

Project Development Phase Sprint-2

Date	05 November 2022
Team ID	PNT2022TMID40411
Project Name	A Novel Method for Handwritten Digit Recognition System

The screenshot displays a Google Colab notebook interface for a project titled "Handwritten Digit Recognition.ipynb". The left sidebar shows a file explorer with a directory structure including "data" (containing 0.png, 1.png, 2.png, 3.png), "drive", "models" (containing mnistCNN.h5), and "sample_data". The main workspace is divided into three sections: "Creating the Model", "Compiling the Model", and "Fitting the Model".

Creating the Model: This section contains a code cell that defines a sequential CNN model. The code is as follows:

```
#create model
model=Sequential()
#adding model Layer
model.add(Conv2D (64, (3, 3), input_shape=(28, 28, 1), activation='relu'))
model.add(Conv2D (32, (3, 3), activation='relu'))
#model.add(Conv2D (32, (3, 3), activation='relu'))
#flatten the dimension of the image
model.add(Flatten())
#output Layer with 10 neurons
model.add(Dense(number_of_classes, activation='softmax'))
```

Compiling the Model: This section shows a code cell that compiles the model with categorical crossentropy loss, Adam optimizer, and accuracy metrics.

```
[11] #compile model
model.compile(loss='categorical_crossentropy', optimizer="Adam", metrics=['accuracy'])
```

Fitting the Model: This section shows a code cell that fits the model to the training and validation data.

```
[12] #fit the model
model.fit(x_train,y_train, validation_data=(X_test,y_test),epochs=5,batch_size=32)
```

The notebook interface includes a top navigation bar with various icons and a bottom status bar showing the system temperature (31°C Cloudy), time (13:55), and date (05-11-2022).

colab.research.google.com/drive/1huptOJ_rj6ly4wertUbqSxVhbaj9n5X#scrollTo=VVILUWpCjmYv

Handwritten Digit Recognition.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

{x}

data

0.png

1.png

2.png

3.png

drive

models

mnistCNN.h5

sample_data

+ Code + Text

RAM

Disk

Editing

Fitting the Model

↑ ↓ ↶ ↷ ↻ ↺ ↻ ↺

[12] #fit the model

model.fit(x_train,y_train, validation_data=(X_test,y_test),epochs=5,batch_size=32)

Epoch 1/5

1875/1875 [=====] - 227s 121ms/step - loss: 0.2526 - accuracy: 0.9527 - val_loss: 0.0826 - val_accuracy: 0.9759

Epoch 2/5

1875/1875 [=====] - 208s 111ms/step - loss: 0.0645 - accuracy: 0.9809 - val_loss: 0.0733 - val_accuracy: 0.9788

Epoch 3/5

1875/1875 [=====] - 208s 111ms/step - loss: 0.0470 - accuracy: 0.9849 - val_loss: 0.0722 - val_accuracy: 0.9788

Epoch 4/5

1875/1875 [=====] - 204s 109ms/step - loss: 0.0374 - accuracy: 0.9878 - val_loss: 0.1071 - val_accuracy: 0.9773

Epoch 5/5

1875/1875 [=====] - 210s 112ms/step - loss: 0.0270 - accuracy: 0.9913 - val_loss: 0.0944 - val_accuracy: 0.9792

<keras.callbacks.History at 0x7f75dd9dcf10>

Observing the Metrics

[13] # final evaluation of the model

metrics = model.evaluate(X_test, y_test, verbose=0)

print("Metrics(Test loss & Test Accuracy): ")

print(metrics)

Metrics(Test loss & Test Accuracy):

[0.09444485604763031, 0.979200005531311]

1s completed at 1:08 PM

Show all

31°C Cloudy

Windows Taskbar

ENG IN 13:56 05-11-2022

colab.research.google.com/drive/1huptOJ_rj6ly4wertUbqSxVhbaj9n5X#scrollTo=t3WKvA1r45Oy

Handwritten Digit Recognition.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

data

0.png

1.png

2.png

3.png

drive

models

mnistCNN.h5

sample_data

85.14 GB available

Taking images as inputs and checking results

```
[17] #importing the keras libraries and packages
from tensorflow.keras.models import load_model
model = load_model(r'/content/models/mnistCNN.h5')
from PIL import Image #used for manipulating image uploaded by the user
import numpy as np #used for numerical analysis
for index in range(4):
    img = Image.open('data/' + str(index) + '.png').convert("L") #convert image to monochrome
    img=img.resize((28,28))# resizing of input image
    im2arr=np.array(img) #converting to image
    im2arr = im2arr.reshape(1,28,28,1) #reshaping according to our requirement
    #Predicting the Test set results
    y_pred=model.predict(im2arr) #predicting the results
    print(y_pred)

1/1 [=====] - 0s 76ms/step
[[9.6278855e-15 1.9075532e-21 1.6254842e-22 4.2402232e-12 3.8528382e-19
 1.0000000e+00 6.7038224e-18 5.1701430e-19 1.4048756e-11 1.6468845e-09]]
1/1 [=====] - 0s 26ms/step
[[1.0000000e+00 6.8269191e-19 2.5597378e-12 6.9047195e-15 2.0105160e-14
 8.2670402e-14 2.5793861e-08 2.8955508e-11 2.4723973e-10 4.0948384e-11]]
1/1 [=====] - 0s 23ms/step
[[8.7009194e-16 1.6748922e-19 2.6306970e-19 9.8592958e-21 1.0000000e+00
 3.6940159e-13 1.3179105e-21 3.6858478e-15 3.5922146e-18 7.2568997e-15]]
1/1 [=====] - 0s 27ms/step
[[6.0339331e-18 9.9999917e-01 5.7398509e-20 1.5878209e-09 4.8178167e-10
 1.0082240e-13 3.6611254e-15 3.3697142e-07 4.2479573e-07 4.0238507e-12]]
```

1s completed at 1:08 PM

Handwritten_Dig...ipynb

Handwritten_Dig...ipynb

Show all

31°C Cloudy

ENG IN

13:57 05-11-2022