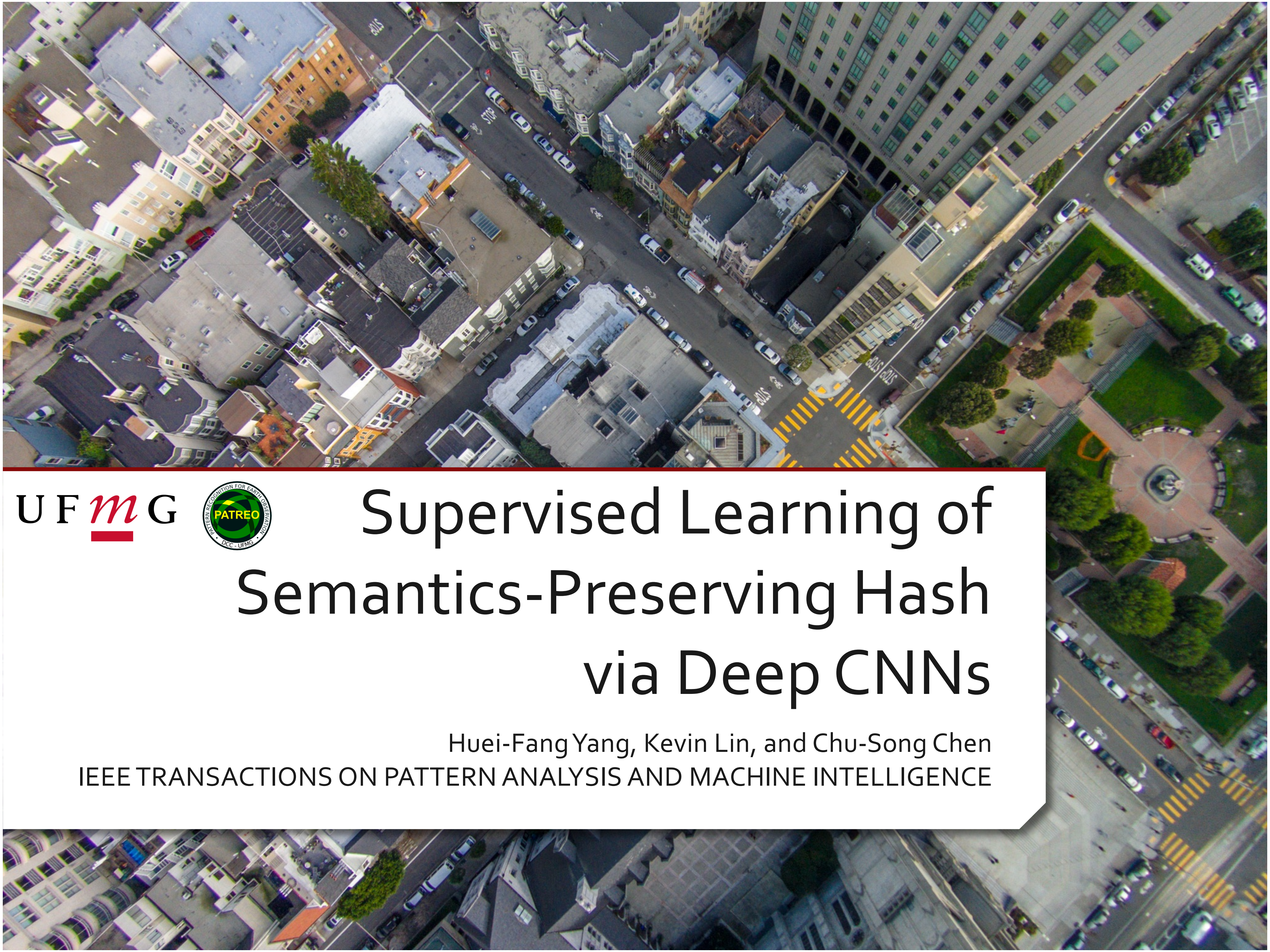


Supervised Learning of Semantics-Preserving Hash via Deep CNNs

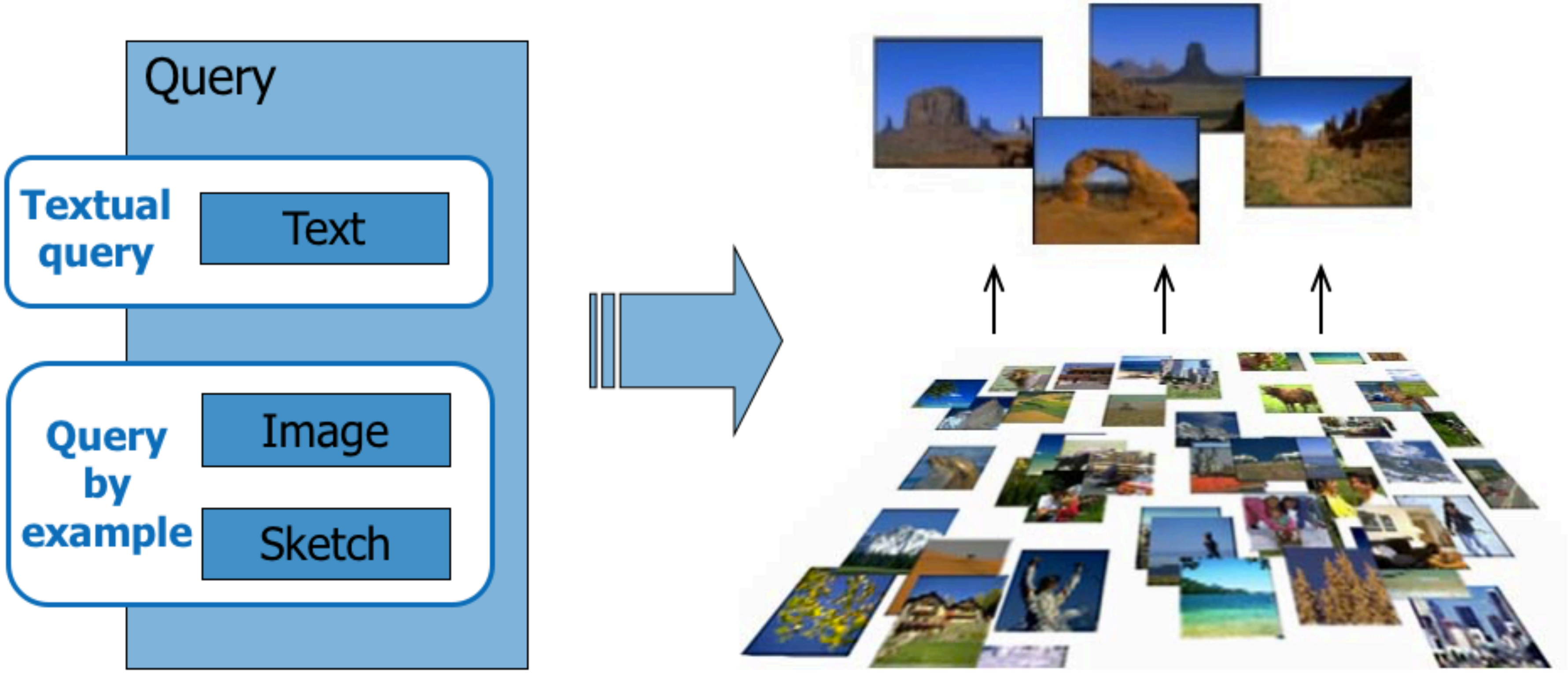
Huei-Fang Yang, Kevin Lin, and Chu-Song Chen

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

Summary

- ✓ Supervised deep hash approach to construct binary hash codes from labeled data for large-scale image search.
- ✓ Assumption: semantic labels are governed by several latent attributes on which classification relies.
- ✓ Classification and retrieval are unified.
- ✓ Joint learning of image representations, hash codes, and classification in a point-wised manner .

CBIR



Query

Textual query

Text

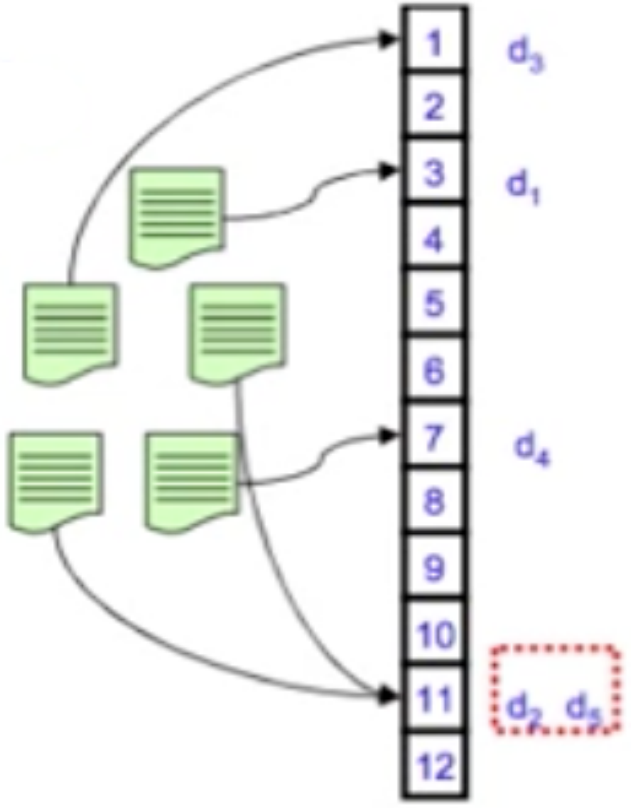
Query by example

Image

Sketch

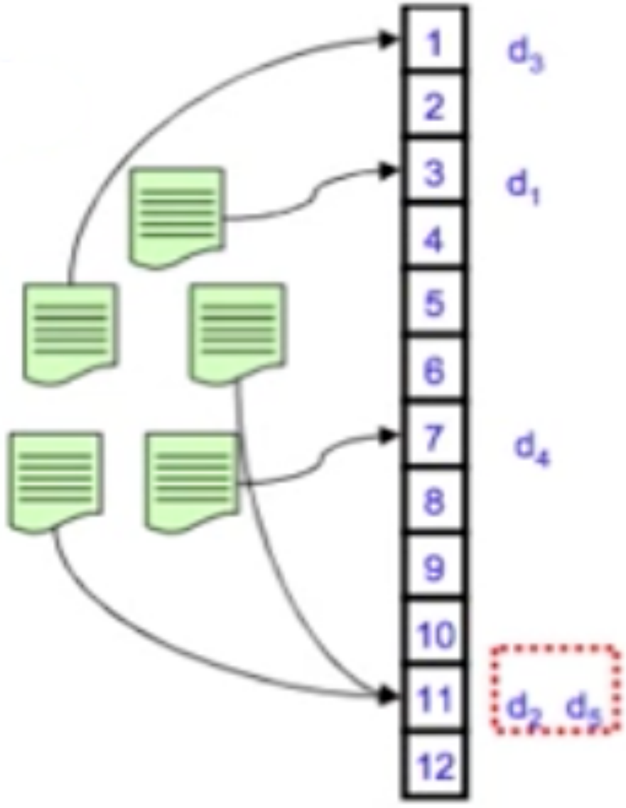
Hashing

- Hash function: any function $\mathcal{H}$ which has, as a minimum, the following two properties:
    - Compression
    - ease of computation



Hashing

- ✓ Hash function: any function $\mathcal{H}$ which has, as a minimum, the following two properties:
    - ✓ Compression
    - ✓ ease of computation
- ✓ Key principle in devising hash functions: map images of similar content to similar binary codes .



Learning to hash

✓ Learned binary codes are more efficient than the ones produced by locality sensitive hashing (LSH) .

Learning to hash

- ✓ Learned binary codes are more efficient than the ones produced by locality sensitive hashing (LSH) .
- ✓ Supervised hashing: exploits supervised information (e.g., pairwised similarities or triple-wised rankings devised by data labels) during the hash function construction.

Learning to hash

- ✓ Learned binary codes are more efficient than the ones produced by locality sensitive hashing (LSH) .
- ✓ Supervised hashing: exploits supervised information (e.g., pairwised similarities or triple-wised rankings devised by data labels) during the hash function construction.
- ✓ Pairs or triplets of the training samples: require long computation time and high storage cost for training.
- ✓ Learning binary codes in a point-wised manner would be a better alternative for the scalability of hash.

Deep Convolutional Neural Networks

- ✓ Capable of learning rich mid-level representations for image classification, object detection, and semantic segmentation.
- ✓ Transfer learning: feature extractors in new domains
- ✓ Fine-tuning
- ✓ *Inductive transfer learning:* one cannot learn how to walk before crawl, or how to run before walk.

Hypothesis

✓ Beyond classification, is the “pre-train + fine-tune” scheme also capable of learning binary hash codes for efficient retrieval? Besides, if it is, how to modify the architecture of a pre-trained CNN to this end?

