Bird Sound Identification Using Deep Learning

Ahaz Bhatti Ha Phan Nilisha Makam Prashantha Shruthi Sathish

I. Motivation

In recent times, there has been significant research on bird species identification contributing to the analysis of migration patterns in specific geographical regions. This enables scientists to enhance their expertise in bird habitats and restoration projects. However, conducting these surveys and collecting data is logistically and financially challenging. Combining passive acoustic monitoring (PAM) with deep learning technology can help with the sampling of larger spatial scales and the acquisition of higher temporal resolution data. This, in turn, would help the scientist by enhancing and aiding the analysis of the avian species.

II. Background

Birds have unique audio-visual characteristics that can help distinguish between species. An automatic wildlife monitoring system for bird song identification using bioacoustics is important for conservation efforts and species protection. The collection of bird sounds is widely collected manually and requires well-trained specialists to gather and analyze data from reliable sources. This research aims to develop an application that is comprehensive and uses sophisticated deep-learning models to classify the birds based on their vocalizations. The proposed approach reduces the resources and time required for the collection, analysis, and classification of data.

III. Literature review

The research by [10] compared SVM models with segmentation algorithms and feature selection to bird call classification models. SVMs demonstrated high accuracy in multi-class bird audio classification for a smaller number of species, but their performance deteriorated with an increase in bird species classes. Recent advancements in machine learning classifiers utilizing spectrograms have shown significant improvements in bird species recognition. To address the challenges of multi-class bird classification, a new technique of bird audio diarization system with faster RCNN was proposed. The research achieved promising results on the Bird Songs from Europe dataset.

The research [7] highlights the use of Vision Transformers (ViT) for audio classification, specifically Mel spectrogram images. By using ViT and time slices of spectrograms as input patches to the STFT Transformers, the researchers achieved promising results. Despite the challenge of training transformers with limited audio data, they employed fine-tuning techniques with pre-trained weights from vision transformers trained on large image datasets, which lead to improved performance and a sored 11th rank in the BirdCLEF 2021 competition.

The research paper [8] discusses the implementation of transfer learning using the VGG16 model. It focuses on classifying images of the German Traffic Sign Recognition Benchmark (GTSRB) with improved accuracy through hyper-parameter tuning. The paper explains Keras as a deep learning framework and VGG16 as a pre-trained architecture for image classification. The paper emphasizes the benefits of transfer learning and its efficiency in tasks like image classification, offering valuable insights for implementing the process.

In their research paper [5], the authors introduce Mobilenet, a novel architecture designed for devices with fewer resources like mobile phones and embedded systems. The depth-wise convolution used by MobileNetresults in fewer parameters and lower cost of computation. The architecture consists of 28 layers, each followed by batch normalization and the ReLU activation function. Experimental evaluations reveal Mobilenet's effectiveness in achieving a balance between complexity and accuracy, with competitive results in object detection, such as achieving a mean Average Precision (mAP) of 19.3% with 6.8 million parameters, outperforming VGG and Inception models with higher parameter counts.

Finally, In the research [11] aims to identify different environmental sounds based on audio data. They propose innovative approaches to address the challenge of recognizing subtle differences between sound classes while minimizing information loss and gradient degradation. Their proposed architecture includes a multi-input 3D CNN model that uses stacked spectrograms and 3D convolutions to capture temporal dependencies. The model outperforms other 1D sequential CNN and 2D ResNet50 models. The fusion network in the proposed model effectively reduces information loss and enhances performance in complex sound scenes, highlighting the limitations of 1D models in capturing discriminative features directly from raw waveforms and the value of alternative approaches like the proposed 3D sub-branch architecture.

Based on all the above literatures, the team has made the decision to use Melspectrogram as the audio representation and employ a combination of pre-trained Convolutional Neural Networks (CNNs) and transformer architectures. These models have been pre-trained on previous BirdCLEF 2021 and 2022 audio data, allowing better feature extraction and classification performance. This approach uses the knowledge and insights gained from previous bird audio classification tasks to enhance accuracy. This research introduces a combination of different architectures, such as Mobilenet-CNN, EfficientNet-CNN, VGG16-RCNN, and a hybrid EfficientNet

VisualTransformer leading to improved accuracy and ability to capture complex features. The hybrid approach may leverage the spatial information captured by CNNs and the attention mechanism of transformers. Finally, the research also provides a comprehensive architectural comparison that has not been done before.

