# Project Report

## Major Project Report Submitted in fulfilment of the degree of B.TECH

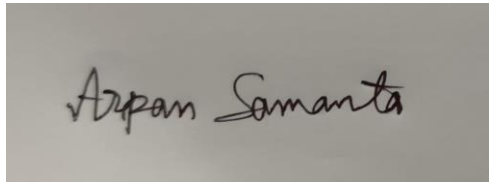
## Project Name: HANDWRITTEN DIGIT RECOGNITION ON MNIST DATASET

By --
**Sajeeb Ghosh(TL), Roll No – 11900121183**
Rohit Deb, Roll No –11900121172
Rishiraj Sarkar, Roll No –11900121182
Prasenjit Sinha, Roll No—11900121177
Priyam Biswas, Roll No– 11900121173

5th Semester students of
SILIGURI INSTITUTE OF TECHNOLOGY under
Maulana Abul Kalam Azad University (MAKAUT)

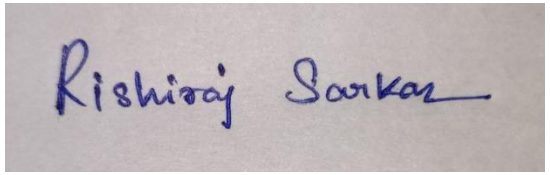
Under the supervision of
**Mr. ARPAN SAMANTA**

I hereby forward the documentation prepared under my supervision by Rishiraj Sarkar entitled Siliguri Institute Of technology be accepted as fulfilment of the requirement for the Degree of Master degree from Siliguri Institute Of Technology affiliated.



**Mr. ARPAN SAMANTA**

**(Sikharthy Infotech pvt. Ltd)**



**Rishiraj Sarkar**

## INTRODUCTION OF MACHINE LEARNING

Machine learning is the process of making systems that learn and improve by themselves, by being specifically programmed.

The ultimate goal of machine learning is to design algorithms that automatically help a system gather data and use that data to learn more. Systems are expected to look for patterns in the data collected and use them to make vital decisions for themselves.

In general, machine learning is getting systems to think and act like humans, show human-like intelligence, and give them a brain. In the real world, there are existing machine learning models capable of tasks like :

- Separating spam from actual emails, as seen in Gmail
- Correcting grammar and spelling mistakes, as seen in autocorrect

Thanks to machine learning, the world has also seen design systems capable of exhibiting uncanny human-like thinking, which performs tasks like:

- Object and image recognition
- Detecting fake news
- Understanding written or spoken words
- Bots on websites that interact with humans, like humans

**STEPS OF MACHINE LEARNING**:

1. Collecting Data:

As you know, machines initially learn from the data that you give them. It is of the utmost importance to collect reliable data so that your machine learning model can find the correct patterns. The quality of the data that you feed to the machine will determine how accurate your model is. If you have incorrect or outdated data, you will have wrong outcomes or predictions which are not relevant.

2. Preparing the Data:

After you have your data, you have to prepare it. You can do this by :

•Putting together all the data you have and randomizing it. This helps make sure that data is evenly distributed, and the ordering does not affect the learning process.

•Cleaning the data to remove unwanted data, missing values, rows, and columns, duplicate values, data type conversion, etc. You might even have to restructure the dataset and change the rows and columns or index of rows and columns.

•Visualize the data to understand how it is structured and understand the relationship between various variables and classes present.

•Splitting the cleaned data into two sets - a training set and a testing set. The training set is the set your model learns from. A testing set is used to check the accuracy of your model after training

### 3. Choosing a Model:

A machine learning model determines the output you get after running a machine learning algorithm on the collected data. It is important to choose a model which is relevant to the task at hand. Over the years, scientists and engineers developed various models suited for different tasks like speech recognition, image recognition, prediction, etc. Apart from this, you also have to see if your model is suited for numerical or categorical data and choose accordingly.

### 4. Training the Model:

Training is the most important step in machine learning. In training, you pass the prepared data to your machine learning model to find patterns and make predictions. It results in the model learning from the data so that it can accomplish the task set. Over time, with training, the model gets better at predicting.

### 5. Evaluating the Model:

After training your model, you have to check to see how it’s performing. This is done by testing the performance of the model on previously unseen data. The unseen data used is the testing set that you split our data into earlier. If testing was done on the same data which is used for training, you will not get an accurate measure, as the model is already used to the data, and finds the same patterns in it, as it previously did. This will give you disproportionately high accuracy.

When used on testing data, you get an accurate measure of how your model will perform and its speed.

6. Parameter Tuning:

Once you have created and evaluated your model, see if its accuracy can be improved in any way. This is done by tuning the parameters present in your model. Parameters are the variables in the model that the programmer generally decides. At a particular value of your parameter, the accuracy will be the maximum. Parameter tuning refers to finding these values.

