APPLICATION OF DATA AUGMENTATION METHODS IN TRANSFER LEARNING ALGORITHMS TO IDENTIFY AMPHIBIAN SPECIES IN BIOACOUSTIC SIGNALS

1st Adriana Lucía Melo Ordóñez
Departamento de Ingeniería Electrónica
Pontificia Universidad Javeriana Cali
Cali, Colombia
adrimelo98@javerianacali.edu.co

***Abstract*—The sensitivity of amphibians, particularly anurans, to temperature changes makes them vital indicators of the impacts of global warming. Passive Acoustic Monitoring (PAM) has been employed to track these species, but the analysis of extensive acoustic data is time-consuming and demanding. Machine learning offers a solution for automating this process, yet it requires substantial data to obtain reliable results. To address data scarcity and improve model performance, this study explores the use of data augmentation and transfer learning. It evaluates the effectiveness of these techniques in classifying multi-label spectrogram samples of anuran calls using three CNN architectures: ResNet, VGG, and EfficientNet. The experiments found that EfficientNet, combined with transfer learning and augmentations, achieved the highest performance with an average F1-score of 0.83.**

***Index Terms*—machine learning, transfer learning, data augmentation, multi-label classification, deep learning, bioacustics, anuran classification.**

I. Introduction

According to the Red List of Threatened Species[1], amphibians have a high percentage of species threatened by extinction. These specimens are susceptible to changes in temperatures. Thus, these changes can affect their body temperature[2] and sexual differentiation[3] and also determine the environmental conditions during the breeding season[4]. Therefore, it is crucial to monitor these species continuously to keep track of their numbers and implement strict measures promptly. Passive Acoustic Monitoring (PAM) is a valuable technique designed to collect wildlife data without causing disruption, as Browning et al. [5] exposed it. This method provides insights into various aspects of species, including their behavior, population, and habitat diversity. However, the data gathered from applying PAM could imply several thousand, even hundreds of thousands of samples, which entails considerable time and effort and can be prone to human error. Hence, machine learning emerges as a valuable tool for efficiently and rapidly analyzing large amounts of data. Nevertheless, achieving exceptional results in training these algorithms requires a substantial quantity of meaningful samples, which may sometimes be challenging to obtain, complicating the training process and the potential of machine learning algorithms.

Data Augmentation and Transfer learning are two techniques proposed to address the lack of data and boost model performance. Data augmentation involves applying various signal processing procedures to introduce distortion to existing samples, thereby creating artificial samples. As Pluščec and Šnajder [6] and Feng et al. [7] demonstrate, data augmentation can enhance the model's performance in several ways, including overall improvement, increased robustness, better identification of unknown data, and addressing imbalance by including samples from underrepresented labels.

Transfer learning involves utilizing a pre-trained model for a specific task in a new but related problem. As stated in Jukes [8], this methodology is specifically developed to enhance the learning of a target task by facilitating faster learning and enhancing performance.

This document will focus on spectrogram classification, specifically multi-label classification, which involves identifying multiple labels within a sample. Some studies have tested augmentations and transfer learning in multiclass classification, demonstrating promising results and providing valuable recommendations [9][10]. On the one hand, other research has explored using Data Augmentation in spectrogram classification, implementing various techniques to modify the spectrogram or the audio signal, and even applying image augmentations [11][12][13][14][15]. On the other hand, some studies have specifically examined the application of transfer learning in classifying spectrograms representing different species calls, such as birds and lemurs[16][17][18] [19]. There are also investigations focused on anuran multi-label classification [20] [21] but did not aim to test the effectiveness of different data augmentation techniques and transfer learning in improving the performance of neural networks for multi-label classification.

This document pretends to evaluate the effectiveness of Data Augmentation and Transfer Learning in classifying multi-

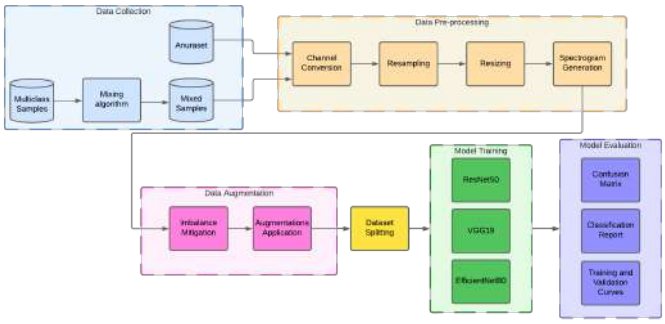

Fig. 1. Block diagram of the system

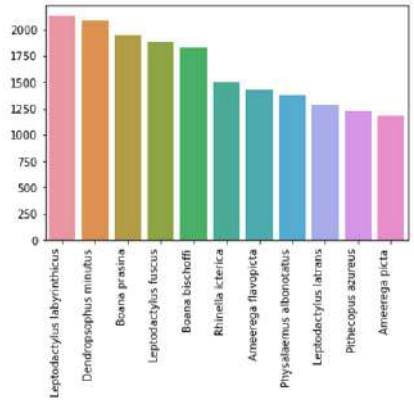

Fig. 2. Data distribution of Mixed Samples dataset

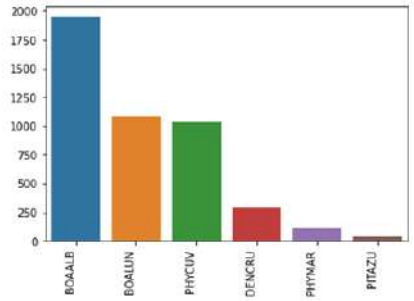

Fig. 3. Data distribution of Anuraset

label spectrogram samples containing various types of anuran calls. Three different neural networks were implemented and tested on two separate datasets, comparing specific metrics described in Section II. The results of all implementations and experiments will be presented in Section III. Finally, Section IV will provide the conclusions drawn from the investigation.

II. Materials and Methods

As suggested in Figure 1, the project followed different phases, each composed of different elements, from data collection to establishing different evaluation metrics. The following sections will describe all these phases.

A. Data Collection

Acquiring multi-label datasets for anurans can be challenging; thus, public data is limited. The number of samples for each subspecies is insufficient; even those available are typically not multi-label. However, luckily for this investigation, two approaches were implemented to tackle this issue which will be explained in the following sections.

1) "Mixed Samples" dataset: This dataset was generated from randomly combined multiclass samples from Toledo [22] and Lis [23]. It was obtained dataset composed by multi-label audios with 11 different species was obtained. The distribution of the dataset is shown in Figure 2, revealing a noticeable class imbalance, especially between "Leptodactylus Labyrinthicus" and "Ameerega Picta".

2) Anuraset: A project presented by Cañas et al. [20]. The dataset is derived from audio recordings from a collaborative PAM program in Brazil, as presented in Figure 3. For this investigation, the audios corresponding to the point "INCT41" were selected, composed of 6 different classes, and where the classes present a more pronounced imbalance compared to the "mixed samples" dataset from Section II-A1. Specifically, the "PITAZU" class has significantly fewer occurrences than the "BOAALB" class.

B. Pre-processing

Before applying the augmentations and starting the training process, the samples must be submitted through pre-processing to ensure that the signals have the same duration, sampling frequency, etc. This approach prevents errors during the subsequent phases. The procedures performed here are based on the pre-processing proposed by Doshi [24]. This process is composed of the following steps:

- Channel Conversion: The signals were adjusted to single-channel audio.
- Resampling:The resampling function from Torchaudio was implemented to guarantee that all samples have the same sample rate.
- Resizing: During this step, the audio durations were equalized.
- Spectrogram generation: The mel-spectrogram function from Torchaudio was utilized, and applying the decibel scale improved amplitude visibility in the spectrogram's colors. Both datasets used a window size of 1024, 128 Mel filterbanks, and a range of 80 dB.

