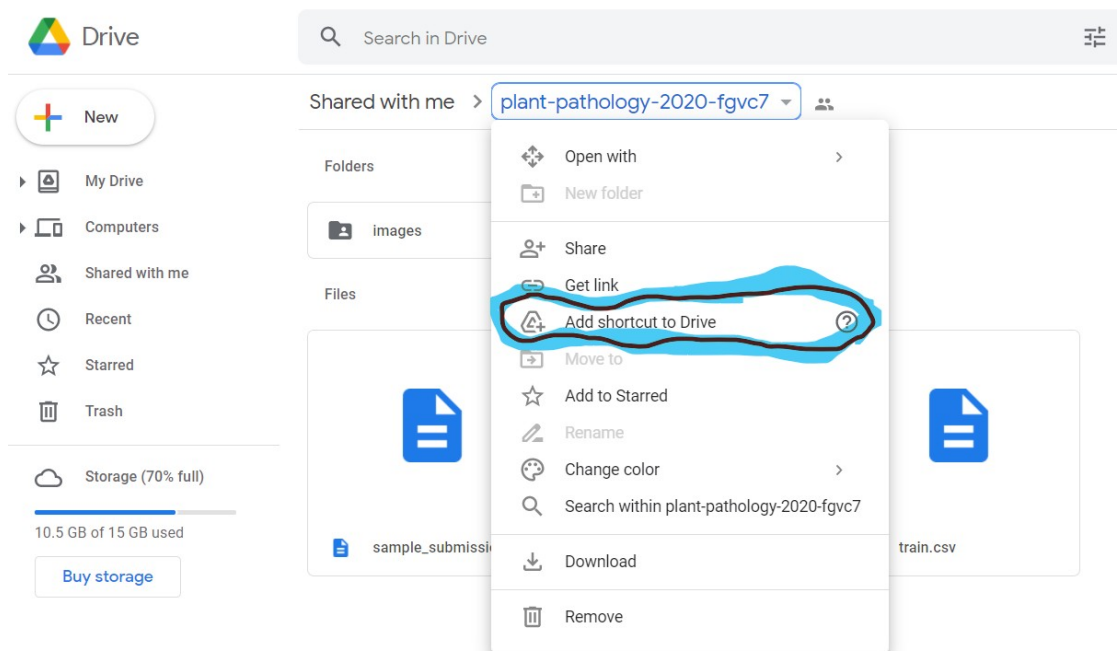


```
import numpy as np
import pandas as pd
import os
from re import search
import shutil
import natsort
from PIL import Image
import matplotlib.pyplot as plt
from tqdm import tqdm
import cv2
from google.colab import drive
```

Add this data set to your drive before proceeding. [Link to Data Set](#)



```
drive.mount('/content/drive')
```

Mounted at /content/drive

```
DIR=r'/content/drive/My Drive/plant-pathology-2020-fgvc7/images'
```

```
train=pd.read_csv(r"/content/drive/My Drive/plant-pathology-2020-fgvc7/train.csv")
```

```
test=pd.read_csv(r"/content/drive/My Drive/plant-pathology-2020-fgvc7/test.csv")
```

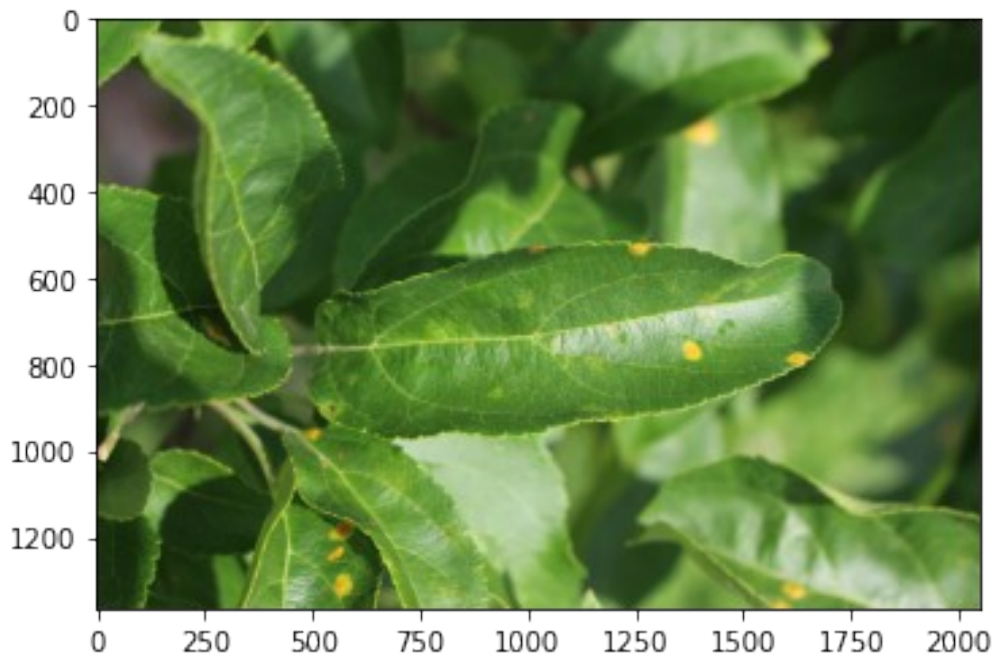
```
print(train.head())
```

	image_id	healthy	multiple_diseases	rust	scab
0	Train_0	0	0	0	1
1	Train_1	0	1	0	0
2	Train_2	1	0	0	0
3	Train_3	0	0	1	0
4	Train_4	1	0	0	0

```
print(test.head())
```

	image_id
0	Test_0
1	Test_1
2	Test_2
3	Test_3
4	Test_4

```
image1=Image.open(r'/content/drive/My Drive/plant-pathology-2020-  
fgvc7/images/Test_0.jpg')  
plt.imshow(image1)  
plt.show()
```



PREPARE TRAINING DATA

```
class_names=train.loc[:, 'healthy:'].columns  
print(class_names)
```

```
Index(['healthy', 'multiple_diseases', 'rust', 'scab'],  
      dtype='object')
```

```

number=0
train['label']=0
for i in class_names:
    train['label']=train['label'] + train[i] * number
    number=number+1

```

```
train.head()
```

	image_id	healthy	multiple_diseases	rust	scab	label
0	Train_0	0	0	0	1	3
1	Train_1	0	1	0	0	1
2	Train_2	1	0	0	0	0
3	Train_3	0	0	1	0	2
4	Train_4	1	0	0	0	0

```
DIR
```

```
{"type": "string"}
```

```
natsort.natsorted(os.listdir(DIR))
```

```

['Test_0.jpg',
 'Test_1.jpg',
 'Test_2.jpg',
 'Test_3.jpg',
 'Test_4.jpg',
 'Test_5.jpg',
 'Test_6.jpg',
 'Test_7.jpg',
 'Test_8.jpg',
 'Test_9.jpg',
 'Test_10.jpg',
 'Test_11.jpg',
 'Test_12.jpg',
 'Test_13.jpg',
 'Test_14.jpg',
 'Test_15.jpg',
 'Test_16.jpg',
 'Test_17.jpg',
 'Test_18.jpg',
 'Test_19.jpg',
 'Test_20.jpg',
 'Test_21.jpg',
 'Test_22.jpg',
 'Test_23.jpg',
 'Test_24.jpg',
 'Test_25.jpg',
 'Test_26.jpg',
 'Test_27.jpg',
 'Test_28.jpg',
 'Test_29.jpg',

