EE603: Audio event detection

Achint Soni

October 12, 2022

1 Introduction

The dataset provided consists of 1000 spectrogram data points, each belonging to a particular class. There are a total of 10 labels. For this task, several machine learning algorithms were used. This include Convolutional Neural Networks, Deep Neural Networks and Support Vector Machines. A total of 4 different types of networks were used. And for a particular CNN model as mentioned in Section 3.2, four different models were trained for different optimizers. So a total of 7 models were trained for prediction.

2 Literature Survey

The authors in [1] propose a system framework for learning acoustic event detectors using only weakly labeled data. A solution is proposed for solving multiple-instance learning, one based on support vector machines (SVM) and the other on neural networks. The proposed approach leads to less time consuming and less expensive process of the manual annotation of data, in order to facilitate fully supervised learning. The system is able to recognize events in the recordings. Results show that events like clanking, scraping and children's voices are easily detectable using SVM and neural network approaches, whereas other events are harder to detect using both of those methods.

3 Methods

3.1 Support Vector Machines

To train the support vector machine, I split the train data into train and test data. The train data is then given for training to the model. Since support vector machines only take one dimensional training data as input, all data were first reshaped using numpy.reshape to 1-D vector. The F1-score on the test dataset came out to be 73.5%. The parameters used for this model were Regularization parameter and Kernel coefficient.

3.2 Convolutional Neural Networks

Moving on, CNN models were used to detect the audio. I used two types of CNN models to predict the data. For the first model, the input of the data were spectrograms as they were given in the data. Since CNN takes fixed input shape of the training data, A common input shape of (128,2500,1) was selected for all data points. The point of smaller size were zero padded and the points of bigger shape were trimmed to (128,2500,1). A total of three convolutional blocks were used in the model which consisted of Conv2D, BatchNormalization, Dropout and MaxPooling layer. After the convolution blocks, two dense layes of size 32 and 10 were connected after using Flatten. Filter size for convolution blocks was used as (3,3), padding = 'same' and stride was kept as 1. A total of 5,118,058 parameters were there in the model. The activation for all layers except the final dense layers was "relu" and for the final dense layer, softmax was used. Adam optimizers was used to train this model. Callbacks included, learning rate scheduler, early stopping and model checkpoints. Training accuracy turned out to be 82% and validation accuracy 76%.

For the second CNN model, the data was first converted to spectrogram images using librosa library and then the images were given as input to the model. The data was split into train, test and validation in ratio (60:20:20). This method showed very promising results. The input shape for the model was (256,256,3). All the images was reshaped to this size. Then we augment the data using ImageDataGenerator function in Keras library. Parameters for data augmentation included rescale, brightness_range = (0.5,1.5), and zoom_range = 0.5. This was performed only of training data. For validation and test datasets, only rescaling = (1./255) of images was performed. Same model architecture was used as used for CNN in previous subsection. Only the input shape was changed. Callbacks were also same. Several optimizers were used to train this model. This included Adam, Nadam, SGD with momentum and RMSprop. The F1 score for all this models on the test dataset turned out to be $[73.00000190734863, 82.99999833106995, 81.99999928474426, 0.7599999904632568]$ respectively. Nadam had the best F1 score. Confusion matrix shown in figure 3.

3.3 Deep Neural Network

Finally, deep neural networks were used to predict the labels. The dataset was split into train and validation dataset in ratio (80:20). Then the data was given to a neural network. The data over here are spectrograms and not images. The input shape of the model is (128,2500,1). The model contains, Flatten and 3 dense layers of shape 64, 32 and 10 respectively. Dropout and BatchNormalization were added after every dense layer to prevent overfitting. Adam optimizer gave the best results for this model with 90% train accuracy

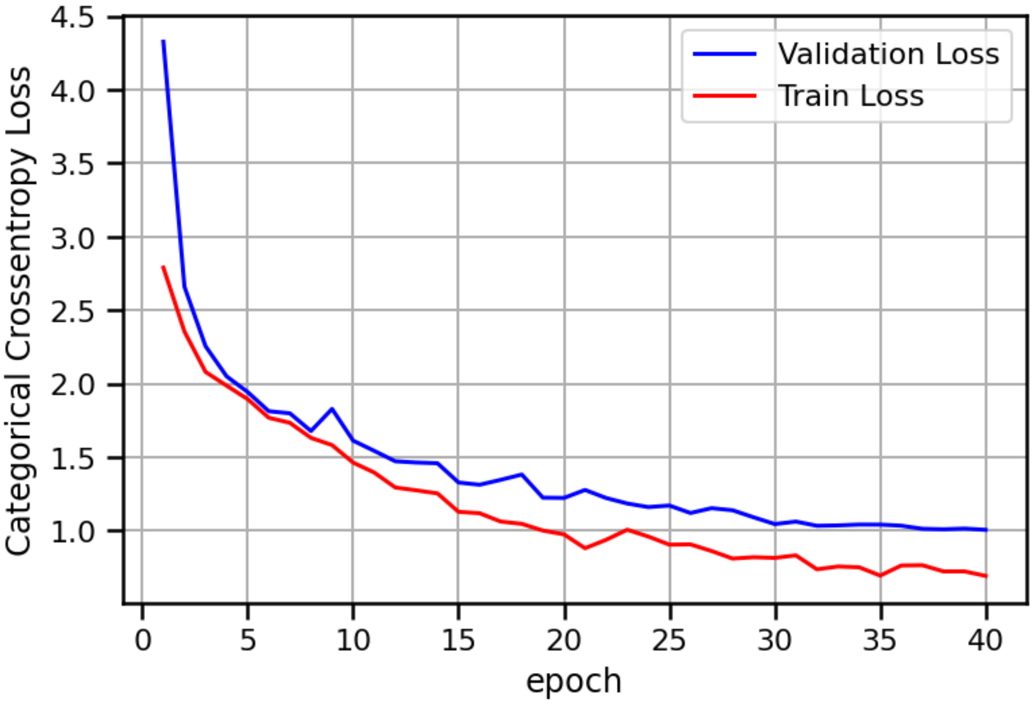

Figure 1: Loss curve of CNN model with images as input

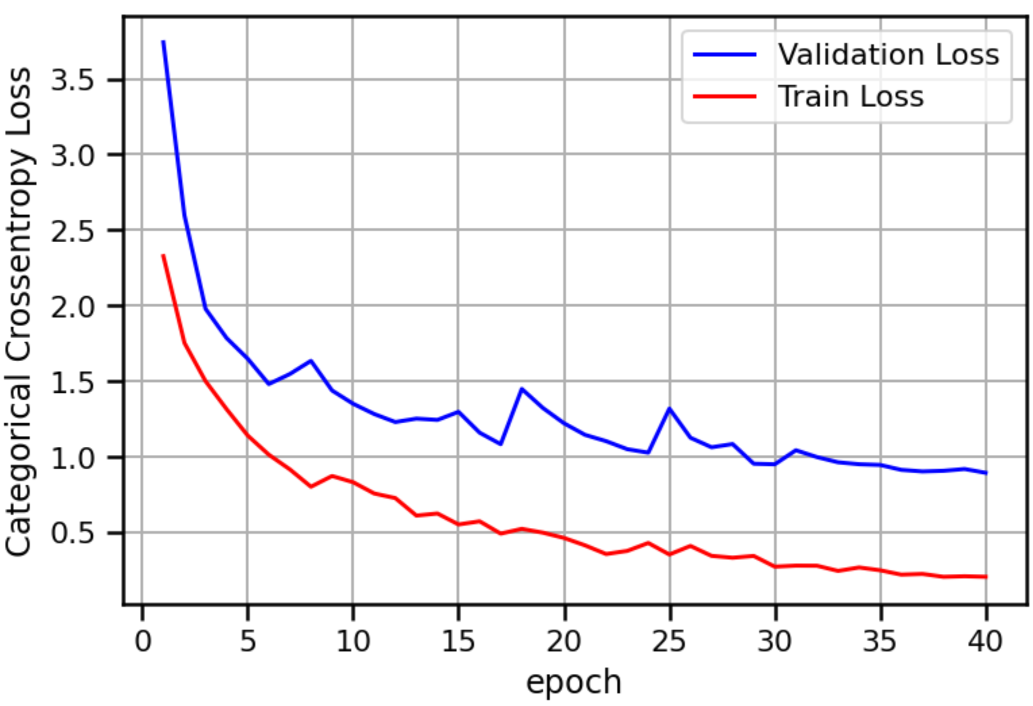

Figure 2: Loss curve for CNN model with spectrogram as input

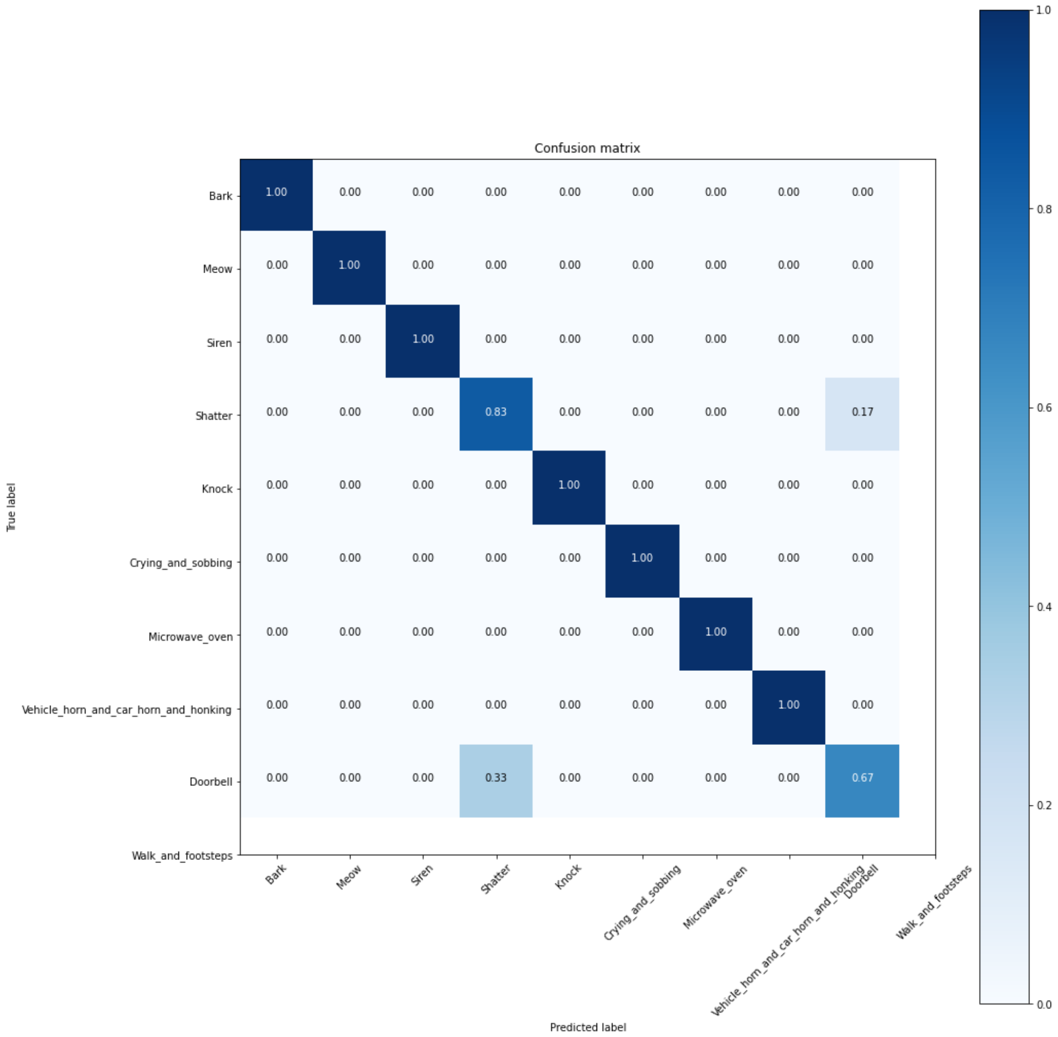

Figure 3: Confusion matrix for Nadam on testing data(using train-test split)

and 84% validation accuracy. Learning rate scheduler, early stopping and model checkpoints were used as callbacks.

4 Results

CNNs with input as images of spectrograms produced the best results out of these models. So they were used to predict the given test dataset.

4.1 Precision

Precision score on the test data turns out to be 0.7710418248761017.

4.2 Recall

Recall score on the test data is 0.7213930348258707.

4.3 F1 score

F1 score on the test data is 0.7228440123418562.

4.4 Confusion Matrix

Shown in figure 4.

5 Observation and Discussion

In this paper we have experimented with audio event classification using different types of classifiers. We have found that a DNN classifier is very useful for this task and performs a little better than SVM. A small additional improvement can be gained by combining the DNN and the SVM scores. CNN model with images of spectrogram as input gave the best results as compared to other models. Although other models performed well for some classes, like SVM model works well for children voices, CNN models in general performed well for all classes. Overfitting was observed in CNN models with spectrograms as input initially, but adding Dropouts and reducing parameters finally solved the issue.

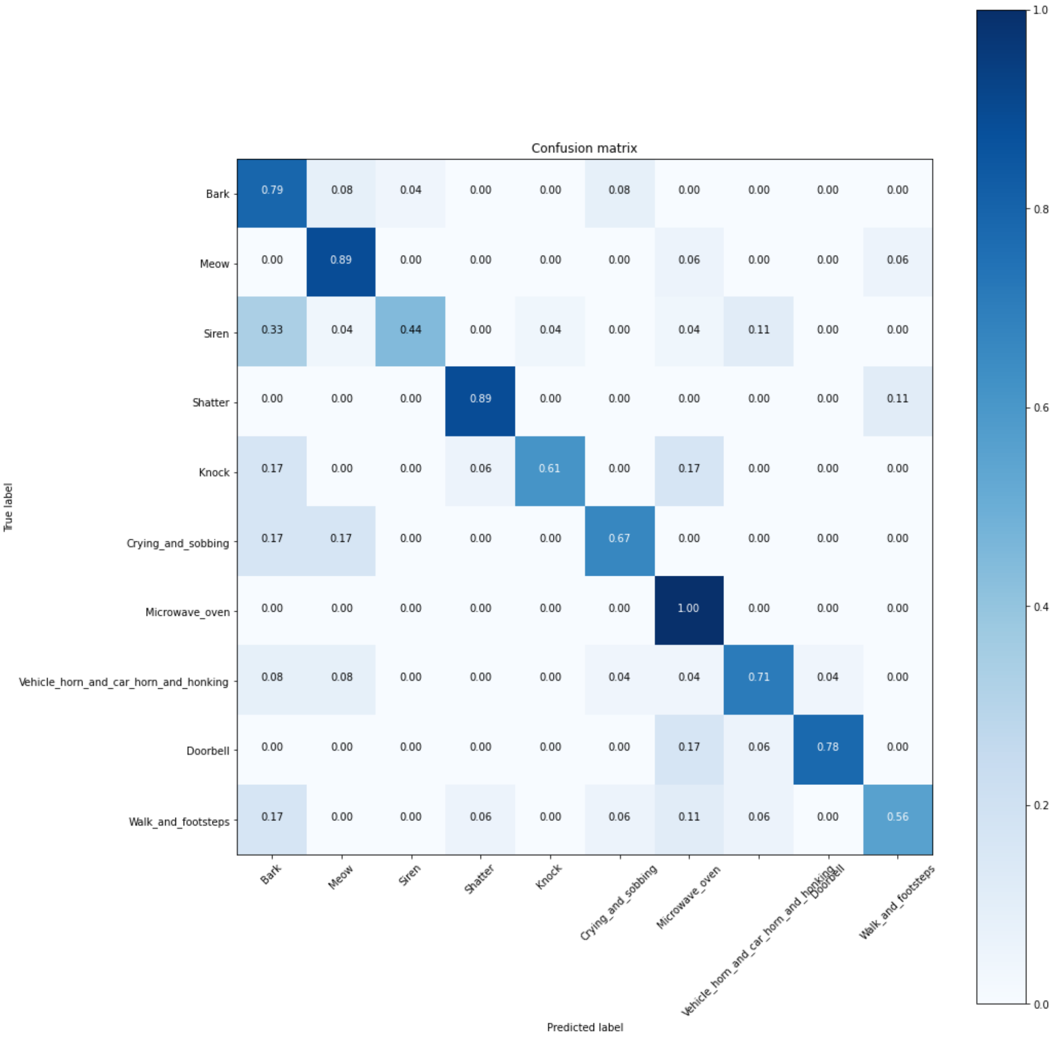

Figure 4: Normalized Confusion matrix on test dataset

The performance of DNN-based and CNN-based classifiers were measured with various model structures. Experimental results exhibited a maximum performance of 82.5 %, which is approximately 10 % higher than the performance of the baseline SVM classifier.

References

[1] Kumar, A. and Raj, B., 2016, October. Audio event detection using weakly labeled data. In Proceedings of the 24th ACM international conference on Multimedia (pp. 1038-1047).

[2] Dandashi, Amal Aljaam, Jihad. (2017). A Survey on Audio Content-Based Classification. 408-413. 10.1109/CSCI.2017.69.

[3] Kons, Z., Toledo-Ronen, O. and Carmel, M., 2013, August. Audio event classification using deep neural networks. In Interspeech (pp. 1482-1486).

[4] Lim, M., Lee, D., Park, H., Kang, Y., Oh, J., Park, J.S., Jang, G.J. and Kim, J.H., 2018. Convolutional neural network based audio event classification. KSII Transactions on Internet and Information Systems (TIIS), 12(6), pp.2748-2760.