C. Data augmentation

The augmentation implemented in this investigation was divided into two phases: imbalance mitigation and augmentation application. These steps were implemented to address the imbalance in the dataset, particularly for classes with the fewest occurrences. Additionally, diverse signal modifications were applied to both the audio and the spectrogram to increase the range of distortions that could be applied to the samples.

1) Imbalance Mitigation: As mentioned, one aim is to address data imbalance with data augmentation. This will

be done by increasing the number of samples with fewer occurrences to increase their detectability by the model. The increment will be made by considering the level of imbalance. First, the proportion of the minority class was calculated based on the number of samples. Specifically, four categories of imbalance were defined: No Imbalance (over 50% minority class), Mildly Imbalanced (40% to 49%), Highly Imbalanced (20% to 40%), and Extremely Imbalanced (less than 20%).

In multi-label datasets, special consideration was given to the overlap between underrepresented and overrepresented classes within individual samples to avoid negatively impacting the detection of overrepresented classes. The algorithm employed for augmentation calculated two vectors: augmentation factors and extra augmentation factors, based on the imbalance levels. The number of augmentations applied to each sample was determined by whether all classes within the sample were below the mean of the dataset. If this condition was met, the maximum values from both vectors were summed; otherwise, the minimum value from the augmentation factors vector was combined with the maximum value of the extra augmentation factors vector, divided by the number of elements within the vector. Additionally, the algorithm imposed a limit on the number of augmentations to prevent overfitting, ensuring that no class received more than 100 augmentations.

2) Augmentation Application: The augmentation techniques were selected according to several of the sources consulted: Nanni et al. [9], Lucas Ferreira-Paiva et al. [15], Park et al. [11], Sun et al. [10], Salamon et al. [14]. The works of Nanni et al. [9] and Lucas Ferreira-Paiva et al. [15] implemented various augmentation techniques, not only focused on spectrogram augmentation, as seen in but also included several audio and image-based techniques. A total of 25 initial augmentation techniques were implemented, divided into the following categories:

- Audio Augmentations: Echo Effect, Dynamic Range Compression, Time Stretch, Shuffling and Mixing, Clipping, “Wow” resampling, Rolling, Shuffling, Noise Injection, Delay, Pitch Shifting, Sound Mix, Harmonic Distortion, Time Axis Flip.device, the sampling rates may differ. The resampling function from Torchaudio was implemented to guarantee that all samples have the same sample rate.
- Spectrogram Augmentations: VTLP, EMDA, Time Masking, Time Swapping.
- Image-Based Augmentations: Negative-Positive Filter, Color Reduction Filters (32 and 18), Color Filters (Red, Green), Brightness and Saturation filters.

To address cases where certain classes required more than 25 augmentations to correct the imbalance, additional ”mixed” augmentation techniques were proposed. These techniques combined some of the previously mentioned augmentations, including Echo with Sound Mix, Rolling with Dynamic Range Compression, Shuffling and Mixing with Noise Injection, Time Masking with VTLP, and Time Swapping with Rolling and ”Wow” Resampling.

The selection and order of augmentation techniques were based on specific criteria, including their ability to simulate plausible real-world data contexts, the extent of frequency axis modification, their effectiveness in improving data visualization, and their impact on classification performance. In scenarios requiring more than 25 augmentations, techniques were applied with consideration for random factors to ensure uniqueness and prevent model overfitting.

D. Model selection and architecture

Three models and their architectures were selected based on their results from previous investigations and through experimentation. While the architecture may differ among models, certain elements remain consistent. All layers were frozen during the application of transfer learning, utilizing the weights obtained from training the models on ”ImageNet.” Additionally, specific hyperparameters, such as a learning rate set at 0.00001, were standardized across all models. The output function was defined as a sigmoid activation function, enabling the calculation of independent probabilities for each label within the label vector. The choice of Adam as the optimizer was based on its rapid convergence and reliable training process, which are advantageous for dealing with imbalanced datasets. Furthermore, the metrics ”binary cross-entropy” and ”binary accuracy” were applied uniformly across all models. Similar to the sigmoid activation, these metrics determine their values by comparing the probabilities of each label in the label vector. Finally, the subsequent sections will provide detailed information on the architecture established for each neural network.

1) Resnet: This architecture is broadly known for its capacity to classify images and has shown excellent results in some experiments of spectrogram classification[20][16][17][25][26]. However, its high complexity could limit its performance in some datasets. This model also requires high memory capacity and computational requirements.

2) VGG: A model developed for image classification tasks with deep neural networks. It can achieve higher accuracy metrics and good generalization with a simpler architecture than other models. Similar works used it to test transfer learning and data augmentation, showing promising results[9][10].

3) EfficientNetB0.: EfficientNet employs a technique that enhances the model’s performance by uniformly scaling three dimensions (width, depth, and resolution) by computing a predetermined set of coefficients. This model presents a simpler architecture while achieving good results, even better than the one obtained with Resnet[25][26].

E. Experimental setup

In Figure 1, the experimental pipeline starts with data collection from ”Anuraset” and a mixed-sample dataset. After data collection, preprocessing is done which includes normalization and generating spectrograms. The next phase involves offline data augmentation, which involves applying augmentations before training. This approach allows for the assessment of

augmentation quality and the use of an algorithm to address dataset imbalances or enhance underrepresented classes. The final dataset is then prepared for training using Keras's "ImageDataGenerator," which efficiently generates batches of images with a specified batch size of 32.

Training begins by dividing the dataset into training and validation sets using the Stratified KFold Cross-Validation algorithm, which ensures label combinations are evenly distributed across five folds. This process allows for a comprehensive evaluation of the model's generalization ability across different partitions. The model is configured to train for 50 epochs, with hyperparameters adjusted as needed based on the behavior of the training and validation curves. Additional adjustments may include incorporating regularization techniques or modifying the model's architecture by adding layers, such as fully connected layers, pooling layers, or batch normalization layers, to enhance performance.

Callbacks are implemented to track training time, monitor GPU memory usage, and save model weights based on validation loss. Four experiments are proposed to evaluate the effectiveness of data augmentation and transfer learning: (1) testing the model with both techniques applied, (2) testing with only transfer learning, (3) testing with only data augmentation, and (4) testing without either technique. The results will be compared using the evaluation metrics detailed in Section III to assess the individual and combined effects of data augmentation and transfer learning on model performance

F. Evaluation metrics.

One of the metrics used to define model performance was the behavior of the training and validation curves. These curves help detect overfitting or underfitting and confirm that the models have appropriately learned from the data.

The classification report and confusion matrices were implemented to measure the model's performance in terms of how good its predictions are. The classification matrix provides insights into the number of samples correctly and incorrectly classified. These are used to calculate the metrics presented in the classification report, which includes the next range of metrics:

- Accuracy: The proportion of correct predictions relative to the total number of evaluated samples.
- Precision: The accuracy of positive predictions for each class.
- Recall: The model's capacity to detect positive samples accurately for each class.
- F1-score: The harmonic mean between precision and recall, providing a balanced evaluation of the model's performance.
- Support: The number of samples in each class.
- Micro Average: Calculated by considering all True Negatives, True Positives, False Positives, and False Negatives across all classes.
- Macro Average: The average of each metric (Precision, Recall, and F1-score) across all classes.

