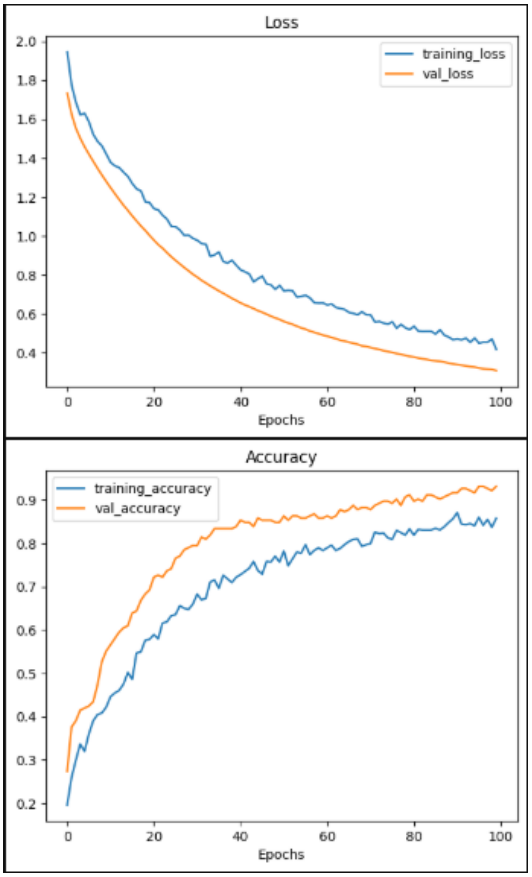
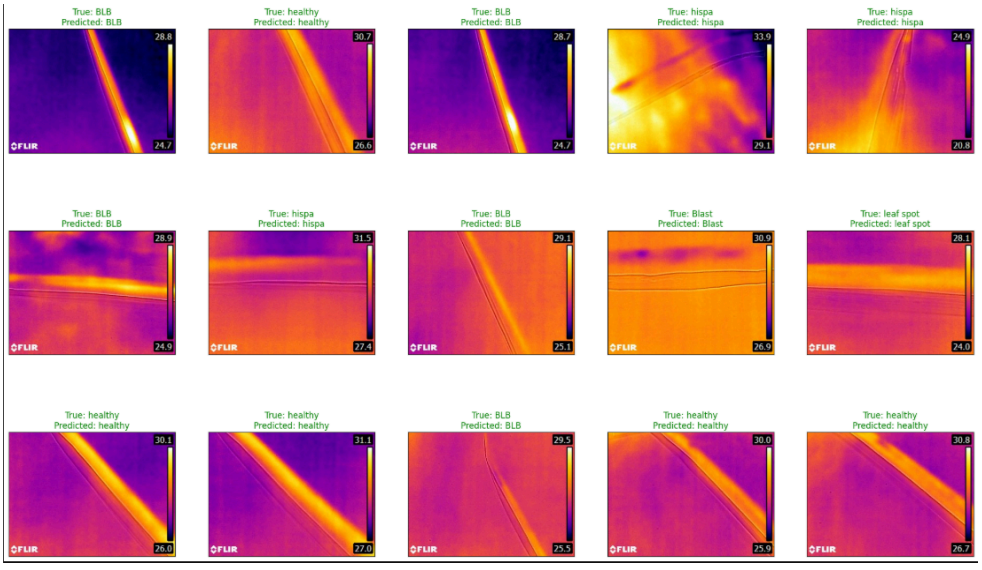
MobileNetv3

Epoch 100
Learning Rate - 0.00001
Optimizer - Adam

```
    Test Loss: 0.40873
Test Accuracy: 84.88%
```



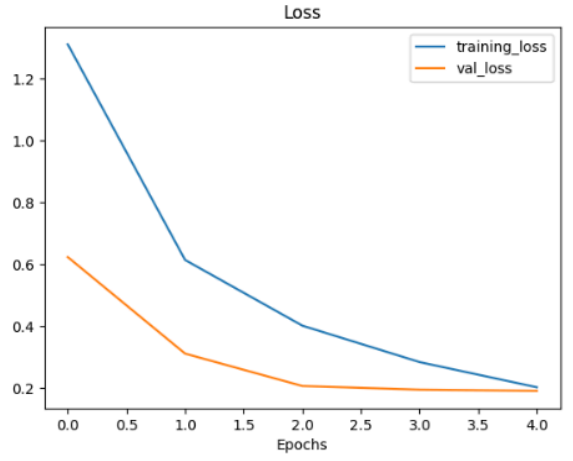


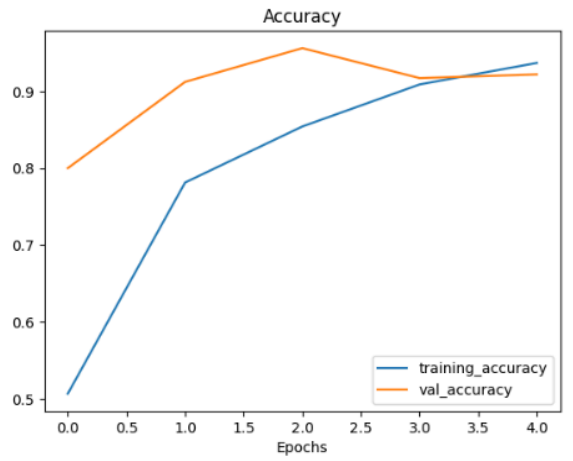
Epoch 5 **Learning Rate** - 0.001 **Optimizer** - Adam

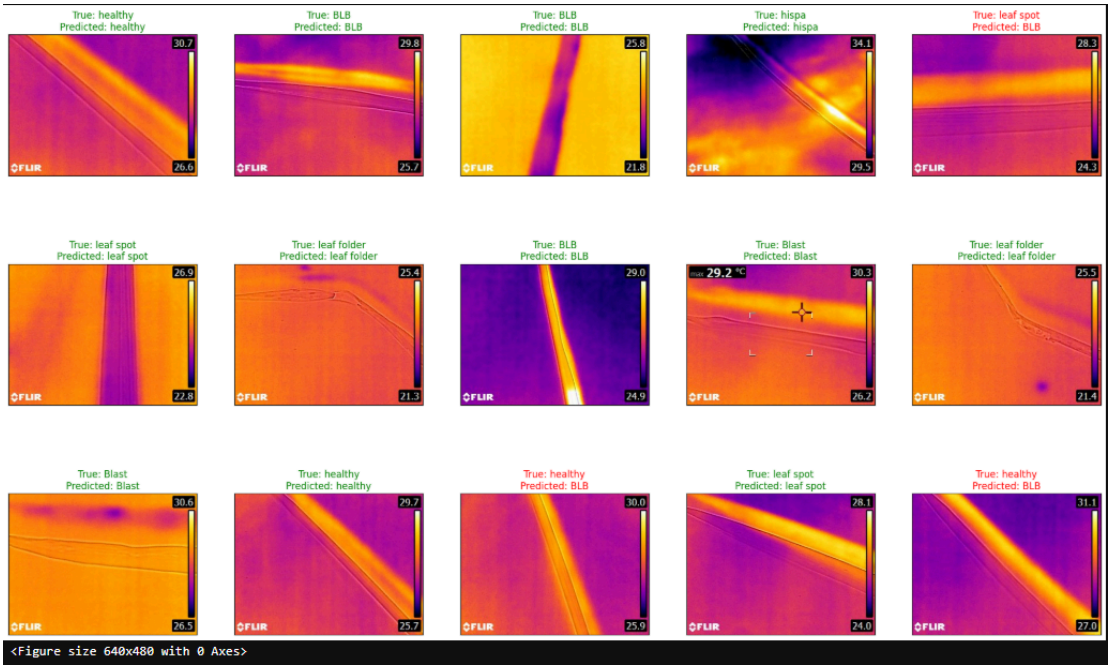
```
Epoch 5: val_accuracy did not improve from 0.95610
26/26 ━━━━━━━━━━━━━━━━━━━━ 29s 1s/step - accuracy: 0.9378 - loss: 0.2032 - val_accuracy: 0.9220 - val_loss: 0.1906
```

```
[23]: results = model.evaluate(test_images, verbose=0)

      print("    Test Loss: {:.5f}".format(results[0]))
      print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.26634
Test Accuracy: 89.15%
```







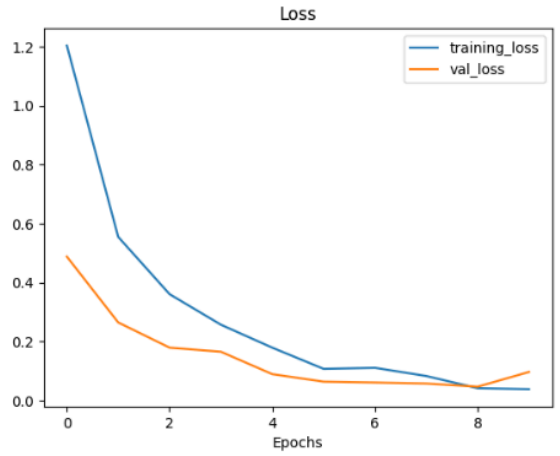

```
<Figure size 640x480 with 0 Axes>
```

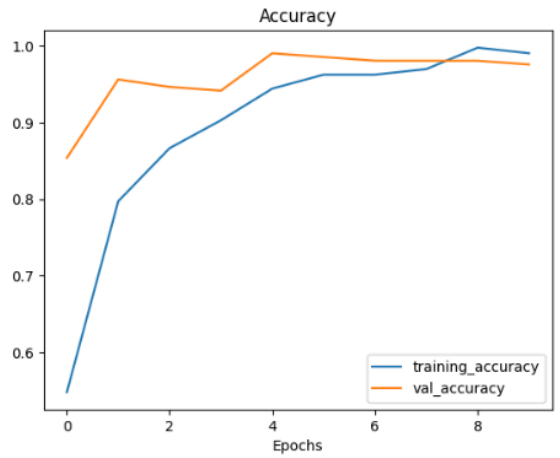
Epoch 10 **Learning Rate** - 0.001 **Optimizer** - Adam

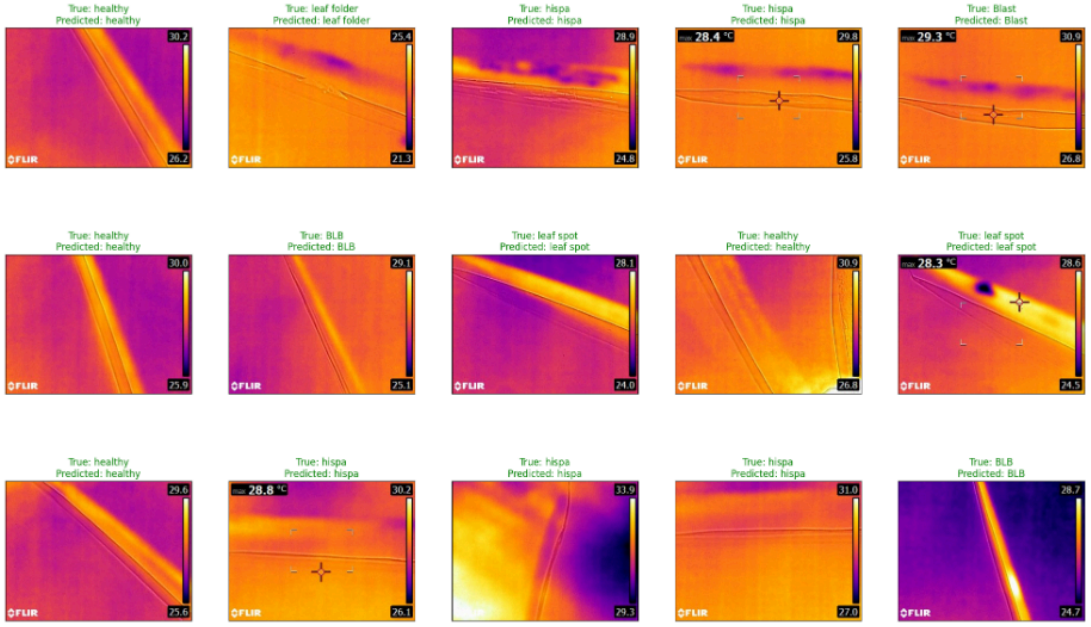
```
Epoch 10: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 27s 1s/step - accuracy: 0.9945 - loss: 0.0317 - val_accuracy: 0.9756 - val_loss: 0.0973
```

```
[42]: results = model.evaluate(test_images, verbose=0)

      print("    Test Loss: {:.5f}".format(results[0]))
      print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.04569
Test Accuracy: 98.06%
```





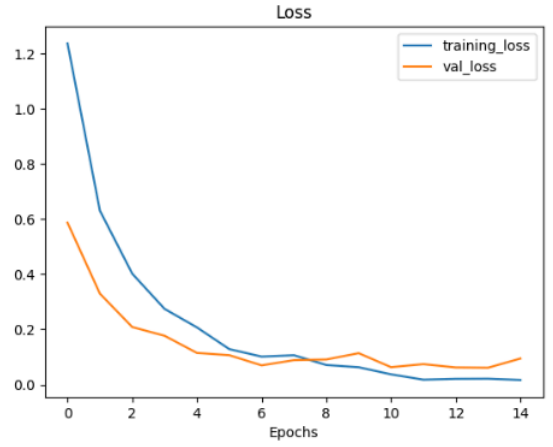


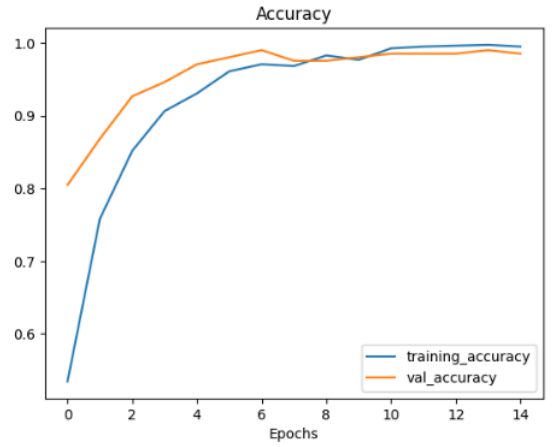
Epoch 15 **Learning Rate** - 0.001 **Optimizer** - Adam

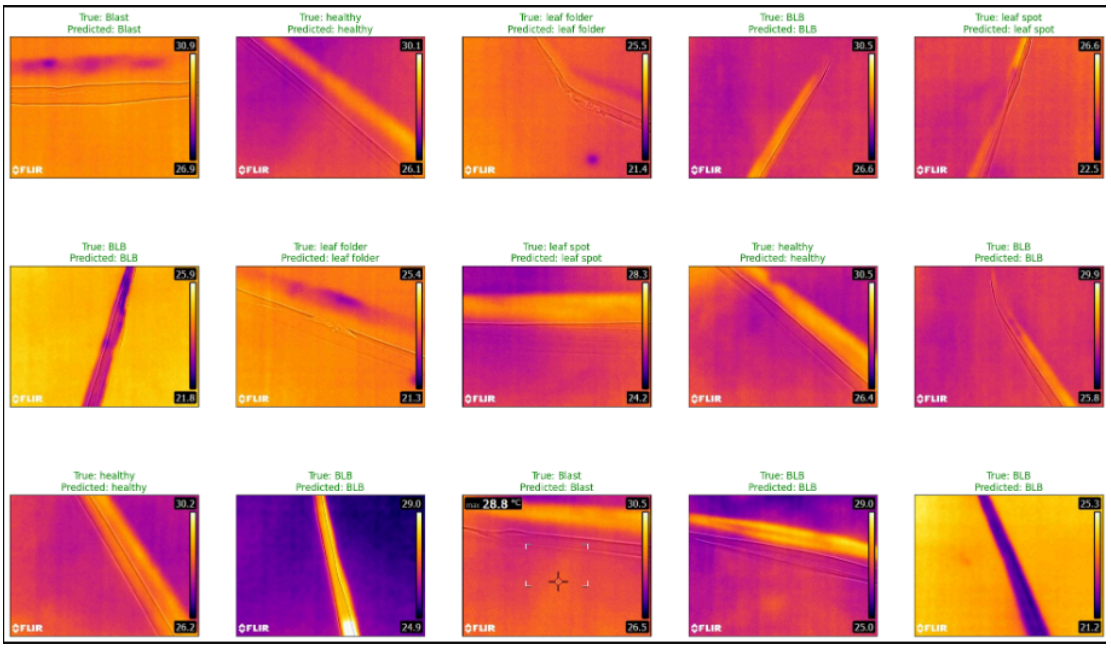
```
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 891ms/step - accuracy: 0.9919 - loss: 0.0213
Epoch 15: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 29s 1s/step - accuracy: 0.9920 - loss: 0.0212 - val_accuracy: 0.9854 - val_loss: 0.0948
```