```

'Test_30.jpg',
'Test_31.jpg',
'Test_32.jpg',
'Test_33.jpg',
'Test_34.jpg',
'Test_35.jpg',
'Test_36.jpg',
'Test_37.jpg',
'Test_38.jpg',
'Test_39.jpg',
'Test_40.jpg',
'Test_41.jpg',
'Test_42.jpg',
'Test_43.jpg',
'Test_44.jpg',
'Test_45.jpg',
'Test_46.jpg',
'Test_47.jpg',
'Test_48.jpg',
'Test_49.jpg',
'Test_50.jpg',
'Test_51.jpg',
'Test_52.jpg',
'Test_53.jpg',
'Test_54.jpg',
'Test_55.jpg',
'Test_56.jpg',
'Test_57.jpg',
'Test_58.jpg',
'Test_59.jpg',
'Test_60.jpg',
'Test_61.jpg',
'Test_62.jpg',
'Test_63.jpg',
'Test_64.jpg',
'Test_65.jpg',
'Test_66.jpg',
'Test_67.jpg',
'Test_68.jpg',
'Test_69.jpg',
'Test_70.jpg',
'Test_71.jpg',
'Test_72.jpg',
'Test_73.jpg',
'Test_74.jpg',
'Test_75.jpg',
'Test_76.jpg',
'Test_77.jpg',
'Test_78.jpg',
'Test_79.jpg',

'Test_80.jpg',
'Test_81.jpg',
'Test_82.jpg',
'Test_83.jpg',
'Test_84.jpg',
'Test_85.jpg',
'Test_86.jpg',
'Test_87.jpg',
'Test_88.jpg',
'Test_89.jpg',
'Test_90.jpg',
'Test_91.jpg',
'Test_92.jpg',
'Test_93.jpg',
'Test_94.jpg',
'Test_95.jpg',
'Test_96.jpg',
'Test_97.jpg',
'Test_98.jpg',
'Test_99.jpg',
'Test_100.jpg',
'Test_101.jpg',
'Test_102.jpg',
'Test_103.jpg',
'Test_104.jpg',
'Test_105.jpg',
'Test_106.jpg',
'Test_107.jpg',
'Test_108.jpg',
'Test_109.jpg',
'Test_110.jpg',
'Test_111.jpg',
'Test_112.jpg',
'Test_113.jpg',
'Test_114.jpg',
'Test_115.jpg',
'Test_116.jpg',
'Test_117.jpg',
'Test_118.jpg',
'Test_119.jpg',
'Test_120.jpg',
'Test_121.jpg',
'Test_122.jpg',
'Test_123.jpg',
'Test_124.jpg',
'Test_125.jpg',
'Test_126.jpg',
'Test_127.jpg',
'Test_128.jpg',
'Test_129.jpg',

'Test_130.jpg',
'Test_131.jpg',
'Test_132.jpg',
'Test_133.jpg',
'Test_134.jpg',
'Test_135.jpg',
'Test_136.jpg',
'Test_137.jpg',
'Test_138.jpg',
'Test_139.jpg',
'Test_140.jpg',
'Test_141.jpg',
'Test_142.jpg',
'Test_143.jpg',
'Test_144.jpg',
'Test_145.jpg',
'Test_146.jpg',
'Test_147.jpg',
'Test_148.jpg',
'Test_149.jpg',
'Test_150.jpg',
'Test_151.jpg',
'Test_152.jpg',
'Test_153.jpg',
'Test_154.jpg',
'Test_155.jpg',
'Test_156.jpg',
'Test_157.jpg',
'Test_158.jpg',
'Test_159.jpg',
'Test_160.jpg',
'Test_161.jpg',
'Test_162.jpg',
'Test_163.jpg',
'Test_164.jpg',
'Test_165.jpg',
'Test_166.jpg',
'Test_167.jpg',
'Test_168.jpg',
'Test_169.jpg',
'Test_170.jpg',
'Test_171.jpg',
'Test_172.jpg',
'Test_173.jpg',
'Test_174.jpg',
'Test_175.jpg',
'Test_176.jpg',
'Test_177.jpg',
'Test_178.jpg',
'Test_179.jpg',

'Test_180.jpg',
'Test_181.jpg',
'Test_182.jpg',
'Test_183.jpg',
'Test_184.jpg',
'Test_185.jpg',
'Test_186.jpg',
'Test_187.jpg',
'Test_188.jpg',
'Test_189.jpg',
'Test_190.jpg',
'Test_191.jpg',
'Test_192.jpg',
'Test_193.jpg',
'Test_194.jpg',
'Test_195.jpg',
'Test_196.jpg',
'Test_197.jpg',
'Test_198.jpg',
'Test_199.jpg',
'Test_200.jpg',
'Test_201.jpg',
'Test_202.jpg',
'Test_203.jpg',
'Test_204.jpg',
'Test_205.jpg',
'Test_206.jpg',
'Test_207.jpg',
'Test_208.jpg',
'Test_209.jpg',
'Test_210.jpg',
'Test_211.jpg',
'Test_212.jpg',
'Test_213.jpg',
'Test_214.jpg',
'Test_215.jpg',
'Test_216.jpg',
'Test_217.jpg',
'Test_218.jpg',
'Test_219.jpg',
'Test_220.jpg',
'Test_221.jpg',
'Test_222.jpg',
'Test_223.jpg',
'Test_224.jpg',
'Test_225.jpg',
'Test_226.jpg',
'Test_227.jpg',
'Test_228.jpg',
'Test_229.jpg',

'Test_230.jpg',
'Test_231.jpg',
'Test_232.jpg',
'Test_233.jpg',
'Test_234.jpg',
'Test_235.jpg',
'Test_236.jpg',
'Test_237.jpg',
'Test_238.jpg',
'Test_239.jpg',
'Test_240.jpg',
'Test_241.jpg',
'Test_242.jpg',
'Test_243.jpg',
'Test_244.jpg',
'Test_245.jpg',
'Test_246.jpg',
'Test_247.jpg',
'Test_248.jpg',
'Test_249.jpg',
'Test_250.jpg',
'Test_251.jpg',
'Test_252.jpg',
'Test_253.jpg',
'Test_254.jpg',
'Test_255.jpg',
'Test_256.jpg',
'Test_257.jpg',
'Test_258.jpg',
'Test_259.jpg',
'Test_260.jpg',
'Test_261.jpg',
'Test_262.jpg',
'Test_263.jpg',
'Test_264.jpg',
'Test_265.jpg',
'Test_266.jpg',
'Test_267.jpg',
'Test_268.jpg',
'Test_269.jpg',
'Test_270.jpg',
'Test_271.jpg',
'Test_272.jpg',
'Test_273.jpg',
'Test_274.jpg',
'Test_275.jpg',
'Test_276.jpg',
'Test_277.jpg',
'Test_278.jpg',
'Test_279.jpg',

