In [1]:
```
from MLModels import CNN_json as cnn_j
from MLModels import CNN_png as cnn_p
```

In [2]:
```
MFCC_JSON = r"C:\datasets\whalefm_mp3\mfcc_data.json"
model, test_inputs, test_labels, mapping =
cnn_j.run(MFCC_JSON,"mfcc", 32, 30, 0.0001, error_function="sparse_categorical_crossentropy")
```

```
Shapes:
Model input shape: (173, 13)
Training samples shape = (8145, 173, 13, 1)
Validation samples shape = (3492, 173, 13, 1)
Testing samples shape = (3880, 173, 13, 1)

Hyper parameters:
Optimizer: {'name': 'Adam', 'learning_rate': 0.0001, 'decay': 0.0, 'beta_1': 0.9, 'beta_2': 0.99
9, 'epsilon': 1e-07, 'amsgrad': False}
Error function: sparse_categorical_crossentropy
Epochs: 30
Batch size: 32

 Model Summary:
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 171, 11, 32)       320

 max_pooling2d (MaxPooling2D  (None, 86, 6, 32)        0
 )

 batch_normalization (BatchN  (None, 86, 6, 32)        128
 ormalization)

 conv2d_1 (Conv2D)           (None, 85, 5, 32)         4128

 max_pooling2d_1 (MaxPooling  (None, 43, 3, 32)        0
 2D)

 batch_normalization_1 (Batc  (None, 43, 3, 32)        128
 hNormalization)

 flatten (Flatten)           (None, 4128)              0

 dense (Dense)               (None, 64)                264256

 dense_1 (Dense)             (None, 2)                 130

=================================================================
Total params: 269,090
Trainable params: 268,962
Non-trainable params: 128
_________________________________________________________________

 Activation Functions:
Layer name             Activation Function
---------------------  ----------------------------------------
conv2d                 <function relu at 0x0000014787905700>
max_pooling2d          No activation attribute
batch_normalization    No activation attribute
conv2d_1               <function relu at 0x0000014787905700>
max_pooling2d_1        No activation attribute
batch_normalization_1  No activation attribute
flatten                No activation attribute
dense                  <function relu at 0x0000014787905700>
dense_1                <function softmax at 0x00000147878FECA0>

 Epochs:
```

```
Epoch 1/30
255/255 - 4s - loss: 0.1230 - accuracy: 0.9504 - val_loss: 0.1083 - val_accuracy: 0.9605 - 4s/ep
och - 14ms/step
Epoch 2/30
255/255 - 1s - loss: 0.0573 - accuracy: 0.9805 - val_loss: 0.0593 - val_accuracy: 0.9794 - 1s/ep
