**DEPTH ESTIMATION OF AN IMAGE**

**A PROJECT REPORT**

***Submitted by***

K.SRAVANI 314126510049

M.DURGA PRASAD 314126510060

B.PAVAN 314126510010

E.UDAY KUMAR 314126510019

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE ENGINEERING**



Under esteemed guidance of

**Mrs.Pritee Parwekar**

(Assistant professor)

**Mrs.G.Santhoshi**

(Assistant Professor)

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**

(Affiliated to Andhra University)

**SANGIVALASA, VISAKHAPATNAM - 531162**

2014-2018

**DEPTH ESTIMATION OF AN IMAGE**

**A PROJECT REPORT**

***Submitted by***

K.SRAVANI 314126510049

M.DURGA PRASAD 314126510060

B.PAVAN 314126510010

E.UDAY KUMAR 314126510019

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE ENGINEERING**



Under esteemed guidance of

**Mrs.Pritee Parwekar**

(Assistant professor)

**Mrs.G.Santhoshi**

(Assistant Professor)

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**

(Affiliated to Andhra University)

**SANGIVALASA, VISAKHAPATNAM - 531162**

2014-2018

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**

**(Affiliated to Andhra University)**

**SANGIVALASA, VISAKHAPATNAM-531162**

**BONAFIDE CERTIFICATE**

Certified that this project report “**DEPTH ESTIMATION OF AN IMAGE”** is the bonafide work of “K.SRAVANI(314126510049),M.DURGA PRASAD(314126510060),B.PAVAN(314126510010),E.UDAYKUMAR(314126510019)“ who carried out the project work under my supervision.

Dr.R.Sivaranjani Mrs.Pritee Parwekar

**HEAD OF THE DEPARTMENT** PROJECT GUIDE

Professor (Assistant Professor)

Department of Computer Science Mrs.G.Santhoshi

& Engineering PROJECT CO-GUIDE

ANITS (Assistant Professor)

Department of computer

Science & Engineering

ANITS

**DECLARATION**

This is to certify that the project work entitled “**DEPTH ESTIMATION OF AN IMAGE**” is a bonafide work carried out by **K.SRAVANI, M.DURGA PRASAD, B.PAVAN, E.UDAY KUMAR** as a part of **B.TECH** final year 2nd semester of **computer science Engineering** of Andhra University, Visakhapatnam during the year 2014-18.

We, **K.SRAVANI, M.DURGA PRASAD, B.PAVAN, E.UDAY KUMAR** of final semester B.Tech., in the department of Computer Science Engineering from ANITS, Visakhapatnam, hereby declare that the project work entitled  **DEPTH ESTIMATION OF AN IMAGE** is carried out by us and submitted in partial fulfillment of the requirements for the award of **Bachelor of Technology in Computer Science Engineering** , under Anil Neerukonda Institute of Technology & Sciences during the academic year 2014-18 and has not been submitted to any other university for the award of any kind of degree.

K. SRAVANI 314126510049

M.DURGA PRASAD 314126510060

B. PAVAN 314126510010

E. UDAY KUMAR 314126510019

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement always boosted the morale. We take a great pleasure in presenting a project, which is the result of a studied blend of both research and knowledge.

We first take the privilege to thank the Head of our Department, **Dr.R.Sivaranjani**, for permitting us in laying the first stone of success and providing the lab facilities, we would also like to thank the other staff in our department and lab assistants who directly or indirectly helped us in successful completion of the project.

We feel great to thank **Mrs.Pritee Parwekar**, **Mrs.G.Santhoshi** ,who are our project guides and who shared their valuable knowledge with us and made us understand the real essence of the topic and created interest in us to work day and night for the project; we also thank our B.Tech coordinator **Mrs. T.Kranthi&&Mr.T.Suresh**, for her support and encouragement.

We also like to thank **Mr.P.Sainath Vital** for providing us the lab resources; we also thank our friends and college staff who extended their part of support in the successful completion of the project.

K.SRAVANI 314126510049

M.DURGA PRASAD 314126510060

B.PAVAN 314126510010

E.UDAY KUMAR 314126510019

**CONTENTS**

**TITLE Page No.**

Abstract 1

Keywords 1

**1. Introduction 2**

1.1 Problem statement 2

1.2 Motivation 2

1.3 Contribution 2

1.4 Research Methodology 3

**2. Literature Survey 4**

2.1 Introduction to image processing 4

2.2 Types of Images 4

2.3 Pre-Processing techniques 6

2.4 Feature extraction 9

2.4.1 Global features 9

2.4.2 Fast Fourier transform 9

2.4.3 Local features 10

2.4.4 Windowed Fourier transform 10

2.5 Principal component Analysis 11

2.6 Support vector machine 13

2.7 Random Forest 16

2.8 Supervised learning 16

2.9 Unsupervised learning 17

**3. System Software Requirements 18**

3.1 Software Requirements 18

3.2 Hardware Requirements 18

3.2.1 Hardware Interface 18

3.2.2 Software Interface 18

**4. Software Methodology 19**

4.1 Introduction to python 19

4.2 Advantages of python 19

4.3 Disadvantages of python 20

**5. Proposed System 21**

5.1 Architecture 21

5.2 Proposed Algorithm 22

**6. Design 23**

6.1 UML Diagrams 23

6.1.1 Sequence Diagram 23

6.1.2 State chart Diagram 24

**7. Implementation 25**

**8.Results 37**

**9. Testing 39**

**10. Conclusion 44**

**References 45**

**ABSTRACT**

The depth of an image is used to find out the distance from where it is taken. This is done by taking a set of images with different depths as the input and extracting Local and Global features from it. First feature extraction based methods are utilized to evaluate image similarities. We propose a model that incorporates both of them to obtain significantly more accurate depth estimates than using either global or local properties alone. The features thus obtained are tested using support vector machines. The trained data is then used for testing to calculate the depth of any image approximately.

**KEYWORDS:**

Depth estimation, local features, global features, PCA, SVM, Random forest.

**1 INTRODUCTION**

**1.1 PROBLEM STATEMENT**

In the absence of cues for absolute depth measurements as binocular disparity, motion, or defocus, the absolute distance between the observer and the scene cannot be measured. One possible source of information for absolute depth estimation is the size of known objects. However, object recognition, under unconstrained conditions, remains difficult and unreliable for current computational approaches.

