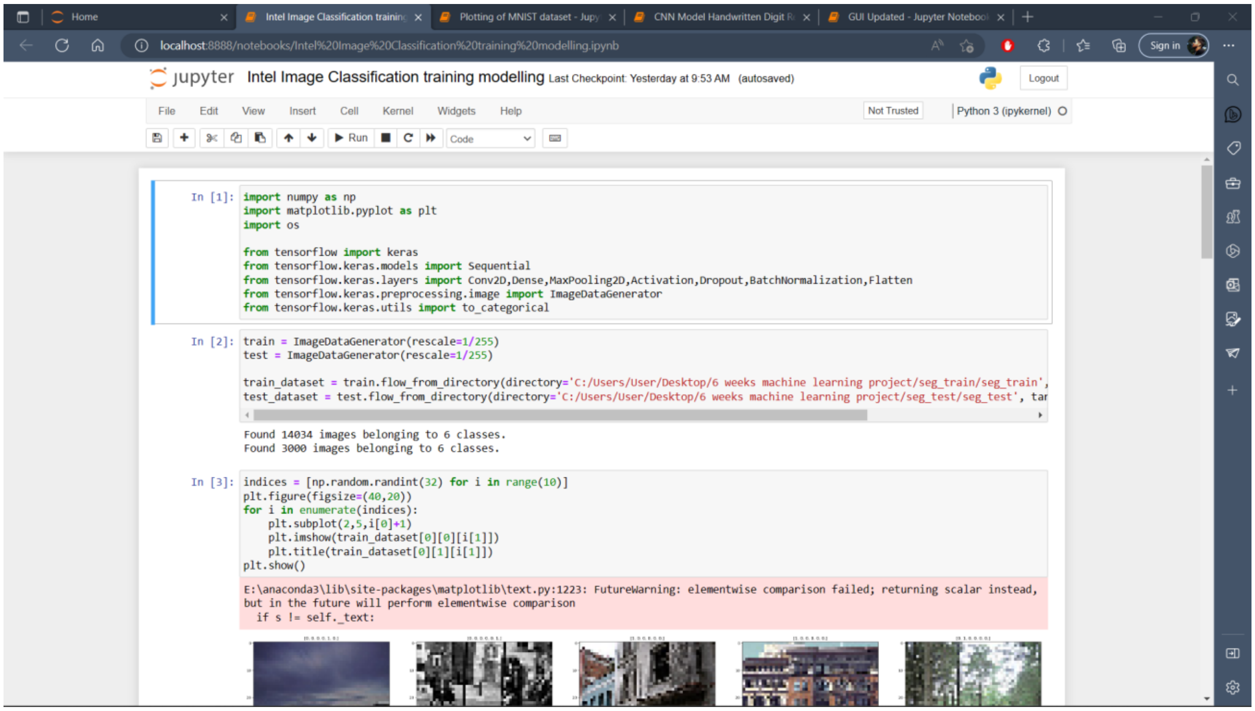
ML PROJECT BATCH 2

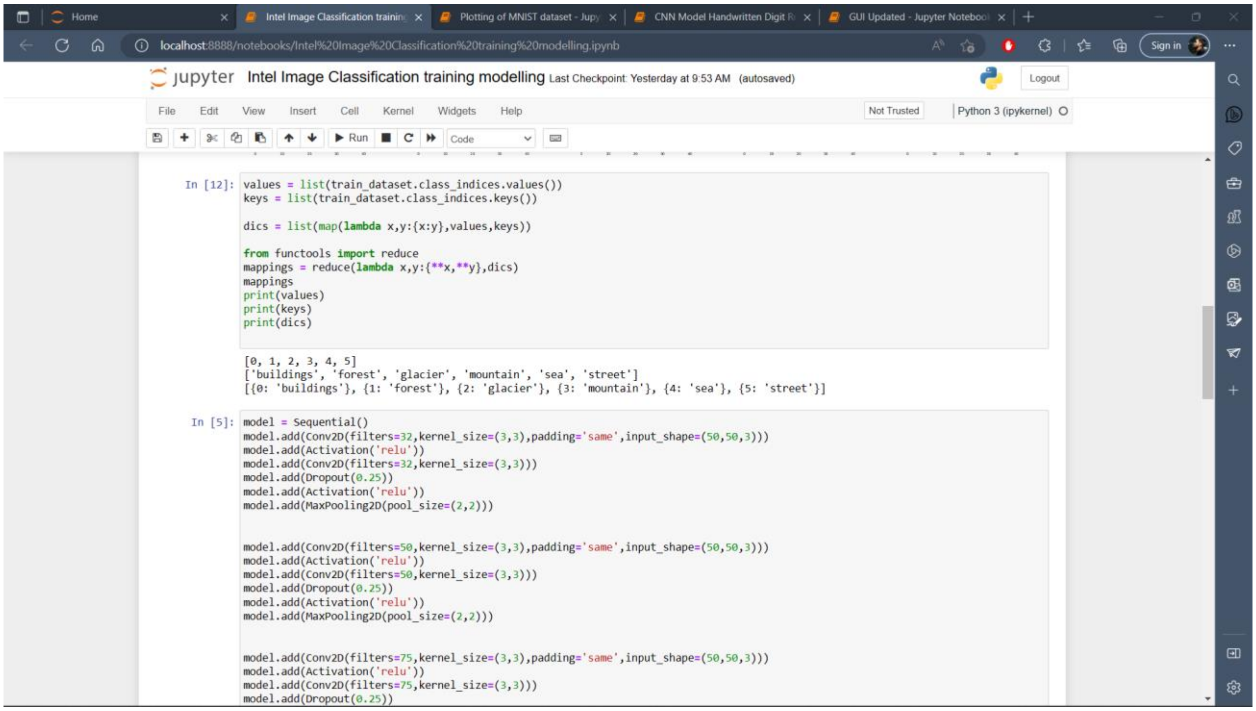
PROJECT TITLE: HANDWRITTEN DIGIT RECOGNITION USING DEEP LEARNING

SCREENSHOTS OF OUTPUTS:

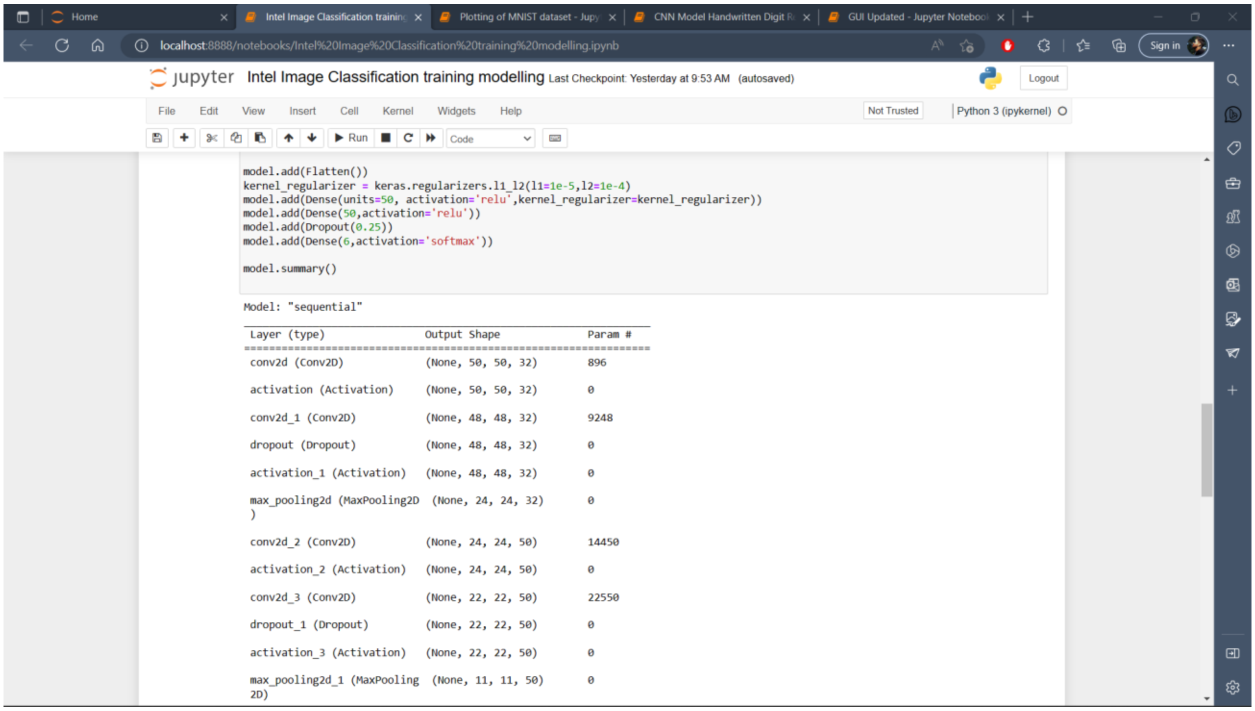
1. Intel Image Classification Training Model


Home
Intel Image Classification trainin
Plotting of MNIST dataset - Jupy
CNN Model Handwritten Digit R
GUI Updated - Jupyter Noteboo
localhost:8888/notebooks/Intel%20Image%20Classification%20training%20modelling.ipynb
Sign in
jupyter
Intel Image Classification training modelling Last Checkpoint: Yesterday at 9:53 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import os

        from tensorflow import keras
        from tensorflow.keras.models import Sequential
        from tensorflow.keras.layers import Conv2D,Dense,MaxPooling2D,Activation,Dropout,BatchNormalization,Flatten
        from tensorflow.keras.preprocessing.image import ImageDataGenerator
        from tensorflow.keras.utils import to_categorical

In [2]: train = ImageDataGenerator(rescale=1/255)
        test = ImageDataGenerator(rescale=1/255)

        train_dataset = train.flow_from_directory(directory='C:/Users/User/Desktop/6 weeks machine learning project/seg_train/seg_train',
        test_dataset = test.flow_from_directory(directory='C:/Users/User/Desktop/6 weeks machine learning project/seg_test/seg_test', tar

        Found 14034 images belonging to 6 classes.
        Found 3000 images belonging to 6 classes.

In [3]: indices = [np.random.randint(32) for i in range(10)]
        plt.figure(figsize=(40,20))
        for i in enumerate(indices):
            plt.subplot(2,5,i[0]+1)
            plt.imshow(train_dataset[0][0][i[1]])
            plt.title(train_dataset[0][1][i[1]])
        plt.show()

        E:\anaconda3\lib\site-packages\matplotlib\text.py:1223: FutureWarning: elementwise comparison failed; returning scalar instead,
        but in the future will perform elementwise comparison
          if s != self._text:
```

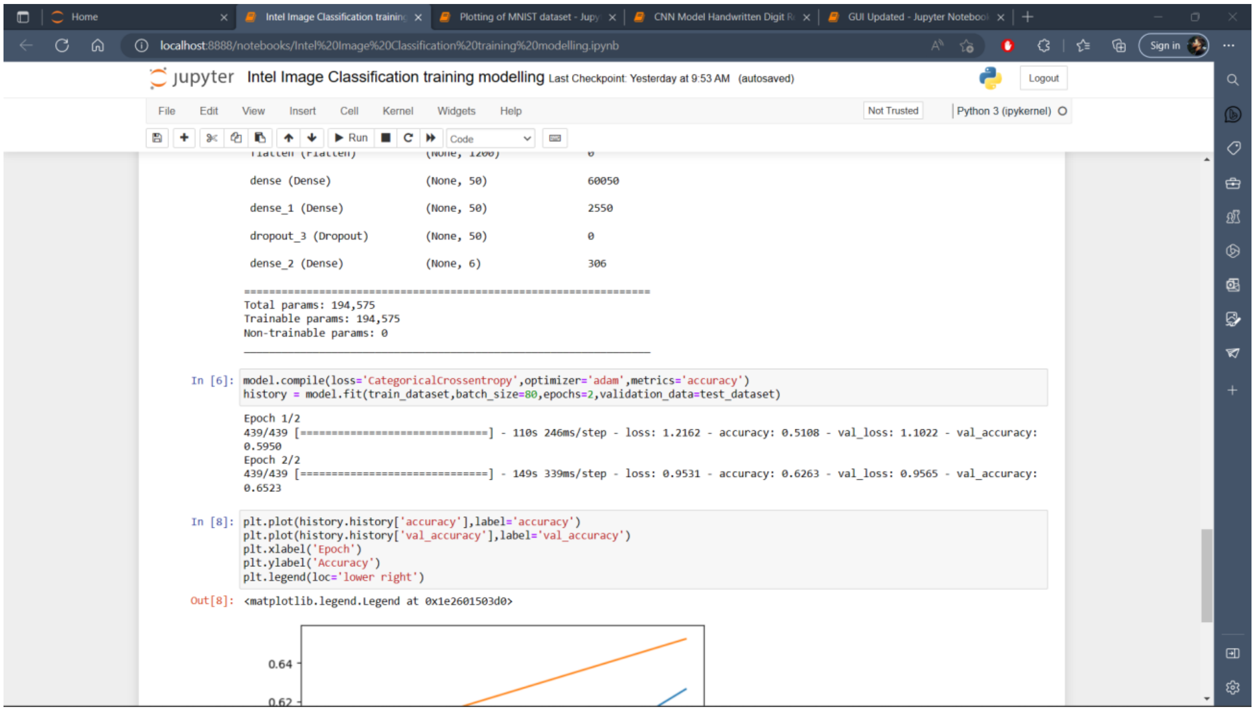

Home
Intel Image Classification trainin
Plotting of MNIST dataset - Jupy
CNN Model Handwritten Digit R
GUI Updated - Jupyter Noteboo
localhost:8888/notebooks/Intel%20Image%20Classification%20training%20modelling.ipynb
Sign in
jupyter
Intel Image Classification training modelling Last Checkpoint: Yesterday at 9:53 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
In [12]: values = list(train_dataset.class_indices.values())
         keys = list(train_dataset.class_indices.keys())

         dics = list(map(lambda x,y:{x:y},values,keys))

         from functools import reduce
         mappings = reduce(lambda x,y:{**x,**y},dics)
         mappings
         print(values)
         print(keys)
         print(dics)

         [0, 1, 2, 3, 4, 5]
         ['buildings', 'forest', 'glacier', 'mountain', 'sea', 'street']
         [{0: 'buildings'}, {1: 'forest'}, {2: 'glacier'}, {3: 'mountain'}, {4: 'sea'}, {5: 'street'}]

In [5]: model = Sequential()
        model.add(Conv2D(filters=32,kernel_size=(3,3),padding='same',input_shape=(50,50,3)))
        model.add(Activation('relu'))
        model.add(Conv2D(filters=32,kernel_size=(3,3)))
        model.add(Dropout(0.25))
        model.add(Activation('relu'))
        model.add(MaxPooling2D(pool_size=(2,2)))


        model.add(Conv2D(filters=50,kernel_size=(3,3),padding='same',input_shape=(50,50,3)))
        model.add(Activation('relu'))
        model.add(Conv2D(filters=50,kernel_size=(3,3)))
        model.add(Dropout(0.25))
        model.add(Activation('relu'))
        model.add(MaxPooling2D(pool_size=(2,2)))


        model.add(Conv2D(filters=75,kernel_size=(3,3),padding='same',input_shape=(50,50,3)))
        model.add(Activation('relu'))
        model.add(Conv2D(filters=75,kernel_size=(3,3)))
        model.add(Dropout(0.25))
```


Home
Intel Image Classification trainin
Plotting of MNIST dataset - Jupy
CNN Model Handwritten Digit R
GUI Updated - Jupyter Noteboo
localhost:8888/notebooks/Intel%20Image%20Classification%20training%20modelling.ipynb
Sign in
jupyter
Intel Image Classification training modelling Last Checkpoint: Yesterday at 9:53 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
model.add(Flatten())
kernel_regularizer = keras.regularizers.l1_l2(l1=1e-5,l2=1e-4)
model.add(Dense(units=50, activation='relu',kernel_regularizer=kernel_regularizer))
model.add(Dense(50,activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(6,activation='softmax'))

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 50, 50, 32)        896

 activation (Activation)     (None, 50, 50, 32)        0

 conv2d_1 (Conv2D)           (None, 48, 48, 32)        9248

 dropout (Dropout)           (None, 48, 48, 32)        0

 activation_1 (Activation)   (None, 48, 48, 32)        0

 max_pooling2d (MaxPooling2D  (None, 24, 24, 32)       0
 )

 conv2d_2 (Conv2D)           (None, 24, 24, 50)        14450

 activation_2 (Activation)   (None, 24, 24, 50)        0

 conv2d_3 (Conv2D)           (None, 22, 22, 50)        22550

 dropout_1 (Dropout)         (None, 22, 22, 50)        0

 activation_3 (Activation)   (None, 22, 22, 50)        0

 max_pooling2d_1 (MaxPooling  (None, 11, 11, 50)       0
 2D)
```


Home
Intel Image Classification trainin
Plotting of MNIST dataset - Jupy
CNN Model Handwritten Digit R
GUI Updated - Jupyter Noteboo
localhost:8888/notebooks/Intel%20Image%20Classification%20training%20modelling.ipynb
Sign in
jupyter
Intel Image Classification training modelling Last Checkpoint: Yesterday at 9:53 AM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
0.64
0.62

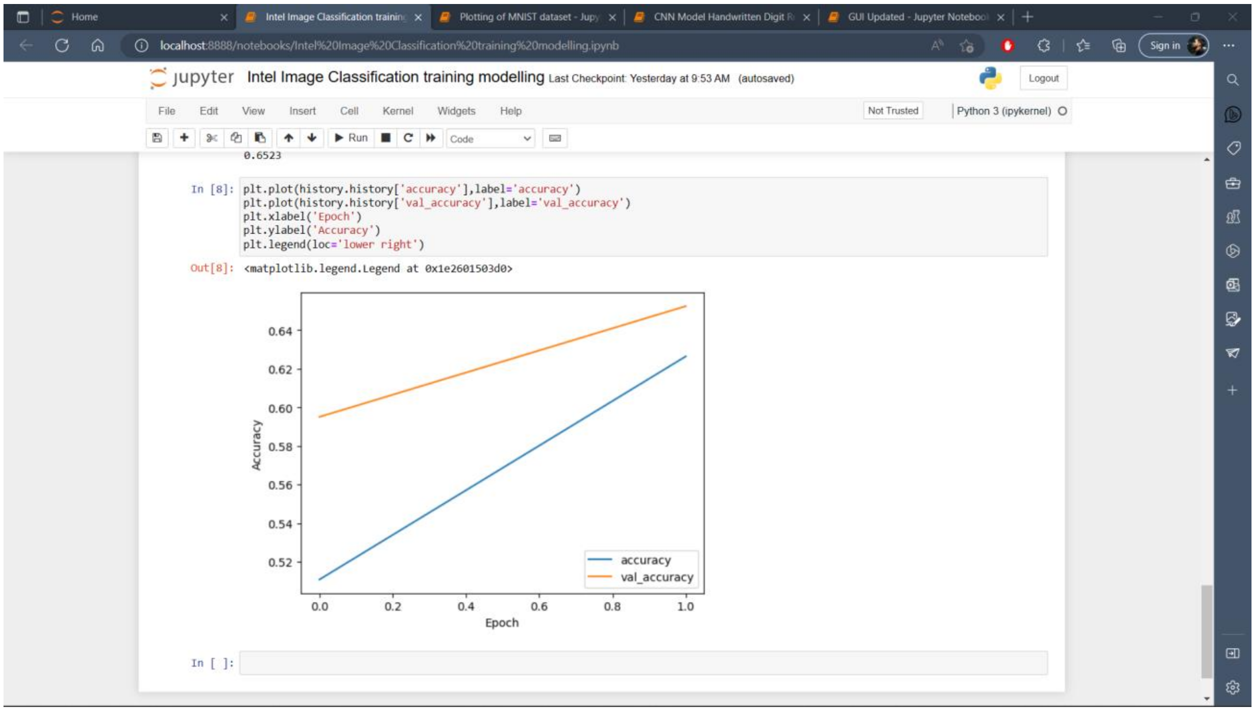
```
 flatten (Flatten)           (None, 1200)              0

 dense (Dense)               (None, 50)                60050

 dense_1 (Dense)             (None, 50)                2550

 dropout_3 (Dropout)         (None, 50)                0

 dense_2 (Dense)             (None, 6)                 306

=================================================================
Total params: 194,575
Trainable params: 194,575
Non-trainable params: 0
_________________________________________________________________
```

In [6]:
```
model.compile(loss='CategoricalCrossentropy',optimizer='adam',metrics='accuracy')
history = model.fit(train_dataset,batch_size=80,epochs=2,validation_data=test_dataset)
```

