SMART MATHEMATICS TUTOR – INTERACTIVE LEARNING APPLICATION FOR STUDENTS USING IBM WATSON

A MINI PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

NEELA DURGA BHAVANI	**18UK1A0541**
GAAJE MANASA	**18UK1A0515**
BOLLOJU VEERENDRACHARY	**18UK1A0565**
KALERU KAVYA	**18UK1A0524**

Under the esteemed guidance of

Ms. B. AMBIKA

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTU Hyderabad & Approved by AICTE,New Delhi) Bollikunta , Warangal – 506005

2018-2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

(Affiliated to JNTU Hyderabad & Approved by AICTE, NewDelhi)

Bollikunta ,Warangal – 506005

2018 – 2022

CERTIFICATE

This is to certify that the Mini Project Report entitled **"SMART MATHEMATICS TUTOR - INTERACTIVE LEARNING APPLICATION FOR STUDENTS USING IBM WATSON"** is being submitted by ***N.DURGA BHAVANI (H.NO:18UK1A0541), G.MANASA (H.NO:18UK1A0515), B.VEERENDRACHARY (H.NO:18UK1A0565), K.KAVYA (H.NO:18UK1A0524)*** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering toJawaharlal Nehru Technological University Hyderabad during the academic year 2021-22, is a record of work carried out by them under the guidance and supervision.

Ms. B . AMBIKA
Project Guide

Dr. R. NAVEEN KUMAR
Head of the Department

EXTERNAL EXAMINAR

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this mini project in the institute.

We extend our heartfelt thanks to **Dr. R. NAVEEN KUMAR**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure andthereby giving us freedom to carry out the mini project.

We express heartfelt thanks to **Mr. SHIVAM**, Program Manager, SmartBridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the mini project and for their support in completing the mini project, mini project and internship.

We express heartfelt thanks to the guide, **Ms. B. AMBIKA** Assistant professor, Department of CSE for her constant support and giving necessary guidance for completion of this mini project.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the this.

INDEX

TITLE	Page. No
1. INTRODUCTION	**6**
1.1 Introduction	
1.2 Overview	
1.3 Purpose	
2. LITERATURE SURVEY	**7**
2.1 Existing Problem	
2.2 Proposed solution	
3. THEORTICAL ANALYSIS	**8**
3.1 Block diagram	
3.2 Hardware/Software designing	
4. EXPERIMENTAL INVESTIGATIONS	**9-20**
4.1 Methods and Implementation	
5. FLOW CHART	**21**
6. RESULTS	**22-23**
7.ADVANTAGES AND DISADVANTAGES	**24**
8. APLLICATIONS	**25-26**
9. CONCLUSION	**27**
10. FUTURE SCOPE	**28-30**
11. BIBILOGRAPHY	**31**
APPENDIX	**32-40**

ABSTRACT

Nationally, the average age at which kids get a phone of their own is 10.3 years. One thing experts agree on is that later is better. Once you open the door, it can be very difficult to close. A 2016 study found that most kids are getting their first social media account between the ages of 10 and 12. Due to lack of interactive learning students doesn't show much interest in learning things. Interactive Learning is a pedagogical approach that incorporates social networking and urban computing into course design and delivery. Interactive Learning has evolved out of the hyper-growth in the use of digital technology and virtual communication, particularly by students. Beginning around 2000, students entering institutes of higher education have expected that interactive learning will be an integral part of their education. The use of interactive technology in learning for these students is as natural as using a pencil and paper were to past generations.

To solve the above mentioned problem we are building a GUI which helps the studing in learning maths and they can easily remember shapes in the application which recognizes the shape gives all the related information like list of formulas. This helps the students to learn interactively.

Smart Maths tutor system is a web based graphical user interface where a user gets to draw shapes of mathematical figures such as square, triangle, circle etc. for which the output would be related formulas to the drawn figure.

1.INTRODUCTION

1.1 Introduction

Our project ‘Smart mathematics Tutor’ includes shape recognition system. The aim of our project is to create tutoring assistant which will prove to be effective in helping Math students to practice shape recognition exercises. For the assistant to provide the needed guidance to a student who are learning to recognize the shapes, it is necessary to take into consideration both the shape that is needed to be recognized, as well as the name of the shape proposed by the learner.

This tutoring assistant will use a shape generator designed to test the knowledge of the student. this shape generator is created by our team to form different shape so that the students can try guess the name of the shapes and find if their answer was correct or not which will help them build their knowledge in an easy way. Shape detection is the identification of a shape in the image along with its localization and classification. It has wide spread applications and is a critical component for AI based software systems. And by using this shape detection our tutor will help students learn about different shapes.

1.2 Overview

Our website being a computer based online system, it can be proved to be efficient and easy in creating different shapes and recognizing them which will be helpful for students who are learning about shapes.

1.3 Purpose

Can we develop a website that efficiently manage shape details, increase accuracy and make it overall user friendly? while doing so website should be secure simple understand.

