Importing Libraries

In [1]:
```
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
```

Loading Model

In [2]:
```
model = tf.keras.models.load_model('trained_plant_disease_model.keras')
```

```
C:\Users\gupta\Desktop\plant disease detection\tfenv\lib\site-packages\keras\src\saving\saving_lib.py:418: UserWarning: Skipping variable l
oading for optimizer 'rmsprop', because it has 26 variables whereas the saved optimizer has 50 variables.
  trackable.load_own_variables(weights_store.get(inner_path))
```

In [3]:
```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 128, 32)	896
conv2d_1 (Conv2D)	(None, 126, 126, 32)	9,248
max_pooling2d (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_2 (Conv2D)	(None, 63, 63, 64)	18,496
conv2d_3 (Conv2D)	(None, 61, 61, 64)	36,928
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_4 (Conv2D)	(None, 30, 30, 128)	73,856
conv2d_5 (Conv2D)	(None, 28, 28, 128)	147,584
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 128)	0
conv2d_6 (Conv2D)	(None, 14, 14, 256)	295,168
conv2d_7 (Conv2D)	(None, 12, 12, 256)	590,080
max_pooling2d_3 (MaxPooling2D)	(None, 6, 6, 256)	0
conv2d_8 (Conv2D)	(None, 6, 6, 512)	1,180,160
conv2d_9 (Conv2D)	(None, 4, 4, 512)	2,359,808
max_pooling2d_4 (MaxPooling2D)	(None, 2, 2, 512)	0
dropout (Dropout)	(None, 2, 2, 512)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1500)	3,073,500
dropout_1 (Dropout)	(None, 1500)	0
dense_1 (Dense)	(None, 38)	57,038

Total params: 15,685,526 (59.84 MB)

Trainable params: 7,842,762 (29.92 MB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 7,842,764 (29.92 MB)

In [5]:
```
pip install opencv-python
```

```
Collecting opencv-pythonNote: you may need to restart the kernel to use updated packages.

  Using cached opencv_python-4.9.0.80-cp37-abi3-win_amd64.whl.metadata (20 kB)
Requirement already satisfied: numpy>=1.21.2 in c:\users\gupta\desktop\plant disease detection\tfenv\lib\site-packages (from opencv-python)
(1.26.4)
Using cached opencv_python-4.9.0.80-cp37-abi3-win_amd64.whl (38.6 MB)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.9.0.80
```

Visualizing Single Image of Test set

In [44]:
```
import cv2
#print(cv2.__version__)
image_path= "test/test/CornCommonRust1.JPG"
#Reading Image
img=cv2.imread(image_path)
img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB) #to convert BGR to RGB
```

```
#Displaying Image
plt.imshow(img)
plt.title("Test Image")
plt.xticks([])
plt.yticks([])
plt.show()
```

Testing Model

In [45]:
```
image= tf.keras.preprocessing.image.load_img(image_path,target_size=(128,128))
input_arr = tf.keras.preprocessing.image.img_to_array(image)
input_arr=np.array([input_arr]) # Convert single image to a Batch
print(input_arr.shape)
```

```
(1, 128, 128, 3)
```

In [46]:
```
prediction=model.predict(input_arr)
prediction,prediction.shape
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
```

Out[46]:
```
(array([[1.48842860e-09, 9.48902179e-10, 2.91154578e-07, 2.21236762e-11,
         2.67099107e-08, 1.43852941e-09, 4.50945420e-10, 6.95892843e-10,
         9.99884963e-01, 1.20070759e-10, 5.76019232e-10, 1.40683699e-12,
         7.96243835e-11, 2.55876462e-12, 1.13090190e-12, 4.11622923e-11,
         9.93791854e-11, 1.66432326e-10, 1.68012729e-10, 4.96581904e-08,
         1.21850052e-09, 1.10418317e-14, 1.64184482e-11, 2.96863992e-11,
         4.34376249e-13, 2.56715593e-10, 2.24094618e-10, 1.73667179e-12,
         1.40791137e-11, 3.97906463e-09, 1.14585986e-04, 4.16578494e-10,
         2.65940470e-09, 1.24699589e-14, 5.77823622e-11, 1.32213056e-12,
         1.57270707e-11, 1.39509586e-08]], dtype=float32),
 (1, 38))
```

In [47]:
```
result_index=np.argmax(prediction)
result_index
```

Out[47]: 8

In [48]:
```
class_name=['Apple___Apple_scab',
 'Apple___Black_rot',
 'Apple___Cedar_apple_rust',
 'Apple___healthy',
 'Blueberry___healthy',
 'Cherry_(including_sour)___Powdery_mildew',
 'Cherry_(including_sour)___healthy',
 'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot',
 'Corn_(maize)___Common_rust_',
 'Corn_(maize)___Northern_Leaf_Blight',
 'Corn_(maize)___healthy',
 'Grape___Black_rot',
 'Grape___Esca_(Black_Measles)',
 'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)',
 'Grape___healthy',
 'Orange___Haunglongbing_(Citrus_greening)',
 'Peach___Bacterial_spot',
 'Peach___healthy',
 'Pepper,_bell___Bacterial_spot',
 'Pepper,_bell___healthy',
 'Potato___Early_blight',
 'Potato___Late_blight',
 'Potato___healthy',
 'Raspberry___healthy',
 'Soybean___healthy',
 'Squash___Powdery_mildew',
 'Strawberry___Leaf_scorch',
 'Strawberry___healthy',
 'Tomato___Bacterial_spot',
```

```
 'Tomato___Early_blight',
 'Tomato___Late_blight',
 'Tomato___Leaf_Mold',
 'Tomato___Septoria_leaf_spot',
 'Tomato___Spider_mites Two-spotted_spider_mite',
 'Tomato___Target_Spot',
 'Tomato___Tomato_Yellow_Leaf_Curl_Virus',
 'Tomato___Tomato_mosaic_virus',
 'Tomato___healthy']
```

In [49]:
```
#Displaying Result of disease prediction
model_prediction=class_name[result_index]
plt.imshow(img)
plt.title(f"Disease Name: {model_prediction}")
plt.xticks([])
plt.yticks([])
plt.show()
```

In [50]:
```
model_prediction
```

Out[50]:
```
'Corn_(maize)___Common_rust_'
```

In [ ]:

In [ ]:

In [ ]: