國立臺灣大學電機資訊學院資訊工程學研究所

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

考慮背景資訊之一般物體辨識

Utilizing Background Information for Generic Object Recognition

謝毓庭

Hsieh, Yu-Ting

指導教授：莊永裕 博士

Advisor: Yung-Yu Chuang, Ph.D.

中華民國九十七年六月

June, 2008

Acknowledgments

This work can not be finished with help from many people. My advisor Dr. Chuang, also as a friend, gave me a chance to decide my research area and help me keep working on the same direction. Dr. Lee-Feng Chien introduced me the picture of research in a short but very helpful chat, that gave me a good beginning of the whole work. Rong-En gave me many intuition of SVM like a kind consultant, which saved my time of making mistake greatly. Tz-Huan shared work with me to settle machine problem in the lab, which relieved my load a lot. And luckily I got a lot of friends to share pressure and happiness during these two years, Yin, Saul, Shiao, Yummy, Tsung-Yu, CCK and Chia-Kai. Also, grateful thanks to my family for giving my financial support, which kept me from being worry about life. Last but not least, I would like to devote this work to Li-Yin, who shared my good and bad in the life.

Abstract

This thesis introduces background information to generic object recognition problem to increase the accuracy. Most of works do not divide images to foreground and background part, or only utilize foreground information. In this thesis, we tried to leverage background information to help object recognition.

A region of interest (ROI) detector is used to find the foreground object in images. Focusing on foreground object can reduce noisy features from unrelevant background region. Furthermore, the complement area of ROI can be considered as background context. Since objects in a category usually appear in specific context, we will show that adding background clue can improve the recognition accuracy in our experiment.

Another challenge problem is how to use different signals together. We compared several methods of feature fusion for machine learning using SVM. Experiment result shows how well these methods can achieve and whether background information benefit them.

摘要

此篇論文主要的研究，是將影像的背景資訊加入一般物體辨識的流程，以提升其準確率。目前大部分的研究並未將影像的前景物體與背景分開考慮，或者只利用前景的資訊。在這一篇論文中，我們試著加入背景資訊以提過一般物體辨別的準確率。

我們使用一個偵測使用者感興趣區域（Region of Interest）的方法來將影像前景的物體偵測出來。更進一步地，使用者感興趣區域周圍的背景資訊可以用來加強物體識別。由於同一個種類的物體通常會出現在某些特定的場合，我們將由實驗說明加入背景資訊對一般物體辨識率的提升。

另一個很有挑戰性的問題是如果將不同的影像特徵合併使用。我們比較了幾個不同的方法在支持向量機（Support Vector Machine）上的表現。實驗結果顯示這些方法在這個問題上的好壞，與他們能否有效運用背景資訊來加提升辨識率。

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image understanding is not a new research topic in computer vision, but it starts to attract more attention from researchers in the pass few years. Knowledge in textual documents can be searched accurately nowadays, but searching knowledge in images still has a long way to go. Many researchers aim to specific types of knowledge from images, such as what or who is in a photo, but they are all very challenging problems to be solved.

For example, face recognition researchers attempt to recognize who are in photos; surveillance is targeting to find out suspicious people or events for warning human beings. These kinds of work focus on specific type of images, and actually can be very helpful to our life.

In this thesis, instead of focusing on any specific kinds of objects, we tried to recognize what kind of object appeared in an image. A generic approach should be developed, instead of any special procedure to a specific cateogry. We refer the problem as *generic object detection*.

1.1 Background

Figure 1.1: Object recognition problem aims to figure out what is in images. These are example images of Caltech101 dataset, which contains 101 categories of generic objects.

In contrast to identifying specific types of objects in images, it is more difficult for computers to recognize generic objects in images. Although such a task is accurate and reliable to human beings, it is surprisingly difficult for computers to archive the same task.

In some sense, computer scientists are aiming to design a procedure for machine to digest low level pixels and reduce to high level knowledge. One of the key point is how to transform pixels to low level features. A good feature design can be more distinguishable between different kinds of objects. SIFT[14], orientation histogram and texture, are examples of popular features.

Once features are collected, ones can construct a model to separate features from different categories into different groups. Features can mostly be represented by a high dimension vector, thus the most popular classifier for this problem is *support vector machine* (SVM). SVM can find a hyper-plane that split different categories into two parts for training data, then prediction can be made based on what side is the vector on in testing phase.

Nevertheless, most people believe that there is no universal feature that can help recognize every category, so that using different kinds of features together is one direct solution. Since SVM works as a classifier of a single feature, one might need to fuse different classifiers together. That introduces a new problem of how to fuse features to archive better accuracy, and we have some comparisons of different methods in later chapter.

In most object recognition approaches, features were extracted from whole images, which includes background context. Background context are often noisy to the problem since the purpose is to know the type of foreground object. Thus *region of interest* (ROI) was introduced to isolated foreground objects from images, so that only features of foreground object will be collected.

In addition to ROI region, we collected the complement area of ROI as background context. Since many kinds of objects usually appear in some specific context, we tried to leverage this clue to improve recognition accuracy.