2.LITERATURE SURVEY

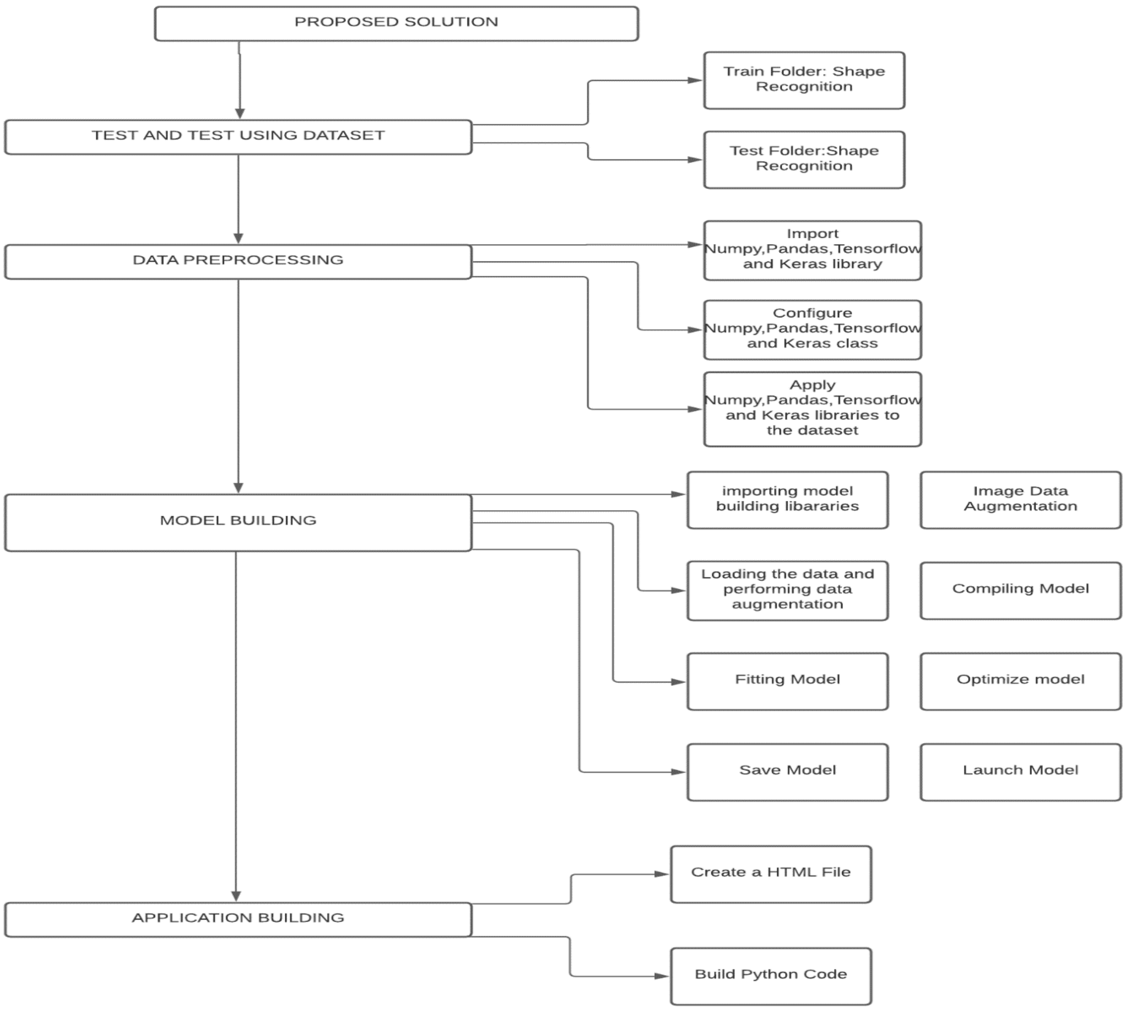
2.1 Existing Problem

Shape detection is the identification of an shape in the image along with its localisation and classification. It has wide spread applications and is a critical component for AI based software systems. This report seeks to perform a rigorous survey of modern shape detection algorithms that use Machine learning. As part of the survey, the topics explored include various algorithms, quality metrics, speed/size trade-offs and training methodologies. This report focuses on the two types of Shape detection algorithms - CNN and Data Pre-processing. Techniques to construct detectors that are portable and fast on low powered devices are also addressed by exploring new lightweight convolutional base architectures. Ultimately, a rigorous review of the strengths and weaknesses of each detector leads us to the present state of the art.

2.2 Problem Solution

"Shape Matching and Object Recognition Using Shape Contexts", proposed shape detection method using a feature called shape context. Shape context describes all boundary points of a shape with respect to any single boundary point. Shape recognition can be achieved by matching this feature with a prior knowledge of the shape context of the boundary points of the object.Problem-solving is an important skill that one must have. Problem-solving in mathematics helps students to experience on how to solve daily life problems by applying their mathematical knowledge and skill. Word problem solving is one of the important components of mathematical problem-solving incorporates real-life problems and applications. however, students cannot perform well in the examination when it comes to word problem-solving question.

3.THEORETICAL ANALYSIS



Hardware /Software designing

Hardware requirement: Laptop.

Software requirement: Python -3.6, Keras -2.2.4, Tensorflow-1.13.0, Spyder, Juypter, Notebook.

4. EXPERIMENTAL INVESTIGATIONS

We investigate shape recognition that models the decision based on supervised learning, where the model is built up based on previously labelled inputs denoted as templates; the set of already known inputs is denoted as training and testing set.

Independently from the exact type and behaviour of the classifier, the classification is a comparison of the input to labelled elements from the training set (or a model built up from the set), where the decision is a function of the representation.

The difference between the representations of the same object is a result of various distortions that occur during the shape acquisition and Data pre-processing.

The task of the recognition is not the reconstruction of the original shape by mathematical operations but to classify independently from transformations that distort the original and the template shapes and thus to estimate the ground truth.

In a real classroom, an expert human teacher who is rich in domain knowledge explains the concepts to the students using various tools evaluations are conducted to test the mastery of the student over subject . During this interactions , the teacher tries to model/ assess individual student behaviour and accordingly decides the pedagogical strategy. A tutoring system should attempt to mimic the best of the collectively human teachers while putting major effort on individualized attention, because this is where teacher fails or falls short to deliver due to time and other constraints.

4.1Methods And Implementation

Importing Neccessary Libraries

```
import numpy as np#used for numerical analysis
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense,Flatten
#Faltten-used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D,MaxPooling2D #Convolutional layer
#MaxPooling2D-for downsampling the image
from keras.preprocessing.image import ImageDataGenerator
```

Initializing The Model

Sequential model is a linear stack of layers. You can create a Sequential model by passing a list of layer instances to the constructor: from keras . models import Sequential from keras as follows.

```
model=Sequential()
```

Adding CNN Layers

For information regarding CNN Layers reer to the link. We are adding a convolution layer with activation function as “relu” and with a small filter size (3,3) and number of filters (32) followed by a max pooling layer.

```
# First convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 1), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
# Second convolution layer and pooling
classifier.add(Conv2D(32, (3, 3), activation='relu'))
# input_shape is going to be the pooled feature maps from the previous convolution layer
classifier.add(MaxPooling2D(pool_size=(2, 2)))

# Flattening the layers
classifier.add(Flatten())
```

Adding Dense Layers

Dense layer is deeply connected neural network layer. It is most common and frequently used layer.

```
model.add(Dense(32))#deeply connected neural network layers.
model.add(Dense(3,activation='softmax'))#output layer with 6 neurons
```

Understanding the model is very important phase to properly use it for training and prediction purposes. Keras provides a simple method, summary to get the full information about the model and its layers.

```
model.summary()#summary of our model

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 62, 62, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 31, 31, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 29, 29, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 6272)              0
_________________________________________________________________
dense (Dense)                (None, 32)                200736
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 99
=================================================================
Total params: 210,979
Trainable params: 210,979
Non-trainable params: 0
_________________________________________________________________
```

Configure The Learning Process

- The compilation is the final step in creating a model. Once the compilation is done, we can move on to training phase. Loss function is used to find error or deviation in the learning process. Keras requires loss function during model compilation process.
- Optimization is an important process which optimize the input weights by comparing the prediction and the loss function. Here we are using adam optimizer
- Metrics is used to evaluate the performance of your model. It is similar to loss function, but not used in training process.

