Autonomous Object Category Learning for Service Robots Using Internet Resources

Md Reaz Ashraful Abedin

November 20, 2016
Master’s Thesis in Computing Science, 30 credits
Supervisor at CS-UmU: Thomas Hellström
Examiner: Ola Ringdahl

UMEÅ UNIVERSITY
DEPARTMENT OF COMPUTING SCIENCE
SE-901 87 UMEÅ
SWEDEN
Abstract

With the developments in the field of Artificial Intelligence (AI), robots are becoming smarter, more efficient and capable of doing more difficult tasks than before. Recent progress in Machine Learning has revolutionized the field of AI. Rather than performing pre-programmed tasks, nowadays robots are learning things, and becoming more autonomous along the way. However, in most of the cases, robots need a certain level of human assistance to learn something. To recognize or classify daily objects is a very important skill that a service robot should possess. In this research work, we have implemented a fully autonomous object category learning system for service robots, where the robot uses internet resources to learn object categories. It gets the name of an unknown object by performing reverse image search in the internet search engines, and applying a verification strategy afterwards. Then the robot retrieves a number of images of that object from internet and use those to generate training data for learning classifiers. The implemented system is tested in actual domestic environment. The classification performance is examined against some object categories from a benchmark dataset. The system performed decently with 78.40% average accuracy on five object categories taken from the benchmark dataset and showed promising results in real domestic scenarios. There are existing research works that deal with object category learning for robots using internet images. But those works use Human-in-the-loop models, where humans assist the robot to get the object name for using it as a search cue to retrieve training images from internet. Our implemented system eliminates the necessity of human assistance by making the task of object name determination automatic. This facilitates the whole process of learning object categories with full autonomy, which is the main contribution of this research.
Acknowledgements

I would like to express my gratitude to my supervisor Professor Dr. Thomas Hellström for his continuous support and guidance throughout the period of this research work. It was quite an experience to work under his supervision. I appreciate his openness to allow me applying different ideas I had, with flexible time frame. I would also like to thank my examiner Dr. Ola Ringdahl for his valuable suggestions and comments, that helped me to finish this thesis work properly.

I am sincerely grateful to Swedish Institute for providing me a fully funded scholarship to pursue my master’s studies. Finally, my heartiest thanks to my family members and my friends, specially Dawit, Shohel and Shaffat for supporting me in many ways and making my days in Umeå University memorable.
Contents

1 Introduction ........................................... 1
   1.1 Motivation ...................................... 1
   1.2 Research Goal .................................. 2
   1.3 Overview ...................................... 2
   1.4 Thesis Outline .................................. 3

2 Background .......................................... 5
   2.1 Literature review .................................. 6

3 Determining Object Name ......................... 11
   3.1 Image Capturing and Reverse Image Search .......... 11
   3.2 Fast Classification and Feedback ...................... 13

4 Learning Object Categories ..................... 17
   4.1 Image Retrieval and Labeling ...................... 17
   4.2 Redundancy Elimination .......................... 18
   4.3 Outlier Removal .................................. 19
   4.4 Bag of Visual Words .............................. 20
   4.5 Spatial Pyramid Representations ..................... 21
   4.6 Learning Algorithm ................................ 24

5 Object Localization ................................ 27

6 Implementation ..................................... 29

7 Results and Evaluations ......................... 31

8 Conclusions ......................................... 39
   8.1 Discussions ..................................... 39
   8.2 Future Work ..................................... 40
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Block diagram of implemented autonomous object category learning</td>
<td>3</td>
</tr>
<tr>
<td>3.1</td>
<td>Object images taken by robot’s camera from different viewpoints</td>
<td>12</td>
</tr>
<tr>
<td>3.2</td>
<td>Histogram of top ten names extracted from the retrieved text resulted by reverse image search with <em>Mug</em> images</td>
<td>13</td>
</tr>
<tr>
<td>3.3</td>
<td>First row: The target image used for reverse image search and the modified list of probable object names in the fast classification and feedback process. Second row: Sample images (partial) retrieved for each object name to use for training the classifier</td>
<td>14</td>
</tr>
<tr>
<td>3.4</td>
<td>Flow chart of the object name determination process</td>
<td>16</td>
</tr>
<tr>
<td>4.1</td>
<td>Image search results (partial) from Google, Bing and Yandex for search cue <em>Teddy bear</em></td>
<td>18</td>
</tr>
<tr>
<td>4.2</td>
<td>Retrieved images (partial) for <em>Eyeglass</em>, based on the appearance of target object. In some images target object is not appeared. Those images are considered as outliers</td>
<td>19</td>
</tr>
<tr>
<td>4.3</td>
<td>Followed steps in BOVW approach</td>
<td>20</td>
</tr>
<tr>
<td>4.4</td>
<td>Schematic diagram of k-means clustering in two dimensional feature space for $k = 3$</td>
<td>21</td>
</tr>
<tr>
<td>4.5</td>
<td>Schematic diagram of visual word histogram generated using visual vocabulary and image features. First row shows three images with their features. Second row shows the visual word histogram for corresponding images.</td>
<td>22</td>
</tr>
<tr>
<td>4.6</td>
<td>Schematic diagram of steps involved in creation of pyramid histogram of visual words. First row shows spatial pyramid representation of an image. Second row is the histogram generated from each sub-region of the image. Third row shows the weighted concatenation of all histogram vector found in different levels.</td>
<td>23</td>
</tr>
<tr>
<td>4.7</td>
<td>Schematic diagram of multi-class SVM with determined optimal hyperplane between two class data</td>
<td>24</td>
</tr>
<tr>
<td>4.8</td>
<td>Schematic diagram of One-class SVM</td>
<td>26</td>
</tr>
</tbody>
</table>
5.1 Sliding window on pyramid of images for scale invariant detection... 28
5.2 Merging multiple detection shown in (a) to single bounding box shown in
(b) using Non-maximum suppression ... ... ... ... ... ... ... ... ... ... ... ... 28

7.1 Object name is accurately determined as Headphone for captured images
in different lighting conditions. No name is determined for the images
with unusual color. The dot on image corner denotes detection status:
Green-detection; red-no detection ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ......
List of Tables

6.1 Number of training and test images for each category used for testing. . 30

7.1 Class-specific accuracy and average accuracy (%) for five categories in Caltech-256 dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34
7.2 Precision, Recall and F1 score for BOVW (SIFT) . . . . . . . . . . . . 34
7.3 Precision, Recall and F1 score for BOVW (SURF) . . . . . . . . . . . . 34
7.4 Precision, Recall and F1 score for PHOW . . . . . . . . . . . . . . . . 36
7.5 Precision, Recall and F1 score for PHOG . . . . . . . . . . . . . . . . 36
7.6 Precision, Recall and F1 score for PHOW+PHOG . . . . . . . . . . . . 38
List of Abbreviations

RIS  Reverse Image Search
VQ   Vector Quantization
SIFT Scale-Invariant Feature Transform
SURF Speeded-Up Robust Features
BOVW Bag of Visual Words
PHOW Pyramid of Histogram of Visual Words
PHOG Pyramid of Histogram of Oriented Gradients
SVM  Support Vector Machine
Chapter 1

Introduction

1.1 Motivation

In the modern world, human life is made easier and more productive with the use of technology. Development of Robots is one of the top achievements of advanced technological research. In last few decades, research in the field of Robotics went through some major breakthroughs that makes it possible to deploy service robots in industry and household to assist human. Now we can hand over many boring and laborious tasks to them, which saves our time and cost. Though the use of service robot firstly started in industries, later it is being designed for personal use also. Today’s robots can assist in our daily household works, play music when the weather is gloomy, tell jokes when we are sad and so on. Statistics show that about 4.7 million service robots are sold worldwide in 2014, which is 28% more than previous year [28]. The sale amount is expected to be doubled within 2015-18. Therefore, it is obvious that, usage of service robots is growing fast. Hence this area demands more research and developments to produce better product in future.

Service robots become popular because of their enhanced capability in automatic navigation and interaction with human. Moreover, Artificial Intelligence (AI) made the robots capable of perceiving different contexts in their environment and taking decisions. To perceive the contexts better, the robot should have to see the world better. Visual information is highly valuable resource in this case to know about different objects and how those are situated in the environment. A service robot needs to recognize and categorize the objects around itself for performing different tasks associated with those objects.

Currently service robots possess good quality cameras and high-speed processors to process acquired images and take quick decisions facilitated by intelligent algorithms. The challenge is having knowledge about different objects and identifying them in cluttered scenes. Available methods for learning object categories include supervised [34], unsupervised [15], semi-supervised [32] or weakly-supervised learning [11, 20, 46]. All of these methods need training images (labeled or unlabeled) to train corresponding classifiers. The number of required training images may vary from very few [5, 9] to hundreds of thousands [34] depending on the methods and required accuracy. Acquiring a large number of training images and labeling them is a challenging job that demands
substantial amount of human effort and time. With the tremendous growth of internet resources (e.g. text, images, videos), we can utilize it to acquire considerable amount of weakly-labeled images from web, which can make the process of learning object categories automated and decrease human labor and cost.

Let's consider a service robot that can learn object categories using the training images acquired by searching in the internet. But before that, it needs to know what is the object name it will search for. The solution could be using human assistance \cite{17,29} to get the linguistic cue about the target object. Another solution could be using captured images of the target object as a reference, and learn the object name directly from the internet. This approach can totally eliminate the need for human assistance to learn object categories, and make the robot self-taught.

1.2 Research Goal

Our goal in this research is to investigate the feasibility of a fully automated system for object categorization that can be applied in robotic applications. We propose to use the internet resources (texts and images) for determining the name/generic type of an object as well as learning multiple object categories. As a part of our investigation, we implement the system and assess its performance in domestic environment.

1.3 Overview

The main idea of this work is, after deploying a service robot in an environment, when it sees an unknown object, it tries to learn the object’s category during a pre-specified learning period. Firstly, it captures images of that object from multiple viewpoints and use those images as input for Reverse Image Search in internet search engine. Some search engines provide reverse image search facility where the users can upload an image and gets similar looking images as a result that are available in the internet. The search engines uses large-scale image retrieval algorithm based on image similarity to perform reverse image search \cite{44}.

The returned images have associated text, which is then retrieved and analyzed by the robot to obtain a list of probable object names. Then an iterative verification method (Fast classification and feedback) helps to make the list shorter and find the highest probable object name at the end of iteration process. Once the object name is known, the robot searches for images in the internet using that particular name as textual cue. Returned images are then filtered and used to generate training data to train a classifier for categorization. Besides categorization, the robot applies a simple object localization method to locate the object within the scene. So, the whole process can be divided into three major steps: determining object names, learning object categories and localizing the object (Figure \ref{fig:overview}). This whole process does not involve any kind of human assistance. As a result the robot is capable of learning new object models completely by itself.
1.4 Thesis Outline

In this chapter, we have introduced the research problem, our motivation behind this research and an overview of whole system. Chapter 2 describes background of this research problem and reviews related works done in this area.

Chapter 3 contains the process of object name determination. It describes how the texts are analyzed and how the verification method works.

Chapter 4 provides the description of object category learning approaches we followed. It includes different feature extraction and representation methods and learning algorithms used in this work.

Chapter 5 describes the method used for scale invariant object localization within a scene.

Chapter 6 contains the description of used software libraries and how the experimentation are performed.

Chapter 7 presents the results of our experiments and evaluation of the whole system.

Chapter 8 discusses the limitations and overall performance of the implemented systems and suggests some future works.
Chapter 2

Background

In this research work, we mainly focus on domestic service robot that has some sort of mobility within its environment and has grasping ability so that it can transport object from one place to another following human commands. This kind of robot usually possesses multiple cameras and depth sensors to see the three dimensional world. We want to provide the robot certain level of intelligence so that it can differentiate between individual objects that belong within its environment. Different machine learning approaches made it possible to learn object categories with decent precision [23]. These learning processes are kind of similar to human child’s learning process. As example, a grown up person shows different objects to a child and utters the object names. Children subconsciously make a model in their brain for each of the objects types. The more is the samples, the better is the model. Similarly for machines, learning object categories demands sample images to train classifiers. In case of service robots, it can be trained for all possible objects prior to the deployment in certain environment. But this approach is not robust since many new objects may appear and the robot has no idea about them. On the other hand, in post-deployment learning, the robot can explore it’s environment and learn object categories along the way. This approach will make the robot capable of dealing with changes around itself and growing it’s knowledge base with time. We adopted the same idea in this research work.

In [38], Thomaz el al proposed interactive learning, where humans act as teacher to the robot by providing feedbacks or directions. Their method deals with various aspects of learning, not only object classification. There are also semi-autonomous learning approaches where humans assist robots to a certain level to learn something [17,29]. In this research, we propose an approach that gives full autonomy to the robot to learn object categories without any means of human assistance. Starting from determining object name, then learning its model and finally localization the object; everything will be done by the robot itself utilizing internet resources. Having enough training images is a major requirement for better learning. As internet is a gigantic source of images, many researchers took this opportunity and used internet images to train learning models [34]. We have chosen similar strategy where the robot uses training images from internet to generate training samples.

Detecting the presence of an object within a scene is not always enough for a service robot. To grasp the object, it needs to know the position of the object. Moreover, if
there are multiple objects in a scene, all of those should be categorized individually. To localize the object, we focus on finding a bounding box around the target object rather than an exact contour. Sliding window method \cite{33,42} is an widely followed object localization strategy, that is also adopted in this work.

### 2.1 Literature review

The proposed approach for fully automated learning of object categorization involves three distinct steps. The first step determines the name of the object that is present in the captured image by utilizing the description of similar images available in the web. Only few research works are found where similar idea is adopted. In \cite{19}, Horváth et al used Google’s Reverse Image Search and retrieved the descriptions associated with the similar images returned by the search engine. Then the texts in the descriptions are analyzed based on a relationship graph between important extracted words. The used algorithm generates probability scores for each words, and suggest the one with highest score as object type. Though the authors claimed high accuracy for ImageNet \cite{34} data sets, their approach has major drawbacks. The approach can be considered as a single-shot method as no learning algorithm is used. As a result, the system needs to repeat the whole classification process every time an object is to be classified, which makes the system very slow and unsuitable for real-time applications.

The second step in the process is learning object categories. A substantial amount of research works are done in the field of object recognition and categorization till now \cite{30}. Recently researchers achieved state-of-art performance in object categorization through deep learning using millions of manually labeled training images \cite{34}. Using weakly-labeled internet images to learn object categories is an alternative, interesting and challenging approach that has been explored by researchers with promising results \cite{1,11,20,46}.

One of the earliest works in object categorization that uses internet images is presented by Fergus et al in \cite{13}. They extended their previously developed constellation model \cite{12} and used it to improve Google’s image search by ranking the output images from image search based on their visual consistency. The extension allows the system to consider heterogeneous parts of the images that may represent the appearance or geometry of the region of an object. The authors used a generative and probabilistic model considering the probability density function (PDF) of the part description, scale, object shape and occlusion as Gaussian distribution. The learning process uses detected features in the images to determine those PDFs so that it maximizes the likelihood of the training data. Two type of features are used in this case. One is the region of pixels (patches) and another is curve segments which represent the appearance and object geometry respectively. The author used both unsupervised and supervised learning. A portion of raw images returned by Google’s image search is used for unsupervised learning. On the other hand, manual selection of images is done in case of supervised learning. Using their model, they re-ranked the images based on their relatedness to searched object and compared with Google’s output. Their overall results show improvement in image ranking over Google’s output as well as failure for few categories. But ultimately it shows that visual consistency ranking is a valid conjecture.

A limitation of the model learnt though this approach is that it is not suitable to
be used in more general setting, where test images are different from the training images. Thus, this model is only good for re-ranking the images returned by web search engine but not good for catagorizing objects by seeing an image which it has never seen before. To deal with this case, in [11], Fergus et al. developed a model named TSI-pLSA to learn object categories from web searched images, which extends the idea of pLSA (Probabilistic Latent Semantic Analysis) by using spatial information to make the system translation and scale invariant. Though pLSA was first developed to apply in textual analysis [15], later in [36] Sivic et al. applied the concept of pLSA to categorize objects in images. This approach considers images as documents; regions found by interest operator as visual words and appeared object as topic or latent variable. Fergus et al. adopted this concept and extended it by using object’s location information. They named it Absolute position pLSA (ABS-pLSA), where the object location within the image is quantized into one of a certain number of bins and then the joint density on the appearance and location of each region is determined. To make this model translation and scale invariant, the authors further modified the model by introducing an extra latent variable, which is a vector containing four elements that represent the object centroid, x-scale and y-scale. To learn the density of the model, expectation maximization (EM) algorithm is used. The authors called this approach as Translation and Scale Invariant pLSA (TSI-pLSA). In spite of using highly noisy internet images as training data, the authors showed that the implemented model is competitive to existing methods that require training images carefully selected by human. Besides, there are many issues in their work that can be modified and improved further to get better performance.

A relatively different approach was followed by Bergamo and Lorenzo in [1]. Here the authors argue that if the training images obtained from internet image search can be combined with few manually annotated strong example images, then the learnt model can address the domain adaption problem, hence can achieve better performance. The domain adaption problem occurs when the test data is obtained from a distribution that is different from the distribution of training data, though they can be related. The authors claimed that, in most of the research work in object categorization using weakly-supervised web images, the learnt models show less accuracy than fully supervised approaches as domain adaption problem plays a role here. They dealt with this problem by systematic empirical analysis and address the distribution difference between test and training images. Combined with the web searched images, a random set of images from an established benchmark Caltech256 are used as training data and another random set are used for testing. The authors used classeme features [39] for image representation and implemented multi-class classification problem using K-binary classifiers trained with one versus the rest approach, where the prediction is performed using winner take all strategy. A linear SVM (Support Vector Machines) is used as the model for the classifiers. Along with three baseline algorithms, they implemented and compared the outcome of four different algorithms based on the strategy of domain adaption. Among them transductive learning (TSVM) showed most superior result which yields 65% improvement over the best published results that time, claimed by the authors. The ultimate results showed that, using domain adaption method and exploiting few strongly labeled images along with weakly-labeled source images as training data can significantly improve the performance of object categorization.

In [46], Yu et al. proposed to use Support Vector Data Description (SVDD) classifier for object category categorization. SVDD does not need any negative sample data for
training, which is advantageous over other methods \[11, 13\] where background images are used as negative samples. SVDD classifier creates a hypersphere around the positive training samples that distinguishes the target category from novel data or outlier. This can be considered as one-class classification problem. They used PCA-SIFT descriptor for image representation. EMD (Earth mover’s distance or $\chi^2$) kernel is used in SVDD framework along with two adjustable parameters. The implemented model can classify seven different object categories. The authors compared their result with previously discussed pLSA \[36\] and TSI-pLSA \[11\] methods. Among seven categories, performance of SVDD is competitive to pLSA in all cases. Moreover, for four categories SVDD performed better than TSI-pLSA. The overall outcomes conclude that SVDD can be considered as an alternative solution for learning from contaminated training data (internet images).

Khan et al. dealt with weakly/ambiguously-labeled internet images using multiple instance learning (MIL) in \[20\]. The whole learning method is divided in three stages. Firstly, an object model is learnt that detects only the presence or absence of target object in the images. With this classifier a number of images are selected with high confidence that contain the target object. Then, considering each image as a bag of objects, an object detector is learnt which can detect the position of target object within the image. The location information is used afterward to train a fully supervised latent SVM part-based model \[10\]. For image representation, the authors used pyramid of histogram of oriented gradients (PHOG) and pyramid of histogram of visual words (PHOW). They adopted the idea of Sparse-MIL \[41\] optimization that adapts a standard SVM formulation to deal with multiple-instance setting. Two publicly available benchmark datasets are used to evaluate the implemented method based on average precision. The results showed that the proposed method performs better than some baseline methods, but performs less than some state-of-art object detection methods that uses strongly annotated training images. But the comparison does not involve other weakly supervised learning algorithms.

Apart from fully probabilistic, pLSA or variant of SVM methods, recently Chen and Gupta approached the same problem with Convolutional Neural Networks (CNN) in \[3\]. They considered the internet as a source of large-scale data (e.g image) and therefore proposed to use deep learning to utilize them. They claimed that, as SVM methods use few parameters, hence the use of large-scale data is unlikely to be effective for those approaches. Their implemented model consists of five convolutional layers followed by two fully connected layers and finally another fully connected layer for classification. The learning is done through two stages. Firstly, easy images (having clean background and appearance) are used for initial training. The authors named those easy images as biased images as they are clean and simple which is not the case in real world. This initial training enables the network to learn low-level filters to represent visual words. To make the system robust, a second stage training is provided with hard images that works as fine-tuning the network. The authors used relationship graph between the images inspired by human’s recognition process to provide additional information of the classes that helps regularizing the network training. Finally, the authors trained a Region-based CNN (R-CNN) with localized objects within the images to make the system capable of not only recognizing objects, but also detecting their locations. Their implementation was tested on PASCAL VOC (2007 and 2012) \[7\], where it showed promising results. The result was competitive to \[14\] for VOC 2007 and outperformed
for VOC 2012. For VOC 2007, the authors reported state-of-art performance without using any VOC training data.

All the research works mentioned above were mainly done with a focus on applying the concept of object categorization using internet images to improve the image rank for search engines, or to recognize objects in images in a general sense. Nevertheless, all of those work can be considered as valuable references if we want to apply those ideas in Robot vision. In [29], [24], [22] and [17], the authors used internet resources for object categorization and directly applied it to robotic applications. Penaloza et al. followed an approach in [29], that emulates learning process of children. When the robot sees a human moving an object in domestic environment, it asks the name of the object to the human. After getting the name as text input, the robot searches for images in the internet using that text. The authors proposed Simile Selector Classifier (SSC) to filter out unrelated images and deal with polysemes. The classifier is trained with both positive and negative images. The positive images include intentionally varied color, illumination and scale for better accuracy. Negative images contain different category objects other than target category. The authors used same SS classifier in two stages. First, they used it for selecting positive images. Then the classifier is re-trained with selected positive and negative images for final categorization. They tested their proposed model for a personal assistant robot named Enon, developed by Fujitsu Corporation. The authors claimed that, the robot was able to learn object models using their method, but no details are provided regarding the number of objects or list of the objects. Besides no comparison is done between the accuracy of their method and any other previously developed method.

Hidago-Penâ et al. proposed another method in [17], where the robot takes human assistance to capture the picture of an object and takes text or speech input to use as cue for image search in the web. The authors implemented a one-class classifier named K-Nearest Neighbor Data Description (K-NNDD) classifier. The classifier is trained using Principal Component Analysis (PCA). The implemented system is tested for a NAO robot and the results are claimed to be satisfactory by the authors. As per their statement, the learning method facilitated the robot with increased autonomy, though empirical result is not fully understood because of lack of details.

Kulvicius et al. proposed a bi-modal solution in [24], where visual and textual cues are combined for retrieving positive images from internet to apply in robotic applications. Inspired by the fact that human uses additional linguistic cues when referring to an object to prevent ambiguity, the authors performed multiple searches with different auxiliary keywords for same object, which is considered as subsearch. This subsearch is basically done to deal with polysemes and filter out unrelated images. The authors named the whole process as Semantic Image Search (SIMSEA). They computed the similarity of subsearched images based on Bag of words representation and using PHOW (pyramid of histogram of visual words) features. Based on similarity measurement, a subset of searched images are obtained which fulfills the semantic expectation of user. Their results are evaluated by comparing with Google’s default search output based on precision and recall. Besides, for all four tested categories (cup, milk, apple, glass), images classified by humans are used as true reference for evaluation. Among four categories, SIMSEA performed satisfactorily for three of them (except milk) compared to human classification. The author also provided reasonable explanation for not performing well in case of milk category. The overall results showed that, semantic image search
is useful for retrieving cleaner images which can later be used to learn classifier with better precision.

The third and final step of our work is to localize the target object within the scene. Within many research works done in this topic, Sliding Window approach [33, 42] is the simplest one and widely used. This approach usually cause multiple detection of an object within nearby windows. In [43], Wojek et al used Non-maximum Suppression, which is a technique to merge all nearby detection windows.
Chapter 3

Determining Object Name

3.1 Image Capturing and Reverse Image Search

In this section, we approach the problem of obtaining the unknown target object name by utilizing captured images of that object. Here, we considered a domestic environment as our robot's work-space. Typically mobile service robots use built-in cameras and depth sensors to map the real world around. A big challenge is analysing visual data, as the real world is messy. In reality, images taken within domestic environment are mostly cluttered. Depth sensors can help to reduce the ambiguity in cluttered scene by providing distance information from the camera to real world objects. Our assumption is that the robot is capable of differentiating between those objects at least to a certain level based on the positions of the objects in the environment. As an example, assuming there are two similar looking mugs situated in the environment separated by at least some distance. We consider the robot can decide that there are two objects, though the name/type of the object is unknown. This task is generally done with the help of image segmentation by using sensors like stereo cameras, depth sensors etc. or fusing multiple sensory data [5, 16, 31, 35]. Generic object detection based on spatial data is an interesting topic of research; but it is out of the scope of this research work. Basically, our implemented system comes into play after this initial detection.