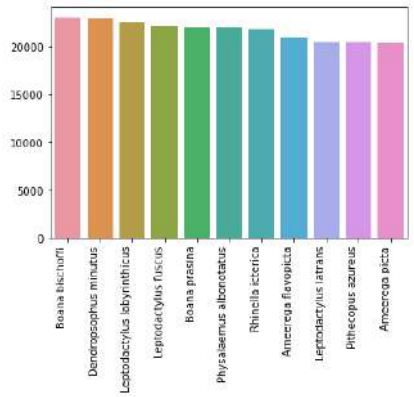

Fig. 4. Data distribution of Mixed Samples dataset after augmentation

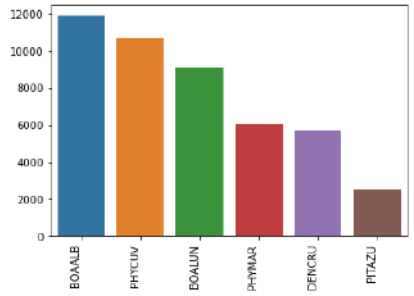

Fig. 5. Data distribution of Anuraset after augmentation

- Weighted Average: The average of each metric weighted by the support of each class.
- Samples Average: The average of Precision, Recall, and F1-score for each instance.

The initial three metrics were computed for each class within the dataset, with subsequent metrics derived from their averages. This process provides a deeper understanding of the model's performance across individual classes, allowing a focus on classes with lower metrics and an investigation into the underlying reasons for these values. It also facilitates an assessment of the overall model performance. Other metrics, such as training time, GPU memory usage, and model convergence, were implemented to explore additional characteristics of the selected models, as mentioned in Section II-E. Cross-validation was also used to evaluate model performance by monitoring the model's behavior across the five training folds, ensuring no indications of overfitting and independence of the training results from the specific data used for training and validation. These metrics allow for a comparison of classification metrics, training time, memory usage, and convergence information, which can be beneficial in potential applications.

III. Results

A. Data augmentation

Figure 4 and Figure 5 display the results of the data distribution after applying augmentation to the Mixed samples and Anuraset, respectively. The imbalance has improved in

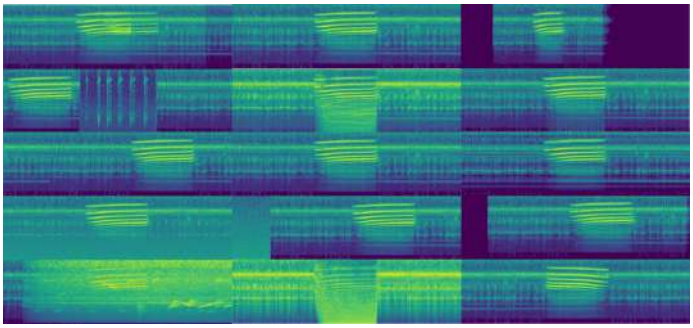

Fig. 6. Audio Augmentation techniques. Augmentation order(from left to right): Echo Effect, Dynamic Range Compression, Time Stretch, Shuffling and Mixing, Clipping, Wow Resampling, Rolling, Shuffling, VTLP, White Noise Injection, Delay, Pitch Shift, Sound Mix, Harmonic Distortion, and Flip.

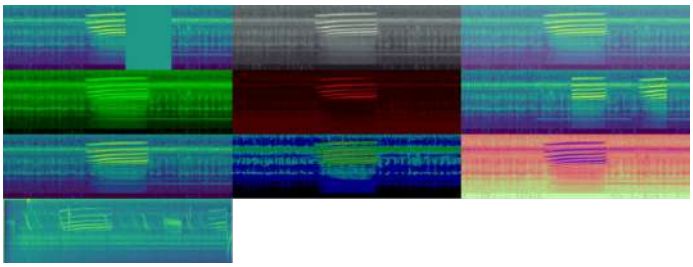

Fig. 7. Spectrogram augmentation techniques. Augmentation order(from left to right): Time Masking, Saturation, Brightness, Green Filter, Red Filter, Time Swapping, Color Reduction 32, Color Reduction 128, Positive Negative and EMDA.

both datasets, although the extent of improvement varies. With the mixed samples, the increase in balance almost reaches the level of the classes with more occurrences compared to Figure 2. It is important to note that the difference between classes was not as significant as the differences observed in Anuraset. The mixed samples dataset also had more individual samples or samples with classes below the mean.

In the case of Anuraset, there was also an improvement, mainly observed in classes such as "PITAZU" and "PHYMAR," where it is clear that the proportion of their samples has increased. However, the dataset still exhibits significant imbalance. As suggested in Section II-C1, these numbers were caused by the number of shared samples between the classes with more occurrences and those with fewer occurrences, limiting the number of augmentations that could be applied. The impacts of these imbalances and the use of data augmentation will be observed in the experiments in the following sections.

In Figure 6 and Figure 7, it is possible to examine the distinctive effects that may influence the augmentations in a sample. These distortions range from alterations in the time axis to modifications in the signal's energy by adjusting the amplitude of various elements within the dataset. Additionally, changes in the visual aspects of the spectrogram, such as its color, can occur. The alterations in the signals are intended to introduce complex situations, thereby augmenting the model's capability to classify samples in challenging environments

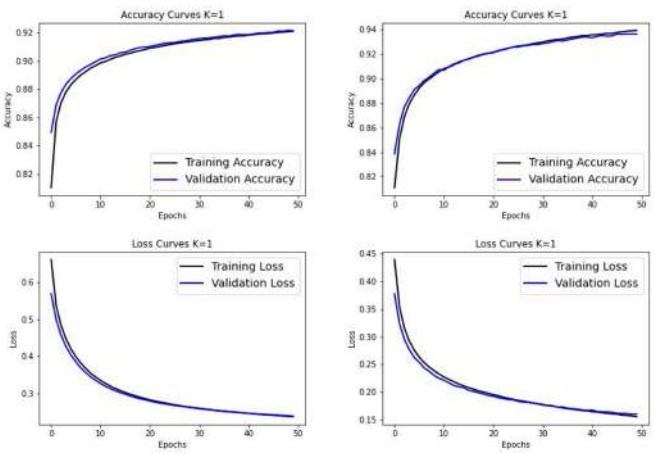

Fig. 8. EfficientNet learning curves with transfer learning and augmentation. Left: Mixed samples dataset. Right: Anuraset.

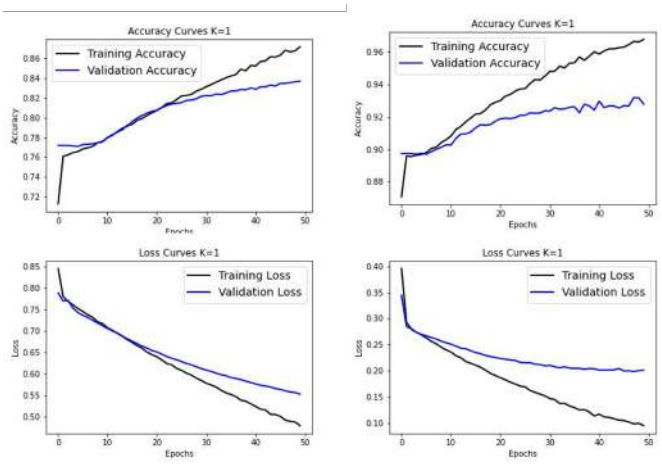

Fig. 9. EfficientNet learning curves with no transfer learning and without augmentations. Left: Mixed samples dataset. Right: Anuraset.

and recognize unknown data. Yet it is essential to note that, when applying augmentations, it is crucial to strike a balance ensuring that the distortions are neither too subtle nor too drastic, as either extreme could diminish the effectiveness of the model.

B. Learning Curves

Figure 8 depicts an example of the learning curves obtained from training EfficientNet with Anuraset and Mixed Samples. The rest of the models presented a similar behavior. These curves display the characteristics of an ideal learning curve, where the accuracy values gradually increase and then stabilize at a specific level, indicating that the model has effectively learned from the data. Furthermore, the minimal distance between the training and validation curves suggests the absence of overfitting, with the values from the validation curve aligning closely with those of the training set. When one or both techniques were absent, the learning curves started to behave erratically, as illustrated in Figure 9. The validation

and training curves began to diverge, with the validation curve fluctuating throughout the epochs. This suggests overfitting during the training, indicating that the validation values do not accurately reflect whether the model has learned from the training set. Furthermore, the curves did not stabilize at a certain point, implying that the model would require more epochs to learn from the dataset thoroughly.

C. Classification Metrics

Table I depicts the classification report calculated from the values of the confusion matrices. Both metrics suggest that EfficientNet performed the best out of the three models. Thus, it achieved high values in metrics, achieving f1-scores that are near or more than 0.7 for Anuraset and fith values near 0.8 for the mixed samples dataset. Therefore, EfficientNet surpasses other models in terms of metrics and remains resilient to class imbalance and data complexity. This can be attributed to integrating a scale-up algorithm that adapts the model's complexity to the data, enabling it to identify and differentiate essential features effectively [25][26].

Meanwhile, the metrics for VGG19 and ResNet50 were lower than those for EfficientNet, especially for Anuraset. These architectures encountered difficulties in accurately identifying certain classes, particularly classes with limited samples, such as "DENCRU" or "PHYMAR." Among these models, ResNet50 was the most sensitive to class imbalance, displaying the lowest numbers for these classes. Furthermore, VGG19 exhibited slightly poorer performance than ResNet50 and EfficientNet for Mixed Samples, indicating potential challenges with datasets containing more complex information or a high number of labels.

There were also classes that presented a recall value that were below than the rest of the classes and that did not belong to classes with few ocurrences, which their low value was attributed to the state of the data. There some samples which their spectral characteristics were not visible due to the background noise or the overlapping between classes, difficulting their recognition. This classes presented also metrics below the rest in the rest of the experiments.

Another important piece of information is the discrepancy between the binary accuracy values from the training and validation sets, the averaged accuracy, and the values encountered in the classification report. Thus, the values from the classification report do not reach values more than 0.9 or 0.8, as suggested in the accuracy values. This discrepancy arises because binary accuracy is calculated by averaging elementwise correct predictions. Therefore, even if the model makes correct predictions for the majority of samples, the overall accuracy might still appear high. Meanwhile, when calculating average accuracy using the confusion matrix, the accuracy value can be disproportionately high if the number of True Negatives is large, even if the model fails to correctly predict many False Positives.

In contrast, Table II illustrates the metrics obtained in the absence of certain techniques. This led to a reduction in most metrics, with classes like "PITAZU" (5) showing a

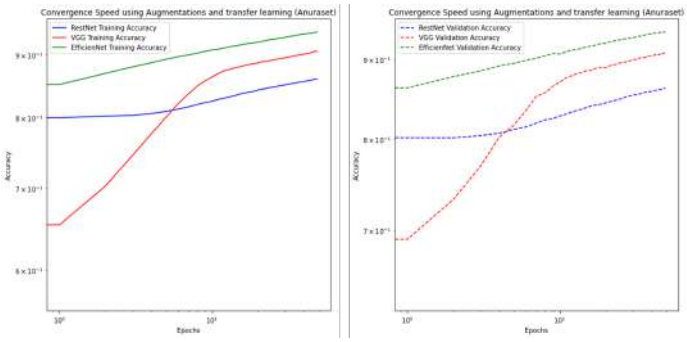

Fig. 10. Convergence speed of models trained with mixed samples dataset.

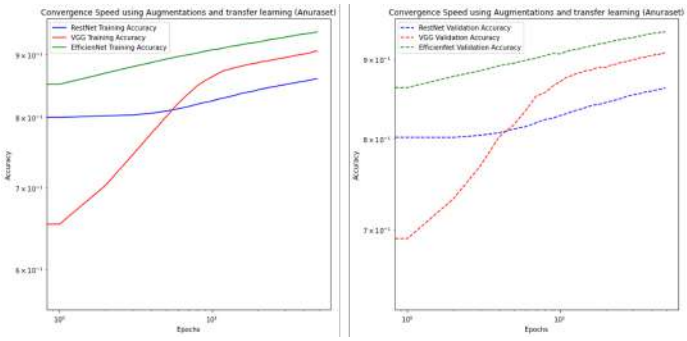

Fig. 11. Convergence speed of models trained with Anuraset.

value of 0, indicating that the models struggled to identify these classes due to their limited presence. This decrease also impacted the metrics in Anuraset. Consequently, the absence of augmentations had a more pronounced impact on datasets with significant class imbalances compared to those with less pronounced imbalances. In Table III, an exception occurred when transfer learning was not used, but the dataset was augmented. In this scenario, the metrics exceeded 0.8, even reaching values of 0.9, indicating precision in the model. However, it's important to consider the possibility of overfitting as a potential cause for these high values. Further investigation is necessary to determine the underlying cause.

D. Training Time and Memory Usage

The results presented in Table IV and Table V illustrate the training time and GPU usage for two experiments. It was observed that there was no consistent pattern between the datasets in terms of which model displayed a shorter training time or used less GPU. EfficientNet, ResNet, and VGG each better performance in each one of the experiments, even in each one of the datasets. However, one consistent finding was that training time increased when augmentations were used, which can be logically attributed to the increase in the number of training sample.

E. Convergence

In considering convergence, Figure 10 and Figure 11illustrates that, with data augmentation and transfer learning, the models consistently and swiftly achieved convergence.

TABLE I
EVALUATION METRICS COMPARISON.(AUGMENTATION AND TRANSFER LEARNING)

