HMR INSTITUTE OF TECHNOLOGY & MANAGEMENT
HAMIDPUR, DELHI - 110036

Affiliated to
GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY
Sector - 16C Dwarka, Delhi - 110075, India
2008-12

Summer Training Report

on

Geometric shapes on web using ML and Shape detection

Bachelor of Technology

In

Computer Science & Engineering

Project Submitted by:

Rohit Kr Pandey (**416**13302716) 5th Sem
Dhanu (**413**13302716) 5th SEM

HMR INSTITUTE OF TECHNOLOGY &MANAGEMENT

HAMIDPUR, DELHI 110 036

Affiliated to

GURU GOBIND SINGH INDRAPRASTHA

UNIVERSITY Sector - 16C Dwarka, Delhi -

110075, India

2016-2020

INDEX

CERTIFICATE

This is to certify that **Geometric Shape on web using ML and Shape Detection**, is a bonafide record of project done by **ABOVE MENTIONED STUDENTS** under my supervision and submitted to Guru Gobind Singh Indraprastha University, Dwarka (New Delhi) in partial fulfilment for the award of the Degree of Bachelor of Technology.

Signature of the supervisor

DECLARATION

I/We, student(s) of Bachelors of Technology (branch) hereby declare that the summer training project entitled " Geometric Shape on web using ML and Shape Detection" which is submitted to Department of "COMPUTER SCIENCE AND ENGINEERING" , HMR Institute of Technology & Management, Hamidpur Delhi, affiliated to Guru Gobind Singh Indraprastha University, Dwarka(New Delhi) in partial fulfilment of requirement for the award of the degree of Bachelor of Technology in ....., has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition. The list of member(s) involved in the project is listed below: -

Student Name	Enrolment Number	Student Signature
Dhanu Bhardwaj	41313302716	
Rohit Kr Pandey	41613302716	

Chapter – 1

Team Details

Name: Dhanu Bhardwaj

College Roll No.: 41313302716

Project Contribution:

“Shape Detection using machine learning”

Name: Rohit Kr Pandey

College Roll No.: 41613302716

Project Contribution:

“Contributed in programming on shape detection coded the web related part ”

Chapter – 2

Abstract

Now almost every engineering related person knows what machine learning is and where it could be implimented . We have also used machine learning and created different application i.e making webpages using machine learning and in our case we are making geometric shape i.e (Rectangle , Pentagon , Triangle , Circle) on webpage by detecting it from live image shown to a computer by a webcam . The shapes are hard coded in the project .
This implimentaton could be extende to a project in which a website is being made live by drawing it on a board with the help of certain gadget.

Humans can expand their knowledge to adapt the changing environment. To do that they must “learn”. Learning can be simply defined as the acquisition of knowledge or skills through study, experience, or being taught. Although learning is an easy task for most of the people, to acquire new knowledge or skills from data is too hard and complicated for machines. Moreover, the intelligence level of a machine is directly relevant to its learning capability. The study of machine learning tries to deal with this complicated task. In other words, machine learning is the branch of artificial intelligence that tries to find an answer to this question: how to make computer learn?

Without further delay, let’s Start.....

Chapter – 3

Algorithms Used

3.1

Canny edge detector

Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Edges in images are areas with strong intensity contrasts – a jump in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. Canny edge detection algorithm is also known as the optimal edge detector. Canny's intentions were to enhance the many edge detectors in the image.

• The first criterion should have low error rate and filter out unwanted information while the useful information preserve.

• The second criterion is to keep the lower variation as possible between the original image and the processed image.