Whenever the robot encounters an unknown object, it takes multiple images of the object from different viewpoints. The images are taken from close distance to the object to minimize background area. Figure 3.1 shows some sample images of five different objects captured in indoor environment. Multiple images from different viewpoints make sure that important features of an object are not occluded. It also helps to get scene variation, which is beneficial for further processing. The captured images are then used as input for Reverse Images Search (RIS) in the internet search engine. We choose Google search engine for this purpose. Reverse image search is a form of content based large-scale image retrieval system where the retrieved images are expected to be similar to the query image. Generally large-scale image retrieval systems use different image features such as SIFT (Scale-Invariant Feature Tranform), SURF (Speeded Up Robust Features), MSER (Maximally Stable Extremal Region) etc. or combinations of multiple features to match query image to others [44].

Each image returned by the reverse image search has associated text. The text is...
supposed to contain information about the object shown in the image. Therefore, we retrieve the text which includes html page name and file name of the corresponding images. We ignore the returned images as those do not provide any valuable information. The texts are cleaned-up by removing unnecessary non-alphanumeric characters and any meaningless words. From the cleaned text, we extract single nouns, bigrams and trigrams. Bigrams are pair of consecutive units like letters, syllables or words used in a text. Similarly, trigrams are composed of three units. In our case, we put some conditions to extract bigrams and trigrams. For bigrams, both of the units should be word as well as noun. This way the compound nouns (e.g. Mobile phone, Teddy bear) are captured by the system. On the other hand, trigrams should contain nouns as the first and last unit, and preposition as middle unit (e.g. packet of chips). We call the extracted single nouns, bigrams and trigrams as names in general, no matter whether it actually denotes a name or not.

A histogram of names is then created based on their number of occurrence (Figure 3.2). The number of occurrence is also called frequency of that word. Frequency of synonymous nouns are summed up and only one of them having relatively higher frequency is retained; others are discarded. Afterwards, a list of names is generated based on the histogram and some other factors. If the number of images returned by the search engine is \( N \), then the number of text chunks \( T \) will be at least \( 2N \) (image file name and associated html page name). We consider the determined list of nouns feasible if following condition holds:

\[
\max_{1 \leq i \leq W} h_i \geq \frac{T}{2}
\]

where \( W \) is the total number of nouns in the list and \( h_i \) is the frequency of \( i \)th noun in the histogram. The equation denotes that, the highest occurring noun should appear in
at least half of the number of retrieved text chunk. If the above condition does not hold, the whole list is discarded and new images are taken to start the query again. After obtaining a valid list of nouns, we make the list shorter by removing the nouns having low frequencies. We consider a word frequency as low if it is less than a threshold value:

$$\text{threshold} = \frac{1}{5} \left( \max_{1 \leq i \leq W} h_i \right)$$  \hspace{1cm} (3.2)

Each noun in the list gets a score value based on its frequency given by $h_i$ value. The score is calculated as the percentage of their frequencies. We call the modified name list as probable names. There is a high probability that, highest scored name is actually the name of our target object. But there are also chances that the object name could be one of the other names in the list. If the object name does not belong to probable names, it is more likely that most of the images returned by the search engine are irrelevant to our query image. So, it is important to verify each of the names in the list. Therefore, we used a method that uses verifies the probable names and modifies the name list for finding a single target object name.

### 3.2 Fast Classification and Feedback

As discussed in previous section, word histogram is not enough for confidently determining the target object name. Nonetheless, it helps to narrow down the number of probable names and suggests which noun is more likely than others to be the object name. Generally we refer to all the names in the list as objects; but in reality some of those nouns may refer to something that is not concrete. To verify the probable names, we used an approach that involves learning category of each object in the list of probable names.
For better explanation, we can consider a real example from our tests. An image of a lamp was used as a query for reverse image search and the obtained list of probable names with their initial score (based on frequencies only) was: Lighting: 57.99, Lamp: 30.0, and Lantern: 12.0 (Figure 3.3). We observe that, the target object lamp has less score than lighting. As the first step in the verification process, we use each of the probable names to perform query in Google image search. The search results contain corresponding images of the query terms. Then a small number of images (ranging from 10 to 15) for each objects are retrieved at first (Figure 3.3). Those images are later used to produce bag of visual words (Section 4.4) based training samples which are feed to a multi-class SVM classifier. More about this classifier can be found in section 4.6.

The multi-class SVM classifier is then used to classify the raw images of the target object captured by robot’s camera. The classifier provides classification probability for each object in the list. The probability scores are then used to modify the initial score for each object. If the score of an object is sufficiently low, then it is removed from the list. If there are more than one name remain in the list, then additional images (2-3 images each time) are retrieved for those names. The new images are used to update the multi-class SVM classifier’s decision boundary. We can consider this as feedback loop that changes the input based on the output. This process is repeated iteratively where in each iteration, the probable names list gets modified. The iteration kept running until we get a single name remaining, or maximum number of iteration is reached. In Figure 3.3, we observe that, score value for lamp increases in each iteration and other names in the list got removed through several iteration and the target object name was detected accurately. The whole process of determination of object name is illustrated.
as a flow chart in Figure 3.4. We named this approach as *fast classification* as the number of training images are very few, hence the computational time for multi-class SVM is very less. Moreover, only initially downloaded images are used for generating visual vocabulary (Section 4.4). Additionally downloaded images do not contribute in vocabulary generation process, which makes the whole classification process faster.
Figure 3.4: Flow chart of the object name determination process.
Chapter 4

Learning Object Categories

4.1 Image Retrieval and Labeling

Once the object name has been determined, as described in previous chapter, the robot needs to learn the object categories so that it can classify that object correctly in future. An important fact should be made clear here. We once performed learning for the target object (i.e. lamp) category during the fast classification process in previous chapter. So, one can question why it is necessary to learn that category again. It is because that learning was not generalized enough. We used small number of images to generate training samples, which might be good enough to categorize the target object correctly as other objects in the list are typically very different than the target object. However, that learned classifier suffers from lack of adequate generality because of small training samples, which might cause poor accuracy for other instances of that target object in future. Therefore, we need larger number of images for training to get adequate performance.

Without experience, there is no learning. In case of machine learning models, the experience is achieved from the training samples. The larger the number of training samples, the better is the learning quality. Our implemented system facilitates the robot to acquire the training images by itself from the internet. Once the robot determines the object name, it uses the name as search cue/query to image search engines to get available images associated with that name in the internet. The robot performs image query to three different search engines: Google, Bing and Yandex, to get larger number of images (Figure 4.1). The number of retrieved images may vary in each query as there are limitations applied by the search engines on the number of resulting images per search.

For learning object categories, we choose supervised learning [27] approach, which needs labeled training data. That means, the class of a particular training sample should be known to the classifier. In our case, the label for each images are same as the object names used for query. Most of the images returned by search engines should contain scenes where the appearance of the query object is prominent. However, many of them may contain the target object, but the appearance is inconspicuous. Some images may even not contain the query object at all. The reason is that search engines use textual data associated with an image to match the query and an image might have wrong
Another fact is, even if for many images the target object is present in the scene, the exact position of it within the images is unknown. Therefore, the labeling of data is not precise enough. This kind of labeling is considered as weakly labeled. On the other hand, training images are considered to be strongly labeled if the positions of the target objects in the images are manually specified by bounding boxes. Intuitively, strongly labeled training samples helps to get better accuracy than weakly labeled samples. However, as we intend to avoid human assistance, we have no other choice but using the weakly labeled images, which makes it challenging to get good classification accuracy.

