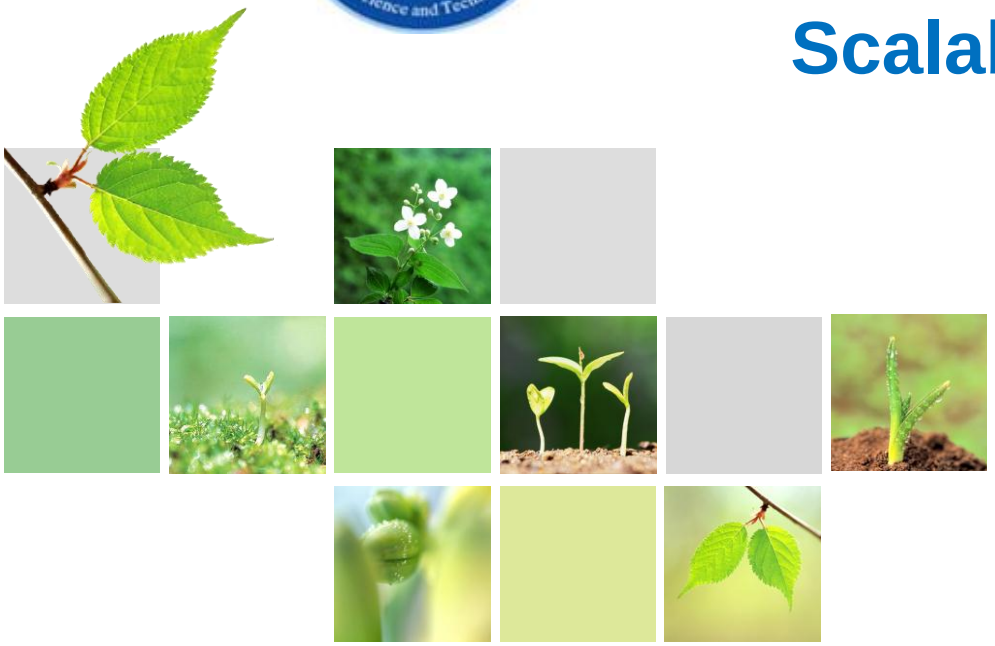
L/O/G/O



Multiview Visual Search based on Scalable Vocabulary Tree

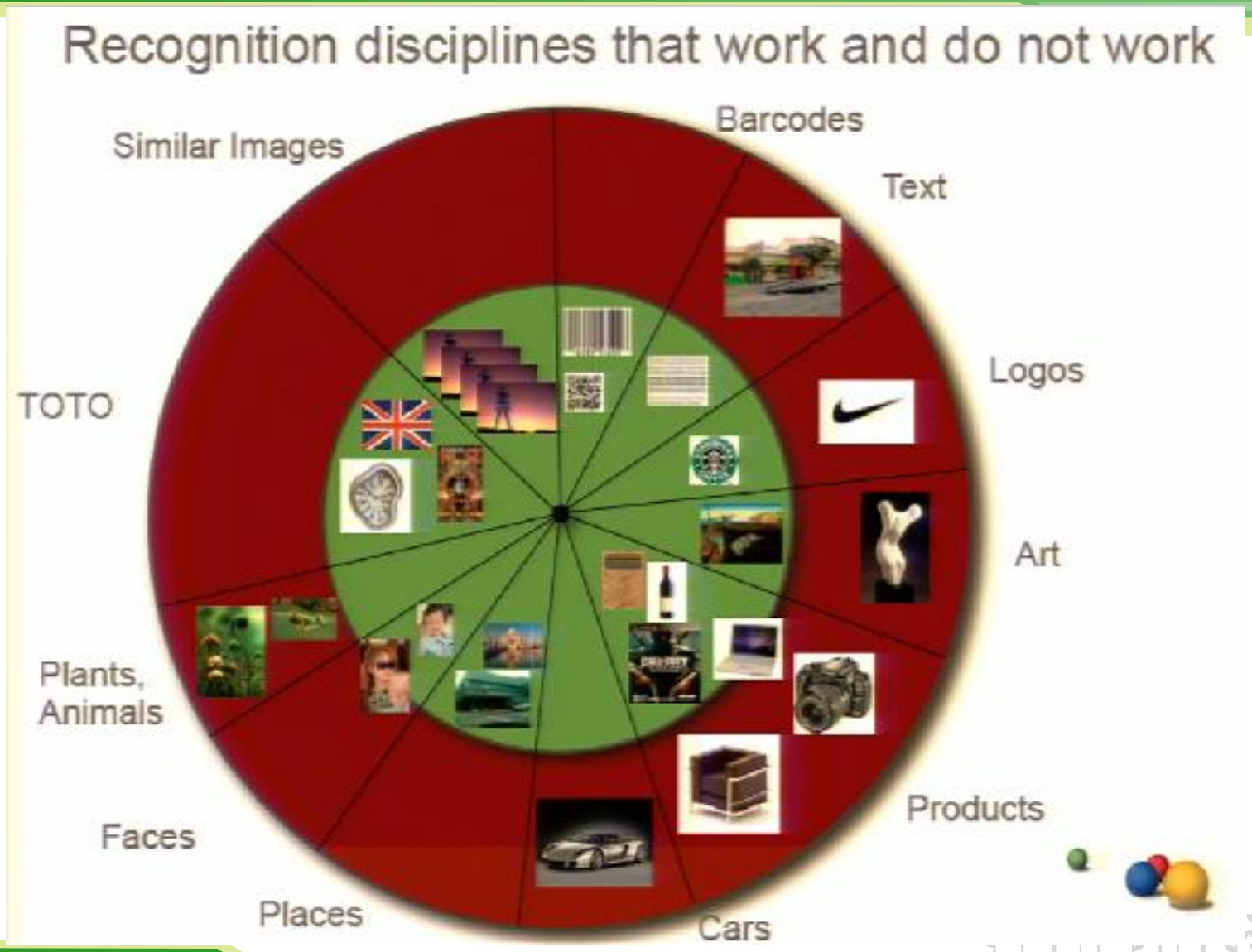
Tao Liu
Dept. EEIS, USTC

Contents

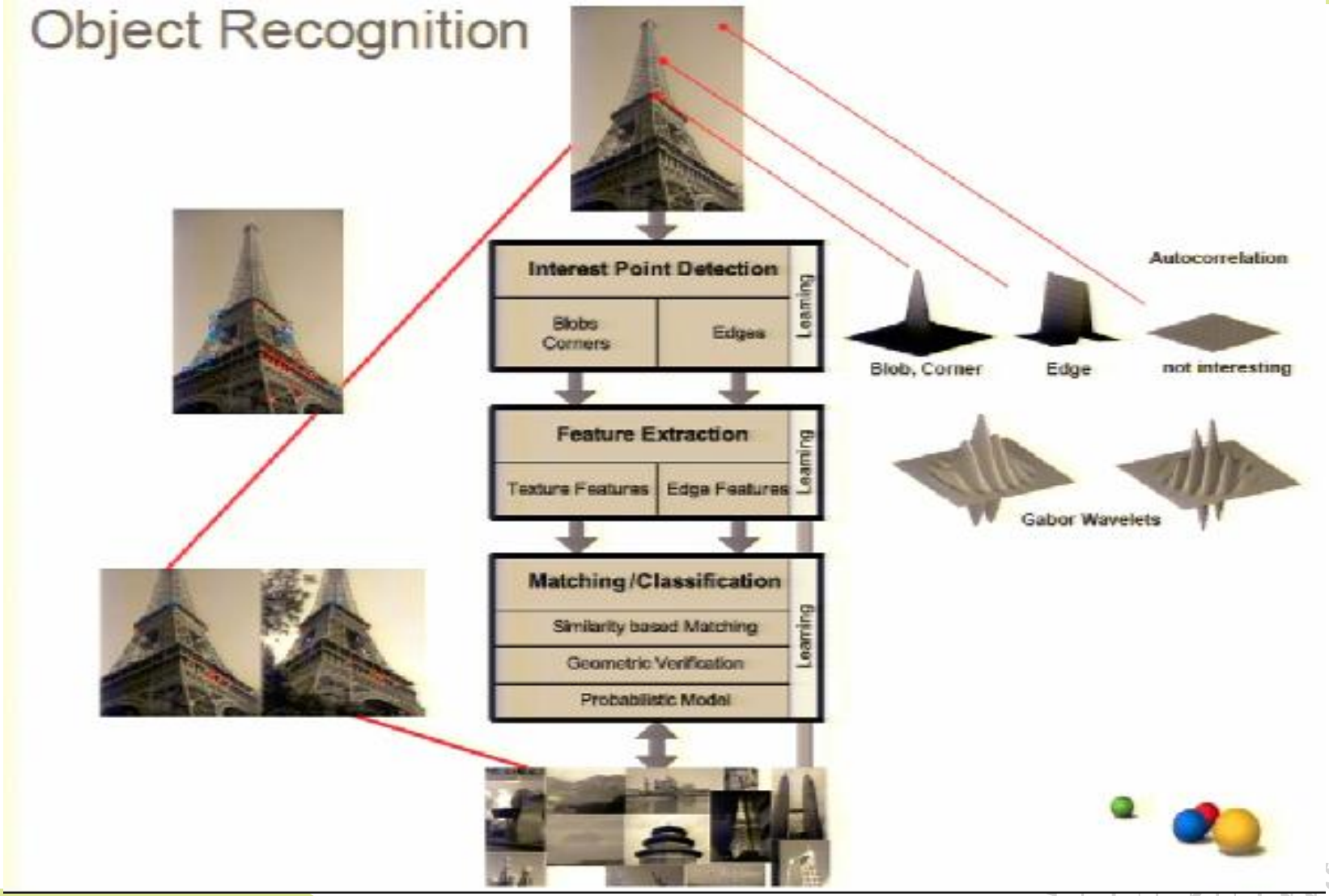
Mobile Visual Search

Company	Platform	Product	Targets
Google	Android	Goggles	Landmark/Book/Artwork/Grocery
Amazon	iPhone	Snaptell	Book/DVD/Game covers
Nokia	Symbian	PointAndFind	Landmark/Barcode/Movie poster
Kooaba	iPhone	Kooaba	Book/DVD/Game covers
oMoby	iPhone	oMoby	General Objects

Google Goggles

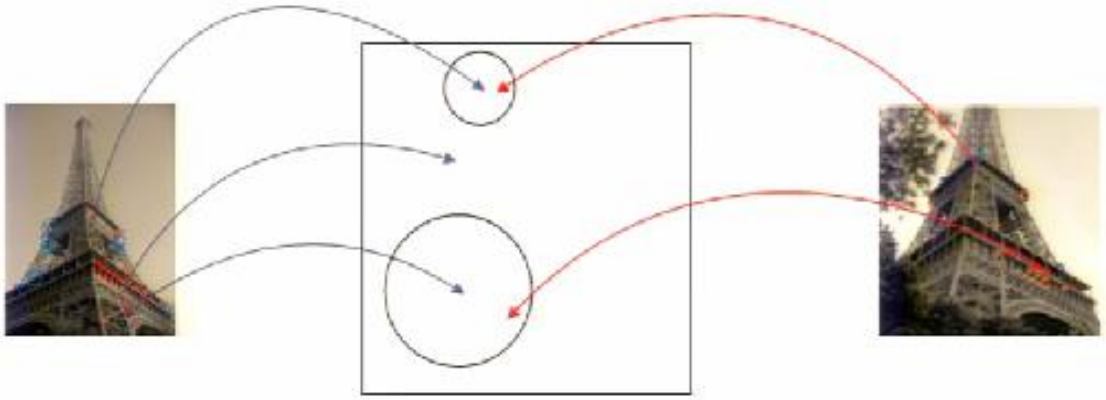


Google Goggles



Approximate nearest neighbor search Full representation

James Philbin, Anand Pillai

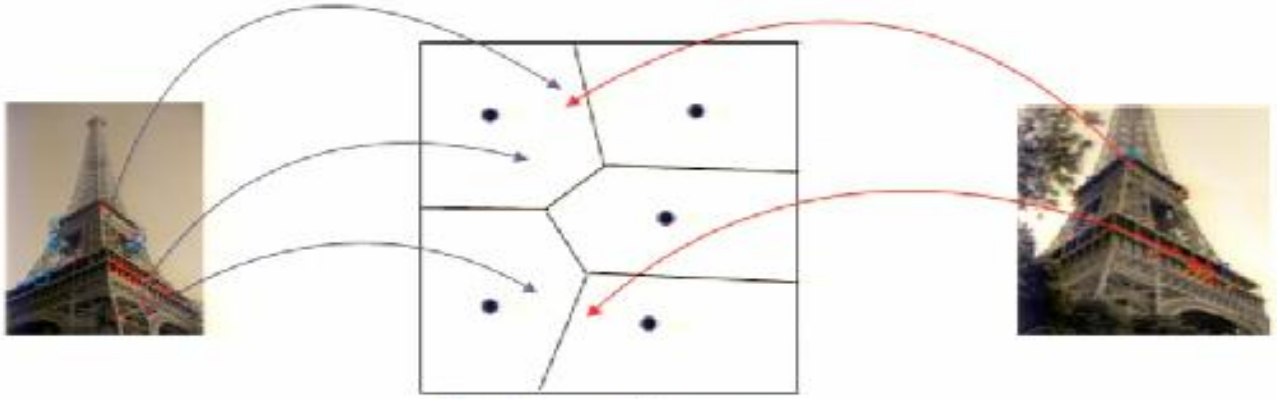


Feature vector representation maintained

Pro: more accurate NN finding

Google Goggles

Approximate nearest neighbor search Visual Words

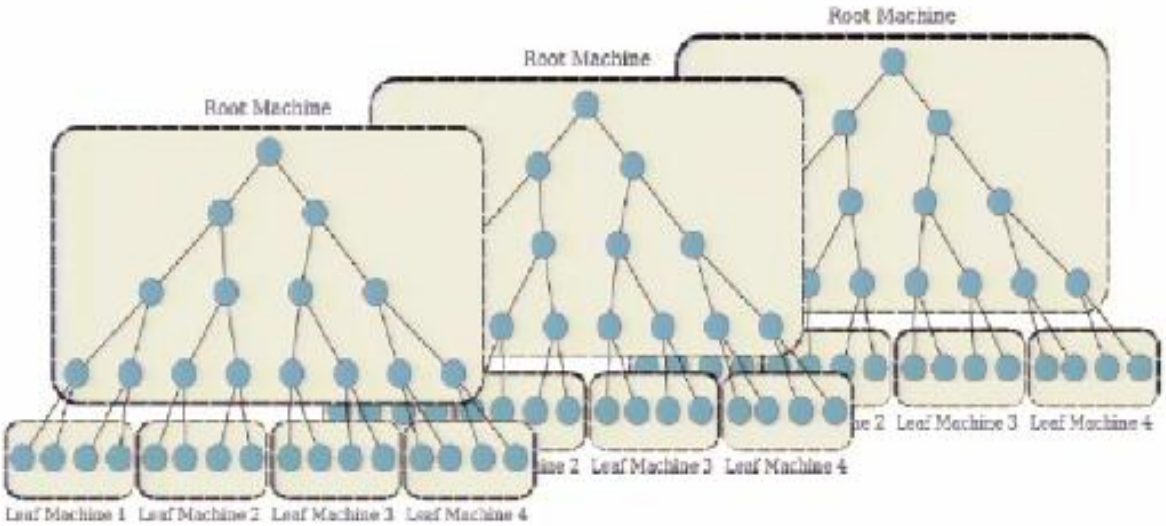


Feature represented by id of Voronoi cell

Pros: less memory, easier to parallelize, index can be upgraded easier

Google Goggles

Before: NN search with parallelized kd-tree



Google Goggles

Now hybrid: Visual words but apply product quantization to each entry in Voronoi cell

Variance balancing and co-occurrence statistics boosts 1-NN precision from 91.5% to 95.3% and 5-NN precision from 86.9% to 91.4%.

