Object Categorization by Local Feature Matching with a Large Number of Web Images

Mizuki Akiyama, Yoshiyuki Kawano, and Keiji Yanai

The University of Electro-Communications, Tokyo

Abstract. Generic object recognition is category-level object recognition, while specific object recognition is instance-level object recognition. Although their objectives are different, instance-level recognition possibly becomes category-level recognition by preparing a large number of various instances that belong to the given categories. Then, in this paper, we examine if generic object recognition can be performed by using specific object recognition methods with a large number of sample images. In the experiments, we used two kinds of the common methods for specific object recognition. One is matching raw SIFT features, and the other is matching visual words. For the twenty-five-category and five-category dataset, we obtained 32.5% and 60.1% for classification rate with SIFT feature matching, respectively. These results are almost equivalent to the results by the generic object recognition method which employs standard bag-of-features and SVM.

1 Introduction

"Generic object recognition" is category-level object recognition, while "specific object recognition" is instance-level object recognition. Recently, "generic object recognition" is drawing much attention in computer vision research. It is still difficult, since unrestricted generic images which are the target of generic object recognition are taken under various situations and conditions. On the other hand, due to the advent of local invariant features such as SIFT [1], "specific object recognition", the objective of which is finding out the same instances as the object registered in a database in advance, can be performed with the high precision rate. In general, searching for the exactly same objects is easier for a computer than classifying objects into one of the given categories.

Then, we propose applying specific object recognition methods to generic object recognition problems. Although the objectives of both recognition are different, instance-level recognition possibly becomes category-level recognition by preparing a large number of various instances that belong to the categories to be classified into. In this paper, we examine if generic object recognition can be realized by using specific object recognition methods with a large number of sample images.

As common methods for specific object recognition, we have two methods. One is matching raw SIFT features [1], and the other is matching visual words [2]. In both of them, instance-based recognition are carried out by matching local patterns between a given image and images in a database. In this paper, we employ both of two methods, and compare the experimental results by both methods changing parameters in various settings. As training data, we collected 145,000 images for five broad categories which contain 25 sub-categories from the Web using three Web image search APIs, and extracted SIFT features from all of them to build a large-scale local feature database. To classify a given image, we use a k-Nearest Neighbor classifier (k-NN). K-NN is a simple method, but it is strong since it can handle a large amount of training data which is difficult for more sophisticated classifiers such as Support Vector Machine (SVM) to treat by using fast indexing methods such as Approximate Nearest Neighbor (ANN) [3] or Locality Sensitive Hashing (LSH) [4].

As mentioned above, in this paper, we propose using instance-based specific object recognition with a large amount of training data for category-based generic object recognition problems. Since nowadays the quantity of image data over the Web is increasing exponentially, our proposed method which is based on simple k-NN and enormous data has possibility to become one of promising methods in the near future for practical use of image recognition.

The rest of this paper is organized as follows: Section 2 describes about related work, and Section 3 explains the overview and the detail of the proposed method. Section 4 shows the experimental results, and Section 5 discusses about the results. We conclude this paper in Section 6.

2 Related Work

Although we treat the problem on general object recognition in this paper, our objective is tackling it with specific object recognition methods. Here, we mainly introduce recent studies on specific object recognition.

The pioneer works of specific object recognition are SIFT by Lowe [1] and Video Google by Sivic et al. [2]. Lowe proposed Scale-Invariant Feature Transform (SIFT) which is a method to detect and describe local patterns around interest points, and showed its discriminative power for instance-based object recognition by matching interest points based on SIFT vectors [1]. To make SIFT-based object matching more large-scale, Sivic et al. proposed the idea of constructing "visual words" by vector-quantization of SIFT vectors [2]. "Visual words" enabled us to search image database for the images containing the given objects very fast in the same ways as text search like Google search engine does.

Ke et al. [5] proposed PCA-SIFT which compresses 128-dim SIFT vectors into 36-dim vectors, and showed that PCA-SIFT is comparable to SIFT in terms of recognition performance in spite of its reduced dimension. Philbin et al. [6] improved how to construct visual words and applied modified visual words to landmark search for a large-scale image database.

