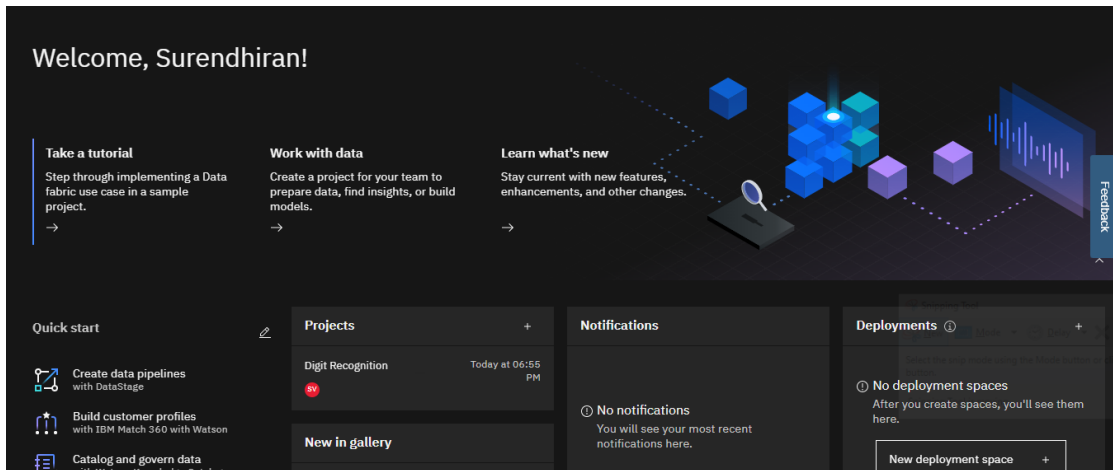


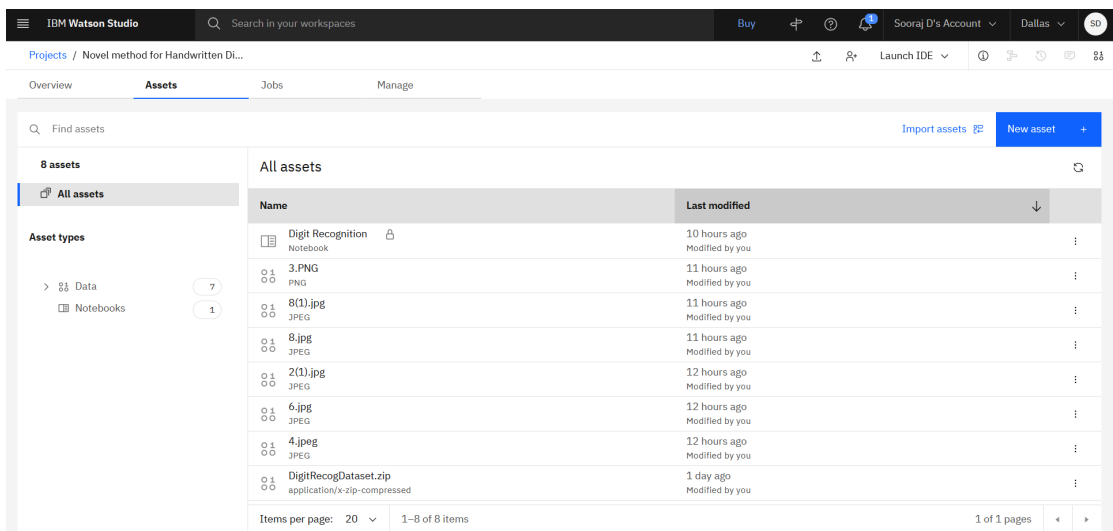
# A Novel Method for Handwritten Digit Recognition

Team ID: PNT2022TMID35407

IBM Cloud Homepage:



IBM Watson Studio:



IBM Notebook:

## LOADING THE DATASET

```
In [66]: import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_names='s3',
                              ibm_api_key_id='g8zhqVn5pbZtmQ0RKL5JZn4MX6TmG1emM2IZ5COKpB',
                              ibm_auth_endpoint='https://iam.cloud.ibm.com/oidc/token',
                              config=Config(signature_version='oauth'),
                              endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'novelmethodforhandwrittendigitrec-donotdelete-pr-sb2ogtldrhegen'
object_key = 'DigitRecogDataset.zip'

streaming_body_1 = cos_client.get_object(Buckets=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/

In [67]: from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)

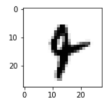
In [68]: train = nd.read_csv("/home/ucuser/work/DigitRecogDataset/train.csv")
```

## DATA PREPROCESSING

```
In [72]: X_train = X_train/255.0
X_test = X_test/255.0

X_train = X_train.values.reshape(-1, 28, 28, 1)
X_test = X_test.values.reshape(-1, 28, 28, 1)

fig, ax = plt.subplots(figsize=(2,2))
plt.imshow(X_train[random.randint(0,len(X_train)), :, :, 0], cmap='Greys', interpolation='nearest')
plt.show()
```



## TRAIN AND TEST SPLIT

```
In [73]: validation_size = 0.2
X_train, X_val, Y_train, Y_val = train_test_split(X_train, Y_train, test_size = validation_size)
```

## ONE HOT ENCODING

```
In [74]: Y_train = keras.utils.np_utils.to_categorical(Y_train, num_classes = 10)
Y_val = keras.utils.np_utils.to_categorical(Y_val, num_classes = 10)
```

## MODEL BUILDING

## TRAINING THE MODEL

```
In [125]: model.fit(X_train, Y_train, validation_data = (X_val, Y_val), epochs = 50)

Epoch 1/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0086 - accuracy: 0.9974 - val_loss: 0.0939 - val_accuracy: 0.9882
Epoch 2/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0069 - accuracy: 0.9978 - val_loss: 0.0968 - val_accuracy: 0.9875
Epoch 3/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0056 - accuracy: 0.9985 - val_loss: 0.0934 - val_accuracy: 0.9874
Epoch 4/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0068 - accuracy: 0.9979 - val_loss: 0.0835 - val_accuracy: 0.9869
Epoch 5/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0055 - accuracy: 0.9982 - val_loss: 0.0982 - val_accuracy: 0.9885
Epoch 6/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0071 - accuracy: 0.9979 - val_loss: 0.1012 - val_accuracy: 0.9873
Epoch 7/50
1050/1050 [=====] - 23s 22ms/step - loss: 0.0066 - accuracy: 0.9980 - val_loss: 0.1014 - val_accuracy: 0.9876
Epoch 8/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0062 - accuracy: 0.9982 - val_loss: 0.0866 - val_accuracy: 0.9888
Epoch 9/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0048 - accuracy: 0.9985 - val_loss: 0.0924 - val_accuracy: 0.9893
Epoch 10/50
1050/1050 [=====] - 23s 22ms/step - loss: 0.0064 - accuracy: 0.9982 - val_loss: 0.1196 - val_accuracy: 0.9863
Epoch 11/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0074 - accuracy: 0.9979 - val_loss: 0.0898 - val_accuracy: 0.9886
Epoch 12/50
1050/1050 [=====] - 21s 20ms/step - loss: 0.0056 - accuracy: 0.9984 - val_loss: 0.1054 - val_accuracy: 0.9886
Epoch 13/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0050 - accuracy: 0.9983 - val_loss: 0.0990 - val_accuracy: 0.9881
Epoch 14/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0040 - accuracy: 0.9990 - val_loss: 0.0863 - val_accuracy: 0.9881
Epoch 15/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0057 - accuracy: 0.9984 - val_loss: 0.1103 - val_accuracy: 0.9890
Epoch 16/50
1050/1050 [=====] - 22s 21ms/step - loss: 0.0063 - accuracy: 0.9985 - val_loss: 0.0977 - val_accuracy: 0.9887
Epoch 17/50
```

Test Case:

IBM Watson Studio

Search in your workspaces

Buy ⓘ ? 🔔 Sooraj D's Account ▾ Dallas ▾ 50

Projects / Novel method for Handwritten Di... / Digit Recognition


```
# @nraoan.ccc
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_botoclient(service_name='s3')
ibm_api_key_id='g8zhqVnSpbZtmQORkLSJZnd4XG6TmG1emM2I25COKpB',
ibm_auth_endpoint='https://iam.cloud.ibm.com/oidc/token',
config=Config(signature_version='oauth'),
endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'novelmethodforhandwrittingdigitrec-donotdelete-pr-sb2ogtldrhgen'
object_key = '3.PNG'

streaming_body_3 = cos_client.get_object(bucket=bucket, key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_botoclient and pandas to learn more about the possibilities to load the data.
# ibm_botoclient documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/

In [148]: img = Image.open(streaming_body_3).convert("L")
img = img.resize( ( 28,28 ) )

In [149]: 
Out[149]: 3

In [150]: im2arr = np.array(img) #converting to image
im2arr = im2arr.reshape(1, 28, 28, 1)

pred = model.predict(im2arr)
print(np.argmax(pred))

3

In [ ]:
```

IBM Deployment space:

IBM Watson Studio

Search in your workspaces

Buy ⓘ ? 🔔 Sooraj D's Account ▾ Dallas ▾ 50

Deployments /

Digit Recognition

Overview

Assets

Deployments

Jobs

Manage

Find assets





Import assets

4 assets

All assets 4

Asset types

Models 4

Name	Last modified	
 Digit Recognition Model	8 hours ago Service	⋮
 Digit Recognition Model	10 hours ago Service	⋮
 Digit Recognition Model	10 hours ago Sooraj D (You)	⋮
 Digit Recognition Model	10 hours ago Service	⋮

Items per page: 20 ▾

1-4 of 4 items

1 of 1 pages