Pseudo-supervised (Deep) Learning for Image Search

Wengang Zhou (周文罡)

EEIS Department, University of Science & Technology of China

zhwg@ustc.edu.cn

Outline

- Background
- Motivation
- Our Work
- Conclusion

Outline

- **Background**
- Motivation
- Our Work
- Conclusion

Background

- Deep learning has been widely and successfully applied in many vision tasks
  - Classification, detection, segmentation, etc.
  - Popular models: AlexNet, VGGNet, ResNet, DenseNets
- What is learnt with deep learning?
  - **Feature representation** to characterize and discriminate visual content
- What make the success of deep learning?
  - Novel techniques in model design
    - ✓ Dropout, batch normalization, ReLU, etc.
  - Powerful computing capability
  - **Big training data**
- Pre-request of deep learning
  - Sufficient training data with **label** as supervision
  - Such as image class, object bounding box, pixel category, etc.

Background

- Content-based Image search
  - Problem definition
    - Given a query image, identify those similar ones from a large corpus
  - Key issues
    - Image representation
      - How to represent the visual content to **measure image relevance**?
      - **Invariant to various transformations**, including rotation, scaling, illumination change, background clutter, etc.
    - Image database index
      - How to enable the fast query response with a large image dataset?
  - Characteristic
    - Large database, real-time query response
    - Unknown number of image category
    - Infeasible to numerate the potential categories
    - Data without label: difficult to train a deep learning model

Outline

- Background
- **Motivation**
- Our Work
- Conclusion

Motivation

- How to leverage deep learning to image search?
  - Apply the pre-trained CNN model from image classification task
    - ✓ Fail to directly optimize towards the goal of image search
    - ✓ Achieve sub-optimal performance in search problem
- Key problem
  - How to make up the **virtual** label to supervise the learning with deep CNN model?
- Our solutions
  - Generate supervision with retrieval-oriented context
    - ✓ Refine the deep learning feature of a pre-trained CNN model
    - ✓ Fine-tune a pre-trained CNN model
  - Leverage the outputs of existing methods as supervision
    - ✓ Binary hashing for ANN search

Outline

- Background
- Motivation
- **Our Work**
- Conclusion

Our Work

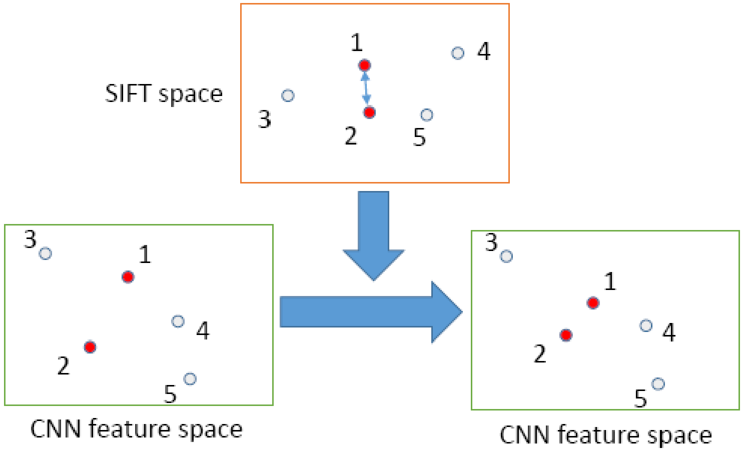
- Generate supervision with retrieval-oriented context
  - Refine the deep learning feature of a pre-trained CNN model
    - ✓ **Collaborative index embedding**
  - Fine-tune a pre-trained CNN model
    - ✓ **Deep Feature Learning with Complementary Supervision**
- Leverage the outputs of existing methods as supervision
  - Learn better binary hash functions for ANN search
    - ✓ **Pseudo-supervised Binary Hashing with linear distance preserving constraints**

Our Work

- Generate supervision with retrieval-oriented context
  - **Refine the deep learning feature of a pre-trained CNN model**
    - ✓ **Collaborative index embedding**
  - Fine-tune a pre-trained CNN model
    - ✓ Deep Feature Learning with Complementary
- Leverage the outputs of existing methods for refinement
  - Learn better binary hash functions for ANN search
    - ✓ **Pseudo-supervised Binary Hashing with linear distance preserving constraints**

Collaborative Index Embedding

- Motivation
  - Images are represented with different features, such as SIFT and CNN
  - How to explore the complementary clue among different features
- Basic idea: neighborhood embedding
  - Ultimate goal: make the nearest neighborhood structure consistent across different feature space
  - If image 1 and 2 are nearest neighbors of each other in SIF space, pull them to be closer in CNN feature space
  - Do similar operation in SIFT feature

