**Project Report on**

# SMART MATHEMATICS TUTOR USING IBM WATSON

**NAME**: **PARKIBANDA NAREN CHARY**
**ROLL NO**: **19R11A0585**
**GROUP**: **CSE**
**TEAM**: **10**

## INTRODUCTION

### ***OVERVIEW:***

It is reported that many students get dissociated and bored from learning math via any mode of teaching. Many students are not interactive towards learning. 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. So, we decided to make math easier in a strategical and explainable way, rather than mugging up the problems.

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

In this project, we have given a simple math learning concept about the shapes and its related formulas information so that every student can understand the related formulas based on the given figure shape interactively.

Also, this project gives a glance knowledge about virtual learning or e - learning in mathematics.

### ***PURPOSE:***

The main purpose of doing this project is to make Math learning interactive and innovative without any boredom and make the primary students get awareness about e - learning or virtual learning.
Moreover, purposefully this project helps the students to change rote learning of mathematics to interactive learning of mathematics in any base, may be formulas or problem solving or proving something etc...

The young students can achieve brief information about the formula for a given shape rather than mugging up the same formula multiple times. This type of project makes a visual description of any learning concepts. This project makes the students run out of fear in math learning. This project helps the students to learn math interactively and also use Interactive Technology in learning these things.

## LITERATURE SURVEY

### Existing Problem:

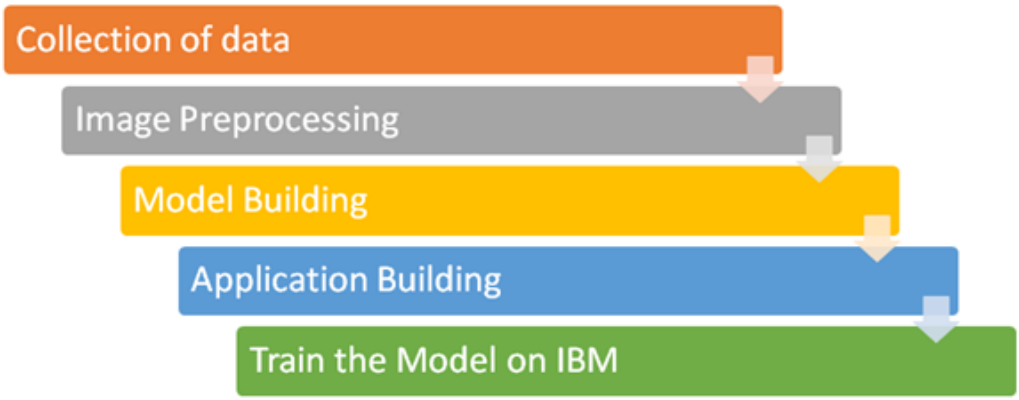
- The way students learn the formulas is by reading them from text book. They have the shape given with its formulas next to it. In this case, they will have to learn the shape names and their formulas is through by heart. It will be boring in that way. So it needs to be interactive for them and not to get bored.

### Proposed Solution:

- The way to solve this kind of problem is using the Convolution Neural Network Algorithm. In the algorithm, we build a model and using a dataset we train the model. After training, we test the model for accuracy. This way we can predict the shape drawn by the user, output the name of the shape and the related formulas to it. The user will draw a shape in the graphical user interface and get the formulas of that shape

# THEORITICAL ANALYSIS

## BLOCK DIAGRAM:




## SOFTWARE DESIGNING:

To complete this project, you must require following software's , concepts and packages

### Anaconda navigator

- Installing Anaconda Python, Jupyter Notebook And Spyder

### Python packages

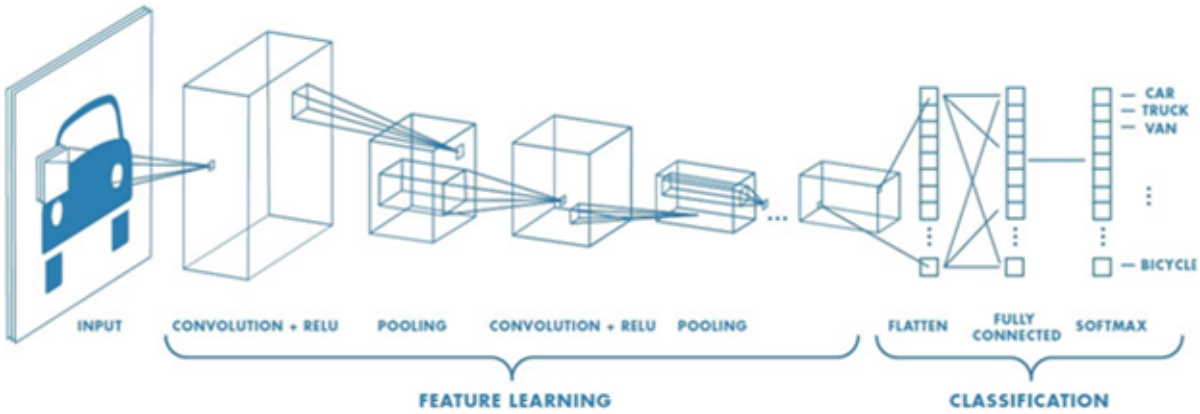
- We need to install TensorFlow, open cv-python, and flask

### Deep Learning Concepts

- **CNN**: A convolution neural network is a class of deep neural networks, most commonly applied to analysing visual imagery.
- **Opencv**: It is an Open-Source Computer Vision Library which is mainly used for image processing, video capture and analysis including features like face detection and object detection.
- **Flask**: Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications.
- **Tkinter**: Tkinter is the standard Python interface to the Tk Graphical User Interface toolkit.

# EXPERIMENTAL INVESTIGATIONS

CNN image classifications take an input image, processes it and classifies it under certain categories (Eg., Dog, Cat, Tiger, Lion). Computers see an input image as an array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d(h = Height, w = Width, d = Dimension ). Example, an image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of Gray scale image. Technically, Deep Learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply SoftMax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.




### Convolution Layer

- Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as an image matrix and a filter or kernel.

### Pooling Layer

- Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling is also called sub sampling or down sampling which reduces the dimensionalities of each map but retains important information. Spatial pooling can be of different types:

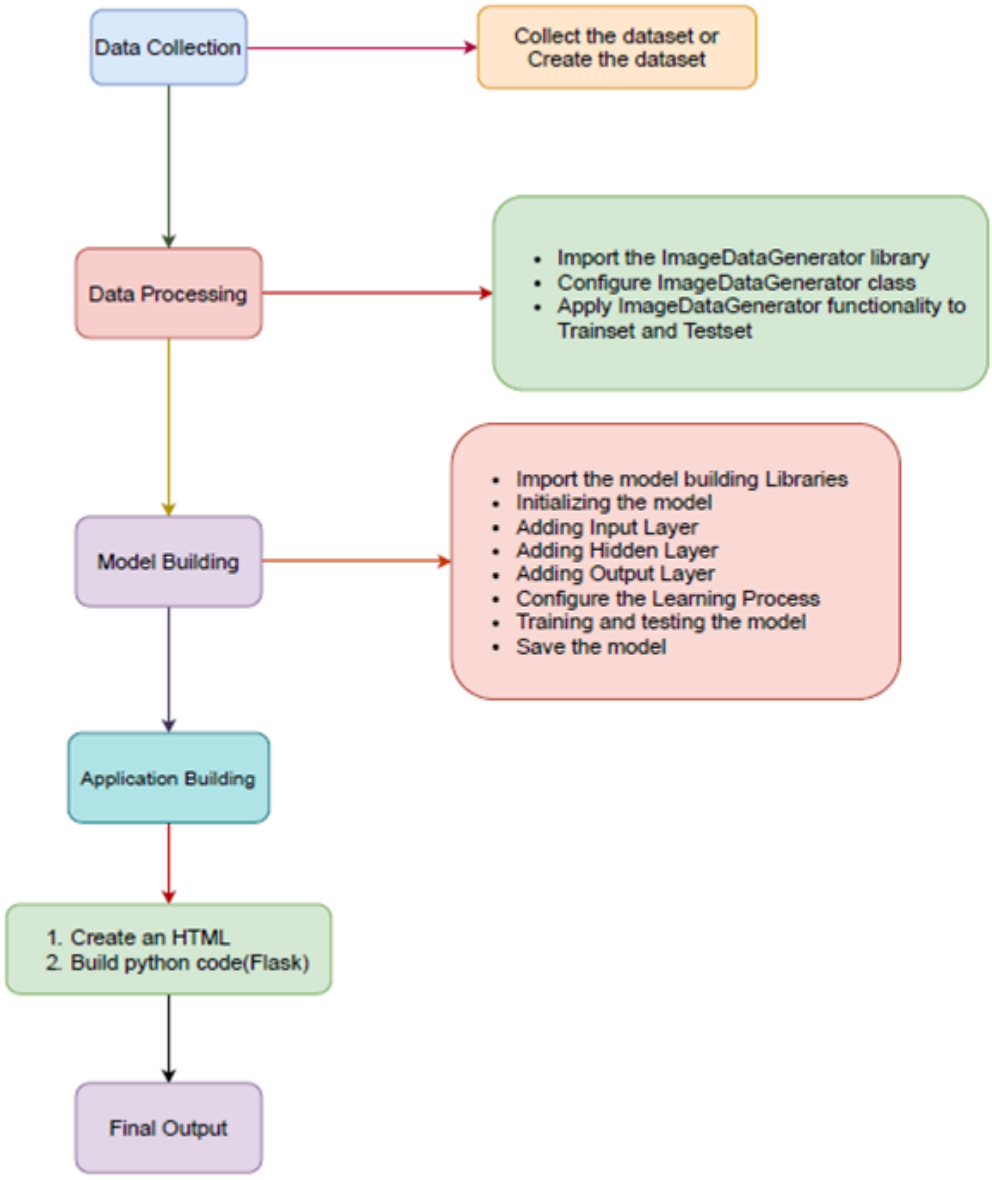
    i. Max Pooling
    ii. Average Pooling
    iii. Sum Pooling

Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. The sum of all elements in the feature map is called as sum pooling.

### Fully Connected Layer

- The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like a neural network.

## FLOWCHART



## ADVANTAGES & DISADVANTAGES

**Advantages: -**

- It is super convenient.
- Easy to learn and easy to understand.
- It is portable and concept-oriented learning.
- You can practice math on your time.

**Disadvantages: -**

- Expensive and complex to implement.
- High maintenance costs.
- Devices such as computers, digital boards, etc. need to be used.
- Highly dependent on electricity and requires proper network connectivity such as LAN, WAN, Internet, etc

## APPLICATIONS

- It creates interest to the children to understand and learn the formulas in a concept-oriented way of learning.
- It is like learning through drawing and follows give-and-take strategical learning.
- This gives your kid an interesting way to learn and improve in math.

## CONCLUSION

- Best use of technology to connect at anytime and anywhere.
- No more boredom in learning process.
- Makes use of concept-oriented learning, interactive analysis.
- Makes a great contribution in global problem like in these pandemic situations.

Smart Math Tutor project makes changes in learning strategy in every student like conceptualization based virtual learning system.

# FUTURE SCOPE

Tutors and parents were excited about the future of online tutors for several reasons:

- Touch screen technology should allow both student and tutor to make hand-written notes in real time.
- With the increasing use of smart phones and iPads/tablets it is likely that online tuition will become more ‘portable’, further enhancing its usefulness.

This project can be made more advanced in the Learning process.