IV. Methodology

A. Data Collection

The audio dataset for the project is collected from the BirdCLEF 2021 [2], 2022 [3], and 2023 [4] competitions hosted on the Kaggle website. The aim of the BirdCLEF competition is to use sound to identify worldwide bird species. This competition provides well-annotated and clean audio recordings of 264 bird categories. This helped the team conduct better experiments and reduce the need for data cleaning. Each folder contains training data along with a metadata.csv file. All audio recordings are downsampled to 32 kHz and converted to .ogg format, which ranges from 5 seconds to 1 minute in length. The BirdCLEF 2021 and 2022 datasets are used for pretraining the models, and the 2023 dataset is used for finetuning the models. The data combined from BirdCLEF 2021 and 2022 contain approximately 64K records, while the 2023 dataset contains approximately 20K audio recordings of birds. The entire dataset is approximately 54.08 GB of memory. Fig. 1 and Fig. 2 show the sample metadata of the BirdCLEF 2021/22 and BirdCLEF 2023.

	filename	filepath	primary_label	secondary_labels	rating	author	xc_id	scientific_name	common_name	target	birdclef	fold
0	XC109605.ogg	/kaggle/input/birdclef-2021/train_short_audio/...	acafly	['amegfl']	2.5	Mike Nelson	XC109605	Empidonax virescens	Acadian Flycatcher	0	21	0
1	XC11209.ogg	/kaggle/input/birdclef-2021/train_short_audio/...	acafly	[]	3.0	Allen T. Chartier	XC11209	Empidonax virescens	Acadian Flycatcher	0	21	1

Fig. 1. Metadata of BirdCLEF 2021 and 2022 Combined

	primary_label	secondary_labels	type	latitude	longitude	scientific_name	common_name	author	license	rating	url	filename
0	abethr1	[]	['song']	4.3906	38.2788	Turdus tephronotus	African Bare-eyed Thrush	Rolf A. de By	Creative Commons Attribution-NonCommercial-Sha...	4.0	https://www.xeno-canto.org/128013	XC128013.ogg
1	abethr1	[]	['call']	-2.9524	38.2921	Turdus tephronotus	African Bare-eyed Thrush	James Bradley	Creative Commons Attribution-NonCommercial-Sha...	3.5	https://www.xeno-canto.org/363501	XC363501.ogg

Fig. 2. Metadata of BirdCLEF 2023

B. Data Preprocessing

1) Insufficient Sample Size: The available dataset has a limited number of samples for certain bird classes, some of which may have only one sample. To ensure these classes are adequately represented in the training data, a filtering approach is implemented. The process involves defining a threshold value (e.g., thr=5) to determine the minimum number of samples required for a class. A new column called "CV" indicates whether a sample should be included in the cross-validation (CV) process. For classes with fewer than five samples, the CV column is set to "False" to make sure these samples are always included in the training data and not used for cross-validation. This approach helps address the issue of insufficient samples for certain bird classes and ensures they contribute to the training of the classification model. The data divided using the filtering process is shown in Fig. 3.

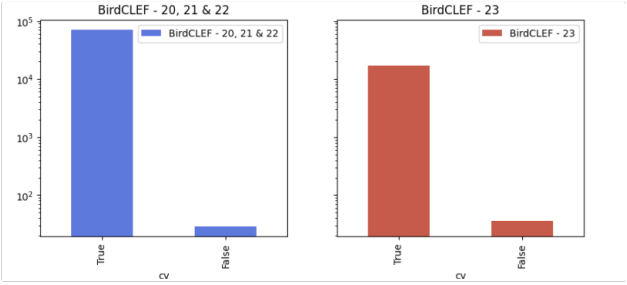

Fig. 3. Applying Cross Validation Filter on Data

2) Handling Biased Data: Despite the filtering process, the filtered dataset still contains minority classes with a limited number of samples, resulting in an imbalanced distribution known as the 'Long Tail' problem. To solve this issue, upsampling and downsampling are done on the data as shown in Fig. 4.