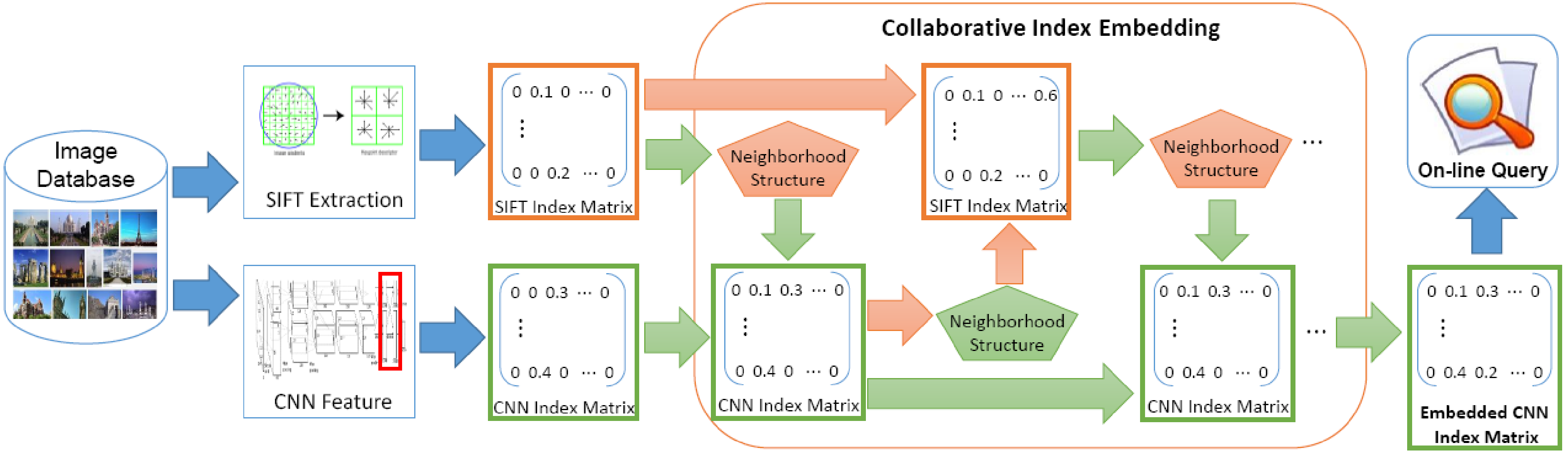


Collaborative Index Embedding

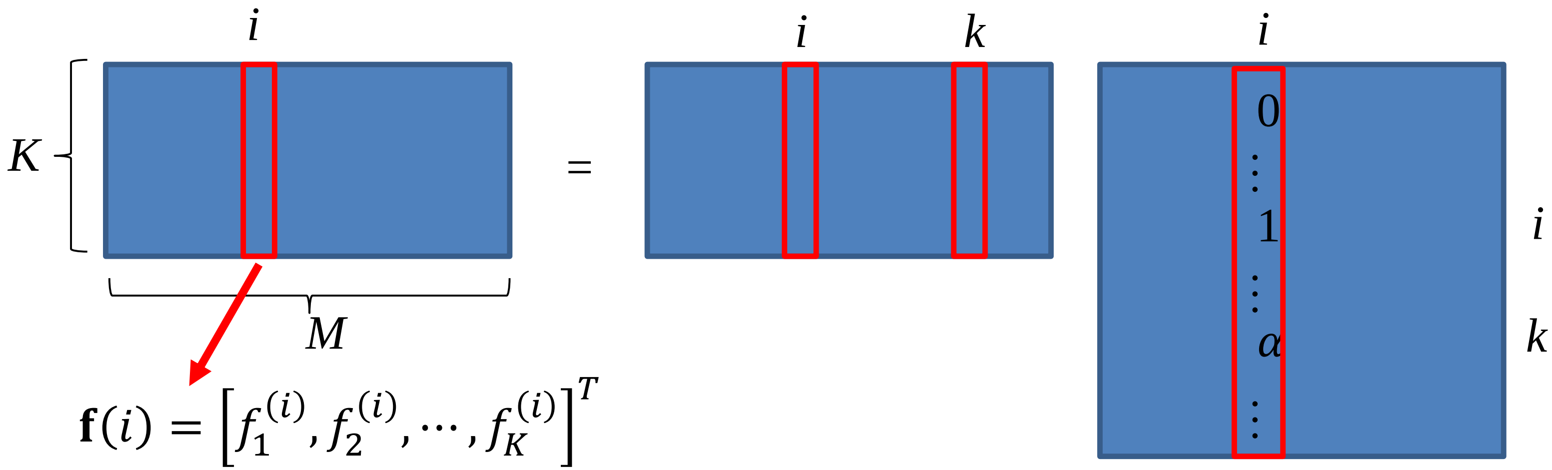
- Optimization formulation

$$\mathbf{C}(\tilde{\mathbf{M}}_C, \tilde{\mathbf{M}}_S) = -\sum_{u \in \mathbf{P}} \frac{\#(\mathcal{R}_C(u) \cap \mathcal{R}_S(u))}{\#(\mathcal{R}_C(u) \cup \mathcal{R}_S(u))} + \mu * ||\mathbf{\Phi}_C||_F + \lambda * ||\mathbf{\Phi}_S||_F,$$

