Team Id : PNT2022TMID45281

Team members :

1. Partheeban K - 812119104023
2. Timple Rosni Augustina N - 812119104037
3. Kamali P - 812119104012
4. Lavanya S - 812119104301
5. Dhayathipathy B - 812119104005

Project Name : A Novel Method for Handwritten Digit Recognition System

Objective :

Handwriting recognition is one of the compelling research works going on because everyindividual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real time applications. MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use artificial neural networks to train these images and build a deep learning model. A web application is created where the user can upload an image of a handwritten digit. this image is analysed by the model and the detected result is returned on to the UI.

Methodology :

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS. Conda is an open-source, cross-platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupiter notebook and spyder. TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers can easily build and deploy ML powered applications. Flask to build web app. Keras leverages various optimization techniques to make high level neural network API easier and more performant. It supports the following features:

- Consistent, simple and extensible API.
- Minimal structure - easy to achieve the result without any frills.

- It supports multiple platforms and backend.
- It is a user-friendly framework which runs on both CPU and GPU.
- Highly scalability of computation.

Novelty :

Handwriting recognition is one of the compelling research works going on because every individual in this world has their own style of writing. It is the capability of the computer to identify and understand handwritten digits or characters automatically. Because of the progress in science and technology, everything is being digitalized to reduce human effort. Hence, there comes a need for handwritten digit recognition in many real time applications. Though there aresome OCR and digit recognition software are available on the market, they are not widely used in the banking industry and many of them are not so accurate. Our goal is to build a digit recognition system with high accuracy.

Problem :

Cheque transactions account for almost 2 - 3 % of the total transactions in India. Almostmore than half a billion cheques are being processed each year. Manual cheque processing leads to delay in processing cheques causing customer dissatisfaction. The department of traffic enforcement does manual monitoring which is error prone. Integrating digit recognition tool with image processing software leads to automatically imposing penalty online for speeding, enforcing law to prevent accidents and ensure road safety.

What would happen when it is fixed :

Business Model Speed up the cheque approval process Store transaction records Social impact Ensure road safety by identifying the owner of the speeding vehicle by using the registration number of that vehicle.

Why is it important to fix this issue :

Automating these tasks removes the need for human effort which is error prone in performing these kinds of tedious works and improves speed as well as efficiency.

Proposed Solution Template:

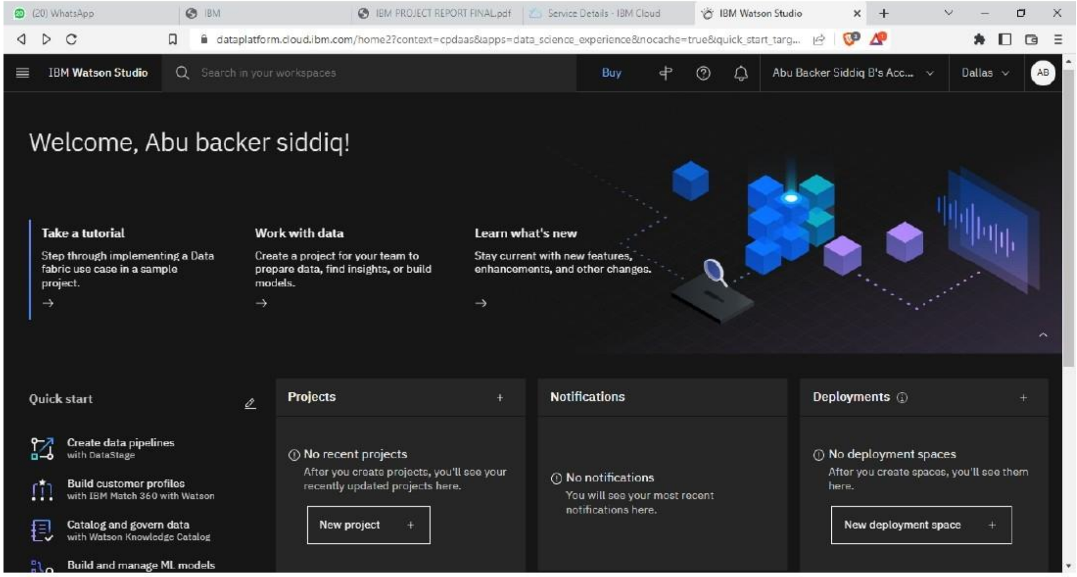
S.No.	**Parameter**	**Description**

1.	Problem Statement (Problem tobe solved)	Handwriting recognition is one of the compelling research works going on because every individual in this world hastheir own style of writing. It is the capability of the computer to identify andunderstand handwritten digits or characters automatically. Because of theprogress in science and technology, everything is being digitized to reduce human effort. Hence, there comes a needfor handwritten digit recognition in many real time applications. Most of the banksare still relying on the manual cheque processing which is both time-consumingand error prone. Handwriting recognition system with a reliable accuracy can have an impact in these business fields.
2.	Idea / Solution description	MNIST data set is widely used for this recognition process and it has 70000 handwritten digits. We use artificial neural networks to train these images and build a deep learning model. We opt to use multi-layer neural networks as deep NN. Due to data is Image, the best type of neural network satisfying our goal is **Convolutional Neural Networks**. As we have to do for most of the data, normalization plays an important role in our process. Before doing any tasks, pre-processing images (our data-set) is highly recommended. Consequently better accuracy will be achieved by pre- processed data. After pre-processing andnormalizing, the prepared data set could be used as input to our deep convolutional neural network. Then deep NN will be run and fit to our data and the result will be produced by that.

