SEMANTIC UNDERSTANDING OF AN IMAGE USING MULTI REGION LABEL ANNOTATION

by

MOHANA PRIYA K 2013103552
SUDHARSAN S 2013103566
GAYATHRI DEVI K 2013103579

A project report submitted to the

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY, CHENNAI – 25

MAY 2017

BONAFIDE CERTIFICATE

Certified that this project report titled **SEMANTIC UNDERSTANDING OF AN IMAGE USING MULTI REGION LABEL ANNOTATION** is the *bonafide* work of **MOHANA PRIYA K (2013103552)**, **SUDHARSAN S (2013103566)** and **GAYATHRI DEVI K (2013103579)** who carried out the project work under my supervision, for the fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. Certified further that to the best of my knowledge, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on these or any other candidates.

Place: Chennai
Date:

Dr.S.Chitrakala
Associate Professor
Department of Computer Science and Engineering
Anna University, Chennai – 25

COUNTERSIGNED

Head of the Department,
Department of Computer Science and Engineering,
Anna University Chennai,
Chennai – 600025

ACKNOWLEDGEMENT

We express our deep gratitude to our guide, Dr. S.Chitrakala for guiding us through every phase of the project. We appreciate her thoroughness, tolerance and ability to share her knowledge with us. We thank her for being easily approachable and quite thoughtful. Apart from adding her own input, she has encouraged us to think on our own and give form to our thoughts. We owe her for harnessing our potential and bringing out the best in us. Without her immense support through every step of the way, we could never have it to this extent. We are extremely grateful to Dr. D.Manjula, Head of the Department of Computer Science and Engineering, Anna University, Chennai 25, for extending the facilities of the Department towards our project and for his unstinting support. We express our thanks to the panel of reviewers Dr. S.Valli, Dr.B.L. Velammal and Dr. S. Sudha for their valuable suggestions and critical reviews throughout the course of our project. We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

Mohana Priya K **Sudharsan S** **Gayathri Devi K**

ABSTRACT

Image annotation is one of the applications of image processing in which a collection of labels assigned to every object in the image.The aim of this project is to understand a given image and provide a short description about the various objects in the image by considering the spatial relationship among objects.

Single Label Image Annotation considers the entire object as a single label,thereby not considering the relationship that exists among the objects in the image.In this project, we propose a semantic understanding of image which is done through multi label image annotation.

Given an image,it undergoes Pre-Processing using Single Haze Removal Algorithm, which results in an enhanced image quality.Then the features in the image are extracted using SURF (Speed Up Robust Features) technique.Using the features extracted, objects in the image are recognized using Blob Detection Algorithm.The directional relations (top, bottom, left, right)between the objects are considered under Spatial Relationship to semantically understand the relation between various objects in the image.Finally, a short description about the various objects in the image is produced.This can result in a better understanding about the objects in an image.The accuracy is found to be roughly around 83%.This finds its applications for visually impaired people to better understand an image. The textual description can also be fed as input to hardware systems like Braille System etc.,

ABSTRACT

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MIML	Multi- Instance Multi-Label
SVM	Support Vector Machine
UI	User Interface
UML	Unified Modelling Language
MLL	Multi Label Learning
JPEG	Joint Photographic Experts Group
PNG	Portable Network Graphics

CHAPTER 1

INTRODUCTION

1.1 GENERAL OVERVIEW

Image processing is a rapidly growing field of research. With the advent of digital photography, Image Understanding has become increasingly important.

Image Semantic understanding is typically formulated as a Single label or Multi-Label learning problem. In Single label learning, each image will be categorized into one and only one of a set of predefined categories. In other words, only one label will be assigned on each image in this setting. The entire image is considered as one Single label. In multi label setting, which is closer to real world applications, each image will be assigned with one or multiple labels from a predefined label set, such as sky, mountain, and water. This paper is about multi-label learning (MLL) for image classification.

1.2 ABOUT THE PROJECT

1.2.1 Objectives

Common people can easily understand the image on seeing the image with their naked eyes but that is not the case with the blind people. This project is intended to help such people.This project gives a better understanding of the objects in the image and also gives the spatial relationship between the objects in the image, which is descried in the form of textual description. This output can later be converted into text to speech.

1.2.2 Problem Statement

Given an image, the image is converted to textual description of that image which can be given as an input. The process involved in image to textual description includes image processing like Pre-Processing, Image Segmentation, Edge Detection, Segmentation, Feature Extraction. In addition to this, Multi Label Annotation is involved for better understanding of image.

1.2.3 Need For Multi-Label Image Annotation

This paper proposes a system to incorporate multiple labels in a single image as opposed to the previous sets of approaches which considers the entire image as a single label.

This approach gives the textual description of the image with respect to the spatial relationship of the various objects present in the image, thus providing a better understanding of the image semantically.

1.3 OVERVIEW OF THESIS

Chapter 2 discusses the related work in Image Understanding and the approaches carried out thus far. It also gives the advantages and disadvantages of each approach. Chapter 3 gives the requirements analysis of the system. It explains the functional and non-functional requirements, constraints and assumptions made in the implementation of the system and the various UML diagrams. Chapter 4 explains the overall system architecture and the design of various modules along with their complexity. Chapter 5 gives the implementation details of each module, describing the algorithms used. Chapter 6 elaborates on the results of the implemented system and gives an idea of its efficiency. It also contains information about the dataset used for testing and other the observations made during testing. Chapter 7 concludes the thesis and gives an overview of its criticisms. It also states the various extensions that can be made to the system to make it function more effectively.

CHAPTER 2

RELATED WORK

This chapter gives a survey of the possible approaches to Image Annotation. We proposed to work with Multi-Label Image Annotation, which identifies all possible objects in the image. The output is a textual description about the objects in the image with respect to its spatial relationship. Thus this survey helped us analyse the various existing approaches to IA and decide on the one which would best cater to our needs.

2.1 MULTI LABEL ANNOTATION

The typical solution of multi-label classification is to translate the multi-label learning task into a set of single-label classification problems. Paper [1] solved the multi-label scene classification problem by building individual classifier for each label. The labels of a new sample are determined by the outputs of these individual classifiers.The disadvantage of the above paper however is that it ignores the correlation among labels.

To exploit these correlations, some researchers have proposed fusion-based methods [3]. Godbole et al. [3] proposed to leverage the correlations by adding a contextual fusion step based on the outputs of the individual classifiers.

More sophisticated MLL approaches model labels and correlations between labels simultaneously [4] [5]. Kang et al. [4] developed a Correlation Label Propagation (CLP) approach to explicitly capture the interactions between labels. Rather than treating labels independently, CLP simultaneously co-propagates multiple labels from training examples to testing examples.

2.2 JOINT MULTI LABEL MULTI INSTANCE LEARNING FOR IMAGE CLASSIFICATION

This method proposed an approach to take care of both multi label and multi instance learning for Image classification. Paper [7] proposed this approach by which both can be taken into consideration. Similarly, Paper [2] also proposed for the combining of Multi Instance and Multi Label into account with the hierarchical context and its application for Multi Media.

However with the use of more classifiers, these approaches takes a longer time as in compared to either multi instance or multi label learning.

Paper [6] proposes a new approach to learning a discriminative model of object classes, incorporating appearance, shape and context information efficiently. However this does not fully satisfy the concept of multi label image annotation where the focus is on the shape, appearance of the object rather than their spatial relationship and other factors.

2.3 OBSERVATIONS FROM THE SURVEY

Following are some of the observations from the survey of the various papers.

- The first paradigm approach treats an image as an indiscrete unit and do not capture the semantic meanings of the regions which actually contribute to the corresponding labels.
- The approaches cannot model the dependencies among the

regions which are also helpful for improving the classification performance.

Therefore we propose a system of Multi-Label Image Annotation, which considers all the objects in the image as well as the spatial relationship between them for better understanding of the image.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

The system outputs a textual description for the given image. The output sentence should adhere to the following requirements:

- The textual description should convey the information about the objects in the image.
- The textual description should consider the spatial relationship between the objects present in the image.

3.2 NON FUNCTIONAL REQUIREMENTS

3.2.1 User Interface

There must be a simple and easy to use user interface where the user should be able to choose an input image. The intermediate images and also the output textual description should be displayed in the screen.

3.2.2 Hardware

No special hardware interface is required for the successful implementation of the system.

3.2.3 Software

- Operating System : Windows
- Programming Language : MATLAB
- Database : Visual Saliency and Categorization of Abstract Images dataset

- Tools : MATLAB R2013a and higher

3.2.4 Performance

The system must be optimized, reliable, consistent and available all the time.

3.3 CONSTRAINTS AND ASSUMPTIONS

3.3.1 Constraints

The system would work only for those sets of images in a particular domain. More sets of images can be added for better results.

- The output generated would be correct only for those domain for which the images are in the database. For other images, the textual description would be incorrect. However, the system can be easily made to work for those by just adding the images in the database for the new domain.
- The accuracy of the Feature Extraction,Similarity Computation determines the efficiency of the Object Recognition and later the textual description. The errors in the Feature Extraction would propagate down to the rest of the modules also.

3.3.2 Assumptions

- The input image is assumed to be in proper JPEG format only.
- The input image should fall under the domain already used to create the database.
- The input image should preferably be an abstract image for better textual description.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The block diagram of the entire system is shown in figure 4.1. The whole system has been implemented using MATLAB and works in all versions starting from version R2013a and higher. Pre-Processing for Noise Removal, Feature Extractor and Object Recognition has been developed. Algorithm for Feature Extractor and Object Recognition has been developed and an algorithm for textual description from the Spatial Relationship of Objects has been devised.

The system aims at identifying all possible objects in a given image and providing a textual description of the image as output by semantically understanding them. Clip-Arts from the original images were manually trained and inserted into the database during the Training Phase. The input to the system is an image from the domain which was considered during Training Phase. The Pre-Processor enhances the image quality by removing Noise(Haze, Mist, Fog etc.,) from the image. Features from the individual clip-arts are already extracted. Features from the input image are then extracted and compared with every individual clip-art for similarity. It is then used to identify Objects in the image using Blob Detector.

The objects are then names using Image Annotation and the spatial relationships between objects are used to give a short textual description about the objects in the image.

