Classifying Birds using Audio

Harsh Kumar Agarwal
2019423
harsh19423@iiitd.ac.in

Eshu Manohare
2019421
eshu19421@iiitd.ac.in

Lakshay Dabas
2019431
lakshay19431@iiitd.ac.in

1. Introduction

There might have been times when you may have heard a bird singing and couldn't quite figure out what the species or the name of the bird is? Here we are trying to solve that problem using ML and deep learning techniques. There can be various problems in solving such a problem such as the background noise present in the bird sound audio, multiple birds sound in a single audio, and imbalance in the data-set. The data-set that we are using is from xeno-canto database. We want to touch upon this problem so that we can help ornithologists, wildlife photographers to easily classify birds in an ecosystem and enable new methods for bird caretakers to groom and provide correct nutrition to the correct species.Also to evaluate the living conditions of a specific environment, it is necessary to know the information about the wild animals living there. Therefore, this may be used to figure out the diversity of birds living in a region.

2. Related Work

The data-set used is taken from xeno-canto database (source - Kaggle Link). We tried to scout for some baseline models on which model accuracy could be compared, however we were unable to find any. We were able to find a kaggle competition which took place on this dataset Link. Also we found a article which used some of the classes from this dataset Link. They were able to achieve 87% accuracy on 27 classes.

3. Data-set and Evaluation

We are using publicly available data of xeno-Canto bird recordings present in Kaggle (source -xeno-canto data-set) The data-set we downloaded is over 17GB in size with over 100 classes of birds. Therefore we reduced the data-set (reduced the number of classes to 5).We selected the classes with the maximum number of samples for our problem. In total we have 3830 samples. We split it in train val test split of 80, 10, 10 % respectively. Learning curves were plotted with 5-fold which helped us to understand the variation of loss on training and validation data with increase in the number of samples.

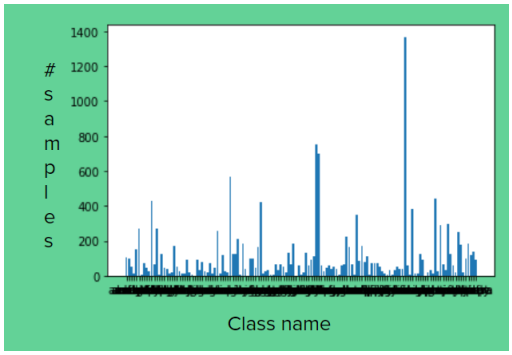

Figure 1. Number of Samples in each class

3.1. Feature extraction

We took mel spectogram image of audio clip as extracted features. In this we converted the audio files in the data to spectrogram images. We read this image as numpy arrays. Flattened the 2D matrix to single array. Then we applied PCA to reduce the number of dimension to get the final features for our ML models.

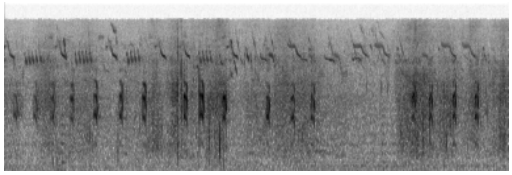

Figure 2. Mel Spectogram of a sample audio(GrayScale)

Then post midsem, various data augmentation techniques were applied to the audio data before converting them to mel spectrogram. The various data augmentation techniques that were applied were:
1. Changing the sampling rate of all audio files to 44100 hz
2. Converting all the mono(1 channel) and stereo(2 channel) audios to mono(Single Channel Audios)
3. The time range of the audio clips was between 3 secs to 2 mins. So the audios were all extended(by adding 0s in the array) or truncated to 4 seconds and 6 seconds(2 Different sets of data).

4. Time Shift was applied to all the audio files in the 6 sec data.

No time shift was applied to 4 seconds long database since 6 seconds long database(with time shift as well) when trained and tested on deep learning models did not perform well(as we will see in Results and Analysis section).

The 4 seconds long audio files database had only applied the first two data augmentation techniques.(More on this in the Results and Analysis Section)

Then after applying the data augmentation techniques, the audio files were converted to Mel-Spectrograms(Color) and saved as images to be used later for training deep learning models.

