This book is about the Classification of emotions based on natural scene images which is a new concept in an innovative field of Image Processing domain. Image processing domain has always proven to be challenging criteria in field of research and development. This book covers concepts related to Image Retrieval, Image Processing and Computer Vision, Emotion detection and CBIR (Content Based Image Retrieval) technique.

Prasad A. Lahare
Pramod R. Gunjal

He is persuing Master of Engineering in Computer Engineering at AVCOE, Sangamner. He has completed his Bachelors Degree in Computer Engineering from MIT, Pune. He is a member of Association of Computer Machinery(ACM).

Classification of Natural Scene Images Based On Emotions

Trends in Image Processing and Computer Vision

978-3-659-86311-0

LAP LAMBERT Academic Publishing

Prasad A. Lahare
Pramod R. Gunjal

Classification of Natural Scene Images Based On Emotions

Prasad A. Lahare
Pramod R. Gunjal

Classification of Natural Scene Images Based On Emotions

Trends in Image Processing and Computer Vision

LAP LAMBERT Academic Publishing

Impressum / Imprint
Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Bibliographic information published by the Deutsche Nationalbibliothek: The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Coverbild / Cover image: www.ingimage.com

Verlag / Publisher:
LAP LAMBERT Academic Publishing
ist ein Imprint der / is a trademark of
OmniScriptum GmbH & Co. KG
Bahnhofstraße 28, 66111 Saarbrücken, Deutschland / Germany
Email: info@lap-publishing.com

Herstellung: siehe letzte Seite /
Printed at: see last page
ISBN: 978-3-659-86311-0

Authors Bibliography:

Mr. Mr. Prasad A. Lahare, persuing masters in Computer Engineering at AVCOE, Sangamner. He has received his Bachelors in Computer Engineering from MIT, Pune. He is a member of Association of Computer Machinery (ACM).

Prof. Pramod R. Gunjal, Assistant Professor in Electronics and Telecommunication at Amrutvahini College of Engineering, Sangamner. He has received his Master's in Electronics in year 2015. He is a member of IEEE, IETE. His interest is in Image Processing, VLSI and Wireless communication.

ACKNOWLEDGEMENT

I thank god almighty for guiding me through this work. I would like to thank all those who have contributed to the completion of this work done and helped me with valuable suggestion for improvement.

I am extremely grateful to Prof. P. R. Gunjal for his guidance and support. I will forever remain grateful for the constant support and guidance extended by him, in making this report. Through our many discussions, he helped me to form and solidify ideas. The invaluable discussions I had with him, the penetrating questions he has put to me and the constant motivation, has all lead to the development of this work.

I wish to express my sincere thanks to my parents for supporting and encouraging me. I would also like to thank to my friends for listening to my ideas, asking questions and providing feedback and suggestions for improving my ideas.

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Emotion in natural scene images plays an important role in the way humans perceive an image. Based on the emotion (happiness, sadness, fear, anger etc.) of any human being the images that are viewed by that person can have a significant impact in a sense that if the person is for example in happy mood and he/she views an image that is pleasing then he/she would have a better sense of attachment towards that image and would not accept an image that depicts sadness as an emotion. Although different people may interpret the same image in different ways, we still can build a universal classification for different emotions. Emotion detection in natural scene images basically means that the natural scene image should be classified properly based on image semantics. Any image can be classified into three levels of semantics (low level, medium level, high level). Our task is to bridge the gap between the different levels. This work deals with the color component of an image.

The tasks to be performed are: Build Classifiers for every mood, retrieve low level semantic information of the chosen image, accordingly classify the image mapping its retrieved information. Although different people may interpret the same image in different ways, we still can build a universal classification for different emotions. However, it is a challenging task for any machine to recognize emotion in any natural scene image we still can build classifiers which can help the machine to adequately classify images according to different emotions.

INDEX

CHAPTER 1
INTRODUCTION

1.1 Overview

Emotion in natural scene images plays an important role in the way humans perceive an image. Based on the emotion (happiness, sadness, fear, anger etc.) of any human being the images that are viewed by that person can have a significant impact in a sense that if the person is for example in happy mood and he/she views an image that is pleasing then he/she would have a better sense of attachment towards that image and would not accept an image that depicts sadness as an emotion. Although different people may interpret the same image in different ways, we still can build a universal classification for different emotions [5].

Emotion detection in natural scene images basically means that the natural scene image should be classified properly based on image semantics. Any image can be classified into three levels of semantics (low level, medium level, high level). Our task is to bridge the gap between the different levels. This work deals with the color component of an image. The tasks to be performed are: Build Classifiers for every mood, retrieve low-level semantic information of the chosen image and accordingly classify the image mapping its retrieved information.

Figure 1.1: Images depicting different emotions.

Although different people may interpret the same image in different ways, we still can build a universal classification for different emotions. However, it is a challenging task for any machine to recognize emotion in any natural scene image we still can build classifiers which can help the machine to adequately classify images according to different emotions.

1.2 Classification of Emotion in Natural Scene Image

1.2.1 Problem statement

To develop system software that can detect an emotion from a natural scenic image, a static image without human intervention or manmade objects.

Area:

- Image Processing.
- Artificial Intelligence.
- Machine Learning

This work mainly focuses on determining the various parameters such as build classifiers for every mood retrieve low-level semantic information of the chosen image and accordingly classify the image mapping its retrieved information.

The modules involved in the Classification of Emotion in Natural Scene Images are as follows:

Module I: Build classifiers for every mood.

This module of work comes under pattern recognition in image processing where classifiers would be built for desired emotions, considering each emotion having some specific pattern and semantics [5].

Module II: Retrieve low-level semantic information of the chosen image.

An image as a whole contains many low-level semantic information like colour, texture, shape which are to be retrieved to gain a pattern.

Module III: Classify the image by mapping its retrieved information.

This module would give final output of the work by classifying the randomly selected natural scene image to one of the desired emotion, for example: happy or sad.

1.2.2 Need of system development

Classification of natural scenic images based on emotion is a novice implementation in the field of image processing. Though depiction of emotion changes from person to person and makes this idea more subjective still it can build a universal classification for different emotions.

This system would play a role of laying a platform on which different applications may be designed bringing out brilliant innovations from the world of artificial intelligence and image processing [3].

Emotion detection in natural scene images basically needed so that the natural scene image should be classified properly based on image semantics. Any image can be classified into three levels of semantics (low level, medium level, high level). Our task is to bridge the gap between the different levels. This work deals with the color component of an image. The tasks to be performed are: Build Classifiers for every mood, retrieve low level semantic information of the chosen image, accordingly classify the image mapping its retrieved information. Although different people may interpret the same image in different ways, we still can build a universal classification for different emotions. However, it is a challenging task for any machine to recognize emotion in any natural scene image we still can build classifiers which can help the machine to adequately classify images according to different emotions.

1.2.3 Application Area and Scope of Research work

- Setting Wallpapers According to Mood:

We can create a system that accepts the mood of the user and automatically sets wallpaper for his/her mood.

This idea can be further implemented as using the mood as an input as setting the system theme accordingly.

1.2.4 Other Future Applications that can be developed

- E-Greeting Cards:

We can use the natural scene images in generating greeting cards. According to the occasion for which we are sending the card we can directly generate the card by just entering the mood for which it being generated.

CHAPTER 2
LITERATURE SURVEY

2.1 Product Perspective

The retrieval accuracy of content-based image retrieval systems, research focus has been shifted from designing sophisticated low-level feature extraction algorithms to reducing the 'semantic gap' between the visual features and the richness of human semantics. Attempts to provide a comprehensive survey of the recent technical achievements in high-level semantic-based image retrieval has been done. Major recent publications are included in this survey covering different aspects of the research in this area, including low-level image feature extraction, pattern recognition, similarity measurement, and deriving high-level semantic features [2].

2.2 Product Features

The Intended product would typically identify the emotion from a natural scene image contained in a database. According to the input of the user for the emotion, these images would be scanned and a suitable image would be presented in return. These images would contain only natural scenes excluding human intervention. Emotion detection in natural scene images basically means that the natural scene image should be classified properly based on image semantics. Any image can be classified into three levels of semantics (low level, medium level, high level). Our task is to bridge the gap between the different levels. This work deals with the color component of an image [3].

2.3 Literature Survey

Research Relevant International Papers Surveyed

- Fuzzy GIST Emotion Detection from Natural Scene Image

Emotion modeling evoked by natural scenes is challenging issue. In this paper, we propose a novel scheme for analyzing the emotion reflected by a natural scene, considering the human emotional status. Based on the concept of original GIST, we developed the fuzzy-GIST to build the emotional feature space. According to the relationship between emotional factors and the characters of image, L*C*H* color and orientation information are chosen to study the relationship between human's low level emotions and image characteristics. And it is realized that we need to analyze the visual features at semantic level, so we incorporate the fuzzy concept to extract features with semantic meanings. Moreover, we treat emotional electroencephalography (EEG) using the fuzzy logic based on possibility theory rather than widely used conventional probability theory to generate the semantic feature of the human emotions. Fuzzy-GIST consists of both semantic visual information and linguistic EEG feature; it is used to represent emotional gist of a natural scene in a semantic level. The emotion evoked by an image is predicted from fuzzy-GIST by using a support vector machine, and the mean opinion score (MOS) is used for performance evaluation for the proposed scheme. The experiments results show that positive and negative emotions can be recognized with high accuracy for a given dataset.

