EE603A Assignment 2

Sanika Gumaste(200865)

1 Introduction

The following is an assignment on Multi Label Audio Classification. Given a data set of 5000 points classified in 10 classes, we need to create a multi-label classification model. To solve the given problem of 10 class audio classification, I have used a Convolutional Neural Network.

2 Literature Survey

Multi-label classification is a technique for mapping data from single labels to multiple labels. These multiple labels are part of the same label set, which includes inconsistent labels. The aim of multi-label classification is to develop a classification model. Various models such as Convolutional Neural Networks, Dense Neural Networks, Recurrent Neural Networks, etc. can be used to train models for Multi Label Classification, however, previous research supports that CNNs provide the best support and accuracy for multi label classification problems.

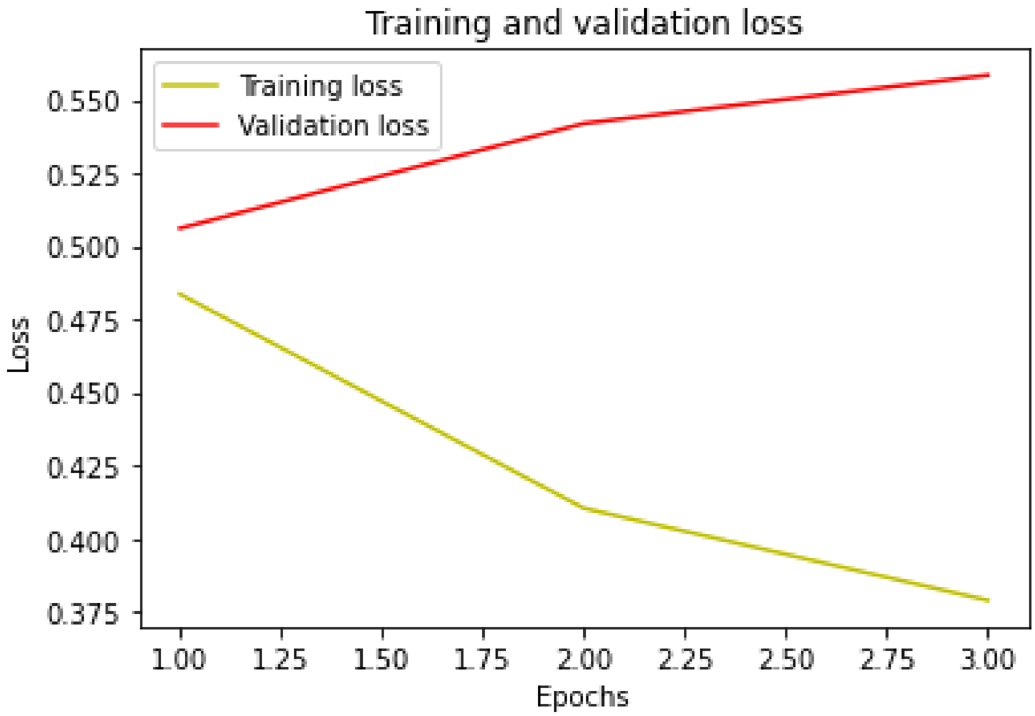
3 Methodology

The steps I followed to build the models are:

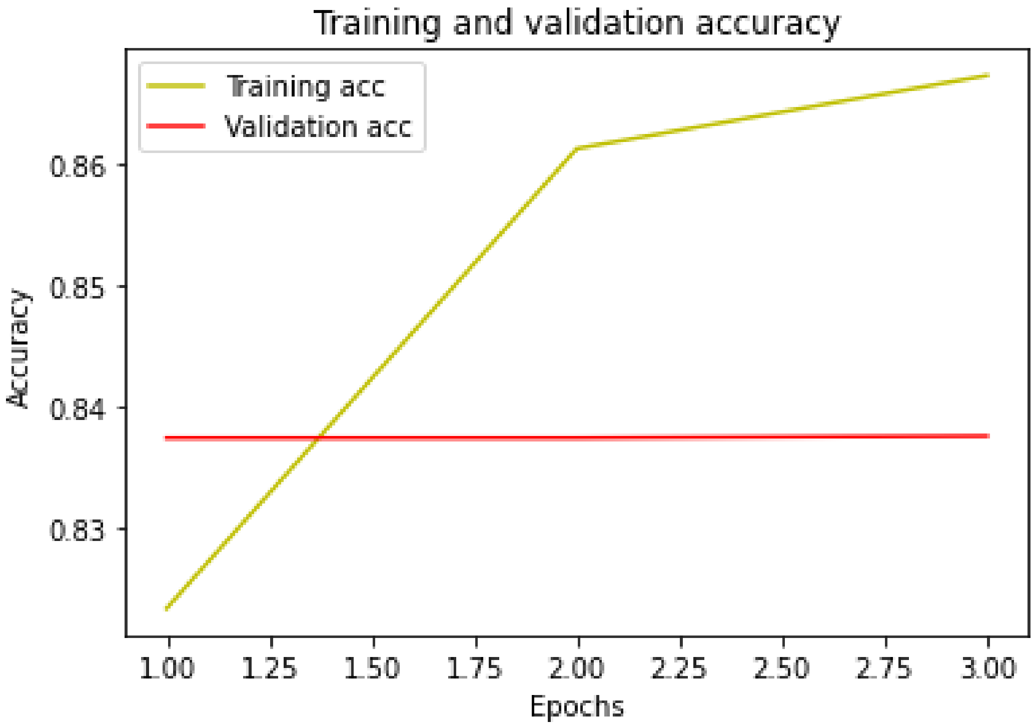
1) **Loading libraries:** The libraries used in the following model are Numpy, Pandas, Keras, Seaborn, Sklearn, and a few other libraries.
2) **Data loading:** Loaded the training data (5000 $*.\mathbf{npy}$ files) into the dictionary 'train', and loaded corresponding 5000 eventroll files containing the labels into a dictionary called labels. Used the given function to convert eventrolls to one-hot vectors.
3) **Data pre-processing:** Defined functions to load the training dataset into the dictionary 'train' and the one-hot encoded vectors into the dictionary 'labels' in a single step. The inputs were padded to a constant size of length 700.
5) **Data Visualisation:** Data visualisation has been implemented using spectrograms and distribution graphs. The spectrograms display the occurrence of the various signals corresponding to every data in the train data. The distribution shows us the number of occurrences of every class in 10 signals picked randomly from the given data set.
4) **Modeling:** The model was trained using Convolutional Neural Networks. The convolutional neural network is implemented using the Keras library from tensorflow. A sequential model with Conv2D layers of kernal size of 8, 16,32, and 64 and 128 respectively were implemented, followed by a maxpooling layer of varying kernel sizes of 2, 3 and 5 are used. the model then passes through two ReLU activated layers of 128 and 64 units. Lastly, the model is passed through a sigmoid activated layer.

The model is analysed on the basis of f1 scores, accuracy, recall and binary validation accuracy. Early stopping is implemented based on the maximisation of binary validation accuracy. In the training data set, the binary validation accuracy turns out to be over 80The model is trained for 10 epochs.

