

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: import tensorflow as tf
import os
import cv2 as cv
```

```
In [3]: from tensorflow import keras
```

```
In [4]: from keras import layers, Sequential, Model
```

```
In [5]: from keras.layers import Flatten, Dropout, Dense, Input, Lambda, MaxPooling3D, Conv2D
```

```
In [6]: path=r"C:\Users\HP\Downloads\banana\BananaLSD\OriginalSet"
```

```
In [7]: dis=os.listdir(path)
```

```
In [8]: dis
```

```
Out[8]: ['cordana', 'healthy', 'pestalotiopsis', 'sigatoka']
```

```
In [9]: Data=[]
```

```
In [10]: for i in dis:
    A=os.path.join(path,i)
    for j in os.listdir(A):
        B=os.path.join(A,j)
        img=cv.imread(B)
        C=cv.resize(img,(130,130))
        T=dis.index(i)
        Data.append([C,T])
```

```
In [11]: import random
random.shuffle(Data)
```

```
In [12]: T=dis.index('cordana')
print(T)
```

0

```
In [13]: F=[]  
        T=[]  
        for i,j in Data:  
            F.append(i)  
            T.append(j)
```

```
In [14]: F=np.array(F)
```

```
In [15]: T=np.array(T)
```

```
In [16]: F=F/25
```

```
In [17]: T=pd.get_dummies(T,dtype="int")
```

VGG16

```
In [18]: from keras.applications.vgg16 import VGG16
```

```
In [19]: from keras.applications.vgg16 import preprocess_input
```

```
In [20]: vgg=VGG16(input_shape=(130,130,3),weights='imagenet',include_top=False)
```

```
In [21]: for i in vgg.layers:  
        i.trainable=False
```

```
In [22]: X=Flatten()(vgg.output)
```

```
In [23]: y=Dense(20,activation='relu')(X)
```


```
In [24]: z=Dense(4,activation='softmax')(y)
```


```
In [25]: model=Model(vgg.input,z)
```


```
In [26]: from keras.losses import CategoricalCrossentropy
```


```
In [27]: model.compile(optimizer="adam",  
                      loss="CategoricalCrossentropy",  
                      metrics=["accuracy"])
```


