Beyond the Bag of Word representation for image classification
DVMM lab seminar, Columbia University

Matthieu Cord

Computer Science dept. (LIP6), UPMC Sorbonne Univ., Paris, France

Dec 2012
Outline

1. Image classification framework
   - BoW extensions: parametrization
   - BossaNova

2. RBM dictionary learning in BoW framework
Image classification pipeline

**Training**
- Training Images
- Image Features
- Classifier Training
- Trained Classifier

**Testing**
- Test Image
- Image Features
- Trained Classifier
- Prediction: Monkey
Image classification pipeline

Bag-of-Visual-Words (BoVW) Model

- 1999 BoW on color [Ma Manjunath]
- 2001 BoW on Gabor [Fournier Cord]
- 2003-4 BoW on SIFT [Csurka]
- Spatial Information [Lazebnik06]
- Soft-assignment, sparse coding, max pooling [Wang10] [Boureau10]

Credit: Prof. Shih-Fu Chang

Image classification: BoW details

Coding/Pooling

- Feature extraction (e.g., SIFT)
- Coding
- Pooling
- Classification (monkey, dog, tree, ...)

monkeys?
**BoW Model**

\[ X = (x_1, \ldots, x_j, \ldots, x_N) \] the set of local descriptors (SIFT) for the image

\[ C = (c_1, \ldots, c_m, \ldots, c_M) \] the visual dictionary

\[
H = \begin{bmatrix}
\alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\
\vdots & & \vdots & & \vdots \\
\alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\
\vdots & & \vdots & & \vdots \\
\alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N}
\end{bmatrix}
\]

\[ g: \text{pooling} \]

\[ f: \text{coding} \]
BoW Model

\( \mathbf{X} = (x_1, \ldots, x_j, \ldots, x_N) \) the set of local descriptors (SIFT) for the image
\( \mathbf{C} = (c_1, \ldots, c_m, \ldots, c_M) \) the visual dictionary

Coding: \( \mathbf{x}_j \rightarrow f(\mathbf{x}_j) = \{\alpha_{m,j}\} \), \( \alpha_{m,j} = 1 \) iff \( m = \arg\min_{k \in \{1, \ldots, M\}} \|\mathbf{x}_j - \mathbf{c}_k\|^2_2 \)
BoW Model

\( \mathbf{X} = (x_1, \ldots, x_j, \ldots, x_N) \) the set of local descriptors (SIFT) for the image

\( \mathbf{C} = (c_1, \ldots, c_m, \ldots, c_M) \) the visual dictionary

**Coding:** \( x_j \rightarrow f(x_j) = \{\alpha_{m,j}\}, \quad \alpha_{m,j} = 1 \text{ iff } m = \arg \min_{k \in \{1, \ldots, M\}} \|x_j - c_k\|_2^2 \)

**Pooling:** \( g(\{\alpha_j\}) = z : \forall m, \quad z_m = \sum_{j=1}^{N} \alpha_{m,j} \)
BoW Model

\( \mathbf{X} = (x_1, \ldots, x_j, \ldots, x_N) \) the set of local descriptors (SIFT) for the image

\( \mathbf{C} = (c_1, \ldots, c_m, \ldots, c_M) \) the visual dictionary

\[
\mathbf{H} = \begin{bmatrix}
  \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\
  \vdots & \ddots & \vdots & \ddots & \vdots \\
  \alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N}
\end{bmatrix}
\]

\[ \Rightarrow g: \text{pooling} \]

\[ f: \text{coding} \]

Coding: \( x_j \to f(x_j) = \{ \alpha_{m,j} \}, \quad \alpha_{m,j} = 1 \text{ iff } m = \arg \min_{k \in \{1, \ldots, M\}} \| x_j - c_k \|_2^2 \)

Pooling: \( g(\{ \alpha_j \}) = \mathbf{z} : \forall m, \quad z_m = \sum_{j=1}^{N} \alpha_{m,j} \)

BoW representation: \( \mathbf{z} = [z_1, z_2, \cdots, z_M]^T \)
Results: Datasets

Many image datasets: Pascal VOC, Scene-15, MirFlickr, ...

Caltech101
- 9,144 images
- 102 categories
- 30 to 800 images per category

Standard evaluation protocol for baseline comparison:
- Train with 15-30 images / class
- Test on the remaining images
- Metric: Multi-class Accuracy
## Performance evaluation on Caltech101

### Average accuracy results

<table>
<thead>
<tr>
<th>Bow-like architectures</th>
<th>15 images</th>
<th>30 images</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Lazebnik &amp; al CVPR06]</td>
<td>56.4</td>
<td>64.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hierarchical and biologically inspired architectures</th>
<th>15 images</th>
<th>30 images</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Mutch &amp; al IJCV08]</td>
<td>51</td>
<td>56</td>
</tr>
<tr>
<td>[Ranzato &amp; al CVPR07]</td>
<td>-</td>
<td>54</td>
</tr>
<tr>
<td>[Jarrett09 &amp; al ICCV09]</td>
<td>-</td>
<td>65.6</td>
</tr>
<tr>
<td>[Zeiler &amp; al CVPR2010]</td>
<td>58.6</td>
<td>66.9</td>
</tr>
</tbody>
</table>
Outline

1. Image classification framework
   - BoW extensions: parametrization
   - BossaNova

2. RBM dictionary learning in BoW framework
Optimization of the BoW pipeline

Expe. from M. Law, N. Thome, M. Cord [ECCVw 2012]

- Parametrization: find the Winner Cocktail
  - SR: Sampling Rate = gap between centers of patches (pixels)
  - Mono/Multi scale SIFT detection
  - Dictionary size
  - Normalization

Figure: BoW pipeline for classification

- Extended coding
- Extended pooling
LSC principle

\[
\alpha_{m,j} = \frac{e^{-\beta \hat{d}(x_j, c_m)}}{\sum_{l=1}^{M} e^{-\beta \hat{d}(x_j, c_l)}}
\]

\[
\hat{d}(x_j, c_m) = \begin{cases} 
  d(x_j, c_m) & \text{if } c_m \in \mathcal{N}_k(x_j)^a \\
  \infty & \text{otherwise}
\end{cases}
\]

followed by max pooling, no normalization of the BoW, and Linear SVM

\[^a\mathcal{N}_k(x_i)\) the k-nearest neighbors
<table>
<thead>
<tr>
<th>SR</th>
<th>Scaling</th>
<th>Codebook Size</th>
<th>Accuracy (no norm)</th>
<th>Acc. ($\ell_2$-norm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>mono</td>
<td>800</td>
<td>70.07 ± 0.96</td>
<td>70.46 ± 1.04</td>
</tr>
<tr>
<td>6</td>
<td>mono</td>
<td>800</td>
<td>71.64 ± 0.99</td>
<td>72.01 ± 0.96</td>
</tr>
<tr>
<td>3</td>
<td>mono</td>
<td>800</td>
<td>72.45 ± 1.05</td>
<td>72.73 ± 0.99</td>
</tr>
<tr>
<td>8</td>
<td>mono</td>
<td>1700</td>
<td>71.67 ± 0.93</td>
<td>71.95 ± 0.90</td>
</tr>
<tr>
<td>8</td>
<td>mono</td>
<td>3300</td>
<td>72.13 ± 0.99</td>
<td>72.50 ± 0.97</td>
</tr>
<tr>
<td>8</td>
<td>multi</td>
<td>800</td>
<td>73.35 ± 0.89</td>
<td>73.83 ± 0.96</td>
</tr>
<tr>
<td>8</td>
<td>multi</td>
<td>1700</td>
<td>75.34 ± 0.92</td>
<td>75.97 ± 0.86</td>
</tr>
<tr>
<td>8</td>
<td>multi</td>
<td>3300</td>
<td>76.91 ± 0.98</td>
<td>77.02 ± 0.94</td>
</tr>
<tr>
<td>3</td>
<td>multi</td>
<td>800</td>
<td>73.81 ± 0.95</td>
<td>73.99 ± 0.86</td>
</tr>
<tr>
<td>3</td>
<td>multi</td>
<td>1700</td>
<td>75.72 ± 1.13</td>
<td>76.00 ± 0.94</td>
</tr>
<tr>
<td>3</td>
<td>multi</td>
<td>3300</td>
<td>77.23 ± 1.02</td>
<td>77.47 ± 0.99</td>
</tr>
<tr>
<td>3</td>
<td>multi</td>
<td>6500</td>
<td>78.00 ± 1.05</td>
<td>78.46 ± 0.95</td>
</tr>
</tbody>
</table>