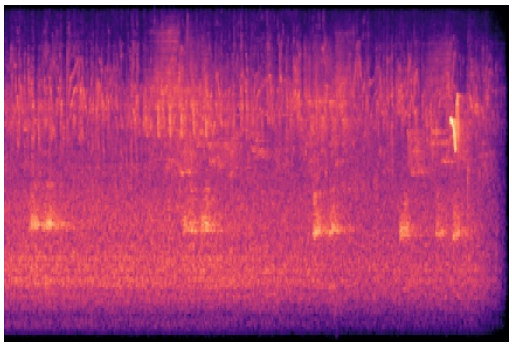

Figure 3. Mel-Spectrogram of a sample audio(Before Data Augmentation)

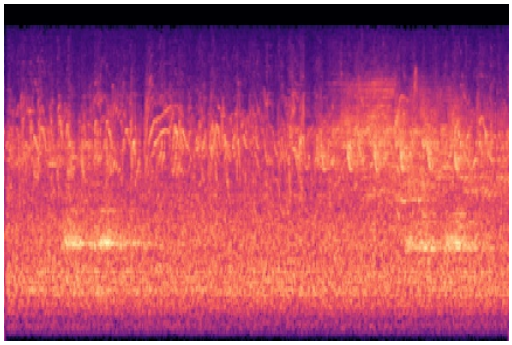

Figure 4. Mel-Spectrogram of a sample audio(After Data Augmentation)

3.2. Evaluation metric

We used classification avg. precision, avg. recall, avg. F1 and accuracy for evaluation.

4. Methodology

We first extracted the top 5 classes with the most number of samples for our data. These classes were comrav,houspa,houwre,redcro and sonspa. These classes had the most number of samples out of the 113 classes in the original dataset. Total number of samples in these top 5 classes were 3830. We then converted these audio samples to mel-spectrograms and stored them as images in a different folder for training the models. We then finally used these images as the data for classification. We first started by apply logistic regression without any penalty. Since we had image data, the flattened 2d array images had large number of features(51,076). We then applied PCA on the data to reduce number of features for training model in less time as 51,076 features for training complex models would take a lot of time. PCA with 0.99 variance also had large number of features and the models still were taking a lot of time to train. After trying multiple values of variance, we concluded variance = 0.95 such that it would reduce the number of features without losing a lot of data.

After applying PCA(variance = 0.95), the number of features reduced to 623. We then applied various model on the data and plotted learning curves (loss vs Training Samples) for various model to assess overfitting and underfitting of the models. In Naive Bayes Classifier we can observe that as training samples increased, the training score decreased and both training and validation score were becoming stagnant. From the graph(Figure 10) its' clear that the model was trained well. Similarly we trained all models and changed the hyperparameter to get the best performance out of them. We changed the hyperparameters by looking at the learning curves. Logistic regression with L2 regularization(400 vs 1500) iterations showed the same performance in terms of loss score. From the curve it can be interpreted that the model is underfitting since the model is very simple. We moved on to other models. SVM with 500 vs Max Iterations both reached a fixed loss value after evaluating on all the training samples and both SVM models (500 iterations vs Max Iterations) showed similar performance in terms of accuracy and F1 score.

Even after training the base models correctly, the highest accuracy we were able to achieve was 58 percent. This indicates that the data used for training is not of good quality. To get better data and better features, we did pre-processing and data augmentation of audio data to get better quality of features for better training and accuracy of models.

The augmented audio data was then converted to mel-spectrograms.We first applied all the above mentioned data augmentation with length of audio files as 6 seconds. Then this audio was converted to mel-Spectrograms(Color) and saved. These images were used to train two different deep learning models resnet(50) and efficient net(B3).

