DFU_CNN

July 19, 2024

[29]:
```
import os
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

[3]:
```
tf.__version__
```

[3]:
```
'2.16.2'
```

[31]:
```
test_dir = 'ThermoDataBase/dataset/val'
```

[5]:
```
# daTa preprocesing of trainign

train_datagen = ImageDataGenerator(
        rescale=1./255,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True)

training_set = train_datagen.flow_from_directory(
        'dataset/train',
        target_size=(64, 64),
        batch_size=32,
        class_mode='binary')
```

```
Found 1444 images belonging to 2 classes.
```

[7]:
```
# Data preprocessing of test

test_datagen = ImageDataGenerator(rescale=1./255)
test_set = test_datagen.flow_from_directory(
        'dataset/val',
        target_size=(64, 64),
        batch_size=32,
        class_mode='binary')
```

```
Found 422 images belonging to 2 classes.
```

[9]:

```
cnn = tf.keras.models.Sequential()
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu',␣
 ↪input_shape=[64, 64, 3]))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))
cnn.add(tf.keras.layers.Flatten())
cnn.add(tf.keras.layers.Dense(units = 128, activation='relu'))
cnn.add(tf.keras.layers.Dense(units = 1, activation='sigmoid'))
```

```
/opt/anaconda3/lib/python3.12/site-
packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not
pass an `input_shape`/`input_dim` argument to a layer. When using Sequential
models, prefer using an `Input(shape)` object as the first layer in the model
instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

[11]:

```
cnn.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics =␣
 ↪['accuracy'])
```

[13]:

```
cnn.fit(x = training_set, validation_data = test_set, epochs = 25)
```

```
Epoch 1/25
```

```
/opt/anaconda3/lib/python3.12/site-
packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121:
UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in
its constructor. `**kwargs` can include `workers`, `use_multiprocessing`,
`max_queue_size`. Do not pass these arguments to `fit()`, as they will be
ignored.
  self._warn_if_super_not_called()
```

```
46/46                 4s 64ms/step -
accuracy: 0.7051 - loss: 0.5142 - val_accuracy: 0.8128 - val_loss: 0.4589
Epoch 2/25
46/46                 3s 54ms/step -
accuracy: 0.8433 - loss: 0.3510 - val_accuracy: 0.8791 - val_loss: 0.3889
Epoch 3/25
46/46                 3s 54ms/step -
accuracy: 0.8442 - loss: 0.3309 - val_accuracy: 0.8294 - val_loss: 0.3024
Epoch 4/25
46/46                 3s 57ms/step -
accuracy: 0.8786 - loss: 0.2861 - val_accuracy: 0.8815 - val_loss: 0.3595
Epoch 5/25
46/46                 3s 57ms/step -
accuracy: 0.8759 - loss: 0.2618 - val_accuracy: 0.8436 - val_loss: 0.2730
Epoch 6/25
46/46                 3s 57ms/step -
```

```
accuracy: 0.9199 - loss: 0.2063 - val_accuracy: 0.8839 - val_loss: 0.2392
Epoch 7/25
46/46            3s 57ms/step -
accuracy: 0.9217 - loss: 0.1677 - val_accuracy: 0.8815 - val_loss: 0.2944
Epoch 8/25
46/46            3s 57ms/step -
accuracy: 0.9130 - loss: 0.1750 - val_accuracy: 0.8673 - val_loss: 0.3385
Epoch 9/25
46/46            3s 56ms/step -
accuracy: 0.9157 - loss: 0.2078 - val_accuracy: 0.9218 - val_loss: 0.2060
Epoch 10/25
46/46            3s 56ms/step -
accuracy: 0.9491 - loss: 0.1238 - val_accuracy: 0.8981 - val_loss: 0.2391
Epoch 11/25
46/46            3s 56ms/step -
accuracy: 0.9588 - loss: 0.0998 - val_accuracy: 0.9052 - val_loss: 0.2600
Epoch 12/25
46/46            3s 56ms/step -
accuracy: 0.9717 - loss: 0.0847 - val_accuracy: 0.9194 - val_loss: 0.2624
Epoch 13/25
46/46            3s 58ms/step -
accuracy: 0.9701 - loss: 0.0861 - val_accuracy: 0.9194 - val_loss: 0.2925
Epoch 14/25
46/46            3s 56ms/step -
accuracy: 0.9739 - loss: 0.0709 - val_accuracy: 0.8934 - val_loss: 0.3576
Epoch 15/25
46/46            3s 59ms/step -
accuracy: 0.9567 - loss: 0.1051 - val_accuracy: 0.8910 - val_loss: 0.3044
Epoch 16/25
46/46            3s 57ms/step -
accuracy: 0.9676 - loss: 0.0869 - val_accuracy: 0.8791 - val_loss: 0.5013
Epoch 17/25
46/46            3s 61ms/step -
accuracy: 0.9657 - loss: 0.0942 - val_accuracy: 0.8981 - val_loss: 0.4807
Epoch 18/25
46/46            4s 75ms/step -
accuracy: 0.9751 - loss: 0.0688 - val_accuracy: 0.9147 - val_loss: 0.3165
Epoch 19/25
46/46            3s 62ms/step -
accuracy: 0.9774 - loss: 0.0662 - val_accuracy: 0.9100 - val_loss: 0.3877
Epoch 20/25
46/46            3s 61ms/step -
accuracy: 0.9848 - loss: 0.0441 - val_accuracy: 0.9100 - val_loss: 0.4050
Epoch 21/25
46/46            4s 74ms/step -
accuracy: 0.9835 - loss: 0.0406 - val_accuracy: 0.9171 - val_loss: 0.3949
Epoch 22/25
46/46            3s 59ms/step -
```

```
accuracy: 0.9860 - loss: 0.0434 - val_accuracy: 0.8839 - val_loss: 0.4378
Epoch 23/25
46/46              3s 56ms/step -
accuracy: 0.9628 - loss: 0.0938 - val_accuracy: 0.9265 - val_loss: 0.3893
Epoch 24/25
46/46              3s 61ms/step -
accuracy: 0.9816 - loss: 0.0482 - val_accuracy: 0.9076 - val_loss: 0.4649
Epoch 25/25
46/46              3s 60ms/step -
accuracy: 0.9862 - loss: 0.0299 - val_accuracy: 0.9100 - val_loss: 0.5348
```