[47]:
```
results = model.evaluate(test_images, verbose=0)

print("    Test Loss: {:.5f}".format(results[0]))
print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.02021
Test Accuracy: 99.22%
```





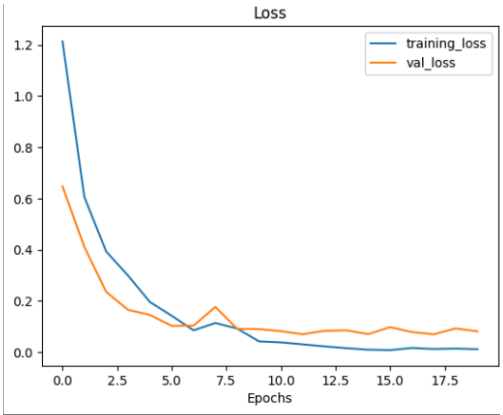


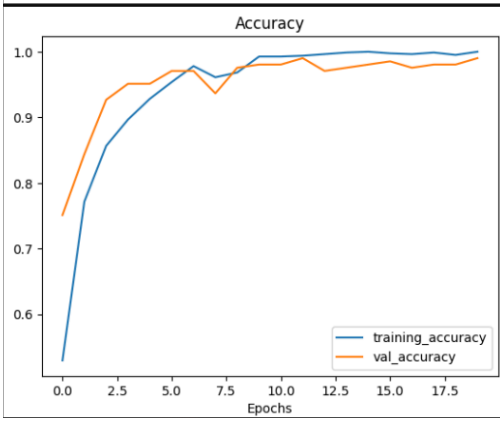
Epoch 20 **Learning Rate** - 0.001 **Optimizer** - Adam

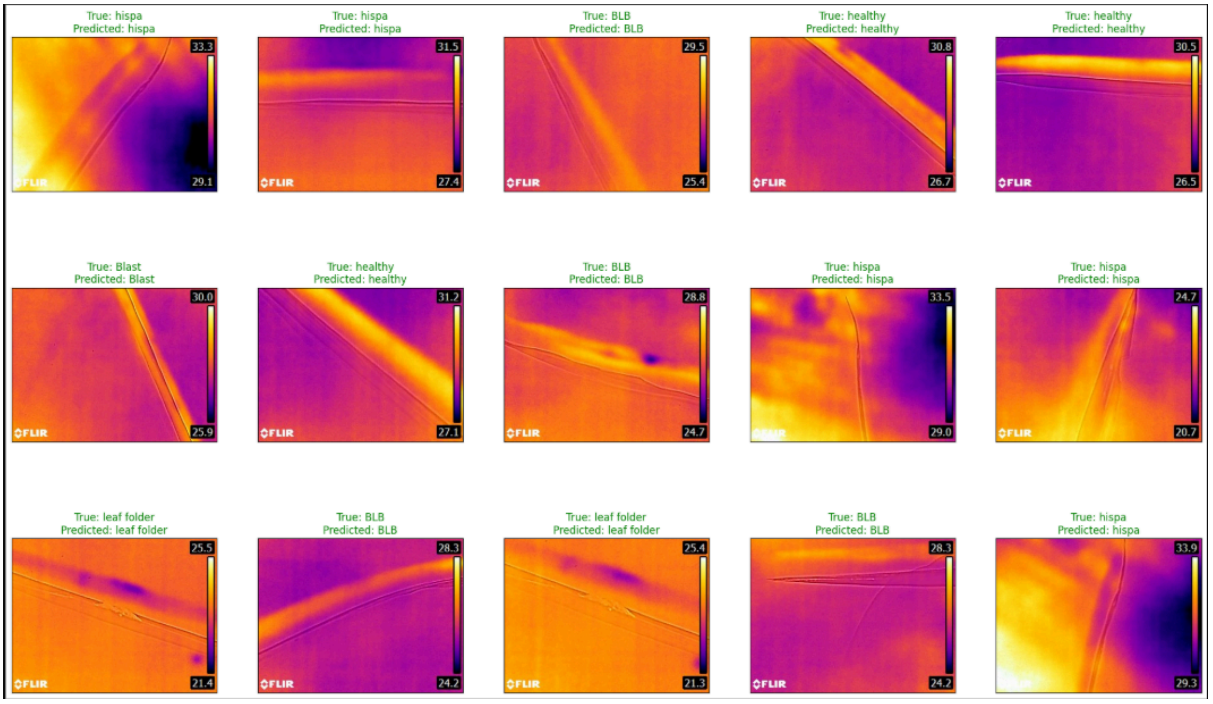
```
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 951ms/step - accuracy: 1.0000 - loss: 0.0129
Epoch 20: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 31s 1s/step - accuracy: 1.0000 - loss: 0.0128 - val_accuracy: 0.9902 - val_loss: 0.0812
```

[52]:
```
results = model.evaluate(test_images, verbose=0)

print("    Test Loss: {:.5f}".format(results[0]))
print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.01367
Test Accuracy: 100.00%
```





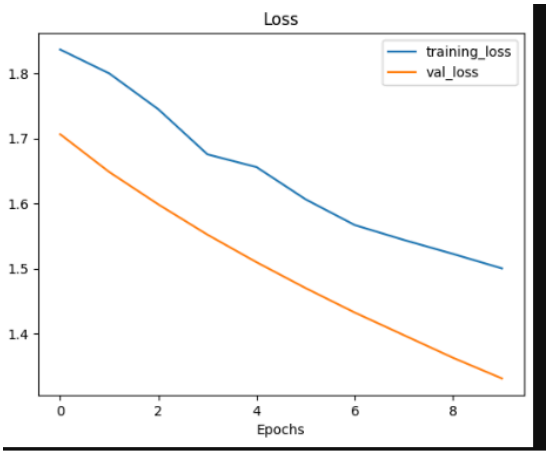


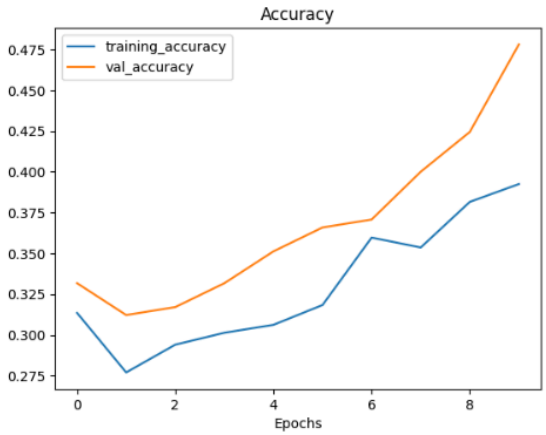
Epoch 10 **Learning Rate** - 0.00001 **Optimizer** - Adam

```
Epoch 10: val_accuracy improved from 0.42439 to 0.47805, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
26/26 ━━━━━━━━━━━━━━━━━━━━ 34s 1s/step - accuracy: 0.3845 - loss: 1.4915 - val_accuracy: 0.4780 - val_loss: 1.3313
```

```
[23]: results = model.evaluate(test_images, verbose=0)

      print("    Test Loss: {:.5f}".format(results[0]))
      print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 1.32523
Test Accuracy: 51.94%
```



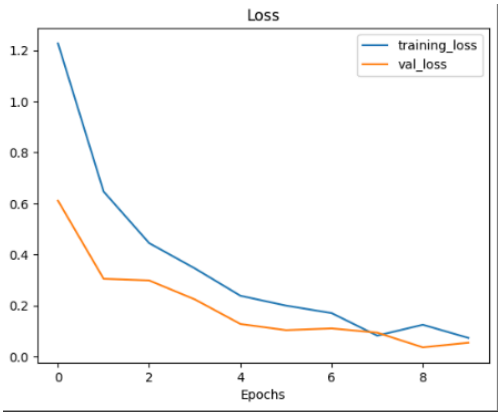


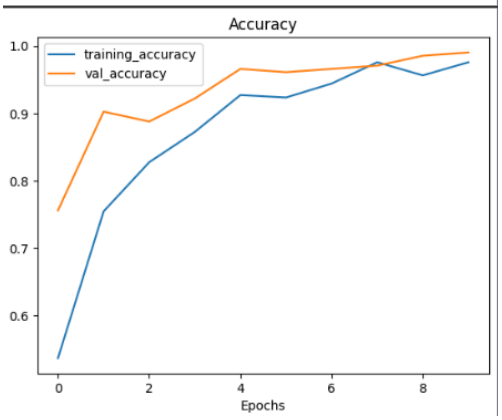
Epoch 10 **Learning Rate** - 0.001 **Optimizer** - RMSprop

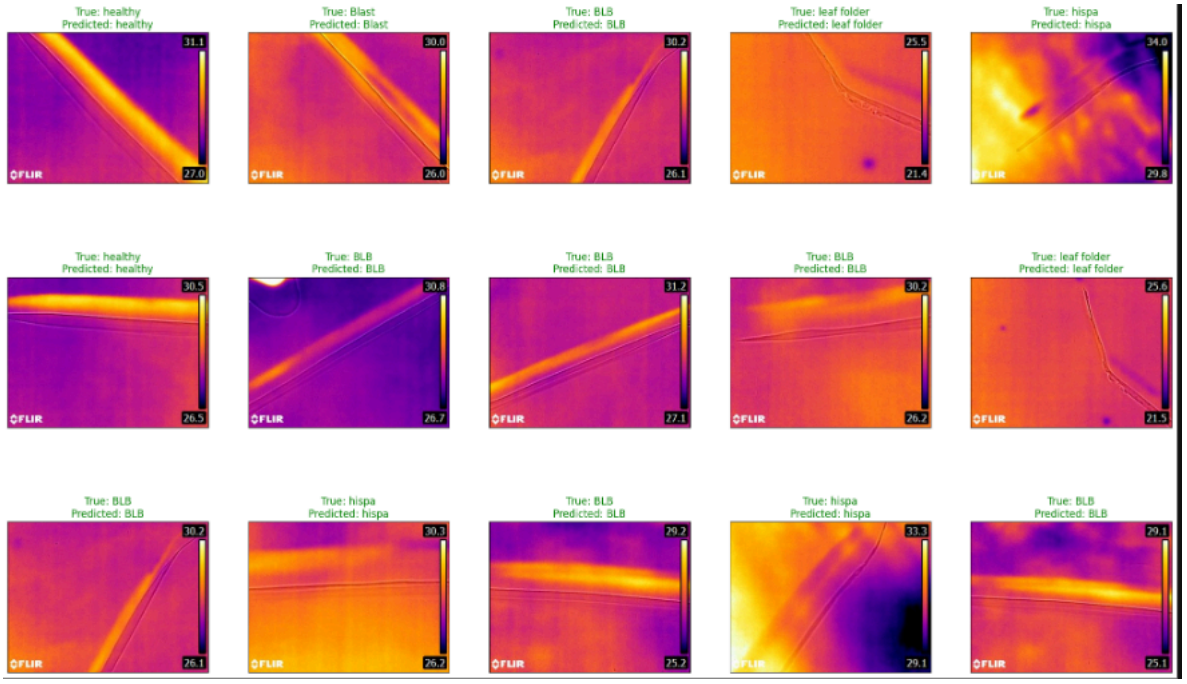
```
Epoch 10: val_accuracy improved from 0.98537 to 0.99024, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
26/26 ━━━━━━━━━━━━━━━━━━━━ 36s 1s/step - accuracy: 0.9756 - loss: 0.0751 - val_accuracy: 0.9902 - val_loss: 0.0541
```

[32]:
```
results = model.evaluate(test_images, verbose=0)

print("    Test Loss: {:.5f}".format(results[0]))
print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.05137
Test Accuracy: 98.06%
```





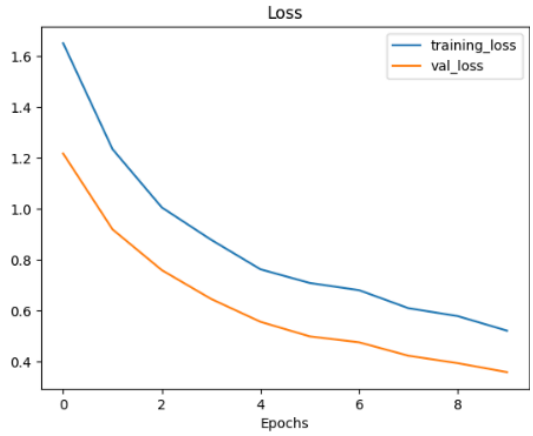


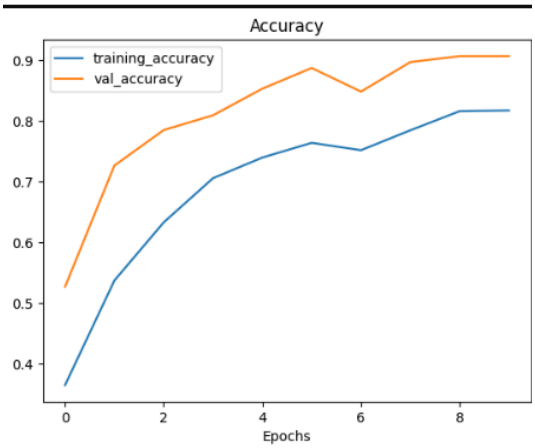
Epoch 10 **Learning Rate** - 0.001 **Optimizer** - SGD

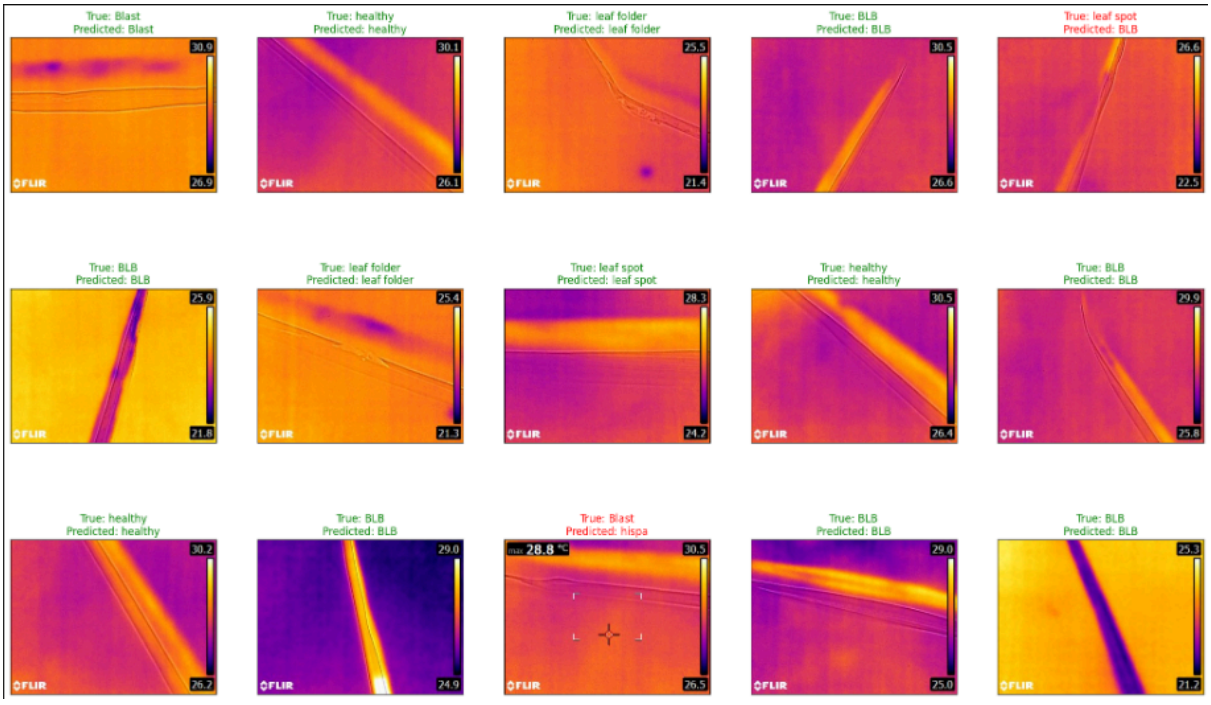
```
Epoch 10: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 85s 3s/step - accuracy: 0.8055 - loss: 0.5543 - val_accuracy: 0.9073 - val_loss: 0.3572
```

```
[38]: results = model.evaluate(test_images, verbose=0)

print("    Test Loss: {:.5f}".format(results[0]))
print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.46146
Test Accuracy: 82.95%
```