4.2 Redundancy Elimination

The training images are retrieved from three different search engines. All of them use the entire internet as source of images to execute the query. So, it is very probable that there are some common images among their search results. This redundancy may or may not affect the decision of the learned classifier, depending on the learning model. In case of SVM, there are two types of margins used for determining decision boundary among different classes—hard and soft margin. Hard margins are not affected by redundant training samples, whereas the soft margin are. Soft-margin SVM may learn a biased decision boundary because of redundant training samples near the margin. Thus, redundancy elimination in training data is an important step to get a non-biased decision. Therefore, all redundant images are eliminated by following a simple technique called Image hashing.

In the Image hashing technique we followed, all retrieved images are converted to hash-code using average hash algorithm. This algorithm takes an image and re-size it to the size of $8 \times 8$. So, total number of pixels becomes 64; each of them contains three color values. The color information is drooped by converting the image to grayscale.
the image contains only 64 pixel values from which the average pixel value is calculated. Then each pixel is represented by a bit depending on whether its value is below or above the average. Eventually, we get 64 bits integer for an image, which is referred to as hash-code. All hash-codes generated from the retrieved images are compared against each other based on their hamming distance, where zero hamming distance denotes same image. In this way, duplicate images are detected and removed.

### 4.3 Outlier Removal

As discussed in section 4.1, some of the retrieved images may not contain the target object. Undoubtedly, those images are wrong training samples and considered as outliers. Outliers cause ambiguity in training samples that affects the accuracy of the classifier. To deal with this problem, we used One-class SVM outlier detection algorithm [26]. One-class SVM is capable of capturing the shape of inliers within contaminated data set. The algorithm first train the classifier with contaminated sample data using carefully chosen hyperparameters. Then, all sample data are classified, where all out-of-class sample data are considered as outliers. Mathematical detail of one-class SVM is provided in section 4.6. The detected outliers are later removed from the training image set. Though, this step does not guarantee hundred percent outlier removal, but it definitely improves the classifier performance.

![ Retrieved images (partial) for Eyeglass, based on the appearance of target object. In some images target object is not appeared. Those images are considered as outliers.](image)

Figure 4.2: Retrieved images (partial) for Eyeglass, based on the appearance of target object. In some images target object is not appeared. Those images are considered as outliers.
4.4 Bag of Visual Words

Bag of Visual Words (BOVW) is an image classification approach, which is simple but widely used [45]. This technique is inspired by Bag of words that was devised for document classification. The idea of BOW is to collect all the words in the documents without any ordering, hence the name bag of words. Then a histogram is built from the words for each document and finally the histogram data is used as training feature to the learning model. Analogously, in BOVW, each image is considered as document. A visual vocabulary is created where image features are considered as visual words. Figure 4.3 shows the steps involved in BOVW approach that we followed.

![Followed steps in BOVW approach.](image)

There are a number of feature extraction and description methods available to determine local or global feature from images. In this work, we have used SIFT (Scale-invariant feature transform) and SURF (Speeded-Up Robust Features) feature detection algorithm, where both of them extracts and describe local features. SIFT is scale and rotation invariant but detection speed in slow. SIFT features are extracted through several steps. Firstly, the original image progressively Gaussian blurred and re-sized to several octave. Then for each octave, Difference of Gaussian (DOG) between two consecutive blurred images are calculated. DOG is basically the approximation of Laplacian of Gaussian (LOG), which reduce the high computational cost LOG. Then keypoints are generated by comparing each pixel in intermediate DOG images with their neighboring pixels within that image, and also in adjacent DOG images. Finally, the relative orientation of each keypoint are calculated as saved as SIFT descriptor.

On the other hand, SURF performs well in case of rotation and blurring of images, but sensitive to illumination and viewpoint changes. SURF is a modified version of SIFT, where the LOG is approximated by applying convolution with Box Filter. This method is very fast as it uses integral images for convolution. It describes the features using wavelet responses around the keypoints, which can also be calculated using pre-computed integral images. As a result SURF becomes several times faster than SIFT. Generally, for good images, SIFT and SURF shows similar performances in object recognition [21]. Nonetheless, we are interested in investigating how they behave in case of object categorization using weakly-labeled images.

After extracting features from all the training images, visual words are created by using vector quantization (VQ). The VQ method clusters the image feature descriptors to a certain number of regions in the feature space. We used k-means algorithm to cluster the image features (Figure: 4.4), where $k$ denotes the number of cluster. The clustering process starts by randomly considering $k$ number of data points as seed. Then
each points in the data set are associated to their nearest seed. This way one cluster of data points is formed for each seed. Then the centroid of each cluster are determined. This is one step in k-means algorithm. The newly determined centroids are considered as seed for next step and same process is repeated. This process continues until optimum clusters are found (when centroids do not change any more). K-means algorithm may get stuck in local minima. Repeated start with different initial seed position can help to get out of local minima and find a optimum solution. The choice of number of cluster \((k)\) is crucial. Too low value of \(k\) may over generalize the system. On the other hand, too high value of \(k\) may cause the system to be unnecessarily discriminative.

Figure 4.4: Schematic diagram of k-means clustering in two dimensional feature space for \(k = 3\).

A visual vocabulary is then created where each cluster area in feature space represents a single visual word in the vocabulary (Figure: 4.4). So the size of the vocabulary is the equal to the number of clusters. Here, each feature descriptor in a single cluster is considered to be representing same local pattern in the image. Then, for each training image, a word histogram is generated using the vocabulary and corresponding image features (Figure: 4.5). So, each histogram is a multi-dimensional vector, where the dimension is equal to the vocabulary size. We call this histogram vector as object features, which should not be confused with images features (e.g SIFT, SURF). The determined object features together with their corresponding classes are used as training samples and fed to a SVM classifier. The classifier learns from the training data by calculating best decision boundaries among the classes in the object feature space.

4.5 Spatial Pyramid Representations

In BOVW the image features are extracted without any specific order. Through this way, the spatial information of image content become lost. Intuitively, incorporation of spatial information with training data should improve classification performance. Pyramid of Histogram of Visual Words (PHOW) is a feature description technique that utilizes the spatial information of the image features. Therefore, we also used PHOW approach as an alternative to BOVW, and investigated the classification performance.
PHOW can be regarded an extension of BOVW. PHOW follows BOVW through the major steps from image feature extraction to building visual vocabulary. However, for feature extraction in PHOW, Dense-SIFT method is used. Simple SIFT features are determined using Lowe’s algorithm [25]. On the other hand, dense-SIFT calculates SIFT descriptor in densely throughout the image with small uniform spacing and multiple scales. Here, 3 pixels spacing and four different scales: 4, 6, 8, 10 are used for extracting dense-SIFT features [40].

In BOVW, we generated a single visual word-histogram for the whole image. On the other hand, PHOW divides an image into layer of increasingly finer spatial grids [2], and generates word-histogram for each of the sub-region. The number of grid cell gets quadrupled compared to previous layer for each iteration. This process continues for several layers. At layer $l$, the number of grid cell along each axes will be $2^l$. Assuming the histogram vector of the full image has $W$ elements, the dimension of PHOW descriptor for an image will be:

$$d_{PHOW} = W \sum_{l=1}^{L} 4^l$$  \hspace{1cm} (4.1)

where, $L$ is total number of layers created through the process.

The generated word-histograms for each subdivision of the image are then concatenated respectively (Figure 4.6), which is called spatial histogram. Finally, the spatial histogram vectors are used as training sample to train a Chi-square based SVM classifier (Section 4.6).

SIFT, SURF and dense-SIFT features capture the essence of appearance of the object in an image. But, we are also interested to investigate if the shape information of an object can help for better classification. Histogram of Oriented Gradient (HOG) [6]
is a feature description method that uses orientation of image gradient to capture the shape context of the object. HOG features were primarily used for human detection [6]. Because of high effectiveness, it became widely used in object detection and categorization. HOG features are found by creating histogram of orientation bins, that contains magnitude of orientation of gradients in small sub-region of image. Pyramid of Histogram of Oriented Gradient (PHOG) is a technique that extends the idea of HOG in a similar way as PHOW extends BOVW. It uses the spatial information of extracted HOG features throughout the image. Along with BOVW and PHOW, we also investigated the PHOG feature representation for learning objects models.

