Deep Unsupervised Image Hashing by Maximizing Bit Entropy

Yunqiang Li and Jan van Gemert
Vision Lab, Delft University of Technology, Netherlands
{y.li-19, j.c.vangemert}@tudelft.nl

Group 2: 2023/01/02

資訊系114 黃芊
資訊系114 **聶怡安**
資訊系114 **楊凱茗**
校際選課 莊上儀

Content

1.1 Introduction: Motivation and Objective

1. Motivation:
   - Unsupervised hashing 是一項可以用較低成本來將圖像、影片 進行分群的方法
   - Hashing 的功能是將資料壓縮但是又不失圖片 重要的資訊

2. Objective:
   - 我們創造了一個unsupervised hash deep layer，命名為Bi-half Net
   - 此layer為一個簡單, **無須額外參數**的layer, 他可以保留圖片最多的資訊
   - 在壓縮圖片時, 會將連續數值轉離散, 為了達到最佳化, 我們將 Wasserstein distance 最小化

2.1 Related Works : Methods and Problems

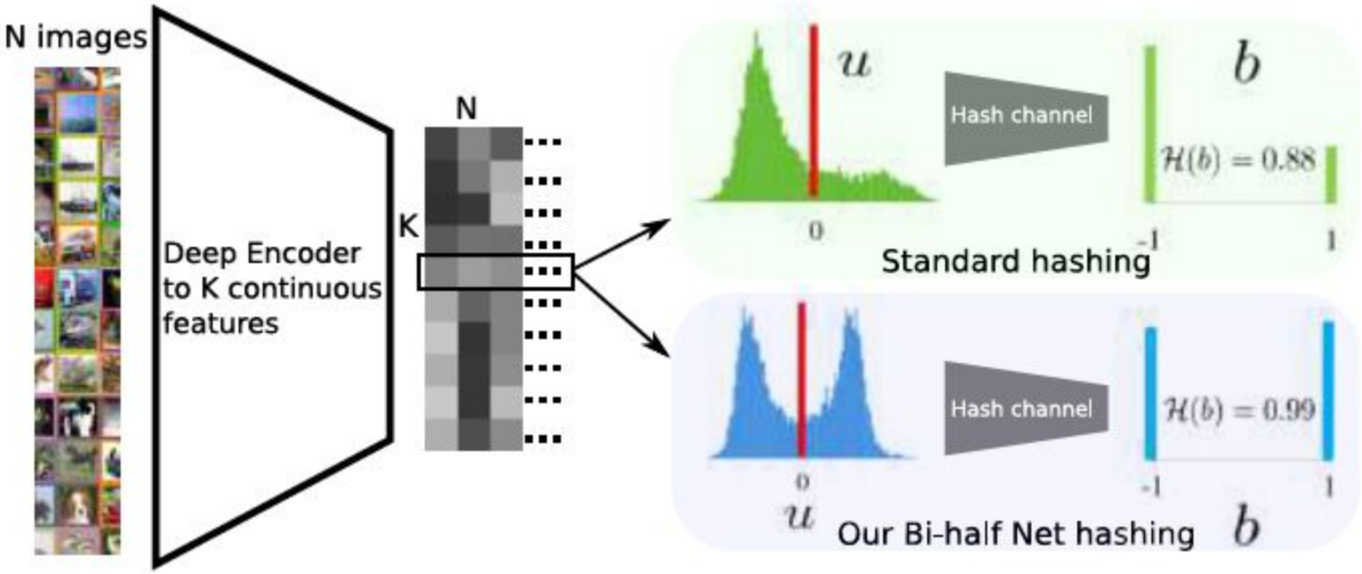
Amount of Supervision	**Quantization from continuous to discrete values.**	**Obtaining gradients for binary codes**	**Information theory in hashing**
supervised的結果很好，但是需要人為預處理(labeled)，很花錢和時間	**當前的Sign function並不能有效利用所有離散資訊，一位會全部分為-1 或是+1**	discrete hash output codes無法直接應用在continuous的activation function中，greedy hash則可能會產生redundant codes	**當前大多數人都是在loss function中加上additional term 以將bit entropy最大化，在不同term之間很難統整，需要有足夠精細的調整**

2.2 Related Works : Contribution

Amount of Supervision	Quantization from continuous to discrete values.	Obtaining gradients for binary codes	Information theory in hashing
專注於處理unsupervised的設定，以減少時間與金錢的額外損耗	利用max bit capacity來提高資訊的使用率	用像是greedy hash 的概念： straight-through estimator，同時**maximizing bit information capacity**	設計一個**新的network layer**，不用設loss function的addition參數，只需在特定的distribution中將 Wasserstein distance最小化

3.1 Approach : System Framework

illustrates the concept of hashing, compressing N images into K bits per image. We propose a novel Bi-half layer that is parameter-free, aiming to maximize the information capacity of the hash channel. This is achieved by minimizing the Wasserstein distance to align continuous features with the optimal discrete distribution.