**Table**: Classification results on Caltech-101 with 30 training images per class

SR: Sampling Rate = gap between centers of patches (pixels)
Conclusion

<table>
<thead>
<tr>
<th></th>
<th>[Law ECCVw 2012]</th>
<th>[Chatfield BMVC 2011]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cal-101</td>
<td>Sc-15</td>
</tr>
<tr>
<td>Sampling Rate</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>XXX</td>
<td>X</td>
</tr>
<tr>
<td>Scaling</td>
<td>XXX</td>
<td>XXX</td>
</tr>
<tr>
<td>Codebook Size</td>
<td>XXX</td>
<td>XXX</td>
</tr>
<tr>
<td>Normalization</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Table: Importance of parameters

- Huge performance difference according to the chain parameter tuning
- the devil is in the (parameter) details ... (Chatfield’s title)
- Fair comparisons: implementation details
- Sampling rate more important in mono-scale setup
- BoW has better results than Chatfield’s reimplementation of FK on Caltech101 and much more compact
Outline

1. Image classification framework
   - BoW extensions: parametrization
   - BossaNova

2. RBM dictionary learning in BoW framework
Poolng extension: BossaNova

- Novel mid-level image representation which offers a *more information*-preserving pooling operation based on a *distance-to-codeword distribution*.

- **BOSSA**: Bag Of Statistical Sampling Analysis
- BossaNova integrates several improvements over the original BOSSA
## BossaNova Model

### Pooling Formalism

$$g : \mathbb{R}^N \rightarrow \mathbb{R}^B$$

$$\alpha_m \rightarrow g(\alpha_m) = z_m$$

$$z_{m,b} = \text{card} \left( x_j : \alpha_{m,j} \in \left[ \frac{b}{B} ; \frac{b+1}{B} \right] \right)$$

$$\frac{b}{B} \geq \alpha_{m}^{\min} \quad \text{and} \quad \frac{b+1}{B} \leq \alpha_{m}^{\max}$$

**B** : number of bins of each histogram $z_m$, and $[\alpha_{m}^{\min} ; \alpha_{m}^{\max}]$ distance range
BossaNova Scheme

Image classification framework

Beyond the Bag of Word representation for image classification
BossaNova Scheme
BossaNova Representation

\[ x_j \rightarrow \alpha_j \]

\[ \sum_j \alpha_{m,j} \mid x_j \in b^{th} = z_{m,b} \]

\[ N_m = \sum_j \alpha_{m,j} \]
BossaNova Representation

\[ x_j \rightarrow \alpha_j \]

\[ \sum_j \alpha_{m,j} \mid x_j \in b^t = z_{m,b} \]

\[ N_m = \sum_j \alpha_{m,j} \]

Power normalization + \( \ell_2 \)-normalization
BossaNova Representation

\[ x_j \rightarrow \alpha_j \]

\[ \sum_j \alpha_{m,j} \mid x_j \in b^h = z_{m,b} \]

Power normalization + \( \ell_2 \)-normalization

\[ \tilde{z}_m \]

\[ N_m = \sum_j \alpha_{m,j} \]

\[ + \]

\[ sN_m \]
BossaNova Representation

\[
\sum_j \alpha_{m,j} \mid x_j \in b^{th} = z_{m,b}
\]

\[
N_m = \sum_j \alpha_{m,j}
\]