In [2], Bosch et al proposed PHOG feature representation for object classification. PHOG captures the object shape and its spatial layout within the image, which are later used to determine the correspondence between two shapes using chi-square kernel [37]. The formulation of image pyramid and spatial histograms is similar to the process of PHOW. The dimension of PHOG descriptor can be found by replacing word-histogram size $W$ in equation 4.2 by number of HOG orientation bin $K$:

$$d_{PHOG} = K \sum_{l=1}^{L} 4^l$$  \hspace{1cm} (4.2)

In [2], the Bosch et al combined appearance and shape context for object categorization and found better results. Inspired by this idea, we combined PHOW and PHOG feature representations for object category learning and compared the results with other methods where PHOW and PHOG are used separately. The feature combination is done by concatenating PHOW and PHOG histogram vectors without any weighting.
4.6 Learning Algorithm

In this work, we have used Support Vector Machines (SVM) as learning algorithm/model. SVM is a supervised learning model and can be used for both classification and regression. For classification, it determines optimal hyperplane with highest possible margin between the data points of two classes in the feature space. In our case, the data points are the visual word histogram vectors generated for all the training images. The multi-dimensional space where the histogram vectors lie is called the feature space.

To determine optimal hyperplane between training data points of two classes, the data set should have linearly separable patterns. Linear SVM is directly applicable for linearly separable data. But, in real world, most of the data are non-linear. Non-linear SVM comes to provide solution for this. Non-linear SVM uses kernel trick, which transforms linearly inseparable data to new higher dimensional space using kernel function, so that it becomes linearly separable.

Considering a two-class classification problem where the training data set is \( \Omega = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \), where \( x_i \in \mathbb{R}^d \) (\( d = 2 \) in this case) is the data point and \( y_i \in -1, 1 \) is the class of the \( i \)th data point. Assuming the data is linearly separable, SVM finds the hyperplane with maximum possible margin between the set of points \( x_i \) from which \( y_i = 1 \) and \( y_i = -1 \). The hyperplane has the following equation:

\[
 w^T + b = 0 \tag{4.3}
\]

where, \( w \in \mathbb{R}^d \) and \( b \in \mathbb{R} \)

The hyperplane lies at equal distance to the nearest data points of both classes. The learning process for SVM can be formulated as a constrained optimization problem like following:
\[
\min_{w \in \mathbb{R}^d, \xi_i \in \mathbb{R}^+} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i
\]
subject to \( y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \) for \( i = 1, 2, \ldots, n \)
and \( \xi_i \geq 0 \)  \hspace{1cm} (4.4)

where, \( \xi_i \) is called slack variable and \( C \) is regularization parameter.

The slack variables allow some data points to lie inside the margin, which may help to prevent overfitting. Value of \( C \) determines the width of the margin considering those data points that lies within the margin. The minimization problem stated above is typically solved by Quadratic Programming with the help of Lagrange Multipliers. Finally, the classification is performed using the decision function:

\[
f(x) = \text{sign} \left( \sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b \right)
\]
where, \( K(x, x_i) \) is Kernel function and \( \alpha_i \) are Lagrange multipliers.

Among several available SVM kernels, linear and RBF kernel are widely used. Linear kernel is nothing but using training data directly without any higher dimensional mapping. On the other hand, when the data is not linearly separable, the RBF kernel uses Gaussian Radial Basis function to indirectly map the data to higher dimension. It is not necessary to explicitly map the data to higher dimension. All we need is the dot product of two data points in higher dimensional feature space. The kernel function does this for us:

\[
K(x, x') = \phi(x)^T \phi(x') = \exp \left( -\frac{\|x - x'\|^2}{2\sigma^2} \right)
\]
where, \( \sigma \) is the kernel parameter.

For BOVW approach, we used SVM with RBF kernel. Beside available popular kernels (e.g. linear, RBF, polynomial, sigmoid), one can use customized kernel mapping to better suit their training data. We used kernel mapping based on chi-square \( (\chi^2) \) distance for PHOW and PHOG approaches because of its superior performance in image categorization \[2\]. For two data points vector \( x \) and \( y \), the chi-square kernel is found by:

\[
k(x, y) = \exp \left( -\gamma \sum_i \frac{(x_i - y_i)^2}{x_i - y_i} \right)
\]
where, \( \gamma \) is a kernel parameter.

We used One-class SVM along with noun histogram to determine object name (section \[3.2\]) and to remove outlier in the training images (section \[4.3\]). One-class SVM algorithm separates the training data from origin and finds optimal hyperplane that maximizes the distance between the data-points and the origin. This can be formulated as a optimization problem:
\[
\min_{w \in \mathbb{R}^d, \xi_i \in \mathbb{R}^+} \frac{1}{2} \|w\|^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i - \rho \\
subject \ to \quad w \cdot \phi(x_i) \geq \rho - \xi_i \quad for \quad i = 1, 2, ..., n \quad \text{and} \quad \xi_i \geq 0
\]

where, \( \nu \) represents an upper bound on the faction of data that are may be outliers.

In this case, the decision function is found by:

\[
f(x) = \text{sign} \left( \sum_{i=1}^{n} \alpha_i K(x, x_i) - \rho \right)
\]

where, \( \rho \) is a hyperparameter that characterizes the hyperplane determined by the classifier.

Figure 4.8: Schematic diagram of One-class SVM
Chapter 5

Object Localization

Knowing the position of an object is equally important as determining the object category; to grasp or move an object from one place to another. Based on the approaches described in previous chapters, the robot is capable of determining the name of an unknown object and learn its category. In this section, we discuss the solution for object localization within the image. Though object localization in the image was not the main focus of our work, this part is necessary to complete the whole system of automated object categorization and to see the learned classifier in action. Because of this, we followed a naive approach to implement object localization.

We adopted popular sliding window approach [33, 42] to find the location of the object within the image. In this approach, a rectangular window created with certain width and height. The window slides across the whole image and along the way image features are extracted within the sub-region of image under the window. The learned classifier then try to determine object category within each of the sub-region. If the classifier finds an object in a image sub-region, then corresponding window position is saved as detected.

As the size of sliding window is fixed, detecting object in different scales becomes an issue. To solve this, we generate pyramid of an image by re-sizing it to different scales. Each of them is called a layer. Then the same sized window is applied to find the object position in each layer (Figure 5.1). If the bounding box found in a certain layer, the re-size ratios for that particular layer is then used to calculate corresponding bounding box size and position in the original image. The classifier may detect the object in multiple overlapping positions. We use Non-maximum suppression to fuse multiple detection data and find a single bounding box around the target object.

The performance of the sliding window method varies with the number of scales used and the step length of the sliding window. Smaller step length causes better detection, though it is computationally expensive.
Figure 5.1: Sliding window on pyramid of images for scale invariant detection.

Figure 5.2: Merging multiple detection shown in (a) to single bounding box shown in (b) using Non-maximum suppression.
Chapter 6

Implementation

The implementation of the whole system consists of three modules in order- object name finder, object category learner and object localizer. These three modules works almost independently. Only dependency between these modules is, one module use the final result produced by preceding module. As we did not have access to any service robot, we attempted to simulate robot’s eye with a Logitech high definition web-camera. We used Python and Matlab programming language to implement the whole system.

We manually captured the images of target objects from multiple viewpoints. The number of images captured for each unknown object was four. This number can vary depending on developers choice. Object name finder module takes those images as input and performs reverse image search. Despite the availability of multiple reverse image search engines, we choose Google because the results returned by Google are found to be more accurate than others. Determination of object name involves text processing. We used Python’s Natural Language Toolkit to process retrieved text associated with the returned images by Google’s reverse image search. In the fast classification process we used ten images per name to train the initial SVM classifier. Number of additional images used for further iterations was three. Both of these numbers can be set by user. Higher number of training images cause increased computational time, and lower number of images cause less accuracy. We used one-vs-rest SVM classifier for fast classification as it is beneficial when number of classes is low.

In BOVW method, SIFT and SURF features are extracted using OpenCV library. To make the feature extraction method faster, FAST (Features from Accelerated Segment Test) algorithm is used to find features, whereas SIFT and SURF are used as feature descriptor. we used 25 as threshold value for FAST to avoid insignificant feature detection. The used size of SIFT and SURF feature descriptor is 128 and 64 respectively.

