**SVM BASED CLASSIFICATION BY USING MOMENT AS FEATURE VALUE**

**Report**

Prepared

**By**

**Love Kumar Yadav**

**ABSTRACT**

We present an object classification technique using feature extraction for gray scale image by second order moment calculation and classification of the object present has been done by using the concept of support vector machine (SVM).

Features are extractable measurements from a sample image summarizing the information content in an image and in the process providing an essential tool in image understanding. In particular, they are useful for image classification into pre-defined classes or grouping a set of image samples (also called clustering) into clusters with similar within-cluster characteristics as defined by such features.

A generic feature extraction process can be built that does not involve the nature of attributes, but just attribute value. In pattern recognition and in image processing, feature extraction is special form of dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously then the input data will be transformed into a reduced representation set of features (also named feature vector). Transforming the input data into the set of features is called feature extraction.

In this project, we tend to extract feature from a given set of input images and create a feature matrix. The method being used for this purpose is moment calculation. Moments are scalar quantities used to characterize a function and to capture its significant features. They have been widely used for hundreds of years in statistics for description of the shape of a probability density function and in classic rigid-body mechanics to measure the mass distribution of a body. From the mathematical point of view, moments are “projections” of a function onto a polynomial basis (similarly, Fourier transform is a projection onto a basis of harmonic functions).

Next part for the project is training. In this project training has been done by using svmtrain() function of MATLAB. Finally the system has classified by using SVM classification technique. It has been noticed that the project is capable to classify any object in which class the object belongs to.

**1. INTRODUCTION**

**1.1 General Overview of the problem**

The general overview about the basic terminology which are used to develop this project ‘SVM based classification by using moment as feature value’ are as follows:

**Image**

The term image refers to two dimensional light intensity function f(x,y), where x and y denote spatial coordinates and value f at any point(x,y) is proportional to the brightness or gray level at that point.

A digital image is an image f(x,y) that has been discredited both in spatial coordinates and brightness. A digital image can be considered as a matrix whose rows and columns indices identify a point in the image and corresponding matrix element value identifies the gray level or RGB values at that point.

The elements of such a digital array are called elements, picture elements and pixels.

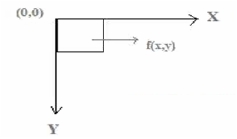


Fig 1: Basic element of digital image

Figure: Digital Image Representation

**Grayscale Image**

A grayscale image is an image in which the value of each pixel is a single sample i.e. it carries only intensity information. Images of this sort are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest.

Often, the grayscale intensity is stored as an 8-bit integer giving 256 possible different shades of gray from black to white. If the levels are evenly spaced then the difference between successive graylevels is significantly better than the graylevel resolving power of human eye.

Grayscale images are very common, in part because much of today’s display and image capture hardware can only support 8-bit images. In addition, grayscale images are entirely sufficient for many tasks and so there is no need to use more complicated and harder-to-process color images.

**Feature**

Features are extractable measurements from a sample image summarizing the information content in an image and in the process providing an essential tool in image understanding.

Defining feature vectors remains the most common and convenient means of data representation for classification problems.

At the lowest level, features may be the intensity levels of a pixel in an image. The intensity levels of the pixels in an image may be derived from a variety of sources. For example, it can be the temperature measurement (using an infra-red camera) of the area representing the pixel or the X-ray attenuation in a given volume element of a 3-d image or it may even represent the dielectric differential in a given volume element obtained from an MIR image.

At a higher level, geometric descriptors of objects of interest in a scene may also be considered as features in the image. Examples of such features are: area, perimeter, aspect ratio and other shape features, or topological features like the number of connected components etc.

This shows that image features can refer to two categories of image properties:

* *Global properties of an image:* i.e. moments, average gray level, shape of intensity, histogram etc.
* *Local properties of an image:*We can refer to some local features as image primitives: circles, lines etc. Other local features are shape of contours etc.

**Moment**

Moments are scalar quantities used to characterize a function and to capture its significant features.

From the mathematical point of view, moments are “projections” of a function onto a polynomial basis (similarly, Fourier transform is a projection onto a basis of harmonic functions).

Further, in image analysis, they are*statistical moments*, as opposed to *mechanical* ones, but the two are analogous. For example,the mechanical moment of inertia describes the rate of change in momentum; the statistical second-order moment describes the rate of change in a shape’s area.

**Second Order Moment**

The second order moments, {*M02 ,M11, M20*}, known as the moments of inertia, may be used to determine an important image feature, orientation. In general, the orientation of an image describes how the image lies in the field of view, or the directions of the principal axes.

In terms of moments, the orientations of the principal axes, , are given by

**(1)**

In (1.11), is the angle of the principal axis nearest to the axis and is the range .

**Moment Invariants**

Moments of an image are invariant to certain class of image degradations. They are:-

* Rotation, translation, scaling
* Affine transform
* Elastic deformations
* Blurring
* Combined invariants

Example: Images used to demonstrate properties of moment invariants

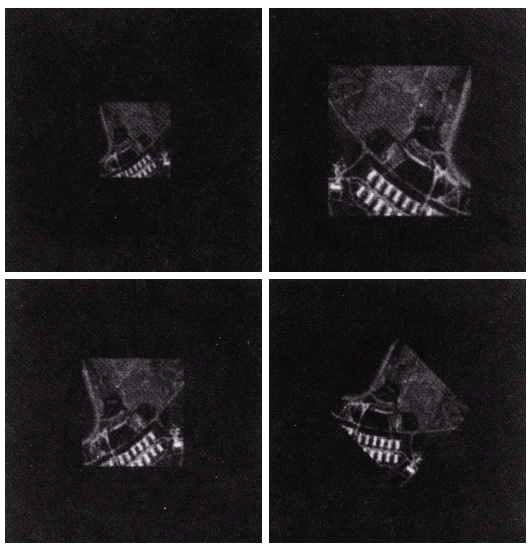
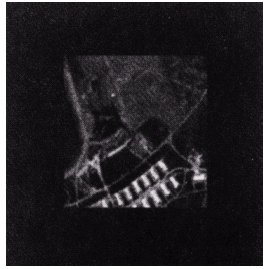
****

Fig (2.1) Original image [7]

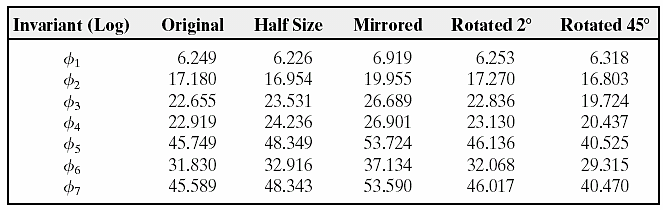
Half-size (top left)

Mirrored (top right)

Rotated 2° (bottom left)

Rotated 45°

Fig (2.2) Altered images [7]

****Fig (2.3) Moment invariants for the images in Figure (2.2)(taken from [7])

**Linear classification models**

Linear classification models, or linear classifiers, are classification models where the separation between the two classes is expressed as a weighted sum of the input variables. Forexample, suppose we have input variables *xi ,*where *i* ranges from 1 to *n*. The prediction of alinear classifier is

**(2)**

Where,*wi*is the weight on the *i*th input variable and *c* is a constant. Notice that linear classifiersare identical to linear regression models up to a finalthresholding operation that converts modeloutputs into one of two possible values. Also, note that in the formulation above, there is no needto add a constant regressor into the model since the flexibility of a constant regressor is alreadycontained in the *c* term.

The boundary that separates the two classes is called the *decision boundary*. For linearclassifiers, the decision boundary is an *n*-dimensional *hyper plane*. This is just a fancy term forthe following concept: When *n* = 1, there is one input variable and the decision boundary is apoint. When *n* = 2, there are two input variables and the decision boundary is a line. When *n* = 3,there are three input variables, and the decision boundary is a plane. And so on.

There are various flavours of linear classifiers, including logistic regression, linear discriminantanalysis (LDA), linear support vector machines (SVM), nearest-prototype classifiers, and Naïve Bayes classifiers. All of these implement the same type of model (described above). However,different classifiers make different assumptions regarding the nature and distribution of the data,and therefore produce different estimates of the parameters of the linear classification model (the*wi*and *c* terms above).

**Support Vector Machines (SVM)**

Support vector machine (SVM) is a powerful technique that performs especially well for high-dimensionalproblems (i.e. problems where there are many input variables compared to datapoints). The key concept in SVM is *maximizing the margin*, that is,maximizing the distancebetween the decision boundary and the nearest data points in the sample. The idea is that a gooddecision boundary is not just one that successfully classifies the data points but also one thatmaximizes the margin, as this will likely generalize well to novel data.

There are both linear and nonlinear versions of SVM. Nonlinearity is achieved through the*kernel trick*. The kernel trick achieves the same result as nonlinear expansion of the input spacebut avoids actual construction of the higher-dimensional space.

**Entropy**

Entropy is a thermodynamic property that is the measure of a system’s thermal energy per unit temperature that is unavailable for doing useful work.

The entropy H is claimed to express a measure of ignorance about the actual structure of the system. The definition of the information entropy is, however, quite general, and is expressed in terms of a distance set of probabilities pi:

**(3)**

In the case of transmitted message, these probabilities were the probabilities that a particular message was actually transmitted, and the entropy of the information system was a measure of the average amount of information in a message.

**1.2 Literature Survey**

[1] Fukunuga et al. states that computational object classification is such a hard problem is that machines take sensory information very literally making object recognition vulnerable to accidental scene information. Such accidental variations include scale, illumination color, viewing angle, background, occlusion, shadows, shading, light intensity, highlights, and many more. One approach to dealing with such photometric variations is found in the use of invariant features. Invariant features remain unchanged under certain operations or transformations and used for various object recognition approaches. For example, the physical laws of image formation can be used to factor out accidental scene effects.