Using Augmentations and Transfer Learning																		
Model	ResNet50						VGG19						EfficientNetB0					
Dataset	Anuraset			Mixed samples			Anuraset			Mixed samples			Anuraset			Mixed samples		
Metric	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
0	0,763 ± 0,014	0,43 ± 0,016	0,55 ± 0,011	0,817 ± 0,016	0,498 ± 0,01	0,619 ± 0,005	0,855 ± 0,023	0,72 ± 0,025	0,781 ± 0,009	0,798 ± 0,027	0,456 ± 0,019	0,58 ± 0,011	0,909 ± 0,026	0,81 ± 0,033	0,856 ± 0,008	0,89 ± 0,016	0,605 ± 0,022	0,72 ± 0,012
1	0,657 ± 0,063	0,122 ± 0,012	0,205 ± 0,018	0,805 ± 0,01	0,572 ± 0,011	0,669 ± 0,008	0,738 ± 0,025	0,473 ± 0,04	0,575 ± 0,024	0,754 ± 0,011	0,466 ± 0,022	0,575 ± 0,013	0,855 ± 0,018	0,648 ± 0,033	0,736 ± 0,016	0,849 ± 0,013	0,687 ± 0,014	0,759 ± 0,006
2	0,779 ± 0,006	0,512 ± 0,004	0,618 ± 0,004	0,809 ± 0,008	0,342 ± 0,009	0,481 ± 0,008	0,83 ± 0,026	0,679 ± 0,04	0,746 ± 0,016	0,737 ± 0,007	0,271 ± 0,009	0,396 ± 0,009	0,859 ± 0,021	0,794 ± 0,031	0,824 ± 0,007	0,853 ± 0,026	0,434 ± 0,018	0,575 ± 0,011
3	0,772 ± 0,008	0,61 ± 0,007	0,682 ± 0,005	0,967 ± 0,003	0,814 ± 0,005	0,884 ± 0,003	0,834 ± 0,027	0,742 ± 0,033	0,784 ± 0,008	0,919 ± 0,005	0,775 ± 0,004	0,841 ± 0,002	0,866 ± 0,019	0,832 ± 0,016	0,848 ± 0,004	0,977 ± 0,003	0,876 ± 0,006	0,923 ± 0,003
4	0,846 ± 0,012	0,437 ± 0,021	0,576 ± 0,021	0,863 ± 0,01	0,583 ± 0,013	0,696 ± 0,009	0,865 ± 0,011	0,696 ± 0,009	0,772 ± 0,007	0,778 ± 0,018	0,518 ± 0,037	0,621 ± 0,022	0,922 ± 0,027	0,834 ± 0,021	0,875 ± 0,005	0,898 ± 0,01	0,704 ± 0,015	0,789 ± 0,007
5	0,77 ± 0,082	0,073 ± 0,008	0,133 ± 0,015	0,827 ± 0,014	0,58 ± 0,009	0,682 ± 0,007	0,828 ± 0,039	0,557 ± 0,023	0,666 ± 0,022	0,801 ± 0,013	0,543 ± 0,021	0,647 ± 0,011	0,909 ± 0,015	0,749 ± 0,044	0,821 ± 0,023	0,879 ± 0,015	0,649 ± 0,022	0,747 ± 0,01
6				0,948 ± 0,004	0,8 ± 0,004	0,868 ± 0,002				0,934 ± 0,015	0,789 ± 0,022	0,855 ± 0,007				0,961 ± 0,007	0,863 ± 0,009	0,909 ± 0,005
7				0,879 ± 0,009	0,606 ± 0,008	0,717 ± 0,004				0,824 ± 0,011	0,534 ± 0,013	0,648 ± 0,007				0,899 ± 0,025	0,733 ± 0,033	0,807 ± 0,01
8				0,931 ± 0,004	0,78 ± 0,005	0,849 ± 0,002				0,874 ± 0,02	0,739 ± 0,025	0,8 ± 0,008				0,964 ± 0,006	0,887 ± 0,007	0,924 ± 0,002
9				0,805 ± 0,014	0,367 ± 0,01	0,504 ± 0,008				0,716 ± 0,055	0,271 ± 0,031	0,391 ± 0,024				0,819 ± 0,031	0,49 ± 0,03	0,612 ± 0,016
10				0,92 ± 0,004	0,77 ± 0,003	0,838 ± 0,003				0,897 ± 0,02	0,705 ± 0,02	0,789 ± 0,006				0,955 ± 0,007	0,888 ± 0,004	0,92 ± 0,002
Micro Avg	0,776 ± 0,005	0,436 ± 0,005	0,558 ± 0,005	0,879 ± 0,003	0,609 ± 0,002	0,72 ± 0,001	0,831 ± 0,011	0,675 ± 0,016	0,745 ± 0,006	0,835 ± 0,003	0,551 ± 0,004	0,664 ± 0,002	0,882 ± 0,005	0,792 ± 0,007	0,835 ± 0,002	0,91 ± 0,004	0,709 ± 0,005	0,797 ± 0,002
Macro Avg	0,765 ± 0,021	0,364 ± 0,005	0,46 ± 0,006	0,87 ± 0,003	0,61 ± 0,002	0,71 ± 0,001	0,825 ± 0,009	0,645 ± 0,014	0,721 ± 0,008	0,821 ± 0,002	0,552 ± 0,004	0,649 ± 0,002	0,887 ± 0,002	0,778 ± 0,007	0,827 ± 0,003	0,904 ± 0,005	0,71 ± 0,005	0,79 ± 0,002
Weighted Avg	0,767 ± 0,012	0,436 ± 0,005	0,535 ± 0,004	0,869 ± 0,003	0,609 ± 0,002	0,709 ± 0,001	0,83 ± 0,01	0,675 ± 0,016	0,742 ± 0,006	0,82 ± 0,003	0,551 ± 0,004	0,649 ± 0,002	0,883 ± 0,004	0,792 ± 0,007	0,833 ± 0,003	0,904 ± 0,005	0,709 ± 0,005	0,789 ± 0,002
Samples Avg	0,39 ± 0,006	0,361 ± 0,007	0,363 ± 0,007	0,874 ± 0,002	0,667 ± 0,002	0,727 ± 0,001	0,602 ± 0,01	0,59 ± 0,014	0,584 ± 0,012	0,818 ± 0,002	0,608 ± 0,003	0,665 ± 0,002	0,707 ± 0,005	0,702 ± 0,006	0,694 ± 0,005	0,922 ± 0,003	0,761 ± 0,004	0,808 ± 0,002
	-2*Average Accuracy		-2*0,862	-2*Average Accuracy		-2*0,897	-2*Average Accuracy		-2*0,907	-2*Average Accuracy		-2*0,878	-2*Average Accuracy		-2*0,937	-2*Average Accuracy		-2*0,921
	-2*Total Average Accuracy			-2*0,879			-2*Total Average Accuracy			-2*0,893			-2*Total Average Accuracy			-2*0,929		

TABLE II
EVALUATION METRICS COMPARISON.(NO AUGMENTATION AND NO TRANSFER LEARNING)

