Image Classification Using Gaussian Mixture and Local Coordinate Coding

Kai Yu
NEC Laboratories America, Cupertino, California, USA

Joint work with
Jinjun Wang, Fengjun Lv, Wei Xu, Yihong Gong
Xi Zhou, Jianchao Yang, Thomas Huang,
Tong Zhang
Chen Wu

NEC Laboratories America
Univ. of Illinois at Urbana-Champaign
Rutgers University
Stanford University

PASCAL VOC Challenge, ICCV, at Kyoto, Japan, October 3rd, 2009
## Where We Are in This Competition

<table>
<thead>
<tr>
<th>Item</th>
<th>Our 4 submissions</th>
<th>Our Best</th>
<th>Other's Best</th>
<th>Our Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aeroplane</td>
<td>88.1 88.0 87.1 87.7</td>
<td>88.1</td>
<td>86.6</td>
<td>1.5</td>
</tr>
<tr>
<td>Bicycle</td>
<td>68.0 68.6 67.4 67.8</td>
<td>68.6</td>
<td>63.9</td>
<td>4.7</td>
</tr>
<tr>
<td>Bird</td>
<td>68.0 67.9 65.8 68.1</td>
<td>68.1</td>
<td>66.7</td>
<td>1.4</td>
</tr>
<tr>
<td>Boat</td>
<td>72.5 72.9 72.3 71.1</td>
<td>72.9</td>
<td>67.3</td>
<td>5.6</td>
</tr>
<tr>
<td>Bottle</td>
<td>41.0 44.2 40.9 39.1</td>
<td>44.2</td>
<td>43.7</td>
<td>0.5</td>
</tr>
<tr>
<td>Bus</td>
<td>78.9 79.5 78.3 78.5</td>
<td>79.5</td>
<td>74.1</td>
<td>5.4</td>
</tr>
<tr>
<td>Car</td>
<td>70.4 72.5 69.7 70.6</td>
<td>72.5</td>
<td>64.7</td>
<td>7.8</td>
</tr>
<tr>
<td>Cat</td>
<td>70.4 70.8 69.7 70.7</td>
<td>70.8</td>
<td>64.2</td>
<td>6.6</td>
</tr>
<tr>
<td>Chair</td>
<td>58.1 59.5 58.5 57.4</td>
<td>59.5</td>
<td>57.4</td>
<td>2.1</td>
</tr>
<tr>
<td>Cow</td>
<td>53.4 53.6 50.1 51.7</td>
<td>53.6</td>
<td>46.2</td>
<td>7.4</td>
</tr>
<tr>
<td>Diningtable</td>
<td>55.7 57.5 55.1 53.3</td>
<td>57.5</td>
<td>54.7</td>
<td>2.8</td>
</tr>
<tr>
<td>Dog</td>
<td>59.3 59.0 56.3 59.2</td>
<td>59.3</td>
<td>53.5</td>
<td>5.8</td>
</tr>
<tr>
<td>Horse</td>
<td>73.1 72.6 71.8 71.6</td>
<td>73.1</td>
<td>68.1</td>
<td>5.0</td>
</tr>
<tr>
<td>Motorbike</td>
<td>71.3 72.3 70.8 70.6</td>
<td>72.3</td>
<td>70.6</td>
<td>1.7</td>
</tr>
<tr>
<td>Person</td>
<td>84.5 85.3 84.1 84.0</td>
<td>85.3</td>
<td>85.2</td>
<td>0.1</td>
</tr>
<tr>
<td>Pottedplant</td>
<td>32.3 36.6 31.4 30.9</td>
<td>36.6</td>
<td>39.1</td>
<td>-2.5</td>
</tr>
<tr>
<td>Sheep</td>
<td>53.3 56.9 51.5 51.7</td>
<td>56.9</td>
<td>48.2</td>
<td>8.7</td>
</tr>
<tr>
<td>Sofa</td>
<td>56.7 57.9 55.1 55.9</td>
<td>57.9</td>
<td>50.0</td>
<td>7.9</td>
</tr>
<tr>
<td>Train</td>
<td>86.0 85.9 84.7 85.9</td>
<td>86.0</td>
<td>83.4</td>
<td>2.6</td>
</tr>
<tr>
<td>Tvmonitor</td>
<td>66.8 68.0 65.2 66.7</td>
<td>68.0</td>
<td>68.6</td>
<td>-0.6</td>
</tr>
<tr>
<td>Average</td>
<td>65.4 66.5 64.3 64.6</td>
<td></td>
<td>66.5</td>
<td>62.8</td>
</tr>
</tbody>
</table>
## Comparative Overview