# APPENDIX

## CNN SHAPE RECOGNITION:

```
In [1]: from keras.preprocessing.image import ImageDataGenerator
        Using TensorFlow backend.

In [2]: train=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
        test=ImageDataGenerator(rescale=1./255)

In [4]: x_train=train.flow_from_directory(r"C:\Users\nagoj\OneDrive\Desktop\Project\Dataset\train1",target_size=(64,64),batch_size=4,cla
        ss_mode="categorical")
        x_test=train.flow_from_directory(r"C:\Users\nagoj\OneDrive\Desktop\Project\Dataset\test1",target_size=(64,64),batch_size=3,class
        _mode="categorical")
        Found 31 images belonging to 3 classes.
        Found 9 images belonging to 3 classes.

In [5]: from keras.models import Sequential
        from keras.layers import Dense
        from keras.layers import Conv2D
        from keras.layers import MaxPooling2D
        from keras.layers import Flatten

In [6]: model=Sequential()

In [7]: model.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation="relu"))
        model.add(MaxPooling2D(2,2))
        WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:4070: The name tf.nn.max
        _pool is deprecated. Please use tf.nn.max_pool2d instead.

In [8]: model.add(Conv2D(32,(3,3),activation="relu"))
        model.add(MaxPooling2D(2,2))

In [9]: model.add(Flatten())

In [10]: model.add(Dense(units=20,activation="relu",kernel_initializer="random_uniform"))

In [11]: model.add(Dense(units=3,activation="softmax",kernel_initializer="random_uniform"))

In [12]: model.compile("adam",loss="categorical_crossentropy",metrics =["accuracy"])

In [14]: model.fit_generator(x_train,steps_per_epoch=7,epochs=25,validation_data= x_test,validation_steps=3)
         Epoch 1/25
         7/7 [==============================] - 0s 17ms/step - loss: 0.0549 - accuracy: 0.9643 - val_loss: 4.0614e-04 - val_accuracy: 1.
         0000
         Epoch 2/25
         7/7 [==============================] - 0s 37ms/step - loss: 0.0547 - accuracy: 0.9615 - val_loss: 2.3842e-07 - val_accuracy: 1.
         0000
         Epoch 3/25
         7/7 [==============================] - 0s 29ms/step - loss: 0.0256 - accuracy: 1.0000 - val_loss: 1.6331e-05 - val_accuracy: 1.
         0000
         Epoch 4/25
         7/7 [==============================] - 0s 33ms/step - loss: 0.0526 - accuracy: 0.9615 - val_loss: 7.9473e-08 - val_accuracy: 1.
         0000
         Epoch 5/25
         7/7 [==============================] - 0s 35ms/step - loss: 0.0560 - accuracy: 0.9286 - val_loss: 9.8148e-06 - val_accuracy: 1.
         0000
         Epoch 6/25
         7/7 [==============================] - 0s 24ms/step - loss: 0.0406 - accuracy: 0.9630 - val_loss: 5.8052e-05 - val_accuracy: 1.
         0000
         Epoch 7/25
         7/7 [==============================] - 0s 33ms/step - loss: 0.0819 - accuracy: 0.9259 - val_loss: 0.0000e+00 - val_accuracy: 1.
         0000
```

```
Epoch 8/25
7/7 [==============================] - 0s 11ms/step - loss: 0.0295 - accuracy: 0.9630 - val_loss: 3.5975e-04 - val_accuracy: 1.
0000
Epoch 9/25
7/7 [==============================] - 0s 22ms/step - loss: 0.0542 - accuracy: 0.9630 - val_loss: 9.0153e-05 - val_accuracy: 1.
0000
Epoch 10/25
7/7 [==============================] - 0s 35ms/step - loss: 0.0515 - accuracy: 0.9630 - val_loss: 3.2424e-05 - val_accuracy: 1.
0000
Epoch 11/25
7/7 [==============================] - 0s 29ms/step - loss: 0.0272 - accuracy: 0.9643 - val_loss: 1.5470e-04 - val_accuracy: 1.
0000
Epoch 12/25
7/7 [==============================] - 0s 27ms/step - loss: 0.0852 - accuracy: 0.9259 - val_loss: 3.1152e-05 - val_accuracy: 1.
0000
Epoch 13/25
7/7 [==============================] - 0s 32ms/step - loss: 0.0551 - accuracy: 0.9615 - val_loss: 7.6404e-05 - val_accuracy: 1.
0000
Epoch 14/25
7/7 [==============================] - 0s 23ms/step - loss: 0.0305 - accuracy: 0.9643 - val_loss: 0.0014 - val_accuracy: 1.0000
Epoch 15/25
7/7 [==============================] - 0s 42ms/step - loss: 0.0783 - accuracy: 0.9259 - val_loss: 3.1789e-07 - val_accuracy: 1.
0000
Epoch 16/25
7/7 [==============================] - 0s 17ms/step - loss: 0.0013 - accuracy: 1.0000 - val_loss: 1.0938e-04 - val_accuracy: 1.
0000
Epoch 17/25
7/7 [==============================] - 0s 17ms/step - loss: 0.0207 - accuracy: 1.0000 - val_loss: 8.1459e-06 - val_accuracy: 1.
0000
Epoch 18/25
7/7 [==============================] - 0s 26ms/step - loss: 0.0694 - accuracy: 0.9615 - val_loss: 1.4782e-05 - val_accuracy: 1.
0000
Epoch 19/25
7/7 [==============================] - 0s 36ms/step - loss: 0.1177 - accuracy: 0.9259 - val_loss: 6.0435e-05 - val_accuracy: 1.
0000
Epoch 20/25
```

```
Epoch 21/25
7/7 [==============================] - 0s 29ms/step - loss: 0.0631 - accuracy: 0.9630 - val_loss: 1.7881e-06 - val_accuracy: 1.
0000
Epoch 22/25
7/7 [==============================] - 0s 33ms/step - loss: 0.0339 - accuracy: 0.9615 - val_loss: 4.1840e-05 - val_accuracy: 1.
0000
Epoch 23/25
7/7 [==============================] - 0s 26ms/step - loss: 0.0785 - accuracy: 0.8929 - val_loss: 0.0000e+00 - val_accuracy: 1.
0000
Epoch 24/25
7/7 [==============================] - 0s 12ms/step - loss: 0.0265 - accuracy: 0.9630 - val_loss: 5.0862e-06 - val_accuracy: 1.
0000
Epoch 25/25
7/7 [==============================] - 0s 25ms/step - loss: 0.2694 - accuracy: 0.9259 - val_loss: 0.0040 - val_accuracy: 1.0000
```

