

CNN Optuna Mels

June 9, 2022

0.0.1 Audio Load by Barney

```
[1]: from barney_functions import *
import librosa
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import random
import re
import soundfile as sf
import itertools
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split

[2]: mels = np.load('mels.npz')['data']
y = np.load('mels_labels.npz', allow_pickle=True)['data']
num_classes = len(np.unique(y))
mels, TEST_images, y, TEST_y = train_test_split(mels, y, test_size=0.2,
↪random_state=42)

[3]: print(mels.shape)
print(y.shape)
print(TEST_images.shape)
print(TEST_y.shape)
```

```
(1804, 128, 51)
(1804,)
(452, 128, 51)
(452,)
```

1 CNN

```
[4]: import keras
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,
↪Dropout
```

```
from tensorflow.keras.losses import categorical_crossentropy
from tensorflow.keras.optimizers import Adadelta, Adam
from optuna.integration import TFKerasPruningCallback
tf.compat.v1.logging.set_verbosity(tf.compat.v1.logging.ERROR)
```

```
/Users/barnabyemmens/opt/anaconda3/envs/appml/lib/python3.8/site-
packages/tqdm/auto.py:22: TqdmWarning: IPProgress not found. Please update
jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
    from .autonotebook import tqdm as notebook_tqdm
```

```
[5]: from sklearn.model_selection import KFold
```

```
[6]: import optuna
```

```
[7]: print("Num CPUs Available: ", len(tf.config.list_physical_devices('CPU')))
```

Num CPUs Available: 1

2 Optimising

```
[8]: def objective(trial):
    n_epochs = trial.suggest_int("n_epochs", low=5, high=20, step=1)
    learning_rate = trial.suggest_float("learning_rate", 1e-5, 1e-2)
    optimiser = Adam(learning_rate=learning_rate)

    filters_1 = trial.suggest_int("filters_1", low=2, high=32, step=2)
    kernel_1 = trial.suggest_int("kernel_size_1", low=8, high=16, step=4)
    strides_1 = 1
    padding_1 = 'same'
    activation_1 = 'relu'
    input_shape = (img_rows, img_cols, 1)
    pool_size_1 = 2

    filters_2 = trial.suggest_int("filters_2", low=2, high=64, step=2)
    kernel_2 = trial.suggest_int("kernel_size_2", low=4, high=12, step=3)
    strides_2 = 1
    padding_2 = 'same'
    activation_2 = 'relu'
    pool_size_2 = 2

    dropout_rate = 0.4

    lin_1_n = trial.suggest_int("lin_1_n", low=4, high=128, step=1)

    batch_size = 128
```

```

model = Sequential()
model.add(Conv2D(filters=filters_1,
                 kernel_size=kernel_1,
                 strides=strides_1,
                 padding=padding_1,
                 activation=activation_1,
                 input_shape=input_shape))
model.add(MaxPooling2D(pool_size=pool_size_1, strides=None))
model.add(Conv2D(filters=filters_2,
                 kernel_size=kernel_2,
                 strides=strides_2,
                 padding=padding_2,
                 activation=activation_2))
model.add(MaxPooling2D(pool_size=pool_size_2, strides=None))
model.add(Dropout(rate=dropout_rate))
model.add(Flatten())
model.add(Dense(units=lin_1_n, activation='relu'))
model.add(Dense(units=num_classes, activation='softmax'))

model.compile(loss=categorical_crossentropy,
              optimizer=optimiser,
              metrics=['categorical_accuracy'])

monitor = "val_categorical_accuracy"

callbacks = [
    tf.keras.callbacks.EarlyStopping(patience=5)]
    #TFKerasPruningCallback(trial, monitor)]

fitted = model.fit(X_train, y_train,
                  batch_size=batch_size,
                  epochs=n_epochs,
                  verbose=1,
                  validation_data=(X_test, y_test),
                  callbacks=callbacks)

score = fitted.history[monitor][-1]

return score
# changed trials to 50 from 10
# removed pruning but not callbacks

```

```

[9]: train_images, val_images, train_labels, val_labels = train_test_split(mels, y,
    ↪test_size=0.2, random_state=42)
X_train, X_test, y_train, y_test, img_rows, img_cols =
    ↪format_for_CNN(train_images, val_images, train_labels, val_labels)
check_class_complete_gen(train_labels, val_labels,y)

```

OK

```
[10]: study = optuna.create_study(direction='maximize')
      study.optimize(objective, n_trials=10)
```

```
[I 2022-06-09 18:20:38,178] A new study created in memory with name:
no-name-f438c8df-005c-4456-8ec5-cec9ab9a8475
```

Epoch 1/7

```
/Users/barnabyemmens/opt/anaconda3/envs/appml/lib/python3.8/site-
packages/optuna/distributions.py:560: UserWarning: The distribution is specified
by [4, 12] and step=3, but the range is not divisible by `step`. It will be
replaced by [4, 10].
```

```
warnings.warn(
2022-06-09 18:20:38.187298: I tensorflow/core/platform/cpu_feature_guard.cc:151]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations: SSE4.1 SSE4.2 AVX AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
```

```
12/12 [=====] - 11s 850ms/step - loss: 9.2160 -
categorical_accuracy: 0.3999 - val_loss: 1.3816 - val_categorical_accuracy:
0.3961
```

Epoch 2/7

```
12/12 [=====] - 10s 872ms/step - loss: 1.3711 -
categorical_accuracy: 0.4297 - val_loss: 1.3646 - val_categorical_accuracy:
0.3961
```

Epoch 3/7

```
12/12 [=====] - 11s 879ms/step - loss: 1.3489 -
categorical_accuracy: 0.4297 - val_loss: 1.3490 - val_categorical_accuracy:
0.3961
```

Epoch 4/7

```
12/12 [=====] - 11s 881ms/step - loss: 1.3303 -
categorical_accuracy: 0.4297 - val_loss: 1.3369 - val_categorical_accuracy:
0.3961
```

Epoch 5/7

```
12/12 [=====] - 11s 890ms/step - loss: 1.3169 -
categorical_accuracy: 0.4297 - val_loss: 1.3285 - val_categorical_accuracy:
0.3961
```

Epoch 6/7

```
12/12 [=====] - 11s 895ms/step - loss: 1.3070 -
categorical_accuracy: 0.4297 - val_loss: 1.3242 - val_categorical_accuracy:
0.3961
```

Epoch 7/7

```
12/12 [=====] - 11s 901ms/step - loss: 1.3016 -
categorical_accuracy: 0.4297 - val_loss: 1.3217 - val_categorical_accuracy:
0.3961
```

[I 2022-06-09 18:21:52,821] Trial 0 finished with value: 0.39612188935279846 and parameters: {'n_epochs': 7, 'learning_rate': 0.00669, 'filters_1': 18, 'kernel_size_1': 12, 'filters_2': 32, 'kernel_size_2': 10, 'lin_1_n': 34}. Best is trial 0 with value: 0.39612188935279846.

Epoch 1/14

/Users/barnabyemmens/opt/anaconda3/envs/appml/lib/python3.8/site-packages/optuna/distributions.py:560: UserWarning: The distribution is specified by [4, 12] and step=3, but the range is not divisible by `step`. It will be replaced by [4, 10].

warnings.warn(

12/12 [=====] - 9s 751ms/step - loss: 1.3308 - categorical_accuracy: 0.3985 - val_loss: 1.2911 - val_categorical_accuracy: 0.4211

Epoch 2/14

12/12 [=====] - 9s 748ms/step - loss: 1.1884 - categorical_accuracy: 0.5080 - val_loss: 1.1514 - val_categorical_accuracy: 0.4460

Epoch 3/14

12/12 [=====] - 9s 746ms/step - loss: 1.0313 - categorical_accuracy: 0.5523 - val_loss: 1.0116 - val_categorical_accuracy: 0.5374

Epoch 4/14

12/12 [=====] - 9s 742ms/step - loss: 0.9106 - categorical_accuracy: 0.5683 - val_loss: 0.9242 - val_categorical_accuracy: 0.5014

Epoch 5/14

12/12 [=====] - 9s 733ms/step - loss: 0.8216 - categorical_accuracy: 0.6133 - val_loss: 0.8422 - val_categorical_accuracy: 0.6288

Epoch 6/14

12/12 [=====] - 9s 735ms/step - loss: 0.7615 - categorical_accuracy: 0.6584 - val_loss: 0.7908 - val_categorical_accuracy: 0.6288

Epoch 7/14

12/12 [=====] - 9s 732ms/step - loss: 0.7181 - categorical_accuracy: 0.6888 - val_loss: 0.7350 - val_categorical_accuracy: 0.6842

Epoch 8/14

12/12 [=====] - 9s 740ms/step - loss: 0.6804 - categorical_accuracy: 0.7082 - val_loss: 0.7050 - val_categorical_accuracy: 0.6814

Epoch 9/14

12/12 [=====] - 9s 728ms/step - loss: 0.6646 - categorical_accuracy: 0.7152 - val_loss: 0.6839 - val_categorical_accuracy: 0.6759

Epoch 10/14

categorical_accuracy: 0.6500 - val_loss: 0.6886 - val_categorical_accuracy: 0.6316

Epoch 6/10

12/12 [=====] - 14s 1s/step - loss: 0.5749 - categorical_accuracy: 0.7270 - val_loss: 0.5878 - val_categorical_accuracy: 0.7368

Epoch 7/10

12/12 [=====] - 15s 1s/step - loss: 0.4887 - categorical_accuracy: 0.7824 - val_loss: 0.5057 - val_categorical_accuracy: 0.8144

Epoch 8/10

12/12 [=====] - 14s 1s/step - loss: 0.4111 - categorical_accuracy: 0.8503 - val_loss: 0.4768 - val_categorical_accuracy: 0.7950

Epoch 9/10

12/12 [=====] - 14s 1s/step - loss: 0.3806 - categorical_accuracy: 0.8455 - val_loss: 0.3357 - val_categorical_accuracy: 0.8726

Epoch 10/10

12/12 [=====] - 14s 1s/step - loss: 0.2838 - categorical_accuracy: 0.8933 - val_loss: 0.3127 - val_categorical_accuracy: 0.8781

[I 2022-06-09 18:26:17,458] Trial 2 finished with value:

0.8781163692474365 and parameters: {'n_epochs': 10, 'learning_rate': 0.00491, 'filters_1': 6, 'kernel_size_1': 16, 'filters_2': 38, 'kernel_size_2': 10, 'lin_1_n': 6}. Best is trial 2 with value: 0.8781163692474365.

Epoch 1/19

/Users/barnabyemmens/opt/anaconda3/envs/appml/lib/python3.8/site-packages/optuna/distributions.py:560: UserWarning: The distribution is specified by [4, 12] and step=3, but the range is not divisible by `step`. It will be replaced by [4, 10].

warnings.warn(

12/12 [=====] - 9s 754ms/step - loss: 9.3184 - categorical_accuracy: 0.2973 - val_loss: 1.3800 - val_categorical_accuracy: 0.3961

Epoch 2/19

12/12 [=====] - 9s 749ms/step - loss: 1.3733 - categorical_accuracy: 0.4297 - val_loss: 1.3646 - val_categorical_accuracy: 0.3961

Epoch 3/19

12/12 [=====] - 9s 768ms/step - loss: 1.3531 - categorical_accuracy: 0.4297 - val_loss: 1.3494 - val_categorical_accuracy: 0.3961

Epoch 4/19

12/12 [=====] - 9s 762ms/step - loss: 1.3361 - categorical_accuracy: 0.4297 - val_loss: 1.3372 - val_categorical_accuracy:

```

0.3961
Epoch 5/19
12/12 [=====] - 9s 749ms/step - loss: 1.3222 -
categorical_accuracy: 0.4297 - val_loss: 1.3290 - val_categorical_accuracy:
0.3961
Epoch 6/19
12/12 [=====] - 9s 750ms/step - loss: 1.3121 -
categorical_accuracy: 0.4297 - val_loss: 1.3236 - val_categorical_accuracy:
0.3961
Epoch 7/19
12/12 [=====] - 9s 747ms/step - loss: 1.3048 -
categorical_accuracy: 0.4297 - val_loss: 1.3206 - val_categorical_accuracy:
0.3961
Epoch 8/19
12/12 [=====] - 9s 744ms/step - loss: 1.2998 -
categorical_accuracy: 0.4297 - val_loss: 1.3191 - val_categorical_accuracy:
0.3961
Epoch 9/19
12/12 [=====] - 9s 741ms/step - loss: 1.2970 -
categorical_accuracy: 0.4297 - val_loss: 1.3188 - val_categorical_accuracy:
0.3961
Epoch 10/19
12/12 [=====] - 9s 745ms/step - loss: 1.2949 -
categorical_accuracy: 0.4297 - val_loss: 1.3193 - val_categorical_accuracy:
0.3961
Epoch 11/19
12/12 [=====] - 9s 746ms/step - loss: 1.2939 -
categorical_accuracy: 0.4297 - val_loss: 1.3201 - val_categorical_accuracy:
0.3961
Epoch 12/19
12/12 [=====] - 9s 758ms/step - loss: 1.2935 -
categorical_accuracy: 0.4297 - val_loss: 1.3213 - val_categorical_accuracy:
0.3961
Epoch 13/19
12/12 [=====] - 9s 749ms/step - loss: 1.2932 -
categorical_accuracy: 0.4297 - val_loss: 1.3222 - val_categorical_accuracy:
0.3961
Epoch 14/19
12/12 [=====] - 9s 743ms/step - loss: 1.2931 -
categorical_accuracy: 0.4297 - val_loss: 1.3234 - val_categorical_accuracy:
0.3961

[I 2022-06-09 18:28:24,187] Trial 3 finished with value:
0.39612188935279846 and parameters: {'n_epochs': 19, 'learning_rate': 0.00689,
'filters_1': 30, 'kernel_size_1': 12, 'filters_2': 58, 'kernel_size_2': 4,
'lin_1_n': 92}. Best is trial 2 with value: 0.8781163692474365.
/Users/barnabyemmens/opt/anaconda3/envs/appml/lib/python3.8/site-
packages/optuna/distributions.py:560: UserWarning: The distribution is specified

```


by [4, 12] and step=3, but the range is not divisible by `step`. It will be replaced by [4, 10].

```
warnings.warn(
```

Epoch 1/11

12/12 [=====] - 5s 366ms/step - loss: 1.3624 - categorical_accuracy: 0.3340 - val_loss: 1.1162 - val_categorical_accuracy: 0.4903

Epoch 2/11

12/12 [=====] - 4s 362ms/step - loss: 0.9674 - categorical_accuracy: 0.5703 - val_loss: 0.9668 - val_categorical_accuracy: 0.5097

Epoch 3/11

12/12 [=====] - 4s 366ms/step - loss: 0.8596 - categorical_accuracy: 0.5946 - val_loss: 0.8822 - val_categorical_accuracy: 0.5402

Epoch 4/11

12/12 [=====] - 4s 362ms/step - loss: 0.7517 - categorical_accuracy: 0.6341 - val_loss: 0.7883 - val_categorical_accuracy: 0.6593

Epoch 5/11

12/12 [=====] - 4s 362ms/step - loss: 0.6735 - categorical_accuracy: 0.7228 - val_loss: 0.7408 - val_categorical_accuracy: 0.6593

Epoch 6/11

12/12 [=====] - 4s 369ms/step - loss: 0.6326 - categorical_accuracy: 0.7283 - val_loss: 0.6751 - val_categorical_accuracy: 0.6925

Epoch 7/11

12/12 [=====] - 4s 363ms/step - loss: 0.6100 - categorical_accuracy: 0.7408 - val_loss: 0.6786 - val_categorical_accuracy: 0.6704

Epoch 8/11

12/12 [=====] - 4s 369ms/step - loss: 0.5679 - categorical_accuracy: 0.7547 - val_loss: 0.6394 - val_categorical_accuracy: 0.6898

Epoch 9/11

12/12 [=====] - 5s 380ms/step - loss: 0.5404 - categorical_accuracy: 0.7762 - val_loss: 0.5845 - val_categorical_accuracy: 0.7091

Epoch 10/11

12/12 [=====] - 5s 384ms/step - loss: 0.4971 - categorical_accuracy: 0.7942 - val_loss: 0.5375 - val_categorical_accuracy: 0.7285

Epoch 11/11

12/12 [=====] - 4s 370ms/step - loss: 0.4400 - categorical_accuracy: 0.8136 - val_loss: 0.4960 - val_categorical_accuracy: 0.7756

```

[I 2022-06-09 18:29:46,829] Trial 4 finished with value:
0.7756232619285583 and parameters: {'n_epochs': 11, 'learning_rate':
0.0024100000000000002, 'filters_1': 2, 'kernel_size_1': 8, 'filters_2': 60,
'kernel_size_2': 10, 'lin_1_n': 11}. Best is trial 2 with value:
0.8781163692474365.
/Users/barnabyemmens/opt/anaconda3/envs/appml/lib/python3.8/site-
packages/optuna/distributions.py:560: UserWarning: The distribution is specified
by [4, 12] and step=3, but the range is not divisible by `step`. It will be
replaced by [4, 10].
    warnings.warn(

Epoch 1/10
12/12 [=====] - 13s 1s/step - loss: 1.0001 -
categorical_accuracy: 0.5433 - val_loss: 0.7320 - val_categorical_accuracy:
0.6648
Epoch 2/10
12/12 [=====] - 13s 1s/step - loss: 0.6056 -
categorical_accuracy: 0.7387 - val_loss: 0.5989 - val_categorical_accuracy:
0.7535
Epoch 3/10
12/12 [=====] - 13s 1s/step - loss: 0.5193 -
categorical_accuracy: 0.7782 - val_loss: 0.5194 - val_categorical_accuracy:
0.7645
Epoch 4/10
12/12 [=====] - 13s 1s/step - loss: 0.4631 -
categorical_accuracy: 0.7970 - val_loss: 0.4789 - val_categorical_accuracy:
0.7867
Epoch 5/10
12/12 [=====] - 13s 1s/step - loss: 0.4067 -
categorical_accuracy: 0.8191 - val_loss: 0.4549 - val_categorical_accuracy:
0.8061
Epoch 6/10
12/12 [=====] - 13s 1s/step - loss: 0.3787 -
categorical_accuracy: 0.8358 - val_loss: 0.4015 - val_categorical_accuracy:
0.8089
Epoch 7/10
12/12 [=====] - 13s 1s/step - loss: 0.3152 -
categorical_accuracy: 0.8739 - val_loss: 0.5724 - val_categorical_accuracy:
0.6925
Epoch 8/10
12/12 [=====] - 13s 1s/step - loss: 0.3392 -
categorical_accuracy: 0.8441 - val_loss: 0.4809 - val_categorical_accuracy:
0.7895
Epoch 9/10
12/12 [=====] - 12s 1s/step - loss: 0.3210 -
categorical_accuracy: 0.8753 - val_loss: 0.3354 - val_categorical_accuracy:
0.8698
Epoch 10/10

```

12/12 [=====] - 12s 1s/step - loss: 0.2541 -
categorical_accuracy: 0.8940 - val_loss: 0.3269 - val_categorical_accuracy:
0.8615

[I 2022-06-09 18:31:54,741] Trial 5 finished with value:
0.861495852470398 and parameters: {'n_epochs': 10, 'learning_rate':
0.00043000000000000004, 'filters_1': 20, 'kernel_size_1': 16, 'filters_2': 22,
'kernel_size_2': 7, 'lin_1_n': 56}. Best is trial 2 with value:
0.8781163692474365.

Epoch 1/15

/Users/barnabyemmens/opt/anaconda3/envs/appml/lib/python3.8/site-
packages/optuna/distributions.py:560: UserWarning: The distribution is specified
by [4, 12] and step=3, but the range is not divisible by `step`. It will be
replaced by [4, 10].

warnings.warn(

12/12 [=====] - 10s 779ms/step - loss: 1.5117 -
categorical_accuracy: 0.4297 - val_loss: 1.1960 - val_categorical_accuracy:
0.4737

Epoch 2/15

12/12 [=====] - 9s 786ms/step - loss: 1.1315 -
categorical_accuracy: 0.5364 - val_loss: 1.0769 - val_categorical_accuracy:
0.5152

Epoch 3/15

12/12 [=====] - 9s 789ms/step - loss: 0.9548 -
categorical_accuracy: 0.6098 - val_loss: 0.9857 - val_categorical_accuracy:
0.6620

Epoch 4/15

12/12 [=====] - 9s 760ms/step - loss: 0.8738 -
categorical_accuracy: 0.6826 - val_loss: 0.8174 - val_categorical_accuracy:
0.7618

Epoch 5/15

12/12 [=====] - 9s 774ms/step - loss: 0.7410 -
categorical_accuracy: 0.7554 - val_loss: 0.7464 - val_categorical_accuracy:
0.7784

Epoch 6/15

12/12 [=====] - 9s 780ms/step - loss: 0.6842 -
categorical_accuracy: 0.7741 - val_loss: 0.6784 - val_categorical_accuracy:
0.8172

Epoch 7/15

12/12 [=====] - 9s 776ms/step - loss: 0.5854 -
categorical_accuracy: 0.8073 - val_loss: 0.6184 - val_categorical_accuracy:
0.7950

Epoch 8/15

12/12 [=====] - 9s 766ms/step - loss: 0.5332 -
categorical_accuracy: 0.8427 - val_loss: 0.6325 - val_categorical_accuracy:
0.8033

Epoch 9/15

12/12 [=====] - 9s 756ms/step - loss: 0.5150 -
categorical_accuracy: 0.8385 - val_loss: 0.5490 - val_categorical_accuracy:
0.8449

Epoch 10/15

12/12 [=====] - 9s 760ms/step - loss: 0.4576 -
categorical_accuracy: 0.8711 - val_loss: 0.5276 - val_categorical_accuracy:
0.8449

Epoch 11/15

12/12 [=====] - 9s 762ms/step - loss: 0.4607 -
categorical_accuracy: 0.8586 - val_loss: 0.4963 - val_categorical_accuracy:
0.8532

Epoch 12/15

12/12 [=====] - 9s 762ms/step - loss: 0.4303 -
categorical_accuracy: 0.8780 - val_loss: 0.4416 - val_categorical_accuracy:
0.8809

Epoch 13/15

12/12 [=====] - 9s 762ms/step - loss: 0.3796 -
categorical_accuracy: 0.8995 - val_loss: 0.4017 - val_categorical_accuracy:
0.9086

Epoch 14/15

12/12 [=====] - 9s 763ms/step - loss: 0.3548 -
categorical_accuracy: 0.9168 - val_loss: 0.4065 - val_categorical_accuracy:
0.9141

Epoch 15/15

12/12 [=====] - 9s 760ms/step - loss: 0.3508 -
categorical_accuracy: 0.9113 - val_loss: 0.3866 - val_categorical_accuracy:
0.8892

[I 2022-06-09 18:34:13,745] Trial 6 finished with value:
0.8891966938972473 and parameters: {'n_epochs': 15, 'learning_rate': 0.00408,
'filters_1': 18, 'kernel_size_1': 12, 'filters_2': 56, 'kernel_size_2': 7,
'lin_1_n': 5}. Best is trial 6 with value: 0.8891966938972473.