3.2 Approach : Maximizing hash channel capacity

- 將連續特徵 U 轉換成 B需要透過一個 lossy communication channel
- 每個 channel 的容量 (channel capacity) C，取決於從 U 轉換成 B 的最大資料量

$$C = \max_{p(u)} I(U;B)$$

- 根據 p(u) 以及 I(U ; B) 去找尋最大值
- p(u) : 所有可能的 input distribution
- I(U ; B) : mutual information between U and B

3.2 Approach : Maximizing hash channel capacity

- I (U ; B) = $H(B) - H(B \mid U)$
- $H(B)$: entropy of B
- $H(B \mid U)$: conditional entropy of B based on U
- 所以 最大化 C 其實就是 最大化 $H(B)$且最小化 $H(B \mid U)$

3.3 Approach : Bi-half layer for quantization

- 使用 Optimal Transport 讓連續特徵分布可以接近理想的 half-half
- 為了找到在兩個機率分布P_r , P_g之間，將單位質量從 x 移動到 y 的最小花費
- π : 有多少質量可以從P_r傳輸到 P_g
- D : cost function matrix, each element ≥ 0

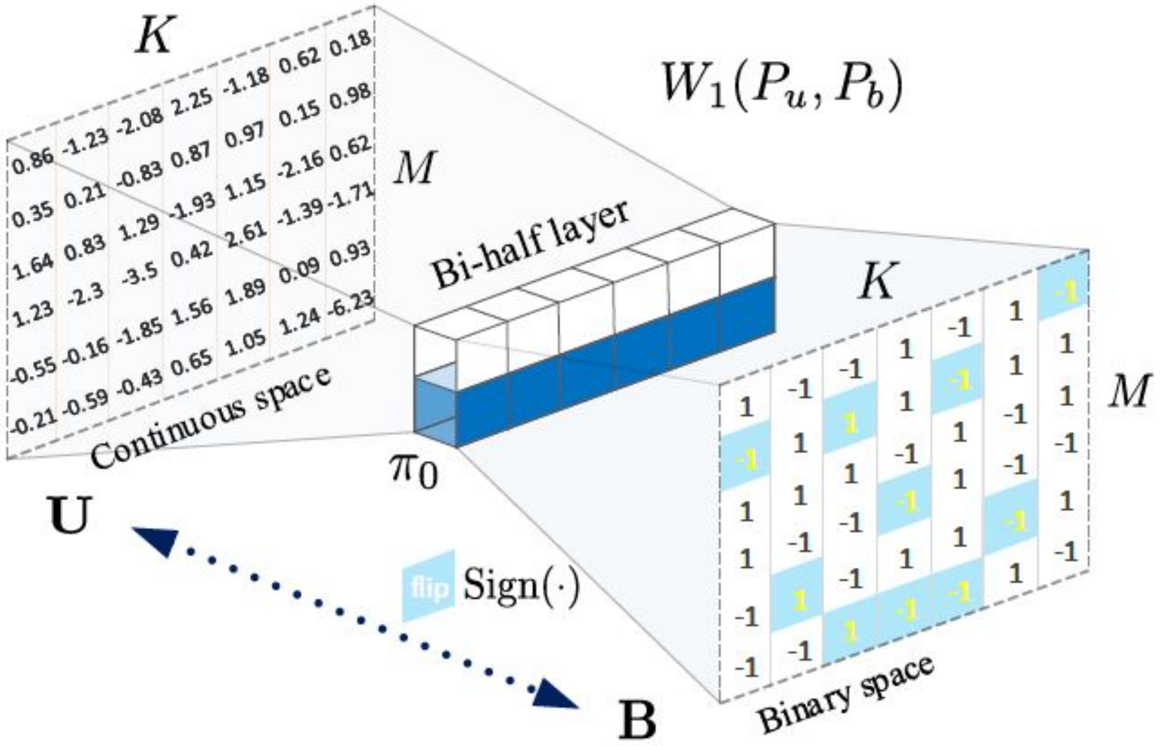
$$\pi_0 = \min_{\pi \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \langle \pi, \mathbf{D} \rangle_F$$

3.3 Approach : Bi-half layer for quantization

- 如果把 OT 的 D 解釋成距離，其實就是 Wasserstein distance
- 這篇論文將距離 D(i, j) = d(x, y) 取 squared Euclidean distance 讓後續的運算以及優化更為順暢

3.3 Approach : Bi-half layer for quantization

- 將此層叫做 bi-half layer
- 將連續變數量子化成 half-half distribution



4.1 Data Collection : Datasets

- Flickr25k
- Nus-wide, Nus-wide(I), Nus-wide(II)
- Cifar-10
- Mscoco
- Mnist
- Ucf-101
- Hmdb-51