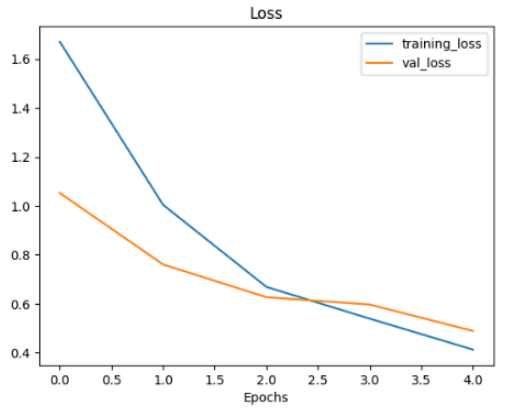
RESNET 50

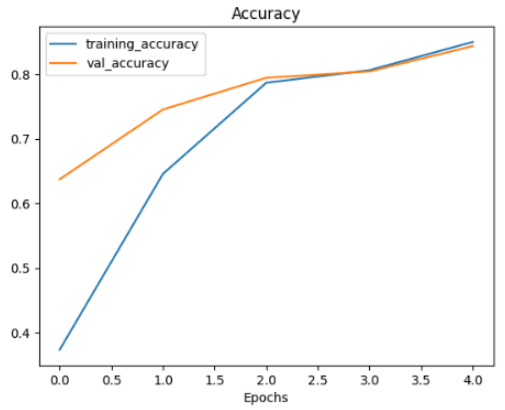
Epoch 5 Learning Rate - 0.001 Optimizer - Adam

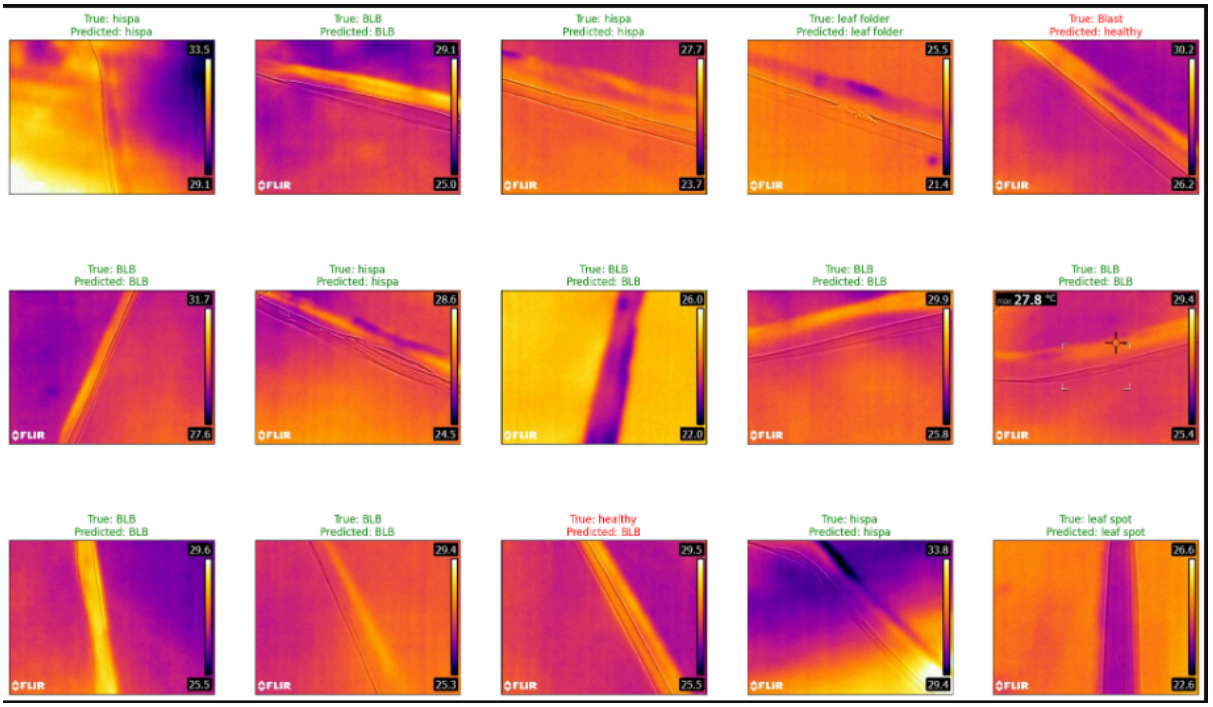
```
Epoch 5: val_accuracy improved from 0.80392 to 0.84314, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
13/13 ━━━━━━━━━━━━━━━━━━━━ 81s 6s/step - accuracy: 0.8419 - loss: 0.4452 - val_accuracy: 0.8431 - val_loss: 0.4889
```

```
[23]: results = model.evaluate(test_images, verbose=0)

      print("    Test Loss: {:.5f}".format(results[0]))
      print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.51547
Test Accuracy: 84.50%
```



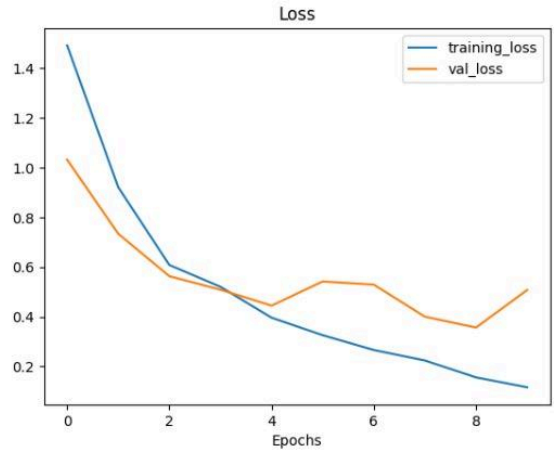


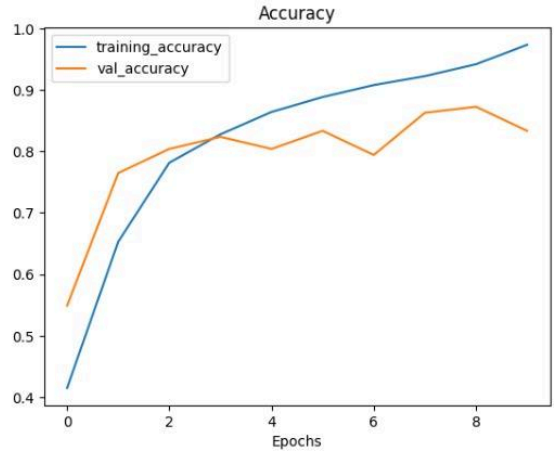


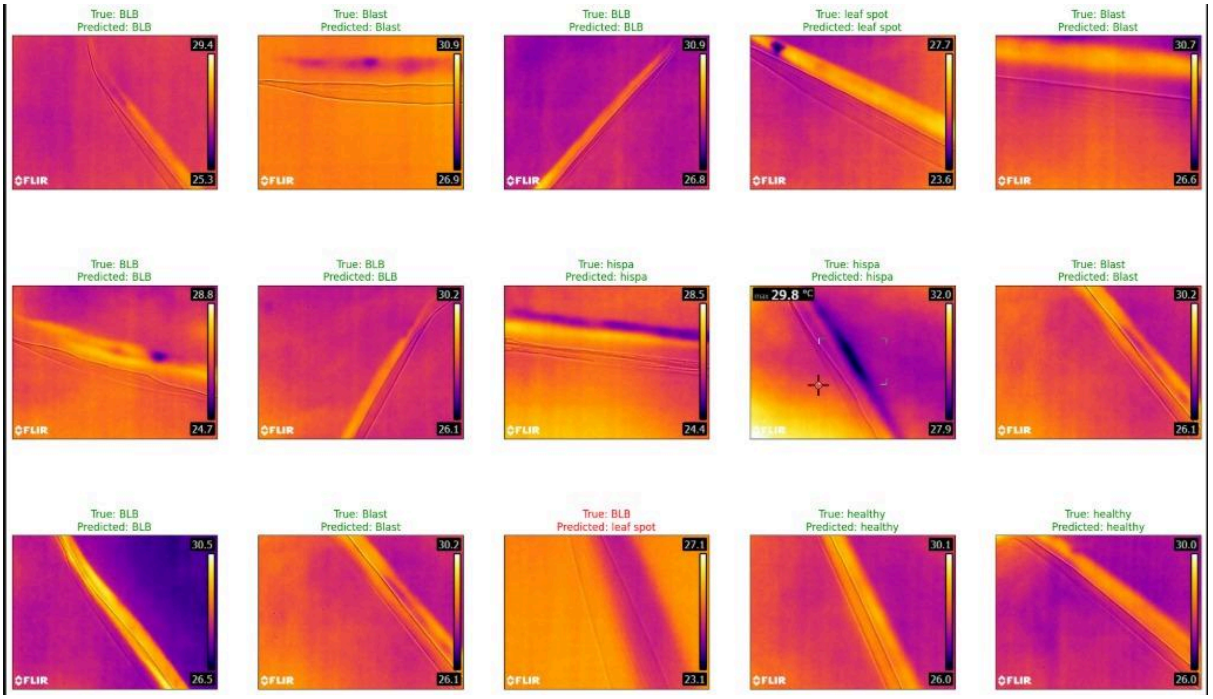
Epoch 10 **Learning Rate** - 0.001 **Optimizer** - Adam

```
[49]: results = model.evaluate(test_images, verbose=0)

      print("    Test Loss: {:.5f}".format(results[0]))
      print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.41489
Test Accuracy: 86.05%
```





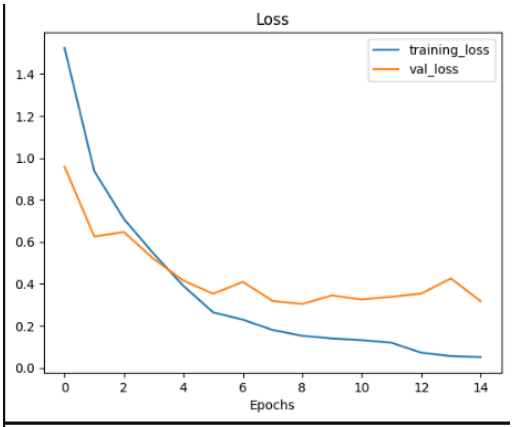


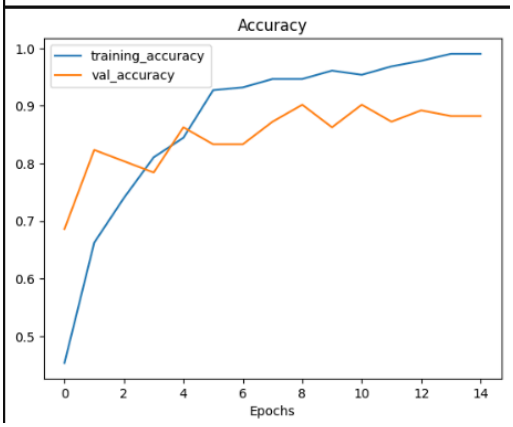
Epoch 15 **Learning Rate** - 0.001 **Optimizer** - Adam

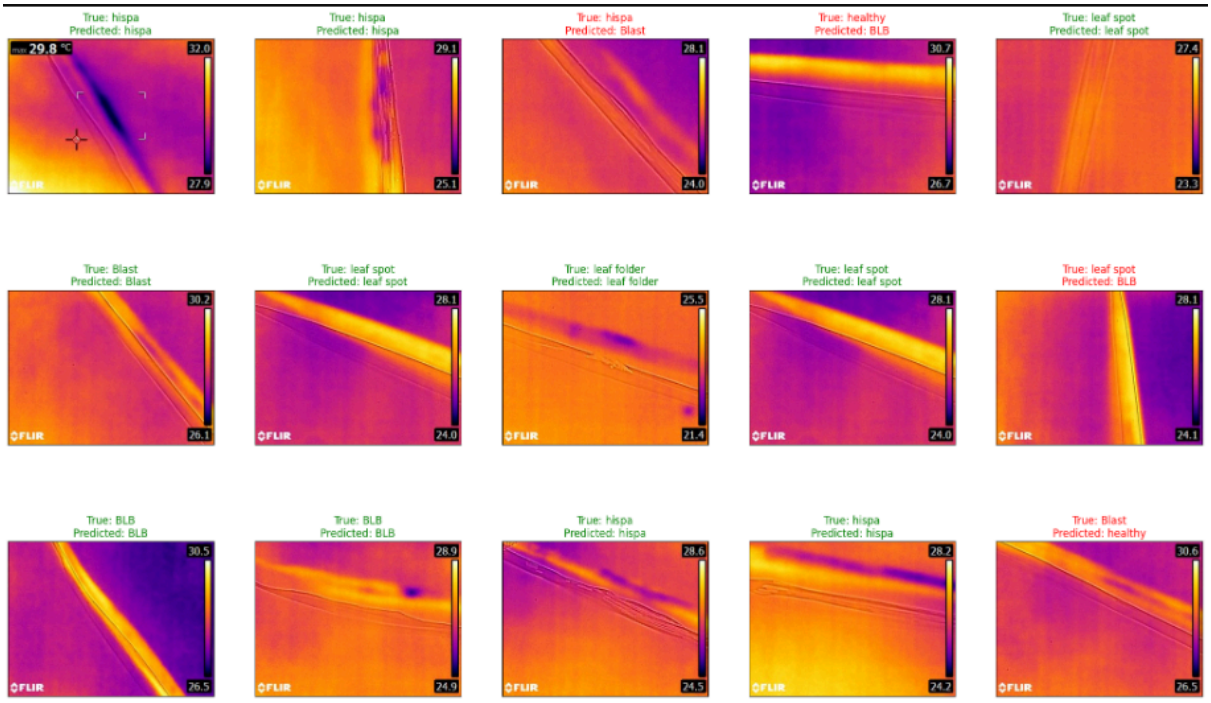
```
Epoch 15: val_accuracy did not improve from 0.90196
13/13 ━━━━━━━━━━━━━━━━━━━━ 100s 8s/step - accuracy: 0.9886 - loss: 0.0572 - val_accuracy: 0.8824 - val_loss: 0.3168
```

```
[49]: results = model.evaluate(test_images, verbose=0)

      print("    Test Loss: {:.5f}".format(results[0]))
      print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.40090
Test Accuracy: 88.37%
```

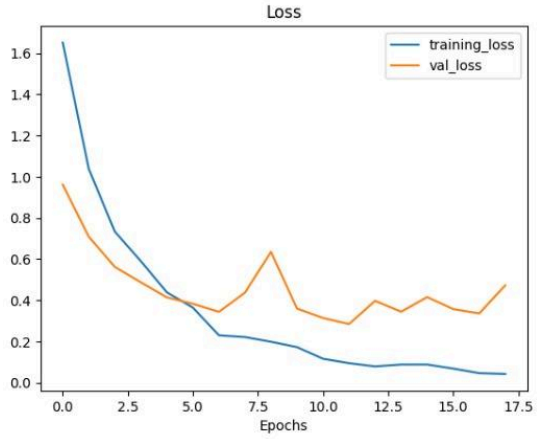


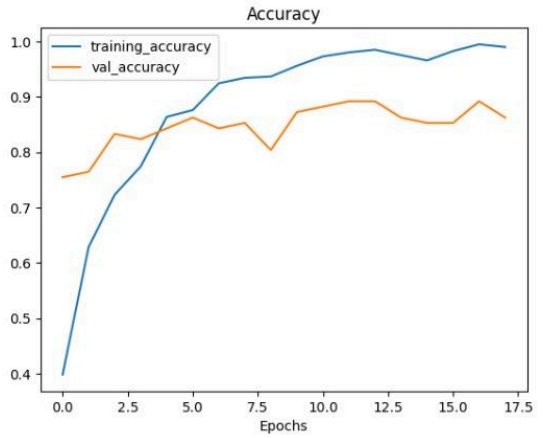


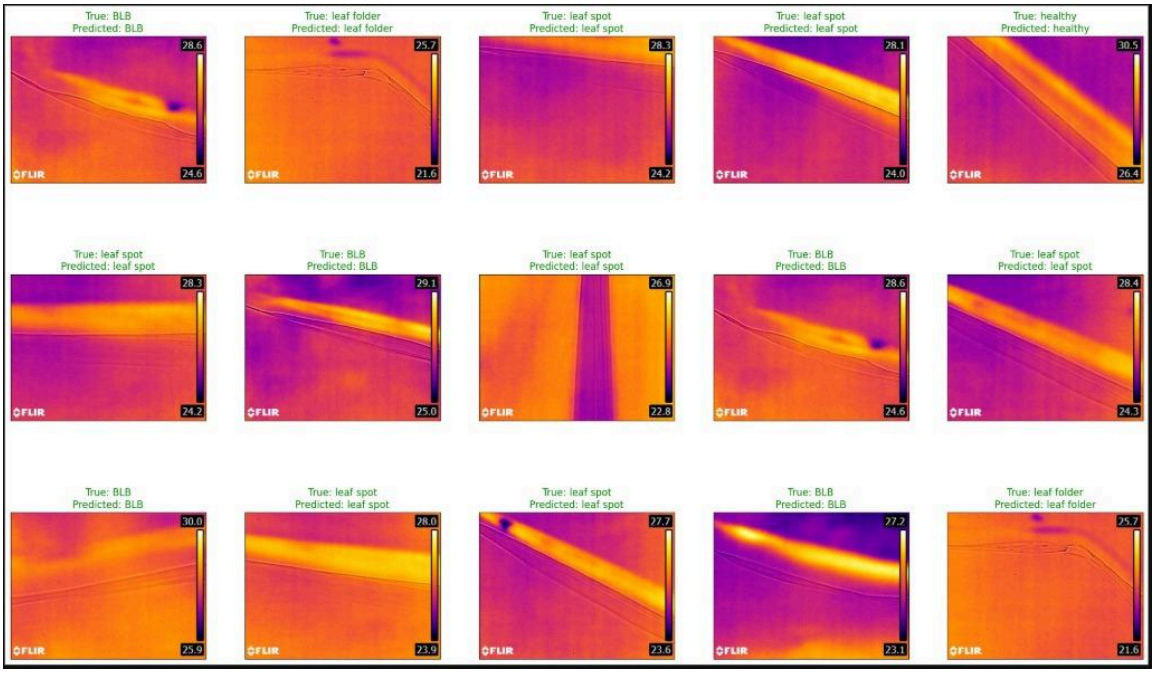


Epoch 20 **Learning Rate** - 0.001 **Optimizer** - Adam

```
Epoch 18: val_accuracy did not improve from 0.89216
13/13 ━━━━━━━━━━━━━━━━━━━━ 182s 14s/step - accuracy: 0.9958 - loss: 0.0396 - val_accuracy: 0.8627 - val_loss: 0.4723

[43]: results = model.evaluate(test_images, verbose=0)

      print("    Test Loss: {:.5f}".format(results[0]))
      print("Test Accuracy: {:.2f}%".format(results[1] * 100))

    Test Loss: 0.39060
Test Accuracy: 88.37%
```