- Implementation framework

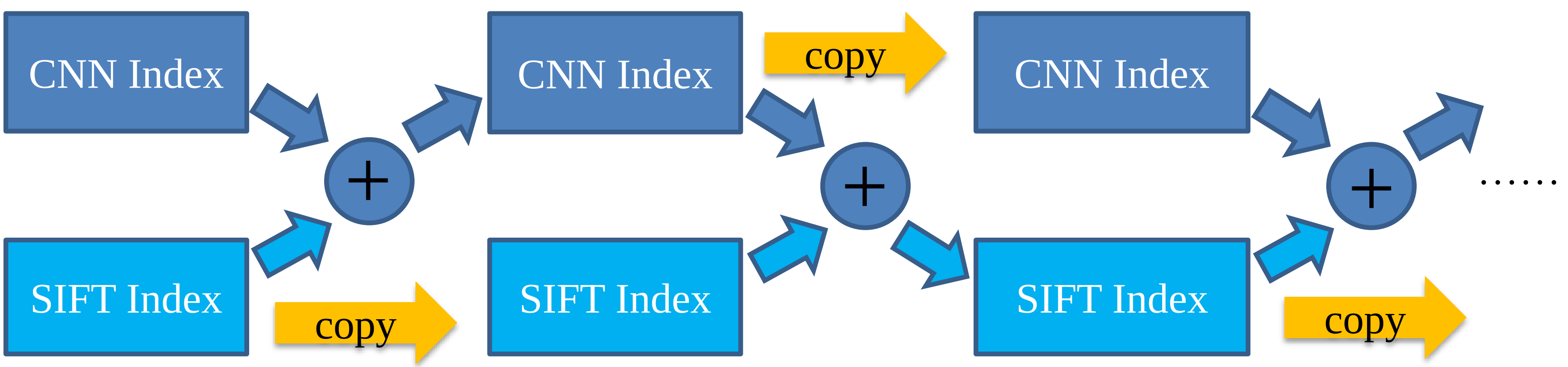


Interpretation of Index Embedding



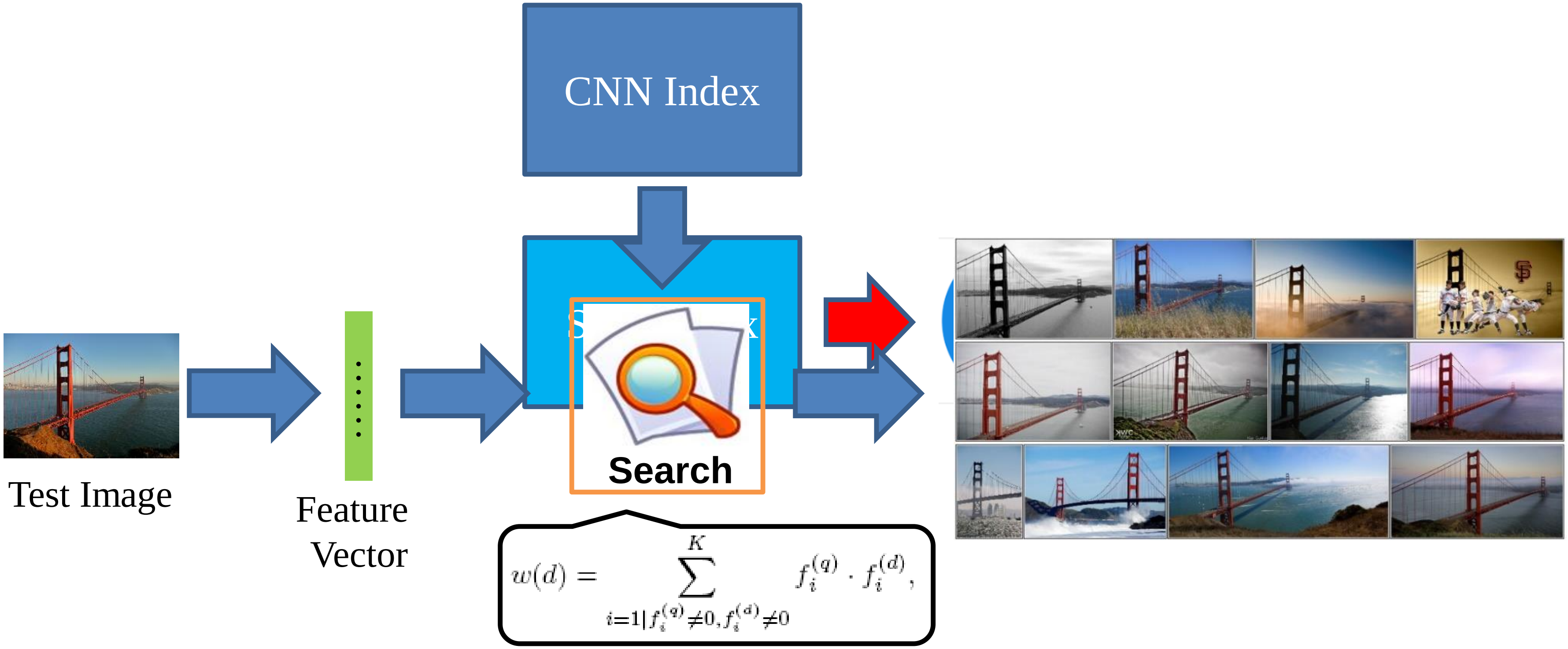
$$\mathbf{f}(i) = \left[f_1^{(i)}, f_2^{(i)}, \cdots, f_K^{(i)}\right]^T$$

$$f_j^{(i)} := \begin{cases} f_j^{(i)} + \alpha \cdot f_j^{(k)}, & \text{if } f_j^{(i)} = 0 \\ f_j^{(i)}, & \text{otherwise} \end{cases}$$



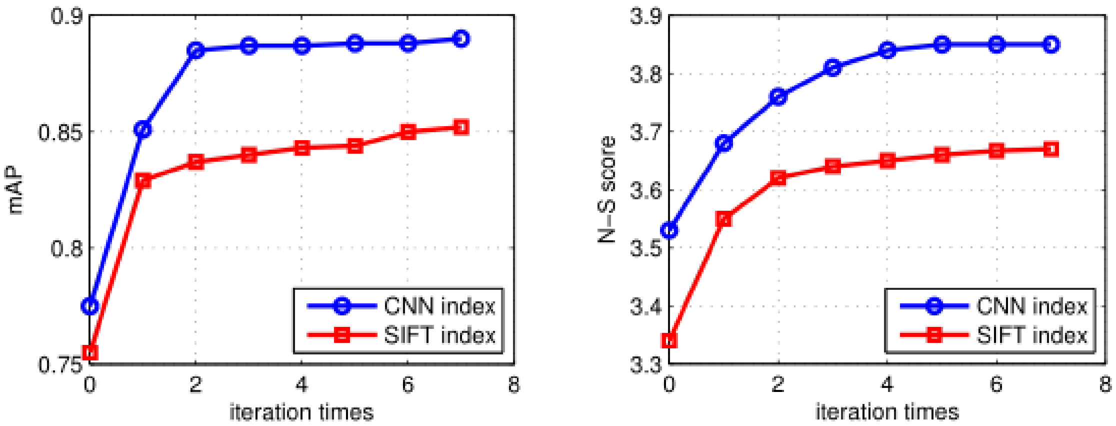
Online Query

- Key only the index of CNN feature
  - Smaller storage, better retrieval accuracy