In both model we freezed the convolution layers and

added fully connected layers as the classifier. The fully connected layer comprised of a linear layer(inputSizeX512) followed by ReLU, dropout, linear layer(512X10) and a softmax layer. We calculated the NLL(negative log likelihood) loss and optimized using Adam optimizer with a learning rate of 0.003. We trained the models for 30 epochs. When analyzing the 6 second database, it was observed that there were a lot of files that were of length 3 seconds and 4 seconds, because of which most of the mel-Spectrograms resulted in black patches in the image.(See Fig Below)

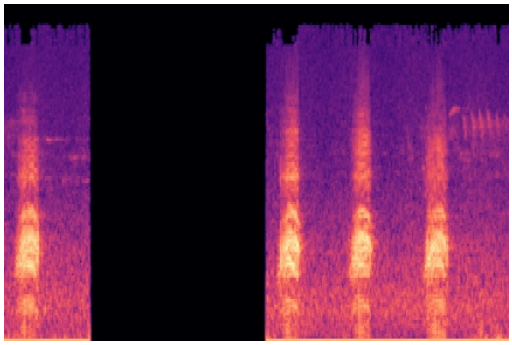

Figure 5. A sample mel-Spectrogram from 6 seconds length audio database

The black patch is the because of the zeros added in the array along with the time shift done to the data. The loss vs iteration graph of this dataset(6 sec) is shown below(Fig.17 and 18)

Hence a new dataset with length of audio file 4 seconds was created after setting audio length = 4 seconds and remove time shift technique from augmentation. This new dataset was used to train the above deep learning models and the loss vs iteration graph of this dataset(4sec) is shown below(Fig.19 and 20)

Then a new dataset with no time constraint was created. Mel-Spectrograms were created of audio files without changing their length. Hence some mel-spectrograms could be of an audio file of 1minute40seconds whereas some mel-spectrograms were of audio files of 4 sec length only. This dataset(No length constraint) was used to train the above deep learning models and their loss vs iteration graph is shown below(Fig.21 and 22).

Now from Figure 20 it can be seen that even after 30 epochs, the validation and training losses are still changing. Hence, the resnet model was trained again on 50 epochs(Figure 21) and it can be seen that the losses did plateu on 30 epochs only. Hence 30 epochs were used on all the models.

Now finally, one more model was trained using the dataset of grayscale mel-Spectrograms that was created first. The train vs loss curve of this model is shown below(Fig. 6). Surprisingly, this model performed best out of all the models trained so far accuracy wise.

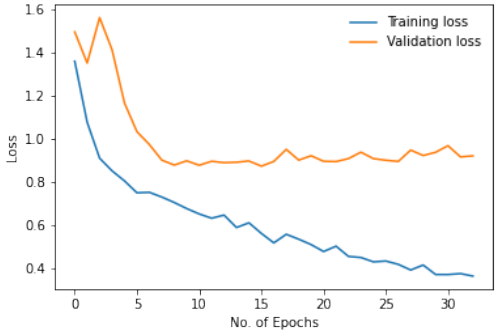

Figure 6. Iteration vs Cross Entropy Loss graph of Efficient Net(Black and White Mel-Spectrogram database)

5. Results and Analysis

The metrics table(Figure 15) shows Precision,Recall Score, F1 Score and Accuracy for all of the base models trained. From the metrics we can see that SVM performed best with a F1 score of 0.52 and an accuracy of 58 percent(confusion matrix Figure 16). Precision and recall scores of the various models tell us about the False Positive and True Positive of the predictions. In case of Random Forest with OVR we can see that there is a large difference different between precision and recall score. This indicates that this model would perform better at detecting false positives in the data but would fail in detecting true positives in the data.This means that though Random Forest with OVR model has an accuracy of 51 percent, but it would fail in detecting true positives in the data. Precision and recall scores for SVM are better than Random Forest indicating that SVM will perform better than Random Forest in every aspect.

Now, the metrics table shown below(Figure 7) shows the accuracy on test set of deep learning models trained on different datasets discussed above.

Model	Accuracy on test set
Resnet(6 sec length database)	70.05208
Efficient Net(6 sec length database)	70.05208
Resnet(4 sec length database)	67.968
Efficient Net(4 sec length database)	69.531
Resnet(No length constraint database)	59.3
Efficient Net(No length constraint database)	63.02
Efficient Net(Black and White spectrogram database)	71.018

Figure 7. Evaluation Metrics for different deep learning models

Now, we can see the accuracies of different models. RestNet and Efficient net with 6 sec database on which all the data augmentation techniques were applied, achieved an accuracy of 70.05 percent. Now, we reduce the length of the audio files from 6 sec to 4 sec and remove the time shift from the files to prevent black patches in the mel-Spectrogram images. After making a new dataset of 4 sec length mel-Spectrograms, the models were trained again accuracy of 67.968 was observed with Resnet and accuracy of 69.531 was observed with efficient net model.