Compiling the model

```
# Compiling the CNN
# categorical_crossentropy for more than 2
classifier.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Now , let us train our model with our image dataset. **fit_generator** functions used to train a deep learning neural network.

Arguments:

- steps_per_epoch : it specifies the total number of steps taken from the generator as soon as one epoch is finished and next epoch has started. We can calculate the value of steps_per_epoch as the total number of samples in your dataset divided by the batch size.

  - Epochs : an integer and number of epochs we want to train our model for.
  - validation_data can be either:
    - an inputs and targets list
    - a generator

- validation_steps : only if the validation_data is a generator then only this argument be used. It specifies the total number of steps taken from the generator before it is stopped at every epoch and its value is calculated as the total number of validation data points in your dataset divided by the validation batch size.

```
# Fit the model
model.fit_generator(generator=x_train,steps_per_epoch = len(x_train),
                    epochs=5, validation_data=x_test,validation_steps = len(x_test))
```

```
WARNING:tensorflow:From <ipython-input-10-a6a0079b99fe>:2: Model.fit_generator (from tensorflow.python.keras.engine.training) is deprecate
d and will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/5
341/341 [==============================] - 369s 1s/step - loss: 0.1025 - accuracy: 0.9632 - val_loss: 6.4172e-04 - val_accuracy: 1.0000
Epoch 2/5
341/341 [==============================] - 101s 296ms/step - loss: 0.0031 - accuracy: 0.9995 - val_loss: 2.1346e-04 - val_accuracy: 1.0000
Epoch 3/5
341/341 [==============================] - 91s 266ms/step - loss: 0.0010 - accuracy: 0.9997 - val_loss: 1.6276e-05 - val_accuracy: 1.0000
Epoch 4/5
341/341 [==============================] - 86s 252ms/step - loss: 0.0036 - accuracy: 0.9987 - val_loss: 0.0112 - val_accuracy: 1.0000
Epoch 5/5
341/341 [==============================] - 87s 254ms/step - loss: 4.8451e-04 - accuracy: 1.0000 - val_loss: 2.7026e-05 - val_accuracy: 1.0
000
```

Save The Model

The model is saved with .h5 extension as follows an H5 file is a data file saved in the Hierarchical Data Format (HDF). It contains multidimensional arrays of scientific data.

Saving our model

```
# Save the model
model.save('shape.h5')
```

Test The Model

Evaluation is a process during development of the model to check whether the model is best fit for the given problem and corresponding data. Load the saved model using load_model.

Predicting our results

```
from tensorflow.keras.models import load_model
from keras.preprocessing import image
model = load_model("shape.h5") #loading the model for testing
```

Taking an image as input and checking the results

```
img = image.load_img(r"E:\PROJECTS\Maths Tutor\shapes\test\square\0.png",
                     target_size= (64,64))#loading of the image
x = image.img_to_array(img)#image to array
x = np.expand_dims(x,axis = 0)#changing the shape
pred = model.predict_classes(x)#predicting the classes
pred
```

By using the model we are predicting the output for the given input image

```
index=['circle', 'square', 'triangle']
result=str(index[pred[0]])
result
```

```
'square'
```

The predicted class index name will be printed here.

Application Building

Now after the model is trained in this particular milestone, we will be building our flask application which will be running in our local browser with a user interface.

Create HTML Pages

- We use HTML to create the front end part of the web page.
- Here, we created 2 html pages- home.html, intro.html
- home.html displays home page.
- intro.html displays all the details about our project.

  For more information regarding HTML

  Link: https://www.w3schools.com/html/

  - We also use JavaScript-main.js and CSS-main.css to enhance our functionality and view of HTML pages.

- **Link :** CSS , JS

Build Python Code

- Let us build flask file ‘app.py’ which is a web framework written in python for server-side scripting. Let’s see step by step procedure for building the backend application.
- App starts running when “__name__” constructor is called in main.
- render_template is used to return html file.
- “GET” method is used to take input from the user.
- “POST” method is used to display the output to the user.

- Importing Libraries

```
from flask import Flask,render_template
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
from tkinter import * # Graphical User Interface package
import PIL
from PIL import ImageGrab # used to copy the contents of the screen
import model # loading our model python file
import cv2  #opencv library
```

- Creating our flask application

```
app = Flask(__name__) # initializing a flask app
```

- Routing to the different html pages

```
@app.route('/')# route to display the home page
def home():
    return render_template('home1.html')#rendering the home page


@app.route('/intro')
def intro():
    return render_template('intro.html')#rendering the intro page
```

The above two route are used to render the home and introduction html pages.

