Computer
SCIENCE

Vishal Sharma
A01789836

Project Objectives: This project has two objectives. The first objective is to train and test four neural networks: one ANN and one ConvNet that classify images and one ANN and one ConvNet that classify audio samples. The second objective is to compare the performance of ANNs and ConvNets on images and audio samples.

Project Goals:

This project aims at delivering classifiers for two problems *(a)* An image classifier to identify Bee *(b)* An audio classifier to identify sound of Bee, Cricket or Noise.

Image Classification

Given dataset contains total of 50863 images, where 25,444 belongs to images with Bee and 25,419 of images with No Bee, stats of dataset shown below:

	Bee	No Bee	**Total**
Train	19,082	19,057	38,139
Test	6,362	6,362	12,724
	25,444	25,419	**50,863**

Goal: Build a binary classifier to identify images containing Bee and No Bee.

Experimental Setup

Given dataset was used 'as is' for training and testing in the experiments, approx. split was 75% (training) - 25% (testing). The images were loaded using openCV and no pre-processing was done to the images. All 3 channels (R,G,B) of images were used in the experiment.

Using ANN

A Multi Layer Perceptron network was used to build a classifier with 6 fully connected layers of 1024, 1024, 512, 512, 128, 2 neurons as shown in Fig 1. Details about activation function used is in code.

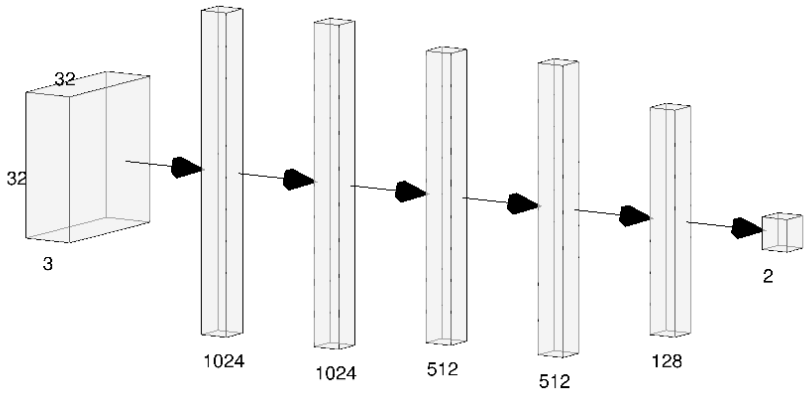

Figure 1: Architechture of ANN used for Image classification

Performance: Training was done for 500 epochs using Stochastic Gradient Descent (SGD) as optimizer with learning rate of 0.0001. Figure 2 displays accuracy during training.

Training Accuracy: 98.85%
Testing Accuracy: 97.52%

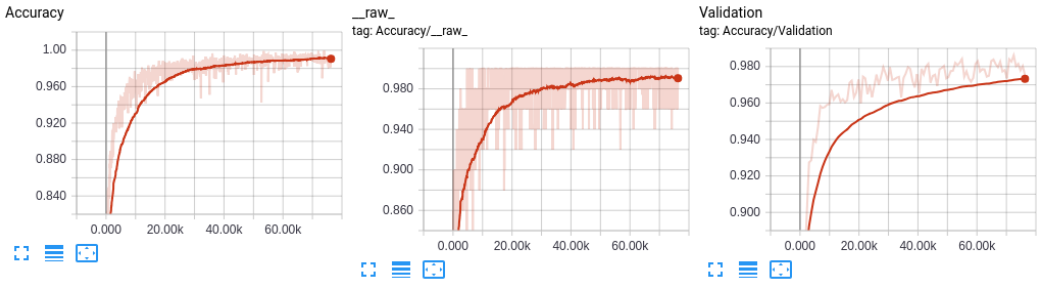

Figure 2: Performance of ANN for Image classification

Using CNN

A network using Convolution layers was used to build classifier, network architecture is shown in Fig 3. The *filter_size* for each convolution was 3 and *number of filters* was 32 and 64 for respective layers, details about activation function used is in code.