Now, one more dataset was created with no-length constraint on the audio files and mel-spectrograms were created using this to check the influence of length on the performance of the model. This dataset performed the worst out of all the deep-learning models with an accuracy of 59.3 percent with resnet and 63.02 percent with efficient net.

Now the best performing deep learning model was the one that was trained with the black and white mel-spectrogram data. No data augmentation techniques were applied on this data and the audio files were simply converted to grayscale mel-Spectrograms. This model(Efficient Net) achieved an accuracy of 71.018 which is the highest among all the models.

The confusion matrix for the models Efficient Net(4 sec length database) and Efficient Net(Black and White Mel-Spectrogram Database) is shown below which are two of the best performing models.

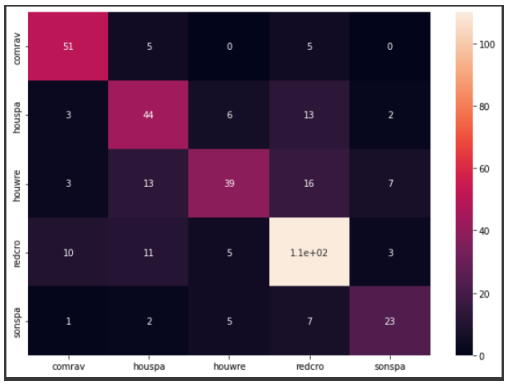

Figure 8. Confusion Matrix for Efficient Net(4 sec length database)

The dataset we are using is not exactly the same as that in the article. Because of computing restraints we are limiting our number of classes to 5. Therefore we cannot directly compare the SOTA(state-of-the-art). Our best model achieved an accuracy of 71.05 percent with 5 classes whereas the model from the article mentioned above(Related Work) achieved an accuracy of 87 Percent on 27 classes.

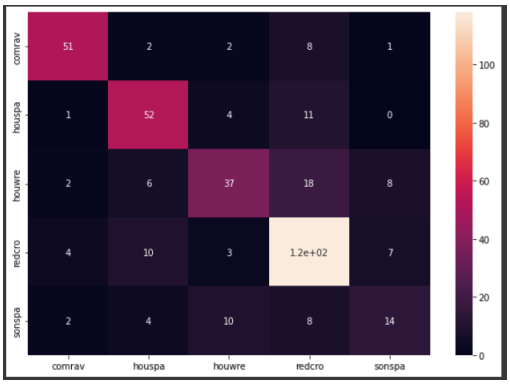

Figure 9. Confusion Matrix for Efficient Net(Black and White Mel-Spectrogram Database)

6. Contributions

6.1. Deliverables

All the deliverables mentioned in the proposal were completed by each team member
Harsh Kumar Agarwal: Work on advance Model(Resnet), sampling data from original downloaded data to suit computing constraints.
Eshu Manohare: Work on advance Model(EfficientNet),looking at evaluation metrics, error analysis and model refinement
Lakshay Dabas: Feature extraction, data augmentation on image data extracted from audio data

6.2. References Citations

Audio Deep Learning made simple sound classification
Sound based bird classification
Reference presentation link
Plotting met spectrogram images
Kaggle competition public available codes for reference
confusion matrix to f1 score
Training model in pytorch
Google colab

6.3. Individual Contributions

Harsh Kumar Agarwal: Wrote scripts for Downloading data, making Final data, training scripts
Eshu Manohare: Wrote script of data visualisation, learning and loss curves
Lakshay Dabas: Wrote scripts for data augmentations(with different folders), confusion matrix plots

All the work was done on google colab. Each file contain many functions, each was the result of contribution from every team member.

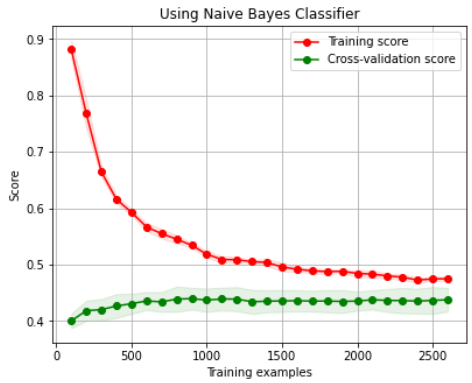

Figure 10. Naive Bayes learning curve

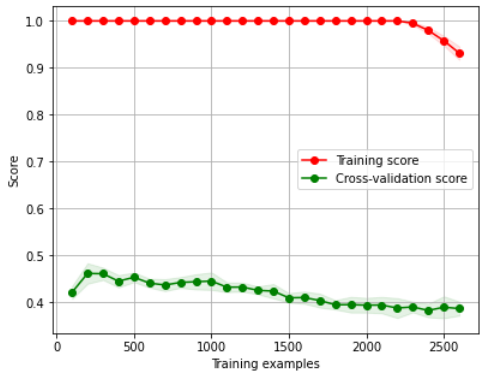

Figure 11. Logistic L2 learning curve(1500 iterations)

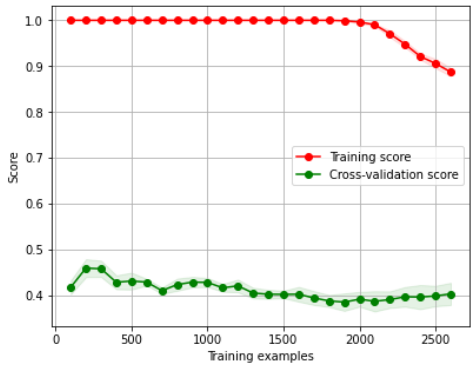

Figure 12. Logistic L2 learning curve(400 iterations)

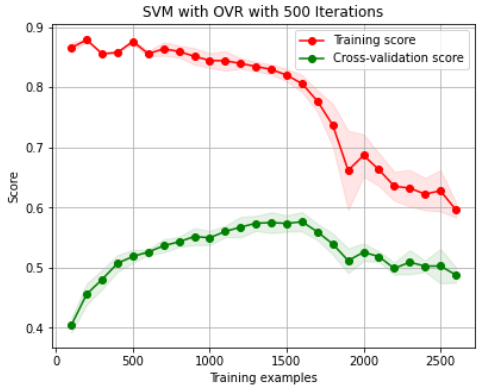

Figure 13. SVM learning curve(500 iterations)

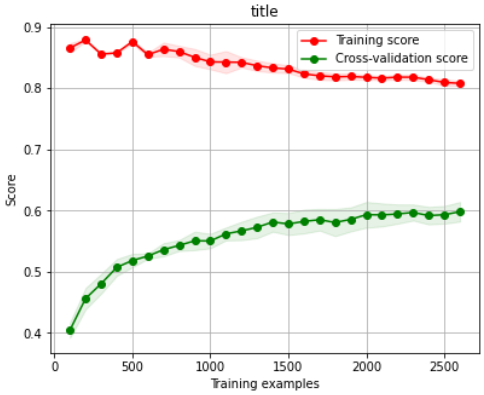

Figure 14. SVM learning curve(Max Iterations)

Model	Precision	Recall Score	F1 Score	Accuracy
Naive Bayes	0.381472096	0.307188070	0.29457828	0.42819843
Logistic without penalty	0.471282227	0.472515357	0.46708056	0.50652741
Logistic with L1 regularization	0.513266388	0.490832437	0.49600994	0.54830287
Logisitc with L2 regularization	0.468811709	0.471834970	0.46493801	0.50130548
Random Forest	0.611998658	0.400983629	0.40480375	0.51958224
Random Forest with OVR	0.700750194	0.380790492	0.38310905	0.51174934
SVM	0.617870200	0.507361762	0.52891103	0.58224543