**1.2 MOTIVATION**

As automation and artificial intelligence become increasingly popular, so does the need for accurate computer vision and scene rendering. One of the most important aspects of computer vision is depth estimation. This problem has broad applications in areas such as robot vision, human computer interfaces, intelligent visual surveillance, 3D image acquisition, and intelligent driver assistance systems. Depth estimation can also be used for artistic purposes, such as image refocus, simulations, and special effects.

**1.3 CONTRIBUTION**

In our work, we use python for our programs. The purpose of this project is to estimate or approximately find the distance between the person and the scene. We tested on many images that are taken as input to check whether we get correct estimated results or not.

**1.4 RESEARCH METHODOLOGY**

Depth Estimation is most interesting research now-a-days because it has wide applications in areas such as robot vision, human computer interfaces and so on. These have many applications and according to our application we change our way of approach to solve a problem. This research mainly focuses on efficient way to estimate the depth of an image using image processing.

**2. LITERATURE SURVEY**

**2.1 INTRODUCTION TO IMAGE PROCESSING**

Image processing is a method to convert an image into digital form and perform some operations on it, in order to get an enhanced image or to extract some useful information from it. It is a type of signal dispensation in which input is image, like video frame or photograph and output may be image or characteristics associated with that image. Usually Image Processing system includes treating images as two-dimensional signals while applying already set signal processing methods to them. Image Processing form s core research area within engineering and computer science disciplines too.

The two types of methods for Image Processing are Analog and Digital Image Processing. The image processing is not just confined to area that has to be studied but on knowledge of analyst. Digital Processing techniques help in manipulation of the digital images by using computers. As raw data from imaging sensors from satellite platform contains deficiencies. To get over such flaws and to get originality of information, it has to undergo various phases of processing. The three general phases that all types of data have to undergo while using digital technique are Pre- processing, enhancement and display, information extraction.

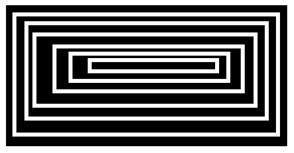
**2.2 TYPES OF IMAGES**

**2.2.1 BINARY IMAGE**

The binary image as it name states, contain only two pixel values 0 and 1. Here 0 refers to black colour and 1 refers to white colour. It is also known as Monochrome. Binary images have a format of PBM (Portable bit map).

**2.2.2 BLACK AND WHITE IMAGE**

The resulting image that is formed hence consist of only black and white colour and thus can also be called as Black and White image.



One of the interesting this about this binary image that there is no gray level in it. Only two colours that are black and white are found in it.

**2.2.3 2, 3, 4, 5, 6 BIT COLOR FORMAT**

The images with a colour format of 2, 3, 4, 5 and 6 bit are not widely used today. They were used in old times for old TV displays or monitor displays. But each of these colours have more than two gray levels, and hence has gray colour unlike the binary image. In a 2 bit 4, in a 3 bit 8, in a 4 bit 16, in a 5 bit 32, in a 6 bit 64 different colours are present.

**2.2.4 8 BIT COLOR FORMAT**

8-bit colour format is one of the most famous image format. It has 256 different shades of colours in it. It is commonly known as Grayscale image. The range of the colours in 8 bit vary from 0-255. Where 0 stands for black, and 255 stands for white, and 127 stands for gray colour.

**2.2.5 24 BIT FORMAT**

24-bit colour format also known as true colour format. Like 16-bit colour format, in a 24-bit colour format, the 24 bits are again distributed in three different formats of Red, Green and Blue.



Since 24 is equally divided on 8, so it has been distributed equally between three different colour channels. Their distribution is like this. 8 bits for R, 8 bits for G, 8 bits for B.

**2.3 PRE-PROCESSING TECHNIQUES**

Image pre-processing is always an important and interesting field as it gives improved pictorial information for human interpretation and processing of image data for storage, transmission and representation of machine perception. Image Pre-Processing is a technique to enhance raw images received from cameras/sensors placed on satellites, space probes and aircrafts or pictures taken in normal day-to-day life for various applications. Pre-Processing is a common name for operations with images at the lowest level of abstraction both input and output are intensity images. The aim of pre-processing is an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing. Some of the pre-processing are described below.

**2.3.1 Converting RGB colored image to grey image:**

After resizing of the image, the colored image will be converted into grayscale so that the red, green, blue will be converted into a single grey value so that we can totally reduce the computations. These all 3 values can be converted into single gray value using different methods like Average Method, Weighted Method. For Average method we will simply add the RGB values and then average it grayscale=r+g+b/3. Similarly for Weighted method we do grayscale= (0.3\*r)+(0.59\*g)+(0.11\*b).



**2.3.2 Sharpening of an image:**

Sharpening of an image involves taking the image as an input and sharpen the image so that to increase the edge content. So, in order to increase edge content in an image, we have to find edges first. Edges can be found by using any operator. Prewitt order, Sobel operator, Robinson Compass mask, Laplacian operator. Here in our project of pre-processing part we will apply negative Laplacian operator, convolving the input image with the kernel of size 3\*3 matrix

**2.3.2.1 Laplacian operator**

Laplacian operator is also a derivative operator which is used to find edges in an image difference between Laplacian and other operators like prewitt, sobel, Robinson and kirsch is that these all are first order derivative masks but Laplacian is a second derivative mask. In this mask we have two further classification Positive and Negative Laplacian Operator. Laplacian takes out the edges in two formats inward and Outward edges.

Sample Kernel Mask for Positive Laplacian filter is:

|  |  |  |
| --- | --- | --- |
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

This positive Laplacian operator is used to take out outward edges in an image whereas negative Laplacian operator is used to take out inward edges in an image. To get sharpen image: if we use positive mask then we subtract with the original image and similarly vice versa if we use negative mask.