- Semantic Categorization and Retrieval of Natural Scene Images

The semantic gap between the digital image representation and the user's image understanding is still a big problem. In our work we try to reduce this semantic gap in a field of natural images. This paper proposes a method for semantic categorization and retrieval of natural scene images with and with-out people. These are typical holiday pictures from hiking outdoors. Our approach comprises of three stages. Pre-processing consists of image segmentation into arbitrary-shaped regions and detection of people in the image. In the next stage, local image regions are classified using low level features into semantic concept classes such as water, sky or sand. Finally, the frequency of occurrence of these semantic concept classes determines the high level scene category. [4] For the classification of local image regions, the k-Nearest Neighbor and Support Vector Machine classifiers are used. The results obtained by both classifiers are validated within the paper. We live in a world where having a digital camera or image scanner is not a problem anymore. People are used to take thousands of pictures during their vacation and they like to share them at the web galleries or social networks. Due to more and more images being generated in digital form around the world, it is important to deal with a problem how to extract the semantic content of images and then retrieve these images effectively. Humans tend to interpret images using high-level concepts, they are able to identify keywords, abstract objects or events presented in the image. However, for a computer the image content is a matrix of pixels, which can be summarized by low-level color, texture or shape features. The lack of correlation between the high-level concepts that a user requires and the low-level features that image retrieval systems offer is the semantic gap. In our work we try to reduce this semantic gap in a field of natural scene images with and without people. These sort of pictures are common in personal family albums. Our method can help the people to search in these albums effectively [8].

- Mining Association Rules between Low-Level Image Features and High Level Concepts

In image similarity retrieval systems, color is one of the most widely used features. Users who are not well versed with the image domain characteristics might be more comfortable in working with an Image Retrieval System that allows specification of a query in terms of keywords, thus eliminating the usual intimidation in dealing with very primitive features. In this paper we present two approaches to automatic image annotation, by finding those rules underlying the links between the low-level features and the high-level concepts associated with images. One scheme uses global color image information and classification tree based techniques. Through this supervised learning approach, we are able to identify relationships between global color-based image features and some textual descriptors. In the second approach, using low-level image features that capture local color information and through a k-means based clustering mechanism, images are organized in clusters such that images that are similar are located in the same cluster. For each cluster, a set of rules is derived to capture the association between the localized color-based image features and the textual descriptors relevant to the cluster [7].

- Content Based Image Retrieval and High Level Semantics

Semantic gap between the visual features and human semantics has become a bottleneck of content-based image retrieval. The need for improving the retrieval accuracy of image retrieval systems and narrowing down the semantic gap is high in view of the fast growing need of image retrieval. In this paper, we first introduce the image semantic description methods, and then we discuss the main technologies for reducing the semantic gap, namely, object-ontology, machine learning, and relevance feedback. Applications of above-mentioned technologies in various areas are also introduced. Finally, some future research directions and problems of image retrieval are presented.

- Image Retrieval- Trends of the New Age

The last decade has witnessed great interest in research on content-based image retrieval. This has paved the way for a large number of new techniques and systems, and a growing interest in associated fields to support such systems. Likewise, digital imagery has expanded its horizon in many directions, resulting in an explosion in the volume of image data required to be organized. In this paper, we discuss some of the key contributions in the current decade related to image retrieval and automated image annotation, spanning 120 references. We also discuss some of the key challenges involved in the adaptation of existing image retrieval techniques to build useful systems that can handle real-world data. We conclude with a study on the trends in volume and impact of publications in the field with respect to venues/journals and sub-topics [6].

- Estimating information from image colors: an application to digital cameras and natural scenes

The colors present in an image of a scene provide information about its constituent elements. But the amount of information depends on the imaging conditions and on how information is calculated. This work had two aims. The first was to derive explicitly estimators of the information available and the information retrieved from the color values at each point in images of a scene under different illuminations. The second was to apply these estimators to simulations of images obtained with five sets of sensors used in digital cameras and with the cone photoreceptors of the human eye. Estimates were obtained for 50 hyper spectral images of natural scenes under daylight illuminants with correlated color temperatures 4000 K, 6500 K, and 25000 K. Depending on the sensor set, the mean estimated information available across images with the largest illumination difference varied from 15.5 to 18.0 bits and the mean estimated information retrieved after optimal linear processing varied from 13.2 to 15.5 bits (each about 85% of the corresponding information available). With the best sensor set, 390% more points could be identified per scene than with the worst. Capturing scene information from image colors depends crucially on the choice of camera sensors.

- Mapping low-level image features to semantic concepts

In this study, a novel offline supervised learning method is proposed to map low-level visual features to high-level semantic concepts for region-based image retrieval. The contributions of this study lie in three folds [10].

(1) For each semantic concept, a set of low-level tokens are extracted from the segmented regions of training images. Those tokens capture the representative information for describing the semantic meaning of that concept;
(2) a set of posteriors are generated based on the low-level tokens through pairwise classification, which denote the probabilities of images belonging to the semantic concepts. The posteriors are treated as high-level features that connect images with high-level semantic concepts. Long-term relevance feedback learning is incorporated to provide the supervisory information needed in the above offline learning process, including the concept information and the relevant training set for each concept: an integrated algorithm is implemented to combine two kinds of information for retrieval: the information from the offline feature-to-concept mapping process and the high-level semantic information from the long-term learned memory. Experimental evaluation on 10,000 images proves the effectiveness of our method.

- Support Vector machines for classification and regression

This document has been written in an attempt to make the Support Vector Machines (SVM), initially conceived of by Cortes and Vapnik, as simple to understand as possible for those with minimal experience of Machine Learning. It assumes basic mathematical knowledge in areas such as calculus, vector geometry and Lagrange multipliers. The document has been split into Theory and Application sections so that it is obvious, after the math’s has been dealt with, how to actually apply the SVM for the different forms of problem that each section is centered on. The document's rest section details the problem of classification for linearly separable data and introduces the concept of margin and the essence of SVM margin maximization. The methodology of the SVM is then extended to data which is not fully linearly separable. This soft margin SVM introduces the idea of slack variables and the trade-o_ between maximizing the margin and minimizing the number of misclassified variables in the second section. The third section develops the concept of SVM further so that the technique can be used for regression. The fourth section explains the other salient feature of SVM – the Kernel Trick. It explains how incorporation of this mathematical sleight of hand allows SVM to classify and regress nonlinear data [12].

2.4 Computational Framework for mood classification

2.4.1 Machine learning

Machine learning, a branch of artificial intelligence, is about the construction and study of systems that can learn from data**.** The core of machine learning deals with representation and generalization. Representation of data instances and functions evaluated on these instances are part of all machine learning systems. Generalization is the property that the system will perform well on unseen data instances; the conditions under which this can be guaranteed are a key object of study in the subfield of computational learning theory.

Machine learning allows automatic creation of classifiers, however, the classical models are generally slow to train, and not interactive. The classical machine-learning (CML) model is summarized in Figure 1. Prior to the training of the classifier, features need to be selected. Training is then performed “off-line” so that classification can be done quickly and efficiently. In this model classification is optimized at the expense of longer training time. Generally, the classifier will run quickly so it can be done real-time. The assumption is that training will be performed only once and need not be interactive. Many machine-learning algorithms are very sensitive to feature selection and suffer greatly if there are very many features.

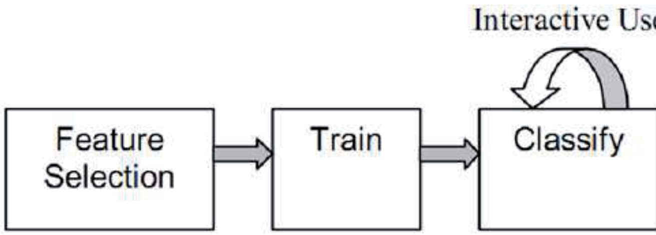


Figure 2.1: Classical machine learning model.

The current approach requires too much technical knowledge on the part of the interface designer. What we would like to do is replace the classical

Machine-learning model with the interactive model shown in Figure 2.1. This interactive training allows the classifier to be coached along until the desired results are met. In this model the designer is correcting and teaching the classifier and the classifier must perform the appropriate feature selection [5].

The pre-selection of features can be eliminated and transferred to the learning part of the IML if the learning algorithm used performs feature selection. This means that a large repository of features is initially calculated and fed to the learning algorithm so it can learn the best features for the classification problem at hand.

The idea is to feed a very large number of features into the classifier training and let the classifier do the filtering rather than the human. The human designer then is focused on rapidly creating training data that will correct the errors of the classifier.

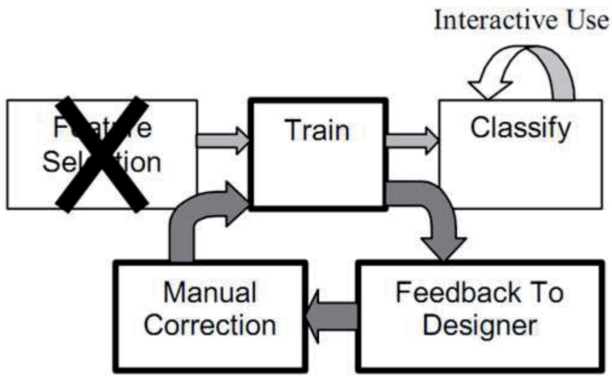


Figure 2.2: Interactive machine learning (IML) model.

Machine learning algorithms can be divided into two types:

1. Supervised learning
2. Unsupervised learning

1. Supervised learning

Supervised learning is the machine learning task of inferring a function from labeled training data. [1] The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyses the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly

In order to solve a given problem of supervised learning, one has to perform the following steps:

1. Determine the type of training examples. Before doing anything else, the user should decide what kind of data is to be used as a training set. In the case of handwriting analysis, for example, this might be a single handwritten character, an entire handwritten word, or an entire line of handwriting.

