CASHEIRS: Cloud Assisted Scalable Hierarchical Encrypted Based Image Retrieval System

Xin Li, Qinghan Xue, Mooi Choo Chuah

INFOCOM, 2017
OUTLINE

• Motivation and challenges

• Selected image retrieval schemes

• Our solution

• Evaluations

• Future work

• References
OUTLINE

• Motivation and challenges
• Selected image retrieval schemes
• Our solution
• Evaluations
• Future work
• References
Motivation

1. Image retrieval techniques are needed in different applications.

- Normal search
- Online shopping
- Biological and medical research
- Social Network
Motivation

2. Sensitive Images

Magnetic resonance images

Famous people
Challenges

• 1. Variation of images:

(Image source: http://cs231n.github.io/classification/)
Challenges

• 2. Semantic gap:

<table>
<thead>
<tr>
<th>Semantic Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
</tr>
<tr>
<td>Dog</td>
</tr>
<tr>
<td>Hat</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Boat</td>
</tr>
</tbody>
</table>

What the computer sees vs. What humans see
Challenges

• 3. Evaluation Criterion:

  - Find this person.
  - Find similar clothes.
OUTLINE

• Motivation and challenges

• Selected image retrieval schemes

• Our solution

• Evaluations

• Future work

• References
Selected image retrieval schemes

B-Hie: Hierarchical Semantic Indexing for Large Scale Image Retrieval [1]

- Jia Deng, Alexander C. Berg, Li Fei-Fei
  - CVPR, 2011

- Goal:
  - This paper aims to address the problem of similar image retrieval, especially in the setting of large-scale datasets with millions to billions of images.
Four steps to build the system:

1. Calculate a similarity matrix based on a given hierarchical semantic tree.
2. Train a 1-vs-all classifier for each category.
3. Generate an attribute vector for each training image using trained classifiers.
4. Hash all images into different bins based on attribute vectors.
Query process

For each candidate image

Calculate Similarity Score:
\[
\text{Sim}(a, b) = f(a)^T S f(b)
\]

\[
= \begin{bmatrix} 0.1 & 0.2 & 0.1 & 0.6 \end{bmatrix} \begin{bmatrix} 1 & 0.6 & 0.6 & 0.4 \\ 0.6 & 1 & 0.6 & 0.4 \\ 0.6 & 0.6 & 1 & 0.4 \\ 0.4 & 0.4 & 0.4 & 1 \end{bmatrix} \begin{bmatrix} 0.05 \\ 0.80 \\ 0.05 \\ 0.10 \end{bmatrix}
\]

\[
= 0.576
\]
Limitation

• 1. The dimension of attribute vector may be very large.

• 2. Use all classifiers for each query image.
  • => Slow query response time.

• 3. Need to re-compute the attribute vectors for all images even if we only add one new category.

• 4. No privacy-aware.
Selected image retrieval schemes

SEISA: Secure and Efficient Encrypted Image Search With Access Control \cite{seisa}

- Jiawei Yuan, Shucheng Yu, Linke Guo
  - INFOCOM, 2015

- Goal:
  - This paper aims to search encrypted images in a secure and efficient way.
1. Build an index tree

a. Extract **Fisher vector** from each image.

b. Employ the **K-Means** to generate the tree.

c. Assigns a **mean vector** to each intermediate node.
2. Image retrieval

Search the most similar nodes through the index tree.
Limitation

• It is possible that the depth of the index tree can become very deep.
• => Slow query response time and low retrieval accuracy.
OUTLINE

• Motivation and challenges

• Selected image retrieval schemes

• Our solution

• Evaluations

• Future work

• References
Our solution

Build an efficient image retrieval system

• **Scalable**
  • Deal with large-scale datasets.

• **Hierarchical**
  • Speed up the query process by using hierarchical structure to quickly identify a small subset of candidate images.

