

Image Preprocessing

Team Id	PNT2022TMID07306
Project Name	AI-powered Nutrition Analyzer for Fitness Enthusiasts

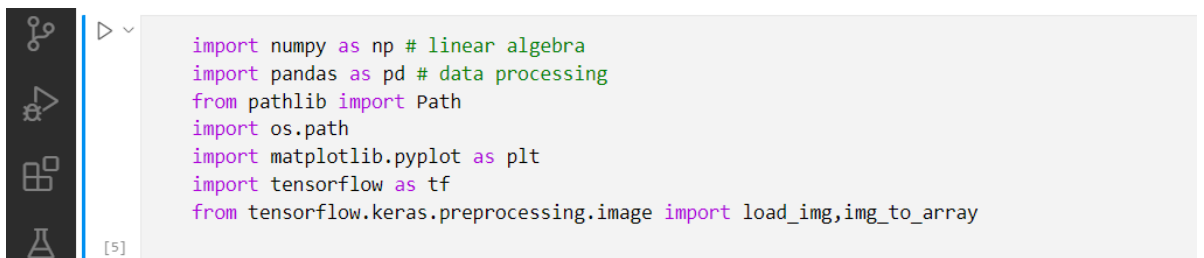
Loading and pre-processing the data:

Data is gold as far as deep learning models are concerned. Your image classification model has a far better chance of performing well if you have a good amount of images in the training set. Also, the shape of the data varies according to the architecture/framework that we use.

Hence, the critical data pre-processing step (the eternally important step in any project). I highly recommend going through the “basics of image processing using Python we use Keras’ ImageDataGenerator class to perform data augmentation. i.e, we are using some kind of parameters to process our collected data. The word “augment” means to make something “greater” or “increase” something (in this case, data), the Keras ImageDataGenerator class actually works by:

- Accepting a batch of images used for training.
- Taking this batch and applying a series of random transformations to each image in the batch (including random rotation, resizing, shearing, etc.).
- Replacing the original batch with the new, randomly transformed batch.
- Training the CNN on this randomly transformed batch (i.e., the original data itself is not used for training)

❖ Import the library

A screenshot of a Jupyter Notebook interface. On the left is a dark sidebar with icons for file explorer, search, and other notebook functions. The main area shows a code cell with the following Python code:

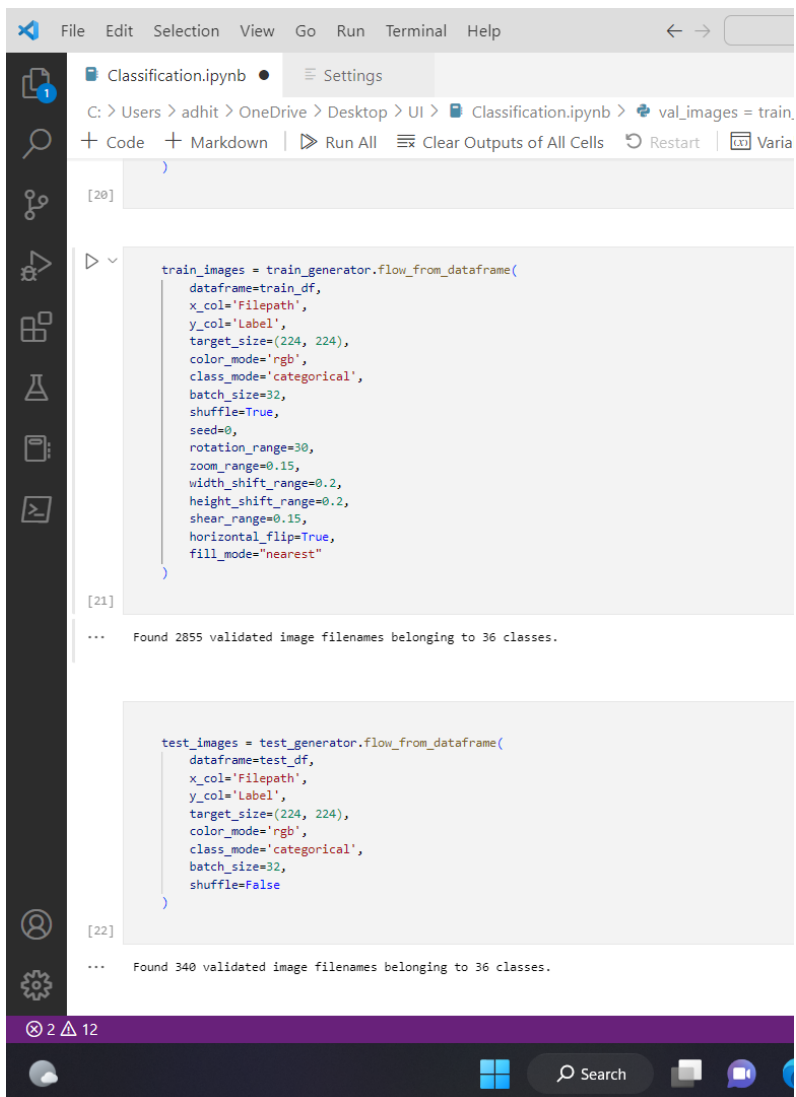
```
import numpy as np # linear algebra
import pandas as pd # data processing
from pathlib import Path
import os.path
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.preprocessing.image import load_img, img_to_array
```

Below the code, the cell number [5] is visible.

❖ Define the parameters /arguments

```
[20] train_generator = tf.keras.preprocessing.image.ImageDataGenerator(  
    preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input  
)  
  
test_generator = tf.keras.preprocessing.image.ImageDataGenerator(  
    preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input  
)
```

❖ Applying functionality to trainset and testset



The screenshot shows a Jupyter Notebook window titled 'Classification.ipynb'. The interface includes a menu bar (File, Edit, Selection, View, Go, Run, Terminal, Help), a toolbar with icons for file operations, search, and execution, and a sidebar with a file explorer. The notebook content is as follows:

```
[20] )
```

```
[21] train_images = train_generator.flow_from_dataframe(  
    dataframe=train_df,  
    x_col='Filepath',  
    y_col='Label',  
    target_size=(224, 224),  
    color_mode='rgb',  
    class_mode='categorical',  
    batch_size=32,  
    shuffle=True,  
    seed=0,  
    rotation_range=30,  
    zoom_range=0.15,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    shear_range=0.15,  
    horizontal_flip=True,  
    fill_mode='nearest'  
)
```

... Found 2855 validated image filenames belonging to 36 classes.

```
[22] test_images = test_generator.flow_from_dataframe(  
    dataframe=test_df,  
    x_col='Filepath',  
    y_col='Label',  
    target_size=(224, 224),  
    color_mode='rgb',  
    class_mode='categorical',  
    batch_size=32,  
    shuffle=False  
)
```

... Found 340 validated image filenames belonging to 36 classes.