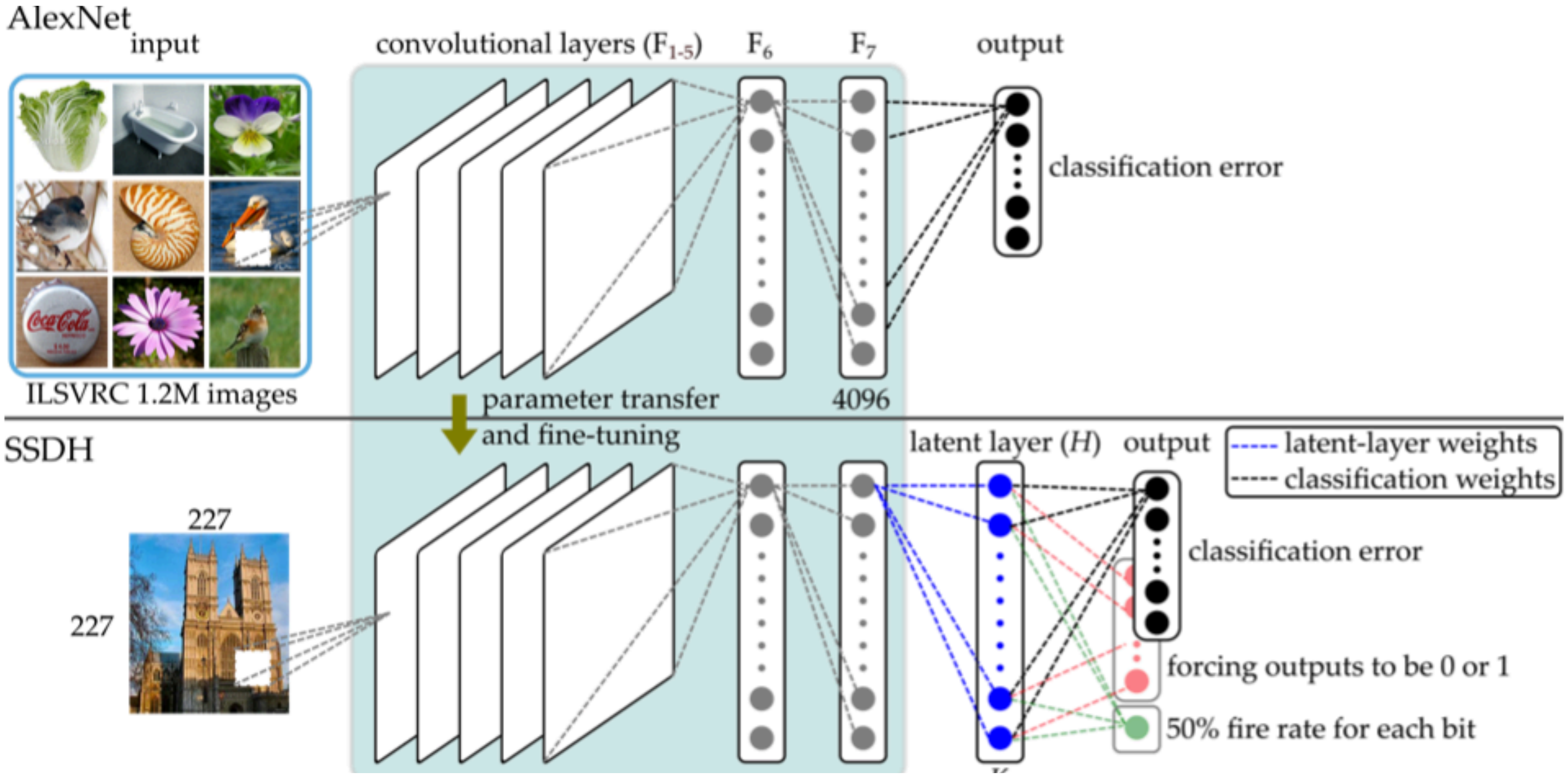
Semantics-preserving Deep Hashing (SSDH)

- ✓ Constructs hash functions as a hidden layer between image representations and classification outputs in a CNN
- ✓ Binary codes are learned by minimizing an objective function defined over classification error and other desired properties on the binary codes.

Semantics-preserving Deep Hashing (SSDH)

- ✓ Goal: design a supervised hashing algorithm that exploits the semantic labels to create binary codes of the following properties:
  - ✓ The codes respect the semantic similarity between image labels. Images that share common class labels are mapped to same (or close) binary codes.
  - ✓ The bits in a code are evenly distributed and discriminative.

Architecture overview



To incorporate the deep representations into the hash function learning, a *latent layer* H with K units is added to the top of layer F7.

Deep Hashing Functions

- ✓ This latent layer is fully connected to F_7 and uses the sigmoid units so that the activations are between 0 and 1.

- ✓ Let $W^H \in \mathbb{R}^{d \times K}$ denote the weights between F_7 and the latent layer. For a given image I_n with the feature vector $a^7{}_n \in \mathbb{R}^d$ in layer F_7, the activations of the units in H can be computed as $a^H{}_n = \sigma(a^7{}_n W^H + b^H)$, where $a^H{}_n$ is a K-dimensional vector, b^H is the bias term and $\sigma(.)$ is the logistic sigmoid function.

- ✓ The binary encoding function is given by:

$$b_n = (\mathrm{sgn}(\sigma(a^7{}_n W^H + b^H) - 0.5) + 1)/2$$

Label Consistent Binary Codes

- ✓ Assumption: semantic labels can be derived from a set of K latent concepts (or hidden attributes).

- ✓ When an input image is associated with binary-valued outputs (in $\{0,1\}^K$), the classification is dependent on these hidden attributes.

- ✓ Implication: through an optimization of a loss function defined on the classification error, ensure that semantically similar images are mapped to similar binary codes.

Label Consistent Binary Codes

- ✓ Consider a matrix $W^C \in \mathbb{R}^{K \times M}$ that performs a linear mapping of the binary hidden attributes to the class labels.

- ✓ Incorporating such matrix amounts to adding a classification layer to the top of the latent layer.

- ✓ In terms of the classification formulation, to solve W^C optimize the following objective function:

$$\arg\min_{W} E_1(W) = \arg\min_{W} \sum_{n=1}^{N} L(y_n, \hat{y}_n) + \lambda ||W||^2$$

L(.) = loss function, W = weights of the network, $\boldsymbol{\gamma}$ = importance of regularization term

Label Consistent Binary Codes

- ✓ The choice of the loss function depends on the problem itself.
- ✓ For single-label classification, uses AlexNet's cross-entropy error function:

$$L(y_n, \hat{y}_n) = -\sum_{m=1}^{M} y_{nm} \ln \hat{y}_{nm}$$

- ✓ For multi-label classification:

$$l(y_{nm}, \hat{y}_{nm}) = \begin{cases} 0 & y_{nm} = 1 \wedge \hat{y}_{nm} \geq 1 \\ 0 & y_{nm} = 0 \wedge \hat{y}_{nm} \leq 0 \\ \frac{1}{2}|y_{nm} - \hat{y}_{nm}|_p^p & \text{otherwise} \end{cases}$$

$$L(y_n, \hat{y}_n) = \sum_{m=1}^{M} l(y_{nm}, \hat{y}_{nm})$$

UFMG

PATREO

Efficient Binary Codes

- ✓ Besides semantically similar images having similar binary codes, we want the activation of each latent node to approximate to {0,1}. Since it has been activated by a sigmoid function, its value is inside the range [0,1].

- ✓ Going further: add the constraint of maximizing the sum of squared errors between the latent layer activations and 0.5:

$$\sum_{n=1}^{N} ||a_n^H - 0.5\mathbf{e}||^2$$

e =K-dimensional vector with all elements 1.

Efficient Binary Codes

- ✓ To make the binary codes balanced, we hope that there is no preference for the hidden values to be 0 or 1.
- ✓ That is, the occurrence probability of each bit's on or off is the same, or the entropy of the discrete distribution is maximized.
- ✓ We want each bit to fire 50% of the time via minimizing:

$$\sum_{n=1}^{N} (\text{mean}(a_n^H) - 0.5)^2$$

Efficient Binary Codes

- ✓ In sum, we aim to optimize the following objective to obtain the binary codes:

$$\arg\min_{W} -\frac{1}{K}\sum_{n=1}^{N} ||a_n^H - 0.5\mathbf{e}||_p^p + \sum_{n=1}^{N} |\text{mean}(a_n^H) - 0.5|^p$$
$$= \arg\min_{W} -E_2(W) + E_3(W),$$

- ✓ The first term encourages the activations of the units in H to be close to either 0 or 1 and the second term further ensures that the output of each node has a nearly 50% chance of being 0 or 1.

- ✓ Each loss term is contributed by only an individual training sample and no cross-sample terms are involved. Hence, the objective remains point-wised and can be minimized through SGD efficiently by dividing the training samples (but not pairs or triples of them) into batches.

Overall Objective and Implementation

$$\arg\min_{W} E_1(W) = \arg\min_{W} \sum^{N} L(y_n, \hat{y}_n) + \lambda||W||^2$$

$$\arg\min_{W} -\frac{1}{K}\sum_{n=1}^{N} ||a_n^H - 0.5\mathbf{e}||_p^p + \sum_{n=1}^{N} |\mathrm{mean}(a_n^H) - 0.5|^p$$

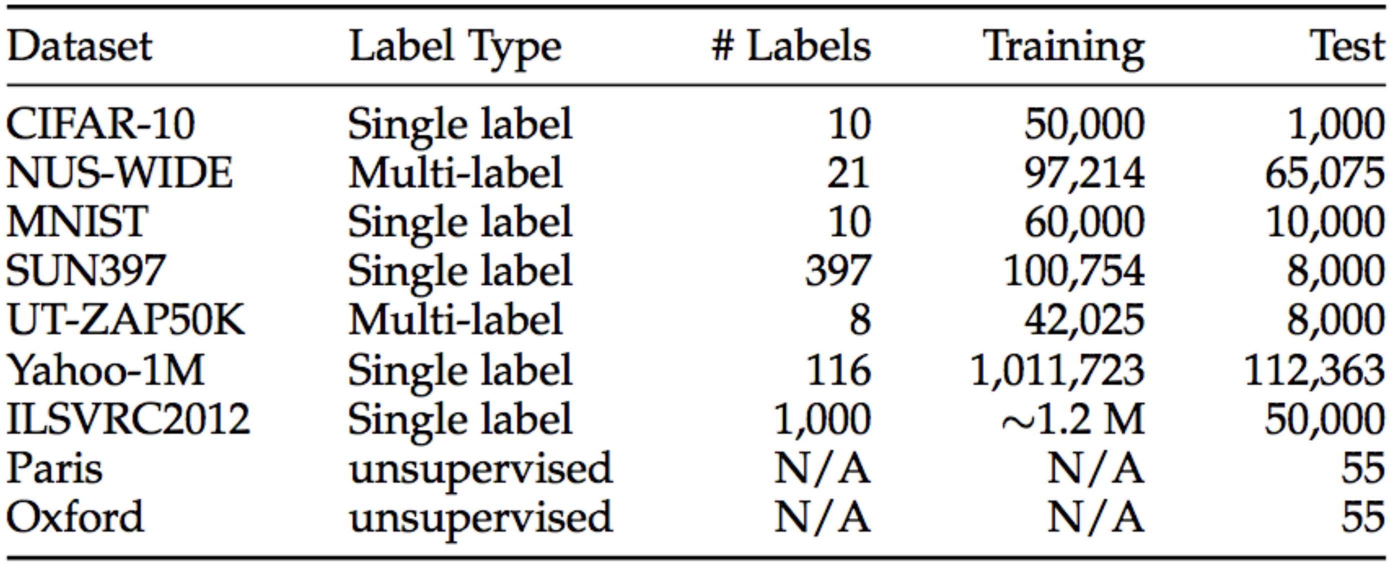
$$= \arg\min_{W} -E_2(W) + E_3(W),$$

Entire objective function:

$$\arg\min_{W} \quad \alpha E_1(W) - \beta E_2(W) + \gamma E_3(W)$$

The initial weights in layers F$_{1\text{-}7}$ of the network are set as the pre-trained ones and the remaining weights are randomly initialized. SGD is applied in conjunction with backpropagation with mini-batches.

Experiments



Dataset	Label Type	# Labels	Training	Test
CIFAR-10	Single label	10	50,000	1,000
NUS-WIDE	Multi-label	21	97,214	65,075
MNIST	Single label	10	60,000	10,000
SUN397	Single label	397	100,754	8,000
UT-ZAP50K	Multi-label	8	42,025	8,000
Yahoo-1M	Single label	116	1,011,723	112,363
ILSVRC2012	Single label	1,000	$\sim$1.2 M	50,000
Paris	unsupervised	N/A	N/A	55
Oxford	unsupervised	N/A	N/A	55

CIFAR-10: tiny objects
NUS-WIDE: web images
MNIST: handwritten digits
UT-ZAP50K: catalog images
SUN397, Oxford, and Paris: scene images
The large datasets, Yahoo-1M and ILSVRC: product and object images with heterogeneous types, respectively.

Evaluation protocols

- ✓ Mean average precision (mAP): indicator of the overall performance of hash functions;
- ✓ Precision at k samples: It is computed as the percentage of true neighbors among the top k retrieved images;
- ✓ Precision within Hamming radius r: We compute the precision of the images in the buckets that fall within the Hamming radius r of the query image, where r = 2 is selected as previous works did.

Results

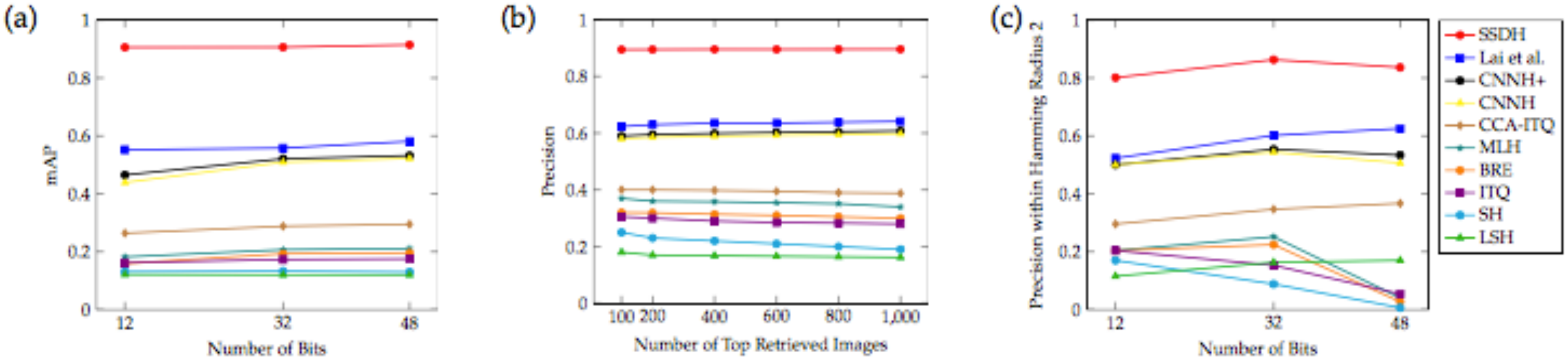


Fig. 4. Comparative evaluation of different hashing algorithms on the CIFAR-10 dataset. (a) mAP curves with respect to different number of hash bits. (b) Precision curves with respect to different number of top retrieved samples when the 48-bit hash codes are used in the evaluation. (c) Precision within Hamming radius 2 curves with respect to different number of hash bits.