```
@app.route('/launch',methods=['GET', 'POST'])# route to show the predictions in a web UI
def launch():
```

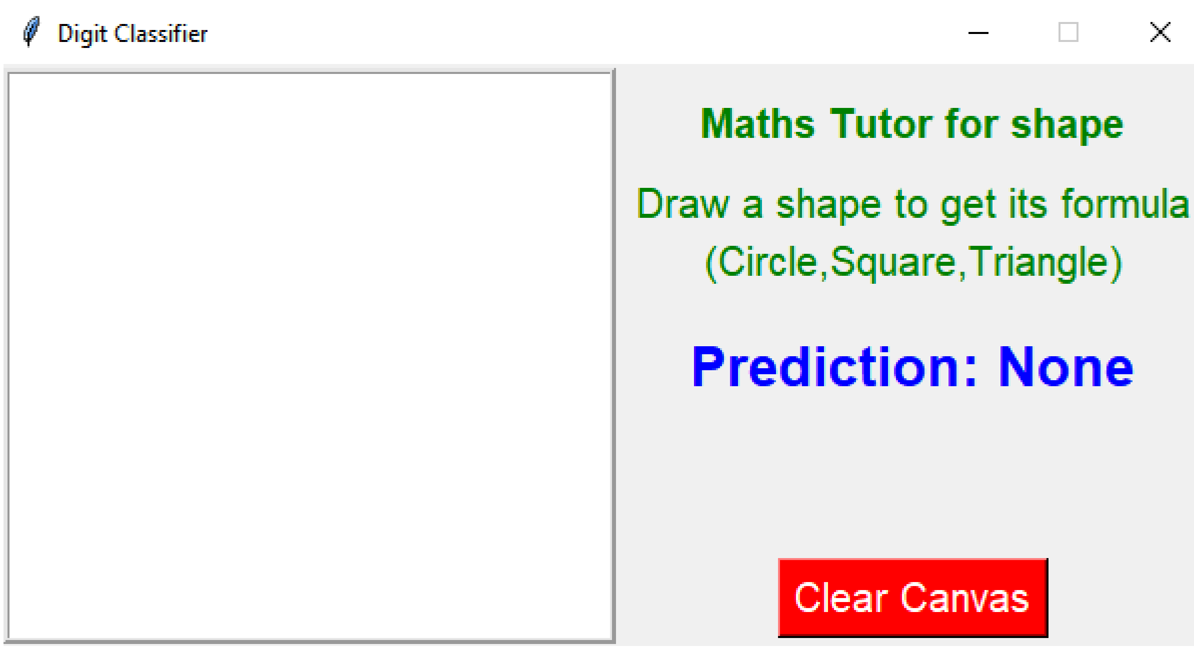
And the launch route is used for prediction and it contains all the codes which are used for predicting our results.

- Firstly, inside launch function we will be creating a class inside which we have different function with different operation and we will be going to do the following operations:
    - Creating a GUI using tkinter concepts.
    - Grab the image and send it to our model which we build.
    - Showcase the results with the help of opencv.
    - Finally run the application.

- **Creating a GUI**

```python
def __init__(self, master):
    self.master = master
    self.res = ""
    self.pre = [None, None]
    self.bs = 4.5
    self.c = Canvas(self.master,bd=3,relief="ridge", width=300, height=282, bg='white')
    self.c.pack(side=LEFT)
    f1 = Frame(self.master, padx=5, pady=5)
    Label(f1,text="Maths Tutor for shape",fg="green",font=("",15,"bold")).pack(pady=10)
    Label(f1,text="Draw a shape to get its formula",fg="green",font=("",15)).pack()
    Label(f1,text="(Circle,Square,Triangle)",fg="green",font=("",15)).pack()
    self.pr = Label(f1,text="Prediction: None",fg="blue",font=("",20,"bold"))
    self.pr.pack(pady=20)

    Button(f1,font=("",15),fg="white",bg="red", text="Clear Canvas",
           command=self.clear).pack(side=BOTTOM)

    f1.pack(side=RIGHT,fill=Y)
    self.c.bind("<Button-1>", self.putPoint)
    self.c.bind("<ButtonRelease-1>",self.getResult)
    self.c.bind("<B1-Motion>", self.paint)
```

In the above code we are creating a function inside which we create a Canvas with some text along with the button. This code result in



- **Grabbing the image and sending it to our model for prediction**

```python
def getResult(self,e):
    x = self.master.winfo_rootx() + self.c.winfo_x()
    y = self.master.winfo_rooty() + self.c.winfo_y()
    x1 = x + self.c.winfo_width()
    y1 = y + self.c.winfo_height()
    img = PIL.ImageGrab.grab()
    img = img.crop((x, y, x1, y1))
    img.save("dist.png")
    self.res = str(model.predict("dist.png"))
    self.pr['text'] = "Prediction: " + self.res
```

Here the shape which we draw in the canvas will be grabbed with the help of ImageGrab function and that should be saved as dist.png in our local system. Then we will send that image to our model which we build earlier by calling the predict function which is saved in the model.py file.

```python
from tensorflow.keras.models import load_model
import numpy as np
from tensorflow.keras.preprocessing import image


model = load_model(r"E:\PROJECTS\Maths Tutor\shape.h5") # loading our model

def predict(InputImg):

    img=image.load_img(InputImg,target_size=(64,64)) #load and reshaping the image
    x=image.img_to_array(img)#converting image to array
    x=np.expand_dims(x,axis=0)
    pred=model.predict_classes(x)
    #pred=np.argmax(model.predict(x), axis=-1)
    print(pred)
    index=['circle', 'square', 'triangle']
    result=str(index[pred[0]])