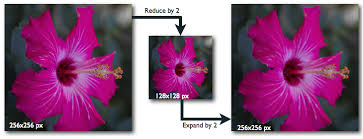
Sample Kernel Mask for Negative Laplacian Operator:

|  |  |  |
| --- | --- | --- |
| 0 | -1 | 0 |
| -1 | 4 | -1 |
| 0 | -1 | 0 |



**2.3.3 Resizing of an image:**

Image Resizing is defined as the image will be reduced to some width and height value and it will be of less value when compared to original width and height of the image. This can be done using resize function present in any programming language for example if an image of size 1024\*496 pixels then this can be reduced to any size of user interest and this reduction will help in reduce computations.



**2.4 FEATURE EXTRACTION**

Feature are the information or list of numbers that are extracted from an image. These are real valued numbers (integers, float or binary). There are a wider range of feature extraction algorithms in open computer vision. Features are the information extracted from the images, suppose we consider the image as data the information extracted from the data is known as features. Based on our application i.e., depth estimation of an image we have chosen two features first one is global feature extraction and second one is local feature extraction of an image.

**2.4.1 Global Feature Extraction**

Global features are generally used in object detection and classification, detection is finding of the object (whether an object exists in image). Global feature means looking the whole image i.e., overall information present in an image. It is also known as global feature descriptors. In our project for extracting the global features we use fast Fourier Transform.

**2.4.2 Fast Fourier Transform**

A Fast Fourier Transform is an algorithm that samples a signal over a period of time or space and divides it into its frequency components. These components are single sinusoidal oscillations at distinct frequencies each with their own amplitude and phase. Fourier analysis converts a signal from its original domain to a representation in the frequency domain. Fast Fourier transform manages to reduce the complexity of computing the Fourier transform from O(n^2) to O(nlogn) where n is the data size. Fast Fourier transform are widely used for many applications in engineering, science and mathematics. The fast Fourier transform is much faster when compared to ft but produces the same result as of discrete Fourier transform. The command or method for fft in python is fft.fft(image) with prerequisite NumPy. In our project the obtained matrix form after applying the command or method or function consists of the combination of amplitude and phase values, from this we take only amplitude values using the function in python abs (matrix form) (matrix form obtained from statement 4 mentioned command) and prerequisite is NumPy.

**2.4.3 Local Feature Extraction**

Local features are generally used for object recognition or identification. Recognition is finding of the object in other words focusing on something. Local features mean looking at particular part of the image and getting the details of that image at that particular area. These are also known as local descriptors. For finding out the local features for an input image, here we use windowed Fourier transform.

**2.4.4 Windowed Fourier Transform**

The windowed Fourier transform is generally defining as the replacement of the Fourier transforms sinusoidal wave by the product of a sinusoidal and a window which is localized in time. In our project we divide the entire image, here image taken as input is the resultant image obtained from the pre-processing technique. The image is then divided into 10\*10 locations, each location is of size 25\*25 i.e., width and height of the one location in that image. For each location we apply Fast Fourier Transform and get the absolute values of the resultant matrix for all 10\*10 locations of that image. In general, the local features describe every part of the image, so when we apply Windowed Fourier Transform the details present in the image are more because the data obtained from that image is more, which indirectly states the large amount of data.

**2.5 PCA (PRINCIPAL COMPONENT ANALYSIS)**

The main purpose of a principal component analysis is the analysis of data to identify patterns and finding patterns to reduce the dimensions of the dataset with minimal loss of information. Principle component analysis is also known as dimensionality reduction algorithm which reduces n dimensional to some d dimensional form for example if n=3 then d can be of any value either 2 or 1. Here in our project after feature extraction we reduce the output data which is in the form of 2 dimensional is converted into 1-Dimensional form by applying PCA algorithm. The PCA is mostly used so that we can reduce the computation part for our image. The PCA algorithm follows the following steps.

**Step 1: Taking the whole dataset**

a) Take the whole dataset i.e., feature values obtained from feature extraction part, consisting of n-dimensional (n=2) samples ignoring the class labels.

Note: Here the value of n is 2 then the resultant of PCA will be in the form of 1-D matrix.

**Step 2: Compute n-Dimensional mean vector**

a) Compute the n-Dimensional mean vector for the whole dataset taken in step1.

b) The function or command used for calculating mean vector in python is mean (matrix, axis=0), the prerequisite is NumPy.

**Step 3: Computing Scatter Matrix**

a) Compute the scatter matrix (alternatively, the covariance matrix) of the whole dataset using mean vector obtained from step 2.

b) The code used to calculate the covariance matrix in python is

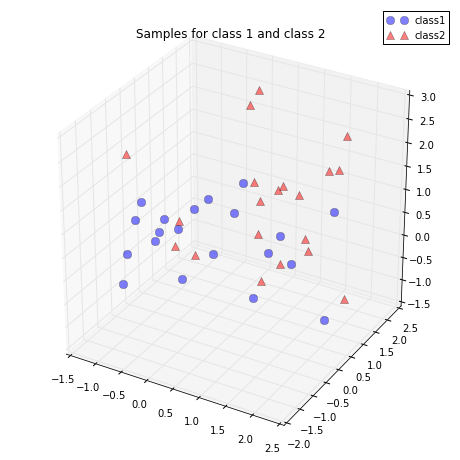
cov\_mat = numpy.cov(X.T)

cov\_mat=(matrix-mean\_vec).T.dot((matrix-mean\_vec))/(matrix.shape[0]-1)

**Step 4: Computing eigen vectors and eigen values**

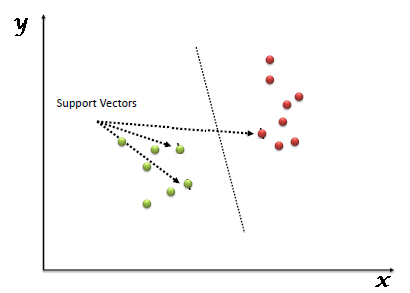
a) After finding covariance matrix we will compute the eigen vectors and corresponding eigen values.

b) These eigen vectors are also called as principal components, and finally the eigen values are the resultant output of the matrix which is of the form 1-D. Using python coding if we want we can plot the values to see the dataset. For example, the below figures will show the dimensionality reduction.