As recent work, Zheng et al. [7] applies a specific object recognition method to landmark database creation. They created large-scale landmark image database containing 5312 landmarks on 1259 cities in 144 countries from 20 million geotagged images on the Web, and they proposed a landmark recognition engine by employing SIFT-matching-based image search. Gammer et al. [8] also applied an instance-based object recognition method to landmark image search, which enables us to get to know the locations and the detail information about landmarks taken in the personal photos. In this way, specific object recognition methods work successfully with high precision for recognition of specific-shaped objects such as landmarks.

On the other hand, as work on generic object recognition using a large amount of image data, "80 million images" by A. Torralba et al. [9] is one of the most important works. They gathered 80 million images each of them is associated with one of WordNet nodes without excluding noise images, and showed that image categorization by k-Nearest Neighbor with 32×32 tiny images achieved comparable performance to the state-of-the-art methods. This result implied that a simple k-NN method with a very large amount of image data was one of promising approaches for generic object recognition, even if the data included noise images.

Our work is inspired by the "80 million images" work [9]. The difference to it is that we adopt SIFT-based local feature matching and voting on images based on the matched features, while they use the sum of squared differences (SAD) between two 32×32 images. Although local feature matching is well-known as specific object methods, it is not explored if it works as generic object recognition methods except for bag-of-features representations [10]. In this work, we also collect a large number of images from the Web, and we use them for experiments on image categorization without excluding noise images.

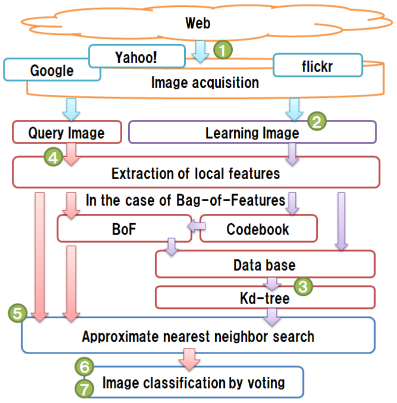

Web
Yahoo!
1
Google
flickr
Image acquisition
Query Image
Learning Image
2
4
Extraction of local features
In the case of Bag-of-Features
BoF
Codebook
Data base
Kd-tree
3
5
Approximate nearest neighbor search
6
7
Image classification by voting

Fig. 1. The processing flow.

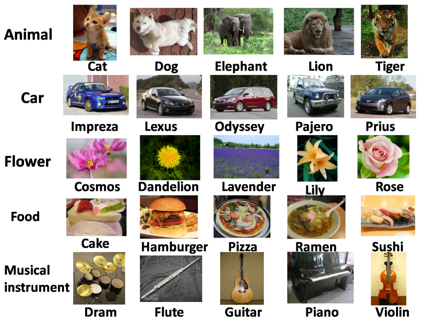


Fig. 2. Examples of images.

3 Proposed Method

In this paper, we tackle generic object recognition problems with specific object recognition methods with a large number of images gathered from the Web. As feature representation, we employ three kinds of well-known methods: SIFT [1], PCA-SIFT [5] and bag-of-features (BoF) [2, 10]. To classify a given image into one of the pre-defined categories, we use a k-nearest neighbor (k-NN) classifier varying k from 1 to 10000.

Basically, the processing is based on the standard specific object recognition methods. The processing flow (Figure 3) is as follows:

1. `[Training phase]` Collect a large number of training image to build a database from the Web.
2. Extract local features from all images with SIFT or PCA-SIFT. In case of the experiment on BoF, after generating a codebook from all features with k-means clustering, convert SIFT vectors into visual words.
3. Build a database structured by kd-tree in case of SIFT and PCA-SIFT. In case of BoF, create an inverted index.
4. `[Classification phase]` Extract local features from a query image.
5. For every single feature of the query image, perform Approximate Nearest Neighbor search (ANN) [3] for the kd-tree database to obtain n nearest neighbor features. In case of BoF, you use the inverted index to get the local features which are assigned as the same visual word as local features of the query image.
6. Vote on the training images having retrieved local features.
7. Classify the query image into one of the given category based on the majority of the categories of the top-voted k images according to the k-Nearest Neighbor classification..

3.1 Image Collection

We collected images from the Web by using Web APIs of the following Web image search engines: Google Image Search, Yahoo! Image Search and Flickr. Because the maximum number of retrieved URLs of Web images per query is limited to 1,000 in case of Google Image Search and Yahoo! Image Search, we prepare instance words which express instances of the categories in addition to main words which express the categories directly, and send instance words and several combinations of the main word and instance words to image search engines as query word sets. For example, we prepared "cat", "dog" and "elephant" as instance words of the "animal" category, and we used "cat", "dog", "elephant", "cat AND animal", "dog AND animal" and "elephant AND animal" to gather "animal" images. In addition, we prepare query words in both English and Japanese, and we gather images with both languages. Although Flickr has no limitation in terms of the number of returned images, we gather images from Flickr in the descending order of Flickr's original relevance scores by sending the same queries as Google Image Search and Yahoo! Image Search.

In this way, for one instance word, we gather Web images 12 times repeatedly (3 image search engines × 2 words combinations × 2 languages). After image gathering, we exclude duplicate images and preserve the ranking of images within each query. In the experiments, we select m images from the higher-ranked images on the search engine ranking as training images. (m depends on experiments.)

3.2 Local Features

In general, local features describe local patterns around interest points by their pixel values or differential values. If the interest points are detected on the same places on the objects and the feature vectors stay unchanged in spite of change of viewpoint, rotation and scale, the corresponding points between two images are easily detected by comparing local feature vectors.

Local features used in this paper are SIFT and PCA-SIFT. Furthermore, we use the bag-of-features (BoF) representation which is expressed by a histogram of frequency of local patterns regarding representative local patterns called as "visual words". The experiments in this paper are conducted in these three feature representations.

Before feature extraction, we normalize the size of images so that the size of long side of images is 640 pixels, when the long side of images is more than 640 pixel long.

SIFT features Scale Invariant Feature Transform (SIFT) proposed by D. Lowe [1] provides a multi-scale representation of an image neighborhood. They are Gaussian derivatives computed at 8 orientation planes over a 4×4 grid of spatial location, giving 128-dimension vector. The biggest advantage of SIFT descriptor is invariant to rotation, scale-change and illumination change.

PCA-SIFT features PCA-SIFT proposed by Ke et al. is an extension of SIFT. It estimates gradient information for a square region of 41×41 around interest points detected in SIFT, and totally extracts a 3,042-dim feature vector. By applying PCA, finally it is compressed to a 36-dim vector. The advantage of PCA-SIFT is that it is more efficient in matching and memory than original SIFT due to reduced dimension of local feature vectors.

Bag-of-Features The main idea of the bag-of-features is representing images as collections of independent local patches, and vector-quantizing them as histogram vectors. The main steps to build a bag-of-features vector are as follows:

1. Extract local feature vectors from all the images by SIFT [1].
2. Construct a vocabulary-tree-based codebook with the hierarchical k-means clustering [11] over extracted feature vectors.
3. Assign all feature vectors to the nearest codeword (visual word) of the codebook by following the vocabulary tree, and convert a set of feature vectors for each image into one k-bin BoF histogram vector regarding assigned codewords.

In the experiments, we explore the best setting of the codebook size k.

3.3 Database

To hold extracted local features, we use a kd-tree for approximate nearest neighbor (ANN) search. In case of experiments of SIFT and PCA-SIFT, we register all the local features associated with image IDs with a kd-tree.

In case of experiments of Bag-of-Features (BoF), we use an inverted index which is a kind of a list structure, each node of which corresponds to visual words and holds image IDs having the corresponding visual words in their BoF vectors [2]. This technique is originally invented for fast text-based document search. We register all the visual words of all the training images with their image IDs with the inverted index.

3.4 Feature search

For searching the kd-tree for similar local features, we use Approximate Nearest Neighbor (ANN) [3] to speed up normal nearest neighbor search.

ANN is a technique of the approximate nearest neighbor search over a kd-tree. First, we build a kd-tree by storing all the local features, and divide the feature space into regions called "cell" where only one data point exists. This means that a cell corresponds to a leaf node of a kd-tree.

