Jain *et al.* (ICCV 2017), "SuBiC: A supervised, structured binary code for image search"

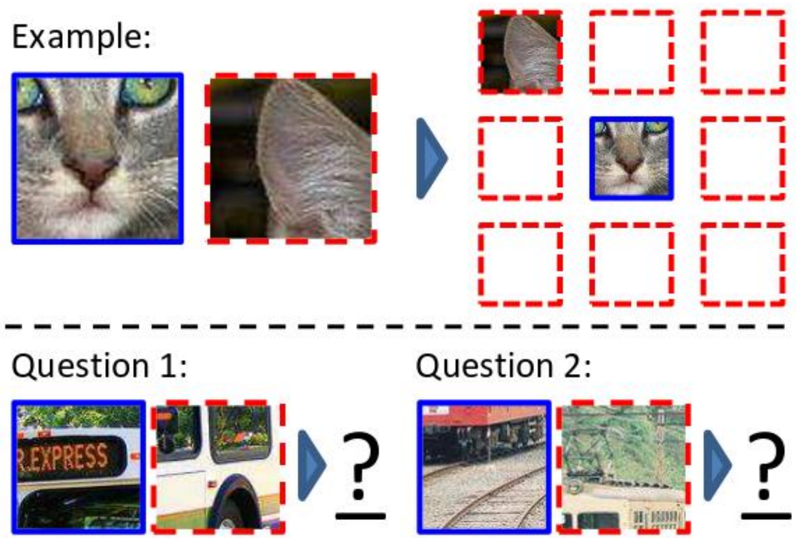
20183385 Huisu Yun

30 October 2018

CS688 Fall 2018 Student Presentation

Review: Doersch *et al.* (ICCV 2015)

- “[S]patial context as a source of [...] signal for training a rich visual representation”