Epoch 1/17

/Users/barnabyemmens/opt/anaconda3/envs/appml/lib/python3.8/site-
packages/optuna/distributions.py:560: UserWarning: The distribution is specified
by [4, 12] and step=3, but the range is not divisible by `step`. It will be
replaced by [4, 10].

warnings.warn(

12/12 [=====] - 8s 630ms/step - loss: 1.3443 -
categorical_accuracy: 0.5246 - val_loss: 1.0226 - val_categorical_accuracy:
0.5623

Epoch 2/17

12/12 [=====] - 7s 619ms/step - loss: 0.6765 -
categorical_accuracy: 0.7006 - val_loss: 0.6573 - val_categorical_accuracy:
0.6953

Epoch 3/17

12/12 [=====] - 8s 626ms/step - loss: 0.5082 -

categorical_accuracy: 0.7852 - val_loss: 0.5032 - val_categorical_accuracy: 0.7812

Epoch 4/17

12/12 [=====] - 7s 609ms/step - loss: 0.4097 - categorical_accuracy: 0.8219 - val_loss: 0.4172 - val_categorical_accuracy: 0.7978

Epoch 5/17

12/12 [=====] - 7s 610ms/step - loss: 0.3146 - categorical_accuracy: 0.8676 - val_loss: 0.2977 - val_categorical_accuracy: 0.8947

Epoch 6/17

12/12 [=====] - 7s 605ms/step - loss: 0.2327 - categorical_accuracy: 0.9085 - val_loss: 0.2047 - val_categorical_accuracy: 0.9252

Epoch 7/17

12/12 [=====] - 7s 610ms/step - loss: 0.2434 - categorical_accuracy: 0.9099 - val_loss: 0.1735 - val_categorical_accuracy: 0.9307

Epoch 8/17

12/12 [=====] - 7s 607ms/step - loss: 0.2435 - categorical_accuracy: 0.9085 - val_loss: 0.2149 - val_categorical_accuracy: 0.9335

Epoch 9/17

12/12 [=====] - 7s 606ms/step - loss: 0.1550 - categorical_accuracy: 0.9411 - val_loss: 0.1979 - val_categorical_accuracy: 0.9003

Epoch 10/17

12/12 [=====] - 7s 606ms/step - loss: 0.1173 - categorical_accuracy: 0.9584 - val_loss: 0.1083 - val_categorical_accuracy: 0.9529

Epoch 11/17

12/12 [=====] - 7s 605ms/step - loss: 0.0759 - categorical_accuracy: 0.9757 - val_loss: 0.1575 - val_categorical_accuracy: 0.9335

Epoch 12/17

12/12 [=====] - 7s 608ms/step - loss: 0.0779 - categorical_accuracy: 0.9688 - val_loss: 0.1161 - val_categorical_accuracy: 0.9640

Epoch 13/17

12/12 [=====] - 7s 606ms/step - loss: 0.0607 - categorical_accuracy: 0.9792 - val_loss: 0.0839 - val_categorical_accuracy: 0.9584

Epoch 14/17

12/12 [=====] - 7s 601ms/step - loss: 0.0464 - categorical_accuracy: 0.9896 - val_loss: 0.0818 - val_categorical_accuracy: 0.9751

Epoch 15/17

12/12 [=====] - 7s 606ms/step - loss: 0.0324 -

categorical_accuracy: 0.9938 - val_loss: 0.1055 - val_categorical_accuracy: 0.9557

Epoch 16/17

12/12 [=====] - 7s 603ms/step - loss: 0.0285 - categorical_accuracy: 0.9924 - val_loss: 0.0889 - val_categorical_accuracy: 0.9612

Epoch 17/17

12/12 [=====] - 7s 606ms/step - loss: 0.0227 - categorical_accuracy: 0.9945 - val_loss: 0.1178 - val_categorical_accuracy: 0.9501

[I 2022-06-09 18:36:18,849] Trial 7 finished with value: 0.950138509273529 and parameters: {'n_epochs': 17, 'learning_rate': 0.0022900000000000004, 'filters_1': 24, 'kernel_size_1': 12, 'filters_2': 54, 'kernel_size_2': 4, 'lin_1_n': 82}. Best is trial 7 with value: 0.950138509273529.

Epoch 1/17

/Users/barnabyemmens/opt/anaconda3/envs/appml/lib/python3.8/site-packages/optuna/distributions.py:560: UserWarning: The distribution is specified by [4, 12] and step=3, but the range is not divisible by `step`. It will be replaced by [4, 10].

warnings.warn(

12/12 [=====] - 13s 1s/step - loss: 9.9000 - categorical_accuracy: 0.3680 - val_loss: 0.9910 - val_categorical_accuracy: 0.5900

Epoch 2/17

12/12 [=====] - 12s 1s/step - loss: 0.8181 - categorical_accuracy: 0.6584 - val_loss: 0.7181 - val_categorical_accuracy: 0.6787

Epoch 3/17

12/12 [=====] - 13s 1s/step - loss: 0.6517 - categorical_accuracy: 0.7242 - val_loss: 0.6193 - val_categorical_accuracy: 0.7202

Epoch 4/17

12/12 [=====] - 13s 1s/step - loss: 0.5306 - categorical_accuracy: 0.7644 - val_loss: 0.5233 - val_categorical_accuracy: 0.7867

Epoch 5/17

12/12 [=====] - 12s 1s/step - loss: 0.4814 - categorical_accuracy: 0.8004 - val_loss: 0.4526 - val_categorical_accuracy: 0.8061

Epoch 6/17

12/12 [=====] - 13s 1s/step - loss: 0.4238 - categorical_accuracy: 0.8011 - val_loss: 0.4118 - val_categorical_accuracy: 0.8366

Epoch 7/17

12/12 [=====] - 12s 1s/step - loss: 0.3565 -

categorical_accuracy: 0.8517 - val_loss: 0.3642 - val_categorical_accuracy: 0.8283

Epoch 8/17

12/12 [=====] - 13s 1s/step - loss: 0.3390 - categorical_accuracy: 0.8607 - val_loss: 0.3227 - val_categorical_accuracy: 0.8670

Epoch 9/17

12/12 [=====] - 14s 1s/step - loss: 0.3160 - categorical_accuracy: 0.8746 - val_loss: 0.2697 - val_categorical_accuracy: 0.8975

Epoch 10/17

12/12 [=====] - 13s 1s/step - loss: 0.2442 - categorical_accuracy: 0.9064 - val_loss: 0.2469 - val_categorical_accuracy: 0.9003

Epoch 11/17

12/12 [=====] - 13s 1s/step - loss: 0.2340 - categorical_accuracy: 0.9037 - val_loss: 0.1742 - val_categorical_accuracy: 0.9335

Epoch 12/17

12/12 [=====] - 13s 1s/step - loss: 0.2025 - categorical_accuracy: 0.9210 - val_loss: 0.2468 - val_categorical_accuracy: 0.9114

Epoch 13/17

12/12 [=====] - 13s 1s/step - loss: 0.1950 - categorical_accuracy: 0.9231 - val_loss: 0.1452 - val_categorical_accuracy: 0.9612

Epoch 14/17

12/12 [=====] - 13s 1s/step - loss: 0.1541 - categorical_accuracy: 0.9404 - val_loss: 0.0956 - val_categorical_accuracy: 0.9778

Epoch 15/17

12/12 [=====] - 13s 1s/step - loss: 0.1490 - categorical_accuracy: 0.9321 - val_loss: 0.1005 - val_categorical_accuracy: 0.9640

Epoch 16/17

12/12 [=====] - 13s 1s/step - loss: 0.1338 - categorical_accuracy: 0.9446 - val_loss: 0.1084 - val_categorical_accuracy: 0.9557

Epoch 17/17

12/12 [=====] - 13s 1s/step - loss: 0.2421 - categorical_accuracy: 0.9064 - val_loss: 0.2951 - val_categorical_accuracy: 0.8892

[I 2022-06-09 18:39:58,177] Trial 8 finished with value: 0.8891966938972473 and parameters: {'n_epochs': 17, 'learning_rate': 0.008700000000000001, 'filters_1': 14, 'kernel_size_1': 16, 'filters_2': 28, 'kernel_size_2': 7, 'lin_1_n': 43}. Best is trial 7 with value: 0.950138509273529.