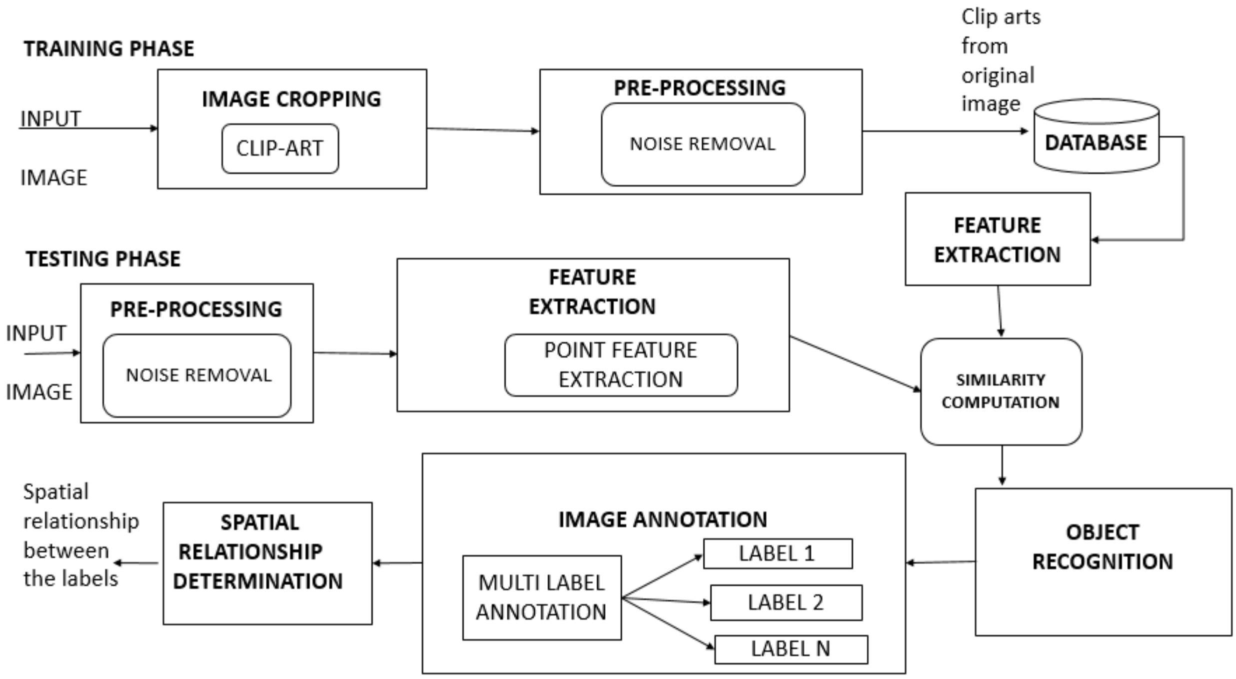

Figure 4.1 System Architecture

4.2 LIST OF MODULES

The list of modules of the whole system is given below.

- Preprocessing
- Feature Extractor
- Similarity Computation
- Object Recognition
- Multi Label Image Annotation
- Spatial Relationship Determination

4.3 SYSTEM MODELS

4.3.1 Sequence Diagram for the System

The sequence diagram for the system is depicted in Figure 4.2.

This sequence diagram depicts the sequential flow of the whole system.

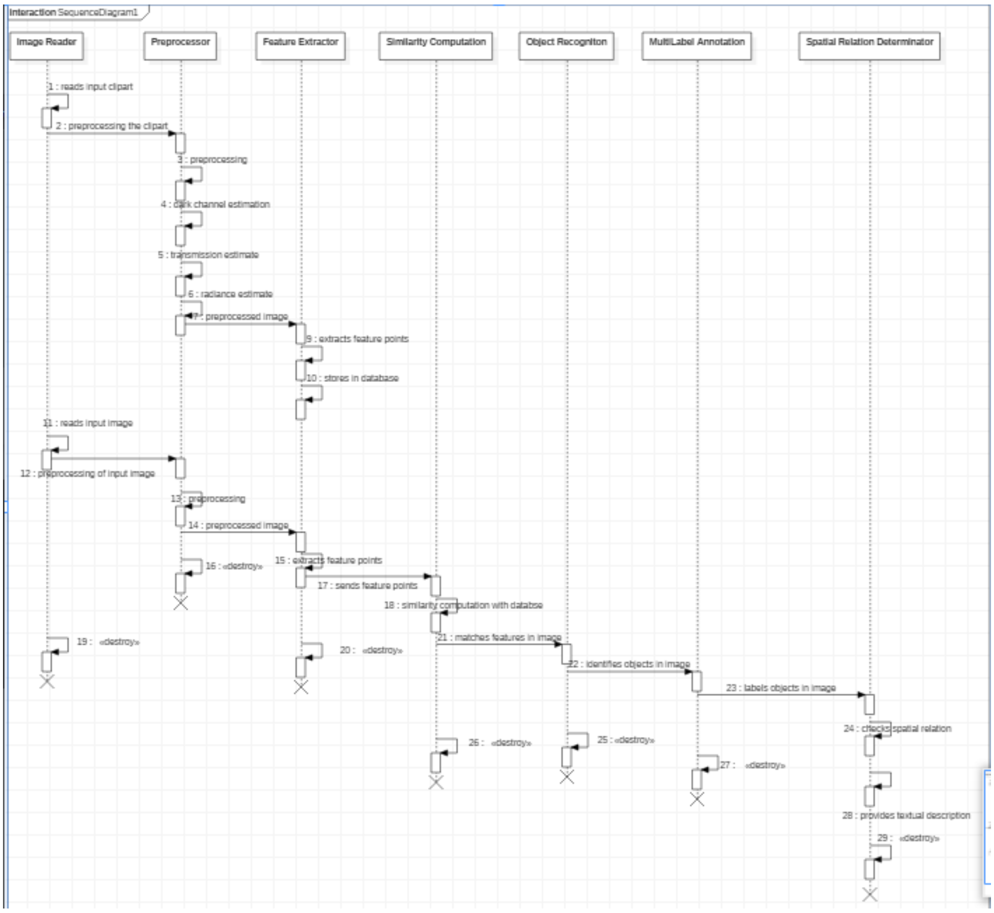

Figure 4.2 Sequence Diagram for the system

4.3.2 Use-Case Diagram for the System

The Use-Case diagram for the system is depicted in Figure 4.3.

4.4 MODULE DESIGN

The process is incorporated in two phases

- Training Phase
- Testing Phase

4.4.1 Training Phase

In this phase, a set of images is trained and saved in database. This phase involves some steps such as, image cropping and pre-processing.

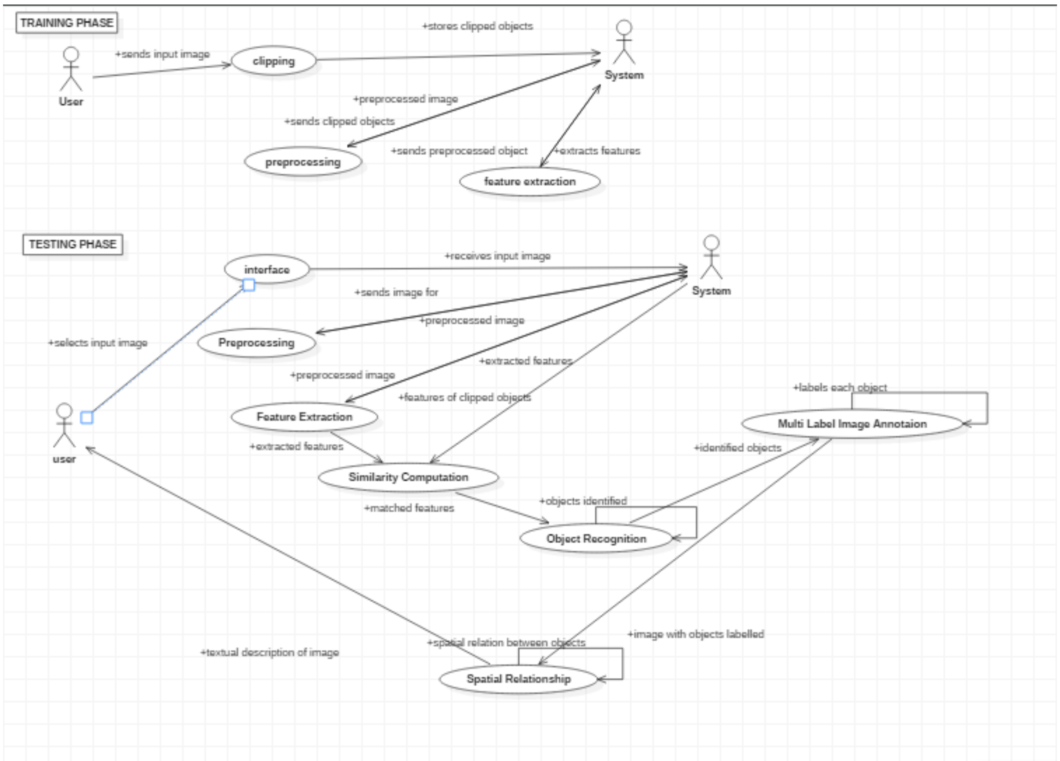

Figure 4.3 Use-Case Diagram for the system

IMAGE CROPPING

Cropping refers to the removal of the outer parts of an image to improve framing, accentuate subject matter or change aspect ratio. A set of images which needs to be trained is cropped using Crop image Interactive Tool. Open Crop Image tool associated with this image. Specify a variable in which to store the cropped image.

[I2, rect] = imcrop(I)

The Crop Image tool blocks the MATLAB command line until you complete the operation. Using the mouse, draw a rectangle over the portion of the image that you want to crop. We can store the cropped object from the image which is then pre-processed to store in the database. A set of images is given as input. The output will be the cropped object from the set of images.

PRE-PROCESSING

Noise is removed from the output object from image cropping stage during this stage using single image haze removal using dark channel prior. The object which has minimum intensity in such a patch should has a very low value. Formally, for an image J, we define,

$$\mathbf{Jdark(x) = min\ cr,g,b\ (\ min\ y(x)\ (Jc\ (y)))}$$

where Jc is a color channel of J and (x) is a local patch centred at x. The minimum intensity is considered as noise thus that should be removed. Thus, final output has the object with maximum intensity. A set of objects is given as input. A set of objects in which noise is removed is got as output which is stored in database for further usage.