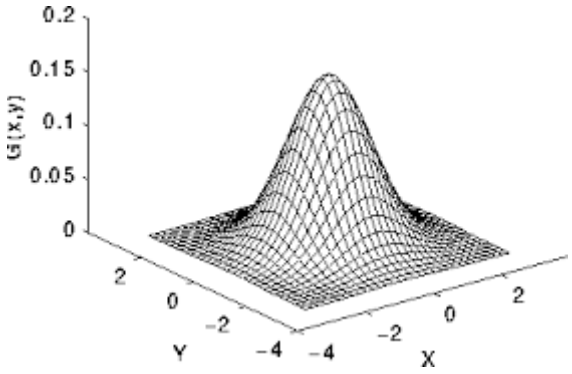
• Third criterion removes multiple responses to an edge. Based on these criteria, the canny edge detector first smoothes the image to eliminate noise. It then finds the image gradient to highlight regions

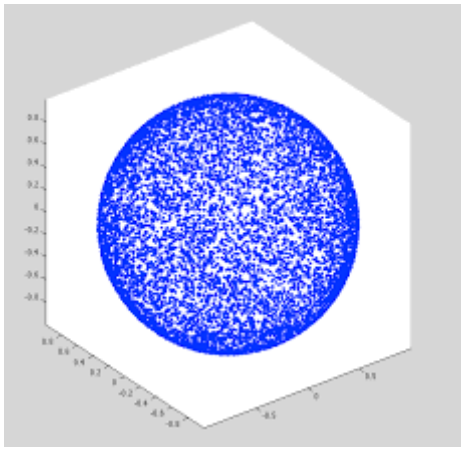
with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum using non-maximum suppression. The gradient array is now further reduced by hysteresis to remove streaking and thinning the edges.

Step 1:

Gaussian filtering to remove noise The first step of canny edge detection is to filter out any noise in the original image before trying to locate and detect any edges. The Gaussian filter is used to blur and remove unwanted detail and noise. By calculating a suitable 5 X 5 mask, the Gaussian smoothing can be performed using standard convolution method. A convolution mask is much smaller than the actual image. As a result, the mask is slid over the image, calculating every square of pixels at a time. Gaussian filter uses 2D distribution to perform convolution. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The weight of the matrix is concentrated at the centre, therefore any noise appearing in the outside columns and rows will be eliminated, as the weight decreases outward from the centre value. The localization error in the detected edges also increases slightly as the Gaussian width is increased. The increasing of standard deviation reduces or blurs the intensity of the noise.

$$\mathbf{K}_{\text{Gauss}} = \frac{\sigma^2}{\pi\delta_x\delta_y Q^2}\begin{pmatrix} \frac{2}{\sigma_x\sigma_y} & 0 & 0 & \frac{-1}{A\sigma_y} & \frac{-1}{A\sigma_x} \\ 0 & \frac{2\sigma_x}{A^2\sigma_y} & 0 & 0 & 0 \\ 0 & 0 & \frac{2\sigma_y}{A^2\sigma_x} & 0 & 0 \\ \frac{-1}{A\sigma_y} & 0 & 0 & \frac{2\sigma_x}{A^2\sigma_y} & 0 \\ \frac{-1}{A\sigma_x} & 0 & 0 & 0 & \frac{2\sigma_y}{A^2\sigma_x} \end{pmatrix},$$


0.2
0.15
0.1
0.05
0
G(X,Y)
2
0
−2
−4
Y
−4
−2
0
2
X



Chapter – 4

Libraries Used

4.1 Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Huguninwith contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Num-array into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

4.1.1. History

The Python programming language was not initially designed for numerical computing, but attracted the attention of the scientific and engineering community early on, so that a special interest group called matrix-sig was founded in 1995 with the aim of defining an array computing package. Among its members was Python designer and maintainer Guido van Rossum, who implemented extensions to Python HYPERLINK "https://en.wikipedia.org/wiki/Python_syntax_and_semantics"' HYPERLINK "https://en.wikipedia.org/wiki/Python_syntax_and_semantics"s syntax (in particular the indexing syntax) to make array computing easier. There was a desire to get Numeric into the Python standard library, but Guido van Rossum decided that the code was not maintainable in its state then. In early 2005, NumPy developer Travis Oliphant wanted to unify the community around a single array package and ported Num-array's features to Numeric, releasing the result as NumPy 1.0 in 2006.[6] This new project was part of SciPy. To avoid installing the large SciPy package just to get an array object, this new package was separated and called NumPy. Support for Python 3 was added in 2011 with NumPy version 1.5.0. In 2011, PyPy started development on an implementation of the NumPy API for PyPy.[13] It is not yet fully compatible with NumPy.[14]

4.1.2. Traits

NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops using NumPy.

Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted,[15] and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars. In comparison, MATLAB boasts a large number of additional toolboxes, notably Simulink, whereas NumPy is intrinsically integrated with Python, a more modern and complete programming language. Moreover, complementary Python packages are available; SciPy is a library that adds more MATLAB-like functionality and Matplotlib is a plotting package that provides MATLAB-like plotting functionality. Internally, both MATLAB and NumPy rely on BLAS and LAPACK for efficient linear algebra computations.

Python bindings of the widely used computer vision library OpenCV utilize NumPy arrays to store and operate on data. Since images with multiple channels are simply represented as three-dimensional arrays, indexing, slicing or masking with other arrays are very efficient ways to access specific pixels of an image. The NumPy array as universal data structure in OpenCV for images, extracted feature points, filter kernels and many more vastly simplifies the programming workflow and debugging.

The core functionality of NumPy is its "ndarray", for *n*-dimensional array, data structure. These arrays are strided views on memory.[6] In contrast to Python's built-in list data structure (which, despite the name, is a dynamic array), these arrays are homogeneously typed: all elements of a single array must be of the same type.

Such arrays can also be views into memory buffers allocated by C/C++, Cython,

and Fortran extensions to the CPython interpreter without the need to copy data around, giving a degree of compatibility with existing numerical libraries. This functionality is exploited by the SciPy package, which wraps a number of such libraries (notably BLAS and LAPACK). NumPy has built-in support for memory-mapped ndarrays.

4.1.3. Limitations

Inserting or appending entries to an array is not as trivially possible as it is with Python's lists. The np.pad(...) routine to extend arrays actually creates new arrays of the desired shape and padding values, copies the given array into the new one and returns it. NumPy's np.concatenate([a1,a2]) operation does not actually link the two arrays but returns a new one, filled with the entries from both given arrays in sequence. Reshaping the dimensionality of an array with np.reshape(...) is only possible as long as the number of elements in the array does not change. These circumstances originate from the fact that NumPy's arrays must be views on contiguous memory buffers. A replacement package called Blaze attempts to overcome this limitation.[16]

Algorithms that are not expressible as a vectorized operation will typically run slowly because they must be implemented in "pure Python", while vectorization may increase memory complexity of some operations from constant to linear, because temporary arrays must be created that are as large as the inputs. Runtime compilation of numerical code has been implemented by several groups to avoid these problems; open source solutions that interoperate with NumPy include scipy.weave, numexpr[17] and Numba.[18] Cython is a static-compiling alternative to these.

4.2 Pandas

In computer programming, **pandas** is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.[2] The name is derived from the term "panel data", an econometrics term for data

sets that include observations over multiple time periods for the same individuals.

4.2.1. Library Features

- Data Frame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Label-based slicing, fancy indexing, and sub setting of large data sets.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on data sets.
- Data set merging and joining.
- Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
- Time series-functionality: Date range generation[3] and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.

The library is highly optimized for performance, with critical code paths written in Python or C.

