OBJECT CATEGORIZATION BY LOCAL FEATURE MATCHING WITH A LARGE NUMBER OF WEB IMAGES

Mizuki Akiyama, Yoshiyuki Kawano, Keiji Yanai
Department of Informatics
The University of Electro-Communication,
Tokyo, JAPAN
Background

- **Object Recognition**
  - Generic object recognition - object, scene, face, ...
  - Specific object recognition - DB search, name search

- TIGER

- [Image of a tiger]
- [Image of a clock]
Object Recognition

- Generic object recognition - object, scene, face, ...
- Specific object recognition - DB search, name search
Background

- **Specific Object Recognition**
  - Large-scale image database + local feature matching
  - High precision for recognition of specific-shaped objects

- **Generic Object Recognition**
  - Small-scale images database + machine learning
  - Low precision due to inter-class ambiguity

Specific object recognition applies to generic object recognition.
Goal

- Generic object recognition
  - Local feature matching based specific object recognition
  - Large-scale training and categorization
    - 150,000 training images

The chair

A chair
Related Work

- Object recognition using a large amount of images
  - Specific object recognition method to landmark database creation [Zheng et al, ICCV2009].
  - 20 million geo-tagged images on the Web
    - Search a very similar image

- SIFT-matching based image search
Related Work

- Generic object recognition using a large amount of image data
  - 80 million images [A. Torralba et al., PAMI (2008)]
  - Image categorization by k-NN with 32x32 tiny images
  - Comparable performance to the state-of-the-art method
  - K-NN method with a very large amount of image data was one of promising approaches for generic object recognition.
Our work is inspired by the “80 million images”

- 80 million images
  - Use the sum of squared differences (SAD) between 32x32 tiny images

Our work:
- Apply SIFT-based local feature matching and voting
- Collect a large number of images from the Web
- Use them for experiments on image categorization without excluding noise images
Overview

**Image collection + Feature extraction**
- Collect 150,000 from the Web
- Extract local features

create a database
- Local features of training images stored in the database

feature matching
- Feature matching for test images
- Vote the k most similar samples stored in the database

recognition
- Majority based categorization
Detail

- Image collection
  - 25 categories, total 150,000
  - Source: Google, Yahoo!, Flickr
  - noise images included
Feature extraction

SIFT
- 128 dimension, invariant to rotation, scale-change and illumination change

PCA-SIFT
- 36 dimension, extension of SIFT

Bag-of-Features (BoF)
- Many local patches, and vector-quantizing
- Explore the best setting of the codebook size k
Detail

- Database
  - kd-tree
    - SIFT, PCA-SIFT
  - Inverted index
    - BoF
Feature matching

- Simple NN search is very costly

Approximate Nearest Neighbor (ANN)

- Kd-tree based approximate nearest neighbor search method
- Search the top n nearest points for each query local feature point
- Vote on the image from which the nearest local features are extracted
 Recognition

- Sort the image having votes in the descending order of the number of votes

- Decide one of the given categories by applying k-Nearest Neighbor classification

- k-Nearest Neighbor
  - Majority of the categories of the top k samples
Experiments

- 5 and 25 class categorization

- Feature representation
  - SIFT, PCA-SIFT, BoF

- Parameters
  - ANN top n and k-NN top k
  - Codebook size
Dataset

- 5 and 25 categories Dataset
  - Examples of images

<table>
<thead>
<tr>
<th>Animal</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>Dog</td>
<td>Elephant</td>
<td>Lion</td>
<td>Tiger</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Car</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Impreza</td>
<td>Lexus</td>
<td>Odyssey</td>
<td>Pajero</td>
<td>Prius</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flower</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosmos</td>
<td>Dandelion</td>
<td>Lavender</td>
<td>Lily</td>
<td>Rose</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Food</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cake</td>
<td>Hamburger</td>
<td>Pizza</td>
<td>Ramen</td>
<td>Sushi</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Musical instrument</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dram</td>
<td>Flute</td>
<td>Guitar</td>
<td>Piano</td>
<td>Violin</td>
<td></td>
</tr>
</tbody>
</table>
Experiments

- The number of training images
  - 32GB linux machine

<table>
<thead>
<tr>
<th>Method</th>
<th># of training images</th>
<th># of images per class</th>
<th># of local features</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>26,250</td>
<td>1,050</td>
<td>15million</td>
<td>20GB</td>
</tr>
<tr>
<td>PCA-SIFT</td>
<td>73,500</td>
<td>2,940</td>
<td>53.50million</td>
<td>25GB</td>
</tr>
<tr>
<td>BoF</td>
<td>145,000</td>
<td>5,800</td>
<td>-</td>
<td>5GB</td>
</tr>
</tbody>
</table>

