Image content search

Jozef Mlích <imlich@fit.vutbr.cz>

Department of computer graphics and multimedia
Brno University of Technology

Abstract. This report makes an overview on approach in the problem of content based image retrieval. It covers description of the implemented baseline system. The description of the system design and the methods related to image processing, computer vision, data processing, search and indexing, are included. Moreover, the evaluation methods for such systems are mentioned. The experimental results achieved on implemented system are shown.

1 Introduction

Nowadays, a very large amount of data is gathered. Since, the manual image and video processing is expansive and time consuming task, the automatic video processing and understanding methods becoming to be more important. The potential is especially in following areas; the video surveillance, the computer gaming industry, television industry and also the home video processing.

In parallel with the text content analysis, the requirement to the image processing stand. There are a couple of very common requirements on manipulation with such amount of image data. For instance data labeling, sorting and search. The problem scope is very wide and includes high level image understanding techniques, such as object class recognition, object instance identity recognition, etc.

The report concern with image retrieval problem. More concrete, with a problem of the image search given by its example. It is structured as follows. In the first section is brief introduction to Information Retrieval problem. Follows brief overview on the state of the art systems. In the next section the problem of image analysis and its comparison is discussed. Follows the section describing efficient ways of image search in large databases using codebooks and indexing techniques using selected image description method. In next section, the evaluation of the image retrieval systems in general and the

evaluation of implemented system is shown. The proposal of further system improvements and overall conclusions about its performance and accuracy is discussed in the last section.

2 Information Retrieval

The key goal of an information retrieval system is to retrieve information which might be relevant to the user. For these systems the inaccuracy and small errors in the search are not important for fulfill its target. The system interpret the documents (logical view on the document) and match the user required information. The information retrieval consist of modeling, classification, categorization, user interface, visualization, filtering, etc.

As presents Baeza-Yates[BYR00], the models vary for different types of data. The *rule model* creates rules according to semantic relations between elements. The *clustering model* divides data into the groups with similar characteristics. The *regression model* adapts it self to the newly acquired data. The *classification model* assigns classes to elements of model according to its annotation.

It seems to be reasonable to describe image by sentence e.g. *Eiffel tower* or *sunset on the beach*. However, this approach requires word description of the image object, which could be very uncertain and difficult on semantic analysis. Also requires additional text meta data to the document. On the other hand, it is possible to describe the image by the its example, which provides in very natural way wide variety of informations about scene type e.g. *indoor/outdoor*, objects in scene e.g. *building, person, car*, etc.

3 State of the art

In the case of the image retrieval problem exists wide variety of algorithms and approaches. The Google Googles [NNS10] relies on fusion of the Optical Character Recognition Engine, Rigid Textured Object Recognition Engine, Face Recognition Engine and Articulate Object Engine. However, the detailed description of algorithm is property of Google and it is not public available. The Google could also profit on the knowledge of the context of the image in the document (i.e. the web page content which contains the image).

The Oracle introduced Multimedia extensions[ACG+] for their database engine, which provides interface for suitable audio-visual data indexing and search. The extensions capabilities are provided by *SQL/MM* language. For these purposes, the *signature* is created. In the case of image search, the signature consist of color, location, shape and texture description. It is possible to set weight which allows to adjust relevance of each part of signature.

There are other publicly available information retrieval engines, which allows [Vii], [Sno], [LTU], [SSZ]

The National institute of Technology performs evaluations TRECVid[1] in image retrieval in several task such as the Content-based copy detection, Rushes summarization, etc.

4 Image processing

This subsection undertakes image processing methods as the first and necessary part for the image based information retrieval. The image feature extraction should provide meta-information which allows to generalize object description to allow its content search with respect to different scale, rotation, position, illumination conditions, etc.

Nowadays, are popular the spare search methods. These methods relies on analysis of the salient points. These methods are based on the idea, that real world images consist of large solid areas, which are not important for human vision. More concrete, the position of the salient points are computed by simple usually rules based threat (e.g. certain frequency in image).

Typical representative of such methods is SURF[2] introduced by Herbert Bay[BETVG08]. The method combines detector and descriptor which is proposed with respect to repeatability, distinctiveness and robustness. This is achieved by computing Hessian Matrix and first order Haar wavelet response on integral image.

The Hessian Matrix $\mathcal{H}(x, \sigma)$ for the point x in an integral image I at scale σ is given by Equation 1, where the $L_{xx}(x, \sigma)$ is convolution of the Gaussian second order derivate $\frac{\delta^2}{\delta x^2} g(\sigma)$ with image I in point

[1] TREC Video Retrieval Evaluation
[2] Speeded-Up Robust Features

x and similar for other matrix members, whereas the repeatability of algorithm seems to be optimal with scale σ around multiplies of $\frac{\pi}{2}$.

$$\mathcal{H}(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma)\, L_{xy}(x,\sigma) \\ L_{xx}(x,\sigma)\, L_{yy}(x,\sigma) \end{bmatrix} \tag{1}$$

The approximation of Hessian determinant $det(\mathcal{H}_{approx}) = L_{xx\ approx} L_{yy\ approx} - (wL_{xy\ approx})^2$ represents the blob response, whereas Ls are computed on certain rectangular region and w is weight used for balance between Gaussian and the approximated Gaussian kernels. Furthermore, the filter responses are normalized with respect to their size. Using the advantage of integral image the extraction on different scale space could be done in constant time. However, the responses are overlapping in scales, therefore it is necessary to suppress non maximal responses. The descriptor in computed for each salient point on rectangular region rotated according to orientation. The orientation is calculated as sums of first order Haar wavelet responses in x and y direction within circular neighborhood of point. The rotated rectangular region is divided to 4×4 sub-regions. As is shown in Figure 1, the feature vector v of each sub-region consist of the Haar wavelet responses sums in x and y direction and sums of its absolute values $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$.

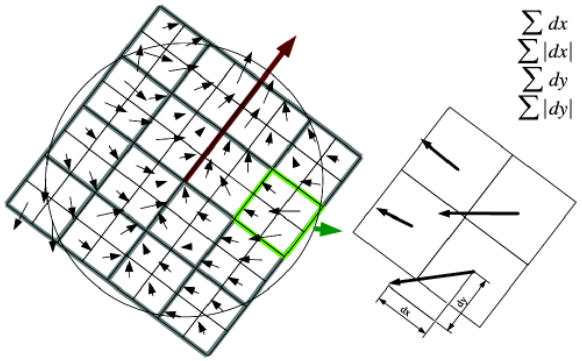


Fig. 1: Haar wavelet reponses for image sub-regions

There are more image processing algorithms for salient point detection which provides very similar features. Noteworthy are SIFT and MSER algorithms.

5 Search

In the case of the two images comparison using SURF method is the approach following. We search the geometric transformation of the salient points to corresponding points in the second image. In the case of the planar transformation (i.e. homography), we need five corresponding pairs of salient points to compute it. Using monte carlo approach RANSAC, we choose random points to create homography and the other points pairs as the evidence of such transformation correctness. This approach is suitable in the case of two similarity measure. However, the complexity of such approach is rather high and it is not suitable for search in large amount of data.

It is possible consider each salient point as a visual word (term) and its occurrence in document suggest document relevance. Even if we skip the geometrical relation between images, the complexity of the similarity measure is still rather high.

As presented Salton [SWY75], for text retrieval could be used tf-idf[3]. The term frequency is defined by Equation 2 describes the ratio of the occurrence of the term t_i within document d_j to all terms in document d_j, whereas the n denotes the frequency of the term.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2}$$

The inverse document frequency (see Equation 3) describes overall importance of term in all documents.

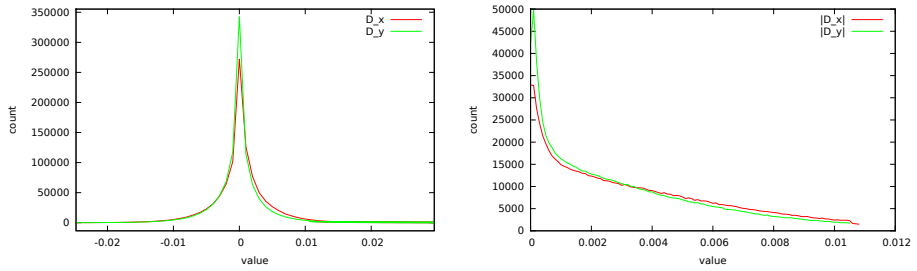
$$idf_i = log \frac{n_i}{\sum_k n_k} \tag{3}$$