```
In [28]: model.fit(F,T,epochs=20,validation_split=0.15)
```


Epoch 1/20
25/25  54s 2s/step - accuracy: 0.4005 - loss: 2.2554 - val_accuracy: 0.5319 - val_loss: 1.2082


Epoch 2/20
25/25  52s 2s/step - accuracy: 0.4982 - loss: 1.1819 - val_accuracy: 0.5390 - val_loss: 0.9672


Epoch 3/20
25/25  54s 2s/step - accuracy: 0.5334 - loss: 0.9057 - val_accuracy: 0.5532 - val_loss: 0.8324


Epoch 4/20
25/25  52s 2s/step - accuracy: 0.5847 - loss: 0.8526 - val_accuracy: 0.4397 - val_loss: 1.2445


Epoch 5/20
25/25  53s 2s/step - accuracy: 0.5846 - loss: 1.0099 - val_accuracy: 0.6738 - val_loss: 0.8477


Epoch 6/20
25/25  52s 2s/step - accuracy: 0.6699 - loss: 0.7863 - val_accuracy: 0.7021 - val_loss: 0.7412


Epoch 7/20
25/25  52s 2s/step - accuracy: 0.7216 - loss: 0.7038 - val_accuracy: 0.6879 - val_loss: 0.7326


Epoch 8/20
25/25  52s 2s/step - accuracy: 0.7003 - loss: 0.7131 - val_accuracy: 0.7305 - val_loss: 0.6671


Epoch 9/20
25/25  53s 2s/step - accuracy: 0.7104 - loss: 0.6928 - val_accuracy: 0.6667 - val_loss: 0.6346

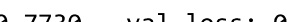
Epoch 10/20
25/25  53s 2s/step - accuracy: 0.7082 - loss: 0.6534 - val_accuracy: 0.7376 - val_loss: 0.6100

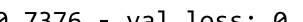
Epoch 11/20
25/25  54s 2s/step - accuracy: 0.7724 - loss: 0.6027 - val_accuracy: 0.6950 - val_loss: 0.7253

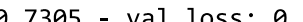
Epoch 12/20
25/25  55s 2s/step - accuracy: 0.7315 - loss: 0.7275 - val_accuracy: 0.7163 - val_loss: 0.6042


Epoch 13/20
25/25  55s 2s/step - accuracy: 0.7380 - loss: 0.6019 - val_accuracy: 0.7801 - val_loss: 0.6060

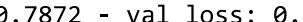
Epoch 14/20
25/25  56s 2s/step - accuracy: 0.7935 - loss: 0.5361 - val_accuracy: 0.7376 - val_loss: 0.6290

Epoch 15/20
25/25  56s 2s/step - accuracy: 0.8334 - loss: 0.4767 - val_accuracy: 0.7730 - val_loss: 0.6996

Epoch 16/20
25/25  55s 2s/step - accuracy: 0.8308 - loss: 0.4663 - val_accuracy: 0.7376 - val_loss: 0.6625

Epoch 17/20
25/25  55s 2s/step - accuracy: 0.7751 - loss: 0.5768 - val_accuracy: 0.7305 - val_loss: 0.9035

Epoch 18/20
25/25  55s 2s/step - accuracy: 0.8091 - loss: 0.5119 - val_accuracy: 0.7730 - val_loss: 0.7505

Epoch 19/20
25/25  55s 2s/step - accuracy: 0.8111 - loss: 0.4996 - val_accuracy: 0.7872 - val_loss: 0.6307

Epoch 20/20

25/25 ————— 55s 2s/step - accuracy: 0.7925 - loss: 0.5100 - val_accuracy: 0.7872 - val_loss: 0.9144

Out[28]: <keras.src.callbacks.history.History at 0x13e5d7a85d0>

```
In [90]: img=r"C:\Users\HP\Downloads\banana\BananaLSD\OriginalSet\sigatoka\5.jpeg"
```

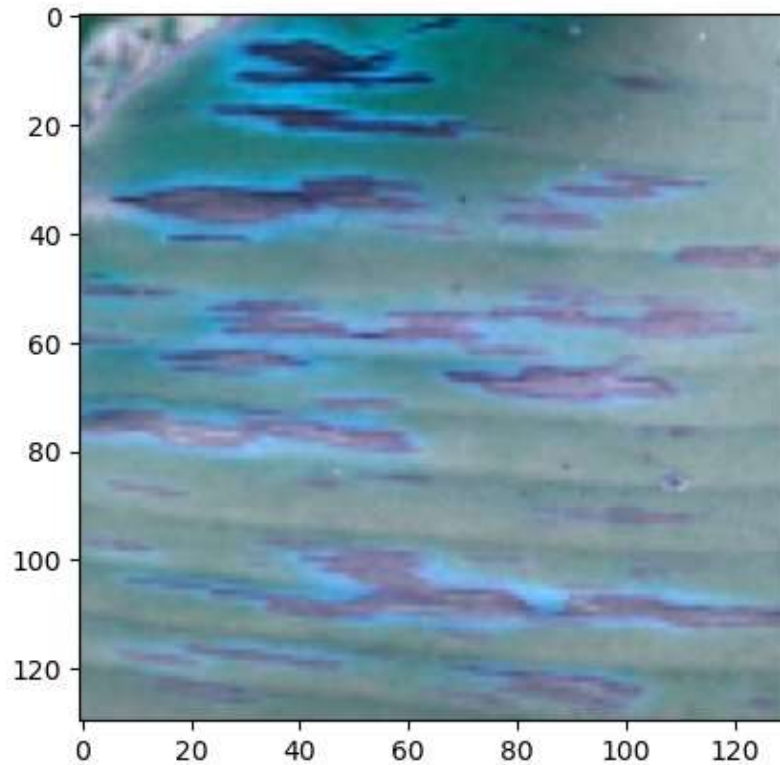
```
In [91]: img1=cv.imread(img)
```