Epoch 1/15

```
/Users/barnabyemmens/opt/anaconda3/envs/appml/lib/python3.8/site-  
packages/optuna/distributions.py:560: UserWarning: The distribution is specified  
by [4, 12] and step=3, but the range is not divisible by `step`. It will be  
replaced by [4, 10].
```

```
warnings.warn(  

```

```
12/12 [=====] - 11s 929ms/step - loss: 2.4020 -  
categorical_accuracy: 0.3680 - val_loss: 1.0696 - val_categorical_accuracy:  
0.4626
```

Epoch 2/15

```
12/12 [=====] - 11s 908ms/step - loss: 0.7733 -  
categorical_accuracy: 0.6507 - val_loss: 0.6238 - val_categorical_accuracy:  
0.7535
```

Epoch 3/15

```
12/12 [=====] - 11s 923ms/step - loss: 0.5632 -  
categorical_accuracy: 0.7720 - val_loss: 0.5436 - val_categorical_accuracy:  
0.7784
```

Epoch 4/15

```
12/12 [=====] - 11s 917ms/step - loss: 0.5011 -  
categorical_accuracy: 0.7956 - val_loss: 0.5167 - val_categorical_accuracy:  
0.7590
```

Epoch 5/15

```
12/12 [=====] - 11s 923ms/step - loss: 0.4202 -  
categorical_accuracy: 0.8191 - val_loss: 0.4100 - val_categorical_accuracy:  
0.8449
```

Epoch 6/15

```
12/12 [=====] - 11s 922ms/step - loss: 0.3842 -  
categorical_accuracy: 0.8226 - val_loss: 0.3871 - val_categorical_accuracy:  
0.8449
```

Epoch 7/15

```
12/12 [=====] - 11s 929ms/step - loss: 0.3332 -  
categorical_accuracy: 0.8593 - val_loss: 0.3776 - val_categorical_accuracy:  
0.8560
```

Epoch 8/15

```
12/12 [=====] - 11s 951ms/step - loss: 0.3089 -  
categorical_accuracy: 0.8649 - val_loss: 0.4103 - val_categorical_accuracy:  
0.8310
```

Epoch 9/15

```
12/12 [=====] - 11s 935ms/step - loss: 0.3373 -  
categorical_accuracy: 0.8559 - val_loss: 0.2851 - val_categorical_accuracy:  
0.8809
```

Epoch 10/15

```
12/12 [=====] - 11s 919ms/step - loss: 0.2391 -  
categorical_accuracy: 0.9044 - val_loss: 0.2073 - val_categorical_accuracy:  
0.9141
```

Epoch 11/15

```
12/12 [=====] - 11s 925ms/step - loss: 0.1902 -
```


categorical_accuracy: 0.9168 - val_loss: 0.2033 - val_categorical_accuracy: 0.9141

Epoch 12/15

12/12 [=====] - 11s 924ms/step - loss: 0.1773 - categorical_accuracy: 0.9328 - val_loss: 0.1581 - val_categorical_accuracy: 0.9418

Epoch 13/15

12/12 [=====] - 11s 921ms/step - loss: 0.1264 - categorical_accuracy: 0.9529 - val_loss: 0.1756 - val_categorical_accuracy: 0.9307

Epoch 14/15

12/12 [=====] - 11s 912ms/step - loss: 0.1235 - categorical_accuracy: 0.9536 - val_loss: 0.1214 - val_categorical_accuracy: 0.9557

Epoch 15/15

12/12 [=====] - 11s 911ms/step - loss: 0.1171 - categorical_accuracy: 0.9556 - val_loss: 0.1731 - val_categorical_accuracy: 0.9280

[I 2022-06-09 18:42:44,545] Trial 9 finished with value: 0.9279778599739075 and parameters: {'n_epochs': 15, 'learning_rate': 0.00636, 'filters_1': 6, 'kernel_size_1': 16, 'filters_2': 14, 'kernel_size_2': 4, 'lin_1_n': 84}. Best is trial 7 with value: 0.950138509273529.

```
[11]: optimised_params = study.best_params
      print(optimised_params)
```

```
{'n_epochs': 17, 'learning_rate': 0.0022900000000000004, 'filters_1': 24,
'kernel_size_1': 12, 'filters_2': 54, 'kernel_size_2': 4, 'lin_1_n': 82}
```

```
[12]: def opt_model(optimised_params):
      n_epochs = optimised_params['n_epochs']
      learning_rate = optimised_params['learning_rate']
      optimiser = Adam(learning_rate=learning_rate)
      # Conv 1
      filters_1 = optimised_params['filters_1']
      kernel_size_1 = optimised_params['kernel_size_1']
      strides_1 = 1
      padding_1 = 'same'
      activation_1 = 'relu'
      input_shape = (img_rows, img_cols, 1)
      # Pool 1
      pool_size_1 = 2
      # Conv 2
      filters_2 = optimised_params['filters_2']
      kernel_size_2 = optimised_params['kernel_size_2']
      strides_2 = 1
      padding_2 = 'same'
```

```

activation_2 = 'relu'
# Pool 2
pool_size_2 = 2
# Dropout
dropout_rate = 0.4
# Lin 1
lin_1_n = optimised_params['lin_1_n']

model = Sequential()
model.add(Conv2D(filters=filters_1,
                  kernel_size=kernel_size_1,
                  strides=strides_1,
                  padding=padding_1,
                  activation=activation_1,
                  input_shape=input_shape))
model.add(MaxPooling2D(pool_size=pool_size_1, strides=None))
model.add(Conv2D(filters=filters_2,
                  kernel_size=kernel_size_2,
                  strides=strides_2,
                  padding=padding_2,
                  activation=activation_2))
model.add(MaxPooling2D(pool_size=pool_size_2, strides=None))
model.add(Dropout(rate=dropout_rate))
model.add(Flatten())
model.add(Dense(units=lin_1_n, activation='relu'))
model.add(Dense(units=num_classes, activation='softmax'))

model.compile(loss=categorical_crossentropy,
              optimizer=optimiser,
              metrics=['categorical_accuracy'])

return model

```

3 Folding

```

[13]: n_folds = 5
kfold = KFold(n_splits=n_folds, shuffle=True)
X = mels
y = y
# Human Optimisation...
check_class_complete(kfold,X,y)
#optimised_params['learning_rate'] = 0.0014280245429123698/2
#optimised_params['n_epochs'] = 30
n_epochs = optimised_params['n_epochs']
fold_loss = np.zeros((n_folds,n_epochs))
print(optimised_params)

```

ALL CLASSES PRESENT

```
{'n_epochs': 17, 'learning_rate': 0.0022900000000000004, 'filters_1': 24,  
'kernel_size_1': 12, 'filters_2': 54, 'kernel_size_2': 4, 'lin_1_n': 82}
```

```
[14]: fold_no = 1  
for train_index, test_index in kfold.split(X, y):  
  
    print("-----")  
    print("FOLD "+str(fold_no))  
    print("-----")  
  
    X_train, X_test = X[train_index], X[test_index]  
    y_train, y_test = y[train_index], y[test_index]  
    X_train, X_test, y_train, y_test, img_rows, img_cols =   
    format_for_CNN(X_train, X_test, y_train, y_test)  
  
    model = opt_model(optimised_params)  
  
    history = model.fit(X_train, y_train,  
                        batch_size=128,  
                        epochs=n_epochs,  
                        verbose=1,  
                        validation_data=(X_test, y_test))  
  
    fold_loss[fold_no-1] = history.history['val_categorical_accuracy']  
    fold_no += 1
```

```
-----  
FOLD 1  
-----
```

Epoch 1/17

```
12/12 [=====] - 8s 672ms/step - loss: 1.9247 -  
categorical_accuracy: 0.4657 - val_loss: 0.8070 - val_categorical_accuracy:  
0.6205
```

Epoch 2/17

```
12/12 [=====] - 8s 668ms/step - loss: 0.7143 -  
categorical_accuracy: 0.6958 - val_loss: 0.6650 - val_categorical_accuracy:  
0.7258
```

Epoch 3/17

```
12/12 [=====] - 8s 667ms/step - loss: 0.6227 -  
categorical_accuracy: 0.7124 - val_loss: 0.5444 - val_categorical_accuracy:  
0.7645
```

Epoch 4/17

```
12/12 [=====] - 8s 670ms/step - loss: 0.5031 -  
categorical_accuracy: 0.7942 - val_loss: 0.4573 - val_categorical_accuracy:  
0.8033
```

Epoch 5/17

```
12/12 [=====] - 8s 665ms/step - loss: 0.3810 -
```

categorical_accuracy: 0.8344 - val_loss: 0.3941 - val_categorical_accuracy: 0.8393

Epoch 6/17

12/12 [=====] - 8s 660ms/step - loss: 0.2991 - categorical_accuracy: 0.8787 - val_loss: 0.2717 - val_categorical_accuracy: 0.8809

Epoch 7/17

12/12 [=====] - 8s 662ms/step - loss: 0.2722 - categorical_accuracy: 0.8822 - val_loss: 0.2577 - val_categorical_accuracy: 0.9114