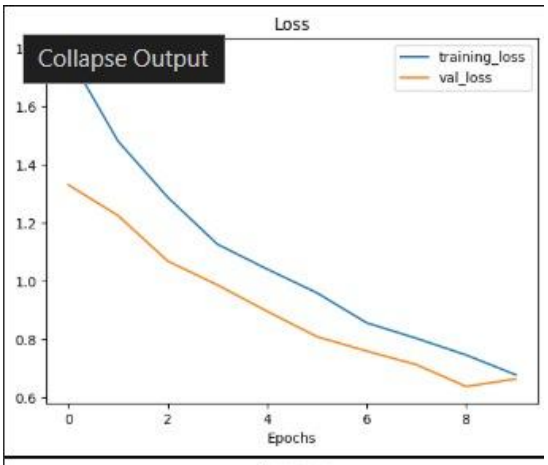


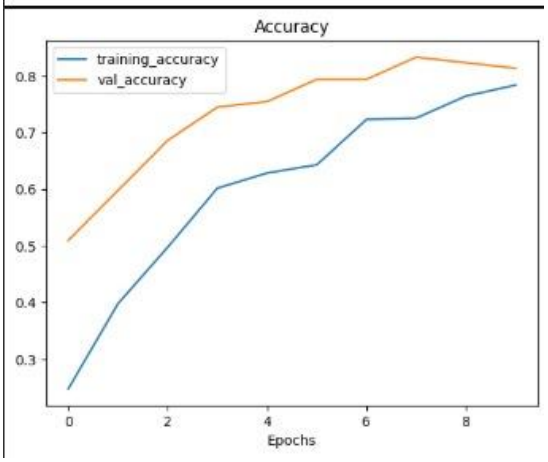


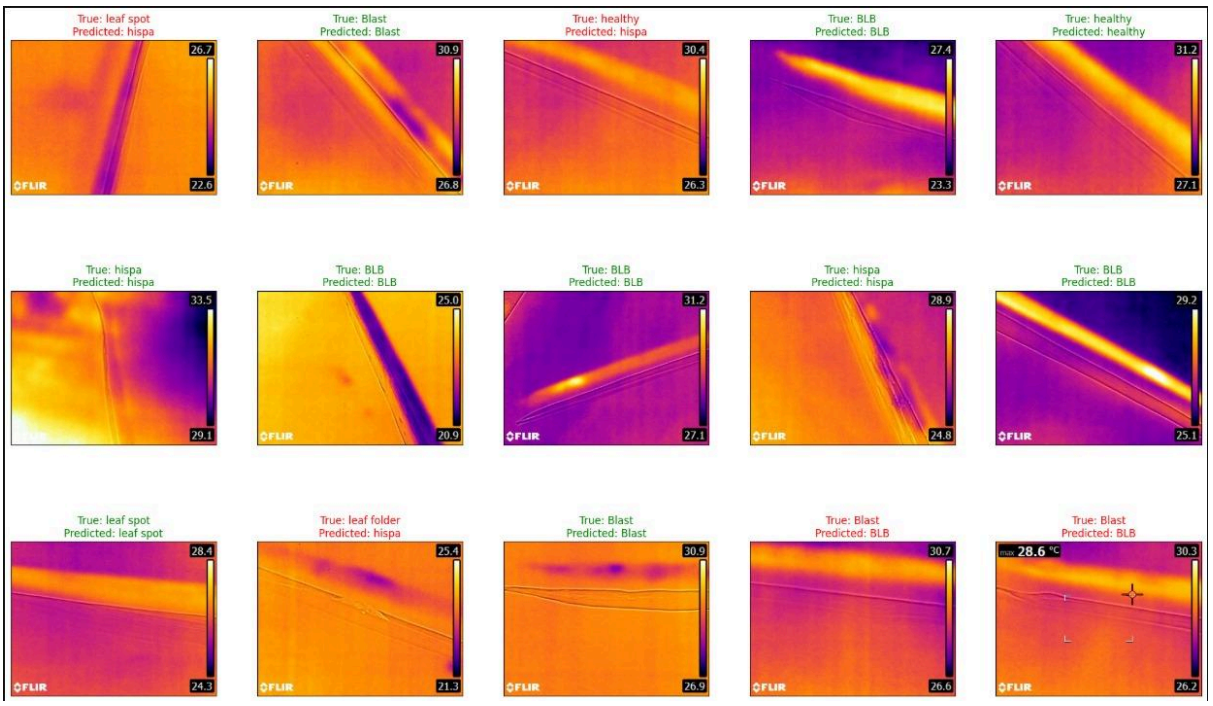


Epoch 10 **Learning Rate** - 0.0001 **Optimizer** - Adam

```
Epoch 10: val_accuracy did not improve from 0.83333
13/13 ━━━━━━━━━━━━━━━━━━━━ 90s 7s/step - accuracy: 0.7812 - loss: 0.6784 - val_accuracy: 0.8137 - val_loss: 0.6634

[23]: results = model.evaluate(test_images, verbose=0)

      print("    Test Loss: {:.5f}".format(results[0]))
      print("Test Accuracy: {:.2f}%".format(results[1] * 100))

    Test Loss: 0.80889
Test Accuracy: 75.97%
```

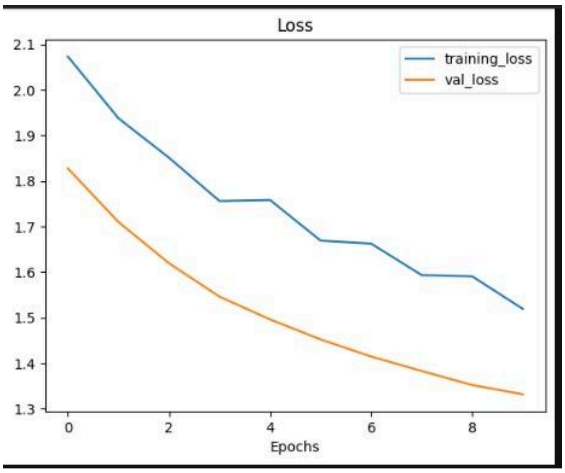


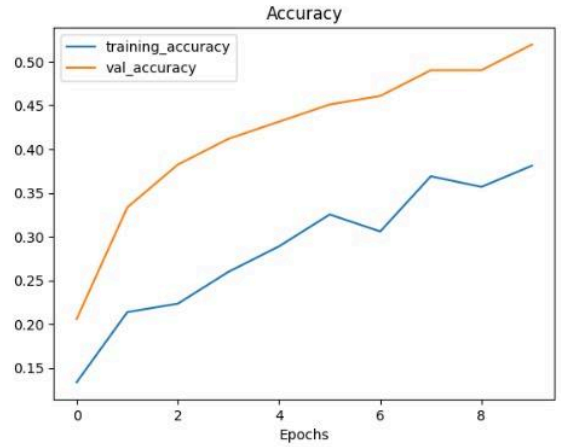


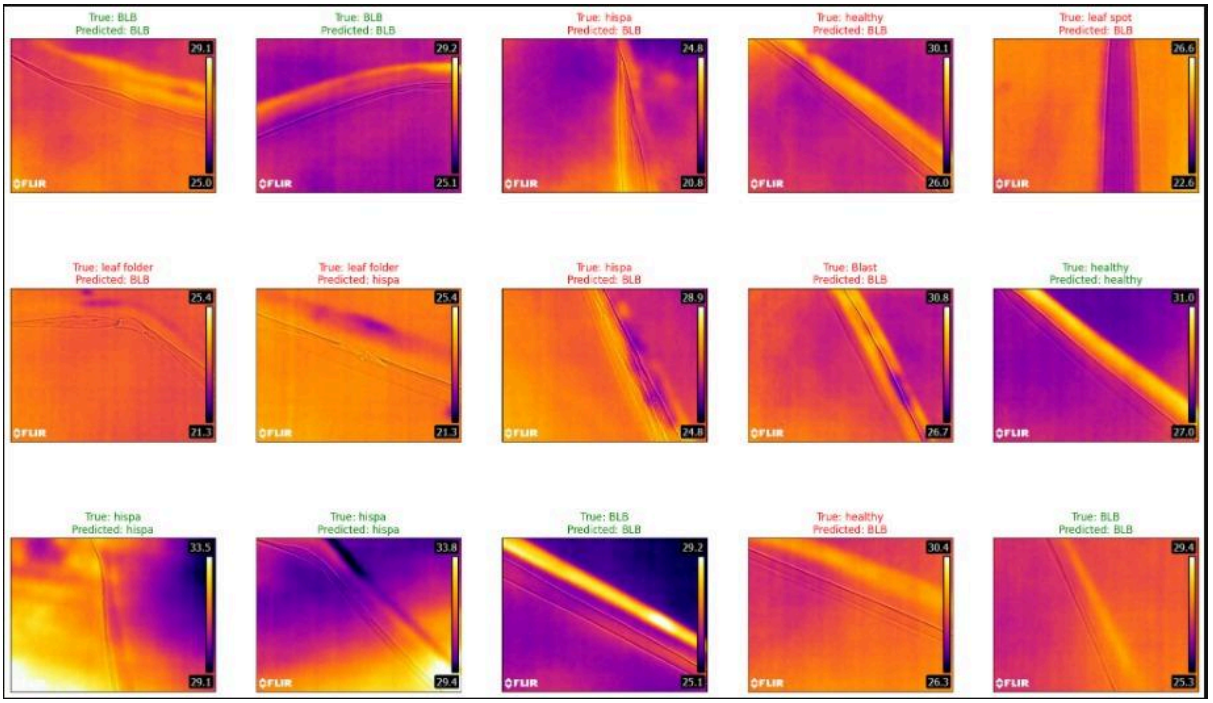


Epoch 10 **Learning Rate** - 0.00001 **Optimizer** - Adam

```
Epoch 10: val_accuracy improved from 0.49020 to 0.51961, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
13/13 ━━━━━━━━━━━━━━━━━━━━ 80s 6s/step - accuracy: 0.3653 - loss: 1.5303 - val_accuracy: 0.5196 - val_loss: 1.3314

[52]: results = model.evaluate(test_images, verbose=0)

      print("    Test Loss: {:.5f}".format(results[0]))
      print("Test Accuracy: {:.2f}%".format(results[1] * 100))

    Test Loss: 1.49933
Test Accuracy: 42.64%
```





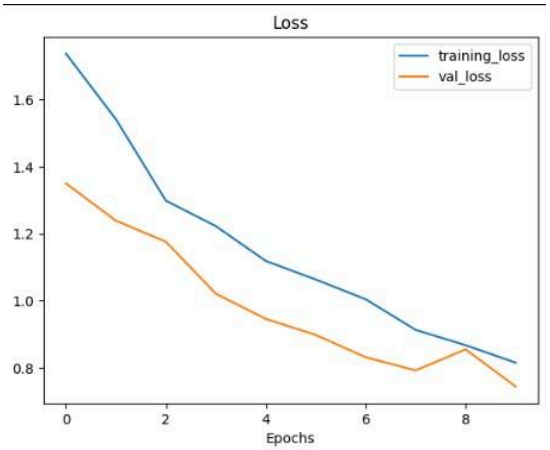


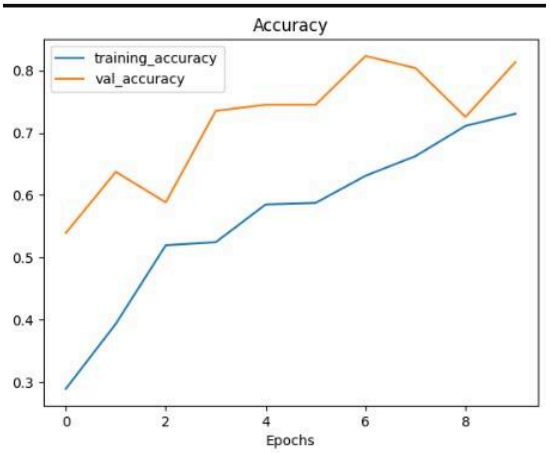
Epoch 10 **Learning Rate** - 0.001 **Optimizer** - SGD

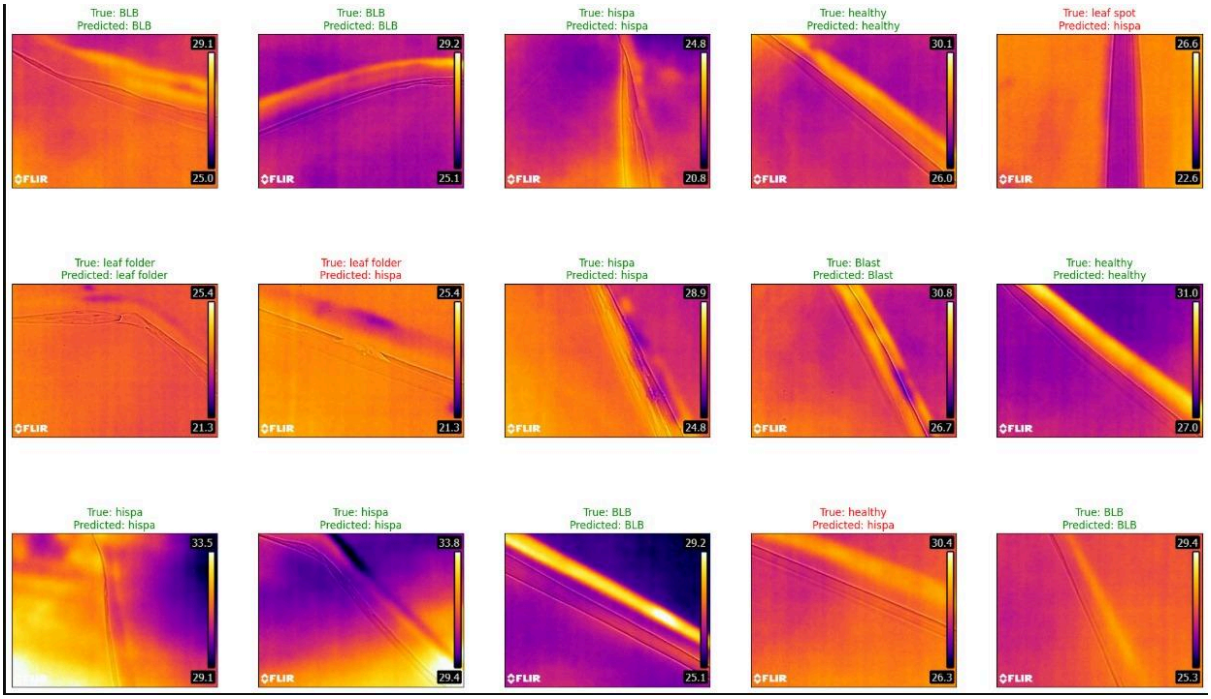
```
Epoch 10: val_accuracy did not improve from 0.82353
13/13 ━━━━━━━━━━━━━━━━━━━━ 81s 6s/step - accuracy: 0.7213 - loss: 0.8078 - val_accuracy: 0.8137 - val_loss: 0.7437
```

```
[79]: results = model.evaluate(test_images, verbose=0)

print("    Test Loss: {:.5f}".format(results[0]))
print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.88047
Test Accuracy: 76.74%
```