Open Cv (Computer Vision)

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV’s deployed uses span the range from stitching streetview images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE

instructions when available. A full-featured CUDAand OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

Chapter – 5

Codes Description and Code

```
#Importing required libraries and functions
import math
import numpy as np
import cv2


#Capturing Image from the Web cam

cap = cv2.VideoCapture(0)
print("press q to exit")
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480))
while(cap.isOpened()):
  #Capture frame-by-frame
    ret, frame = cap.read()
    if ret==True:

#Implimenting Canny Edge Detetion Algorithm

        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        #Canny
        canny = cv2.Canny(frame,80,240,3)
```

```
        #contours
        canny2, contours, hierarchy =
cv2.findContours(canny,cv2.RETR_EXTERNAL,cv2.CHAIN_AP
PROX_SIMPLE)


#Detection of  shapes
for i in range(0,len(contours)):
            #approximate the contour with accuracy
proportional to
            #the contour perimeter
            approx =
cv2.approxPolyDP(contours[i],cv2.arcLength(contours[i],Tru
e)*0.02,True)
            cv2.drawContours(frame, contours[i], -1, (0, 0 ,
255), 3)
            #Skip small or non-convex objects
            if(abs(cv2.contourArea(contours[i]))<100 or
not(cv2.isContourConvex(approx))):
                continue


            #triangle
            if(len(approx) == 3):
                x,y,w,h = cv2.boundingRect(contours[i])
                cv2.putText(frame,'TRIANGLE',
(x,y),cv2.FONT_HERSHEY_SIMPLEX,scale,
(255,255,255),2,cv2.LINE_AA)
```

```
        elif(len(approx)>=4 and len(approx)<=6):
            vtc = len(approx)
            x,y,w,h = cv2.boundingRect(contours[i])
            if(vtc==4):
                cv2.putText(frame,'RECTANGLE',
(x,y),cv2.FONT_HERSHEY_SIMPLEX,scale,
(255,255,255),2,cv2.LINE_AA)
            elif(vtc==5):
                cv2.putText(frame,'PENTAGAN',
(x,y),cv2.FONT_HERSHEY_SIMPLEX,scale,
(255,255,255),2,cv2.LINE_AA)
            elif(vtc==6):
                cv2.putText(frame,'HEXA',
(x,y),cv2.FONT_HERSHEY_SIMPLEX,scale,
(255,255,255),2,cv2.LINE_AA)
        else:

            #detect and label circle
            area = cv2.contourArea(contours[i])
            x,y,w,h = cv2.boundingRect(contours[i])
            radius = w/2
            if(abs(1 - (float(w)/h))<=2 and abs(1-(area/
(math.pi*radius*radius)))<=0.2):

                cv2.putText(frame,'CIRCLE',
(x,y),cv2.FONT_HERSHEY_SIMPLEX,scale,
(255,255,255),2,cv2.LINE_AA)
```

```
        #Display the resulting frame
        out.write(frame)
        cv2.imshow('frame',frame)
        cv2.imshow('canny',canny)
        if cv2.waitKey(1) == 27: #if q is pressed
            break

#When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
```

#opening the browser

```
import webbrowser as browser
string = "pentago"
if (string == "triangle"):
        browser.open("triangle.html")
elif (string == "rectangle"):
        browser.open("rectangle.html")
elif (string == "circle"):
        browser.open("circle.html")
elif(string=="pentagon"):
        browser.open("pentagon.html")
else:
        print("The shape is not defined")
```

#HTML code for the shapes

#For the triangle

```
DOCTYPE html>
<html>
<head>
        <title>Traingle</title>
        <link rel="stylesheet" type="text/css" href="stylesheet.css">
</head>
<body>
        <div class="container">
                        <div id="textspecial"><p><h3>This is a
Triangle</h3></p></div>
                        <div class="triangle"></div>
        </div>
</body>
</html>
```

 # For the Rectangle

```
<!DOCTYPE html>
<html>
<head>
        <title>Rectangle</title>
        <link rel="stylesheet" type="text/css" href="stylesheet.css">
</head>
<body>
        <div class="container">
                        <div id="text"><p><h3>This is a
Rectangle</h3></p></div>
                        <div class="rectangle"></div>

        </div>
</body>
```

```
</html>
```