```
In [92]: img2=cv.resize(img1,(130,130))
```

```
In [93]: img31=img2.reshape(1,130,130,3)
```

```
In [94]: plt.imshow(img2)
```

Out[94]: <matplotlib.image.AxesImage at 0x13eec51b390>



```
In [95]: pred1=model.predict(img31)
```

1/1 ————— 0s 48ms/step

```
In [96]: np.argmax(pred1)
```

Out[96]: 3

VGG19

```
In [36]: from keras.applications.vgg19 import VGG19
```

```
In [37]: from keras.applications.vgg19 import preprocess_input
```

```
In [38]: vgg=VGG19(input_shape=(130,130,3),weights='imagenet',include_top=False)
```

```
In [39]: for i in vgg.layers:  
        i.trainable=False
```

```
In [40]: X=Flatten()(vgg.output)
```

```
In [41]: y=Dense(20,activation='relu')(X)
```


```
In [42]: z=Dense(4,activation='softmax')(y)
```


```
In [43]: model=Model(vgg.input,z)
```


```
In [44]: from keras.losses import CategoricalCrossentropy
```


```
In [45]: model.compile(optimizer="adam",  
                      loss="CategoricalCrossentropy",  
                      metrics=["accuracy"])
```


```
In [46]: model.fit(F,T,epochs=20,validation_split=0.15)
```


Epoch 1/20
25/25  73s 3s/step - accuracy: 0.3825 - loss: 5.1380 - val_accuracy: 0.5532 - val_loss: 0.9374


Epoch 2/20
25/25  71s 3s/step - accuracy: 0.5313 - loss: 0.9899 - val_accuracy: 0.5674 - val_loss: 0.8830


Epoch 3/20
25/25  70s 3s/step - accuracy: 0.5550 - loss: 0.9217 - val_accuracy: 0.5887 - val_loss: 0.8596


Epoch 4/20
25/25  72s 3s/step - accuracy: 0.6041 - loss: 0.9165 - val_accuracy: 0.5461 - val_loss: 0.8855


Epoch 5/20
25/25  70s 3s/step - accuracy: 0.5819 - loss: 0.9033 - val_accuracy: 0.6312 - val_loss: 0.8009


Epoch 6/20
25/25  70s 3s/step - accuracy: 0.6159 - loss: 0.9088 - val_accuracy: 0.6170 - val_loss: 0.7900


Epoch 7/20
25/25  69s 3s/step - accuracy: 0.6276 - loss: 0.8117 - val_accuracy: 0.7234 - val_loss: 0.7246

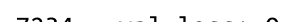
Epoch 8/20
25/25  69s 3s/step - accuracy: 0.6178 - loss: 0.8143 - val_accuracy: 0.5957 - val_loss: 0.8203

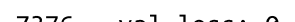
Epoch 9/20
25/25  70s 3s/step - accuracy: 0.5882 - loss: 0.8613 - val_accuracy: 0.6525 - val_loss: 0.7306

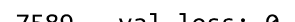
Epoch 10/20
25/25  72s 3s/step - accuracy: 0.6352 - loss: 0.8277 - val_accuracy: 0.6312 - val_loss: 0.7413

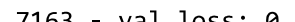
Epoch 11/20
25/25  71s 3s/step - accuracy: 0.6042 - loss: 0.7981 - val_accuracy: 0.6809 - val_loss: 0.6839


Epoch 12/20
25/25  71s 3s/step - accuracy: 0.6193 - loss: 0.8511 - val_accuracy: 0.6596 - val_loss: 0.7162


Epoch 13/20
25/25  70s 3s/step - accuracy: 0.6469 - loss: 0.7930 - val_accuracy: 0.7234 - val_loss: 0.5743


Epoch 14/20
25/25  70s 3s/step - accuracy: 0.6846 - loss: 0.6941 - val_accuracy: 0.7376 - val_loss: 0.5724

Epoch 15/20
25/25  73s 3s/step - accuracy: 0.7043 - loss: 0.6282 - val_accuracy: 0.7589 - val_loss: 0.4965

Epoch 16/20
25/25  71s 3s/step - accuracy: 0.7193 - loss: 0.5547 - val_accuracy: 0.7163 - val_loss: 0.6175

Epoch 17/20
25/25  70s 3s/step - accuracy: 0.7084 - loss: 0.5551 - val_accuracy: 0.6667 - val_loss: 0.7286

Epoch 18/20
25/25  69s 3s/step - accuracy: 0.6725 - loss: 0.6653 - val_accuracy: 0.7660 - val_loss: 0.4877

Epoch 19/20
25/25  75s 3s/step - accuracy: 0.7440 - loss: 0.6033 - val_accuracy: 0.8014 - val_loss: 0.4482

Epoch 20/20

25/25 ————— 91s 4s/step - accuracy: 0.7449 - loss: 0.5392 - val_accuracy: 0.7376 - val_loss: 0.5823

Out[46]: <keras.src.callbacks.history.History at 0x13e00859f50>