Next, when searching the kd-tree for nearest neighbor features, we follow the kd-tree from the root of the tree to leaf nodes to find out the cell covering the query local feature q. However, the feature point p included in the found cell is not always the nearest point to the query feature Since its neighboring cells possibly contain the true nearest point, we have to search all the cells overlapped with the circle of center q and radius $d(q, p)$ where $d(q, p)$ represents distance between the query point q and the feature point p both of which belong to the same cell. We visit the other cells overlapped the circle, and calculate distances to the feature vector included in the cell to find out the true nearest point. Furthermore, we does not use distance of $d(q, p)$ to shorten the search time, but multiply it by $1/(1 + \varepsilon)$ to reduce the number of cells to be searched. In the experiments, we obtain not only one nearest point but the top n nearest points to each query local feature point.

In case of BoF, to search for similar local features, we just look up the nodes of the inverted index corresponding to the query visual words.

3.5 Recognition

After carrying out ANN to find out the most similar local features stored in the database to local features extracted from a query image or looking up the inverted index, we voted on the images from which the nearest local features are extracted. After voting in terms of all the local features of the query images, we sort the images having votes in the descending order of the number of votes, and we decide one of the given categories applying k-Nearest Neighbor classification.

k-Nearest Neigbor classification is a method to classify a sample into one of the given categories by examining the k most similar samples stored in the database. The category is decided by majority of the categories of the top k samples.

4 Experiments

4.1 Experimental setting

In the experiments, we collect images regarding 25 categories which belong to five kinds of broad categories. 25 kinds of categories are shown in Figure 2. In the left

Table 1. Statistics of the image database in terms of the number of images (imgs) and local features (feats).

	# imgs/class	Total # imgs	# feats/class	Total # feats
SIFT	1,050	26,250	600,000	15,000,000
PCA-SIFT	2,900	72,500	2,140,000	53,500,000
BoF	5,800	145,000	4,300,000	107,000,000

column, broad category words are shown, while instance words are shown under each sample image.

As described in Section 3.1, we gather Web images using three Web image search engines. We select fifty query images per class from the low-ranked collected images in terms of outputs of Web image search engines by hand, and use the rest of the images as training images. Note that training images contains some irrelevant images, while query images consist of only relevant images. The number of training images and extracted local features which are used in each experiment are shown in Table 4.1. Due to memory limitation, we could use only 26,250 images for the experiment with SIFT, while we used 72,500 and 14,500 images for the experiment with PCA-SIFT and BOF, respectively.

We made experiments with three kinds of local feature representations, SIFT, PCA-SIFT and Bag-of-Features (BoF). If there was big unevenness in terms of the number of local features between different categories, voting was performed profitably for the classes having larger number of local features. Therefore, we reduced the number of features by random sampling so that all the categories have the same number of local features.

In the experiments, we varied several parameters. In case of SIFT and CA-SIFT, we changed n which represents the number of retrieved local features for one query point when ANN search. In case of BoF, we changed the size of the codebook s. In addition, we also varied k which represents the number of top-voted images used for k-Nearest Neighbor classification.

We evaluated results of each category with the precision rate and the recall rate, and evaluated results over all the categories with the classification rate. The precision, the recall and the classification rate are calculated by (# correctly classified images)/(# classified images), (# correctly classified images)/(# relevant images), and (# correctly classified images over all the class)/(all # test images), respectively.

All the experiments are carried out on the PC with Intel(R) Xeon(R) Dual-core CPU 2.33GHz and 32GB memory.

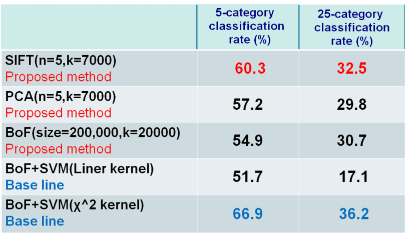
4.2 Experimental results

First, we made experiments on both 5-category and 25-category classification by changing the number of retrieved neighbor points n and k of k-NN image classification with SIFT and PCA-SIFT. To build a kd-tree from features, it took about one hour in the case of SIFT about 2 hour in case of PCA-SIFT. Regarding memory usage, the kd-tree for SIFT required 26GB, while the kd-tree for PC-SIFT required 20GB. In case of BoF, the used memory was only about 5GB, which showed that visual word representation can save a lot of memory. Although exact query time depends on the number of local features of a query image, it took about 2 or 3 seconds for one query image in the experiments with any of the three representations.