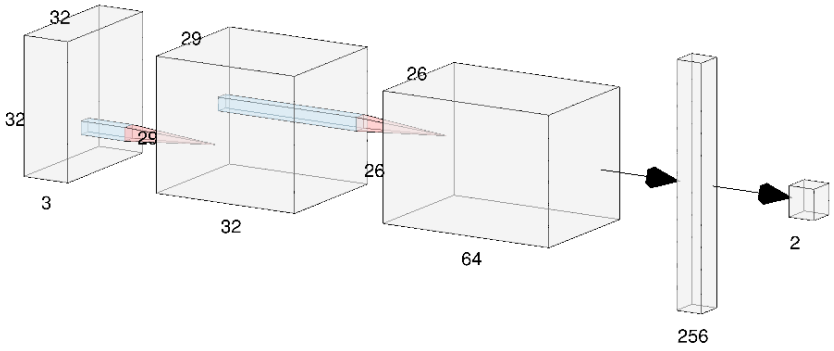

Figure 3: Architechture of CNN used for Image classification

Performance: Training was done for 500 epochs using Stochastic Gradient Descent (SGD) as optimizer with learning rate of 0.01. Figure 4 displays accuracy during training.

Training Accuracy: 100.00%
Testing Accuracy: 99.34%

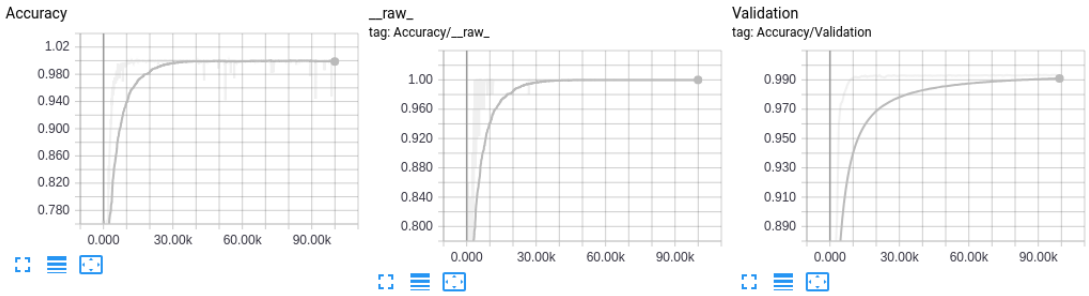

Figure 4: Performance of CNN for Image classification

Audio Classification

The objective of this project is to build a multi class classifier to identify sound of a bee, cricket or noise.

Audio Dataset

Given dataset contains total of 9,914 audio sample, where 3,300 belongs to Bee, 3,500 belongs to Cricket and 3,114 belongs to noise. Each audio sample is approximately about 2 sec long and has 44,100 amplitude samples/sec. Given dataset was merged and experiments were performed on 80%-20% split.

Table 1: Audio Dataset

	Bee	Cricket	Noise	**Total**
Train	2,402	3,000	2,180	7,582
Test	898	500	934	2,332
	3,300	3,500	3,114	**9,914**

Audio Data Preprocessing:

Audio dataset given has very high frame rate, on an average every file had 80,000 frames (amplitude/sec). With frames/sec being so high we have a lot of data and it needs some preprocessing. Reduction of audio frame rate and length was performed using interpolation technique. The audio sample was reduced to 15k sample and total length of 22,000 (approximately 1/4 reduction of the given audio).

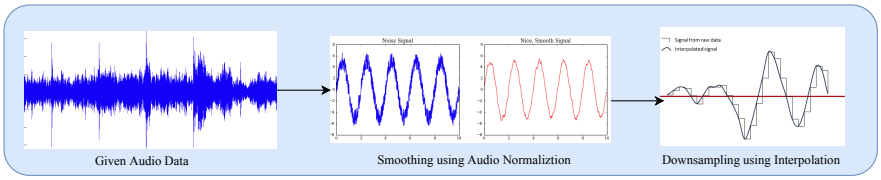

Figure 5: Audio Downsampling

Using CNN

A network using Convolution layers was used to build classifier, network architecture is shown in Fig 6. The *number of filters* for both convolution was 64 and *filter_size* was 10 and 3 for respective layers followed by 3 fully connected layers, details about activation function used is in code.

Max pooling was used after each convolution layer. During training over fitting was observed, to handle that dropout of 50% (keep) was used after first two fully connected layers and also 'L2' regularization was added to both layers. Input length was fixed as 22,000 with 1 channel.

During training it was also observed, without downsampling data model was not able to generalize well between bee and noise data. Adding downsampling technique helped the model in generalization.

PERFORMANCE: Training was done for 500 epochs using Adaptive Moment Estimation (adam) as optimizer with learning rate of 0.0001. Figure 7 displays accuracy during training.

