

**Image content based classification of  
Vacation/ tourism related images**

Software Requirements Specification

Project ID: 16-074

Author : Gunathilaka M.A.S.S  
IT13018238

Object Identification and Classification

**B.Sc. Special (Honors) Degree in Information Technology**

Date of Submission

01-04-2016

## **Declaration**

I hereby declare that the Software Requirements Specification entitled “Image content based classification of vacation/ tourism related images”, submitted to the Sri Lanka Institute of Information Technology is a record of an original work done by me, under the guidance of our supervisor Dr.Rohana Thilakumara. This project work is submitted in the partial fulfillment of the requirement for the award of the degree of Bachelor of Science in Information Technology. The results embodied in this report have not been submitted to any other university or Institution for the award of any degree or diploma. Information derived from the published or unpublished work of others has been acknowledged in the text and a complete list of references is given.

Name: Gunathilaka M.A.S.S

Signature: .....

## Table of Contents

1.	Introduction.....	5
1.1.	Purpose .....	5
1.2.	Scope .....	5
1.3.	Definitions, Acronyms, and Abbreviations.....	6
1.4.	Overview .....	7
2.	Overall Descriptions .....	8
2.1.	Product perspective .....	8
2.1.1.	System interfaces .....	9
2.1.2.	User interfaces .....	9
2.1.3.	Hardware interfaces .....	10
2.1.4.	Software interfaces.....	10
2.1.5.	Communication interfaces .....	11
2.1.6.	Memory constraints .....	11
2.1.7.	Operations .....	11
2.1.8.	Site adaptation requirements.....	11
2.2.	Product functions.....	12
2.3.	User characteristics .....	15
2.4.	Constraints.....	15
2.5.	Assumptions and dependencies.....	15
2.6.	Apportioning of requirements .....	15
3	Specific requirements.....	16
3.1.	External interface requirements .....	16
3.1.1.	User interfaces .....	16
3.1.2.	Hardware interfaces .....	16
3.1.3.	Software interfaces.....	16
3.1.4	Communication interfaces .....	17
3.2	Classes/Objects .....	18
	<b>3.3 Performance requirements .....</b>	<b>18</b>
	<b>3.4 Design constraints .....</b>	<b>19</b>
3.5	Software system attributes .....	19
3.5.1	Reliability.....	19
3.5.2	Availability .....	19
3.5.3	Security .....	19

3.5.4 Maintainability .....	20
3.6 Other requirements.....	20
4 Supporting information .....	20
4.1 References .....	20
4.2 Appendices.....	21

## List of tables

Table 2.2.1 Use case scenario1 .....	14
Table 2.2.2 Use case scenario2.....	15
Table 3.1.1.1 User interface details for figure 2.1.2.1 .....	17

## List of figures

Figure 2.1.2.1 Upload image interface.....	9
Figure 2.1.2.2 Categorized images interface.....	10
Figure 2.2.1 High level diagram for the system.....	12
Figure 2.2.2 Use case diagram.....	13
Figure 3.2.1 Class Diagram.....	18
Figure 4.2.1 Pseudo code for object training algorithm.....	21
Figure 4.2.1 Pseudo code for back propagation algorithm.....	22

# 1. Introduction

## 1.1. Purpose

This document will provide the description about the Image content based classification of vacation/ tourism related images and describe the functionality of object identification and classification. This SRS is mainly introduced for the users and the developers who are interested of improving this algorithm for further classification.

## 1.2. Scope

This document covers the requirements for object identification and classification of image content based classification of vacation/ tourism related images which is an algorithm that classify images and predict what kind of category it belongs to. We can use this algorithm as a web service in future. The purpose of this is to guide developers in selecting a design that will be able to accommodate the full-scale of proposed algorithm. This document will cover the functionality of object identification and classification and the scope is limited to explain only the function. It will not explain the implementation phase in detail.

Object identification and classification will identify the objects in an image based on the trained objects. Therefore in this document we describe the aspects of training the objects and identifying objects based on trained images.

### 1.3. Definitions, Acronyms, and Abbreviations

#### Acronyms

SRS	Software Requirement Specification
RNN	Recursive Neural Network
CNN	Convolutional Neural Network

#### Definitions

SRS	A document that completely describes all the functions of a proposed algorithm and the constraints under which it must operate.
Database	Collection of data stored in the system
Stakeholder	Any person who can affect the project in development.

