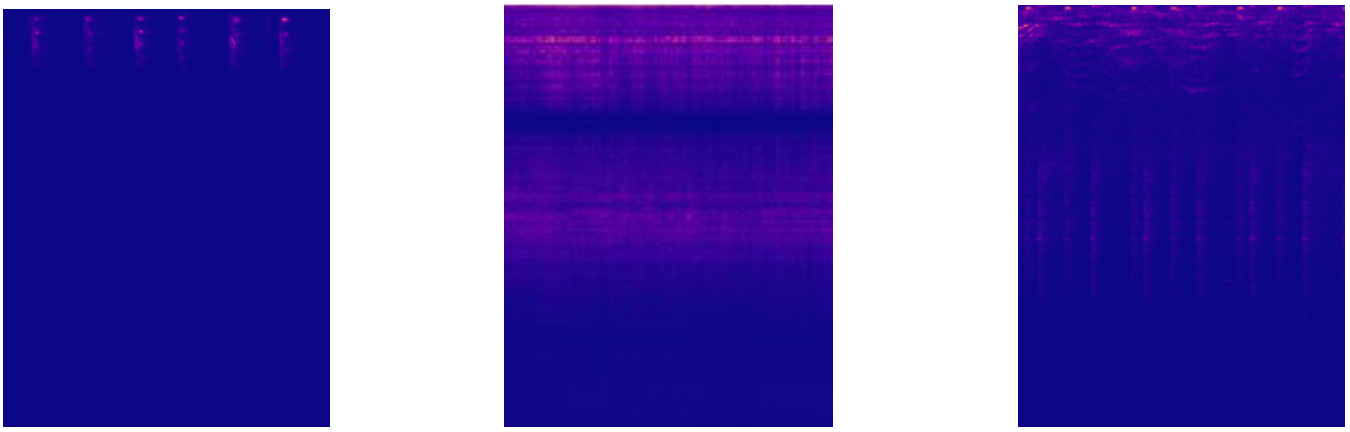
EE698V Project Report

- Saksham Gupta (170613) and Aryan Jain (170153)

INTRODUCTION

The objective is to develop a model, using Machine Learning techniques to classify audio events (Task - 1). Task - 2 is to identify the sequence of audio events. The tasks are commonly known as 'Audio Event Detection'.

DATA: For training, we have 1761 audio '.wav' files. The duration varied from 4 seconds to 0.07 seconds. For each file, we calculate its Short-time Fourier transform (STFT), sampled at 44100Hz and window length 1024 (n_fft), which outputs a 1-D image of height equal to 513 ($= 1 + \text{n\_fft}/2$) and varying width (depending upon the duration of audio). The models are primarily deep neural networks, so we use GPUs for faster computations.



(a) dog bark (b) drilling (c) street music

Figure-1: Some sample STFT images of different classes

TASK – 1

Data Pre-processing: We resize each STFT to 513 x 401, which corresponds to 4-second duration (75% + files are of the same dimensions). If STFT image is shorter, add zero padding; else select randomly 401 consecutive columns. We normalise the training data (zero mean and unit variance). We use one-hot encoding for labels/target.

Model: We use a deep convolution neural network which has an architecture as follows:

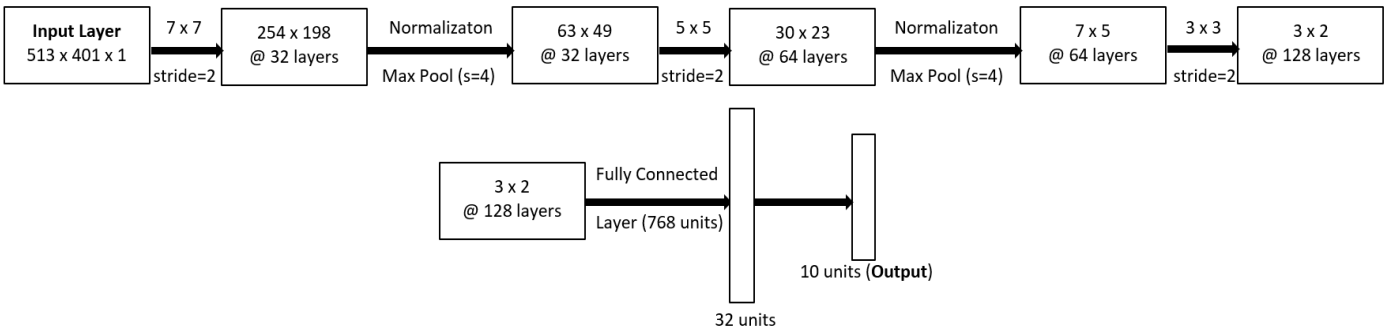


Figure-2: A model architecture for Task - 1

We use 10 fold cross-validation and save the 'best' model weights (maximum validation accuracy) in each case. The learning rate, l is exponentially decreasing: $l = 10^{-3} * (0.95)^{x}$; x is the epoch number. Also, we use 'early stopping' with the patience of 10 epochs. The overall model summary is as follows:

Mean Training accuracy = 93.54%; Mean Validation accuracy = 90.07%.

TASK – 2

Data Generation: In addition to the audio event classification, the model should predict the sequence of the events. We selected the minimum threshold duration of 0.5 seconds for any event in the sequence of audio events. We split each STFT into multiple files of size 513 x 50. The size of the training data is 12718 samples. We normalise the training data and use one-hot encoding for labels/target.

Model: We use a CNN based model similar to the Task – 1. As in Task - 1, we use 10 fold cross-validation and only save the 'best' model, use exponentially decreasing learning rate and 'early stopping' with patience equal to 10 epochs. The convolution neural network architecture is as follows:

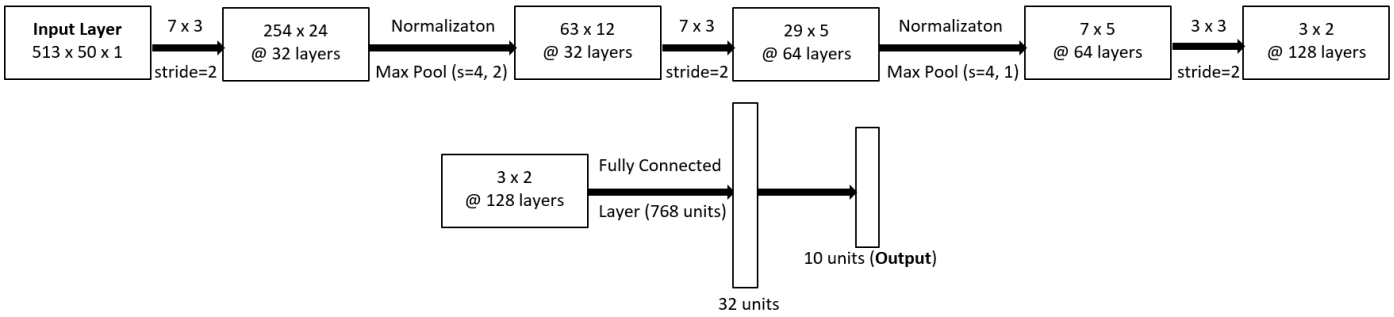
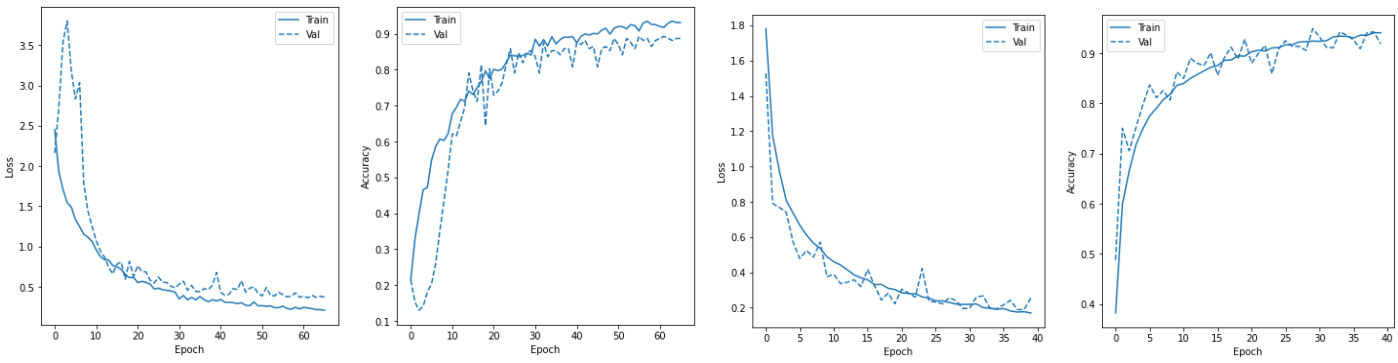


Figure-3: A model architecture for Task - 2

The model only gives event classification for each 'slice', i.e. STFT of duration equal to 0.5 seconds. We integrate predictions from each slice of the audio file. We reduce the error due to misclassification of individual slices by setting a minimum threshold 'prediction number', i.e. at least 6 out of 10 models must predict the majority class, else the prediction for that slice is set to null. We concatenate the predictions. Model summary:

Mean Training accuracy = 95.74%; Mean Validation accuracy = 94.67%.



(a) Task - 1 (b) Task - 2

Figure-4: Result analysis for both Tasks training for one iteration of 10-fold cross-validation

The pseudo-code to convert slice labels to the sequence of events is as follows:

```
// Step – 1: Remove susceptible errorous slice labels
// predictions is a NumPy array of size (number of slices, number of classes)
threshold = 6
for i in range(number of slices):
    row = predictions[i]
    if (all row elements < threshold):
        predictions[i] = NULL
```

```
// Step - 2: predictions are mapped to the event.
index = 0 // Tracks the slice number.
for i in range(number of samples):
    ans = ""
    curr = ""
    for j in range(number of slices in sample[i]):
        if (predictions!= curr):
            ans += "-" + result[idx]
            curr = result[idx]
        index += 1
    final_ans[i] = ans[1:]
```

POSSIBLE IMPROVEMENTS IN MODEL

- The training data was divided randomly into train and validation data. Due to possible repetitions in the training samples, we observe high accuracy in Task – 1 when we test the model on the training data itself. The confusion matrix for the same is as follows:

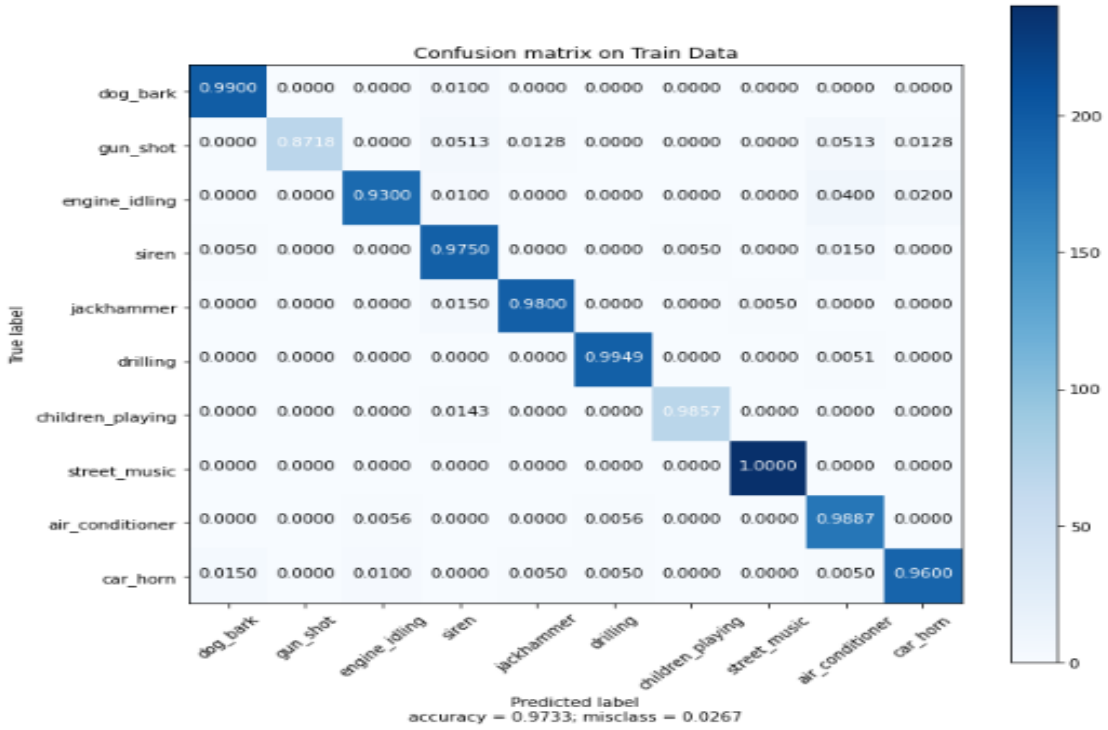


Figure-5: Confusion Matrix on Train data in Task – 1

To improve, we must adequately analyse the training data and ensure no repetitions in train and validation data.

- We assumed the minimum time window for event detection in Task – 2 equal to 0.5 seconds. Events may be of a lot smaller duration; it will result in an incorrect prediction. Also, hop size is equal to window size, i.e. we divide the audio into non-overlapping slices. Events with duration not equivalent to a multiple of 0.5 will lead to low confidence predictions.

  To improve, we may take overlapping slices of smaller with shorter duration