No Augmentations and No Transfer Learning																		
Model	ResNet50						VGG19						EfficientNetB0					
Dataset	Anuraset			Mixed samples			Anuraset			Mixed samples			Anuraset			Mixed samples		
Metric	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
0	0,793 ± 0,092	0,653 ± 0,113	0,705 ± 0,036	0,61 ± 0,086	0,583 ± 0,098	0,585 ± 0,017	0,806 ± 0,369	0,876 ± 0,087	0,773 ± 0,29	0,68 ± 0,26	0,824 ± 0,137	0,701 ± 0,114	0,675 ± 0,028	0,53 ± 0,017	0,594 ± 0,014	0,609 ± 0,033	0,327 ± 0,032	0,425 ± 0,031
1	0,496 ± 0,201	0,334 ± 0,216	0,32 ± 0,092	0,698 ± 0,08	0,687 ± 0,088	0,683 ± 0,021	0,518 ± 0,361	0,366 ± 0,145	0,557 ± 0,28	0,812 ± 0,263	0,737 ± 0,216	0,714 ± 0,12	0,1 ± 0,224	0,003 ± 0,008	0,007 ± 0,015	0,637 ± 0,013	0,485 ± 0,048	0,55 ± 0,035
2	0,75 ± 0,131	0,766 ± 0,077	0,747 ± 0,054	0,58 ± 0,103	0,446 ± 0,061	0,495 ± 0,013	0,915 ± 0,034	0,882 ± 0,04	0,897 ± 0,007	0,626 ± 0,227	0,611 ± 0,2	0,561 ± 0,078	0,695 ± 0,039	0,598 ± 0,037	0,642 ± 0,03	0,615 ± 0,075	0,304 ± 0,038	0,406 ± 0,047
3	0,813 ± 0,051	0,683 ± 0,101	0,736 ± 0,046	0,875 ± 0,049	0,813 ± 0,035	0,842 ± 0,008	0,872 ± 0,117	0,852 ± 0,081	0,852 ± 0,034	0,703 ± 0,3	0,934 ± 0,085	0,759 ± 0,198	0,731 ± 0,05	0,65 ± 0,022	0,687 ± 0,013	0,827 ± 0,026	0,744 ± 0,032	0,783 ± 0,025
4	0,383 ± 0,286	0,061 ± 0,039	0,103 ± 0,065	0,778 ± 0,109	0,688 ± 0,107	0,718 ± 0,028	0,396 ± 0,237	0,691 ± 0,173	0,445 ± 0,159	0,728 ± 0,275	0,831 ± 0,17	0,724 ± 0,115	0 ± 0	0 ± 0	0 ± 0	0,685 ± 0,032	0,454 ± 0,053	0,544 ± 0,038
5	0,2 ± 0,447	0,025 ± 0,056	0,044 ± 0,099	0,793 ± 0,056	0,596 ± 0,078	0,675 ± 0,036	0,676 ± 0,205	0,5 ± 0,25	0,516 ± 0,196	0,693 ± 0,257	0,855 ± 0,103	0,727 ± 0,158	0 ± 0	0 ± 0	0 ± 0	0,636 ± 0,019	0,213 ± 0,02	0,319 ± 0,024
6				0,904 ± 0,091	0,686 ± 0,112	0,77 ± 0,059				0,821 ± 0,24	0,872 ± 0,073	0,822 ± 0,115				0,909 ± 0,017	0,649 ± 0,026	0,757 ± 0,022
7				0,851 ± 0,061	0,68 ± 0,079	0,751 ± 0,037				0,964 ± 0,049	0,744 ± 0,118	0,833 ± 0,059				0,72 ± 0,018	0,496 ± 0,033	0,587 ± 0,024
8				0,947 ± 0,028	0,781 ± 0,056	0,855 ± 0,026				0,865 ± 0,177	0,956 ± 0,025	0,898 ± 0,107				0,827 ± 0,032	0,644 ± 0,038	0,723 ± 0,02
9				0,507 ± 0,081	0,286 ± 0,064	0,356 ± 0,025				0,272 ± 0,146	0,908 ± 0,161	0,387 ± 0,112				0,426 ± 0,056	0,051 ± 0,012	0,091 ± 0,021
10				0,906 ± 0,045	0,745 ± 0,077	0,814 ± 0,037				0,74 ± 0,318	0,969 ± 0,026	0,795 ± 0,271				0,787 ± 0,035	0,704 ± 0,038	0,743 ± 0,029
Micro Avg	0,743 ± 0,054	0,652 ± 0,063	0,691 ± 0,02	0,744 ± 0,027	0,637 ± 0,02	0,686 ± 0,005	0,707 ± 0,255	0,858 ± 0,044	0,749 ± 0,195	0,551 ± 0,068	0,828 ± 0,027	0,659 ± 0,046	0,706 ± 0,027	0,546 ± 0,015	0,615 ± 0,006	0,725 ± 0,009	0,457 ± 0,005	0,561 ± 0,005
Macro Avg	0,572 ± 0,128	0,42 ± 0,036	0,442 ± 0,015	0,768 ± 0,013	0,636 ± 0,018	0,686 ± 0,008	0,697 ± 0,153	0,778 ± 0,054	0,673 ± 0,115	0,719 ± 0,066	0,84 ± 0,028	0,72 ± 0,04	0,367 ± 0,034	0,297 ± 0,009	0,322 ± 0,005	0,698 ± 0,007	0,461 ± 0,005	0,539 ± 0,004
Weighted Avg	0,756 ± 0,052	0,652 ± 0,063	0,682 ± 0,022	0,756 ± 0,016	0,637 ± 0,02	0,682 ± 0,007	0,83 ± 0,1	0,858 ± 0,044	0,812 ± 0,088	0,721 ± 0,062	0,828 ± 0,027	0,717 ± 0,035	0,643 ± 0,02	0,546 ± 0,015	0,587 ± 0,006	0,693 ± 0,007	0,457 ± 0,005	0,537 ± 0,004
Samples Avg	0,251 ± 0,013	0,247 ± 0,022	0,242 ± 0,017	0,787 ± 0,022	0,698 ± 0,018	0,706 ± 0,005	0,307 ± 0,054	0,341 ± 0,019	0,315 ± 0,038	0,561 ± 0,08	0,861 ± 0,025	0,654 ± 0,055	0,194 ± 0,007	0,197 ± 0,007	0,189 ± 0,006	0,727 ± 0,007	0,519 ± 0,009	0,568 ± 0,009
	-2*Average Accuracy		-2*0,940	-2*Average Accuracy		-2*0,836	-2*Average Accuracy		-2*0,920	-2*Average Accuracy		-2*0,800	-2*Average Accuracy		-2*0,930	-2*Average Accuracy		-2*0,834
	-2*Total Average Accuracy			-2*0,888			-2*Total Average Accuracy			-2*0,860			-2*Total Average Accuracy			-2*0,882		

TABLE III
EVALUATION METRICS COMPARISON.(AUGMENTATION AND NO TRANSFER LEARNING)