#### Abbreviation

The System	The project “Image content based classification of vacation/ tourism related images” will hereafter be simply referred as “The System”

## 1.4. Overview

Grouping images into meaningful categories using low-level visual features is a challenging and important problem in content-based image retrieval and classification. Image content based classification of vacation/ tourism related images is an algorithm which is used to classify the content of the image and categories them. The proposed system will be initially developed as a desktop application and then we will develop it as a service for the web users. It provides the facility to categories large no of images within few minutes.

This document is divided into three parts. Section 1 we discussed previously.

### **Section2**

In this section we discuss about the overall description of the function and product perspective which compare our system with other related or competing product. Under product perspective we discuss the system interfaces, user interfaces, hardware interfaces, software interfaces, communication interfaces, memory constraints and the operations. Next we discuss about the product function, user characteristics and the constraints.

### **Section 3**

In this section we discuss about the external interface requirements, Classes/Object, performance requirement, design constraint and the software system attributes. Under software system attributes we discuss the non-functional requirements like availability, reliability, security and maintainability.

## 2. Overall Descriptions

Content based image classification is a major requirement for web users in these days. Even though human can understand the content of the image, still there is a problem of identifying content in the image for computers.

Object identification and classification parts is a major role in this proposed system. Before moving to identifying objects in the image, we need to train sample objects in order to identify objects. Then there after the segmented and well-formed regions are input to the neural network and match them with the trained objects. So that the major concern of this component is to train the objects in an image more accurately. From the trained objects we can identify the content of the input image easily.

### 2.1. Product perspective

Object identification and classification can be basically done in two pattern recognition algorithms called appearance-based and feature based. In order to do that lot of researchers have used neural network.

[1] and [2] have used convolutional neural network (CNN) which trains large no of images for object identification purpose. The problem of using that is their network's performance degrades if a single convolutional layer is removed.

We use Recursive Neural Network (RNN) for the purpose of identifying objects in the image. The advantage of using RNN is it provides a better solution to minimize the error rate.



### 2.1.1. System interfaces

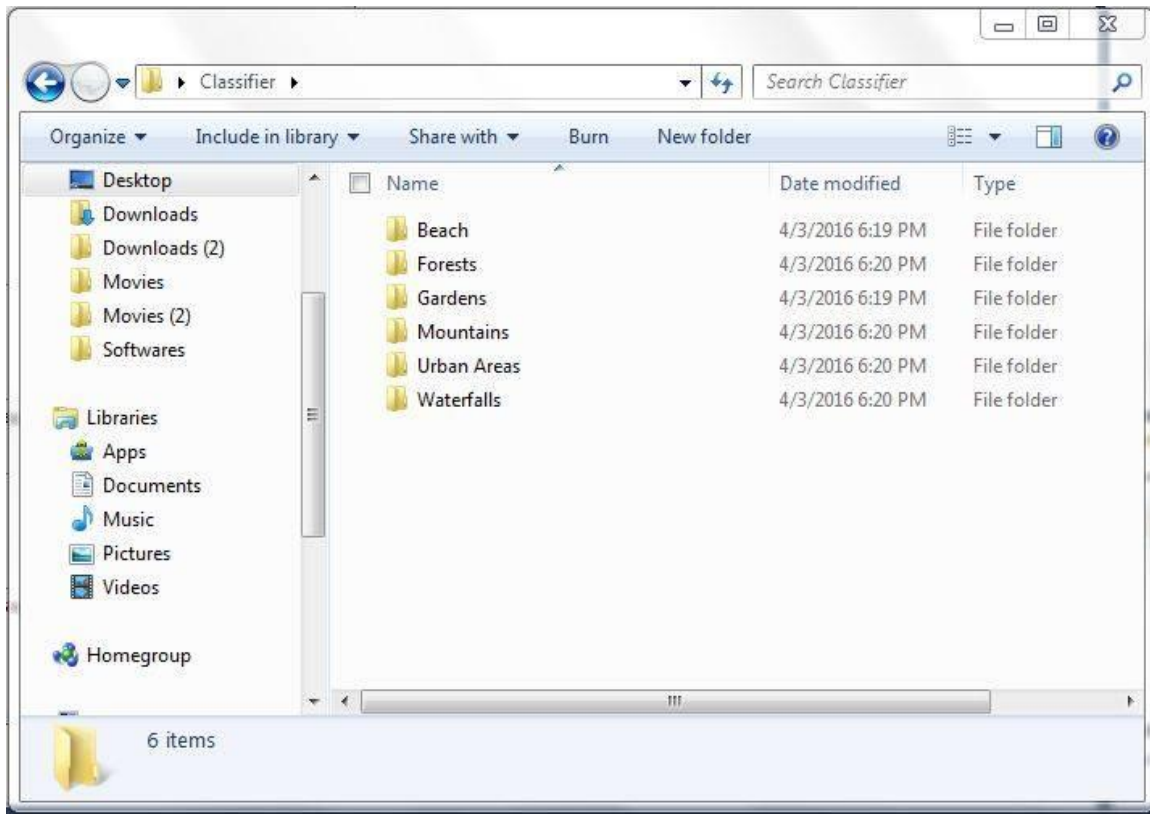
Since this proposed system is based on some algorithms, there is no any external interface involved with. It only needs a proper Internet connection and a computer running on windows 7 or above to operate the system.