```
Epoch 1/2
439/439 [==============================] - 110s 246ms/step - loss: 1.2162 - accuracy: 0.5108 - val_loss: 1.1022 - val_accuracy:
0.5950
Epoch 2/2
439/439 [==============================] - 149s 339ms/step - loss: 0.9531 - accuracy: 0.6263 - val_loss: 0.9565 - val_accuracy:
0.6523
```

In [8]:
```
plt.plot(history.history['accuracy'],label='accuracy')
plt.plot(history.history['val_accuracy'],label='val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(loc='lower right')
```

Out[8]:
```
<matplotlib.legend.Legend at 0x1e2601503d0>
```

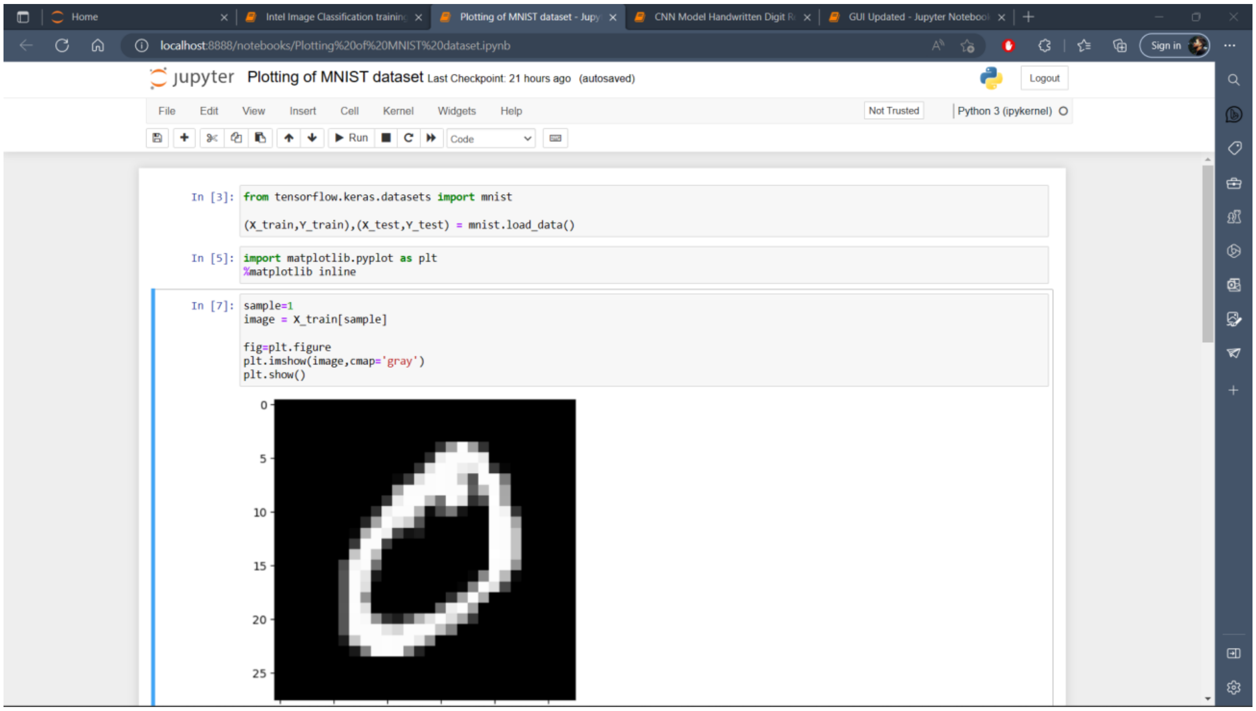

Home
Intel Image Classification training
Plotting of MNIST dataset - Jupy
CNN Model Handwritten Digit R
GUI Updated - Jupyter Noteboo
localhost:8888/notebooks/Intel%20Image%20Classification%20training%20modelling.ipynb
Sign in
jupyter
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
0.64
0.62
0.60
0.58
0.56
0.54
0.52
Accuracy
0.0
0.2
0.4
0.6
0.8
1.0
Epoch
accuracy
val_accuracy

Intel Image Classification training modelling Last Checkpoint: Yesterday at 9:53 AM (autosaved)

```
0.6523
```

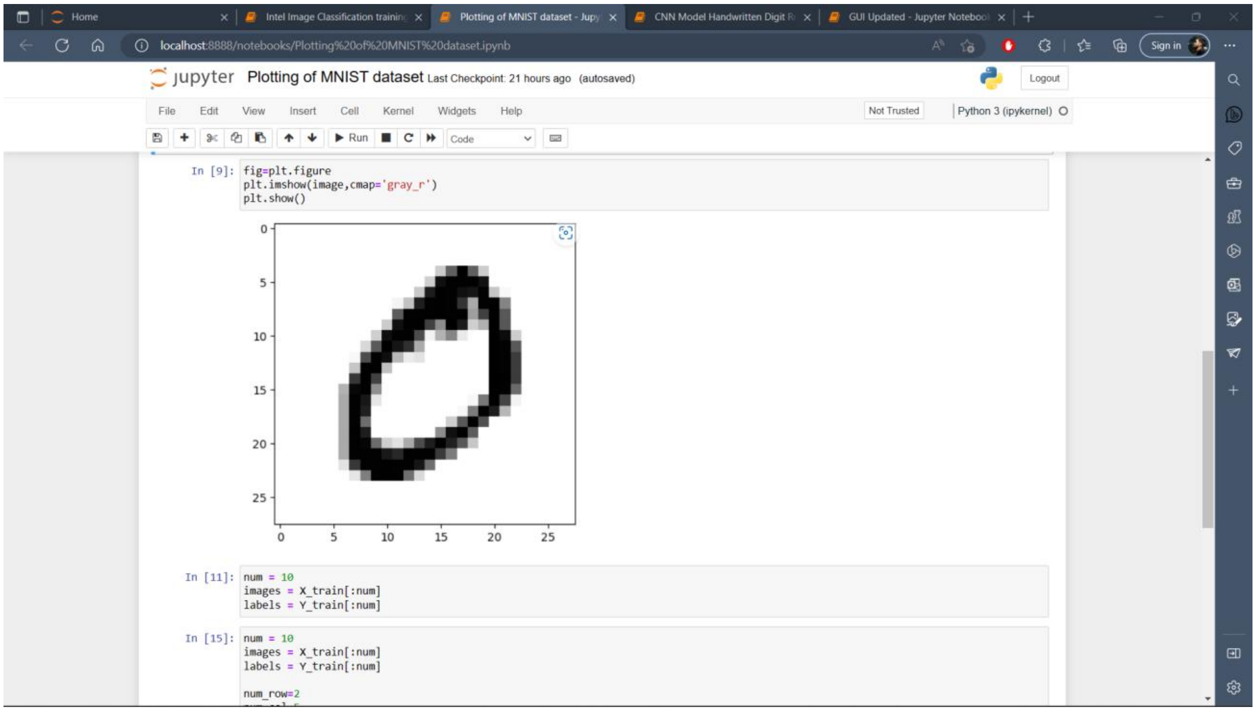
```
In [8]: plt.plot(history.history['accuracy'],label='accuracy')
        plt.plot(history.history['val_accuracy'],label='val_accuracy')
        plt.xlabel('Epoch')
        plt.ylabel('Accuracy')
        plt.legend(loc='lower right')

Out[8]: <matplotlib.legend.Legend at 0x1e2601503d0>

In [ ]:
```

2. Plotting of MNIST dataset


Home
Intel Image Classification training
Plotting of MNIST dataset - Jupy
CNN Model Handwritten Digit R
GUI Updated - Jupyter Noteboo
localhost:8888/notebooks/Plotting%20of%20MNIST%20dataset.ipynb
Sign in
jupyter
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
0
5
10
15
20
25

Plotting of MNIST dataset Last Checkpoint: 21 hours ago (autosaved)

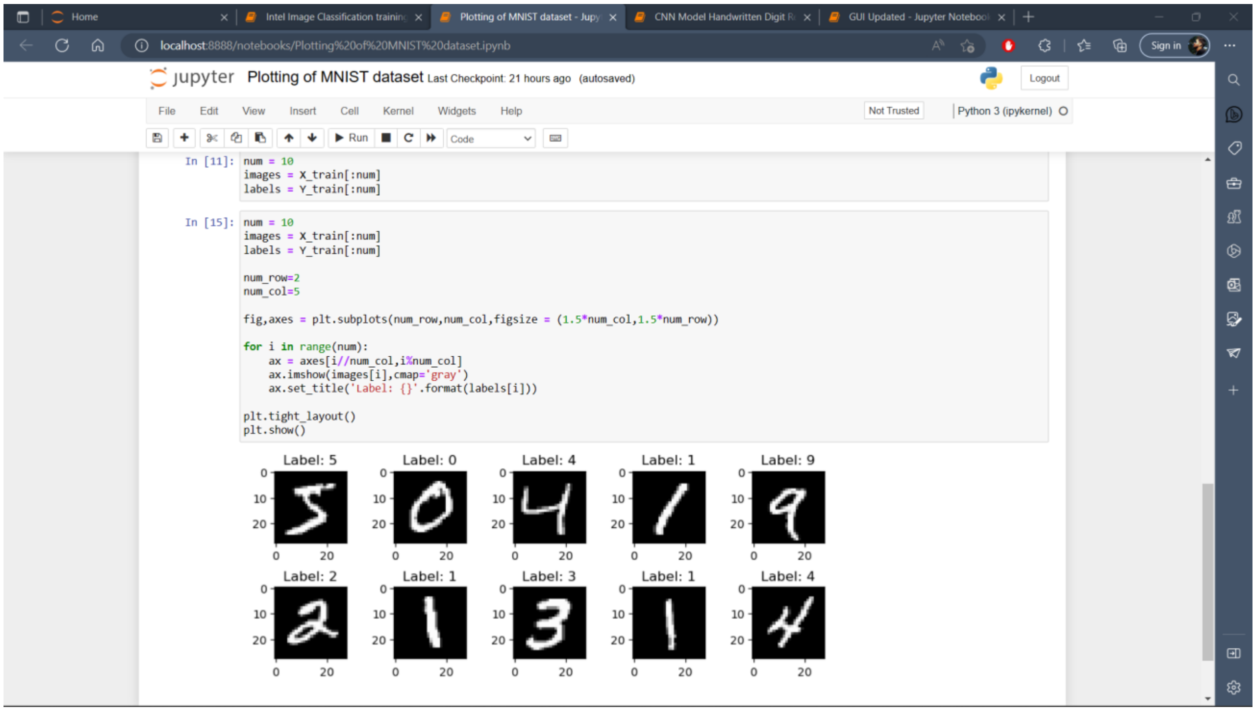
```
In [3]: from tensorflow.keras.datasets import mnist

        (X_train,Y_train),(X_test,Y_test) = mnist.load_data()

In [5]: import matplotlib.pyplot as plt
        %matplotlib inline

In [7]: sample=1
        image = X_train[sample]

        fig=plt.figure
        plt.imshow(image,cmap='gray')
        plt.show()
```


Home
Intel Image Classification training
Plotting of MNIST dataset - Jupy
CNN Model Handwritten Digit R
GUI Updated - Jupyter Notebook
localhost:8888/notebooks/Plotting%20of%20MNIST%20dataset.ipynb
Sign in
jupyter
Plotting of MNIST dataset Last Checkpoint: 21 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
0
5
10
15
20
25
0
5
10
15
20
25

```
In [9]: fig=plt.figure
        plt.imshow(image,cmap='gray_r')
        plt.show()
```

```
In [11]: num = 10
         images = X_train[:num]
         labels = Y_train[:num]
```

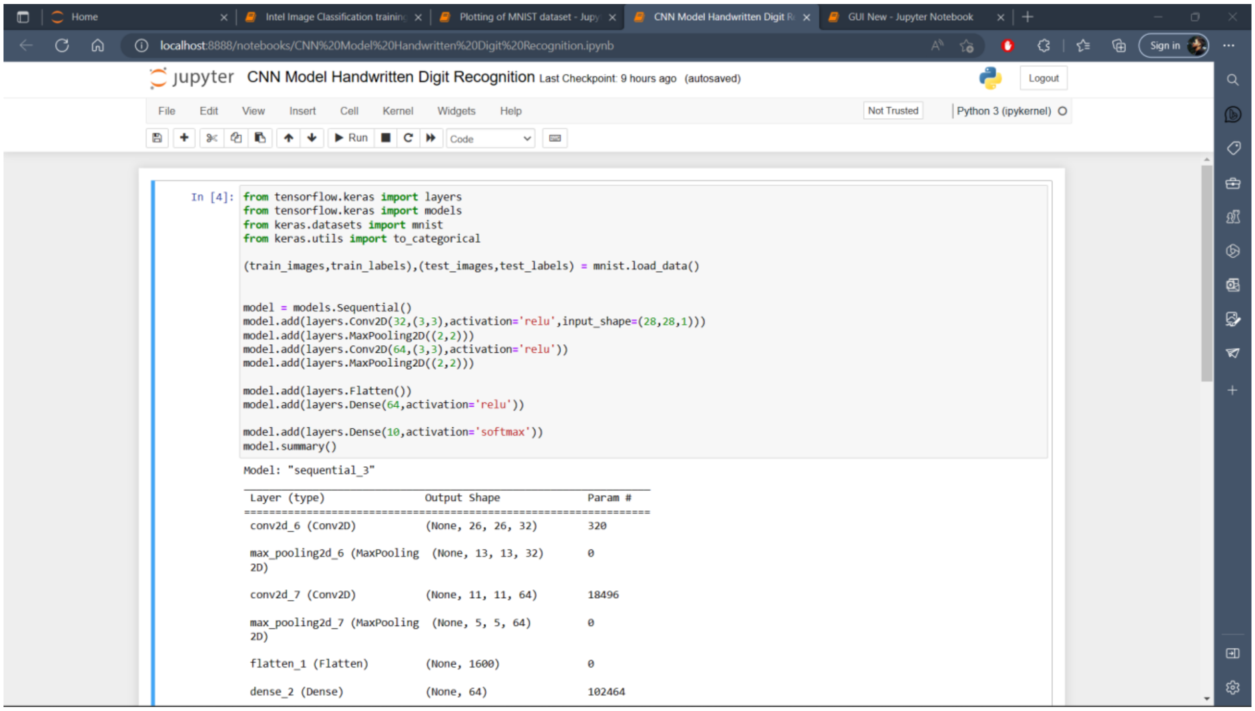
```
In [15]: num = 10
         images = X_train[:num]
         labels = Y_train[:num]

         num_row=2
```


Home
Intel Image Classification training
Plotting of MNIST dataset - Jupy
CNN Model Handwritten Digit R
GUI Updated - Jupyter Notebook
localhost:8888/notebooks/Plotting%20of%20MNIST%20dataset.ipynb
Sign in
jupyter
Plotting of MNIST dataset Last Checkpoint: 21 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Label: 5
Label: 0
Label: 4
Label: 1
Label: 9
Label: 2
Label: 1
Label: 3
Label: 1
Label: 4
0
10
20
0
20