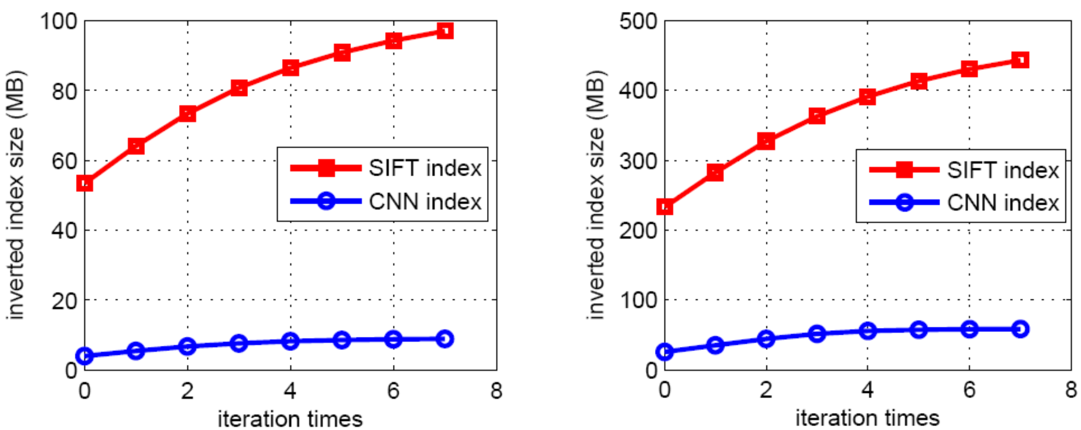


Experiments

- Retrieval accuracy in each iteration



- Index size in each iteration



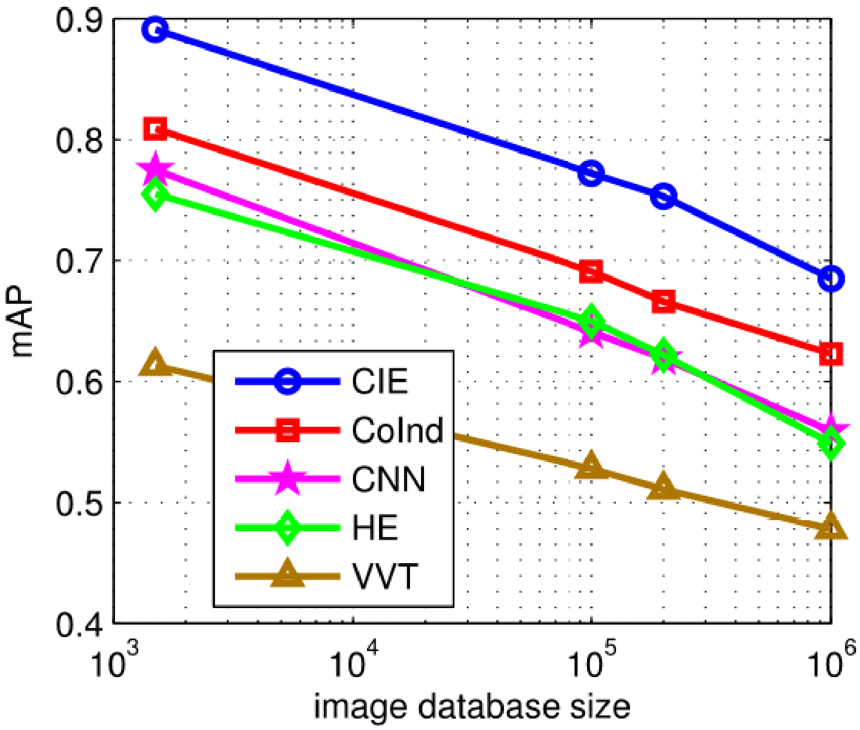
Experiments

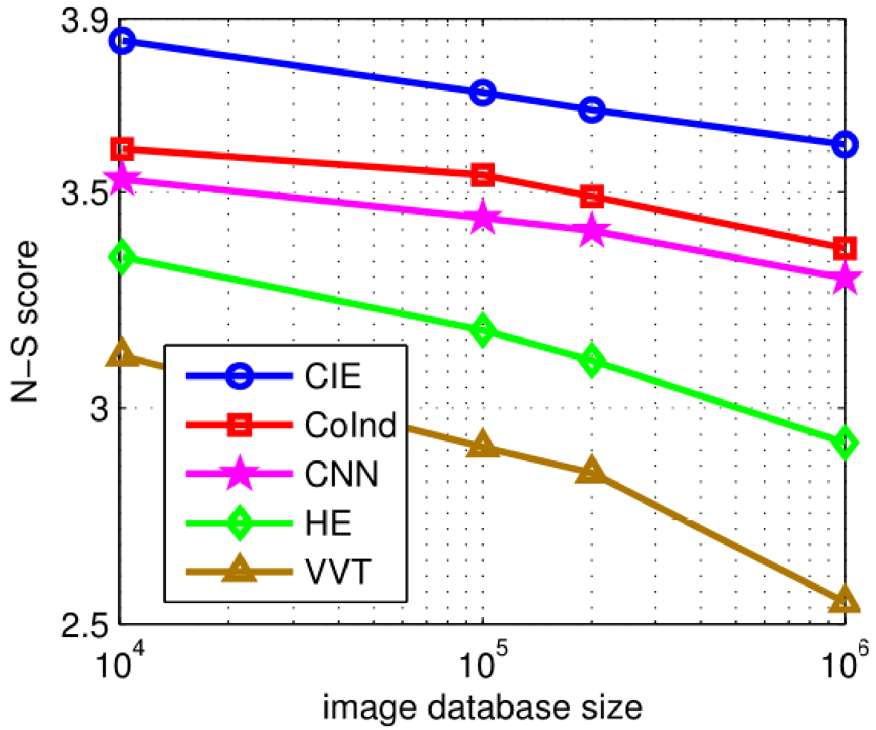
- Comparison with existing retrieval algorithms

Methods	UKbench (N-S score)	Holidays (mAP)	Involved visual features	On-line memory cost per indexed image (Bytes)
CWVT[18]	3.56	0.781	SIFT	16K
SCSM[40]	3.52	0.762	SIFT	18K
HE+WGC[13]	3.42	0.813	SIFT	24K
CDM[10]	3.68	NA	SIFT	16K
KrNN[11]	3.67	NA	SIFT	22K
QSF[44]	3.77	0.846	SIFT, HSV	20K
CoInd[21]	3.60	0.809	SIFT, attributes	24K
c-MI[22]	3.85	0.858	SIFT, color names	13.5K
MsOP[34]	NA	0.802	dense CNN	48K
QaLF[46]	3.84	0.880	SIFT, holistic CNN, HSV, GIST	62K
CIE	**3.86**	**0.892**	SIFT, holistic CNN	4K
CIE+	**3.91**	**0.903**	SIFT, holistic CNN	52K

Experiments

- Evaluation on different database scales





Our Work

- Generate supervision with retrieval-oriented context
  - Refine the deep learning feature of a pre-trained CNN model
    - ✓ Collaborative index embedding (TPAMI 2017)
  - Fine-tune a pre-trained CNN model
    - ✓ Deep Feature Learning with Complementary Supervision (TIP, under review)
- Leverage the outputs of existing methods for refinement
  - Learn better binary hash functions for ANN search
    - ✓ **Pseudo-supervised Binary Hashing with linear distance preserving constraints** （TIP-2017, MM-2016）

Deep Feature Learning with Complementary Supervision Mining

- Motivation
  - Database images are not independent of each other
  - Makes use of the complementary clues from different visual features as supervision to guide the learning with deep CNN
- Complementary Supervision Mining
  - Makes use of the relevance dependence among database images
  - Reversible nearest neighbourhood

$$R_C(I_i) = \{I_j | I_j \in \mathcal{N}_C(I_i, p), I_i \in \mathcal{N}_C(I_j, m)\}$$

$$R_S(I_i) = \{I_j | I_j \in \mathcal{N}_S(I_i, q), I_i \in \mathcal{N}_S(I_j, m)\}.$$

  - How to use it?
    - ✓ Select similar image pairs by SIFT matching to compose a training set