[13]:
```
<keras.src.callbacks.history.History at 0x176bdca70>
```

[33]:
```python
y_true = []
y_pred = []

# Iterate through each class directory
for class_name in os.listdir(test_dir):
    class_dir = os.path.join(test_dir, class_name)
    if os.path.isdir(class_dir):
        for img_name in os.listdir(class_dir):
            img_path = os.path.join(class_dir, img_name)
            if img_path.endswith('.png') or img_path.endswith('.jpg'):
                # Load and preprocess image
                test_image = image.load_img(img_path, target_size=(64, 64))
                test_image = image.img_to_array(test_image)
                test_image = np.expand_dims(test_image, axis=0)
                test_image /= 255.0

                # Predict
                result = cnn.predict(test_image)
                if result[0][0] > 0.5:
                    y_pred.append(1)  # Assuming 1 corresponds to 'DFU'
                else:
                    y_pred.append(0)  # Assuming 0 corresponds to 'Normal skin'

                # Append true label
                if class_name == 'DMG':  # Replace 'DMG' with the actual class␣
↪directory name for 'DFU'
                    y_true.append(1)
                else:
                    y_true.append(0)

# Generate confusion matrix
cm = confusion_matrix(y_true, y_pred)

# Visualize the confusion matrix
```

```
plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Normal skin',␣
 ↪'DFU'], yticklabels=['Normal skin', 'DFU'])
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()

# Alternatively, using sklearn's ConfusionMatrixDisplay
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Normal␣
 ↪skin', 'DFU'])
disp.plot(cmap='Blues')
plt.show()
```

```
1/1             0s 15ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 7ms/step
1/1             0s 7ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 7ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 10ms/step
1/1             0s 7ms/step
1/1             0s 10ms/step
1/1             0s 7ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
```

```
1/1             0s 7ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 10ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 10ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 7ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 10ms/step
1/1             0s 10ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 10ms/step
1/1             0s 7ms/step
1/1             0s 9ms/step
1/1             0s 7ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 26ms/step
1/1             0s 9ms/step
1/1             0s 7ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
```

```
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 9ms/step
1/1               0s 8ms/step
1/1               0s 10ms/step
1/1               0s 9ms/step
1/1               0s 9ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 9ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 9ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 23ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 9ms/step
1/1               0s 9ms/step
1/1               0s 11ms/step
1/1               0s 11ms/step
1/1               0s 25ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 9ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 27ms/step
1/1               0s 8ms/step
1/1               0s 7ms/step
1/1               0s 9ms/step
1/1               0s 9ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 9ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 11ms/step
```

```
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 14ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 13ms/step
1/1             0s 10ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 10ms/step
1/1             0s 10ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 23ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 24ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
```

```
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 11ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 22ms/step
1/1             0s 15ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 29ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 29ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 27ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 10ms/step
1/1             0s 13ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
```

