Harvesting Image Databases from the Web

Author: F. Schroff, A. Criminisi, A. Zisserman

Presenter: Duo Xu, Bo Feng
Problems with “Animals on the Web”
Problems with “Animals on the Web”

• For the training dataset, after applying the LDA algorithm to get the 10 topics, need users to identify which topics are relevant (positive) or background (negative), i.e. need manual intervention.

• Given a new class, e.g. “shark”, how could we automatically get images to match it?
Overview

- Retrieve dataset from web
- Remove noise (drawings & symbolic Images)
- Rank images using text feature
- Train visual classifier
- Rank images
Overview

- Retrieve dataset from web
- Remove noise (drawings & symbolic Images)
- Rank images using text feature
- Train visual classifier

- Web Search ; Image Search ; Google Images
- Image features + SVM (with RBF kernel)
- Text features + Bayes classifier
- Image features + SVM (with RBF kernel)
- Rank based on SVM score
Step 1 - Dataset

- Retrieve dataset from web
- Remove noise (drawings & symbolic images)
- Rank images using text feature
- Train visual classifier
- Rank images

Web Search; Image Search; Google Images
Retrieve dataset from web - I

• Web Search
  – Query word by using Google text web search, get all images that are linked within the returned web pages

• Image Search
  – Using Google image search find image, using each image as a “seed” – further images are downloaded from the web page from where the seed image originated.

• Google Images
  – Only images directly returned by Google image search
Retrieve dataset from web - II

• Query words not only include the specific “word”, but also related word, e.g. for “penguin”, also search “penguin animal”, “penguin or penguins”
• Images smaller than 120 x 120 are discarded.
• Also download text surrounding the image in HTML and meta-data, image filenames, etc.
Retrieve dataset from web - III

- Annotated manually into 3 types:
  - In-class-good: in one class in a clearly visible way
  - In-class-ok: show parts of a class instance, or obfuscated due to lighting, etc.
  - Non-class: not belonging to in-class

- Further, “good” and “ok” sets divided into 2 types:
  - Abstract: not realistic natural, (e.g. comics)
  - Non-abstract
Figure 2. **Image annotations**: Example images corresponding to annotation categories for the class penguin.
Retrieve dataset from web - IV

- ImageSearch is not used due to a very low precision

<table>
<thead>
<tr>
<th>Service</th>
<th>in-class</th>
<th>non-class</th>
<th>precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebSearch</td>
<td>8773</td>
<td>25252</td>
<td>26%</td>
</tr>
<tr>
<td>ImageSearch</td>
<td>5963</td>
<td>135432</td>
<td>4%</td>
</tr>
<tr>
<td>GoogleImages</td>
<td>4416</td>
<td>6766</td>
<td>39%</td>
</tr>
</tbody>
</table>

Table 1. **Statistics by source**: The statistics of downloaded images for different retrieval techniques.
Step 2 – Remove noise

- Retrieve dataset from web
- Remove noise (drawings & symbolic Images)
- Rank images using text feature
- Train visual classifier
- Rank images

Image features + SVM (with RBF kernel)
Remove noise - I

• Goal : remove drawings and symbolic images from the downloaded images, like comics, graphs, plot, maps, charts, drawings.
Remove noise - II

• Prepare training dataset
  – Using ImageSearch, such as “sketch” or “drawing” or “draft” to download images
  – Manually select suitable training images
  – 1400 drawing & symbolic images, 2000 non drawing & symbolic images
Remove noise - III

• Image features:
  – Histogram of color
  – Histogram of L2-norm of gradient
  – Histogram of the angles (0 ... π) weighted by the L2-norm of the corresponding gradient

• Classifier:
  – SVM with RBF kernel
Remove noise -IV

• Training & Testing
  – Using two-fold cross-validation
  – Achieve about 90% accuracy on the drawing & symbolic images

• Apply to the downloaded image dataset in the first step.
  – Removed 42% non-class images
  – Removed 60% in-class abstract images
  – Removed 13% in-class non-abstract images
Remove noise -V

<table>
<thead>
<tr>
<th>Class</th>
<th>downloaded images</th>
<th>after filtering</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>in-cl.</td>
<td>non-cl.</td>
<td>prec.</td>
</tr>
<tr>
<td>airplane (ap)</td>
<td>843</td>
<td>1720</td>
<td>32.89%</td>
</tr>
<tr>
<td>beaver (bv)</td>
<td>201</td>
<td>3156</td>
<td>5.99%</td>
</tr>
<tr>
<td>bikes (bk)</td>
<td>1236</td>
<td>1963</td>
<td>38.64%</td>
</tr>
<tr>
<td>boat (bt)</td>
<td>861</td>
<td>2170</td>
<td>28.41%</td>
</tr>
<tr>
<td>camel (cm)</td>
<td>492</td>
<td>1910</td>
<td>20.48%</td>
</tr>
<tr>
<td>car (cr)</td>
<td>1125</td>
<td>1045</td>
<td>51.84%</td>
</tr>
<tr>
<td>dolphin (dp)</td>
<td>663</td>
<td>1544</td>
<td>30.04%</td>
</tr>
<tr>
<td>elephant (ep)</td>
<td>660</td>
<td>1835</td>
<td>26.45%</td>
</tr>
<tr>
<td>giraffe (gf)</td>
<td>779</td>
<td>1433</td>
<td>35.22%</td>
</tr>
<tr>
<td>guitar (gr)</td>
<td>1261</td>
<td>1993</td>
<td>38.75%</td>
</tr>
<tr>
<td>horse (hs)</td>
<td>963</td>
<td>1986</td>
<td>32.66%</td>
</tr>
<tr>
<td>kangaroo (kg)</td>
<td>295</td>
<td>1886</td>
<td>13.53%</td>
</tr>
<tr>
<td>motorbikes (mb)</td>
<td>704</td>
<td>981</td>
<td>41.78%</td>
</tr>
<tr>
<td>penguin (pg)</td>
<td>664</td>
<td>1484</td>
<td>30.91%</td>
</tr>
<tr>
<td>shark (sk)</td>
<td>522</td>
<td>1771</td>
<td>22.76%</td>
</tr>
<tr>
<td>tiger (tr)</td>
<td>333</td>
<td>2114</td>
<td>13.61%</td>
</tr>
<tr>
<td>wristwatch (ww)</td>
<td>916</td>
<td>982</td>
<td>48.26%</td>
</tr>
<tr>
<td>zebra (wb)</td>
<td>427</td>
<td>1718</td>
<td>19.91%</td>
</tr>
<tr>
<td>total</td>
<td>12945</td>
<td>31691</td>
<td>29.00%</td>
</tr>
</tbody>
</table>

Table 2. Image class statistics of the original downloaded images using WebSearch&GoogleImages only, and after applying the drawing&symbolic images removal filter.
Step 3 – Text features

- Retrieve dataset from web
- Remove noise (drawings & symbolic Images)
- Rank images using text feature
- Train visual classifier
- Rank images

Text features + Bayes classifier
Rank images using text feature - I

• 7 textual features:
  – contextR : words on the web page between 11 and 50 words away from the image-link.
  – Context10 : 10 words on either side of the image-link
  – filedir
  – filename
  – image alt
  – image title
  – website title
Penguin

From Wikipedia, the free encyclopedia

For other uses, see Penguin (disambiguation).

Penguins (order Sphenisciformes, family Spheniscidae) are a group of aquatic, flightless birds living almost exclusively in the southern hemisphere, especially in Antarctica. Highly adapted for life in the water, penguins have countershaded dark and white plumage, and their wings have evolved into flippers. Most penguins feed on krill, fish, squid, and other forms of sealife caught while swimming underwater. They spend about half of their lives on land and half in the oceans.

Although all penguin species are native to the southern hemisphere, they are not found only in cold climates, such as Antarctica. In fact, only a few species of penguin live so far south. Several species are found in the temperate zone, and one species, the Galápagos Penguin, lives near the equator.