### 2.1.2. User interfaces

The screenshot shows a Windows application window titled 'Form1'. It displays classification statistics and a list of image categories with their respective percentages. Below this, there are two rows of six dashed boxes, each labeled 'Image Category' underneath. At the bottom left, there is an 'Upload Images' button.

Image Categories	Percentage
Beach	20%
Forests	20%
Mountains	18%
Urban Areas	20%
Gardens	20%
Waterfalls	1%

*Figure 2.1.2.1 Upload image interface*



*Figure 2.1.2.2 Categorized images interface*

### 2.1.3. Hardware interfaces

Since this project based on an algorithm, there is no any hardware interface involved with the proposed system.

### 2.1.4. Software interfaces

- Python 2.7 -  
Main environment for building the algorithms
- MySQL -  
To storing the sample images in the database
- MATLAB -  
Testing environment to check the algorithms are working properly.

#### 2.1.5. Communication interfaces

All users must have a proper Internet connection. Therefore a modem or a wifi router is required.

#### 2.1.6. Memory constraints

User's computer machine is expected to use at least 4GB of Ram and 1GB of external storage.

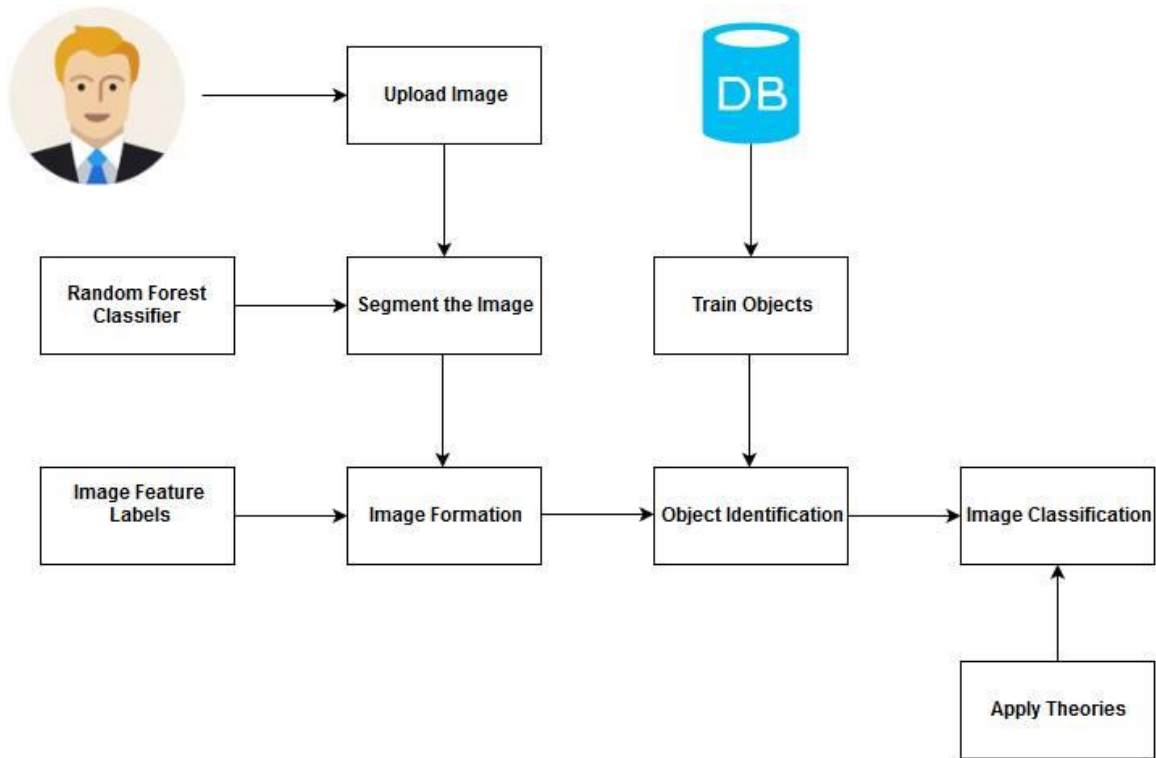
#### 2.1.7. Operations

- User must enable the web access of the computer
- User should upload images for classification

#### 2.1.8. Site adaptation requirements

Since the system will be initially developed to be as a desktop application to be run on a computer with windows 7 or above the system will not be adapted to other operating systems or environments until this research phase is complete. Since the user can be of any nationality the user interface must be created for English language.

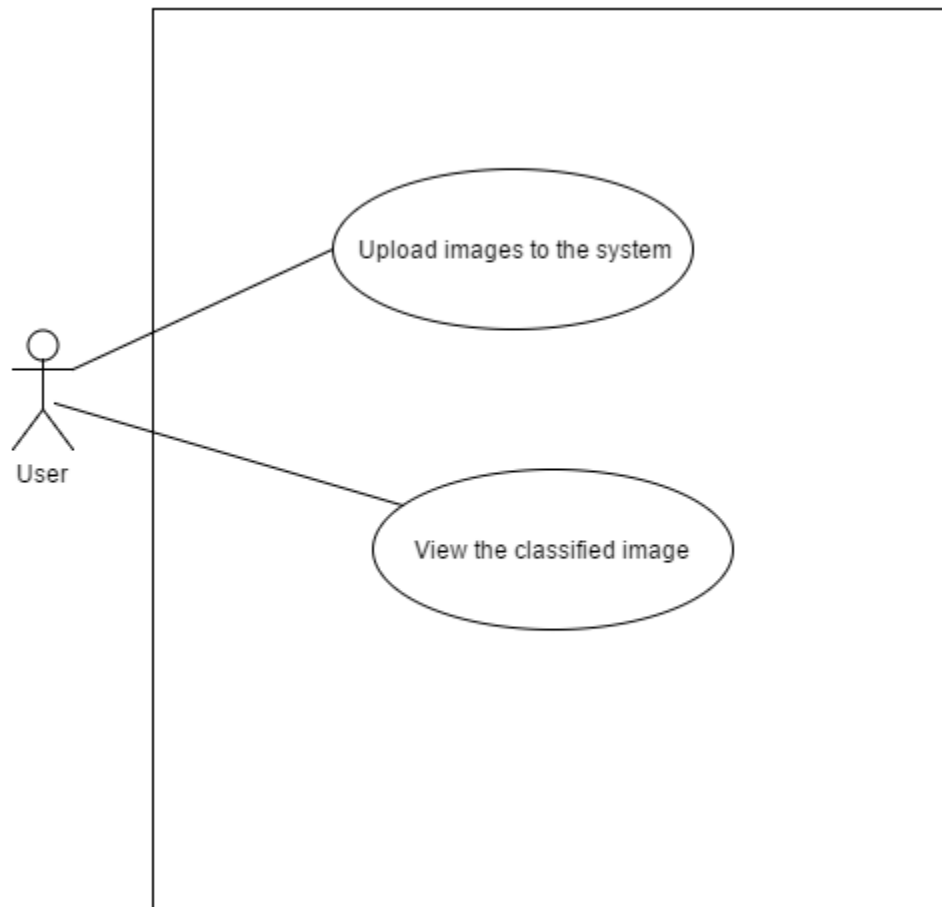
## 2.2. Product functions



*Figure 2.2.1 High level diagram for the system*

The system consists of mainly four functions namely image spatial representation and segmentation, object formation from segmented images parts, object classification and identification, Image context identification and classification. This SRS document mainly covers the requirements for the “Object identification and classification” component of the application.