Figure 15. Final Metric Table for base models

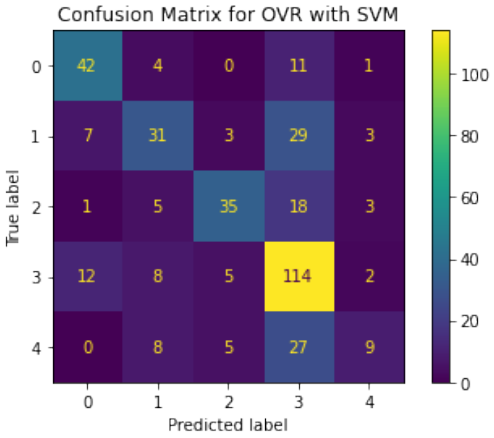

Figure 16. Confusion matrix of SVM

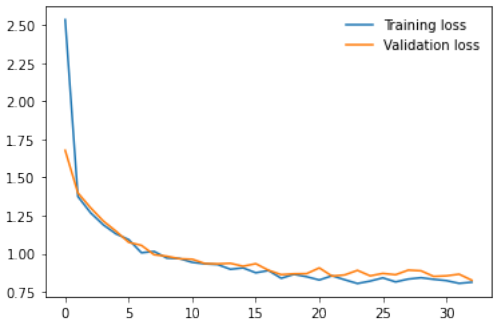

Figure 17. Iteration vs Cross Entropy Loss Graph of Resnet(6 sec database)

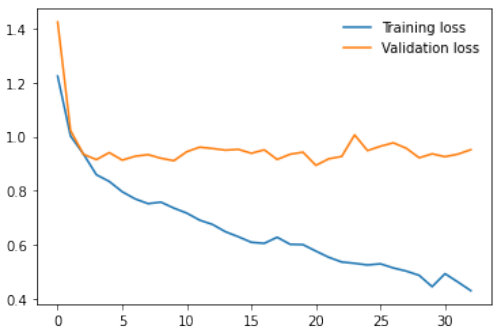

Figure 18. Iteration vs Cross Entropy Loss graph of Efficient Net(6 sec database)

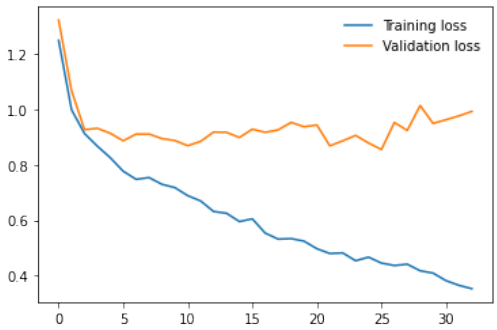

Figure 19. Iteration vs Cross Entropy Loss graph of Efficient Net(4 sec database)

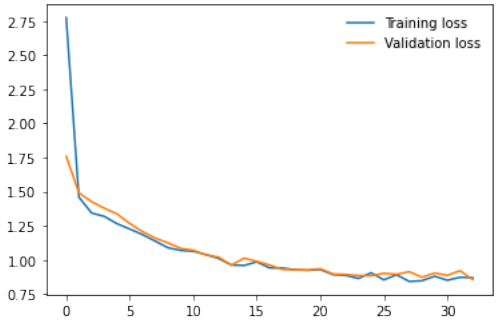

Figure 20. Iteration vs Cross Entropy Loss graph of Res Net(4 sec database)

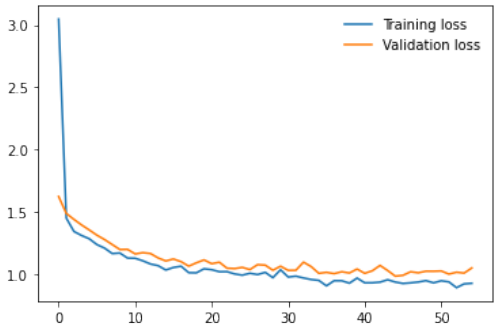

Figure 21. Iteration vs Cross Entropy Loss graph of Res Net(No length constraint database)

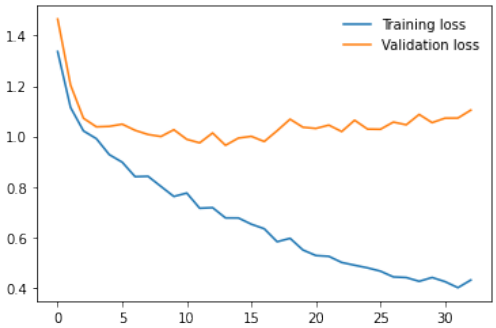

Figure 22. Iteration vs Cross Entropy Loss graph of Efficient Net(No length constraint database)