The relevance of the term is given by Equation 4.

$$tf\text{-}idf_{i,j} = tf_{i,j} \times idf_i \tag{4}$$

[3] term frequency-inverse document frequency

The full dictionary covering all SURFs would be rather large, considering the dimension of the SURF descriptor (64 dimensions on 32-bit according to IEEE-754 i.e. 64^{32}) and distribution of its components (non fully utilized 32-bit float) as is shown in Figure 2.



a) b)

Fig. 2: Histogram of SURF values in corpus for a) D_x, D_y and b) $|D_x|$, $|D_y|$ with bin size a) 10^{-3} b) 10^{-4}

Despite of this, we can build dictionary using subset of all terms. In the analogy with fulltext search, where the stemming is done, we can very easily measure distance between surfs descriptors and assign any of them to the descriptor contained in dictionary. According to amount of descriptors occurred in document we can build *signature* which have constant rather small dimension which is feasible for search.

The signature construction relies on selection of proper and significant descriptors and consequent assignment of descriptors from document. In such amount of descriptors it is possible to choose random subset with rather high distinguish abilities. It is reasonable to use clustering techniques such as k-means or Gaussian Mixture Models to find better representatives of the data set. However, these techniques for such amount of data relies on non trivial memory efficient implementation.

The second step of this transformation is based on assignment of the descriptor to the dictionary. Van Gemert, et. al. [vGVSG10] described and evaluated precision of hard assignment and different soft assignment methods. This process is shown in Figure 3, whereas the red points $\{1, ..., 6\}$ represents the dictionary, the blue points

$\{A, ..., H\}$ represents the terms is searched document. The example shows transformation $6 \cdot 2 \cdot 8 \cdot 2 \rightarrow 6$

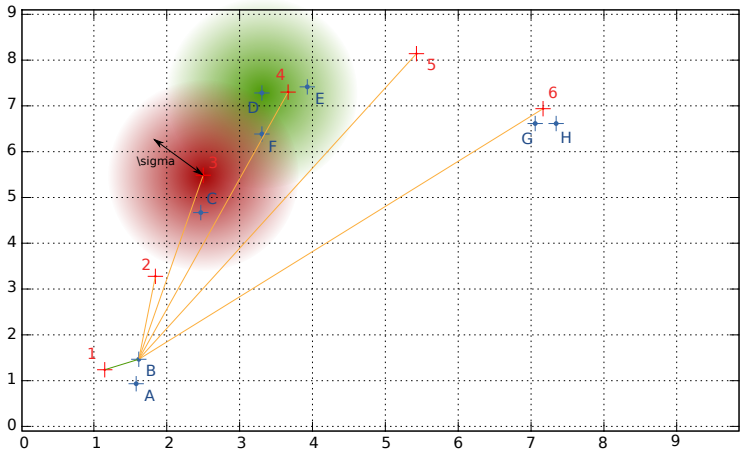


Fig. 3: Example of the assignment in two dimensional space

The hard assignment find for each descriptor w the descriptor in dictionary v with smallest distance $D(w, v)$. The signature is consequently normalized to the amount of surfs. The process could is described in Equation 5. The resulting signature for above mentioned example will be $\frac{1}{8} \cdot (2, 0, 1, 3, 0, 2)$.

$$HA(w) = \frac{1}{n} \sum_{i=1}^{n} \begin{matrix} 1, if\ w = argmin_{v \in V}(D(v, r_i)) \\ 0, otherwise \end{matrix} \tag{5}$$

The soft assignment methods take into the account the distance between descriptors in the document and in the dictionary. The basic method is the *codeword plausibility*. It could be described by Equation 6. The distance is additionally weighted by probability for normal distribution K_σ. The weighting is highlighted in the example for 3th and 4th term of dictionary. The estimation of the resulting signature for given example is about $\frac{1}{8} \cdot (1.8, 0, 0.7, 2.5, 0, 1.9)$.

$$PLA(w) = \frac{1}{n} \sum_{i=1}^{n} \begin{matrix} K_\sigma(D(w_i, v))\ if\ w = argmin_{v \in V}(D(w_i, v)) \\ 0, \qquad\qquad\qquad otherwise \end{matrix} \tag{6}$$

The *codeword uncertainty* (see Equation 7) accumulates the signature not only for the words with minimal distance, but for all neighborhood terms. The estimated UNC signature in this case will be about $\frac{1}{8} \cdot (1.8, 0.2, 0.8, 2.5, 0.1, 1.9)$.

$$UNC(w) = \frac{1}{n} \sum_{i=1}^{n} \frac{K_\sigma(D(w, r_i))}{\sum_{i=1}^{|V|} K_\sigma(D(v_i, r_i))} \tag{7}$$

However, the normal distribution K_σ parameter σ is given by dictionary distribution and could be obtained from clustering parameters or could be choosen ad hoc.