Table 2 shows the classification rate of the 5-category and 25-category classification with SIFT, PCA-SIFT and BoF in case of the best combination of n (s for BoF) and k. This table also includes the results by the baseline methods. As the baseline methods, we employed BoF with relatively a small number of visual words (s=1000) and support vector machine (SVM) with a linear kernel and a chi-square RBF kernel, both of which are standard methods of generic object recognition. To carry out 5- and 25-category multi-class classification, we adopted the one-vs-rest classification strategy. Although the results by the second baseline outperformed all the results by the proposed methods slightly, the differences are not so much. With slight improvement such as the number of training samples, the results possibly become comparable or superior to the baseline. In addition, since the second baseline employed one-vs-rest strategy and non-linear kernel, it is time-consuming and requires much more processing time compared to the proposed methods for classification, especially in case of increasing the number of the categories.

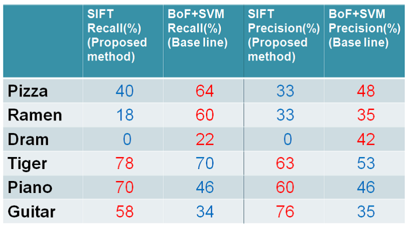
Table 3 shows the precision and the recall on six categories for comparison between SIFT with the best setting, $n = 5, k = 7000$, and BoF+SVM with a chi-square kernel. For "tiger", "guitar" and "piano", the proposed method outperformed the standard generic object recognition methods, since they have characteristic local features such as striped structures of "tigers" and "pianos" and strings of "guitars". On the other hand, the baseline results were much better than the SIFT results for "pizza", "ramen" and "dram", which have no characteristic local patterns.

Table 2. Results over all the categories.



	5-category classification rate (%)	25-category classification rate (%)
SIFT(n=5,k=7000) Proposed method	60.3	32.5
PCA(n=5,k=7000) Proposed method	57.2	29.8
BoF(size=200,000,k=20000) Proposed method	54.9	30.7
BoF+SVM(Liner kernel) Base line	51.7	17.1
BoF+SVM(χ^2 kernel) Base line	66.9	36.2

Table 3. Results on six categories.



	SIFT Recall(%) (Proposed method)	BoF+SVM Recall(%) (Base line)	SIFT Precision(%) (Proposed method)	BoF+SVM Precision(%) (Base line)
Pizza	40	64	33	48
Ramen	18	60	33	35
Dram	0	22	0	42
Tiger	78	70	63	53
Piano	70	46	60	46
Guitar	58	34	76	35

Figure 3 and 4 show the classification rate of the experiments of 5-category and 25-category classification with SIFT and PCA-SIFT varying k from 1 to 10000 and setting n as 1, 5, 10, and 25. Both of the figures show that the classification rate rises until $k = 7000$ as k increases, and becomes saturated at $k = 7000$ in terms of improvement of the results for both SIFT and PCA-SIFT. The best result achieved at $n = 5$ and $k = 7000$ for SIFT and PCA-SIFT.

Figure 5 and 6 show the results of the experiments with BoF varying the codebook size s with 50,000, 200,000, 500,000 and 1,000,000 and k from 1 to 100,000 Note that the varying range of k is ten times as broader as the SIFT experiments, since the number of training samples used in the BoF experiments is five times as many as in the raw SIFT experiments. Although the best results obtained at $s = 200,000$ and $k = 20,000$,

Figure 7 and 8 indicates the results with SIFT and the best setting, $k = 7000, n = 5$ varying the number of training samples with 1250, 3125, 6250, 12500, and 26250. These results implies that the classification rate also rises up, as the number of training images increases. The more improvement is expected if we can use more samples than 26,250. To do that, we need parallel computing or a PC with more memory than 32GB.

Table 4, 5, 6 and 7 show the confusion matrices of the experiments for 5-category classification with SIFT, PCA-SIFT and BoF and for 25-category classification with SIFT, respectively. The best parameter setting is used for each experiment. From these tables, "car" and "musical instrument" tends to be confused in case of 5-category classification. On the other hand, the sub-categories under the "car" category are easy to be confused with each other, and "flutes" were not classified correctly at all in case of 25-category classification.

