Scalable Logo Recognition in Real-World Images

Stefan Romberg* (romberg@informatik.uni-augsburg.de)
Lluis Garcia Pueyo (lluis@yahoo-inc.com)
Roelof van Zwol (roelof@yahoo-inc.com)
Rainer Lienhart (lienhart@informatik.uni-augsburg.de)
Outline

• Introduction
• Idea
• Training
• Recognition / Cascaded Index
• Dataset
• Results
What is a logo?

A logo

- is the visual representation of a trademark/brand
- is a 2-D plane, a “flat” object: Only one distinct perspective
- is made to be visually outstanding.
  (e.g. single colored background, high contrast, high saturation)
- is designed to be immediately recognized by humans:
  (high distinctiveness and consistent appearance)

A logo class = logos of a certain brand with a single consistent appearance

vs. Same brand, two different classes
Applications

- Content-related advertising
- Product recognition
- Screening of advertisements on TV
  Companies have paid for it and want to check what they get
- Other motivation:
  Logo recognition is a sub-problem of object class recognition
Goal

Application requirements:

- Fast and efficient recognition of multiple logo classes
  - Approach should be potentially extensible to thousands of classes
  - Cannot use 1-vs-all classifier for each class (problems with speed and classifier agreement)

- High precision more important than recall for most applications
- Must work with real-world images

Immediate design choices:
- Use sparse local features
- Quantize high-dimensional descriptors to visual words
  - Efficient storage
Fundamental idea:

- **Training**: index spatial layout of local features
  - training images

- **Recognition**: use index as dictionary
  - Verify if spatial layout in test image is present in logo class by index lookup

**Idea:** Indexing of spatial structure
Feature Triples

Description of spatial layout by geometric primitives

- Can be efficiently stored in an index (=hash tables)
- Each class represented by many items in index

Notation:

- two local features = feature pair => representation: edge
- three local features = feature triple => representation: triangle

Choice: Encoding of local feature layout by feature triples

- Encode relative position and orientation
  => Resulting shape is triangle => Store triangle signature in index
Feature order given by visual word labels $i, j, k$ sorted by ID
- Quantized angles and quantized relative feature orientations
- Scale and rotation invariant (in-plane)
- Quantization deals with out-of-plane rotations
Training

- Training by determining correspondences in image pairs

Training images of class "Adidas"

representation

Adidas

store

Index of all classes
Problem:
- How to test unknown image with N features for specific feature triples?
  - No prior knowledge (position/size)!
  - Draw random features?
  - \( N^3 \) possible combinations to create triple!

Solution:
- Do not test feature triples directly
  => Use cascaded index
Cascaded Index

Cascaded index = two linked indexes:

- Edge index holds edge representations
- Triangle index holds triangle representations

Cascaded Index:

- All edges in edge index are part of 1+ triangles in the triangle index
- One hit in edge index \( \iff \) edge found
  \( \iff \) part of a triangle in triangle index found

- Edge index describes smaller space of feature pairs
- Triangle index is superspace of edge index
Given test image with set of local features:

Step (1): Determine **edges** that are part of *any* logo:

1. Scan local feature neighborhoods
   - Select a local feature and each of its spatial nearest neighbors
   - Tests feature pairs close to each other
2. Monte Carlo method
   - Randomly sample feature pairs
   - Large number of samples (e.g. 100K random samples)
   - Covers bigger areas in image quickly

All found edges $E$ are then used for querying the second index holding triangle representations.
After Step (1):
All edges $E$ that have been found in the edge index are part of a triangle
⇒ Use these edges as hint and test feature triples with triangle index

Step (2): Scan test image for triangles part of any logo:
- For the desired number of random samples:
  - Construct triangle from feature triple $(i, j, k)$:
    - Draw $(i, j)$ from edge set $E$, draw $k$ from set of points given by $E$
    - If signature for $(i, j, k)$ is found in triangle index
      => One vote for certain logo class
Recognition:
A logo is considered to be present in test image if the number of detected triangles of a certain class $> T_{class}$

Observations:
- Logos vary in visual complexity / common observed size
- Number of detections vary

=> Adjust $T_{class}$ for each class separately with validation set:
- Increase threshold until 95% precision is met

Adjustment of thresholds is done for all experiments:
⇒ Fixed precision, optimization for recall
⇒ Increasing noise leads to decreasing recall
FlickrLogos-32 dataset

32 logo classes downloaded from Flickr

- training set: 10 manually picked images per class
- validation set: 30 images per class, 3000 non-logo images
- test set: 30 images per class, 3000 non-logo images

Dataset available for download:
http://www.multimedia-computing.de/flickrlogos
Examples
Varying number of queries to cascaded index
Scanning of spatial neighbors is beneficial
Experiment 2: Vocabulary sizes

- Varying number of vocabulary sizes
- Larger vocabularies perform better
- Color SIFT performs slightly better than grayscale SIFT
Experiment 3: Triangle shape

- Evaluation of parameters determining coarseness of triangle representation
Results

ICMR 2011, Scalable Logo Recognition in Real-World Images
Summary

- Multi-class logo recognition by indexing of class-specific feature layout
- Representation of spatial layout of local features as triangles
- Indexing of spatial layout within cascaded index
  - lower-dimensional space (features pairs)
  - high-dimensional space (feature triples)
- High precision: 95%, Recall: 60%
Thank you!