There are several popular published image datasets, including Caltech101/256, Pascal VOC. For evaluation, we used Caltech101 collected by Fei-Fei et al. [5] in our experiment, which provides a unified framework for performance evaluation and comparison. In Caltech dataset, all images contain only one class of object.

1.2 Contribution

There are several contributions in this thesis. First, we introduced an efficient vision based ROI detection system to this problem. Second, we additionally tried to leverage background information to help foreground object recognition. Third, different trials of fusion method have been compared.

We will introduce related work first in chapter 2, and then feature representation used in this thesis. ROI extraction and combination with feature extraction are followed in chapter 4. Learning strategy for feature fusion is then discussed in chapter 5. Experiment configuration is shown in chapter 6, and the conclusion in chapter 7.

Chapter 2

Related Work

We split related work to several categories, including feature, region of interest (ROI) and feature fusion, each will be introduced in later section.

2.1 Feature

Currently, most popular and effective features are based on *bag of feature* (BOF) model[11, 8, 1]. BOF model describes an image as a group of order-less, simplified representation of image patches, and no geometric relation between patches are considered. By measuring co-occurrence of visual words in categories, machines can learn a model to tell the most likely category of an object in given an image.

BOF model does not regard spatial relation between patches. Some part-based models try to model objects by compositing components with spatial relation, but it often took longer time to measure spatial similarity, and the accuracy does not usually compete with BOF model[6, 4].

One interest variance of BOF is called *pyramid of histogram of visual word* (PHOW), which collects visual word histograms in different grid size. Images are divided from 1×1 to $n \times n$ grids, like a n level pyramid. Histogram intersection in smaller grids contribute more than larger grids. It has been demonstrated to be very effective in many work[8, 11, 2, 1].

There is another similar pyramid model called *pyramid of histogram of orientation gradient* (PHOG). It accumulates orientation information in different grid size as pyramid of histogram, combining with PHOW feature can further increase the recognition accuracy[2].

Unlike those designed features, Fidler et al. [7] tried to borrow idea of visual processing system of human brain from cognitive psychology research, but it's not easy to be further developed for lacking of knowledge of brain. Also, the same logic might not be suitable since our brains are not Turing machines.

2.2 ROI

ROI was recently introduced to object recognition problem by Bosch et al. for filtering background noise, and has been shown helpful in this problem[1]. Nevertheless, the proposed method of ROI detection needs pair-wise comparison for all training images in the same category. Besides, during testing phase, every image needs to compare to all training images in all categories to find the ROI. Therefore, the whole process can take very long computation time when number of images and categories increase, so that the scalibility is quite limited.

Liu et al. has developed an learning based ROI detection system. They created an image database with human labeled ground truth of ROI, each image contains exact one object. A few weightings of the ROI detector need to be learned from some training images based on some visual clue, then the object in each testing image can be detected efficiently.

2.3 Feature Fusion

None of feature along is able to differentiate every kind of categories. Different types of features are often used together since different categories can be better described by different features. But how to combine features is still a very challenge problem. A naive way is to performed exhaustive search to find weighting of different features[2].

Lin et al. [12] introduced ensemble kernel to fuse several types of kernels to make them as similar as possible to a target kernel, but the design keeps histogram intersection kernel used in PHOW and PHOG unapplicable. Detail will be mentioned in chapter 5.

Super kernel[16] targets to build an upper level SVM to combine outputs of different SVM models. Prediction score of classifiers are transformed to probability vector. Vectors can then be learned or predicted again in super level SVM. This method is compared in our experiment.

Chapter 3

Feature Extraction

Pyramid feature[8] has been demonstrated to be an effective feature representation, especially the two variances PHOW (pyramid of histogram of visual word)[11] and PHOG (pyramid of histogram of orientation gradient)[2]. In our work, these two features are further applied to foreground sub-image isolated by *region of interest* (ROI)[1] bounding box.

In PHOW and PHOG features, images are subdivided to pyramid composed by l-by-l grid where l is from 0 to L where L is the max level of pyramid. Example is shown in figure 3.1. For both features, every grid has a histogram in every level, and that becomes a histogram of pyramid. Similarity between two images is defined by accumulated histogram intersection of corresponding grids in two histograms of pyramids, with different weighting for different levels. The similarity function is called *pyramid match kernel*.

[1] ROI will be introduced in chapter 4

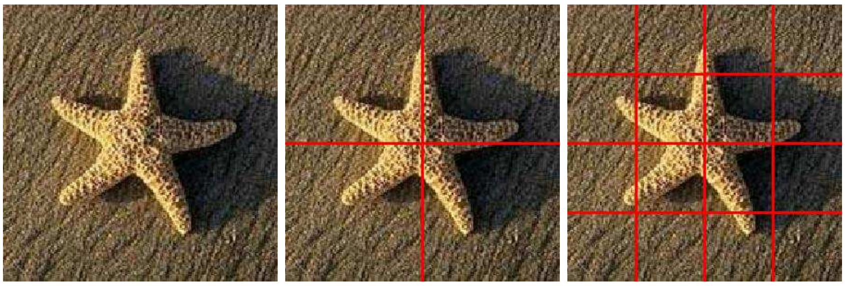


Figure 3.1: Visualization of pyramid from $l = 0$ to $l = 2$

In this chapter, we will first introduce histogram composition for PHOW and PHOG features, and then pyramid match kernel for these two features.