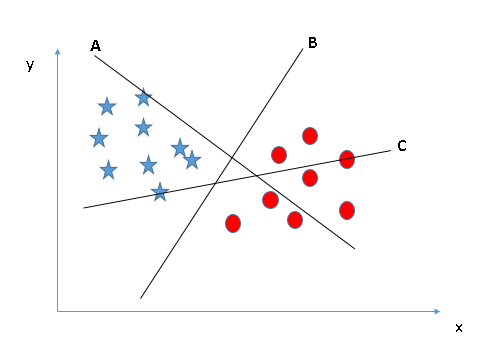
**2.6 SVM (SUPPORT VECTOR MACHINES)**

Support Vector Machine (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot). Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).



The main point in support vector machines is that process of segregating the two classes with a hyper-plane, here we should choose the right hyperplane, for that we have to follow following procedure. Following are some scenarios in which the right hyperplane is chosen.

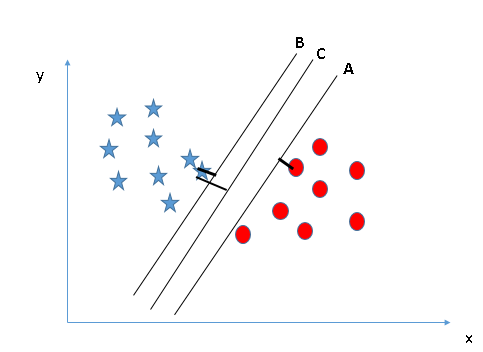
Scenario 1: Identify the hyper plane

Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.  
  


You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.

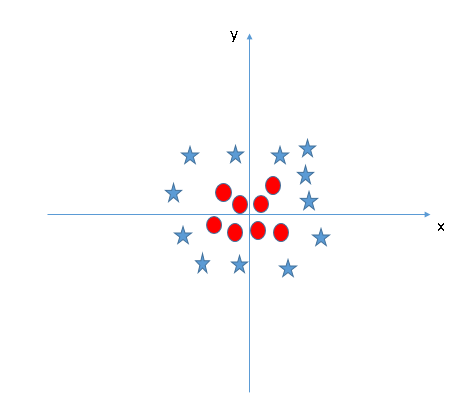
Scenario 2: Identify the hyper plane

Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as margin. Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.



Scenario 3:

In the scenario below, we can’t have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.



SVM has a technique called the kernel trick. These are functions which takes low dimensional input space and transform it to a higher dimensional space i.e. it converts not separable problem to separable problem, these functions are called kernels. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then find out the process to separate the data based on the labels or outputs you’ve defined.

Prons and cons of support vector machine:

Prons:

It works very well in very clear margin of separation. It is effective in high dimensional spaces. It uses a subset of training points in decision function called support vectors so it is called memory efficient.

Cons:

It doesn’t perform well when we have large dataset because training time is higher. It does not perform very well if there is large noise i.e., target classes are overlapping.

**2.7 Random Forest**

Random Forest is an ensemble learning (both classification and regression) technique. It is one of the commonly used machine learning technique. It that it operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. Ranger is a fast implementation of random forests. In a random forest algorithm number of decision trees are built during the process. A vote from each of the decision trees is considered in deciding the final class of a case or an object, this is called ensemble process. Since, many decision trees are built and used in a process of Random Forest algorithm, it is called a forest. For a building a decision tree, samples of a data frame are selected with replacement along with selecting a subset of variables for each of the decision tree. Both sampling of data frame and selection of subset of the variables are done randomly, so first word “random” is arrived.

Key advantages of using Random Forest:

Reduce chances of over-fitting

Higher model performance or accuracy

**2.8 Supervised Learning**

Supervised machine learning systems provide the learning algorithms with known quantities to support future judgments. Self-driving cars, face recognition programs and robots are among the systems that may use either supervised or Unsupervised learning. Supervised learning systems are mostly associated with retrieval-based AI but they may also be capable of using a generative learning model. Training data for supervised learning includes a set of examples with paired input subjects and desired output (which is also referred to as the supervisory signal). In supervised learning for image processing, for example, an AI system might be provided with labelled pictures of vehicles in categories such as cars and trucks. After a sufficient amount of observation, the system should be able to distinguish between and categorize unlabeled images, at which time training can be said to be complete.

**2.9 Unsupervised Learning**

In unsupervised learning, an AI system is presented with unlabelled and uncategorised data and the system’s algorithms act on the data without prior training. The output is dependent upon the coded algorithms. Subjecting a system to unsupervised learning is one way of testing AI. Unsupervised learning algorithms can perform more complex processing tasks than supervised learning systems. However, unsupervised learning can be more unpredictable than the alternate model. While an unsupervised learning AI system might, for example, figure out on its own how to sort cats from dogs, it might also add unforeseen and undesired categories to deal with unusual breeds, creating clutter instead of order. Expectation-maximization algorithm is one of the unsupervised algorithm.

**3. SYSTEM REQUIREMENT SPECIFICATION**

**3.1 SOFTWARE USED IN THIS PROJECT**

**Language:** Python Programming

**Python version:** Python 3.6 32 bit

**Operating System:** Windows

**3.2 HARDWARES USED IN THIS PROJECT:**

**PROCESSOR**: intel Multi Core processor

**RAM**: 4 GB or above

**HARDDISK**:500 GB or above

**3.2.1 HARDWARE INTERFACE:**

MONITOR: The outputs are displayed on the monitor screen.

**3.2.2 SOFTWARE INTERFACE:**

In PYTHON

We have a console python ide where the outputs are displayed. The graphs and plots are displayed in the plots field. We used notepad++ where the code is written and used .csv files to store the data.

**4. SOFTWARE METHODOLOGIES**

**4.1 INTRODUCTION TO PYTHON**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a programming language that lets you work quickly and integrate systems more efficiently. There are two major Python versions- python 2 and 3. Both are quite different.Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP. You can actually sit at a Python prompt and interact with the interpreter directly to write your programs. Python supports Object-Oriented style or technique of programming that encapsulates code within objects. Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**4.2 ADVANTAGES OF PYTHON:**

The Python language has diversified application in the software development companies such as in gaming, web frameworks and applications, language development, prototyping, graphic design applications, etc. This provides the language a higher plethora over other programming languages used in the industry. One of its advantage is.