[2] A. K Jain introduced feature extraction theory; and said that features are very application oriented and often found by heuristic methods and interactive data analysis. It is not possible to give an overview of such interactive feature extraction methods; in any specific problem such as, e.g., character or speech recognition, there is an accumulated knowledge of the most feasible ways to extract the relevant information. Speed and accuracy of processing are two important factors for feature extraction method. The accuracy may be more important than the speed.

[3] Levine M D reviews number of techniques for feature extraction which can be used in image processing application. Feature extraction or selection is a pivotal procedure considerably for currency recognition, which effects on design and performance of the classifier intensively. If the differences of selected features are so large, it can easily construct a classifier with good recognition performance. It is difficult to get it with the contrary situation. The essential task of feature task of feature extraction and selection is how to find the correspondingly effective features out of many pending features.

[4] Liang-Chi et al. categorized feature types as follows:

* Structural features: It describes geometrical and topological characteristics of a pattern by representing its global and local properties.
* Statistical features: Statistical features are derived from the statistical distribution of pixels and describe the characteristic measurements of the pattern.
* Global transformation: Global transformation technique transforms the pixel representation to a more compact form. This reduces the dimensionality of the feature vector and provides feature invariants to global deformation like translation, dilation and rotation.

[5] Schalkoff R J states that, feature extraction is a special form of dimensionality reduction. Defining feature vectors remains the most common and convenient means of data representation for classification and regression problems.When the input data is too large to be processed and it is redundant in nature then the input data will be transformed into a reduced representation set of features (also named features vector). Transforming the input data into the set of features is called feature extraction. Feature extracted provides the relevant information from the input images which can be used for their classifications. Feature extraction involves simplifying the amount of resources required to describe a large set of data accurately. It can be used in the area of image processing which involves using algorithms to detect and isolate various desired portions or shapes (features) of a digitized image.

**1.3 Problem Definition**

Humans are capable of distinguishing the same object from millions of different images.Machines on the other hand have significant difficulty with this seemingly trivial task.

One of the reasons that computational object recognition is such a hard problem is that machine takes sensory information very literally like illumination, viewing angle, and shading.

Feature extraction deals with extraction of feature from an image that is invariant to transformations like rotation,translation,and affine transformation etc. classification deals with Categorizing objects into different categories.

**1.4 Problem Analysis**

The main task of our system is to extract feature from a given set of input images and create feature matrix .The method being used for this purpose is moment calculation .Moments are scalar quantities used to characterize a function and to capture it significant feature .after creation of feature matrix training the system to classify the images taken as input using SVM(support vector machine). The final implementation of the project will result in being capable to distinguish any object whether the object belongs to pre-defined class of input data set which has been used for training the system.

The steps involved in the system are given below:-

* {Input :Raw data}
* {Output: class of input image}
* 1.Data acquisition
* 2. Conversion of image to gray scale
* 3. Create feature matrix by moment calculation
* 4. Train the system for classification
* 5. Classify the given input image

**System Requirements and Specification**

**Software Specification:**

Platform : MATLAB R2012a

Operating System : Windows 7 or Service Pack 1

Windows VistaTM Service Pack 2

Windows XP Service Pack 3

Windows XP x64 Edition Service Pack 2

**Hardware Specification:**

Processor : Any Intel® or AMD x86 Processor

RAM : At least 1024 MB (2048 MB recommended)

Disk Space : 1 GB (MATLAB only)

3-4 GB for a typical installation

**1.5 Proposed Solution Strategy**

There is no well-developed theory for feature extraction; mostly features are very application oriented and often found by heuristic methods and interactive data analysis. It is not possible to give an overview of such interactive feature extraction methods; in any specific problem such as, e.g., character or speech recognition, there is an accumulated knowledge of the most feasible ways to extract the relevant information. Speed and accuracy of processing are two important factors for feature extraction method. The accuracy may be more important than the speed. Currently, there are a number of techniques for feature extraction which using image processing.

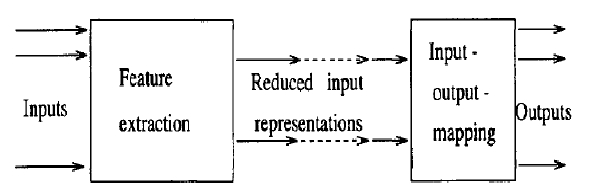
Feature extraction or selection is a pivotal procedure considerably for currency recognition, which effects on design and performance of the classifier intensively. If the differences of selected features are so large, it can easily construct a classifier with good recognition performance. It is difficult to get it with the contrary situation. The essential task of feature extraction and selection is how to find the correspondingly effective features out of many pending features.

