Image Preprocessing

Date	19 September 2022
Team ID	PNT2022TMID43802
Project Name	AI-powered nutrition analyzer for fitness enthusiasts
Maximum Marks	4 Marks

In this milestone, 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.

Loading and pre-processing the data:

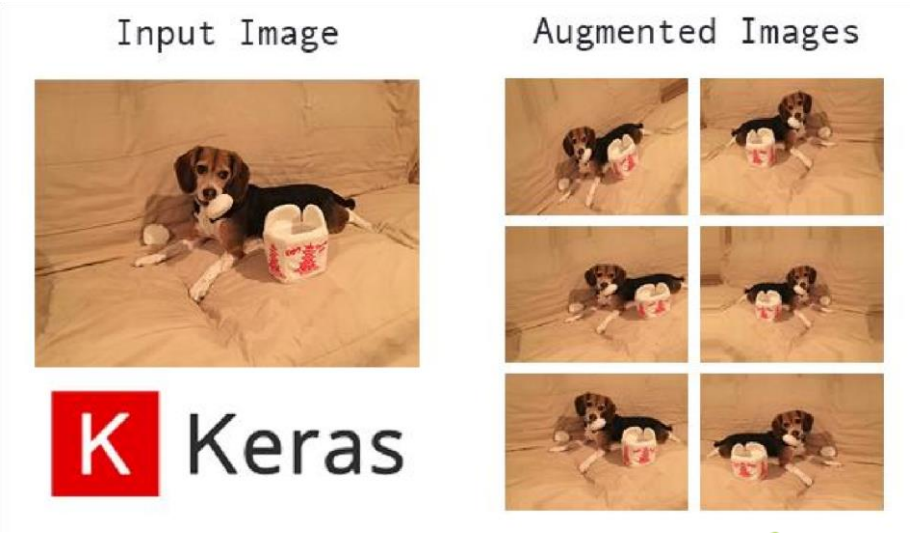
Data is gold as far as deep learning models are concerned. Your image classification model has a far better chance of performing well if you have a good amount of images in the training set. Also, the shape of the data varies according to the architecture/framework that we use.

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:

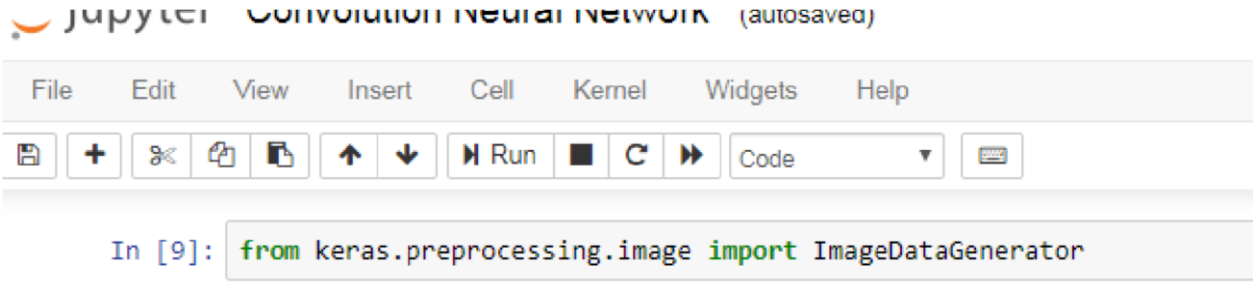
- ✓ Accepting a batch of images used for training.
- ✓ Taking this batch and applying a series of random transformations to each image in the batch (including random rotation, resizing, shearing, etc.).
- ✓ Replacing the original batch with the new, randomly transformed batch.
- ✓ Training the CNN on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

Note: The ImageDataGenerator accepts the original data, randomly transforms it, and returns only the new, transformed data.



SMARTBRIDGE
Let's Bridge the Gap

❖ **Import the library**



❖ **Define the parameters /arguments for ImageDataGenerator class**

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

Note: The `ImageDataGenerator` transforms each image in the batch by a series of random translations, these translations are based on the arguments

❖ **Applying `ImageDataGenerator` functionality to trainset and testset**

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

```
x_train = train_datagen.flow_from_directory(r'E:\dataset\training_set',target_size=(64,64),batch_size=32,class_mode='categorical')
x_test = train_datagen.flow_from_directory(r'E:\dataset\test_set',target_size=(64,64),batch_size=32,class_mode='catagorical')
```

```
Found 8000 images belonging to 2 classes.
Found 2000 images belonging to 2 classes.
```