Project Development Phase

Delivery of Sprint 1

Date - 29 October 2022

Team ID - PNT2022TMID30319

Project Name - Fertilizers Recommendation System For Disease Prediction

Maximum Marks -

Image Preprocessing

```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
    Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mou
```

```
!unzip gdrive/MyDrive/Fertilizers_Recommendation_\ System_For_Disease_\ Prediction.zip
```

```
    Archive:  gdrive/MyDrive/Fertilizers_Recommendation_ System_For_Disease_ Prediction.zip
    replace Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___Black_rot/00e909a
```

Import Library

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import pandas as pd
import tensorflow
import keras
import os
import cv2
import glob
from skimage import io
import random
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img,img_to_array
from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img,img_to_array
from numpy import expand_dims
```

```
path = 'Dataset Plant Disease'
```

```
path1 = 'Dataset Plant Disease/fruit-dataset/fruit-dataset/train'
```

```
path2 = 'Dataset Plant Disease/fruit-dataset/fruit-dataset/test'
```

```
train_data_gen = ImageDataGenerator(rescale = 1./255,
                             shear_range = 0.2,
                             zoom_range = 0.2,
                             horizontal_flip = True,
                             validation_split = 0.30)
test_data_gen = ImageDataGenerator(rescale = 1./255,validation_split = 0.30)
```

Train and Test

```
training_set = train_data_gen.flow_from_directory(path,
                                                 target_size=(64,64),
                                                 batch_size=100,
                                                 class_mode='categorical',
                                                 shuffle=True,
                                                 color_mode='rgb',
                                                 subset = 'training')

testing_set = test_data_gen.flow_from_directory(path,
                                                 target_size=(64,64),
                                                 batch_size=100,
                                                 class_mode='categorical',
                                                 shuffle=True,
                                                 color_mode='rgb',
                                                 subset = 'validation')
```

```
Found 15311 images belonging to 2 classes.
Found 6561 images belonging to 2 classes.
```

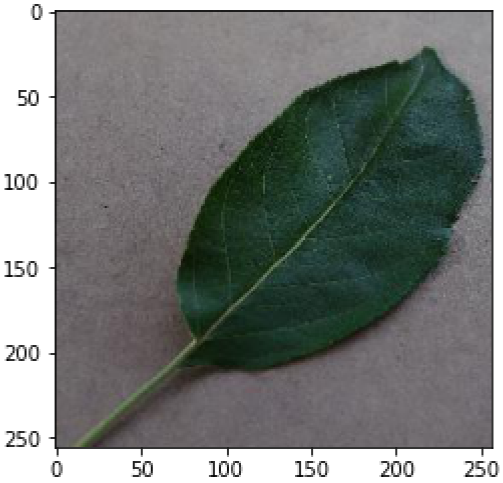
```
training_set = train_data_gen.flow_from_directory(path1,
                                                 target_size=(128,128),
                                                 batch_size=32,
                                                 class_mode='categorical',
                                                 shuffle=True,
                                                 color_mode='rgb',
                                                 subset = 'training')

testing_set = test_data_gen.flow_from_directory(path2,
                                                 target_size=(128,128),
                                                 batch_size=32,
                                                 class_mode='categorical',
                                                 shuffle=True,
                                                 color_mode='rgb',
                                                 subset = 'validation')
```

```
Found 3770 images belonging to 6 classes.
Found 503 images belonging to 6 classes.
```

Image Classification

```
image = io.imread('/content/Dataset Plant Disease/fruit-dataset/fruit-dataset/train/Apple___h
i, (im1) = plt.subplots(1)
i.set_figwidth(15)
im1.imshow(image)
```

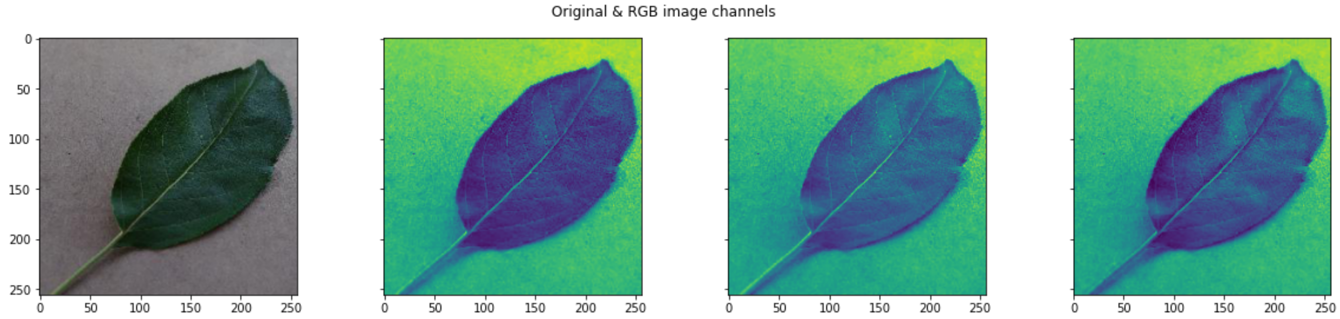
```
<matplotlib.image.AxesImage at 0x7f1b22ba9d10>
```



Original image to RGB images

```
i, (im1, im2, im3, im4) = plt.subplots(1, 4, sharey=True)
i.set_figwidth(20)

im1.imshow(image)  #Original image
im2.imshow(image[:, : , 0]) #Red
im3.imshow(image[:, : , 1]) #Green
im4.imshow(image[:, : , 2]) #Blue
i.suptitle('Original & RGB image channels')
```

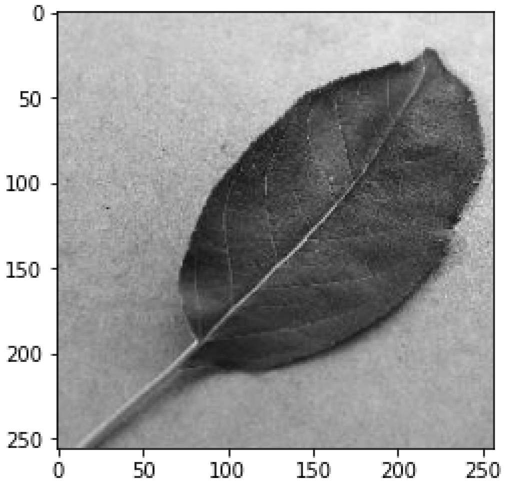
```
Text(0.5, 0.98, 'Original & RGB image channels')
```



Grayscale Conversion

```
gray_image = image.convert("L")
plt.imshow(gray_image, cmap = 'gray')
```

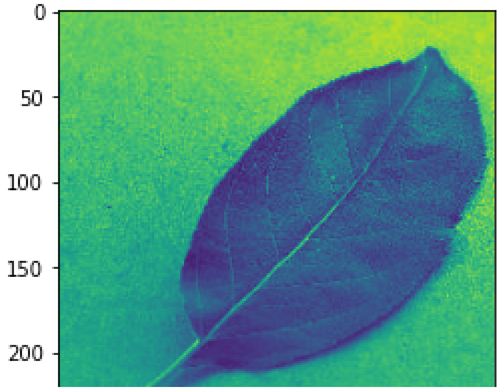
```
<matplotlib.image.AxesImage at 0x7f1b157875d0>
```



Normalization

```
norm_image = (gray_image - np.min(gray_image)) / (np.max(gray_image) - np.min(gray_image))
plt.imshow(norm_image)
```

```
<matplotlib.image.AxesImage at 0x7f1b2539ed50>
```



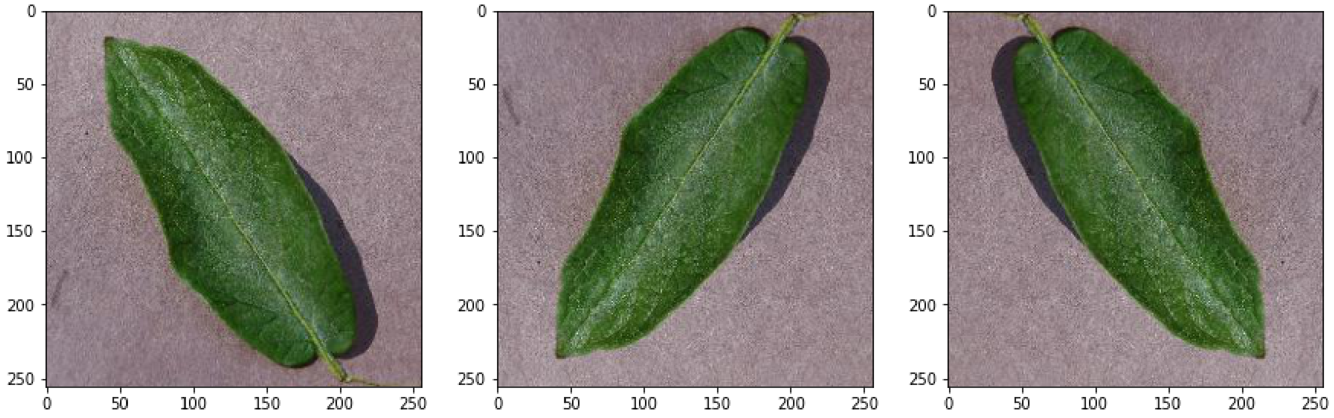
Data Augmentation

Shifting

```
# Convert to numpy array
image = load_img('/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Potato___h
data = img_to_array(image,data_format = None,dtype =None)

# Expand dimensions
samples = expand_dims(image, 0)

# Data augmentation generator
datagen = ImageDataGenerator(width_shift_range=[-100,100])

it = datagen.flow(samples, batch_size=1)
fig, im = plt.subplots(nrows=1, ncols=3, figsize=(15,15))

# Generate images
for i in range(3):

    # Convert unsigned
    image = next(it)[0].astype('uint8')

    # Plot image
    im[i].imshow(image)
```

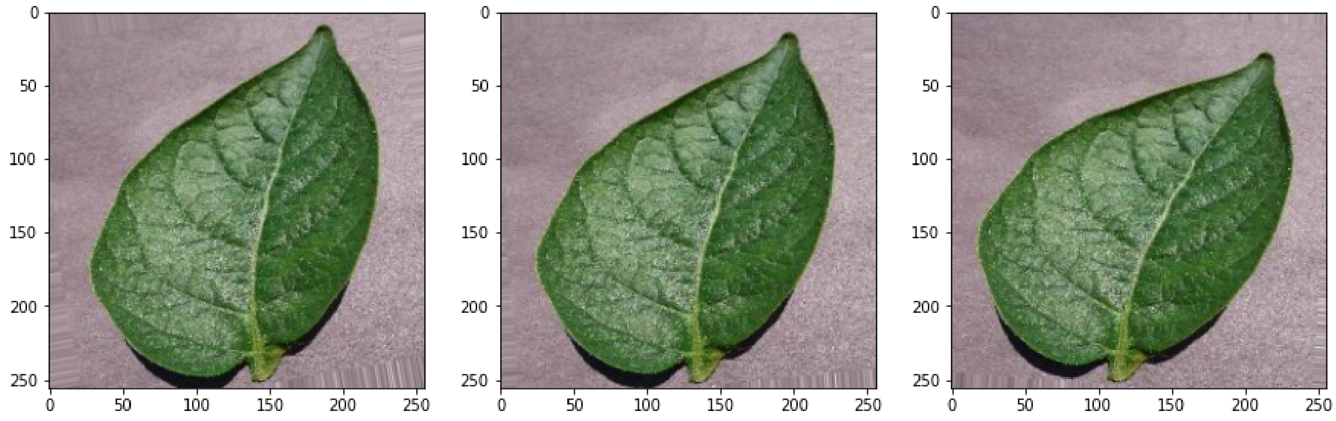
Flipping

```
# Convert to numpy array
image = load_img('/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Potato___h
data = img_to_array(image,data_format = None,dtype =None)

# Expand dimensions
samples = expand_dims(image, 0)

# Data augmentation generator
datagen = ImageDataGenerator(horizontal_flip=True, vertical_flip=True)
it = datagen.flow(samples, batch_size=1)
fig, im = plt.subplots(nrows=1, ncols=3, figsize=(15,15))

# Generate images
for i in range(3):

    # Convert unsigned
    image = next(it)[0].astype('uint8')

    # Plot image
    im[i].imshow(image)
```



Rotation

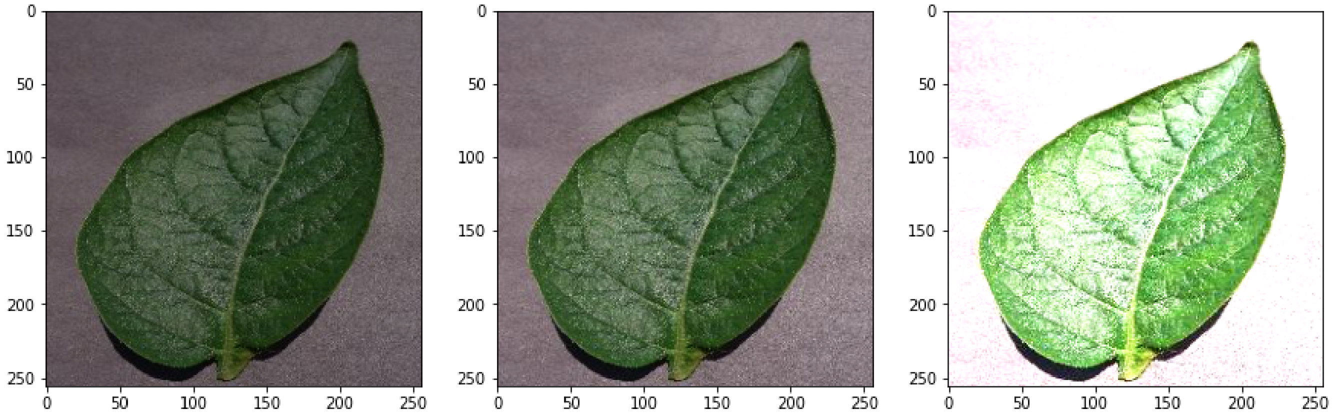
```
# Convert to numpy array
image = load_img('/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Potato___h
data = img_to_array(image,data_format = None,dtype =None)

# Expand dimensions
samples = expand_dims(image, 0)

# Data augmentation generator
datagen = ImageDataGenerator(rotation_range=20, fill_mode='nearest')
it = datagen.flow(samples, batch_size=1)
fig, im = plt.subplots(nrows=1, ncols=3, figsize=(15,15))

# Generate images
for i in range(3):

    # Convert unsigned
    image = next(it)[0].astype('uint8')

    # Plot image
    im[i].imshow(image)
```



Changing brightness

```
# Convert to numpy array
image = load_img('/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Potato___h
data = img_to_array(image,data_format = None,dtype =None)
```

```
# Expand dimensions
samples = expand_dims(image, 0)

# Data augmentation generator
datagen = ImageDataGenerator(brightness_range=[0.5,2.0])
it = datagen.flow(samples, batch_size=1)
fig, im = plt.subplots(nrows=1, ncols=3, figsize=(15,15))

# Generate images
for i in range(3):

    # Convert unsigned
    image = next(it)[0].astype('uint8')

    # Plot image
    im[i].imshow(image)
```

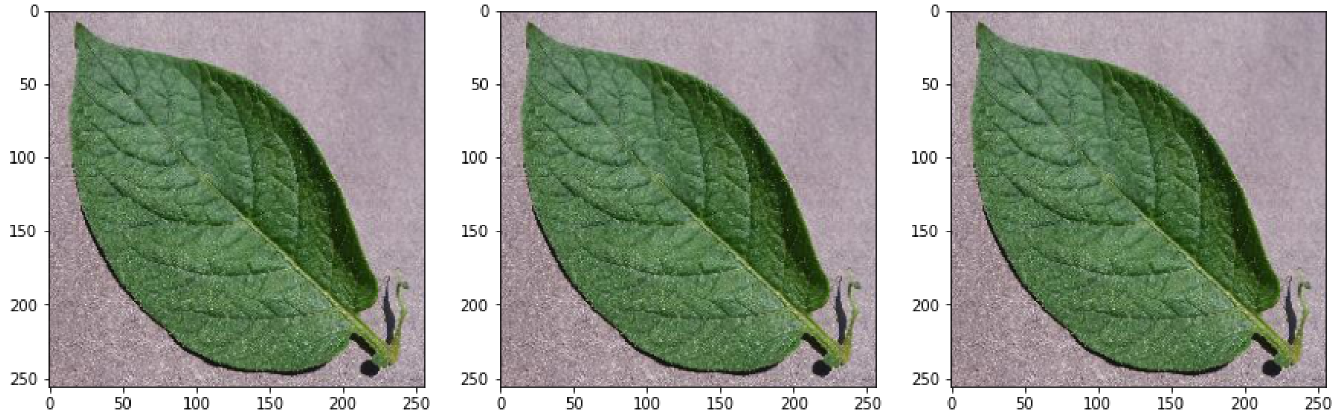


Standardizing images

```
# Convert to numpy array
image = load_img('/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Potato___h
data = img_to_array(image,data_format = None,dtype =None)

# Expand dimensions
samples = expand_dims(image, 0)

# Data augmentation generator
datagen = ImageDataGenerator(featurewise_center = False,
      featurewise_std_normalization = True)
it = datagen.flow(samples, batch_size=1)
fig, im = plt.subplots(nrows=1, ncols=3, figsize=(15,15))

# Generate images
```

```
for i in range(3):

    # Convert unsigned
    image = next(it)[0].astype('uint8')

    # Plot image
    im[i].imshow(image)
```

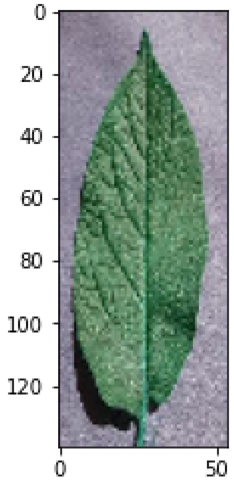
```
 /usr/local/lib/python3.7/dist-packages/keras/preprocessing/image.py:1297: UserWarning: T
   warnings.warn('This ImageDataGenerator specifies '
 /usr/local/lib/python3.7/dist-packages/keras/preprocessing/image.py:1663: UserWarning: T
   warnings.warn('This ImageDataGenerator specifies '
 /usr/local/lib/python3.7/dist-packages/keras/preprocessing/image.py:1671: UserWarning: T
   warnings.warn('This ImageDataGenerator specifies '
```



Cropping

```
img = cv2.imread('/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Potato___h
res = cv2.resize(img, dsize=(54, 140), interpolation=cv2.INTER_CUBIC)
plt.imshow(res)
```

```
<matplotlib.image.AxesImage at 0x7f1b25254c90>
```


0
20
40
60
80
100
120
0
50