2. Gather a training set. The training set needs to be representative of the real-world use of the function. Thus, a set of input objects is gathered and corresponding outputs are also gathered, either from human experts or from measurements.
3. Determine the input feature representation of the learned function. The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is transformed into a feature vector, which contains a number of features that are descriptive of the object. The number of features should not be too large, because of the curse of dimensionality; but should contain enough information to accurately predict the output.
4. Determine the structure of the learned function and corresponding learning algorithm. For example, the engineer may choose to use support vector machines or decision trees.
5. Complete the design. Run the learning algorithm on the gathered training set. Some supervised learning algorithms require the user to determine
certain control parameters. These parameters may be adjusted by optimizing performance on a subset (called a validation set) of the training set, or via cross-validation.
6. Evaluate the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set.

- Approaches to supervised learning include:
    - Analytical learning
    - Artificial neural network
    - Bayesian statistics
    - Decision tree learning
    - Gaussian process regression
    - Minimum message length (decision trees, decision graphs, etc.)
    - Nearest Neighbor Algorithm
    - Symbolic machine learning algorithms
    - Support vector machines
    - Random Forests

2. Unsupervised learning

In machine learning, unsupervised learning refers to the problem of trying to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning. Unsupervised learning is closely related to the problem of density estimation in statistics. However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data. Many methods employed in unsupervised learning are based on data mining methods used to pre-process data [11].

- Approaches to unsupervised learning include:
  - Clustering (e.g., k-means, mixture models, hierarchical clustering)
  - Blind signal separation using feature extraction techniques for dimensionality reduction (e.g., Principal component analysis, Independent component analysis, Non-negative matrix factorization, Singular value decomposition).

CHAPTER 3

PLAN OF THE WORK

3.1 Research Scope

The scope of this work is classification of an emotion in scenic images. (Natural and human faces). A natural scene image can be described as an image which has no human intervention or any manmade objects to be detected or otherwise be classified based on emotions. An image would be classified as happy or sad as follows: By extracting the features of that image and then applying it to a suitable classification algorithm for detection of an emotion.

Emotion detection in natural scene images basically means that the natural scene image should be classified properly based on image semantics. Any image can be classified into three levels of semantics (low level, medium level, high level). Our task is to bridge the gap between the different levels. This work deals with the color component of an image. The tasks to be performed are: Build Classifiers for every mood, retrieve low level semantic information of the chosen image, accordingly classify the image mapping its retrieved information. Although different people may interpret the same image in different ways, we still can build a universal classification for different emotions. However, it is a challenging task for any machine to recognize emotion in any natural scene image we still can build classifiers which can help the machine to adequately classify images according to different emotions.

Application area of work:

- Setting Wallpapers According to Mood:

We can create a system that accepts the mood of the user and automatically sets wallpaper for his/her mood.

This idea can be further implemented as using the mood as an input as setting the system theme accordingly.

Other Future Applications that can be developed:

- E-Greeting Cards:

We can use the natural scene images in generating greeting cards. According to the occasion for which we are sending the card we can directly generate the card by just entering the mood for which it being generated.

3.2 Feasibility study

3.2.1 Operational

Time:

The time required will depend on the size of the input work which may have image retrieval, training dataset and also on the review constraints provided by the end user.

Space:

The space requirements will depend on the size of the source file which in turn will depend on the size of input work.

Efficiency:

Efficiency of the system will depend on review constraint provided by the reviewer i.e. End user whether it is correct or not and if not then it can be achieved by classifier and training set.

3.2.2 Financial

The software used in developing the work is free ware i.e. Java and an open source Net Beans-IDE by SUN Microsystems. Therefore, the work was found to be cost feasible.

3.3 Resources

3.3.1 Human resources

This work is performed by a group of four members and all its work was well divided between its team members.

3.3.2 Reusable software components

The components developed for the work can be distributed for use in other work. The components used in this research are as follows:

SVM module:

SVM i.e. Support Vector machine is an algorithm used as a classifier hence called as a classification algorithm which comes under machine learning. Its segregates an image into different classes which further helps in classification of that image into a specific emotion. Hence SVM is a module of our work which can be re-used for classification purpose in the field of image processing.

3.3.3 Software tools

Integrated Development Environment (IDE) is useful tool for developing software programs. So, Net beans IDE 7.1.2 and Java are being used in this work.

3.3.4 Requirements

3.3.4.1 Minimum hardware requirements

1] Pentium processor 166 MHz

2] 64 MB RAM.

3] 750 MB of free disc space for program development.

4] 100 MB for program execution.

3.3.4.2 Software requirements

1] JDK 1.7.

2] JRE 1.7.

3] Net Beans 7.2 IDE

3.4 Time Line Schedule specifications:

Table 3.1: Time line specifications.

Activity	**Number of Days**
Literature Survey	June - August
Requirement engineering	September- November
Design	December - January
Implementation: Training Phase Recognition Phase Feature Extraction Building Classifier Pattern Models	January onwards
Documentation	March- April

3.5 Risk management

3.5.1 Technology Awareness:

In today’s world, everyday a new idea is being implemented. Under such circumstances the developer may develop some new algorithms for generating Emotion form Natural Scene Images which may have some new features. So the algorithm may become obsolete. Hence there is need for technology awareness so that we can include the new features.

3.5.2 Risk Factors

A risk management strategy can be included in work plan or as an independent Risk mitigation, monitoring and management plan. RMMM plan depicts the work done as a part of risk analysis by the software work team. Risk mitigation and monitoring commences once the RMMM plan is ready and the risks are documented. The objective of risk management is to assist a work team in defining risks, assessing their impact and probability, and tracking risks throughout a software work.

a. For efficient and accurate image detection, the classifier must be trained for maximum number of different types of images
b. Integration and communication of different components must work properly.

3.5.3 RMMM Plan

After the risks had been categorized, prioritized and their probability of occurrences determined, action was taken to control these risks, which involved mitigating and monitoring these risks. The steps taken were:

1) Strict Completion Deadline

Risk Mitigation:

a. Schedule the work in such a fashion that system design is completed first so that they get enough time for testing and debugging.
b. Follow an iterative model of software development.

Risk Monitoring and Management:

a. Keep track of schedule slips.
b. Put in extra hours to make up for the lost hours.

2) Lack of training on technology and tools

Risk Mitigation:

a. Given enough time to learning all technologies required before actual coding starts.
b. Develop sample applications using these technologies.

Risk Monitoring and Management:

a. Consult technical guide in case of any problems.

3) Inexperience in programming software environment

Risk Mitigation:

a. Learn the required programming thoroughly before starting with the coding.
b. Code simple applications using the new programming languages.

Risk Monitoring and Management:

a. Consult the technical guide in case of any difficulties.

4) Selection of Images

Risk Mitigation

a. Select different types of natural images.
b. Try to remove every type of impurities available in image.

Risk Monitoring and Management

a. Keep a track of different types of images.

5) Feature Extraction

Risk Mitigation:

a. Implement the different types of algorithm.

Risk Monitoring and Management:

a. Test the output which is generated by the algorithm.
b. Modify algorithm when needed for efficiency.

6) Image Recognition

Risk Mitigation:

a. Train the network for maximum possible patterns.

Risk Monitoring and Management:

a. Increase the no. of hidden layers if needed.
b. Training activity must be continued for higher accuracy and efficiency.

3.6 Functional Specifications

3.6.1 Internal Interfaces

The interface design describes how the software communicates within itself, with systems that interoperate with it, and with humans who use it. An interface implies a flow of information (e.g., data and/or control) and a specific type of behavior.

1] Training Dataset

2] Testing Dataset

3] User Interface

4] Database

3.6.2 External interfaces

3.6.2.1 Hardware Requirements

1] Pentium processor 166 MHz

2] 64 MB RAM.

3] 750 MB of free disc space for program development.

4] 100 MB for program execution

3.6.2.2 Software requirements

1] JDK 1.7.

2] JRE 1.7.

3] NetBeans7.2 IDE.

CHAPTER 4
SOFTWARE REQUIREMENT AND ANALYSIS

4.1 Introduction

4.1.1 Purpose

Emotion in natural scene images plays an important role in the way humans perceive an image. Based on the emotion (happiness, sadness, fear, anger etc.) of any human being the images that are viewed by that person can have a significant impact in a sense that if the person is for example in happy mood and he/she views an image that is pleasing then he/she would have a better sense of attachment towards that image and would not accept an image that depicts sadness as an emotion. Although different people may interpret the same image in different ways, we still can build a universal classification for different emotions.

4.1.2 Research Scope

Emotion detection in natural scene images basically means that the natural scene image should be classified properly based on image semantics. Any image can be classified into three levels of semantics (low level, medium level, high level). Our task is to bridge the gap between the different levels. This work deals with the color component of an image. The tasks to be performed are: Build Classifiers for every mood, retrieve low level semantic information of the chosen image, accordingly classify the image mapping its retrieved information. Although different people may interpret the same image in different ways, we still can build a universal classification for different emotions. However, it is a challenging task for any machine to recognize emotion in any natural scene image we still can build classifiers which can help the machine to adequately classify images according to different emotions.

Application Area of the work:

- Setting Wallpapers According to Mood:

We can create a system that accepts the mood of the user and automatically sets wallpaper for his/her mood. This idea can be further implemented as using the mood as an input as setting the system theme accordingly.

Other Future Applications that can be developed

- E-Greeting Cards:

We can use the natural scene images in generating greeting cards. According to the occasion for which we are sending the card we can directly generate the card by just entering the mood for which it being generated.

4.2 Detail description

4.2.1 User Characteristics

- Emotion Detection from Natural Scene Image.
- Semantic Categorization of images.
- Mining Association Rules between Low-Level Image Features and High Level Concepts.
- Content Based Image Retrieval and High Level Semantics.
- Image Retrieval- Trends of the New Age.
- Estimating information from image colors: an application to digital cameras and natural scenes.
- Mapping low-level image features to semantic concepts.

4.2.2 Operating Environment

This software is intended to work on Windows Operating System with standard software and hardware requirement.