3.	Novelty / Uniqueness	A web application is created where the user can upload an image of a handwritten digit. this image is analysedby the model and the detected result is returned on to the UI. One of the major decisions that had to be made was choosing the suitable programming language to satisfy our goal of extracting knowledge from our data. After some searching the suitable decision has beenmade by selecting Python as the project programming language. Due to the fact that, a lot of tools and frameworks are available for Python to create powerful Artificial Neural Networks. Also IBM Watson helps to predict future outcomes,automate complex processes,and optimize user’s time. Andalso the result accuracy will be increasedfrom 70% which is the accuracy of the test results that the previous developed codes produced.
4.	Social Impact / Customer Satisfaction	Can ensure road safety by identifying the owner of the speeding vehicle by using the registration number of that vehicle. Can automate data entry jobs and speed up cheque approval process
5.	Business Model (Revenue Model)	Can collaborate with banks to speed-upthe cheque approval process, which improves customer experience.
6.	Scalability of the Solution	This project will help us to detect digits more precisely. Also we can develop this model to recognize alphabets.

IBM Cloud model :

CREATING AN IBM WATSON STUDIO:


(20) WhatsApp
IBM
IBM PROJECT REPORT FINAL.pdf
Service Details - IBM Cloud
IBM Watson Studio
dataplatform.cloud.ibm.com/home2?context=cpdaas&apps=data_science_experience&nocache=true&quick_start_targ...
IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
AB

Welcome, Abu backer siddiq!

Take a tutorial
Step through implementing a Data fabric use case in a sample project.

Work with data
Create a project for your team to prepare data, find insights, or build models.

Learn what's new
Stay current with new features, enhancements, and other changes.

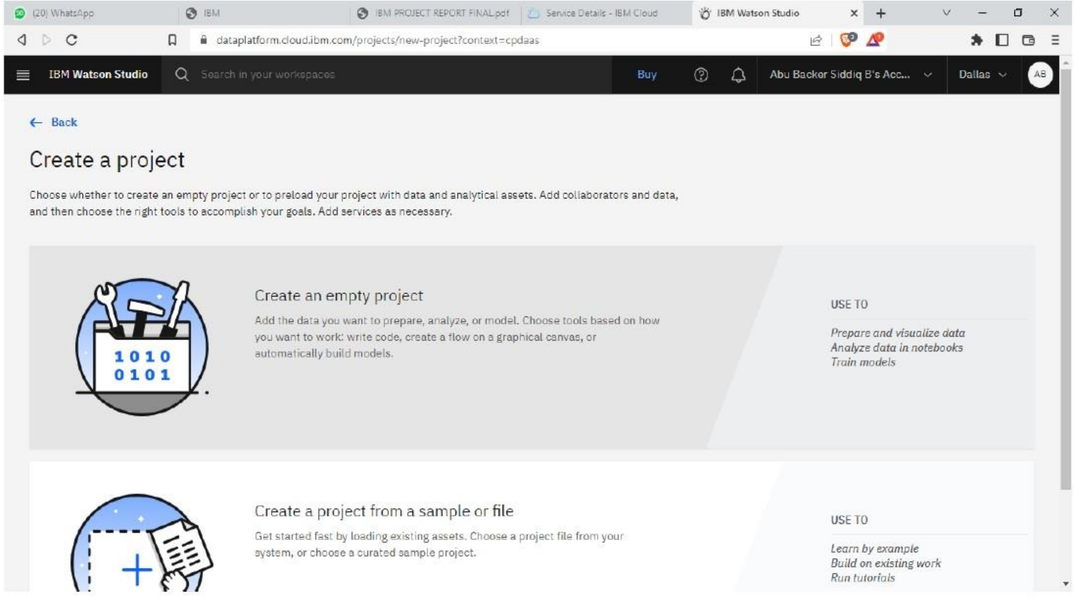
Quick start
- Create data pipelines with DataStage
- Build customer profiles with IBM Match 360 with Watson
- Catalog and govern data with Watson Knowledge Catalog
- Build and manage ML models

Projects
No recent projects
After you create projects, you'll see your recently updated projects here.
New project +

Notifications
No notifications
You will see your most recent notifications here.

Deployments
No deployment spaces
After you create spaces, you'll see them here.
New deployment space +

CREATING A PROJECT:


(20) WhatsApp
IBM
IBM PROJECT REPORT FINAL.pdf
Service Details - IBM Cloud
IBM Watson Studio
dataplatform.cloud.ibm.com/projects/new-project?context=cpdaas
IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
AB
1010
0101

← Back

Create a project

Choose whether to create an empty project or to preload your project with data and analytical assets. Add collaborators and data, and then choose the right tools to accomplish your goals. Add services as necessary.

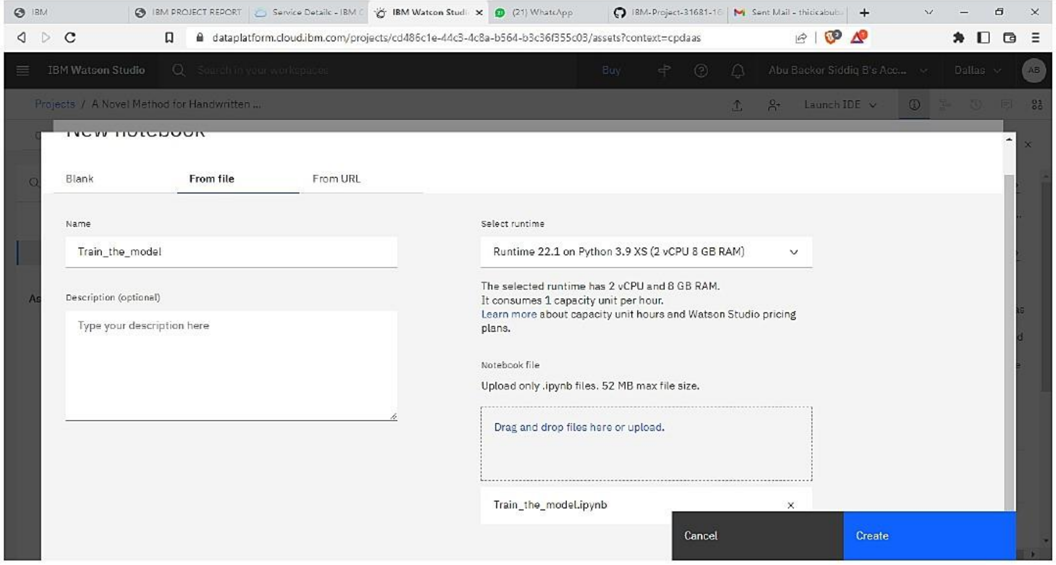
Create an empty project
Add the data you want to prepare, analyze, or model. Choose tools based on how you want to work: write code, create a flow on a graphical canvas, or automatically build models.

USE TO
Prepare and visualize data
Analyze data in notebooks
Train models

Create a project from a sample or file
Get started fast by loading existing assets. Choose a project file from your system, or choose a curated sample project.

USE TO
Learn by example
Build on existing work
Run tutorials

CREATING A NEW EVIRONMENT:



dataplatform.cloud.ibm.com/projects/cd486c1e-44c3-4c8a-b564-b3c36f355c03/assets?context=cpdaas
IBM Watson Studio
Projects / A Novel Method for Handwritten ...
Launch IDE

Blank | From file | From URL

Name: Train_the_model

Description (optional): Type your description here

Select runtime: Runtime 22.1 on Python 3.9 XS (2 vCPU 8 GB RAM)

The selected runtime has 2 vCPU and 8 GB RAM.
It consumes 1 capacity unit per hour.
Learn more about capacity unit hours and Watson Studio pricing plans.

Notebook file

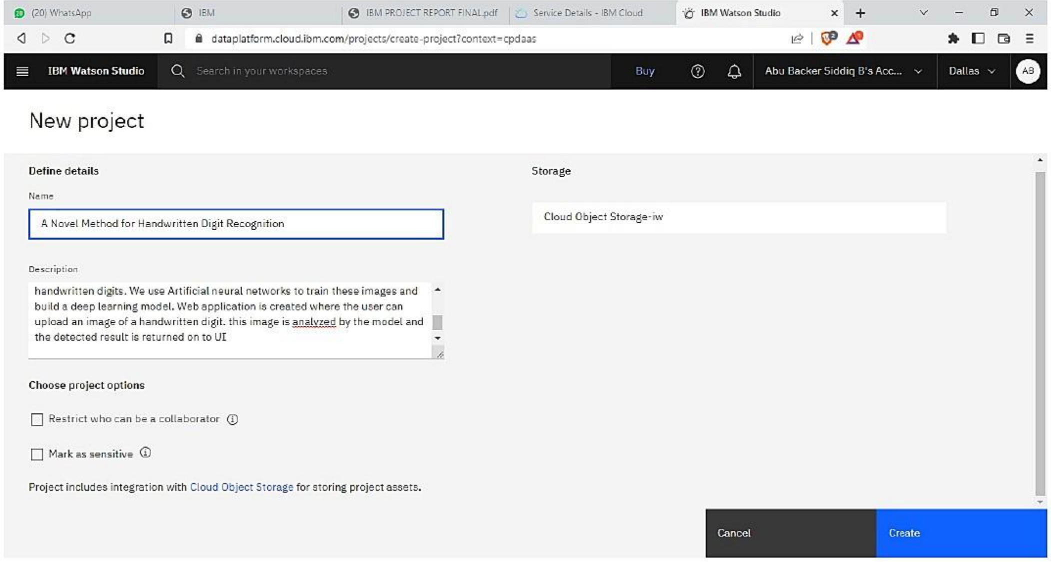
Upload only .ipynb files. 52 MB max file size.

Drag and drop files here or upload.

Train_the_model.ipynb

Cancel | Create

CREATING CLOUD SPACE:



dataplatform.cloud.ibm.com/projects/create-project?context=cpdaas
IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas

New project

Define details

Name: A Novel Method for Handwritten Digit Recognition

Description: handwritten digits. We use Artificial neural networks to train these images and build a deep learning model. Web application is created where the user can upload an image of a handwritten digit. this image is analyzed by the model and the detected result is returned on to UI

Storage: Cloud Object Storage-iw

Choose project options

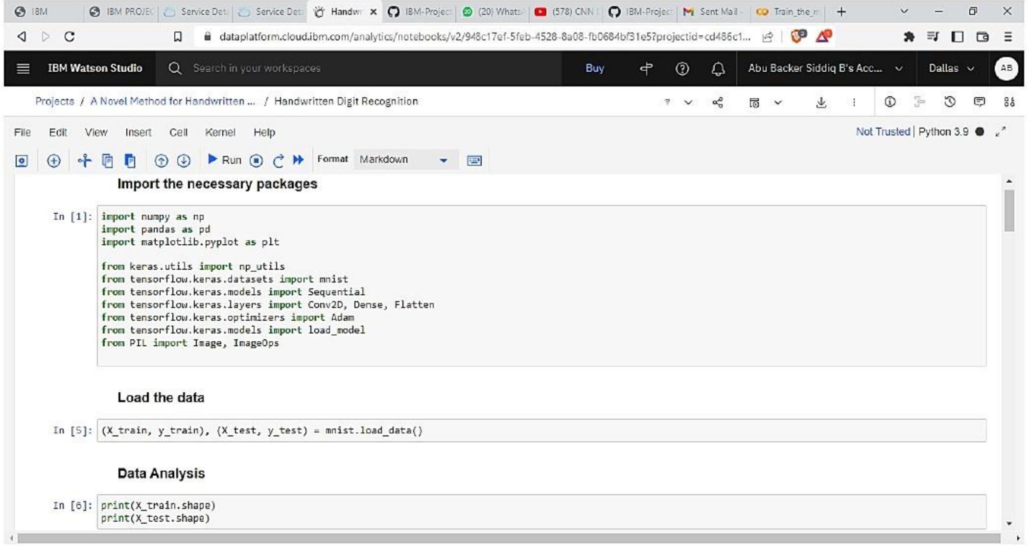
□ Restrict who can be a collaborator

□ Mark as sensitive

Project includes integration with Cloud Object Storage for storing project assets.

Cancel | Create

TRAINING THE MODEL ON IBM CLOUD:


IBM Watson Studio
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.9
Run
Format Markdown

Import the necessary packages

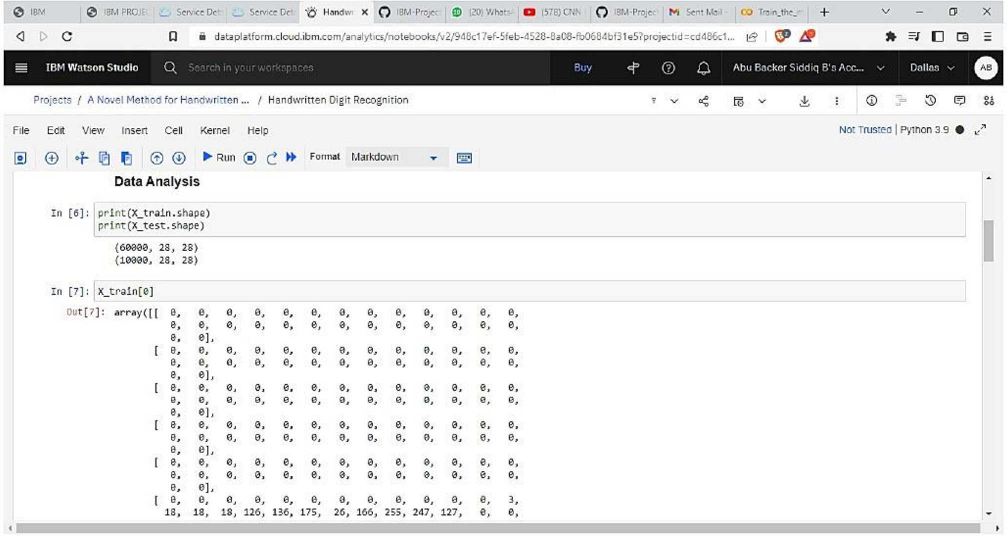
```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt

        from keras.utils import np_utils
        from tensorflow.keras.datasets import mnist
        from tensorflow.keras.models import Sequential
        from tensorflow.keras.layers import Conv2D, Dense, Flatten
        from tensorflow.keras.optimizers import Adam
        from tensorflow.keras.models import load_model
        from PIL import Image, ImageOps
```

Load the data

```
In [5]: (X_train, y_train), (X_test, y_test) = mnist.load_data()
```

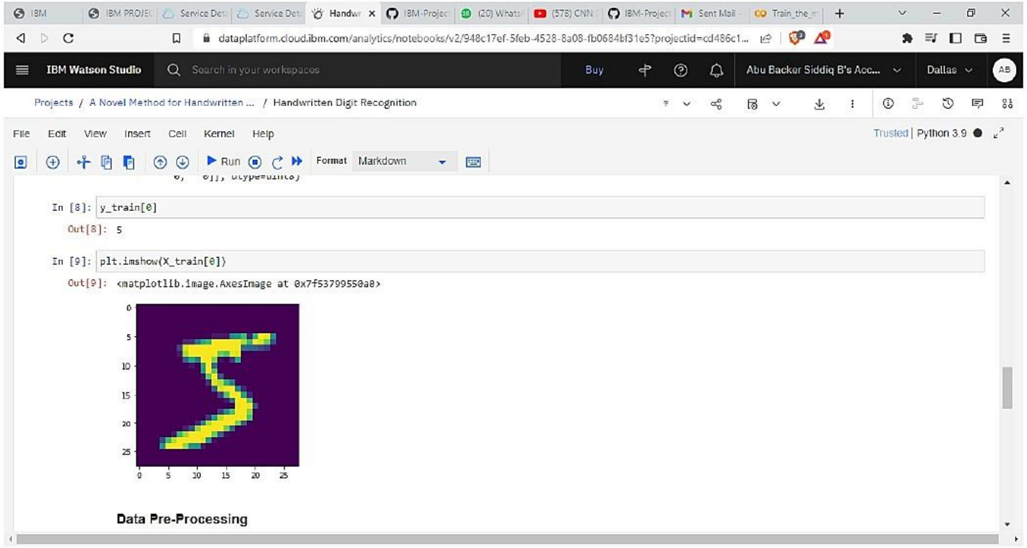
Data Analysis

```
In [6]: print(X_train.shape)
        print(X_test.shape)
```


IBM Watson Studio
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Not Trusted | Python 3.9
Run
Format Markdown

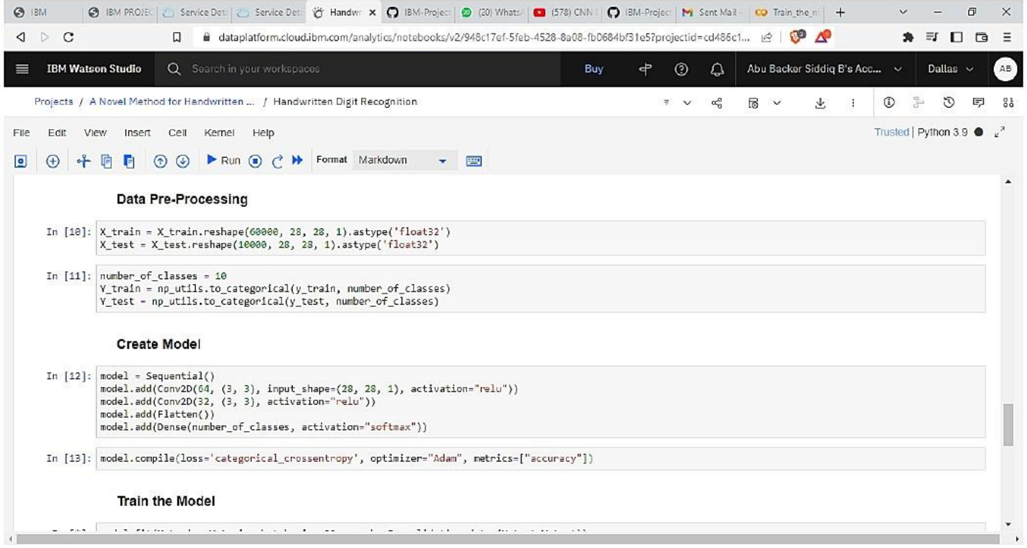
Data Analysis

```
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,
                  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
                [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
                [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
                [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
                [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
                  0,   0],
                [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   3,
                 18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,   0,
```


IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run
Format Markdown
0
5
10
15
20
25
0
5
10
15
20
25

```
In [8]: y_train[0]
Out[8]: 5

In [9]: plt.imshow(X_train[0])
Out[9]: <matplotlib.image.AxesImage at 0x7f53799550a0>
```

Data Pre-Processing


IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run
Format Markdown

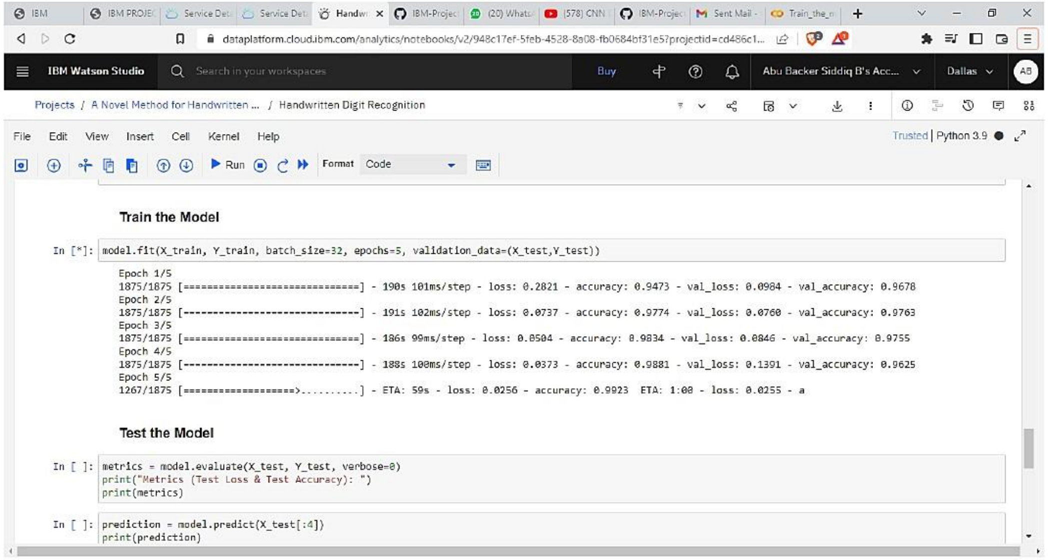
Data Pre-Processing

```
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

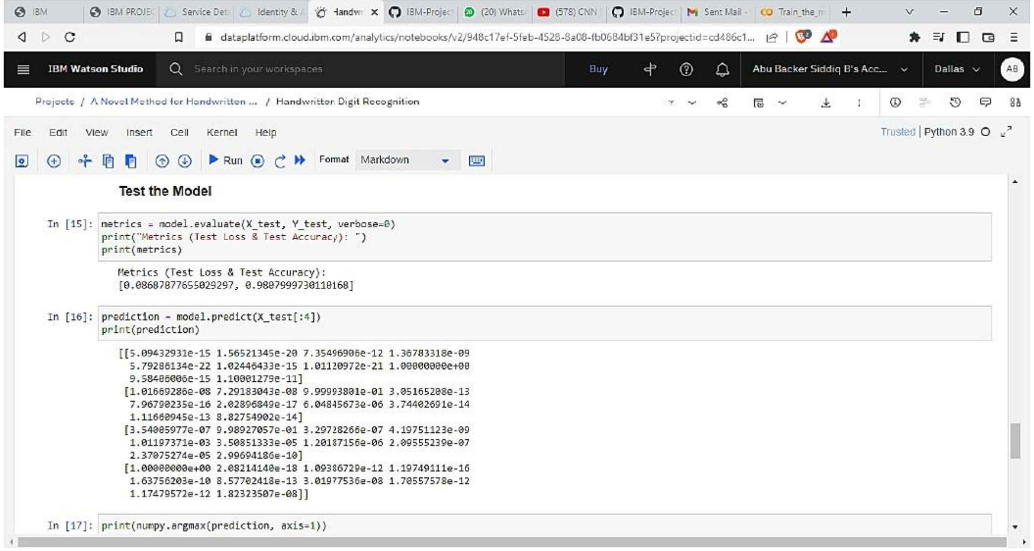

IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run
Format Code

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
1267/1875 [===================>..........] - ETA: 59s - loss: 0.0256 - accuracy: 0.9923  ETA: 1:00 - loss: 0.0255 - a
```

