```
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix

import keras
from keras.models import Sequential
from keras.layers import Conv2D, Lambda, MaxPooling2D
from keras.layers import Dense, Dropout, Flatten
from tensorflow.keras.layers import BatchNormalization

from keras.preprocessing.image import ImageDataGenerator

from keras.utils.np_utils import to_categorical

from keras.datasets import mnist
```

Load the data

In [3]:
```
(X_train, Y_train), (X_test, Y_test) = mnist.load_data()
print(X_train.shape)
print(X_test.shape)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11493376/11490434 [==============================] - 0s 0us/step
11501568/11490434 [==============================] - 0s 0us/step
(60000, 28, 28)
(10000, 28, 28)
```

```
468/468 [==============================] - 351s 750ms/step - loss: 0.0373 - accuracy: 0.9884 - val_loss: 0.0353 - val_accuracy: 0.9894 - lr: 0.0010
Epoch 5/10
468/468 [==============================] - 355s 759ms/step - loss: 0.0346 - accuracy: 0.9894 - val_loss: 0.0454 - val_accuracy: 0.9873 - lr: 0.0010
Epoch 6/10
468/468 [==============================] - 353s 754ms/step - loss: 0.0300 - accuracy: 0.9907 - val_loss: 0.0311 - val_accuracy: 0.9903 - lr: 0.0010
Epoch 7/10
468/468 [==============================] - 353s 754ms/step - loss: 0.0305 - accuracy: 0.9911 - val_loss: 0.0338 - val_accuracy: 0.9908 - lr: 0.0010
Epoch 8/10
468/468 [==============================] - 352s 752ms/step - loss: 0.0274 - accuracy: 0.9919 - val_loss: 0.0365 - val_accuracy: 0.9895 - lr: 0.0010
Epoch 9/10
468/468 [==============================] - 353s 753ms/step - loss: 0.0254 - accuracy: 0.9923 - val_loss: 0.0350 - val_accuracy: 0.9893 - lr: 0.0010
Epoch 10/10
468/468 [==============================] - ETA: 0s - loss: 0.0247 - accuracy: 0.9927
Epoch 00010: ReduceLROnPlateau reducing learning rate to 0.00020000000949949026.
468/468 [==============================] - 353s 754ms/step - loss: 0.0247 - accuracy: 0.9927 - val_loss: 0.0363 - val_accuracy: 0.9898 - lr: 0.0010
```

Save the Model

In [13]:
```
model.save("mnistcnn.h5")
```

In [14]:
```
!tar -zcvf model.tgz mnistcnn.h5
```

```
mnistcnn.h5
```

Install necessary packages(IBM WATSON)

ibm cloud login - Yahoo India Se | Service Details - IBM Cloud | Model - IBM Watson Studio

dataplatform.cloud.ibm.com/analytics/notebooks/v2/b9214ca5-99da-40ad-a134-6471bc7378e3?projectid=3b973fe3-c545-41d1-8da9-06e08f004002&context=cpdaas

Gmail YouTube Maps

IBM Watson Studio | Search in your workspaces | Buy | DHANACHEZIAN S's Accou... | Dallas | DS

Projects / A Novel Method For Handwritten ... / Model

File Edit View Insert Cell Kernel Help | Not Trusted | Python 3.9

Run Format Code

```
                    'secret_access_key': '8ba50bfd75dda79178baa328d3470ada4da1226581455ad7',
                    'service_id': 'ServiceId-fe49cce4-b58f-415d-811a-81112b58f718'},
                   'viewer': {'access_key_id': '7587710545614916bc659f81e3f43ce3',
                    'api_key': 'AYWpyBz6md-GnqNbUwT493bddyxmkvtgQX0F6Ydmpa7z',
                    'resource_key_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/9c85d3e704a74497ae0f6d94b3ca1507:4864478b-6f1a-40da-8ccb-bccbfa863b2e::',
                    'secret_access_key': '5f010dd57bde7b012864dbdce955c6e3dd43157aa57637f4',
                    'service_id': 'ServiceId-9b111335-99b8-4d35-a02d-28d6cc764268'}},
                  'endpoint_url': 'https://s3.us-south.cloud-object-storage.appdomain.cloud',
                  'guid': '4864478b-6f1a-40da-8ccb-bccbfa863b2e',
                  'resource_crn': 'crn:v1:bluemix:public:cloud-object-storage:global:a/9c85d3e704a74497ae0f6d94b3ca1507:4864478b-6f1a-40da-8ccb-bccbfa863b2e::'},
                 'type': 'bmcos_object_storage'}},
               'metadata': {'created_at': '2022-11-17T00:31:51.533Z',
                'creator_id': 'IBMid-668000ETK4',
                'id': '69eb14d9-dd12-410e-a848-fc25becb11e5',
                'updated_at': '2022-11-17T00:32:09.260Z',
                'url': '/v2/spaces/69eb14d9-dd12-410e-a848-fc25becb11e5'}}]}
```

In [19]:
```
def guid_from_space_name(client, ibm_deploy):
    space = client.spaces.get_details()
    return(next(item for item in space['resources'] if item['entity']['name'] == ibm_deploy)['metadata']['id'])
```

In [25]:
```
space_uid = guid_from_space_name(client, 'Handwritten-Digit-Recognition')
space_uid
```

Out[25]: '69eb14d9-dd12-410e-a848-fc25becb11e5'

In [26]:
```
client.set.default_space(space_uid)
```

Out[26]: 'SUCCESS'

Search | ENG IN | 14:17 17-11-2022