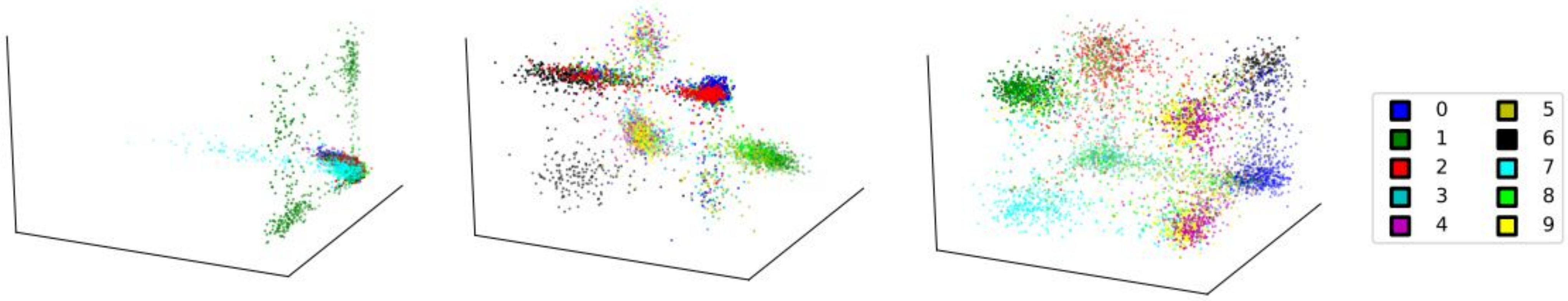
4.2 Data Collection : Implementation Details

- For MNIST
  - Training an AutoEncoder from scratch
- For other images
  - VGG-16 as backbone
  - Bi-half append to generate binary code
  - Use SGD, momentum = 0.9, weight decay = 5*10^(-4), batch size = 32
  - Learning rate = 0.0001
  - Divided by 10 when the loss stop decreasing
  - Hyper-parameters γ is tuned by cross-validation on training set and set as $\gamma = 3 * (1/N*K)$

Evaluation Metrics

- Mean Average Precision (mAP)
- Precision-Recall curves (PR)
- TopN-precision curves (N = 5000)

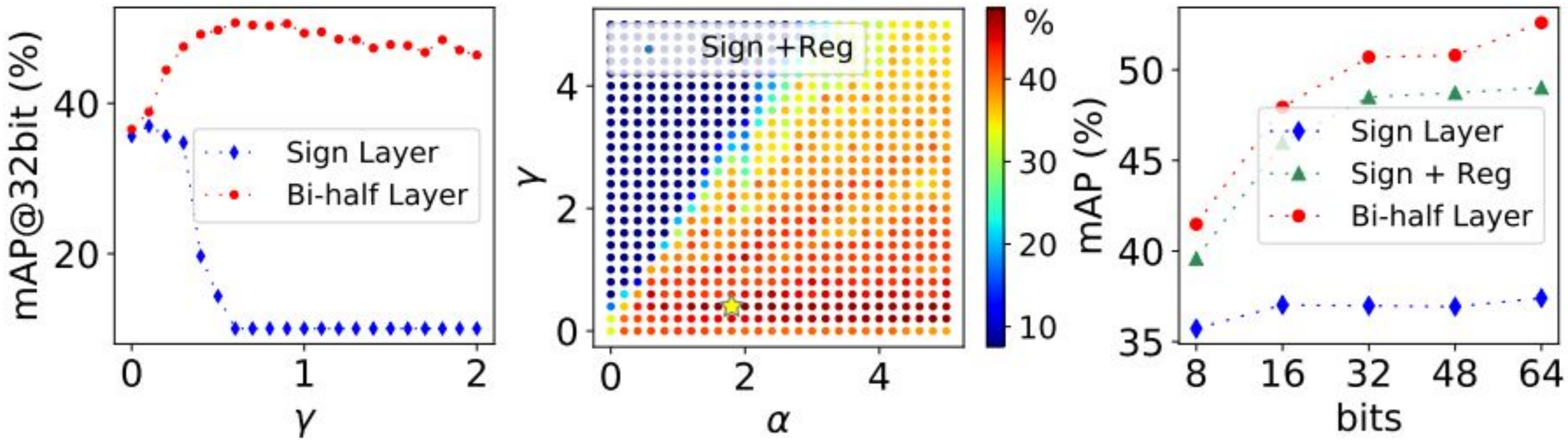
4.3 Experiment : Training an AutoEncoder from scratch



(a) Sign Layer (b) Sign + Reg (c) Bi-half Layer

Train an AutoEncoder from scratch on Mnist dataset. The top row (a, b, c) visualizes the continuous feature distributions before binarization over different methods by training the network with 3 hash bits.