'Test_280.jpg',
'Test_281.jpg',
'Test_282.jpg',
'Test_283.jpg',
'Test_284.jpg',
'Test_285.jpg',
'Test_286.jpg',
'Test_287.jpg',
'Test_288.jpg',
'Test_289.jpg',
'Test_290.jpg',
'Test_291.jpg',
'Test_292.jpg',
'Test_293.jpg',
'Test_294.jpg',
'Test_295.jpg',
'Test_296.jpg',
'Test_297.jpg',
'Test_298.jpg',
'Test_299.jpg',
'Test_300.jpg',
'Test_301.jpg',
'Test_302.jpg',
'Test_303.jpg',
'Test_304.jpg',
'Test_305.jpg',
'Test_306.jpg',
'Test_307.jpg',
'Test_308.jpg',
'Test_309.jpg',
'Test_310.jpg',
'Test_311.jpg',
'Test_312.jpg',
'Test_313.jpg',
'Test_314.jpg',
'Test_315.jpg',
'Test_316.jpg',
'Test_317.jpg',
'Test_318.jpg',
'Test_319.jpg',
'Test_320.jpg',
'Test_321.jpg',
'Test_322.jpg',
'Test_323.jpg',
'Test_324.jpg',
'Test_325.jpg',
'Test_326.jpg',
'Test_327.jpg',
'Test_328.jpg',
'Test_329.jpg',

'Test_330.jpg',
'Test_331.jpg',
'Test_332.jpg',
'Test_333.jpg',
'Test_334.jpg',
'Test_335.jpg',
'Test_336.jpg',
'Test_337.jpg',
'Test_338.jpg',
'Test_339.jpg',
'Test_340.jpg',
'Test_341.jpg',
'Test_342.jpg',
'Test_343.jpg',
'Test_344.jpg',
'Test_345.jpg',
'Test_346.jpg',
'Test_347.jpg',
'Test_348.jpg',
'Test_349.jpg',
'Test_350.jpg',
'Test_351.jpg',
'Test_352.jpg',
'Test_353.jpg',
'Test_354.jpg',
'Test_355.jpg',
'Test_356.jpg',
'Test_357.jpg',
'Test_358.jpg',
'Test_359.jpg',
'Test_360.jpg',
'Test_361.jpg',
'Test_362.jpg',
'Test_363.jpg',
'Test_364.jpg',
'Test_365.jpg',
'Test_366.jpg',
'Test_367.jpg',
'Test_368.jpg',
'Test_369.jpg',
'Test_370.jpg',
'Test_371.jpg',
'Test_372.jpg',
'Test_373.jpg',
'Test_374.jpg',
'Test_375.jpg',
'Test_376.jpg',
'Test_377.jpg',
'Test_378.jpg',
'Test_379.jpg',

'Test_380.jpg',
'Test_381.jpg',
'Test_382.jpg',
'Test_383.jpg',
'Test_384.jpg',
'Test_385.jpg',
'Test_386.jpg',
'Test_387.jpg',
'Test_388.jpg',
'Test_389.jpg',
'Test_390.jpg',
'Test_391.jpg',
'Test_392.jpg',
'Test_393.jpg',
'Test_394.jpg',
'Test_395.jpg',
'Test_396.jpg',
'Test_397.jpg',
'Test_398.jpg',
'Test_399.jpg',
'Test_400.jpg',
'Test_401.jpg',
'Test_402.jpg',
'Test_403.jpg',
'Test_404.jpg',
'Test_405.jpg',
'Test_406.jpg',
'Test_407.jpg',
'Test_408.jpg',
'Test_409.jpg',
'Test_410.jpg',
'Test_411.jpg',
'Test_412.jpg',
'Test_413.jpg',
'Test_414.jpg',
'Test_415.jpg',
'Test_416.jpg',
'Test_417.jpg',
'Test_418.jpg',
'Test_419.jpg',
'Test_420.jpg',
'Test_421.jpg',
'Test_422.jpg',
'Test_423.jpg',
'Test_424.jpg',
'Test_425.jpg',
'Test_426.jpg',
'Test_427.jpg',
'Test_428.jpg',
'Test_429.jpg',

'Test_430.jpg',
'Test_431.jpg',
'Test_432.jpg',
'Test_433.jpg',
'Test_434.jpg',
'Test_435.jpg',
'Test_436.jpg',
'Test_437.jpg',
'Test_438.jpg',
'Test_439.jpg',
'Test_440.jpg',
'Test_441.jpg',
'Test_442.jpg',
'Test_443.jpg',
'Test_444.jpg',
'Test_445.jpg',
'Test_446.jpg',
'Test_447.jpg',
'Test_448.jpg',
'Test_449.jpg',
'Test_450.jpg',
'Test_451.jpg',
'Test_452.jpg',
'Test_453.jpg',
'Test_454.jpg',
'Test_455.jpg',
'Test_456.jpg',
'Test_457.jpg',
'Test_458.jpg',
'Test_459.jpg',
'Test_460.jpg',
'Test_461.jpg',
'Test_462.jpg',
'Test_463.jpg',
'Test_464.jpg',
'Test_465.jpg',
'Test_466.jpg',
'Test_467.jpg',
'Test_468.jpg',
'Test_469.jpg',
'Test_470.jpg',
'Test_471.jpg',
'Test_472.jpg',
'Test_473.jpg',
'Test_474.jpg',
'Test_475.jpg',
'Test_476.jpg',
'Test_477.jpg',
'Test_478.jpg',
'Test_479.jpg',

'Test_480.jpg',
'Test_481.jpg',
'Test_482.jpg',
'Test_483.jpg',
'Test_484.jpg',
'Test_485.jpg',
'Test_486.jpg',
'Test_487.jpg',
'Test_488.jpg',
'Test_489.jpg',
'Test_490.jpg',
'Test_491.jpg',
'Test_492.jpg',
'Test_493.jpg',
'Test_494.jpg',
'Test_495.jpg',
'Test_496.jpg',
'Test_497.jpg',
'Test_498.jpg',
'Test_499.jpg',
'Test_500.jpg',
'Test_501.jpg',
'Test_502.jpg',
'Test_503.jpg',
'Test_504.jpg',
'Test_505.jpg',
'Test_506.jpg',
'Test_507.jpg',
'Test_508.jpg',
'Test_509.jpg',
'Test_510.jpg',
'Test_511.jpg',
'Test_512.jpg',
'Test_513.jpg',
'Test_514.jpg',
'Test_515.jpg',
'Test_516.jpg',
'Test_517.jpg',
'Test_518.jpg',
'Test_519.jpg',
'Test_520.jpg',
'Test_521.jpg',
'Test_522.jpg',
'Test_523.jpg',
'Test_524.jpg',
'Test_525.jpg',
'Test_526.jpg',
'Test_527.jpg',
'Test_528.jpg',
'Test_529.jpg',

'Test_530.jpg',
'Test_531.jpg',
'Test_532.jpg',
'Test_533.jpg',
'Test_534.jpg',
'Test_535.jpg',
'Test_536.jpg',
'Test_537.jpg',
'Test_538.jpg',
'Test_539.jpg',
'Test_540.jpg',
'Test_541.jpg',
'Test_542.jpg',
'Test_543.jpg',
'Test_544.jpg',
'Test_545.jpg',
'Test_546.jpg',
'Test_547.jpg',
'Test_548.jpg',
'Test_549.jpg',
'Test_550.jpg',
'Test_551.jpg',
'Test_552.jpg',
'Test_553.jpg',
'Test_554.jpg',
'Test_555.jpg',
'Test_556.jpg',
'Test_557.jpg',
'Test_558.jpg',
'Test_559.jpg',
'Test_560.jpg',
'Test_561.jpg',
'Test_562.jpg',
'Test_563.jpg',
'Test_564.jpg',
'Test_565.jpg',
'Test_566.jpg',
'Test_567.jpg',
'Test_568.jpg',
'Test_569.jpg',
'Test_570.jpg',
'Test_571.jpg',
'Test_572.jpg',
'Test_573.jpg',
'Test_574.jpg',
'Test_575.jpg',
'Test_576.jpg',
'Test_577.jpg',
'Test_578.jpg',
'Test_579.jpg',