```
In [97]: img=r"C:\Users\HP\Downloads\banana\BananaLSD\OriginalSet\pestalotiopsis\84.jpeg"
```

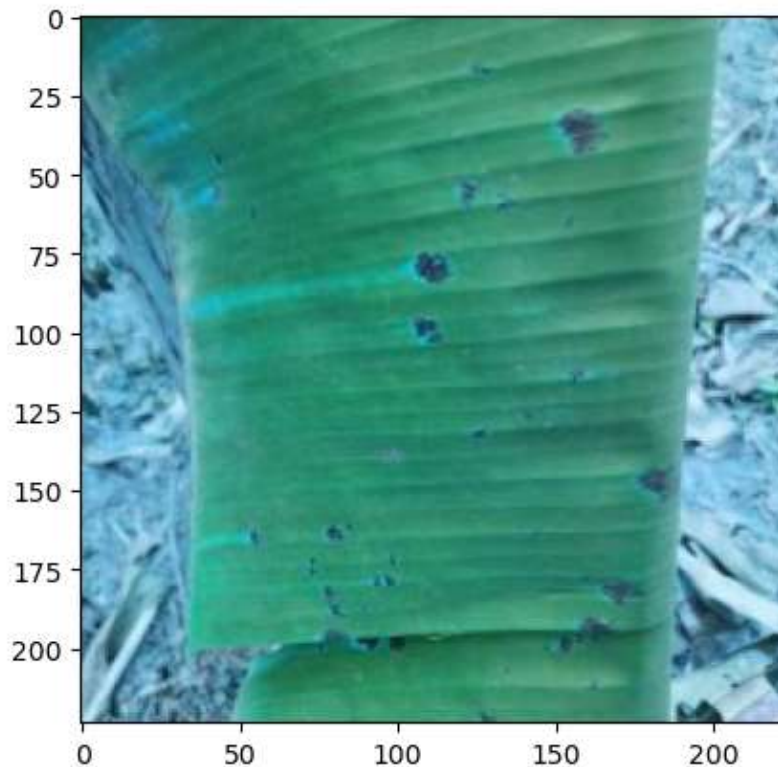
```
In [98]: img1=cv.imread(img)
```

```
In [99]: img2=cv.resize(img1,(130,130))
```

```
In [100]: img3=img2.reshape(1,130,130,3)
```

```
In [101]: plt.imshow(img1)
```

Out[101]: <matplotlib.image.AxesImage at 0x13eef8b00d0>



```
In [102]: pred2=model.predict(img3)
```

1/1 ————— 0s 48ms/step

```
In [53]: np.argmax(pred2)
```

Out[53]: 2

Resnet

```
In [54]: from keras.applications.resnet import ResNet50
```

```
In [55]: from keras.applications.resnet import preprocess_input
```

```
In [56]: resnet=ResNet50(input_shape=(130,130,3),weights='imagenet',include_top=False)
```

