VALSE webinar， 2015年5月27日

Feature Selection in Image and Video Recognition

Jianxin Wu

National Key Laboratory for Novel Software Technology

Nanjing University

Learning And Mining from DatA

http://lamda.nju.edu.cn

南京大學

NANJING UNIVERSITY

1902

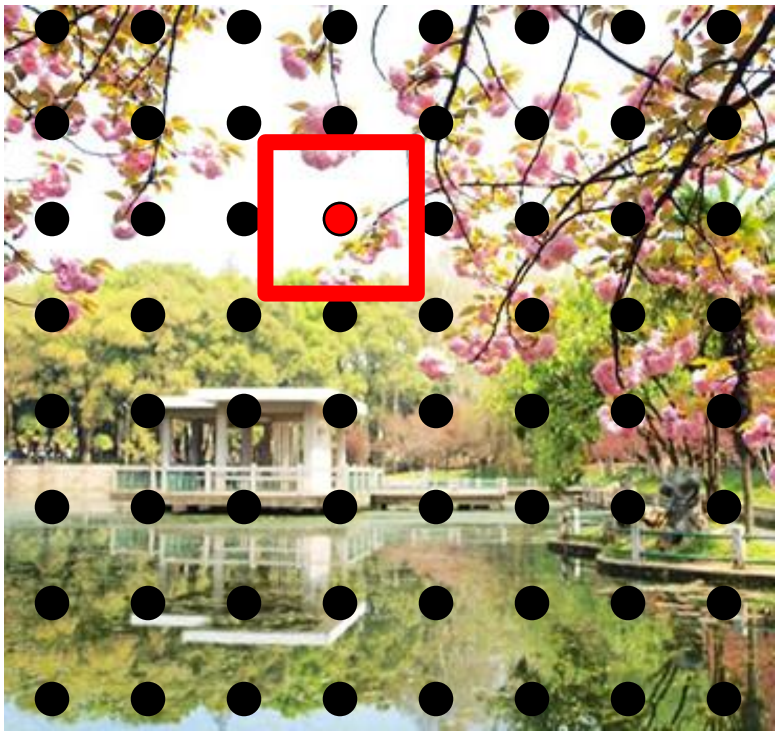
Introduction

For image classification, how to represent an image?

With

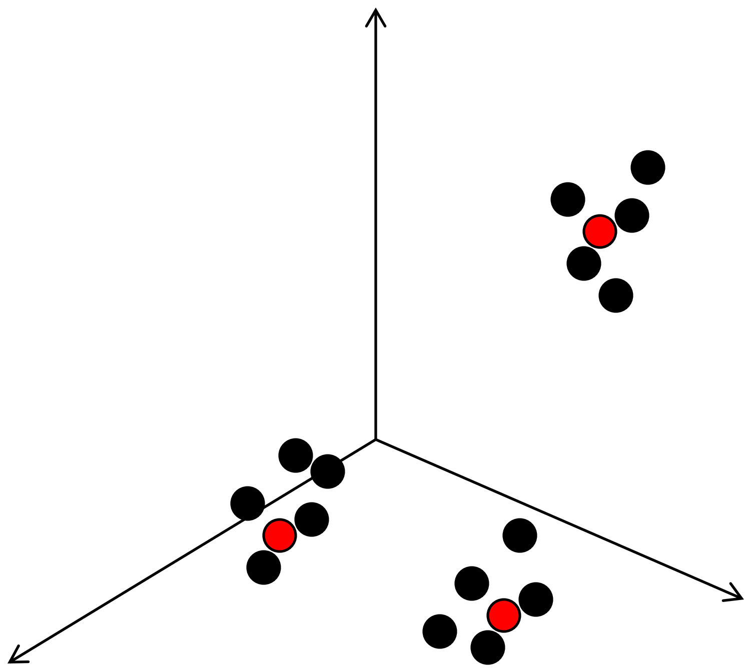
- strong discriminative power; and,
- manageable storage and CPU costs

Bag of words



- Dense sample
- Extract visual descriptor (e.g. SIFT or CNN) at every sample location, usually PCA to reduce dimensionality
- Learning a visual codebook by k-means

