DigitRecognition-dataload

January 26, 2020

1 Digit Recognition - Load new testdata!

Sample code given below to show how to load a test image dataset from outside.

```
In [30]: # TensorFlow and tf.keras
         import tensorflow as tf
         from tensorflow import keras
         import sys
         import os
         from PIL import Image, ImageOps

         # Helper libraries
         import numpy as np
         import matplotlib.pyplot as plt
         import random
         print(tf.__version__)
```

```
2.1.0
```

```
In [35]: # To load images to features and labels
         def load_images_to_data(image_label, image_directory, features_data, label_data):
             list_of_files = os.listdir(image_directory)
             for file in list_of_files:
                     full_path = image_directory
                     image_file_name = os.path.join(full_path, file)
                     if ".png" in image_file_name:
                         img = Image.open(image_file_name)
                             #.convert("L")
                         img = img.resize((28,28))
                         img = ImageOps.invert(img)
                         im2arr = np.array(img)
                         im2arr = im2arr[:,:,0]
                         print ("---------")
                         print(im2arr.shape)
                         print("---------")
                         im2arr = im2arr.reshape(1,28,28)
```

```
                    features_data = np.append(features_data, im2arr, axis=0)
                    label_data = np.append(label_data, [image_label], axis=0)
            return features_data, label_data
```

```
In [36]: # Keras provides a handy API to download the MNIST dataset, and split them into
         # "train" dataset and "test" dataset.
         mnist = keras.datasets.mnist
         (train_images, train_labels), (test_images,
                                        test_labels) = mnist.load_data()

         print (test_labels.shape)
         print(test_images.shape)
         print (type(train_images))
         print (type(test_labels))
```

```
(10000,)
(10000, 28, 28)
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
```

1.0.1 Load the new test data

Please find code below that will load each test user one by one separately.

```
In [43]: user1_data = np.zeros(28*28).reshape(1,28,28)
         user1_label = np.zeros(1)
         DATA_IMG_PATH = "data_images/User1/"

         for path in os.listdir(DATA_IMG_PATH):
             full_path = os.path.join(DATA_IMG_PATH, path)
             user1_data, user1_label = load_images_to_data(
                 int(path), full_path, user1_data, user1_label)
         user1_data = np.delete(user1_data, 0, axis=0)
         user1_label = np.delete(user1_label, 0, axis= 0)
         print ("++++++++++++++++++")
         print(user1_data.shape)
         print (user1_label.shape)
         print ("++++++++++++++++++")
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
++++++++++++++++++
(142, 28, 28)
(142,)
++++++++++++++++++
```

```
In [42]: user2_data = np.zeros(28*28).reshape(1,28,28)
         user2_label = np.zeros(1)
         DATA_IMG_PATH = "data_images/User2/"

         for path in os.listdir(DATA_IMG_PATH):
             full_path = os.path.join(DATA_IMG_PATH, path)
             user2_data, user2_label = load_images_to_data(
                 int(path), full_path, user2_data, user2_label)
         user2_data = np.delete(user2_data, 0, axis=0)
```

```
        user2_label = np.delete(user2_label, 0, axis= 0)
        print ("+++++++++++++++++")
        print(user2_data.shape)
        print (user2_label.shape)
        print ("+++++++++++++++++")
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
++++++++++++++++++
(8, 28, 28)
(8,)
++++++++++++++++++
```

```
In [45]: user3_data = np.zeros(28*28).reshape(1,28,28)
         user3_label = np.zeros(1)
         DATA_IMG_PATH = "data_images/User3/"

         for path in os.listdir(DATA_IMG_PATH):
             full_path = os.path.join(DATA_IMG_PATH, path)
             user3_data, user3_label = load_images_to_data(
                 int(path), full_path, user3_data, user3_label)
         user3_data = np.delete(user3_data, 0, axis=0)
         user3_label = np.delete(user3_label, 0, axis= 0)
         print ("+++++++++++++++++")
         print(user3_data.shape)
```

```
        print (user3_label.shape)
        print ("++++++++++++++++++")
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
```

```
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
---------
(28, 28)
---------
++++++++++++++++++
(41, 28, 28)
(41,)
++++++++++++++++++
```

```
In [46]: print("User1:")
         print(user1_label)
         print("User2:")
         print(user2_label)
         print("User2:")
         print(user3_label)
```

```
User1:
[9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9. 9.
 9. 9. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 7. 7. 7. 7. 7. 7.
 7. 7. 7. 7. 7. 7. 7. 7. 7. 7. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6. 6.
 6. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 8. 8. 8. 8. 8. 8. 8. 8. 8. 8. 8. 8.
 8. 8. 8. 8. 8. 8. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 3. 3. 3. 3. 3.
 3. 3. 3. 3. 3. 3. 3. 3. 2. 2. 2. 2. 2. 2. 2. 2. 2. 5. 5. 5. 5. 5.]
User2:
[0. 0. 7. 6. 1. 8. 8. 3.]
User2:
[9. 9. 9. 9. 9. 0. 0. 0. 7. 7. 6. 6. 6. 1. 1. 1. 1. 8. 8. 8. 4. 4. 4. 4.
 4. 3. 3. 3. 3. 3. 3. 3. 3. 2. 2. 2. 2. 2. 2. 5. 5.]
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```