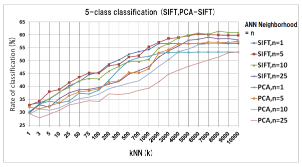


Fig. 3. 5-class classification by SIFT and PCA-SIFT.

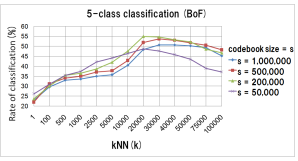


Fig. 5. 5-class classification by BoF.

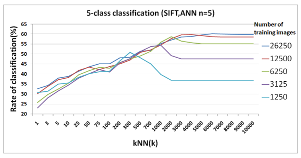


Fig. 7. 5-class classification by SIFT in case of changing the number of training samples.

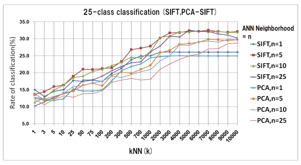


Fig. 4. 25-class classification by SIFT and PCA-SIFT.

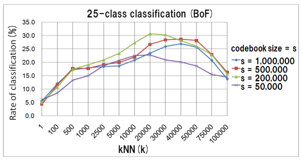


Fig. 6. 25-class classification by BoF.

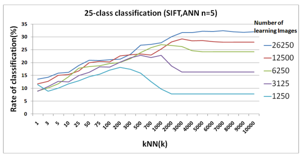


Fig. 8. 25-class classification by SIFT in case of changing the number of training samples.

5 Discussions

5.1 On differences of three approaches

Overall, SIFT outperformed PCA-SIFT and BOF, although the number of training images are least among the experiments with three kinds of local feature representations.

In addition, it is considered that SIFT experiment was more effective than PCA-SIFT experiment, because in the case of n=25 where many noise images possibly were retrieved, the classification rate by SIFT became lower than the other results with $n = 1, 5or10$, but rate of classification was improved in proportion to the range of k. On the other hand, the classification rate of PCA-SIFT was not improved.

In BoF experiment, the experimental results has become a little worse than SIFT and PCA-SIFT. Because the memory had room in BoF, we can use much more training samples for future work.

5.2 On n of ANN neighbors and k of k-NN votes

The classification rates for both SIFT and PCA-SIFT fell when enlarging the range of the number of n neighbors in ANN(Approximate Nearest Neighbor), but recognition accuracy was improved when enlarging the range of the number of k in kNN. In case of increasing the range of n, local features extracted from irrelevant images are also expected to be retrieved within top n features in ANN feature search. Therefore, when noise features are extracted more than the effective feature for images, it is considered that recognition accuracy may have fallen down.

On the other hand, in case of enlarging the range of k, it is thought that recognition accuracy may have risen as increasing k to 7000, because the effective features voted to the image of the correct class to some extent but the noise features voted scatteringly to various class images.

5.3 On classification rate

In case of 5-category classification, good accuracy was obtained for the categories such as keyboard of "piano" and string of "guitar" that were characteristic parts of the

Table 4. Confusion matrix of 5-class classification by SIFT

	1	2	3	4	5	Recall (%)
1: Animal	155	14	37	37	7	62
2: Car	10	228	3	4	5	91
3: Flower	35	15	150	41	9	60
4: Food	43	24	40	135	8	54
5: Musical instrument	12	128	10	14	86	34
Precision (%)	61	56	63	58	75	60.3

Table 5. Confusion matrix of 5-class classification by PCA-SIFT

	1	2	3	4	5	Recall (%)
1: Animal	166	15	46	17	6	66
2: Car	14	216	6	4	10	86
3: Flower	44	20	163	20	3	65
4: Food	73	28	64	79	6	32
5: Musical instrument	27	107	19	6	91	36
Precision (%)	51	56	55	63	78	57.2

Table 6. Confusion matrix of 5-class classification by BoF

	1	2	3	4	5	Recall (%)
1: Animal	109	23	71	17	30	43.6
2: Car	4	203	13	0	30	81.2
3: Flower	7	19	197	3	24	78.8
4: Food	16	32	89	73	40	29.2
5: Musical instrument	5	121	16	4	104	41.6
Precision (%)	77	51	51	75	46	54.9