The VLAD pipeline



- K code words $\boldsymbol{c}_i \in \mathbb{R}^D$
- Pooling

$$\boldsymbol{f}_i = \sum_{\boldsymbol{x} \in \boldsymbol{c}_i} (\boldsymbol{x} - \boldsymbol{c}_i)$$

- Concatenation

$$[\boldsymbol{f}_1 \boldsymbol{f}_2 \cdots \boldsymbol{f}_K]$$

- Dimensionality: $D \times K$

Jegou et al. Aggregating local images descriptors into compact codes. TPAMI, 2012

Effect of High Dimensionality

- Blessing
  - Fisher Vector: $K \times (2D + 1)$
  - Super Vector: $K \times (D + 1)$
  - State-of-the-art results in many application domains

- Curse
  - 1 million images
  - 8 spatial pyramid regions
  - $K = 256,\ D = 64,$ 4 bytes to store a floating number

  - **1056G bytes!**

J. Sanchez *et al*. Image classification with the fisher vector: Theory and practice. IJCV, 2013.
X. Zhou *et al*. Image classification using super-vector coding of local image descriptors. ECCV, 2010.

Solution?

- Use fewer example / dimensions?
  - Reduce accuracy quickly
- Feature compression
  - Introduction soon
- Feature selection
  - This talk

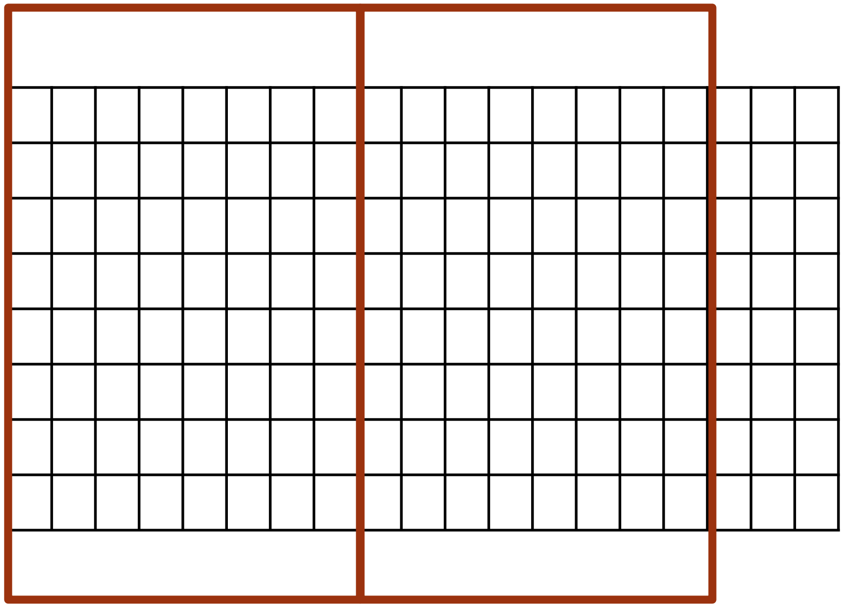
To compress?

Methods in the literature: feature compression

Compress the long feature vectors so that

- Much fewer bytes to store them
- (possibly) faster learning

Product Quantization illustration



- For every 8 dimensions
  1. Generate a codebook with 256 words
  2. VQ a 8d vector (32 bytes) into a index (1 byte)

- On-the-fly decoding
  1. Get stored index $\boldsymbol{i}$
  2. Expand into 8d $\boldsymbol{c}_i$

Do not change learning time

Jegou *et al*. Product quantization for nearest neighbor search. TPAMI, 2011.
Vedaldi & Zisserman. Sparse kernel approximations for efficient classification and detection. CVPR, 2012.

Thresholding

- A simple idea

$$x \leftarrow \begin{cases} -1, & x < 0 \\ +1, & x \geq 0 \end{cases}$$

- 32 times compression

- Working surprisingly well!

- But, why?

Perronnin *et al*. Large-scale image retrieval with compressed Fisher vectors. CVPR, 2010.

Bilinear projections (BPBC)

- FV or VLAD requires rotation
  - A large matrix times the long vector
- Bilinear projection + binary feature
  - Example: KD vector $\boldsymbol{x}$ reshape into $K \times D$ matrix X
  - Bilinear projection / rotation

$$\mathrm{sgn}(R_1^T X R_2)$$

  - $R_1: K \times K$, $R_2: D \times D$
  - Smaller storage and faster computation than PQ
- But, learning R is very time consuming (circulant?)

Gong *et al*. Learning binary codes for high-dimensional data using bilinear projections. CVPR, 2013.

The commonality

- Linear projection!
  - *New features are linear combinations of multiple dimensions from the original vector*
- What does this mean?
  - *Assuming strong multicollinearity exists!*
- Is this true in reality?

Collinearity and multicollinearity

Examining real data find that:

- Collinearity almost never exist
- Too expensive to examine the existence of multicollineairty, but we have something to say

Collinearity

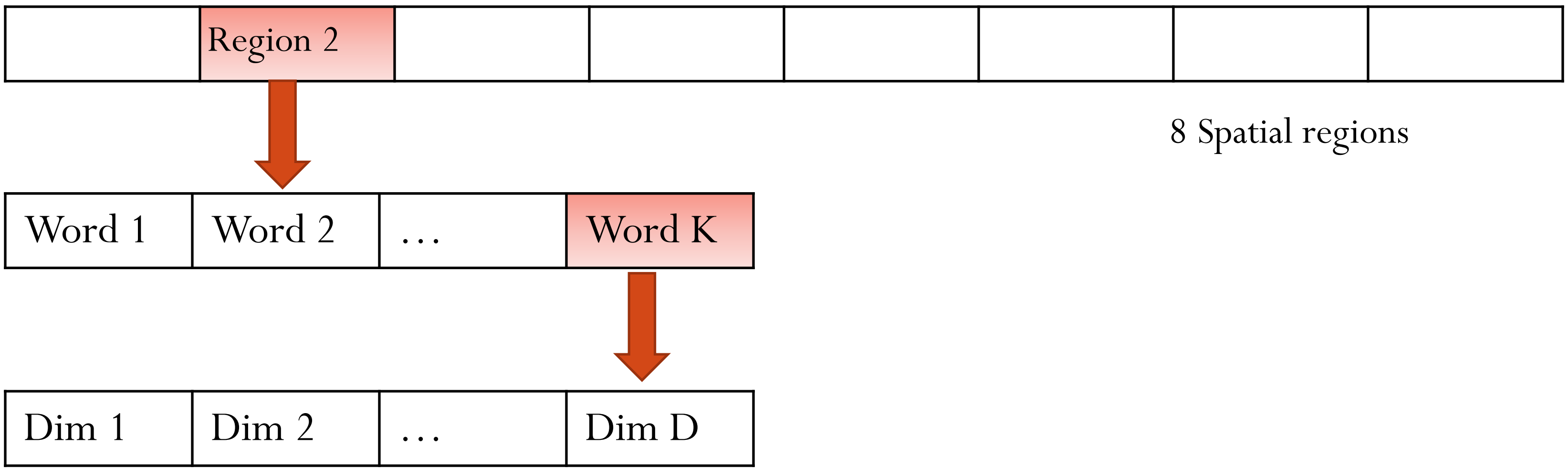
- Existence of strong linear dependencies between two dimensions in the VLAD / FV vector