It provides large standard libraries that include the areas like string operations, Internet, web service tools, operating system interfaces and protocols. Most of the highly used programming tasks are already scripted into it that limits the length of the codes to be written in Python

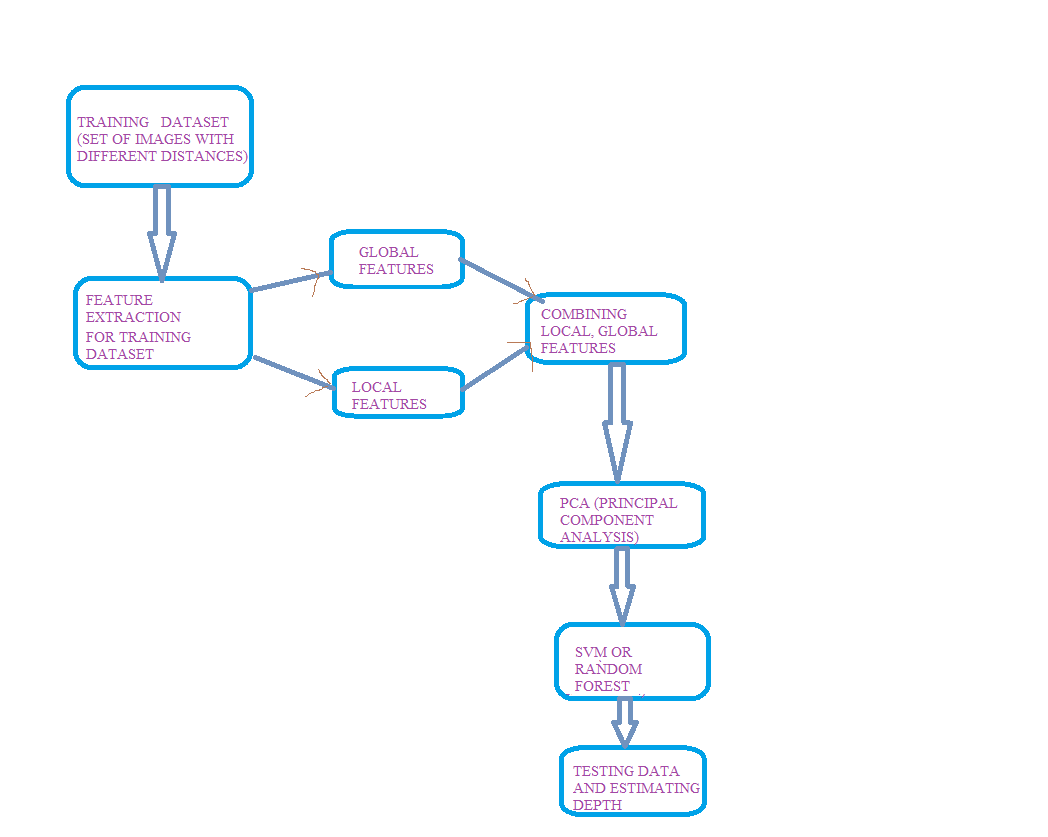
**4.3 DISADVANTAGES OF PYTHON:**

The Python lovers become so accustomed to its features and its extensive libraries, so they face problem in learning or working on other programming languages. Python experts may see the declaring of cast “values” or variable “types”, syntactic requirements of adding curly braces or semi colons as an onerous task.

Python executes with the help of an interpreter instead of the compiler, which causes it to slow down because compilation and execution help it to work normally. On the other hand, it can be seen that it is fast for many web applications too.

**5. PROPOSED SYSTEM**

**5.1 ARCHITECTURE**

****

**5.2 PROPOSED ALGORITHM STEPS**

Our project depends on the different applications and the approach to project changes depending on the application chosen, the different applications can be when travelling on the road the car should automatically stop if the distance between the car and person is less, clearly this can be done using depth estimation. In our project the existing system in our base paper is finding features and then using probabilistic model one of the clustering algorithm.

Proposed System uses the application that it calculates the distance between person and top of the building.

**5.2.1 Algorithm Illustration:**

1. First the training set is taken with different known distances and are kept into four different folders basing upon their height, our training data set are restricted to the buildings.

2. After taking training dataset we will get do pre-processing techniques which include converting RGB image to grayscale, sharpening of image, and finally resizing of the images this will in whole results in best computation results which will increase accuracy.

3. Features are extracted here in our project we extract global and local features for global features we use fast Fourier transform and for local features we use windowed Fourier transform, these all are done for all images present in all folders.

4. After extracting features we apply PCA (principle component analysis) so that to reduce the dimension i.e., the result of step3 will be 2-D and this will have reduced to 1-D using PCA.

4. After extracting the features and applying PCA we will combine these features for each image and finally get a data for all folders and we will store them in .csv file and also their classes are stored in separate folder.

5. Collect the test dataset and store it in testing folder.

6. Using this training dataset storage we test the images present in testing folder using SVM support vector machines algorithm or random forest by fitting them using the fit function present in python and repeat the steps 3,4,5 except classes value and then predict the exact class representing the depth that it can fall in any of the four folders and calculate the result approximately.

**6. SYSTEM DESIGN**

**6.1 UML DIAGRAMS**

The uml is a language. It provides vocabulary and the results for combining words in that vocabulary for the purpose of communication.(Ref[7]) A modelling language is language whose vocabulary and rules flows on the conceptual and physical representation of a system. A modelling language such as uml is a standard language for software blue prints. The uml is a language for visualizing, specifying, constructing and documenting. The software intensive articrafts of a system. UML diagram are classified into two categories:

1. Structural or static

2. Dynamic or behavioral

Structural Model contains Classes, object, use case, component and deployment

Behavioral Model contains: Collaboration, State chart and activity.