'Test_580.jpg',
'Test_581.jpg',
'Test_582.jpg',
'Test_583.jpg',
'Test_584.jpg',
'Test_585.jpg',
'Test_586.jpg',
'Test_587.jpg',
'Test_588.jpg',
'Test_589.jpg',
'Test_590.jpg',
'Test_591.jpg',
'Test_592.jpg',
'Test_593.jpg',
'Test_594.jpg',
'Test_595.jpg',
'Test_596.jpg',
'Test_597.jpg',
'Test_598.jpg',
'Test_599.jpg',
'Test_600.jpg',
'Test_601.jpg',
'Test_602.jpg',
'Test_603.jpg',
'Test_604.jpg',
'Test_605.jpg',
'Test_606.jpg',
'Test_607.jpg',
'Test_608.jpg',
'Test_609.jpg',
'Test_610.jpg',
'Test_611.jpg',
'Test_612.jpg',
'Test_613.jpg',
'Test_614.jpg',
'Test_615.jpg',
'Test_616.jpg',
'Test_617.jpg',
'Test_618.jpg',
'Test_619.jpg',
'Test_620.jpg',
'Test_621.jpg',
'Test_622.jpg',
'Test_623.jpg',
'Test_624.jpg',
'Test_625.jpg',
'Test_626.jpg',
'Test_627.jpg',
'Test_628.jpg',
'Test_629.jpg',

'Test_630.jpg',
'Test_631.jpg',
'Test_632.jpg',
'Test_633.jpg',
'Test_634.jpg',
'Test_635.jpg',
'Test_636.jpg',
'Test_637.jpg',
'Test_638.jpg',
'Test_639.jpg',
'Test_640.jpg',
'Test_641.jpg',
'Test_642.jpg',
'Test_643.jpg',
'Test_644.jpg',
'Test_645.jpg',
'Test_646.jpg',
'Test_647.jpg',
'Test_648.jpg',
'Test_649.jpg',
'Test_650.jpg',
'Test_651.jpg',
'Test_652.jpg',
'Test_653.jpg',
'Test_654.jpg',
'Test_655.jpg',
'Test_656.jpg',
'Test_657.jpg',
'Test_658.jpg',
'Test_659.jpg',
'Test_660.jpg',
'Test_661.jpg',
'Test_662.jpg',
'Test_663.jpg',
'Test_664.jpg',
'Test_665.jpg',
'Test_666.jpg',
'Test_667.jpg',
'Test_668.jpg',
'Test_669.jpg',
'Test_670.jpg',
'Test_671.jpg',
'Test_672.jpg',
'Test_673.jpg',
'Test_674.jpg',
'Test_675.jpg',
'Test_676.jpg',
'Test_677.jpg',
'Test_678.jpg',
'Test_679.jpg',

'Test_680.jpg',
'Test_681.jpg',
'Test_682.jpg',
'Test_683.jpg',
'Test_684.jpg',
'Test_685.jpg',
'Test_686.jpg',
'Test_687.jpg',
'Test_688.jpg',
'Test_689.jpg',
'Test_690.jpg',
'Test_691.jpg',
'Test_692.jpg',
'Test_693.jpg',
'Test_694.jpg',
'Test_695.jpg',
'Test_696.jpg',
'Test_697.jpg',
'Test_698.jpg',
'Test_699.jpg',
'Test_700.jpg',
'Test_701.jpg',
'Test_702.jpg',
'Test_703.jpg',
'Test_704.jpg',
'Test_705.jpg',
'Test_706.jpg',
'Test_707.jpg',
'Test_708.jpg',
'Test_709.jpg',
'Test_710.jpg',
'Test_711.jpg',
'Test_712.jpg',
'Test_713.jpg',
'Test_714.jpg',
'Test_715.jpg',
'Test_716.jpg',
'Test_717.jpg',
'Test_718.jpg',
'Test_719.jpg',
'Test_720.jpg',
'Test_721.jpg',
'Test_722.jpg',
'Test_723.jpg',
'Test_724.jpg',
'Test_725.jpg',
'Test_726.jpg',
'Test_727.jpg',
'Test_728.jpg',
'Test_729.jpg',

'Test_730.jpg',
'Test_731.jpg',
'Test_732.jpg',
'Test_733.jpg',
'Test_734.jpg',
'Test_735.jpg',
'Test_736.jpg',
'Test_737.jpg',
'Test_738.jpg',
'Test_739.jpg',
'Test_740.jpg',
'Test_741.jpg',
'Test_742.jpg',
'Test_743.jpg',
'Test_744.jpg',
'Test_745.jpg',
'Test_746.jpg',
'Test_747.jpg',
'Test_748.jpg',
'Test_749.jpg',
'Test_750.jpg',
'Test_751.jpg',
'Test_752.jpg',
'Test_753.jpg',
'Test_754.jpg',
'Test_755.jpg',
'Test_756.jpg',
'Test_757.jpg',
'Test_758.jpg',
'Test_759.jpg',
'Test_760.jpg',
'Test_761.jpg',
'Test_762.jpg',
'Test_763.jpg',
'Test_764.jpg',
'Test_765.jpg',
'Test_766.jpg',
'Test_767.jpg',
'Test_768.jpg',
'Test_769.jpg',
'Test_770.jpg',
'Test_771.jpg',
'Test_772.jpg',
'Test_773.jpg',
'Test_774.jpg',
'Test_775.jpg',
'Test_776.jpg',
'Test_777.jpg',
'Test_778.jpg',
'Test_779.jpg',

'Test_780.jpg',
'Test_781.jpg',
'Test_782.jpg',
'Test_783.jpg',
'Test_784.jpg',
'Test_785.jpg',
'Test_786.jpg',
'Test_787.jpg',
'Test_788.jpg',
'Test_789.jpg',
'Test_790.jpg',
'Test_791.jpg',
'Test_792.jpg',
'Test_793.jpg',
'Test_794.jpg',
'Test_795.jpg',
'Test_796.jpg',
'Test_797.jpg',
'Test_798.jpg',
'Test_799.jpg',
'Test_800.jpg',
'Test_801.jpg',
'Test_802.jpg',
'Test_803.jpg',
'Test_804.jpg',
'Test_805.jpg',
'Test_806.jpg',
'Test_807.jpg',
'Test_808.jpg',
'Test_809.jpg',
'Test_810.jpg',
'Test_811.jpg',
'Test_812.jpg',
'Test_813.jpg',
'Test_814.jpg',
'Test_815.jpg',
'Test_816.jpg',
'Test_817.jpg',
'Test_818.jpg',
'Test_819.jpg',
'Test_820.jpg',
'Test_821.jpg',
'Test_822.jpg',
'Test_823.jpg',
'Test_824.jpg',
'Test_825.jpg',
'Test_826.jpg',
'Test_827.jpg',
'Test_828.jpg',
'Test_829.jpg',