- Pearson's correlation coefficient

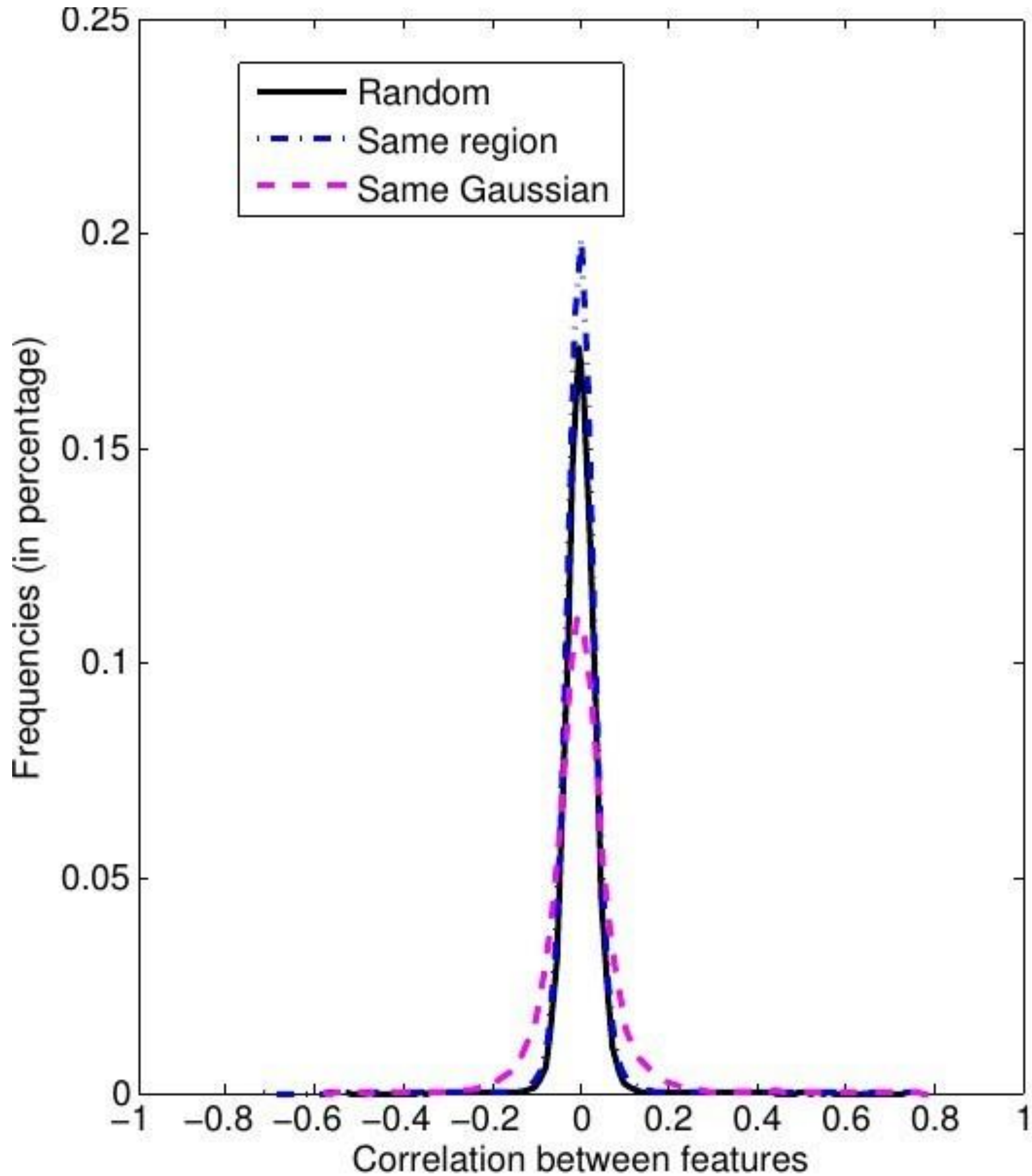
$$r = \frac{\boldsymbol{x}_{:i}^T \boldsymbol{x}_{:j}}{\|\boldsymbol{x}_{:i}\| \|\boldsymbol{x}_{:j}\|}$$

  - $r = \pm 1$: perfect collinearity
  - $r = 0$: no linear dependency at all

Three types of checks



1. Random pair
2. In the same spatial region
3. In same code word / Gaussian component (all regions)



- Same Gaussian shows a little stronger correlation
- Mostly no correlation at all!

From 2 to n

- Multicollinearity – strong linear dependency among > 2 dimensions

- Given the missing of collinearity, the chance of multicollinearity is also small

- PCA is essential for FV and VLAD
  - Dimensions in PCA are uncorrelated

- Thus, we should choose, not compress!

MI based feature selection

A simple mutual information based importance sorting algorithm to choose features

- Computationally very efficient
- When ratio changes, no need to repeat
- Highly accurate

Yes, to choose!

- Choose is better than compress
  - Given that multicollinearity is missing

- Cannot afford expensive feature selection
  - Features too big to put into memory
  - Complex algorithms take too long

Usefulness measure

- Mutual information

$$I(\boldsymbol{x}, \boldsymbol{y}) = H(\boldsymbol{x}) + H(\boldsymbol{y}) - H(\boldsymbol{x}, \boldsymbol{y})$$

  - H: entropy
  - $\boldsymbol{x}$: one dimension
  - $\boldsymbol{y}$: image label vector

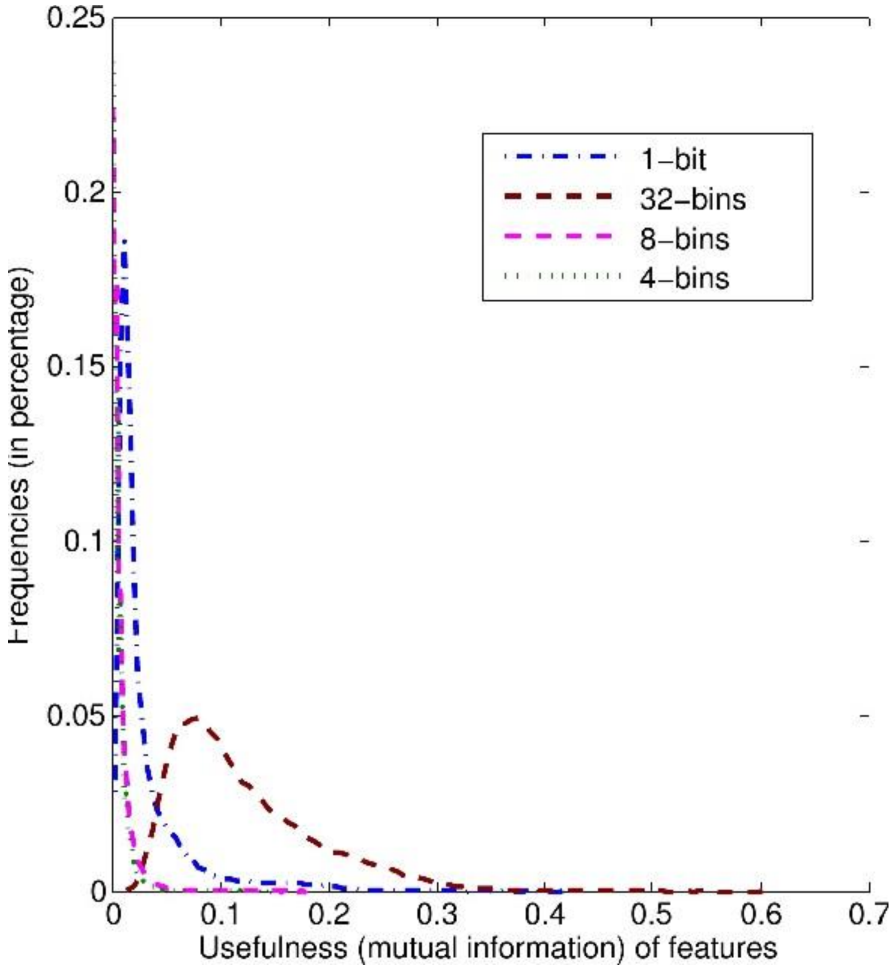
- Selection
  - Sort all MI values, choose the top D'
  - Only one pass of data
  - No addition work if D' changes

Entropy computation

- Too expensive using complex methods
  - e.g. kernel density estimation

- Use discrete quantization
  - 1-bit: $x \leftarrow \begin{cases} -1, & x < 0 \\ +1, & x \geq 0 \end{cases}$
  - N-bins: uniformly quantize into N bins
  - 1-bit and 2-bins are different

  - Discrete entropy: $H = -\sum_j p_j \log_2 p_j$
  - Larger N, bigger H value



- Most features are not use
- Choose a small subset is not only for speed or scalability, but also for accuracy!
- 1-bit >> 4/8 bins – keep the threshold at 0 is important!

