

PROJECT DEVELOPMENT PHASE

SPRINT-1 DATA COLLECTION,IMAGE PREPROCESSING

TEAM -ID :	PNT2022TMID15785
PROJECT NAME:	AI-powered Nutrition Analyzer for Fitness Enthusiasts

P.Lokesh Reddy
J.Nitish Gopala Krishna
M.Saketh Chowdary
Vishnu deepak
Mahendra

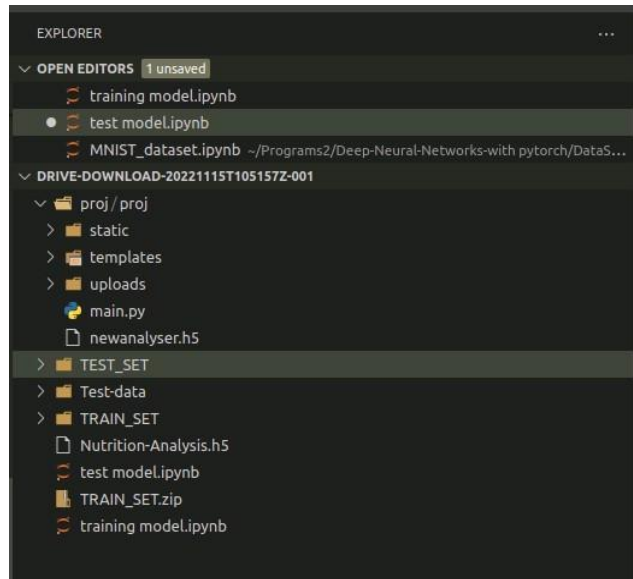
DATA COLLECTION

Data Collection Collect images of different food items organized into subdirectories based on their respective names as shown in the project structure. Create folders of types of food items that need to be recognized. In this project, we have collected images of 5 types of food items apples, 'banana', 'orange', , they are saved in the respective subdirectories with their respective names.

DATASET LINK:

The link provided for accessing the test and train datasets for the CNN model is :

https://drive.google.com/drive/folders/1yNVuLA2hxIstOcDV58enyD7_4Y9drEs6Y?usp=sharing



Shared with me > Dataset > TEST_SET > ORANGE ▾ 👤











Name ↑	Owner	Last modified	File size
 3_100.jpg 👤	Shivani Kapoor	10 Jun 2020 Shivani Kapoor	5 KB
 4_100.jpg 👤	Shivani Kapoor	10 Jun 2020 Shivani Kapoor	5 KB
 5_100.jpg 👤	Shivani Kapoor	10 Jun 2020 Shivani Kapoor	5 KB
 6_100.jpg 👤	Shivani Kapoor	10 Jun 2020 Shivani Kapoor	5 KB
 7_100.jpg 👤	Shivani Kapoor	10 Jun 2020 Shivani Kapoor	5 KB
 8_100.jpg 👤	Shivani Kapoor	10 Jun 2020 Shivani Kapoor	5 KB
 9_100.jpg 👤	Shivani Kapoor	10 Jun 2020 Shivani Kapoor	5 KB
 30_100.jpg 👤	Shivani Kapoor	10 Jun 2020 Shivani Kapoor	5 KB

IMAGE PREPROCESSING

We will be improving the image data that suppresses unwilling distortions or enhances some image features important for further processing, although performing some geometric transformations of images like rotation, scaling, translation, etc.

- **Import The ImageDataGenerator Library**
- **Configure ImageDataGenerator Class**
- **Apply Image DataGenerator Functionality To Training dataset And Testing dataset**

The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class.

There are five main types of data augmentation techniques for image data; specifically:

Image shifts via the `width_shift_range` and `height_shift_range` arguments. The image flips via the `horizontal_flip` and `vertical_flip` arguments. Image rotations via the `rotation_range` argument. Image brightness via the `brightness_range` argument. Image zoom via the `zoom_range` argument.

Let us apply ImageDataGenerator functionality to Train dataset and Test dataset by using the following code

For Training set using flow_from_directory function.

```
1 import numpy as np
2 import tensorflow
3 from tensorflow.keras.models import Sequential
4 from tensorflow.keras import layers
5 from tensorflow.keras.layers import Dense, Flatten
6 from tensorflow.keras.layers import Conv2D, MaxPooling2D
7 from keras.preprocessing.image import ImageDataGenerator
```

```
1 train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
2 test_datagen=ImageDataGenerator(rescale=1./255)
```

```
1 x_train = train_datagen.flow_from_directory(r'TRAIN_SET',
2                                           target_size=(64, 64),
3                                           batch_size=5,
4                                           color_mode='rgb',
5                                           class_mode='sparse')
6
7 x_test = test_datagen.flow_from_directory(r'TEST_SET',
8                                          target_size=(64, 64),
9                                          batch_size=5,
10                                         color_mode='rgb',
11                                         class_mode='sparse')
12
```

```
Found 2626 images belonging to 5 classes.
Found 1055 images belonging to 5 classes.
```

```
1 print(x_train.class_indices)
```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
```

```
1 from collections import Counter as ct
2 ct(x_train.labels)
```

```
Counter({0: 606, 1: 445, 2: 479, 3: 621, 4: 475})
```