The Use case diagram for this module is given in the Figure 4.4

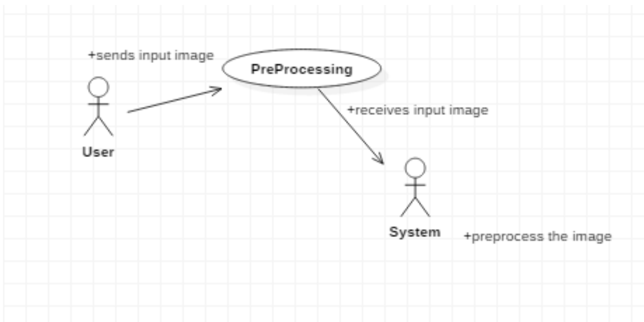

Figure 4.4 Use-Case Diagram for Pre Processing

4.4.2 Testing Phase

In this phase, a single image is given as input in which Similarity Computation is with the object in the database. Then the objects in the image are labelled using Object Recognition using Clustering and Blob Detector method. Then the Spatial Relationship between the objects are found from the given image and a short Textual description is given.

PRE-PROCESSING

The Pre-Processing stage here is similar to the one in the Testing Phase.

Noise is removed from the given image during this stage using single image haze removal using dark channel prior. The image with minimum intensity in such a patch should has a very low value. Formally, for an image J, we define,

Jdark(x) = min cr,g,b (min y(x) (Jc (y)))

where Jc is a color channel of J and (x) is a local patch centred at x. The minimum intensity is considered as noise thus that should be removed. Thus, final output has the image with maximum intensity. A single image is given as input. A single image in which noise is removed which is as input to feature extraction. The algorithm for the same is similar to the Pre-processing algorithm given in the Training Phase.

The Sequence Diagram for this module is given in Figure 4.5

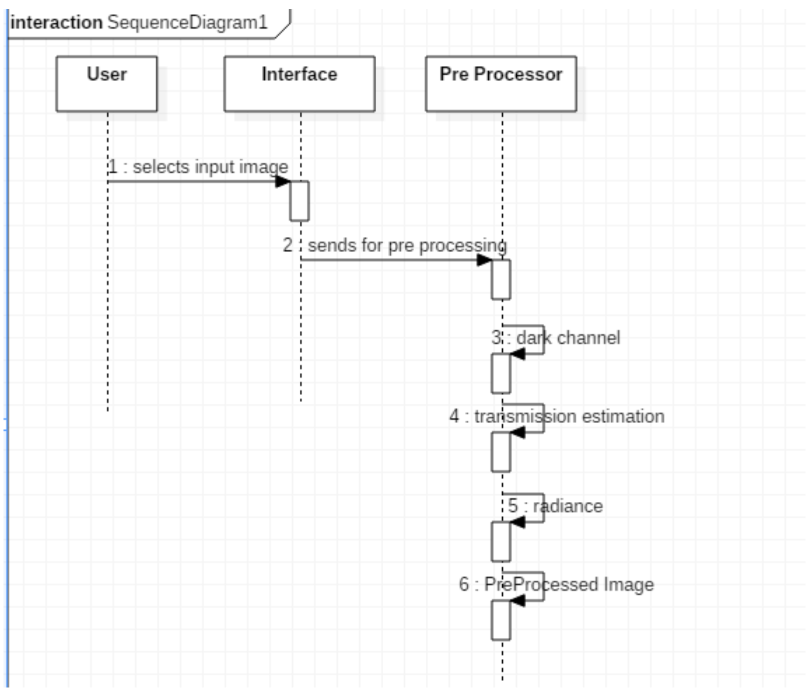

Figure 4.5 Sequence Diagram for Pre Processing

FEATURE EXTRACTION

Noise removed image and object from the database is given as input to feature extraction. In this module feature are extracted using SURF (Speeded Up Robust Features) method which include point features. Interest points can be found at different scales, partly because the search for correspondences often requires comparison images where they are seen at different scales. In other feature detection algorithms, the scale space is usually realized as an image pyramid. Images are repeatedly smoothed with a Gaussian filter, then they are sub sampled to get the next higher level of the pyramid. Therefore, several floors or stairs with various measures of the masks are calculated:

$$\sigma approx = Current filtersize * \frac{BaseFilterScale}{BaseFilterSize}$$

The Use case diagram for the module is given in Figure 4.6

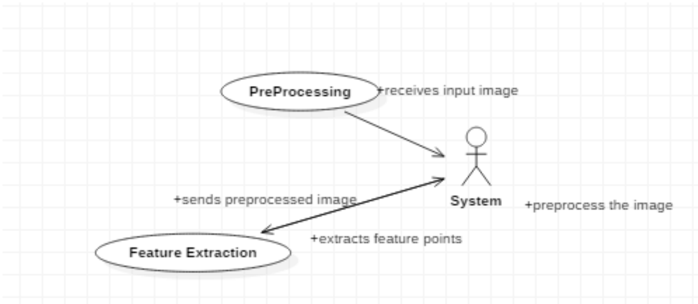

Figure 4.6 Use Case Diagram for Feature Extraction

The Sequence Diagram for this module is given in Figure 4.7

SIMILARITY COMPUTATION

In this module, feature extracted from image is computed for similarity to features extracted from object which are stored in the database. The determinant of a Hessian matrix expresses the extent of the response and is an expression of the local change around the area.

The Use case diagram for the module is given in Figure 4.8

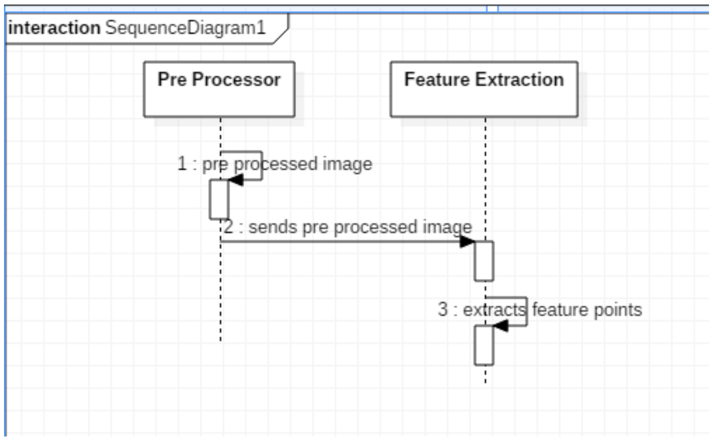

Figure 4.7 Sequence Diagram for Feature Extraction

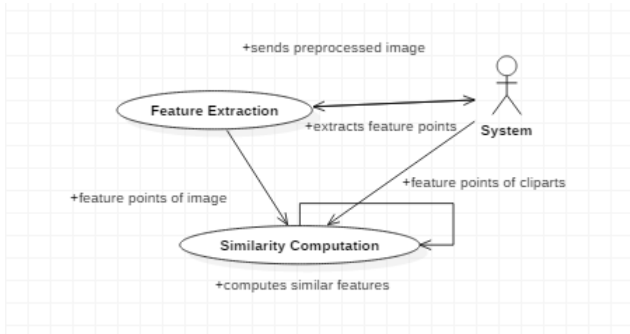

Figure 4.8 Use Case Diagram for Similarity Computation

The Sequence Diagram for this module is given in Figure 4.9

OBJECT RECOGNITION

In this module, object is recognized in image using clustering method if feature matches between the object and image. The output of this module is shown as rectangle is drawn in the image where the object is recognized. Finally locate the object in scene using matched points.

In order to locate the object in scene estimate geometric transformations i.e. affine transform. Estimate Geometric Transform calculates the transformation relating the matched points, while

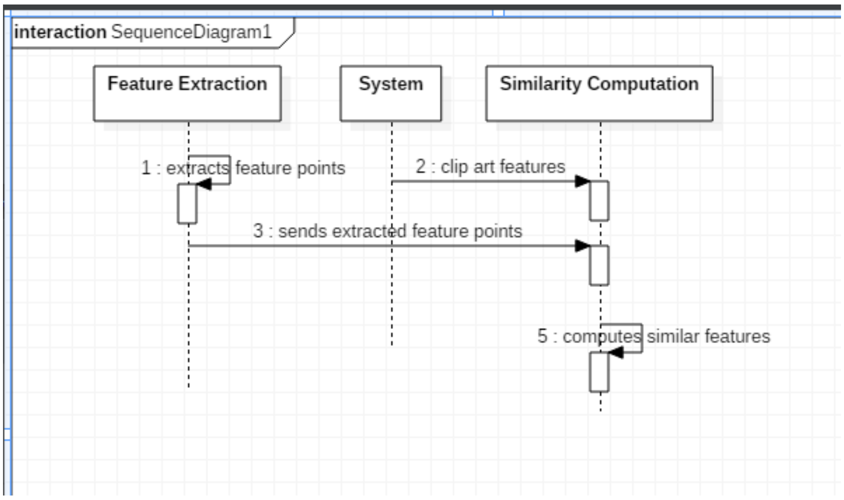

Figure 4.9 Sequence Diagram for Similarity Computation

eliminating outliers. This transformation allows us to localize the object in the scene and finally transform the reference image into the coordinate system of the target image. The transformed image indicates the location of the object in the scene.

The Use case diagram for the module is given in Figure 4.10

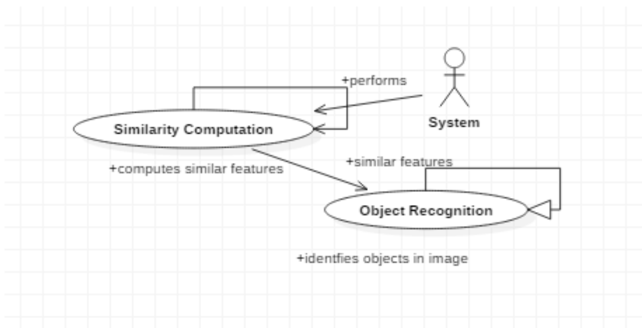

Figure 4.10 Use Case Diagram for Object Recognition

The Sequence Diagram for this module is given in Figure 4.11

MULTI LABEL IMAGE ANNOTATION

In this module, the objects which are recognized in before stage are labelled using multi label annotation. The output of this stage will be

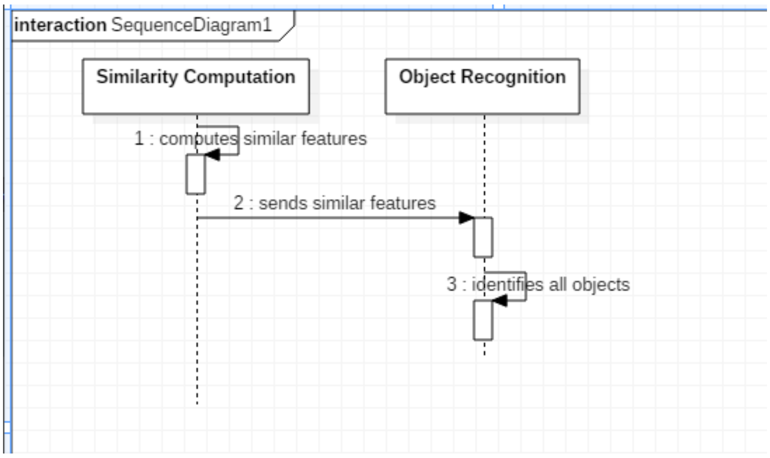

Figure 4.11 Sequence Diagram for Object Recognition

shown as multi-label is annotated in the given image for objects which are matched from the database.

