

Elephant Rumble Model Report

A Convolutional Neural Network based architecture to detect Elephant-Rumble:

CNN Model:

Parameter for GFCC extraction:

- 1- Down sampling the sample rate to 2000
- 2- Number of extracted cepstrum coefficients: 18
- 3- preEmphasisFactor = 0 {no preemphasis}
- 4- lifterParameter = 0 {no liftering}
- 5- lowFrequency = 10Hz
- 6- highFrequency = 500Hz
- 7- windoLength = 0.075
- 8- windowStep = 0.05
- 9- Hamming using hann window from scipy.signal.windows

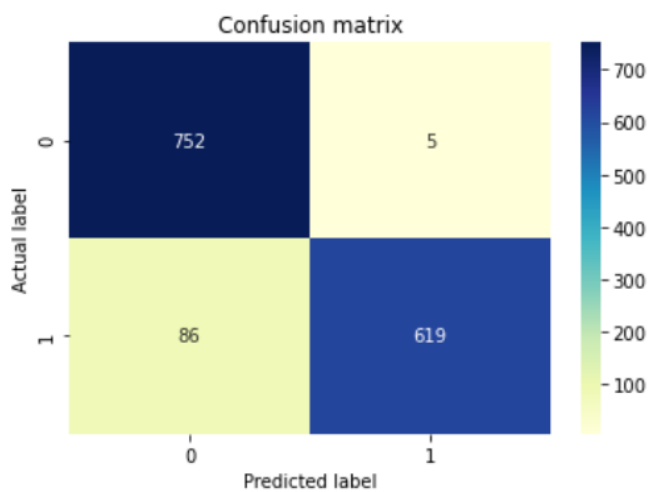
Results:

Train Accuracy: 99.7%

Validation Accuracy: 94.48%

Test Accuracy: 93.8%

(The CNN model is an improvement over the dense model which gave accuracy score of 90% and 78% on train and test sets respectively)



1 denotes rumble & 0 denotes noise

The model has high False -ve share as compared to False +ves: 86/ 1462 predictions

The False +ve rate is menial: 5/1462

F1-score of 0.94

Model Architecture:

```
model = Sequential()  
model.add(Conv2D(64, kernel_size=3, input_shape=X_train.shape[1:]))  
model.add(BatchNormalization())  
model.add(Activation("relu"))  
model.add(MaxPooling2D(pool_size=(2, 2)))  
model.add(Conv2D(32, kernel_size=3))  
model.add(BatchNormalization())  
model.add(Activation("relu"))  
model.add(MaxPooling2D(pool_size=(2, 2)))  
model.add(Flatten())  
model.add(Dense(32))  
model.add(BatchNormalization())  
model.add(Activation("relu"))  
model.add(Dense(2, activation='softmax'))
```

Model Compilation factors:

Loss = categorical_crossentropy

Metrics = accuracy based

Optimizer = adam