The largest living species is the Emperor Penguin (Aptenodytes forsteri): on average adults are about 1.1 m (3 ft 7 in) tall and weigh 25 kg (55 lb) or more. The smallest penguin species is the Little

![Image of a penguin](https://upload.wikimedia.org/wikipedia/commons/thumb/b/be/Pygoscelis_papua.jpg/250px-Pygoscelis_papua.jpg)

Gentoo Penguin. Gentoo Penguin
Rank images using text feature - II

• Text processing :
  – Use stop word list and Porter stemmer
  – Also ignore html tags and “html stop words”, like &nbsp;

• Each of these 7 features is treated as a binary feature vector, “True” (1) if it contains the query word (e.g. penguin), otherwise, “False” (0)
  – $a = (a_1, a_2, \ldots, a_7)$, where $a_1, a_2, \ldots, a_7$ belongs to \{0, 1\}
Rank images using text feature - III

• Prepare dataset
  – Suppose we want to re-rank images in penguin class, using other 17 classes to learn the probabilities

• In-class :
  – (1, 1, 1, 0, 1, 0, 1), (1, 1, 1, 0, 0, 0, 1), (1, 1, 0, 1, 1, 0, 0) ...

• Non-class:
  – (1, 1, 0, 0, 0, 0, 0), (0, 1, 0, 0, 0, 0, 1) ...
Rank images using text feature - IV

• Bayesian Posterior estimation is used, with:

\[ P(y|a) = \frac{P(a|y)P(y)}{P(a)} \]

• where:

\[ P(a|y) = P(a_1, \ldots, a_4|y) \prod_{5}^{7} P(a_i|y) \]

\( y \in \{\text{in-class, non-class}\} \)
Rank images using text feature - V

• Advantages:
  – A completely automatic class independent image ranker; i.e. for any new and unknown class, the images can be re-ranked without ever using labeled truth knowledge of that class.
Rank images using text feature - VI

- Precision & Recall for bayes classifier
Step 4 – Train visual classifier

- Retrieve dataset from web
- Remove noise (drawings & symbolic Images)
- Rank images using text feature
- Train visual classifier

Image features + SVM (with RBF kernel)
Train visual classifier - I

• Prepare training dataset:
  – Positive: Top (150/250) images from the text ranked list
  – Negative: Any images (250/500/1000) from all downloaded images
Train visual classifier - II

• Build features:
  – Detect regions by using 4 detectors:
    • Difference of Gaussians
    • Multiscale-Harris
    • Kadir’s saliency operator
    • Canny edge points
  – Region is represented as a 72 dimensional SIFT descriptor
  – 100 visual words is learnt for each detector using k-means, combined to 400 visual words.
  – Assign descriptor of each region to this vocabulary, each image can be represented as a 400 dimension vector.
Train visual classifier - III

- **Classifier:**
  - SVM with RBF kernel

- **Tools:**
  - SVM \textsuperscript{light}
Step 5 – Re-ranking

- Retrieve dataset from web
- Remove noise (drawings & symbolic Images)
- Rank images using text feature
- Train visual classifier
- Rank images
- Rank based on SVM score
Rank images - I

- The trained SVM is used to re-rank the filtered image set, based on the SVM classification score.
Rank images - II
Rank images - III

Figure 6. Top ranked 36 images of zebra, wristwatch and car using the text+vision algorithm.
Rank images - IV

Figure 7. **Comparison with Google image search.** Precision at 100 image recall.
Figure 8. **Comparison with Berg et al. [5]**. Precision at 100-image recall level for the 10 animal classes made available by the authors of [5]. Note that our automatic algorithm is superior in many cases, even though the method of [5] involves manual intervention.
Summary

• Download images and meta-data for new class (e.g. “lion”) using WebSearch & GoogleImages
• Filter images: remove drawings & symbolic images
• Rank images based on text-attributes using the Bayes classifier.
• Train visual SVM classifier on text-ranked images
• Rank all images from 1. using the visual classifier.
Summary

Retrieve dataset from web

Remove noise (drawings & symbolic Images)

Rank images using text feature

Train visual classifier

Rank images

Web Search ; Image Search ; Google Images

Image features + SVM (with RBF kernel)

Text features + Bayes classifier

Image features + SVM (with RBF kernel)

Rank based on SVM score
Discussion

• Practical. May build your own image search engine with a fairly good result.

• Not many innovations, combine different features and different classifiers (bayes, SVM) together

• Improvement ? e.g. Better features ?
Thank you