Results

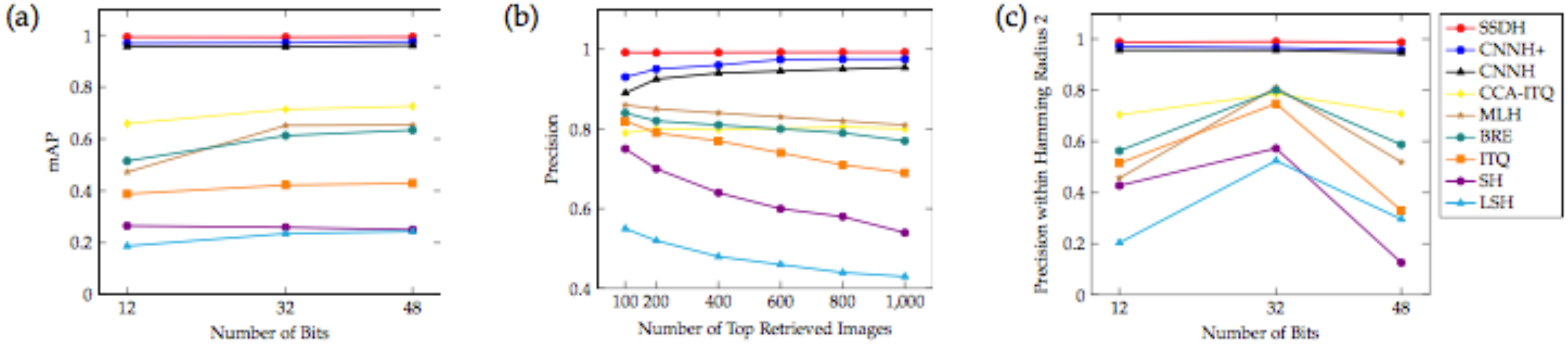


Fig. 5. Comparative evaluation of different hashing algorithms on the MNIST dataset. (a) mAP curves with respect to different number of hash bits. (b) Precision curves with respect to different number of top retrieved samples when the 48-bit hash codes are used in the evaluation. (c) Precision within Hamming radius 2 curves with respect to different number of hash bits.

Results

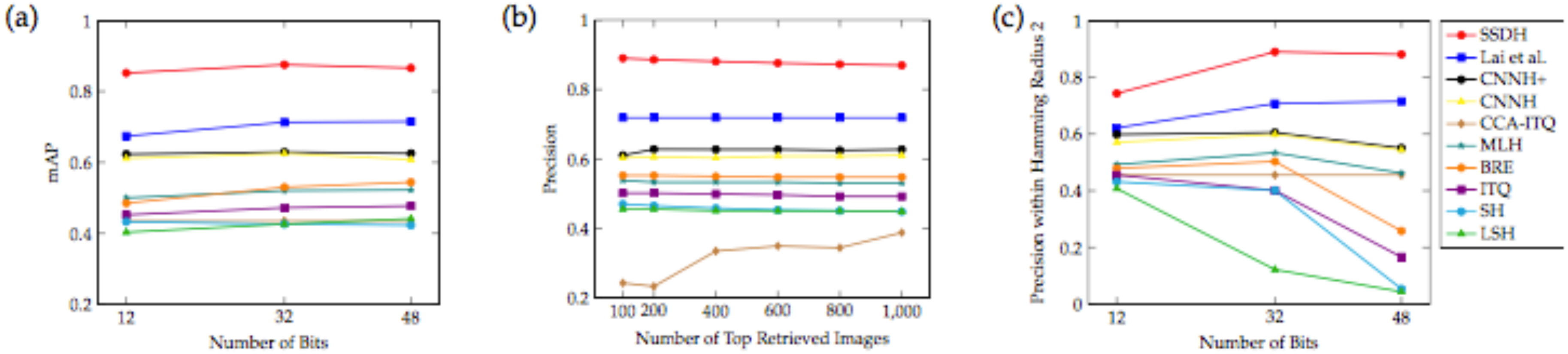


Fig. 6. Comparative evaluation of different hashing algorithms on the NUS-WIDE dataset. (a) mAP curves of top 5,000 returned images with respect to different number of hash bits. (b) Precision curves with respect to different number of top retrieved samples when the 48-bit hash codes are used in the evaluation. (c) Precision within Hamming radius 2 curves with respect to different number of hash bits.