• **Secure**
  • Prevent the sensitive information of images from being leaked.
System Model
Visual feature representation

Image feature --> binary code:

High-dimensional descriptor vectors

Binary encoding

1011001011...0101
1110101101...0010
0001111000...1010
1111111001...0001
1010101010...1001
0011111110...1010
0101101001...1111
1001111000...1010
0001001001...0010
1001110010...1010
1101111000...0010
1101111001...0001
0001111000...0010
1010011011...1111
Visual feature representation

Image feature --> binary code:

Similar images
\[ \| x - y \| \approx 0 \]
Visual feature representation

Image feature --> binary code:

Dissimilar images $\|x - y\|$ is large
Visual feature representation

Image feature --> binary code (ITQ):

Visual feature representation

Representative Vector (RV):

<table>
<thead>
<tr>
<th>Binary Code:</th>
<th>Image 1</th>
<th>Image 2</th>
<th>Image 3</th>
<th>...</th>
<th>Image M</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Image 2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Image 3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image M</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Mean Vector:

| 0.4 | 0.6 | 1.0 | 0.8 | 0.4 | 0.4 | 0.8 | 0.8 | 0.8 | ... | 0.8 | 0.4 | 0.6 | 0.6 |

Representative Vector:

| 0   | 0   | 1   | 1   | 0   | 0   | 0   | 1   | 0   | 1   | ... | 1   | 0   | 0   | 0   |
Hierarchical index tree building

Cluster Index: Lev h, Ind i

\[
\frac{|D(a,b) - \max(IND_a, IND_b)|}{\max(IND_a, IND_b)} > 0.5 \text{ (threshold)}
\]

Level 2:

Level 1:

Level 0:

K-means Result:

Images:
Hierarchical index tree building

Cluster Index: Lev h, Ind j

RV E IND

\[
\frac{|D(a,b) - \max(IND_a, IND_b)|}{\max(IND_a, IND_b)} > 0.5 \text{ (threshold)}
\]

Level 3:

Level 2:

Level 1:

Level 0:

K-means Result:

Images:
Query process

$$w_{hj} = 1 - \frac{D(Q, RV_{hj})}{\sum_{i=0}^{Z} D(Q, RV_{hi})}$$

$$\text{Sum}_\text{Chosen} = \sum_{c=0}^{k} w_c > 0.8 \text{ (threshold)}$$
# Privacy

<table>
<thead>
<tr>
<th>Binary Code:</th>
<th>RV₁ (Cloud)</th>
<th>RV₂ (User)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 1 1 0 0 1</td>
<td>0 1 1 1 0 0 0</td>
<td></td>
</tr>
</tbody>
</table>

**Step 1:**

\[
RV₁' = -1 \quad -1 \quad 1 \quad 1 \quad -1 \quad -1 \quad 1
\]

\[
S₁ = 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1
\]

**Step 2:**

\[
RV₁₁' = -1 \quad a₁ \quad a₂ \quad 1 \quad a₄ \quad -1 \quad 1
\]

\[
RV₁₂' = -1 \quad b₁ \quad b₂ \quad 1 \quad b₄ \quad -1 \quad 1
\]

\[
RV₂' = -1 \quad 1 \quad 1 \quad 1 \quad -1 \quad -1 \quad -1
\]

\[
S₂ = 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0
\]

**Step 3:**

Generate invertible random matrices \(M₁\) and \(M₂\)

**Step 4:**

\[Enc(RV₁') = \{M₁^T RV₁₁', \quad M₂^T RV₁₂'\}\]

\[Enc(RV₂') = \{M₁⁻¹ RV₂₁', \quad M₂⁻¹ RV₂₂'\}\]

**Distance:**

\[D(RV₁, RV₂) = \frac{n - RV₁' \cdot RV₂'}{2} = \frac{n - Enc(RV₁') \cdot Enc(RV₂')}{2}\]
Query process (with encryption)
OUTLINE

• Motivation and challenges

• Selected image retrieval schemes

• Our solution

• Evaluations

• Future work

• References
Evaluations

• Metrics:
  • Precision at top k (P@k)
    \[ P@k = \frac{\text{num\_correct}}{k} \]
  • Mean average precision (mAP)
    \[ AP(q) = \frac{\sum_{k=1}^{n} (P@k \times \text{rel}(k))}{N} \]

• Dataset:
  • Caltech256
    • 30608 images, 256 object categories
  • INRIA Holiday
    • 1491 images, 500 image groups

• Evaluate on:
  • A laptop running OS X
  • 2.5GHz Intel Core i7 CPU
  • 16GB Memory
Evaluations – Effectiveness of RV

- 50 images per category are used
- Using image category label to build the index tree instead of K-Means.
Evaluations – System Construction

Computation time of building a hierarchical index tree.