```
In [57]: for i in resnet.layers:  
        i.trainable=False
```

```
In [58]: X=Flatten()(resnet.output)
```

```
In [59]: y=Dense(20,activation='relu')(X)
```

```
In [60]: z=Dense(4,activation='softmax')(y)
```

```
In [61]: model=Model(resnet.input,z)
```

```
In [62]: from keras.losses import CategoricalCrossentropy
```

```
In [63]: model.compile(optimizer="adam",  
                      loss="CategoricalCrossentropy",  
                      metrics=["accuracy"])
```

```
In [64]: model.fit(F,T,epochs=20,validation_split=0.15)
```

```
Epoch 1/10
25/25 ————— 76s 2s/step - accuracy: 0.5902 - loss: 2.6507 - val_accu
acy: 0.5319 - val_loss: 3690005.0000
Epoch 2/10
25/25 ————— 44s 2s/step - accuracy: 0.7678 - loss: 0.6864 - val_accu
acy: 0.5319 - val_loss: 6139782.5000
Epoch 3/10
25/25 ————— 2631s 110s/step - accuracy: 0.8477 - loss: 0.5850 - val_a
ccuracy: 0.4397 - val_loss: 7428.7617
Epoch 4/10
25/25 ————— 43s 2s/step - accuracy: 0.9180 - loss: 0.2681 - val_accu
acy: 0.1773 - val_loss: 1.4149
Epoch 5/10
25/25 ————— 44s 2s/step - accuracy: 0.9460 - loss: 0.1602 - val_accu
acy: 0.1348 - val_loss: 2.8440
Epoch 6/10
25/25 ————— 45s 2s/step - accuracy: 0.9716 - loss: 0.0972 - val_accu
acy: 0.1348 - val_loss: 4.9927
Epoch 7/10
25/25 ————— 44s 2s/step - accuracy: 0.9862 - loss: 0.0400 - val_accu
acy: 0.1348 - val_loss: 7.1636
Epoch 8/10
25/25 ————— 45s 2s/step - accuracy: 0.9698 - loss: 0.1068 - val_accu
acy: 0.1348 - val_loss: 12.4321
Epoch 9/10
25/25 ————— 44s 2s/step - accuracy: 0.9766 - loss: 0.0857 - val_accu
acy: 0.1348 - val_loss: 16.6639
Epoch 10/10
25/25 ————— 44s 2s/step - accuracy: 0.9778 - loss: 0.0830 - val_accu
acy: 0.1560 - val_loss: 8.0376
```

```
Out[64]: <keras.src.callbacks.history.History at 0x13e01b74850>
```

```
In [137]: img=r"C:\Users\HP\Downloads\banana\BananaLSD\OriginalSet\sigatoka\51.jpeg"
```

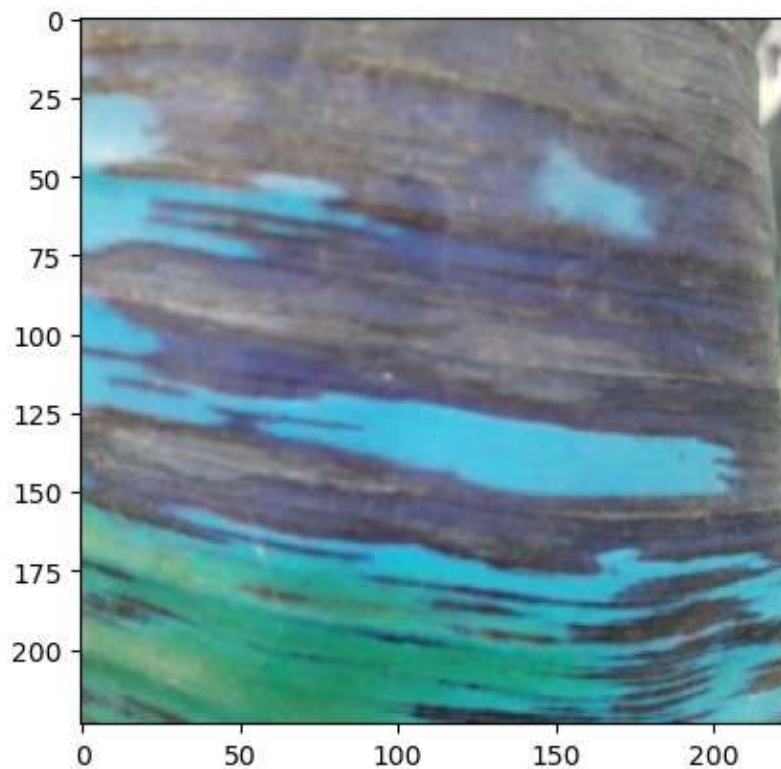
```
In [138]: img1=cv.imread(img)
```