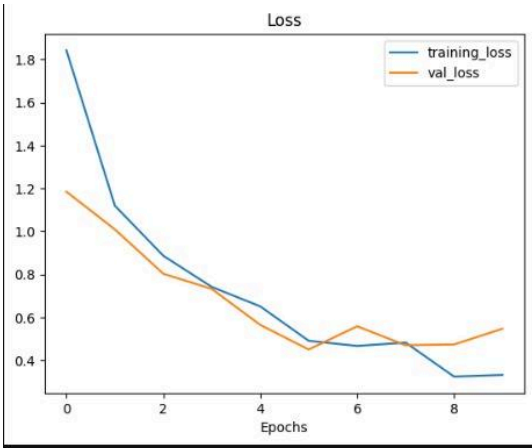


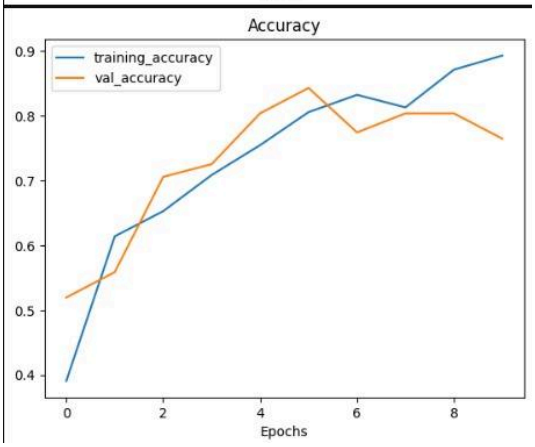


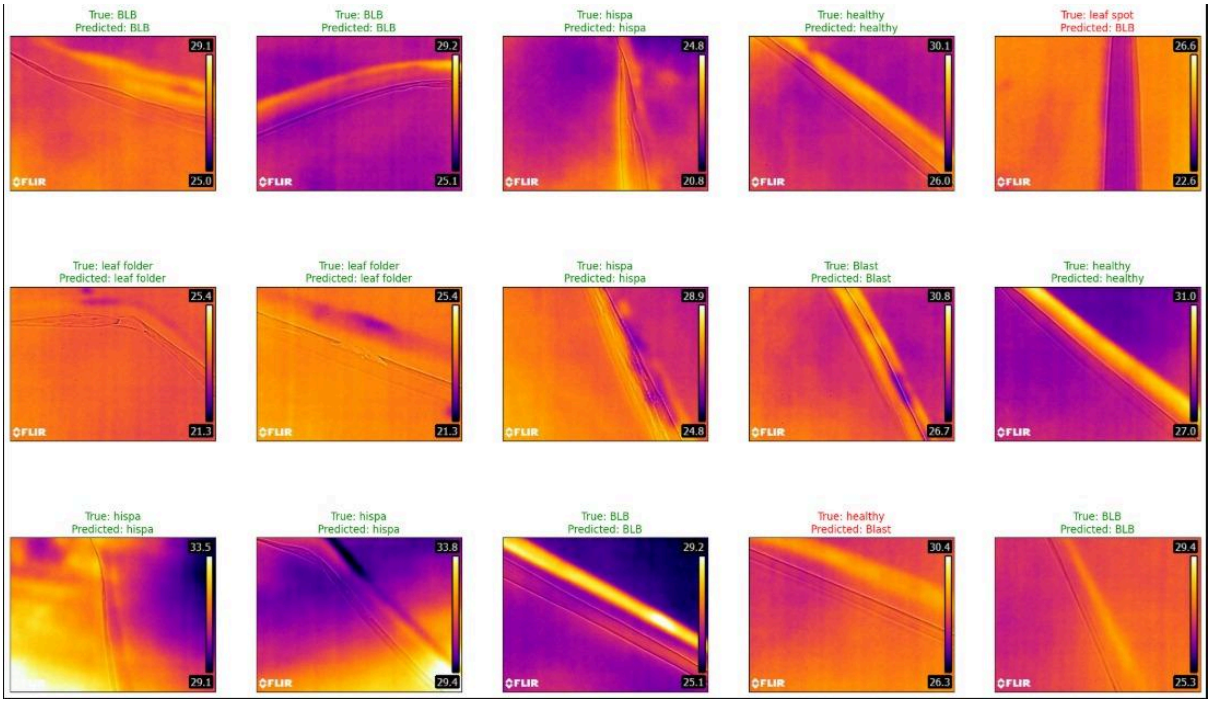


Epoch 10 **Learning Rate** - 0.001 **Optimizer** - RMSprop

```
Epoch 10: val_accuracy did not improve from 0.84314
13/13 ━━━━━━━━━━━━━━━━━━━━ 49s 4s/step - accuracy: 0.9080 - loss: 0.2876 - val_accuracy: 0.7647 - val_loss: 0.5467

[106]: results = model.evaluate(test_images, verbose=0)

       print("    Test Loss: {:.5f}".format(results[0]))
       print("Test Accuracy: {:.2f}%".format(results[1] * 100))

    Test Loss: 0.54817
Test Accuracy: 82.95%
```



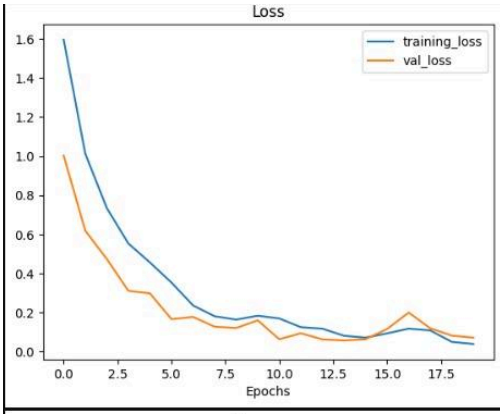


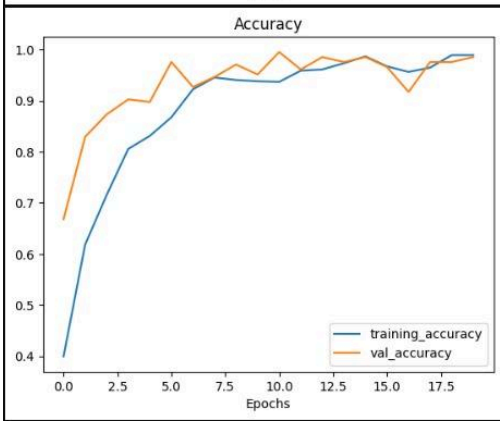


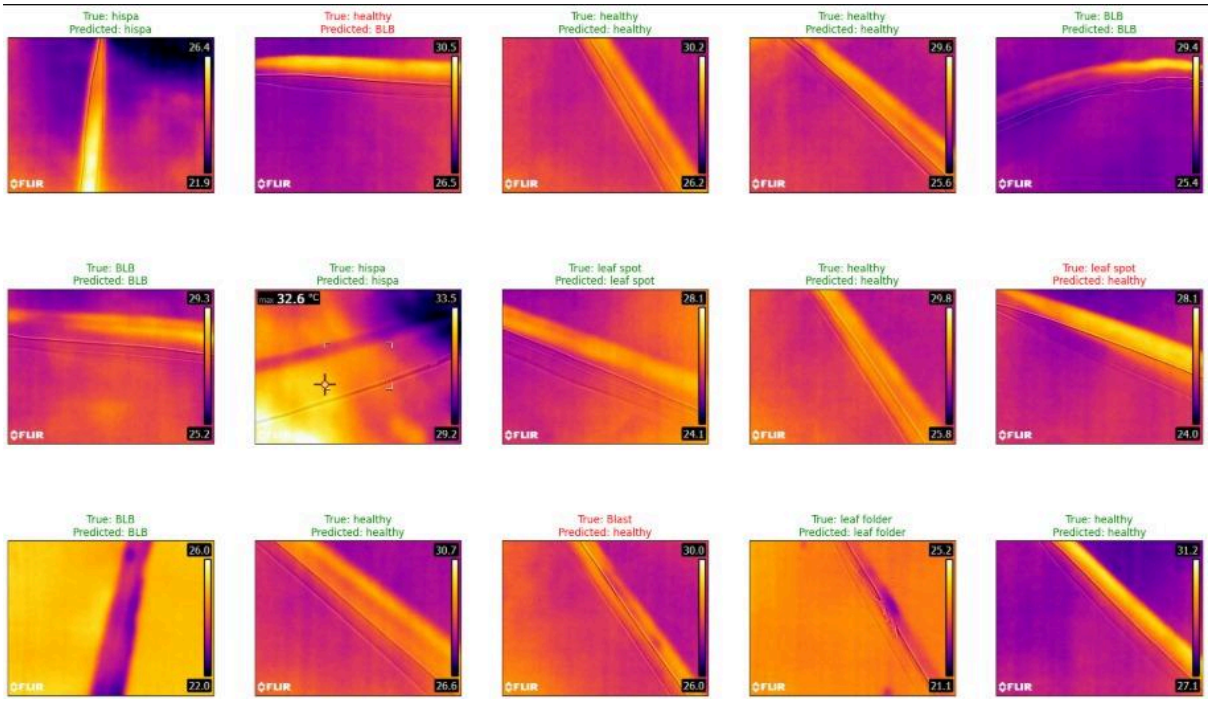
INCEPTION V3

Epoch 20 **Learning Rate** - 0.001 **Optimizer** - Adam

```
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.9944 - loss: 0.0317
Epoch 20: val_accuracy did not improve from 0.99512
26/26 ━━━━━━━━━━━━━━━━━━━━ 245s 9s/step - accuracy: 0.9942 - loss: 0.0320 - val_accuracy: 0.9854 - val_loss: 0.0716

[24]: results = model.evaluate(test_images, verbose=0)

      print("    Test Loss: {:.5f}".format(results[0]))
      print("Test Accuracy: {:.2f}%".format(results[1] * 100))

    Test Loss: 0.16853
Test Accuracy: 94.96%
```

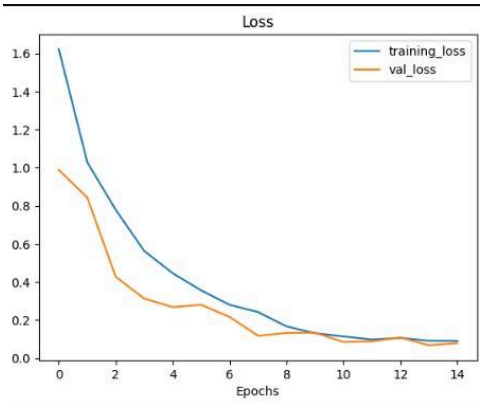


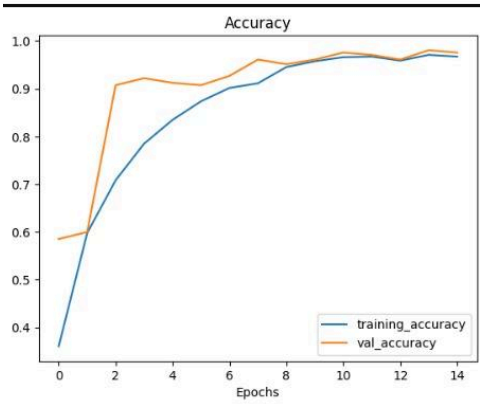


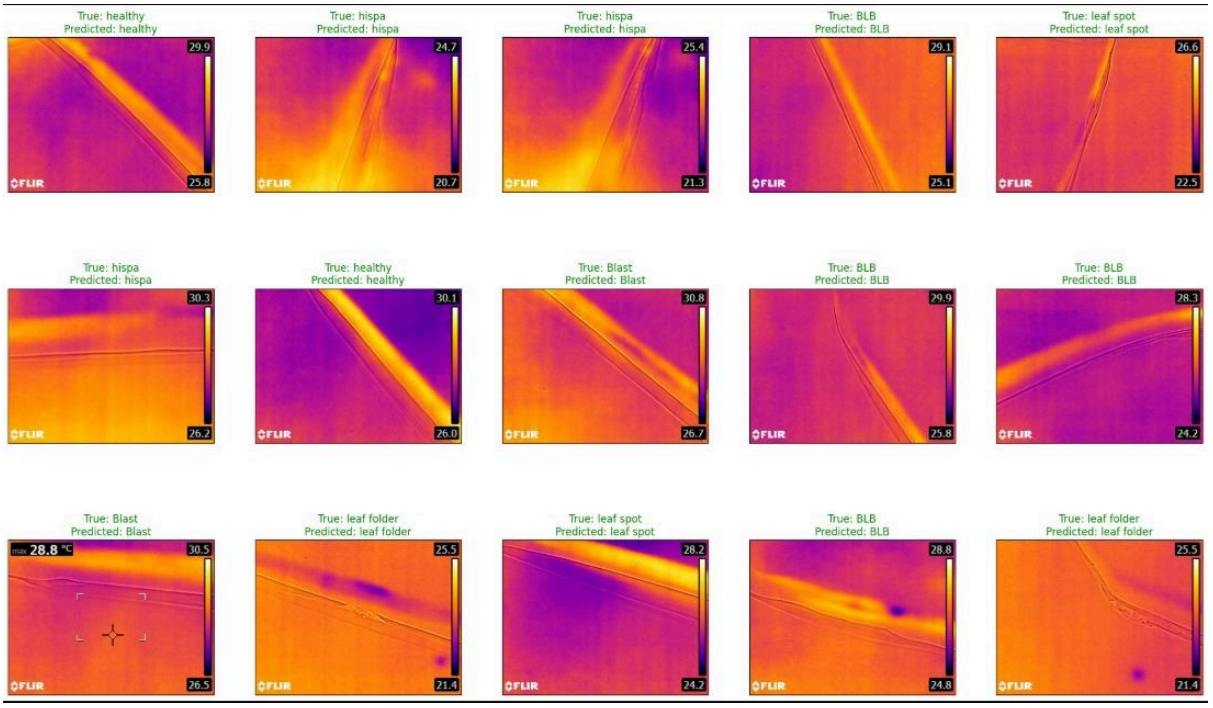


Epoch 15 **Learning Rate** - 0.001 **Optimizer** - Adam

```
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 8s/step - accuracy: 0.9704 - loss: 0.0831
Epoch 15: val_accuracy did not improve from 0.98049
26/26 ━━━━━━━━━━━━━━━━━━━━ 249s 10s/step - accuracy: 0.9703 - loss: 0.0834 - val_accuracy: 0.9756 - val_loss: 0.0789

[50]: results = model.evaluate(test_images, verbose=0)

      print("    Test Loss: {:.5f}".format(results[0]))
      print("Test Accuracy: {:.2f}%".format(results[1] * 100))

    Test Loss: 0.16544
Test Accuracy: 95.35%
```





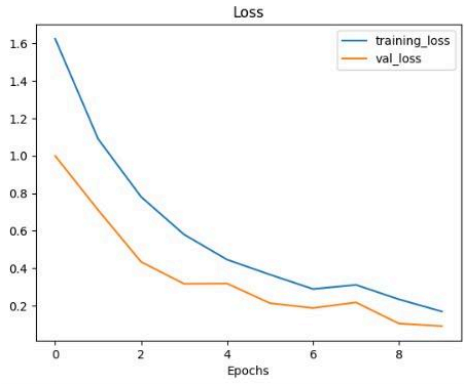


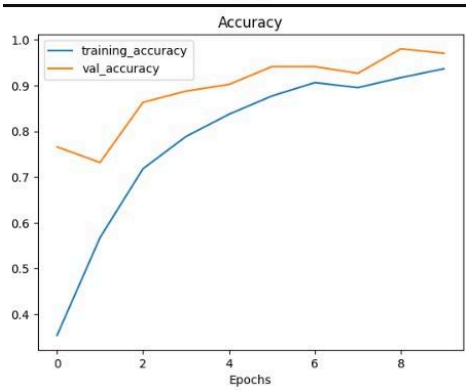
Epoch 10 **Learning Rate** - 0.001 **Optimizer** - Adam