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>State of the Art</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Detection</td>
<td>multiple detectors</td>
<td>dense sampling</td>
</tr>
<tr>
<td>Feature Extraction</td>
<td>multiple descriptors</td>
<td>SIFT (gray)</td>
</tr>
<tr>
<td>Coding Scheme</td>
<td>VQ</td>
<td>GMM, LCC</td>
</tr>
<tr>
<td>Spatial Pooling</td>
<td>SPM</td>
<td>SPM</td>
</tr>
<tr>
<td>Classifier</td>
<td>nonlinear classifiers</td>
<td>linear classifiers</td>
</tr>
</tbody>
</table>
Our Strategy

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>State of the Art</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Detection</td>
<td>multiple detectors</td>
<td>dense sampling</td>
</tr>
<tr>
<td>Feature Extraction</td>
<td>multiple descriptors</td>
<td>SIFT (gray)</td>
</tr>
<tr>
<td>Coding Scheme</td>
<td>VQ</td>
<td>GMM, LCC</td>
</tr>
<tr>
<td>Spatial Pooling</td>
<td>SPM</td>
<td>SPM</td>
</tr>
<tr>
<td>Classifier</td>
<td>nonlinear classifiers</td>
<td>linear classifiers</td>
</tr>
</tbody>
</table>

We bet on machine learning techniques.
Pipeline Overview - I

Input gray image

- extract SIFT on a grid of locations
  - Grid Step Size: every 4 pixels
  - Patch Size: 16x16, 24x24, 32x32
  - PCA on SIFT: 128 dim -> 80 dim

• GMM coding & SPM
• LCC coding & SPM

Unsupervised codebook learning

• linear classifiers
  • WCCN
  • Gaussian process
  • SVM Universum

Submission Entry: NECUIUC_LL-CDCV
Overall AP=64.29%

Submission Entry: NECUIUC_LN-CDCV
Overall AP=64.63%
Pipeline Overview - II

Input gray image

- extract SIFT on a grid of locations

1. GMM coding & SPM
2. LCC coding & SPM
3. Fast LCC coding & SPM
4. Sliding window by object det.

Linear classifiers

1. Max pooling

Submission Entry: NECUIUC_CLS-DTCT
Overall AP=66.48%

Note: 1. Overall AP is around 58.0%; 2. Overall AP is around 46% (estimation based on 5-fold cross validation)
Prior Publications

- **Local Coordinate Coding**
  - Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification
    Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang, *CVPR 2009*
  
  - Nonlinear Learning using Local Coordinate Coding
    Kai Yu, Tong Zhang, and Yihong Gong, *NIPS 2009*, to appear

- **GMM**
  - Hierarchical Gaussianization for Image Classification
    Xi Zhou, Na Cui, Zhen Li, Feng Liang, and Thomas S. Huang, *ICCV 2009*
  
  - SIFT-Bag Kernel for Video Event Analysis
    Xi Zhou, Xiaodan Zhuang, Shuicheng Yan, Shih-Fu Chang, Mark Hasegawa-Johnson, Thomas S. Huang, *ACM Multimedia 2008*

In our work on PASCAL challenge, we made further extensions of the above work in both engineering and theory.
A Unified Framework

- What matters is to learn nonlinear function on SIFT vectors.
- This boils down to learning a good coding scheme of SIFT.
Coding of SIFT