- Upsampling: The train data for minority classes with very few samples are increased by duplicating or generating synthetic samples. This technique helps to balance the representation of the minority classes and provides more training examples.
- Downsampling: It is applied to ensure that the maximum number of samples for each class is controlled. This involves reducing the number of instances for classes that have an excessive number of samples, ensuring more balanced data for all classes.

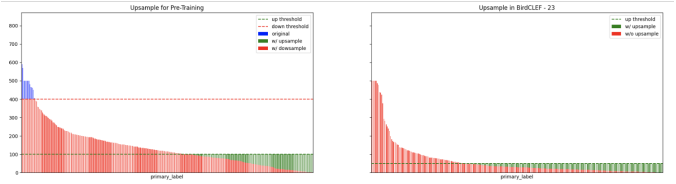

Fig. 4. Upsampling and Downsampling Training Data

3) Pre-processing Pipeline: The preprocessing pipeline involves audio decoding that converts raw audio from .ogg into a different format that can be used for processing and extracts spectrograms for analysis. Audio augmentation techniques such as adding noise, cropping, and mixing samples increase dataset diversity. Some commonly used audio augmentation techniques include adding Gaussian noise to simulate real-world variations, performing random crop and padding to extract different sections of the audio, applying CutMix and MixUp to combine audio samples from different sources, enhancing the dataset's robustness, and reducing overfitting. Normalization is used to standardize the data, which involves calculating the mean and standard deviation of the data and then scaling the values to a range between 0 and 1. Normalization helps to remove any biases and ensures that the data is centered around zero. The Spectrogram augmentation

techniques like TimeFreqMask introduce variations in the spectrogram to enhance model performance where random time and frequency segments of the spectrogram are masked or removed, helping to model to generalize well. These steps collectively contribute to the effective preprocessing and preparation of audio data for machine learning tasks, as shown in Fig. 5.

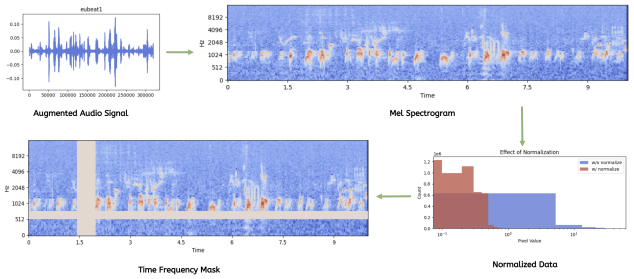

Fig. 5. Data Pre-processing Pipeline

C. Data Preparation

During data preparation, StratifiedKFold is used on the data to split it into five cross-validation folds to contain proportional target variables. However, in some cases, certain classes may have a limited number of samples, resulting in not all folds containing all the classes, and the data is shuffled to avoid overfitting. For pre-training, the data from BirdCLEF competitions in 2021 and 2022, with an image size of (128 and 384) is used as input. The data is divided into batches of size 256 for efficient processing during training. The training set consists of 60,146 samples, while the validation set had 14,183 samples. For fine-tuning, the dataset is first divided into five cross-folds. Fold0 with 3,381 samples is used as the validation data, and fold4 with 3,416 samples is used as test data. The rest of the samples are used for training data after upsampling. So, the training data set contains 22,208 samples. Also, the data marked with cross-validation filters are not added to the training data. The pre-processing and data preparation is done based on the code available at [2]. However, the data split is done for testing using the same 2023 dataset for our research, and the evaluation metrics are changed to manual accuracy calculation based on correct and misclassified records in the test set.