Results

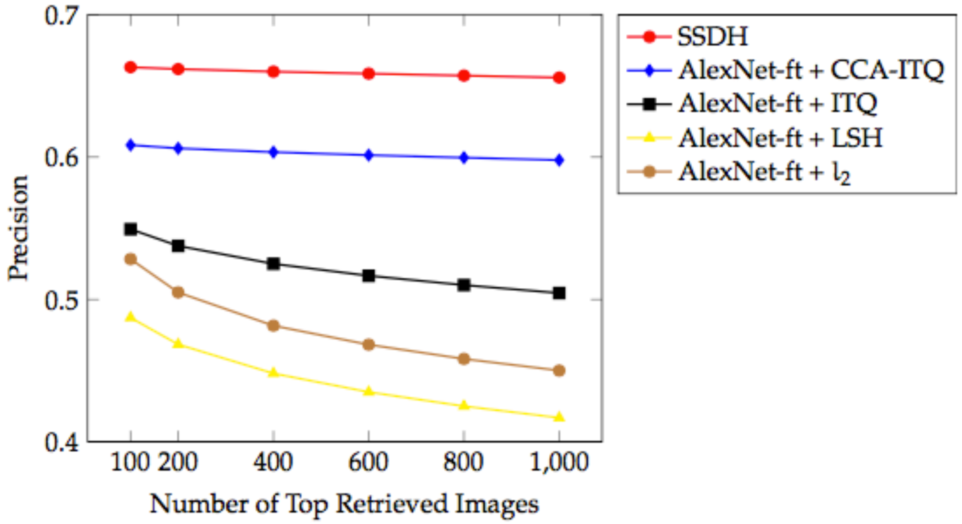


Fig. 8. Precision curves with respect to different number of top retrieved samples on the Yahoo-1M dataset when the 128-bit hash codes are used in the evaluation. AlexNet-ft denotes that the features from layer F_7 of AlexNet fine-tuned on Yahoo-1M are used in learning hash codes.

Results

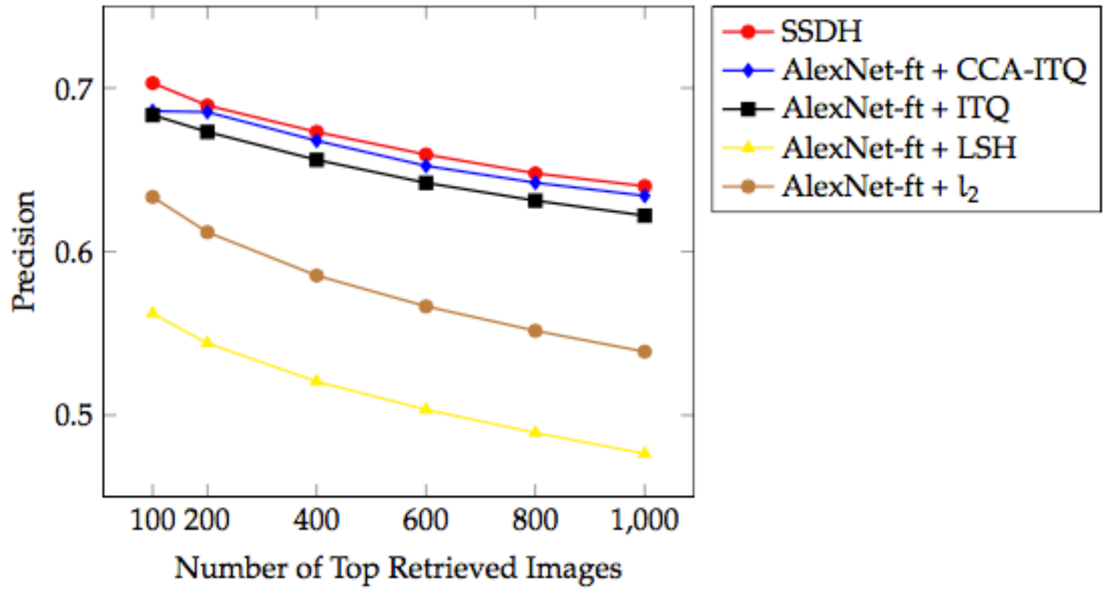


Fig. 9. Precision curves with respect to different number of top retrieved samples on the UT-ZAP50K dataset when the 48-bit hash codes are used in the evaluation. AlexNet-ft denotes that the features from layer F_7 of AlexNet fine-tuned on UT-ZAP50K are used in learning hash codes.

Acknowledgments

PATTERN RECOGNITION AND EARTH OBSERVATION
PATREO
DCC - UFMG

UFMG

30 ANOS
FAPEMIG
1986 - 2016

CAPES

CNPq
Conselho Nacional de Desenvolvimento
Científico e Tecnológico