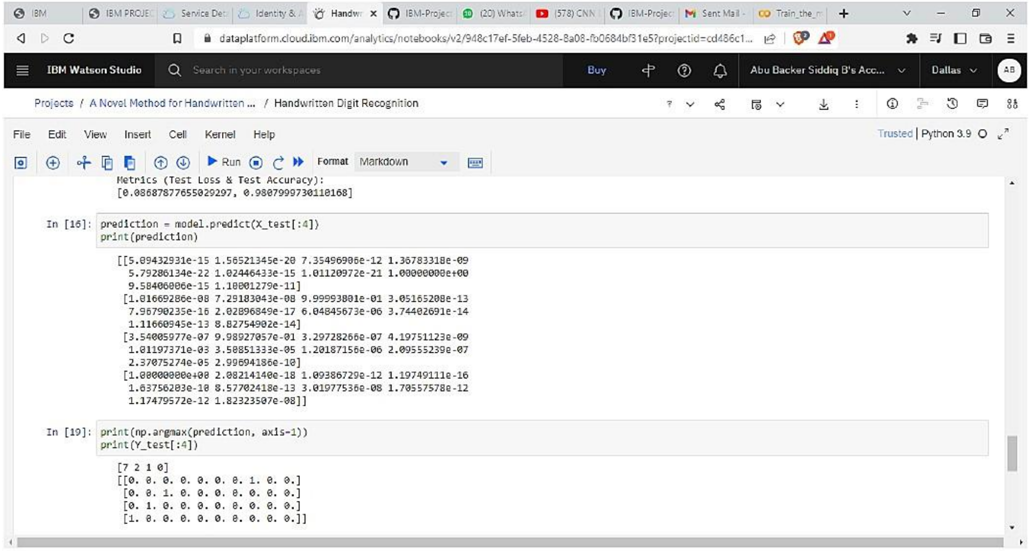
Test the Model

```
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)
```


IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
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 7.35496906e-12 1.36783318e-09
   5.79286134e-22 1.02446433e-15 1.01120972e-21 1.00000000e+00
   9.58406006e-15 1.10001279e-11]
  [1.01669286e-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 9.98927057e-01 3.29728266e-07 4.19751123e-09
   1.01197371e-03 3.50851333e-05 1.20187156e-06 2.09555239e-07
   2.37075274e-05 2.99694186e-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))
```



IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run
Format
Markdown

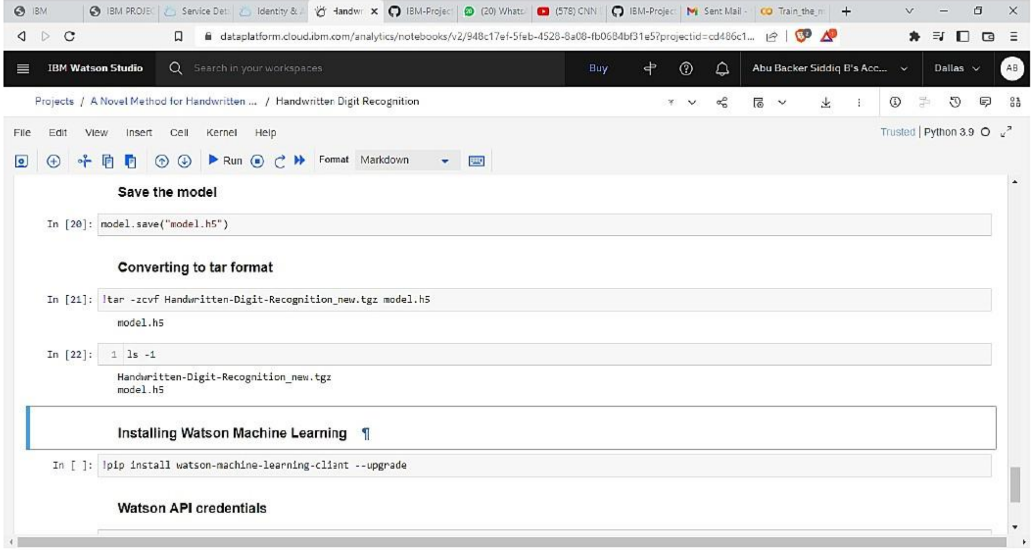
```
Metrics (Test Loss & Test Accuracy):
[0.08687877655029297, 0.9807999730110168]
```

```
In [16]: prediction = model.predict(X_test[:4])
         print(prediction)
```

```
[[5.09432931e-15 1.56521345e-20 7.35496906e-12 1.36783318e-09
  5.79286134e-22 1.02446433e-15 1.01120972e-21 1.00000000e+00
  9.58406006e-15 1.10001279e-11]
 [1.01669286e-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 9.98927057e-01 3.29728266e-07 4.19751123e-09
  1.01197371e-03 3.50851333e-05 1.20187156e-06 2.09555239e-07
  2.37075274e-05 2.99694186e-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.]]
```



IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
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
```