3.1 Grid of Pyramid

We need to define how the histogram is accumulated in each grid. For PHOW feature, visual words are simply counted in every grid. A visual word is usually defined by first collecting SIFT features[14] from training images, and then quantizing SIFT vectors into fixed number of groups. The numbers of different visual words in a grid compose the visual word histogram of the grid.

For PHOG feature, Sobel mask is used to retrieve an orientation map from an image. In practical implementation, orientations are usually uniformly divided to several slices. Orientation on a pixel is then projected to some slice, and corresponding intensity is added to the slice. At the end, the feature will be a vector of accumulated quantity of every orientation slice.

At this point, we have defined the pyramid features. Pyramid match kernel will be introduced in next section.

3.2 Pyramid of Histogram

For PHOW, histogram of visual word is counted for the i-th grid at level l, which can be considered as a pyramid of histogram. PHOG with histogram of orientation is defined similarly. Additionally, histogram intersection function of two images X and Y is defined as

$$I(H_X^l, H_Y^l) = \sum_{i=1}^{2^{dl}} \min(H_X^l(i), H_Y^l(i))$$

where d is the dimension of histogram. $I(H_X^l, H_Y^l)$ will be abbreviated to I^l in the following.

3.3 Pyramid Match Kernel

Thought there exists many popular SVM kernels (e.g. radial basis function and linear kernel), they are not able to fully utilize pyramid features. Pyramid match kernel is able to better leverage pyramid feature since it has prior knowledge of pyramid features.

Pyramid match kernel gives more weighting to histogram intersection of grid at finer level. Specifically, the kernel sums histogram intersection of levels inversely proportional to the grid number of the level. Following is the definition of pyramid match kernel of PHOW[11]:

$$\begin{aligned}\kappa_1^L(X,Y) &= I^L + \sum_{l=0}^{L-1} \frac{1}{2^{L-l}}(I^l - I^{l+1}) \\ &= \frac{1}{2^L} I^0 + \sum_{l=0}^{L} \frac{1}{2^{L-l+1}}(I^l)\end{aligned}$$

And also kernel used in PHOG[2]:

$$\kappa_2^L(X,Y) = \sum_{l \in L} \alpha_l d_l(X,Y)$$

α_l is the weighting of l-th level of pyramid. The weighting used in this thesis is chose from GLW experiment in [2].

An important attribute of these two kernels are positive definiteness. Histogram intersection kernel or the linear combination have been proved to be positive definite[15, 8]. Without this attribute, SVM might trap into local minimum.

After pyramid features are collected, we use SVM with pyramid match kernel to classify categories. Detail will be described in chapter 5.

3.4 Foreground Representation

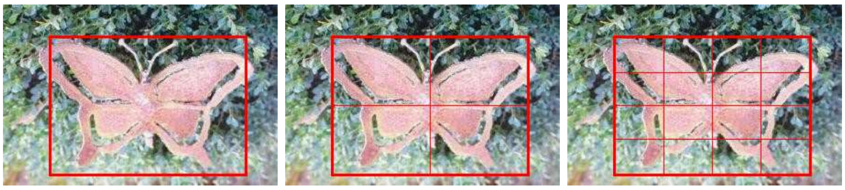


Figure 3.2: Visualization of pyramid from $L = 0$ to $L = 2$ in an ROI

Usually foreground objects do not often occupied the image totally. Focusing on the object can reduce unnecessary information and may improve the classification accuracy[1]. Figure 3.2 visualizes the pyramid feature in a given ROI.

In section 4, we will introduce an algorithm of ROI detection. Every image will have an ROI represented as a bounding box. Foreground features are extracted by treating sub-image inside the bounding box instead of the full image. In this work, PHOW and PHOG are extracted from the bounded sub-image.

Chapter 4

Region of Interest

Region of interest (ROI) of an image often refers to the attractive area for human beings. For object classification problem, focusing on the object can reduce background noise greatly, thus ROI was introduced to object recognition problem recently[1]. An ROI detector is adapted from ROI research to this work. In addition, we tried to leverage background information to help recognition of foreground objects.

4.1 ROI for Object Detection

In this section, we will talk about previously purposed algorithm of ROI detection for object recognition problem and the limitation. Then an algorithm based on visual clue will be introduced after.

4.1.1 Low-level Feature-based Exhaustive Search

Bosch et al. [1] has modeled ROI using bounding boxes specifically to images contain only one object in the middle. The method is only able to find a sub-optimal ROI due to computational complexity, and is still not efficient enough to find good ROIs.

In the proposed algorithm, every training image has a sliding window as hypothezises ROI for every category. A hypothesis ROI can be considered as a smaller image. Every ROI sub-image is compared to all other ROI sub-images of training images within fixed, centered bounding boxes. They assumed that objects are likely to appear in the middle of images overall. The hypothesis region which is most similar to all other ROI images is chosen as the ROI.

