A New Content-Based Image Retrieval Method Based on the Google Cloud Vision API

Shih-Hsin Chen[a], Yi-Hui Chen[b,c]

[a]*Department of Information Management, Cheng Shiu University, No.840, Chengcing Rd., Niaosong Dist., Kaohsiung City 83347, Taiwan (R.O.C.)*

[b]*Department of M-Commerce and Multimedia Applications, Asia University, No. 500, Lioufeng Rd., Wufeng Dist., Taichung City 41354, Taiwan (R.O.C.)*

[c]*Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 404, Taiwan (R.O.C.)*

Abstract

Content-based image retrieval (CBIR) methods analyze the content of an image, extract the features that describe images, and yield image annotations or labels. A machine learning (ML) algorithm is commonly used to acquire these annotations. The need to import a large number of training images and use of many CPU hours are the two primary difficulties of using existing ML algorithms. Google Cloud Vision application programming interface (API) might overcome these two shortcomings.

Cloud Vision API is trained by Google; therefore, it saves computational time in obtaining image labels. We test whether this API can outperform existing ML algorithms in describing annotations. To the best of our knowledge, this paper is the first to illustrate the effectiveness of Cloud Vision API for image labeling.

Our programs are coded in the R language, which calls Cloud Vision API. We retain at most ten labels—with scores over 50—for each image. Because there is a semantic gap between the labels returned by Cloud Vision API and the image datasets, we define a transformation method to map the terms.

We selected a well-known dataset and 4972 figures, and we used them to compare Cloud Vision API with the corresponding image annotation algorithms. This work found that Cloud Vision API yields 42.4% correctness among the 4972 images. In each dataset, Cloud Vision API is more effective than the ML algorithms.

This paper compares the CBIR performance of Google Cloud Vision API and some ML algorithms. According to the extensive experimental results, Cloud Vision API is quite competitive compared with other image annotation algorithms. Hence, this API could be extended to test other image datasets and be used as a benchmark method for

Email addresses: shchen@csu.edu.tw (Shih-Hsin Chen), chenyh@asia.edu.tw (Yi-Hui Chen)

Preprint submitted to Computer Vision and Image Understanding *October 25, 2017*

evaluating ML algorithm performance.

Keywords: Content Based Image Retrieval, Image Annotation, Google Cloud Vision API, Deep Learning

1. Introduction

Content-based image retrieval (CBIR) [30, 16, 25, 30, 38] is a typical image retrieval method. CBIR methods analyze the content of an image and extract features, such as color, texture, and shape. They use these low-level image features to retrieve images with similar features from a dataset. Image annotation [19] is roughly categorized into classification-based methods [23, 17, 35] and probabilistic modeling-based methods [6, 8, 19, 20, 44]. For classification-based methods, a machine learning (ML) algorithm is commonly used to extract low-level features from images that are trained as patterns by using supervised learning methods [2]. An unlabeled image is classified according to the comparison of its features with these patterns. Subsequently, the labels of categories, such as "animal" and "building," become the annotations for the image. Thus, classification-based methods recognize unlabeled images and add the corresponding labels according to the trained classifier.

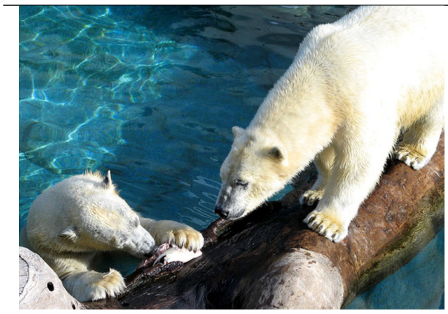
The classification-based approach performs well, but the training required is a time-consuming process [40, 42], and the cost is even higher if experts are involved in the label assignment [32, 33]. To enhance the efficiency of image retrieval, some frameworks [4, 14] are proposed. However, if the classifier is not trained in a corresponding category, the model is not able to assign a correct label for that figure. To overcome these difficulties, in this paper, we propose a new approach using Google Cloud Vision application programming interface (API)[1], which applies deep learning algorithms and convolution neural networks [21, 11]. Because Cloud Vision API is trained with a large number of pictures for millions of CPU hours, researchers and practitioners could benefit from this model.

Table 1 depicts a figure acquired from ImageNet [18]. We display the three returned labeling results in JavaScript Object Notation (JSON) format. In each JSON record, the mid, description, and score are the message ID, image annotation, and the score of this annotation, respectively. The score value represents the confidence of this annotation. In this example, the image annotation with the highest score is "polar bear" (98.55%). The second is "mammal" (95.87%) and the third is "animal." The number of available labels is different for each figure. In this polar bear example, we could obtain more than ten labels. For example, the 12th label is "biology" and has a score of 60.71%.

[1] https://cloud.google.com/vision/

However, for some figures, Cloud Vision API may not return any labels because their scores are too low.

Table 1: Image labels of a polar bear photo taken from ImageNet

	{”mid”: ”/m/0633h”, ”description”: ”polar bear”, ”score”: 0.985539}, {”mid”: ”/m/04rky”, ”description”: ”mammal”, ”score”: 0.9587503}, {”mid”: ”/m/0jbk”, ”description”: ”animal”, ”score”: 0.9344022}

By using this example, Cloud Vision API might be capable of extracting annotations from images. The reasons are due to this API applies the deep learning algorithms and convolution neural network. Also, the Vision API receives extensive image annotation training taken millions of CPU hours of the training time. This work, consequently, likes to compare the performance of this API with existing ML algorithms. Besides, because there is a semantic gap between the labels returned by Cloud Vision API and the image datasets, we design a transformation approach to mapping the terms.

This paper is organized as follows. Section 3 presents the transformation method which could map the image annotations of the Cloud Vision API to the testing datasets. In Section 4, we take one well-known dataset and compare the Vision API with some state-of-art algorithms in the literature.

2. Review

The main way of CBIR is to analyze the content of image and extract the features (e.g. colors, texture, shape, and so on) which describe images. It makes use of the low-level image features to retrieve similar images from the image dataset according to the similarity of image features. Nevertheless, there is a problem that computer uses a series of numerical value to express an image, which is widely divergent from the languages and words of human being, called semantic gap [3, 13, 23, 23, 26]. To overcome the problem of the semantic gap, image annotation is a way to annotate text information on images. The image annotation [19] is roughly divided into classification-based methods [17, 23, 35]and probabilistic modeling-based methods [8, 19, 20, 6, 44]. Classification-based methods extract low-level features from images to train as patterns by supervised learning methods. An un-labeled image will be classified into the categories according to the results that the features compared to the patterns. Later on, the labels of categories, such as human, building, etc., are as the annotations for the image. That is, the classification-based methods can recognize the un-labeled images to add the corresponding labels according to the trained classifier. We show the two important parts in Section 2.1 and Section 2.2, respectively.

2.1. Classification-based methods

One of the classification-based methods is based on SIFT features to generate bag-of-visual words (BVW) for object recognition [17, 23, 29]. The first step of BVW is to extract the SIFT keypoints of the training images; then, to cluster the keypoints into several groups by k-means. Then, for each image, it calculates the number of SIFT keypoints in each cluster and translates into vector to re-describe the image. Each category is trained to generate a classifier by supervised learning approach such as SVM. However, there are thousands of SIFT keypoint in an image, which takes a lot of time to train classifiers. Also, the accuracy of classification is affected by the noises.

Kesorn and Poslad [17] proposed a method to improve the qualities of visual words. The idea of [17] is to combine the close keypoints and removes the cluster that has high document frequency and small statistical association with all the categories (concept) in the dataset. Lu and Wang [27] developed a semantic regularized matrix factorization based on Laplacian regularization to improve the efficiency during the training process of BVW. In addition to BVW model method, AICMD [39] proposed by Su et al. to create different models to represent the images. AICMD extracts six different low-level features for clustering to generate patterns. The patterns integrate entropy, tf-idf and association rules as image features to represent the images. After that, all the features are used to train the classifiers by SVM (Support Vector Machine). Instead of SVM, some researches use Hidden Markov Models [1, 22].