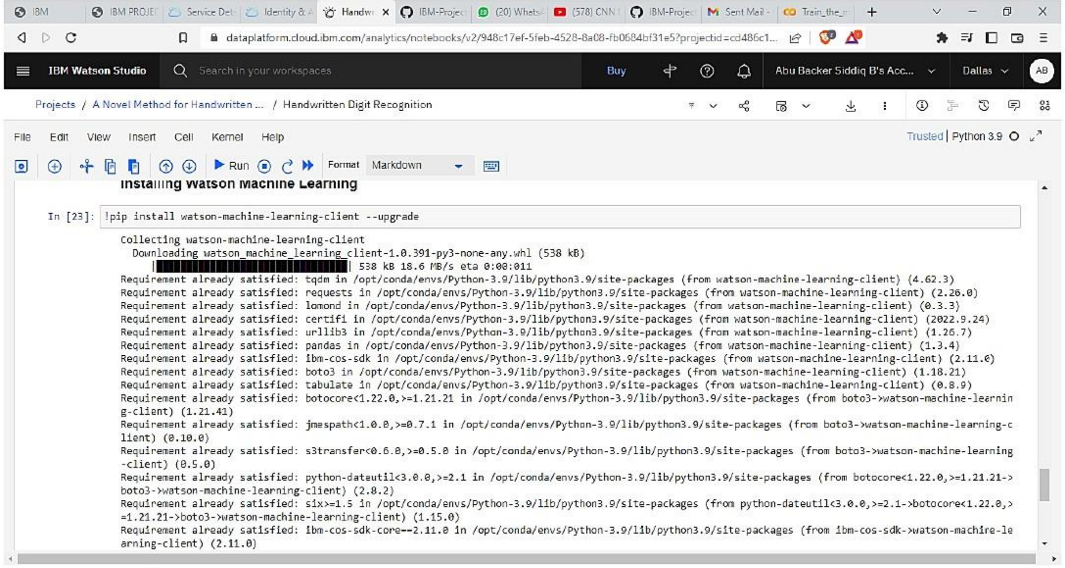
```
In [22]: ls -1
```

```
Handwritten-Digit-Recognition_new.tgz
model.h5
```

Installing Watson Machine Learning ¶

```
In [ ]: !pip install watson-machine-learning-client --upgrade
```

Watson API credentials

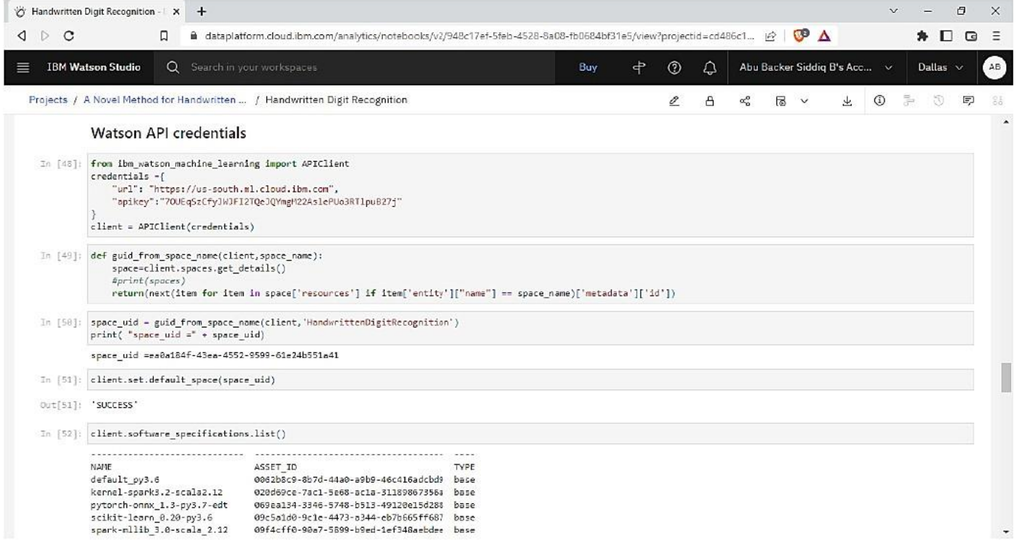


IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition
File Edit View Insert Cell Kernel Help
Trusted | Python 3.9
Run
Format Markdown

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-learnin
g-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-c
lient) (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-machire-le
arning-client) (2.11.0)
```

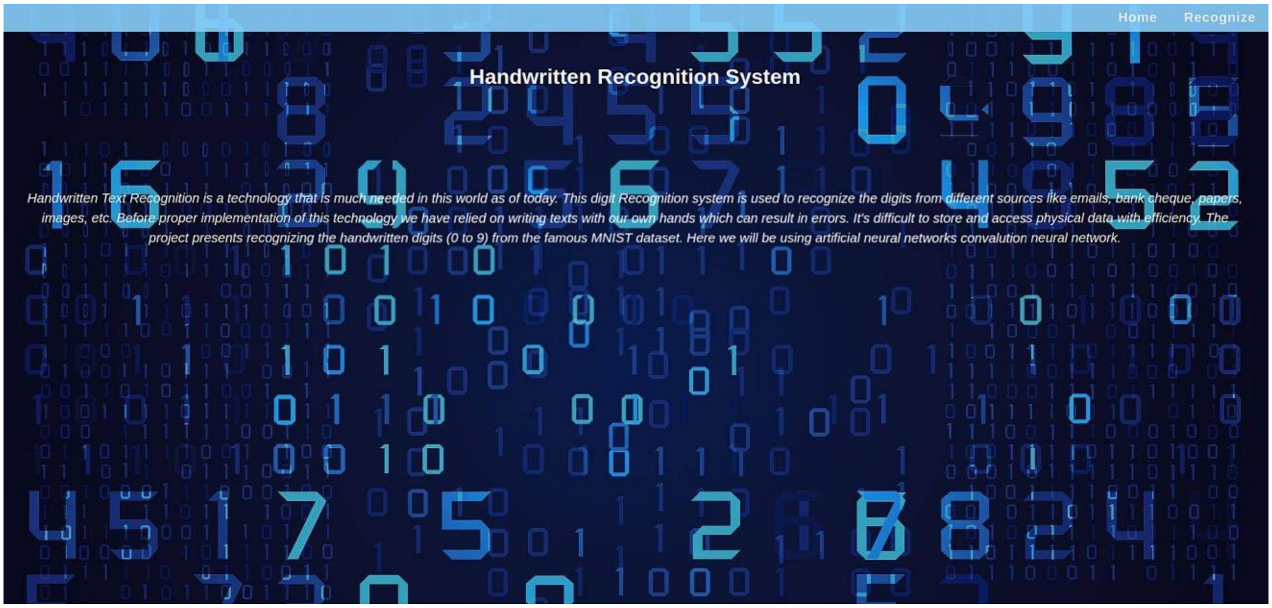
WATSON API CONFIGURATION:



Handwritten Digit Recognition
IBM Watson Studio
Search in your workspaces
Buy
Abu Backer Siddiq B's Acc...
Dallas
Projects / A Novel Method for Handwritten ... / Handwritten Digit Recognition

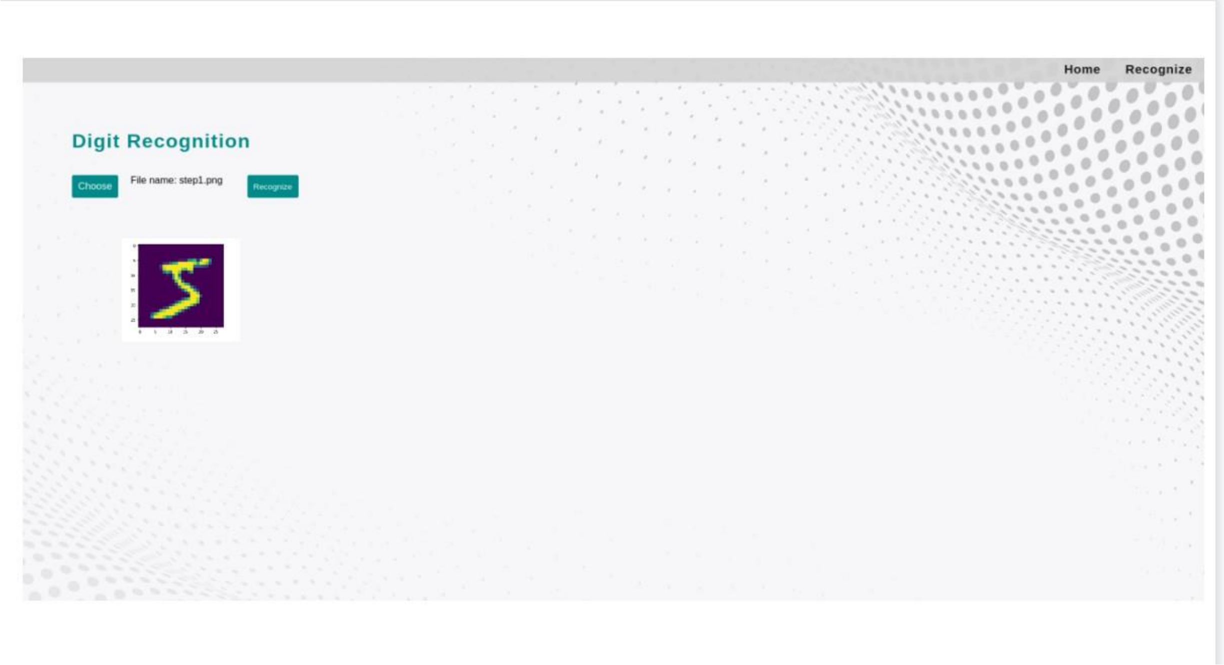
Watson API credentials

```
In [48]: from ibm_watson_machine_learning import APIClient
         credentials ={
             "url": "https://us-south.ml.cloud.ibm.com",
             "apikey":"7OUEqSzCfyJWJFI2TQeJQYmgM22AslePUo3RTlpuB27j"
         }
         client = APIClient(credentials)

In [49]: def guid_from_space_name(client,space_name):
             space=client.spaces.get_details()
             #print(spaces)
             return(next(item for item in space['resources'] if item['entity']["name"] == space_name)['metadata']['id'])

In [50]: space_uid = guid_from_space_name(client,'HandwrittenDigitRecognition')
         print( "space_uid =" + space_uid)

         space_uid =ea0a184f-43ea-4552-9599-61e24b551a41

In [51]: client.set.default_space(space_uid)

Out[51]: 'SUCCESS'

In [52]: client.software_specifications.list()

         ----------------------------  ------------------------------------  ----
         NAME                          ASSET_ID                              TYPE
         default_py3.6                 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
         kernel-spark3.2-scala2.12     020d69ce-7ac1-5e68-ac1a-31189867356a  base
         pytorch-onnx_1.3-py3.7-edt    069ea134-3346-5748-b513-49120e15d288  base
         scikit-learn_0.20-py3.6       09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
         spark-mllib_3.0-scala_2.12    09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
```



Handwritten Recognition System

Handwritten Text Recognition is a technology that is much needed in this world as of today. This digit Recognition system is used to recognize the digits from different sources like emails, bank cheque, papers, images, etc. Before proper implementation of this technology we have relied on writing texts with our own hands which can result in errors. It's difficult to store and access physical data with efficiency. The project presents recognizing the handwritten digits (0 to 9) from the famous MNIST dataset. Here we will be using artificial neural networks convalution neural network.



Digit Recognition

File name: step1.png