Time complexity is a serious problem. Given training images, it needs calculate pair-wise similarity. For each calculation, a sliding window is shifted to find a good match of sub-optimal. Note that to find a good ROI bounding box, different area and aspect-ratio of sliding windows need to be tried since the object may have variant size and aspect-ratio. Each sliding window moves around the image for many times, which depends on the image size. Thus the computation time to find all ROIs can be very long.

Another problems of this algorithm is the definition of ROI similarity. Similarity is calculated by adding PHOW (pyramid of histogram of visual word) and PHOG (pyramid of histogram of orientation gradient) (see chapter 3) with equal weighting, whereas different classes may have different type of effective feature. There is no prior of preference for classes, thus the preferred weighting between features are unknown.

Bosch et al. used a fixed weighting of PHOW and PHOG features in their work.

In the testing phase, a testing image needs to be searched against all training images for every category, since one doesn't know which category is it. Thus testing process is very time consuming.

4.1.2 Learning-based Detection with Visual Cue

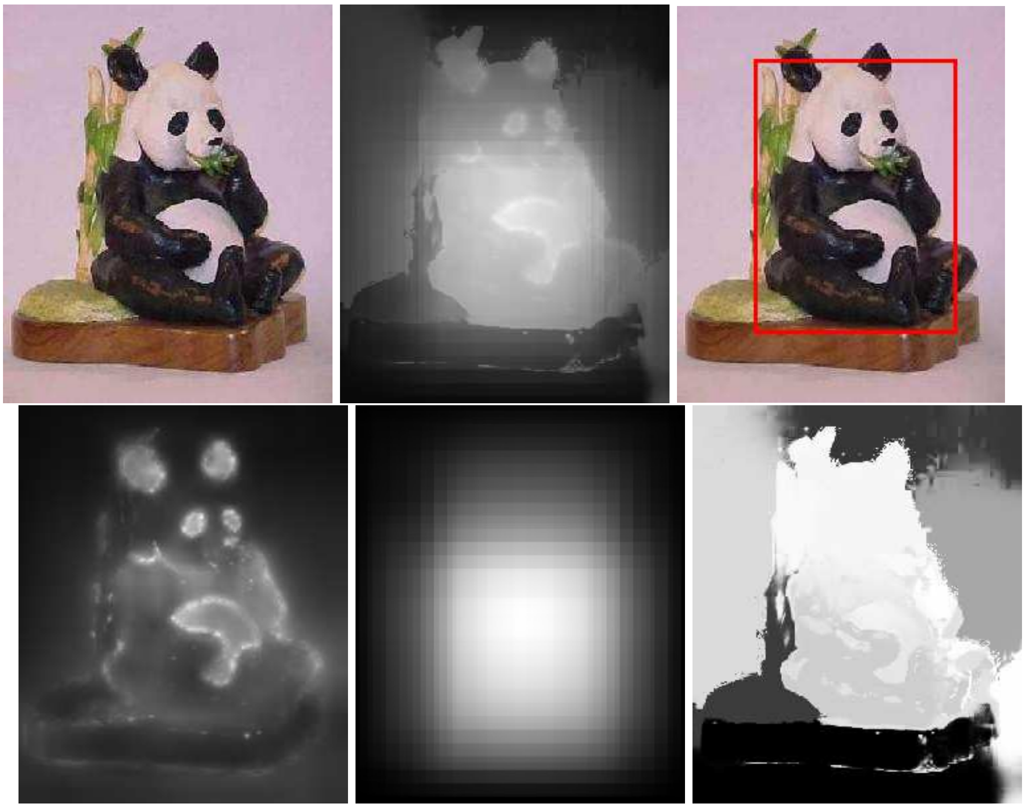


Figure 4.1: The first image is the original image, followed by the salient map and corresponding bounding box of ROI. The next three images are features, respectively multi-scale contrast, center-surround histogram and color spatial-distribution.

Instead of exhaustive search and dilemma of choosing feature types, we introduced a salient object detection method in learning approach developed by Liu et al. [13]. Given that only one object in an image, the proposed algorithm finds a salient image that indicates likelihood of location of the object. In this work, three types of images as signal is computed for each training image, respectively multi-scale contrast, center-surround histogram and color spatial-distribution, as shown in the bottom row of figure 4.1.

Three Feature Maps

Multi-scale contrast tries to focus on detail information and drop coarseness in an image. The original image is scaled down to half for several times to reduce high frequency, and *difference of Gaussian* (DoG) between levels are added together to form the final salient map. Specifically, the salient map is defined as following:

$$f_c(x, I) = \sum_{l=1}^{L} \sum_{x' \in N(x)} ||I^l(x) - I^l(x')||^2$$

where I^l is the l-th level of image and N is a window for blurring.

Center-surround histogram first defines a rectangle R in the given image, and R_S surrounds R with the same center and same area as R, and both R and R_S together form a bigger rectangle similar to R but with twice of area. The idea is that the more difference between pixels in R and in R_S, the more chance R bounds the foreground. Then color histogram is collected for both regions and χ^2 distance is used to calculate the similarity. The salient map is defined as

$$f_h(x, I) \propto \sum_{\{x'|x \in R^*(x')\}} w_{xx'} \chi^2(R^*(x', R_S^*(x'))$$

where $w_{xx'} = \exp(-0.5\sigma_{x'}^{-2}||x - x'||^2)$.