4.3 System Features

4.3.1 Functional Requirements:

- From a database of images sequentially retrieve images and based on selected semantic perform feature extraction.

- Apply SVM algorithm on the training dataset for classification of images.

4.3.2 Non-Function Requirements

- Allowing the user to insert new images in the database of images.
- Providing enhanced GUI improving the applications usability.
- Providing Extendibility for integration of facial recognition with natural scene images increasing the human computer interaction.

4.4 Graphical User Interfaces

The GUI implemented in this work depicts the following functionality:

- Selection of natural scene image dataset.
- Display feature extraction of dataset.
- Display the testing accuracy of that dataset
- Select any input image.
- Display the image category.
- Display the emotion of image (Happy, Sad, Angry etc.).
- Display the response time.
- Saving of the intermediate results preformed over the image.

CHAPTER 5
SYSTEM DESIGN

5.1 Mathematical Model Specifications

The different functionalities and use cases identified in this work are:

1. Take Mood as an Input.
2. Identify Emotion.
3. Set as Desktop Background.

5.1.1 Mathematical Model for Building Classifiers:

Input: Test Images Dataset.

Processing:

1. Take input images from database (of the mood for which classifier is to be built).
2. Apply learning algorithm to generate the patterns matching the input image emotion entered to the system.

Output: Pattern of the entered mood.

Functions: RetrieveFromDB(), RecogPattern(), StorePattern()

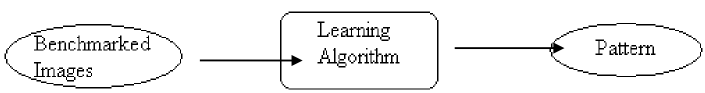


Figure 5.1: Mathematical model for building classifiers.

5.1.2 Mathematical Model for Recognizing Mood of any input Image:

Input: Unclassified Natural Scenic Images.

Processing:

1. Take input Image from a Database of images.
2. Apply classifier to identify the emotion depicted in the image.
3. Store the image according to the emotion class.

Output: Images Sorted according to Emotion.

Functions: RetrieveFromDB(), classifyImage(), StoreEmotion().

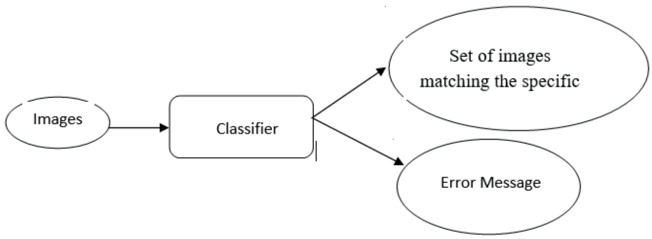


Figure 5.2: Mathematical model for recognizing mood of any input image.

5.1.3 Mathematical Model for System:

Table 5.1: Mathematical model for system.

Sr. No	Description	Observations/ Remarks
1	Let S be the System S={S1, S2} Where, S1- module that Builds classifiers. S2- the module that Recognizing Emotion of any input Image.	S identifies system set
2	S1={D, R, S} Where, D- Retrieve images from benchmarked dataset. R- Recognize Pattern. S- Stores the Pattern Recognized.	S1 is the module that builds classifier.
3	S2={U, C, O} Where, U- User database of Images. C- Classifies the images according to patterns. O- Output (Classified Emotion of the Image).	S2 is the module that recognizes emotion of any image.

5.2 System Architecture

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structure of the system which comprises system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them, and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system.

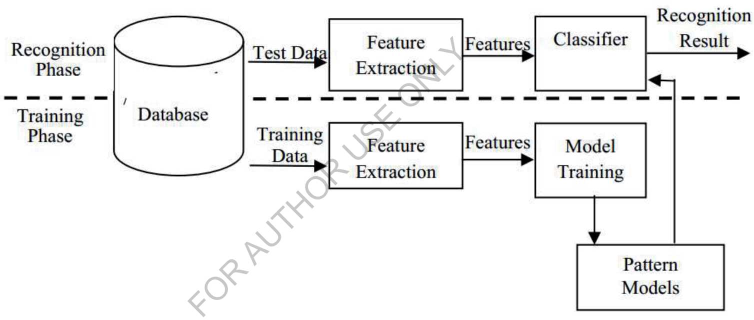


Figure 5.3: System Architecture.

The work is divided into two phases:

1. Training phase
2. Recognition Phase
    - To start with the work, training phase first comes into the picture. In training phase, the image is taken from the training dataset and then passed to the feature extraction block.
    - Feature extraction is responsible for CBIR i.e. content based image retrieval. The low level features are extracted from the specific image.
    - In model training block an image would undergo some pattern recognition to develop different pattern models.

- These pattern models would be taken as a defining model to bring out the emotion from a new image in further stages.
- A pattern model would help the classifier to recognize an emotion from a new image which is not a bench marked image i.e. from the training dataset.

5.3 SVM Classification

The classification problem can be restricted to consideration of the two-class problem without loss of generality. In this problem the goal is to separate the two classes by a function which is induced from available examples. The goal is to produce a classifier that will work well on unseen examples, i.e. it generalizes w ell. Consider the example in Figure. Here there are many possible linear classifier that can separate the data, but there is only one that maximizes the margin maximizes the distance between it and the nearest data point of each class. This linear classifier is termed the optimal separating hyper plane. Intuitively, we would expect this boundary to generalize well as opposed to the other possible boundaries.

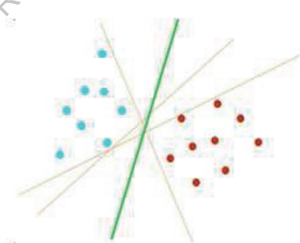


Figure 5.4: Optimal separating hyperplane.

5.3.1 The Optimal Separating Hyper plane

Consider the problem of separating the set of training vectors belonging to two separate classes,

$$\mathcal{D} = \left\{(x^1, y^1), \ldots, (x^l, y^l)\right\}, \quad x \in \mathbb{R}^n, y \in \{-1, 1\}, \qquad (2.1)$$

With a hyper plane,

$$\langle w, x \rangle + b = 0. \tag{2.2}$$

The set of vectors is said to be optimally separated by the hyperplane if it is separated without error and the distance between the closest vectors to the hyperplane is maximal.

There is some redundancy in Equation 2.2, and without loss of generality it is appropriate to consider a canonical hyperplane (Vapnik, 1995), where the parameters w, b is constrained by,

$$\min_{i} \left| \langle w, x^i \rangle + b \right| = 1. \tag{2.3}$$

This incisive constraint on the parameterization is preferable to alternatives in simplifying the formulation of the problem. In words it states that: the norm of the weight vector should be equal to the inverse of the distance, of the nearest point in the dataset to the hyperplane. The idea is illustrated in Figure, where the distance from the nearest point to each hyper plane is shown.

A separating hyperplane in canonical form must satisfy the following constraints,

$$y^i \left[\langle w, x^i \rangle + b \right] \geq 1, \quad i = 1, \ldots, l. \tag{2.4}$$

The distance d (w, b; x) of a point x form the hyperplane (w, b) is,

The optimal hyperplane is given by maximizing the margin, $\dot{\rho}$, subject to the constraints of equation 2.4, The margin is given by,

$$\begin{aligned} \rho(w, b) &= \min_{x^i : y^i = -1} d(w, b; x^i) + \min_{x^i : y^i = 1} d(w, b; x^i) \\ &= \min_{x^i : y^i = -1} \frac{\left| \langle w, x^i \rangle + b \right|}{\|w\|} + \min_{x^i : y^i = 1} \frac{\left| \langle w, x^i \rangle + b \right|}{\|w\|} \\ &\models \frac{1}{\|w\|} \left(\min_{x^i : y^i = -1} \left| \langle w, x^i \rangle + b \right| + \min_{x^i : y^i = 1} \left| \langle w, x^i \rangle + b \right| \right) \\ &= \frac{2}{\|w\|} \end{aligned} \tag{2.6}$$

Hence the hypeplane that optically separate the data is one that minimizes

$$\Phi(w) = \frac{1}{2} \|w\|^2. \tag{2.7}$$

It is independent of b becouse provided Equation 2.4 is satisfies (i.e. it is a sepaprating hyperplane) changing b will b will move it in the normal direction to itself. Accordingly the margiin remains unchanged but the hyperrplane is no longer optimal in that it will be nearer to one class than other. To consider how minimising Equation 2.7 is equivalent to implimenting the SRM proncipile, suppose that the following bound holds,

$$\|w\| < A. \tag{2.8}$$

The VC dimension, h, of the set of canonical hyperplanes in n dimensional space is bounded by,

$$h \leq \min[R^2 A^2, n] + 1, \tag{2.10}$$

where R is the radius of a hypersphere enclosing all the data points. Hence minimising Equation 2.7 is equivalent to minimising an upper bound on the VC dimension. The solution to the optimisation problem of Equation 2.7 under the constraints of Equation 2.4 is given by the saddle point of the Lagrange functional (Lagrangian) (Minoux, 1986),

$$\Phi(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{\ell} \alpha_i \left(y^i \left[\langle w, x^i \rangle + b\right] - 1\right), \tag{2.11}$$

where α are the Lagrange multipliers. The Lagrangian has to be minimised with respect to w, b and maximised with respect to $\alpha \geq \mathbf{0}$. Classical Lagrangian duality enables the primal problem, Equation 2.11, to be transformed to its dual problem, which is easier to solve. The dual problem is given by,

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left(\min_{w,b} \Phi(w, b, \alpha) \right). \tag{2.12}$$

The minimum with respect to w and b of the Lagrangian, Φ, is given by,

$$\begin{aligned} \frac{\partial \Phi}{\partial b} = 0 \quad &\Rightarrow \quad \sum_{i=1}^{l} \alpha_i y_i = 0 \\ \frac{\partial \Phi}{\partial w} = \mathbf{0} \quad &\Rightarrow \quad w = \sum_{i=1}^{l} \alpha_i y_i x_i. \end{aligned} \tag{2.13}$$

Hence from Equations 2.11, 2.12 and 2.13, the dual problem is,

$$\max_{\alpha} W(\alpha) = \max_{\alpha} -\tfrac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{k=1}^{l} \alpha_k, \tag{2.14}$$

and hence the solution to the problem is given by,

$$\alpha^* = \arg\min_{\alpha} \tfrac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle - \sum_{k=1}^{l} \alpha_k, \tag{2.15}$$

with constraints,

$$\begin{aligned} \alpha_i \geq 0 \quad & i = 1, \ldots, l \\ \sum_{j=1}^{l} \alpha_j y_j &= 0. \end{aligned} \tag{2.16}$$

-

Solving Equation 2.15 with constraints Equation 2.16 determines the Lagrange multipliers, and the optimal separating hyperplane is given by,

$$\begin{aligned} w^* &= \sum_{i=1}^{l} \alpha_i y_i x_i \\ b^* &= -\frac{1}{2} \langle w^*, x_r + x_s \rangle. \end{aligned} \tag{2.17}$$

where x_r and x_s are any support vector from each class satisfying,

$$\alpha_r, \alpha_s > 0, \quad y_r = -1, y_s = 1. \tag{2.18}$$

The hard classifier is then,

$$f(x) = \operatorname{sgn}(\langle w^*, x\rangle + b) \tag{2.19}$$

Alternatively, a soft classifier may be used which linearly interpolates the margin,

$$f(x) = h(\langle w^*, x\rangle + b) \quad \text{where} \quad h(z) = \begin{cases} -1 & : \quad z < -1 \\ z & : \quad -1 \leq z \leq 1 \\ +1 & : \quad z > 1 \end{cases} \tag{2.20}$$

This may be more appropriate than the hard classifier of Equation 2.19, because it produces a real valued output between -1 and 1 when the classifier is queried within the margin, where no training data resides. From the Kuhn-Tucker conditions,

$$\alpha_i \left(y^i \left[\langle w, x^i\rangle + b\right] - 1\right) = 0, \quad i = 1, \ldots, l, \tag{2.21}$$

and hence only the points x^i which satisfy,

$$y^i \left[\langle w, x^i\rangle + b\right] = 1 \tag{2.22}$$

will have non-zero Lagrange multipliers. These points are termed Support Vectors (SV). If the data is linearly separable all the SV will lie on the margin and hence the number of SV can be very small. Consequently the hyperplane is determined by a small subset of the training set; the other points could be removed from the training set and recalculating the hyperplane would produce the same answer. Hence SVM can be used to summarise the information contained in a data set by the SV produced. If the data is linearly separable the following equality will hold,

$$\|w\|^2 = \sum_{i=1}^{l} \alpha_i = \sum_{i \in SVs} \alpha_i = \sum_{i \in SVs} \sum_{j \in SVs} \alpha_i \alpha_j y_i y_j \langle x_i, x_j\rangle. \tag{2.23}$$

Hence from Equation 2.10 the VC dimension of the classifier is bounded by,

$$h \leq \min[R^2 \sum_{i \in SVs}, n] + 1, \tag{2.24}$$

and if the training data, x, is normalised to lie in the unit hypersphere,

$$h \leq 1 + \min[\sum_{i \in SVs}, n], \tag{2.25}$$

5.4 UML Diagram

5.4.1 Activity Diagram

The activity diagram describes the overall behavior and flow of the system. There is separate activity diagram for each type of assignment statements found in the source program.

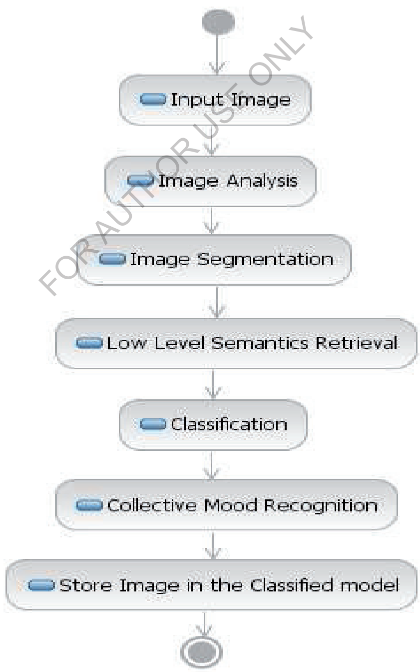


Figure 5.5: Activity diagram

Explanation:

- An input of an image is given from the set of database containing two fragments, one as the training dataset and another as the end user input.
- Analyze the particular image in terms of low level semantics.
- The specific image is put through different levels of segmentation.
- Retrieval of low level semantics in terms of color, hue, Chroma, luminance, brightness, contrast.
- Apply Support Vector Machine (SVM) classifier to classify the image into regions.
- This classifier would then identify the mood of an image.
- Store this image in classified model respectively.

4.2 Sequence Diagram

The sequence diagram describes the sequence of the processes taking place and interaction between different modules. A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

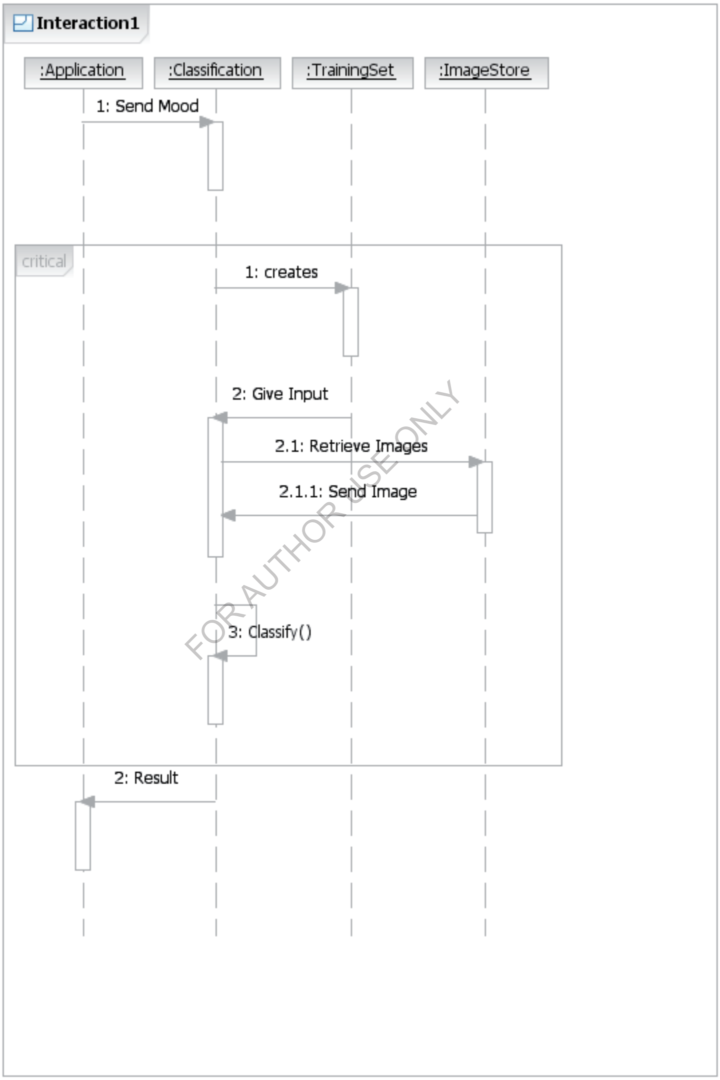


Figure 5.6: Sequence diagram

5.4.3 Class Diagram

The class diagram describes the different classes of objects in our system. The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematics of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the static relations between those objects. With detailed modeling, the classes of the conceptual design are often split into a number of subclasses.

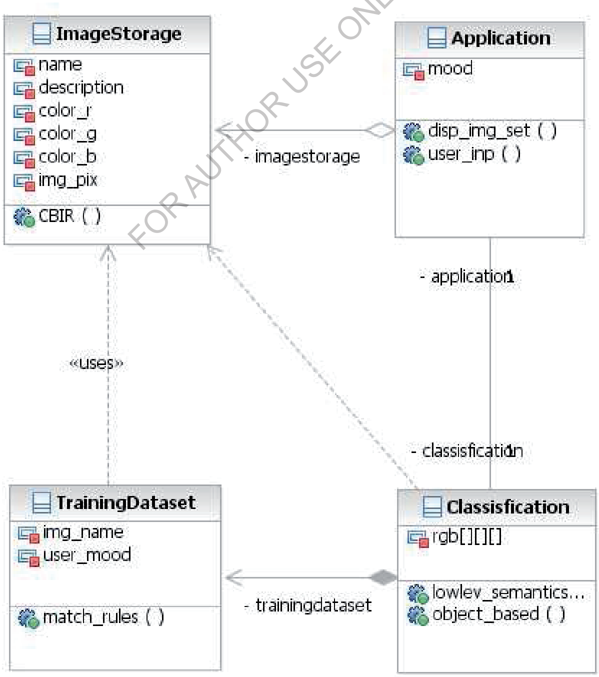


Figure 5.7: Class diagram

5.4.4 Package Diagram

The package diagram shows that our system uses 2 main packages found in our work i.e. User visible package and system package.

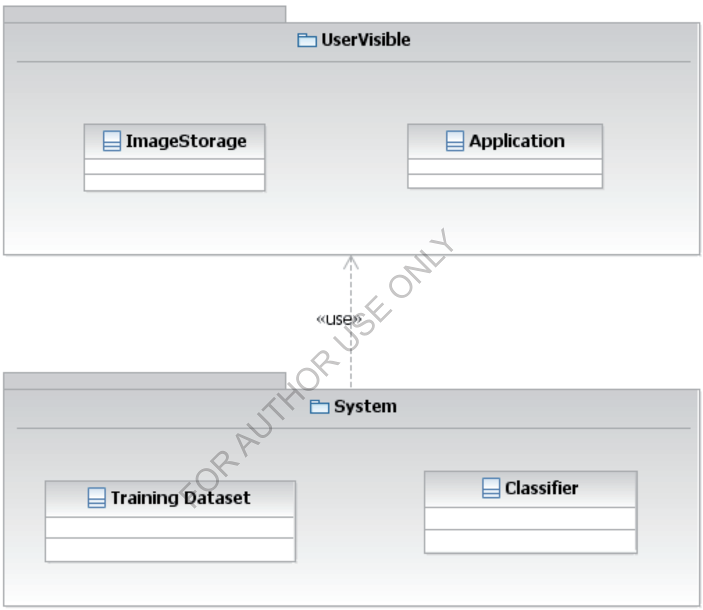


Figure 5.8: Package diagram

5.4.5 State Machine Diagram

The State Machine diagram shows the transition of states of the system based on processes taking place.

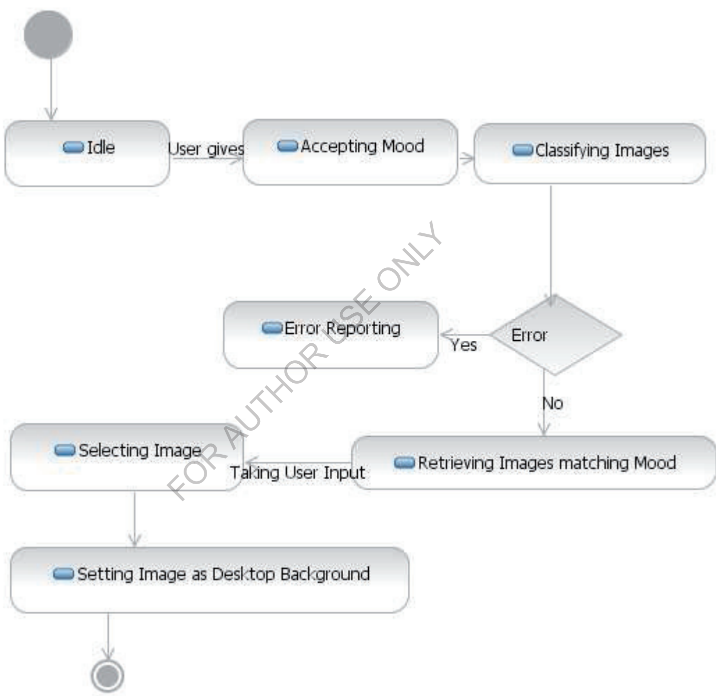


Figure 5.9: State machine diagram

5.4.6 Component Diagram

Component diagram depicts how components in our system are wired together to form larger components and/or software systems. This diagram illustrates the structure of our system in terms of its components.

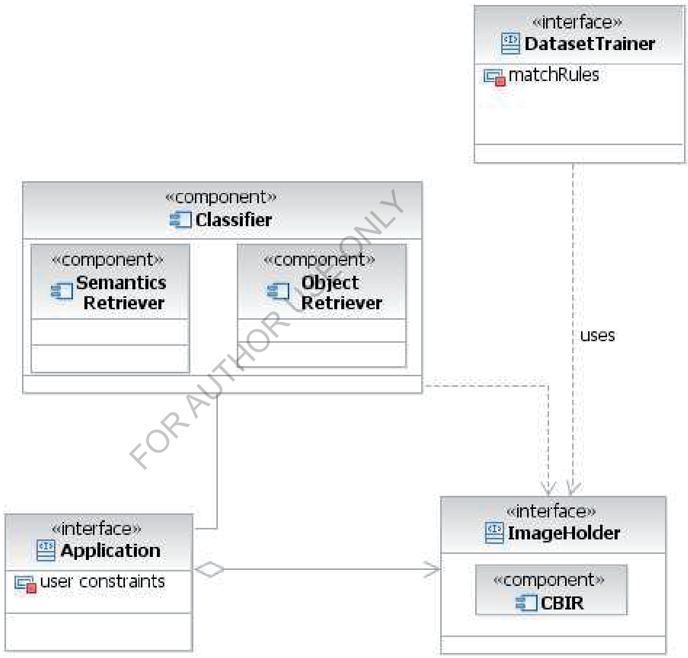


Figure 5.10: Component diagram

5.4.7 Deployment Diagram

A deployment diagram in the unified modeling language models the physical deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI). The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

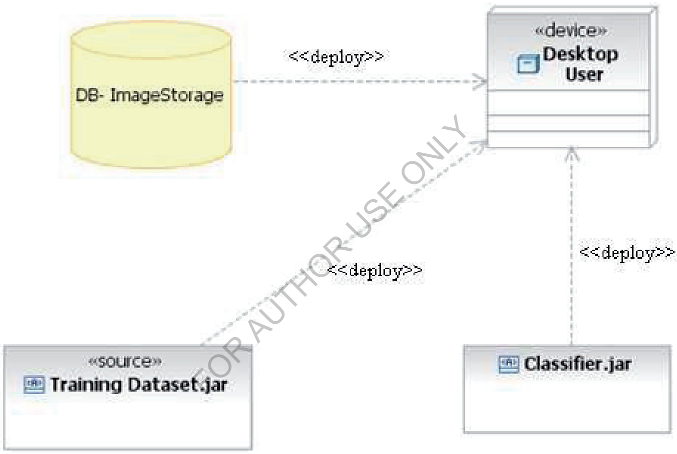


Figure 5.11: Deployment diagram

5.5 Use Case

5.5.1 Use Case Diagram

Our use case diagram shows the different use cases pertaining to the various functions performed by our system.

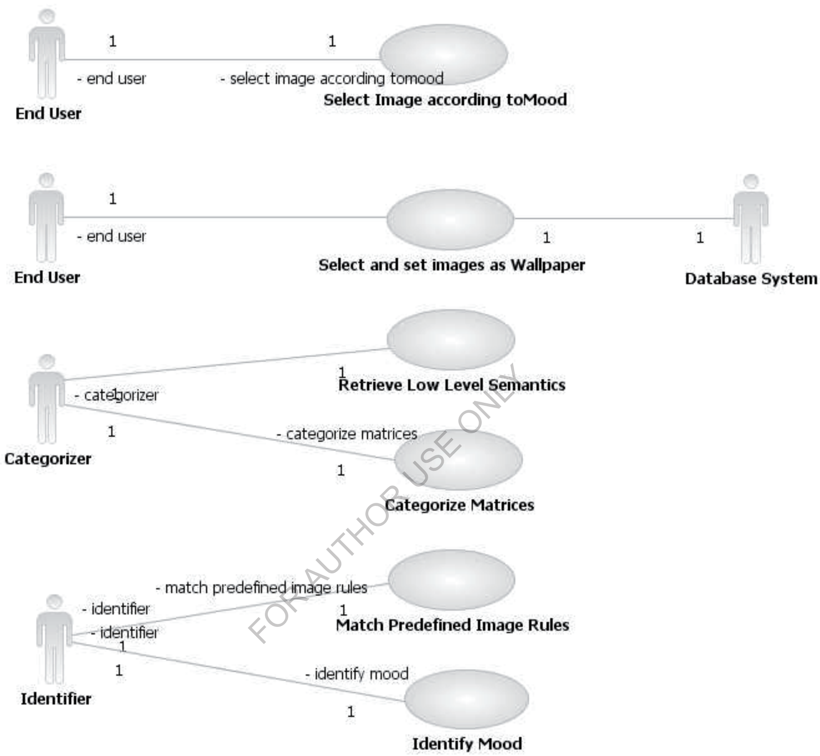


Figure 5.12: Use case diagram

5.5.2 Use Case Specifications

Use case 1: Select Image according to Mood

Primary Actor: User

Precondition:

A database containing collection of images categorized according to a specific mood.

Main Scenario:

1. User specifies mood or emotion.
2. The system provides the user with set of images matching and categorized according to the mood.

Alternate Scenario:

No such image of desired mood is found in the database and return with no images found.

Post condition:

1. Set of images with user desired mood. OR
2. No Images found.

Use case 2: Select and Set Image as Wallpaper.

Primary Actor: End User.

Precondition:

The user needs to specify the mood for natural scene image. There must be an availability of the required image in the database.

Main Scenario:

1. The system would provide the user with set of images matching to the mood given.
2. The user would select an image from the set which would be applied as wallpaper respectively.

Alternative Scenario:

The system would provide user with no images.

Post condition:

The selected image would be applied as desktop background.

Use case 3: Retrieve Low Level Semantics.

Primary Actor: Categorizer

Precondition:

Image should be converted to grey-scale.

Main Scenario:

The image is categorized on the basis of different parameters like Edge, Texture and Color with the help of classifiers.

Post condition:

This categorization would be stored to database in the form of Image information.

Use case 4: Identify Mood.

Primary Actor: Identifier

Precondition:

The low level Semantics of an image is retrieved and is categorized on the basis of its parameters by the categorizer.

Main Scenario:

1. The system will differentiate the image to different parameters.
2. On the basis of these parameters the identifier will identify the mood of an Image.

Post condition:

The identified mood will be set as a mood description of an image and will be updated to the database for further categorization.

Use case 5: Match Predefined Image Rules.

Primary Actor: Identifier.

Precondition:

The mood of an image is identified by the Identifier.

Main Scenario:

The identified mood of an image will be matched to the input i.e. the emotion specified by the user.

Alternative Scenario:

The mood of a specific image is not identified by the identifier.

Post condition:

If the emotion user specified matches the predefined image rules the image would appear in the 'set' (resembling to particular emotion only) for further selection as desktop background.

CHAPTER 6
CODING TECHNIQUE

6.1 Code specification

Software used:

This work is developed using the Java technology which is an object-oriented, platform-independent, multithreaded programming environment. Java SE 7 (JDK 1.7 and JRE 1.7) is used for the work. Integrated Development Environment (IDE) is useful tool for developing software programs. The Net Beans Platform is a reusable framework for simplifying the development of Java Swing Desktop Applications.

So, Net beans IDE 7.1.2 and Java are being used in this work.

6.2 Coding Style

Coding conventions are a set of guidelines for a specific programming language that recommend programming style, practices and methods for each aspect of a piece program written in this language. These conventions usually cover file organization, indentation, comments, declarations, statements, white space, naming conventions, programming practices, principles, programming, architectural best practices, etc. These are guidelines for software structural quality. Software programmers are highly recommended to follow these guidelines to help improve the readability of their source code and make software maintenance easier. Coding conventions are only applicable to the human maintainers and peer reviewers of a software work. Conventions may be formalized in a documented set of rules that an entire team or company follows, or may be as informal as the habitual coding practices of an individual. Coding conventions are not enforced by compilers. As a result, not following some or all of the rules has no impact on the executable programs created from the source code.

The following coding standards are used:

1. The normal coding style for variable naming is used in Java coding. All the variables are consistent through the programs.
2. Declaration of the variable is supplemented with a comment describing its usage and purpose.
3. Each function is supported with a comment to describe its functionality.
4. Each call of the function has an attached comment that tells what all parameters are passed and what each parameter means.
5. Standard procedure for indentation has been followed.
6. Code has been modularized to support maximum reuse.

CHAPTER 7
SYSTEM TESTING

7.1 Test Specification

7.1.1 Objectives of Testing

Software testing is an important element of software quality assurance and represents ultimate review of specification, design and coding of the system.

A successful test is one that reveals errors that are yet undiscovered. Testing demonstrates that software functions behave according to specification and also that the performance requirements have been met also the data collected while testing provides a good indication of software reliability and quality. It is general principle of testing that all tests should be traceable to the customer requirements.

Following rules serve well as testing objectives:

. Executing all the modules with intent of finding errors.
. Testing all the modules to find an as yet undiscovered error.
. Testing that specifications are implemented as documented.
. Check that the documented specifications are what the business needs.
. Check that the system performs in the desired manner in the target environment.

Testing Principles:

. All tests should be traceable to customer requirements.
. Tests should be planned long before the testing begins.
. Testing should begin "in the small" and progress towards testing "the large".
. To be most effective a third should carry out the testing.
. Testing should begin from requirement gathering phase.
. Testing should be left at the end of the system development like cycle.
. Continuous testing concept should be applied instead of Big Bang Testing.

As the potentials failure cases errors are discovered and corrected the reliability of the product increases. Thus by testing we can ensure better quality of the product, which in turn increases its chance of acceptance by the customer.

7.2 Testing Plan:

Software test plan is the document that defines the overall approach and objectives of testing for the SUT (System under Test). This document provides the directions of all the testing activities. Test plan to be created at the beginning of the Requirements phase and contains Test Scope, Test Objectives, Test Design, Test Schedules and Test Tools details. The first task is to establish, and seek confirmation from the customer, clear understandings of the work and its deliverables.

Exhaustive analysis will ensure that there is no mismatch between our understandings and the customer requirements. The entire relevant product, interface, components, and other external dependences are identified timeframe for delivering the result is computed. The key steps considered while creating a test plan are:

- Define release criteria.
- Test Scope and Objectives.
- What is to be tested and what is not to be tested.
- Test design.
- Test tools to be used.
- Defect tracking system.
- Outline and prioritize the testing effort.
- Identify resource requirements at various stages of testing.
- Set up calendar based activity plan.

The biggest challenge is to ensure that there are no product failures or production delays due to inadequate testing. Time to market and quality are paramount and for number of devices, can meet your requirements and provide customized testing services to fulfill your needs.

The best approach to testing is to start with basic functionality and gradually add levels of complexity at each successive stage. As each test completes the result is documented and verified against the work requirements. Any problem if found, should be investigated and resolved.

7.3 Testing Strategy

Testing begins "in the small" and progresses "to the large". By this we mean that early testing focuses on a single component and applies white and black-box tests to uncover errors in program logic and function. After individual components are tested they must be integrated. Testing continues as the software is constructed. Finally, a series of high order tests are executed once full program is operational.

7.3.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design-the software component or module. The unit test is white box oriented and can be conducted in parallel for multiple components. In computer programming, unit testing is a procedure used to validate that individual units of source code are working properly. A unit is the smallest testable part of an application**.**

Module 1 – Preprocessing Testing

Table 7.1: Pre-processing module testing

Sr. No	Test Objective	Expected output	Actual output
1	To check whether color features to be extracted.	Feature extracted.	Same as expected

Module 2 – Image Mapping Testing

Table 7.2: Image mapping module testing

Sr. No	Test Objective	Expected output	Actual output
1	To check whether the image mapping its retrieved information	Image mapped.	Same as expected.

Module 3 –Emotion Recognition Testing

Table 7.3: Emotion Recognition module testing

Sr No	Test Objective	Expected output	Actual output
1	To check for all images emotion if the emotions are recognized correctly.	Emotion recognition rate should be 100%	Emotion recognition rate is 75%

7.3.2 Validation Testing

This type of usability testing usually occurs later in the software life cycle, close to release time, and is intended to certify the software's usability. A principal objective of validation usability testing is to evaluate how the product compares to some predetermined usability standard or benchmark. Testers want to determine whether the software meets the standards prior to release; if it does not, the reasons for this need to be established.

7.3.3 GUI Testing

GUI software testing is the process of testing graphical user interface-based software to ensure it meets its written specifications. This is normally done through the use of a variety of test cases.

To generate a ‘good’ set of test cases, the test designer must be certain that their suite covers all the functionality of the system and also has to be sure that the suite fully exercises the GUI itself.

Table 7.4: GUI testing

Sr. No	**Test objectives**	**Expected output**	**Actual output**
1	To check whether button click gives the desired output.	Desired dialog box opens.	Same as expected

7.3.4 System Testing

This is type of testing to confirm that all code modules work as specified, and that system as a whole performs adequately on the platform on which it will be deployed. System testing is performed by the testers who are trained to plan, execute and report on application and system code.

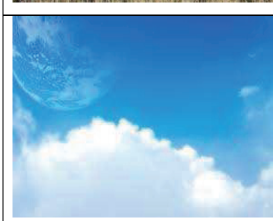
Table 7.5: System Testing

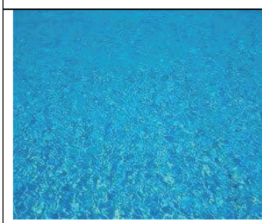
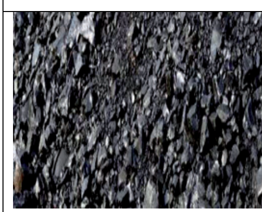
Sr. No	**Test Objective**	**Expected output**	**Actual output**
1	To check for all images emotion if the emotions are recognized correctly.	Emotion recognition rate should be 100%	Emotion recognition rate is 85%

CHAPTER 8
PERFORMANCE MEASUREMENTS

8.1 Testing Result based on Human Perception

Table 8.1: Performance analysis: Result based on Human Perception

Images	Output Expected	Emotion Detected by a Particular Age Group (10 people in each group)						Majority say
		12-25		25-40		41-60		
		Happy	Sad	Happy	Sad	Happy	Sad	
	Happy Grass	8	2	7	3	8	2	Happy
	Sad Grass	3	7	1	9	2	8	Sad
	Happy Sky	10	0	9	1	8	2	Happy

	Happy Tree	6	4	7	3	7	3	Happy
	Sad Tree	3	7	2	8	1	9	Sad
	Happy Water	7	3	6	4	9	1	Happy
	Sad Rock	1	9	2	8	2	8	Sad
	Sad Sky	1	9	0	10	1	9	Sad

By above performance analysis we observe that even though the matter of emotion interpretation differs from person to person, still an emotion of a particular pattern image can be universally identified.

CHAPTER 9

FUTURE ENHANCEMENTS AND APPLICATIONS

9.1 Future Enhancement

Our application works only for two emotions which is happy and sad. It can be enhanced in future to detect all other emotions like angry, gloomy, chirpy etc.

Emotion detection in natural scene images basically means that the natural scene image should be classified properly based on image semantics. Any image can be classified into three levels of semantics (low level, medium level, high level). Our task is to bridge the gap between the different levels. This work deals with the color component of an image. The tasks to be performed are: Build Classifiers for every mood retrieve low level semantic information of the chosen image accordingly classify the image mapping its retrieved information. Although different people may interpret the same image in different ways, we still can build a universal classification for different emotions. However, it is a challenging task for any machine to recognize emotion in any natural scene image we still can build classifiers which can help the machine to adequately classify images according to different emotions.

9.2 Proposed Applications

- E-Greeting Cards:

We can use the natural scene images in generating greeting cards. According to the occasion for which we are sending the card we can directly generate the card by just entering the mood for which it being generated.

CHAPTER 10
CONCLUSION

10.1 Conclusion

Emotion of an image depicting natural scene is identified correctly by our work. The features of the images were successfully extracted which were then given to the classification algorithm which is a Support Vector machine algorithm to segregate the image into classes and therefore detect the emotion of an image respectively.

