Recognition of Object Categories in Realistic Scenes

[1]Ruddy Suhatril
[2]Adang Suhendra
[3]Lew F.C. Lew Yan Voon
[4]Eric Fauvet

[1]Gunadarma University(ruddyjs@staff.gunadarma.ac.id)
[2]Gunadarma University(adang@staff.gunadarma.ac.id)
[3]Universite de Bourgogne
[4]Universite de Bourgogne

Abstract

Classification of images maps the image content into a certain semantic term such as categories, domain, object. Image classification should be able automatically check the existence of certain object (e.g. car, animal, and scene) in the image content. This task is still challenging in computer vision since we have to deal with the realistic image. The objective of this works is to discover the image classification methods by mixturing the existing techniques with the aim of the best results in classification. In this work, we implemented sparse coding method with spatial pyramid matching to classify the images. Beside gray SIFT, four SIFT color descriptors were also used as a local descriptor. Linear Super Vector Machine (SVM) is conducted for training and testing the images. The result of this work has shown that color descriptors improve significantly the classification rate compared to gray SIFT.

1 Introduction

Image classification is a technique to classify the image into such semantic term such as categories, domain, object and other semantics. In other word, image classification should be able to recognize the object inside the image based on a semantic, and automatically check the existence of the desired object.

In computer vision, many scientists have persued to solve image classification problems since more than half century. The one of the problems is the severity of possible circumferences when images are taken such as unsuitable illumination conditions, occlusions, poses. Such circumferences tend to increase the difficulty of the classification. The objectives of this work are to develop the image classification method by mixturing the existing techniques to give the best results in classification.

2 Problem Definition

The fact that image classification should be able to work with realistic image makes this task is still challenging in computer vision. Automated image classification cannot cope with the wide variety of realistic images of the same object due to acquisition conditions. Therefore, in computer vision the solutions of those problems should be discovered. Problems of realistic images are as follow:

- Illumination.

  In the real world, the pictures are taken under different illumination conditions; it could be from nature (e.g. sun, moon) or from artificial light source (e.g. car light, camera flash, etc). For image classification, varying illumination conditions increase the difficulty of the task since the object may change.

- Intra-class variability

  This is one of the most difficult tasks to overcome, since the problem is caused by the wide variation inside the object class. Inside the same class, the shape or the size of object could be varied.

- Inter-class variability

  Due to the similarity properties of the objects, where two or more objects have the same properties. For example bicycle and motorbike, where they have the same number of tire and the same shape in general.

- Rotation and Occlusion.

  Same as scale variation problem, rotation and occlusion problem could also make an object

difficult to be classified. A robust method is needed since the method should be able to classify the object even when it is partially visible.

- Scale.

  The pose of the object in the image result in size variations.

Figure 1: Image classification task should be able to classify the image into their high semantic category.

3 State of the Art

The state of the art in image classification methods will be briefly described. It will present about Bag of Words model as a general framework and its implementation including feature detection, feature descriptor, codebook generation and classifier which are mostly used in this work.

3.1 Bag of Words

Bag of Words (BoW) method has proven to be an effective way to classify text document [8]. The idea of bag of words is that the classification is made based on the occurrences of words in the text document.

In computer vision, the BOW method has been adopted for image classification [2]. In term of image, the word is replaced by a visual word. At the end, the occurrences of the visual words are represented by a histogram [2].

In general, a bag of words method can be described as follows:

1. *Image Representation*. In BoW model, image is treated as a document where consisting of visual words. Each visual word is represented as a feature. There are two main steps which are: feature detection and feature descriptor. A feature detection aim is to detect the sample point so called interest point where the local feature descriptor will be utilized.

2. *Developing the codebook or Codebook Generation*. The codebook is a collection of visual words that will be used to train and classify the image. Unlike text document, there is no such fixed dictionary in the image. Moreover, the images are taken under various conditions such as angle, size, quality and etc. Therefore this step is critical since we have to develop the dictionary that contains a visual word that covers the entire image set.

3. *Classification*. The idea of this step is to produce classifiers that provide a rule that can be used to classify a novel image. This process consists of learning and recognition phase.

3.2 Feature Detection

There are two approaches in feature detection:

1. *Dense sampling*. This method samples the image uniformly on the entire image. In dense sampling, the image is uniformly sampled using a grid. It has two parameters which are size of the region and step between regions.