Power normalization + \(\ell_2\)-normalization

\[
\tilde{z}_m + sN_m
\]

\[
M \times (B + 1)
\]
BossaNova Representation

**BossaNova (BN) Parameters**

- **B** (number of bins): \{2, 4, 6, 8, 10\}
- **\(\alpha_{\text{min}}\)**: \{0, 0.6\}
- **\(\alpha_{\text{max}}\)**: \{1.5, 2.0\}
- **s** (cross weight): \{10^{-4};1\}
- **M** (codebook): \{128; 8192\}
Experimental Results

- Implemented methods: Bag-of-Words (BoW), Fisher Vector (FV), BOSSA, BossaNova (BN), BN + FV
- MIRFLICKR: 25000 images, manually annotated for 38 concepts.
- ImageCLEF 2011 Photo Annotation: 18000 images, 99 concepts
### Experimental Results – PASCAL VOC 2007

<table>
<thead>
<tr>
<th>Implemented methods</th>
<th>MAP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW [1]</td>
<td>53.2</td>
</tr>
<tr>
<td>BOSSA [2]</td>
<td>54.4</td>
</tr>
<tr>
<td>FV [3]</td>
<td>59.5</td>
</tr>
<tr>
<td>BN (ours)</td>
<td>58.5</td>
</tr>
<tr>
<td>BN + FV (ours)</td>
<td>61.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Published results</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Krapac et al. [4]</td>
<td>56.7</td>
</tr>
<tr>
<td>Wang et al. [5]</td>
<td>59.3</td>
</tr>
<tr>
<td>Chatfield et al. [6]</td>
<td><strong>61.7</strong></td>
</tr>
</tbody>
</table>
## Experimental Results – 15-Scenes

<table>
<thead>
<tr>
<th>Implemented methods</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW [1]</td>
<td>81.1 ± 0.6</td>
</tr>
<tr>
<td>BOSSA [2]</td>
<td>82.9 ± 0.5</td>
</tr>
<tr>
<td>FV [3]</td>
<td>88.1 ± 0.2</td>
</tr>
<tr>
<td>BN (ours)</td>
<td>85.3 ± 0.4</td>
</tr>
<tr>
<td>BN + FV (ours)</td>
<td><strong>88.9 ± 0.3</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Published results</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang et al. [7]</td>
<td>80.3 ± 0.9</td>
</tr>
<tr>
<td>Lazebnik et al. [8]</td>
<td>81.4 ± 0.5</td>
</tr>
<tr>
<td>Boureau et al. [9]</td>
<td>85.6 ± 0.2</td>
</tr>
<tr>
<td>Krapac et al. [4]</td>
<td>88.2 ± 0.6</td>
</tr>
</tbody>
</table>
## Experimental Results – MIRFLICKR

<table>
<thead>
<tr>
<th>Method</th>
<th>MAP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our methods</strong></td>
<td></td>
</tr>
<tr>
<td>BossaNova [Avila et al., 2012]</td>
<td>54.4</td>
</tr>
<tr>
<td>BossaNova + FV [Avila et al., 2012]</td>
<td>56.0</td>
</tr>
<tr>
<td><strong>Implemented methods</strong></td>
<td></td>
</tr>
<tr>
<td>BoW [Sivic and Zisserman, 2003]</td>
<td>51.5</td>
</tr>
<tr>
<td>FV [Perronnin et al., 2010]</td>
<td>54.3</td>
</tr>
<tr>
<td><strong>Published results</strong></td>
<td></td>
</tr>
<tr>
<td>[Huiskes et al., 2010]</td>
<td>37.5</td>
</tr>
<tr>
<td>[Guillaumin et al., 2010]</td>
<td>53.0</td>
</tr>
</tbody>
</table>
## Experimental Results – MIRFLICKR