D. Models Used

1) MobileNet CNN: This project experimented with a combination of the MobileNet model and a custom CNN on the birds sound dataset. MobileNet is known as a deep neural network architecture that utilizes depthwise convolution instead of normal convolution. By using this method of convolution, MobileNet is effective in terms of less complexity, number of parameters, and computational cost compared to other deep neural network architecture. Basically, the original MobileNet has 28 layers. Using pre-trained MobileNet on ImageNet dataset as parameter initialization, then remove its classifier layer and add a custom CNN with 3 conv layers along with classifier layer to train our bird sound dataset. For custom CNN, kernel size 3x3, 2D convolution, and RELU activation function are used. The number of kernels is increased after each layer with values of 32, 64, and 128, respectively. This custom CNN architecture is used since it gives the best performance among multiple variations experimented on the dataset. The architecture of this MobileNet-CNN is shown the Fig. 6.

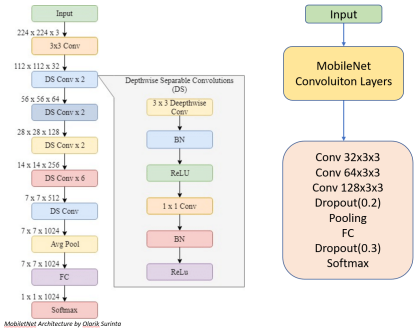

Fig. 6. MobileNet-CNN Architecture

2) VGG16 RCNN: When recognizing bird sounds, we first convert the audio into a visual representation called a spectrogram. This helps us understand which frequencies are present at different moments in time. After the Randomization/Masking layer, the processed spectrogram goes through the VGG16 backbone model that analyzes the spectrogram to find important features representing different bird sounds. The features are then passed to subsequent layers, including the RoI pooling layer, which focuses on the most important parts of the data. This pooling step summarizes the features and the dense layers and further analyzes and processes these summarized features to classify and predict the bird species based on the patterns it has learned. These layers work together in a step-by-step manner to extract and use meaningful features to accurately recognize bird sounds, as shown in Fig. 7.

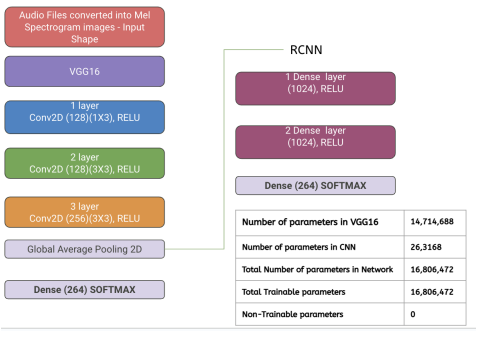

Fig. 7. VGG16-RCNN Architecture

3) EfficientNet CNN: The project's third variant involves experimentation with a basic Convolutional Neural Network (CNN) model, drawing inspiration from VGG principles built

upon the EfficientNet B1 backbone using transfer learning. However, certain modifications have been made, such as incorporating global average pooling for flattening and introducing dropout layers as a regularization technique before flattening. The decision to incorporate a 2-dimensional Convolutional Neural Network (CNN) was inspired by the findings of [11], which indicated that 2D or 3D CNNs are more effective in capturing spatiotemporal characteristics of input signals such as Mel spectrograms. The complete model architecture is presented in Fig. 8. The network contained a grand total of 7,569,560 parameters.

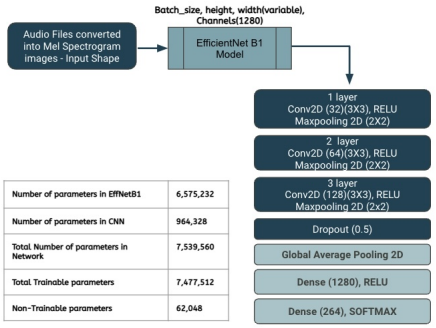

Fig. 8. EfficientNet-CNN Architecture

4) Hybrid EfficientNet VisualTransformer: The architecture depicted in Fig. 9 presents the Hybrid EfficientNet VisualTransformer, which incorporates various input normalization and augmentation techniques such as TimeFrequency Mask and CutMix. The backbone of the model consists of the pre-trained EfficientNet that employs a compound scaling method to uniformly scale the network's depth, width, and resolution, ensuring a balance between accuracy and efficiency. The data is sent to the normalization layer before passing through a multi-head self-attention layer to capture intricate relationship patterns. Dropout regularization is implemented to reduce overfitting and enhance the model's generalization abilities. The output of the normalization layer and the multi-head attention layer undergo residual connection, followed by another round of normalization and subsequent processing through a multi-layer perceptron (MLP) with an additional dropout layer for regularization. The output of the second normalization layer and the MLP are combined in a residual block, average-pooled, and connected to an output layer with Softmax activation, producing 264 classification outputs. The trainable parameters for this network vary between 23 million to 63 million, depending on the number of attention heads. For this research, eight attention heads are used, making the total trainable parameters approximately 63 million.