Contents

Recognition scheme

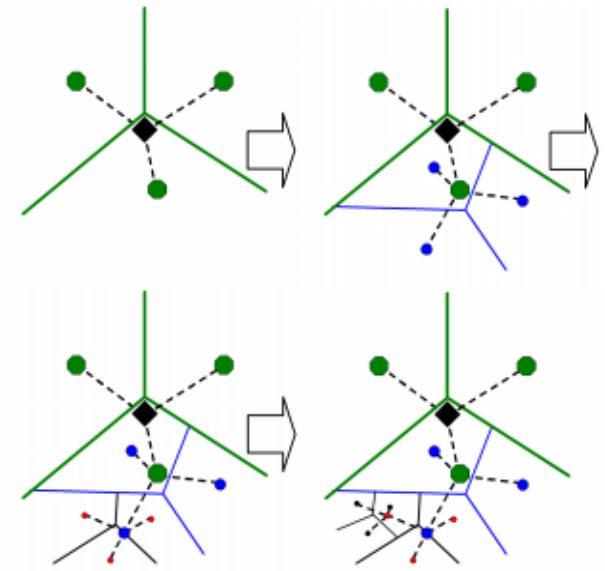
- **Vocabulary Tree** defined using an offline unsupervised training stage.
- **Hierarchical scoring** based on Term Frequency Inverse Document Frequency (TF-IDF).
- **Local features** Colored Scale Invariant Feature Transform (CSIFT).
- **Multiview search** Using multi-view images to search.
- **Fast geometric re-rank** shorten the list of candidates for the complex geometric verification.
- **Geometric consistency check** reduces false positives and allows spatial localization of the object within the query frame.

Vocabulary Tree

- The vocabulary tree defines a hierarchical quantization built by hierarchical k-means clustering

➢ A large set of representative descriptor vectors are used in the unsupervised training of the tree.

➢ K defines the branch factor of the tree.



Advantages of Vocabulary Tree

Efficient

- more efficient training through a hierarchical k-means approach
- on-the-fly insertion of new objects into the database
- speed up queries via inverted index compression

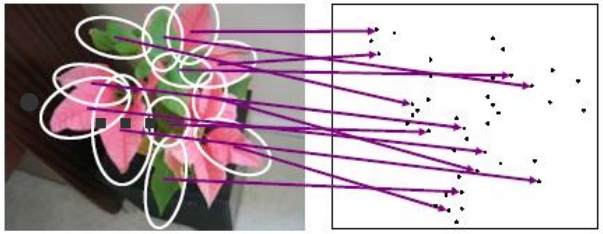
Compact

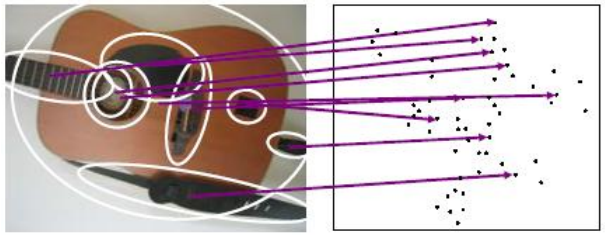
- the tree directly defines the quantization
- the quantization and the indexing are therefore fully integrate
- representation of an image patch is simply one or two integers

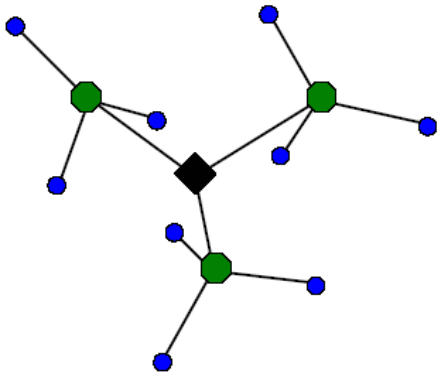
Better retrieval quality

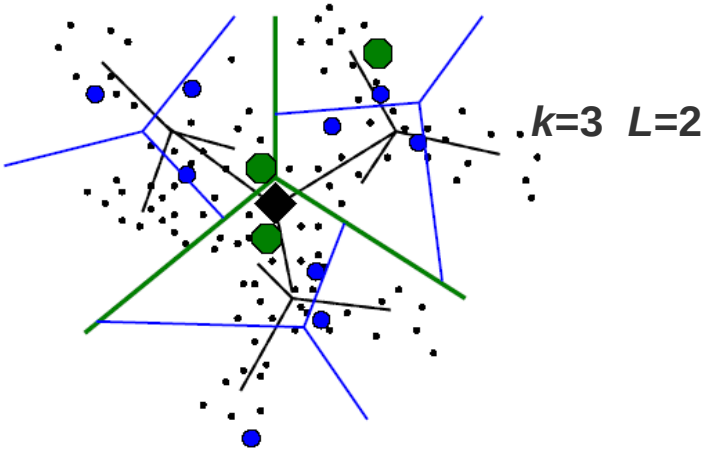
- allows a larger and more discriminatory vocabulary to be used efficiently, which leads to a dramatic improvement in retrieval quality

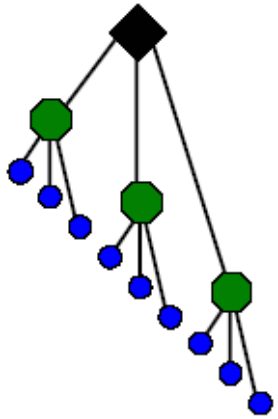
Building the Vocabulary Tree



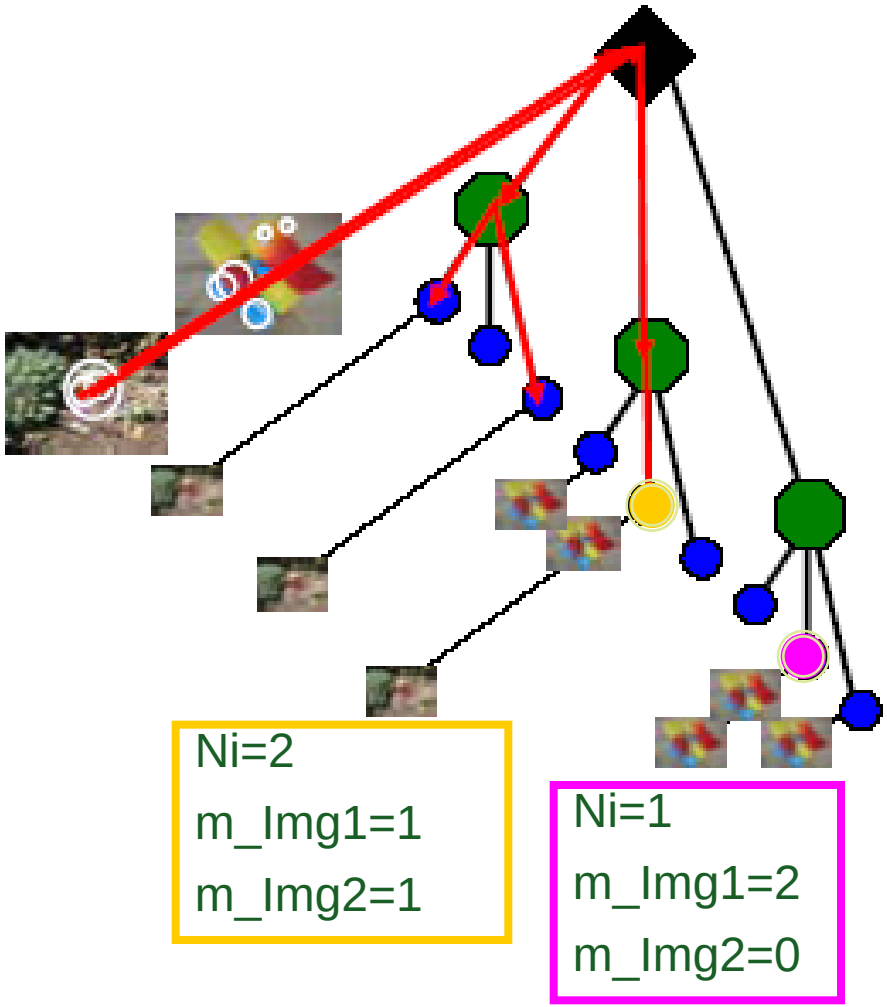








Definition of Scoring



- Number of the descriptor vectors of each image with a path along the node *i* (n_i query, m_i database)
- Number of images in the database with at least one descriptor vector path through the node *i* (N_i)

Definition of Scoring

- Weights are assigned to each node

$$w_i = \ln \frac{N}{N_i}$$

- Query and database vectors are defined according to their assigned weights

$$q_i = n_i w_i$$
$$d_i = m_i w_i$$

- Each database image is given a relevance score based on the normalized difference between the query and the database vectors

$$s(q,d) = \| \frac{q}{\| q \|} - \frac{d}{\| d \|} \|$$

$$\mathrm{sim}(\mathbf{v}_q, \mathbf{v}_d) = \frac{\mathbf{v}_q^\top \mathbf{v}_d}{\|\mathbf{v}_q\|_2 \, \|\mathbf{v}_d\|_2},$$

Implementation of Scoring

- Analogy with text retrieval inverted file systems and document rankings are used.
- Every node in the vocabulary tree is associated with an inverted file.
- Inverted files stored the id-numbers of the images in which a particular node occurs and the term frequency of that image.
- Decrease the fraction of images in the database that have to be explicitely considered for a query.

Hierarchical TF-IDF scoring

- TF-IDF value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others.

- The leaf nodes are simply much more powerful than the inner nodes.

- In the experiment, I just use the score of leaf nodes.

An improved method to compute IDF

A large representative database to determine the IDF(entropies)

- Track the path of each sift feature when building the tree.

- Get a path matrix when tree completed.

  Path matrix: one column per sift feature and height equal to the depth of the tree. Each column encodes the branch of the tree that correspond to each sift feature.

- Compute the IDF using the matrix and the sift number of each image.
  - For each image i in database:

    get the index of each sift feature in leaf node (index(i)=∑(Path(m,n)-1)*power(K,depth-m)

    get a unique index array of each image. (unique(index))
  - Calculate the number of images with at least one sift feature path through each node.
  - Entropy weight= $$w_i = \ln \frac{N}{N_i}$$

Local features-CSIFT

- SIFT has been proven to be the most robust local invariant feature descriptor.

- SIFT is designed mainly for gray images. However, color provides valuable information in object description and matching tasks.

- The built Colored SIFT (CSIFT) is more robust than the conventional SIFT with respect to color and photometrical variations.

Multiview Search

- How to distinguish different views?
- How to select the optimal views?
- How to use distinctive views to achieve more accurate search?

Algorithm for selecting optimal views

1.Obtain many views of the object from different view points from the video frames.

2.Extract the SIFT feature of the images obtained in step 1.

3.Select a threshold value I which will be used to define which views are similar. If the matching of SIFT feature between two views is less than t they are marked as similar.

4.Assign a rank to each view defined as the number of views that are similar to it.

Algorithm for selecting optimal views

5.Sort all the views according to their ranks.

6.Make a sorted list L of all views. Each view will have a pointer to other views similar to it.

7.Start from the top of L and place the first view in the set C of characteristic views. Remove all views similar to the first view from L to obtain a reduced list.

8.Move down the reduced list L and repeat the procedure in 7 until the end of L is reached. Now we get the optimal views.

Multiview scoring method

Object recognition accuracy can be improved when information from multiple views is integrated.

- Get the scores of each query result.

$[S_{q1,d1},\ S_{q1,d2},\ S_{q1,d3,}\ \ldots\ldots\ldots\ S_{q1,dN}\ ]$

$[S_{q2,d1},\ S_{q2,d2},\ S_{q2,d3,}\ \ldots\ldots\ldots\ S_{q2,dN}\ ]$

…….

$[S_{qv,d1},\ S_{qv,d2,}\ S_{qv,d3}\ \ldots\ldots\ldots\ S_{qv,dN}\ ]$

- For each image in the database, add the scores of each query image to get a new score.
- Sort the score to get the ranked query results.

Some improvements

- If we can get some prior information about each view, we can assign different weights to different view to get better results.

- We can add weights to the score of the top 100 results of each query.

Some improvements

In the paper ***Less is More: Efficient 3-D Object Retrieval With Query View Selection*****,** it suggests a real-time user interactive scheme to retrieve 3-D objects.

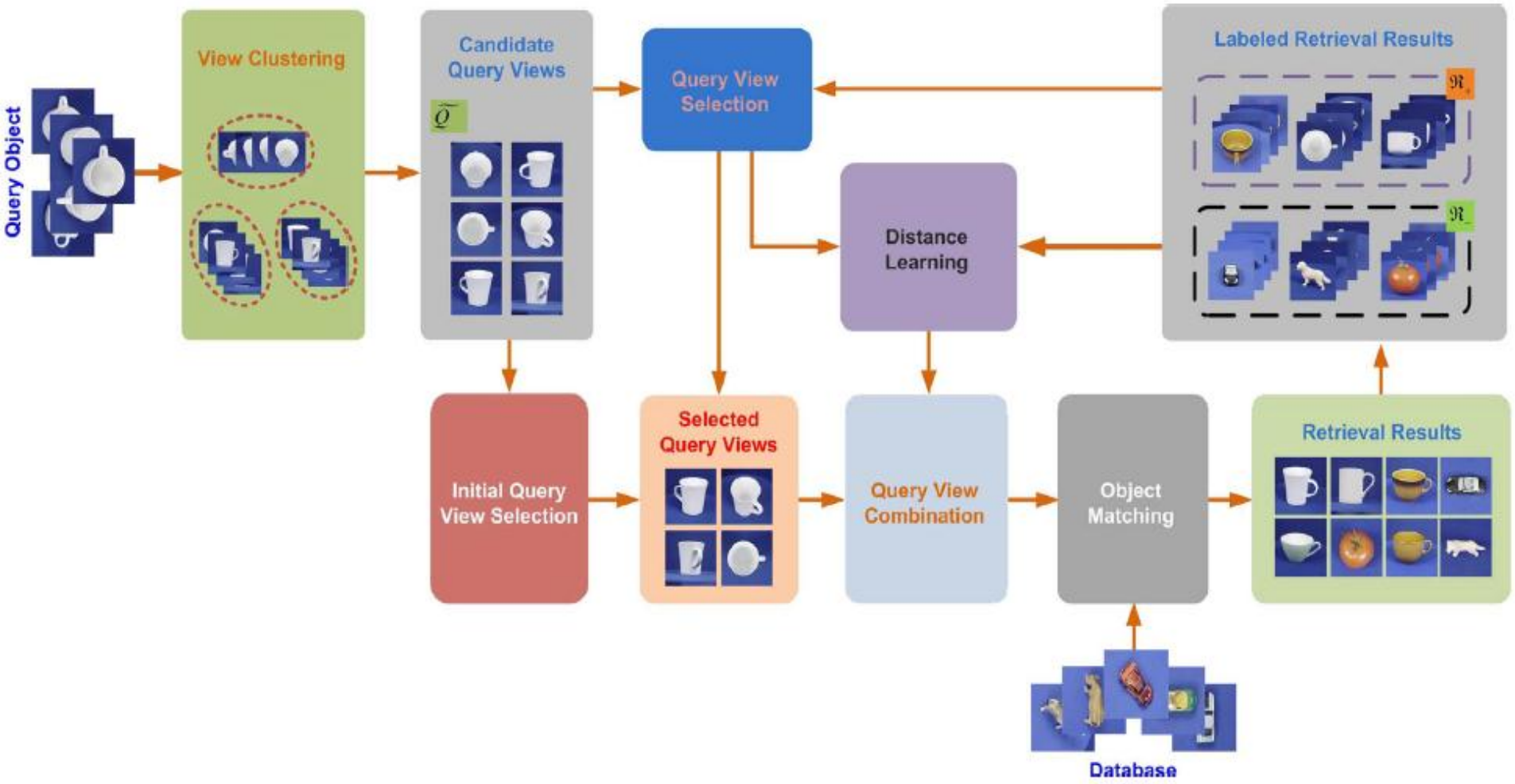


Fig. 1. Framework of the proposed interactive 3-D object retrieval algorithm.

Intelligent query

- The views selected in an unsupervised manner will not be informative enough.
- It incrementally selects a subset of query views based on the users' relevance feedback.