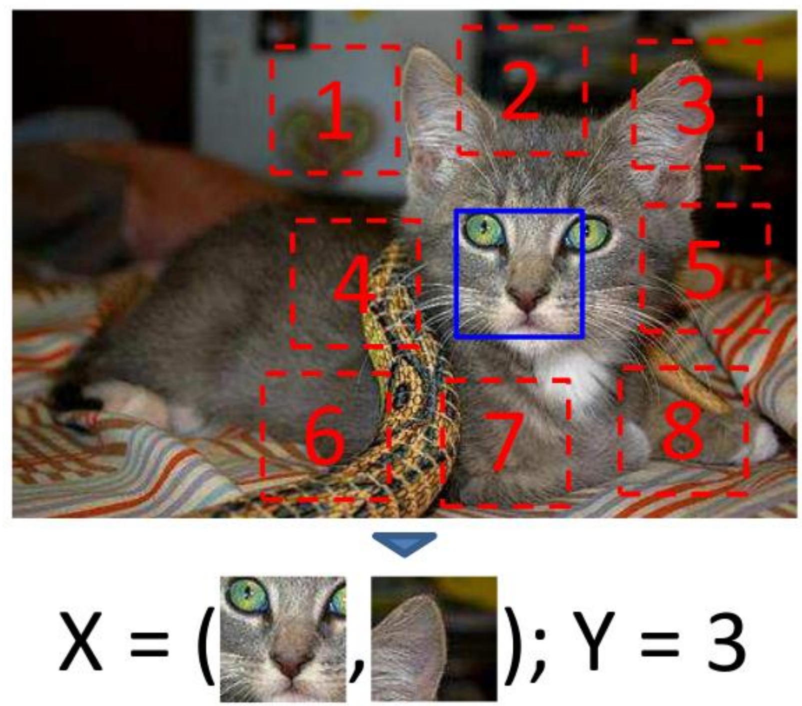


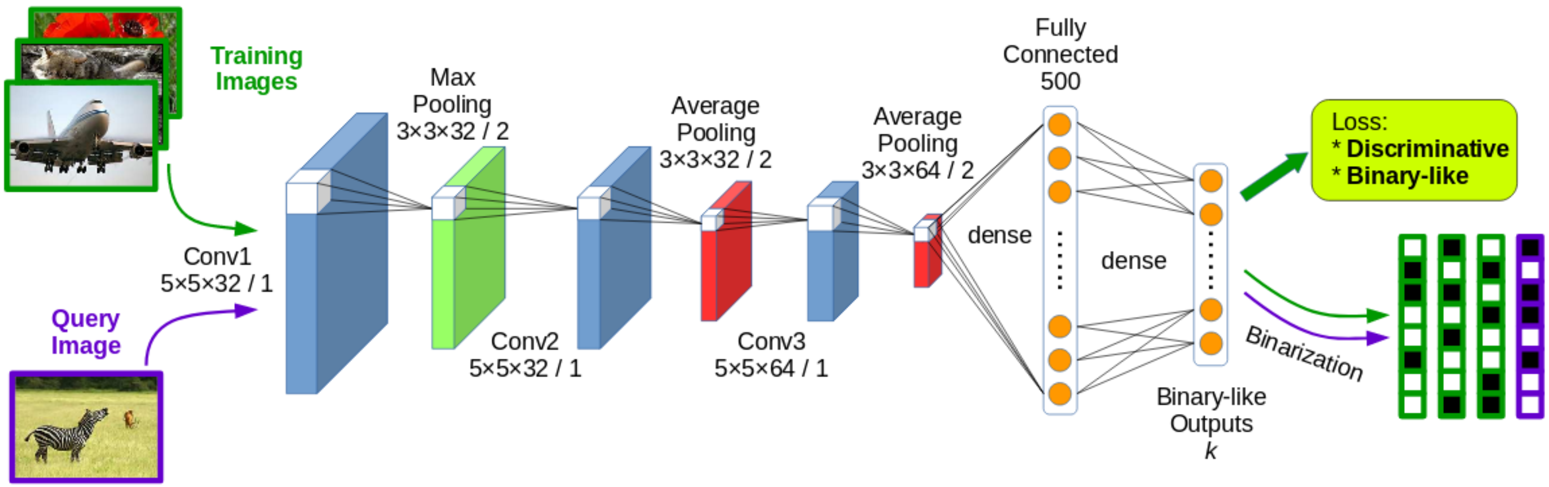
 Image reproduced from Doersch et al. *2015. “Unsupervised visual representation learning by context prediction”*

Motivation

- Raw feature vectors are **very long** (*cf.* PA2)
  - ...which is why we want to use specialized binary codes
- Binary codes for image search (*cf.* lecture slides)
  - ...should be of **reasonable length**
  - ...and provide **faithful representation**

Background: Supervised codes (1/2)

- Liu *et al.* (CVPR 2016): pairwise supervision



Pairwise loss function

(Hamming distance approximated using Euclidean distance)

$$L_r(\mathbf{b}_1, \mathbf{b}_2, y) = \frac{1}{2}(1-y)||\mathbf{b}_1 - \mathbf{b}_2||_2^2$$

Similar images—similar codes

$$+\frac{1}{2}y\ \max(m - ||\mathbf{b}_1 - \mathbf{b}_2||_2^2,\ 0)$$

Dissimilar images—different codes

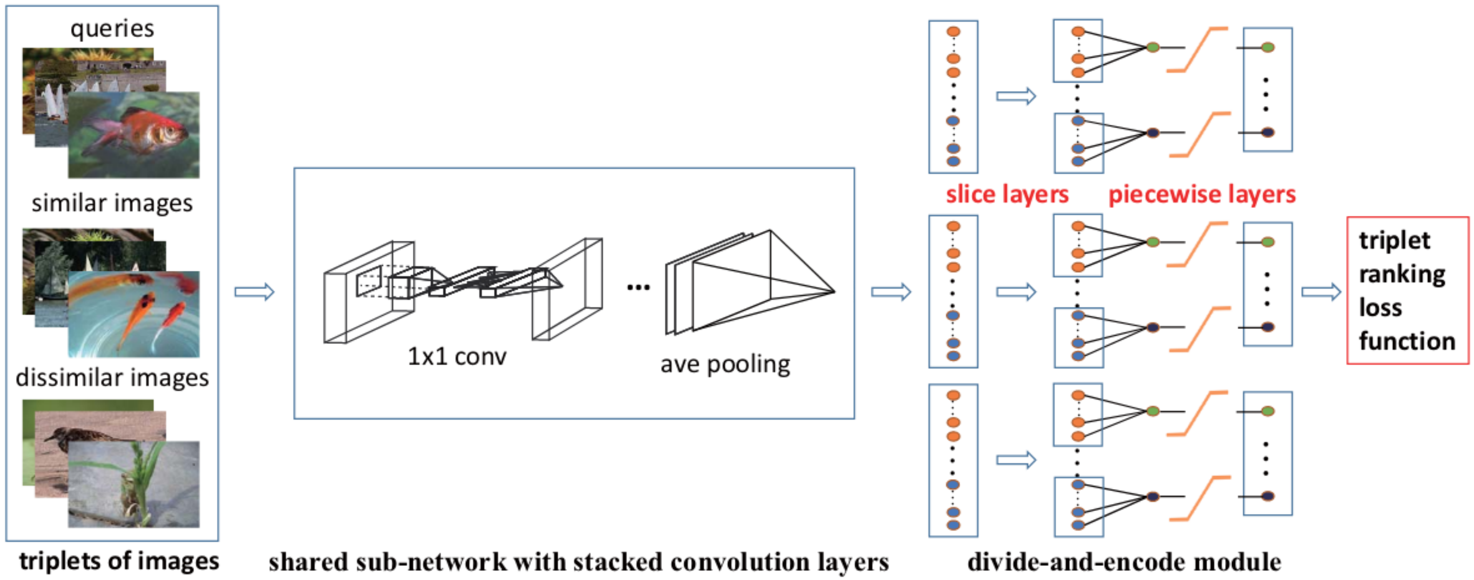
$$+\alpha(||\,|\mathbf{b}_1| - \mathbf{1}||_1 + ||\,|\mathbf{b}_2| - \mathbf{1}||_1)$$