Out[14]: <keras.callbacks.callbacks.History at 0x2b17aaae898>

In [13]:
```
model.save("shape.h5")
```

In [6]:
```
from keras.models import load_model
from tensorflow.keras.preprocessing import image
import cv2
import numpy as np
```

In [7]:
```
index=["circle","square","triangle"]
```

In [8]:
```
model=load_model(r"shape.h5")
```

In [11]:
```
img=image.load_img(r"dist.png",target_size = (64,64))
x=image.img_to_array(img)
print(x.shape)
x=np.expand_dims(x,axis=0)
print(x.shape)
pred = model.predict_classes(x)
print(index[pred[0]])
```

```
(64, 64, 3)
(1, 64, 64, 3)
triangle
```

## FLASK APP CODE :

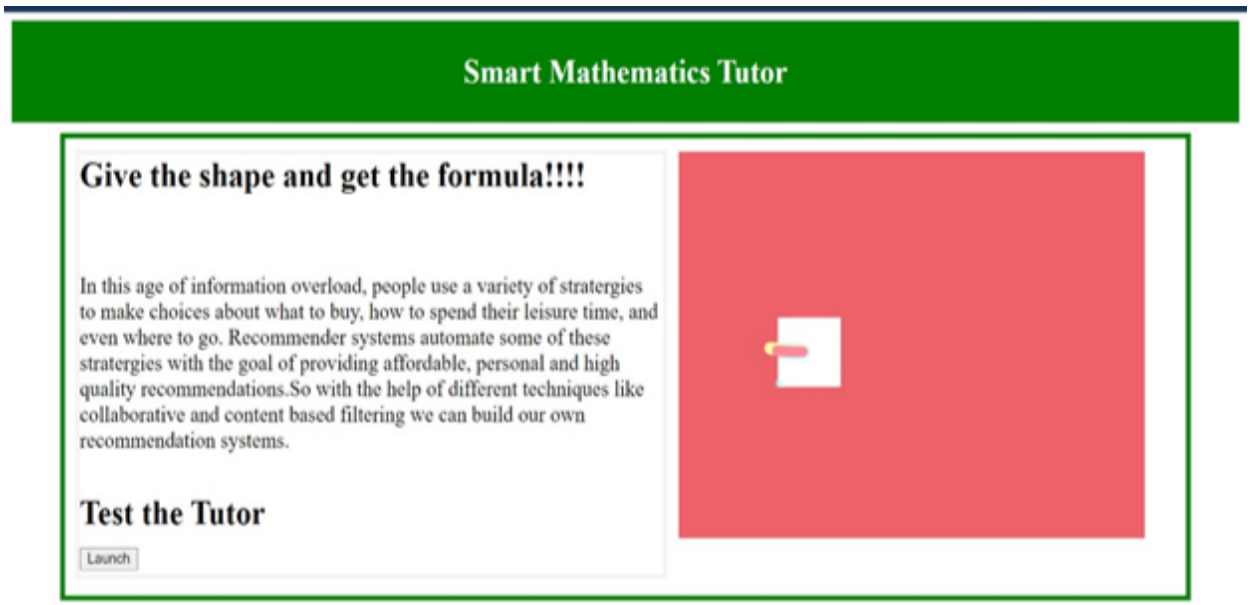
```
import tensorflow as tf
from tensorflow.python.keras.backend import set_session
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from flask import Flask,render_template
from tkinter import *
import PIL
from PIL import ImageTk,Image,ImageDraw,ImageGrab
import cv2
import tkcap
import numpy as np

app = Flask(__name__)

sess=tf.Session()
graph = tf.get_default_graph()
set_session(sess)
model = load_model(r"shape.h5")

@app.route("/",methods=["GET","POST"])
def index():
    return render_template('index.html')

@app.route('/launch',methods=['GET', 'POST'])
def launch():
    class pred:
        def __init__(self, master):
            self.master = master
            self.res = ""
            self.d=[]
            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)
            self.c.pack(expand=YES,fill=BOTH)

            f1 = Frame(self.master, padx=5, pady=5)
            Label(f1,text="Maths Tutor for shape",fg="green",font=("",15,"bold")).pack(pady=5)
            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=("",15,"bold"))
```

```
        self.pr.pack(pady=20)

        Button(f1,font=("",15),fg="white",bg="red",
text="ClearCanvas",command=self.clear).pack(side=BOTTOM)
        Button(f1,font=("",15),fg="white",bg="red",
text="Predict",command=self.getResult).pack(side=BOTTOM)
        f1.pack(side=RIGHT,fill=Y)
        self.c.bind("<Button-1>", self.putPoint)
        self.c.bind("<B1-Motion>", self.paint)

    def getResult(self):
        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()
        cap=tkcap.CAP(self.master)
        cap.capture('dist.png',overwrite=True)
        im = Image.open(r"dist.png")
        width, height = im.size
        im1 = im.crop((5,50,width-420,height))
        im1.save('dist.png')
        index=['circle','square','triangle']
        global sess
        global graph
        with graph.as_default():
            set_session(sess)
            img=image.load_img("dist.png",target_size=(64,64))
            x=image.img_to_array(img)
            x=np.expand_dims(x,axis=0)
            p=model.predict_classes(x)
            self.res=str(index[p[0]])
            self.pr['text'] = "Prediction:"+self.res

        if self.res=='circle':
            i = cv2.imread('circle.jpg')
            cv2.imshow('circle',i)

        elif self.res=='square':
            i = cv2.imread('square.jpg')
            im=cv2.resize(i,(500,400))
            cv2.imshow('square',im)

        elif self.res=='triangle':
            i = cv2.imread('triangle.png')
```