Dense SIFT

Nonlinear Coding on SIFT

Linear Pooling

Lin. Classifier

cat
Some Notation

\[ X \in \mathbb{R}^D \]  
\( \Phi(X) : \mathbb{R}^D \rightarrow \mathbb{R}^L \)  
\( f(X) : \mathbb{R}^D \rightarrow \mathbb{R} \)  
\( \hat{f}(X) = W^\top \Phi(X) \)

- a SIFT feature vector
- encoding function
- unknown function on local features
- approximating function

Supervised Learning  Unsupervised Learning
Example 1: Vector Quantization Coding (VQ)

The approximating function is

\[ \hat{f}(X) = W^T \Phi(X), \]

where \( W = [W_1, W_2, \ldots, W_K]^T \), \( \Phi(X) \) is the code of \( X \).

If \( X \) belongs to class 2, \( \Phi(X) = [0, 1, 0, \ldots, 0]^T \), then \( \hat{f}(X) = W^T \Phi(X) = W_2 \).
Example 2: “Supervector” Coding

- Given $K$ clusters in $X$ space, let $W = [W_1^T, W_2^T, \ldots, W_K^T]^T$, where $W_k \in \mathbb{R}^D$, and

$$
\Phi(X) = [C_1(X) * X^T, C_2(X) * X^T, \ldots, C_K(X) * X^T]^T,
$$

with $C_k(X) = 1$ if $X$ belongs to cluster $k$, otherwise $C_k(X) = 0$.

- Then $\hat{f}(X) = W^T \Phi(X) = \sum_k C_k(X) * W_k^T X$. — a locally piecewise linear function

- $C_k(X)$ can be soft probability given by GMM, then $\Phi(X)$ is **GMM supervector**.
Example 3: Local Coordinate Coding

Given anchor points \([B_1, \ldots, B_K]\), if the coding scheme \(\Phi(X) = [\phi_1, \ldots, \phi_K]\) satisfies

1. low reconstruction error: \(X \approx \sum_{k=1}^{K} \phi_k B_k\);
2. good locality: \(\phi_k\) tends to be nonzero if \(B_k\) is in \(X\)'s neighborhood, otherwise 0.

Then \(\hat{f}(X) = W^\top \Phi(X)\) provides a close approximation to \(f(X)\).
LCC: How It Works

\[ \hat{f}(X) = \sum_{k=1}^{K} \Phi_k W_k = \sum_{k=1}^{K} \Phi_k \hat{f}(B_k) \] forms a **local interpolation**

\[
\Phi(X) = \arg \max_{\Phi} \left\| X - \sum_{k=1}^{K} \Phi_k B_k \right\|^2 + \lambda \sum_{k} \alpha_k(X) |\Phi_k|
\]

where \( \alpha_k(X) \) is a distance from \( X \) to \( B_k \)
Comparison of Coding Methods

<table>
<thead>
<tr>
<th>Function Approximation</th>
<th>Poor</th>
<th>Good</th>
<th>Excellent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Locality</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Caltech-101</td>
<td>~65% ¹</td>
<td>~73% ²</td>
<td>~73% ³</td>
</tr>
</tbody>
</table>

1. Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, CVPR, 2006
2. Xi Zhou, Na Cui, Zhen Li, Feng Liang, and Thomas S. Huang, ICCV, 2009
Improve GMM Supervector Coding

\[ f(X) \]

- "local linear" → "local nonlinear"
- the code of \( X \) is

\[
\Phi(X) = \left[ C_1(X) \ast (X, X^2)^\top, \ldots, C_K(X) \ast (X, X^2)^\top \right]
\]
Improve LCC’s Efficiency

- Pre-computation: partition data and anchor points
- Eliminate those anchor points in different partitions
Equivalent to “Mixture of Coding Experts”