Object identification and classification can be divided into two parts. In first part, we develop an algorithm to train the algorithm using objects in sample images. In second part, we develop an algorithm for identify objects in the uploaded images.



*Figure 2.2.2 Use case diagram*

### **Use case scenario**

Use case 01	Upload images to the system
Primary Actor(s)	User
Pre-Condition	Images should be related to vacation/tourism related images
Post-Condition	Uploaded image should be display in the interface.

Main Scenario	<ol style="list-style-type: none"> <li>1. Start with opening the application</li> <li>2. User click on the browse image button</li> <li>3. Select image and click OK button.</li> </ol>
Extensions	If user input a different file to the system, system will display an error message

*Table 2.2.1 Use case scenario1*

Use case 02	View the classified image
Primary Actor(s)	User
Pre-Condition	Images should be related to vacation/tourism related images
Post-Condition	
Main Scenario	<ol style="list-style-type: none"> <li>1. Start with opening the application</li> <li>2. User click on the browse image button</li> <li>3. Select image and click OK button.</li> <li>4. User click “Apply Classification” button</li> </ol>
Extensions	If user input a different file to the system, system will display an error message

*Table 2.2.2 Use case scenario2*

### 2.3. User characteristics

1. General computer user.
2. Software professionals.

### 2.4. Constraints

- This algorithm will only consider about the images which belong to traveling/tourism related images.
- This algorithm shall operate on PCs running Windows 7 or later at a minimum speed of 2.2 GHZ.
- User must have a proper Internet connection.

### 2.5. Assumptions and dependencies

- User have an average speed of computer machine and good Internet connection.
- User inputs only the vacation/tourism related images

### 2.6. Apportioning of requirements

#### Essential Requirement

1. Allow user to upload images to be classified
2. Objects of the image should be identified by the system.

#### Desirable Requirements

1. Use this algorithm as a web service for the travelling/tourism related applications for get the user preferences and suggest the places where user wish to visit.

#### Optional Requirements

1. Connect with social profiles, check the contents and suggest the places according to their desires.

### 3 Specific requirements

#### 3.1. External interface requirements

##### 3.1.1. User interfaces

Figure 2.1.2.1 Upload image interface

Name of the item	“Upload images” button
Description of purpose	Pop up a new window to browse images from the computer
Source of input or destination of output	Mouse point
Valid range, accuracy and/or tolerance	N/A
Units of measure	No
Timing	No
Relationships to other inputs/outputs	No
Screen formats/organization	Bottom left
Window formats/organization	Left of screen
Data formats	N/A