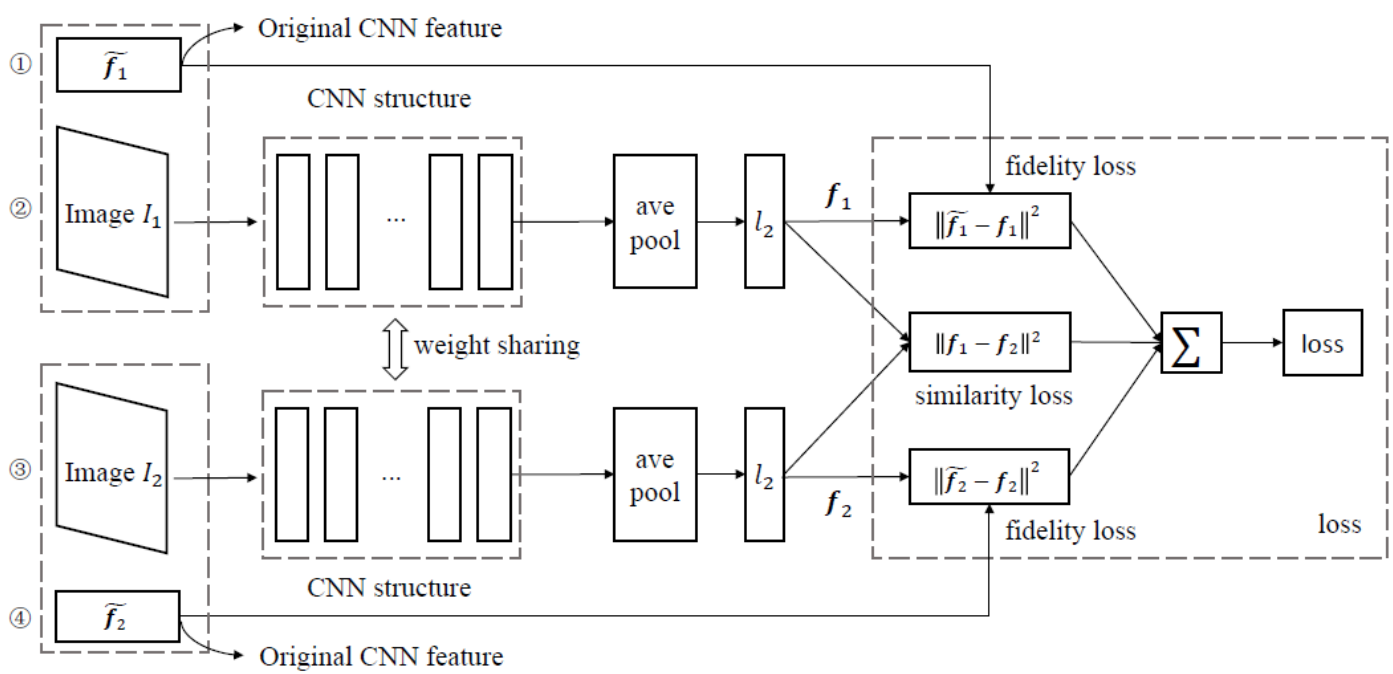
Deep Feature Learning with Complementary Supervision Mining

- Optimization formulation
  - Loss definition

$$\mathcal{L}(I_1, I_2) = \alpha \|\boldsymbol{f}_1 - \boldsymbol{f}_2\|^2 + \|\boldsymbol{f}_1 - \tilde{\boldsymbol{f}}_1\|^2 + \|\boldsymbol{f}_2 - \tilde{\boldsymbol{f}}_2\|^2,$$

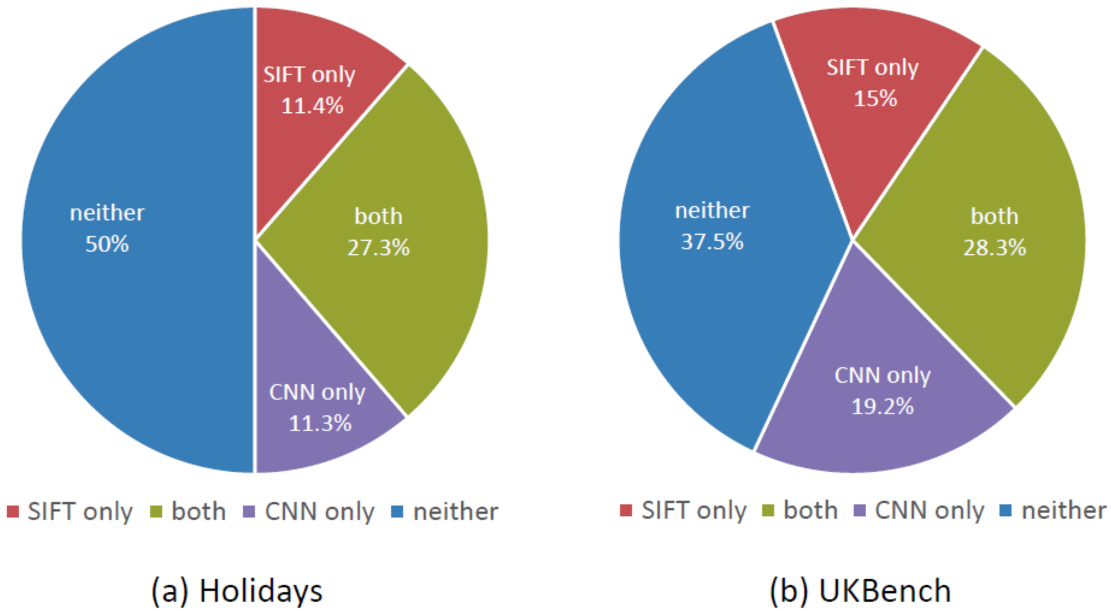
$\boldsymbol{f}_1$: CNN feature of I_1 after fine-tuning