The soft assignment measure *best weighted distance*, described by Equation 8, takes into account the ratio between minimal distance of the observed term and terms in the dictionary.

$$BWD(w) = \begin{array}{ll} \frac{1}{n} \sum_{i=1}^{n} \left(1 - \frac{minD(w,r_i)}{\sum_{i=1}^{N}(D(v_i,r_i))}\right) & if\ w = argmin_{v \in V}(D(v, r_i)) \\ 0, & otherwise \end{array} \tag{8}$$

The estimated signature vector in Example would be $\frac{1}{8} \cdot (1.9, 0, 0.8, 2.6, 0, 1.8)$. In the example the Figure shows, the term B in image and highlighted distances with the dictionary terms.

Consequent image retrieval consist of search within signatures. The distribution of randomly selected firetruck image distance with rest of indexed images is shown in Figure 4. The distribution of distances from other annotated images in the same class is also shown in the Figure. To be efficient it is recommended use of proper indexing method. For indexing of the n-dimensional data are often used following methods: K-D-Tree, BSP-Tree, Point Quad Tree.

6 Evaluation

For the evaluation was selected data corpus consisting of dataset Caltech 256 [GHP], Kentucky dataset [SN] and TRECVid 2005 dataset[Nat]. It consist of about 200000 pictures with various parameters and various content. The evaluation is usually measured in the performance view and the correctness view.

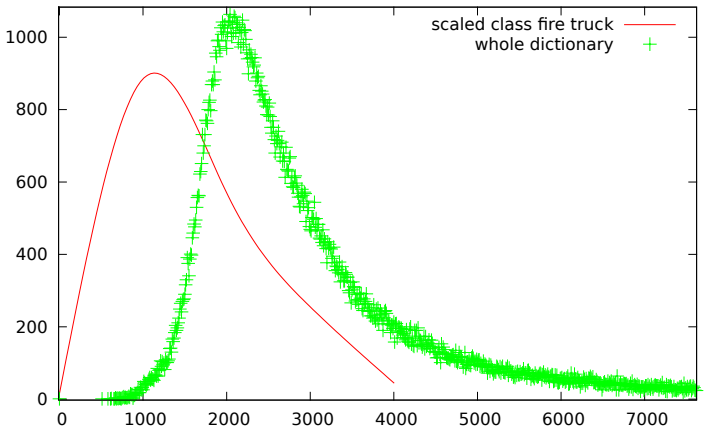


Fig. 4: Distance of randomly selected image with other images in database

First, the performance evaluation is discussed. Intel® Core™2 Duo CPU E6750 @ 2.66GHz with 2 GB RAM. In the case of image retrieval problem the performance is measured in the indexing and the search task. The performance consist of two separated phases. The image processing and the indexing or search. The image processing phase is dependent on image resolution and is constant for both tasks. In the case of SURF descriptors the duration was from 0.1 up to 5 seconds. More detailed analysis of the SURF method is in [BETVG08].

The signature extraction and indexing on data corpus with 500 dimensional random dictionary took 3.2s per image in average. The search task takes about 7s.

The second evaluation direction examines the result correctness. The taxonomy for correctness evaluation in terms of detection denotes the positive and negative detection which is compared with ground truth. Hence, there are 4 categories of detection *true positive*, *true negative*, *false positive* and *false negative*. Based on these, the accuracy $ACC = \frac{TP+TN}{TP+TN+FP+FN}$ and the precision $PRC = \frac{TP}{TP+FP}$, is defined.

