**Practical 7: Convolutional Neural Network**

**Implementation:**

import keras

from keras.datasets import mnist

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten

from keras.layers import Conv2D, MaxPooling2D

import numpy as np

batch\_size = 128

num\_classes = 10

epochs = 12

*# input image dimensions*

img\_rows, img\_cols = 28, 28

*# the data, split between train and test sets*

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

x\_train = x\_train.reshape(60000, 28, 28, 1)

x\_test = x\_test.reshape(10000, 28, 28, 1)

print("x\_train shape:", x\_train.shape)

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

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

*# convert class vectors to binary class matrices*

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

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

model = Sequential()

model.add(Conv2D(32, *kernel\_size*=(3, 3), *activation*="relu", *input\_shape*=(28, 28, 1)))

model.add(Conv2D(64, (3, 3), *activation*="relu"))

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

model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(128, *activation*="relu"))

model.add(Dropout(0.5))

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

model.compile(

*loss*=keras.losses.categorical\_crossentropy,

*optimizer*=keras.optimizers.Adadelta(),

*metrics*=["accuracy"],

)

model.fit(

    x\_train,

    y\_train,

*batch\_size*=batch\_size,

*epochs*=epochs,

*verbose*=1,

*validation\_data*=(x\_test, y\_test),

)

score = model.evaluate(x\_test, y\_test, *verbose*=0)

print("Test loss:", score[0])

print("Test accuracy:", score[1])

**Output**