$\tilde{\boldsymbol{f}}_1$: CNN feature of I_1 before fine-tuning



Experiments

- Study of complement on image nearest neighbors with SIFT or CNN



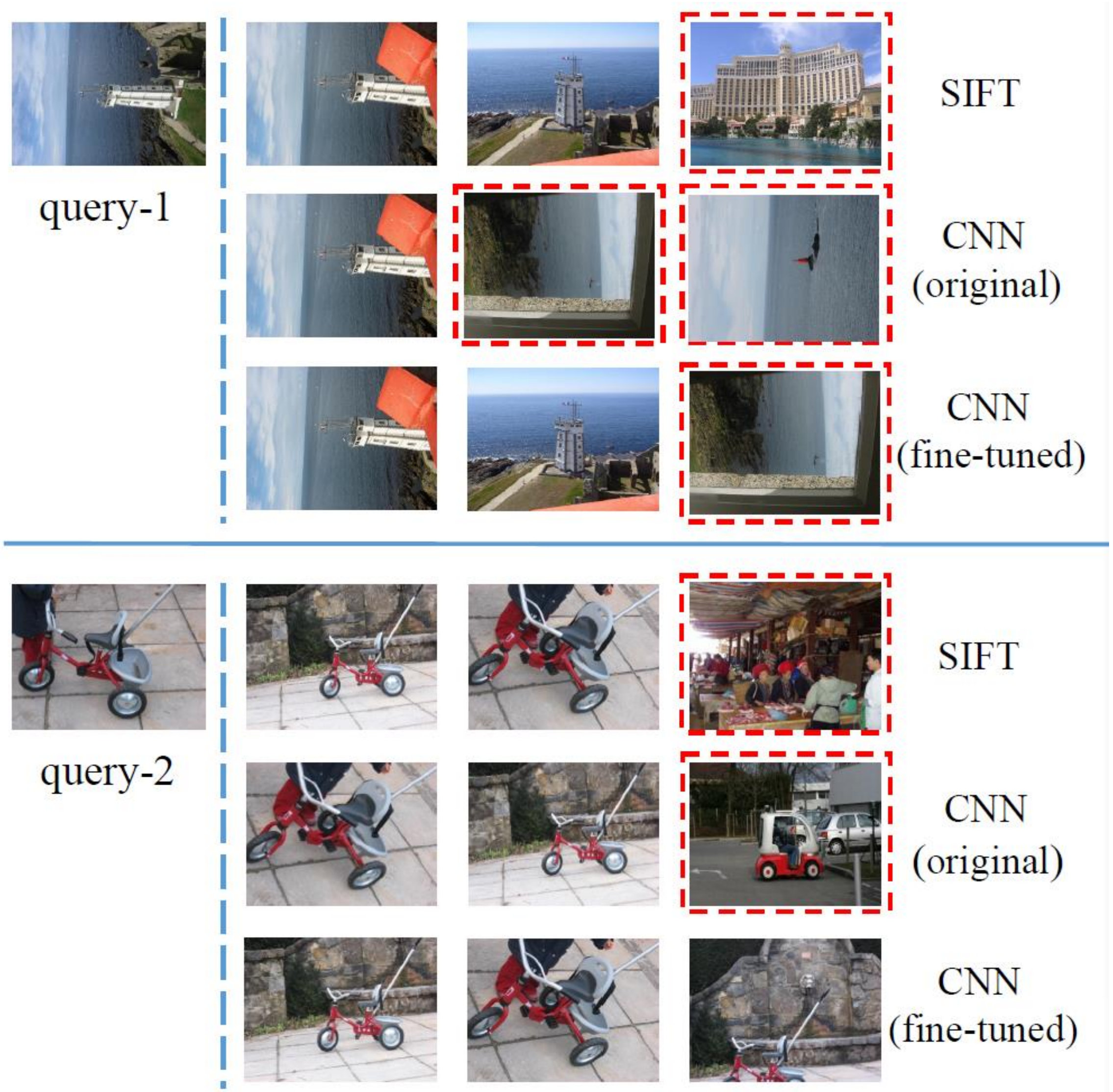
- Comparison of different features

Method	Holidays	UKBench
SIFT	0.735	3.33
CNN (AlexNet)	0.801	3.63
Ours (AlexNet)	0.878	3.88
CNN (VGG-Net16)	0.793	3.67
Ours (VGG-Net16)	0.880	3.90

- Comparison of different query settings

Method	Holidays	UKBench
CNN (pre-trained)	0.801	3.62
CNN (without query)	0.821	3.86
CNN (with query)	0.878	3.88

Qualitative Results



Experiments

- Comparison with multi-feature fusion retrieval methods

Method	Holidays	UKBench	MEM (Bytes)
QaLF [33]	0.880	3.84	16K
OR [32]	0.837	3.81	16K
Zheng *et al.* [28]	0.862	3.78	62K
CIE [31]	**0.892**	3.86	4K
Ours (VGG-Net16)	0.880	**3.90**	2K

- Comparison with deep feature based retrieval methods

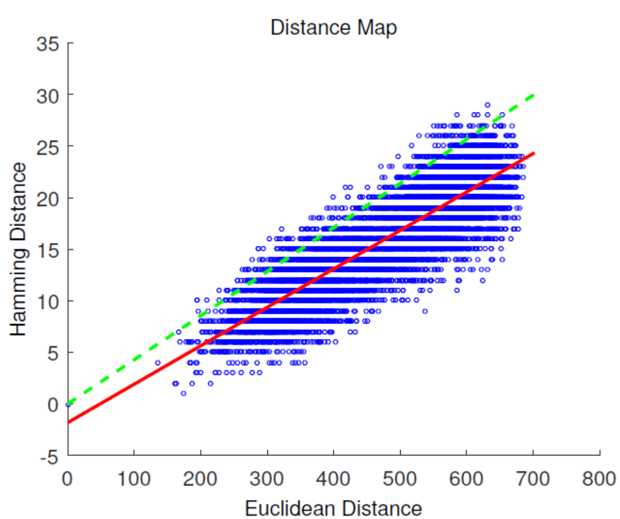
Method	Network	Dim	Holidays	UKBench
SPoC [5]	V	256	0.802	3.65
NetVlad [27]	V	256	0.86	-
CroW [10]	V	512	0.849	-
Neural codes [18]	FA	128	0.789	3.55
R-MAC [19]	FA	256	0.815	-
Ours	FA	256	0.878	3.88
NetVlad [27]	FV	256	0.843	-
R-MAC [19]	FV	512	0.825	-
Gordo et al. [20]	FV	512	0.864	3.55
Ours	FV	512	**0.880**	**3.90**

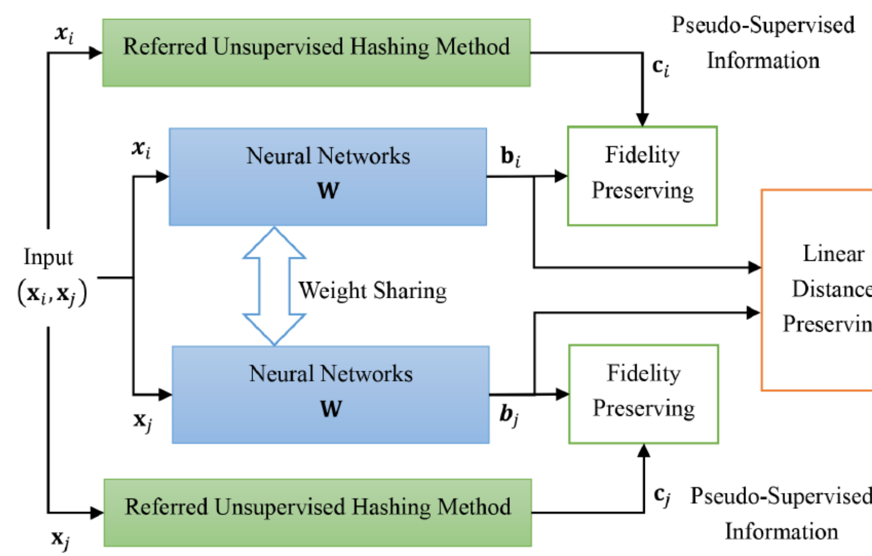
Our Work

- Generate supervision with retrieval-oriented context
  - Refine the deep learning feature of a pre-trained CNN model
    - ✓ Collaborative index embedding
  - Fine-tune a pre-trained CNN model
    - ✓ Deep Feature Learning with Complementary Supervision
- Leverage the outputs of existing methods for refinement
  - Learn better binary hash functions for ANN search
    - ✓ **Pseudo-supervised Binary Hashing with linear distance preserving constraints**