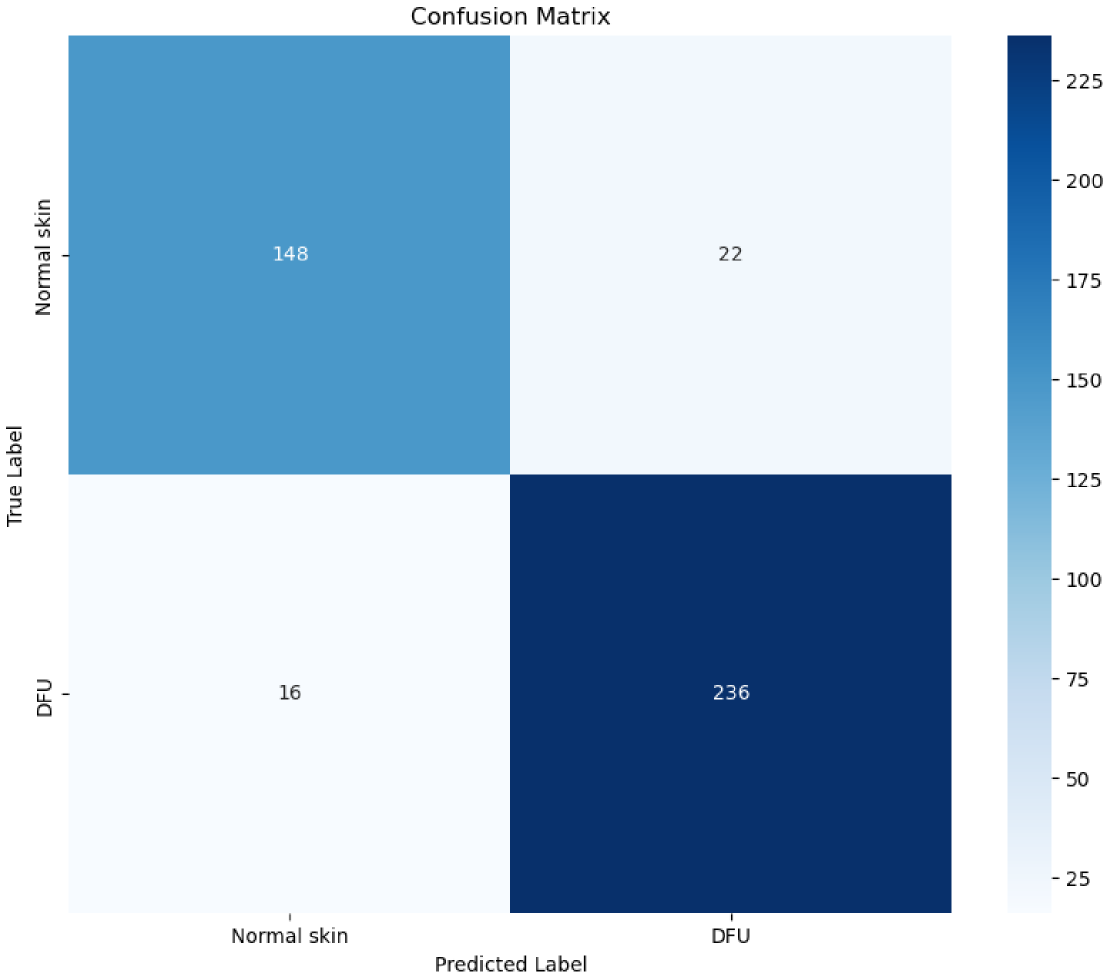
```
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 13ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 15ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 27ms/step
1/1             0s 16ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 14ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 13ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
```

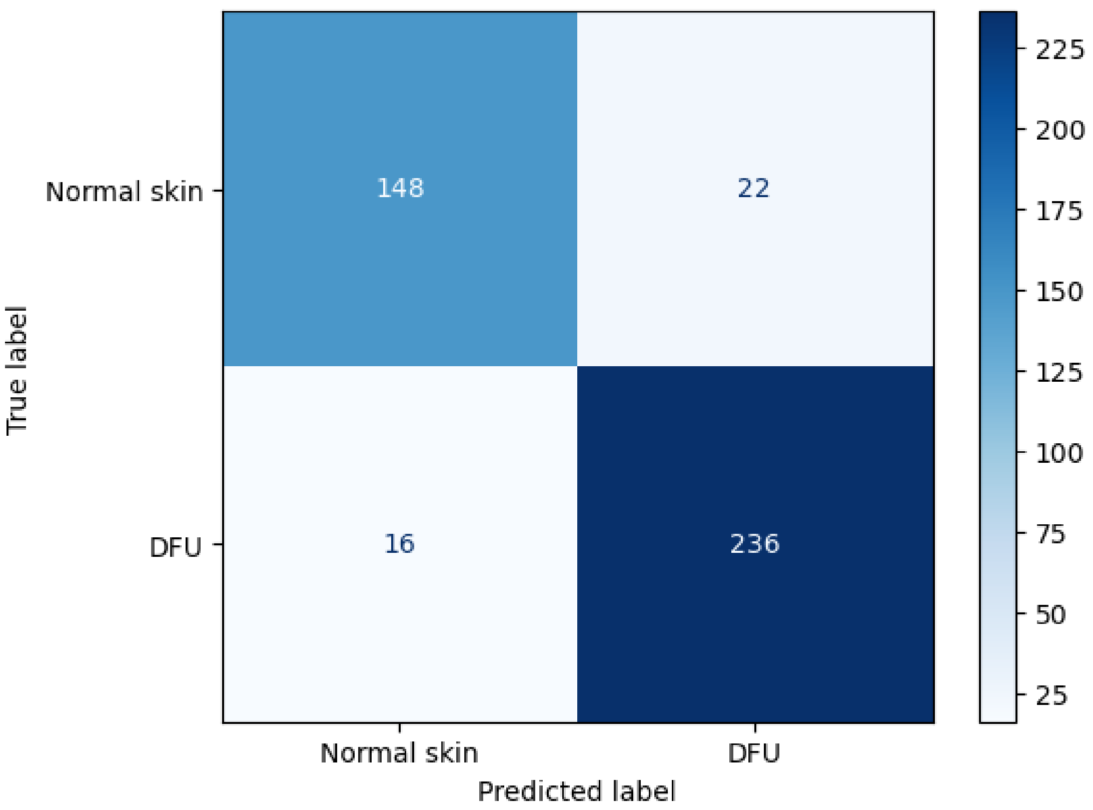
```
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 11ms/step
1/1             0s 9ms/step
1/1             0s 10ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 25ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 33ms/step
1/1             0s 11ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 11ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 10ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
```

```
1/1             0s 8ms/step
1/1             0s 13ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 37ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 11ms/step
1/1             0s 12ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 29ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 29ms/step
1/1             0s 10ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 11ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
```

```
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 12ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 14ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 15ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 9ms/step
1/1             0s 12ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 33ms/step
1/1             0s 9ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 8ms/step
1/1             0s 14ms/step
1/1             0s 8ms/step
```

```
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 8ms/step
1/1               0s 9ms/step
1/1               0s 12ms/step
```




Normal skin
DFU
148
22
16
236
True label
Normal skin
DFU
Predicted label
225
200
175
150
125
100
75
50
25