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

loading model

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

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)      9248

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 conv2d_2 (Conv2D)           (None, 63, 63, 64)        18496

 conv2d_3 (Conv2D)           (None, 61, 61, 64)        36928

 max_pooling2d_1 (MaxPooling  (None, 30, 30, 64)       0
 2D)

 conv2d_4 (Conv2D)           (None, 30, 30, 128)       73856

 conv2d_5 (Conv2D)           (None, 28, 28, 128)       147584

 max_pooling2d_2 (MaxPooling  (None, 14, 14, 128)      0
 2D)

 conv2d_6 (Conv2D)           (None, 14, 14, 256)       295168

 conv2d_7 (Conv2D)           (None, 12, 12, 256)       590080

 max_pooling2d_3 (MaxPooling  (None, 6, 6, 256)        0
 2D)

 conv2d_8 (Conv2D)           (None, 6, 6, 512)         1180160

 conv2d_9 (Conv2D)           (None, 4, 4, 512)         2359808

 max_pooling2d_4 (MaxPooling  (None, 2, 2, 512)        0
 2D)

 dropout (Dropout)           (None, 2, 2, 512)         0

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 1500)              3073500

 dropout_1 (Dropout)         (None, 1500)              0

 dense_1 (Dense)             (None, 38)                57038

=================================================================
Total params: 7,842,762
Trainable params: 7,842,762
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:
```
```

Visualising images of test set

In [4]:
```
import cv2
image_path = "test/test/CornCommonRust2.JPG"
#read img
img = cv2.imread(image_path)
#convert bgr to rgb
img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)

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

Testing Model

In [5]:
```
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 [6]:
```
prediction = model.predict(input_arr)
prediction,prediction.shape
```

```
1/1 [==============================] - 15s 15s/step
```

Out[6]:
```
(array([[8.7829467e-18, 2.8660604e-19, 4.7152817e-21, 6.3119323e-22,
         3.2426425e-21, 2.2499923e-18, 9.6758780e-19, 1.9149978e-13,
         1.0000000e+00, 2.2833669e-19, 1.2302450e-17, 8.4352367e-26,
         4.7415217e-23, 8.9849995e-23, 1.2010913e-24, 1.8906961e-20,
         2.2671302e-17, 4.1544132e-23, 3.9089671e-16, 2.5080584e-17,
         1.2075649e-12, 2.9245322e-24, 1.2035763e-18, 2.6732390e-24,
         7.2733559e-24, 3.4162246e-20, 2.8513275e-19, 3.0768677e-24,
         7.7857995e-26, 7.8720473e-19, 2.9442566e-16, 1.0691177e-22,
         2.3213051e-21, 4.6419847e-28, 7.0633841e-25, 2.0332089e-27,
         2.3638059e-27, 5.2839713e-20]], dtype=float32),
 (1, 38))
```

In [7]:
```
result_index=np.argmax(prediction)
result_index
```

Out[7]:
```
8
```

In [8]:
```
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 [9]:
```
#Displaying Result Of diseas predaction
model_prediction = class_name[result_index]
plt.imshow(img)
plt.title(f"Disease Name: {model_prediction}")
plt.xticks([])
plt.yticks([])
plt.show()
```

In [10]:
```
model_prediction
```

Out[10]: 'Corn_(maize)___Common_rust_'

In [ ]: