PRACTICAL 8

Conversion of Handwritten Digit to CSV Format

ACTIVITY - Submission Required!

The popular MNIST dataset can be downloaded in two formats: Image and CSV. The dataset is in grayscale format. In this exercise, we will be converting the Dzongkha handwritten digit into CSV (Comma Separated Values). In this tutorial, we will be using three different classes of Dzongkha handwritten digits (zero, one and two).

Dataset Details:

The dataset contains 100 images per class (a subset of the original dataset) and was created by Mr. Yonten Jamtsho, Ms. Sonam Wangmo, Ms. Pema Yangden and Mrs. Nima Dema (Faculty of GCIT) for the Annual University Research Grant. The images in the dataset have been already resized to 28 by 28 dimensions and are in grayscale format.

Perform the following steps to complete this exercise:

1. Click here to download the dataset (zip folder). Save the zipped folder inside the working directory (Google Drive).
2. Open a new Jupyter Notebook file.
3. Mount the Google Drive.

```
from google.colab import drive
drive.mount('/content/drive')
path = ".................."
```

NOTE: The path value will vary based on your working directory.

4. Import the required libraries.

```
import zipfile
import pandas as pd
import glob
from PIL import Image
import numpy as np
import csv
import matplotlib.pyplot as plt
```

5. Extract the zip folder from the Google Colab and save it back to the working directory.

```
import zipfile
```

```
with zipfile.ZipFile(path+"Dzo_MNIST.zip", 'r') as zipObj:
  zipObj.extractall(path)
```

6. Define the names of the columns.

```
# define an empty list
column_names = list()
# Loop up to 784
# 28 * 28 = 784
for i in range(784):
  pixel = 'pixel-'
  pixel += str(i)
  column_names.append(pixel)
print(column_names)
```

Expected Output:

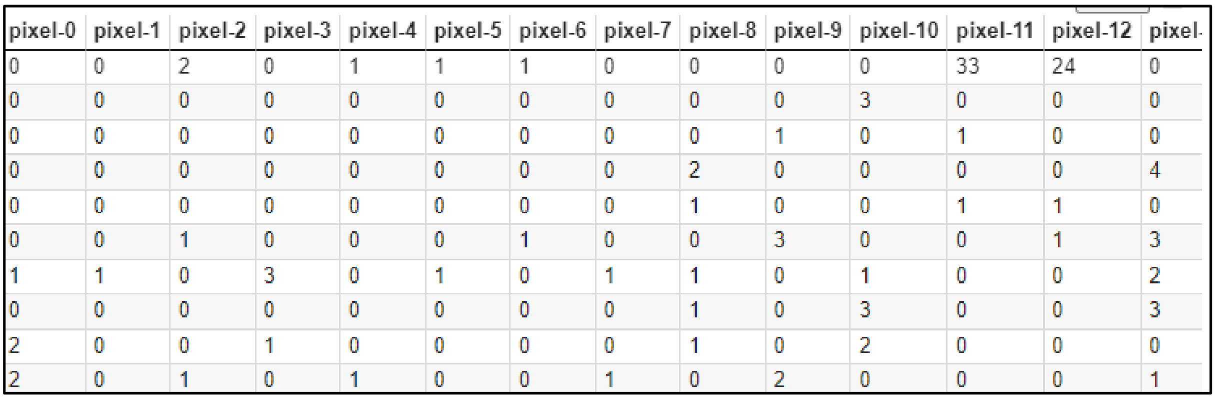
```
['pixel-0', 'pixel-1', 'pixel-2', 'pixel-3', 'pixel-4', 'pixel-5', 'pixel-6',
'pixel-7', 'pixel-8',......... 'pixel-783']
```

7. Convert the pixels of handwritten digits to a CSV file.

```
# Create a DataFrame
df = pd.DataFrame(columns = column_names)

# NOTE: You need to change the directory
filenames = [img for img in glob.glob(path+"/0/*.jpg")]

j = 0
for i in filenames:
  img = Image.open(i)
  rawData = img.load()
  data = []

  for y in range(28):
    for x in range(28):
      data.append(rawData[x, y][0])

  k = 0
  df.loc[j] = [data[k] for k in range(784)]
  j = j + 1
```

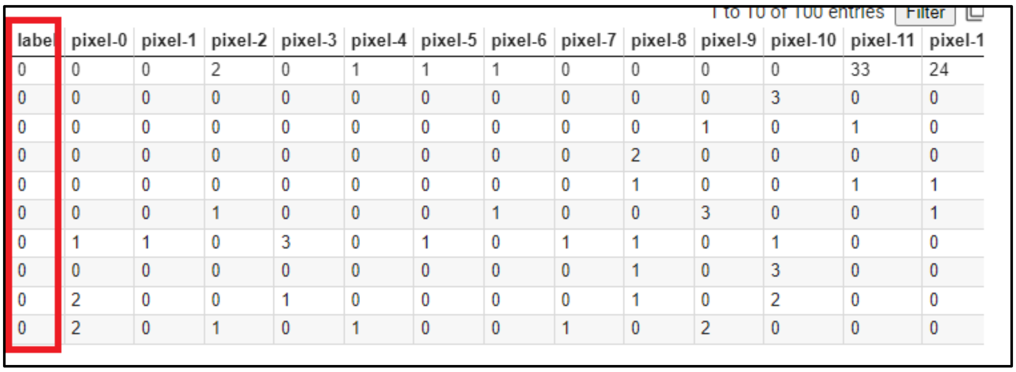
```
# Save to CSV
df.to_csv("zero.csv", index = False)
```

A zero.csv file will be created. You can open the CSV file using the Jupyter Notebook.



pixel-0	pixel-1	pixel-2	pixel-3	pixel-4	pixel-5	pixel-6	pixel-7	pixel-8	pixel-9	pixel-10	pixel-11	pixel-12	pixel-
0	0	2	0	1	1	1	0	0	0	0	33	24	0
0	0	0	0	0	0	0	0	0	0	3	0	0	0
0	0	0	0	0	0	0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	0	2	0	0	0	0	4
0	0	0	0	0	0	0	0	1	0	0	1	1	0
0	0	1	0	0	0	1	0	0	3	0	0	1	3
1	1	0	3	0	1	0	1	1	0	1	0	0	2
0	0	0	0	0	0	0	0	1	0	3	0	0	3
2	0	0	1	0	0	0	0	1	0	2	0	0	0
2	0	1	0	1	0	0	1	0	2	0	0	0	1

NOTE: Open the CSV and add a label as the first column. Fill the column with the class name.



1 to 10 of 100 entries Filter

label	pixel-0	pixel-1	pixel-2	pixel-3	pixel-4	pixel-5	pixel-6	pixel-7	pixel-8	pixel-9	pixel-10	pixel-11	pixel-1
0	0	0	2	0	1	1	1	0	0	0	0	33	24
0	0	0	0	0	0	0	0	0	0	0	3	0	0
0	0	0	0	0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	0	0	2	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	1	1
0	0	0	1	0	0	0	1	0	0	3	0	0	1
0	1	1	0	3	0	1	0	1	1	0	1	0	0
0	0	0	0	0	0	0	0	0	1	0	3	0	0
0	2	0	0	1	0	0	0	0	1	0	2	0	0
0	2	0	1	0	1	0	0	1	0	2	0	0	0

8. Now, all the pixel representation of an image is converted to a CSV file. The code below will plot the CSV.

Note: From the zero.csv file, remove the column header.

```
import numpy as np
import csv
import matplotlib.pyplot as plt

with open('zero.csv', 'r') as csv_file:
    # Skip header
    next(csv_file)
    for data in csv.reader(csv_file):
        # The first column is the label
        label = data[0]

        # The rest of columns are pixels
        pixels = data[0:]
        # Make those columns into a array of 8-bits pixels
        # This array will be of 1D with length 784
        # The pixel intensity values are integers from 0 to 255
        pixels = np.array(pixels, dtype='uint8')

        # Reshape the array into 28 x 28 array (2-dimensional array)
        pixels = pixels.reshape((28, 28))

        # Plot
        plt.title('Label is {label}'.format(label=label))
        plt.imshow(pixels, cmap='gray')
        plt.show()

        break # This stops the loop, I just want to see one
```

Activity

1. Generate CSV files for digits one and two. In total, there should be three CSV files.
2. Write a Python program to merge the three CSV files that you have created in Q1. You should have a single CSV file containing a pixel representation for all the 300 digits.
3. Write a Python program to convert the CSV file back to the image.

****