Table 7. Confusion matrix of 25-class classification by SIFT

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Recall(%)
Animal	1: Cat	7	4	5	6	3	2	0	2	1	0	1	0	5	3	3	1	2	3	0	0	0	0	1	0	1	14
	2: Dog	6	2	0	3	2	1	0	0	1	5	0	3	7	4	3	4	3	0	0	2	0	1	1	0	2	4
	3: Elephant	2	3	9	7	3	2	0	1	0	0	4	2	13	0	0	0	1	1	0	0	0	1	1	0	0	18
	4: Lion	2	1	1	21	6	0	0	0	0	0	2	3	8	0	1	1	1	3	0	0	0	0	0	0	0	42
	5: Tiger	3	0	0	4	39	1	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	78
Car	6: Impreza	0	1	0	2	0	21	3	6	6	6	1	0	1	0	0	0	0	0	1	0	0	0	0	2	0	42
	7: Lexus	0	0	0	0	0	13	12	7	6	7	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	24
	8: Odyssey	0	0	0	0	1	10	8	7	7	12	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	14
	9: Pajero	0	0	0	0	0	7	5	7	16	14	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	32
	10: Prius	1	1	2	0	0	5	6	8	5	15	1	0	0	0	0	0	1	0	1	0	0	0	1	2	1	30
Flower	11: Cosmos	0	0	1	1	0	1	0	1	0	0	28	3	1	4	2	0	0	4	0	1	1	0	0	1	1	56
	12: Dandelion	1	0	0	4	0	0	0	1	0	0	9	22	9	0	0	1	0	1	0	0	0	2	0	0	0	44
	13: Lavender	0	0	2	4	2	0	0	0	0	0	6	0	30	0	0	2	0	2	0	0	0	0	0	2	0	60
	14: Lily	4	1	0	1	0	2	0	1	0	0	6	2	2	12	7	3	0	4	1	2	0	0	0	2	0	24
	15: Rose	0	1	0	1	2	0	0	1	0	0	4	0	3	4	25	3	2	1	0	1	0	1	0	0	1	50
Food	16: Cake	2	1	2	2	0	0	3	3	2	1	5	0	4	1	2	6	5	4	1	0	0	0	0	3	3	12
	17: Hamburger	2	4	1	4	1	0	0	0	3	0	3	0	1	2	4	9	4	4	3	3	0	1	0	0	1	8
	18: Pizza	3	3	0	5	1	0	0	0	0	0	3	0	4	0	3	3	2	20	2	0	0	0	0	0	1	40
	19: Ramen	1	0	3	1	1	0	2	2	0	1	2	1	2	2	3	9	3	5	9	2	0	0	0	1	0	18
	20: Sushi	0	4	0	1	1	1	1	1	0	1	3	0	1	3	4	13	3	2	1	6	0	1	0	0	3	12
Musical instrument	21: Dram	3	1	1	1	0	3	10	3	2	3	2	1	1	0	3	2	4	3	2	0	0	0	2	1	2	0
	22: Flute	0	0	0	1	0	7	11	4	4	2	0	0	0	0	2	1	2	0	3	0	1	5	0	5	2	10
	23: Guitar	1	1	0	0	0	3	0	2	2	5	0	0	0	0	0	0	0	1	0	1	0	1	29	1	3	58
	24: Piano	0	0	0	0	0	0	4	3	3	3	0	0	0	0	0	0	0	0	1	0	0	0	1	35	0	70
	25: Violin	0	0	1	1	0	5	4	5	1	2	0	0	0	0	0	2	0	0	0	0	0	0	1	2	26	52
	Precision (%)	18	7	32	30	63	25	17	11	27	19	35	59	32	33	40	10	11	33	33	32	0	38	76	60	55	32.5

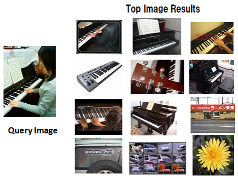


Table 8. Images classified as "piano" correctly.