From the image retrieval problem point of view, we interested in a first couple of results (i.e. with best score). The precision of the implemented system with 500 dimensional randomly selected

dictionary is shown in Figure 5. The evaluation were performed on a whole data set as ground truth was used the Caltech 256 corpus. The system with 30 best results outperform with precision about 5.4 % in average. It means that in 30 first results will be about 2 relevant picutres. The very good results was achieved on fire truck class where in 30 first results will be about 7 relevant pictures. The best results was achieved on class chess board.

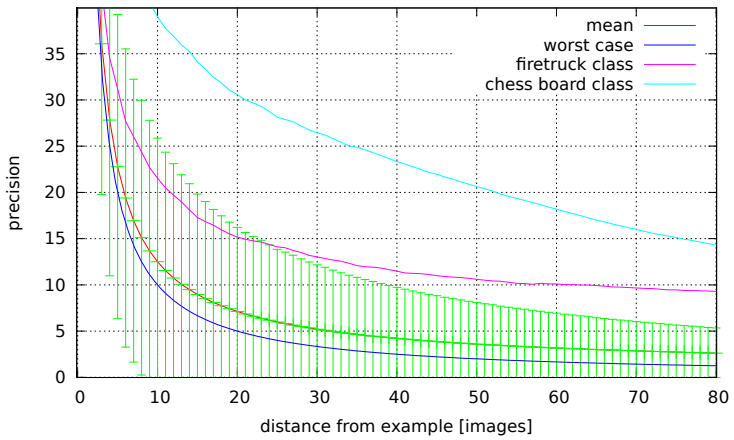


Fig. 5: mid-average precision

As the worst case is considered to be response containing only picture contained in the query.

7 Conclusions

This report presents a system which, in comparison with other state of the art systems brings no advantage. Describes the known approaches suitable for image search based on similarity and provides a basic platform for further experimentation in this area. It could be used as a demonstration application in education.

The achieved results suggest focus on improvements in following areas.

– (accuracy) using dictionary created by clustering method

- (accuracy) to design different soft assignment method e.g. Soft assignment based on probabilistic model with full covariance matrix.
- (accuracy) describing the objects by method distinguishing the color information.
- (accuracy) describing the objects by shape models
- (accuracy) fusion of different image description techniques (face detection, color model, shape model, semantic informations)
- (performance) dimension reduction using Principal component analysis
- (performance) better utilization of the indexing methods e.g. using K-D Trees.

References

ACG+. R. Abbott, F. Chen, B. Gettys, D. Guo, D. Lin, S. Mavris, P. Parida, J. Steiner, Y. Sun, S. Watt, et al. Oracle Multimedia Reference, 11g Release 1 (11.1) B28414-02.

BETVG08. Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision Image Understanding*, 110(3):346–359, 2008.

BYR00. R. Baeza-Yates and B.A.N. Ribeiro. *Modern information retrieval*. ACM Press Books. Pearson, Addison-Wesley, 2000.

GHP. G. Griffin, AD. Holub, and P. Perona. The Caltech-256. http://www.vision.caltech.edu/Image_Datasets/Caltech256/. [Online; accessed 29-January-2011].

LTU. LTU Technologies. Corbis. http://corbis.ltutech.com/. [Online; accessed 29-January-2011].

Nat. National Institute of Standards and Technology. TREC Video Retrieval Evaluation dataset. http://trecvid.nist.gov/. [Online; accessed 29-January-2011].

NNS10. H. Neven and H. Neven Sr. Mobile image-based information retrieval system, July 6 2010. US Patent 7,751,805.

SN. Henrik Stewénius and David Nistér. Kencutcky dataset. http://vis.uky.edu/ stewe/ukbench/. [Online; accessed 29-January-2011].

Sno. Cees G.M. Snoek. Mediamill semantic video search engine. http://mediamill.nl/. [Online; accessed 29-January-2011].

SSZ. Josef Sivic, Frederik Schaffalitzky, and Andrew Zisserman. Video google demo. http://www.robots.ox.ac.uk/∼vgg. [Online; accessed 29-January-2011].

SWY75. G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

vGVSG10. Jan C. van Gemert, Cor J. Veenman, Arnold W.M. Smeulders, and Jan-Mark Geusebroek. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1271–1283, 2010.

[vii] Jukka Viitala. Octagon - content based image retrieval software. http://octagon.viitala.eu/. [Online; accessed 29-January-2011].

A Caltech 256 precision on 20 nearest results

class	precision [%]
chess board	30.58333
saturn	16.92708
tower pisa	16.83333
baseball glove	16.28378
fire truck	15.16949
mountain bike	13.71951
t shirt	13.49162
fireworks	13.40000
tweezer	12.99180
sheet music	12.97619
brain 101	12.59036
waterfall	12.21053
french horn	11.25000
ak47	9.79592
snowmobile	9.77679
cactus	9.60526
toad	9.58333
microscope	9.57265
tennis racket	9.44444
homer simpson	9.43299
cockroach	9.31452
faces easy 101	9.24138
self propelled lawn mower	9.20833
palm tree	9.07767
american flag	9.02062
theodolite	8.92857
license plate	8.90110
stained glass	8.90000
diamond ring	8.72881
lathe	8.66667
camel	8.59091
minotaur	8.47561
binoculars	8.37963

butterfly	8.30357
cereal box	8.27586
backpack	8.11258
triceratops	8.10526
house fly	8.09524
bulldozer	8.00000
tricycle	7.94737
comet	7.89256
teepee	7.87770
gorilla	7.83019
steering wheel	7.73196
chopsticks	7.70588
treadmill	7.58503
mushroom	7.50000
top hat	7.43750
penguin	7.38255
spider	7.17593
starfish 101	7.03704
bear	6.96078
grapes	6.94030
bonsai 101	6.92623
centipede	6.90000
soccer ball	6.89655
grand piano 101	6.78947
hummingbird	6.72414
teddy bear	6.63366
covered wagon	6.54639
clutter	6.51753
cannon	6.50485
vcr	6.44444
horse	6.44444
guitar pick	6.44231
elk	6.43564
grasshopper	6.42857
harmonica	6.40449
baseball bat	6.37795

birdbath	6.37755
sushi	6.32653
desk globe	6.28049
cormorant	6.27358
raccoon	6.17857
crab 101	6.17647
tambourine	6.10526
harp	6.10000
skunk	6.04938
cd	6.02941
dice	6.02041
photocopier	6.01942
computer monitor	6.01504
ladder	6.01240
spaghetti	5.96154
chimp	5.95455
sneaker	5.94595
cartman	5.94059
stirrups	5.93407
tomato	5.92233
minaret	5.88462
billiards	5.88129
picnic table	5.87912
sunflower 101	5.87500
floppy disk	5.84337
sword	5.83333
refrigerator	5.83333
giraffe	5.83333
frog	5.81897
hamburger	5.81395
paperclip	5.76087
speed boat	5.75000
harpsichord	5.75000
superman	5.74713
traffic light	5.70707
computer keyboard	5.70588

laptop 101	5.70312
iguana	5.70093
segway	5.70000
golden gate bridge	5.68750
light house	5.68421
saddle	5.68182
telephone box	5.65476
watermelon	5.64516
dog	5.63725
golf ball	5.61224
welding mask	5.61111
screwdriver	5.58824
socks	5.58036
playing card	5.55556
teapot	5.55147
cowboy hat	5.52632
fire hydrant	5.50505
tuning fork	5.50000
bathtub	5.49569
syringe	5.49550
knife	5.49505
mattress	5.49479
roulette wheel	5.48193
toaster	5.47872
washing machine	5.47619
frying pan	5.47368
megaphone	5.46512
blimp	5.46512
smokestack	5.45455
boxing glove	5.44355
xylophone	5.43478
computer mouse	5.42553
ipod	5.41322
unicorn	5.41237
tennis ball	5.40816
pyramid	5.40698

horseshoe crab	5.40230
calculator	5.40000
watch 101	5.39801
windmill	5.38462
pez dispenser	5.36145
football helmet	5.35714
coffee mug	5.34483
fried egg	5.33333
radio telescope	5.32609
beer mug	5.31915
ketch 101	5.31532
llama 101	5.29412
ibis 101	5.29167
spoon	5.28571
buddha 101	5.25773
joy stick	5.23077
car tire	5.22222
flashlight	5.21739
hawksbill 101	5.21505
skyscraper	5.21053
ewer 101	5.18072
fire extinguisher	5.17857
soda can	5.17241
coffin	5.11494
ice cream cone	5.11364
video projector	5.10309
lightbulb	5.05435
conch	5.04854
kangaroo 101	5.00000
car side 101	5.00000