Evaluation

- Recall, precision, classification rate
- Baseline: Bag-of-Features + SVM
# Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>5 class Classification rate(%)</th>
<th>25 class Classification rate(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT(n=5,k=7,000)</td>
<td>60.1</td>
<td>32.5</td>
</tr>
<tr>
<td>Proposed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCA(n=5,k=7,000)</td>
<td>57.2</td>
<td>29.8</td>
</tr>
<tr>
<td>Proposed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BoF(size=200,000,k=20,000)</td>
<td>54.9</td>
<td>30.7</td>
</tr>
<tr>
<td>Proposed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BoF+SVM(linear kernel)</td>
<td>51.7</td>
<td>17.1</td>
</tr>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BoF+SVM($\chi^2$ kernel)</td>
<td>66.9</td>
<td>36.2</td>
</tr>
<tr>
<td>Baseline</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experimental Results (SIFT, PCA, 5class)
Experimental Results (SIFT, PCA, 25 class)

25 class classification (SIFT, PCA-SIFT)

Classification rate vs. \( k \) (top-k) for different methods and values of \( n \):
- **ANN: \( n \)**
- **SIFT, \( n=1 \)**
- **SIFT, \( n=5 \)**
- **SIFT, \( n=10 \)**
- **SIFT, \( n=25 \)**
- **PCA, \( n=1 \)**
- **PCA, \( n=5 \)**
- **PCA, \( n=10 \)**
- **PCA, \( n=25 \)**
Experimental Results (BoF, 5class)

Classification rate

5 class classification (BoF)

Classification rate

kNN (top-k)

code book size: s

[thousand]

S : 100
S : 50
S : 20
S : 5
Experimental Results (BoF, 25 class)

Classification rate: 100 [thousand]: 50

25 class classification (BoF)

Classification rate

kNN (top-k)

code book size: s

[thousand]

S = 100
S = 50
S = 20
S = 5
## Experimental Result (SIFT, 5class)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Recall(%)</th>
<th>Precision(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1: Animal</strong></td>
<td>155</td>
<td>14</td>
<td>37</td>
<td>37</td>
<td>7</td>
<td><strong>62</strong></td>
<td><strong>61</strong></td>
</tr>
<tr>
<td><strong>2: Car</strong></td>
<td>10</td>
<td>228</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td><strong>91</strong></td>
<td><strong>56</strong></td>
</tr>
<tr>
<td><strong>3: Flower</strong></td>
<td>35</td>
<td>15</td>
<td>150</td>
<td>41</td>
<td>9</td>
<td><strong>60</strong></td>
<td><strong>63</strong></td>
</tr>
<tr>
<td><strong>4: Food</strong></td>
<td>43</td>
<td>24</td>
<td>40</td>
<td>135</td>
<td>8</td>
<td><strong>54</strong></td>
<td><strong>58</strong></td>
</tr>
<tr>
<td><strong>5: Musical instrument</strong></td>
<td>12</td>
<td>128</td>
<td>10</td>
<td>14</td>
<td>86</td>
<td><strong>34</strong></td>
<td><strong>75</strong></td>
</tr>
</tbody>
</table>

**Precision(%)**

| 61 | 56 | 63 | 58 | 75 | **60.3** |
**Experimental Results (SIFT, 25class)**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Car</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>8</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>7</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>22</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>15</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Flower</td>
<td>10: Prius</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Flower</td>
<td>10: Prius</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Flower</td>
<td>10: Prius</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Flower</td>
<td>10: Prius</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Flower</td>
<td>10: Prius</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Flower</td>
<td>10: Prius</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Precision (%)**

18 | 7 | 32 | 30 | 63 | 25 | 17 | 11 | 27 | 19 | 35 | 59 | 32 | 33 | 40 | 10 | 11 | 33 | 33 | 32 | 0 | 38 | 76 | 60 | 55 | 32.5
Experimental Results (SIFT, 25 class)
Conclusions

- Generic object recognition by feature matching based specific object recognition

- It performed well with a large number of sample images
  - Almost equivalent to the result by generic recognition
    - 5class - 60.3%, 25class - 32.5%
Future works

- More large-scale experiments
  - Using parallel computing

- Informative feature
  - Use only discriminative features for image categorization.
Experimental Results (SIFT, 25 class)

25 class classification (SIFT, ANN n=5)

Classification rate vs. kNN (topk)

- # of training images:
  - 26250
  - 12500
  - 6250
  - 3125
  - 1250

Classification rate increases with increasing kNN (topk) and number of training images.