```
In [139]: img2=cv.resize(img1,(130,130))
```

```
In [140]: img3=img2.reshape(1,130,130,3)
```

```
In [141]: plt.imshow(img1)
```

```
Out[141]: <matplotlib.image.AxesImage at 0x13ef0a3e250>
```



```
In [142]: pred3=model.predict(img3)
```

1/1 ————— 0s 63ms/step

```
In [143]: np.argmax(pred3)
```

```
Out[143]: 3
```

```
In [ ]:
```

InceptionV3

```
In [72]: from keras.applications import InceptionV3  
model = InceptionV3(weights='imagenet')
```

```
In [73]: from keras.applications.inception_v3 import preprocess_input
```

```
In [74]: inception_v3=InceptionV3(input_shape=(130,130,3),weights='imagenet',include_top=False
```

```
In [75]: for i in inception_v3.layers:  
         i.trainable=False
```

```
In [76]: X=Flatten()(inception_v3.output)
```


```
In [77]: y=Dense(20,activation='relu')(X)
```


```
In [78]: z=Dense(4,activation='softmax')(y)
```


```
In [79]: model=Model(inception_v3.input,z)
```


```
In [80]: model.compile(optimizer="adam",  
                      loss="CategoricalCrossentropy",  
                      metrics=["accuracy"])
```


```
In [81]: model.fit(F,T,epochs=20,validation_split=0.15)
```


Epoch 1/20
25/25  50s 833ms/step - accuracy: 0.5034 - loss: 1.3270 - val_accuracy: 0.3546 - val_loss: 1.2859


Epoch 2/20
25/25  19s 747ms/step - accuracy: 0.8105 - loss: 0.5382 - val_accuracy: 0.1560 - val_loss: 1.3928


Epoch 3/20
25/25  19s 740ms/step - accuracy: 0.9198 - loss: 0.2880 - val_accuracy: 0.1844 - val_loss: 498.3741


Epoch 4/20
25/25  18s 739ms/step - accuracy: 0.8603 - loss: 0.3318 - val_accuracy: 0.3546 - val_loss: 1.3311


Epoch 5/20
25/25  19s 742ms/step - accuracy: 0.8600 - loss: 0.3101 - val_accuracy: 0.4539 - val_loss: 2.6314


Epoch 6/20
25/25  19s 749ms/step - accuracy: 0.9147 - loss: 0.2146 - val_accuracy: 0.4894 - val_loss: 6.1162


Epoch 7/20
25/25  19s 779ms/step - accuracy: 0.9025 - loss: 0.2625 - val_accuracy: 0.2695 - val_loss: 2.4585


Epoch 8/20
25/25  20s 800ms/step - accuracy: 0.8657 - loss: 0.4041 - val_accuracy: 0.6667 - val_loss: 2.3900


Epoch 9/20
25/25  20s 809ms/step - accuracy: 0.8861 - loss: 0.4321 - val_accuracy: 0.6099 - val_loss: 4.6354


Epoch 10/20
25/25  21s 831ms/step - accuracy: 0.9250 - loss: 0.2590 - val_accuracy: 0.6950 - val_loss: 2.5515


Epoch 11/20
25/25  20s 788ms/step - accuracy: 0.9447 - loss: 0.2114 - val_accuracy: 0.6454 - val_loss: 1.3842


Epoch 12/20
25/25  21s 823ms/step - accuracy: 0.9202 - loss: 0.3020 - val_accuracy: 0.6241 - val_loss: 3.7017


Epoch 13/20
25/25  21s 832ms/step - accuracy: 0.8976 - loss: 0.3499 - val_accuracy: 0.5319 - val_loss: 1.3736


Epoch 14/20
25/25  20s 807ms/step - accuracy: 0.9120 - loss: 0.2321 - val_accuracy: 0.5035 - val_loss: 1.3500

Epoch 15/20
25/25  20s 802ms/step - accuracy: 0.9249 - loss: 0.2016 - val_accuracy: 0.6028 - val_loss: 1.4164

Epoch 16/20
25/25  21s 830ms/step - accuracy: 0.9541 - loss: 0.1866 - val_accuracy: 0.7092 - val_loss: 1.3011

Epoch 17/20
25/25  21s 839ms/step - accuracy: 0.9439 - loss: 0.1615 - val_accuracy: 0.5319 - val_loss: 1.8429

Epoch 18/20
25/25  20s 789ms/step - accuracy: 0.9240 - loss: 0.1888 - val_accuracy: 0.2270 - val_loss: 7.0645

Epoch 19/20
25/25  20s 783ms/step - accuracy: 0.9466 - loss: 0.1466 - val_accuracy: 0.6809 - val_loss: 1.7003

Epoch 20/20

25/25 ————— 19s 775ms/step - accuracy: 0.8865 - loss: 0.3664 - val_accuracy: 0.5319 - val_loss: 9.9839

Out[81]: <keras.src.callbacks.history.History at 0x13e046dc850>