Classification of emotions based on natural scene images is a new concept in an innovative field of Image Processing domain. Image processing domain has always proven to be challenging criteria in field of research and development. This work demands a thorough study of every concept related to Image retrieval, emotion detection and CBIR (Content Based Image Retrieval) technique. Emotion in natural scene images plays an important role in the way humans perceive an image. Based on the emotion (happiness, sadness, fear, anger etc.) of any human being the images that are viewed by that person can have a significant impact in a sense that if the person is for example in happy mood and he/she views an image that is pleasing then he/she would have a better sense of attachment towards that image and would not accept an image that depicts sadness as an emotion. Although different people may interpret the same image in different ways, we still can build a universal classification for different emotions.

CHAPTER 11

SCREEN SHOTS

11.1 Application

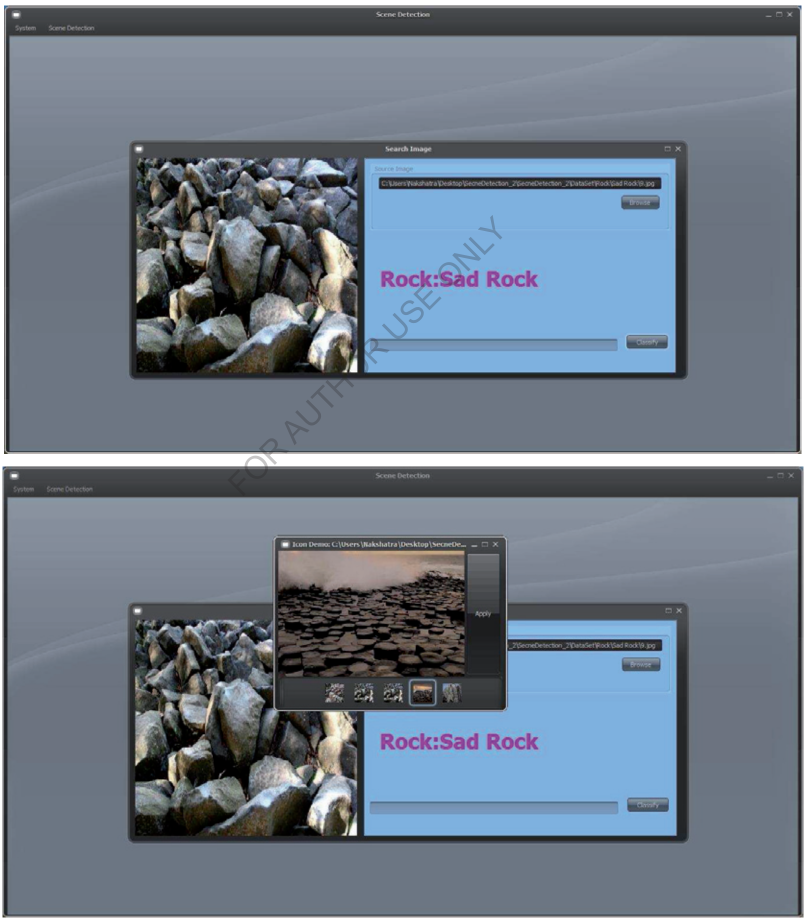


Figure 11.1: Screen shot (Application)

11.2 Feature Extraction-

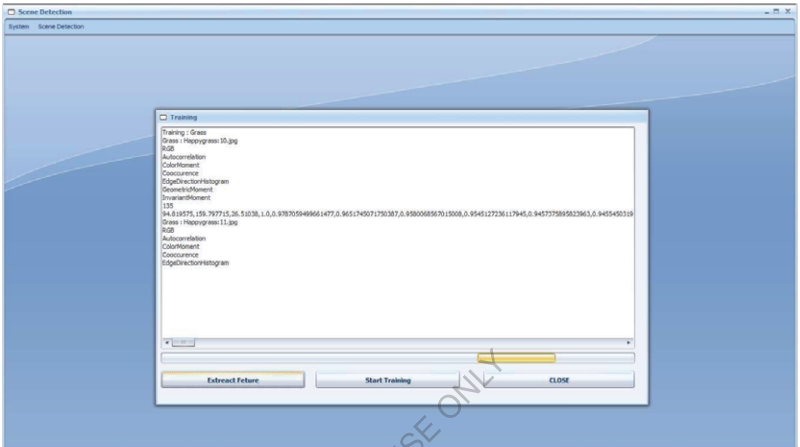


Figure 11.2: Screen shot (Feature Extraction)

11.3 Training Dataset

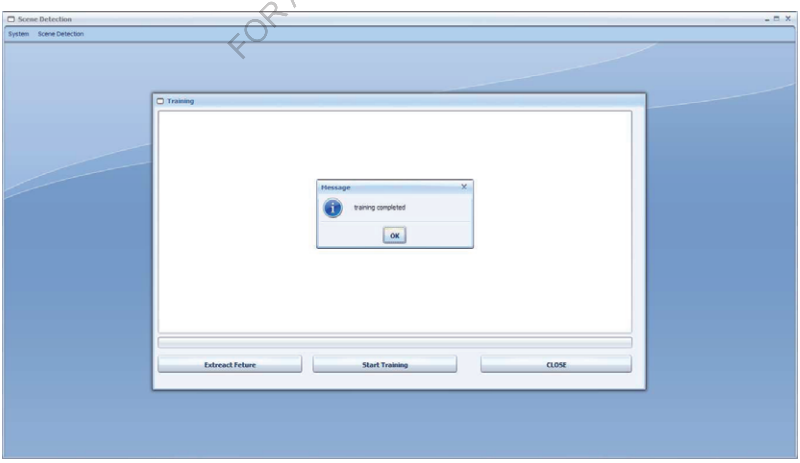


Figure 11.3: Screen shot (Training dataset)

11.4 Testing Dataset-

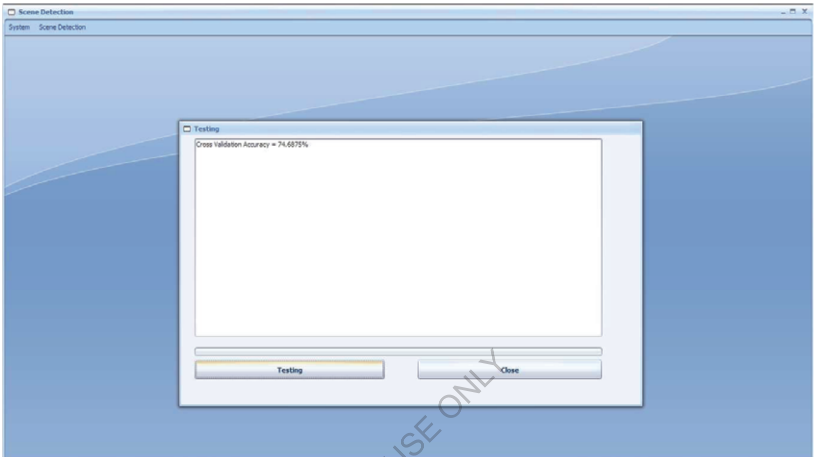


Figure 11.3: Screen shot (Testing Dataset)

APPENDIX A

REFERENCES

[1] A. Mojsilovic and B. Gomes, J.and Rogowitz, "Semantic-Friendly Indexing and Quering of Images Based on the Extraction of the Objective Semantic Cues", International Journal of Computer Vision, 56:79–107, 2004.

[2] Ch. Chang and Ch. Lin., "LIBSVM: A library for sup-port vector machines", ACM transactions on Intel-ligent Systems and Technology, 2:27:1–27:27, 2011.

[3] D. Comaniciu, P. Meer, et.al, "A robust approach toward feature space analysis", IEEE transactions on Pattern Analysis and Ma-chine Intelligence, 24:603 – 619, 2002.

[4] E. Sikudova´, "On Some Possibilities of Automatic Image Data Classification" PhD thesis, Comenius University, Bratislava, Slovakia, March 2006.

[5] H. B. Kekre, et.al, "Cbir using upper six FFT sectors of color images for feature vector generation", International Journal of Engineering and Technology, 2:49–54, 2010.

[6] Jianjiang Lu, Zhenghui Xie, Ran Li, Yafei Zhang, and Jiabao Wang., "A Framework of Cbir System Based On Relevance Feedback", In Proceedings of the 3rd international conference on Intelligentinfor-mation technology application, IITA'09, pages 175– 178, Piscataway, NJ, USA, 2009. IEEE Press.

[7] J. Vogel and B. Schiele, "Semantic Modeling of Natural Scenes for Content-Based Image Retrieval. Int. J. Comput. Vision" 72:133–157, April 2007.

[8] J. Wu, Z. Lin, and M. Lu, "Asymmetric semi-supervised boosting for SVM active learning in cbir", In Proceedings of the ACM International Conference on Image and Video Retrieval, CIVR '10, pages 182– 188, New York, NY, USA, 2010. ACM.

[9] R. S. Choras, “Image feature extraction techniques and their application for cbir and biometrics systems”, International Journal of Biology and Biomedical Engineering, 1:6–16, 2007.

[10] Available online: V. Kazemi. Face detector (boosting haarfea-tures), 2010. http://www.mathworks.com/matlabcentral/fileexchange/27150\ -face-detector-boostinghaar-features.

[11] V. Mezaris, I. Kompatsiaris, and M.G. Strintzis, “An ontology approach to object-based image retrieval”, International Conference on Image Processing (ICIP 2003), volume 2, pages II – 511–14 vol.3, sept. 2003.

[12] Y. Liu, D. Zhang, G. Lu, and W. Ma., “A Survey of Content-Based Image Retrieval with High-Level Seman-Tics” Pattern Recognition, 40(1):262 – 282, 2007.

[13] Y. Zhuang, X. Liu, and Y. Pan, “Apply semantic template to support content-based image retrieval”, In Proceeding of IST and SPIE Storage and Retrieval for Media Databases, pages 23–28, 2000.