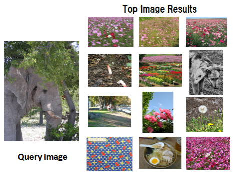


Table 9. Images classified as "elephant" incorrectly.

objects. The identification to car models was difficult, but recognition of "car" which is the upper class of car models was successfully recognized to some extent. It is thought that recognition was possible without being affected by noise features too much, because all pianos have keyboards the shape of which are characteristic, and "pianos" and "guitars" rarely appear in images of other categories. Some "piano" images which are retrieved as similar images to the query image are shown in Figure 8. Some wrong images were selected as similar images, but the correctly class was chosen for all the k.

In case of 25-category classification, we obtained the good recall 78% for "tiger" class, 70% for "piano" class and 58% for "guitar" class, since they have characteristic local features such as striped patterns of tigers, black-and-white strips of pianos and strings of guitar. In case of 5-category classification, we obtained the precision 56%, but obtained the good recall 91% for "car" class.

However, features in the "animal" class was influenced by features in the flower class such as "the trees and plants" which are background features. So we understood it was problem that handling about the features that common to multiple classes.

On the other hand,the class classification of the food and animal were difficult, cosmos had gotten to be pulled the other flowers only see the flower class. In case of animal recognition,most animal images are tend to drawn out in the flower class. It is thought that it has been recognized to flower class so that many of animal images include grasses and trees such as information in the background. In case of an experiment only in an animal class,such "trees and plants" features are scattered in an animal class so these features are not concerned possibly with many results. However,it is thought that became noise features affecting results greatly because there is a flower class by this experiment. Figure 9 is some examples that has been voted as elephant images. These images are selected by the local textures extracted from "tree" regions. It is necessary as possible as to reduce the vote of features would come out across a number of classes.

In addition in the case of food classification, it seems that the change of the image in the class is too big and was not able to express the food class. It is necessary to collect more images for the class where a change is big.

6 Conclusions

In this paper, we examined if generic object recognition can be performed by using specific object recognition methods with a large number of sample images. In the experiments, we used two kinds of the common methods for specific object recognition. One was matching SIFT features, and the other was matching visual words. For the twenty-five-category and five-category dataset, we obtained 32.5% and 60.1% for classification rate with SIFT feature matching, respectively. These results are almost equivalent to the results by the generic object recognition method which employs standard bag-of-features and SVM, which shows that generic image recognition with local feature matching is one of the promising future direction.

As future work, first of all, we plan to make more large-scale experiments. Due to memory limitation, it is impossible to add training images anymore on a single computer. On the other hand, by using parallel computing, we can use as much memory as we like. We will implement this system over a large-scale PC cluster system for a large amount of memory and speeding up. In addition, it is expected to be effective to select informative feature points and to register only selected features to a kd-tree or an inverted index. We use only discriminative features for image categorization. For examples, we should not use features coming from grassy plains on the background for recognizing tiger, and should use only features extracted from the foreground regions of tigers themselves. It is a possible solution to remove features having many similar ones over many categories by searching for similar local features among local features extracted from training images over all the categories.

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
2. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. of IEEE International Conference on Computer Vision. Volume 2. (2003) 1470–1477
3. Arya, S., Mount, D., Netanyahu, N., Silverman, R., Wu, A.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of the ACM **45** (1998) 891–923
4. Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proc. of the thirtieth annual ACM Symposium on Theory of Computing. (1998) 604–613
5. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: Proc. of IEEE Computer Vision and Pattern Recognition. Volume 2. (2004)
6. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proc. of IEEE Computer Vision and Pattern Recognition. Volume 3613. (2007) 1575–1589
7. Zheng, Y., Zhao, M., Song, Y., Adam, H., Buddemeier, U., Bissacco, A., Brucher, F., Chua, T., Neven, H.: Tour the world: building a web-scale landmark recognition engine. In: Proc. of IEEE International Conference on Computer Vision. (2009)
8. Gammeter, S., Bossard, L., Quack, T., Van-Gool, L.: I know what you did last summer: object-level auto-annotation of holiday snaps. In: Proc. of IEEE International Conference on Computer Vision. (2009)
9. Torralba, A., Fergus, R., Freeman, W., MIT, C.: 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **30** (2008) 1958–1970
10. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: Proc. of ECCV Workshop on Statistical Learning in Computer Vision. (2004) 59–74
11. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proc. of IEEE Computer Vision and Pattern Recognition. (2006)