Classification-based approach has great performances, but it is a time consuming process during training. Nevertheless, it is difficult to recognize the object as an instance class, e.g., Barack Obama, St. Peter's Basilica. In addition, users might use the same word to tag different things, called ambiguity problem. For the reason, Feng et al. [9] rank tags in the descending order of their relevance to the given image to reduce the learning space for signicantly simplifying the time consuming problem.

Zhang et al. and Xia et al. proposed the method about refining and enriching the imprecise tag words in 2013 and 2014 separately to solve the ambiguity problem [41, 44]. Zhang et al. used Random Walk with Restart (RWR) algorithm to refine the imprecise tag of the query image, called CTSTag. The generated precise tags are help to connect different images with the related tags. In addition, to enrich the image tag, Xia et al. used the concept of ontology to increase the precision of the tags in the image social networking service (e.g. Flickr). As for hierarchical concept, Yuan et al. proposed a hierarchical image annotation system to generate the hierarchical tags for images [43].

Moreover, Fang et al. [5]proposed an ontology hierarchically concepts and concept relationships to create the semantic understanding for information retrieval.

2.2. Probabilistic modeling-based method

The probabilistic modeling-based method calculates the joint probability between image content and the corresponding annotations. Probabilistic modeling-based methods divide the un-label image into several image segments. Then, the probabilities of the labels mapped to the image segments are calculated to find the labels with high probabilities as the image annotations. Mori [29] calculates the co-occurrence as the relationship between the sub-images and the corresponding labels. First, the image is divided into several sub-images. Second, it extracts low-level features for clustering. Next, the co-occurrence between each cluster and the related labels is calculated. Although it takes shorter time of the process than classification-based method, the accuracy is worse than that one.

Kuric and Bielikov [19] consider both the local and global features of the images. The local and global features are both extracted from the regions of the images; then, locality-sensitive hashing (LSH) is used to represent the regions for clustering. For an un-label image, the similar region in dataset is chosen and the weight of each label is calculated to renew the probability for labeling the un-label images.

Zhang et al. [44] propose ObjectPatchNet, which is the hybrid scheme of BVW and probability. ObjectPatchNet calculates the co-occurrence among each cluster and also considers the probability between image patches and labels to present the relationship. The limits of probabilistic modeling-based method are (1) the low-level feature is lack of semantic, and (2) the low-level features of the same individual objects, but different orientation are judged as dissimilar. Compare to classification-based method, probabilistic modeling-based method is more effective, and it can be applied to social platform, such as Flickr. Generally, the concrete expression for images enables to use instance classes to defined the labeled of categories. However, the low-level features with no semantic information to cause two images indicating to the same object but with different angles or orientations to be judged as two different objects.

Applied to large amount of data, Hong et al. [12] creates the relationship between semantic concepts based on the data from image commercial engine. The relationship of each pair of concepts is classified into a specific relationship.

The methods mentioned above are based on image recognition, but lots of abstract concepts, e.g., location, cannot be defined according to the image features. No doubts that it is difficult to get accurate results because the images are not with abstract concepts. That is, the retrieval results are incomplete if the datasets are not clearly defined with domain knowledge. Consequently, it desires that the labels for image annotation are with semantic meaning by using ontology theory to identify various definitions, attributes and the relationships between individuals [15, 31, 34, 37].

3. Methods

In this study, there are three methods to annotate the images. First of all, Google Cloud Vision API was applied to annotate the labels of figures, which is named Method 1. The testing images are directly processed by GCV. Because Method 1 does not consider the synonyms, it decreases the output accuracy. For example, terms for the polar bear that we illustrated in Section 1 include "ice bear," "Thalarctos maritimus," and "Ursus maritimus" [28] according to WordNet.

As a result, we further propose Method 2. The approach is that after we validated whether the labels generated by Google Cloud Vision API were consistent with the instance label, we made a further comparison using WordNet. WordNet has defined a set of nouns, verbs, adjectives, and adverbs that are grouped into sets of cognitive synonyms [28]. Using these synonyms, we verified the words provided by the names that were generated by Google Cloud Vision API from the image datasets. The research framework is shown in Figure 1.

Even though WordNet supplies synonyms of a given term, there remain a fundamental problem caused by the category labels named by a dataset. Take Pascal VOC 2007 for example, they combine two words into one, e.g. the potted plant to be "pottedplant and dining table is named "diningtable" . Besides, they use "tvmonitor" instead of tv or television. It is not likely for WordNet to find synonyms of these words. We intend to propose Method 3 to solve this problem. The

We coded our programs in R language to parse the images and obtain labels. The labels of each image were stored in a database. To help researchers conduct further studies, the code for these programs is available on Github [2].

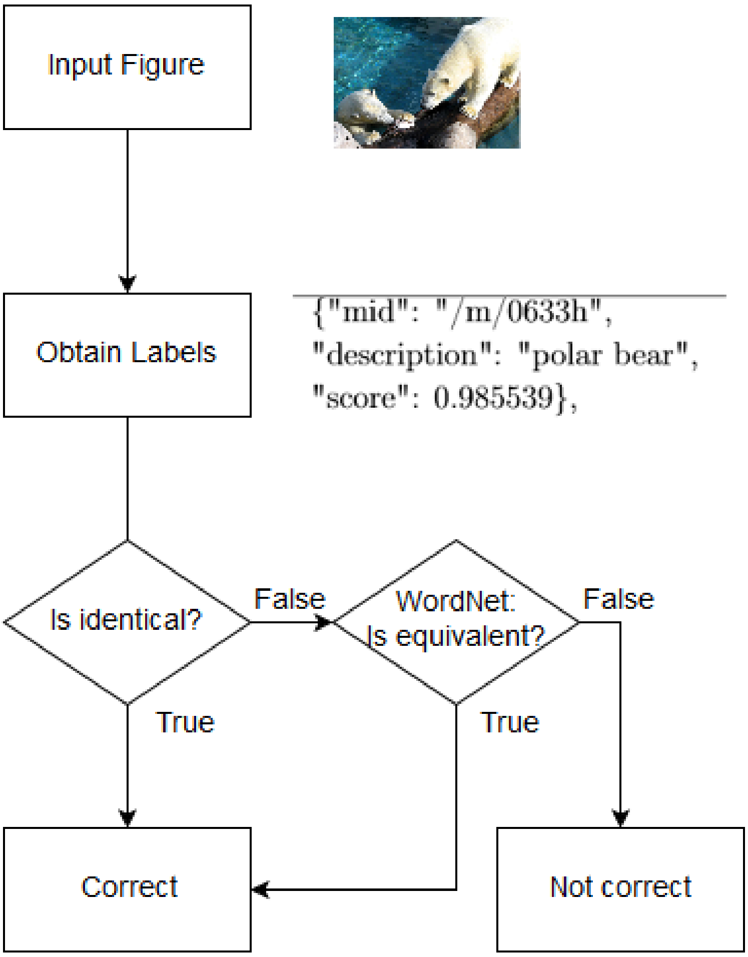
Because the annotations selected by Cloud Vision API might be different from the original category names of the selected image datasets, we designed a transformation to map the terms. This approach involved applying the method provided by WordNet. This mapping method was also completed using R language.

4. Experiment Results

To verify the effectiveness of Google Cloud Vision, we examined a prominent image dataset (Pascal VOC 2007). This dataset contains 4952 images in 20 categories. We input all the images as test cases and obtained their labels with Google Cloud Vision API. Then, in Method 1, we validated the correctness of the labels provided by Google Cloud Vision API. Finally, in Method 2, we used WordNet to close the semantic gap of the labels. The equivalent names obtained from WordNet are shown in Table 2.