The training and Validation data loss can be represented as follows

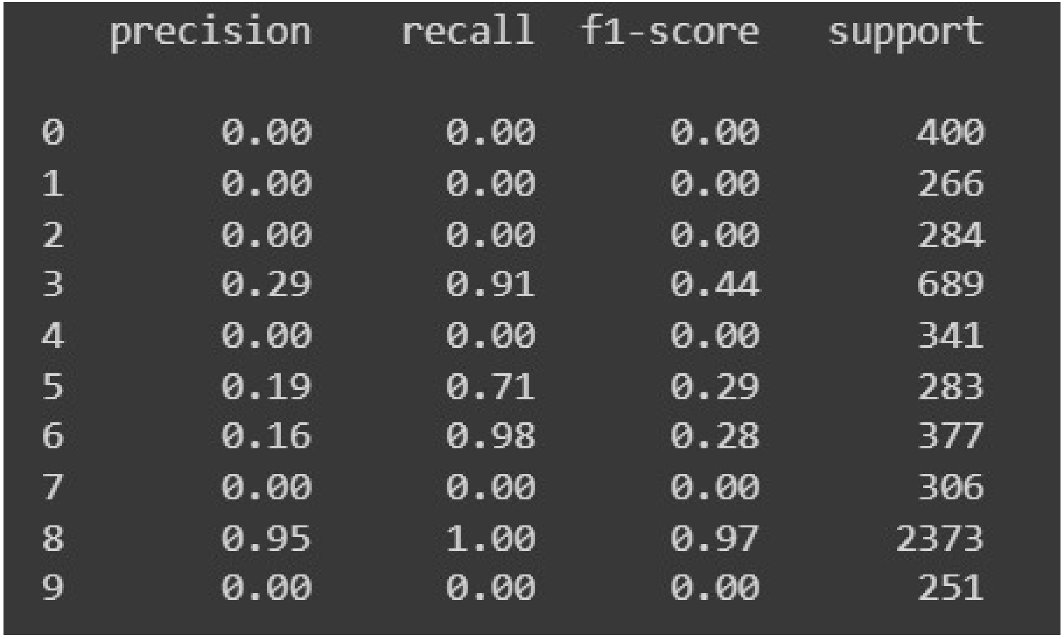


The training and Validation data accuracy can be represented as follows



4 Metric

For the given training data, test data and ground truth-

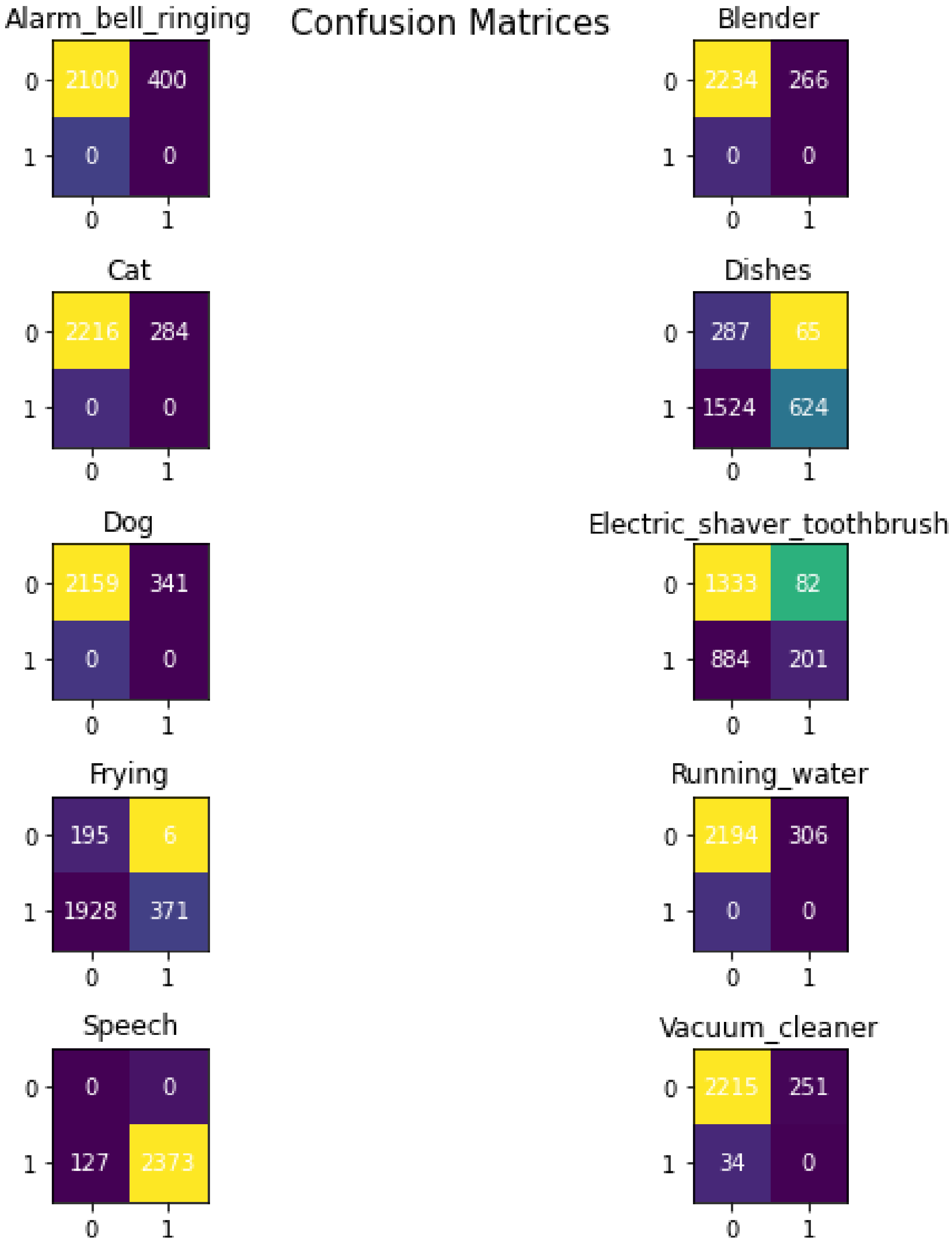


```
     precision    recall  f1-score   support


0         0.00      0.00      0.00       400
1         0.00      0.00      0.00       266
2         0.00      0.00      0.00       284
3         0.29      0.91      0.44       689
4         0.00      0.00      0.00       341
5         0.19      0.71      0.29       283
6         0.16      0.98      0.28       377
7         0.00      0.00      0.00       306
8         0.95      1.00      0.97      2373
9         0.00      0.00      0.00       251
```

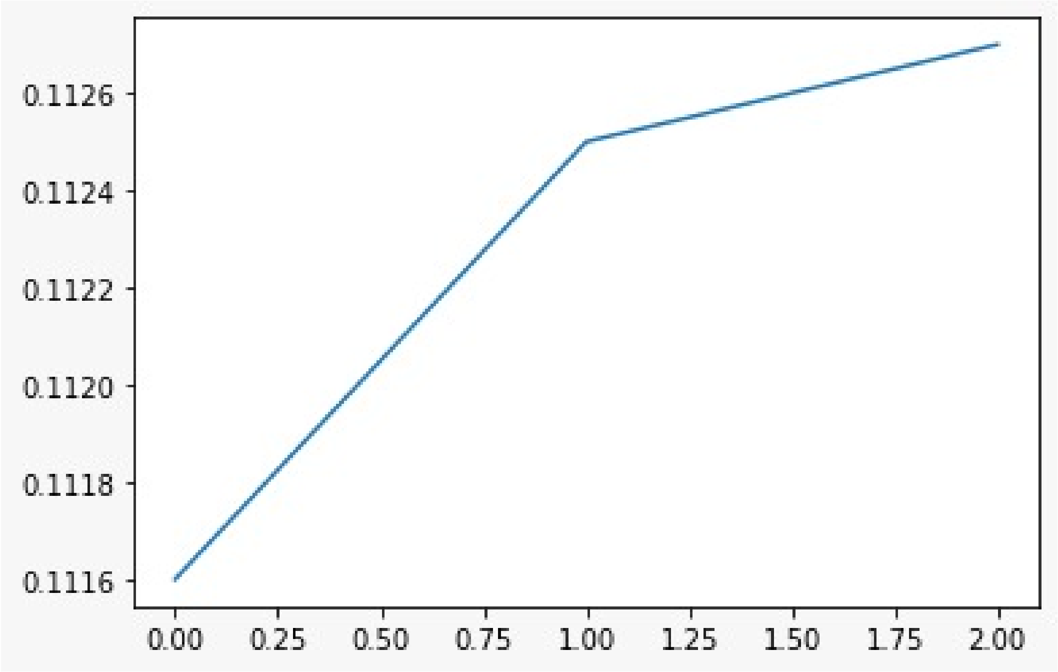
The f1 score is:- [0, 0, 0, 0.44, 0, 0.29, 0.28, 0, 0.97, 0]
The recall is:- [0, 0, 0, 0.91, 0, 0.71, 0.98, 0, 1, 0]
The precision score is:- [0, 0, 0, 0.29, 0, 0.19, 0.16, 0, 0.95, 0]

The confusion heat-map is displayed below.



5 Observations and discussion

The Convolutional Neural Network model was able to predict the correct labels of the given data set with accuracy as follows


0.1126
0.1124
0.1122
0.1120
0.1118
0.1116
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00