The Use case diagram for the module is given in Figure 4.12

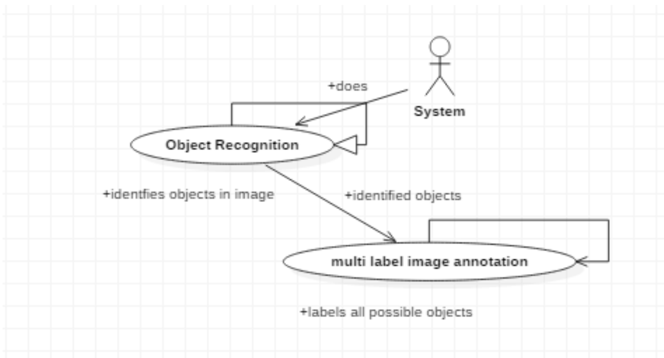

Figure 4.12 Use Case Diagram for Image Annotation

The Sequence Diagram for this module is given in Figure 4.13

SPATIAL RELATIONSHIP DETERMINATION

In this module, spatial relationship is found between the objects in the given image which include location of the particular object in that image. Thus output will be textual description of the object in the image based on their spatial relationship.

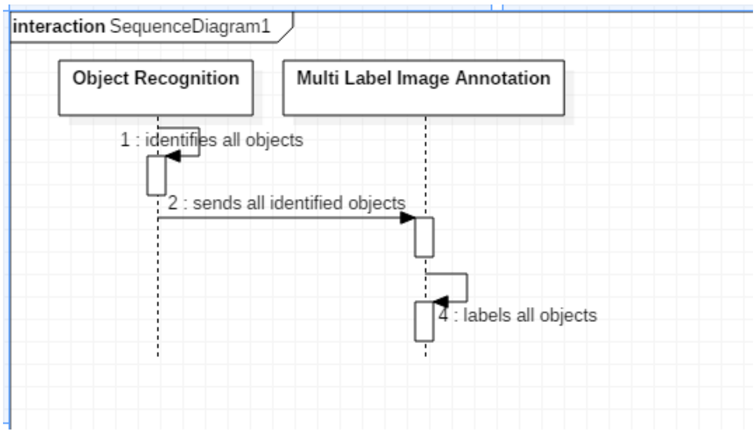

Figure 4.13 Sequence Diagram for Image Annotation

The Use case diagram for the module is given in Figure 4.14

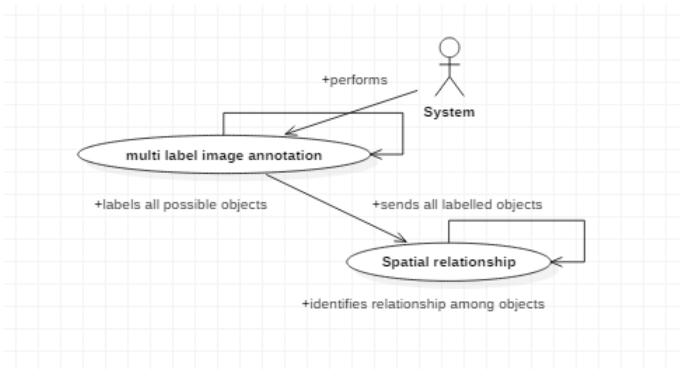

Figure 4.14 Use Case Diagram for Spatial Relationship

The Sequence Diagram for this module is given in Figure 4.15

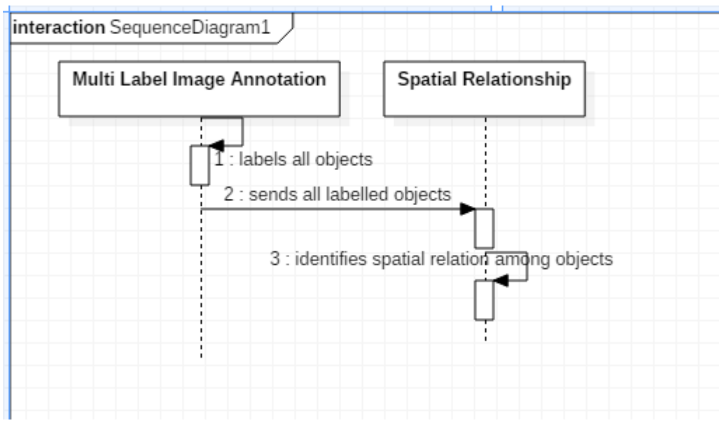

Figure 4.15 Sequence Diagram for Spatial Relationship

CHAPTER 5

SYSTEM DEVELOPMENT

An overview of the entire algorithm of the system is given below. The input image is given to the Pre-Processor which produces the pre-processed image of better quality, followed by Feature Extractor which identifies all the feature points in the image and stores them. It is then compared for Similarity with the features of the Clip-Art images, which are already stored in the database. Based on the similarity, the Objects in the image are recognized and labelled using Multi-Label Annotation. These labels are then compared to check the spatial relationship among them and a short description is produced as output to the user.

5.1 PROTOTYPES ACROSS THE MODULES

The input and output to each module of the system is described in this section.

- **PRE-PROCESSING:** The input to this module is the clipped objects from the image during the Training Phase and the entire image during the Testing Phase. The output is the enhance quality of the image where the noise such as Mist,Haze,Fog are removed.
- **FEATURE EXTRACTION:** This module takes the pre-processed image as input and gives out all the feature points in the image and clip-arts as the output.
- **SIMILARITY COMPUTATION:** This module takes two inputs, features of the clipped object and the features of the

original image and compares them. The output is the mapping of the points in the image whose features matches with any of the features of any of the clipped object stored in the database.

- **OBJECT RECOGNITION:** The input to this module is the matched feature points on the original image. This module recognizes the individual object in the image and marks them as output.
- **MULTI-LABEL IMAGE ANNOTATION:** The input to this module are the various objects recognized in the entire image. This module labels multiple objects in the same image as the output.
- **SPATIAL RELATIONSHIP DETERMINATION:** This module takes as input all the possible objects identified and labelled in the previous stages. The output is the short textual description about the various objects in the image.

5.2 ALGORITHMS OF THE MODULES

5.2.1 Pre-Processing

The Pre-Processor algorithm is given below. It takes the input image and removes noise from the image. The algorithm used is Haze Removal Algorithm.

The intermediate output of the Pre Processing Module is given in Figure 5.1

5.2.2 Feature Extraction - Training Phase

The Feature Extraction Algorithm is given below. The algorithm used is SURF(Speed Up Robust Features) Algorithm. In Training Phase, the features are extracted for Clip Art Images.

The intermediate output of the Feature Extraction Module in the

Algorithm 1 Preprocessing

$filename =' image';$
$ext =' .jpg';$
$scalingFactor = 0.5;$
$imageRGB = imresize(readIm(filename, ext), scalingFactor);$
$imwrite(imageRGB, fullfile('Images', strcat(filename, ext)));$
$dark = darkChannel(imageRGB);$
$imwrite(dark, fullfile('Images', strcat(filename,'_d ark', ext)));$
$transmission = transmissionEstimate(imageRGB, dark);$
$imwrite(transmission, fullfile('Images', strcat(filename,'_t rans', ext)));$
$radiance = getRadiance(atm, imageRGB, transmission);$
$imwrite(radiance, fullfile('Images', strcat(filename,'_r ad', ext)));$

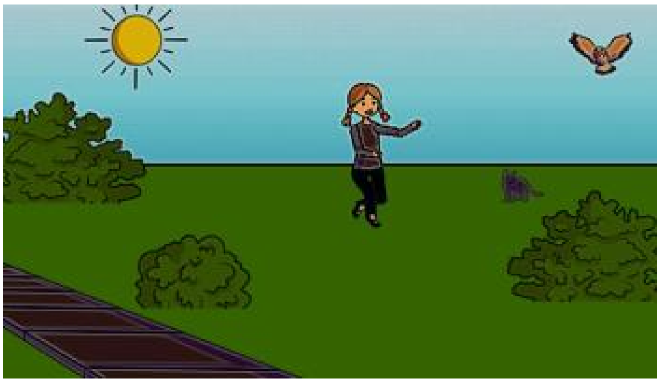

Figure 5.1 Pre-Processed Image

Algorithm 2 Feature Extraction of Clip-Art Images

$filenames = dir(fullfile(image_folder,' *.jpg'));$
$total_images = numel(filenames);$
for $n = 1 : total_images$ **do**
  $boxImage1 = imread(fullname);$
  $boxImage = rgb2gray(boxImage1);$
  $boxPoints = detectSURFFeatures(boxImage);$
  $plot(selectStrongest(boxPoints, 100));$
  $[boxFeatures, boxPoints] = extractFeatures(boxImage, boxPoints);$
end for

Figure 5.2 Feature extraction from individual object

Training Phase is given in Figure 5.2

5.2.3 Feature Extraction - Testing Phase

The Feature Extraction Algorithm is given below. In the Testing Phase, the features are extracted for the entire image.

