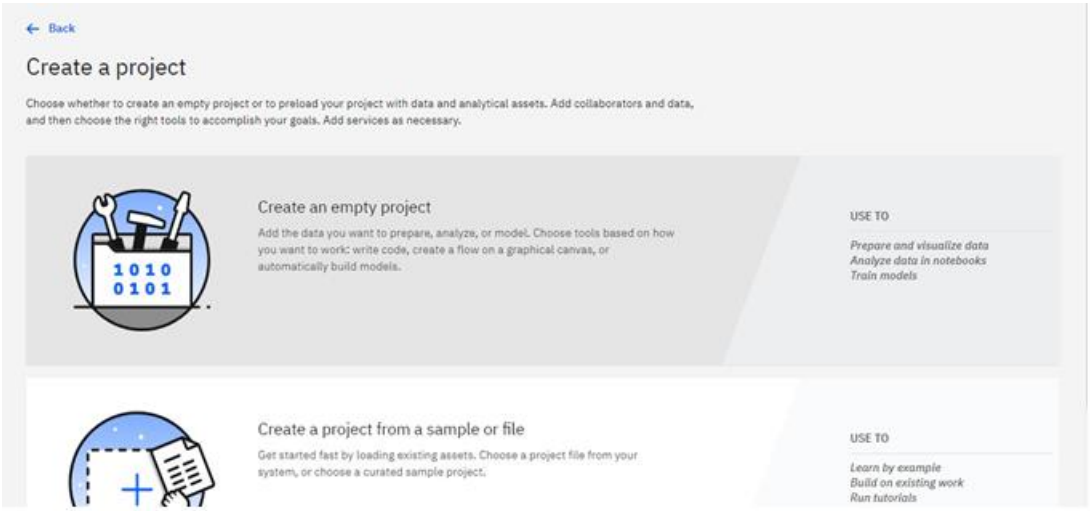
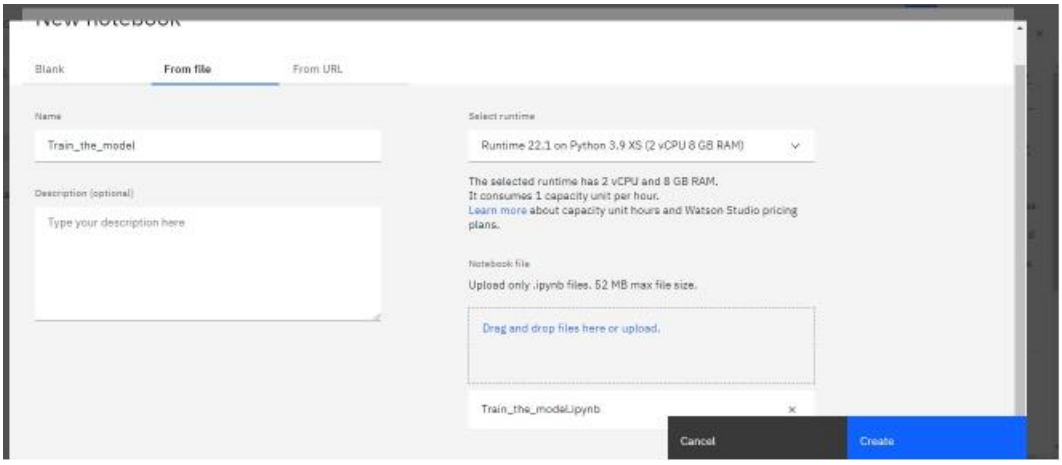
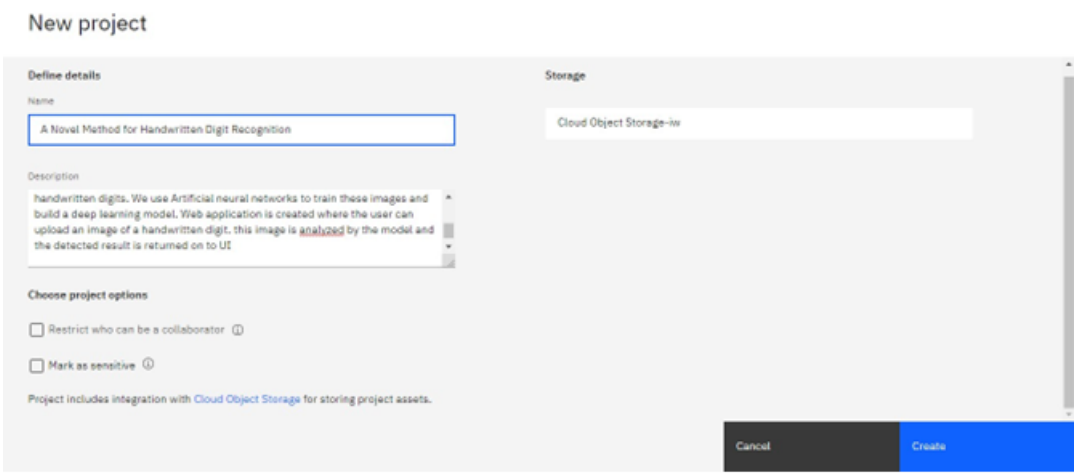
CREATING A PROJECT:



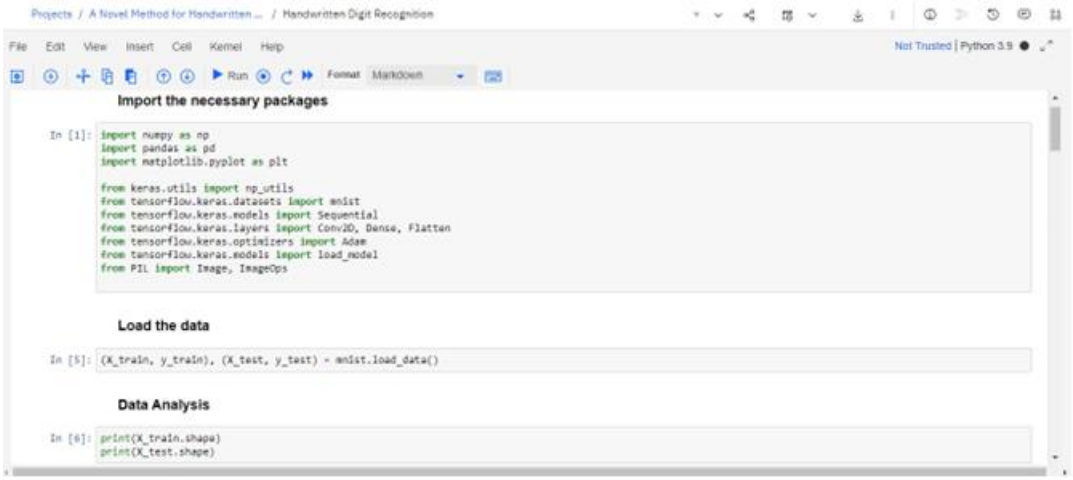
CREATING A NEW EVIRONMENT:

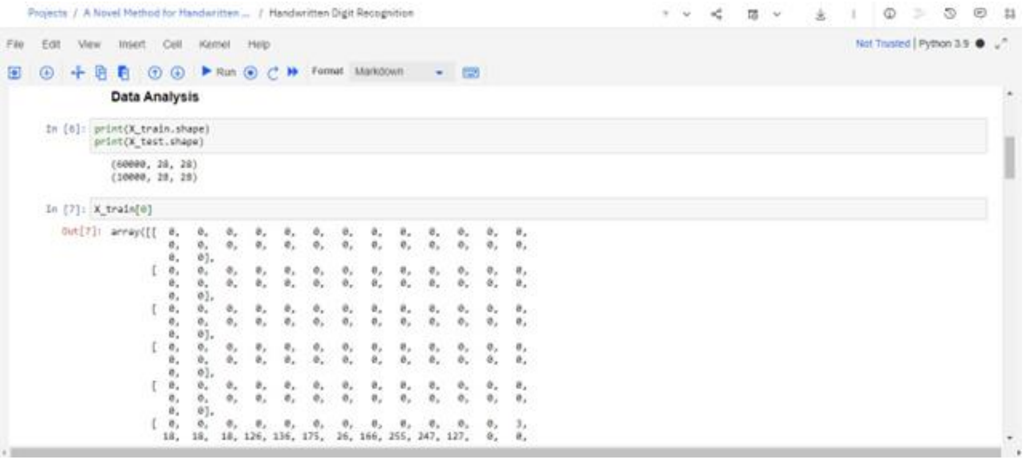


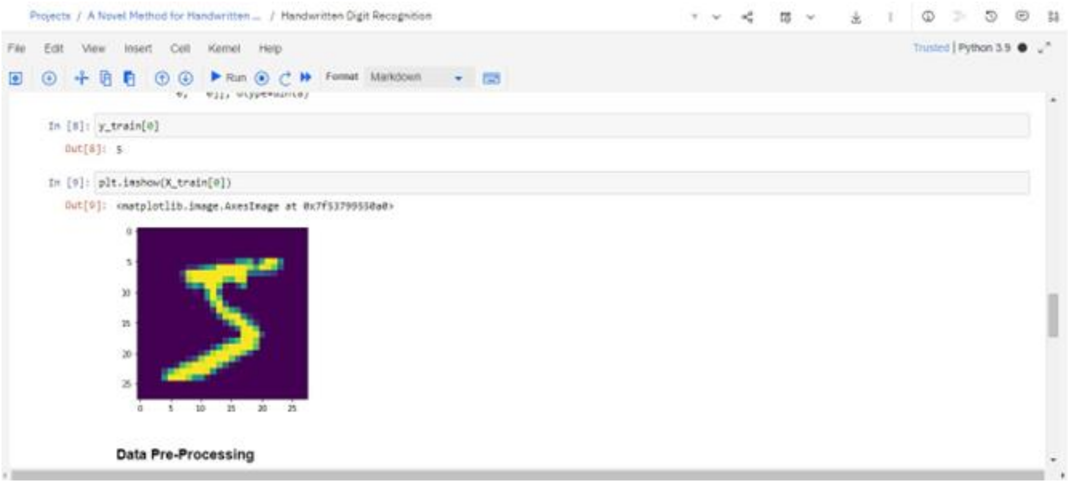
CREATING CLOUD SPACE:

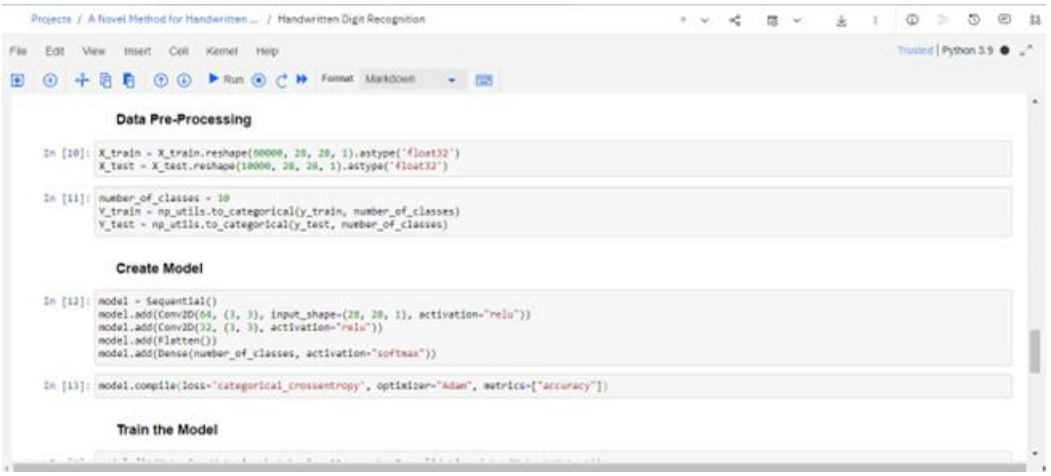


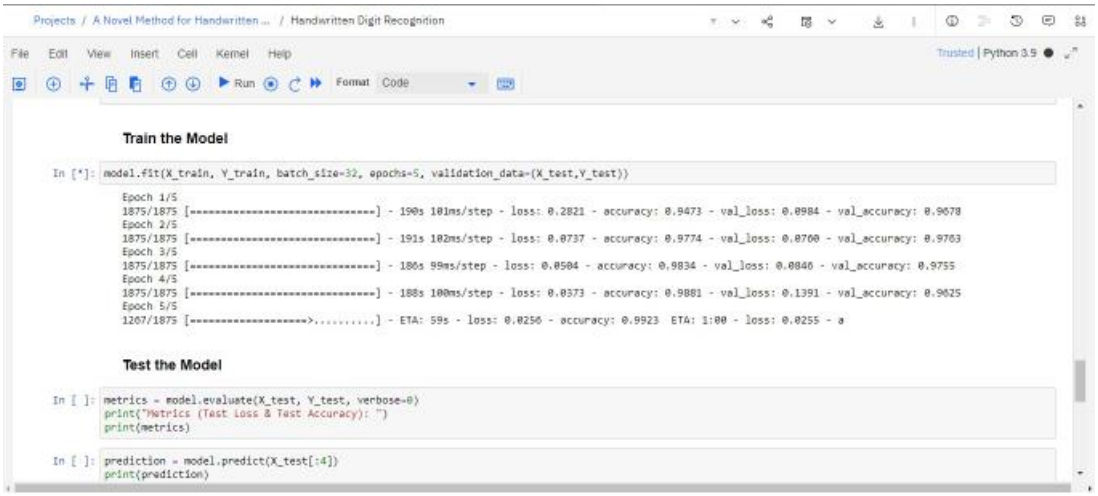
TRAINING THE MODEL ON IBM CLOUD:

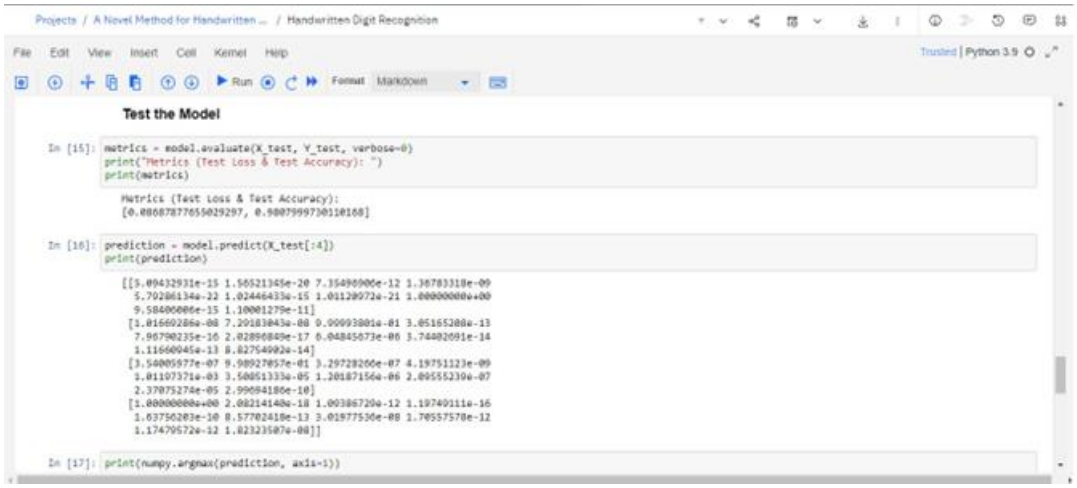


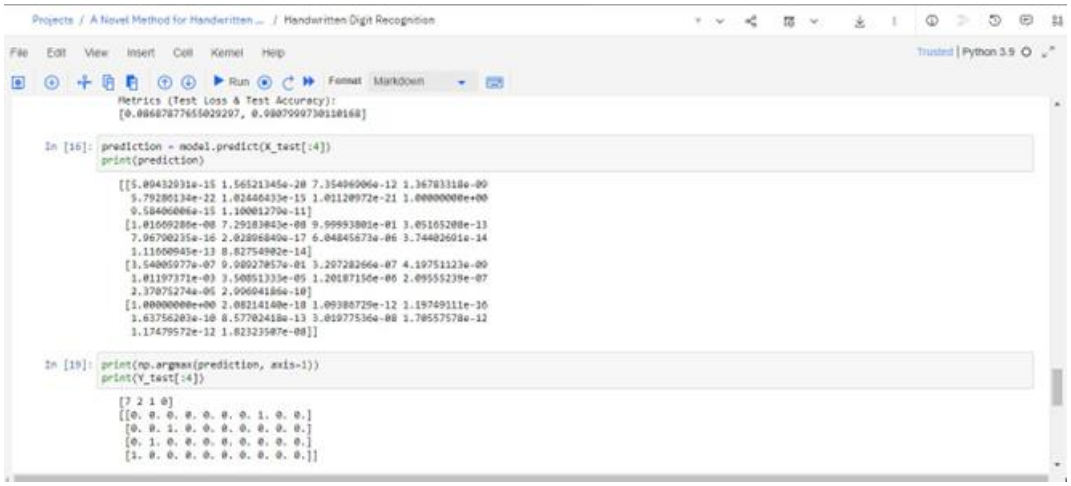

Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.9
Run
Format Markdown
Data Analysis
In [6]: print(X_train.shape)
print(X_test.shape)
(60000, 28, 28)
(10000, 28, 28)
In [7]: X_train[0]
Out[7]: array([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
18, 18, 18, 126, 136, 175, 26, 166, 255, 247, 127, 0, 0,

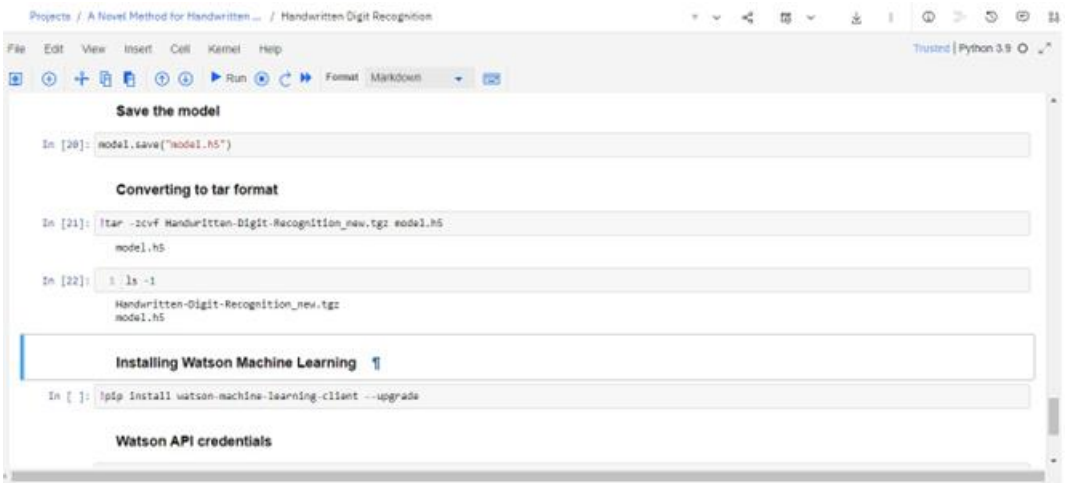

Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run
Format Markdown
In [8]: y_train[0]
Out[8]: 5
In [9]: plt.imshow(X_train[0])
Out[9]: <matplotlib.image.AxesImage at 0x7f53799550a0>
0
5
10
15
20
25
0 5 10 15 20 25
Data Pre-Processing

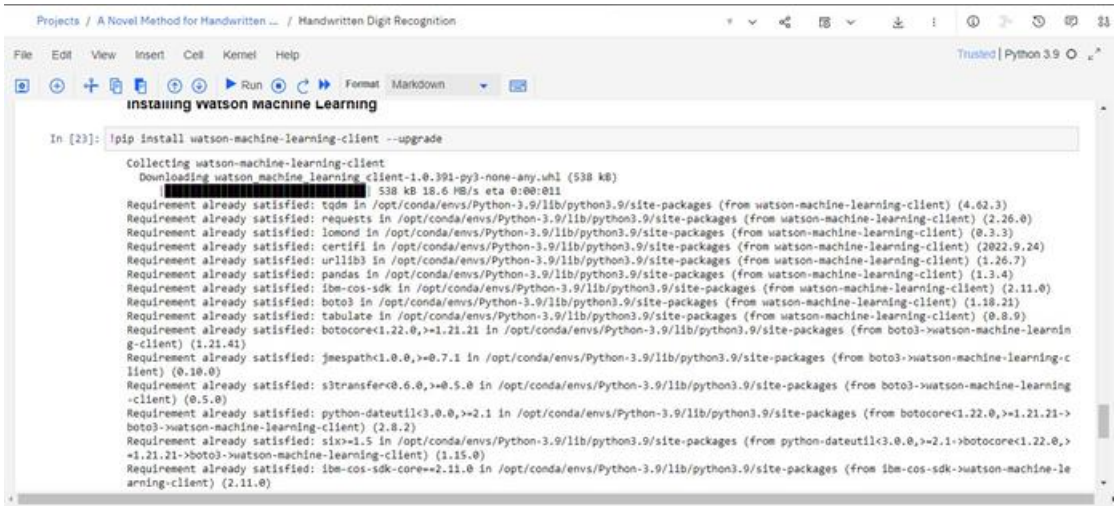

Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run Format Markdown
Data Pre-Processing
In [10]:
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')
In [11]:
number_of_classes = 10
Y_train = np_utils.to_categorical(y_train, number_of_classes)
Y_test = np_utils.to_categorical(y_test, number_of_classes)
Create Model
In [12]:
model = Sequential()
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation="relu"))
model.add(Conv2D(32, (3, 3), activation="relu"))
model.add(Flatten())
model.add(Dense(number_of_classes, activation="softmax"))
In [13]:
model.compile(loss='categorical_crossentropy', optimizer="Adam", metrics=["accuracy"])
Train the Model


Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run Format Code
Train the Model
In [*]:
model.fit(X_train, Y_train, batch_size=32, epochs=5, validation_data=(X_test,Y_test))
Epoch 1/5
1875/1875 [==========] - 190s 101ms/step - loss: 0.2821 - accuracy: 0.9473 - val_loss: 0.0984 - val_accuracy: 0.9678
Epoch 2/5
1875/1875 [==========] - 191s 102ms/step - loss: 0.0737 - accuracy: 0.9774 - val_loss: 0.0760 - val_accuracy: 0.9763
Epoch 3/5
1875/1875 [==========] - 186s 99ms/step - loss: 0.0504 - accuracy: 0.9834 - val_loss: 0.0846 - val_accuracy: 0.9755
Epoch 4/5
1875/1875 [==========] - 188s 100ms/step - loss: 0.0373 - accuracy: 0.9881 - val_loss: 0.1391 - val_accuracy: 0.9625
Epoch 5/5
1207/1875 [==========>.....] - ETA: 59s - loss: 0.0256 - accuracy: 0.9923 ETA: 1:00 - loss: 0.0255 - a
Test the Model
In [ ]:
metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)
In [ ]:
prediction = model.predict(X_test[:4])
print(prediction)


Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run Format Markdown
Test the Model
In [15]:
metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)
Metrics (Test Loss & Test Accuracy):
[0.08687877655029297, 0.9807999730110168]
In [16]:
prediction = model.predict(X_test[:4])
print(prediction)
[[5.09432931e-15 1.56521345e-20 7.35406906e-12 1.36783318e-09
5.79286134e-22 1.02446433e-15 1.01120972e-21 1.00000000e+00
9.58406086e-15 1.10001279e-11]
[1.01660286e-08 7.29183043e-08 9.99993801e-01 3.05165208e-13
7.96790235e-16 2.02896849e-17 6.04845673e-06 3.74402691e-14
1.11660945e-13 8.82754902e-14]
[3.54005977e-07 9.98927857e-01 3.29728266e-07 4.19751123e-09
1.01197371e-03 3.50851333e-05 1.20187156e-06 2.09555239e-07
2.37075274e-05 2.99684186e-10]
[1.00000000e+00 2.08214140e-18 1.09386729e-12 1.19749111e-16
1.63756203e-10 8.57702418e-13 3.01977536e-08 1.70557578e-12
1.17479572e-12 1.82323507e-08]]
In [17]:
print(numpy.argmax(prediction, axis=1))


Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run Format Markdown
Metrics (Test Loss & Test Accuracy):
[0.08687877655029297, 0.9807999730110168]
In [16]:
prediction = model.predict(X_test[:4])
print(prediction)
[[5.09432931e-15 1.56521345e-20 7.35406906e-12 1.36783318e-09
5.79286134e-22 1.02446433e-15 1.01120972e-21 1.00000000e+00
9.58406086e-15 1.10001279e-11]
[1.01660286e-08 7.29183043e-08 9.99993801e-01 3.05165208e-13
7.96790235e-16 2.02896849e-17 6.04845673e-06 3.74402691e-14
1.11660945e-13 8.82754902e-14]
[3.54005977e-07 9.98927857e-01 3.29728266e-07 4.19751123e-09
1.01197371e-03 3.50851333e-05 1.20187156e-06 2.09555239e-07
2.37075274e-05 2.99684186e-10]
[1.00000000e+00 2.08214140e-18 1.09386729e-12 1.19749111e-16
1.63756203e-10 8.57702418e-13 3.01977536e-08 1.70557578e-12
1.17479572e-12 1.82323507e-08]]
In [19]:
print(np.argmax(prediction, axis=1))
print(Y_test[:4])
[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
[0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
[0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
[1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]


Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run Format Markdown
Save the model
In [20]: model.save("model.h5")
Converting to tar format
In [21]: !tar -zcvf Handwritten-Digit-Recognition_new.tgz model.h5
model.h5
In [22]: ls -l
Handwritten-Digit-Recognition_new.tgz
model.h5
Installing Watson Machine Learning
In [ ]: !pip install watson-machine-learning-client --upgrade
Watson API credentials


Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run Format Markdown
Installing Watson Machine Learning
In [23]: !pip install watson-machine-learning-client --upgrade
Collecting watson-machine-learning-client
Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (538 kB)
538 kB 18.6 MB/s eta 0:00:011
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (1.21.41)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.10.0)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.5.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (1.15.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)

WATSON API CONFIGU