- First perform clustering to obtain several candidates.
- Then incrementally select query views for object Matching.
- In each round of relevance feedback, only add the query view that is judged to be the most informative one based on the labeling information.
- In addition, an efficient approach is proposed to learn a distance metric for the newly selected query view and the weights for combining all of the selected query views.

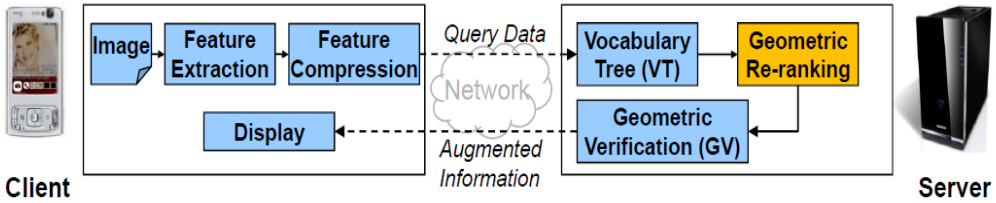
Geometric re-ranking

- Fast geometric re-ranking can shorten the list of candidates for the complex geometric Verification.
- A geometric similarity score of the query image and the candidate image is generated. This score can be computed efficiently by comparing the geometric properties of the VT visual word matches a location geometric similarity scoring method that is invariant to rotation, scale, and translation.
- Location geometric similarity scoring

$$S_{LDR} = \left\{ \log\left(\frac{dist(l_{q,i}, l_{q,m})}{dist(l_{d,j}, l_{d,n})} \right) \mid (i,j),(m,n) \in M \right\}$$

$$C_{LDR}(\alpha) = \sum_{z \in S_{LDR}} I\left(\frac{\alpha}{c} \leq z < \frac{\alpha+1}{c} \right)$$

$$Score_{LDR} = \max_{\alpha} C_{LDR}(\alpha)$$



Geometric verification

- The GV finds a coherent spatial pattern between features of the query image and the candidate database image to ensure that the match is plausible.
- The GV step rejects all matches with feature locations that cannot be plausibly explained by a change in viewing position.
- The geometric transform between query and database image is estimated using robust regression techniques such as RANSAC or the Hough transform.

Contents

Database

- Database for training Vocabulary Tree:

  Caltech256, 10000 images

- Database for recognition:

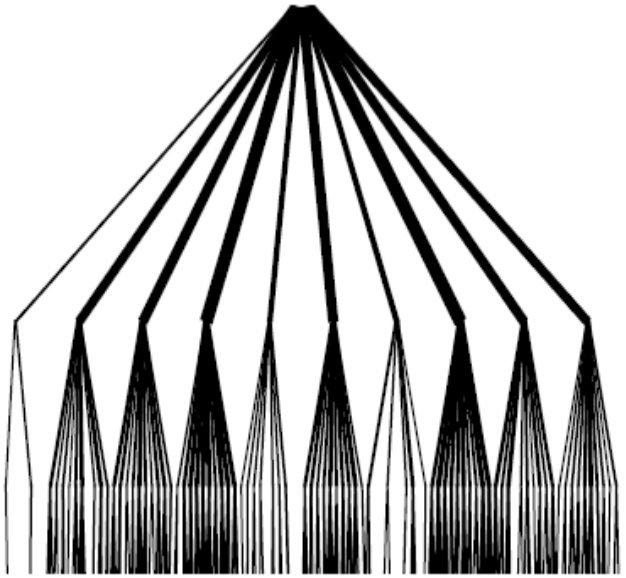
  Caltech256, 30 objects, 1500 images

Caltech-256

- Smallest category size now 80 images
- About 30K images
  - Harder
  - Not left-right aligned
  - No artifacts
  - Performance is halved
  - More categories
  - Performance are halved (even less)
  - New and larger clutter category

Vocabulary tree

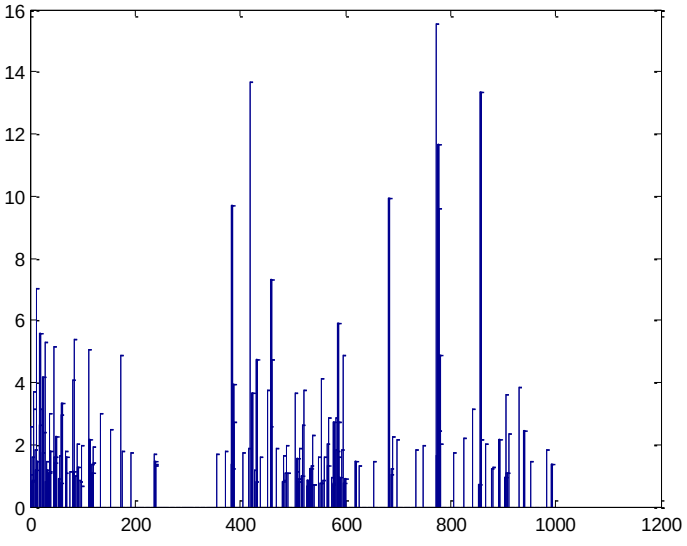
- Training images:10000 images
- SIFT feature: 4544348 sift feature
- Cluster: 10
- Depth: 4
- Visual words: 10000
- Iteration of k-means:50

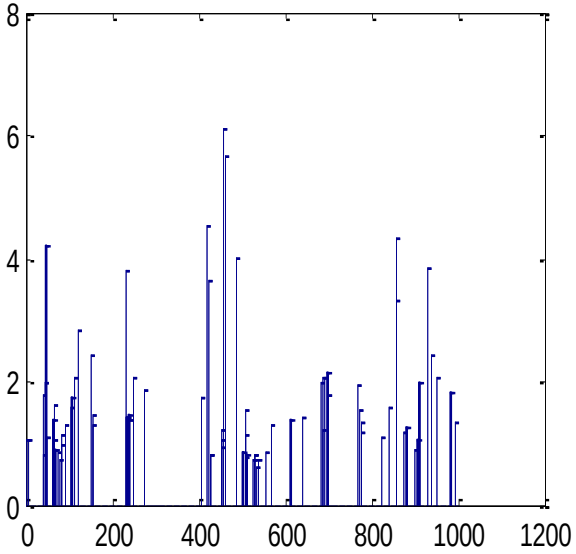


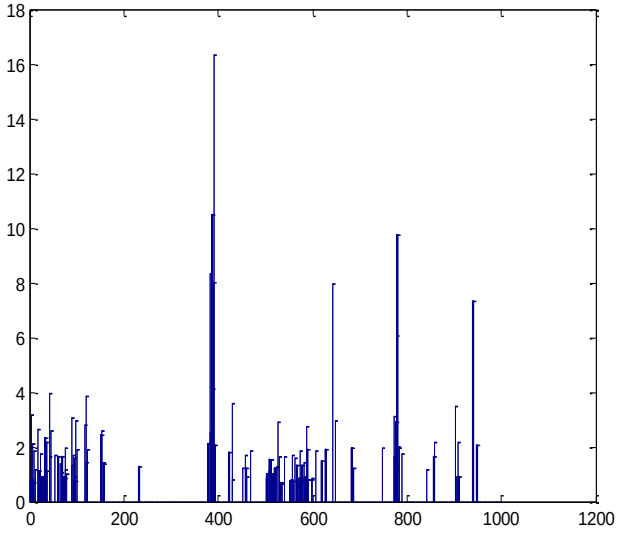
Time Complexity

- Extract SIFT feature(10000images): about 1 d
- Read SIFT feature from files: about 2 hours
- Building the vocabulary tree: about 1 hours
- Query in the database: about 10s

Distance measurement



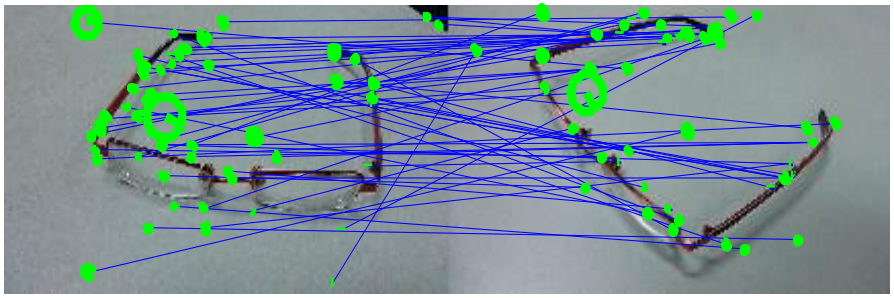


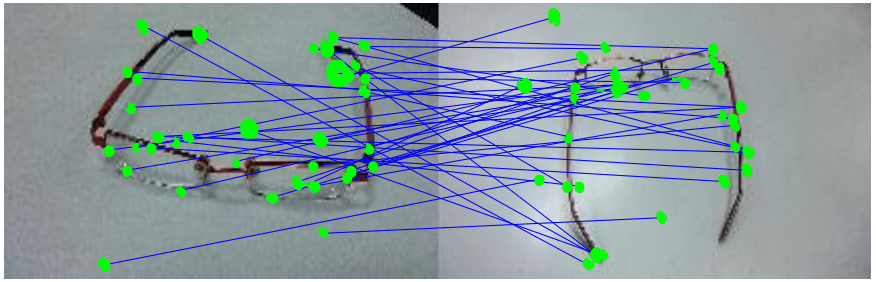


Cosine distance

1.0000 **0.4158** **0.3257**

Sift match

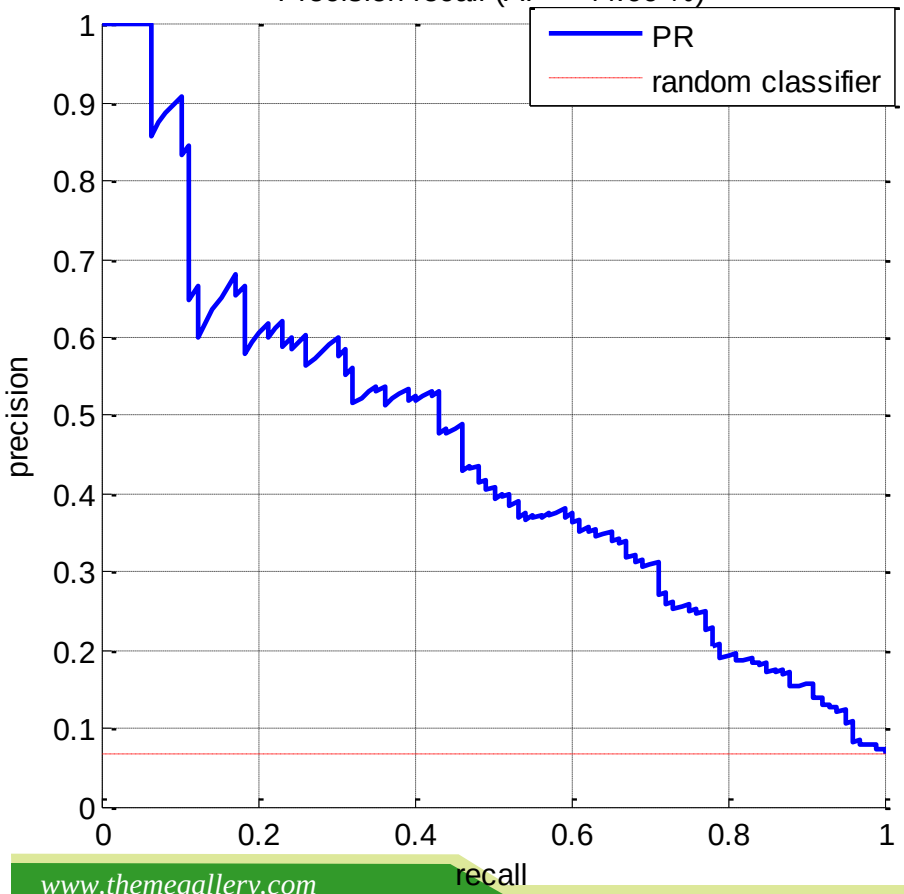


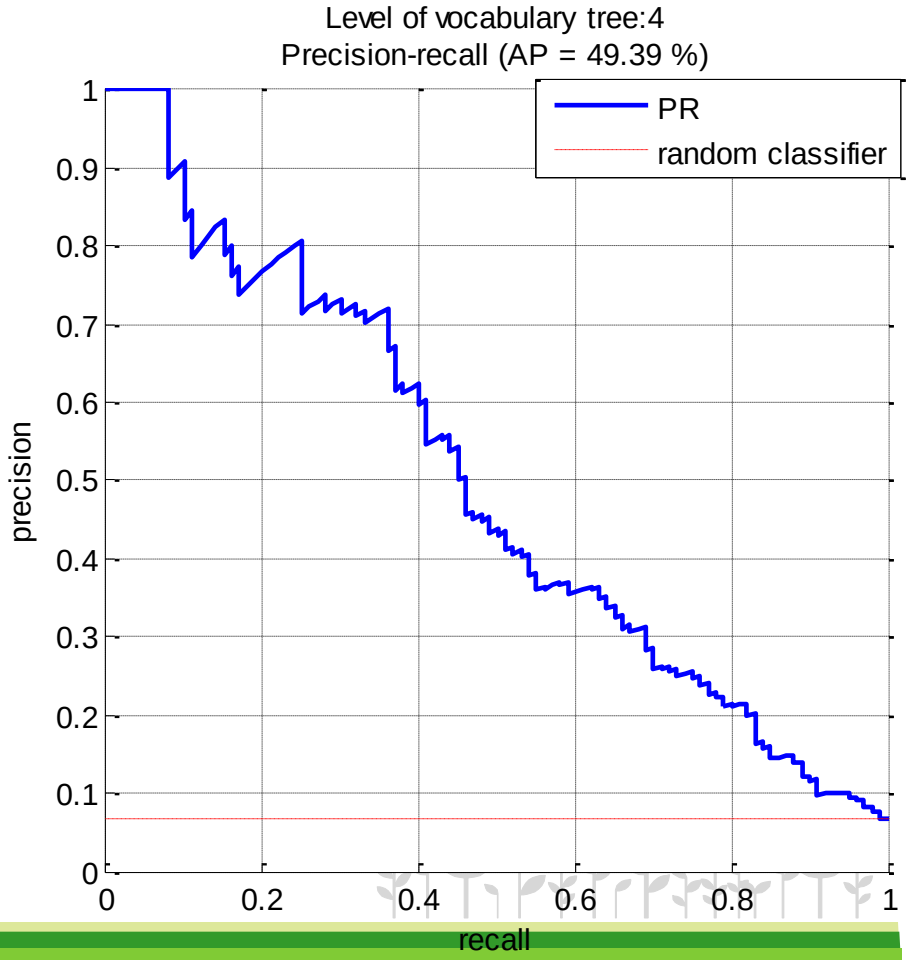


the number of sift of image1 is:1507
the number of sift of image2 is:2934
the number of matched sift is:58

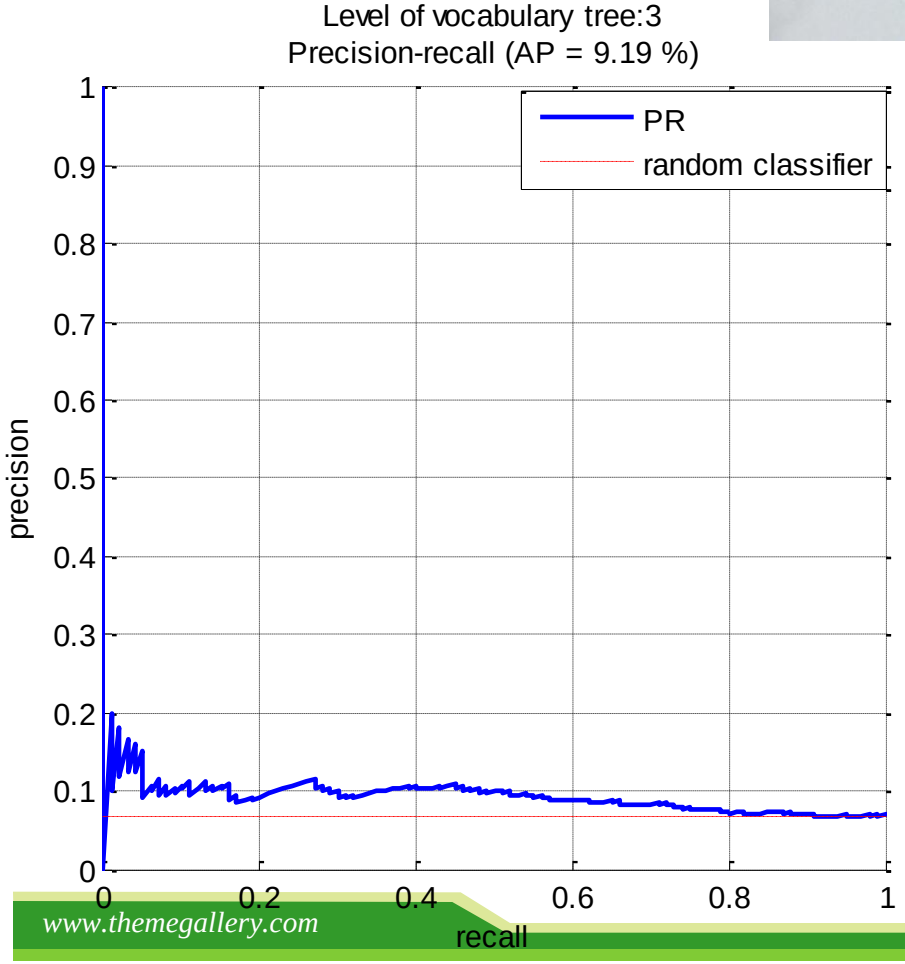
the number of sift of image1 is:1507
the number of sift of image2 is:2334
the number of matched sift is:39

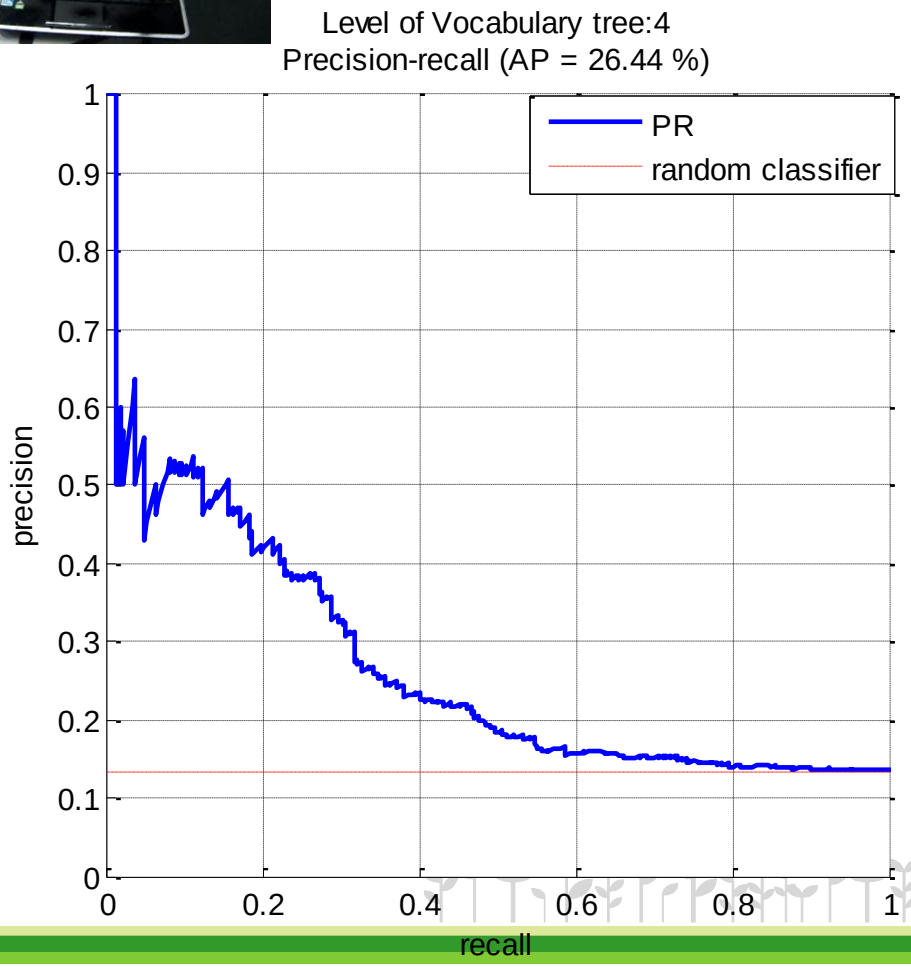
Depth of vocabulary tree





Depth of vocabulary tree



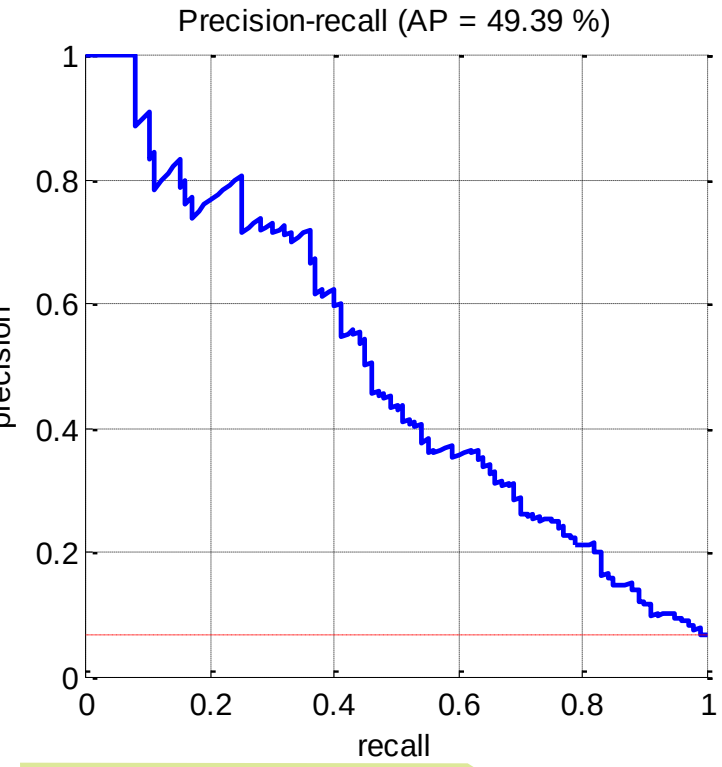


Depth of vocabulary tree

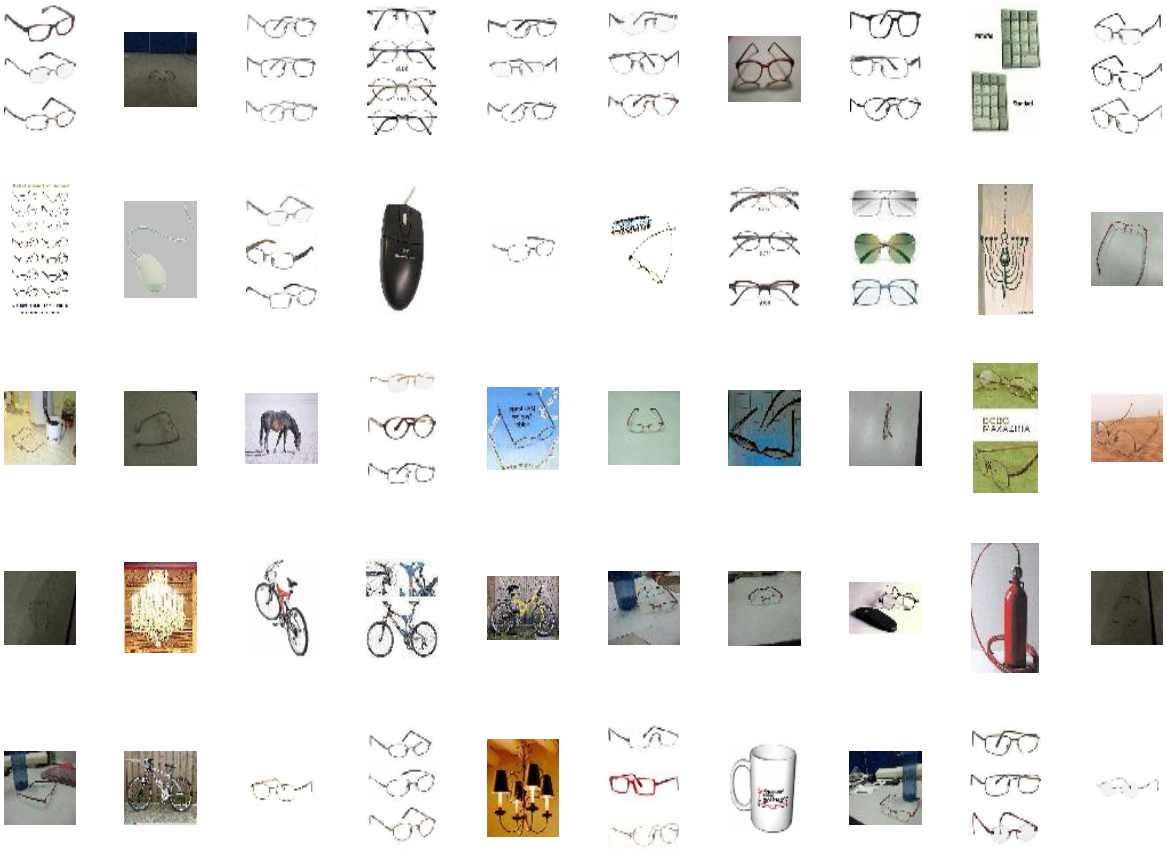
The larger vocabulary tree(the large number of leaf nodes), the better retrieval quality.

In principle, the vocabulary size must eventually grow too large, so that the variability and noise in the descriptor vectors frequently move the descriptor vectors between different quantization cells.

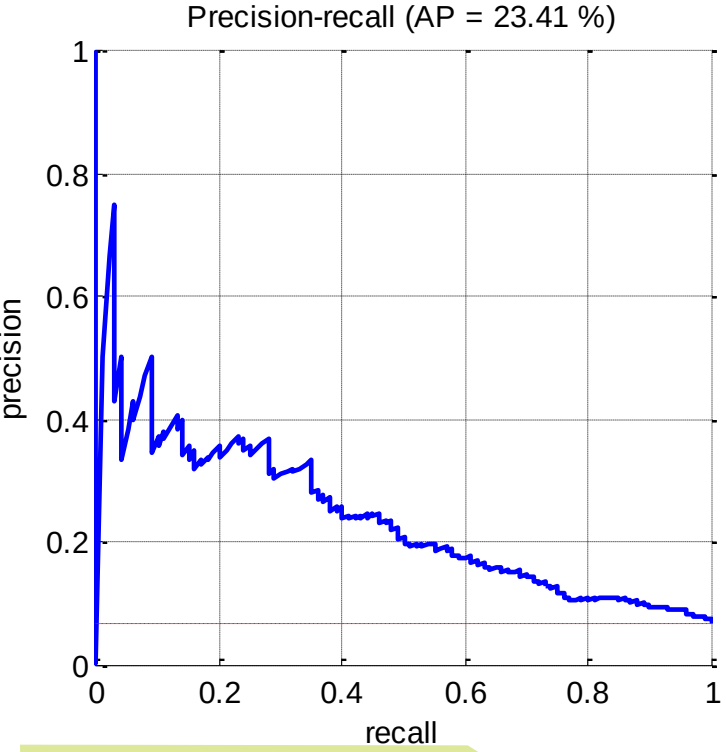
Single view search (view 1)



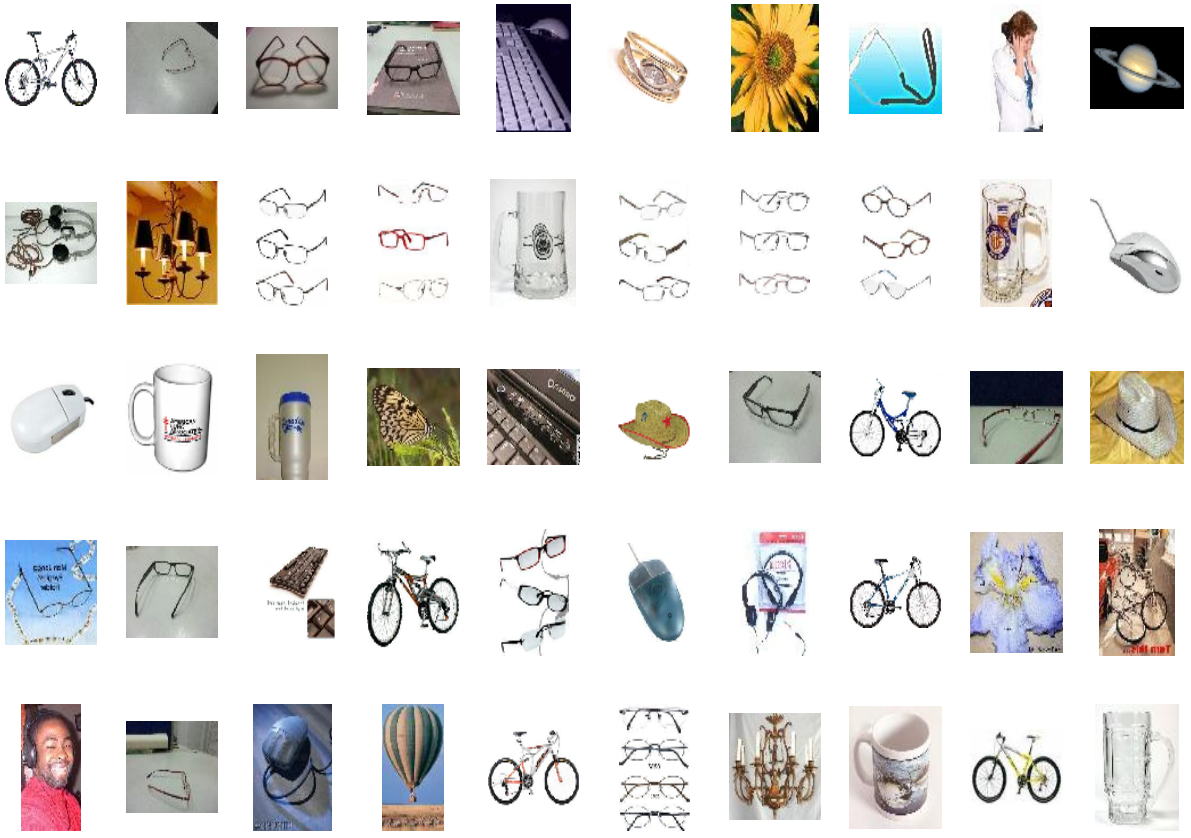
Top 50 results



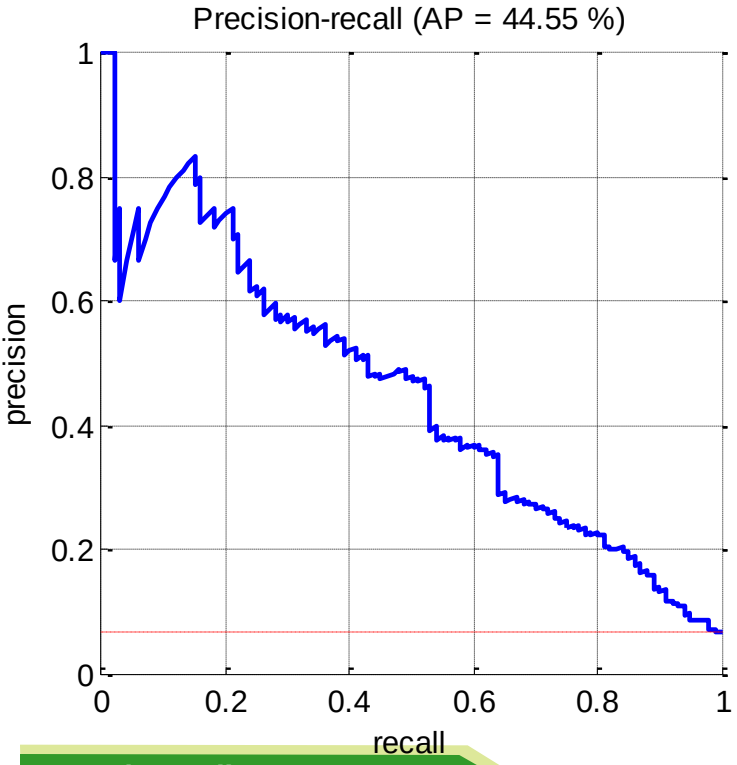
Single view search(view 2)



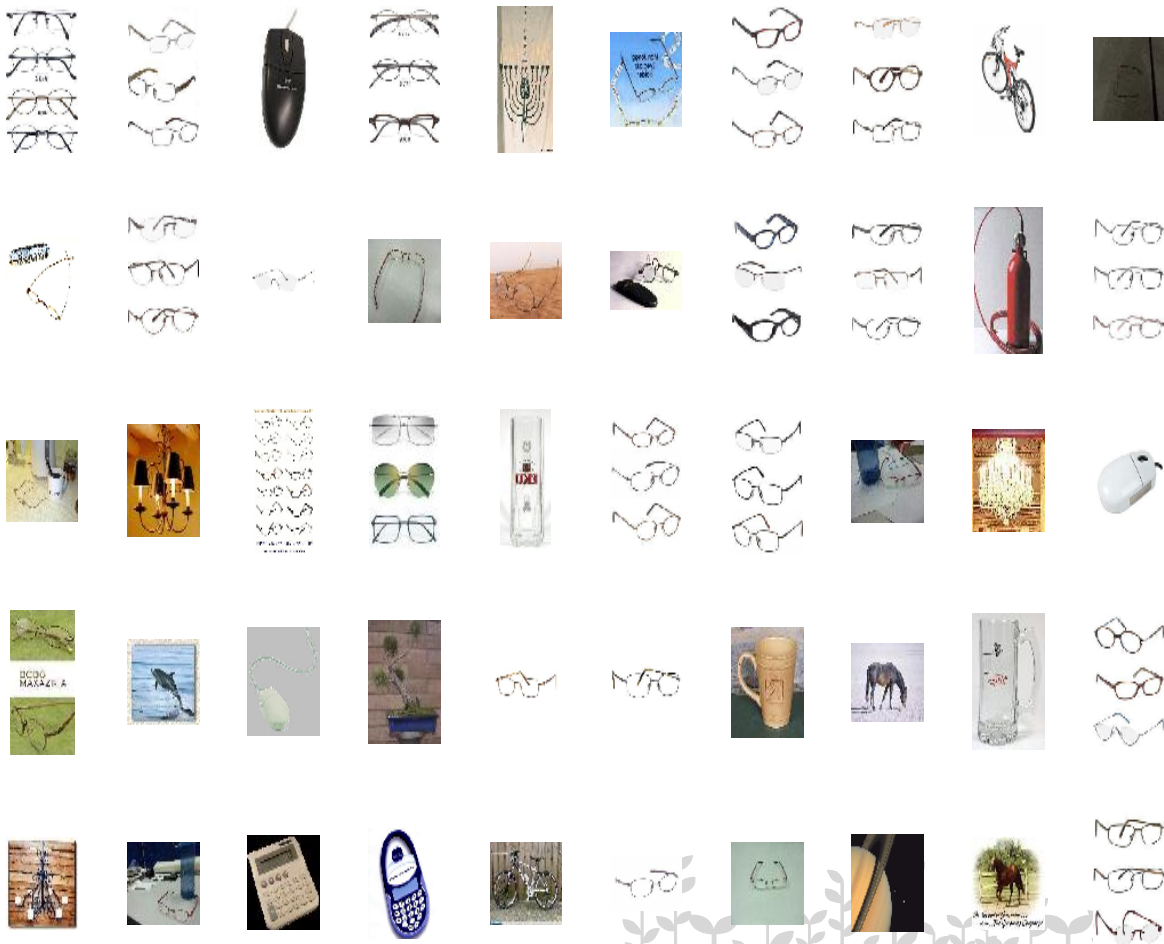
Top 50 results



Single view search (view 3)

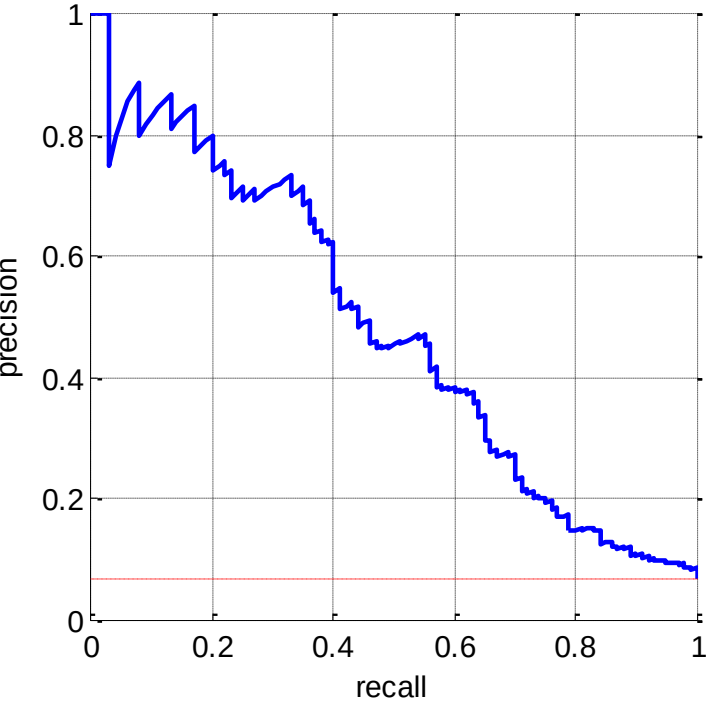


Top 50 results

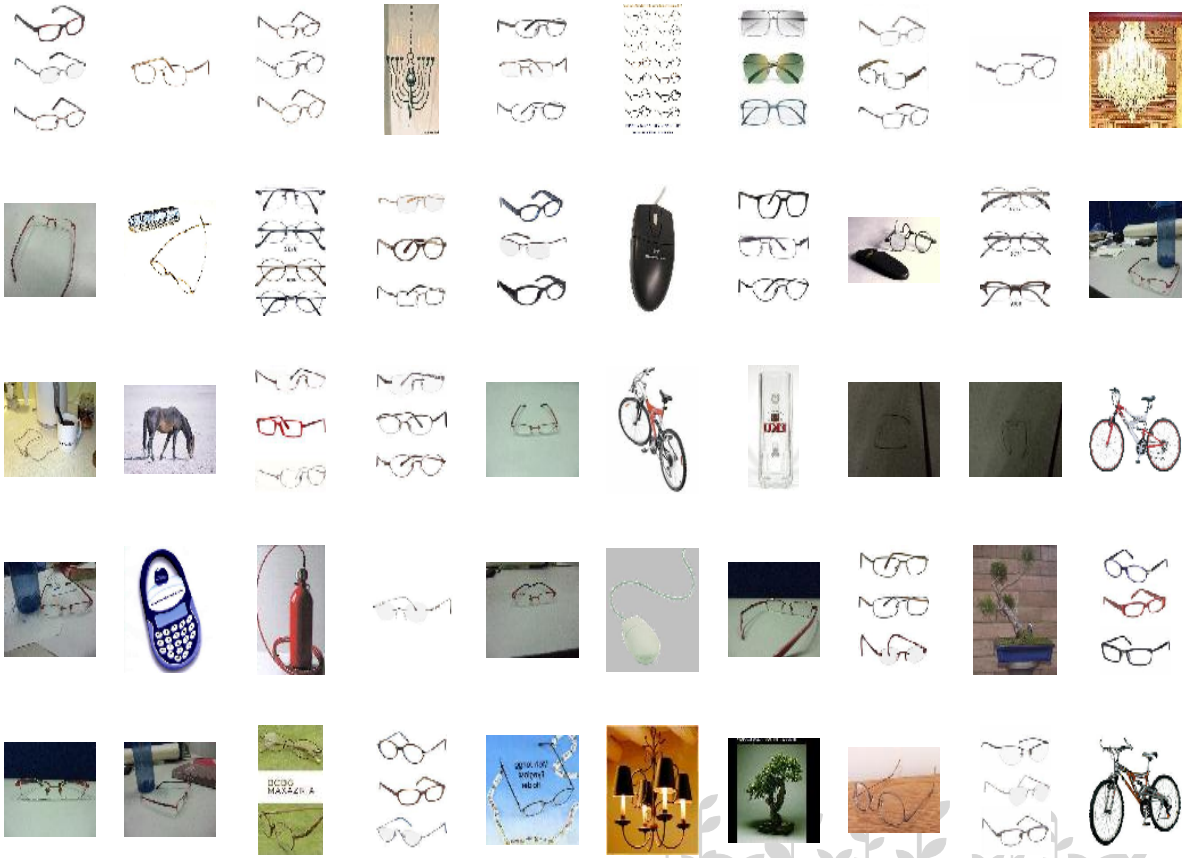


Single view search (view4)

Precision-recall (AP = 47.66 %)

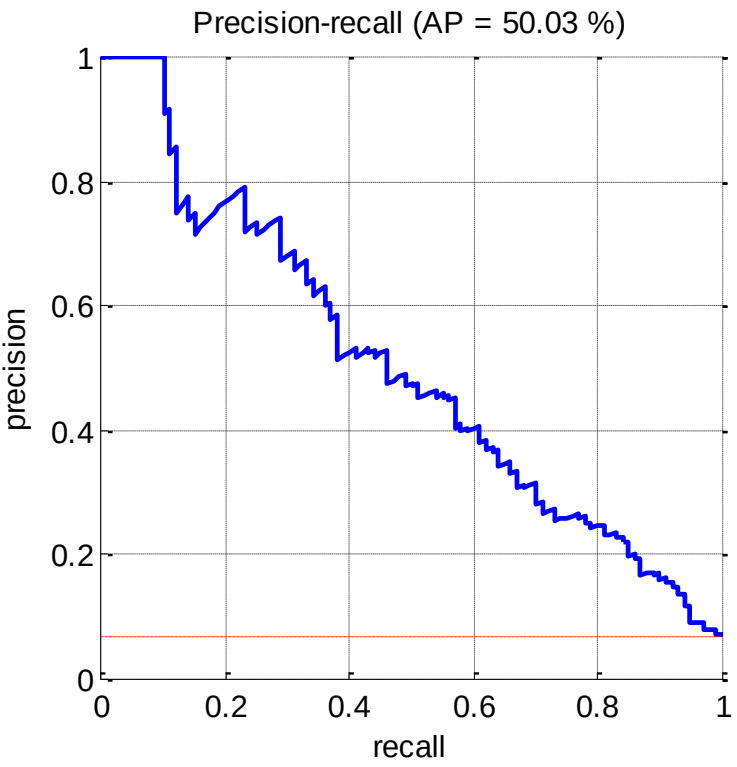


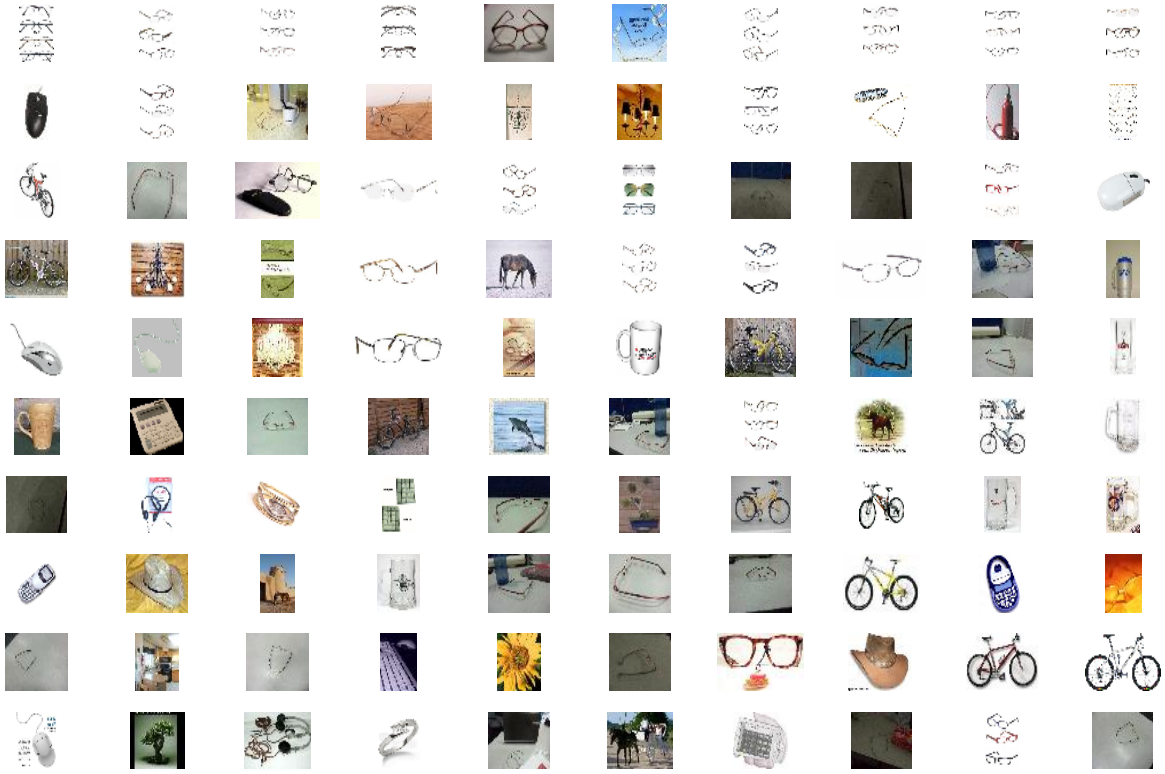
Top 50 results



Multiple views search

Top 100 results

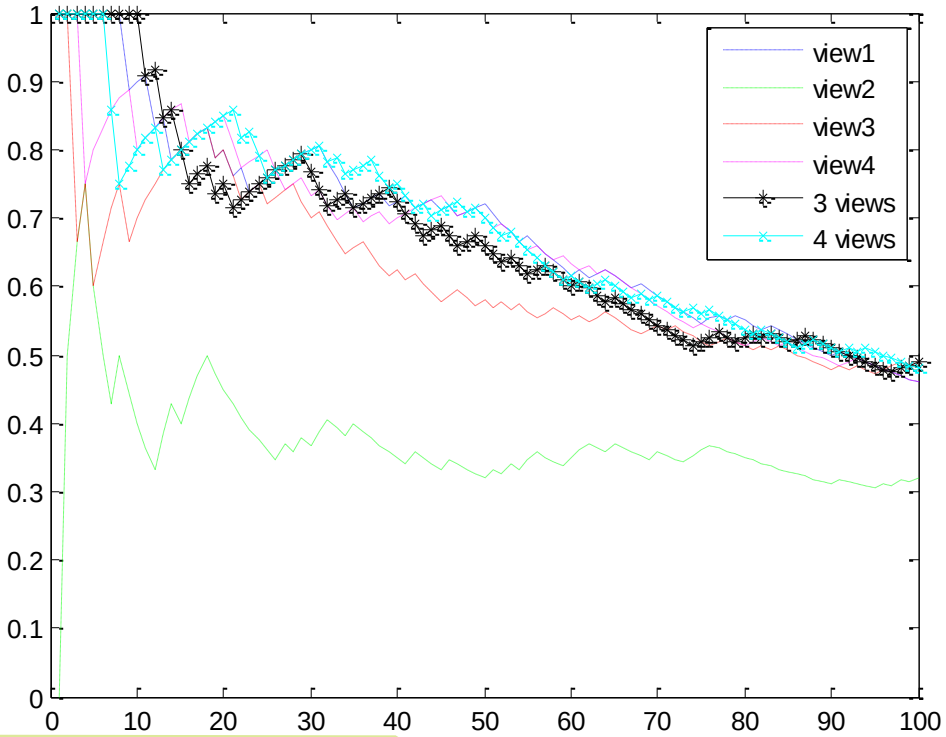




Single view & multiple view search

Average precision(%)

Single view1	Single view2	Single view3	Single view4	3 views	3 views (weight)	4 views	4 views (weight)
49.39	23.41	44.55	47.66	49.41	50.03	50.39	50.61



Curves showing percentage (y-axis) of the query images that make it into the top x (x-axis) frames of the query for a 1500 image database. The curves are shown up to 100 images in the database .