Table 2 shows that the synonyms of "aeroplane" are "airplane" and "plane" and the synonyms of "car" include "auto" and "automobile." However, the

[2] https://github.com/worldstar/GoogleCloudVisionAPI_RLanguage



```
{"mid": "/m/0633h",
"description": "polar bear",
"score": 0.985539},
```

Figure 1: Research procedure

terms WordNet provided for ” chair” were unwanted because the results included “chairman,” “chairperson,” “chairwoman,” “electric chair,” and “hot seat.” Moreover, the synonyms of “motorbike” did not include “motorcycle” and categories including “dining table,” “potted plant,” “sheep,” and “TV monitor” did not yield any synonyms. Therefore, the provision of correct synonyms could be improved by WordNet. Semantic gaps remained between category names defined in the dataset and the synonyms given by WordNet.

Table 3 displays the detailed results of each category for Methods 1 and 2. The mean average percentage of Methods 1 and 2 were 32.1% and 42.4%, which indicated an improvement with the incorporation of WordNet. The 10.3% improvement was mainly due to the average precisions of “aeroplane,” “car,” and “sofa” of 0% in Method 1. After we applied WordNet, their average precisions were 78.5%, 77.4%, and 19.1%,

Algorithm 1 Method3

N: The number of categories
i: Index of a category where i = 1,. . .,N
π: A set of training figures collected by N categories
S: A set of testing figures of N categories
k: Sample size
θ: A threshold value where θ is [0, 1]
r: Successful rate
ω: A set of category index should be trained
M: A model trained by an existing algorithm

```
$i \leftarrow 1$
while $i <= N$ do
    Randomly select $k$ figures from $\pi_i$
    $r \leftarrow$ Evaluate the $k$ figures by Method 2
    if $r <= \theta$ then
        $\omega \leftarrow i$
    end if
    i ← i + 1
end while
Train model $M$ by using the figures in $\omega$
for all $j \in S$ do
    if $M$ gives a label for $j$ then
        Output label for figure $j$
    else
        Process this image by Method 2
    end if
end for
```

respectively. Hence, researchers or practitioners could apply WordNet to improve accuracy.

However, the results for “diningtable,” “motorbike,” “pottedplant,” and “tvmonitor” were zero because WordNet was unable to supply adequate terms. If WordNet could supply the term “television” and “motorcycle” for “motorbike” and “tvmonitor”, respectively, the number of corrects is 135 and 24. Their average average correctness become 74.59% and 12.5% and the overall mAP is 46.76%. Solving this problem would enable further increases in the accuracy of the proposed method. It is the reason why we propose a more general method named Method 3.

In Method 3, we sample k figures from the training dataset. If the average is precision is less than a threshold θ, we activate an algorithm to train the figures in that

Table 2: The corresponding synonym(s) given by WordNet

Category	Synonym(s)
aeroplane	airplane, plane
bicycle	bike, cycle, wheel
bird	birdie, boo, Bronx cheer, chick, dame, doll, fowl, hiss, hoot, raspberry, razz, razzing, shuttle, shuttlecock, skirt, snort, wench
boat	gravy boat, gravy holder, sauceboat
bottle	bottleful, feeding bottle, nursing bottle
bus	autobus, busbar, bus topology, charabanc, coach, double decker, heap, jalopy, jitney, motorbus, motorcoach, omnibus, passenger vehicle
car	auto, automobile, cable car, elevator car, gondola, machine, motorcar, railcar, railroad car, railway car
cat	African tea, Arabian tea, big cat, bozo, CAT, Caterpillar, cat-o'-nine-tails, computed axial tomography, computed tomography, computerized axial tomography, computerized tomography, CT, guy, hombre, kat, khat, qat, quat, true cat
chair	chairman, chairperson, chairwoman, death chair, electric chair, hot seat, president, professorship
cow	moo-cow
diningtable	-
dog	andiron, blackguard, bounder, cad, Canis familiaris, click, detent, dog-iron, domestic dog, firedog, frank, frankfurter, frump, heel, hotdog, hot dog, hound, pawl, weenie, wiener, wienerwurst
horse	buck, cavalry, Equus caballus, gymnastic horse, horse cavalry, knight, sawbuck, sawhorse
motorbike	minibike
person	individual, mortal, somebody, someone, soul
pottedplant	-
sheep	-
sofa	couch, lounge
train	caravan, gear, gearing, geartrain, power train, railroad train, string, wagon train
tvmonitor	-