5) Evaluation Metrics: Loss refers to a measure used to quantify the inconsistency between the predicted output of a machine learning model and the true or target output. In classification tasks, Categorical Cross Entropy (CCE) is a commonly used loss function. It compares the predicted class probabilities with the true class labels and calculates the average logarithmic loss across all classes. The formula to

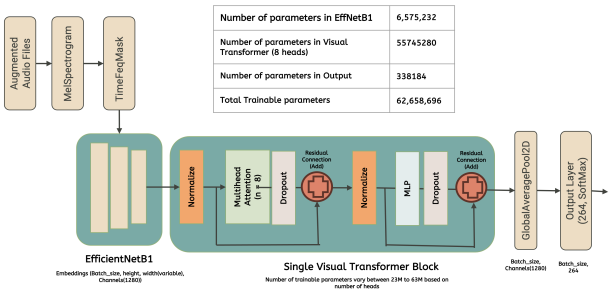

Fig. 9. Hybrid EfficientNet Visual Transformer Architecture

calculate CCE is shown in equation 1 where y_true is the true or target probability distribution (one-hot encoded) of the class labels and y_pred is the predicted probability distribution of the class labels.

$$CCE = -\sum(y_{true} * log(y_{pred})) \tag{1}$$

Performance metrics are used to evaluate the effectiveness of a model. Two commonly used metrics in classification tasks are AUC (Area Under the Curve) and Accuracy. AUC represents the overall quality of the model's predictions by measuring the area under the Receiver Operating Characteristic (ROC) curve. It provides a single value that summarizes the model's ability to discriminate between different classes. Accuracy measures the proportion of correctly classified instances out of the total number of instances. It is a straightforward metric that is easy to interpret and understand. The formula used to calculate accuracy in this research is as shown in equation 2 where y_pred is the total number of predictions made by the model, and y_missclass is the number of misclassified samples.

$$Accuracy = ((y_{pred} - y_{missclass})/y_{pred}) * 100 \tag{2}$$

V. Experiment Results

The models used in this research were pre-trained with a batch size of 256 and image size of (128, 384) on a dataset with 60,146 training samples and 14,183 validation samples. For the fine-tuning phase, a dataset of 19,627 training samples, 3,381 validation samples, and 3,416 test samples is utilized. A cosine learning rate scheduler is applied that generates a learning rate function to compute the appropriate learning rate for each epoch as shown in Fig. 10, and the learning curve is plotted based on the area under the receiver operating characteristic (AUC) curve, and the accuracy is also calculated. The number of epochs for pre-training and fine-tuning varied depending on when the models ceased to exhibit further improvement. The hybrid transformer model was trained for up to 25 epochs, whereas the CNN models were trained for a maximum of 15 epochs. All experiments were performed on a TPU VM v3-8 on Kaggle, with the average time required for pre-training the models being approximately three hours, while the duration of fine-tuning varies between 20 to 40 minutes, depending on the complexity of the model architecture.

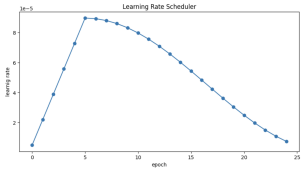

Fig. 10. Learning Rate Scheduler

1) MobileNet CNN: For the MobileNet-CNN, various experiments were conducted with variations of parameters like dropout, optimizer, and learning rate to evaluate the performance of each experiment and to find out the model achieving the best results. Total number of parameters of this model is approximately 3.3 million. The final model gave the result of 0.64 and 2.5 for AUC and Loss, respectively, for validation. Fine-tuning is performed on the 2023 dataset, achieving an AUC of 0.70 and a loss of 1.5 for validation. Fig. 11 shows the learning curve for the pre-training and fine-tuning phase. It can be seen that in both the pre-training and fine-tuning phases, the performance of validation is higher than in training. This could be because the validation dataset is simpler for the model to classify compared to the training dataset.