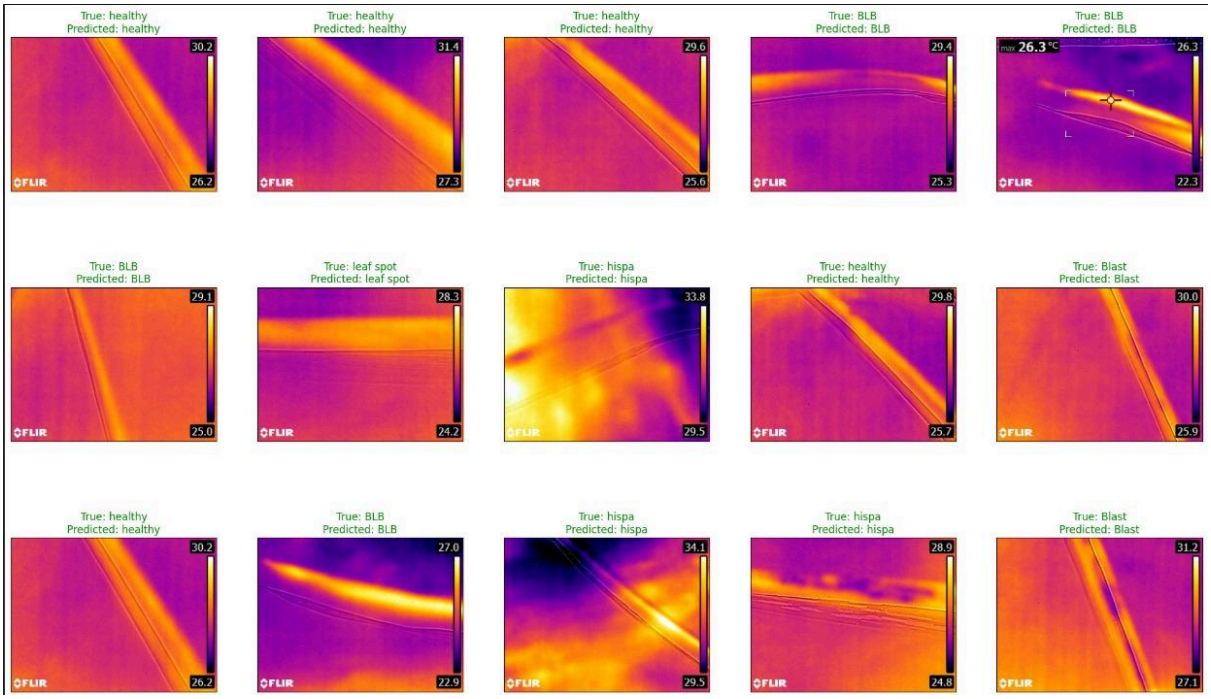
```
Epoch 10: val_accuracy did not improve from 0.98049
26/26 ━━━━━━━━━━━━━━━━━━━━ 251s 10s/step - accuracy: 0.9371 - loss: 0.1794 - val_accuracy: 0.9707 - val_loss: 0.0899
```

```
[79]: results = model.evaluate(test_images, verbose=0)

      print("    Test Loss: {:.5f}".format(results[0]))
      print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.22235
Test Accuracy: 92.64%
```





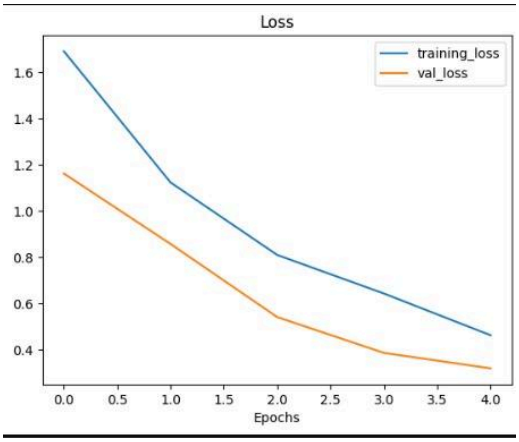


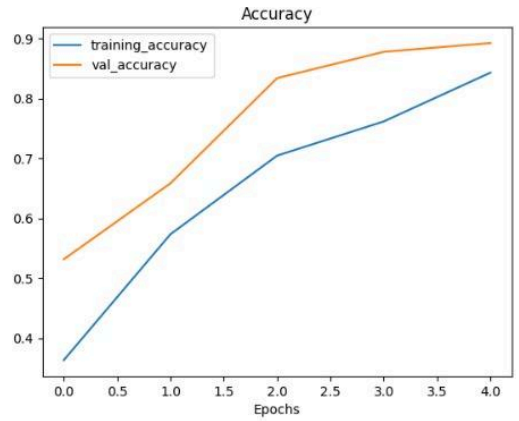
Epoch 5 **Learning Rate** - 0.001 **Optimizer** - Adam

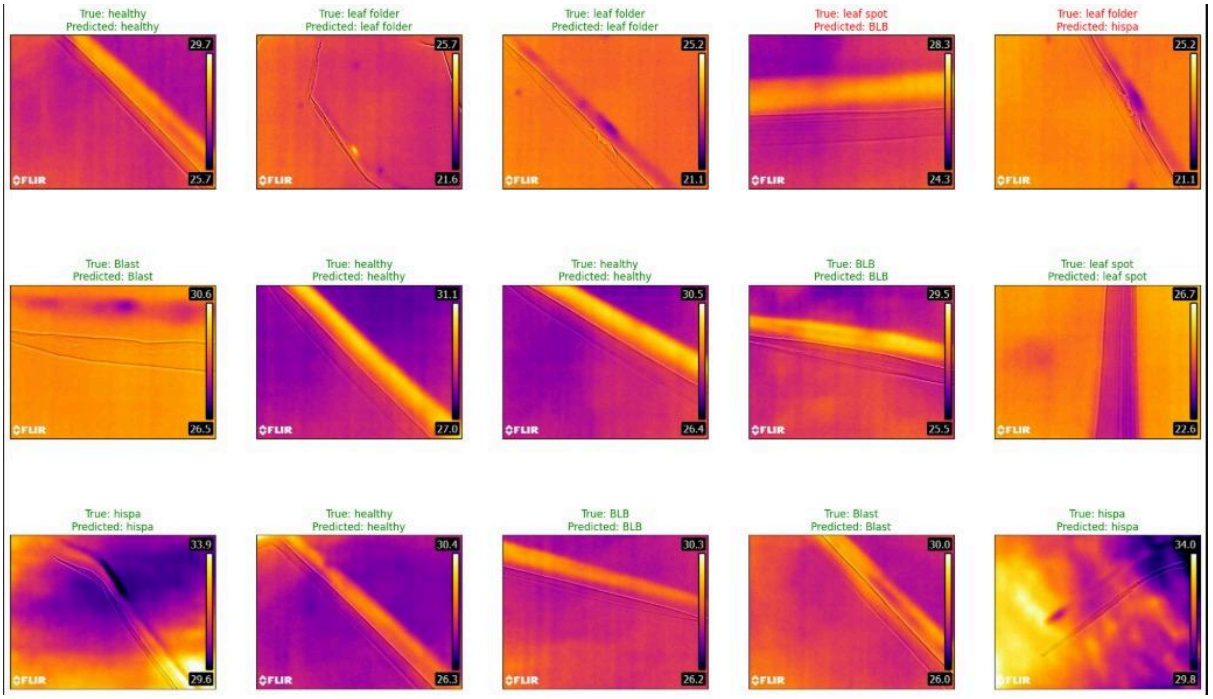
```
Epoch 5: val_accuracy improved from 0.87805 to 0.89268, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
26/26 ━━━━━━━━━━━━━━━━━━━━ 266s 10s/step - accuracy: 0.8488 - loss: 0.4519 - val_accuracy: 0.8927 - val_loss: 0.3181
```

```
[106]: results = model.evaluate(test_images, verbose=0)

       print("    Test Loss: {:.5f}".format(results[0]))
       print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.46634
Test Accuracy: 81.78%
```

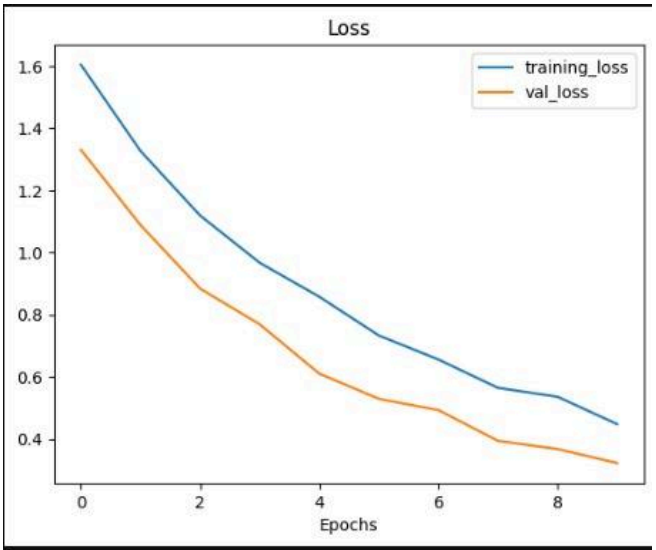


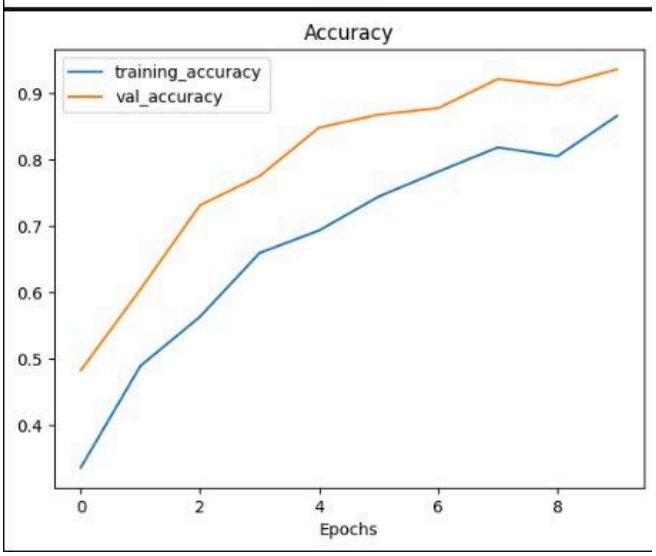




Epoch 10 **Learning Rate** - 0.0001 **Optimizer** - Adam

```
Epoch 10: val_accuracy improved from 0.92195 to 0.93659, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
26/26 ━━━━━━━━━━━━━━━━━━━━ 228s 9s/step - accuracy: 0.8662 - loss: 0.4445 - val_accuracy: 0.9366 - val_loss: 0.3231

[134]: results = model.evaluate(test_images, verbose=0)

       print("    Test Loss: {:.5f}".format(results[0]))
       print("Test Accuracy: {:.2f}%".format(results[1] * 100))

    Test Loss: 0.41573
Test Accuracy: 85.66%
```



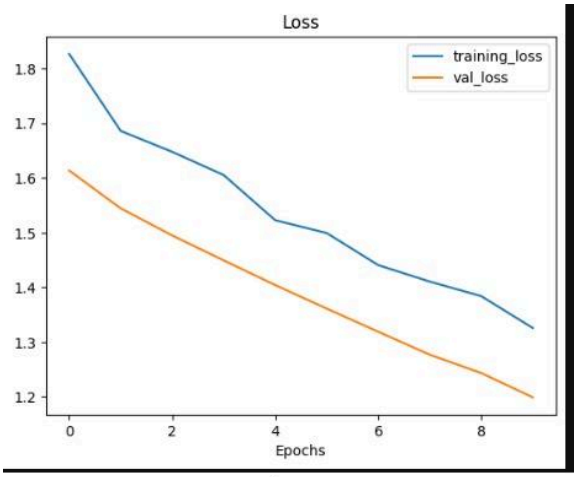


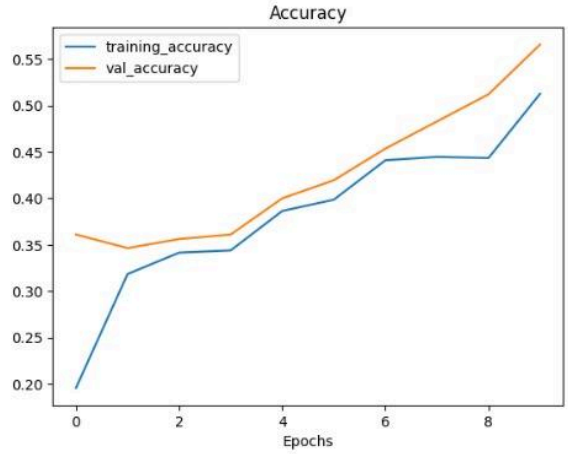
Epoch 10 **Learning Rate** - 0.00001 **Optimizer** - Adam

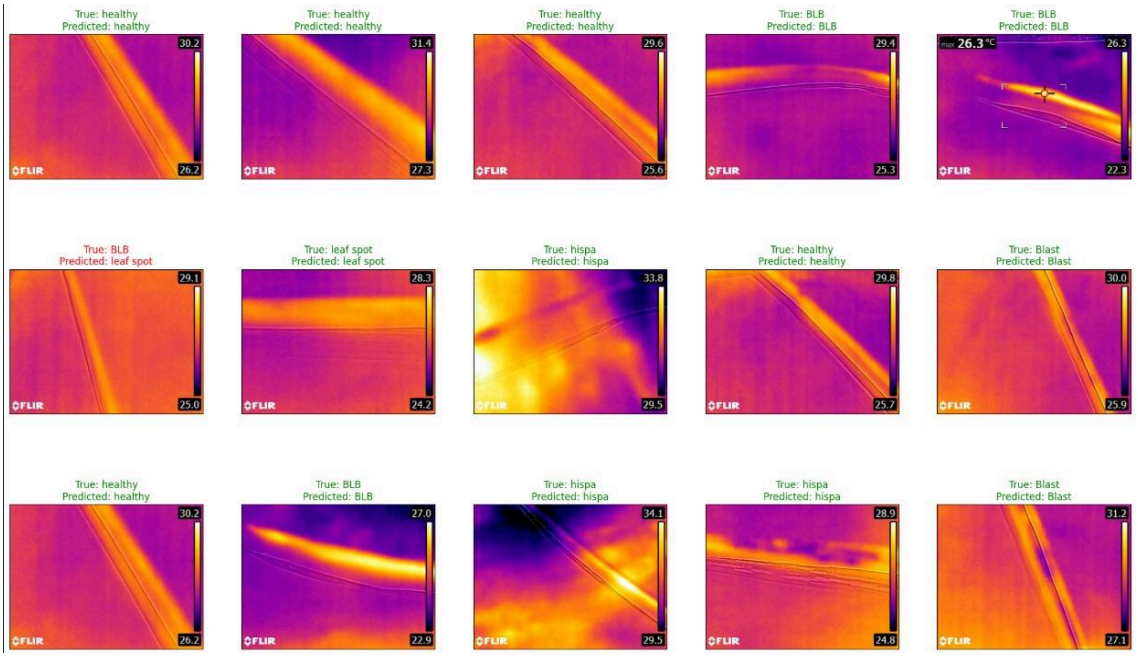
```
Epoch 10: val_accuracy improved from 0.51220 to 0.56585, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
26/26 ━━━━━━━━━━━━━━━━━━━━ 244s 9s/step - accuracy: 0.5148 - loss: 1.3040 - val_accuracy: 0.5659 - val_loss: 1.1989
```

[161]:
```
results = model.evaluate(test_images, verbose=0)

print("    Test Loss: {:.5f}".format(results[0]))
print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 1.25263
Test Accuracy: 56.98%
```





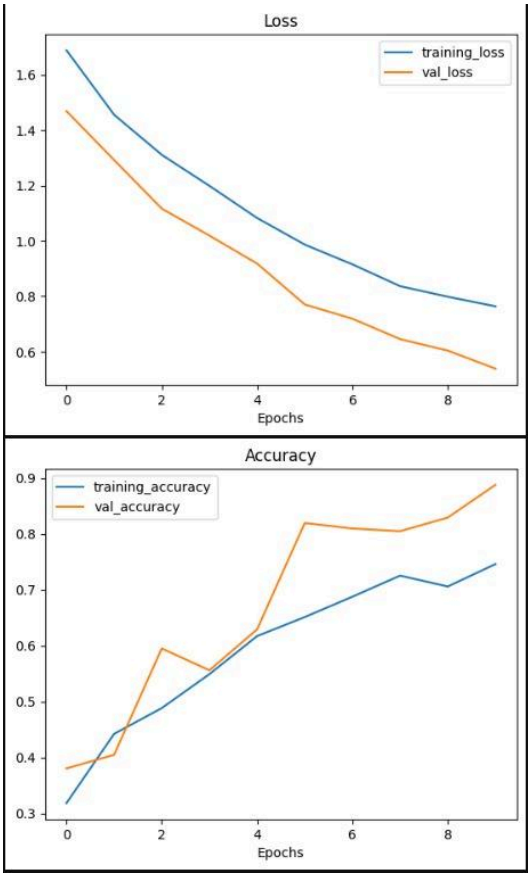


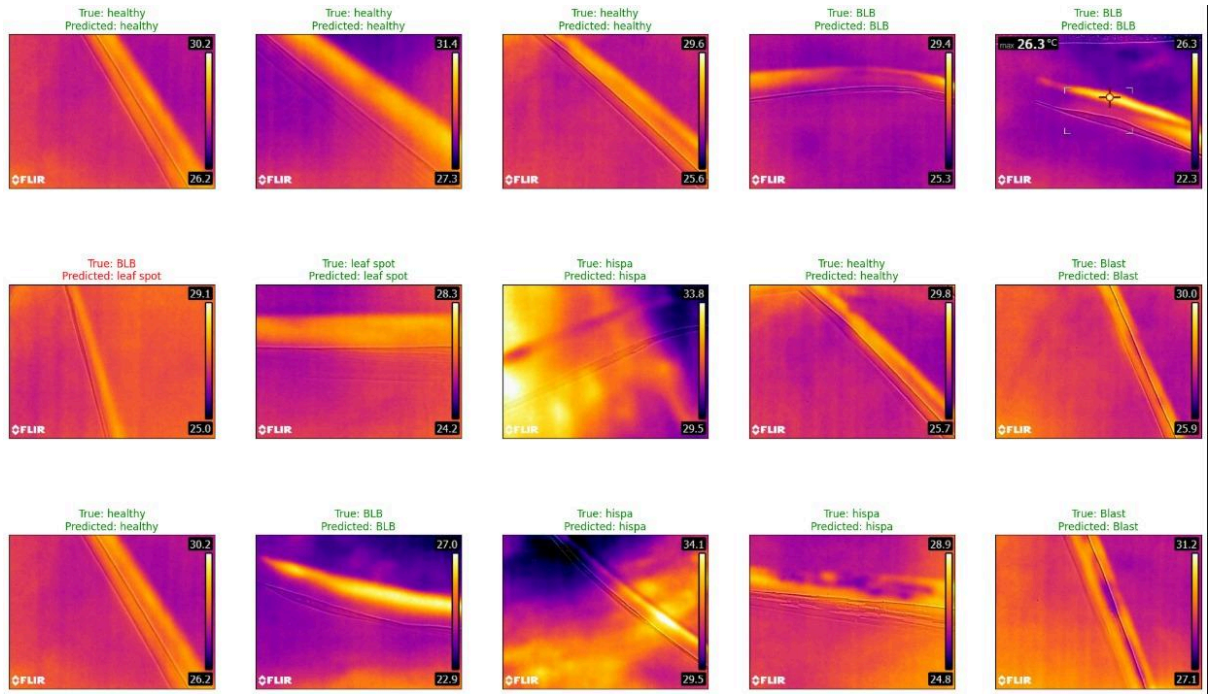
Epoch 10 **Learning Rate** - 0.001 **Optimizer** - SGD

```
Epoch 10: val_accuracy improved from 0.82927 to 0.88780, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
26/26 ━━━━━━━━━━━━━━━━━━━━ 223s 9s/step - accuracy: 0.7225 - loss: 0.7606 - val_accuracy: 0.8878 - val_loss: 0.5381
```

```
[187]: results = model.evaluate(test_images, verbose=0)

       print("    Test Loss: {:.5f}".format(results[0]))
       print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.65049
Test Accuracy: 81.01%
```



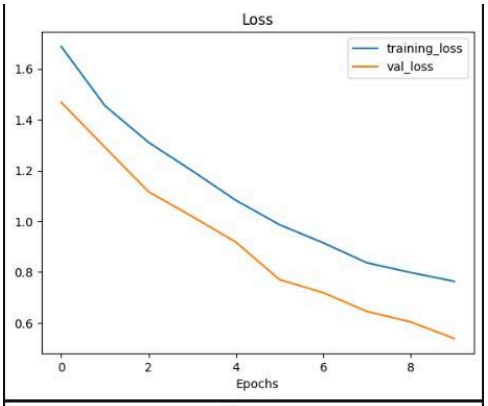


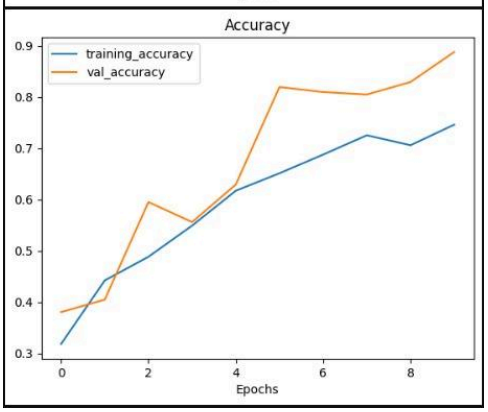
Epoch 10 **Learning Rate** - 0.001 **Optimizer** - SGD

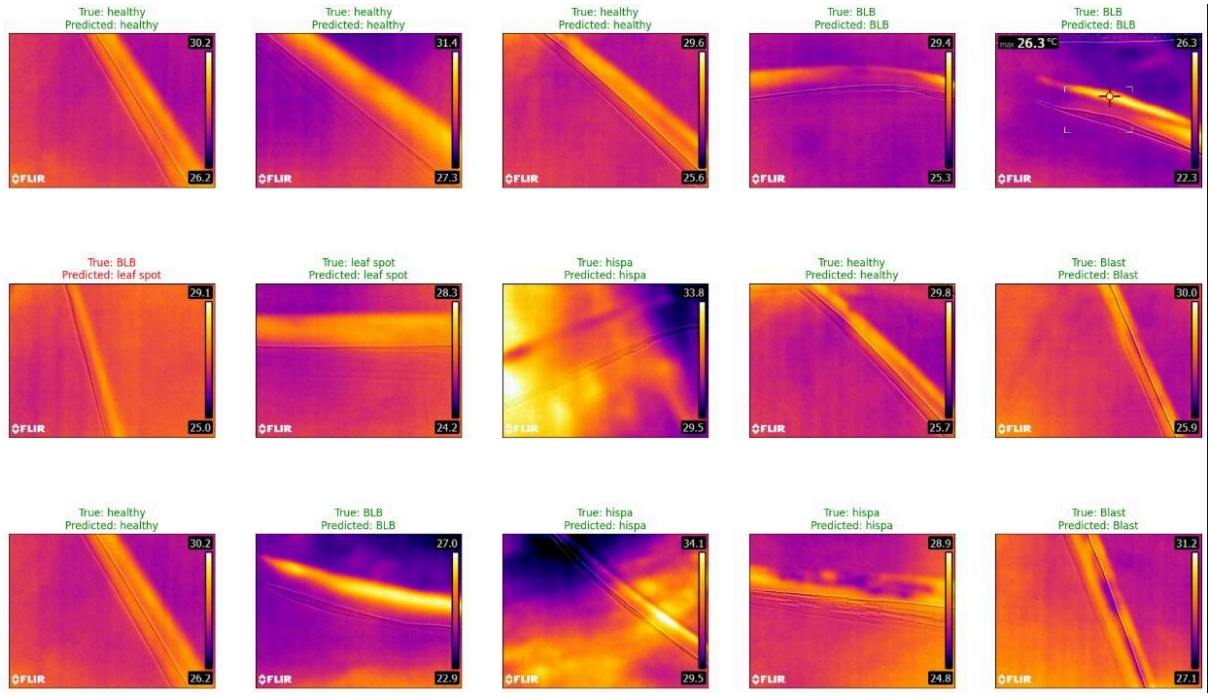
```
Epoch 10: val_accuracy did not improve from 0.94634
26/26 ━━━━━━━━━━━━━━━━━━━━ 221s 9s/step - accuracy: 0.8652 - loss: 0.3832 - val_accuracy: 0.7805 - val_loss: 0.5070
```

```
[215]: results = model.evaluate(test_images, verbose=0)

       print("    Test Loss: {:.5f}".format(results[0]))
       print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.31918
Test Accuracy: 87.21%
```

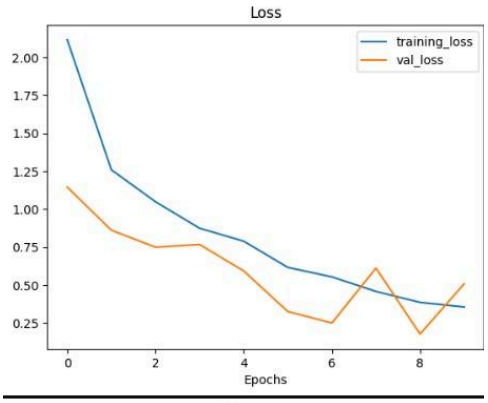


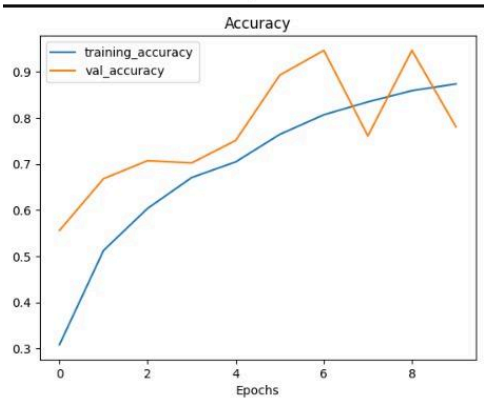


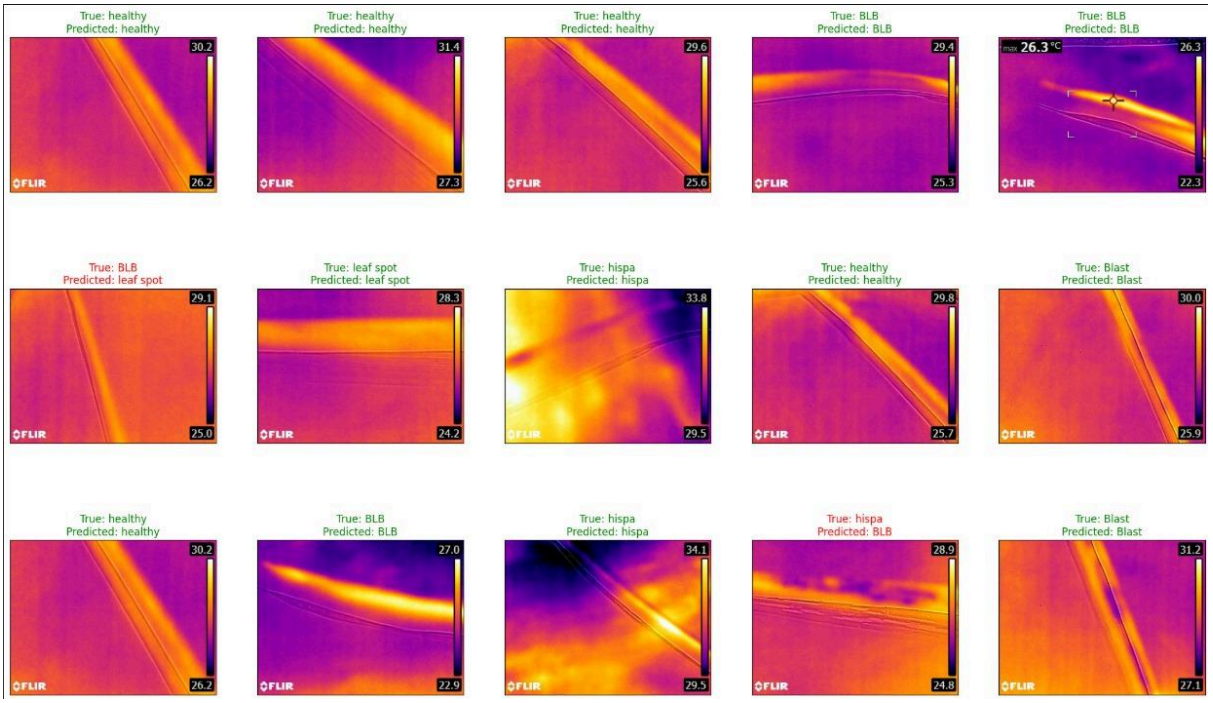


Epoch 10 **Learning Rate** - 0.001 **Optimizer** - RMSProp

```
Epoch 10: val_accuracy did not improve from 0.94634
26/26 ━━━━━━━━━━━━━━━━━━━━ 221s 9s/step - accuracy: 0.8652 - loss: 0.3832 - val_accuracy: 0.7805 - val_loss: 0.5070

[215]: results = model.evaluate(test_images, verbose=0)

       print("    Test Loss: {:.5f}".format(results[0]))
       print("Test Accuracy: {:.2f}%".format(results[1] * 100))

    Test Loss: 0.31918
Test Accuracy: 87.21%
```





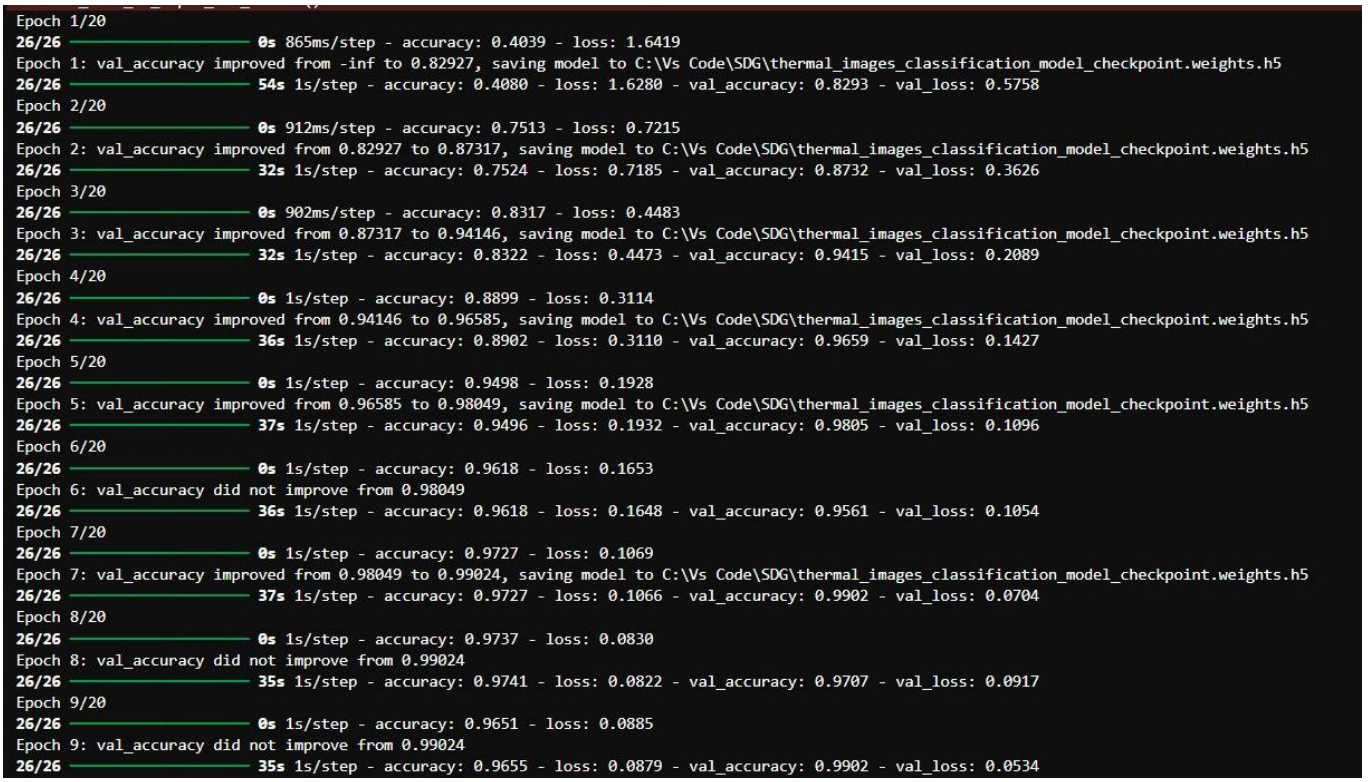


PERFORMANCE TABLE

Model	Epoch	Learning Rate	Optimizer	Test Accuracy	Loss
MobileNetV3	5	0.001	Adam	89.15%	0.26634
MobileNetV3	10	0.001	Adam	98.06%	0.04569
MobileNetV3	15	0.001	Adam	99.22%	0.02021
MobileNetV3	20	0.001	Adam	100%	0.01367
MobileNetV3	10	0.00001	Adam	51.94%	1032523
MobileNetV3	10	0.001	SGD	82.95%	0.46146
MobileNetV3	10	0.001	RMSprop	98.06%	0.05137
Resnet50	5	0.001	Adam	84.50%	0.51547
Resnet50	10	0.001	Adam	86.05%	0.41489
Resnet50	15	0.001	Adam	88.37%	0.40090
Resnet50	20	0.001	Adam	88.37%	0.39060
Resnet50	10	0.0001	Adam	75.97%	0.80889
Resnet50	10	0.00001	Adam	42.64%	1.49933
Resnet50	10	0.001	SGD	76.74%	0.88047
Resnet50	10	0.001	RMSprop	82.95%	0.54817
InceptionNet	5	0.001	Adam	81.78%	0.46634
InceptionNet	10	0.001	Adam	92.64%	.022235
InceptionNe	15	0.001	Adam	95.35%	0.16544

t					
InceptionNet	20	0.001	Adam	94.96%	0.16853
InceptionNet	10	0.0001	Adam	85.66%	0.41573
InceptionNet	10	0.00001	Adam	56.98%	1.25263
InceptionNet	10	0.001	SGD	81.01%	0.65049
InceptionNet	10	0.001	RMSprop	87.21%	0.31918