Algorithm 3 Feature Extraction of Input Image

$filename = dir(fullfile(image_{f}older,' *.jpg'));$

$total_{i}mages = numel(filenames);$

$sceneImage1 = imread(fullname);$

$sceneImage = rgb2gray(sceneImage1);$

$scenePoints = detectSURFFeatures(sceneImage);$

$plot(selectStrongest(scenePoints, 300));$

$[sceneFeatures, scenePoints] = extractFeatures(sceneImage, scenePoints);$

The intermediate output of the Feature Extraction Module in the Testing Phase is given in Figure 5.3

5.2.4 Similarity Computation

The code for the Similarity Computation method is given below. This module compares the clip art image and the original image for similar feature points.

The intermediate output of the Similarity Computation Module is given in Figure 5.4

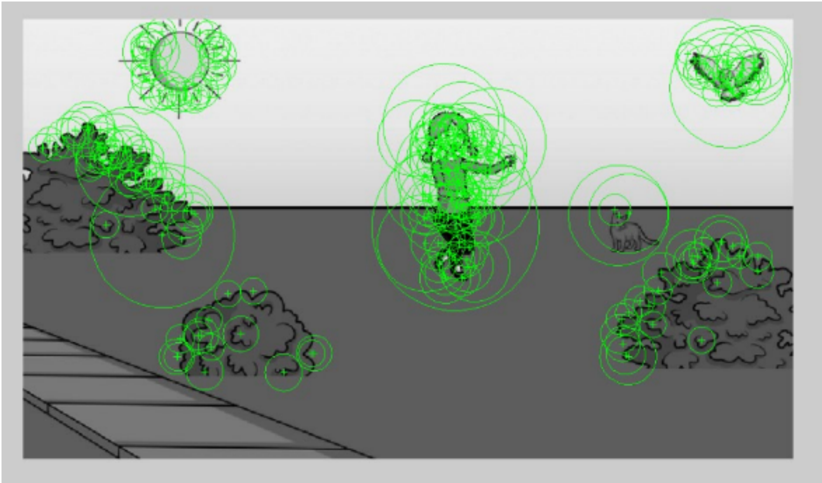

Figure 5.3 Feature Extraction from the input image

Algorithm 4 Similarity Computation

$boxPairs = matchFeatures(boxFeatures, sceneFeatures);$
if isempty(boxPairs) **then**
  $disp('Nomatchesfound');$
else
  $matchedBoxPoints = boxPoints(boxPairs(:,1),:);$
  $matchedScenePoints = scenePoints(boxPairs(:,2),:);$
  $figure;$
  $showMatchedFeatures(boxImage, sceneImage, matchedBoxPoints, \ldots$
  $matchedScenePoints, 'montage');$
end if

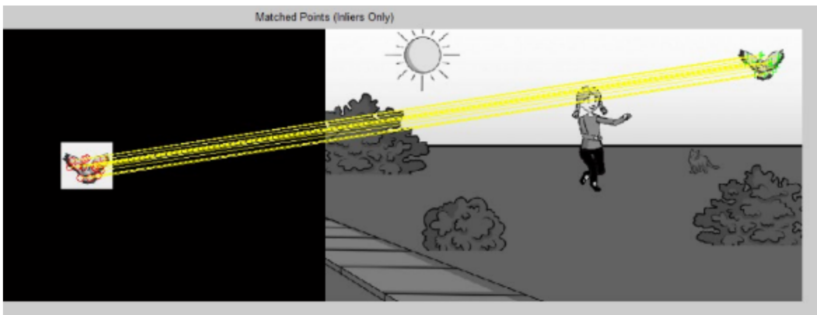

Figure 5.4 Similarity Computation

5.2.5 Object Recognition

This module draws a box around the objects that have been recognized in the image. The algorithm for the same is given below:

Algorithm 5 Object Recognition

$[tform, inlierBoxPoints, inlierScenePoints, status] = ...$
$estimateGeometricTransform(matchedBoxPoints, matchedScenePoints,'affine'$
if (status==1) **then**
  $disp('notenoughpoints')$
else
  $figure;$
  $showMatchedFeatures(boxImage, sceneImage, inlierBoxPoints, ...$
  $inlierScenePoints,'montage');$
  $title('MatchedPoints(InliersOnly)');$
  $boxPolygon = [1, 1; ...$
  $size(boxImage, 2), 1; ...$
  $size(boxImage, 2), size(boxImage, 1); ...$
  $1, size(boxImage, 1); ...$
  $1, 1];$
  $newBoxPolygon = transformPointsForward(tform, boxPolygon);$
  $figure;$
end if

The intermediate output of the Object Recognition Module is given in Figure 5.5

5.2.6 Multi-Label Image Annotation

This module labels all the objects recognized in the image. The algorithm is as follows:

The intermediate output of the Multi Label Image Annotation Module is given in Figure 5.6

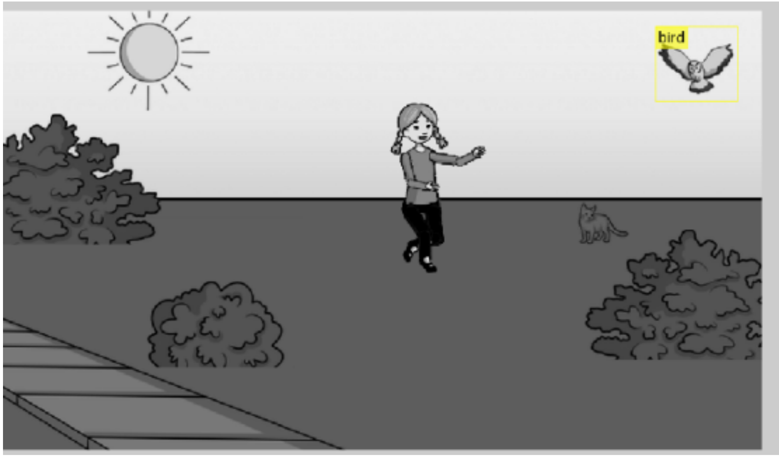

Figure 5.5 Object Recognition

Algorithm 6 Multi Label Image Annotation

$position = [newBoxPolygon(5,1), newBoxPolygon(1,2)];$
$[pathstr, name, ext] = fileparts(filenames(n).name);$
$Rgb = insertText(sceneImage, position, name);$
$imshow(Rgb);$
$holdon;$
$line(newBoxPolygon(:,1), newBoxPolygon(:,2), 'Color', 'y');$
$sceneImage1 = Rgb;$
$sceneImage = rgb2gray(sceneImage1);$
$imshow(sceneImage);$

Figure 5.6 Multi Label Image Annotation

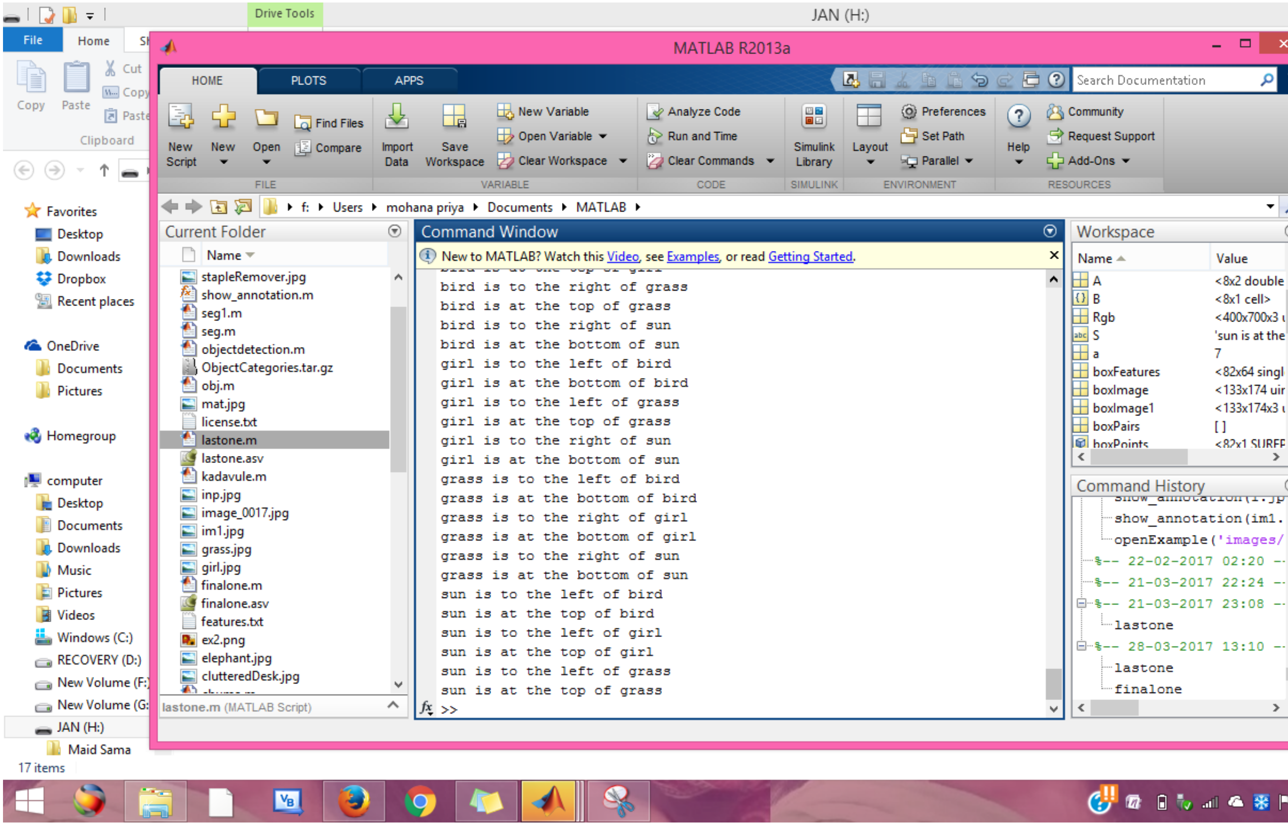

Figure 5.7 Textual Description

5.2.7 Spatial Relationship Determination

The output to this module is a Textual description, that is displayed on the command window. For the above image, some of the possible description are as follows;

Girl is to the left of the bird

Bird is to the right of the sun The algorithm for the Spatial Relationship Module is as follows:

5.3 DEPLOYMENT DETAILS

The deployment of the system requires any version of Matlab, most preferably from MatLab version R2013a and higher.

