Aayush Goyal
A02259952

CS 6600
Project Report

Training & Testing Results:

Bee2 Image Set

	Training Accuracy	Validation Accuracy
ANN	94.3 %	95.5 %
CNN	99.4 %	98.88 %

Buzz Audio Set

	Training Accuracy	Validation Accuracy
ANN	42.9%	43.1%
CNN	90.1%	63.2%

Image Data

Preprocessing:

The number of neurons for image data were 1024 , so not much pre-processing was required except for converting the raw images into grey scale images and scaling them afterwards .

Training :

For ANN different architectures were tried with 2 , 3 , 4 or more hidden layers and neurons ranging between 50-120 in each layer. Mini-batch sizes and ETA values were randomly picked up from a list.
For CNN I tried 2 architectures , one with single convolution and 2 fully connected layers and other with 2 convolution and 2 fully connected layers. Second Architecture gave better results so I chose that one.
For Activation I used ReLu . An advantage to ReLU other than avoiding vanishing gradients problem is that it has much lower run time. max(0,a) runs much faster than any sigmoid function.
I kept the learning rate to 0.002 and Adam Optimizer simply because it has higher convergence rate. CNN gave far better results than ANN for image data with accuracy reaching 99% .

Audio Data

Preprocessing:
Processing Audio data was a very difficult task as each .wav file had around 82,000 input neurons so they had to be reduced in order to feed them into our neural network for faster training. Moreover training and test data had different neurons so they had to be the same. As we were not allowed to do the feature extraction using Spectrogram and similar techniques , Different approaches that I took to reduce input neurons:

1. Extracted data with maximum variance in the audio and only using that data to train the network.
2. Calculating information gain from the audio files and getting the data with maximum information gain. These two methods didn't help increasing the accuracy above 30%.
3. Down sampling the audio using Librosa reduced the neurons but the audio quality was not acceptable.
4. I also converted the .wav files into .mp3 files which reduced the size of each .wav file from 160kb to around 20 kb, but It was not happening in Realtime and there was no proper way or library to read .mp3 files and use them in training like .wav files.
5. Resampling with ***scipy.signal's*** resample method : It reduced the number of neurons according to number of seconds in signal but didn't improve accuracy.
6. ***Fast Fourier Transform*** – Lastly With SciPy's FFT I tried to perform Fourier transform on audio files which helped increase accuracy up to 63%.

Observations :

Validation accuracy on audio came on low number of EPOCH.

To reduce the number of neurons I divided took chunks of input data. This chunks were taken from starting, middle and last values of audio.

There was a lot of overfitting with audio CNN , to reduce it I used dropout with L2 regularization .

Accuracy on image data was pretty high, maybe due to less number of neurons and less preprocessing compared to audio data.

In audio data ANN sometimes predicts cricket sounds as noise.