category. We suppose θ is zero and we adopt the R-CNN FT fc$_7$ BB [10] to train the images.

We further compared our approaches with previously reported algorithms including the R-CNN FT fc $_7$ BB [10], DPM v5 [7], DPM HSC [36], and DPM HSC [36] algorithms . Their accuracy is represented as the mean average precision (mAP). These four algorithms were tested with 4952 images from the Pascal VOC 2007 dataset. We found

Table 3: The detail results of each category

		Method 1		Method 2		Method 3
Category	Total	Corrects	Avg (%)	Corrects	Avg (%)	Avg (%)
aeroplane	200	0	0	157	78.5	78.5
bicycle	189	128	67.7	131	69.3	69.3
bird	275	196	71.3	197	71.6	71.6
boat	162	104	64.2	104	64.2	64.2
bottle	130	17	13.1	17	13.1	13.1
bus	156	86	55.1	129	82.7	82.7
car	580	0	0	449	77.4	77.4
cat	305	213	69.8	213	69.8	69.8
chair	238	42	17.6	42	17.6	17.6
cow	123	0	0	0	0	63.5
diningtable	109	0	0	0	0	54.5
dog	384	271	70.6	274	71.4	71.4
horse	212	131	61.8	131	61.8	61.8
motorbike	181	0	0	0	0	68.6
person	845	194	23	194	23	23
pottedplant	119	0	0	0	0	33.4
sheep	93	56	60.2	56	60.2	60.2
sofa	215	0	0	41	19.1	19.1
train	250	171	68.4	171	68.4	68.4
tvmonitor	192	0	0	0	0	64.8
mAP			32.1		42.4	56.7

that the R-CNN FT fc$_7$ BB was the best algorithm. Method 2 of this paper achieved the second highest performance, outperforming the DPM v5, DPM HSC, and DPM HSC algorithms. This result showed that Method 2 yields at least an 8% improvement over the other algorithms.

Table 4: The comparison results

Methods	Corrects	mAP
Method 1 (Raw results)	1609	32.1%
Method 2 (Using WordNet)	2306	42.4%
Method 3 (With Training)	2805	56.7%
R-CNN FT fc$_7$ BB [10]	2897	58.5%
DPM v5 [7]	1669	33.7%
DPM ST [24]	1441	29.1%
DPM HSC [36]	1699	34.3%

The overall results obtained using the Pascal VOC 2007 image dataset show that Google Cloud Vision API combined with WordNet yielded an acceptable result, even though we did not train their models. Omitting a prior training step saves substantial computational time. Hence, although the CPU time of the compared algorithms was not reported, our proposed method is faster than the algorithms in the literature.

5. Conclusions

This study is the first to investigate the effectiveness of Google Cloud Vision API compared with some efficient algorithms in the literature. Despite the absence of some category names, GCV and WordNet together provide acceptable precision compared with previously reported algorithms because WordNet closes the semantic gap between the labels generated by GCV and the image dataset. Most importantly, the computational effort is reduced because the proposed methods do not require training. Therefore, this approach might be a worthwhile direction for researchers and practitioners. Researchers could employ our proposed algorithm in their own research framework. In industry, this framework could be applied directly to fit their image recognition requirements. In future research, we will modify the category names to fit the labels given by GCV. Moreover, we will evaluate the proposed scheme to test more well-known datasets.

Acknowledgement

We are grateful for the Ministry of Science and Technology, R.O.C., supports this research with ID MOST 105-2221-E-230-009.

Reference List

[1] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, “Object trajectory-based activity classification and recognition using hidden markov models,” *Image Processing, IEEE Transactions on*, vol. 16, no. 7, pp. 1912–1919, 2007.

[2] S.-F. Chang, W.-Y. Ma, and A. Smeulders, “Recent advances and challenges of semantic image/video search,” in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4. IEEE, 2007, pp. IV–1205.

[3] C. Dorai and S. Venkatesh, “Bridging the semantic gap with computational media aesthetics,” *IEEE multimedia*, vol. 10, no. 2, pp. 15–17, 2003.

[4] N. D. F Golshani, “Retrieval and delivery of information in multimedia database systems,” *Information and Software Technology*, vol. 36, no. 4, pp. 235–242, 1994.

[5] Q. Fang, C. Xu, J. Sang, M. Hossain, and A. Ghoneim, “Folksonomy-based visual ontology construction and its applications,” 2016.

[6] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories,” *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part based models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[8] S. Feng, R. Manmatha, and V. Lavrenko, “Multiple bernoulli relevance models for image and video annotation,” in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–1002.

[9] S. Feng, Z. Feng, and R. Jin, “Learning to rank image tags with limited training examples,” *Image Processing, IEEE Transactions on*, vol. 24, no. 4, pp. 1223–1234, 2015.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[11] ——, “Region-based convolutional networks for accurate object detection and segmentation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 38, no. 1, pp. 142–158, 2016.

[12] R. Hong, Y. Yang, M. Wang, and X.-S. Hua, “Learning visual semantic relationships for efficient visual retrieval.”

[13] X. Hu, K. Li, J. Han, X. Hua, L. Guo, and T. Liu, “Bridging the semantic gap via functional brain imaging,” *Multimedia, IEEE Transactions on*, vol. 14, no. 2, pp. 314–325, 2012.

[14] K.-H. H. Hung-Yi Lin, Po-Whei Huang, “A new indexing method with high storage utilization and retrieval efficiency for large spatial databases,” *Information and Software Technology*, vol. 49, no. 8, pp. 817–826, 2007.

[15] D.-H. Im and G.-D. Park, “Linked tag: image annotation using semantic relationships between image tags,” *Multimedia Tools and Applications*, vol. 74, no. 7, pp. 2273–2287, 2015.

[16] H. Kekre, T. K. Sarode, S. D. Thepade, and V. Vaishali, “Improved texture feature based image retrieval using kekre’s fast codebook generation algorithm,” in *Thinkquest˜2010*. Springer, 2011, pp. 143–149.

[17] K. Kesorn and S. Poslad, “An enhanced bag-of-visual word vector space model to represent visual content in athletics images,” *Multimedia, IEEE Transactions on*, vol. 14, no. 1, pp. 211–222, 2012.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[19] E. Kuric and M. Bielikova, “Annor: Efficient image annotation based on combining local and global features,” *Computers & Graphics*, vol. 47, pp. 1–15, 2015.

[20] V. Lavrenko, R. Manmatha, and J. Jeon, “A model for learning the semantics of pictures,” in *Advances in neural information processing systems*, 2003, p. None.

[21] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[22] J. Li and J. Z. Wang, “Automatic linguistic indexing of pictures by a statistical modeling approach,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 9, pp. 1075–1088, 2003.

[23] L.-J. Li, C. Wang, Y. Lim, D. M. Blei, and L. Fei-Fei, “Building and using a semantivisual image hierarchy,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3336–3343.

[24] J. J. Lim, C. L. Zitnick, and P. Dollár, “Sketch tokens: A learned mid-level representation for contour and object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3158–3165.