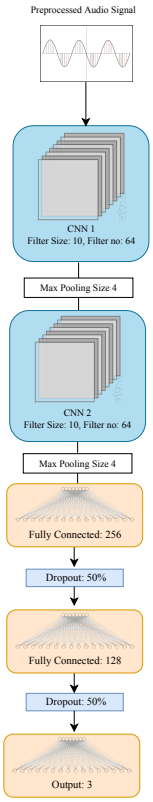

Figure 6: CNN for Audio Classification

Training Accuracy: 99.88%
Testing Accuracy: 99.45%

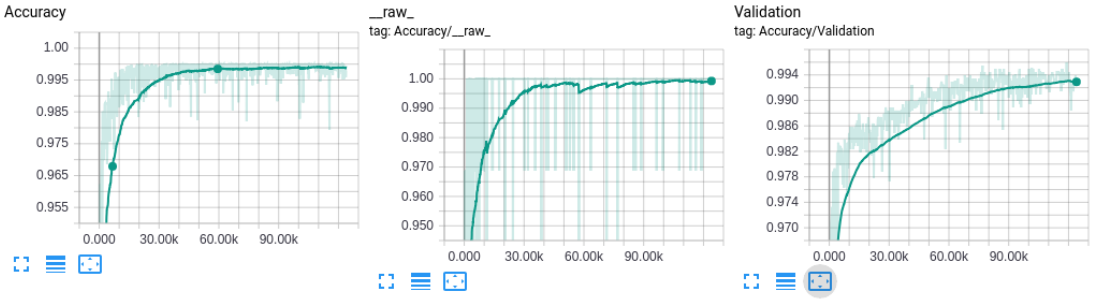

Figure 7: Performance of CNN on audio classification

Using ANN

During initial experiments ANN was not performing good and later after several experiments a Multi Layer Perceptron (MLP) model was build based on intuition of CNN. Where before feeding audio data in network it was max pooled in 3 different layers and output of pooled layers was given input to the fully connected layers as shows in Fig 8. To merge features extracted from different pooling layers output of fully connected layer was merged.

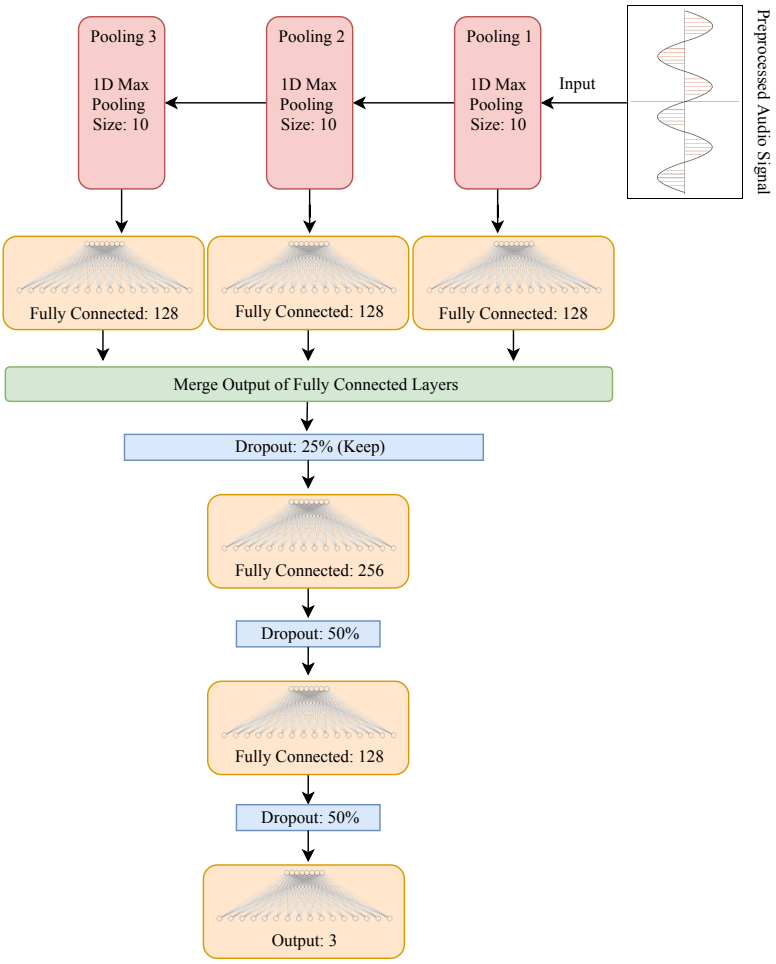

Figure 8: ANN for Audio Classification

PERFORMANCE: Training was done for 500 epochs using Adaptive Moment Estimation (adam) as optimizer with learning rate of 0.0005. Figure 9 displays accuracy during training.