    return result
```

Our model will predict the prediction and return the result which will be saved in self.res.

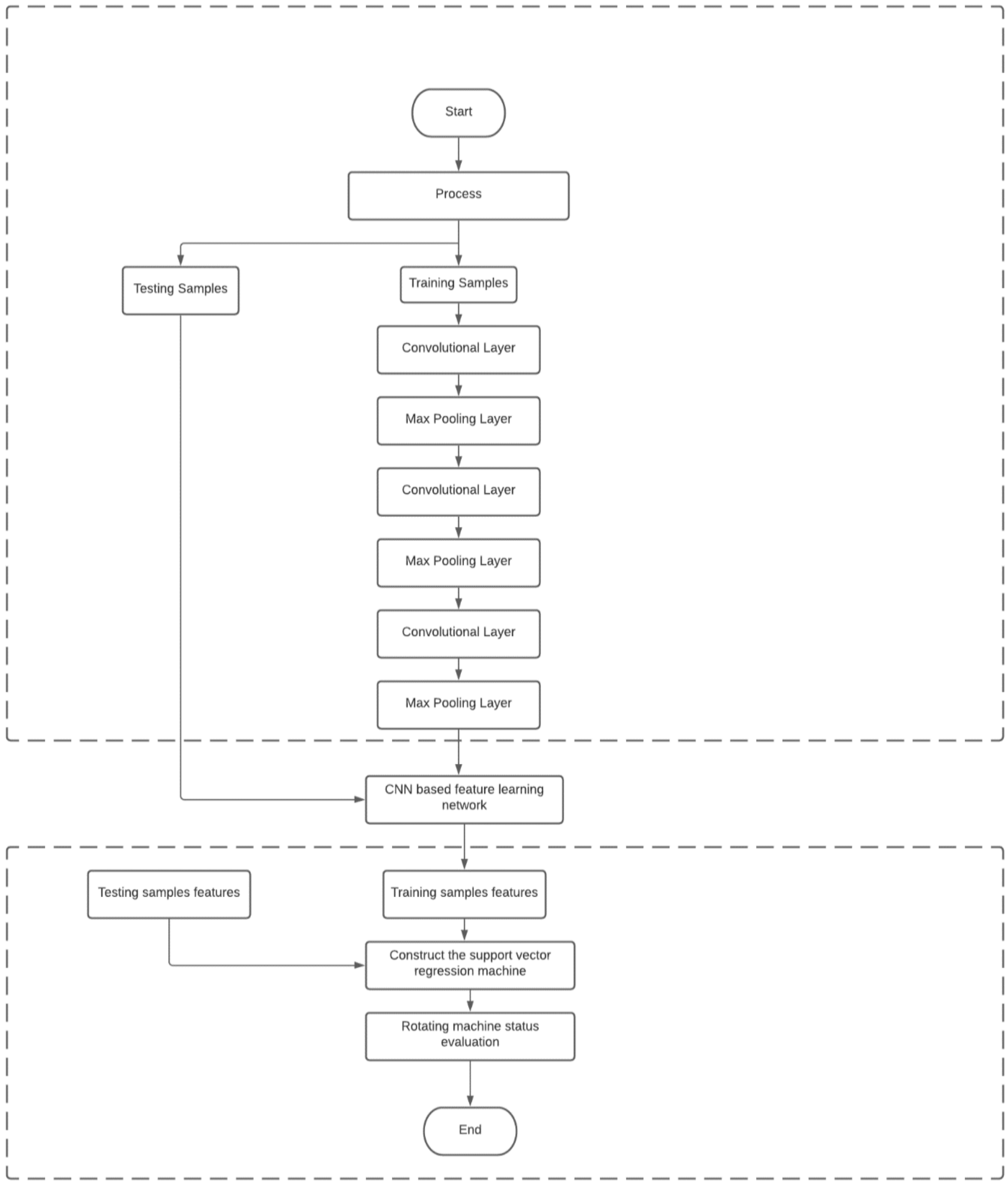
- **Showcase the results with the help of opencv**

Now as we know our result are stored in self.res so we will be creating nested if statements and with the help of opencv we will showcase the results according to our predictions.

```
if self.res=='circle':
    image = cv2.imread('circle.png')
    cv2.imshow('circle', image)
    key=cv2.waitKey(0)

    if (key & 0xFF) == ord("c"):
        cv2.destroyWindow("circle")

elif self.res=='square':
    image = cv2.imread('square.png')
    cv2.imshow('square', image)
    key=cv2.waitKey(0)

    if (key & 0xFF) == ord("s"):
        cv2.destroyWindow("square")
else:

    image = cv2.imread('triangle.png')
    cv2.imshow('triangle', image)
    key=cv2.waitKey(0)

    if (key & 0xFF) == ord("t"):
        cv2.destroyWindow("triangle")
```

Some of the functions which are needed for our GUI to run i.e., clear canvas, drawing with black color etc.

5.FLOWCHART



Start
Process
Testing Samples
Training Samples
Convolutional Layer
Max Pooling Layer
Convolutional Layer
Max Pooling Layer
Convolutional Layer
Max Pooling Layer
CNN based feature learning network
Testing samples features
Training samples features
Construct the support vector regression machine
Rotating machine status evaluation
End

6.RESULT

Smart Mathematics Tutor developed by us is an achievement that could change the education and learning experience of a student. It keeps it easy for the students to learn and for the teachers to perform their duties in a better way.

The Modules of Smart Mathematics Tutor developed by our team shows their own efficiency and effectiveness. With the help of this system, it will be easy to manage and upgrade the tutor system according to the demand and requirements.

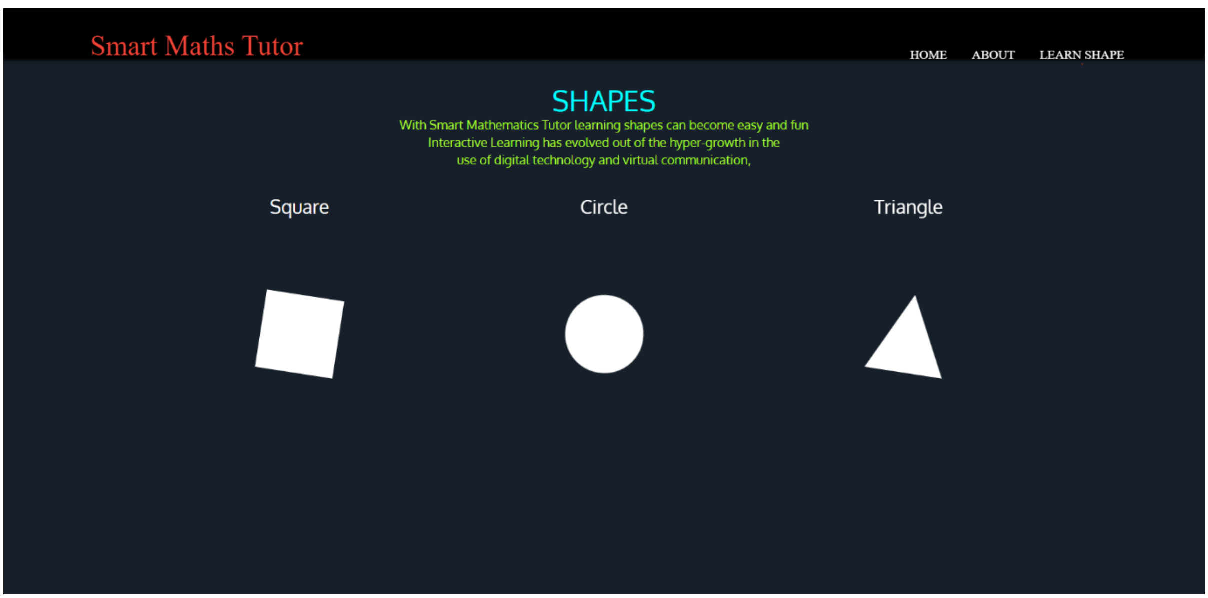


Fig 1. Home page

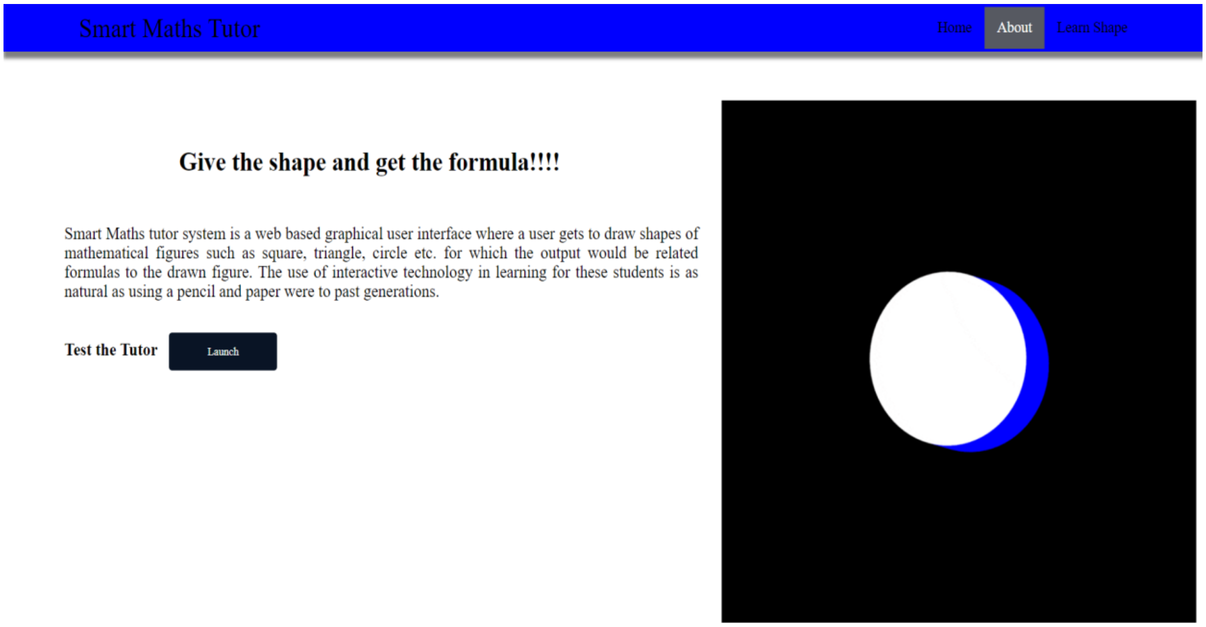


Fig 2. About Us page

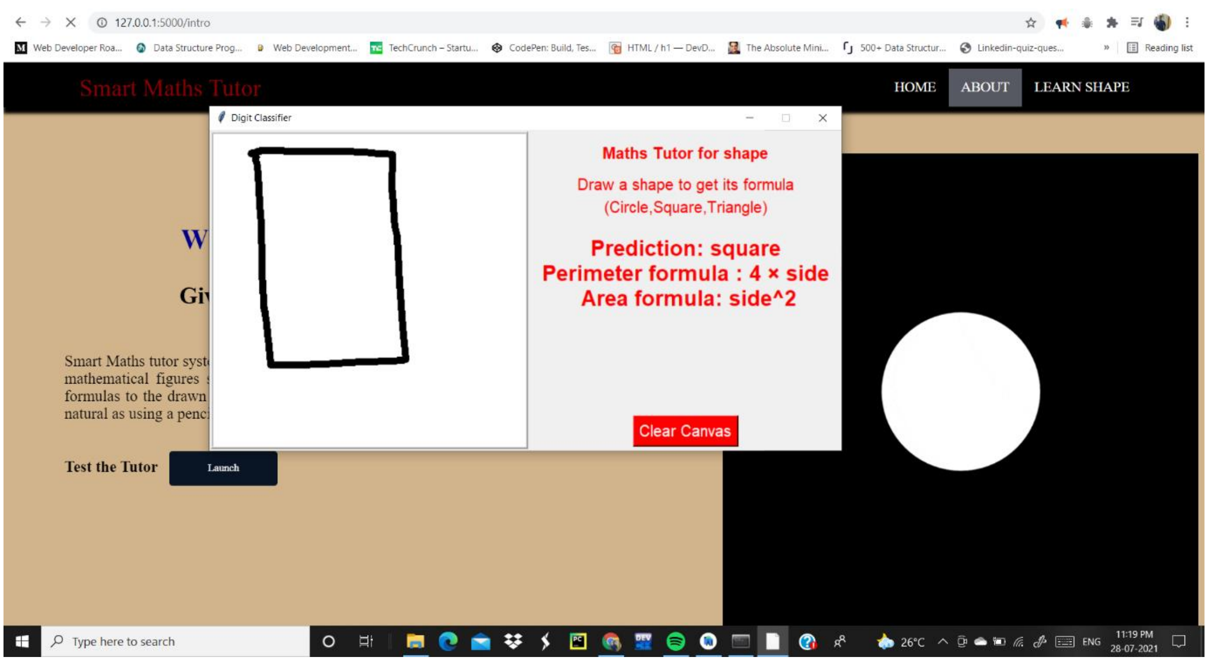


Fig. 3 Learn Shape Page

7. ADVANTAGES AND DISADVANTAGES

to refine teaching **Advantages** of Smart Mathematics tutoring systems are:

1. Be available at any time of the day, even late at night before an exam.

2. Provide real-time data to instructors and developers looking methods.
3. Reduce the dependence on human resources

4. Help students better understand material by allowing them to first explain what they know, then by catering responses accordingly

5. Afford educators the opportunity to create individualized programs due to their personalized nature.
6. Yield higher test scores than traditional systems, especially in students from special education, non-native English, and low-income backgrounds.

7. Provide immediate yes/no feedback, individual task selection, on-demand hints, and support for mastery learning.

Disadvantages of Smart Mathematics tutoring systems are:

1. It is difficult to assess the effectiveness of SMT programs.

2. Immediate feedback and hint sequences fail to develop deep learning in students.

3. Systems fail to ask questions of students which might explain their actions.

4. The implementation of SMTs may be difficult to justify to an administrative staff.

5. Evaluation of an intelligent tutoring system is often difficult, costly, and time consuming.

8. APPLICATION

Shape detection is breaking into a wide scope of enterprises, with use cases extending from. individual security to efficiency in the working environment. Shape detection is applied in numerous territories of image processing, including picture retrieval, security, observation, computerized vehicle systems and machine investigation. Critical difficulties remain in the field of object detection. The potential outcomes are inestimable with regards to future use cases for object detection.

Tracking objects

A Shape detection framework is additionally utilized in tracking the shapes/objects, for instance tracking a ball during a match in the football world cup, tracking the swing of a cricket bat, tracking an individual in a video.

Object tracking has an assortment of uses, some of which are surveillance and security, traffic checking, video correspondence, robot vision and activity.

People Counting

Shape detection can be additionally utilized for People counting. It is utilized for dissecting store execution or group measurements during festivals. These will, in general, be progressively troublesome as individuals move out of the frame rapidly (likewise in light of the fact that individuals are non-inflexible objects).

Automated CCTV surveillance

Surveillance is a necessary piece of security and watch. Ongoing advances in computer vision innovation need to prompt the improvement of different programmed surveillance systems. Be that as it may, their viability is influenced by numerous factors and they are not totally dependable. This examination researched the capability of an automated surveillance system to diminish the CCTV administrator outstanding task at hand in both discovery and following exercises.

Person Detection

Person detection is necessary and critical work in any intelligent video surveillance framework, as it gives the essential data to semantic comprehension of the video recordings. It has a conspicuous augmentation to automotive applications because of the potential for improving security frameworks. Person detection is undertakings of Computer vision frameworks for finding and following individuals. Person detection is the task of finding all examples of individuals present in a picture, and it has been most broadly achieved via looking through all areas in the picture, at all potential scales, and contrasting a little region at every area with known layouts or examples of individuals. Person detection is commonly viewed as the initial procedure in a video surveillance pipeline and can take care of into more significant level thinking modules, for example, action recognition and dynamic scene analysis.

Vehicle Detection

Vehicle Detection is one of the most important part in our daily life. As the world is moving faster and the numbers of cars are kept on increasing day by day, Vehicle detection is very important. By using Vehicle Detection technique, we can detect the number plate of a speeding car or accident affected car.

9.CONCLUSION

We presented a new shape description and classification method. Key characteristics of our approach are the compound descriptor and classifier that join the region and contour-based features. We suggested an online learning method to extend the representative set and increase performance. We proposed a representative set optimizing algorithm as well.