2. *Sparse sampling*. The idea is to detect the interest point. The interest point could be anything that distinguishes the object such as corner, blob, line or any other properties of the object. Several methods can be used to obtain the interest point such as Harris Corner Detector, Laplacian of Gaussian, Difference of Gaussian, and Harris-Laplace.

3.3 Feature Descriptor

Instead of working with the real image patches, it is more efficient (i.e. low memory consumption) to have a uniform descriptor to describe their properties. A good feature descriptor should be distinctive, compact, efficient and robust. In this paper, Scale-invariant Feature Transform (SIFT) and its variance will be presented.

3.4 Scale-invariant Feature Transform (SIFT)

It is the most widely used feature descriptor in image classification or object detection [12, 6, 14]. Mikolajczyk and Schmid [10] have made an evaluation on local key descriptor. It showed that SIFT gives better result that any other descriptor. It was proposed by Lowe[9].

There are four main steps in SIFT algorithm:

1. *Scale space extrema detection*. This step is performed to identify interest point which invariant to scale and orientation. Difference-of-Gaussian is used to determine the point and the scale.

2. *Keypoint localization*. This step measures the keypoint stability. The unstable point is defined as a point with low contrast and a point that is positioned along an edge. Thus, the unstable points are eliminated.

3. *Orientation assignment*. It is necessary to achieve rotation invariance. First, compute the gradient and magnitude and orientation at selected scale. Then, a weighted orientation histogram is formed. The peak of histogram is defined as the keypoint orientation and any other peaks within 80% of the peak are defined as additional keypoint orientations.

4. *Keypoint descriptor*. The descriptor is built as follows: A 8-bin histogram is made by accumulating orientations of samples weighted by their magnitude. By considering each 4x4 array histogram with 8 bins each; the descriptor is formed by 4x4x8 = 128 element descriptor vectors.

It is invariant to scale and rotation, but it is not fully invariant to change of illumination and change of 3D viewpoint.

Since it was well known as a distinctive descriptor, several modifications based on Lowe's work have been proposed. Yan Ke et. al. proposed a modification of SIFT called PCASIFT [5]. It is well known that PCA is a dimension reduction technique, therefore the idea of PCA-SIFT is to achieve more distinctive and compact descriptor by applying PCA on the normalize gradient patch.

Another extension of SIFT was proposed by Mikolajczyk and Schmid [10]. It is called GLOH (Gradient Location and Orientation Histogram). This descriptor uses more region histogram compared to SIFT, and at the end the descriptor size is reduced. PCA is also employed to reduce the dimension size.

3.5 Color SIFT Descriptors

Sande et. al. evaluated color descriptor using SIFT [12]. 4 color descriptors are recommended as they achieved best result in its evaluation. It was tested with PASCAL VOC 2008 dataset. These 4 methods which are: Opponent, C-SIFT, Transform SIFT and rgSIFT are presented.

1. Opponent SIFT: In the opponent color space, O_3 channel contains intensity information. Color information is stored in O_1 and O_2. The opponent color space is defined as follows:

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix} \quad (1)$$

Suppose the values in all layers are equal, the offset will cancel due to subtraction in O_1 and O_2 layer (e.g. a white light source). Thus, based on their photometric analysis, layers O_1 and O_2 are shift invariant. SIFT descriptor is utilized in each layer and concatenated; therefore the size of its descriptor is $128 \times 3 = 384$. Normalization in SIFT descriptor makes them invariant to intensity changes.

2. Transformed Color SIFT: The transformed color distribution is defined as follows:

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} \frac{R-\mu_R}{\sigma_R} \\ \frac{G-\mu_G}{\sigma_G} \\ \frac{B-\mu_B}{\sigma_B} \end{pmatrix} \quad (2)$$

where μ_C the mean and σ_C the standard deviation of distribution in $_C$ layer. Each value is calculated independently in the desired area, patches or image. At the end each layer has $\mu = 0$ and $\sigma = 1$. SIFT descriptor is computed for every layer after normalization. It is invariant to scale, shift and light color changes.

3. rgSIFT: rg color model is calculated as normalized RGB layer. The color information in the image is described by its chromatic component (r and g). It is defined as follow:

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} \frac{R}{R+G+B} \\ \frac{G}{R+G+B} \\ \frac{B}{R+G+B} \end{pmatrix} \quad (3)$$

Normalization makes them scale-invariant. It is also invariant to light intensity changes, shadows and shading. The descriptor is applied in each layer.

4. C-SIFT: [1] proposes the C-invariant which is added to SIFT descriptor. The invariant is obtained from normalization of O_1 and O_2 with intensities in layer O_3 in opponent color space. It is scale invariant with respect to light intensity but not shift invariance.

3.6 Codebook Generation

The visual words are obtained in this phase. The idea is how to get the distinctive word by quantizing the feature descriptor into a defined number of words. There are several coding schemes such as vector quantization, sparse coding, and gaussian mixture model. In this paper only Sparse Coding is presented.

Sparse Coding is inspired by mammalian brain that contains million of neuron. The information is represented in their brain as a pattern to activate those neurons. Sparse Coding provides an algorithm to represent a given stimuli into base and

sparse representation. In codebook generation, the process to obtain visual word is also called dictionary learning. With sparse coding the dictionary learning is done by obtaining the base function of a given set of observation. Given an unlabeled input data, the base function will be learned to capture a high-perspective of it. Unlike vector quantization, sparse coding can be employed to learn an over complete dictionary where the dimension of base is greater than the input number.

In sparse coding, a representation of signal y could be described as $y \approx Dx$, where $D \epsilon R^{n \times K}$ is the base dictionary and vector $x \epsilon R^K$ contains sparse coefficients of signal y.

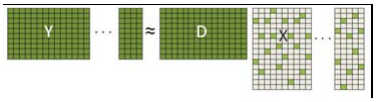

Figure 2: Sparse Coding. The original signal Y can be approximate with base (D) and its sparse coefficients (X).

3.7 Spatial Pyramid Matching

The Bag of Words model does not take into account the spatial information of the features. Moreover, spatial information of object is an important cue in image classification. In order to overcome this problem, Lazebnik et. al proposed a method called Spatial Pyramid Matching [6]. It consists in dividing the image into several level resolutions and the histogram is calculated for each cell in those levels. Then, all of them are concatenated to form a single feature descriptor. The illustration of this method can be seen in figure 3.

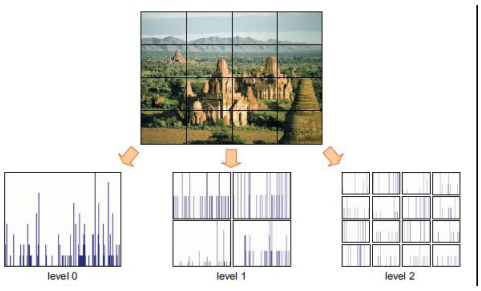

Figure 3: Spatial Pyramid Matching

3.8 Classifier

The next step in image classification is learning and recognition steps. In learning phase, the classifier is utilized to produce a rule or decision function given a set of observation. In recognition step, the decision function is applied to a novel data i.e. testing the novel data in order to determine to which class they belong.

One of the most widely used classifier in image classification is Support Vector Machine [6, 14, 7, 11, 12]. It yielded state-of-the-art result in many image classification tasks. It does not need a prior knowledge and fits with high dimensional data. Thus, it is consider as a good classifier in image classification. The idea behind this classifier is to suppose that there are two classes, each class will be separated with hyper plane; the empty area (i.e. margin) around the decision boundary is considered as an optimization criterion. The nearest vector to the margin is defined as support vector. To overcome non linear problem, the kernel is introduced to determine the boundary.

4 Methodology

In general, this work follows the same strategy as the one implemented by Jianchao Yang [14]. The method called ScSPM, It used sparse coding and linear Spatial Pyramid Matching strategy to do classification task. The gray SIFT is employed as a feature descriptor thus the sparse coding is built on it. And finally a linear SVM is applied to build the classifier.

As it is widely known that the color information is an important cue in task like classification, instead of using only grey SIFT, this work tried to use color information as feature descriptor. Therefore, 4 color SIFTs (Opponent SIFT, rgSIFT, Transformed Color SIFT, and C-SIFT) were employed in order to observe usefulness of color information incorporated with sparse coding and SPM.

In order to implement the method, the framework was developed using Matlab. The framework is detailed in figure 4. The framework contains three main parts: Dictionary learning, Training and Testing. Dictionary learning is performed to obtain the codebook. The learning and recognition process are performed in the training and testing part. Each part can be run separately except for two things, sparse coding and testing; since it requires an input from dictionary learning and training.