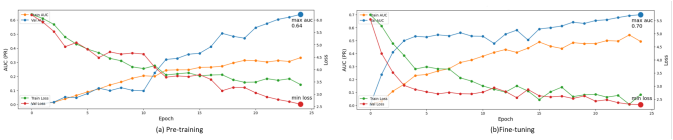

Fig. 11. Mobilenet-CNN (a) Pre-training (b) Fine-tuning

2) VGG16 RCNN: The VGG-RCNN model utilized ReLU activation in all its CNN layers. The RoI feature maps use convolutional layers with varying kernel sizes. The first layer used a (1, 1) kernel and ReLU activation, while subsequent layers had (3, 3) kernels, ReLU activation, and spatial padding. After convolution, global max-pooling is done to pool the features and fed into two fully connected layers with 1024 units and ReLU activation. The final dense output layer has 264 outputs, using softmax activation for classification. The highest validation AUC achieved during pre-training was 0.77, while the most notable validation accuracy reached during fine-tuning was 0.87. In conclusion, the pre-training phase demonstrated a solid performance with an AUC of 0.77. However, the fine-tuning process further enhanced the model's performance as shown in Fig. 12.

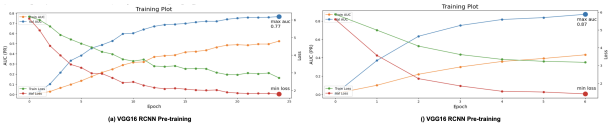

Fig. 12. VGG16 RCNN Learning curve (a) Pre-Training (b) Fine-Tuning

3) ENet CNN: The EfficientNet-CNN model uses the ReLU activation function across all of its convolutional layers. A dropout rate of 0.5 was applied, and the ADAM optimizer was utilized. Furthermore, the number of filters was incrementally increased following each max-pooling operation. Initially, various modifications were explored by the team, such as incorporating batch normalization and dropout after each CNN layer, as well as substituting ReLU with leaky ReLU at different stages. However, our research findings revealed that the implementation of a straightforward model yielded superior precision when compared to the utilization of complex models for this specific dataset. Fig. 13 represents a comparison learning curve for pretraining and finetuning. It shows that the validation AUC (0.69) surpassed the training AUC (0.31). However, it is noteworthy that the training AUC in both pretraining and finetuning remained below 0.50. These results indicate that the model did not exhibit robust generalization due to the limited amount of data used for validation.

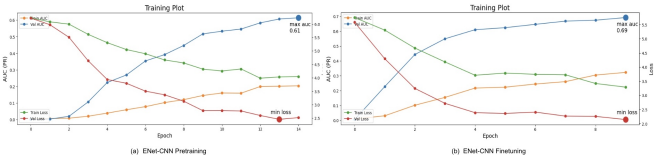

Fig. 13. EfficientNet CNN Learning curve (a) Pre-Training (b) Fine-Tuning

4) ENet VIT: In the experiments from ENet VIT, the model was evaluated under various configurations of dropout rates, activation functions, and optimizers. The final model employed the GeLU activation function for the MLP layer, while the remaining layers performed optimally with their default ReLU activation. Layer normalization was applied using an epsilon value of 1e-6, and a dropout rate of 0.2 was identified as best. The Adam optimizer was utilized for training the network. Two models were developed, one with two attention heads and another with eight attention heads. The model with eight heads, demonstrated superior accuracy compared to the two-head model. Fig. 14 shows the learning curve for the eight-head model. The best pre-training validation AUC was 0.82, and the best fine-tuning validation accuracy achieved was 0.74. While the network exhibited slight underfitting during pre-training, this model represented the best fit among all other models in this study, as evidenced by the AUC and loss values for both training and validation, indicating neither overfitting nor underfitting. Table. I shows the comparison of all four models in this research.