For clustering image features and training classifier, k-means algorithm is used. Choosing the number of clusters \( k \) is important for having good visual vocabulary, hence better classification. To obtain a good value of \( k \), the classification is performed with different value of \( k \) ranging from 30 to 500. Because of limitations in computational resources, we could not assign the value of \( k \) more than 500. Number of iteration for k-means is set to 30 to avoid local minima. Within the specifying range, \( k = 50 \) provides highest accuracy for both SIFT and SURF.
Both one-class and multi-class SVM are implemented by using Python’s *Scikit-learn* machine learning library. To get the best classifier for BOVW, we experimented with three different SVM kernels, and combinations of hyperparameters. The tested kernels are: linear, RBF and Chi2-square. The optimal hyperparameters are found by applying *Grid Search* with a set of different values. To prevent overfitting, five-fold cross-validation is performed in all the cases. Based on the cross-validation results and grid search, the RBF kernel with \( C = 50 \) and \( \gamma = 0.0078125 \) is found as best combination for the multi-class SVM classifier in BOVW method. We choose one-against-one approach for the SVM to deal with multi-classes. It usually performs better than One-against-rest when number of classes is large.

To implement the PHOW and PHOG and combined technique, we used *VLFeat* computer vision library. VLFeat provides library for extracting dense-SIFT features, which are advantageous for PHOW method. Besides, it supports homogenous kernel mapping, k-means clustering and SVM training that are used in the implementation of PHOW and PHOG. To obtaining PHOG features, we used Anna Bosch’s implementation of PHOG feature extraction method \([2]\). The used bin size of the histogram of oriented gradients is 40, where range of orientation is 360 and number of pyramid layer is 2. For both PHOW and PHOG, chi-square kernel is used to train the classifier.

We used five different categories of objects for experimentation and to investigate the performance of the learned classifier. The objects are: *Eyeglass, Headphone, Mug, Spoon* and *Teddy Bear*. Reason for choosing these categories is, they are included in the *Caltech-256* benchmark dataset and we had access to those objects as well to test whole system in real scenarios. We got 174 training images on average per category after combining the results from Google, Bing and Yandex (Table: 6.1). Here, the internet images are the source of our training data and the *Caltech-256* dataset provides us the test data. We observe that the training data set is *imbalanced* or in other words the number of training images are not equal. To deal with imbalance dataset, a weight value is assigned to each class based on corresponding number of training images (Equation: 6.1). The weight values modifies the regularization parameter of the classifier and solve the problem of imbalanced training set.

\[
weight_i = \frac{\text{Total number of samples}}{\text{number of classes} \times \text{number of samples for class } i} \quad (6.1)
\]

Caltech-256 is a popular dataset containing images of 256 categories used in many research works in computer vision. We choose it so that we can have a standard and the performance can be compared with other research works. Moreover, the system is also tested in real scenarios. In this work, our proposed method is mainly aimed to be used for domestic service robots. Therefore, the reported results are produced from experiments done within domestic environment.
Chapter 7

Results and Evaluations

The accuracy of determining object name is assessed mainly by extensively observing the system output rather than using numerical data. Because, performance of this step varies highly depending on many facts. Here, we mention and analyze some important facts and discuss how the implemented system behaves in those cases. For ease of explanation, when an object name is accurately determined, we refer to it as detection.

The system is found to be robust for different lighting conditions, but sensitive to unusual scene color (Figure 7.1). The cause of sensitivity to unusual scene color has reasonable explanation. Most of the internet images are realistic or refined to make the image visually appealing. That’s why unusual scene color causes less or no matches in reverse image search.

![Figure 7.1: Object name is accurately determined as Headphone for captured images in different lighting conditions. No name is determined for the images with unusual color. The dot on image corner denotes detection status: Green-detection; red-no detection.](image)

We observed that image background has important effect in the output accuracy. Figure 7.2 shows a case where the object with cleaner background got detected, but was remain undetected for with highly cluttered background. Background clutter decrease the prominence of appearance of the target object in the images. As image search engines try to match the whole scene of the image, too much background clutter may result unrelated images. Nonetheless, we observed that, in many cases, object images
with too clean background cause wrong detection because of lacking realisticness.

On the other hand, background clutter does not matter that much if the appearance of the target object is sufficiently prominent and the object features are highly distinctive (Figure 7.3).

![Figure 7.2](image1.png)

Figure 7.2: The system was unable to detect a *Lamp* with high background clutter. But accurate detection is obtained with the image captured from close distance to the target object with less background area.

![Figure 7.3](image2.png)

Figure 7.3: Despite of reasonable amount of background clutter, the system detects target objects (*Coca-cola* and *Guitar*) because of highly distinctive object features and prominent appearance.

Using multiple images captured from different viewpoints showed better detection than using single image. Visibility of important features of an object may vary with viewpoints. Therefore, multiple viewpoint images ensures the visibility of those features, which ultimately causes better detection. Another criteria for better detection is the conventionality of the object. As an example, a typical looking Mug in an image is more likely to be detected, where an unconventional looking chair may remain undetected.

For any query image, the reverse image search always returns some visually similar images, which may or may not contain the target object. So the word histogram generated from associated text does not ensure the target object name, rather it suggests some name. Its the fast multi-class SVM classifier, that learns the categories of the suggested objects very quickly, classify the captured images and use probability score as feedback to find the accurate object name from the list of probable object names. Based on our observation, this classifier helps to prevent the system from detecting false positives. The system may fail to detect any object because of the issues discussed before,
but it rarely performed false detection. Overall, the performance of this first step was promising. Figure 7.4 shows some examples of correctly detected object images.

![Correctly detected object images](image)

Figure 7.4: Images (partial) with correctly detected object names.

For object category learning, five different classifiers are learned based on: simple BOVW (with SIFT and SURF), PHOW, PHOG and combination of PHOW and PHOG. We compared their classification performance for five selected objects: Eyeglass, Headphone, Mug, Spoon and Teddy bear from the benchmark dataset Caltech-256. We have used several performance measures to access the learned classifiers.

The average accuracy denotes output accuracy of a classifier on average for all classes. On the other hand, class-specific accuracy provides classification accuracy for individual class, which is more detailed than previous one. Only using average accuracy may cause misjudgment of the system performance. For example, a two class classifier with 50% average accuracy may seem descent, but class-specific accuracy for Class A and Class B could be 95% and 5%, which clearly shows a biased or inaccurate classifier. So, we need to use some other performance measures to evaluate the system properly.

Two widely used performance measures are Precision and Recall, which can be found by following equations:

\[
\text{Precision} = \frac{\sum \text{True positive}}{\sum \text{True positive} + \sum \text{False positive}}
\]

\[
\text{Recall} = \frac{\sum \text{True positive}}{\sum \text{True positive} + \sum \text{False negative}}
\]

F1-score is another important performance measure that is calculated as:

\[
F1 - score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]
Table 7.1: Class-specific accuracy and average accuracy (%) for five categories in Caltech-256 dataset

<table>
<thead>
<tr>
<th>Class</th>
<th>BOVW (SIFT)</th>
<th>BOVW (SURF)</th>
<th>PHOW</th>
<th>PHOG</th>
<th>PHOW+PHOG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eyeglass</td>
<td>0.71</td>
<td>0.39</td>
<td>0.80</td>
<td>0.60</td>
<td>0.77</td>
</tr>
<tr>
<td>Headphone</td>
<td>0.65</td>
<td>0.75</td>
<td>0.75</td>
<td>0.65</td>
<td>0.73</td>
</tr>
<tr>
<td>Mug</td>
<td>0.41</td>
<td>0.24</td>
<td>0.87</td>
<td>0.61</td>
<td>0.89</td>
</tr>
<tr>
<td>Spoon</td>
<td>0.12</td>
<td>0.03</td>
<td>0.59</td>
<td>0.52</td>
<td>0.59</td>
</tr>
<tr>
<td>Teddy bear</td>
<td>0.70</td>
<td>0.76</td>
<td>0.96</td>
<td>0.78</td>
<td>0.98</td>
</tr>
<tr>
<td>Average</td>
<td>52.33</td>
<td>46.10</td>
<td>75.68</td>
<td>63.61</td>
<td>78.40</td>
</tr>
</tbody>
</table>

Table 7.2: Precision, Recall and F1 score for BOVW (SIFT)

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eyeglass</td>
<td>0.35</td>
<td>0.71</td>
<td>0.62</td>
</tr>
<tr>
<td>Headphone</td>
<td>0.56</td>
<td>0.65</td>
<td>0.60</td>
</tr>
<tr>
<td>Mug</td>
<td>0.47</td>
<td>0.41</td>
<td>0.44</td>
</tr>
<tr>
<td>Spoon</td>
<td>0.42</td>
<td>0.12</td>
<td>0.19</td>
</tr>
<tr>
<td>Teddy bear</td>
<td>0.51</td>
<td>0.70</td>
<td>0.59</td>
</tr>
<tr>
<td>Average</td>
<td>0.51</td>
<td>0.52</td>
<td>0.49</td>
</tr>
</tbody>
</table>

Precision measures classifier *exactness* or how much one class interfere to others. On the other hand, Recall measures classifier *completeness* or how well it classifies individual classes. F1-score measures the balance between Precision and Recall. The value of Precision, Recall and F1-score has the range from 0 to 1, where higher values denotes better performance. We also provided *Confusion Matrix* for each classifier, which depicts the classification results in detail (true test classes and predicted classes).