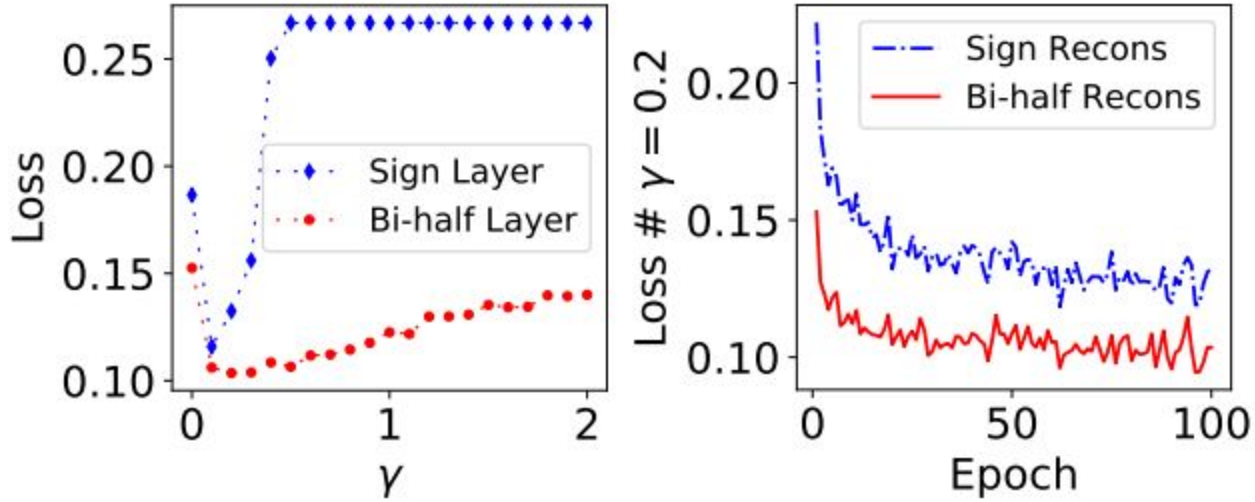
4.3 Experiment : Training an AutoEncoder from scratch



(d) Retrieval Results

shows the corresponding retrieval results. We compare bi-half layer with sign layer and sign+reg. In specific, sign+reg uses an additional entropy regularization term to optimize entropy, while it is hard to balance the added term

4.3 Experiment : Training an AutoEncoder from scratch

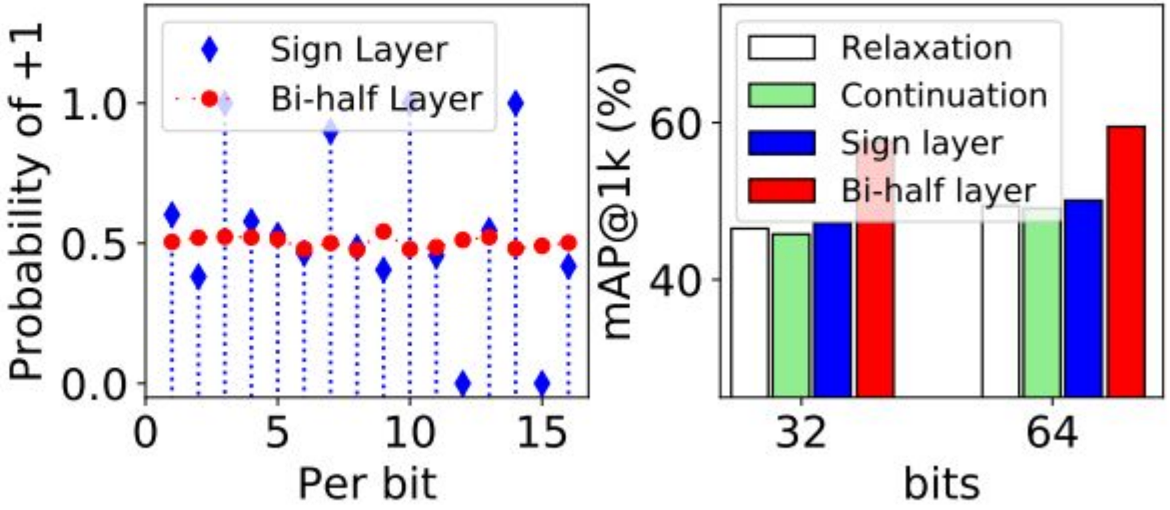


(e) Reconstruction Loss

Shows the reconstruction loss curves for sign layer and bi-half layer. Generating informative binary codes in latent space can help to do reconstruction

4.4 Experiment : Empirical analysis

- How are individual hashing bits distributed?