Multiple view & single view

- The average precision is better when using multiple views.

- It shows that when using multiview search, correct images from the database make it to the very top of the query better.

Voting scheme

- To do object recognition, we can first label the database images and then cast a vote on the top 100 results.
- The vote of the 30 objects:

1	5	2	0	3	0	4	4	2	5
1	2	0	1	0	32	2	3	3	0
0	0	11	1	0	1	0	0	1	16

- The 16th (in caltech256)and 30th (photos by myself)objects are both eyeglasses.

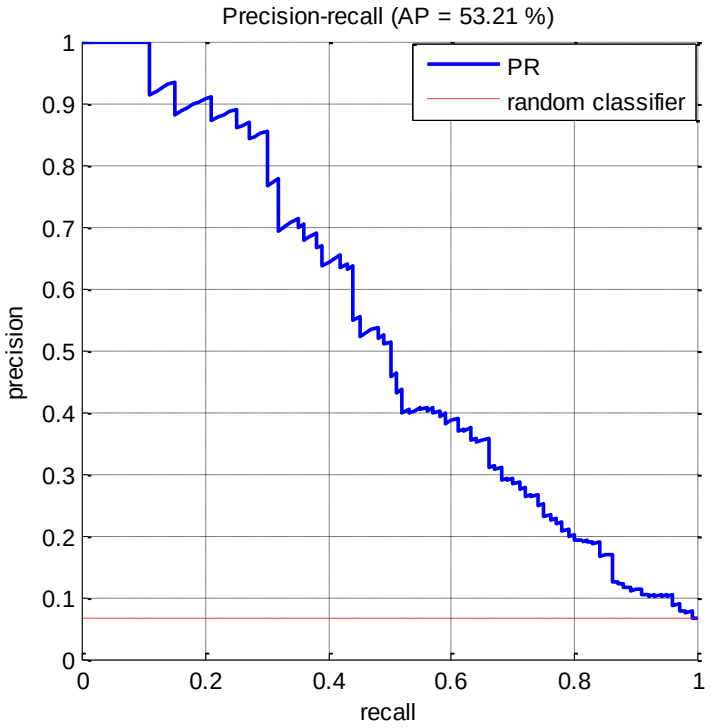
Other ideas---1

- Max pooling

In the paper *Analysis of Feature Learning and Feature Pooling for Image Recognition*, the author indicated that max pooling should be preferred over average pooling when features have a low probability of being active (e.g., with large codebooks) and the pool cardinality is large enough.

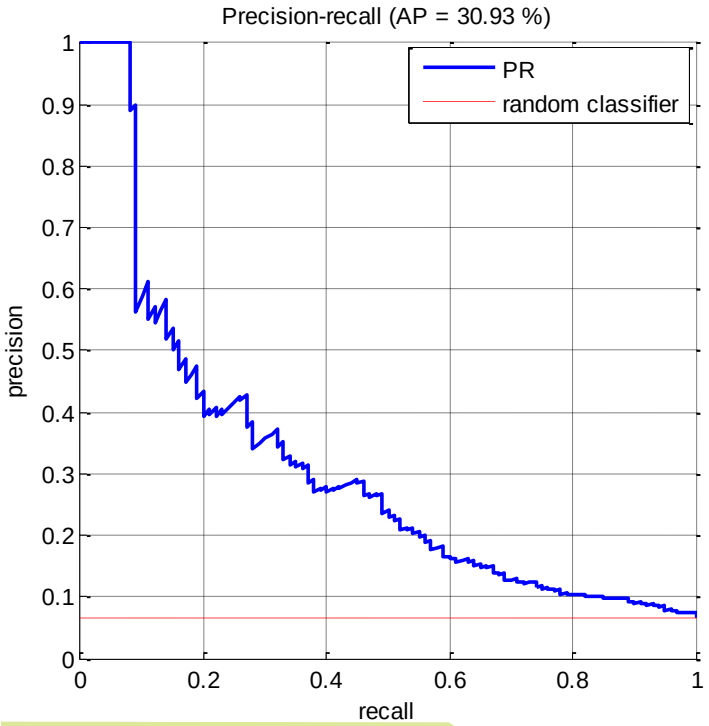
- If we do max pooling instead of TF (Term Frequency) , what's the result?