Epoch 8/17

12/12 [=====] - 8s 664ms/step - loss: 0.1968 - categorical_accuracy: 0.9245 - val_loss: 0.1825 - val_categorical_accuracy: 0.9280

Epoch 9/17

12/12 [=====] - 8s 664ms/step - loss: 0.1561 - categorical_accuracy: 0.9404 - val_loss: 0.1703 - val_categorical_accuracy: 0.9169

Epoch 10/17

12/12 [=====] - 8s 664ms/step - loss: 0.1477 - categorical_accuracy: 0.9362 - val_loss: 0.1267 - val_categorical_accuracy: 0.9584

Epoch 11/17

12/12 [=====] - 8s 682ms/step - loss: 0.1247 - categorical_accuracy: 0.9515 - val_loss: 0.1454 - val_categorical_accuracy: 0.9418

Epoch 12/17

12/12 [=====] - 8s 681ms/step - loss: 0.0927 - categorical_accuracy: 0.9716 - val_loss: 0.1256 - val_categorical_accuracy: 0.9584

Epoch 13/17

12/12 [=====] - 8s 664ms/step - loss: 0.0748 - categorical_accuracy: 0.9744 - val_loss: 0.1191 - val_categorical_accuracy: 0.9640

Epoch 14/17

12/12 [=====] - 8s 681ms/step - loss: 0.0566 - categorical_accuracy: 0.9778 - val_loss: 0.0826 - val_categorical_accuracy: 0.9695

Epoch 15/17

12/12 [=====] - 8s 669ms/step - loss: 0.0567 - categorical_accuracy: 0.9834 - val_loss: 0.0907 - val_categorical_accuracy: 0.9723

Epoch 16/17

12/12 [=====] - 8s 670ms/step - loss: 0.0439 - categorical_accuracy: 0.9868 - val_loss: 0.0699 - val_categorical_accuracy: 0.9778

Epoch 17/17

12/12 [=====] - 8s 664ms/step - loss: 0.0511 -

categorical_accuracy: 0.9827 - val_loss: 0.0715 - val_categorical_accuracy:
0.9751

FOLD 2

Epoch 1/17

12/12 [=====] - 9s 689ms/step - loss: 1.7964 -
categorical_accuracy: 0.4726 - val_loss: 0.9193 - val_categorical_accuracy:
0.6066

Epoch 2/17

12/12 [=====] - 8s 664ms/step - loss: 0.7416 -
categorical_accuracy: 0.7013 - val_loss: 0.6365 - val_categorical_accuracy:
0.7452

Epoch 3/17

12/12 [=====] - 8s 664ms/step - loss: 0.5650 -
categorical_accuracy: 0.7616 - val_loss: 0.5404 - val_categorical_accuracy:
0.7673

Epoch 4/17

12/12 [=====] - 8s 663ms/step - loss: 0.4993 -
categorical_accuracy: 0.7879 - val_loss: 0.4234 - val_categorical_accuracy:
0.8449

Epoch 5/17

12/12 [=====] - 8s 665ms/step - loss: 0.3989 -
categorical_accuracy: 0.8288 - val_loss: 0.3242 - val_categorical_accuracy:
0.8864

Epoch 6/17

12/12 [=====] - 8s 672ms/step - loss: 0.2862 -
categorical_accuracy: 0.8836 - val_loss: 0.2592 - val_categorical_accuracy:
0.8975

Epoch 7/17

12/12 [=====] - 8s 678ms/step - loss: 0.2183 -
categorical_accuracy: 0.9113 - val_loss: 0.2213 - val_categorical_accuracy:
0.9003

Epoch 8/17

12/12 [=====] - 8s 663ms/step - loss: 0.1848 -
categorical_accuracy: 0.9217 - val_loss: 0.1670 - val_categorical_accuracy:
0.9418

Epoch 9/17

12/12 [=====] - 8s 665ms/step - loss: 0.1760 -
categorical_accuracy: 0.9279 - val_loss: 0.1167 - val_categorical_accuracy:
0.9612

Epoch 10/17

12/12 [=====] - 8s 665ms/step - loss: 0.1217 -
categorical_accuracy: 0.9563 - val_loss: 0.0871 - val_categorical_accuracy:
0.9723

Epoch 11/17

12/12 [=====] - 8s 662ms/step - loss: 0.0984 -
categorical_accuracy: 0.9681 - val_loss: 0.0729 - val_categorical_accuracy:

```

0.9861
Epoch 12/17
12/12 [=====] - 8s 664ms/step - loss: 0.0804 -
categorical_accuracy: 0.9702 - val_loss: 0.1671 - val_categorical_accuracy:
0.9280
Epoch 13/17
12/12 [=====] - 8s 664ms/step - loss: 0.0837 -
categorical_accuracy: 0.9716 - val_loss: 0.0531 - val_categorical_accuracy:
0.9806
Epoch 14/17
12/12 [=====] - 8s 667ms/step - loss: 0.0435 -
categorical_accuracy: 0.9868 - val_loss: 0.0482 - val_categorical_accuracy:
0.9806
Epoch 15/17
12/12 [=====] - 8s 684ms/step - loss: 0.0438 -
categorical_accuracy: 0.9854 - val_loss: 0.0362 - val_categorical_accuracy:
0.9889
Epoch 16/17
12/12 [=====] - 8s 673ms/step - loss: 0.0339 -
categorical_accuracy: 0.9882 - val_loss: 0.0333 - val_categorical_accuracy:
0.9917
Epoch 17/17
12/12 [=====] - 8s 674ms/step - loss: 0.0303 -
categorical_accuracy: 0.9931 - val_loss: 0.0273 - val_categorical_accuracy:
0.9917
-----
FOLD 3
-----
Epoch 1/17
12/12 [=====] - 8s 669ms/step - loss: 1.5085 -
categorical_accuracy: 0.5149 - val_loss: 0.7694 - val_categorical_accuracy:
0.6565
Epoch 2/17
12/12 [=====] - 8s 665ms/step - loss: 0.6181 -
categorical_accuracy: 0.7450 - val_loss: 0.4974 - val_categorical_accuracy:
0.8006
Epoch 3/17
12/12 [=====] - 8s 661ms/step - loss: 0.4670 -
categorical_accuracy: 0.7990 - val_loss: 0.4652 - val_categorical_accuracy:
0.8144
Epoch 4/17
12/12 [=====] - 8s 665ms/step - loss: 0.3708 -
categorical_accuracy: 0.8413 - val_loss: 0.3107 - val_categorical_accuracy:
0.8809
Epoch 5/17
12/12 [=====] - 8s 664ms/step - loss: 0.2847 -
categorical_accuracy: 0.8794 - val_loss: 0.3938 - val_categorical_accuracy:
0.8504

```

Epoch 6/17
12/12 [=====] - 8s 661ms/step - loss: 0.2747 -
categorical_accuracy: 0.8940 - val_loss: 0.1896 - val_categorical_accuracy:
0.9446

Epoch 7/17
12/12 [=====] - 8s 666ms/step - loss: 0.1612 -
categorical_accuracy: 0.9432 - val_loss: 0.1396 - val_categorical_accuracy:
0.9418

Epoch 8/17
12/12 [=====] - 8s 662ms/step - loss: 0.1204 -
categorical_accuracy: 0.9501 - val_loss: 0.1129 - val_categorical_accuracy:
0.9501

Epoch 9/17
12/12 [=====] - 8s 683ms/step - loss: 0.1087 -
categorical_accuracy: 0.9605 - val_loss: 0.1223 - val_categorical_accuracy:
0.9640

Epoch 10/17
12/12 [=====] - 8s 661ms/step - loss: 0.0867 -
categorical_accuracy: 0.9695 - val_loss: 0.1146 - val_categorical_accuracy:
0.9557

Epoch 11/17
12/12 [=====] - 8s 661ms/step - loss: 0.0654 -
categorical_accuracy: 0.9757 - val_loss: 0.1205 - val_categorical_accuracy:
0.9501

Epoch 12/17
12/12 [=====] - 8s 666ms/step - loss: 0.0514 -
categorical_accuracy: 0.9854 - val_loss: 0.1237 - val_categorical_accuracy:
0.9529

Epoch 13/17
12/12 [=====] - 8s 661ms/step - loss: 0.0472 -
categorical_accuracy: 0.9868 - val_loss: 0.1266 - val_categorical_accuracy:
0.9529