Table 7.1 shows class-specific accuracy and average accuracy for each classifier. It is observed that, BOVW-SIFT approach has just above fifty percent accuracy (52.33%). So, one can claim that it learned something. But it terribly fails to detect Spoon (class accuracy = 12%), whereas it performed well for the category: Eyeglass, Headphone and Teddy bear. From corresponding confusion matrix (Figure 7.5), we can see that, it confuses Spoon mostly with Mug and Teddy bear. Besides, the low average F1-score clearly shows this methods vulnerability (Table 7.2).

BOVW-SURF approach shows more degraded results. It has average accuracy less than 50%, which is not feasible to use in real applications. It performs pretty good for categories: Headphone and Teddy bear (Figure 7.6). But similar to BOVW-SIFT, it
Figure 7.5: Confusion matrix for BOVW (SIFT).

Figure 7.6: Confusion matrix for BOVW (SURF).
Figure 7.7: Confusion matrix for PHOW.

<table>
<thead>
<tr>
<th>Object</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eyeglass</td>
<td>0.59</td>
<td>0.80</td>
<td>0.68</td>
</tr>
<tr>
<td>Headphone</td>
<td>0.90</td>
<td>0.64</td>
<td>0.75</td>
</tr>
<tr>
<td>Mug</td>
<td>0.84</td>
<td>0.87</td>
<td>0.85</td>
</tr>
<tr>
<td>Spoon</td>
<td>0.78</td>
<td>0.59</td>
<td>0.67</td>
</tr>
<tr>
<td>Teddy bear</td>
<td>0.72</td>
<td>0.96</td>
<td>0.82</td>
</tr>
<tr>
<td>Average</td>
<td>0.78</td>
<td>0.76</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Table 7.4: Precision, Recall and F1 score for PHOW

<table>
<thead>
<tr>
<th>Object</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eyeglass</td>
<td>0.47</td>
<td>0.60</td>
<td>0.53</td>
</tr>
<tr>
<td>Headphone</td>
<td>0.84</td>
<td>0.65</td>
<td>0.73</td>
</tr>
<tr>
<td>Mug</td>
<td>0.63</td>
<td>0.61</td>
<td>0.62</td>
</tr>
<tr>
<td>Spoon</td>
<td>0.71</td>
<td>0.52</td>
<td>0.60</td>
</tr>
<tr>
<td>Teddy bear</td>
<td>0.56</td>
<td>0.78</td>
<td>0.66</td>
</tr>
<tr>
<td>Average</td>
<td>0.67</td>
<td>0.64</td>
<td>0.64</td>
</tr>
</tbody>
</table>

Table 7.5: Precision, Recall and F1 score for PHOG
Figure 7.8: Confusion matrix for PHOG.

Figure 7.9: Confusion matrix for PHOW+PHOG.
<table>
<thead>
<tr>
<th>Object</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eyeglass</td>
<td>0.67</td>
<td>0.77</td>
<td>0.72</td>
</tr>
<tr>
<td>Headphone</td>
<td>0.95</td>
<td>0.73</td>
<td>0.83</td>
</tr>
<tr>
<td>Mug</td>
<td>0.84</td>
<td>0.89</td>
<td>0.86</td>
</tr>
<tr>
<td>Spoon</td>
<td>0.78</td>
<td>0.59</td>
<td>0.67</td>
</tr>
<tr>
<td>Teddy bear</td>
<td>0.70</td>
<td>0.98</td>
<td>0.82</td>
</tr>
<tr>
<td>Average</td>
<td>0.80</td>
<td>0.78</td>
<td>0.78</td>
</tr>
</tbody>
</table>

Table 7.6: Precision, Recall and F1 score for PHOW+PHOG

fails to detect Spoon as well as Mug.

PHOW demonstrated fairly good performance with average accuracy 75.68%. It classifies Teddy bear with very high accuracy (Recall = 0.96). It also classifies Spoon with acceptable accuracy (Recall = 0.59). The individual as well as average F1-score are also very desirable (≥ 0.75).

PHOG produced descent average (63.61%) and class accuracy (Table 7.1). The F1-score is marginally acceptable (≥ 0.64) for most of the classes (Table 7.5). It classified most of the Teddy bear images correctly (Recall = 0.78), but it also confuses Spoon with Teddy bear in many cases (0.22 in confusion matrix).

The combination of PHOW and PHOG approach provides highest accuracy (78.40%) among all. It has best class accuracy for three of the five classes. The classification accuracy for Teddy bear is nearly perfect (0.98) (Table 7.6). Besides, Average F1-score is more than PHOW (≥ 0.78).
Chapter 8

Conclusions

8.1 Discussions

We have implemented an autonomous object category learning method, which can help a service robot to be self-taught. Using this approach, a newly deployed robot in a domestic environment can learn to categorize objects it has never seen before without asking for any kind of human assistance. Despite the descent results obtained, there are some challenges and limitations of the implemented system.

The object categories learned by this system are very much generic in some cases. As an example, the system may not distinguish between an acoustic guitar and an electric guitar; or between a tennis bat and a badminton racket, because of their visual similarities. It may detect both raspberry and strawberry as their generic kind berry.

Unconventional object categories may not get learned by the system because of unavailability of similar images in the internet. But this challenge is also valid for many other object categorization approaches, because of shortage of available training images. Another limitation of our system is, it can not categorized objects based on color, as we did not incorporate any color information in learning. So, a red T-shirt and blue T-shirt are same to the classifier. Despite of these limitations, the implemented system exhibited good results in determining object name. The verification process using fast multi-class SVM helps preventing the system from generating false positive results and proved to be effective.

In object category learning, it is observed that, most of the classifiers detected Teddy bear and Headphone more accurately. This has reasonable cause. These two category objects have strong and distinct features, hence produce more data that helps the classifier to learn the categories with good accuracy. On the other hand, the classifiers had difficulty with detection of Spoon, because of having low amount and weak features.

We also observed that, adding spatial information to the image features can improve the classification performance to a great extent. In case of PHOW, the accuracy is increased by almost 44% compared to BOVW. Combining appearance context (PHOW) with shape context (PHOG) improved the accuracy by 3.5% compared to second best approach PHOW. Zeiler and Fergus reported the best result for Caltech – 256 dataset.
in [17] with average accuracy $70.6 \pm 0.2\%$, where they used deep convolutional network as learning model, and trained the model using ImageNet – 2012 dataset [34] along with 6 images per class from Caltech-256 dataset. As we used only five of 256 available categories, our result is not directly comparable to the reported best result. Nonetheless, our focus is to investigate the feasibility of the whole system we have proposed, rather than finding better learning approach than existing ones.

Here, we basically attempt to make an indirect comparison to have a high level idea of the feasibility of our system. ImageNet – 2012 dataset contains 1300 manually annotated training images per category for 1000 categories. On the other hand, we got 174 weakly-labeled images on average for 5 categories retrieved from internet search results. It is obvious that, average accuracy decreases with the number categories. However, despite using weakly-labeled small number of training images, getting 78.40\% percent accuracy is absolutely promising. Our overall observations indicate that the proposed method is feasible and demand further investigations.

### 8.2 Future Work

This research work can be extended in several ways in future. Semantic analysis of retrieved texts should help narrowing down the choice of probable object names and deal with homonyms. Intelligent auto image segmentation can be applied to prevent background clutter problem. Probabilistic analysis may help to deal with the ambiguity of weakly label images. Besides, more advanced image feature representation technique and learning methods can be used to increase the classification accuracy.
Bibliography