Fig 3: The Feature Extraction Approach

**ALGORITHIM**

Input: {Training data set, testing data set}

Output: {Line separating the two classes of objects}

Step1. Start

Step2. Read all image in training data set.

Step3. Convert all images in training data set in gray scale image.

Step4. Resize all gray scale images to size (50\*50).

Step5. Plot color image gray scale image & histogram in single window.

Step6. Extract feature for each gray scale image by calculating second order moment.

Step7. Generate feature matrix for training data set.

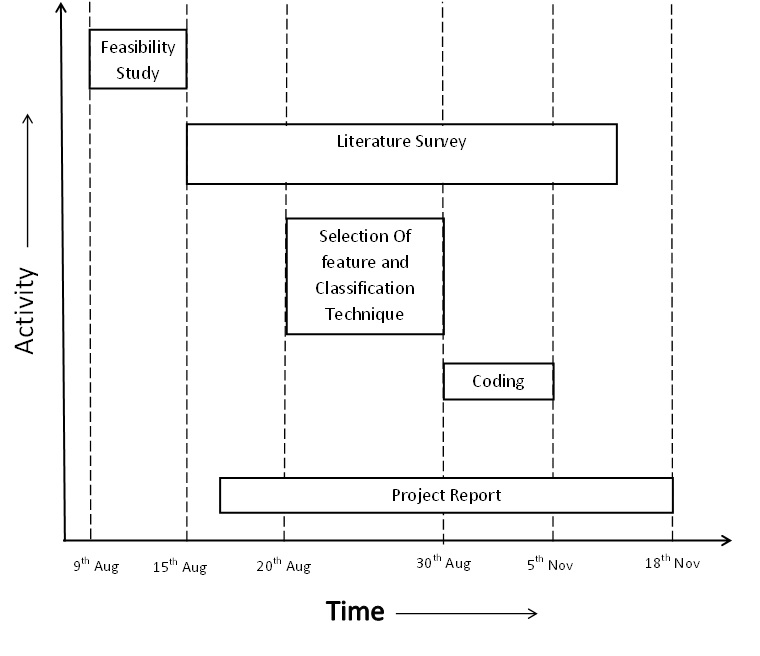
Step8. Read all image from testing data set.

Step9. Convert all images in testing data set to gray scale image.

Step10. Use SVM to classify two classes of objects.

Step11. Generate plot with line separating two classes of objects.

**1.6 Project Schedule**

**Gantt Chart**

**1.7 Organization of the report**

This project report is organized according to the different development phase of software Development:

* At the very outset, the Abstract of the project is provided.
* This is followed by a brief, general introduction to the initial problem. A formal definition to the problem and then Analysis follows. Then the Software Requirement Specification describes the requirements of the system. Finally, the proposed solution strategy has been presented.
* Next the detailed Algorithm for the project is elucidated in which the detailed step-by-step approach for gathering data and implementing the clustering algorithm and feature extraction is provided. Also, the fundamental concepts of the algorithm are mentioned here.
* The implementation details come next which outline the need for the platform used to develop the feature extraction model.
* The Result and Discussions cover the outcome of the various classifications carried out using different algorithms, and how they helped to discover and analyze faults and errors.
* The Result and Discussions cover the outcome of the various classifications carried out using different algorithm, and how they helped to discover and analyze faults and errors.
* The Summary and conclusion gives a list of all that has been achieved and accomplished. It lists the main difficulties encounter during the implementation and how they were overcome. It also specific the limitations of the project and gives a list of all future modification and improvements.

**2. DESIGN STRATEGY FOR THE SOLUTION**

**2.1 Architecture Diagram**

Software design is the process through which requirements are translated into a representation of software. Initially the representation depicts a hostile view of the software. Subsequent refinement leads to a design representation that is very close to the source code. From a project management point of view, software design is conducted in two steps namely Architectural design and Detailed design.

**Block Diagram**

Calculate Second Order Moment

Input Training Data Set

In Gray Scale

Train the system using svmtrain()

Create Feature Matrix

Convert Image into Gray Scale

Generate Feature Matrix

Input Target Image

Calculate Second Order Moment

Classify using SVM

Generate discriminant analysis plot after classification

**2.2 Flowchart**

Start

Read the source image and resize it to the size of 50\*50

Convert the set of images into gray scale

Calculate the second order moment for each image in training data set

Generate feature matrix

Read the target image & convert it into gray scale.

Calculate Second Order moment.

Generate feature matrix.

Perform classification using SVM

Generate graph showing classification

Stop

**3. DETAILED TEST PLAN**

Tests are frequently grouped by where they are added in the software development process or by the level of specificity of the test. The main levels in testing that we are going to follow are Unit testing, Integration testing and System testing that is distinguished by the test target without implying a specific process model. Other test levels are classified by testing objective.

**3.1 Unit Testing**

Unit testing also known as component testing refers to the tests that verify the functionality of a specific section of code usually at the function level. These types of tests are usually written by developers as they work on code (white box style), to ensure that the specific function is working as expected. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses works independently of each other. In our project various modules are:

* Gray scale conversion
* Resizing of gray scale image
* Feature extraction
* Object classification



Fig 4: RGB image image

Fig 5: Gray scale image

**3.2 Integration Testing**

Integration testing is any type of software testing that seeks to verify the interfaces between components against the software design. Software components may be integrated in an iterative way or all together. Normally the former is considered a better practice since it allows interface issues to be localized more quickly and fixed. Integration testing works to expose defects in the interfaces and interaction between integrated components (module).

In our project various modules of integration testing are:

* Data acquisition module
* Feature matrix formation
* Training the system
* Classification

**3.3 System Testing**

System testing tests a completely integrated system to verify that it meets its requirements. In addition, the software testing should ensure that the program, as wellas working as expected ,does not also destroy or partially corrupt its operating environment or cause other processes within that environment to become inoperative (this includes not corrupting shared memory ,not consuming or locking up excessive resources and leaving any parallel processes unharmed by its presence).

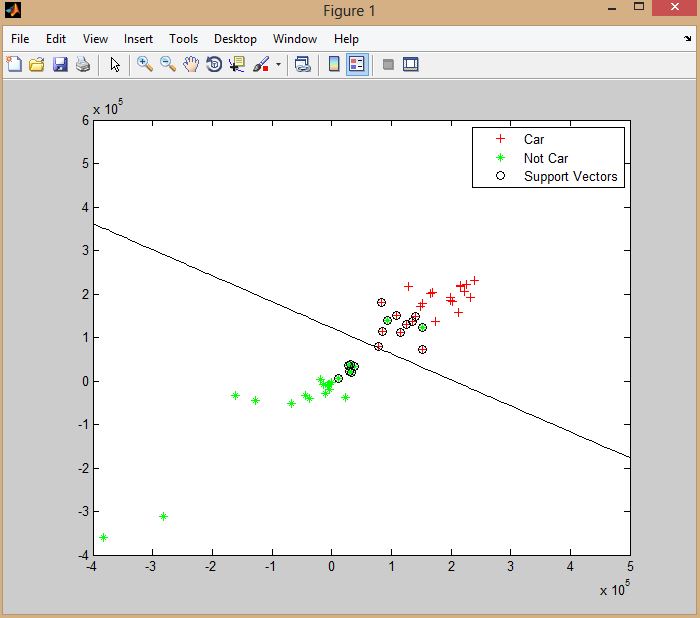
* ****Classification of input images.

Fig 6: Snapshot of final output

**4. IMPLEMENTATION DETAILS**

**Pseudo code for algorithm**

For 1 to size of file

imread(filename)

imresize(input\_im,[50 50])

rgb2gray(input\_im)

Show subplot

subplot(131),

imshow(input\_im)

input\_gray\_hist:=imhist(input\_im), subplot(132);hist(input\_gray\_hist)

input\_im:=double(input\_im), m:=moment(input\_im,3)

m3ent:=entropy(m)

m3ent1:=[m3ent1;m3ent]

m2:=[m2;m]

save('train\_dataset.mat','m3t','m3','m3ent1')

group = cell(50,1)

strArray = java\_array('java.lang.String', 50)

For i:=1 to 50

Convert numbers into string for entropy values

group = cell(strArray)

save('group.mat','group')

loadtrain\_dataset.mat

loadgroup.mat

loadfisheriris

For I : =1 to 50

xdata = m3(1:end,20:21)

group1 = group(1:end)

svmStruct = svmtrain(xdata,group1,'showplot',true)

Show the plot with line separating two classes of objects.

**5. RESULT AND DISCUSSIONS**

**Testing Protocol and Result of Testing**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, coding. The purpose of the testing phase is to integrate the component, programs, subsystem and other interfacing entities into tested, finished system. This may or may not include acceptance testing. It does not include acceptance testing. It does not include installation and checkout.

The software testing phase is initiated when the code has been developed. Software component have been independently verified, and software is ready for testing. This phase ends when the software system components are integrated and verified with the total hardware and software system, the final product are audited and a software product baseline is identified and accepted by customer, user and developers.

During the software testing phase, the software is verified against the build to baseline to determine if the final program properly implements the design. At roughly the same time the software system is tested and verified against the original system and software requirements specification to see if requirements specifications to see if the product was built correctly.