Single view- view1

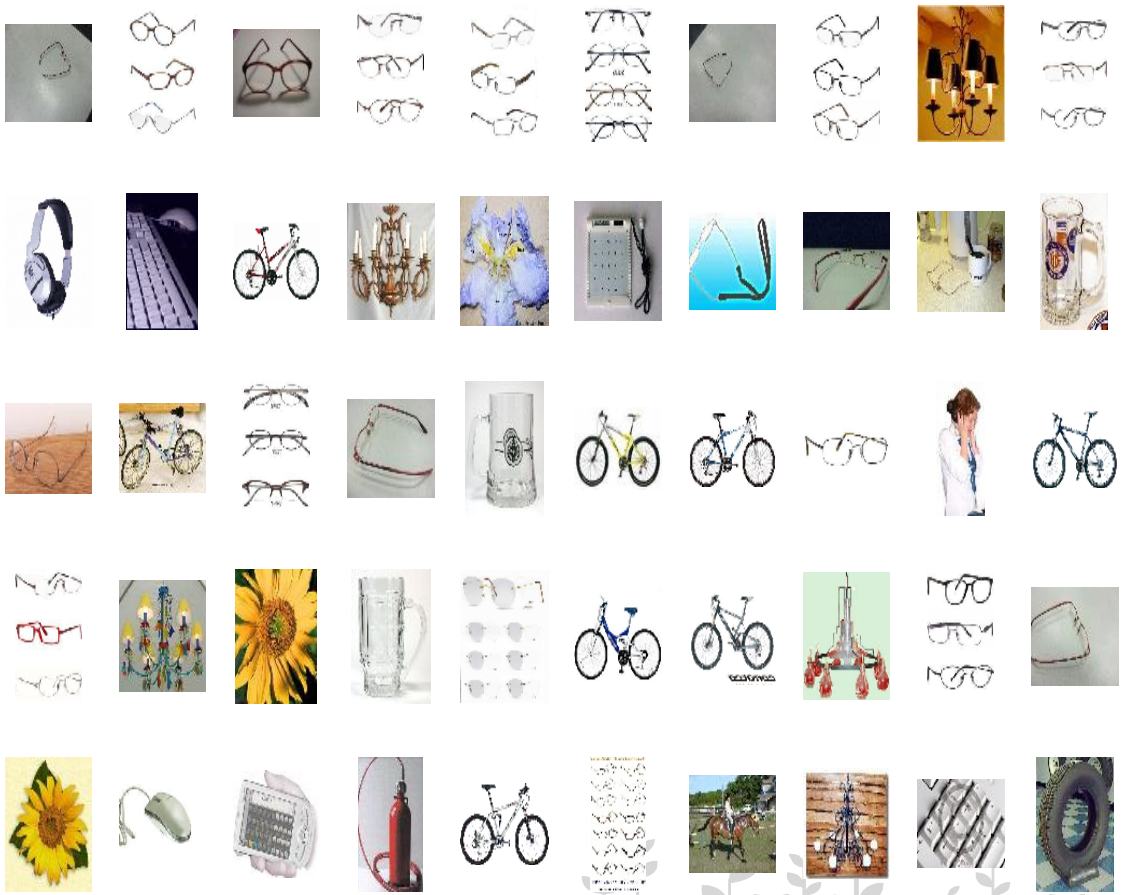


Top 50 results

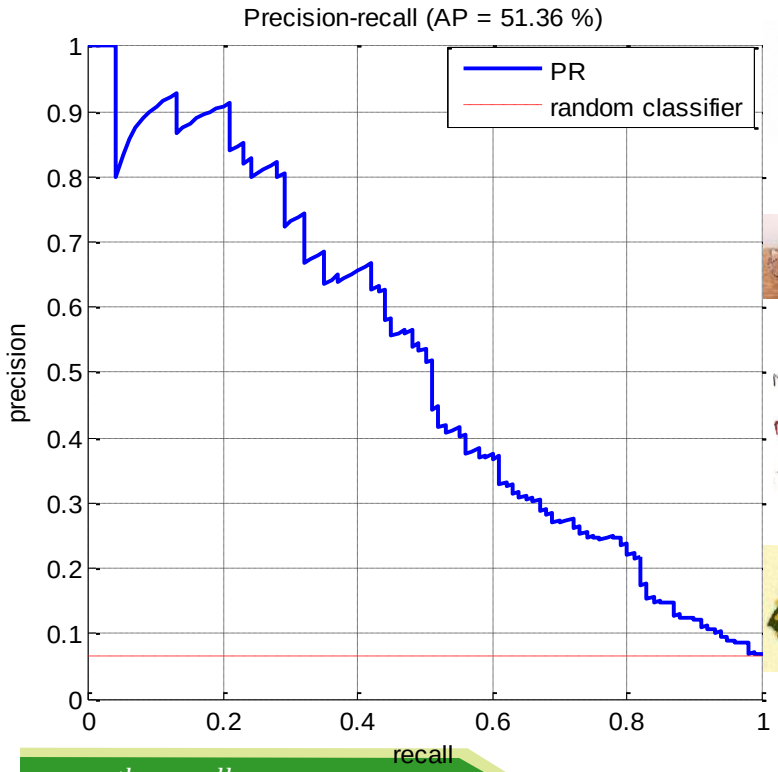
Single view- view2



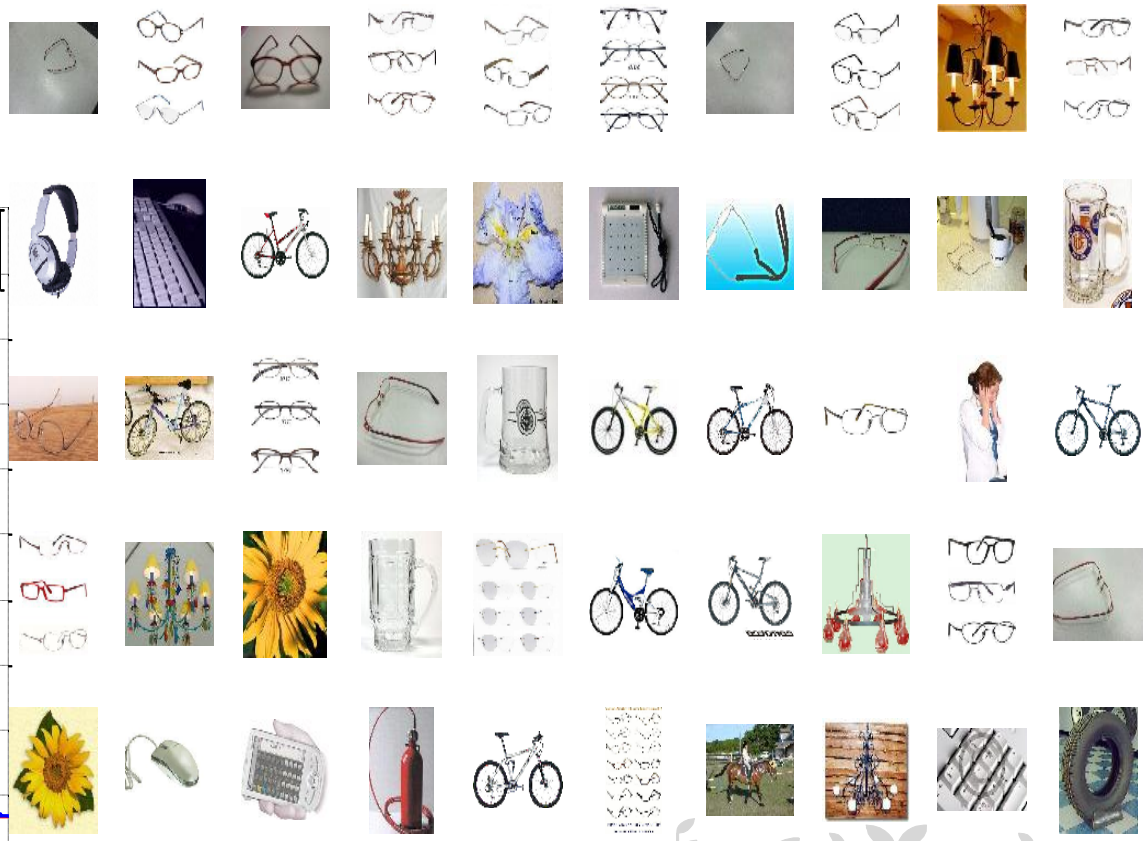
Top 50 results



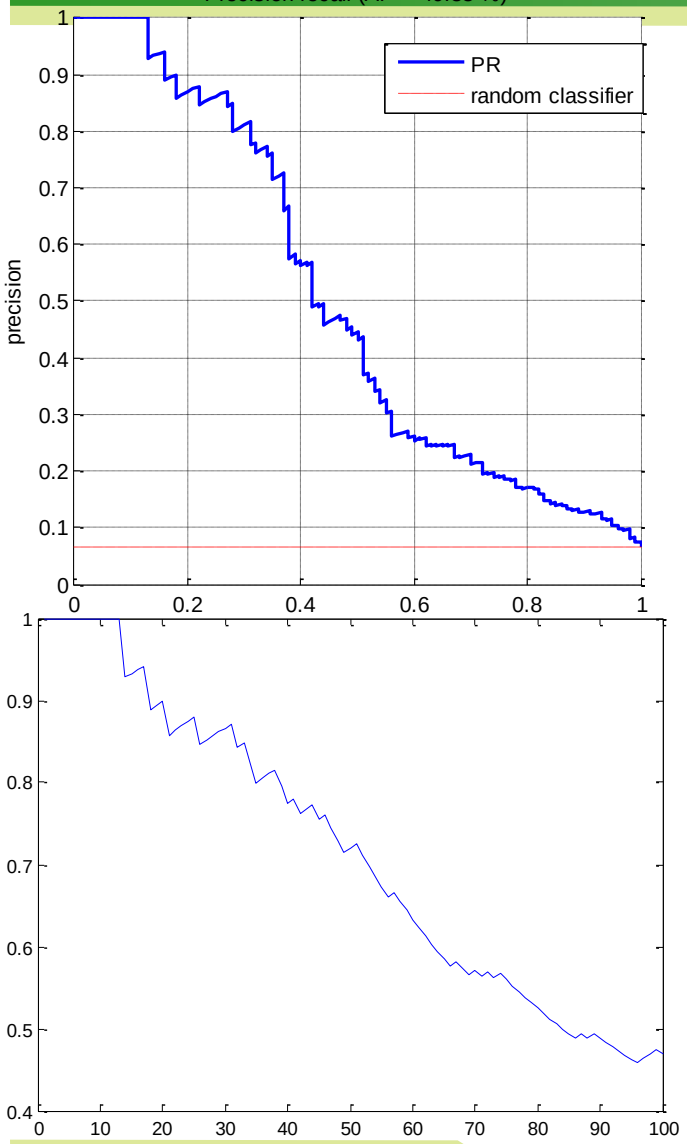
Single view- view3



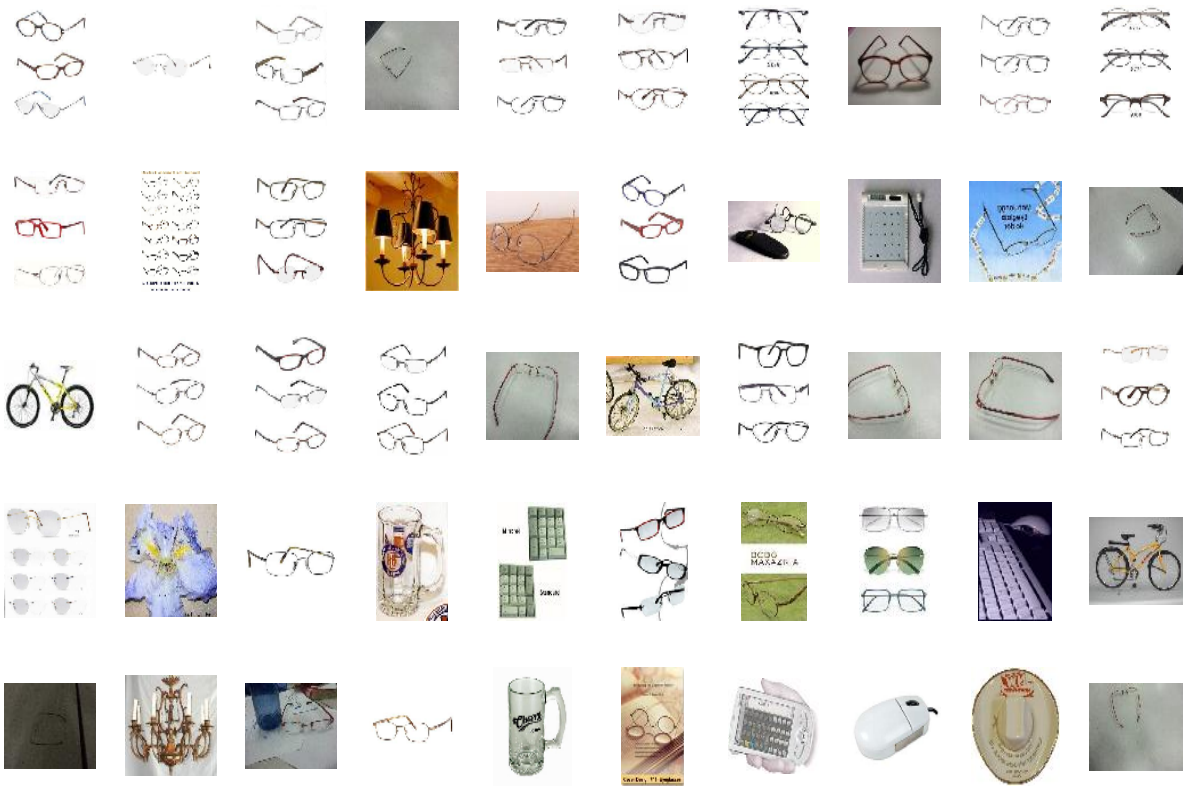
Top 50 results



Max pooling using multiple views



Top 50 results



Max pooling Results

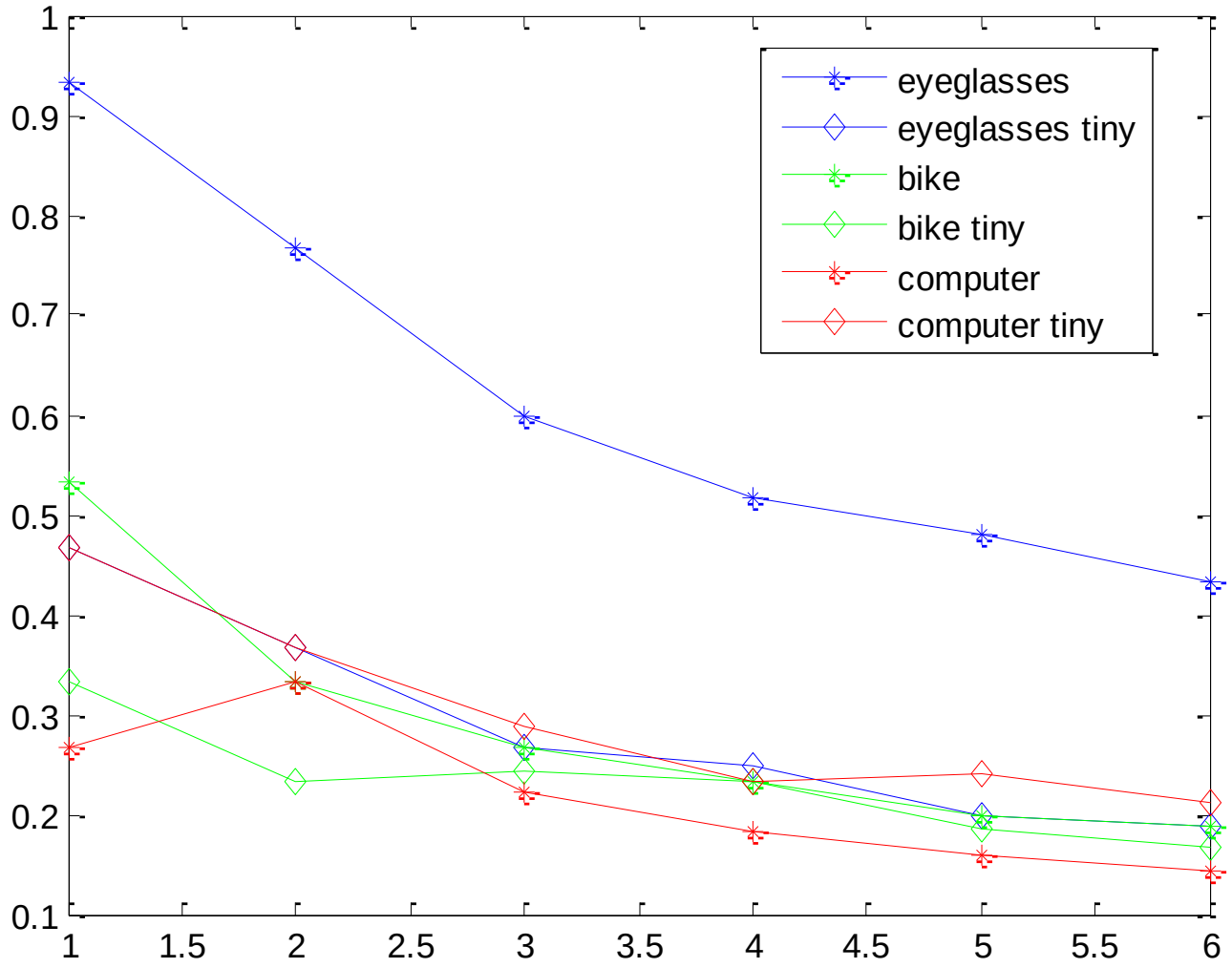
- Compared with the result of TF-IDF scoring method, the search result of Max pooling is better.

- Reason: The codebook is much larger (10000 visual words) than the number of sift feature, and the features are not active with the codebook, so max pooling works well.

Other ideas---2

- In the paper *80 million tiny images: a large dataset for non-parametric object and scene recognition,* the author indicated that we can achieve good recognition result even when we lower the resolution of the images in the database.
- The tiny image can reduce some details in the image, which may be good for the search for objects of same class.

Original database & tiny database



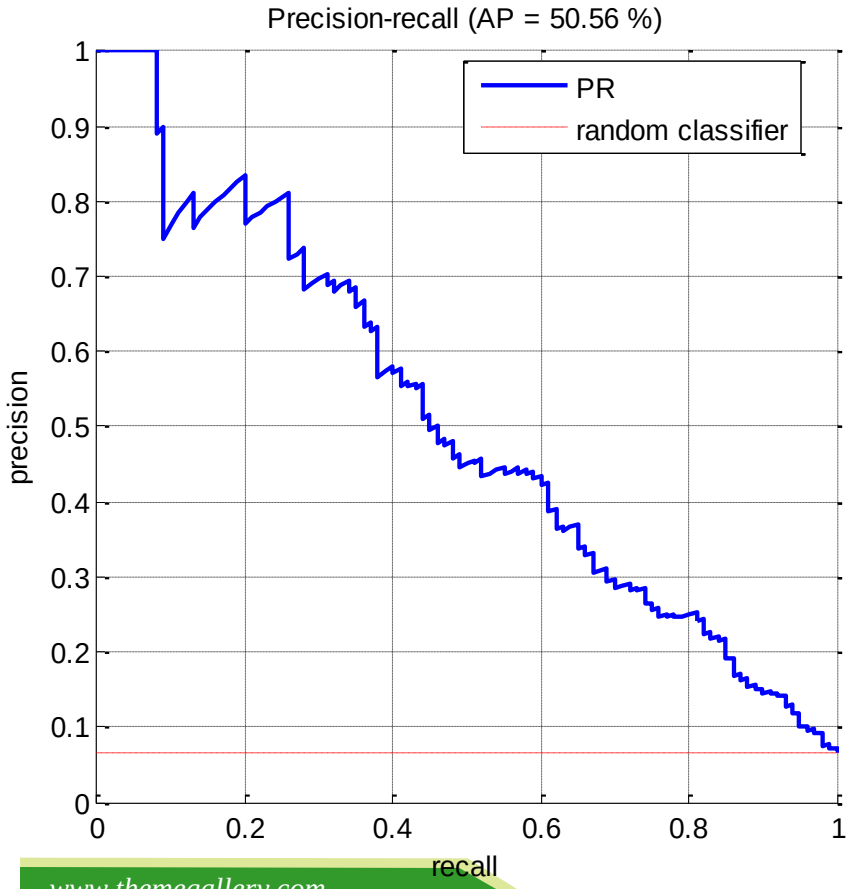
Curves showing percentage (y-axis) of the query images that make it into the top x percent (x-axis) frames of the query for a 1500 image database. The curves are shown up to 6% images in the database.

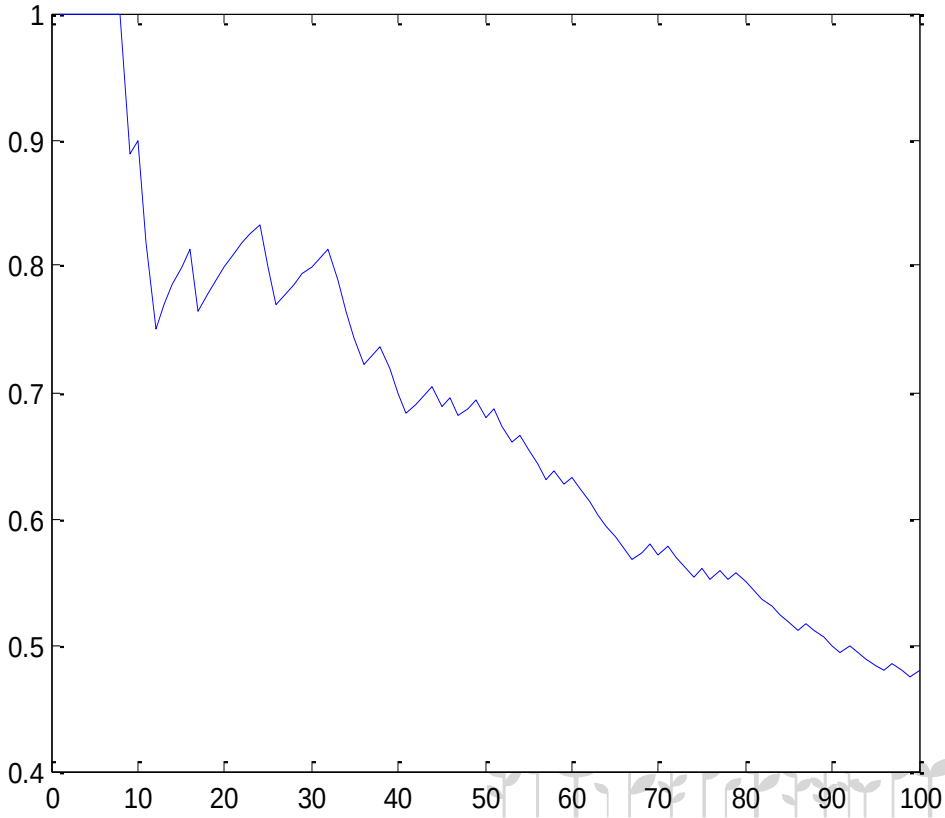
Original database & tiny database

- In the experiment, I do visual search based on the database of the original images and tiny images.
- It show that for object with complex structure, such as computer, the tiny image database can achieve better search result.
- While for object with simple structure, the result isn’t improved.
- Actually, the sampling process is in the SIFT feature extract process (DOG). So maybe we can make some change in the sift parameter.

Other ideas ---3

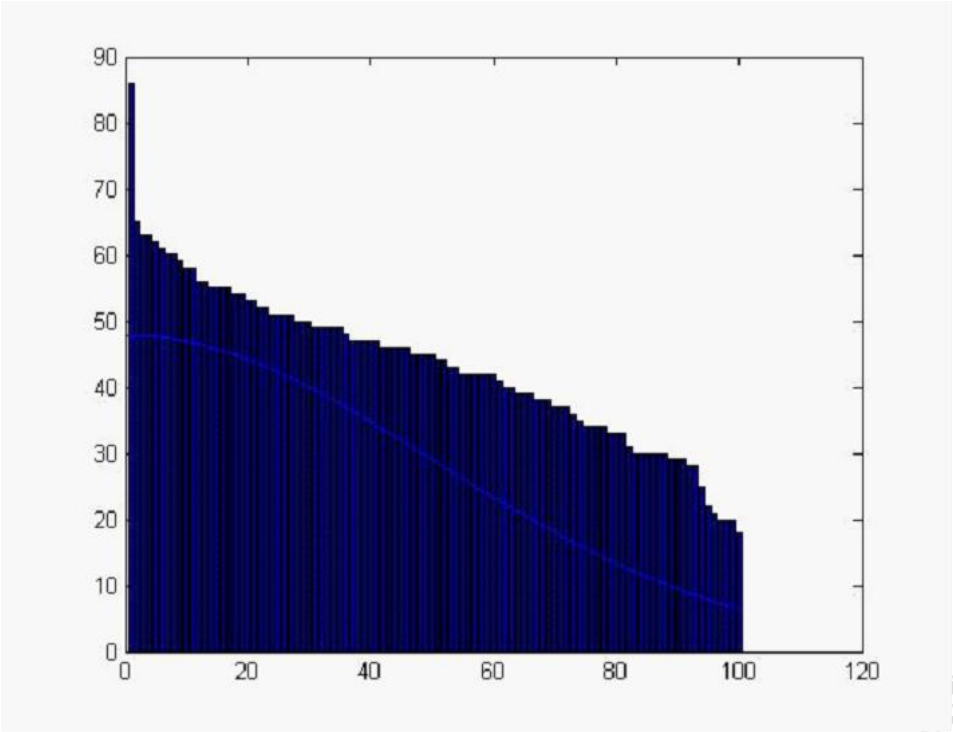
- Using all the features of the multiple view images as a whole, and then query in the database.





Other ideas-4

- There is feature redundancy across different views. We can build a model to select features to reduce the redundancy based on Gaussian model or Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy.