och - 6ms/step
Epoch 3/30
255/255 - 1s - loss: 0.0447 - accuracy: 0.9831 - val_loss: 0.0661 - val_accuracy: 0.9759 - 1s/ep
och - 5ms/step
Epoch 4/30
255/255 - 1s - loss: 0.0337 - accuracy: 0.9892 - val_loss: 0.0525 - val_accuracy: 0.9808 - 1s/ep
och - 5ms/step
Epoch 5/30
255/255 - 1s - loss: 0.0242 - accuracy: 0.9925 - val_loss: 0.0480 - val_accuracy: 0.9828 - 1s/ep
och - 6ms/step
Epoch 6/30
255/255 - 1s - loss: 0.0192 - accuracy: 0.9942 - val_loss: 0.0495 - val_accuracy: 0.9828 - 1s/ep
och - 5ms/step
Epoch 7/30
255/255 - 1s - loss: 0.0150 - accuracy: 0.9956 - val_loss: 0.0552 - val_accuracy: 0.9820 - 1s/ep
och - 6ms/step
Epoch 8/30
255/255 - 1s - loss: 0.0099 - accuracy: 0.9977 - val_loss: 0.0542 - val_accuracy: 0.9825 - 1s/ep
och - 5ms/step
Epoch 9/30
255/255 - 1s - loss: 0.0076 - accuracy: 0.9984 - val_loss: 0.0500 - val_accuracy: 0.9837 - 1s/ep
och - 5ms/step
Epoch 10/30
255/255 - 1s - loss: 0.0055 - accuracy: 0.9993 - val_loss: 0.0501 - val_accuracy: 0.9842 - 1s/ep
och - 5ms/step
Epoch 11/30
255/255 - 1s - loss: 0.0042 - accuracy: 0.9995 - val_loss: 0.0502 - val_accuracy: 0.9834 - 1s/ep
och - 5ms/step
Epoch 12/30
255/255 - 1s - loss: 0.0029 - accuracy: 0.9999 - val_loss: 0.0501 - val_accuracy: 0.9860 - 1s/ep
och - 5ms/step
Epoch 13/30
255/255 - 1s - loss: 0.0024 - accuracy: 0.9998 - val_loss: 0.0587 - val_accuracy: 0.9842 - 1s/ep
och - 6ms/step
Epoch 14/30
255/255 - 1s - loss: 0.0016 - accuracy: 1.0000 - val_loss: 0.0557 - val_accuracy: 0.9854 - 1s/ep
och - 5ms/step
Epoch 15/30
255/255 - 1s - loss: 0.0013 - accuracy: 1.0000 - val_loss: 0.0564 - val_accuracy: 0.9848 - 1s/ep
och - 5ms/step
Epoch 16/30
255/255 - 1s - loss: 0.0012 - accuracy: 1.0000 - val_loss: 0.0551 - val_accuracy: 0.9851 - 1s/ep
och - 5ms/step
Epoch 17/30
255/255 - 1s - loss: 8.9144e-04 - accuracy: 1.0000 - val_loss: 0.0641 - val_accuracy: 0.9845 - 1
s/epoch - 5ms/step
Epoch 18/30
255/255 - 1s - loss: 7.8007e-04 - accuracy: 1.0000 - val_loss: 0.0572 - val_accuracy: 0.9854 - 1
s/epoch - 5ms/step
Epoch 19/30
255/255 - 1s - loss: 9.0150e-04 - accuracy: 0.9999 - val_loss: 0.0597 - val_accuracy: 0.9840 - 1
s/epoch - 5ms/step
Epoch 20/30
255/255 - 1s - loss: 7.9607e-04 - accuracy: 1.0000 - val_loss: 0.0598 - val_accuracy: 0.9842 - 1
s/epoch - 6ms/step
Epoch 21/30
255/255 - 1s - loss: 4.6678e-04 - accuracy: 1.0000 - val_loss: 0.0594 - val_accuracy: 0.9848 - 1
```

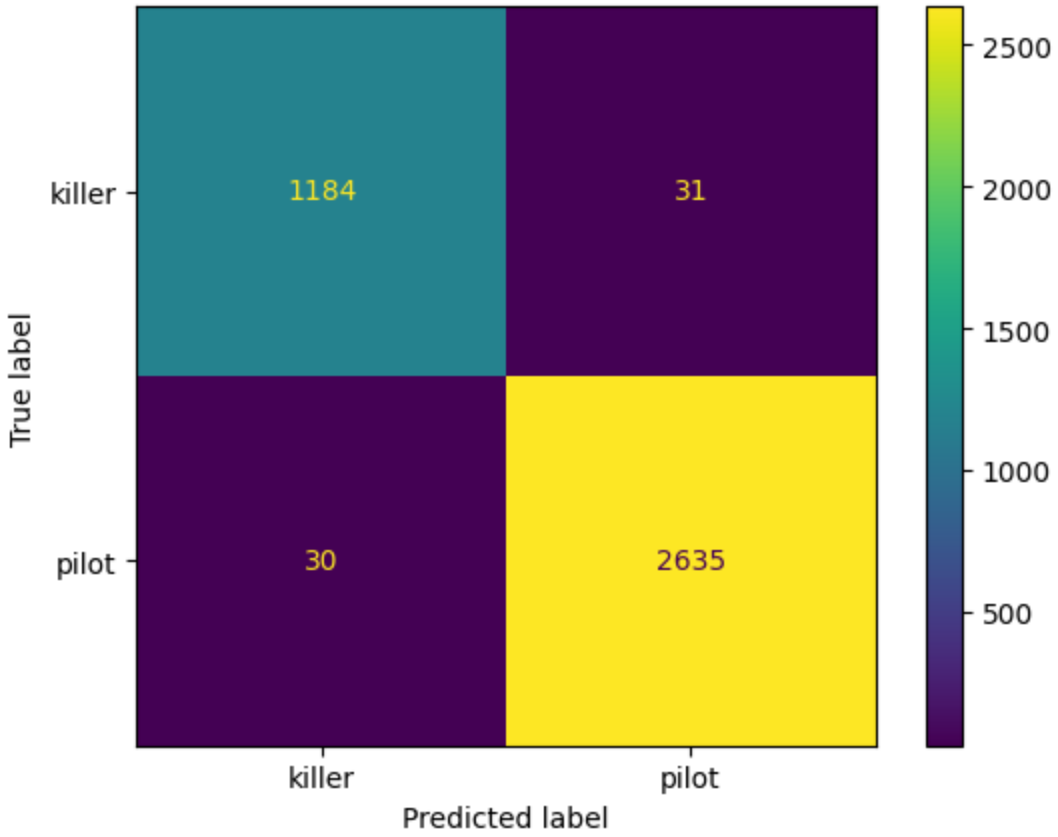
```
s/epoch - 5ms/step
Epoch 22/30
255/255 - 1s - loss: 4.6539e-04 - accuracy: 1.0000 - val_loss: 0.0727 - val_accuracy: 0.9831 - 1
s/epoch - 6ms/step
Epoch 23/30
255/255 - 1s - loss: 4.9138e-04 - accuracy: 1.0000 - val_loss: 0.0655 - val_accuracy: 0.9842 - 1
s/epoch - 5ms/step
Epoch 24/30
255/255 - 1s - loss: 2.1693e-04 - accuracy: 1.0000 - val_loss: 0.0655 - val_accuracy: 0.9860 - 1
s/epoch - 6ms/step
Epoch 25/30
255/255 - 1s - loss: 1.5569e-04 - accuracy: 1.0000 - val_loss: 0.0637 - val_accuracy: 0.9865 - 1
s/epoch - 6ms/step
Epoch 26/30
255/255 - 1s - loss: 1.7135e-04 - accuracy: 1.0000 - val_loss: 0.0658 - val_accuracy: 0.9845 - 1
s/epoch - 6ms/step
Epoch 27/30
255/255 - 1s - loss: 1.8676e-04 - accuracy: 1.0000 - val_loss: 0.0668 - val_accuracy: 0.9851 - 1
s/epoch - 6ms/step
Epoch 28/30
255/255 - 1s - loss: 1.1624e-04 - accuracy: 1.0000 - val_loss: 0.0689 - val_accuracy: 0.9854 - 1
s/epoch - 6ms/step
Epoch 29/30
255/255 - 1s - loss: 4.1221e-04 - accuracy: 1.0000 - val_loss: 0.0976 - val_accuracy: 0.9805 - 1
s/epoch - 5ms/step
Epoch 30/30
255/255 - 1s - loss: 0.0181 - accuracy: 0.9958 - val_loss: 0.0633 - val_accuracy: 0.9840 - 1s/ep
och - 5ms/step

 Final test results:
Accuracy on test set is 0.9842783212661743
Error on test set it 0.0554669052362442
```

In [3]:
```
predictions = model.predict(test_inputs)
```

```
122/122 [==============================] - 0s 2ms/step
```

In [4]:
```
import numpy as np
predictions = np.argmax(predictions, axis=-1)
```

In [6]:
```
cnn_j.get_confusion_matrix(predictions=predictions, test_labels=test_labels, display_labels=mapp
```


killer
pilot
True label
1184
31
30
2635
killer
pilot
Predicted label
2500
2000
1500
1000
500