IMAGE PREPROCESSING

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.

The first step is usually importing the libraries that will be needed in the program.
Import Keras library from that library import the ImageDataGenerator Library to your Python script:
After each code block in this tutorial, you should type ALT + ENTER or SHIFT+ENTER to run the code and move into a new code block within your notebook.

Let us import the Image Data Generator class from Keras

Import Image Data Generator Library and Configure it

Image Data Generator class is used to load the images with different modifications like considering the zoomed image, flipping the image and rescaling the images to range of 0 and 1

```
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)
```

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.
- The image rotates via the rotation_range argument
- Image brightness via the brightness_range argument.
- The image zooms via the zoom_range argument
- An instance of the ImageDataGenerator class can be constructed for train and test.
-

- **Apply Image Data Generator functionality to Train and Test set**
- Specify the path of both the folders in the flow_from_directory method. We are importing the images in 128*128 pixels.

```
x_train = train_datagen.flow_from_directory('fruit-dataset/train',
                                            target_size = (128,128),batch_size = 32, class_mode = 'categorical')
x_test =  test_datagen.flow_from_directory('fruit-dataset/test',
                                           target_size = (128,128),batch_size = 32, class_mode = 'categorical')
```

```
Found 5384 images belonging to 6 classes.
Found 1686 images belonging to 6 classes.
```

- his write-up/tutorial will take you through different ways of using **flow_from_directory** and **flow_from_dataframe**, which are methods of ImageDataGenerator class from Keras Image Preprocessing. Below we will consider different scenarios on how to generate batches of augmented/normalized data using **ImageDataGenerator methods**.

1. **directory** value : The path to parent directory containing sub-directories(class/label) with images
2. **classes** value : Name of the class/classes for which images should be loaded. *If not specified it will load all the images — (Optional argument)*
3. Other arguments to suit your need and model, which are self explanatory here

flow_from_dataframe Method

This method is useful when the *images are clustered in only one folder*. To put in other words images from different class/labels reside in only one folder. *Generally, with such kind of data, some text files containing information on class and other parameters are provided.* In this case, we will create a dataframe using pandas and text files provided, and create a meaningful dataframe with columns having file name (**only the file names, not the path**) and other classes to be used by the model. For this method, arguments to be used are:

1. **dataframe** value : Dataframe having meaningful data (*file name, class columns are a must*)
2. **directory** value : The path to the parent directory containing all images.
3. **x_col** value : which will be the name of column(in dataframe) having file names
4. **y_col** value : which will be the name of column(in dataframe) having class/label
5. Other arguments to suit your need and model, which are self-explanatory here