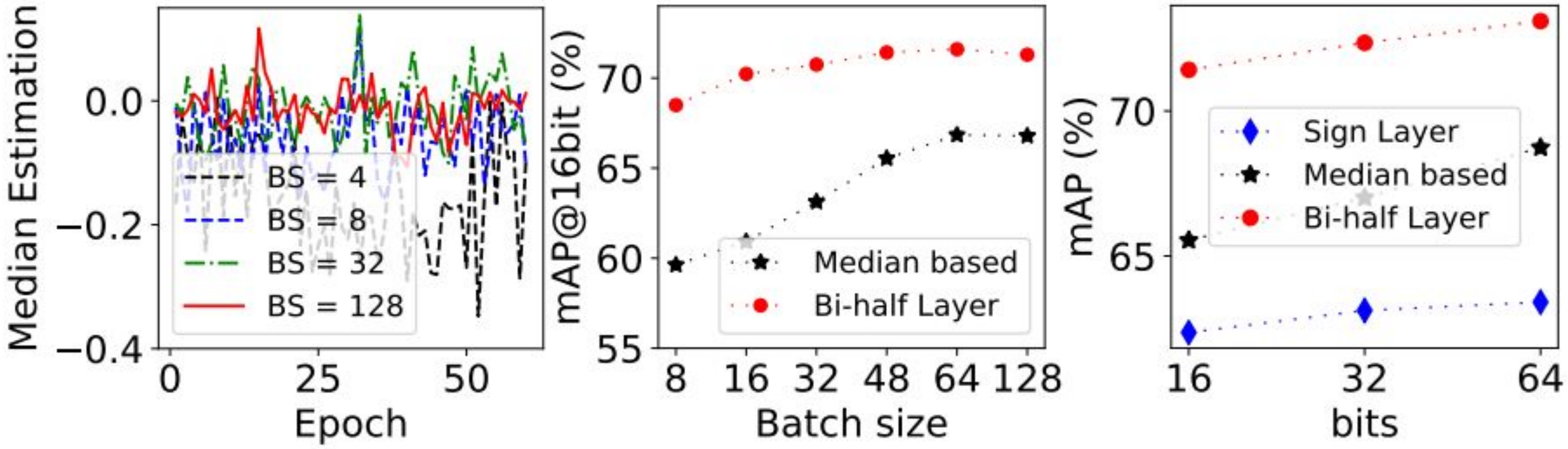


(a) left two subfigures on Cifar-10

Bi-half layer can generate informative hash bits and outperforms other coding methods.

4.4 Experiment : Empirical analysis

- Other hash coding strategies

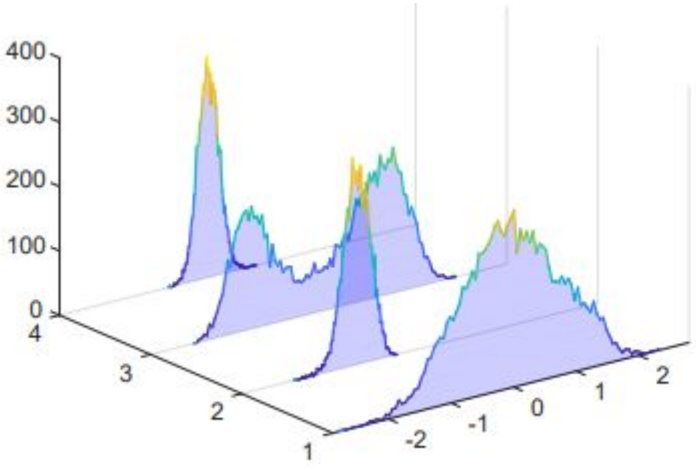


(b) right three subfigures on Flickr25k

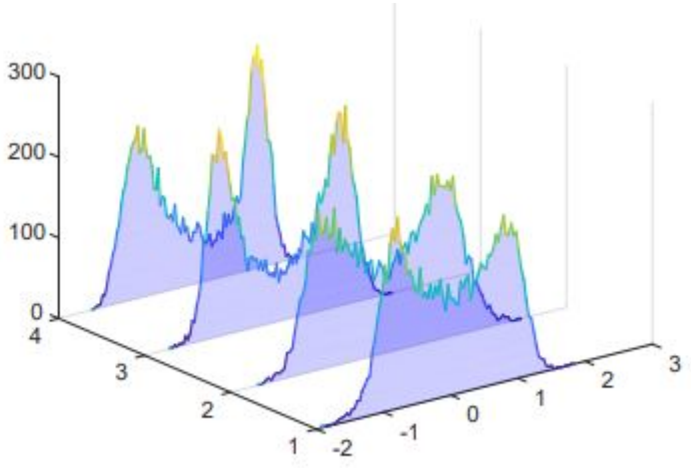
The alternative median based method performs worse than bi-half layer

4.4 Experiment : Empirical analysis

- How are the continuous features distributed?



Sign layer



Bi-half layer

Bi-half method approximates the ideal half-half distribution.

4.5 Experiment : Comparison with state-of-the art

Method	Cifar-10(I) 16 bits	Cifar-10(I) 32 bits	Cifar-10(I) 64 bits	Nus-wide(I) 16 bits	Nus-wide(I) 32 bits	Nus-wide(I) 64 bits
DeepBit	19.40	24.90	27.70	39.22	40.32	42.06
SAH	41.80	45.60	47.40	–	–	–
SADH	–	–	–	60.14	57.99	56.33
HashGAN	44.70	46.30	48.10	–	–	–
GreedyHash*	44.80	47.20	50.10	55.49	57.47	60.93
Ours	**56.10**	**57.60**	**59.50**	**65.12**	**66.31**	**67.26**

Table 1: mAP@1000 results on Cifar-10(I) and mAP@All results on Nus-wide(I). The * denotes that we run the experiments with the released code by the authors.

Greedy hash is effective to solve the vanishing gradient problem and maintain the discrete constraint in hash learning, but it cannot maximize hash bit capacity. In contrast, our method does maximize hash bit capacity and clearly outperforms all other methods on this two datasets.

