import keras

from keras.datasets import mnist

from keras.models import Sequential

from keras.layers import Conv2D, Dense, Dropout, MaxPooling2D, Flatten

batch\_size = 128

num\_classes = 10

epochs = 2

(x\_train,y\_train),(x\_test,y\_test)=mnist.load\_data()

print(x\_train.shape)

x\_train = x\_train.astype('float32')

x\_test = x\_test.astype('float32')

x\_train /= 255

x\_test /= 255

print(x\_train.shape)

print(x\_train.shape[0], 'train samples')

print(x\_test.shape[0], 'test samples')

y\_train = keras.utils.to\_categorical(y\_train, num\_classes)

y\_test = keras.utils.to\_categorical(y\_test, num\_classes)

x\_train = x\_train.reshape(x\_train.shape[0], 28, 28, 1)

x\_test = x\_test.reshape(x\_test.shape[0], 28, 28, 1)

model = Sequential()

model.add(Conv2D(filters = 6, kernel\_size = 5, strides = 1, activation = 'relu', input\_shape = (28,28,1)))

model.add(MaxPooling2D(pool\_size = 2, strides = 2))

model.add(Conv2D(filters = 16, kernel\_size = 5, strides = 1, activation = 'relu'))

#Pooling Layer 2

model.add(MaxPooling2D(pool\_size = 2, strides = 2))

#Flatten

model.add(Flatten())

model.add(Dense(512, activation='relu'))

model.add(Dropout(0.2))

model.add(Dense(num\_classes, activation='softmax'))

model.summary()