```
In [11]: num = 10
         images = X_train[:num]
         labels = Y_train[:num]
```

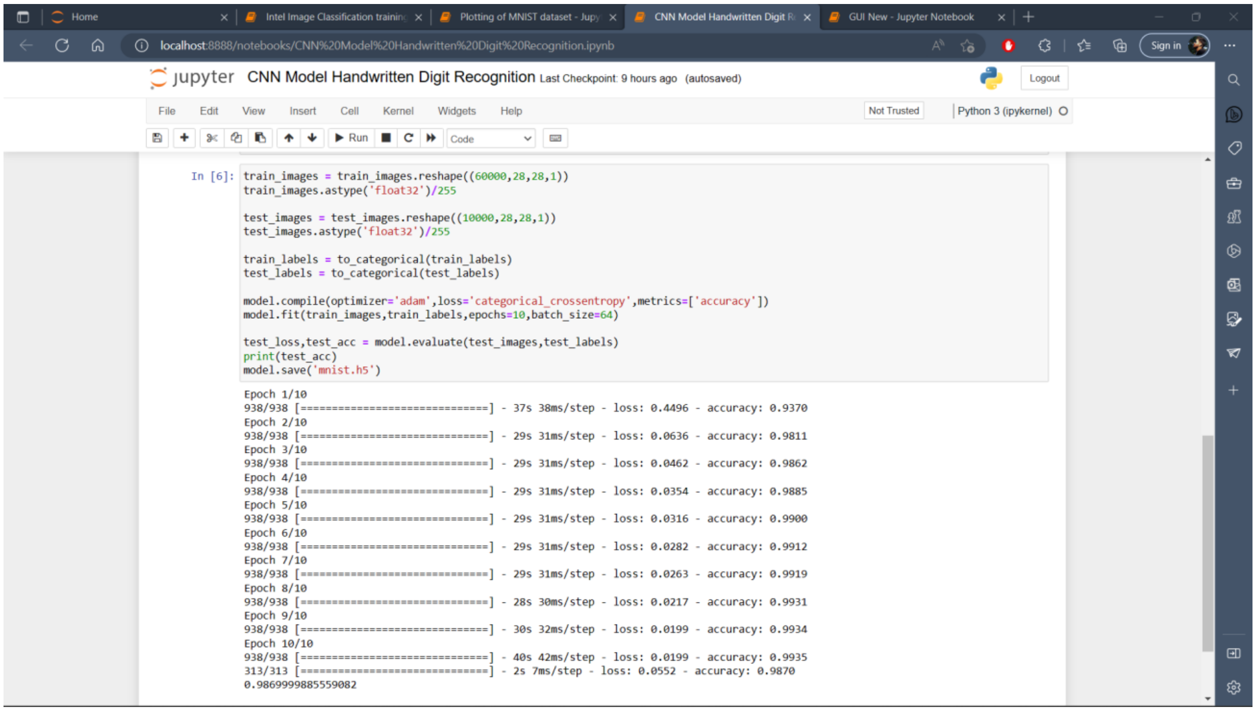
```
In [15]: num = 10
         images = X_train[:num]
         labels = Y_train[:num]

         num_row=2
         num_col=5

         fig,axes = plt.subplots(num_row,num_col,figsize = (1.5*num_col,1.5*num_row))

         for i in range(num):
             ax = axes[i//num_col,i%num_col]
             ax.imshow(images[i],cmap='gray')
             ax.set_title('Label: {}'.format(labels[i]))

         plt.tight_layout()
         plt.show()
```

3. CNN Model of Handwritten Digit Recognition


Home
Intel Image Classification training
Plotting of MNIST dataset - Jupyter
CNN Model Handwritten Digit R
GUI New - Jupyter Notebook
localhost:8888/notebooks/CNN%20Model%20Handwritten%20Digit%20Recognition.ipynb
Sign in
jupyter CNN Model Handwritten Digit Recognition Last Checkpoint: 9 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel) ○
Run
Code

```
In [4]: from tensorflow.keras import layers
        from tensorflow.keras import models
        from keras.datasets import mnist
        from keras.utils import to_categorical

        (train_images,train_labels),(test_images,test_labels) = mnist.load_data()


        model = models.Sequential()
        model.add(layers.Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))
        model.add(layers.MaxPooling2D((2,2)))
        model.add(layers.Conv2D(64,(3,3),activation='relu'))
        model.add(layers.MaxPooling2D((2,2)))

        model.add(layers.Flatten())
        model.add(layers.Dense(64,activation='relu'))

        model.add(layers.Dense(10,activation='softmax'))
        model.summary()
```

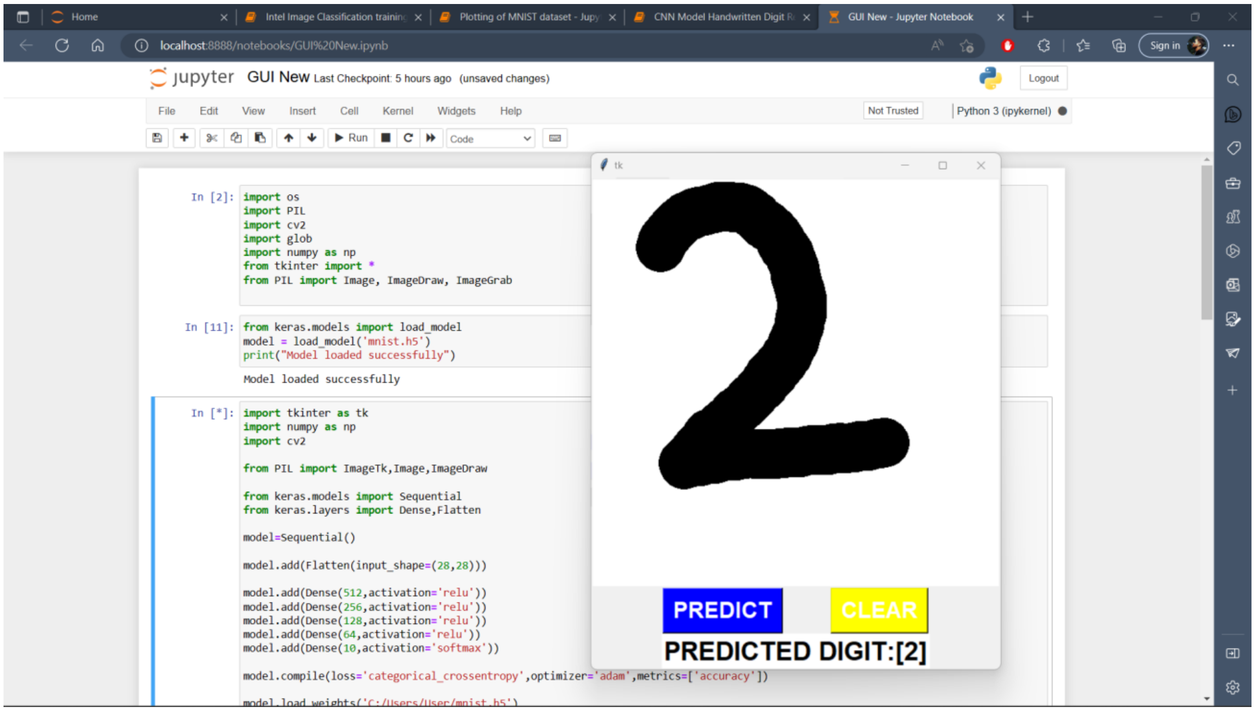
```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 26, 26, 32)        320