```
In [144]: img=r"C:\Users\HP\Downloads\banana\BananaLSD\OriginalSet\sigatoka\51.jpeg"
```

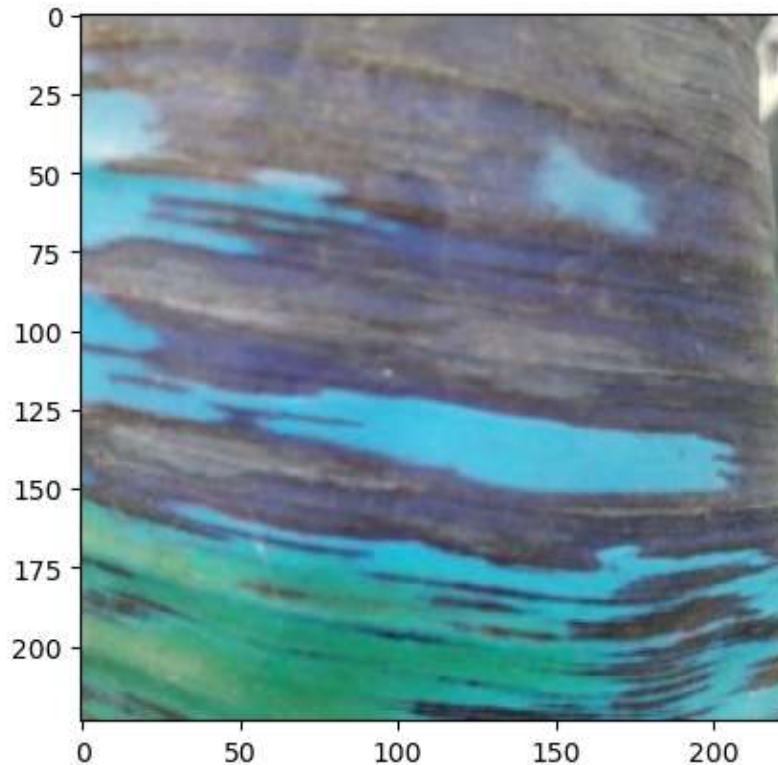
```
In [145]: img1=cv.imread(img)
```

```
In [146]: img2=cv.resize(img1,(130,130))
```

```
In [147]: img3=img2.reshape(1,130,130,3)
```

```
In [148]: plt.imshow(img1)
```

Out[148]: <matplotlib.image.AxesImage at 0x13ef0abae50>



```
In [149]: pred4=model.predict(img3)
```

1/1 ————— 0s 48ms/step

```
In [150]: np.argmax(pred4)
```

Out[150]: 3

```
In [ ]:
```