7. Making Predictions

In the end, you can use your model on unseen data to make predictions accurately.

EXPLORER
GIT
python project.py 3, U

Get Started | Untitled-1.ipynb | python project.py 3, U | Plots

+ Code + Markdown | Run All | Clear Outputs of All Cells | Restart | Interrupt | Variables | Outline | Python 3.9.7 64-bit

```
import numpy as np
import matplotlib.pyplot as plt
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Conv2D, MaxPool2D, Flatten,Dropout
(X_train,y_train),(X_test,y_test)=mnist.load_data()
X_train.shape,y_train.shape,X_train.shape,y_test.shape

def plot_input_img(i):
    plt.imshow(X_train[i],cmap="binary")
    plt.title(y_train[i])
    plt.show()

for i in range(10):
    plot_input_img(i)
```

[ ] Python

```
X_train=X_train.astype(np.float32)/255
X_test=X_test.astype(np.float32)/255

X_train=np.expand_dims(X_train,-1)
X_test=np.expand_dims(X_test,-1)

y_train=keras.utils.to_categorical(y_train)
y_test=keras.utils.to_categorical(y_test)
```

[40] ✓ 1.2s Python

OUTLINE
TIMELINE

File Edit Selection View Go Run Terminal Help • Untitled-1.ipynb - git - Visual Studio Code

EXPLORER

GIT

python project.py 3, U

Get Started | Untitled-1.ipynb | python project.py 3, U | Plots

+ Code + Markdown | Run All | Clear Outputs of All Cells | Restart | Interrupt | Variables | Outline | Python 3.9.7 64-bit

```
model1.summary()
```
[ ] Python

```
model1.compile(optimizer='adam',loss=keras.losses.categorical_crossentropy,metrics=['accuracy'])
```
[43] ✓ 0.1s Python

```
from keras.callbacks import EarlyStopping,ModelCheckpoint
es=EarlyStopping(monitor='val_acc',min_delta=0.01,patience=4,verbose=1)
mc=ModelCheckpoint("./bestmodel.h5",monitor="val_acc",verbose=1,save_best_only=True)
cb=[es,mc]
```
[44] ✓ 0.1s Python

```
his=model1.fit(X_train,y_train,epochs=5,validation_split=0.3,callbacks=cb)
```
[ ] Python

```
model1_S=keras.models.load_model("E://python_project//python projects//bestmodel.h5")
```
[ ] Python

```
score=model1_S.evaluate(X_test,y_test)
print(f"the model accuracy is {score[1]}")
```
[ ] Python

[ ] Python

OUTLINE

TIMELINE

master* 0 3 Jupyter Server: local Cell 9 of 10

Type here to search 27°C ENG 11:14 PM 14-09-2022

File Edit Selection View Go Run Terminal Help • python project.py - git - Visual Studio Code

EXPLORER

GIT

python project.py 3, U

Get Started | Untitled-1.ipynb | python project.py 3, U | Plots

python project.py > ...