[25] G.-H. Liu and J.-Y. Yang, “Content-based image retrieval using color difference histogram,” *Pattern Recognition*, vol. 46, no. 1, pp. 188–198, 2013.

[26] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, “A survey of content-based image retrieval with high-level semantics,” *Pattern Recognition*, vol. 40, no. 1, pp. 262–282, 2007.

[27] Z. Lu and L. Wang, “Learning descriptive visual representation for image classification and annotation,” *Pattern Recognition*, vol. 48, no. 2, pp. 498–508, 2015.

[28] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[29] Y. Mori, H. Takahashi, and R. Oka, "Image-to-word transformation based on dividing and vector quantizing images with words," in *First International Workshop on Multimedia Intelligent Storage and Retrieval Management.* Citeseer, 1999, pp. 1–9.

[30] S. Murala, R. Maheshwari, and R. Balasubramanian, "Local tetra patterns: a new feature descriptor for content-based image retrieval," *Image Processing, IEEE Transactions on*, vol. 21, no. 5, pp. 2874–2886, 2012.

[31] T. Osman, D. Thakker, and G. Schaefer, "Utilising semantic technologies for intelligent indexing and retrieval of digital images," *Computing*, vol. 96, no. 7, pp. 651–668, 2014.

[32] Y. Pan, T. Yao, T. Mei, H. Li, C.-W. Ngo, and Y. Rui, "Click-through-based cross-view learning for image search," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval.* ACM, 2014, pp. 717–726.

[33] Y. Pan, T. Yao, K. Yang, H. Li, C.-W. Ngo, J. Wang, and T. Mei, "Image search by graph-based label propagation with image representation from dnn," in *Proceedings of the 21st ACM international conference on Multimedia.* ACM, 2013, pp. 397–400.

[34] C. Pesquita, J. D. Ferreira, F. M. Couto, and M. J. Silva, "The epidemiology ontology: an ontology for the semantic annotation of epidemiological resources." *J. Biomedical Semantics*, vol. 5, p. 4, 2014.

[35] S. Poslad and K. Kesorn, "A multi-modal incompleteness ontology model (mmio) to enhance information fusion for image retrieval," *Information Fusion*, vol. 20, pp. 225–241, 2014.

[36] X. Ren and D. Ramanan, "Histograms of sparse codes for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3246–3253.

[37] M. Á. Rodríguez-García, R. Valencia-García, F. García-Sánchez, and J. J. Samper-Zapater, "Ontology-based annotation and retrieval of services in the cloud," *Knowledge-Based Systems*, vol. 56, pp. 15–25, 2014.

[38] I. H. Sarker and S. Iqbal, "Content-based image retrieval using haar wavelet transform and color moment," *SmartCR*, vol. 3, no. 3, pp. 155–165, 2013.

[39] J.-H. Su, C.-L. Chou, C.-Y. Lin, and V. S. Tseng, “Effective semantic annotation by image-to-concept distribution model,” *Multimedia, IEEE Transactions on*, vol. 13, no. 3, pp. 530–538, 2011.

[40] L. Von Ahn, R. Liu, and M. Blum, “Peekaboom: a game for locating objects in images,” in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 55–64.

[41] Z. Xia, J. Peng, X. Feng, and J. Fan, “Automatic abstract tag detection for social image tag refinement and enrichment,” *Journal of Signal Processing Systems*, vol. 74, no. 1, pp. 5–18, 2014.

[42] J. Yang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1794–1801.

[43] Z. Yuan, C. Xu, J. Sang, S. Yan, and M. S. Hossain, “Learning feature hierarchies: A layer-wise tag-embedded approach,” *Multimedia, IEEE Transactions on*, vol. 17, no. 6, pp. 816–827, 2015.

[44] S. Zhang, Q. Tian, G. Hua, Q. Huang, and W. Gao, “Objectpatchnet: Towards scalable and semantic image annotation and retrieval,” *Computer Vision and Image Understanding*, vol. 118, pp. 16–29, 2014.