Pseudo-supervised Binary Hashing

- Binary hashing
  - Transform data from Euclidean space to Hamming space
  - Speedup the approximate nearest neighbor search
  - Problem: the optimal output of binary hashing is unknown
- Our solution
  - Take an existing method as Reference and take its output as supervision
  - Impose novel transformation constraints: linear distance preserving
  - Learn a better hashing transformation with neural network





Alternative scheme

- Optimization objective:

$$\min_{\mathbf{W},a,b} \ \frac{\lambda}{N_p}\|\mathbf{h}-a\mathbf{d}-b\|_2^2+\frac{\alpha}{N_p}\left\|\widetilde{\mathbf{U}}-\widetilde{\mathbf{C}}\right\|_F^2+\beta\|\mathbf{W}^T\mathbf{W}-\mathbf{I}\|_F^2$$

- An alternative solution:
  - a, b -step: $\min_{a,b} \|\mathbf{h}-a\mathbf{d}-b\|_2^2$
    - ✓ Linear Regression Problem: Least Square Method
  - $\mathbf{W}$ -step: $\min_{\mathbf{W}} \ \frac{\lambda}{N_p}\|\mathbf{h}-a\mathbf{d}-b\|_2^2+\frac{\alpha}{N_p}\left\|\widetilde{\mathbf{U}}-\widetilde{\mathbf{C}}\right\|_F^2+\beta\|\mathbf{W}^T\mathbf{W}-\mathbf{I}\|_F^2$
    - ✓ Dual Neural Networks: Stochastic Gradient Descent

a, b -step

Repeat until convergence

$\mathbf{W}$ -step

Experimental Results

Precision(%)@500 Comparison

Dataset	Code Length	LSH[15]/LSH-Ours	BRE[28]/BRE-Ours	MLH[21]/MLH-Ours	SpH[20]/SpH-Ours	ITQ[18]/ITQ-Ours	**LDTH**
ANN_SIFT1M	16	0.94 / 1.12	0.83 / 1.69	0.71 / 0.84	1.38 / 1.51	1.32 / **2.30**	1.66
	32	2.52 / 2.95	2.20 / 4.27	2.63 / 2.73	3.65 / 3.93	3.54 / **5.09**	4.12
	64	5.23 / 6.20	4.45 / 7.15	5.84 / 6.39	7.06 / 7.51	7.03 / **7.71**	7.46
	128	9.30 / 10.21	7.70 / 8.62	8.64 / 9.12	10.63 / **11.42**	10.82 / 10.78	11.22
ANN_GIST1M	16	0.32 / 0.69	0.86 / 0.99	0.65 / 0.92	0.76 / 0.91	1.09 / **1.24**	1.23
	32	0.76 / 1.35	1.85 / 1.95	1.38 / 2.00	1.87 / 1.91	2.22 / 2.37	**2.45**
	64	1.61 / 2.79	3.06 / 3.05	2.74 / **3.75**	3.51 / 3.55	3.37 / 3.55	3.74
	128	3.25 / 4.63	4.62 / 4.76	4.07 / 5.36	5.39 / **5.49**	4.40 / 5.46	5.18
CIFAR-10 (1000d fc8)	16	51.33 / 52.18	28.96 / 26.53	48.36 / 53.01	**54.76** / 52.85	49.68 / 53.37	51.86
	32	57.38 / **62.16**	35.30 / 36.05	56.73 / 61.38	59.99 / 60.59	56.35 / 61.38	59.05
	64	64.05 / **68.70**	45.11 / 48.94	62.11 / 67.26	66.14 / 64.14	60.77 / 64.58	65.57
	128	69.04 / **71.88**	49.47 / 57.85	65.35 / 70.91	69.20 / 66.93	64.52 / 68.86	69.98