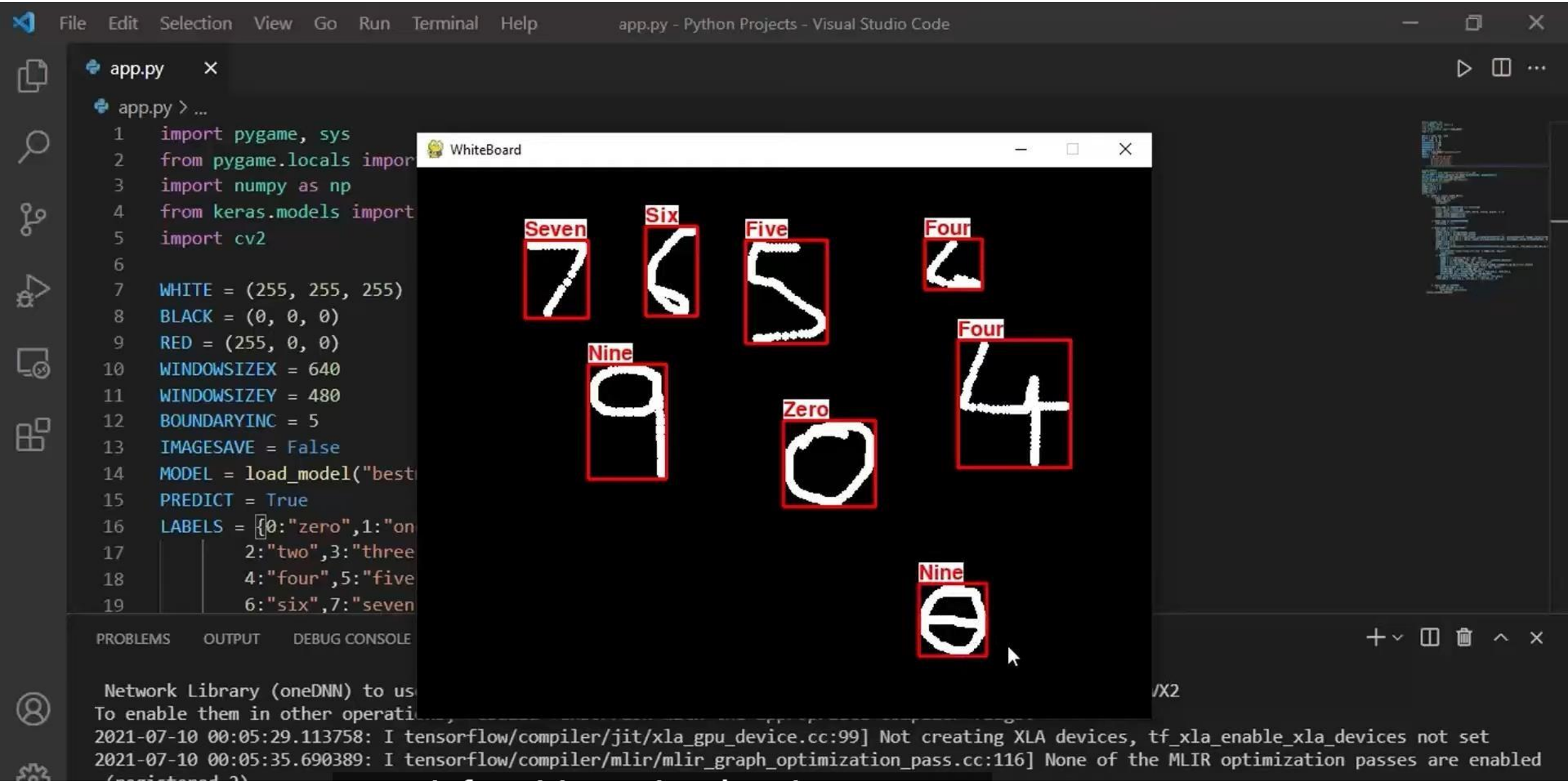
```python
from multiprocessing.dummy.connection import families
import pygame,sys
from pygame.locals import *
import numpy as np
from keras.models import load_model
import cv2

WINDOWSIZEX=640
WINDOWSIZEW=480
WHITE=(255,255,255)
BLACK=(0,0,0)
RED=(255,0,0)
IMAGESAVE=False
MODEL=load_model("bestmodel.h5")
LABELS={0:"zero",1:"one",2:"two",
        3:"three",4:"four",5:"five",
        6:"six",7:"seven",8:"eight",
        9:"nine"}

pygame.init()

DISPLAYSURF=pygame.display.set_mode((WINDOWSIZEX,WINDOWSIZEW))
pygame.display.set_caption("Digit Board")
iswriting=False
number_xcord=[]
number_ycord=[]

while True:
    for event in pygame.event.get():
        if event.type==QUIT:
            pygame.quit()
            sys.exit()

if event.type==MOUSEMOTION and iswriting:
    xcord,ycord=event.pos
    pygame.draw.circle(DISPLAYSURF,WHITE(xcord,ycord),4,0)
    number_xcord.append(xcord)
    number_ycord.append(ycord)
```

OUTLINE

TIMELINE

master* ⊗0 ⚠3 Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.9.7 64-bit

Type here to search 27°C ENG 11:14 PM 14-09-2022


File Edit Selection View Go Run Terminal Help
app.py - Python Projects - Visual Studio Code
app.py
app.py > ...
1 import pygame, sys
2 from pygame.locals impor
3 import numpy as np
4 from keras.models import
5 import cv2
6
7 WHITE = (255, 255, 255)
8 BLACK = (0, 0, 0)
9 RED = (255, 0, 0)
10 WINDOWSIZEX = 640
11 WINDOWSIZEY = 480
12 BOUNDARYINC = 5
13 IMAGESAVE = False
14 MODEL = load_model("best
15 PREDICT = True
16 LABELS = {0:"zero",1:"on
17 2:"two",3:"three
18 4:"four",5:"five
19 6:"six",7:"seven
WhiteBoard
Seven
Six
Five
Four
Nine
Four
Zero
Nine
PROBLEMS
OUTPUT
DEBUG CONSOLE
Network Library (oneDNN) to us
To enable them in other operati
2021-07-10 00:05:29.113758: I tensorflow/compiler/jit/xla_gpu_device.cc:99] Not creating XLA devices, tf_xla_enable_xla_devices not set
2021-07-10 00:05:35.690389: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR optimization passes are enabled

# Thank you