 max_pooling2d_6 (MaxPooling  (None, 13, 13, 32)       0
 2D)

 conv2d_7 (Conv2D)           (None, 11, 11, 64)        18496

 max_pooling2d_7 (MaxPooling  (None, 5, 5, 64)         0
 2D)

 flatten_1 (Flatten)         (None, 1600)              0

 dense_2 (Dense)             (None, 64)                102464
```


Home
Intel Image Classification training
Plotting of MNIST dataset - Jupyter
CNN Model Handwritten Digit R
GUI New - Jupyter Notebook
localhost:8888/notebooks/CNN%20Model%20Handwritten%20Digit%20Recognition.ipynb
Sign in
jupyter CNN Model Handwritten Digit Recognition Last Checkpoint: 9 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel) ○
Run
Code

```
In [6]: train_images = train_images.reshape((60000,28,28,1))
        train_images.astype('float32')/255

        test_images = test_images.reshape((10000,28,28,1))
        test_images.astype('float32')/255

        train_labels = to_categorical(train_labels)
        test_labels = to_categorical(test_labels)

        model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
        model.fit(train_images,train_labels,epochs=10,batch_size=64)

        test_loss,test_acc = model.evaluate(test_images,test_labels)
        print(test_acc)
        model.save('mnist.h5')
```

```
Epoch 1/10
938/938 [==============================] - 37s 38ms/step - loss: 0.4496 - accuracy: 0.9370
Epoch 2/10
938/938 [==============================] - 29s 31ms/step - loss: 0.0636 - accuracy: 0.9811
Epoch 3/10
938/938 [==============================] - 29s 31ms/step - loss: 0.0462 - accuracy: 0.9862
Epoch 4/10
938/938 [==============================] - 29s 31ms/step - loss: 0.0354 - accuracy: 0.9885
Epoch 5/10
938/938 [==============================] - 29s 31ms/step - loss: 0.0316 - accuracy: 0.9900
Epoch 6/10
938/938 [==============================] - 29s 31ms/step - loss: 0.0282 - accuracy: 0.9912
Epoch 7/10
938/938 [==============================] - 29s 31ms/step - loss: 0.0263 - accuracy: 0.9919
Epoch 8/10
938/938 [==============================] - 28s 30ms/step - loss: 0.0217 - accuracy: 0.9931
Epoch 9/10
938/938 [==============================] - 30s 32ms/step - loss: 0.0199 - accuracy: 0.9934
Epoch 10/10
938/938 [==============================] - 40s 42ms/step - loss: 0.0199 - accuracy: 0.9935
313/313 [==============================] - 2s 7ms/step - loss: 0.0552 - accuracy: 0.9870
0.9869999885559082
```

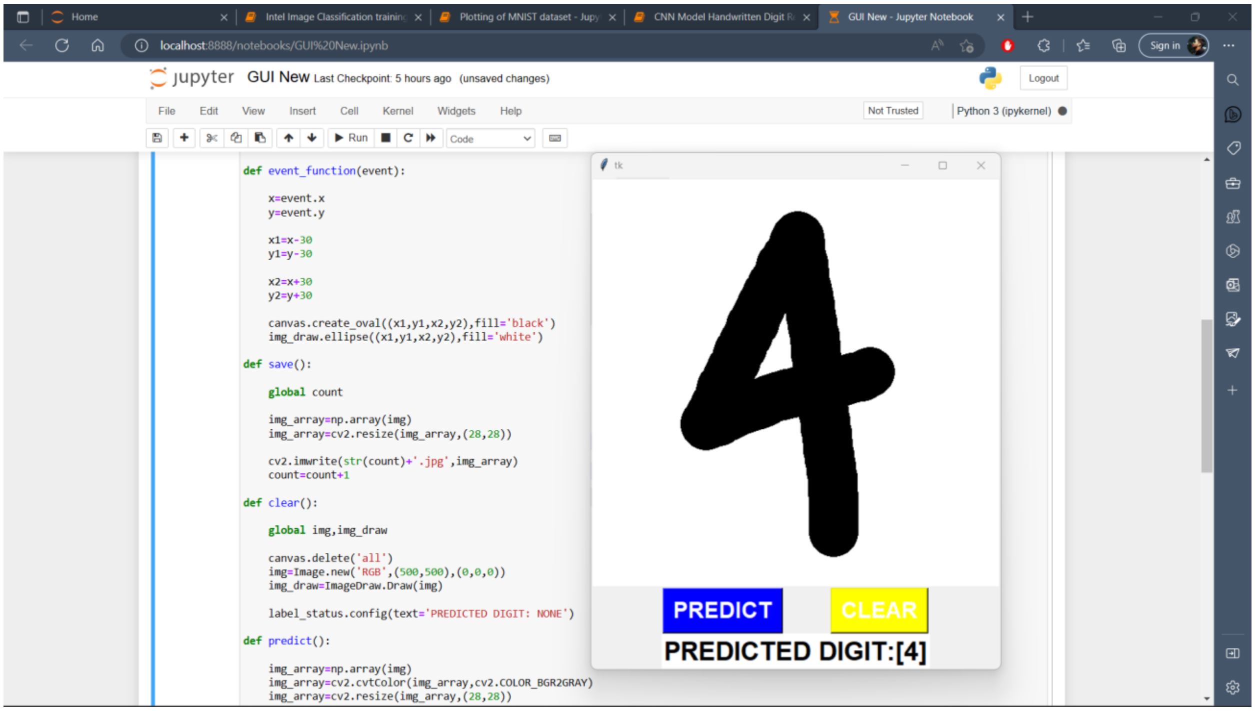
4. GUI


Home
Intel Image Classification trainin
Plotting of MNIST dataset - Jupy
CNN Model Handwritten Digit R
GUI New - Jupyter Notebook
localhost:8888/notebooks/GUI%20New.ipynb
Sign in
jupyter
GUI New Last Checkpoint: 5 hours ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
tk
PREDICT
CLEAR
PREDICTED DIGIT:[2]

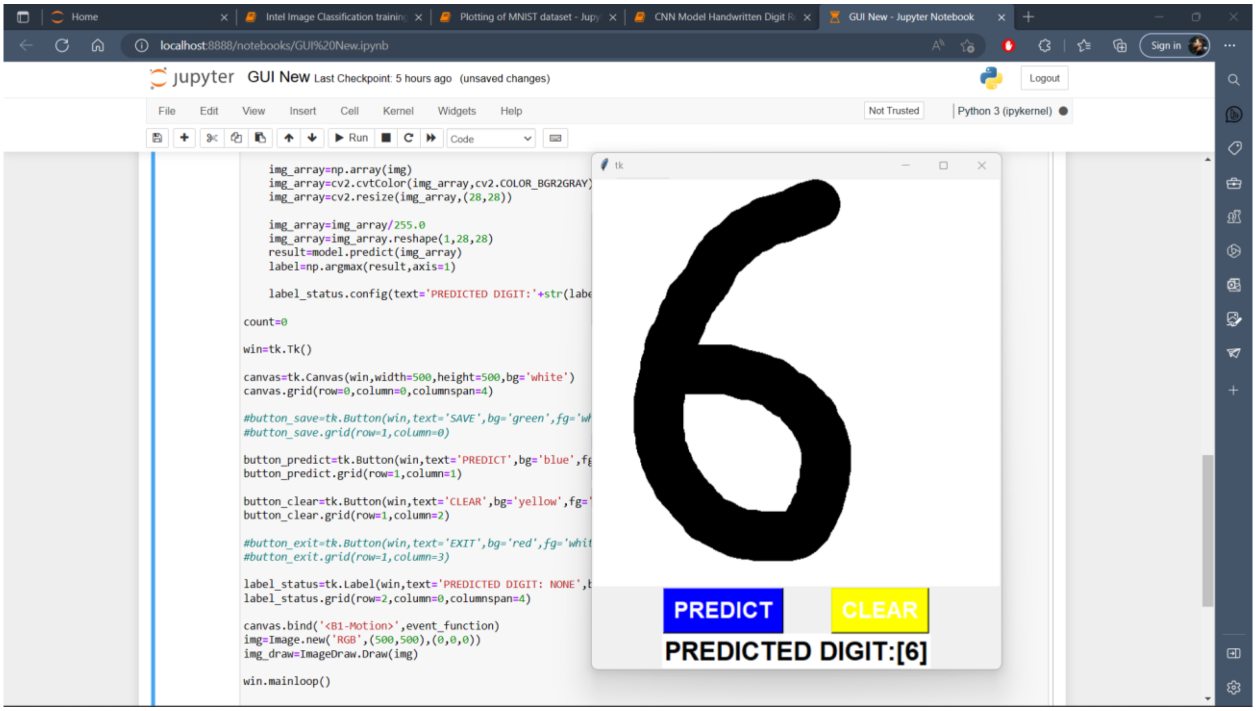
```
In [2]: import os
        import PIL
        import cv2
        import glob
        import numpy as np
        from tkinter import *
        from PIL import Image, ImageDraw, ImageGrab
```

```
In [11]: from keras.models import load_model
         model = load_model('mnist.h5')
         print("Model loaded successfully")

         Model loaded successfully
```

```
In [*]: import tkinter as tk
        import numpy as np
        import cv2

        from PIL import ImageTk,Image,ImageDraw

        from keras.models import Sequential
        from keras.layers import Dense,Flatten

        model=Sequential()

        model.add(Flatten(input_shape=(28,28)))

        model.add(Dense(512,activation='relu'))
        model.add(Dense(256,activation='relu'))
        model.add(Dense(128,activation='relu'))
        model.add(Dense(64,activation='relu'))
        model.add(Dense(10,activation='softmax'))

        model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

        model.load_weights('C:/Users/User/mnist.h5')
```


Home
Intel Image Classification trainin
Plotting of MNIST dataset - Jupy
CNN Model Handwritten Digit R
GUI New - Jupyter Notebook
localhost:8888/notebooks/GUI%20New.ipynb
Sign in
jupyter
GUI New Last Checkpoint: 5 hours ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
tk
PREDICT
CLEAR
PREDICTED DIGIT:[4]

```
def event_function(event):

    x=event.x
    y=event.y

    x1=x-30
    y1=y-30

    x2=x+30
    y2=y+30

    canvas.create_oval((x1,y1,x2,y2),fill='black')
    img_draw.ellipse((x1,y1,x2,y2),fill='white')

def save():

    global count

    img_array=np.array(img)
    img_array=cv2.resize(img_array,(28,28))

    cv2.imwrite(str(count)+'.jpg',img_array)
    count=count+1

def clear():

    global img,img_draw

    canvas.delete('all')
    img=Image.new('RGB',(500,500),(0,0,0))
    img_draw=ImageDraw.Draw(img)

    label_status.config(text='PREDICTED DIGIT: NONE')

def predict():

    img_array=np.array(img)
    img_array=cv2.cvtColor(img_array,cv2.COLOR_BGR2GRAY)
    img_array=cv2.resize(img_array,(28,28))
```



Home
Intel Image Classification training
Plotting of MNIST dataset - Jupy
CNN Model Handwritten Digit R
GUI New - Jupyter Notebook
localhost:8888/notebooks/GUI%20New.ipynb
Sign in
jupyter
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python 3 (ipykernel)
Run
Code
tk
PREDICT
CLEAR
PREDICTED DIGIT:[6]

GUI New Last Checkpoint: 5 hours ago (unsaved changes)

```
    img_array=np.array(img)
    img_array=cv2.cvtColor(img_array,cv2.COLOR_BGR2GRAY)
    img_array=cv2.resize(img_array,(28,28))

    img_array=img_array/255.0
    img_array=img_array.reshape(1,28,28)
    result=model.predict(img_array)
    label=np.argmax(result,axis=1)

    label_status.config(text='PREDICTED DIGIT:'+str(labe

count=0

win=tk.Tk()

canvas=tk.Canvas(win,width=500,height=500,bg='white')
canvas.grid(row=0,column=0,columnspan=4)

#button_save=tk.Button(win,text='SAVE',bg='green',fg='wh
#button_save.grid(row=1,column=0)

button_predict=tk.Button(win,text='PREDICT',bg='blue',fg
button_predict.grid(row=1,column=1)

button_clear=tk.Button(win,text='CLEAR',bg='yellow',fg='
button_clear.grid(row=1,column=2)

#button_exit=tk.Button(win,text='EXIT',bg='red',fg='whit
#button_exit.grid(row=1,column=3)

label_status=tk.Label(win,text='PREDICTED DIGIT: NONE',t
label_status.grid(row=2,column=0,columnspan=4)

canvas.bind('<B1-Motion>',event_function)
img=Image.new('RGB',(500,500),(0,0,0))
img_draw=ImageDraw.Draw(img)

win.mainloop()
```