Color spatial-distribution models an image by several *Gaussian mixture models* (GMMs), and then gives each pixel in salient map smaller intensity if the spatial variance of Gaussian component is big. Intuitively, it can give a background pixel lower intensity since background usually has a big spatial variance. Specifically, the definition is

$$f_s(x, I) \propto \sum_c p(c|I_x)\dot{(}1 - V(c))$$

where c is Gaussian component and $V(c)$ is spatial variance of component c.

These features are linear combined as the final salient map. To learn the combination weight, the problem was modeled using conditional random field (CRF), and learned with human labeled ROI given as bounding box in training images. A learned vector of weighting will be used in testing phase to combine three feature images. Thus, this algorithm is much efficient than exhaustive search considering time complexity for both training and testing phase.

Reduction To Binary Label

The output of this algorithm is a salient image, indicating which pixel is more confident to be the salient object. Graphcut[10] algorithm is then applied to separate foreground and background pixels. We use Graphcut to split foreground and background pixels on salient map. Graphcut is modeled as a minimization problem with the following energy function:

$$E(I) = \sum_{p \in P} |I_p - I_p^o| + \sum_{(p,q) \in N} K_{(p,q)} T(I_p \neq I_q)$$

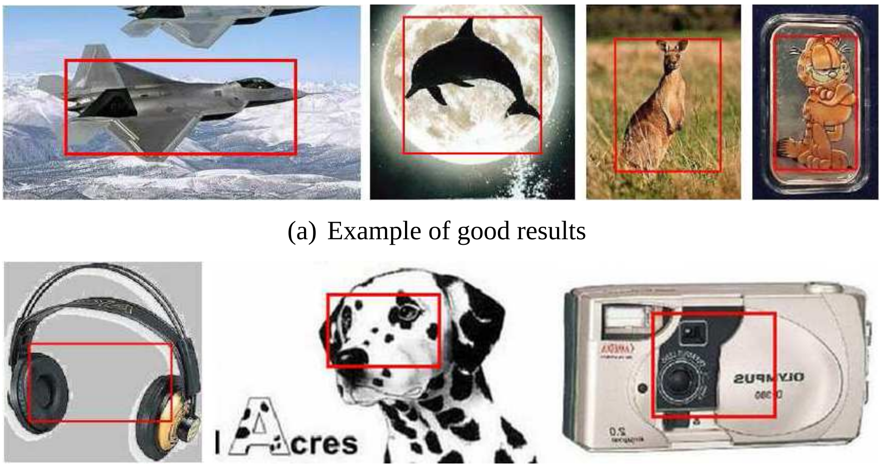
where I_p^o is the observed pixel intensity on salient map and I_p is the truth. The truth pixels here are defined by simply threshold. $K(p,q)$ is the cost of giving different label to two neighboring pixels, depending on color difference and intensity difference. Once pixels are labeled, a bounding box is used to represent ROI for simplicity in this work.

Limitation

An issue of applying this learning method is that the training images need ground truth of ROI labeling. Lacking of the ground truth in Caltech101, the same weighting from Liu et al. [13] is borrowed in our experiment. Some results are shown in figure 4.2

4.2 Apply ROI to Classification Problem

After the ROIs are found, images can be separated into foreground and background. With bounding box of detected ROI, PHOW and PHOG features can be applied in the sub-image, which has been proved helpful in recent research[1].



(a) Example of good results

(b) Example of bad results

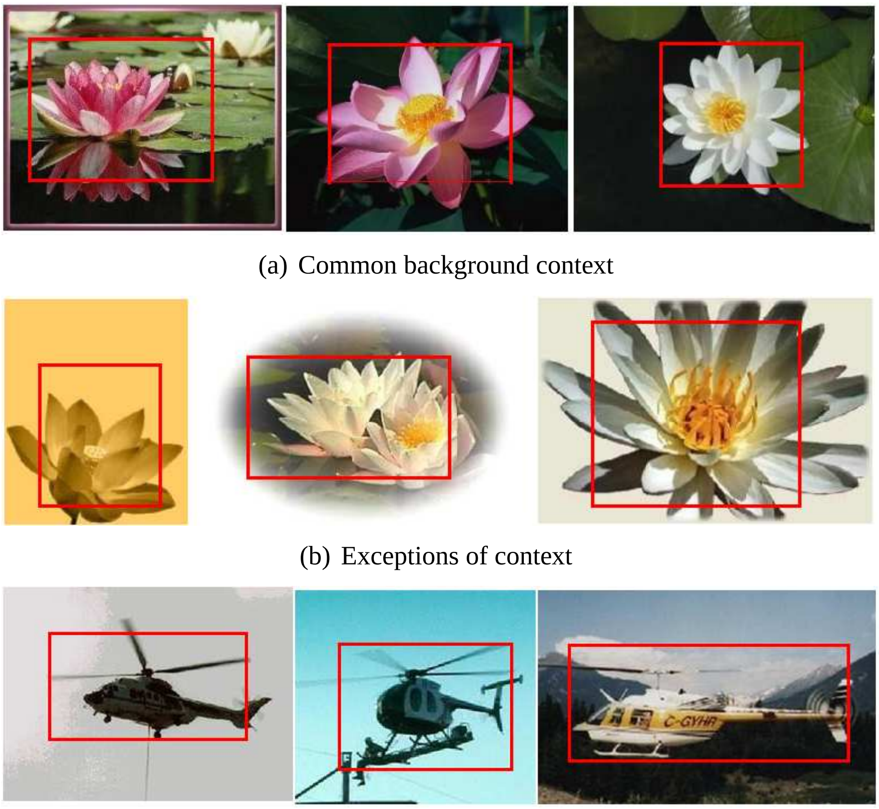
Figure 4.2: Example of good and bad results of ROI detection

4.2.1 Background Representation

Besides foreground features, remaining background can still be useful for other kinds of features. Objects of some categories often appear in similar context, which can give us a clue to better recognize foreground object. Figure 4.3(a) shows an example of water lilies. Though exceptions often exist (4.3(b)), it's still helpful to distinguish from other categories, such as helicopter in figure 4.3(c).

To describe the background context, we use BHOW (background HOW) and BHOG (background HOG) feature in our work. Both features are collected from the area outside ROI, and form a histogram of words and orientation gradient.

For background features, histogram intersection is used as kernel function for learning. Although background information may not be strong knowledge to recognize specific categories, it could be helpful to differentiate from some categories which often appear in irrelevant background context. Consequently, we use BHOW and BHOG



(a) Common background context

(b) Exceptions of context

(c) Different background in other category

Figure 4.3: Background plays an important role for object recognition. In (a), water lilies appear in similar context, while sometimes there are exceptions as shown in (b). Obviously, background context of (a) are very different to helicopter's in (c).

together with PHOW and PHOG to boost our recognition.

An upcoming problem is how to use different features together. We will introduce our solution in the next solution.

Chapter 5

Supervised Learning of Categories

Features are vectors representation of images. Based on features, a supervised learning approach tries to separate features from a class to another. State-of-the-art research shows that support vector machine (SVM) is a very efficient approach for this problem[8, 11, 2, 12].

5.1 Support Vector Machine

SVM classifies input instances based on kernel function. Specifically, given vectors (y, x) where the class label $y = 1, -1$ and x is the vector, SVM can find a hyper plane that separates vectors of two classes with a positive definite kernel function $k(x_1, x_2)$.

κ_1 and κ_2 are two kernel functions for foreground object as defined in chapter 3. Additionally, histogram intersection is the kernel function applied to background features BHOW and BHOG. Fusion method will be introduced in the later section.

Note that the kernel functions used in this thesis are histogram intersection, or linear combination of histogram intersection. Both form of kernel were proved positive-definite[15, 8].

One-versus-rest learning for different categories was performed in the experiments.

5.2 Feature Fusion

A followed up problem is how to combine multiple features together to compose the final classifier. In this section, we will compare different fusion strategies, including averaged kernel, ensemble learning and adaptive search in kernel level, and super kernel works in super level.

5.2.1 Averaged Kernel

A naive method is to simply average the kernel functions. We compared this method to others, and found its superior than some purposed methods if input features are good.

5.2.2 Ensemble Learning

Local ensemble kernel[12] attempts to find a weighting between different kernels by solving a maximization problem of *kernel alignment* $\hat{A}(K_1, K_2)$. It tries to approximate target matrix G with fused kernel matrix. Kernel alignment is defined as:

$$\hat{A}(K_1, K_2) = \frac{< K_1, K_2 >_F}{\sqrt{< K_1, K_1 >_F < K_2, K_2 >_F}}$$

$$\text{where} \quad < K_1, K_2 >_F = \sum_{i,j=1}^{l} K_1(x_i, x_j) K_2(x_i, x_j)$$

And the vector of weighting of kernels α:

$$\begin{aligned} \arg\max_{\alpha} \quad & \hat{A}(K, G) \\ \text{subject to} \quad & K = \sum_{r=1}^{M} \alpha^r K^r, \\ & trace(K) = 1 \\ & \alpha^r \geq 0, for 1 \leq r \leq M. \end{aligned}$$

In which K is the fused kernel and G is the target kernel matrix to approach:

$$G(i,j) = \begin{cases} +1 & \text{if } y_i = y_j, \\ -1 & \text{otherwise.} \end{cases}$$

In fact, Hoi et al. [9] have reduced the above problem to a quadratic programming problem. Nevertheless, the same reduction can not be applied for histogram intersection features, which has domain in $(0, 1)$ but not $(-1, 1)$ as G and make the terms in kernel alignment $\hat{A}$ unbalance. Shifting the kernel matrix from $(0, 1)$ to $(-1, 1)$ will remove the positive definiteness and could trap the optimization into local minimum.

Another possible solution is to represent cell of different class in G as 0 instead of -1, denoted by G'. But it will remove the information of negative examples. The other way to utilize both positive and negative examples could be to maximize $\hat{A}(K, G') + \hat{A}(1 - K, 1 - G')$, but the optimization problem becomes harder. Even though, we still simply searched on different weighting to evaluate the performance.

5.2.3 Adaptive Grid Search of Weighting

We will try to select parameters to find the best one which maximize overall accuracy. Basically it's selected manually with some strategy. At the end, this strategy will select a sub-optimal solution.

Features are separated into foreground and background groups at the beginning. By intuition, foreground features are stronger signals comparing to background, since similar background often appears with different foreground categories. Thus we fixed foreground features first, and then use additional background information for tuning.

In the first step, different weighting vectors for kernel fusion in foreground group are evaluated on the validation data, by exhaustively trying with some fixed weighting. For the weighting with the best validation accuracy, one can try additional neighbor points in the weighting space to find a local maximum.

Then for the fixed weighting, we added background features with different combination of weighting for more validation. Again, one can keep trying other parameters which are near the weighting for local maximum accuracy, until a satisfying weighting is achieved.

This manner can find a sub-optimal solution, and provide a possible upper bound for other kernel level fusion algorithms in the experiment.

5.2.4 Super Kernel Fusion

Super kernel Fusion[16] doesn't attempt to fusion kernel matrix in contrast with previous method. In super kernel, different features are trained separately as several SVM models. For each model, training instances are predicted and the returning scores are converted to probabilities given a mapping function. PHOW, PHOG, BHOW and BHOG are trained individually by SVM, and sigmoid function is applied to convert scores to probability.

Concatenation of probabilities forms a higher level of feature vector for a new SVM. Different kernel can be selected. The kernel used in our implementation is RBF (Radial basis function).

We had deployed the mentioned strategies in our experiment. Results will be shown in chapter 6.

Chapter 6

Experiment

In this section, we will talk about the configuration and result of our experiment. We run the experiments for three times, and the reported accuracy is averaged.

6.1 Caltech 101

The dataset used in the experiment is Caltech101[5], collected by Fei-Fei et al. . This dataset provides a benchmark to evaluate one's method. There are totally 101 categories in it, each contains about 40 to 800 images, mostly about 50. Image resolutions are roughly 300 x 200 pixels.

Each image in the dataset contains exactly one category, and left-right well-aligned. Rotation artifacts exist in the dataset.

6.2 Feature Extraction in ROI

In the experiment, SIFT features are collected in images every 8 pixels and quantized to 300 vocabularies in the codebook using KMean algorithm. For PHOG feature, orientations are quantized to 8 directions, and corresponding magnitude are added to histogram for every pixels.

ROI detection method by Liu et al. [13] needs a weighting vector for three feature maps. Since Caltech dataset do not have ground truth of ROI, the learned weighting $\vec{\lambda} = 0.24, 0.54, 0.22$ in Liu et al. 's work was applied.

One difference between our implementation and Liu et al. 's suggestion is that in color spatial-distribution feature, they add an additional term to reduce weighting of distant pixels from the center. Since some objects in images of Caltech101 occupy most images, we remove the suggested term in the experiment.

Both maximum pyramid level of PHOW and PHOG are 2, i.e. 4-by-4 in the bottom level in our implementation.

6.3 Feature Fusion

The well-known library LIBSVM[3] was used in our experiment, and one-versus-reset learning strategy was chosen.

We compared different methods for using foreground only and also using both foreground and background information in table 6.1.

	FG	FG + BG
Averaged Kernels	71.53%	68.72
Adaptive Search	73.61%	75.38
Kernel Alignment	58.32%	-
Super Kernel Fusion	64.79%	66.08%
[1]	81.3%	-
[2]	77.08%	
[11]	64.6%	

Table 6.1: Accuracy of different methods. "FG" column shows the accuracy of using only foreground information, while "FG + BG" uses foreground and background together. For last two rows, they doesn't segment image, and the numbers are for comparison.

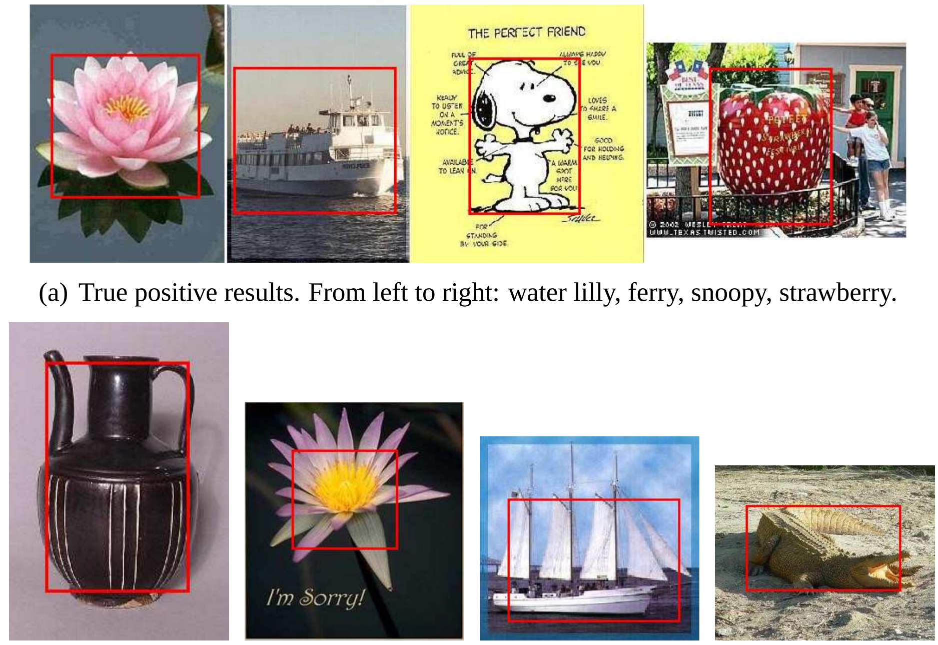
6.3.1 Discussion

Finding a good weighting at kernel level is a difficult problem. In a sense, the problem is to estimate which fused kernel would work better in SVM later. In the experiment, we can see that only super kernel and adaptive search can utilize background features instead hurt by them. Specifically for adaptive search strategy, the best weighting between PHOW, PHOG, BHOW, BHOG in three runs are $1 : 1.5 : 0.2 : 0.2$, $1 : 1.5 : 0.2 : 0.3$ and $1 : 1 : 0.1 : 0.3$, which shows using background information can improve the accuracy during adaptive Search procedure. Accuracy of super kernel, however, is still far from adaptive search.

Average kernel works surprisingly well, but it does not have ability to reduce weighing of worse signal. Kernel alignment are not able to select a good weighting, but suggested a ratio $1 : 0$ between PHOW and PHOG, i.e. to use PHOW along. Thus we do not further run it with background information in the experiment.

Comparing to the result of Bosch et al. [1], our adaptive search with only foreground is about 7.7% behind. We believe that the ROI feature weighting plays an important role here, and can improve our result a lot if we had ROI ground truth for Caltech101. ROI detectors can even be learned for each category. Again, if ROI ground truth was supported, the accuracy might be improved here.

6.4 Example of Result Image



(a) True positive results. From left to right: water lilly, ferry, snoopy, strawberry.

(b) Predicted ewer as barrel (20%)

(c) Predicted lotus as water lilly (15%)

(d) Predicted schooner as ketch (25%)

(e) Predicted crocodile as crocodile head (20%)

Figure 6.1: Some positive and negative image by adaptive search. (a) shows positive results, (b) to (e) are incorrect predictions. Description of negative results shows the percentage of error to predict class A as B. Negative examples are picked from high error rate of A-B prediction.

Chapter 7

Conclusion

In this thesis, we introduced an ROI detection method to object recognition problem. For this salient object detection method, it's more timely efficient than the previously purposed method. Although the final result is not as good as the best one of state-of-the-arts, fusion of foreground features is close to them. We believe that it can be further improved if human-labeled ROI of the dataset is supplied. In this case, we will have weighting vectors $\vec{\lambda_C}$ for each category C, and likely better ROIs can be found.

Though not all of the ROIs are perfectly fit, the background information outside ROI area can still provides clue and improves the classification accuracy. Experiment shown that methods can benefit from background information.

To fuse different features well is a challenge problem. In this thesis, we have compared different methods and shown restriction of some methods. Adaptive search provides an approximate upper bound for development of fusion strategy, which indicates there are much improvement can be done in the future.

Bibliography

[1] A. Bosch, A. Zisserman, and X. Munoz. Image Classification using Random Forests and Ferns. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 2007.

[2] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408, 2007.

[3] C. Chang and C. Lin. LIBSVM: a library for support vector machines. *Software available at http://www. csie. ntu. edu. tw/˜cjlin/libsvm*, 80:604–611, 2001.

[4] D. Crandall and D. Huttenlocher. Composite Models of Objects and Scenes for Category Recognition. *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 2007.

[5] 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*, 106(1):59–70, 2007.

[6] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1, 2005.

[7] S. Fidler and A. Leonardis. Towards Scalable Representations of Object Categories:

Learning a Hierarchy of Parts. *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 2007.

[8] K. Grauman and T. Darrell. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2, 2005.

[9] S. Hoi, M. Lyu, and E. Chang. Learning the unified kernel machines for classification. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 187–196, 2006.

[10] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics*, 22(3):277–286, 2003.

[11] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proc. CVPR*, 2(2169-2178):1, 2006.

[12] Y. Lin, T. Liu, and C. Fuh. Local Ensemble Kernel Learning for Object Category Recognition. *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 2007.

[13] T. Liu, J. Sun, N. Zheng, X. Tang, and H. Shum. Learning to Detect A Salient Object. *Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition (CVPR)*, 2007.

[14] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[15] F. Odone, A. Barla, and A. Verri. Building kernels from binary strings for image matching. *Image Processing, IEEE Transactions on*, 14(2):169–180, 2005.

[16] Y. Wu, E. Chang, K. Chang, and J. Smith. Optimal multimodal fusion for multimedia data analysis. *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 572–579, 2004.