

# The Import Image Data Generator Library

Date	23 OCTOBER 2022
Team Id	PNT2022TMID41375
Project Name	AI-POWERED NUTRITION ANALYZER FOR FITNESS ENTHUSIASTS
Maximum Marks	4 MARKS

- ✓ Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset.
- ✓ 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

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

## Importing Relevant Libraries:

- ✓ Let's start with importing different libraries we need to annotate and visualize our images.
- ✓ You must all be familiar with NumPy # Numerical Python, Pandas, a data analysis and manipulation tool, and Matplotlib for visualization.
- ✓ Glob module, as its name suggests it's the shortest form of Global which means it searches all the file paths for the given pattern.
- ✓ CV2 is another version of the CV which comes in handy for image processing.
- ✓ OS, as you know, stands for the operating system and is Python standard library .

## CODE:

- ✓ Pigeon, as mentioned, is used for annotation. It is a python library and can be installed by Github or by pip. PigeonXT is just an extension of Pigeon and helps us to annotate the unlabelled dataset.
- ✓ pip install pigeonXT-jupyter
- ✓ Python Imaging Library (PIL) modules provide an image library for editing.
- ✓ From IPython.display module, we will import the display and image used for displaying data and images. Here is the [dataset](#), which we'll be using.

```
[1]: import os
import glob
import cv2
import numpy as np
import pandas as pd
from PIL import Image
from pathlib import Path
from keras.models import Sequential, Model, load_model
from keras.applications.vgg16 import VGG16, preprocess_input
from keras.preprocessing.image import ImageDataGenerator, load_img, img_to_array
from keras.layers import Conv2D, MaxPooling2D, Dense, Dropout, Input, Flatten, Activation
from keras.optimizers import Adam, SGD, RMSprop
from keras.callbacks import Callback, EarlyStopping
from keras.utils import to_categorical
from sklearn.metrics import confusion_matrix
from keras import backend as K
import tensorflow as tf
import matplotlib.pyplot as plt
matplotlib.rcParams['font.family'] = 'serif'
```