In Feature Extraction, dense sampling strategy and Vedaldi SIFT [13] descriptor is employed as the SIFT descriptor. An efficient sparse coding proposed by Honglak lee et. al. [7] is used since it is shown to require less computational time. They proposed algorithm called Feature-sign algorithm to obtain the sparse coefficient. Then, the base is obtained using Lagrange dual.

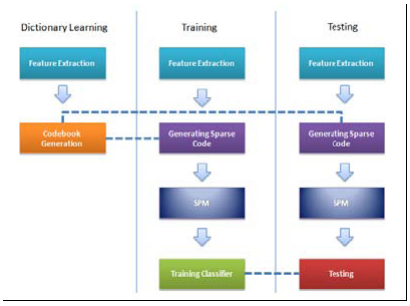

Figure 4: Implementation Framework

5 Results

The entire experiment was run on Intel Core2 Duo 2.13 GHz processor and 4 GB RAM. Two data sets were used in the experiment: Caltech [4] and Pascal VOC 2009 dataset [3]. The number of vocabulary (codeword) used in this experiment was 256 for every descriptor. The window size used for extracting the feature was 16×16 with step size was 8. Max method was implemented to pool the feature.

5.1 Caltech2

The first experiment was thus performed on the Caltech2 dataset. The number 2 stands for 2 classes of objects: airplane and camera. There are 50 images of each class.

For n=1 to 6, 5n randomly chosen training images are considered for each value of n and the classification process performed 5 times. For each experiment the mean accuracy is calculated.

The best performance is achieved by opponent SIFT. After 30 training images, it achieved 100% accuracy.

Table 1: Accuracy of Classification on Caltech2

Training No	5	10	15	20	25	30
Gray	0.676	0.688	0.726	0.740	0.744	0.765
Opponent	0.964	0.980	0.986	0.993	0.996	1
rg	0.742	0.903	0.911	0.927	0.928	0.955
Transform	0.960	0.970	0.983	0.953	0.972	0.980
C-SIFT	0.956	0.975	0.983	0.960	0.976	0.960

5.2 Caltech10

The next experiment was carried out on bigger subset of Caltech. We called it Caltech10; it is comprised of 10 classes: accordion, anchor, ant, barrel, bass, beaver, bonsai, brain, brontosaurus, and Buddha. The same procedure was still taken.

The accuracy of classification result can be seen in table VIII. As it can be observed in the result, same as the previous experiment, the color descriptors shows their strength. Opponent SIFT still achieved best performance among color descriptors, it is also noticed that rgSIFT did not perform as well as the others.

Table 2: Accuracy of Classification on Caltech10

Training No	5	10	15	20	25	30
Gray	0.2	0.22	0.24	0.25	0.27	0.27
Opponent	0.49	0.59	0.63	0.62	0.7	0.73
rg	0.36	0.43	0.47	0.5	0.53	0.58
Transform	0.49	0.57	0.61	0.63	0.67	0.68
C-SIFT	0.5	0.59	0.65	0.64	0.69	0.7

5.3 Pascal VOC 2009

This work follows Pascal competition evaluation criterion where the classification result is measured by its average precision. It is proportional with the area under Recall and Precision graph.

The result of experiment on Pascal dataset can be seen in table (3,4,5,6). And to be fair, before discussing about these results, due to the computational time of sparse coding, we were not able to run the experiment on the entire dataset. In this result, the number of training images for each class is 900 and 200 for testing images.

Although it is only a partial result (classification was not done on the entire dataset). If we observe the result, there is no superior descriptor that able to get better performance. But in general, they improved the gray SIFT in every class except for Car, Horse, and TV/monitor.

Table 3: Average Precision on Pascal VOC 2009 (1)

Feature / Class	Bus	Car	Cat	Chair	Cow
Gray	0.848	0.464	0.364	0.172	0.022
Opponent	0.295	0.302	0.357	0.290	0.108
rg	0.697	0.358	0.288	0.143	0.044
Transform	0.86	0.416	0.468	0.186	0.024
C-SIFT	0.855	0.382	0.390	0.315	0.045

Table 4: Average Precision on Pascal VOC 2009 (2)

Feature / Class	Aeroplane	Bicycle	Bird	Boat	Bottle
Gray	0.109	0.108	0.213	0.108	0.117
Opponent	0.149	0.171	0.330	0.157	0.115
rg	0.337	0.116	0.231	0.282	0.256
Transform	0.338	0.119	0.291	0.148	0.134
C-SIFT	0.144	0.087	0.275	0.121	0.277