4.5 Experiment : Comparison with state-of-the art

Method	Flickr25k			Nus-wide(II)			Cifar-10(II)		
	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
LSH + VGG (Datar and Indyk 2004)	58.31	58.85	59.33	43.24	44.11	44.33	13.19	15.80	16.73
SH + VGG (Weiss, Torralba, and Fergus 2008)	59.19	59.23	60.16	44.58	45.37	49.26	16.05	15.83	15.09
ITQ + VGG (Gong and Lazebnik 2011)	61.92	63.18	63.46	52.83	53.23	53.19	19.42	20.86	21.51
DeepBit (Lin et al. 2016)	59.34	59.33	61.99	45.42	46.25	47.62	22.04	24.10	25.21
SGH (Dai et al. 2017)	61.62	62.83	62.53	49.36	48.29	48.65	17.95	18.27	18.89
SSDH (Yang et al. 2018)	66.21	67.33	67.32	62.31	62.94	63.21	25.68	25.60	25.87
DistillHash (Yang et al. 2019)	69.64	70.56	70.75	66.67	67.52	67.69	28.44	28.53	28.67
GreedyHash* (Su et al. 2018)	62.36	63.12	63.41	51.39	55.80	59.27	28.71	31.72	35.47
Ours	**71.42**	**72.35**	**73.10**	**67.12**	**68.05**	**68.21**	**42.87**	**43.29**	**44.13**

Table 2: mAP@All for various methods on three Flickr25k, Nus-wide(II) and Cifar-10(II) datasets. Our method with 16 bits outperforms others that use 64 bits.

4.5 Experiment : Comparison with state-of-the art

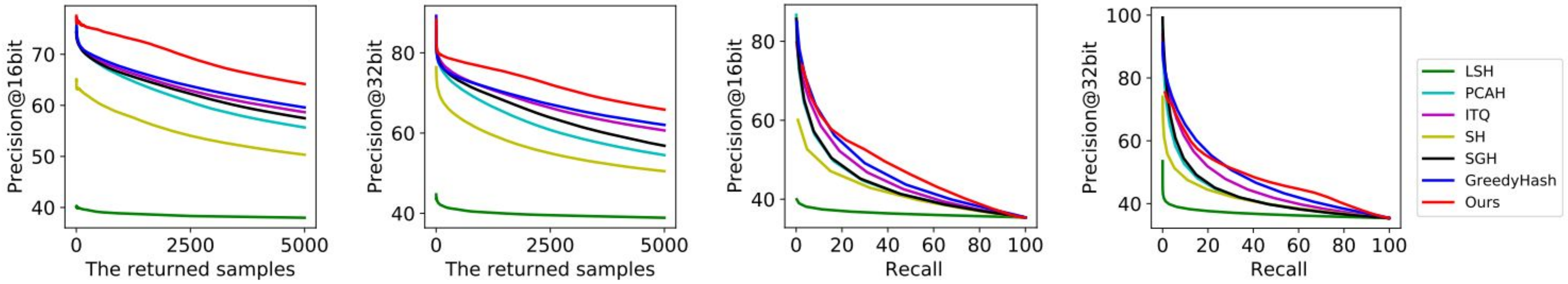


Figure 6: Top N precision and precision-recall curves on Mscoco. The proposed bi-half layer performs best.

Use Dataset of Mscoco

4.5 Experiment : Comparison with state-of-the art

Backbone	Method	Ucf-101			Hmdb-51		
		16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
ResNet-34	GreedyHash*	45.49	57.24	64.77	30.32	37.55	40.53
	Ours	50.83	60.30	65.89	34.21	38.67	41.74
ResNet-101	GreedyHash*	39.29	58.35	67.23	27.60	39.96	42.07
	Ours	**59.30**	**66.13**	**68.47**	**36.68**	**41.48**	**43.03**

Table 3: mAP@100 results on two video datasets using kinetics pre-trained 3D ResNet-34 and 3D ResNet-101. The * denotes we run the experiments with the released code.

For both datasets and both ResNet models our bi-half method consistently outperforms the sign layer method over all hash bit length, especially for short bits. In hashing, fewer bits is essential to save storage and compute.

5.1 Discussion : Conclusion

- 藉由優化 Bit Entropy 來實現Bi-half Net 的想法
- 和藉由更改參數的 Loss function 比起來，Bi-half layer 更勝一籌
- Bi-half layer 可以輕鬆的加進 Deep Structure，像是AutoEncoder，來產生更好的Binary codes，並做出更好的分類。
- 其他的hashing method通常要用到64 bits，而bi-half layer只需用到16 bits

5.2 Discussion : Future Works

- 在Maximizing hash channel capacity時，並沒有考慮到每個bit之間的相依性關聯，因此在之後的研究中會將此因素也考慮進去。

6 References

[Belghazi et al. 2018] Belghazi, M. I.; Baratin, A.; Rajeswar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, R. D. 2018. Mine: mutual information neural estimation. *ICML*.