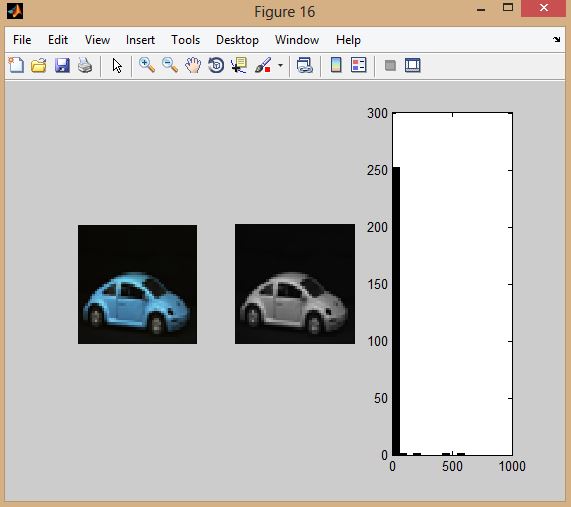
The objective of the testing are:-

* Testing is a process of executing a program with the intent of finding an error.
* A good test case is one that has high probability of finding an undiscovered error.

A successful test is one that undercover an yet undiscovered error.

If testing is conducted successfully, it will uncover errors in the software. As a second benefit, testing demonstrates the software function appears to be working correctly to specification that performance requirements appear to have been met. In addition, data collected as testing is conducted provide a good indication of reliability and some induction of software quality as a whole. But there is one thing that testing cannot do: testing cannot show the absence of defects, it can only show that defects exit in system.

**Snapshots**

Fig 7: Snapshot showing color image, gray scale image and histogram

**Feature matrix of training data set**

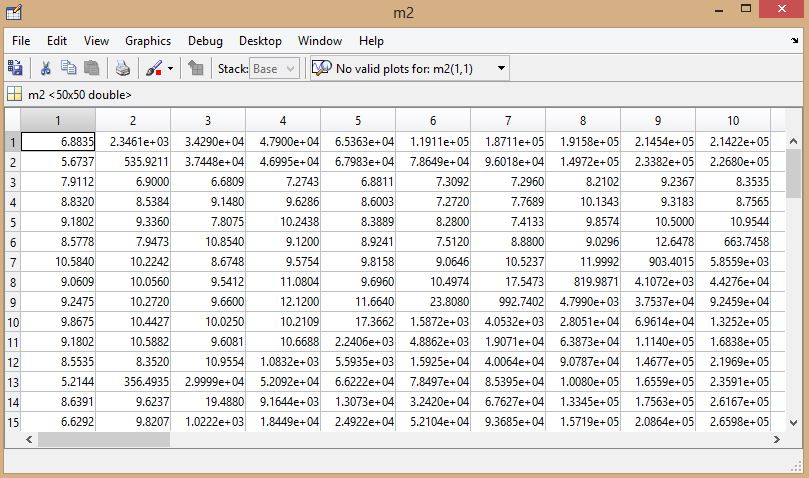
****

Fig 8: Snapshot of feature matrix of training data set

**Snapshot of testing data set**

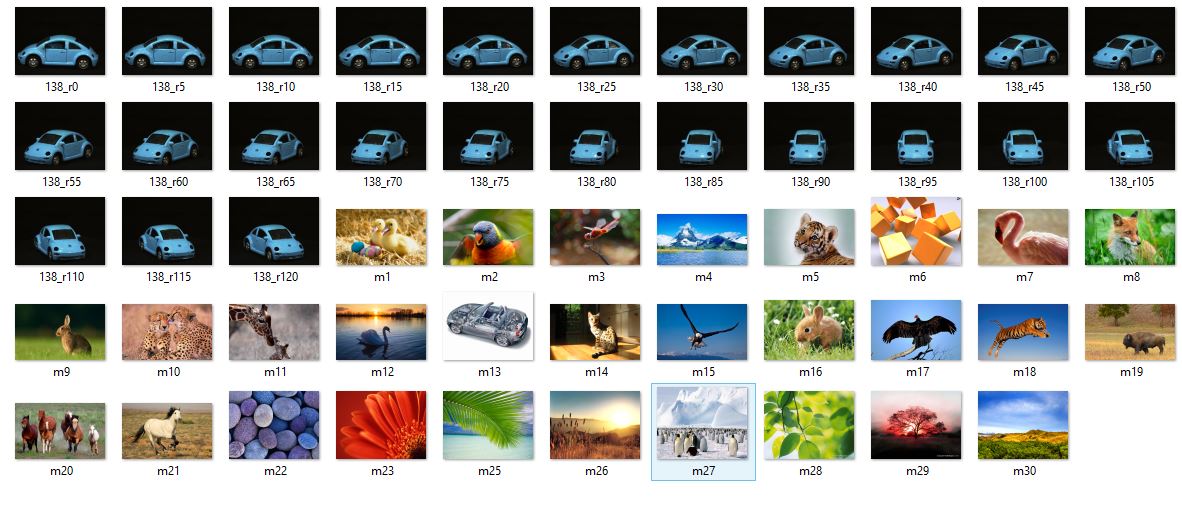
****

Fig 9: Testing data set considering rotation

**Snapshot showing classification**

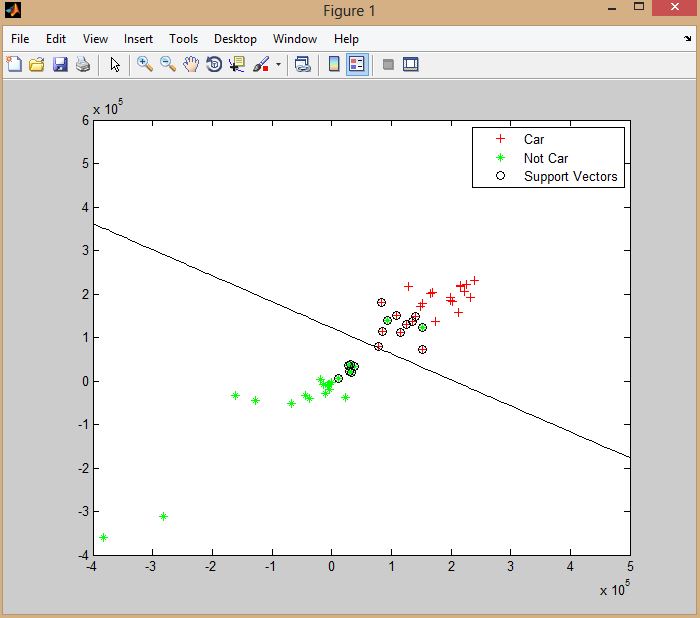
****

Fig 10: Snapshot for classification of car from other objects

**Snapshot of testing data set**

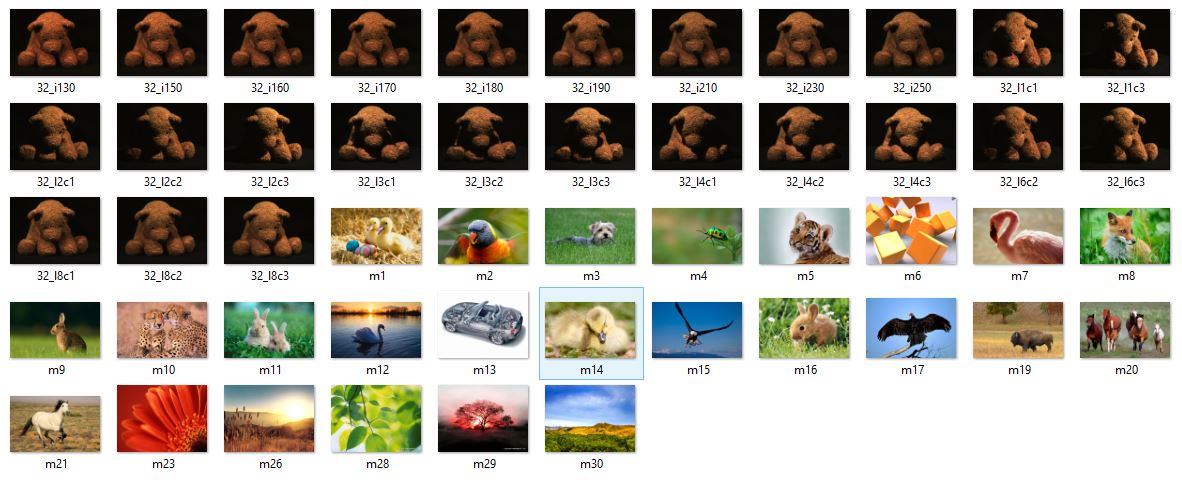
****

Fig 11: Testing data set considering illumination variation

**Snapshot showing classification**

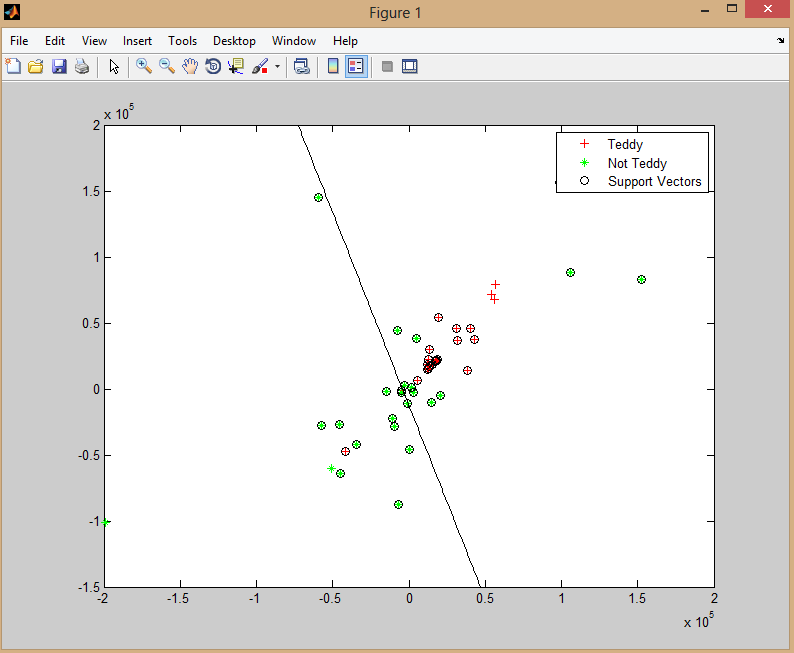


Fig 12: Snapshot for classification of teddy from other objects

**6. SUMMARY AND CONCLUSION**

**6.1 Summary of Achievements**

Digital image processing is being widely used in several fields presently. It has shown a tremendous growth in past three decades. Color image classification has become popular in many applications. Learning about the subject and being able to process the image is not an easy job and is a big achievement to start with. Feature extraction is an important part of object classification. Finally being able to extract a feature that is invariant to image transformations like rotation, translation, affine transformation and illumination variations and determining entropy required a lot of hard work and detailed study. To be able to classify a color images from different classes in the testing data set is the biggest achievement and would definitely be appreciated by a person having good knowledge about the subject.

**6.2 Main difficulties encountered**

While designing the system that can classify objects into different categories, the main difficulties that were encountered during implementation:-

* Selection of proper feature that is invariant to sensory transformation.
* Selection of proper classification technique whether to use SVM or KNN classification.
* Training the system for classification.
* Matching the features of training data set and testing data set.
* Implementation of SVM classification and generating plot with line classifying two classes of object.

**6.3 Limitations of the project**

The drawbacks of SVM classification technique mainly lead to the limitations of our project.

* SVM is a binary classifier. To do a multi-class classification, pair-wise classifications can be used (one class against all others, for all classes).
* The second limitation is speed and size (mostly in training- for large training sets, it typically selects a small number of support vectors, thereby minimizing the computational requirements during testing).
* The optimizing criterion is the width of the margin between the classes (i.e., the empty area around the decision boundary defined by the distance to the nearest training patterns).

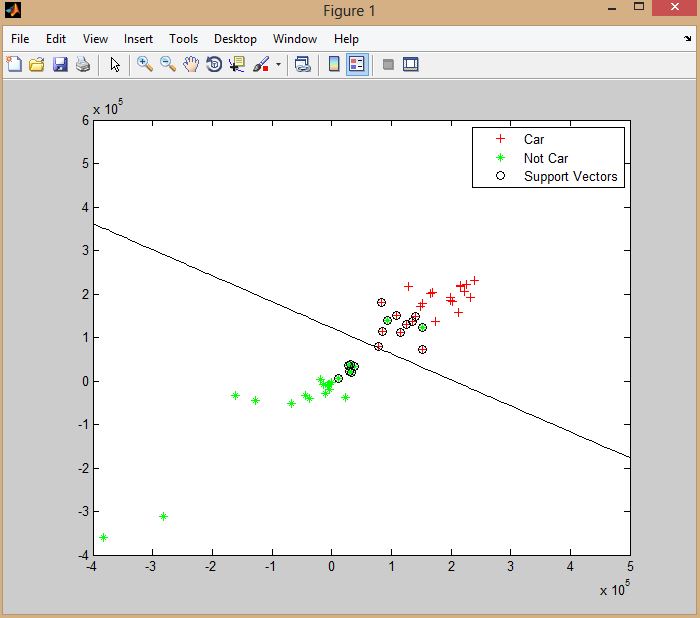
****Although we have taken images of different classes as the input image for testing the system, the system classifies those images into only two categories of objects.

Fig 13: SVM classification of objects

**6.4 Future scope of the work**

The feature extraction technique which has been used can be applied in various following fields:

1. Can be implemented for object identification.

2. Can be implemented for security purpose.

3. Can be implemented in GIS

**7. REFERENCES AND BIBLIOGRAPHY**

[1] Fukunuga, K., Introduction to Statistical Pattern Recognition, 2nd ed., Academic Press, San Diego, CA, 1990.

[2] Jain A K, Fundamentals of Digital Image Processing, Prentice Hall, Englewood Cliffs, NJ 1989.

[3] Levine M D, Vision in Man and Machine, McGraw Hill, Inc., New York, 1985.

[4] Liang-Chi Chiu, Tian-Sheuan Chang, Fast SIFT Design for Real-TimeVisual Feature Extraction, IEEE Transcations on Image Proseccing, V22 N8, August 2013.

[5] Schalkoff R J, Digital Image Processing and Computer Vision, J. Wiley, Inc. NewYork, 1989.

[6] SubhransuMaji, Alexander C. Berg, and Jitendra Malik, Efficient Classification for Additive Kernel SVMs, IEEE Transcations on Pattern Analysis And Machine Intelligence, 2012.

[7] R. E. Woods R. C. Gonzalez, Digital Image Processing, Prentice Hall, 1992.