Algorithm 7 Spatial Relationship Determination

$filenames = dir(fullfile(image_folder,'*.jpg'));$
$total_images = numel(filenames);$
for $n = 1 : total_images$ **do**
  if $obj1(A(n,1)) \geq obj2(A(a,1))$ **then**
    $obj1\ is\ to\ the\ right\ of\ obj2$
  else
    $obj1\ is\ to\ the\ left\ of\ obj2$
    if $obj1(A(n,2)) \geq obj2(A(a,2))$ **then**
      $obj1\ is\ to\ the\ bottom\ of\ obj2$
    else
      $obj1\ is\ to\ the\ top\ of\ obj2$
    end if
  end if
end for

CHAPTER 6

RESULTS AND DISCUSSION

6.1 DATA SET FOR TESTING

The input to the entire system consists of atleast 100 abstract clipped objects and images, each image in the form of JPG/JPEG/PNG with size not exceeding 500KB in the least. Each module of the system was also tested separately. The system was tested with abstract images for the domains considered.

The Dataset for the project consists of Clipped images from the project which are trained and stored in the database in the Training Phase, which is shown in Figure 6.1

The Dataset for the project consists of images which are used to test the working of the entire project. Several sets of images were considered among the Abstract Images Domain, which is shown in Figure 6.2

6.2 SAMPLE SCREENSHOTS DURING TESTING

The input image and the output image windows are shown below. The interface in which the input image is selected is shown in Figure 6.3 The output image in which the Textual description of the image is produced is displayed in the command window in MATLAB. The output image is shown in Figure 6.4 and its corresponding textual description is shown in Figure 6.3

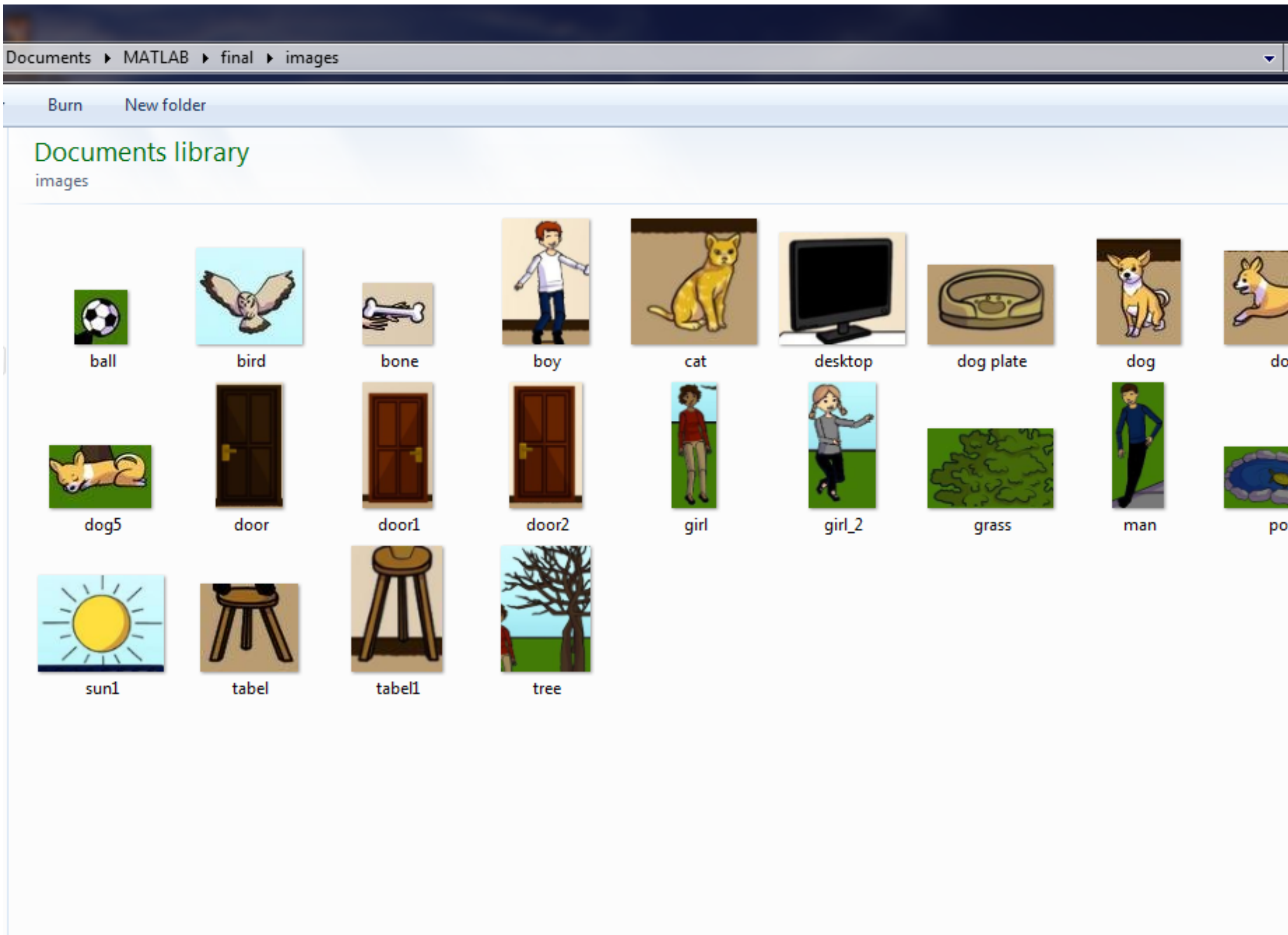

Figure 6.1 Data Set containing clipped images

6.3 PERFORMANCE EVALUATION

The performance of the entire system is evaluated using the standard parameters described below.

6.3.1 Precision , Recall - Feature Extraction

Precision is the fraction of retrieved feature points in the image that are *relevant* to the query:

$$\frac{T_p}{T_p + F_p} \tag{6.1}$$

where

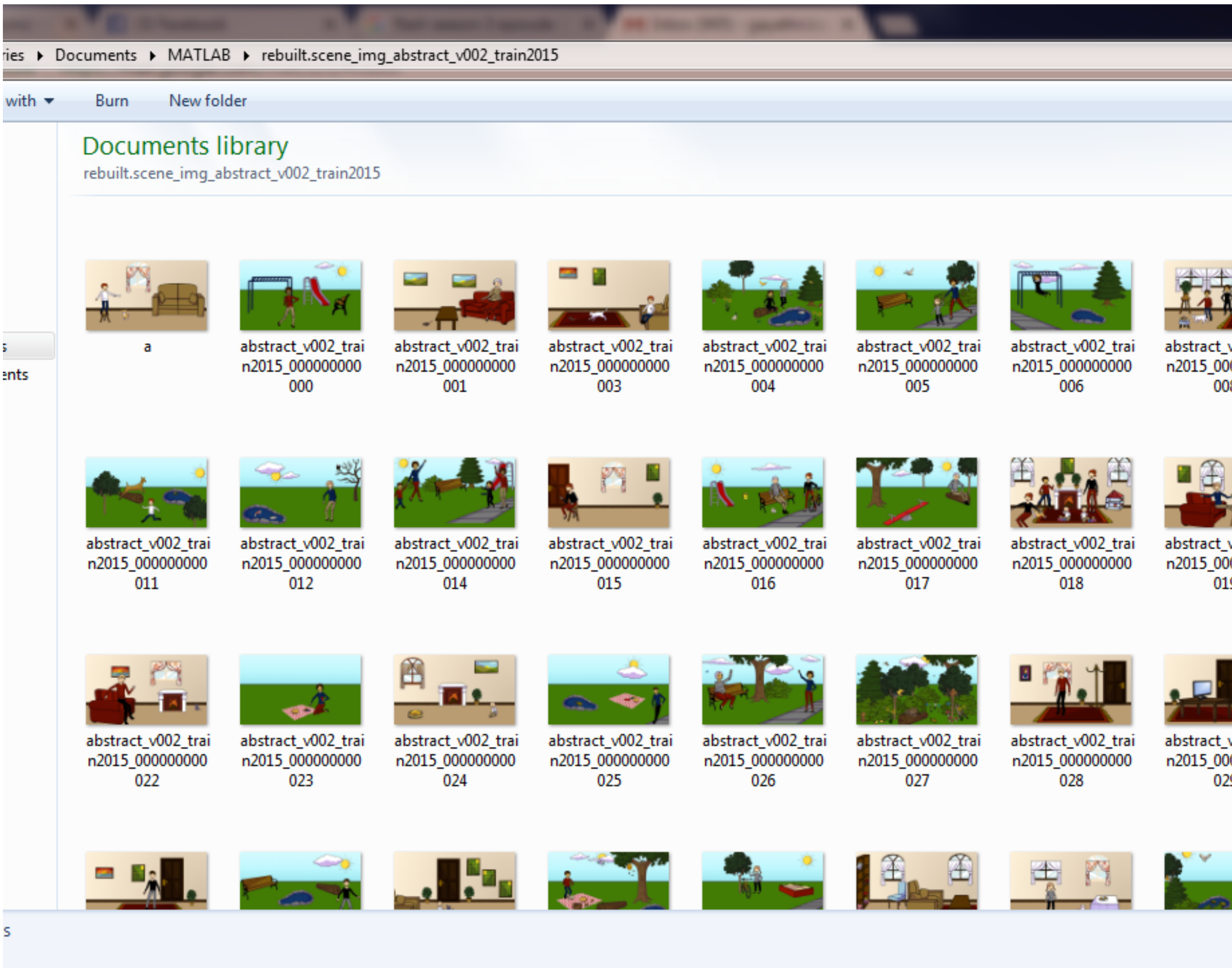

Figure 6.2 Data Set containing Test Images

- T_p : No. of precisely found feature points in the image.
- F_p : No. of false positives

Recall is the fraction of relevant feature points that are *extracted*

$$\frac{T_p}{T_p + F_n} \times 100\% \tag{6.2}$$

where

- T_p : number of correctly retrieved feature points.
- F_n : number of false negatives.

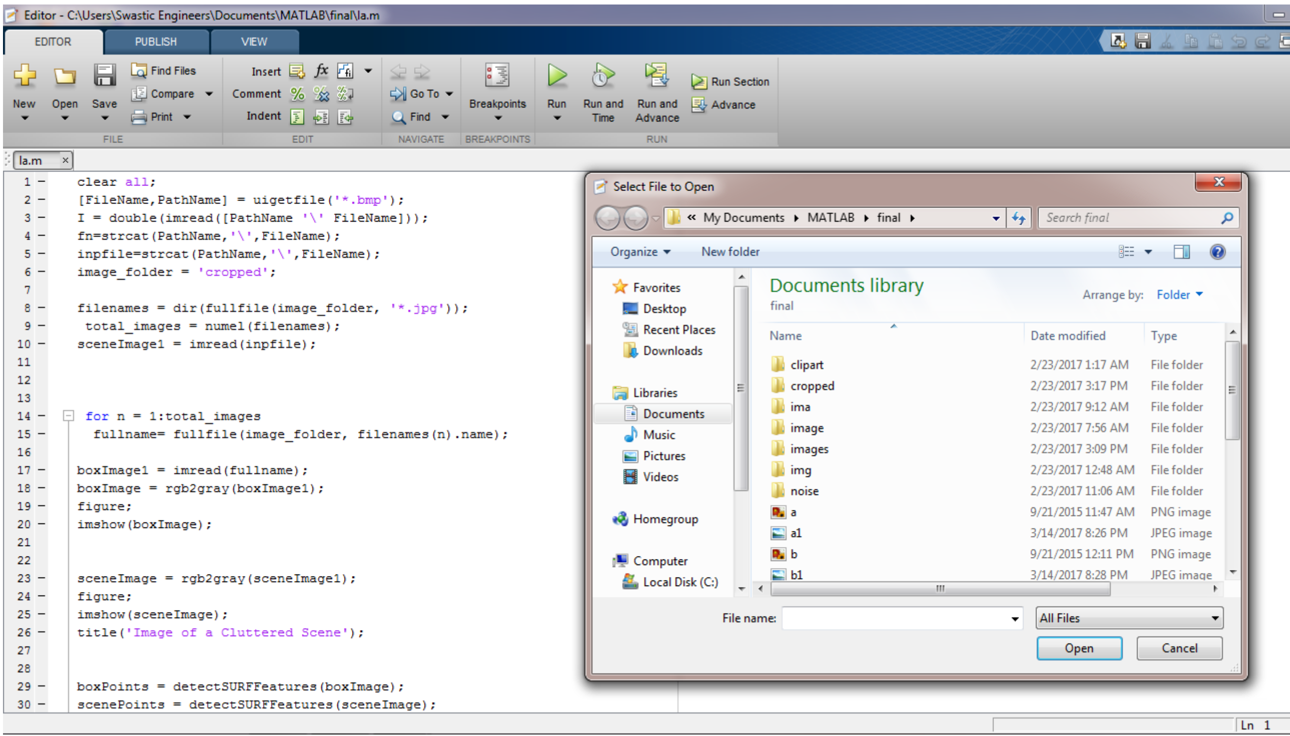

Figure 6.3 Interface to select input image

Figure 6.4 Output of Multi Label Annotation

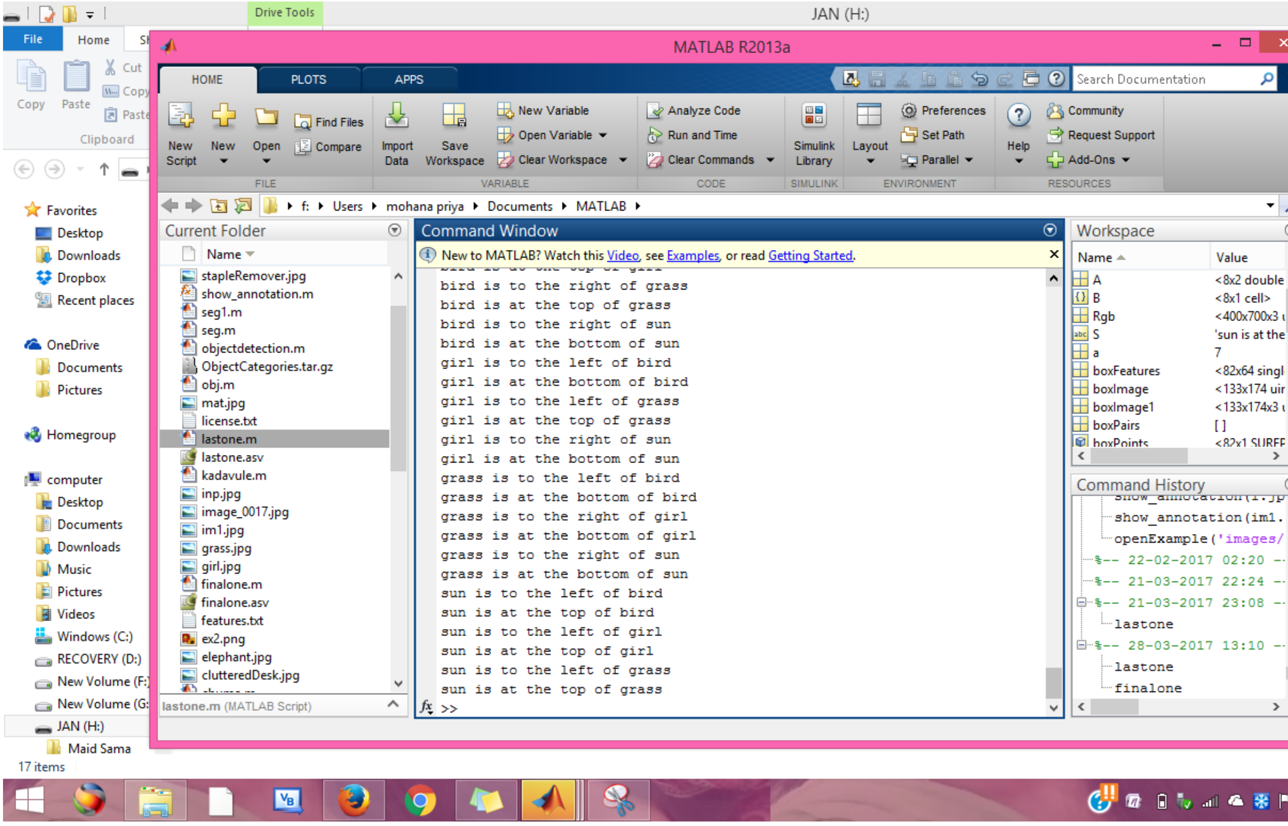

Figure 6.5 Textual Description

6.4.1.1 Result 1

The precision,recall and f-score values for the Feature extraction are depicted below:

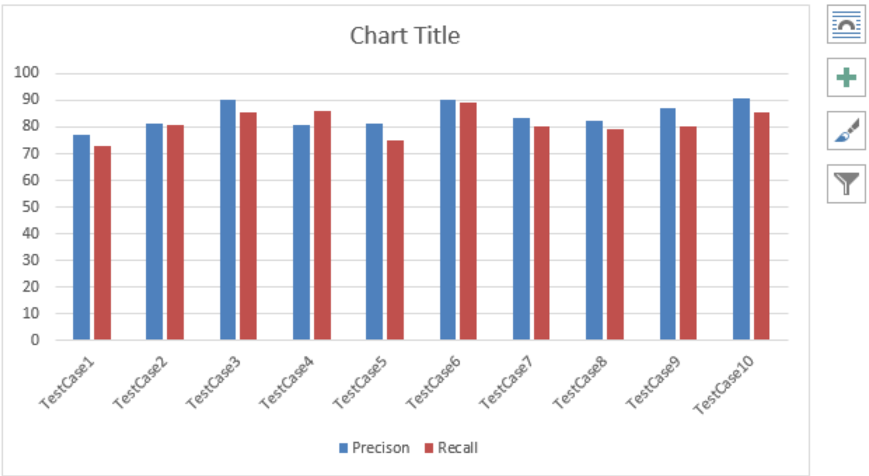

Figure 6.6 Bar chart of Performance Metric of Feature Extraction

Table 6.1 Precision, Recall and F-score for FE

Domain-Abstract Images	Precision	Recall	F-Score
TestCase1	77.29	72.91	0.85
TestCase2	81.29	85.02	0.87
TestCase3	90.22	85.36	0.87
TestCase4	80.99	86.13	0.88
TestCase5	81.35	74.78	0.78
TestCase6	90.14	89.34	0.90
TestCase7	83.36	80.42	0.82
TestCase8	82.26	79.34	0.82
TestCase9	87.13	80.04	0.82
TestCase10	90.84	85.52	0.88

For different sample images, Precision, Recall and F-score values vary different because of the quality in each of the images and the size of the images. On an average, the F-Score value is 0.84, which is very good for extracting the features.

The bar chart of the values computed is shown in Figure 6.5

6.3.2 Precision, Recall - Object Recognition

Precision is the fraction of recognized objects in the image that are *relevant* to the query:

$$\frac{T_p}{T_p + F_p} \tag{6.3}$$

where

- T_p : No. of precisely recognized objects in the image.
- F_p : No. of false positives

Recall is the fraction of relevant objects that are *recognized*

$$\frac{T_p}{T_p + F_n} \times 100\% \tag{6.4}$$

where

- T_p : number of correctly retrieved objects.
- F_n : number of false negatives.

6.4.1.2 Result 2

The precision,recall and f-score values for the Object Recognition are depicted below:

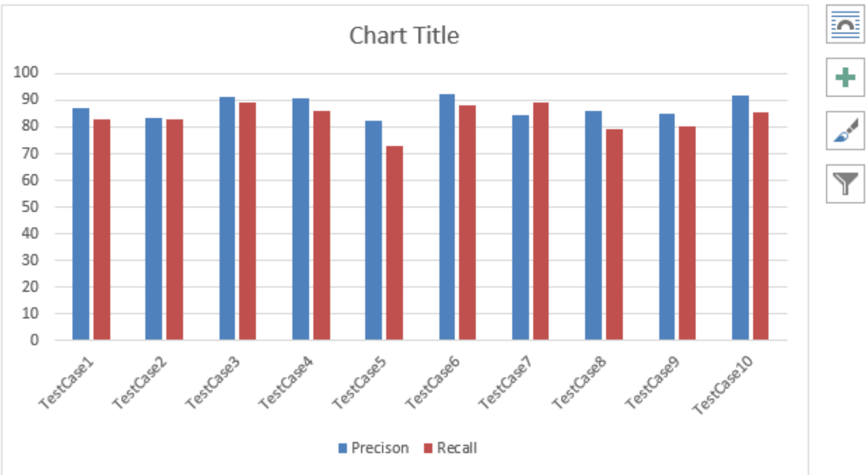

Figure 6.7 Bar chart of Performance Metric of Object Recognition

Table 6.2 Precision, Recall and F-score for OR

Domain-Abstract Images	Precision	Recall	F-Score
TestCase1	87.29	82.91	0.85
TestCase2	83.29	75.02	0.87
TestCase3	91.22	84.36	0.87
TestCase4	90.99	86.13	0.88
TestCase5	82.35	74.78	0.78
TestCase6	92.14	89.34	0.90
TestCase7	84.36	80.42	0.82
TestCase8	86.26	79.34	0.82
TestCase9	85.13	80.04	0.82
TestCase10	91.84	85.52	0.88

For different sample images, Precision, Recall and F-score values vary different because of the quality in each of the images and the size of the images. On an average, the F-Score value is 0.84, which is very good for object recognition.

The bar chart of the values computed is shown in Figure 6.6

6.3.3 Precision, Recall - Textual Description

Precision is the fraction of retrieved textual description in the image that are *relevant* to the query:

$$\frac{T_p}{T_p + F_p} \tag{6.5}$$

where

- T_p : No. of precisely retrieved textual description from the image.
- F_p : No. of false positives.

Recall is the fraction of relevant textual description that are *produced*

$$\frac{T_p}{T_p + F_n} \times 100\% \tag{6.6}$$

where

- T_p : number of correctly retrieved textual description.
- F_n : number of false negatives.

6.4.1.3 Result 3

The precision,recall and f-score values for the Textual Description are depicted below:

Table 6.3 Precision, Recall and F-score for TD

Domain-Abstract Images	Precision	Recall	F-Score
TestCase1	82.29	81.91	0.85
TestCase2	81.29	74.02	0.87
TestCase3	90.22	74.36	0.87
TestCase4	70.99	88.13	0.88
TestCase5	84.35	89.78	0.78
TestCase6	90.14	79.34	0.90
TestCase7	81.36	88.42	0.82
TestCase8	78.26	89.34	0.82
TestCase9	95.13	70.04	0.82
TestCase10	81.84	75.52	0.88

For different sample images, Precision, Recall and F-score values vary different because of the quality in each of the images and the size of the images. On an average, the F-Score value is 0.84, which is very good for extracting the features.

The bar chart for the performance metrics of Textual description is shown in Figure 6.7

6.3.4 Accuracy

The accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined.To make the context clear by the semantics, it is often referred to as the ”Rand accuracy” or ”Rand index”. It is a parameter of the test.

$$\frac{tp+tn}{tp+fp+tn+fn} \tag{6.7}$$

The accuracy will be 0.83 (i.e) 83%

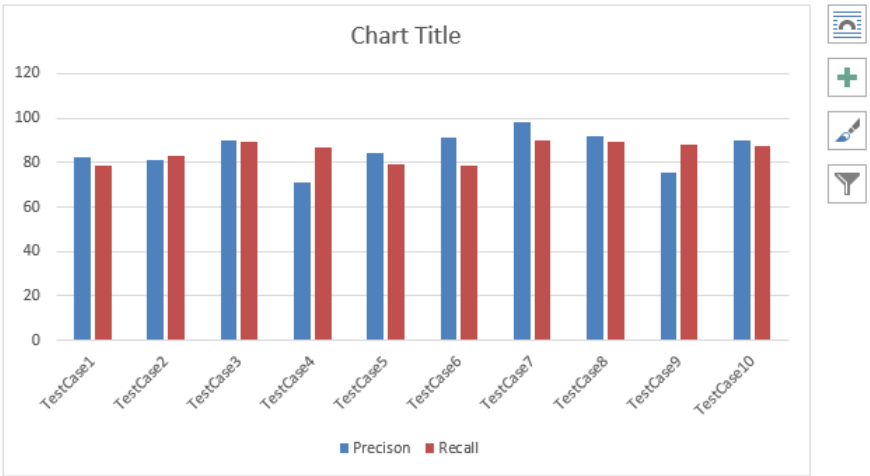

Figure 6.8 Bar chart of Performance Metric of Textual Description

6.4 TEST CASES FOR EACH MODULE

6.4.1 Pre-Processor

6.4.1.1 Test Pre-Requisite

The Input Image should be in a proper JPEG/PNG format.

6.4.1.2 Description

This module removes noise such as Haze from the given image thus enhancing the image quality.

6.4.1.3 Test Cases

- **TC-ID : 01**

  Input:

  An Input Image in any invalid format.

Expected Output:

Error resulting in termination of process.

- **TC-ID : 02**

  Input:

  An Input Image in which there is a presence of external factor such as Haze.

  Expected Output:

  Pre-Processed image with haze removed.

- **TC-ID : 03**

  Input:

  An Input Image in which there is no external factor such as Haze.

  Expected Output:

  The original image as same as the input image.

6.4.2 Feature Extraction

6.4.2.1 Test Pre-Requisite

The Input image in proper JPEG/PNG format with pre-processing done on the image.

6.4.2.2 Description

This module extracts all possible feature points from the image or the clipped object.

6.4.2.3 Test Cases

- **TC-ID : 01**

  Input:

  Pre-Processed Image is given as input.

  Expected Output:

  All possible features are found in the image.

- **TC-ID : 02**

  Input:

  Pre-Processed Image of relatively smaller size is given as input

  Expected Output:

  Error due to lack of feature points due to smaller image size.

- **TC-ID : 03**

  Input:

  Pre-Processed clipped object is given as input.

  Expected Output:

  Extraction of almost all feature points.

6.4.3 Similarity Computation

6.4.3.1 Test Pre-Requisite

The input to this module should be two input images for it to compute the similarity between them.The Input Image should be in a proper JPEG/PNG format.

6.4.3.2 Description

This module compares the similarity between the original image and the clipped image.

6.4.3.3 Test Cases

- **TC-ID : 01**

  Input:

  One Image in which all features are extracted and another clipped object stored in database.

  Expected Output:

  Matching of feature points in the input image with the nearest matching feature points in any of the clipped object in the database.

- **TC-ID : 02**

  Input:

  One Image in which all features are not extracted and another clipped object stored in database.

  Expected Output:

  Improper Matching of feature points in the input image with the nearest matching feature points in any of the clipped object in the database.

- **TC-ID : 03**

Input:

One Image in which all features are extracted and another clipped object stored in database.

Expected Output:

Matching of feature points in the input image with the nearest matching feature points in any of the clipped object in the database.

6.4.4 Object Recognition

6.4.4.1 Test Pre-Requisite

Input Image in which the features are matched with any of the clipped image in the database.

6.4.4.2 Description

This module identifies the objects present in the image, indicated by a box.

6.4.4.3 Test Cases

- **TC-ID : 01**

  Input:

  Image in which feature points are correctly matched

  Expected Output:

  All possible objects are found in the image.

- **TC-ID : 02**

Input:

Image in which feature points are not accurately matched.

Expected Output:

Not recognizing some objects in the image due to lack of feature matching.

- **TC-ID : 03**

Input:

Image in which feature points are correctly matched

Expected Output:

All possible objects are found in the image.

6.4.5 Multi Label Image Annotation

6.4.5.1 Test Pre-Requisite

Image in which atleast more than one object is identified.

6.4.5.2 Description

This module labels all the objects found in the image with its corresponding name.

6.4.5.3 Test Cases

- **TC-ID : 01**

Input:

Image in which all possible objects are found.

Expected Output:

All possible objects are labelled in the image.

- **TC-ID : 02**

  Input:

  Image in which all objects are not found.

  Expected Output:

  Labelling only those objects which are found.

- **TC-ID : 03**

  Input:

  Image in which no object is found.

  Expected Output:

  No Labelling is done to the image..

6.4.6 Spatial Relationship Determination

6.4.6.1 Test Pre-Requisite

The input image should atleast contain two labels to find the spatial relationship between them.

6.4.6.2 Description

This module compares a label with all other identified labels for the relationship between the label and all other labels in the image.

6.4.6.3 Test Cases

- **TC-ID : 01**

  Input:

  Image in which multiple labels are found.

  Expected Output:

  Textual Description about each label based on its spatial relation with every other label.
- **TC-ID : 02**

  Input:

  Image in which only two labels are found.

  Expected Output:

  Textual Description of the label based on its relation with other label.
- **TC-ID : 03**

  Input:

  Image in which only one label is found.

  Expected Output:

  No Textual Description as there is a lack of objects to understand their spatial relationship.

CHAPTER 7

CONCLUSIONS

7.1 SUMMARY

Visually impaired people cannot see the image and understand the semantic meaning in that image. This project helps them in knowing what is there in an image. Understanding all the objects in the image is not the only part in an Image Understanding concept. It is essential to know the relationship that exists among the objects within an image.

This paper proposes the concept of Multi-Label Image Annotation, in which almost all the objects in the image are found and labelled. Then the spatial relationship that exists between the objects in the image are taken into consideration to provide the final output, which is a short textual description about the relationship between the objects in the image. this can also be used as an input to Hardware system such as Braille through which they can understand the images.

7.2 FUTURE WORK

The Efficiency of the Similarity Computation module is not upto the expected level for the system. Therefore, better techniques can be deployed to ensure that the features are matched perfectly. This would provide better results to the entire system. Thus there are a lot of improvements and extensions possible and this system offers a wide scope in the field of Image Understanding.

REFERENCES

[1] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown, “Learning multi-label scene classification”, 2004.

[2] X. Ding, B. Li, W. Xiong, W. Guo, W. Hu, and B. Wang, “Multi-instance multi-label learning combining hierarchical context and its application to image annotation”, *IEEE Transactions on Multimedia*, vol. 18, num. 8, pp. 1616–1627, Aug 2016.

[3] Shantanu Godbole and Sunita Sarawagi, “Discriminative methods for multi-labeled classification”, In *In Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 22–30. Springer, 2004.

[4] Feng Kang, Rong Jin, and R. Sukthankar, “Correlated label propagation with application to multi-label learning”, In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1719–1726, 2006.

[5] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang, “Correlative multi-label video annotation”, In *Proceedings of the 15th ACM international conference on Multimedia*, pp. 17–26. ACM, 2007.

[6] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi, “Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation”, 2006.

[7] Zheng-Jun Zha, Xian-Sheng Hua, Tao Mei, Jingdong Wang, Guo-Jun Qi, and Zengfu Wang, “Joint multi-label multi-instance

learning for image classification", In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2008.