Suitability for visual search on CalTech256

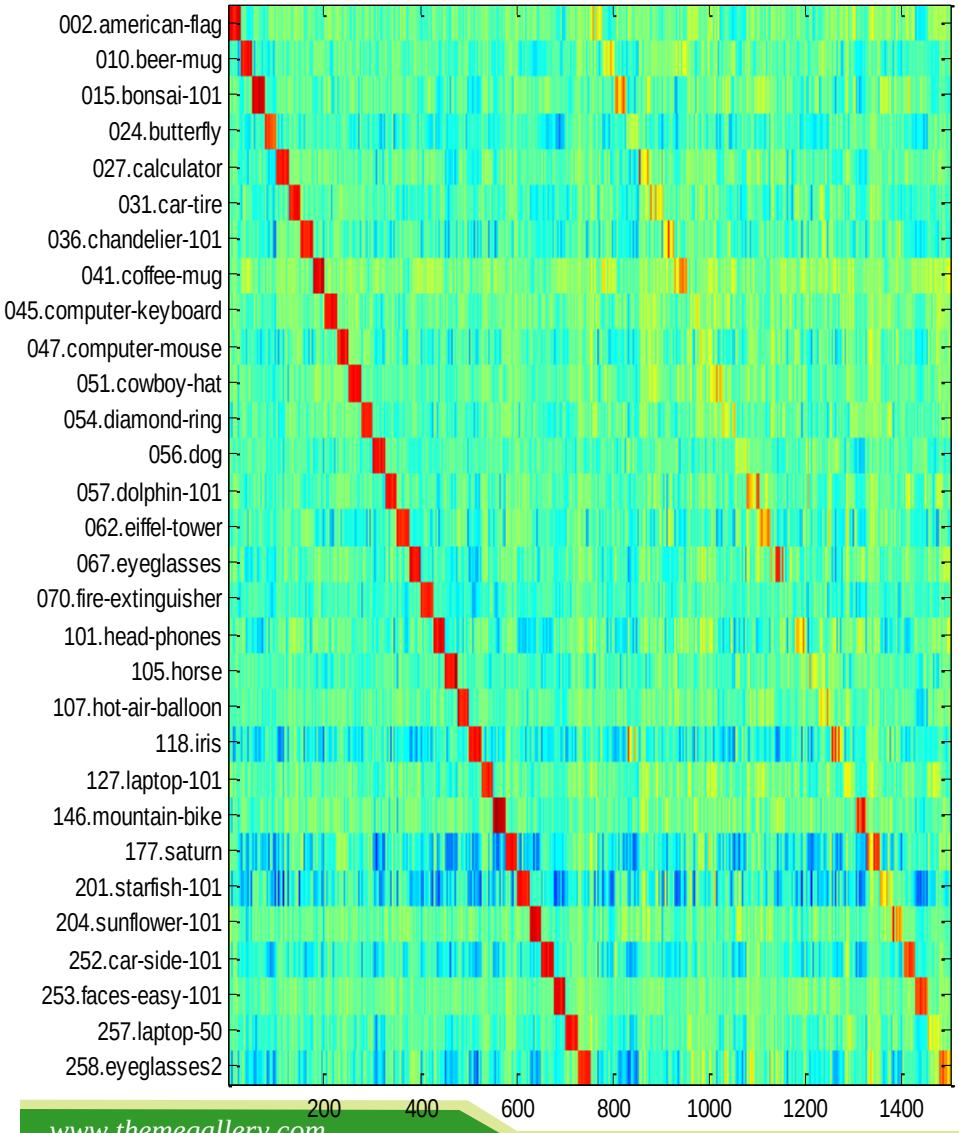
- Actually, CalTech256 is not suitable for visual search. The difficulty lies in capturing the variability of appearance and shape of different objects belonging to the same class, while avoiding confusing objects from different classes. So I use the simple object—eyeglasses--as query object.

- To do object recognition in CalTech256, it's better to use supervised method, such as SVM classifier.

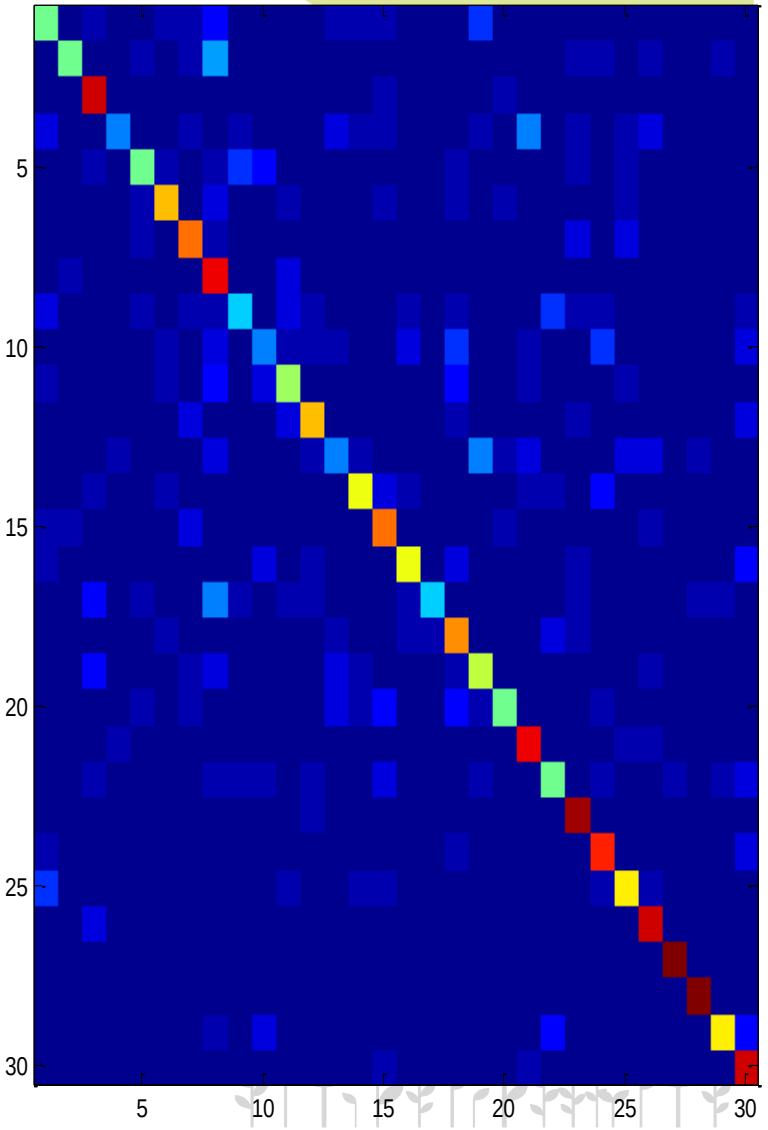
- For multiview object recognition, we can first classifier all the single view image and then cast a vote on all the results to get the final result.

SVM classifier

Scores



Confusion matrix (64.13 % accuracy)

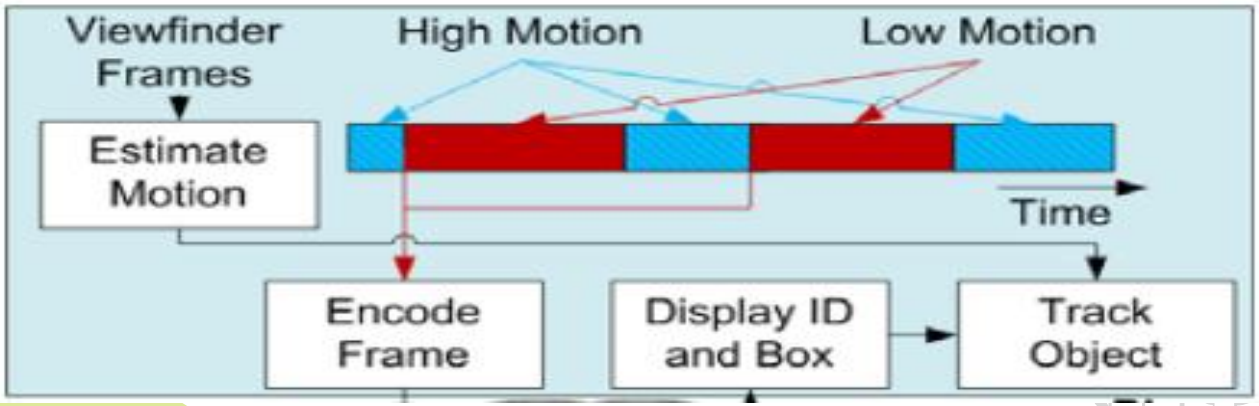


Contents

For mobile visual search

- 1.Streaming recognition

An efficient motion estimator is used to determine camera movement, which enables to selectively send query data to the server only when needed and to track an object after initial recognition.

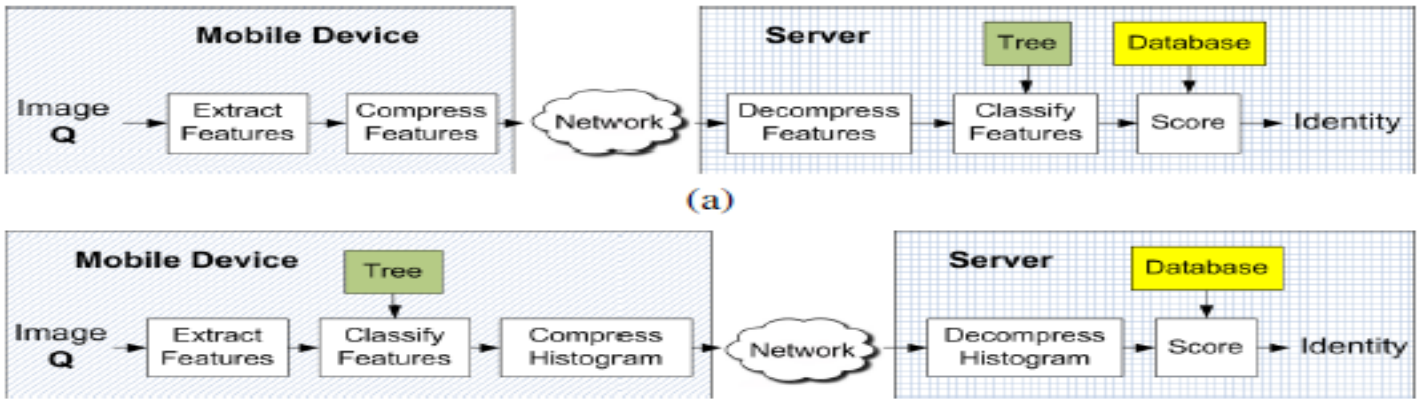


For mobile visual search

- 2.Use SURF or CHoG feature to speed up feature extraction and reduce query latency in mobile image retrieval systems.
- 3. Use BFOS Algorithm to prune vocabulary tree until it reaches the subtree with the fewest number of leaves that achieves a given rate distortion trade-off.
- 4. Use a soft binning scheme or sparse coding to mitigate the effect of quantization errors for a large VT.

For Mobile visual search

- 5. Use run-length encoding algorithm to encode the tree histogram.
- 6. Inverted file compression to reduce the memory storage.
- 7. Build Multiview Vocabulary Trees for Severe Perspective Queries.



Reference

- [1].D. Nister and H. Stewenius. **Scalable recognition with a vocabulary tree**. In c*onference on Computer Vision and Pattern Recognition*, New York, NY, USA, June 2006.
- [2]J. Sivic and A. Zisserman**, "Video google: a text retrieval approach to object matching in videos,"** in International Conference on Computer Vision, 2003, vol. 2, pp. 1470–1477.
- [3]Girod, B., Chandrasekhar, V., Chen, D. M., Cheung, N. M.,Grzeszczuk, R., Reznik, Y., Takacs, G., Tsai, S. S., & Vedantham,R**. Mobile visual search**. IEEE signal processing magazine,2010.
- [4] Yuriy Reznik, Vijay Chandrasekhar, Gabriel Takacs, David Chen, Sam S. Tsai, and Bernd Girod. **Fast quantization and matching of histogram-based image features**.In Proceedings of Applications of Digital Image Processing XXXIII, SPIE vol. 7798, San Diego, CA, August 2010.
- [5] D. Lowe. **Distinctive image features from scale-invariant keypoints**. *International Journal of Computer Vision*,November 2004.
- [6]Vijay Chandrasekhar, David M. Chen, Zhi Li, Gabriel Takacs, Sam S. Tsai, Radek Grzeszczuk, and Bernd Girod .**Low-Rate Image Retrieval with Tree Histogram Coding** International Mobile Multimedia Communications Conference, MobiMedia, London, U.K., September 2009.
- [7]Yue Gao, Meng Wang, Zheng-Jun Zha, Qi Tian,Qionghai Dai, and Naiyao Zhang**. Less is More: Efficient 3-D Object Retrieval With Query View Selection**. IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 13, NO. 5, OCTOBER 2011.

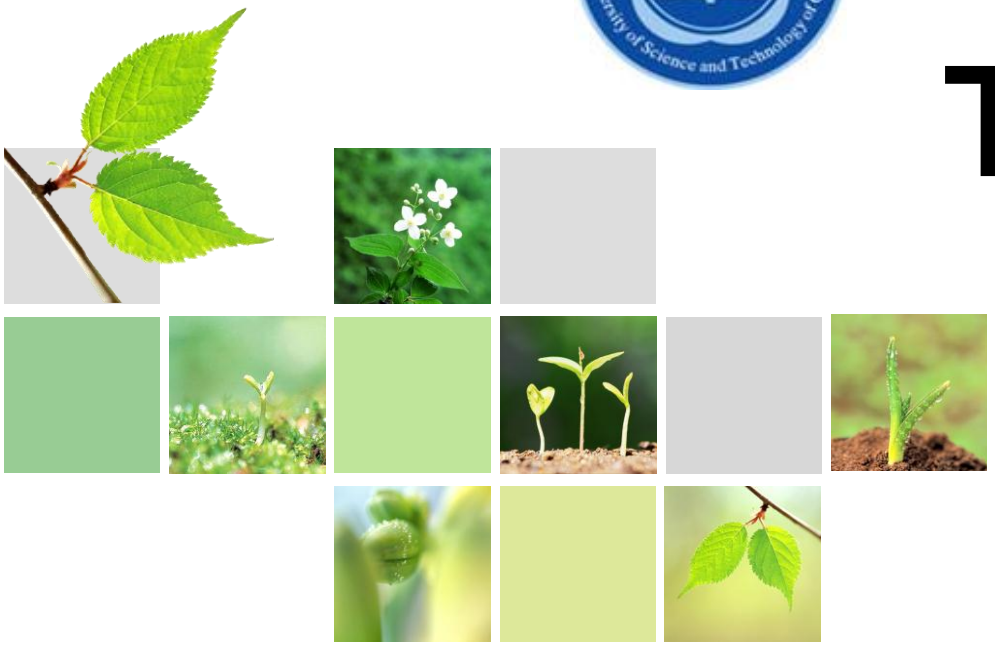
Reference

- [8]. S. S. Tsai, D. M. Chen, G. Takacs, V. Chandrasekhar,R. Vedantham, R. Grzeszczuk, and B. Girod**. Fast geometric re-ranking for image-based retrieval**. In *International Conference on Image Processing*, Hong Kong,September 2010.
- [9]J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "**Lost in quantization—Improving particular object retrieval in large scale image databases**," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Anchorage,AK, June 2008
- [10]J. Yang, K. Yu, Y. Gong, and T. Huang. **Linear spatial pyramid matching using sparse coding for image classification**. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.
- [11] R. Fergus, L. Fei-Fei, P. Perona and A. Zisserman. **Learning Object Categories from Google's Image Search.** *IEEE Inter. Conf. Computer Vision.* 2005.
- [12]D. M. Chen, S. S. Tsai, V. Chandrasekhar, G. Takacs,R. Vedantham, R. Grzeszczuk, and B. Girod**. Inverted index compression for scalable image matching**. In *Data Compression Conference*, Snowbird, UT, USA, 2010.
- [13]AY Yang, S Maji, CM Christoudias, T Darrel. **Multiple-View Object Recognition in Smart Camera Network.Video Sensor Networks**, 2011 – Springer.
- [14]Torralba, A., Fergus, R., and Freeman, W. T. (2008). **80 million tiny images: A large data set for nonparametric object and scene recognition**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(11):1958–1970

Reference

- [15]S. S. Tsai, D. M. Chen, G. Takacs, V. Chandrasekhar,R. Vedantham, R. Grzeszczuk, and B. Girod. **Location coding for mobile image retrieval**. In Proc. 5th International Mobile Multimedia Communications Conference, 2009.
- [16]D. M. Chen, S. S. Tsai, R. Grzeszczuk, R. Vedantham, and B. Girod, "**Streamingmobile augmented reality on mobile phones**," in *Proc. Int. Symp. Mixed and Augmented Reality (ISMAR)*, Orlando, FL, Oct. 2009.
- [17]C. Christoudias, R. Urtasun, and T. Darrell. **Unsupervised feature selection via distributed coding for multi-view object recognition**. In CVPR, 2008.
- [18]Stefano Melacci, Marco Maggini, and Marco Gori. **Semi–supervised Learning with Constraints for Multi–view Object Recognition.** Artificial Neural Networks–ICANN 2009, 2009 – Springer
- [19]Amir Saffari, Christian Leistner, Martin Godec, and Horst Bischof. **Robust Multi-View Boosting with Priors.**Computer Vision–ECCV 2010, 2010 – Springer
- [20]. Vedaldi and B. Fulkerson. **VLFeat: An open and portable library of computer vision algorithms.** http://www.vlfeat.org/,2008
- [21]Google goggles. http://www.google.com/mobile/goggles/.
- [22]Machine Learning in Google goggles. http://techtalks.tv/talks/54457/#

L/O/G/O



Thank You!

http://home.ustc.edu.cn/~liutao/