'Test_830.jpg',
'Test_831.jpg',
'Test_832.jpg',
'Test_833.jpg',
'Test_834.jpg',
'Test_835.jpg',
'Test_836.jpg',
'Test_837.jpg',
'Test_838.jpg',
'Test_839.jpg',
'Test_840.jpg',
'Test_841.jpg',
'Test_842.jpg',
'Test_843.jpg',
'Test_844.jpg',
'Test_845.jpg',
'Test_846.jpg',
'Test_847.jpg',
'Test_848.jpg',
'Test_849.jpg',
'Test_850.jpg',
'Test_851.jpg',
'Test_852.jpg',
'Test_853.jpg',
'Test_854.jpg',
'Test_855.jpg',
'Test_856.jpg',
'Test_857.jpg',
'Test_858.jpg',
'Test_859.jpg',
'Test_860.jpg',
'Test_861.jpg',
'Test_862.jpg',
'Test_863.jpg',
'Test_864.jpg',
'Test_865.jpg',
'Test_866.jpg',
'Test_867.jpg',
'Test_868.jpg',
'Test_869.jpg',
'Test_870.jpg',
'Test_871.jpg',
'Test_872.jpg',
'Test_873.jpg',
'Test_874.jpg',
'Test_875.jpg',
'Test_876.jpg',
'Test_877.jpg',
'Test_878.jpg',
'Test_879.jpg',

'Test_880.jpg',
'Test_881.jpg',
'Test_882.jpg',
'Test_883.jpg',
'Test_884.jpg',
'Test_885.jpg',
'Test_886.jpg',
'Test_887.jpg',
'Test_888.jpg',
'Test_889.jpg',
'Test_890.jpg',
'Test_891.jpg',
'Test_892.jpg',
'Test_893.jpg',
'Test_894.jpg',
'Test_895.jpg',
'Test_896.jpg',
'Test_897.jpg',
'Test_898.jpg',
'Test_899.jpg',
'Test_900.jpg',
'Test_901.jpg',
'Test_902.jpg',
'Test_903.jpg',
'Test_904.jpg',
'Test_905.jpg',
'Test_906.jpg',
'Test_907.jpg',
'Test_908.jpg',
'Test_909.jpg',
'Test_910.jpg',
'Test_911.jpg',
'Test_912.jpg',
'Test_913.jpg',
'Test_914.jpg',
'Test_915.jpg',
'Test_916.jpg',
'Test_917.jpg',
'Test_918.jpg',
'Test_919.jpg',
'Test_920.jpg',
'Test_921.jpg',
'Test_922.jpg',
'Test_923.jpg',
'Test_924.jpg',
'Test_925.jpg',
'Test_926.jpg',
'Test_927.jpg',
'Test_928.jpg',
'Test_929.jpg',

'Test_930.jpg',
'Test_931.jpg',
'Test_932.jpg',
'Test_933.jpg',
'Test_934.jpg',
'Test_935.jpg',
'Test_936.jpg',
'Test_937.jpg',
'Test_938.jpg',
'Test_939.jpg',
'Test_940.jpg',
'Test_941.jpg',
'Test_942.jpg',
'Test_943.jpg',
'Test_944.jpg',
'Test_945.jpg',
'Test_946.jpg',
'Test_947.jpg',
'Test_948.jpg',
'Test_949.jpg',
'Test_950.jpg',
'Test_951.jpg',
'Test_952.jpg',
'Test_953.jpg',
'Test_954.jpg',
'Test_955.jpg',
'Test_956.jpg',
'Test_957.jpg',
'Test_958.jpg',
'Test_959.jpg',
'Test_960.jpg',
'Test_961.jpg',
'Test_962.jpg',
'Test_963.jpg',
'Test_964.jpg',
'Test_965.jpg',
'Test_966.jpg',
'Test_967.jpg',
'Test_968.jpg',
'Test_969.jpg',
'Test_970.jpg',
'Test_971.jpg',
'Test_972.jpg',
'Test_973.jpg',
'Test_974.jpg',
'Test_975.jpg',
'Test_976.jpg',
'Test_977.jpg',
'Test_978.jpg',
'Test_979.jpg',

```

'Test_980.jpg',
'Test_981.jpg',
'Test_982.jpg',
'Test_983.jpg',
'Test_984.jpg',
'Test_985.jpg',
'Test_986.jpg',
'Test_987.jpg',
'Test_988.jpg',
'Test_989.jpg',
'Test_990.jpg',
'Test_991.jpg',
'Test_992.jpg',
'Test_993.jpg',
'Test_994.jpg',
'Test_995.jpg',
'Test_996.jpg',
'Test_997.jpg',
'Test_998.jpg',
'Test_999.jpg',
...]

def get_label_img(img):
    if search("Train",img):
        img=img.split('.')[0]
        label=train.loc[train['image_id']==img]['label']
        return label

def create_train_data():
    images=natsort.natsorted(os.listdir(DIR))
    for img in tqdm(images):
        label=get_label_img(img)
        path=os.path.join(DIR,img)

        if search("Train",img):
            if (img.split("_")[1].split(".")[0]) and label.item()==0:
                shutil.copy(path,r'/content/drive/My Drive/plant-
pathology-2020-fgvc7/images/train/healthy')

            elif(img.split("_")[1].split(".")[0]) and label.item()==1:
                shutil.copy(path,r'/content/drive/My Drive/plant-
pathology-2020-fgvc7/images/train/multiple_disease')

            elif(img.split("_")[1].split(".")[0]) and label.item()==2:
                shutil.copy(path,r'/content/drive/My Drive/plant-
pathology-2020-fgvc7/images/train/rust')

            elif(img.split("_")[1].split(".")[0]) and label.item()==3:
                shutil.copy(path,r'/content/drive/My Drive/plant-
pathology-2020-fgvc7/images/train/scab')

```

```

        elif search("Test",img):
            shutil.copy(path,r'/content/drive/My Drive/plant-
pathology-2020-fgvc7/images/test')

try:
    shutil.os.mkdir(r'/content/drive/My Drive/plant-pathology-2020-
fgvc7/images/train')
    shutil.os.mkdir(r'/content/drive/My Drive/plant-pathology-2020-
fgvc7/images/train/healthy')
    shutil.os.mkdir(r'/content/drive/My Drive/plant-pathology-2020-
fgvc7/images/train/multiple_disease')
    shutil.os.mkdir(r'/content/drive/My Drive/plant-pathology-2020-
fgvc7/images/train/rust')
    shutil.os.mkdir(r'/content/drive/My Drive/plant-pathology-2020-
fgvc7/images/train/scab')
    shutil.os.mkdir(r'/content/drive/My Drive/plant-pathology-2020-
fgvc7/images/test')
except FileExistsError:
    print("Already exists");

```

Already exists

train_dir=create_train_data()

100%|██████████| 3644/3644 [16:41<00:00, 3.64it/s]