**6.1.1 Sequential Diagram**

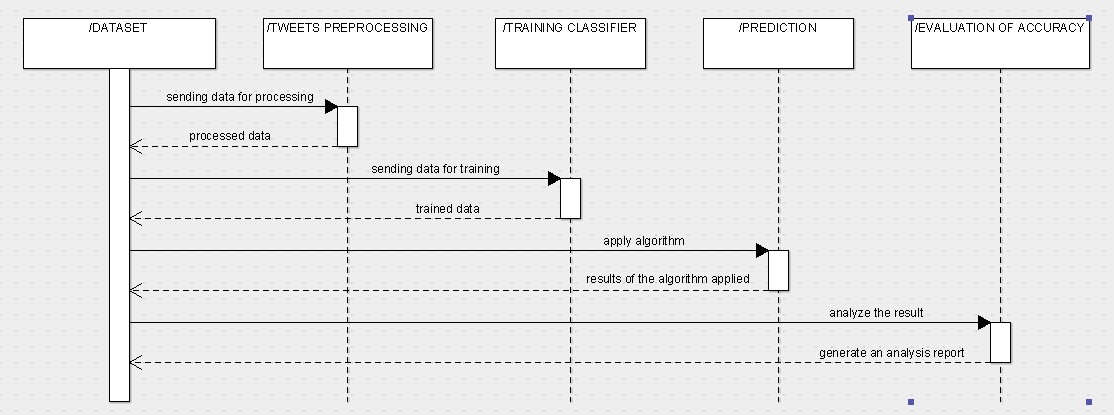
Interaction diagram is called is sequence diagram. Interaction diagram describes patterns of communication among a set of interaction objects. An object interacts with another object by sending messages, arguments may be passed along with a message and they are found to be parameters of executing methods in the receiving objects.

In this Diagram, the Objects are:

1.Training set

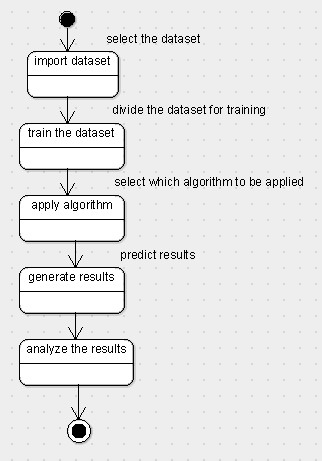
2.Feature Extracter

3.Testing using classifier



**6.1.2 State Chart Diagram:**

It describes the dynamic nature of the diagram how the objects changes its state. It has various States and Transitions.



**7. IMPLEMENTATION**

**7.1 SAMPLE CODE:**

**Training data set of images:**

import csv

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import MinMaxScaler

import os

import numpy as np

from matplotlib import pyplot as plt

import scipy.misc

from PIL import Image

from os import listdir

from os.path import isfile, join

import glob

import numpy

import cv2

import h5py

fixed\_size=tuple((250,250))

test\_size=0.10

train\_path="dataset/training folder images"

train\_labels=os.listdir(train\_path)

features=[]

features1=[]

features2=[]

labels=[]

i,j,e,=0,0,0

seed=9

images\_per\_class=25

def featured\_global(image, mask=None):

print('entered global')

fg= np.fft.fft2(image)

fg=np.abs(fg)

X=np.array(fg)

mean\_vec=np.mean(X,axis=0)

cov\_mat = np.cov(X.T)

cov\_mat=(X-mean\_vec).T.dot((X-mean\_vec))/(X.shape[0]-1)

eig\_vals1, eig\_vecs1 = np.linalg.eig(cov\_mat)

eig\_vals1=np.abs(eig\_vals1)

print('end global')

return eig\_vals1

def featured\_local(image, mask=None):

print('entered local')

a,b,m=0,0,0

p,q=0,0

fg= [[0 for x in range(10)] for y in range(10)]

M\_selection= [[0 for x in range(10)] for y in range(10)]

eig\_vals=[0 for x in range(100)]

eig\_vecs=[0 for x in range(100)]

while p<250 or q<250:

i=p

j=i+25

k=q

l=k+25

M\_selection[a][b] = image[i:j, k:l]

fg[a][b]= np.fft.fft2(M\_selection[a][b])

fg[a][b]=np.abs(fg[a][b])

X=np.array(fg[a][b])

mean\_vec=np.mean(X,axis=0)

cov\_mat = np.cov(X.T)

cov\_mat=(X-mean\_vec).T.dot((X-mean\_vec))/(X.shape[0]-1)

eig\_vals[m], eig\_vecs[m] = np.linalg.eig(cov\_mat)

eig\_vals[m]=np.abs(eig\_vals[m])

m=m+1

q=q+25

b=b+1

if l==250 and p!=250:

if l==250 and p!=225:

q=0

b=0

p=p+25

a=a+1

pcavalues=eig\_vals[0]

o=1

while o<100:

pcavalues=np.append(pcavalues,eig\_vals[o])

o=o+1

print('end local')

return pcavalues

yy=0

fea=[0 for x in range(100)]

for training\_name in train\_labels:

dir=os.path.join(train\_path,training\_name)

current\_label=training\_name

e=1

for xx in range(1,images\_per\_class+1):

file=dir+"/"+str(xx)+".jpg"

print('file',file)

image=cv2.imread(file)

image=cv2.resize(image, fixed\_size)

image = cv2.cvtColor(image,cv2.COLOR\_BGR2GRAY)

kernel = np.zeros( (9,9), np.float32)

kernel[4,4] = 2.0 #Identity, times two!

boxFilter = np.ones( (9,9), np.float32) / 81.0

kernel = kernel - boxFilter

image = cv2.filter2D(image, -1, kernel)

gl\_fea=featured\_global(image)

lo\_fea=featured\_local(image)

all\_features=np.hstack([gl\_fea,lo\_fea])

labels.append(current\_label)

fea[yy]=all\_features

yy=yy+1

features.append(all\_features)

if yy>25:

features1.append(all\_features)

if yy>50:

features2.append(all\_features)

i+=1

e+=1

print("[status] processed folder:{}",format(current\_label))

j+=1

print("[status] completed all features extraction")

csvfile = "output 4classes/data\_csv.csv"

with open(csvfile, "w") as output1:

writer = csv.writer(output1, lineterminator='\n')

writer.writerows(features)

csvfile="output 4classes/labels\_csv.csv"

with open(csvfile, "w") as output1:

writer = csv.writer(output1, lineterminator='\n')

writer.writerows(labels)