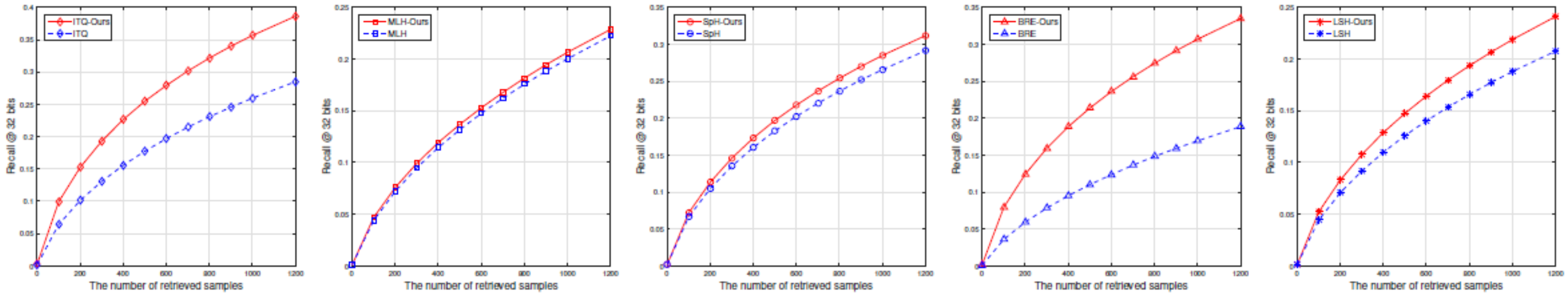
mAP Comparison

Dataset	Code Length	LSH[15]/LSH-Ours	BRE[28]/BRE-Ours	MLH[21]/MLH-Ours	SpH[20]/SpH-Ours	ITQ[18]/ITQ-Ours	**LDTH**
ANN_SIFT1M	16	0.56 / 0.70	0.57 / 1.09	0.48 / 0.56	0.84 / 0.98	0.93 / **1.62**	1.13
	32	2.12 / 2.55	1.75 / 4.18	2.17 / 2.33	3.37 / 3.75	3.31 / **5.52**	4.07
	64	6.29 / 7.92	4.71 / 9.73	6.88 / 7.92	9.47 / 10.38	9.34 / **11.29**	10.53
	128	15.71 / 18.26	11.11 / 13.11	12.94 / 14.19	19.42 / **21.74**	19.91 / 19.68	21.40
ANN_GIST1M	16	0.18 / 0.38	0.43 / 0.52	0.38 / 0.53	0.41 / 0.49	0.64 / **0.75**	0.74
	32	0.56 / 1.11	1.26 / 1.40	0.92 / 1.46	1.34 / 1.44	1.77 / 1.96	**2.03**
	64	1.38 / 2.54	2.75 / 2.77	2.45 / 3.68	3.43 / 3.51	3.39 / 3.57	**3.86**
	128	3.51 / 5.29	5.18 / 5.28	4.35 / 6.23	6.45 / 6.49	5.53 / **6.56**	6.30
CIFAR-10 (1000d fc8)	16	30.65 / 33.80	20.36 / 20.40	33.06 / 34.59	30.00 / 33.85	34.64 / **35.16**	34.60
	32	34.79 / 38.62	22.20 / 25.68	38.04 / 40.03	31.69 / 37.07	39.40 / **41.08**	39.42
	64	38.84 / 43.93	26.52 / 32.35	41.94 / **44.88**	36.82 / 37.59	43.21 / 43.85	43.64
	128	43.56 / 46.74	28.03 / 36.53	44.80 / **47.86**	39.80 / 38.09	46.28 / 47.42	47.23

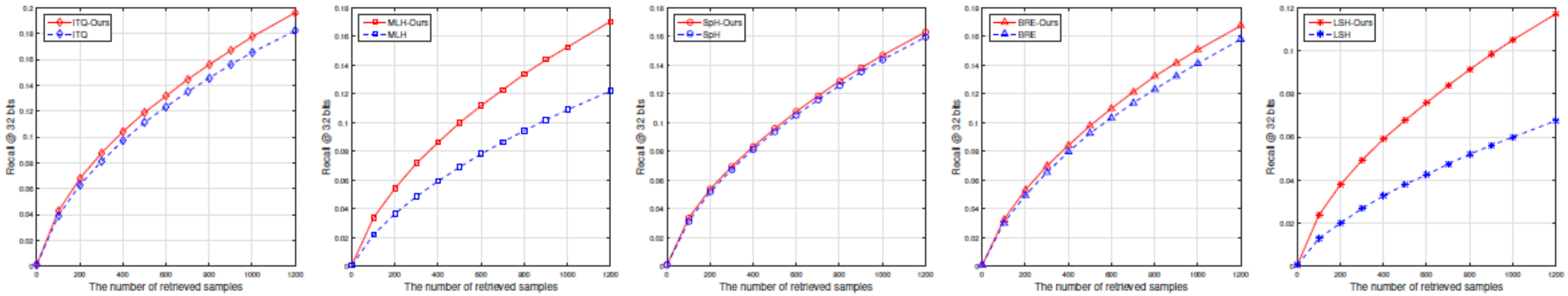
Experimental Results

- Recall@K Comparison on different feature datasets

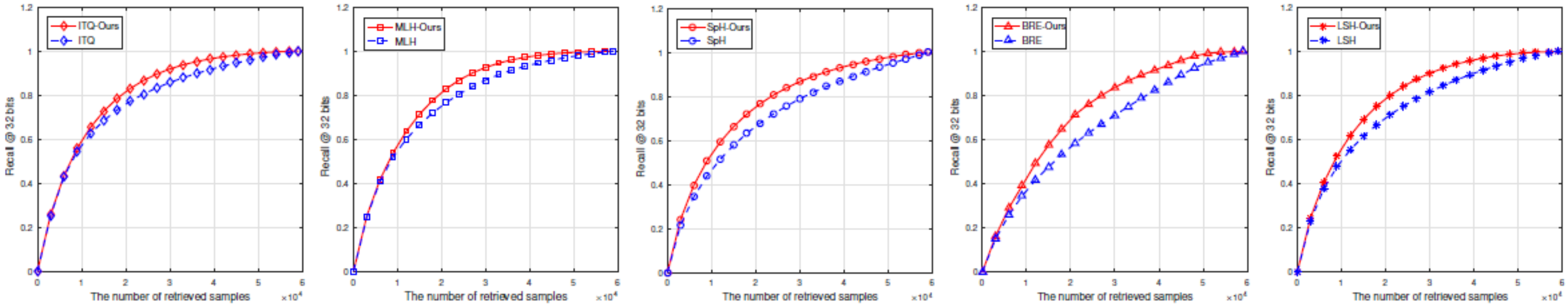
SIFT-1M



GIST-1M



CIFAR-10



Experimental Results

- mAP Comparison for the supervised binary hashing methods

CIFAR-10 IMAGE DATASET

Methods	Architectures	12-bits	24-bits	32-bits	48-bits
CNNH [33]	3 convs, 2 fcs	0.429	0.511	0.509	0.522
CNNH* [32]	Net. in Net.	0.484	0.476	0.472	0.489
NINH [32]	Net. in Net.	0.552	0.566	0.558	0.581
DHN [53]	AlexNet	0.555	0.594	0.603	0.621
DPSH [52]	CNN-F	**0.713**	0.727	0.744	**0.757**
LDSH	CNN-F	0.704	**0.733**	**0.758**	**0.757**

NUS-WIDE DATASET

Methods	Architectures	12-bits	24-bits	32-bits	48-bits
CNNH [33]	3 convs, 2 fcs	0.611	0.618	0.625	0.608
CNNH* [32]	Net. in Net.	0.617	0.663	0.657	0.688
NINH [32]	Net. in Net.	0.674	0.697	0.713	0.715
DHN [53]	AlexNet	0.708	0.735	0.748	0.758
LDSH	CNN-F	0.674	0.719	0.728	0.738

Reference

- **Wengang Zhou**, Houqiang Li, Jian Sun, and Qi Tian, “Collaborative Index Embedding for Image Retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*TPAMI*), Feb. 2017.
- Min Wang, **Wengang Zhou**, Qi Tian, and Houqiang Li, “A General Framework for Linear Distance Preserving Hashing,” *IEEE Transactions on Image Processing* (*TIP*), Aug. 2017.
- Min Wang, **Wengang Zhou**, Qi Tian, et al., "Linear Distance Preserving Pseudo-Supervised and Unsupervised Hashing," *ACM International Conference on Multimedia* (*MM*), pp. 1257-1266, long paper, 1257-1266, 2016.

Outline

- Background
- Motivation
- Our Work
- **Conclusion**

Conclusion

- Feature representation is the fundamental issue in image search
- Image search suffers a gap from image classification in labeled data to supervise deep learning
- Supervision clues can be designed to orient deep learning for search task
  - Refine the feature learning process
  - Generate better features for image search

Thank You!