<table>
<thead>
<tr>
<th>Method</th>
<th>MAP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our methods</strong></td>
<td></td>
</tr>
<tr>
<td>BossaNova [Avila et al., 2012]</td>
<td>54.4</td>
</tr>
<tr>
<td>BossaNova + FV [Avila et al., 2012]</td>
<td>56.0</td>
</tr>
<tr>
<td><strong>Implemented methods</strong></td>
<td></td>
</tr>
<tr>
<td>BoW [Sivic and Zisserman, 2003]</td>
<td>51.5</td>
</tr>
<tr>
<td>FV [Perronnin et al., 2010]</td>
<td>54.3</td>
</tr>
<tr>
<td><strong>Published results</strong></td>
<td></td>
</tr>
<tr>
<td>[Huiskes et al., 2010]</td>
<td>37.5</td>
</tr>
<tr>
<td>[Guillaumin et al., 2010]</td>
<td>53.0</td>
</tr>
</tbody>
</table>
### Experimental Results – ImageCLEF 2011

<table>
<thead>
<tr>
<th>Our methods</th>
<th>MAP (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOSSA [Avila et al., 2011]</td>
<td>32.9</td>
</tr>
<tr>
<td>BN [Avila et al., 2012]</td>
<td>35.3</td>
</tr>
<tr>
<td>BN + FV [Avila et al., 2012]</td>
<td>38.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Implemented methods</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW [Sivic and Zisserman, 2003]</td>
<td>31.2</td>
</tr>
<tr>
<td>FV [Perronnin et al., 2010]</td>
<td>36.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Best results ImageCLEF 2011</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[Binder et al., 2011]</td>
<td>38.8</td>
</tr>
</tbody>
</table>

- ImageCLEF’12, rank second for Fully Visual track
- Project Web page with codes available
  [https://sites.google.com/site/bossanovasite/](https://sites.google.com/site/bossanovasite/)
Outline

1. Image classification framework
2. RBM dictionary learning in BoW framework
Extended Coding: dictionary learning

- **Unsupervised Dictionary Learning**
  - Non-Learned assignment coding
  - Sparse coding
  - Restricted Boltzmann machines

- **Supervised Dictionary Learning**
  - Local optimization
  - Global optimization

- **Strategy based on RBM with supervised fine tuning:**
  - [ECCV 2012] with Hanlin Goh, Lim Joo Hwee, Nicolas Thome
  - Accurate image categorization
  - Small & concise visual dictionary
  - Fast inference
Image Categorization Framework

- **Bag of Words (BOW) Model**

![Diagram of Image Categorization Framework]

(Deep) Visual Dictionary
Restricted Boltzmann Machine

\[ P(z_j \mid x) = \text{sigm} \left( b_j + \sum_{i=1}^{I} w_{ij} x_i \right) \]

\[ P(x_i \mid z) = \text{sigm} \left( c_i + \sum_{j=1}^{J} w_{ij} z_j \right) \]

\[ E(x, z) = - \log P(x, z) = - \sum_{i=1}^{I} \sum_{j=1}^{J} x_i w_{ij} z_j - \sum_{i=1}^{I} c_i x_i - \sum_{j=1}^{J} b_j z_j. \]

**Optimization**
- Maximum likelihood approximation
- Contrastive divergence learning algorithm
Contrastive Divergence Learning

(1) Alternating Gibbs Sampling
- Fix activations of one layer
- Stochastically sample opposite layer

(2) Parameter Updates
\[ \Delta w_{ij} = \varepsilon (\langle x_i z_j \rangle_{data} - \langle x_i z_j \rangle_{recon}) \]
\[ \Delta b_j = \varepsilon (\langle z_j \rangle_{data} - \langle z_j \rangle_{recon}) \]
\[ \Delta c_i = \varepsilon (\langle x_i \rangle_{data} - \langle x_i \rangle_{recon}) \]
Selective & Sparse Feature Coding

- **Selectivity**
  - Each codeword should respond only to a small subset of input descriptors

- **Sparsity**
  - Each input descriptor should only have a small subset of codewords responding to it
Sparse & Selective Coding Schemes