Regularization (+1 or −1)

Image reproduced from Liu et al. *2016. "Deep supervised hashing for fast image retrieval"*

Background: Supervised codes (2/2)

- Lai *et al.* (CVPR 2015): triplet supervision



Triplet ranking loss

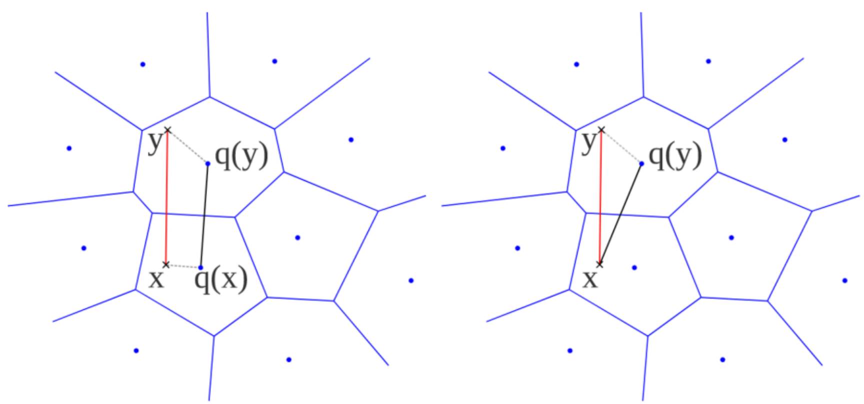
$$\begin{aligned} &\ell_{triplet}(\mathcal{F}(I), \mathcal{F}(I^+), \mathcal{F}(I^-)) \\ =&\max(0, ||\mathcal{F}(I) - \mathcal{F}(I^+)||_2^2 - ||\mathcal{F}(I) - \mathcal{F}(I^-)||_2^2 + 1) \\ &s.t.\ \mathcal{F}(I),\ \mathcal{F}(I^+),\ \mathcal{F}(I^-) \in [0,1]^q. \end{aligned}$$

 Image reproduced from Lai et al. *2015. "Simultaneous feature learning and hash coding with deep neural networks"*

Background: Vector quantization

- Group similar vectors
  - ...such that each group has approximately the same members
  - Vectors are represented by the group (centroid) they belong to
- Jégou *et al.* (TPAMI 2011): Product Quantization (PQ)
  - Split the vector into small subvectors; quantize them separately
  - Results in **structured codes** *(why?)*

Symmetric distance computation



Asymmetric distance computation

 Image reproduced from Jégou et al. *2011. "Product quantization for nearest neighbor search"*

Introduction

- SuBiC — Supervised, structured binary codes
  - Supervised: trained such that **class labels** can be predicted; **point-wise supervision**
  - Structured: one-hot blocks (*cf.* quantized subvectors in PQ)

*One-hot blocks (*cf. *PQ)*

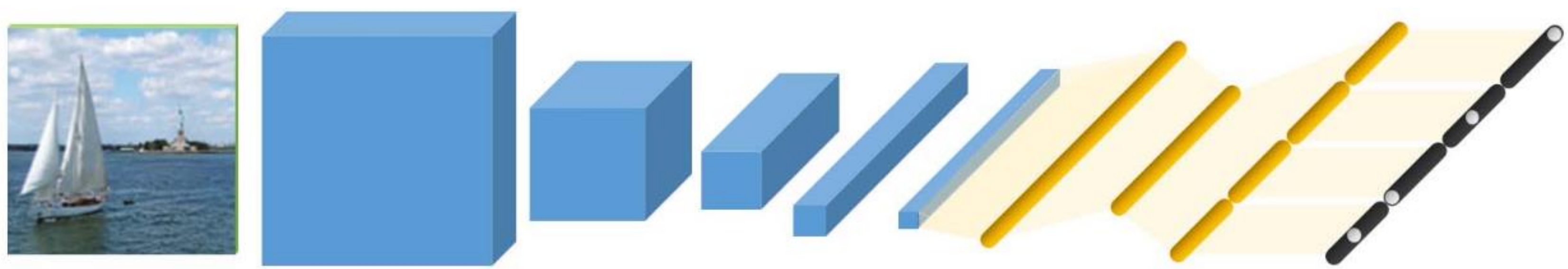
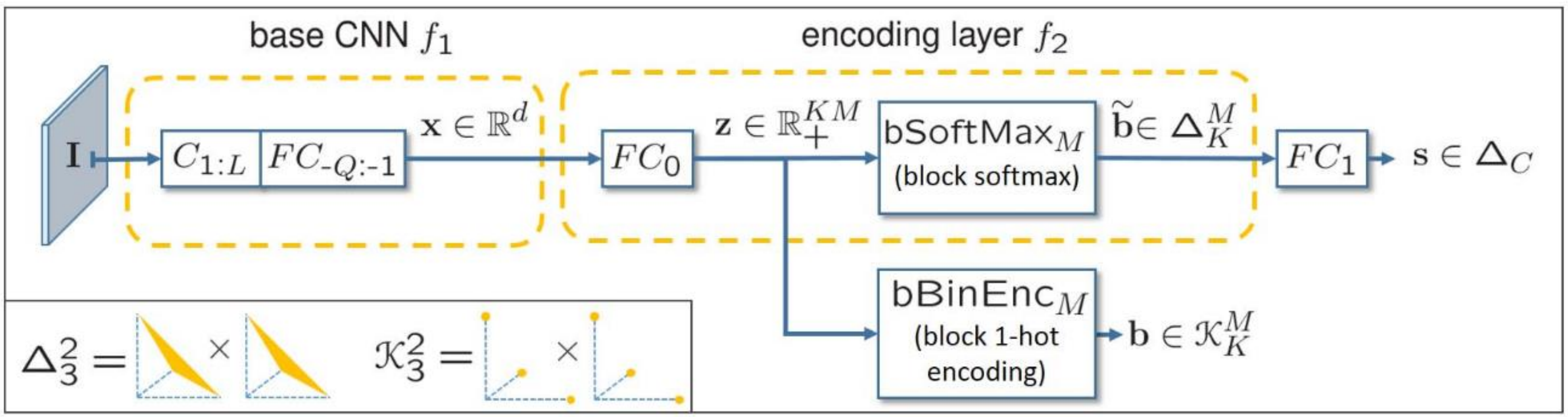


 Image reproduced from Jain et al. *2017. "SuBiC: A supervised, structured binary code for image search"*

Overview

- Code length: *KM* (*M* blocks, each having *K* dimensions)
  - Training time: produced by **block softmax nonlinearity**
  - Test time: produced by **block one-hot encoder**



$\Delta_K \triangleq \{\mathbf{d} \in [0,1]^K \text{ s.t. } \|\mathbf{d}\|_1 = 1\}$ *Convex hull of below (training time output)*

$\mathcal{K}_K \triangleq \{\mathbf{d} \in \{0,1\}^K \text{ s.t. } \|\mathbf{d}\|_1 = 1\}$ *Set of one-hot vectors (test time output)*

$$\widetilde{\mathbf{b}}_m = \frac{1}{\|\exp(\mathbf{z}_m)\|_1} \exp(\mathbf{z}_m)$$

$$\mathbf{b}_m[k] = \begin{cases} 1 & \text{if } k = \operatorname{argmax}_r \mathbf{z}_m[r] \\ 0 & \text{otherwise,} \end{cases}$$

 Image reproduced from Jain et al. *2017. "SuBiC: A supervised, structured binary code for image search"*