The core idea behind our method is the two-level description and classification: for an input shape, low-level, global statistical information is extracted to roughly select the set of similar objects and to reject obviously different templates. In the second stage, local edge information is investigated to find the closest known shape but with the ability to reject the match. The refusal is based on the acceptance radius that is specified individually for every item in the representative set according to the properties of the local proximity in the feature set.

Results demonstrate a high precision rate (99.83%) and an acceptable recall rate (60.53%), which fulfil the requirements for a safety-oriented visual application processing an image flow. The reason to have lower cover is that input frames contain highly deformed shapes, which, for sake of reliability, are classified as nonrelevant inputs. The recall is acceptable, as long as a continuous input is available. Compared to other classifiers, none of the tested ones could outperform the AL-NN in precision, and the same recall could only be reproduced with significantly lower precision. If a final decision is made based on multiple input frames and multiple clues, the false-positive error can be minimized to be practically negligible.

The computation time of the descriptor (~30 ms) and the classification time (~2 ms) allow real-time recognition even on standard CPUs in computers and phones, and the architecture core of the algorithm is easily adaptable to locally connected cellular array processors.

10. FUTURE SCOPE

This report elucidates shape detection, one of the highly computational applications that has become possible in recent years. Although detecting shapes in a given image or video frame has been around for years, it is becoming more widespread across a range of industries now more than ever before.

Shape detection in images and video has received lots of attention in the computer vision and pattern recognition communities over recent years. We have had great progress in the field, processing a single image used to take 20 seconds per image and today it takes less than 20 milliseconds of the problems related to these fields, analysing an image and recognizing all shapes remains to be one of the most challenging ones.

For humans and many other animals, visual perception is one of the most important senses; we heavily rely on vision whenever we interact with our environment. In order to pick up a glass, we need to first determine which part of our visual impression corresponds to the glass before we can find out where we have to move our hands in order to grasp it.

The same code that can be used to recognize Stop signs or pedestrians in a self-driving vehicle signs can also be used to find cancer cells in a tissue biopsy.

If we want to recognize another human, we first have to find out which part of the image we see represents that individual, as well as any distinguishing factors of their face.

Notably, we generally do not actively consider these basic steps, but these steps pose a major challenge for artificial systems dealing with image processing.

Most existing algorithms only tackle a small subset of the different tasks necessary for understanding an image and are very expensive computationally. In order to reproduce a fraction of the average person’s ability to detect shapes, one would have to combine several different algorithms to make a combined system that runs in real time, an enormous challenge with today’s hardware.

Indeed, shape detection is a key task for most computer and robot vision systems. Although there has been great progress in the last several years, there will be even bigger improvements in the future with the advent of artificial intelligence in conjunction with existing techniques that are now part of many consumer electronics or have been integrated in assistant driving technologies.

However, we are still far from achieving human-level performance in open-world learning. Furthermore, shape detection has not been applied in many areas where it could be of great help. Consider for example the possibility of applications of shape detection systems to robotic excavation when venturing into previously unexplored territory, such as the deep sea or other planets, in which the detection systems will have to learn new shape classes on the job. In such cases, a real-time, open-world learning ability will be critical. This fascinating computer technology related to computer vision and image processing that detects and defines shapes, such as persons, vehicles, and animals from digital images and videos, will be incredibly important in the near future.

We have developed many methods for shape detection, but the application of deep learning promises higher accuracy for a wider variety of shape classes.
Shape detection is breaking into a wide range of industries, including computer vision, image retrieval, security, surveillance, automated vehicle systems, and machine inspection. Although the possibilities are endless when it comes to future use cases for shape detection, there are still significant challenges remaining.

Herewith are some of the main useful applications of shape detection: Vehicle's Plates recognition, self-driving cars, tracking shapes, face recognition, medical imaging, shape counting, shape extraction from an image or video, person detection.

The future of shape detection technology is in the process of proving itself, and much like the original Industrial Revolution, it has the potential to free people from menial jobs that can be done more efficiently and effectively by machines.

It will also open up new avenues of research and operations that will reap additional benefits in the future.

Thus, these challenges circumvent the need for a lot of training requiring a massive number of datasets to serve more nuanced tasks, with its continued evolution, along with the devices and techniques that make it possible, it could soon become the next big thing in the future.

11. BIBLIOGRAPHY

- https://www.geeksforgeeks.org/
- https://www.w3schools.com/
- https://stackoverflow.com/
- https://www.lucidchart.com/pages/
- https://smartinternz.com/Student/guided_project_info/4652#

APPENDIX

SOURCE CODE: Github

HTML

