70.00</b>
	GHM-Fix	90.02	86.83	78.70	84.69	80.43	69.35

- [4] V. Nguyen and M. N. Do, “Deep learning based supervised hashing for efficient image retrieval,” in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, 2016, pp. 1–6.
- [5] J. Bai, B. Ni, M. Wang, Z. Li, S. Cheng, X. Yang, C. Hu, and W. Gao, “Deep progressive hashing for image retrieval,” *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3178–3193, 2019.
- [6] H. Zhang, Y. Gu, Y. Yao, Z. Zhang, L. Liu, J. Zhang, and L. Shao, “Deep unsupervised self-evolutionary hashing for image retrieval,” *IEEE Transactions on Multimedia*, pp. 1–1, 2020.
- [7] G. Song and X. Tan, “Hierarchical deep hashing for image retrieval,” *Frontiers Comput. Sci.*, vol. 11, no. 2, pp. 253–265, 2017.
- [8] K. Lin, J. Lu, C.-S. Chen, J. Zhou, and M.-T. Sun, “Unsupervised deep learning of compact binary descriptors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1501–1514, 2019.
- [9] H.-F. Yang, K. Lin, and C.-S. Chen, “Supervised learning of semantics-preserving hash via deep convolutional neural networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 437–451, 2018.
- [10] T.-T. Do, T. Hoang, D.-K. L. Tan, T. Pham, H. Le, N.-M. Cheung, and I. D. Reid, “Binary constrained deep hashing network for image retrieval without manual annotation,” in *IEEE Winter Conference on Applications of Computer Vision WACV Waikoloa Village, USA*, 2019, pp. 695–704.
- [11] T.-T. Do, T. Hoang, D.-K. L. Tan, H. Le, T. V. Nguyen, and N.-M. Cheung, “From selective deep convolutional features to compact binary representations for image retrieval,” *ACM Trans. Multim. Comput. Commun. Appl.*, vol. 15, no. 2, pp. 43:1–43:22, 2019.

**Table 4:** Multi-class classification accuracy (%) on Googlenet features for Caltech101 and Caltech256 data-sets. The best mean accuracy is highlighted in bold.

No. of bits	Method	Caltech101	Caltech256
16384	Original	90.31	98.88
1024	i.i.d. Gauss	86.80± 0.32	99.69± 0.20
	GCM [14]	87.24± 0.31	99.64 ± 0.21
	GHM-Rand	86.81	<b>99.74</b>
	GHM-Fix	<b>87.75</b>	<b>99.74</b>
512	i.i.d. Gauss	84.00± 0.37	97.12 ± 0.20
	GCM [14]	<b>84.48± 0.36</b>	97.14± 0.22
	GHM-Rand	83.4± 0.32	97.43 ± 0.19
	GHM-Fix	<b>84.49</b>	<b>97.47</b>
256	i.i.d. Gauss	78.59± 0.38	84.00 ± 0.19
	GCM [14]	78.20± 0.32	84.49 ± 0.21
	GHM-Rand	79.41± 0.28	84.93 ± 0.19
	GHM-Fix	<b>80.13</b>	<b>84.96</b>

- [12] W. Johnson and L. Lindenstrauss, “Extensions of Lipschitz mappings into Hilbert space,” *Contemporary mathematics*, vol. 26, pp. 189–206, 1984.
- [13] L. Gan, L. Liu, and Y. Shen, “Golay sequence for partial Fourier and Hadamard compressive imaging,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6048–6052.
- [14] F. Yu, A. Bhaskara, S. Kumar, Y. Gong, and S. Chang, “On binary embedding using circulant matrices,” *J. Mach. Learn. Res.*, vol. 18, pp. 150:1–150:30, 2017.
- [15] Y. Plan and R. Vershynin, “Dimension reduction by random hyperplane tessellations,” *Discret. Comput. Geom.*, vol. 51, no. 2, pp. 438–461, 2014.
- [16] X. Yi, C. Caramanis, and E. Price, “Binary embedding: Fundamental limits and fast algorithm,” in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France*, 2015, vol. 37, pp. 2162–2170.
- [17] A. Choromanska, K. Choromanski, M. Bojarski, T. Jebara, S. Kumar, and Y. LeCun, “Binary embeddings with structured hashed projections,” in *Proceedings of the 33rd International Conference on Machine Learning (ICML), New York City, USA*, 2016, vol. 48, pp. 344–353.
- [18] A. Andoni, P. Indyk, T. Laarhoven, I-P. Razenshteyn, and L. Schmidt, “Practical and optimal LSH for angular distance,” in *Annual Conference on Neural Information Processing Systems 2015, Montreal, Canada*, 2015, pp. 1225–1233.
- [19] S. Dirksen and A. Stollenwerk, “Binarized Johnson-Lindenstrauss embeddings,” Available at <https://arxiv.org/abs/2009.08320> (Sept. 2020).
- [20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [21] M. Golay, “Complementary series,” *IEEE Trans. Inf. Theory*, vol. 7, no. 2, pp. 82–87, Apr. 1961.
- [22] N. Ailon and E. Liberty, “Fast dimension reduction using Rademacher series on dual BCH codes,” *Discret. Comput. Geom.*, vol. 42, pp. 615, 2009.
- [23] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” in *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007)*. 2007, IEEE Computer Society.
- [24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Lost in quantization: Improving particular object retrieval in large scale image databases,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.
- [25] F. Radenovic, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, “Revisiting Oxford and Paris: Large-scale image retrieval benchmarking,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA*, 2018, pp. 5706–5715.
- [26] Jérôme Revaud, Jon Almazán, Rafael S. Rezende, and César Roberto de Souza, “Learning with average precision: Training image retrieval with a listwise loss,” in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, South Korea (South)*. 2019, pp. 5106–5115, IEEE.
- [27] Artem Babenko, Anton Slesarev, Alexander Chigorin, and Victor S. Lempitsky, “Neural codes for image retrieval,” in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland*. 2014, vol. 8689, pp. 584–599, Springer.
- [28] J. Almazan and Y. Cabon, “Deep image retrieval,” Available at <https://github.com/naver/deep-image-retrieval> (2019).
- [29] F-F. Li, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories, washington, usa,” in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 2004, pp. 178–178.
- [30] G. Gregory, H. Alex, and P. Pietro, “Caltech-256 object category dataset,” Available at [http://www.vision.caltech.edu/Image\\_Datasets/Caltech256/](http://www.vision.caltech.edu/Image_Datasets/Caltech256/) (2007).
- [31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.