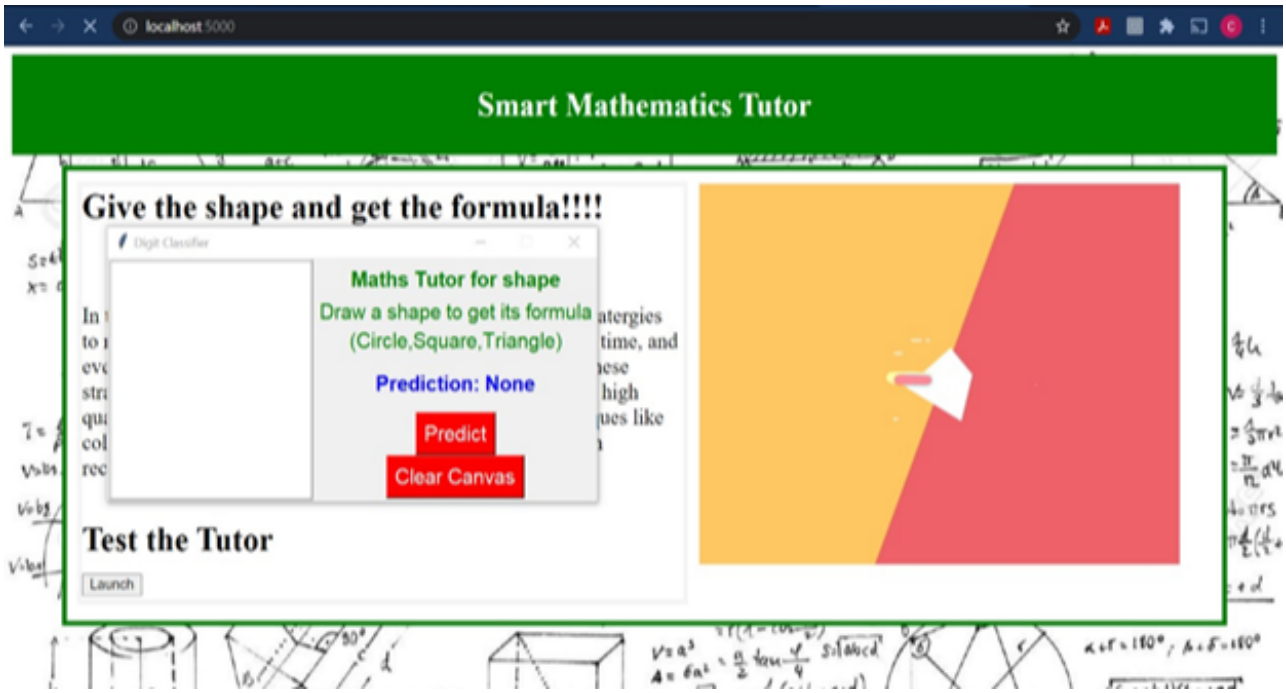
```
            im=cv2.resize(i,(500,455))
            cv2.imshow('triangle',im)

        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,sm
ooth=True)
            self.pre = [e.x, e.y]

    if __name__ == "__main__":
        root = Tk()
        pred(root)
        root.title('Digit Classifier')
        root.resizable(0, 0)
        root.mainloop()
    return render_template("index.html")

if __name__=='__main__':
    app.run(debug=True)
```