5.4 Computational Time

These tables (7, 8, 9) depict the time needed to generate codebooks for each descriptor and the av-

Table 5: Average Precision on Pascal VOC 2009 (3)

Feature / Class	Dining Table	Dog	Horse	M.bike	Person
Gray	0.18	0.113	0.075	0.091	0.646
Opponent	0.161	0.117	0.046	0.076	0.610
rg	0.029	0.247	0.043	0.082	0.656
Transform	0.418	0.123	0.044	0.065	0.612
C-SIFT	0.425	0.291	0.041	0.403	0.611

Table 6: Average Precision on Pascal VOC 2009 (4)

Feature / Class	P.plane	Sheep	Sofa	Train	TV/Monitor
Gray	0.038	0.423	0.164	0.184	0.621
Opponent	0.039	0.105	0.119	0.356	0.374
rg	0.035	0.077	0.07	0.068	0.35
Transform	0.041	0.148	0.122	0.152	0.543
C-SIFT	0.036	0.424	0.209	0.065	0.357

erage time needed to generate the sparse coding per image (for each local descriptor). The sparse coding time is including the spatial pyramid matching method inside. But the most time was taken by sparse coding. The sparse coding is generated once per image, and for each cell in SPM, the descriptor code is pooled based on their spatial location. These tables proves the computational expensive of sparse coding.

Table 7: Computational Time on Caltech2 Dataset

	D. learning (hours)	Sparse Coding (seconds)
Gray	15.8	296
Opponent	21.45	421
rg	21.34	443
Transform	21.61	454
C-SIFT	20.99	451

6 Conclusions and Future Works

6.1 Conclusions

In the experiment, the color descriptor combined with sparse coding and spatial pyramid matching is indicating a better performance. The results were improved instead of just using the original SIFT (gray SIFT).

In Caltech dataset, the opponent SIFT has indicated the best performance compare to another color descriptor. The result is also proportional with the number of training for each class. Among color descriptors, rgSIFT performed less well than the others.

In Pascal VOC 2009 dataset, each color descriptor has indicated different performance in each class.

Table 8: Computational Time on Caltech10 Dataset

	D. learning (hours)	Sparse Coding (seconds)
Gray	63.91	420
Opponent	81.56	630
rg	80.07	621
Transform	83.52	625
C-SIFT	83.25	653

Table 9: Computational Time on Subset of Pascal VOC 2009

	D. learning (hours)	Sparse Coding (seconds)
Gray	64.47	499
Opponent	100.02	658
rg	75.67	639
Transform	74.91	648
C-SIFT	78.68	673

In other word, there is no superior color descriptor that achieve good result in every class. But in general, color descriptor achieved better performance than gray SIFT.

The computational time of sparse coding can be considered as the weakness of this method. In addition, the sizes of color descriptors are bigger than the gray one; it makes this situation getting worse. As we can see in the experiment, the color descriptor was taking longer time than the gray one.

6.2 Future Works

Due to the limitation of time and computational time, we did not succeed to evaluate the entire dataset. In order to see the ”real” performance of this work, it is needed to run the experiment in the whole dataset and with several parameters that have been set such as various numbers of codebooks, various window sizes and steps. Computational time of sparse coding is proportional to the

7 Acknowledgments

This work was conducted in Laboratoire Electronique, Informatique et Image, Le Creusot, France.

References

[1] Alaa E. Abdel-Hakim and Aly A. Farag. Csift: A sift descriptor with color invariant characteristics. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1978–1983, Washington, DC, USA, 2006. IEEE Computer Society.

[2] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results.

[4] Li Fei-Fei, Rod Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. page 178, 2004.

[5] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. pages 506–513, 2004.

[6] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. volume 2, pages 2169–2178, 2006.

[7] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In *In NIPS*, pages 801–808. NIPS, 2007.

[8] David D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. pages 4–15. Springer Verlag, 1998.

[9] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[10] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[11] Fahad Shahbaz Khan, Joost van de Weijer, and M Vanrell. Top-down color attention for object recognition. In *IEEE Conference on Computer Vision (ICCV'09)*, 2009.

[12] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(in press), 2010.

[13] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

[14] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801. IEEE, 2009.