[Bengio, Léonard, and Courville 2013] Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *Technical Report*.

[Cao et al. 2017] Cao, Z.; Long, M.; Wang, J.; and Yu, P. S. 2017. Hashnet: Deep learning to hash by continuation. In *ICCV*.

[Cao et al. 2018] Cao, Y.; Long, M.; Bin, L.; and Wang, J. 2018. Deep cauchy hashing for hamming space retrieval. In *CVPR*.

[Chang 2012] Chang, S. F. 2012. Supervised hashing with kernels. In *CVPR*.

[Chen et al. 2016] Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2172–2180.

[Chen et al. 2018] Chen, Z.; Yuan, X.; Lu, J.; Tian, Q.; and Zhou, J. 2018. Deep hashing via discrepancy minimization. In *CVPR*, 6838–6847.

[Chen, Cheung, and Wang 2018] Chen, J.; Cheung, W. K.; and Wang, A. 2018. Learning deep unsupervised binary codes for image retrieval. In *IJCAI*.

[Chua et al. 2009] Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. Nus-wide: a real-world web image database from national university of singapore. In *CIVR*.

[Cover and Thomas 2012] Cover, T. M., and Thomas, J. A. 2012. *Elements of information theory*. John Wiley & Sons.

[Dai et al. 2017] Dai, B.; Guo, R.; Kumar, S.; He, N.; and Song, L. 2017. Stochastic generative hashing. In *ICML*, 913–922.

[Datar and Indyk 2004] Datar, M., and Indyk, P. 2004. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the ACM Symposium on Computational Geometry*, 253–262.

[Do et al. 2017] Do, T.-T.; Le Tan, D.-K.; Pham, T. T.; and Cheung, N.-M. 2017. Simultaneous feature aggregating and hashing for large-scale image search. In *CVPR*, 6618–6627.

[Erin Liong et al. 2015] Erin Liong, V.; Lu, J.; Wang, G.; Moulin, P.; and Zhou, J. 2015. Deep hashing for compact binary codes learning. In *CVPR*, 2475–2483.

[Ghasedi Dizaji et al. 2018] Ghasedi Dizaji, K.; Zheng, F.; Sadoughi, N.; Yang, Y.; Deng, C.; and Huang, H. 2018. Unsupervised deep generative adversarial hashing network. In *CVPR*, 3664–3673.

[Gionis, Indyk, and Motwani 2000] Gionis, A.; Indyk, P.; and Motwani, R. 2000. Similarity search in high dimensions via hashing. In *Proceedings of International Conference on Very Large Databases*, 518–529.

[Gong and Lazebnik 2011] Gong, Y., and Lazebnik, S. 2011. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*.

[Goyal et al. 2017] Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; and He, K. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

[Hara, Kataoka, and Satoh 2017] Hara, K.; Kataoka, H.; and Satoh, Y. 2017. Learning spatio-temporal features with 3d residual networks for action recognition. In *ICCV*, 3154–3160.

[Hara, Kataoka, and Satoh 2018] Hara, K.; Kataoka, H.; and Satoh, Y. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 6546–6555.

[He et al. 2016] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

[He, Wen, and Sun 2013] He, K.; Wen, F.; and Sun, J. 2013. K-means hashing: An affinity-preserving quantization method for learning binary compact codes. In *CVPR*.

[Huiskes and Lew 2008] Huiskes, M. J., and Lew, M. S. 2008. The mir flickr retrieval evaluation. In *ACM Multimedia*, 39–43.

[Jegou, Douze, and Schmid 2008] Jegou, H.; Douze, M.; and Schmid, C. 2008. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 304–317. Springer.

[Jiang and Li 2015] Jiang, Q.-Y., and Li, W.-J. 2015. Scalable graph hashing with feature transformation. In *IJCAI*.

[Jiang and Li 2017] Jiang, Q.-Y., and Li, W.-J. 2017. Deep cross-modal hashing. In *CVPR*.

[Jolliffe 2002] Jolliffe, I. T. 2002. Principal component analysis. *Journal of Marketing Research* 87(100):513.

6 References

[Kay et al. 2017] Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *CORR*.

[Kong and Li 2012] Kong, W., and Li, W. J. 2012. Isotropic hashing. In *NIPS*.

[Krizhevsky and Hinton 2009] Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

[Krizhevsky 2014] Krizhevsky, A. 2014. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*.

[Kuehne et al. 2011] Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. Hmdb: a large video database for human motion recognition. In *ICCV*, 2556–2563.

[Kulis and Grauman 2009] Kulis, B., and Grauman, K. 2009. Kernelized locality-sensitive hashing for scalable image search. In *ICCV*.

[Kulis, Jain, and Grauman 2009] Kulis, B.; Jain, P.; and Grauman, K. 2009. Fast similarity search for learned metrics. *IEEE TPAMI* 31(12):2143.

[Lai et al. 2015a] Lai, H.; Pan, Y.; Liu, Y.; and Yan, S. 2015a. Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*.

[Lai et al. 2015b] Lai, H.; Pan, Y.; Liu, Y.; and Yan, S. 2015b. Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*.