Epoch 14/17
12/12 [=====] - 8s 668ms/step - loss: 0.0434 -
categorical_accuracy: 0.9841 - val_loss: 0.1284 - val_categorical_accuracy:
0.9612

Epoch 15/17
12/12 [=====] - 8s 677ms/step - loss: 0.0604 -
categorical_accuracy: 0.9730 - val_loss: 0.1168 - val_categorical_accuracy:
0.9612

Epoch 16/17
12/12 [=====] - 8s 677ms/step - loss: 0.0435 -
categorical_accuracy: 0.9868 - val_loss: 0.0751 - val_categorical_accuracy:
0.9806

Epoch 17/17
12/12 [=====] - 8s 689ms/step - loss: 0.0295 -
categorical_accuracy: 0.9917 - val_loss: 0.0990 - val_categorical_accuracy:
0.9723

FOLD 4

Epoch 1/17

12/12 [=====] - 8s 675ms/step - loss: 1.5127 -
categorical_accuracy: 0.4269 - val_loss: 0.7794 - val_categorical_accuracy:
0.6233

Epoch 2/17

12/12 [=====] - 8s 669ms/step - loss: 0.6809 -
categorical_accuracy: 0.7110 - val_loss: 0.5431 - val_categorical_accuracy:
0.7812

Epoch 3/17

12/12 [=====] - 8s 668ms/step - loss: 0.5171 -
categorical_accuracy: 0.7678 - val_loss: 0.4637 - val_categorical_accuracy:
0.8089

Epoch 4/17

12/12 [=====] - 8s 669ms/step - loss: 0.3983 -
categorical_accuracy: 0.8281 - val_loss: 0.3195 - val_categorical_accuracy:
0.8947

Epoch 5/17

12/12 [=====] - 8s 669ms/step - loss: 0.2827 -
categorical_accuracy: 0.8891 - val_loss: 0.4429 - val_categorical_accuracy:
0.7950

Epoch 6/17

12/12 [=====] - 8s 671ms/step - loss: 0.2562 -
categorical_accuracy: 0.8919 - val_loss: 0.2151 - val_categorical_accuracy:
0.9141

Epoch 7/17

12/12 [=====] - 8s 669ms/step - loss: 0.1758 -
categorical_accuracy: 0.9328 - val_loss: 0.1882 - val_categorical_accuracy:
0.9446

Epoch 8/17

12/12 [=====] - 8s 671ms/step - loss: 0.1443 -
categorical_accuracy: 0.9459 - val_loss: 0.1808 - val_categorical_accuracy:
0.9307

Epoch 9/17

12/12 [=====] - 8s 669ms/step - loss: 0.1120 -
categorical_accuracy: 0.9598 - val_loss: 0.1284 - val_categorical_accuracy:
0.9557

Epoch 10/17

12/12 [=====] - 8s 668ms/step - loss: 0.0885 -
categorical_accuracy: 0.9716 - val_loss: 0.1450 - val_categorical_accuracy:
0.9391

Epoch 11/17

12/12 [=====] - 8s 668ms/step - loss: 0.0671 -
categorical_accuracy: 0.9785 - val_loss: 0.1415 - val_categorical_accuracy:
0.9418

Epoch 12/17


```

12/12 [=====] - 8s 670ms/step - loss: 0.0503 -
categorical_accuracy: 0.9841 - val_loss: 0.1136 - val_categorical_accuracy:
0.9640
Epoch 13/17
12/12 [=====] - 8s 677ms/step - loss: 0.0357 -
categorical_accuracy: 0.9875 - val_loss: 0.1115 - val_categorical_accuracy:
0.9612
Epoch 14/17
12/12 [=====] - 8s 678ms/step - loss: 0.0507 -
categorical_accuracy: 0.9827 - val_loss: 0.1029 - val_categorical_accuracy:
0.9751
Epoch 15/17
12/12 [=====] - 8s 691ms/step - loss: 0.0473 -
categorical_accuracy: 0.9834 - val_loss: 0.1028 - val_categorical_accuracy:
0.9834
Epoch 16/17
12/12 [=====] - 8s 675ms/step - loss: 0.0334 -
categorical_accuracy: 0.9889 - val_loss: 0.4304 - val_categorical_accuracy:
0.8670
Epoch 17/17
12/12 [=====] - 8s 682ms/step - loss: 0.1051 -
categorical_accuracy: 0.9591 - val_loss: 0.1237 - val_categorical_accuracy:
0.9695
-----
FOLD 5
-----
Epoch 1/17
12/12 [=====] - 9s 655ms/step - loss: 1.6636 -
categorical_accuracy: 0.4564 - val_loss: 0.9197 - val_categorical_accuracy:
0.6056
Epoch 2/17
12/12 [=====] - 8s 653ms/step - loss: 0.7691 -
categorical_accuracy: 0.6738 - val_loss: 0.8919 - val_categorical_accuracy:
0.6889
Epoch 3/17
12/12 [=====] - 8s 653ms/step - loss: 0.6480 -
categorical_accuracy: 0.7361 - val_loss: 0.6388 - val_categorical_accuracy:
0.7222
Epoch 4/17
12/12 [=====] - 8s 652ms/step - loss: 0.5129 -
categorical_accuracy: 0.7874 - val_loss: 0.4478 - val_categorical_accuracy:
0.8056
Epoch 5/17
12/12 [=====] - 8s 654ms/step - loss: 0.4085 -
categorical_accuracy: 0.8269 - val_loss: 0.3422 - val_categorical_accuracy:
0.8444
Epoch 6/17
12/12 [=====] - 8s 657ms/step - loss: 0.3255 -

```

categorical_accuracy: 0.8587 - val_loss: 0.2785 - val_categorical_accuracy: 0.8722

Epoch 7/17

12/12 [=====] - 8s 653ms/step - loss: 0.2463 - categorical_accuracy: 0.8871 - val_loss: 0.2332 - val_categorical_accuracy: 0.9056

Epoch 8/17

12/12 [=====] - 8s 655ms/step - loss: 0.2180 - categorical_accuracy: 0.9141 - val_loss: 0.1821 - val_categorical_accuracy: 0.9194

Epoch 9/17

12/12 [=====] - 8s 671ms/step - loss: 0.1732 - categorical_accuracy: 0.9294 - val_loss: 0.1590 - val_categorical_accuracy: 0.9389

Epoch 10/17

12/12 [=====] - 8s 661ms/step - loss: 0.1446 - categorical_accuracy: 0.9488 - val_loss: 0.1126 - val_categorical_accuracy: 0.9500

Epoch 11/17

12/12 [=====] - 8s 663ms/step - loss: 0.1104 - categorical_accuracy: 0.9543 - val_loss: 0.1007 - val_categorical_accuracy: 0.9556

Epoch 12/17

12/12 [=====] - 8s 661ms/step - loss: 0.0894 - categorical_accuracy: 0.9688 - val_loss: 0.0745 - val_categorical_accuracy: 0.9750

Epoch 13/17

12/12 [=====] - 8s 664ms/step - loss: 0.0680 - categorical_accuracy: 0.9785 - val_loss: 0.0648 - val_categorical_accuracy: 0.9778

Epoch 14/17

12/12 [=====] - 8s 660ms/step - loss: 0.0510 - categorical_accuracy: 0.9827 - val_loss: 0.0699 - val_categorical_accuracy: 0.9667

Epoch 15/17

12/12 [=====] - 8s 663ms/step - loss: 0.0599 - categorical_accuracy: 0.9771 - val_loss: 0.0610 - val_categorical_accuracy: 0.9778