Training

- Newly introduced entropy-based losses
  - **Mean entropy loss** (weighted by γ): for one-hot structure
  - **Batch entropy loss** (weighted by μ): for uniform block support
- Cross entropy loss
  - Our usual choice for classification problems

$$\mathrm{Loss}\big(\{(\mathbf{I}^{(i)}, y^{(i)})\}_{i\in\mathcal{T}}\big) \triangleq \frac{1}{T}\sum_{i\in\mathcal{T}}\Big[\underbrace{\ell(\mathbf{s}^{(i)}, y^{(i)})}_{\text{Classification loss}} + \underbrace{\frac{\gamma}{M\log_2 K}\mathrm{E}(\widetilde{\mathbf{b}}^{(i)})}_{\text{Mean entropy loss}} - \underbrace{\frac{\mu}{M\log_2 K}\mathrm{E}(\overline{\mathbf{b}})}_{\text{Batch entropy loss}}\Big]$$

Cross entropy

$$\ell(\mathbf{s}, y) \triangleq -\frac{1}{\log_2 C}\log_2 \mathbf{s}[y]$$

Image search with SuBiC

- While the code length in the SuBiC neural network architecture is KM, the actual storage footprint of the produced codes can be easily reduced to $M \log_2 K$
  - *e.g.* the 16-bit code ((0, 0, 0, 0, 0, 0, 0, 1), (0, 0, 1, 0, 0, 0, 0, 0)) can be compacted to (7, 2) = ((1, 1, 1), (0, 1, 0)) of length 6

- Only M additions required for asymmetric distance computation (*i.e.* between a binary code and its real-valued cousin)

Results

Method	12-bit	24-bit	36-bit	48-bit
CNNH+ [45]	0.5425	0.5604	0.5640	0.5574
DLBHC [32]	0.5503	0.5803	0.5778	0.5885
DNNH [31]	0.5708	0.5875	0.5899	0.5904
DSH [33]	0.6157	0.6512	0.6607	0.6755
KSH-CNN [35]	-	0.4298	-	0.4577
DSRH [48]	-	0.6108	-	0.6177
DRSCH [46]	-	0.6219	-	0.6305
BDNN [17]	-	0.6521	-	0.6653
SuBiC (ours)	**0.6349**	**0.6719**	**0.6823**	**0.6863**

Table 2: **Single-domain category retrieval.** Comparison against published mAP values on Cifar-10 for various supervised deep hashing methods. See the *ImageNet* column of Table 3 for single-domain results on ImageNet.

Method	VOC2007	Caltech-101	ImageNet
PQ [24]	0.4965	0.3089	0.1650
CKM [38]	0.4995	0.3179	0.1737
LSQ [37]	0.4993	0.3372	0.1882
DSH-64 [33]	0.4914	0.2852	0.1665
SuBiC 2-layer	**0.5600**	0.3923	0.2543
SuBiC 3-layer	0.5588	**0.4033**	**0.2810**

Table 3: **Cross-domain category retrieval.** Performance (mAP) using 64-bit encoders across three different datasets using VGG-128 as base feature extractor. For completeness, results on ImageNet validation set (*i.e.* single-domain retrieval) are provided in the third column.

[Table 2] K = 64; M = one of {2, 4, 6, 8}

Method	Oxford5K	Paris6K
PQ [24]	0.2374	0.3597
LSQ [37]	0.2512	0.3764
DSH-64 [33]	0.2108	0.3287
SuBiC	**0.2626**	**0.4116**

Table 4: **Instance retrieval.** Performance (mAP) comparison using 64-bit codes for all methods.

	ImageNet		VOC2007
	Top-1 acc.	*Top-5 acc.*	*mAP*
VGG-128*	53.80	77.32	73.79
PQ 64-bit	39.88	67.22	65.94
CKM 64-bit	41.15	69.66	67.25
SuBiC soft*	50.07	74.11	70.20
SuBiC 64-bit	47.77	72.16	67.86

Table 5: **Classification performance with different compact codes**. The rows marked (*) are non-binary codes. See the text for details.

[Table 5] SuBiC soft: using the block softmax nonlinearity instead of block one-hot encoder in test architecture

Discussion

- Combining the self-structuring properties of unsupervised learning with the strength of supervised deep hashing approaches

- The decent cross-domain performance would make SuBiC a good candidate for use in systems without much parallelism (*e.g.* GPU assistance) available
  - However, the block one-hot structure might be an obstacle; deep hash codes might be faster to compare on modern CPUs