#For the circle

```
<!DOCTYPE html>
<html>
<head>
	<title>circle</title><link rel="stylesheet" type="text/css" href="stylesheet.css">
</head>
<body>
	<div class="container">
		<div id="text"><p><h3>This is a Circle</h3></p></div>
		<div class="circle"></div>

	</div>
</body>
</html><!DOCTYPE html>
<html>
<head>
	<title>circle</title><link rel="stylesheet" type="text/css" href="stylesheet.css">
</head>
<body>
	<div class="container">
		<div id="text"><p><h3>This is a Circle</h3></p></div>
		<div class="circle"></div>

	</div>
</body>
</html>
```

#For the Pentagon

```
<!DOCTYPE html>
<html>
<head>
	<title>circle</title><link rel="stylesheet" type="text/css"
href="stylesheet.css">
</head>
<body>
	<div class="container">
		<div id="text"><p><h3>This is a Circle</h3>
	</p></div>
		<div class="circle"></div>

	</div>
</body>
</html>
```

#For the square

```
<!DOCTYPE html>
<html>
<head>
	<title>Square</title>
	<link rel="stylesheet" type="text/css" href="stylesheet.css">
</head>
<body>
	<div class="container">

		<div id="text"><p><h3>This is a Square</h3>
	</p></div>
		<div class="square"></div>

	</div>
</body>
</html>
```

#Css Code for Styling the webpage

```
body{
        padding:0;
        margin: 0;
        width: 100%;
        height:100vh;

}
.container{
        display: flex ;
        width :100%;
        height: 100vh;
        background-color: white ;
        justify-content: center ;
        border:1px solid black ;
}

.triangle{
        width :0px;
        height:0px;
        border-left:125px solid transparent;
        border-right:125px solid transparent;
        border-bottom:150px solid black;
```

```
        position:absolute ;
        top:35vh;
        text-align: center;

}
.rectangle{
        width :500px;
        height: 300px;
        background-color: black ;
        position:absolute;
        top :30vh;
        box-shadow: 10px 10px 5px gray;

}
.circle{
        width :300px;
        height: 300px;
        background-color: black ;
        position:absolute;
        top :30vh;
        border : 1px solid transparent;
        border-radius: 50%;
        box-shadow: 10px 10px  5px gray;
```

```
}
.square{
        width:300px;
        height: 300px;
        background-color: black ;
        position:absolute;
        top :30vh;
        box-shadow: 10px 10px  5px gray;

}
#pentagon{
        width :400px;
        height:400px;
        position:relative;
        top:25vh;
        position :fixed;
}
 h3{
        color :maroon;
        font-family: halvatica;
}
#text{
        position: relative;
```

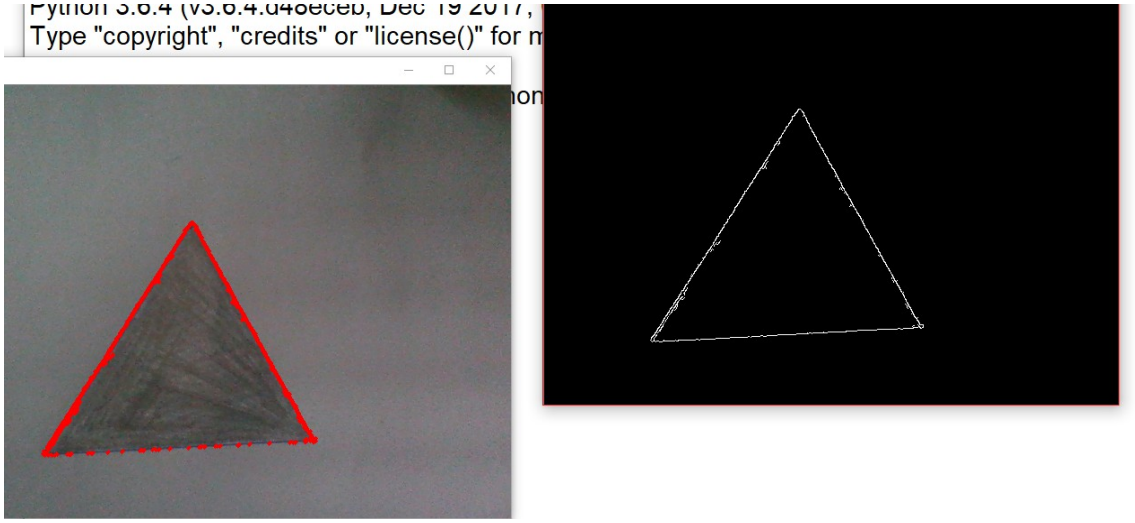
```
    animation-name: allshapes;
    animation-duration: 4s;
}


@keyframes allshapes {
    0%   { left:0px; top:0px;font-size: 80px;}
    25%  { left:200px; top:0px;}
    50%  { left:450px; top:450px;}
    75%  {left:0px; top:450px;}
    100% { left:0px; top:0px; font-size: 80px;}
}
#textspecial{
        position: relative;
    animation-name: triangle;
    animation-duration: 4s;
}
@keyframes triangle {
    0%   { left:0px; top:0px;font-size: 30px;}
    25%  { left:200px; top:0px;}
    50%  { left:350px; top:350px;}
    75%  {left:0px; top:350px;}
    100% { left:0px; top:0px; font-size: 30px;}
```