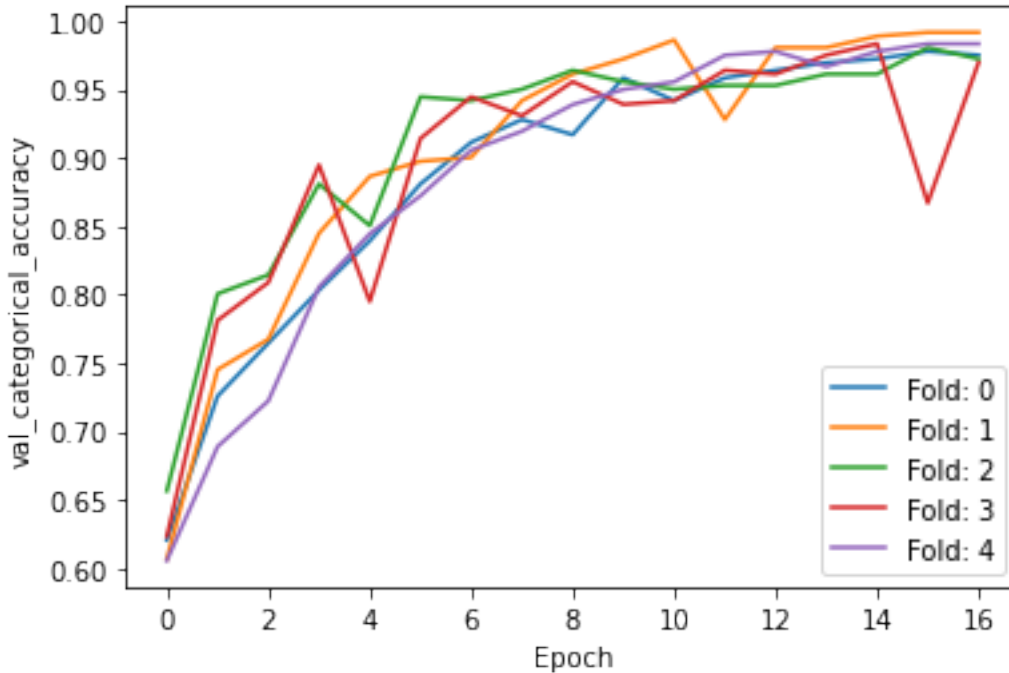
Epoch 16/17

12/12 [=====] - 8s 662ms/step - loss: 0.0483 - categorical_accuracy: 0.9848 - val_loss: 0.0495 - val_categorical_accuracy: 0.9833

Epoch 17/17

12/12 [=====] - 8s 659ms/step - loss: 0.0351 - categorical_accuracy: 0.9917 - val_loss: 0.0416 - val_categorical_accuracy: 0.9833

```
[15]: for i in range(n_folds):
        plt.plot(fold_loss[i], label='Fold: '+str(i))
plt.xlabel('Epoch')
plt.ylabel('val_categorical_accuracy')
plt.legend()
plt.show()
mean_acc = np.mean(fold_loss[:,-1])
print('Mean val_categorical_accuracy: ',np.around(mean_acc,5))
```



Mean val_categorical_accuracy: 0.97838

```
[16]: # model.layers will print a list of layer parameters/values
filters1, biases1 = model.layers[0].get_weights()
filters2, biases2 = model.layers[2].get_weights()

# normalize filter values to range 0-1 for better colormapping during plotting
def norm_filter(kernel):
    return (kernel - np.min(kernel)) / (np.max(kernel) - np.min(kernel))

n_1 = optimised_params['filters_1']
print('1st convolution layer:')
fig, axs = plt.subplots(1,n_1, figsize=(10, 6))
axs = axs.ravel()
for i in range(n_1):
    axs[i].imshow(norm_filter(filters1[:, :, 0, i]), cmap=plt.cm.binary)
```

```

    axs[i].set_xticks([]); axs[i].set_yticks([]); axs[i].grid(False)
plt.show()

n_2 = optimised_params['filters_2']
print('2nd convolution layer:')
fig, axs = plt.subplots(1,n_2, figsize=(10, 6))
axs = axs.ravel()
for i in range(n_2):
    axs[i].imshow(norm_filter(filters2[:, :, 0, i]), cmap=plt.cm.binary)
    axs[i].set_xticks([]); axs[i].set_yticks([]); axs[i].grid(False)

```

1st convolution layer:



2nd convolution layer:



3.0.1 Evaluation

```

[17]: predicted_prob = model.predict(TEST_images)
      predictions = np.argmax(predicted_prob, axis=1)

```

```

[18]: print(predictions.shape)
      print(TEST_y.shape)

```

(452,)

(452,)

```

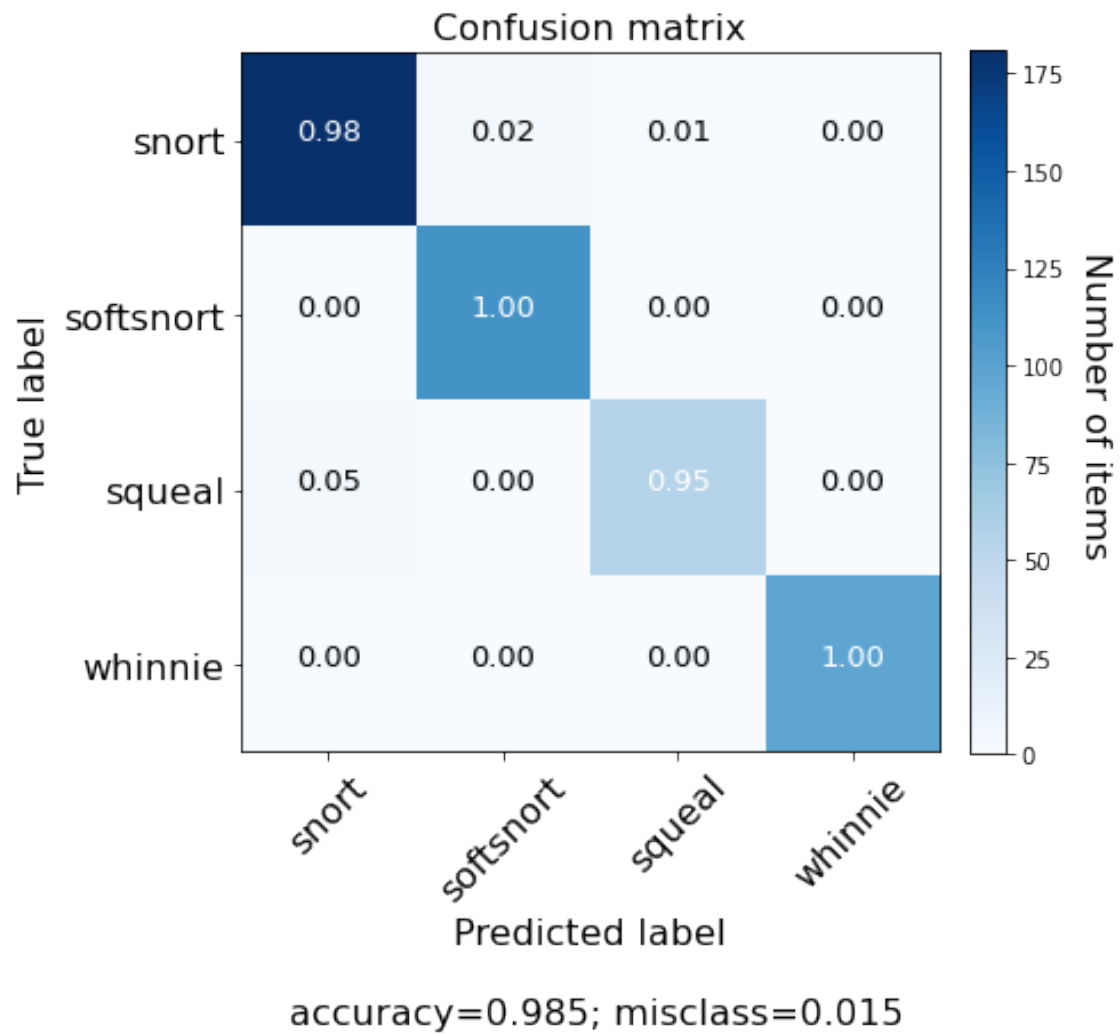
[19]: labels_list = np.unique(TEST_y)
      onehot_v = pd.get_dummies(TEST_y)
      onehot_y = onehot_v.to_numpy()
      truths = np.argmax(onehot_y, axis=1)
      cm = confusion_matrix(truths, predictions, labels=range(num_classes))
      print(cm)
      plot_confusion_matrix(cm, labels_list)

```

```

[[181  3  1  0]
 [  0 112  0  0]
 [  3  0 55  0]
 [  0  0  0 97]]

```



[]: