PREDICT function:

'predict' function takes in path where images of plants' leaves are located and for each one of them individually predicts whether they are healthy or not.

In [20]:

```
def final_fun_1(path):
  import tensorflow as tf
  from tqdm import tqdm
  import numpy as np
  import keras

  #add '*' after path name
  if path[-1]=='/':
    path=path+"*"
  else:
    path= path+"/*"
  images_ds= tf.data.Dataset.list_files(path) #get path to images

  #function to get image
  def process_image(file_path):
    img= tf.io.read_file(file_path)
    img= tf.image.decode_jpeg(img)
    img= tf.image.resize(img, [254,254])
    img= img/255
    return img

  x=[]
  for image in tqdm(images_ds.map(process_image)):
    x.append(image)
  x= np.array(x)

  #get file names:
  image_name=[]
  for images in images_ds:
    image_name.append(images)

  file_name_arr= np.array(image_name)
  file_name= []
  for i  in file_name_arr:
    i= ((str(i).split(',')[0][12:])[:-1])
    file_name.append(i)
  del file_name_arr

  file_names= []
  for file in file_name:
    file_names.append(file.split('/')[-1])

  #load model:
  path_to_model= "/content/gdrive/MyDrive/Plant_disease/aug_20_2.hdf5"
  model = keras.models.load_model(path_to_model)

  y_pred=[]

  for pred in ((model.predict(x))): #custom loop with threshold as 0.1
    if pred >= 0.5:
      y_pred.append('without_disease')
    else:
      y_pred.append('with_disease')

  #join file_name and it's prediction:
  final_pred=[]
```

```
  for file, pred in zip(file_names, y_pred):
    final_pred.append(str(file)+ ': ' + str(pred))

  return final_pred
```

In [21]:
```
final_fun_1("/content/gdrive/MyDrive/Plant_disease/self_collected/predict/")
```

```
100%|██████████| 23/23 [00:00<00:00, 38.91it/s]
```

Out[21]:
```
['4w.jpg: without_disease',
 '17w.jpg: with_disease',
 '20w.jpg: without_disease',
 '12I.jpg: without_disease',
 '2w.jpg: with_disease',
 '10I.jpg: with_disease',
 '8I.jpg: without_disease',
 '19w.jpg: with_disease',
 '18w.jpg: without_disease',
 '7I.jpg: with_disease',
 '16I.jpg: with_disease',
 '22w.jpg: without_disease',
 '9I.jpg: without_disease',
 '13I.jpg: without_disease',
 '14I.jpg: with_disease',
 '11I.jpg: with_disease',
 '3w.jpg: with_disease',
 '5U.jpg: without_disease',
 '15I.jpg: without_disease',
 '21w.jpg: without_disease',
 '1w.jpg: without_disease',
 '6I.jpg: with_disease',
 '0w.jpg: with_disease']
```

ACCURACY function:

'accuracy' function takes in path where plants' leaves images are located in 'healthy_plants' and 'unhealthy_plants' folders, loads pretrained model, inputs the image in model and gives output in form of accuracy.

In [17]:
```
def final_fun_2(path):
  import tensorflow as tf
  from tqdm import tqdm
  import numpy as np
  from sklearn.preprocessing import LabelBinarizer
  import keras

  if path[-1]=='/':
    path=path+"*/*"
  elif path[-1]!= '/':
    path= path+"/*/*"
  else:
    path= path

  images_ds= tf.data.Dataset.list_files(path) #get path to images

  # function to get label
  def get_label(file_path):
    import os
    return tf.strings.split(file_path, os.path.sep)[-2]

  #function to get image and label
  def process_image(file_path):
```

```
    label= get_label(file_path)

    img= tf.io.read_file(file_path)
    img= tf.image.decode_jpeg(img)
    img= tf.image.resize(img, [254,254])
    img= img/255
    return img, label

  x=[]
  y=[]

  for image, label in tqdm(images_ds.map(process_image)):
    x.append(image)
    y.append(label)

  x= np.array(x)
  y_arr= np.array(y)

  y= []
  for i  in y_arr:
    i= ((str(i).split(',')[0][12:])[:-1])
    y.append(i)
  del y_arr

  y= np.asarray(y)

  #converting strings of y into label (1s and 0s):
  label_binarizer = LabelBinarizer()
  y_label = label_binarizer.fit_transform(y)
  n_classes = len(label_binarizer.classes_)

  #load model:
  path_to_model= "/content/gdrive/MyDrive/Plant_disease/aug_20_2.hdf5"
  model = keras.models.load_model(path_to_model)

  scores = model.evaluate(x, y_label)
  print(f"ACCURACY for test dataset is: {scores[1]}")
```

In [14]:

```
final_fun_2("/content/gdrive/MyDrive/Plant_disease/self_collected/Test")
```

```
100%|██████████| 425/425 [00:05<00:00, 82.37it/s]
14/14 [==============================] - 18s 1s/step - loss: 0.3670 - accurac
y: 0.8424
ACCURACY for test dataset is: 0.8423529267311096
```