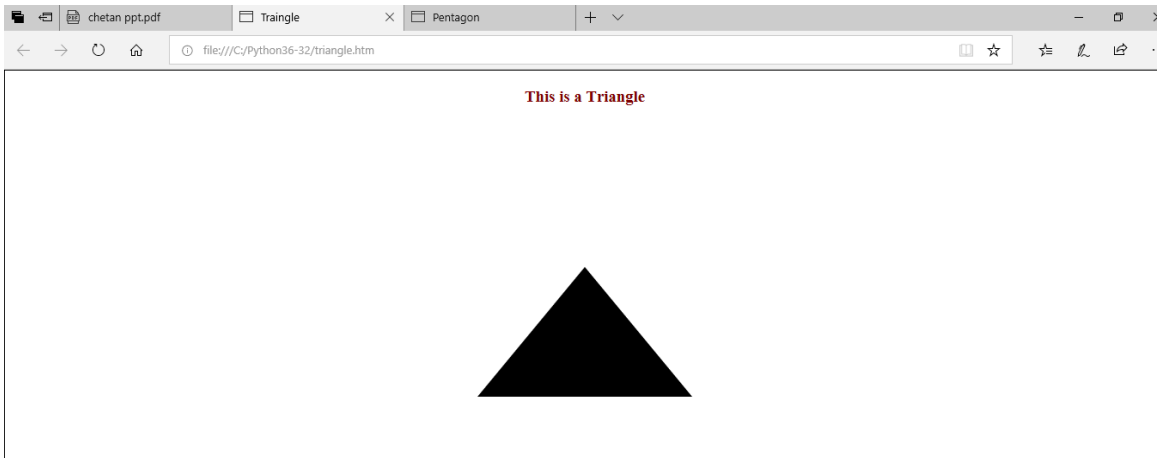
```
}
#pentagontxt{
        position: relative;
    animation-name: penta;
    animation-duration: 4s;

}

@keyframes penta {
    0%   { left:-10px; top:0px;font-size: 80px;}
    25%  { left:200px; top:0px;}
    50%  { left:500px; top:500px;}
    75%  {left:0px; top:450px;}
    100% { left:0px; top:0px; font-size: 80px;}
}
@media screen and (max-width : 550px ) and ( max-height : 550px )
        {
                .rectangle{
                        width: 350px;
                        height: 200px ;

                }
                .triangle{
```

```
                    border-left:55px solid transparent;
                    border-right:55px solid
transparent;
                    border-bottom:90px solid black;
                }
                .circle{
                    width:150;
                    height:150;
                    background-color: blue;
                    position:relative;
                    top:2px;
                }
                .square{

                    width:150;
                    height:150;
                    display:block;
                }
            }
```

Chapter – 6

(Screen Shots)