- Use a **soft-max gating function** $G_k(X)$ indicating if $X$ is in local partition $k$.
- Optimize the following cost

$$\Phi(X) = \arg \min_{\Phi} \sum_{k=1}^{K} G_k(X) \left( \left\| X - \sum_{m=1}^{M} \Phi_m^{(k)} B_m^{(k)} \right\|^2 + \lambda \sum_{m} \left| \Phi_m^{(k)} \right| \right)$$

- This is equivalent to

$$\Phi(X) = \arg \max_{\Phi} \left\| X - \sum_{k=1}^{M*K} \Phi_k B_k \right\|^2 + \lambda \sum_{k=1}^{M*K} \alpha_k(X) \left| \Phi_k \right|$$

where $\alpha_k(X)$ is 1 if $X$ and $B_k$ belong to the same partition, otherwise $+\infty$. 
Linear Pooling

- Dense SIFT
- Nonlinear Coding on SIFT
- Linear Pooling
- Lin. Classifier
- cat
(Local) Linear Pooling

data in an image codes image representation

$$Z_I = \sum_{k=1}^{K} \frac{\sum_{i \in I} G_k(X_i)\Phi_{nmlz}(X_i)}{\sqrt{\sum_{j \in I} G_k(X_j)}}$$

where $\Phi_{nmlz}(X)$ is the normalized version of $\Phi(X)$, obtained by subtracting mean and then dividing by variance.

The classification function on image $I$ is

$$c(I) = W^T Z_I = \sum_{k=1}^{K} \frac{\sum_{i \in I} G_k(X_i)W^T \Phi_{nmlz}(X_i)}{\sqrt{\sum_{j \in I} G_k(X_j)}} = \sum_{k=1}^{K} \frac{\sum_{i \in I} G_k(X_i)f(X_i)}{\sqrt{\sum_{j \in I} G_k(X_j)}}$$

Nonlinear function on local features
SPM representation

See also in “SurreyUVA_SRKDA method”, presentation at PASCAL VOC workshop 08.
Linear Classifier

- Dense SIFT
- Nonlinear Coding on SIFT
- Linear Pooling
- Lin. Classifier
- Cat
Support Vector Machines

- Use our own implementation, training using gradient based method LBFGS.

\[
\min_W \left\{ J(W) = \|W\|^2 + C \sum_{i=1}^{n} \ell(W; Y_i, Z_i) \right\}
\]

- Use a differentiable hinge loss

\[
\ell(W; Y_i, Z_i) = \left[ \max \left( 0, W^\top Z_i \cdot Y_i - 1 \right) \right]^2
\]
Use the **Universum approach**: if image $i$ is a difficult case, let the loss be

$$\ell (W; Y_i, Z_i) = (W^\top Z_i)^2$$
Within-class Covariance Normalization

- Within-class normalization

\[ K_{i,j} = Z_i^T (\gamma S + (1 - \gamma)I)^{-1} Z_j \]

where \( S \) is the average within-class covariance matrix.
Improve SPM using Gaussian Process

- The SPM approach uses 8 linear kernels.
- We can learn the kernel weights.

\[
\min_{\{\alpha_s \geq 0\}} - \log P \left( Y \bigg| \sum_{s=1}^{8} \alpha_s K_s \right) + \lambda \sum_{s=1}^{8} (\alpha_s - \alpha_0)^2
\]

- We learn a set of global weights for all classes.
Some Details

- Number of partitions or components
  - GMM: 1024 and 2048
  - LCC: 1024 and 2048

- Dimensionality of feature vector for each image (e.g. in case of 1024 partitions)
  - GMM: 1024x80x8 (1024 components, 80 PCA-SIFT, 8 SPM sub kernels)
  - LCC: 1024x256x8 (1024 partitions, 256 codebook size, 8 SPM sub kernels)
Conclusion Remarks

• Highly nonlinear, highly local encoding of image local features make difference!

• Still a long way to go
  – No high-level (semantic) features used so far
  – how to get compact image representations?
  – Supervised training of coding schemes
  – Better methods to use the bounding box information

• More details will be provided in an upcoming TR.