The pipeline

1. Generate a FV / VLAD vector
2. Only keep the chosen D' dimensions
3. Further quantize the D' dimensions into D' bits

- Compression ratio is $\frac{32D}{D'}$
- Store 8 bits in a byte

Image Results

- Much faster in feature dimensionality reduction, learning
- Requires almost no extra storage
- In general, significantly higher accuracy with same ratio

Features

- Use the Fisher Vector
- D=64
  - 128 dim SIFT, reduced by PCA
- K=256
- Use mean and variance part
- 8 spatial regions

- Total dimensionality:

$$256 \times 64 \times 2 \times 8 = 262{,}144$$

VOC2007: accuracy

Table 1. Mean average precision (mAP) on VOC 2007. The loss of mAP to original dense feature (ratio 1) is also computed.

Method	Compression ratio	mAP (%)	Loss (%)
MI	1	58.57 ± 0.19	0
	32	60.09 ± 0.09	-1.52
	64	60.05 ± 0.16	-1.48
	128	58.97 ± 0.23	-0.40
	256	56.82 ± 0.49	1.75
	512	52.70 ± 0.44	5.87
	1024	46.52 ± 0.40	12.05
PQ [25]	1	58.8	0
	$32(d=6)$	58.2	0.6
	$64(d=8)$	56.6	2.2
	$128(d=8)$	54.0	4.8
	$256(d=8)$	50.3	8.5
PQ [21]	1	58.3	0
	$32(d=8)$	57.3	1.0
	$64(d=8)$	55.9	2.4
	$64(d=16)$	56.2	2.1

- #classes: 20
- #training: 5000
- #testing: 5000

ILSVRC2010: accuracy

Table 2. Top-5 accuracy on the ILSVRC 2010 dataset.

Method	Compression ratio	Accuracy (%)
MI	64	61.06
	128	56.64
	256	50.15
PQ [21]	$32(d = 8)$	56.2
	$64(d = 8)$	54.2
	$64(d = 16)$	54.9

- #classes: 1000
- #training: 1,200,000
- #testing: 150,000

SUN397: accuracy

Table 3. Top-1 accuracy on the SUN 397 dataset.

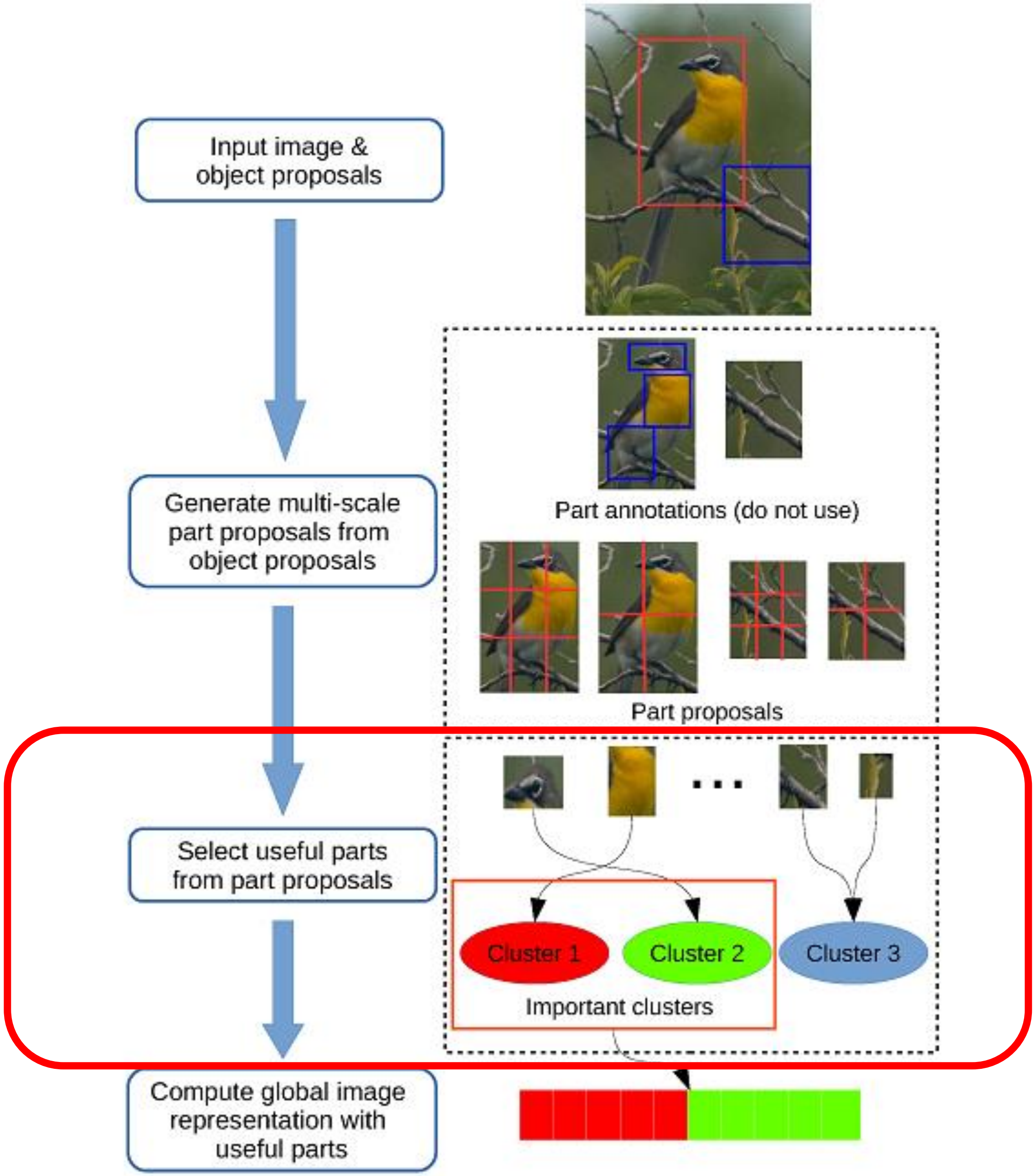
Method	Compression ratio	Accuracy (%)
dense FV [22]	1	43.3
multiple features [27]	1	38.0
spatial HOG [7]	1	26.8
MI	32	41.88±0.31
	64	42.05±0.36
	128	40.42±0.40
	256	37.36±0.34
PQ	32	42.72±0.45
	64	41.74±0.38
	128	40.13±0.33
	256	37.84±0.33

- #classes: 397
- #training: 19,850
- #testing: 19,850

Fine-Grained Categorization

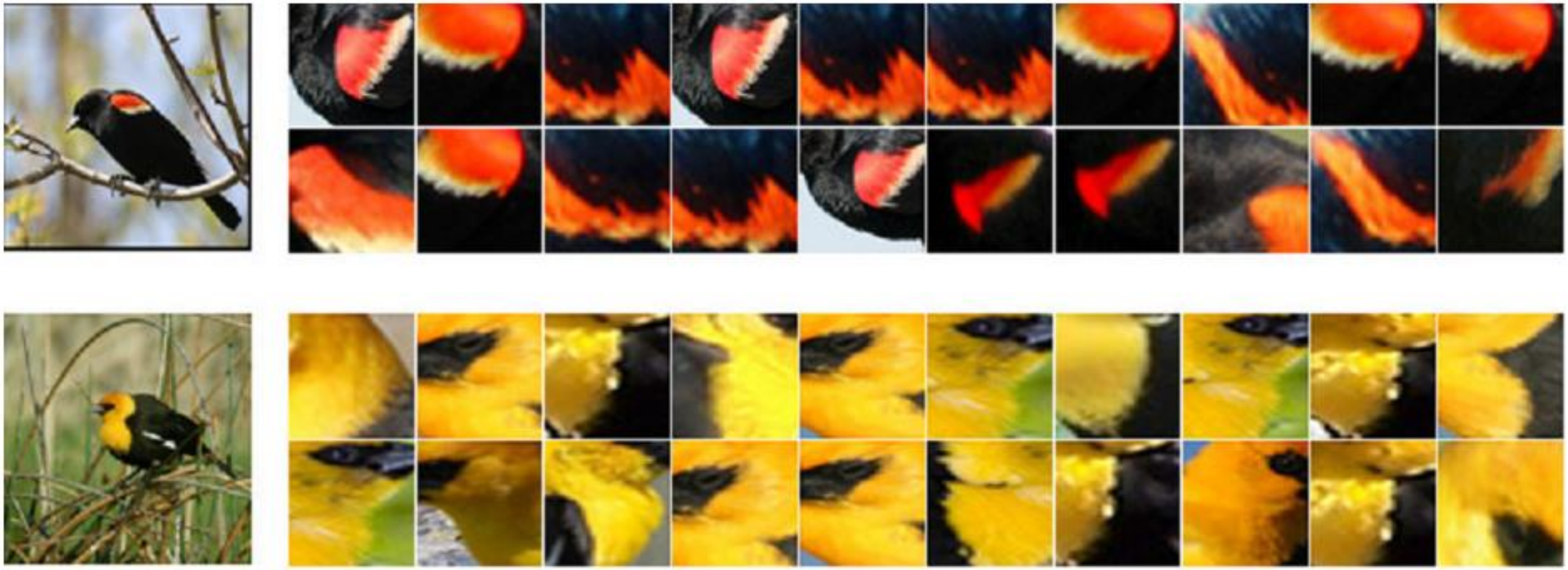
Selecting features is more important

Selection of subtle differences?



What features (parts) are chosen?

(a) Red-bellied Woodpecker vs. Red-headed Woodpecker



(b) Red-winged Blackbird vs. Yellow-headed Blackbird

(c) Blue Jay vs. Green Jay

How about accuracy?

Table 2: Classification accuracy on Caltech-UCSD Birds 200-2011.

Without annotations in both training and testing		
Methods	Selection fraction	Acc. (%)
Proposed	100% (All)	71.04
	75% (3/4)	71.67
	50 % (1/2)	73.34
	25% (1/4)	**75.02**
	12.5% (1/8)	73.82
Two-level attention [28]		69.70
Use annotations in training, not in testing		
DPD+DeCAF [6]		44.94
Part based R-CNN (without parts) [32]		52.38
Part based R-CNN-ft (without parts) [32]		62.75
Part based R-CNN-ft (with parts) [32]		73.89
Pose Normalized CNN [3]		75.70

Table 3: Classification accuracy on StanfordDogs.

Without annotations in both training and testing		
Methods	Selection fraction	Acc. (%)
Proposed	100% (All)	77.23
	75% (3/4)	78.28
	50% (1/2)	79.36
	25% (1/4)	**79.92**
	12.5% (1/8)	78.18
Two-level attention [28]		71.90
Use annotations in both training and testing		
Edge templates [29]		38.00
Unsupervised alignments [11]		50.10
MTL [21]		39.30

Published results

Compact Representation for Image Classification: To Choose or to Compress? Yu Zhang, Jianxin Wu, Jianfei Cai CVPR 2014

Towards Good Practices for Action Video Encoding
Jianxin Wu, Yu Zhang, Weiyao Lin CVPR 2014

New methods & results in arXiv

- VOC 2012: 90.7%, VOC 2007: 92.0%
  - http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=2
  - http://arxiv.org/abs/1504.05843
- SUN 397: 61.83%
  - http://arxiv.org/abs/1504.05277
  - http://arxiv.org/abs/1504.04792
- Details of fine-grained categorization
  - http://arxiv.org/abs/1504.04943

DSP

- An intuitive, principled, efficient, and effective image representation for image recognition
  - Using only the convolutional layers of CNN
    - Very efficient, but impressive representational power
    - No fine-tuning at all
  - Extremely small but effective FV / VLAD encoding (K=1, or 2)
    - Small memory footprint
  - New normalization strategy
    - Matrix norm to utilize global information
  - Spatial pyramid
    - Natural and principled way to integrate spatial information

D3

- Discriminative Distribution Distance
  - FV, VLAD and Super Vectors are *generative* representations
  - They ask "how one set is generated?"
  - But for image recognition, we care about "how two sets are *separated*?"
  - Proposed directional distribution distance to compare two sets
  - Proposed using a classifier MPM to *robustly* estimate the distance

  - D3 is very stable
  - D3 is very efficient

Multiview image representation

- Using DSP as the global view
- But context is also important: what are the neighborhood structure?
  - Solving distance metric learning as a DNN
  - Called the label view
- Integrated (global+label) views
  - 90.7% @VOC2012 recognition task
  - 92.0% @VOC2007 recognition task

Thanks!