[LeCun et al. 1998] LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.; et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

[Li et al. 2017] Li, Q.; Sun, Z.; He, R.; and Tan, T. 2017. Deep supervised discrete hashing. In *NIPS*.

[Li et al. 2019] Li, Y.; Pei, W.; van Gemert, J.; et al. 2019. Push for quantization: Deep fisher hashing. In *BMVC*.

[Lin et al. 2014a] Lin, G.; Shen, C.; Shi, Q.; Hengel, A. V. D.; and Suter, D. 2014a. Fast supervised hashing with decision trees for high-dimensional data. In *CVPR*.

[Lin et al. 2014b] Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014b. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

[Lin et al. 2016] Lin, K.; Lu, J.; Chen, C.-S.; and Zhou, J. 2016. Learning compact binary descriptors with unsupervised deep neural networks. In *CVPR*, 1183–1192.

[Liu et al. 2011] Liu, W.; Wang, J.; Kumar, S.; and Chang, S.-F. 2011. Hashing with graphs. In *ICML*.

[Liu et al. 2014] Liu, W.; Kumar, S.; Kumar, S.; and Chang, S. F. 2014. Discrete graph hashing. In *NIPS*.

[Liu et al. 2016] Liu, H.; Wang, R.; Shan, S.; and Chen, X. 2016. Deep supervised hashing for fast image retrieval. *CVPR*.

[Liu et al. 2017] Liu, H.; Wang, R.; Shan, S.; and Chen, X. 2017. Learning multifunctional binary codes for both category and attribute oriented retrieval tasks. In *CVPR*.

[Lu, Liong, and Zhou 2017] Lu, J.; Liong, V. E.; and Zhou, J. 2017. Deep hashing for scalable image search. *IEEE TIP* 26(5):2352–2367.

[Mu and Yan 2010] Mu, Y., and Yan, S. 2010. Non-metric locality-sensitive hashing. In *AAAI*.

[Raginsky 2009] Raginsky, M. 2009. Locality-sensitive binary codes from shift-invariant kernels. In *NIPS*.

[Raziperchikolaei and Carreira-Perpiñán 2016] Raziperchikolaei, R., and Carreira-Perpiñán, M. A. 2016. Optimizing affinity-based binary hashing using auxiliary coordinates. In *NIPS*.

[Shannon 1948] Shannon, C. E. 1948. A mathematical theory of communication. *Bell system technical journal* 27(3):379–423.

[Shen et al. 2015] Shen, F.; Shen, C.; Liu, W.; and Shen, H. T. 2015. Supervised discrete hashing. In *CVPR*.

[Shen et al. 2018] Shen, F.; Xu, Y.; Liu, L.; Yang, Y.; Huang, Z.; and Shen, H. T. 2018. Unsupervised deep hashing with similarity-adaptive and discrete optimization. *IEEE TPAMI* 40(12):3034–3044.

[Simonyan and Zisserman 2015] Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. *ICLR*.

6 References

[Soomro, Zamir, and Shah 2012] Soomro, K.; Zamir, A. R.; and Shah, M. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CORR*.

[Su et al. 2018] Su, S.; Zhang, C.; Han, K.; and Tian, Y. 2018. Greedy hash: Towards fast optimization for accurate hash coding in cnn. In *NIPS*. 798–807.

[Villani 2003] Villani, C. 2003. *Topics in optimal transportation*. Number 58. American Mathematical Soc.

[Weiss, Torralba, and Fergus 2008] Weiss, Y.; Torralba, A.; and Fergus, R. 2008. Spectral hashing. In *NIPS*.

[Xu et al. 2013] Xu, B.; Bu, J.; Lin, Y.; Chen, C.; He, X.; and Cai, D. 2013. Harmonious hashing. In *IJCAI*.

[Yang et al. 2018] Yang, E.; Deng, C.; Liu, T.; Liu, W.; and Tao, D. 2018. Semantic structure-based unsupervised deep hashing. In *IJCAI*, 1064–1070.

[Yang et al. 2019] Yang, E.; Liu, T.; Deng, C.; Liu, W.; and Tao, D. 2019. Distillhash: Unsupervised deep hashing by distilling data pairs. In *CVPR*, 2946–2955.

[Yuan et al. 2018] Yuan, X.; Ren, L.; Lu, J.; and Zhou, J. 2018. Relaxation-free deep hashing via policy gradient. In *ECCV*, 134–150.

[Zhang et al. 2010] Zhang, D.; Wang, J.; Cai, D.; and Lu, J. 2010. Self-taught hashing for fast similarity search. In *ACM SIGIR*, 18–25.

[Zhang et al. 2014] Zhang, P.; Zhang, W.; Li, W. J.; and Guo, M. 2014. Supervised hashing with latent factor models. In *SIGIR*.

[Zhao et al. 2015] Zhao, F.; Huang, Y.; Wang, L.; and Tan, T. 2015. Deep semantic ranking based hashing for multi-label image retrieval. In *CVPR*.