## RESULT

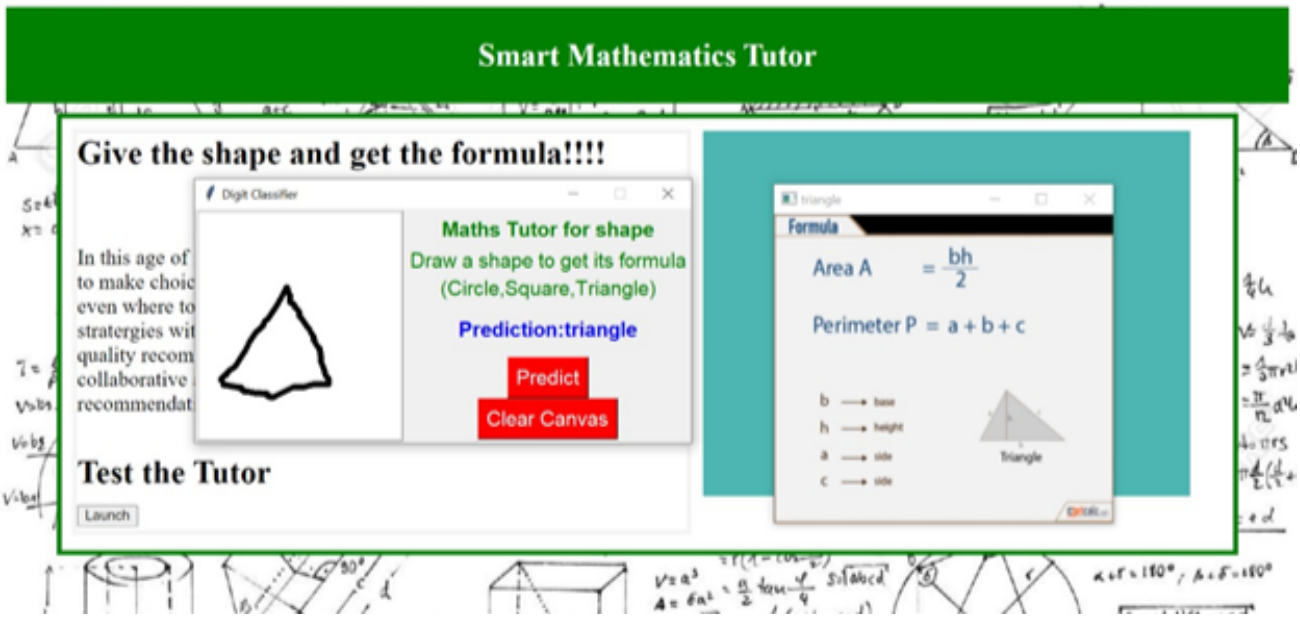
*Introduction page:*




*Launch page:*



*Output page:*


Smart Mathematics Tutor
Give the shape and get the formula!!!!
In this age of
to make choic
even where to
stratergies wit
quality recom
collaborative
recommendat
Digit Classifier
Maths Tutor for shape
Draw a shape to get its formula
(Circle,Square,Triangle)
Prediction:triangle
Predict
Clear Canvas
Test the Tutor
Launch
triangle
Formula
Area A = bh/2
Perimeter P = a + b + c
b → base
h → height
a → side
c → side
Triangle