

Case Study

- 1. Name : Vaibhav Bichave
- 2. class : BE 10 (IT)
- 3. Roll No.: 43209

Problem Statement :

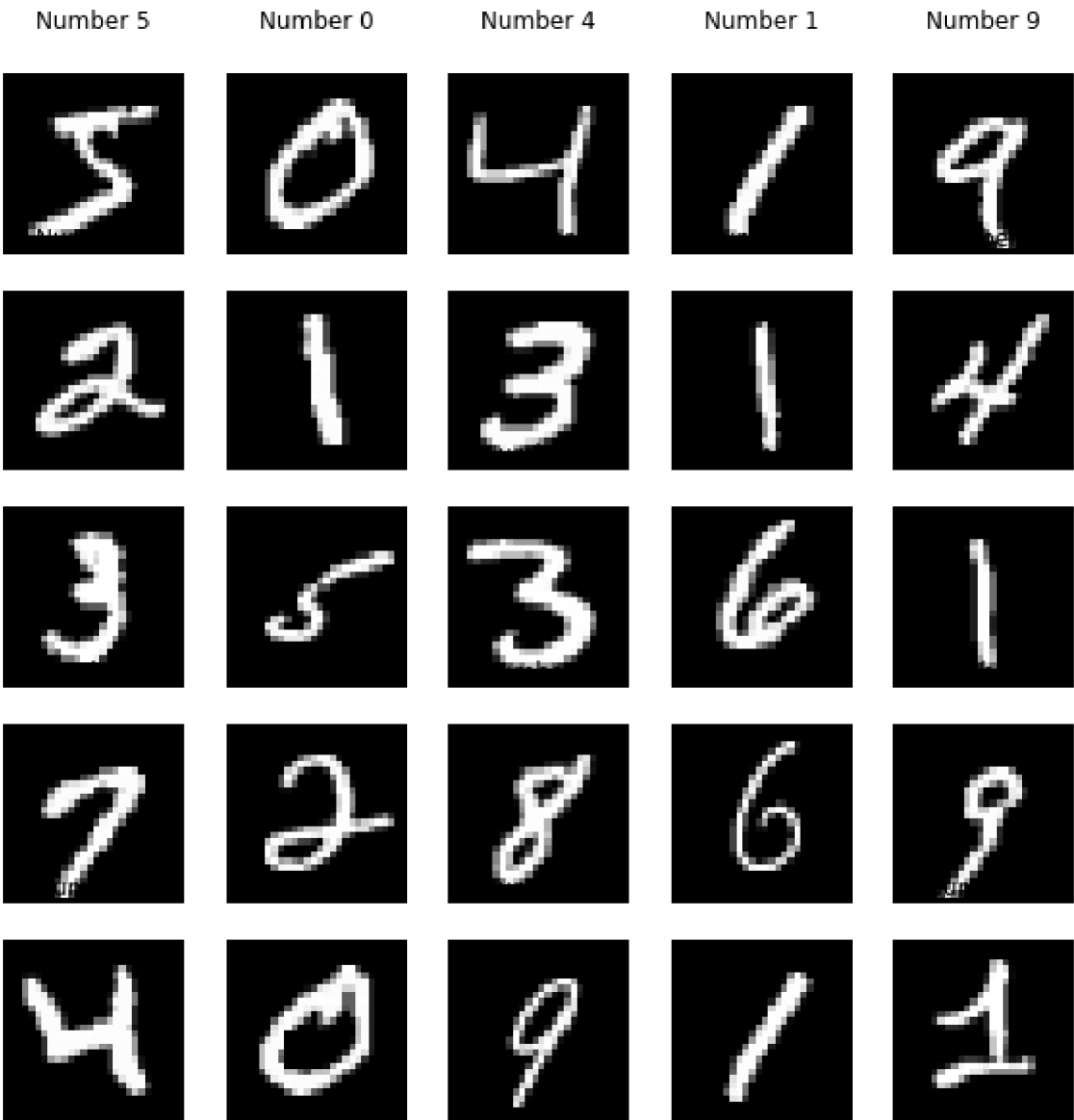
Recognize optical character using ANN

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings
%matplotlib inline
import tensorflow as tf
```

```
In [2]: (x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
x_test.shape
```

Out[2]: (10000, 28, 28)

```
In [3]: fig, axs = plt.subplots(5, 5, figsize=(10,10))
plt.gray()
for i,a in enumerate(axs.flat):
    a.matshow(x_train[i])
    a.axis('off')
    a.set_title('Number {}'.format(y_train[i]))
```



```
In [4]: x_train = x_train.reshape(x_train.shape[0], 28,28,1)
x_test = x_test.reshape(x_test.shape[0], 28,28,1)
inputshape=(28,28,1)

In [5]: x_train=x_train.astype('float32')
x_test=x_test.astype('float32')
x_train/=255
x_test/=255

In [6]: from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooli
model = Sequential()
model.add(Conv2D(28, kernel_size=(3,3), input_shape=inputshape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation=tf.nn.relu))
model.add(Dropout(0.2))
model.add(Dense(10,activation=tf.nn.softmax))

In [7]: model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metri
model.fit(x=x_train,y=y_train, epochs=10)
```

```
Epoch 1/10
1875/1875 [=====] - 32s 17ms/step - loss: 0.2066 -
accuracy: 0.9382
Epoch 2/10
1875/1875 [=====] - 33s 17ms/step - loss: 0.0816 -
accuracy: 0.9748
Epoch 3/10
1875/1875 [=====] - 32s 17ms/step - loss: 0.0569 -
accuracy: 0.9821
Epoch 4/10
1875/1875 [=====] - 33s 18ms/step - loss: 0.0431 -
accuracy: 0.9867
Epoch 5/10
1875/1875 [=====] - 37s 20ms/step - loss: 0.0351 -
accuracy: 0.9881
Epoch 6/10
1875/1875 [=====] - 33s 17ms/step - loss: 0.0275 -
accuracy: 0.9908
Epoch 7/10
1875/1875 [=====] - 34s 18ms/step - loss: 0.0267 -
accuracy: 0.9912
Epoch 8/10
1875/1875 [=====] - 32s 17ms/step - loss: 0.0212 -
accuracy: 0.9927
Epoch 9/10
1875/1875 [=====] - 33s 17ms/step - loss: 0.0176 -
accuracy: 0.9937
Epoch 10/10
1875/1875 [=====] - 32s 17ms/step - loss: 0.0175 -
accuracy: 0.9938
```

Out[7]: <keras.callbacks.History at 0x8188580>

```
In [8]: scores = model.evaluate(x_test, y_test)
print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))

313/313 [=====] - 2s 6ms/step - loss: 0.0649 - accu
racy: 0.9852

accuracy: 98.52%
```

```
In [9]: !pip install ann_visualizer

Requirement already satisfied: ann_visualizer in c:\users\admin\appdata\loca
l\programs\python\python38\lib\site-packages (2.5)
```

```
In [18]: from ann_visualizer.visualize import ann_viz;
ann_viz(model, title="Neural Network")
```