**Testing the set of images:**

import csv

import numpy as np

import h5py

import matplotlib.pyplot as plt

import warnings

import os

import glob

import cv2

from sklearn.model\_selection import train\_test\_split

from sklearn import svm

from sklearn.ensemble import RandomForestClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn import linear\_model

from sklearn.linear\_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

num\_trees=100

test\_size=0.10

seed=9

fixed\_size=(250,250)

fearow=[0 for x in range(100)]

labrow=[0 for x in range(100)]

meanrow=[0 for x in range(100)]

n=0

rowfeature=[]

with open('output 4classes/data\_csv.csv','r') as File:

reader = csv.reader(File)

for rows in reader:

fearow[n]=np.array(rows)

n=n+1

rowfeature.append(rows)

rowlabel=[]

n=0

with open('output 4classes/labels\_csv.csv','r') as File:

reader = csv.reader(File)

for rows in reader:

labrow[n]=np.array(rows)

n=n+1

rowlabel.append(rows)

print('3:top floor 2:mtech floor 1:hod 0:ground')

print('.............SUPPORT VECTOR MACHINES.............')

from sklearn import svm

lin\_clf = svm.SVC()

lin\_clf.fit(np.array(fearow), np.array(labrow).ravel())

clf = RandomForestClassifier()

clf.fit(np.array(fearow), np.array(labrow).ravel())

linear=linear\_model.LinearRegression()

linear.fit(np.array(fearow), np.array(labrow).ravel())

knn = KNeighborsClassifier()

knn.fit(np.array(fearow), np.array(labrow).ravel())

def featured\_global(image, mask=None):

#print('entered global')

fg= np.fft.fft2(image)

fg=np.abs(fg)

X=np.array(fg)

mean\_vec=np.mean(X,axis=0)

cov\_mat = np.cov(X.T)

cov\_mat=(X-mean\_vec).T.dot((X-mean\_vec))/(X.shape[0]-1)

eig\_vals1, eig\_vecs1 = np.linalg.eig(cov\_mat)

eig\_vals1=np.abs(eig\_vals1)

#print('end global')

return eig\_vals1

def featured\_local(image, mask=None):

#print('entered local')

a,b,m=0,0,0

p,q=0,0

fg= [[0 for x in range(10)] for y in range(10)]

M\_selection= [[0 for x in range(10)] for y in range(10)]

eig\_vals=[0 for x in range(500)]

eig\_vecs=[0 for x in range(500)]

while p<250 or q<250:

i=p

j=i+25

k=q

l=k+25

M\_selection[a][b] = image[i:j, k:l]

fg[a][b]= np.fft.fft2(M\_selection[a][b])

fg[a][b]=np.abs(fg[a][b])

X=np.array(fg[a][b])

mean\_vec=np.mean(X,axis=0)

cov\_mat = np.cov(X.T)

cov\_mat=(X-mean\_vec).T.dot((X-mean\_vec))/(X.shape[0]-1)

eig\_vals[m], eig\_vecs[m] = np.linalg.eig(cov\_mat)

eig\_vals[m]=np.abs(eig\_vals[m])

m=m+1

q=q+25

b=b+1

if l==250 and p!=250:

if l==250 and p!=225:

q=0

b=0

p=p+25

a=a+1

pcavalues=eig\_vals[0]

o=1

while o<100:

pcavalues=np.append(pcavalues,eig\_vals[o])

o=o+1

#print('end local')

return pcavalues

testfearow=[0 for x in range(1)]

test\_path = "dataset/testing folder images"

train\_path="dataset/training folder images"

train\_labels=os.listdir(test\_path)

for file in glob.glob(test\_path + "/\*.jpg"):

image=cv2.imread(file)

print('file',file)

image = cv2.cvtColor(image,cv2.COLOR\_BGR2GRAY)

#cv2.imshow('original image',image)

kernel = np.zeros( (9,9), np.float32)

kernel[4,4] = 2.0 #Identity, times two!

boxFilter = np.ones( (9,9), np.float32) / 81.0

kernel = kernel - boxFilter

image = cv2.filter2D(image, -1, kernel)

image=cv2.resize(image, fixed\_size)

print('entered')

#cv2.imshow('image',image)

gl\_fea=featured\_global(image)

lo\_fea=featured\_local(image)

all\_features=np.hstack([gl\_fea,lo\_fea])

predicted\_label = lin\_clf.predict (all\_features.reshape(1,-1))[0]

print('support vector machine label is',predicted\_label)

predicted\_label =clf.predict(all\_features.reshape(1,-1))[0]

print('random forest classifier label is',predicted\_label)

predicted\_label= linear.predict(all\_features.reshape(1,-1))[0]

print('Linear Regression label is:',int(predicted\_label))

predicated\_label = knn.predict(all\_features.reshape(1,-1))[0]

print('knn label is:',int(predicated\_label))

if int(predicted\_label)==0:

print('image is taken from 50feet')

elif int(predicted\_label)==1:

print('image is taken from 40 feet')

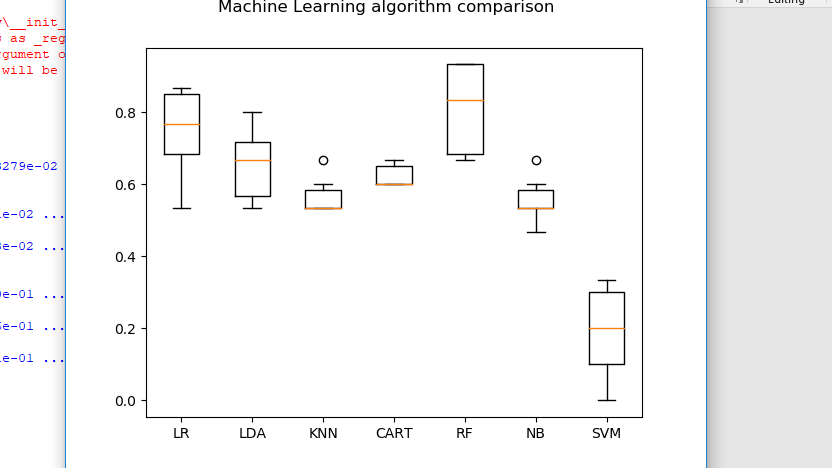
elif int(predicted\_label)==2:

print('image is taken from 30 feet')

elif int(predicted\_label)==3:

print('image is taken from 15 feet')

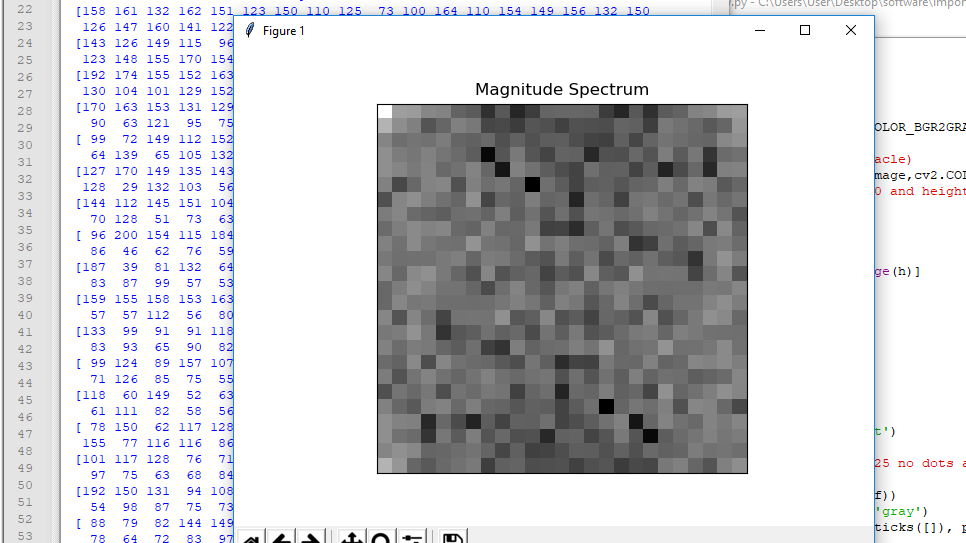
**8. RESULT**

****

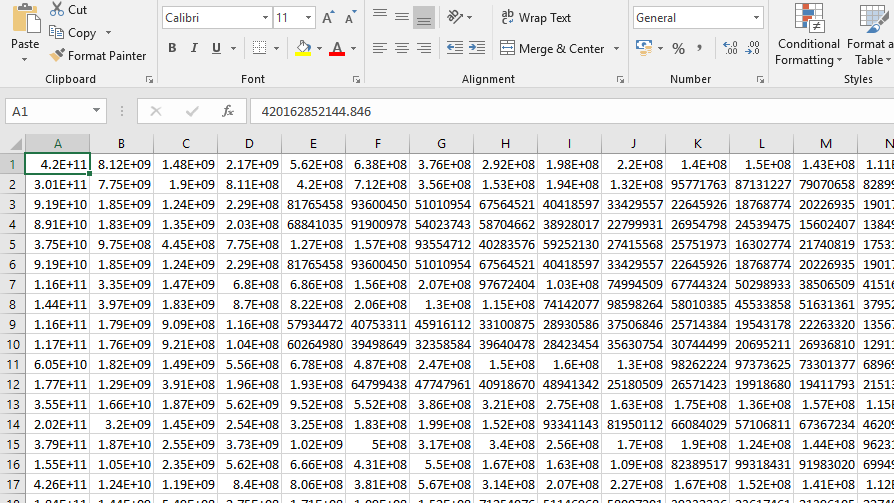
Result of global features:



Result of local features:

****

Features storage: the features stored in files

****

**10. TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and or/a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations does not fail in unacceptable manner. There are various types of test. Each test type addresses a specific requirement.

**TYPES OF TESTS**

**UNIT TESTING**

Unit testing involves the design of test cases that validate the internal program logic is functioning properly, and that program inputs procedure valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion o an individual unit before integration. This is structural testing, that relies on knowledge of its construction and is invasive. Unit test perform basic test at component level and test a specific business process, application, and/or system configuration. Unit test ensures that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**INTEGRATION TESTING**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown successfully by unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**FUNCTIONAL TEST**

Functional tests provide systematic demonstrations that function tests are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: identified classes of valid input must be accepted

Invalid Input: identified classes of invalid input must be rejected

Functions: identified function must be exercised

Output: identified classes of application outputs must be exercised

Procedures: interfacing systems or procedures must be invoked

Organization and preparation of function tests is focused on requirements, key functions or special test cases. In addition, systematic coverage pertaining to identify business process flow; data fields, predefined process, and successive process must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current test is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is configuration-oriented system integration test.

**White Box Testing**

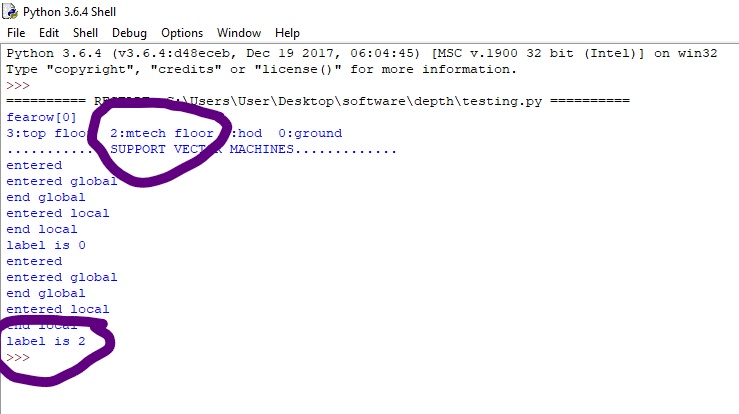
White Box Testing is a testing in which the software tester has and of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kind of tests, must be written from a definitive source document, such as specification or requirements document. It is a testing in which the software under test is treated, as black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.



****





**11. CONCLUSION**

In this, we have estimated depth of an image by performing training of dataset containing images and storing them in files, and the testing the new data and roughly predicting the depth using machine learning algorithm SVM OR Random forest classifier using python programming language and found that random forest classifier gives more accuracy when compared to support vector machines.

**REFERENCES**

1) <https://www.analyticsvidhya.com/>

2) <http://sebastianraschka.com>

3) <https://stackoverflow.com/>

4) <https://www.quora.com/>

5) <http://ieeexplore.ieee.org>

6) <https://docs.python.org/3/>

7) <https://www.tutorialspoint.com/dip/>

**Future Scope:**

We are not able to achieve to, how to find the sky texture (i.e., how to recognize the sky after the immediate end of the building). For this we will train the images at different times from different heights. This will be the future extension of our project.