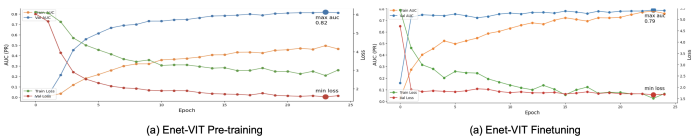

Fig. 14. EfficientNet Visual Transformer Learning curve (a) Pre-Training (b) Fine-Tuning

VI. Discussion and Future Improvements

As shown in Table. I, the ENet-CNN and MobileNet-CNN models have similar validation accuracies of 0.65 and

TABLE I
MODEL EVALUATION RESULTS

	Fine-tuning Accuracy	Testing Accuracy
ENet-CNN	***0.65***	***0.86***
MobileNet-CNN	***0.66***	***0.85***
VGG16-RCNN	***0.81***	***0.78***
ENet-VIT	***0.74***	***0.97***

0.66, respectively. However, the MobileNet-CNN model has a slightly higher testing accuracy of 0.85 compared to the ENet-CNN model's 0.86. The VGG16-RCNN model has a higher validation accuracy of 0.81 and a slightly lower testing accuracy of 0.78. It performs better than both the ENet-CNN and MobileNet-CNN models in terms of validation accuracy but slightly worse in terms of testing accuracy. The ENet-VIT model stands out with a higher validation accuracy of 0.74 and an impressive testing accuracy of 0.97. It outperforms all the other models in both validation and testing accuracy, indicating its superior performance. The ENet-VIT model, which combines EfficientNet-CNN with Vision Transformer, demonstrates the best performance among the mentioned models, achieving significantly higher testing accuracy. It highlights the effectiveness of leveraging both architectures for image classification tasks and showcases the potential of hybrid models to improve accuracy.

It is evident that most of the models are generalizing very well with the test data. However, it has to be noted that the high test accuracy might be an indication that the test data is too easy to predict for the model since the training data provided by the Kaggle BirdCLEF 2023 competition is split to train, test and validate. Using scraped data from XenoCanto that is not part of the BirdCLEF competition might give a better idea of how the model generalizes.

As a future scope, the team will collect more diverse and real-world data to eliminate bias, optimize the models to use fewer resources, and improve the training time. Also, using unsupervised machine learning with clustering would prove to be more beneficial for the research

VII. REFERENCES

REFERENCES

[1] Awsaf, "BirdCLEF23: Pretraining is All you Need [Train]," Kaggle, Apr. 2023, [Online]. Available: https://www.kaggle.com/code/awsaf49/birdclef23-pretraining-is-all-you-need-train

[2] BirdCLEF 2021 - Birdcall Identification — Kaggle. (n.d.). https://www.kaggle.com/competitions/birdclef-2021

[3] BirdCLEF 2022 - Birdcall Identification — Kaggle. (n.d.). https://www.kaggle.com/competitions/birdclef-2022

[4] BirdCLEF 2023 - Birdcall Identification — Kaggle. (n.d.). https://www.kaggle.com/competitions/birdclef-2023

[5] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1704.04861

[6] K. Elissa, "Title of paper if known," unpublished.

[7] Park, K., & Lee, J. S. (2022). SwinE-Net: hybrid deep learning approach to novel polyp segmentation using convolutional neural network and Swin Transformer. Journal of Computational Design and Engineering, 9(2), 616–632. https://doi.org/10.1093/jcde/qwac018

[8] Persson, S. (2018). Application of the German Traffic Sign Recognition Benchmark on the VGG16 network using transfer learning and bottleneck features in Keras. DIVA. https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1188243&dswid=1875

[9] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[10] Shrestha, R., Glackin, C., Wall, J., & Cannings, N. (2021). Bird Audio Diarization with Faster R-CNN. In Lecture Notes in Computer Science (pp. 415–426). Springer Science+Business Media.

[11] Y. Qu, X. Li, Z. Qin, and Q. Lu, "Acoustic scene classification based on three-dimensional multi-channel feature-correlated deep learning networks," Scientific Reports, vol. 12, no. 1, Aug. 2022, doi: 10.1038/s41598-022-17863-z.