DATA PREPROCESSING

```

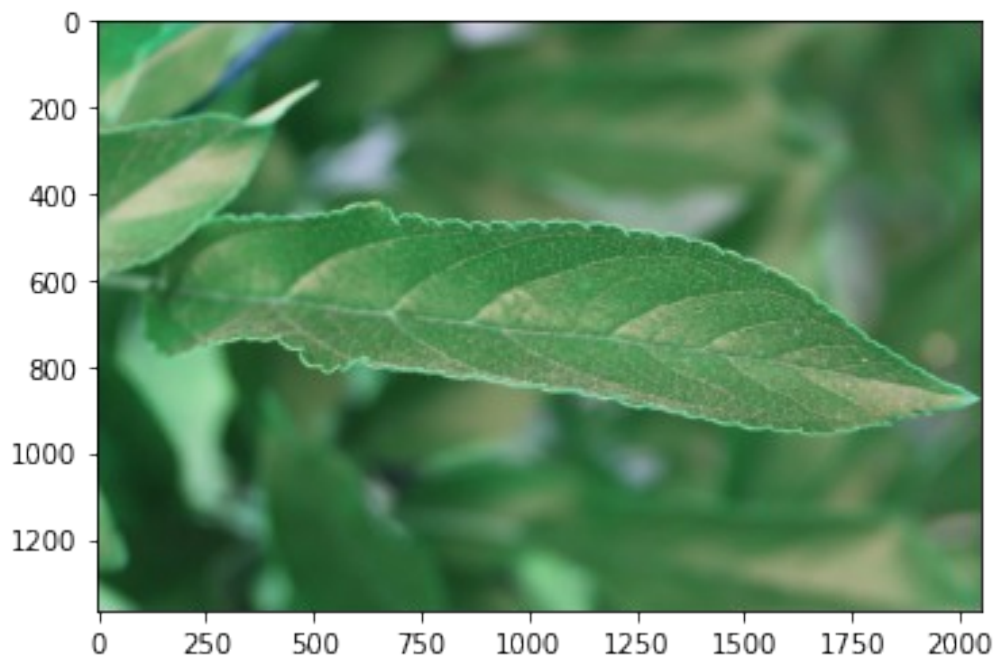
Train_DIR=r'/content/drive/My
Drive/plant-pathology-2020-fgvc7/images/train'
Categories=['healthy','multiple_disease','rust','scab']

```

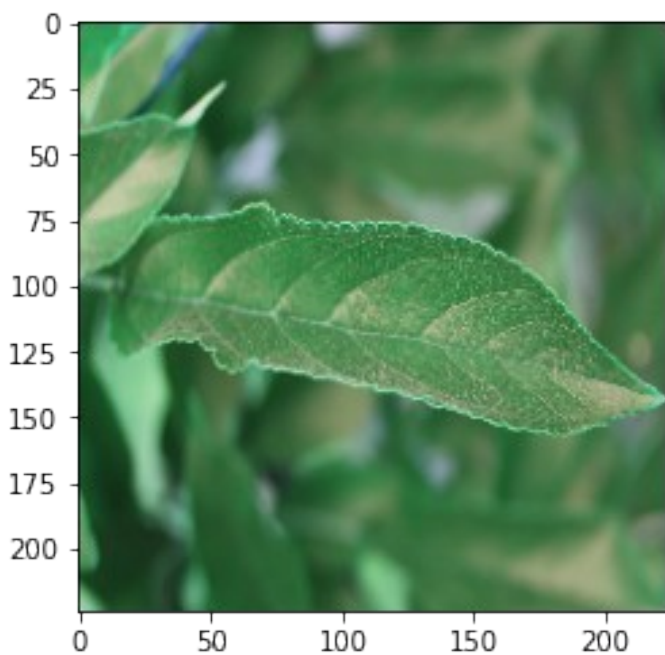
```

for j in Categories:
    path=os.path.join(Train_DIR,j)
    for img in os.listdir(path):
        old_image=cv2.imread(os.path.join(path,img),cv2.COLOR_BGR2RGB)
        plt.imshow(old_image)
        plt.show()
        break
    break

```

```
IMG_SIZE=224  
new_image=cv2.resize(old_image,(IMG_SIZE,IMG_SIZE))  
plt.imshow(new_image)  
plt.show()
```



MODEL PREPARATION

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Dense, Activation, Flatten, Conv2D,
MaxPooling2D

datagen=ImageDataGenerator(rescale=1./255,
                           shear_range=0.2,
                           zoom_range=0.2,
                           horizontal_flip=True,
                           vertical_flip=True,
                           validation_split=0.2)

train_datagen=datagen.flow_from_directory(r'/content/drive/My
Drive/plant-pathology-2020-fgvc7/images/train',

target_size=(IMG_SIZE,IMG_SIZE),

                                     batch_size=16,
                                     class_mode='categorical',
                                     subset='training')

val_datagen=datagen.flow_from_directory(r'/content/drive/My
Drive/plant-pathology-2020-fgvc7/images/train',

target_size=(IMG_SIZE,IMG_SIZE),

                                     batch_size=16,
                                     class_mode='categorical',
                                     subset='validation')

Found 1458 images belonging to 4 classes.
Found 363 images belonging to 4 classes.

model=Sequential()
model.add(Conv2D(64,
(3,3),activation='relu',padding='same',input_shape=(IMG_SIZE,IMG_SIZE,
3)))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64,(3,3),activation='relu',padding='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64,(3,3),activation='relu',padding='same'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(128,(3,3),activation='relu',padding='same'))
model.add(MaxPooling2D(2,2))
model.add(Flatten())
model.add(Dense(4,activation='softmax'))

# Compile the Model
```

```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 224, 224, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_1 (Conv2D)	(None, 112, 112, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 64)	0
conv2d_2 (Conv2D)	(None, 56, 56, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 64)	0
conv2d_3 (Conv2D)	(None, 28, 28, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 128)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 4)	100356
Total params: 249,860		
Trainable params: 249,860		
Non-trainable params: 0		

```
checkpoint=ModelCheckpoint(r'/content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5',
                           monitor='val_loss',
                           mode='min',
                           save_best_only=True,
                           verbose=1)
earlystop=EarlyStopping(monitor='val_loss',
                         min_delta=0,
                         patience=10,
                         verbose=1,
                         restore_best_weights=True)
```

```
callbacks=[checkpoint,earlystop]
model_history=model.fit(train_datagen,validation_data=val_datagen,
                        epochs=30,

steps_per_epoch=train_datagen.samples//16,

validation_steps=val_datagen.samples//16,
                        callbacks=callbacks)
```

```
Epoch 1/30
91/91 [=====] - ETA: 0s - loss: 1.2906 -
accuracy: 0.3585
Epoch 00001: val_loss did not improve from 1.23252
91/91 [=====] - 318s 3s/step - loss: 1.2906 -
accuracy: 0.3585 - val_loss: 1.2613 - val_accuracy: 0.3494
Epoch 2/30
91/91 [=====] - ETA: 0s - loss: 1.2425 -
accuracy: 0.3245
Epoch 00002: val_loss did not improve from 1.23252
91/91 [=====] - 291s 3s/step - loss: 1.2425 -
accuracy: 0.3245 - val_loss: 1.2515 - val_accuracy: 0.3125
Epoch 3/30
91/91 [=====] - ETA: 0s - loss: 1.2462 -
accuracy: 0.3322
Epoch 00003: val_loss did not improve from 1.23252
91/91 [=====] - 290s 3s/step - loss: 1.2462 -
accuracy: 0.3322 - val_loss: 1.2629 - val_accuracy: 0.2983
Epoch 4/30
91/91 [=====] - ETA: 0s - loss: 1.2463 -
accuracy: 0.3051
Epoch 00004: val_loss did not improve from 1.23252
91/91 [=====] - 288s 3s/step - loss: 1.2463 -
accuracy: 0.3051 - val_loss: 1.2398 - val_accuracy: 0.3381
Epoch 5/30
91/91 [=====] - ETA: 0s - loss: 1.2377 -
accuracy: 0.3412
Epoch 00005: val_loss did not improve from 1.23252
91/91 [=====] - 291s 3s/step - loss: 1.2377 -
accuracy: 0.3412 - val_loss: 1.2364 - val_accuracy: 0.3210
Epoch 6/30
91/91 [=====] - ETA: 0s - loss: 1.2330 -
accuracy: 0.3655
Epoch 00006: val_loss did not improve from 1.23252
91/91 [=====] - 292s 3s/step - loss: 1.2330 -
accuracy: 0.3655 - val_loss: 1.2363 - val_accuracy: 0.3835
Epoch 7/30
91/91 [=====] - ETA: 0s - loss: 1.2294 -
accuracy: 0.3745
```