Input Feature Space

Feature Coding

Descriptors

Distributed

Sparse But Not Selective

Selective But Not Sparse

Sparse and Selective

Visual Codewords
Two-Step Joint RBM Regularization

**STEP 1: Compute Target Codeword Responses**

**STEP 1A:** Map every column  
**STEP 1B:** Remap every row

**STEP 2: Regularize RBM Learning**

\[
\arg \min_w \left( - \sum_{k=1}^{K} \log \sum_z \Pr(x_k, z_k) - \lambda \sum_{j=1}^{J} \sum_{k=1}^{K} p_{jk} \log z_{jk} + (1 - p_{jk}) \log (1 - z_{jk}) \right)
\]

RBM  
(maximum likelihood approximation)  

Cross-Entropy Penalty  
(per descriptor & codeword)
SIFT Visual Codewords
Local Supervised Fine-Tuning

- Supervised learning is performed on the codebook initialized by the unsupervised regularized RBM.
- The error backpropagation algorithm is used to fine-tune the local descriptor codebook using image labels.
Shallow & Deep Visual Dictionaries

Dense Sampling:

Category Labels

Coded MF

Spatial Aggregation

Macro Feature

Image

Macro Code

Spatial Aggregation

Coded MF

Shared Spatial Pooling

Image

Category Labels

Coded MF

Shared Spatial Pooling

Macro Feature
Image Categorization Results

- Achieved high accuracy
- Visual dictionaries are small and concise
- Inference is fast

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Caltech-101 (30 tr.)</th>
<th>Caltech-256 (60 tr.)</th>
<th>15-Scenes (100 tr.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unsupervised Shallow</td>
<td>78.0%</td>
<td>46.1%</td>
<td>85.7%</td>
</tr>
<tr>
<td>Supervised Shallow</td>
<td>78.9%</td>
<td>46.0%</td>
<td>86.0%</td>
</tr>
<tr>
<td>Unsupervised Deep</td>
<td>72.8%</td>
<td>44.7%</td>
<td>82.5%</td>
</tr>
<tr>
<td>Supervised Deep</td>
<td>79.7%</td>
<td>47.2%</td>
<td>86.4%</td>
</tr>
<tr>
<td>Bach’s Team</td>
<td>75.7%</td>
<td>-</td>
<td>84.3%</td>
</tr>
</tbody>
</table>

- Deep unsupervised does not do as well as shallow unsupervised.
- Supervision is crucial for deep architectures; less important for shallow architectures.
Deep Transfer Learning

- Learn dictionary from Caltech-101 and evaluate on Caltech-256

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Layer 1 Training Set</th>
<th>Layer 2 Training Set</th>
<th>Unsupervised Results</th>
<th>Supervised Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shallow</td>
<td>Caltech-256</td>
<td>-</td>
<td>46.1%</td>
<td>46.0%</td>
</tr>
<tr>
<td>Deep</td>
<td>Caltech-256</td>
<td>Caltech-256</td>
<td>44.7%</td>
<td>47.2%</td>
</tr>
<tr>
<td>Shallow</td>
<td>Caltech-101</td>
<td>-</td>
<td>45.8%</td>
<td>45.9%</td>
</tr>
<tr>
<td>Deep</td>
<td>Caltech-101</td>
<td>Caltech-101</td>
<td>41.4%</td>
<td>44.2%</td>
</tr>
<tr>
<td>Deep</td>
<td>Caltech-101</td>
<td>Caltech-256</td>
<td>44.0%</td>
<td>47.0%</td>
</tr>
</tbody>
</table>

- Transfer learning works well on the lower layer
- Performance drops when transferring the higher layer
What’s next?

Comeback of Deep Networks

- Convolutional networks: [LeCun98], improvements [Jarrett09, Lee09]


RBM dictionary learning in BoW framework

People LIP6, Univ. UPMC-PARIS VI
Matthieu Cord, Nicolas Thome, matthieu.cord@lip6.fr

- PhD students: Sandra Avila, Hanlin Goh, Mar Law, Denis Pitzalis
- Post-Docs: Christian Theriault
- Research Inge. J. Guyomard

BossaNova Project Web page with codes available:
https://sites.google.com/site/bossanovasite/
JKernelMachines (Java) with D. Picard:
https://mloss.org/software/view/409/

http://webia.lip6.fr/~cord/
References I