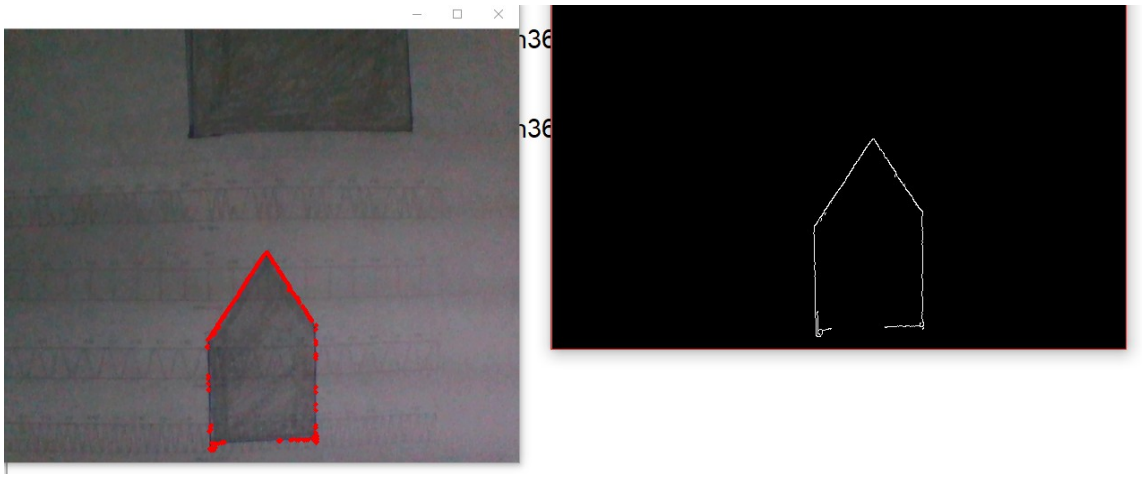


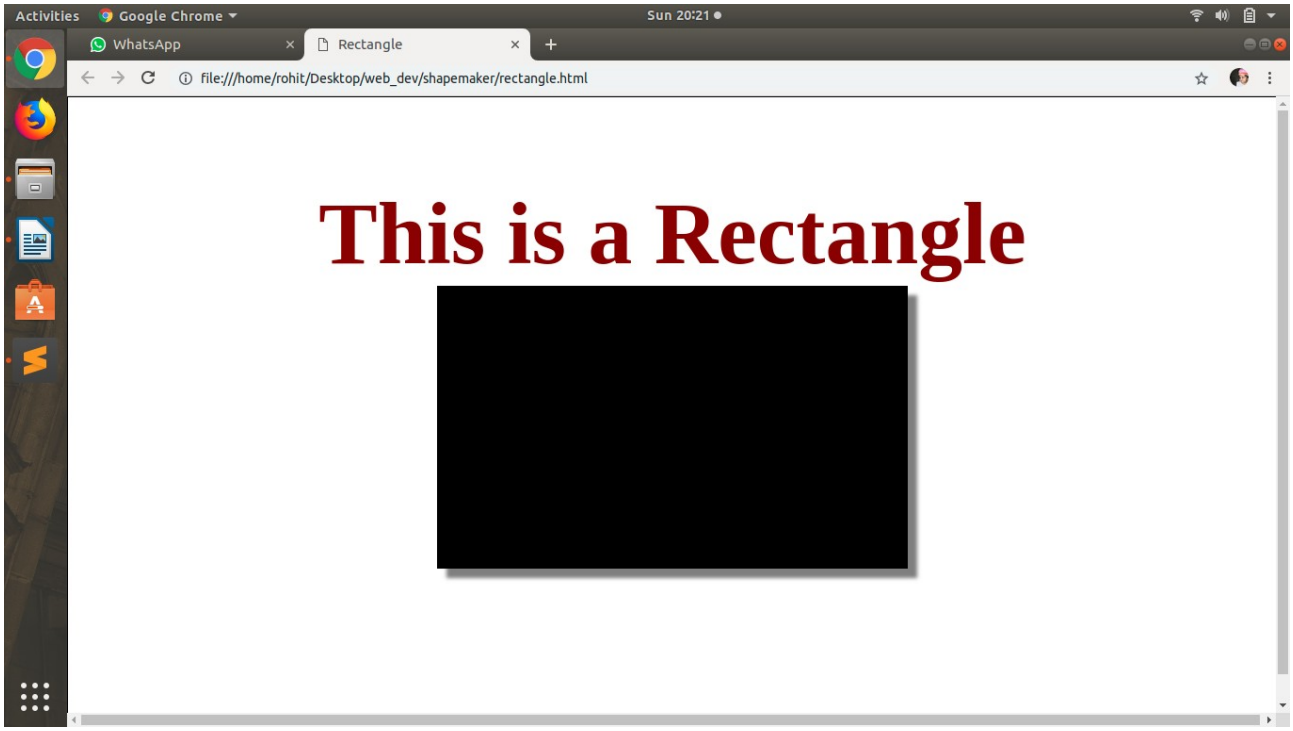
Chapter – 7

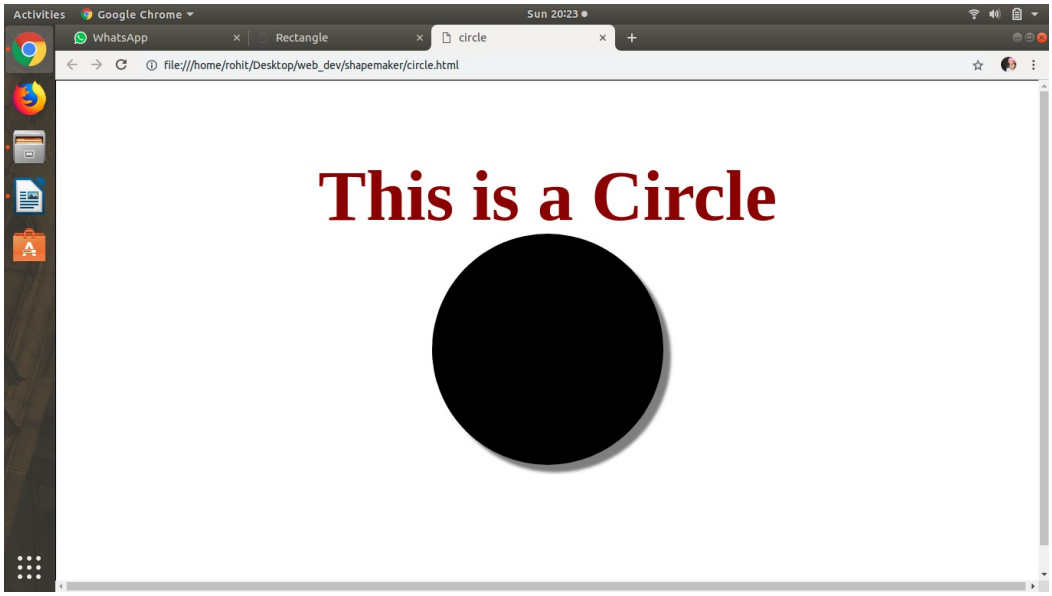
Idel Console

```
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:\Python36-32\shape_detection.py ==================
triangle
>>>
```

(Screen Shots)




Activities
Google Chrome
Sun 20:21
WhatsApp
Rectangle
file:///home/rohit/Desktop/web_dev/shapemaker/rectangle.html
This is a Rectangle


Activities
Google Chrome
Sun 20:23
WhatsApp
Rectangle
circle
file:///home/rohit/Desktop/web_dev/shapemaker/circle.html
This is a Circle

Chapter – 9

Summary

This project can be designed in such a way that it becomes capable of making a pure html page by just drawimg things on a Whiteboard , the code will be generated using machine learning responsive design will be made using flexes or etc .