

PROJECT DEVELOPMENT PHASE

SPRINT 1

- DATA COLLECTION
- IMAGE PREPROCESSING

TOPIC : AI powered nutrition analyzer for fitness enthusiasts











Team id : PNT2022TMID15800

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:

<https://drive.google.com/drive/folders/1yNVuLA2hxIstOcDV58enyD74Y9drEs6Y?usp=sharing>

- ▼  MyDrive
 - ▶  106106182
 - ▶  Colab Notebooks
 - ▼  Dataset
 - ▶  TEST_SET
 - ▶  TRAIN_SET
 -  IBM_review.pptx
 -  Nutrition.h5
 -  nutrition(1).h5
 -  photo-1589820296156

My Drive > Dataset > TEST_SET > APPLES ▾

Files

Name ↑



 n07740461_20.jpg



 n07740461_40.jpg




 n07740461_41.jpg



 n07740461_51.jpg



 n07740461_81.jpg



 n07740461_91.jpg



Image Preprocessing

we will be improve 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.

1)Import The ImageDataGenerator Library

2)Configure ImageDataGenerator Class

3)Apply Image DataGenerator Functionality To Trainset And Testset

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

Let us import the ImageDataGenerator class from Keras

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 Trainset and Testset by using the following code

For Training set using flow_from_directory function.

+ Code + Text

✓ RAM
Disk Editing ^

```
[ ] from keras.preprocessing.image import ImageDataGenerator
```

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

```
[ ] test_datagen=ImageDataGenerator(rescale=1./255)
```

```
[1] from google.colab import drive  
drive.mount('/content/drive')
```

Mounted at /content/drive

```
import pandas as pd
```

```
[ ] Data_trainpath='/content/drive/MyDrive/Dataset/TRAIN_SET'
```

```
[ ] Data_testpath='/content/drive/MyDrive/Dataset/TEST_SET'
```

```
[ ] x_train = train_datagen.flow_from_directory(Data_trainpath, target_size=(64,64), batch_size=5, color_mode='rgb', class_mode='sparse')  
Found 1711 images belonging to 3 classes.
```

```
[ ] x_test = train_datagen.flow_from_directory(Data_testpath, target_size=(64,64), batch_size=5, color_mode='rgb', class_mode='sparse')  
Found 929 images belonging to 3 classes.
```

```
[ ] print(x_train.class_indices)
```

+ Code + Text

✓ RAM
Disk Editing

import pandas as pd

[] Data_trainpath='/content/drive/MyDrive/Dataset/TRAIN_SET'

[] Data_testpath='/content/drive/MyDrive/Dataset/TEST_SET'

```
[ ] x_train = train_datagen.flow_from_directory(Data_trainpath,target_size=(64,64),batch_size=5,color_mode='rgb',class_mode='sparse')
Found 1711 images belonging to 3 classes.
```

```
[ ] x_test = train_datagen.flow_from_directory(Data_testpath,target_size=(64,64),batch_size=5,color_mode='rgb',class_mode='sparse')
Found 929 images belonging to 3 classes.
```

```
[ ] print(x_train.class_indices)
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2}
```

```
[ ] print(x_test.class_indices)
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2}
```

```
[ ] from collections import Counter as c
c(x_train.labels)
Counter({1: 692, 2: 1019})
```