- 50 images per category are used
- binary code: PCA-128
Evaluations – Storage Cost

CASHEIRS: (binary code: PCA-128 )

• CNN model: 244.7 MB [Data owner | Data user]
• PCA matrix: 3.3 MB [Data owner | Data user]
• Rotation matrix: 105 KB [Data owner | Data user]
• Encrypted key: 1.7 KB [Data owner | Data user]

• Encrypted images: 14.28 KB/image (300 x 200 pixels) [Data owner | Cloud]
• Encrypted index tree: 440 KB (cat_50) [Data owner | Cloud]
• Image features: 5.3 MB (cat_50) [Data owner | Cloud]

<table>
<thead>
<tr>
<th>Cloud Storage Cost (MB)</th>
<th>Cat_50</th>
<th>Cat_100</th>
<th>Cat_150</th>
<th>Cat_200</th>
<th>Cat_250</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASHEIRS</td>
<td>40.6</td>
<td>81.0</td>
<td>121.5</td>
<td>161.4</td>
<td>201.7</td>
</tr>
<tr>
<td>B-Hie</td>
<td>59.8</td>
<td>120.4</td>
<td>181.0</td>
<td>241.7</td>
<td>302.4</td>
</tr>
<tr>
<td>OASIS</td>
<td>109.3</td>
<td>197.4</td>
<td>285.5</td>
<td>373.6</td>
<td>461.7</td>
</tr>
<tr>
<td>SEISA (PCA-128)</td>
<td>40.3</td>
<td>80.8</td>
<td>122.3</td>
<td>167.1</td>
<td>211.8</td>
</tr>
<tr>
<td>SEISA (PCA-512)</td>
<td>55.7</td>
<td>113.9</td>
<td>175.9</td>
<td>239.6</td>
<td>303.3</td>
</tr>
</tbody>
</table>
Evaluations – Search Evaluation

We randomly generate 10 Million image features (128 dimension), and then mix them with the features extracted from INRIA Holiday dataset. (Same set-up as SEISA)

**TABLE II: Comparison Results**

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Search Time (ms)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Category-10</td>
<td>Category-20</td>
<td>Category-50</td>
</tr>
<tr>
<td>CASHEIRS</td>
<td>4.1</td>
<td>9.6</td>
<td>12.9</td>
</tr>
<tr>
<td>B-Hie</td>
<td>13.3</td>
<td>23.2</td>
<td>52.2</td>
</tr>
<tr>
<td>OASIS</td>
<td>131.9</td>
<td>132.6</td>
<td>133.9</td>
</tr>
</tbody>
</table>

**INRIA Holiday (10 million images)**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Search Time (ms)</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASHEIRS</td>
<td>95.2</td>
<td>0.64</td>
</tr>
<tr>
<td>SEISA</td>
<td>87.5</td>
<td>0.55</td>
</tr>
</tbody>
</table>

Accuracy on Caltech256 dataset

- 10 classes
- 20 classes
- 50 classes
Evaluations – Search Evaluation

Communication Cost:

Data owner to the cloud:

11.2 MB encrypted index tree [one time]
14.28 KB/encrypted image (300 x 200 pixels) [one time]

A user:

[one time]
receive 248.1 MB CNN model and encryption keys from the Data owner

[Querying]
send 2 KB encrypted query information to the cloud
receive 14.28 KB/encrypted image (300 x 200 pixels) from the cloud
Prototype

• Cloud server:
  • A laptop running OS X
  • 2.5GHz Intel Core i7 CPU
  • 16GB Memory

• Client
  • Samsung S5 phone
  • Snapdragon 801 chip with 2.5GHz Quad-core CPU
  • 2GB RAM

• Communication
  • WIFI router
OUTLINE

• Motivation and challenges
• Selected image retrieval schemes
• Our solution
• Evaluations
• Future work
• References
Future work

- Using larger image datasets, e.g. ImageNet.
- More efficient security solution.
  - 128 bit binary code into 2 KB encrypted information.
- Using deep learning model to generate a binary code directly.
Conclusion

• Challenges in the field of image retrieval.
• Related papers and their limitations.
• Propose our solution.
References