Table 3.1.1.1 User interface details for figure 2.1.2.1

##### 3.1.2. Hardware interfaces

- Core i3 processor may be required in order to process the large no of images

##### 3.1.3. Software interfaces

- Python 2.7 -

This is the main environment for building the algorithms



- MySQL -

This is for storing the sample images in the database

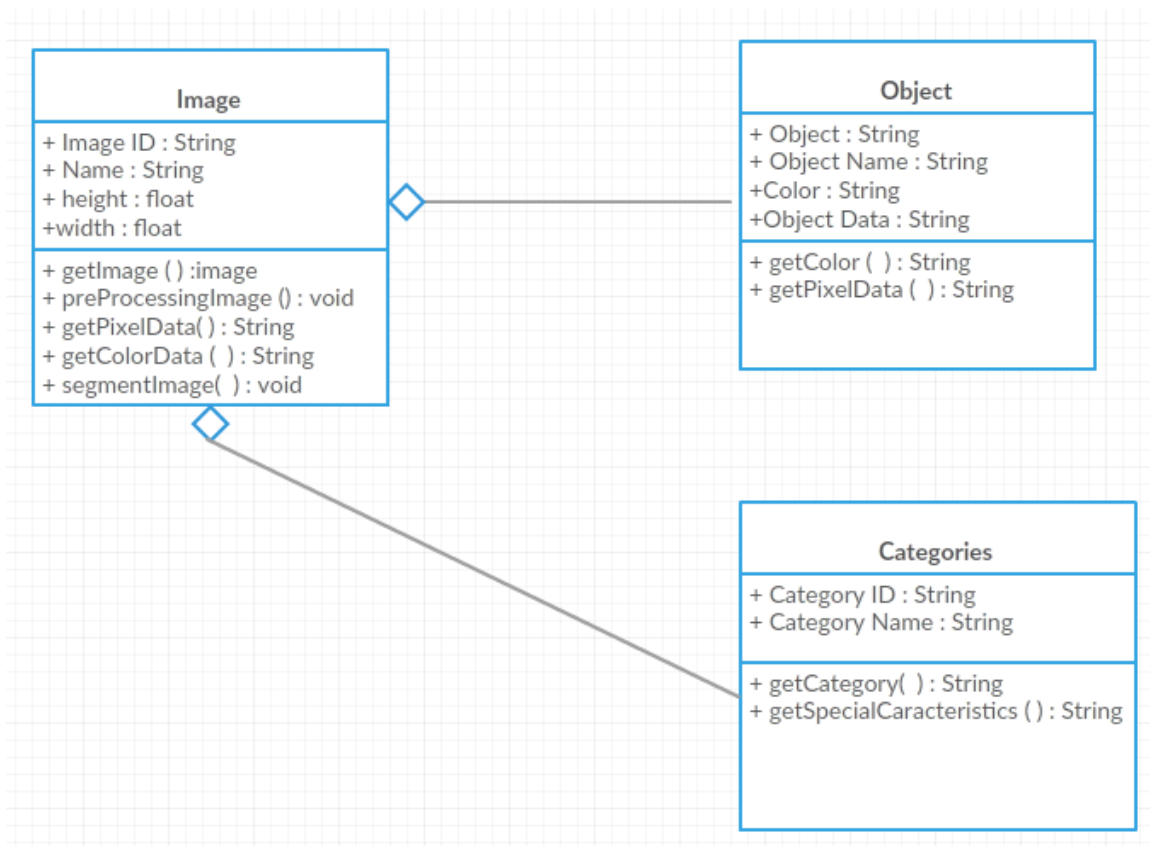
- MATLAB -

This environment is used to check the algorithms are working properly or not.

#### 3.1.4 Communication interfaces

All users must have a proper Internet connection. Therefore a modem or a wifi router is required.

### 3.2 Classes/Objects



3.2.1 Figure Class Diagram

### 3.3 Performance requirements

- Since large no of objects training is main part of this algorithm, the system will require at least 2GB RAM
- The processor speed should be at least 2.2 GHZ.

### **3.4 Design constraints**

Since we are developing an algorithm for image classification, we are not considering about the designing interfaces for now. In future we could use this algorithm for various applications.

### **3.5 Software system attributes**

#### **3.5.1 Reliability**

This algorithm should be able to identify objects in the image more accurately with the limited time. Internet connection should be in consistent stage in order to maintain the process of identifying objects consistency.

#### **3.5.2 Availability**

This system should be available for any web user and they must have a proper Internet connection for maintaining availability.

#### **3.5.3 Security**

The sample images are stored in the database. Therefore security is needed. And when user closed the application, uploaded images must be removed from the system.(Because all they want is to classify the given image)

### 3.5.4 Maintainability

The system should be able to classify the images properly after the maintenance has done and updated categories also need to be classified and give output correctly.

### 3.6 Other requirements

- User friendly -

User can easily upload images to the system without finding it difficult and the predicted results should be displayed in well formed.

- Modifiability -

The proposed system should be able to accept modifications.

## 4 Supporting information

### 4.1 References

[1] Howard, A. G., "Some Improvements on Deep Convolutional Neural Network Based Image Classification", *arXiv preprint arXiv:1312.5402*, 2013.

[2] Krizhevsky, A., Sutskever I., and Hinton, G., "Imagenet classification with deep convolutional neural networks", *Advances in neural information processing systems*, 2012.

## 4.2 Appendices

1. **PROCEDURE TRAIN\_OBJECTS** (object)
2.     **for** each training example in object **do**
3.         Present the example to the network
4.         Extract the features of the example
5.         **for** each child of the output node **do**
6.             **BACK\_PROPAGATE**(child , features)
7.         **end for**
8.         Update the example features
9.     **end for**
10. **End procedure**

Figure 4.2.1 Pseudo code for object training algorithm

1. **PROCEDURE BACK\_PROPAGATE** (node, features)
2.     Accumulate values for the node
3.     Compute the error for features
4.     **for** each child of node **do**
5.         **BACK\_PROPAGATE** (node, features)
6.     **End for**
7. **End procedure**

Figure 4.2.2 Pseudo code for back propagation algorithm