```
 <html>

<script>

document.body.addEventListener("mousemove", evt => {

  const mouseX = evt.clientX;

  const mouseY = evt.clientY;

   gsap.set(".cursor", {

    x: mouseX,

    y: mouseY

   })

   gsap.to(".shape", {

    x: mouseX,

    y: mouseY,

    stagger: -0.1

  })

})

</script>

<style>

body {

  font-family: Montserrat, sans-serif;

  font-weight: 900;
```

```
  &, * {
    cursor: none;
  }
}
.shapes {
  position: relative;
  height: 100vh;
  width: 100vw;
  background: #2128bd;
  overflow: hidden;
}
.shape {
  will-change: transform;
  position: absolute;
  border-radius: 50%;
   $shapes: (#005ffe: 650px, #ffe5e3: 440px, #ffcc57: 270px);
  @each $color, $size in $shapes {
  &.shape-#{index($shapes, ($color $size))} {
      background: $color;
      width: $size;
      height: $size;
      margin: (-$size/2) 0 0 (-$size/2);
    }
  }
```

```
}
.content {
  top: 0;
  left: 0;
  position: absolute;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  width: 100vw;
  background: #fff;
  mix-blend-mode: screen;
}
h1 {
  @include rf(100, 200); // defined in imported pen;
  color:#000;
margin: 0;
  text-align: center;
}
.cursor {
  position: fixed;
  background: #2128bd;
  width: 20px;
  height: 20px;
```

```
  margin: -10px 0 0 -10px;
  border-radius: 50%;
  will-change: transform;
  user-select: none;
  pointer-events: none;
  z-index: 10000;
}
</style>
::selection {
  color: #fff;
  background: #2128bd;
}
<head>
<link href="https://fonts.googleapis.com/css?family=Montserrat:900&display=swap"
rel="stylesheet">
<meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
</body>
</html>
```

APP.PY

```
 from flask import Flask,render_template
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
from tkinter import * # Graphical User Interface package
import PIL
from PIL import ImageGrab # used to copy the contents of the screen
import model # loading our model python file
import cv2  #opencv library
app = Flask(_name_) # initializing a flask app
@app.route('/')# route to display the home page
def home():
   return render_template('home1.html')#rendering the home page
@app.route('/intro')
def intro():
   return render_template('intro.html')#rendering the intro page
@app.route('/launch',methods=['GET', 'POST'])# route to show the predictions in a web
UI
def launch():
   class main:
       def _init_(self, master):
           self.master = master
           self.res = ""
```

```
        self.pre = [None, None]

        self.bs = 4.5

        self.c = Canvas(self.master,bd=3,relief="ridge", width=400, height=400,
bg='white')

        self.c.pack(side=LEFT)

        f1 = Frame(self.master, padx=5, pady=5)

        Label(f1,text="Maths Tutor for
shape",fg="red",font=("",15,"bold")).pack(pady=10)

       Label(f1,text="Draw a shape to get its formula",fg="red",font=("",15)).pack()

        Label(f1,text="(Circle,Square,Triangle)",fg="red",font=("",15)).pack()

        self.pr = Label(f1,text="Prediction: None",fg="red",font=("",20,"bold"))

        self.pr.pack(pady=20)

        Button(f1,font=("",15),fg="white",bg="red", text="Clear Canvas",

            command=self.clear).pack(side=BOTTOM)

      f1.pack(side=RIGHT,fill=Y)

        self.c.bind("<Button-1>", self.putPoint)

        self.c.bind("<ButtonRelease-1>",self.getResult)

        self.c.bind("<B1-Motion>", self.paint)

    def getResult(self,e):

        x = self.master.winfo_rootx() + self.c.winfo_x()

        y = self.master.winfo_rooty() + self.c.winfo_y()

        x1 = x + self.c.winfo_width()

        y1 = y + self.c.winfo_height()

        img = PIL.ImageGrab.grab()
```

```
img = img.crop((x, y, x1, y1))

img.save("dist.png")

self.res = str(model.predict("dist.png"))

self.pr['text'] = "Prediction: "

if self.res == 'square':

self.pr['text'] = self.pr['text'] + self.res + "\n Perimeter formula : 4 × side " + "\n Area formula: side^2"

elif self.res == 'circle':

self.pr['text'] = self.pr['text'] + self.res + "\n Perimeter formula: 2 × pi × radius " + "\n Area formula:   pi × radius2"

elif self.res == 'triangle':

self.pr['text'] = self.pr['text'] + self.res + "\n Perimeter formula: side1 + side2 + side3 " + "\n Area formula: base × height / 2"

if self.res=='circle':

image = cv2.imread('circle.png')

cv2.imshow('circle', image)

key=cv2.waitKey(0)

if (key & 0xFF) == ord("c"):

cv2.destroyWindow("circle")

elif self.res=='square':

image = cv2.imread('square.png')

cv2.imshow('square', image)

key=cv2.waitKey(0)

if (key & 0xFF) == ord("s"):
```

```
cv2.destroyWindow("square")
          else:  image = cv2.imread('triangle.png')
             cv2.imshow('triangle', image)
             key=cv2.waitKey(0)

             if (key & 0xFF) == ord("t"):
                cv2.destroyWindow("triangle")
def clear(self):
          self.c.delete('all')
  def putPoint(self, e):
          self.c.create_oval(e.x - self.bs, e.y - self.bs, e.x + self.bs, e.y + self.bs,
                        outline='black', fill='black')
          self.pre = [e.x, e.y]
   def paint(self, e):
          self.c.create_line(self.pre[0], self.pre[1], e.x, e.y, width=self.bs * 2,
                        fill='black', capstyle=ROUND,
                        smooth=TRUE)
   self.pre = [e.x, e.y]
     if _name_ == "_main_":
       root = Tk()
main(root)
       root.title('Digit Classifier')
       root.resizable(0, 0)
       root.mainloop()
```

```
    # showing the prediction results in a UI
  return render_template("home1.html")
  if _name_ == "_main_":
 # running the app
  app.run(debug=False)
```

© 20