Augmentations and No Transfer Learning																		
Model	ResNet50						VGG19						EfficientNetB0					
Dataset	Anuraset			Mixed samples			Anuraset			Mixed samples			Anuraset			Mixed samples		
Metric	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
0	0,94 ± 0,012	0,897 ± 0,023	0,918 ± 0,008	0,938 ± 0,03	0,831 ± 0,051	0,88 ± 0,019	0,92 ± 0,069	0,929 ± 0,042	0,922 ± 0,022	0,96 ± 0,024	0,913 ± 0,05	0,935 ± 0,019	0,848 ± 0,011	0,856 ± 0,007	0,852 ± 0,005	0,905 ± 0,013	0,806 ± 0,014	0,852 ± 0,003
1	0,887 ± 0,03	0,889 ± 0,02	0,888 ± 0,016	0,944 ± 0,03	0,898 ± 0,05	0,919 ± 0,015	0,987 ± 0,011	0,756 ± 0,123	0,851 ± 0,077	0,863 ± 0,123	0,935 ± 0,074	0,889 ± 0,038	0,782 ± 0,013	0,761 ± 0,02	0,771 ± 0,006	0,933 ± 0,013	0,872 ± 0,017	0,902 ± 0,004
2	0,941 ± 0,044	0,882 ± 0,034	0,909 ± 0,009	0,809 ± 0,08	0,844 ± 0,05	0,822 ± 0,021	0,886 ± 0,087	0,946 ± 0,041	0,911 ± 0,034	0,86 ± 0,145	0,882 ± 0,051	0,861 ± 0,065	0,862 ± 0,009	0,856 ± 0,016	0,859 ± 0,007	0,841 ± 0,023	0,71 ± 0,021	0,77 ± 0,006
3	0,934 ± 0,025	0,914 ± 0,018	0,923 ± 0,004	0,965 ± 0,026	0,965 ± 0,012	0,965 ± 0,008	0,897 ± 0,064	0,961 ± 0,02	0,927 ± 0,029	0,985 ± 0,017	0,977 ± 0,01	0,981 ± 0,005	0,873 ± 0,008	0,865 ± 0,004	0,869 ± 0,004	0,984 ± 0,002	0,945 ± 0,004	0,964 ± 0,001
4	0,956 ± 0,013	0,915 ± 0,012	0,935 ± 0,005	0,959 ± 0,048	0,907 ± 0,045	0,93 ± 0,013	0,96 ± 0,024	0,901 ± 0,13	0,924 ± 0,069	0,988 ± 0,01	0,932 ± 0,039	0,958 ± 0,017	0,881 ± 0,009	0,847 ± 0,017	0,863 ± 0,006	0,944 ± 0,009	0,895 ± 0,006	0,919 ± 0,004
5	0,938 ± 0,04	0,895 ± 0,023	0,915 ± 0,012	0,95 ± 0,02	0,909 ± 0,024	0,929 ± 0,005	0,986 ± 0,01	0,893 ± 0,053	0,936 ± 0,027	0,907 ± 0,146	0,944 ± 0,039	0,918 ± 0,077	0,823 ± 0,035	0,83 ± 0,029	0,826 ± 0,03	0,923 ± 0,011	0,858 ± 0,018	0,889 ± 0,006
6				0,973 ± 0,003	0,932 ± 0,004	0,952 ± 0,003				0,986 ± 0,017	0,899 ± 0,103	0,937 ± 0,056				0,976 ± 0,003	0,913 ± 0,005	0,943 ± 0,002
7				0,98 ± 0,004	0,91 ± 0,015	0,944 ± 0,006				0,986 ± 0,018	0,912 ± 0,072	0,946 ± 0,036				0,943 ± 0,005	0,9 ± 0,011	0,921 ± 0,004
8				0,988 ± 0,004	0,958 ± 0,007	0,973 ± 0,003				0,919 ± 0,153	0,978 ± 0,012	0,941 ± 0,088				0,979 ± 0,003	0,942 ± 0,004	0,96 ± 0,003
9				0,907 ± 0,03	0,815 ± 0,03	0,858 ± 0,008				0,983 ± 0,007	0,802 ± 0,044	0,883 ± 0,024				0,891 ± 0,016	0,735 ± 0,018	0,805 ± 0,006
10				0,985 ± 0,006	0,955 ± 0,01	0,97 ± 0,003				0,873 ± 0,159	0,952 ± 0,083	0,9 ± 0,085				0,969 ± 0,003	0,944 ± 0,004	0,956 ± 0,001
Micro Avg	0,933 ± 0,016	0,9 ± 0,012	0,916 ± 0,004	0,942 ± 0,013	0,902 ± 0,012	0,921 ± 0,004	0,917 ± 0,04	0,914 ± 0,022	0,915 ± 0,021	0,92 ± 0,035	0,921 ± 0,019	0,92 ± 0,015	0,852 ± 0,003	0,844 ± 0,006	0,848 ± 0,004	0,936 ± 0,002	0,865 ± 0,001	0,899 ± 0,001
Macro Avg	0,933 ± 0,014	0,899 ± 0,01	0,915 ± 0,005	0,945 ± 0,01	0,902 ± 0,011	0,922 ± 0,004	0,939 ± 0,024	0,898 ± 0,031	0,912 ± 0,024	0,937 ± 0,027	0,921 ± 0,019	0,923 ± 0,013	0,845 ± 0,005	0,836 ± 0,007	0,84 ± 0,005	0,935 ± 0,002	0,865 ± 0,001	0,898 ± 0,001
Weighted Avg	0,934 ± 0,016	0,9 ± 0,012	0,916 ± 0,004	0,945 ± 0,011	0,902 ± 0,012	0,921 ± 0,004	0,925 ± 0,034	0,914 ± 0,022	0,913 ± 0,022	0,937 ± 0,026	0,921 ± 0,019	0,923 ± 0,013	0,852 ± 0,003	0,844 ± 0,006	0,848 ± 0,004	0,935 ± 0,002	0,865 ± 0,001	0,898 ± 0,001
Samples Avg	0,808 ± 0,006	0,804 ± 0,01	0,8 ± 0,006	0,951 ± 0,009	0,919 ± 0,01	0,922 ± 0,004	0,799 ± 0,024	0,81 ± 0,021	0,798 ± 0,022	0,932 ± 0,029	0,934 ± 0,017	0,919 ± 0,016	0,736 ± 0,004	0,751 ± 0,005	0,732 ± 0,005	0,947 ± 0,002	0,888 ± 0,001	0,901 ± 0,001
	-2*Average Accuracy		-2*0,967	-2*Average Accuracy		-2*0,966	-2*Average Accuracy		-2*0,966	-2*Average Accuracy		-2*0,965	-2*Average Accuracy		-2*0,939	-2*Average Accuracy		-2*0,958
	-2*Total Average Accuracy			-2*0,967			-2*Total Average Accuracy			-2*0,965			-2*Total Average Accuracy			-2*0,949		

TABLE IV
TRAINING TIME AND MEMORY UTILIZATION OF EACH MODEL.(AGUMENTATION AND TRANSFER LEARNING)

	ResNet50		VGG19		EfficientNetB0	
	Anuraset	Mixed Samples	Anuraset	Mixed Samples	Anuraset	Mixed Samples
Avg. Training Time	227,4728	616,1866478	226,1307	584,8391693	221,9355	603,9588
Avg. Memory utilization	0,855439	0,898107422	0,882914	0,862926758	0,858161	0,8730

TABLE V
TRAINING TIME AND MEMORY UTILIZATION OF EACH MODEL.(NO AUGMENTATION AND TRANSFER LEARNING)

	ResNet50		VGG19		EfficientNetB0	
	Anuraset	Mixed Samples	Anuraset	Mixed Samples	Anuraset	Mixed Samples
Avg. Training Time	46,8896	44,09062121	41,55936	43,61567439	44,8245	43,1138
Avg. Memory utilization	0,867244	0,850817871	0,889703	0,88885498	0,908022	0,8605

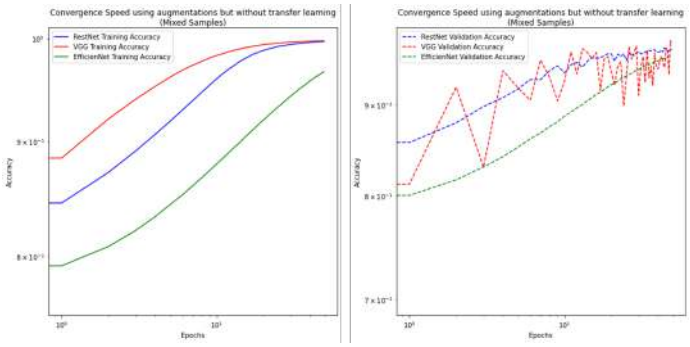

Fig. 12. Convergence speed of models trained with mixed samples dataset.

Notably, the efficientNet model demonstrated the fastest convergence. Conversely, in Figure 12 and Figure 13, the absence of these techniques resulted in slower convergence for the models, with the validation curve displaying a more noisy behavior. VGG was notably sensitive to these changes, whereas ResNet and efficientNet consistently maintained stable curves even in the absence of data augmentation and transfer learning.

IV. CONCLUSIONS AND DISCUSSION

This investigation aimed to test the effectiveness of using transfer learning and data augmentation in classifying anuran species. Among the three neural network architectures evaluated, EfficientNet benefited most from this technique. It presented the best metrics with f1-scores near or more than 0.7. Furthermore, EfficientNet could be precise independent of the type of dataset because it achieved high metrics for

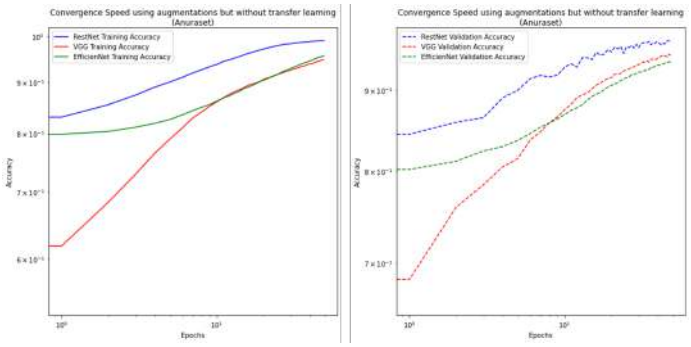

Fig. 13. Convergence speed of models trained with Anuraset.

both a dataset with a higher number of classes although less imbalanced ("Mixed samples") while also maintaining this performance in a dataset with fewer classes but with a more marked imbalance ("Anuraset"). T he results from VGG19 suggest that it is less sensible to imbalance; it performs better in datasets such as Anuraset, whereas Resnet could perform better in more complex datasets, such as datasets with more classes, though it is more sensitive to data imbalances. Focusing more on the effects of data augmentation, the results indicated that data augmentation could be more helpful for datasets with considerable marked imbalance; thus, it could improve the detectability of rare classes for some models, as was possible to see with cases such as "PHYMAR" and "PITAZU" from Anuraset. Nevertheless, its impact was less pronounced in datasets with less marked imbalance.