MobileNet V3 Epoch 20 , Learning Rate 0.001 , Adam optimizer



```
Epoch 10/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.9905 - loss: 0.0504
Epoch 10: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 35s 1s/step - accuracy: 0.9903 - loss: 0.0507 - val_accuracy: 0.9902 - val_loss: 0.0597
Epoch 11/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.9956 - loss: 0.0303
Epoch 11: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 36s 1s/step - accuracy: 0.9956 - loss: 0.0303 - val_accuracy: 0.9854 - val_loss: 0.0518
Epoch 12/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.9884 - loss: 0.0352
Epoch 12: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 35s 1s/step - accuracy: 0.9885 - loss: 0.0349 - val_accuracy: 0.9854 - val_loss: 0.0569
Epoch 13/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.9903 - loss: 0.0330
Epoch 13: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 36s 1s/step - accuracy: 0.9902 - loss: 0.0331 - val_accuracy: 0.9854 - val_loss: 0.0689
Epoch 14/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.9998 - loss: 0.0163
Epoch 14: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 35s 1s/step - accuracy: 0.9998 - loss: 0.0162 - val_accuracy: 0.9902 - val_loss: 0.0707
Epoch 15/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.9976 - loss: 0.0170
Epoch 15: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 36s 1s/step - accuracy: 0.9976 - loss: 0.0169 - val_accuracy: 0.9902 - val_loss: 0.0685
Epoch 16/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 1.0000 - loss: 0.0099
Epoch 16: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 36s 1s/step - accuracy: 1.0000 - loss: 0.0099 - val_accuracy: 0.9854 - val_loss: 0.0577
Epoch 17/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 1.0000 - loss: 0.0055
Epoch 17: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 39s 1s/step - accuracy: 0.9999 - loss: 0.0056 - val_accuracy: 0.9902 - val_loss: 0.0491
Epoch 18/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 1.0000 - loss: 0.0091
Epoch 18: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 36s 1s/step - accuracy: 1.0000 - loss: 0.0091 - val_accuracy: 0.9902 - val_loss: 0.0512
Epoch 19/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 1.0000 - loss: 0.0086
Epoch 19: val_accuracy did not improve from 0.99024
```

```
Epoch 19: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 39s 2s/step - accuracy: 1.0000 - loss: 0.0085 - val_accuracy: 0.9902 - val_loss: 0.0705
Epoch 20/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 1.0000 - loss: 0.0048
Epoch 20: val_accuracy did not improve from 0.99024
26/26 ━━━━━━━━━━━━━━━━━━━━ 36s 1s/step - accuracy: 1.0000 - loss: 0.0049 - val_accuracy: 0.9854 - val_loss: 0.1007
```

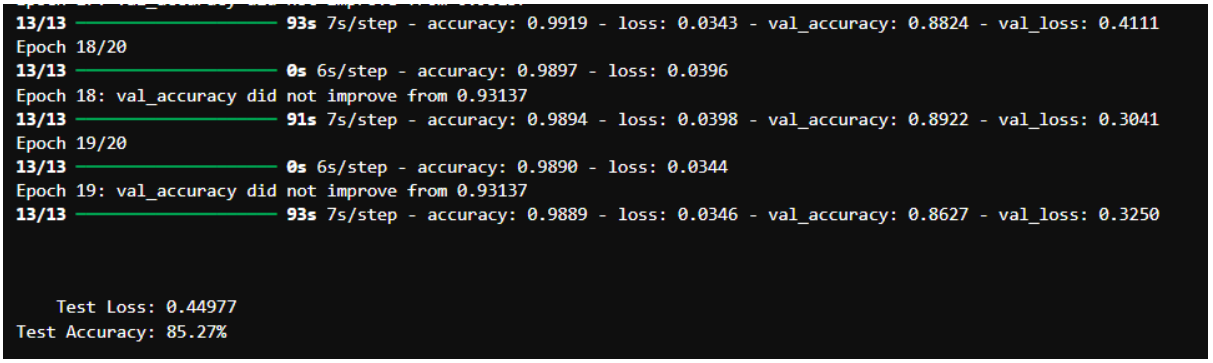
```
results = model.evaluate(test_images, verbose=0)

print("    Test Loss: {:.5f}".format(results[0]))
print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.00780
Test Accuracy: 100.00%
```

Resnet50 Epoch 20 Learning Rate 0.001 Adam Optimizer (Next Page)

```
Epoch 1/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.3846 - loss: 1.6039
Epoch 1: val_accuracy improved from -inf to 0.76471, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
13/13 ━━━━━━━━━━━━━━━━━━━━ 122s 8s/step - accuracy: 0.3899 - loss: 1.5948 - val_accuracy: 0.7647 - val_loss: 0.9176
Epoch 2/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.5389 - loss: 1.0959
Epoch 2: val_accuracy did not improve from 0.76471
13/13 ━━━━━━━━━━━━━━━━━━━━ 92s 7s/step - accuracy: 0.5427 - loss: 1.0894 - val_accuracy: 0.7451 - val_loss: 0.7028
Epoch 3/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.7432 - loss: 0.6825
Epoch 3: val_accuracy improved from 0.76471 to 0.79412, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
13/13 ━━━━━━━━━━━━━━━━━━━━ 94s 7s/step - accuracy: 0.7442 - loss: 0.6825 - val_accuracy: 0.7941 - val_loss: 0.5933
Epoch 4/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.8131 - loss: 0.5332
Epoch 4: val_accuracy did not improve from 0.79412
13/13 ━━━━━━━━━━━━━━━━━━━━ 92s 7s/step - accuracy: 0.8147 - loss: 0.5304 - val_accuracy: 0.7941 - val_loss: 0.5011
Epoch 5/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.8715 - loss: 0.4204
Epoch 5: val_accuracy improved from 0.79412 to 0.82353, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
13/13 ━━━━━━━━━━━━━━━━━━━━ 93s 7s/step - accuracy: 0.8699 - loss: 0.4203 - val_accuracy: 0.8235 - val_loss: 0.4276
Epoch 6/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.9160 - loss: 0.3057
Epoch 6: val_accuracy improved from 0.82353 to 0.83333, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
13/13 ━━━━━━━━━━━━━━━━━━━━ 92s 7s/step - accuracy: 0.9163 - loss: 0.3050 - val_accuracy: 0.8333 - val_loss: 0.4314
Epoch 7/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.9505 - loss: 0.2042
Epoch 7: val_accuracy improved from 0.83333 to 0.85294, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
13/13 ━━━━━━━━━━━━━━━━━━━━ 91s 7s/step - accuracy: 0.9497 - loss: 0.2057 - val_accuracy: 0.8529 - val_loss: 0.4027
Epoch 8/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 4s/step - accuracy: 0.9241 - loss: 0.2384
Epoch 8: val_accuracy improved from 0.85294 to 0.86275, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
13/13 ━━━━━━━━━━━━━━━━━━━━ 73s 6s/step - accuracy: 0.9249 - loss: 0.2357 - val_accuracy: 0.8627 - val_loss: 0.3741
```

```
Epoch 9/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 4s/step - accuracy: 0.9413 - loss: 0.1643
Epoch 9: val_accuracy did not improve from 0.86275
13/13 ━━━━━━━━━━━━━━━━━━━━ 72s 6s/step - accuracy: 0.9415 - loss: 0.1642 - val_accuracy: 0.8529 - val_loss: 0.3533
Epoch 10/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 5s/step - accuracy: 0.9798 - loss: 0.1066
Epoch 10: val_accuracy did not improve from 0.86275
13/13 ━━━━━━━━━━━━━━━━━━━━ 87s 7s/step - accuracy: 0.9798 - loss: 0.1066 - val_accuracy: 0.8529 - val_loss: 0.4149
Epoch 11/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.9763 - loss: 0.0912
Epoch 11: val_accuracy improved from 0.86275 to 0.89216, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
13/13 ━━━━━━━━━━━━━━━━━━━━ 93s 7s/step - accuracy: 0.9761 - loss: 0.0915 - val_accuracy: 0.8922 - val_loss: 0.2950
Epoch 12/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.9582 - loss: 0.0751
Epoch 12: val_accuracy improved from 0.89216 to 0.93137, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
13/13 ━━━━━━━━━━━━━━━━━━━━ 93s 7s/step - accuracy: 0.9585 - loss: 0.0754 - val_accuracy: 0.9314 - val_loss: 0.2706
Epoch 13/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.9758 - loss: 0.0824
Epoch 13: val_accuracy did not improve from 0.93137
13/13 ━━━━━━━━━━━━━━━━━━━━ 91s 7s/step - accuracy: 0.9763 - loss: 0.0821 - val_accuracy: 0.8725 - val_loss: 0.2642
Epoch 14/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 5s/step - accuracy: 0.9800 - loss: 0.0854
Epoch 14: val_accuracy did not improve from 0.93137
13/13 ━━━━━━━━━━━━━━━━━━━━ 86s 7s/step - accuracy: 0.9799 - loss: 0.0856 - val_accuracy: 0.8725 - val_loss: 0.3871
Epoch 15/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.9943 - loss: 0.0485
Epoch 15: val_accuracy did not improve from 0.93137
13/13 ━━━━━━━━━━━━━━━━━━━━ 91s 7s/step - accuracy: 0.9938 - loss: 0.0491 - val_accuracy: 0.8431 - val_loss: 0.5025
Epoch 16/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.9677 - loss: 0.0745
Epoch 16: val_accuracy did not improve from 0.93137
13/13 ━━━━━━━━━━━━━━━━━━━━ 93s 7s/step - accuracy: 0.9688 - loss: 0.0734 - val_accuracy: 0.9020 - val_loss: 0.2950
Epoch 17/20
13/13 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.9924 - loss: 0.0337
Epoch 17: val_accuracy did not improve from 0.93137
13/13 ━━━━━━━━━━━━━━━━━━━━ 93s 7s/step - accuracy: 0.9919 - loss: 0.0343 - val_accuracy: 0.8824 - val_loss: 0.4111
```



InceptionNet Epoch 20 Adam Optimizer Learning Rate 0.001

```
Epoch 1/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.2835 - loss: 1.8110
Epoch 1: val_accuracy improved from -inf to 0.54146, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
26/26 ━━━━━━━━━━━━━━━━━━━━ 269s 10s/step - accuracy: 0.2857 - loss: 1.8038 - val_accuracy: 0.5415 - val_loss: 1.0244
Epoch 2/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 8s/step - accuracy: 0.5643 - loss: 1.0764
Epoch 2: val_accuracy improved from 0.54146 to 0.83415, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
26/26 ━━━━━━━━━━━━━━━━━━━━ 268s 10s/step - accuracy: 0.5653 - loss: 1.0745 - val_accuracy: 0.8341 - val_loss: 0.6376
Epoch 3/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.7154 - loss: 0.8003
Epoch 3: val_accuracy improved from 0.83415 to 0.88780, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
26/26 ━━━━━━━━━━━━━━━━━━━━ 253s 9s/step - accuracy: 0.7162 - loss: 0.7972 - val_accuracy: 0.8878 - val_loss: 0.4157
Epoch 4/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 8s/step - accuracy: 0.7820 - loss: 0.5428
Epoch 4: val_accuracy improved from 0.88780 to 0.93659, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
26/26 ━━━━━━━━━━━━━━━━━━━━ 256s 10s/step - accuracy: 0.7823 - loss: 0.5427 - val_accuracy: 0.9366 - val_loss: 0.3252
Epoch 5/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 8s/step - accuracy: 0.8769 - loss: 0.4398
Epoch 5: val_accuracy did not improve from 0.93659
26/26 ━━━━━━━━━━━━━━━━━━━━ 253s 10s/step - accuracy: 0.8762 - loss: 0.4394 - val_accuracy: 0.9171 - val_loss: 0.2654
Epoch 6/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.8519 - loss: 0.3894
Epoch 6: val_accuracy did not improve from 0.93659
26/26 ━━━━━━━━━━━━━━━━━━━━ 234s 9s/step - accuracy: 0.8518 - loss: 0.3893 - val_accuracy: 0.9366 - val_loss: 0.2291
Epoch 7/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.8681 - loss: 0.3434
Epoch 7: val_accuracy did not improve from 0.93659
26/26 ━━━━━━━━━━━━━━━━━━━━ 220s 9s/step - accuracy: 0.8688 - loss: 0.3414 - val_accuracy: 0.9220 - val_loss: 0.1775
Epoch 8/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.9144 - loss: 0.2702
Epoch 8: val_accuracy improved from 0.93659 to 0.96585, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
26/26 ━━━━━━━━━━━━━━━━━━━━ 232s 9s/step - accuracy: 0.9146 - loss: 0.2700 - val_accuracy: 0.9659 - val_loss: 0.1440
Epoch 9/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 6s/step - accuracy: 0.9436 - loss: 0.1825
Epoch 9: val_accuracy did not improve from 0.96585
26/26 ━━━━━━━━━━━━━━━━━━━━ 237s 8s/step - accuracy: 0.9436 - loss: 0.1822 - val_accuracy: 0.9659 - val_loss: 0.1037
Epoch 10/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.9565 - loss: 0.1402
Epoch 10: val_accuracy did not improve from 0.96585
26/26 ━━━━━━━━━━━━━━━━━━━━ 212s 8s/step - accuracy: 0.9563 - loss: 0.1407 - val_accuracy: 0.9561 - val_loss: 0.1252
Epoch 11/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.9754 - loss: 0.0953
Epoch 11: val_accuracy improved from 0.96585 to 0.98049, saving model to C:\Vs Code\SDG\thermal_images_classification_model_checkpoint.weights.h5
26/26 ━━━━━━━━━━━━━━━━━━━━ 237s 9s/step - accuracy: 0.9750 - loss: 0.0959 - val_accuracy: 0.9805 - val_loss: 0.0772
Epoch 12/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.9665 - loss: 0.1111
Epoch 12: val_accuracy did not improve from 0.98049
26/26 ━━━━━━━━━━━━━━━━━━━━ 258s 9s/step - accuracy: 0.9663 - loss: 0.1114 - val_accuracy: 0.9317 - val_loss: 0.1452
Epoch 13/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.9321 - loss: 0.1867
Epoch 13: val_accuracy did not improve from 0.98049
26/26 ━━━━━━━━━━━━━━━━━━━━ 260s 9s/step - accuracy: 0.9326 - loss: 0.1857 - val_accuracy: 0.9756 - val_loss: 0.0876
Epoch 14/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.9562 - loss: 0.1143
Epoch 14: val_accuracy did not improve from 0.98049
26/26 ━━━━━━━━━━━━━━━━━━━━ 233s 9s/step - accuracy: 0.9564 - loss: 0.1142 - val_accuracy: 0.9707 - val_loss: 0.0807
Epoch 15/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.9610 - loss: 0.1085
Epoch 15: val_accuracy did not improve from 0.98049
26/26 ━━━━━━━━━━━━━━━━━━━━ 241s 9s/step - accuracy: 0.9612 - loss: 0.1083 - val_accuracy: 0.9707 - val_loss: 0.1014
Epoch 16/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.9791 - loss: 0.0721
Epoch 16: val_accuracy did not improve from 0.98049
26/26 ━━━━━━━━━━━━━━━━━━━━ 238s 9s/step - accuracy: 0.9789 - loss: 0.0720 - val_accuracy: 0.9756 - val_loss: 0.0808
Epoch 17/20
26/26 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.9762 - loss: 0.0757
Epoch 17: val_accuracy did not improve from 0.98049
26/26 ━━━━━━━━━━━━━━━━━━━━ 241s 9s/step - accuracy: 0.9764 - loss: 0.0752 - val_accuracy: 0.9756 - val_loss: 0.0831
```

```
results = model.evaluate(test_images, verbose=0)

print("    Test Loss: {:.5f}".format(results[0]))
print("Test Accuracy: {:.2f}%".format(results[1] * 100))
```

```
    Test Loss: 0.18538
Test Accuracy: 93.02%
```