Train Accuracy: 91.11%
Test Accuracy: 88.25%

Mentioned by Prof. Kulyukin, Fig 10 shows an attempt to analyze what could be happening when using multi pooling layer, followed by fully connected layer.

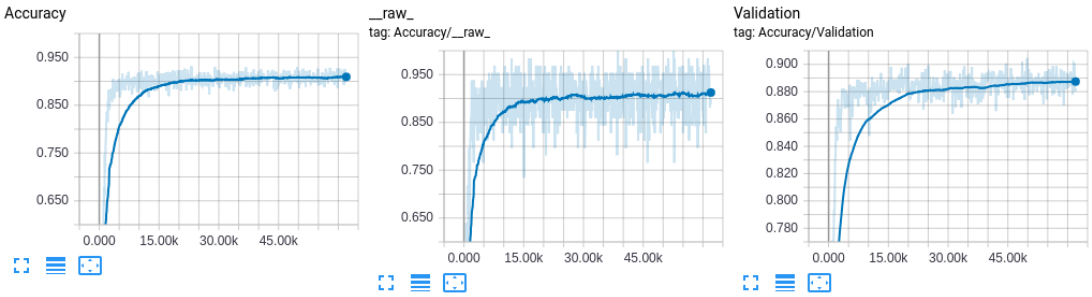

Figure 9: Performance of ANN on audio classification

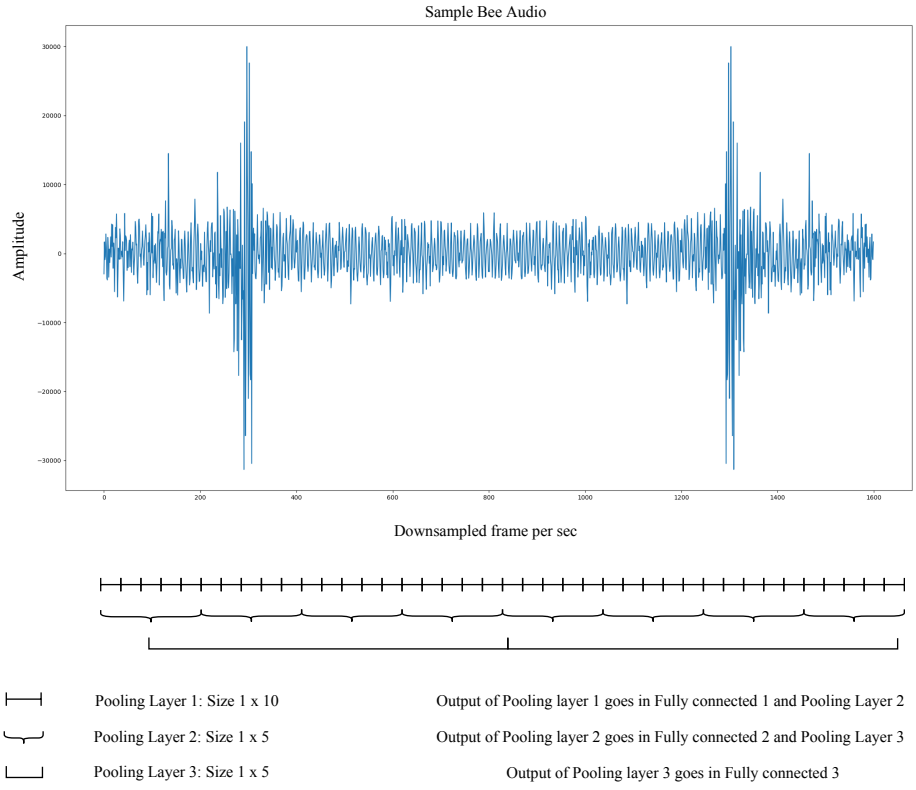

Figure 10: Sample Bee Audio and expected feature extraction using pooling layers and merging fully connected layers

Note: I would like to bring this to attention that during several stages of this project there were discussions about ideas between me (Vishal) and Prateek. The idea of using pooling for initial layers and merging fully connected layers was also one of the outcomes of such discussions.