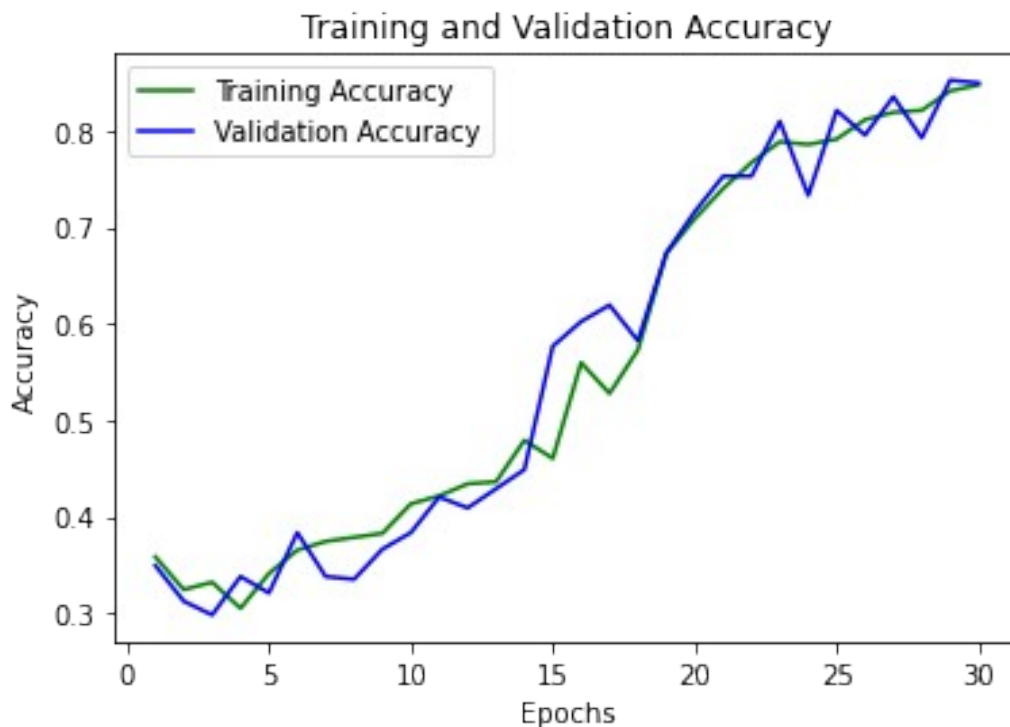
Epoch 00007: val_loss did not improve from 1.23252
91/91 [=====] - 289s 3s/step - loss: 1.2294 -
accuracy: 0.3745 - val_loss: 1.2435 - val_accuracy: 0.3381
Epoch 8/30
91/91 [=====] - ETA: 0s - loss: 1.2292 -
accuracy: 0.3786
Epoch 00008: val_loss did not improve from 1.23252
91/91 [=====] - 289s 3s/step - loss: 1.2292 -
accuracy: 0.3786 - val_loss: 1.2392 - val_accuracy: 0.3352
Epoch 9/30
91/91 [=====] - ETA: 0s - loss: 1.2304 -
accuracy: 0.3828
Epoch 00009: val_loss improved from 1.23252 to 1.21881, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 289s 3s/step - loss: 1.2304 -
accuracy: 0.3828 - val_loss: 1.2188 - val_accuracy: 0.3665
Epoch 10/30
91/91 [=====] - ETA: 0s - loss: 1.2122 -
accuracy: 0.4133
Epoch 00010: val_loss did not improve from 1.21881
91/91 [=====] - 289s 3s/step - loss: 1.2122 -
accuracy: 0.4133 - val_loss: 1.2307 - val_accuracy: 0.3835
Epoch 11/30
91/91 [=====] - ETA: 0s - loss: 1.2014 -
accuracy: 0.4216
Epoch 00011: val_loss improved from 1.21881 to 1.20129, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 290s 3s/step - loss: 1.2014 -
accuracy: 0.4216 - val_loss: 1.2013 - val_accuracy: 0.4205
Epoch 12/30
91/91 [=====] - ETA: 0s - loss: 1.1904 -
accuracy: 0.4341
Epoch 00012: val_loss did not improve from 1.20129
91/91 [=====] - 300s 3s/step - loss: 1.1904 -
accuracy: 0.4341 - val_loss: 1.2174 - val_accuracy: 0.4091
Epoch 13/30
91/91 [=====] - ETA: 0s - loss: 1.1880 -
accuracy: 0.4362
Epoch 00013: val_loss improved from 1.20129 to 1.16994, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 289s 3s/step - loss: 1.1880 -
accuracy: 0.4362 - val_loss: 1.1699 - val_accuracy: 0.4290
Epoch 14/30
91/91 [=====] - ETA: 0s - loss: 1.1554 -
accuracy: 0.4792
Epoch 00014: val_loss did not improve from 1.16994
91/91 [=====] - 289s 3s/step - loss: 1.1554 -
accuracy: 0.4792 - val_loss: 1.1990 - val_accuracy: 0.4489
Epoch 15/30
91/91 [=====] - ETA: 0s - loss: 1.1583 -

accuracy: 0.4605
Epoch 00015: val_loss improved from 1.16994 to 1.02666, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 291s 3s/step - loss: 1.1583 -
accuracy: 0.4605 - val_loss: 1.0267 - val_accuracy: 0.5767
Epoch 16/30
91/91 [=====] - ETA: 0s - loss: 1.0474 -
accuracy: 0.5596
Epoch 00016: val_loss improved from 1.02666 to 0.99252, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 289s 3s/step - loss: 1.0474 -
accuracy: 0.5596 - val_loss: 0.9925 - val_accuracy: 0.6023
Epoch 17/30
91/91 [=====] - ETA: 0s - loss: 1.0925 -
accuracy: 0.5277
Epoch 00017: val_loss did not improve from 0.99252
91/91 [=====] - 288s 3s/step - loss: 1.0925 -
accuracy: 0.5277 - val_loss: 1.0239 - val_accuracy: 0.6193
Epoch 18/30
91/91 [=====] - ETA: 0s - loss: 1.0197 -
accuracy: 0.5728
Epoch 00018: val_loss improved from 0.99252 to 0.98166, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 289s 3s/step - loss: 1.0197 -
accuracy: 0.5728 - val_loss: 0.9817 - val_accuracy: 0.5824
Epoch 19/30
91/91 [=====] - ETA: 0s - loss: 0.8323 -
accuracy: 0.6727
Epoch 00019: val_loss improved from 0.98166 to 0.80770, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 289s 3s/step - loss: 0.8323 -
accuracy: 0.6727 - val_loss: 0.8077 - val_accuracy: 0.6733
Epoch 20/30
91/91 [=====] - ETA: 0s - loss: 0.7403 -
accuracy: 0.7087
Epoch 00020: val_loss improved from 0.80770 to 0.69359, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 289s 3s/step - loss: 0.7403 -
accuracy: 0.7087 - val_loss: 0.6936 - val_accuracy: 0.7159
Epoch 21/30
91/91 [=====] - ETA: 0s - loss: 0.6544 -
accuracy: 0.7399
Epoch 00021: val_loss improved from 0.69359 to 0.65146, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 290s 3s/step - loss: 0.6544 -
accuracy: 0.7399 - val_loss: 0.6515 - val_accuracy: 0.7528
Epoch 22/30
91/91 [=====] - ETA: 0s - loss: 0.6044 -
accuracy: 0.7670
Epoch 00022: val_loss improved from 0.65146 to 0.62033, saving model

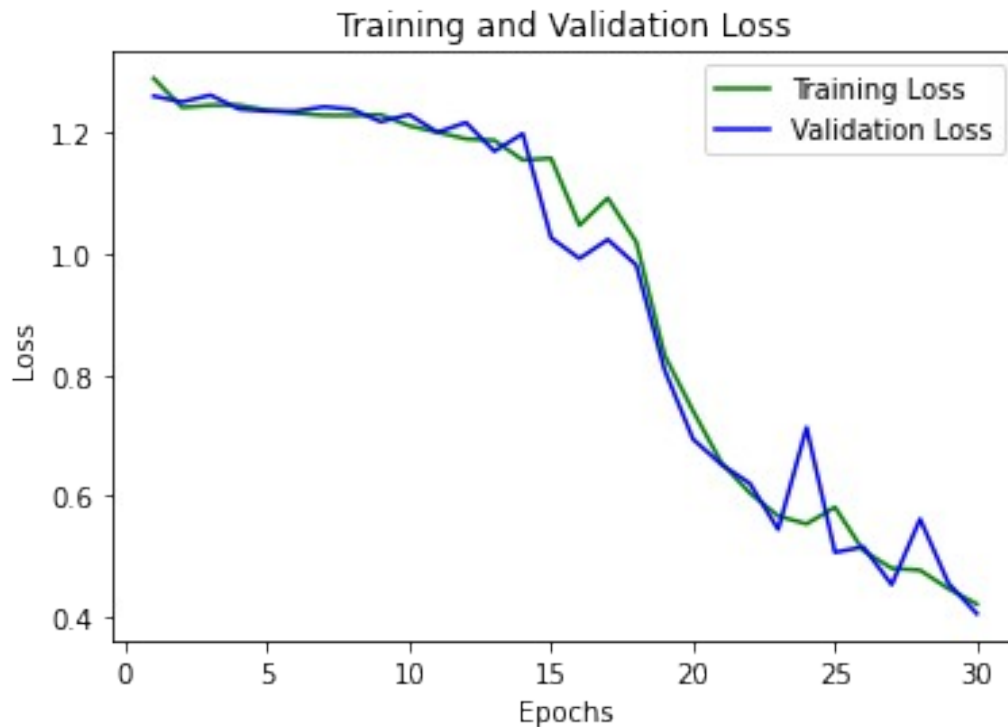
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 288s 3s/step - loss: 0.6044 -
accuracy: 0.7670 - val_loss: 0.6203 - val_accuracy: 0.7528
Epoch 23/30
91/91 [=====] - ETA: 0s - loss: 0.5663 -
accuracy: 0.7885
Epoch 00023: val_loss improved from 0.62033 to 0.54409, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 287s 3s/step - loss: 0.5663 -
accuracy: 0.7885 - val_loss: 0.5441 - val_accuracy: 0.8097
Epoch 24/30
91/91 [=====] - ETA: 0s - loss: 0.5532 -
accuracy: 0.7857
Epoch 00024: val_loss did not improve from 0.54409
91/91 [=====] - 287s 3s/step - loss: 0.5532 -
accuracy: 0.7857 - val_loss: 0.7128 - val_accuracy: 0.7330
Epoch 25/30
91/91 [=====] - ETA: 0s - loss: 0.5809 -
accuracy: 0.7913
Epoch 00025: val_loss improved from 0.54409 to 0.50619, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 288s 3s/step - loss: 0.5809 -
accuracy: 0.7913 - val_loss: 0.5062 - val_accuracy: 0.8210
Epoch 26/30
91/91 [=====] - ETA: 0s - loss: 0.5090 -
accuracy: 0.8114
Epoch 00026: val_loss did not improve from 0.50619
91/91 [=====] - 289s 3s/step - loss: 0.5090 -
accuracy: 0.8114 - val_loss: 0.5152 - val_accuracy: 0.7955
Epoch 27/30
91/91 [=====] - ETA: 0s - loss: 0.4798 -
accuracy: 0.8190
Epoch 00027: val_loss improved from 0.50619 to 0.45259, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 290s 3s/step - loss: 0.4798 -
accuracy: 0.8190 - val_loss: 0.4526 - val_accuracy: 0.8352
Epoch 28/30
91/91 [=====] - ETA: 0s - loss: 0.4765 -
accuracy: 0.8211
Epoch 00028: val_loss did not improve from 0.45259
91/91 [=====] - 289s 3s/step - loss: 0.4765 -
accuracy: 0.8211 - val_loss: 0.5612 - val_accuracy: 0.7926
Epoch 29/30
91/91 [=====] - ETA: 0s - loss: 0.4459 -
accuracy: 0.8412
Epoch 00029: val_loss did not improve from 0.45259
91/91 [=====] - 289s 3s/step - loss: 0.4459 -
accuracy: 0.8412 - val_loss: 0.4548 - val_accuracy: 0.8523
Epoch 30/30
91/91 [=====] - ETA: 0s - loss: 0.4204 -

```
accuracy: 0.8474
Epoch 00030: val_loss improved from 0.45259 to 0.40454, saving model
to /content/drive/My Drive/plant-pathology-2020-fgvc7/models/apple2.h5
91/91 [=====] - 291s 3s/step - loss: 0.4204 -
accuracy: 0.8474 - val_loss: 0.4045 - val_accuracy: 0.8494
```

```
acc_train=model_history.history['accuracy']
acc_val=model_history.history['val_accuracy']
epochs=range(1,31)
plt.plot(epochs,acc_train,'g',label='Training Accuracy')
plt.plot(epochs,acc_val,'b',label='Validation Accuracy')
plt.title("Training and Validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()
```



```
loss_train=model_history.history['loss']
loss_val=model_history.history['val_loss']
epochs=range(1,31)
plt.plot(epochs,loss_train,'g',label='Training Loss')
plt.plot(epochs,loss_val,'b',label='Validation Loss')
plt.title("Training and Validation Loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()
```

MAKING THE PREDICTION ON A SINGLE IMAGE

```
test_image=r'/content/drive/My
Drive/plant-pathology-2020-fgvc7/images/train/rust/Train_3.jpg'
image_result=Image.open(test_image)
```

```
from tensorflow.keras.preprocessing import image
test_image=image.load_img(test_image,target_size=(224,224))
test_image=image.img_to_array(test_image)
test_image=test_image/255
test_image=np.expand_dims(test_image,axis=0)
result=model.predict(test_image)
print(np.argmax(result))
Categories=['healthy','multiple_disease','rust','scab']
image_result=plt.imshow(image_result)
plt.title(Categories[np.argmax(result)])
plt.show()
```

rust