The study also demonstrated that data augmentation and transfer learning could improve stability, accelerate a model's convergence, and produce more consistent results during the validation folds. The outcomes regarding memory usage and training time were inconclusive, as no pattern defined which model was better than the others within these characteristics. Therefore, further research could be made comparing other developed models, such as EfficientNetV2, with a focus on memory usage. One potential research could involve comparing the effectiveness of training a model from scratch versus using transfer learning. The experiments without augmentations yielded values close to or even exceeding 0.9. Therefore, it is essential to determine whether these values resulted from a better understanding of spectral patterns or from overfitting.

Further experiments could be carried out based on architectural changes, such as testing different types of machine learning architectures or using alternative versions of the neural networks implemented in this study. Other potential changes for further investigation include testing different datasets, such as alternative subsets from Anuraset, or implementing alternative data representations instead of using spectrograms.

References

[1] "The IUCN Red List of Threatened Species," 2023. [Online]. Available: https://www.iucnredlist.org

[2] E. Sanabria and L. Quiroga, "The body temperature of active desert anurans from hyper-arid environment of South America: The reliability of WorldClim for predicted body temperatures in anurans," *Journal of Thermal Biology*, vol. 85, p. 102398, 10 2019.

[3] A. Ruiz-García, S. Roco, and M. Bullejos, "Sex Differentiation in Amphibians: Effect of Temperature and Its Influence on Sex Reversal," *Sexual Development*, vol. 15, no. 1-3, pp. 157–167, 2021.

[4] K. Ceron, D. J. Santana, E. M. Lucas, J. J. Zocche, and D. B. Provete, "Climatic variables influence the temporal dynamics of an anuran metacommunity in a nonstationary way," *Ecology and Evolution*, vol. 10, no. 11, pp. 4630–4639, 6 2020.

[5] E. Browning, R. Gibb, P. Glover-Kapfer, and K. Jones, *Passive acoustic monitoring in ecology and conservation*, 10 2017.

[6] D. Pluščec and J. Šnajder, "Data Augmentation for Neural NLP," 5 2023.

[7] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, "A survey of data augmentation approaches for NLP," *arXiv preprint arXiv:2105.03075*, 2021.

[8] E. Jukes, "Encyclopedia of Machine Learning and Data Mining (2nd edition)," *Reference Reviews*, vol. 32, no. 7/8, pp. 3–4, 9 2018.

[9] L. Nanni, G. Maguolo, and M. Paci, "Data augmentation approaches for improving animal audio classification," *Ecological Informatics*, vol. 57, 2020.

[10] Y. Sun, T. M. Maeda, C. Solis-Lemus, D. Pimentel-Alarcon, and Z. Burivalova, "Classification of animal sounds in a hyperdiverse rainforest using Convolutional Neural Networks," 11 2021.

[11] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," 4 2019.

[12] X. Song, Z. Wu, Y. Huang, D. Su, and H. Meng, "SpecSwap: A Simple Data Augmentation Method for End-to-End Speech Recognition," in *Interspeech 2020*. ISCA: ISCA, 10 2020, pp. 581–585.

[13] T. Inoue, P. Vinayavekhin, S. Wang, D. Wood, A. Munawar, B. J. Ko, N. Greco, and R. Tachibana, "Shuffling and mixing data augmentation for environmental sound classification," 2019.

[14] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello, "Scaper: A library for soundscape synthesis and augmentation," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 10 2017, pp. 344–348.

[15] Lucas Ferreira-Paiva, Elizabeth Alfaro-Espinoza, Vinicius M. Almeida, Leonardo B. Felix, and Rodolpho V. A. Neves, "A Survey of Data Augmentation for Audio Classification," 10 2022.

[16] E. Dufourq, C. Batist, R. Foquet, and I. Durbach, "Passive acoustic monitoring of animal populations with transfer learning," *Ecological Informatics*, vol. 70, p. 101688, 9 2022.

[17] K. Palanisamy, D. Singhania, and A. Yao, "Rethinking CNN Models for Audio Classification," 7 2020.

[18] J. Xie, R. Zeng, C. Xu, J. Zhang, and P. Roe, "Multi-Label Classification of Frog Species via Deep Learning," in *2017 IEEE 13th International Conference on e-Science (e-Science)*. IEEE, 10 2017, pp. 187–193.

[19] A. Khalighifar, R. M. Brown, J. Goyes Vallejos, and A. T. Peterson, "Deep learning improves acoustic biodiversity monitoring and new candidate forest frog species identification (genus Platymantis) in the Philippines," *Biodiversity and Conservation*, vol. 30, no. 3, pp. 643–657, 3 2021.

[20] J. S. Cañas, M. P. Toro-Gómez, L. S. M. Sugai, H. D. Benítez Restrepo, J. Rudas, B. Posso Bautista, L. F. Toledo, S. Dena, A. H. R. Domingos, F. L. de Souza, S. Neckel-Oliveira, A. da Rosa, V. Carvalho-Rocha, J. V. Bernardy, J. L. M. M. Sugai, C. E. dos Santos, R. P. Bastos, D. Llusia, and J. S. Ulloa, "A dataset for benchmarking Neotropical anuran calls identification in passive acoustic monitoring," *Scientific Data*, vol. 10, no. 1, p. 771, 11 2023.

[21] I. Moummad, N. Farrugia, R. Serizel, J. Froidevaux, and V. Lostanlen, "Mixture of Mixups for Multi-label Classification of Rare Anuran Sounds," *arXiv preprint arXiv:2403.09598*, 2024.

[22] L. F. Toledo, "Fonoteca Neotropical Jacques Vielliard (FNJV)," 8 2016. [Online]. Available: www.ib.unicamp.br/fnjv/

[23] "List of Call and Video Files on AmphibiaWeb." [Online]. Available: https://amphibiaweb.org/lists/sound.shtml

[24] K. Doshi, "Audio Deep Learning Made Simple (Part 3): Data Preparation and Augmentation," https://towardsdatascience.com/audio-deep-learning-made-simple-part-3-data-preparation-and-augmentation-24c6e1f6b52, 12 2021.

[25] Y. Yang, L. Zhang, M. Du, J. Bo, H. Liu, L. Ren, X. Li, and M. J. Deen, "A comparative analysis of eleven neural networks architectures for small datasets of lung images of COVID-19 patients toward improved clinical decisions," *Computers in Biology and Medicine*, vol. 139, p. 104887, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482521006818

[26] K. W. Gunawan, A. A. Hidayat, T. W. Cenggoro, and B. Pardamean, "A Transfer Learning Strategy for Owl Sound Classification by Using Image Classification Model with Audio Spectrogram," *International Journal on Electrical Engineering and Informatics*, vol. 13, no. 3, pp. 546–553, 9 2021.