Rainforest Species Identification

Ben Rapp, Phil Genovese, Jared Huzar, Craig Fox

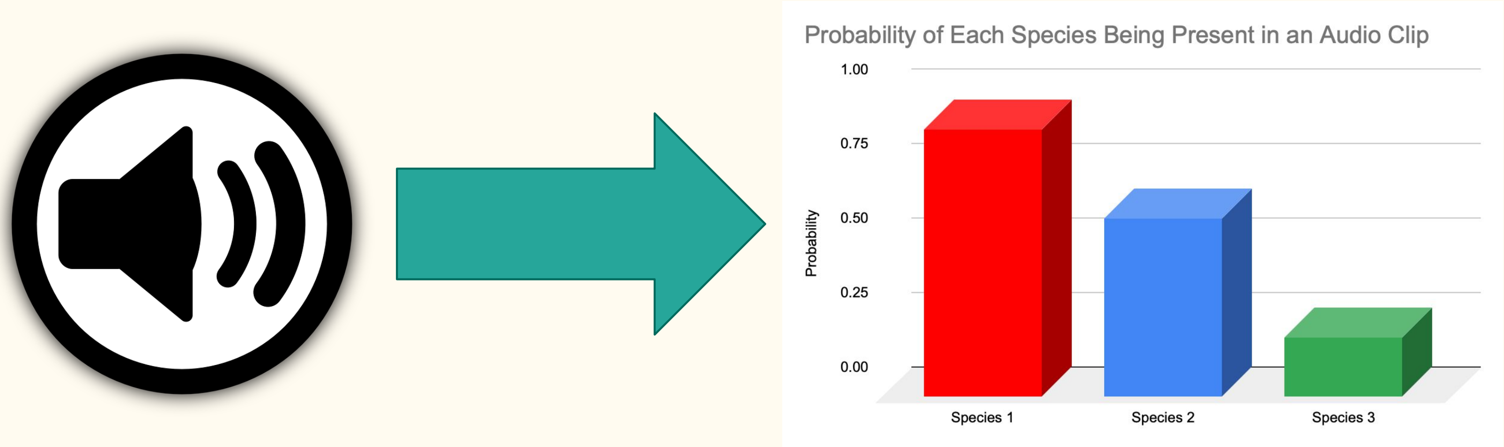
Problem

- Identifying species presence is a good proxy for rainforest health
- It is easier to collect meaningful audio data instead of video
- Expert identifiers are expensive and slow
- Basic machine learning models require huge amounts of data especially as the number of potential species increase

Puerto Rican Tody - eBird

Objective

Develop a machine learning model which can accurately predict relative probabilities of each species being present in an audio clip

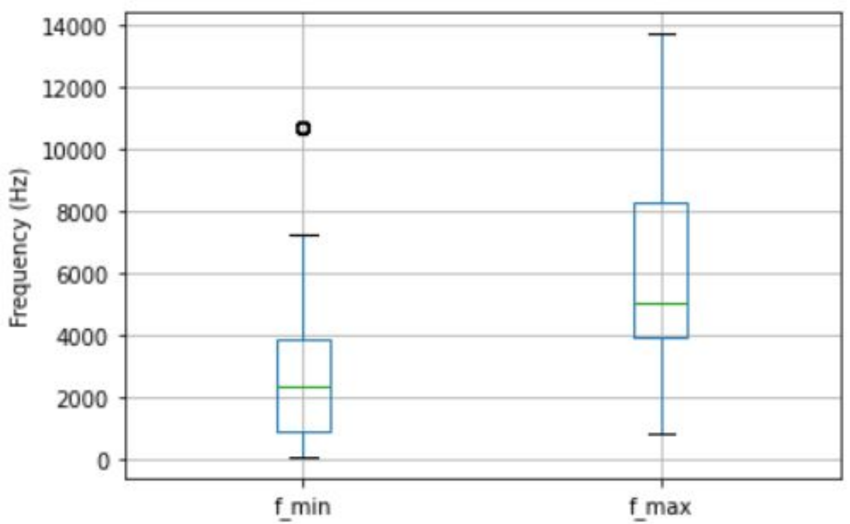


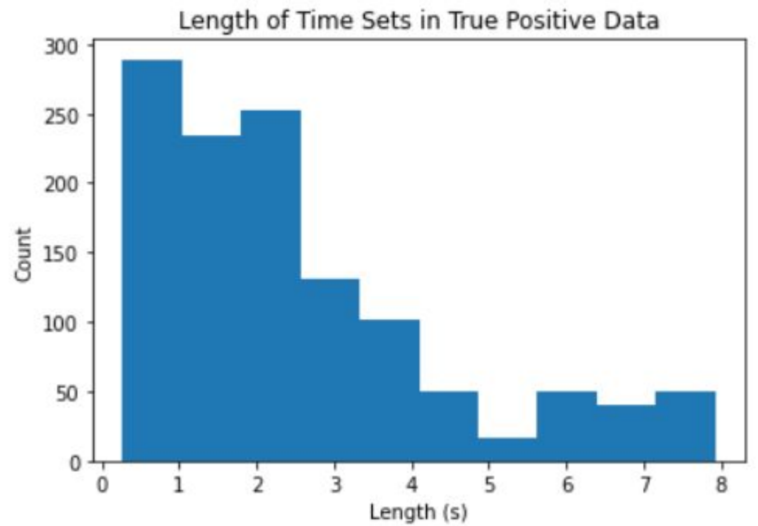
Machine Learning Hurdles

Issues		Solutions
Small Training Set • 1,216 TPs • 7,781 FPs	→	**Data Augmentation**
Noisy Data • Insects • Other Species	→	**Smoothing and Other Preprocessing**
Audio Classification Understudied • Few popular architectures	→	**Image Classification**

Data Exploration

- Even distribution of species in training set
- Wide variance in background noise
- Species are a mix of frogs and birds





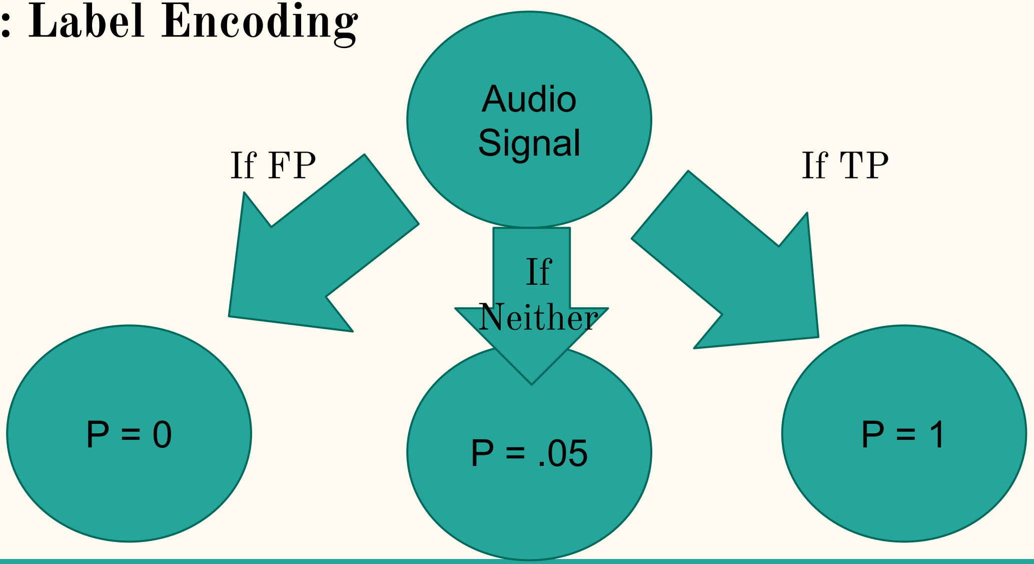
Preprocessing

- Temporal Splicing to 8 seconds surrounding a clip
- Frequency Splicing from 0 to 14000 Hz
- Nearest Neighbors Filtering
- Created “noisy versions” of the data set
    - Duplicates of the original with .2% and .5% maximum amplitude added
- Split into two 75% length clips with 50% overlap
- Converted audio into image

False Positive Data Labels

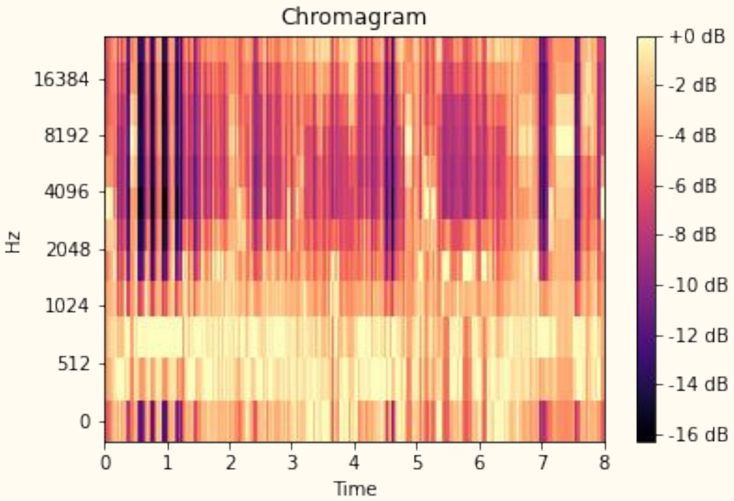
- Experts manually annotated false positives predicted by a baseline model
- Signals which are likely to be classified incorrectly by the model

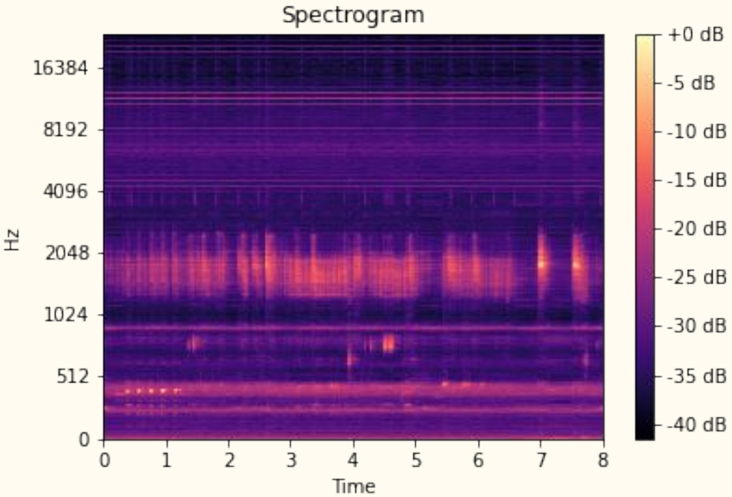
Our Solution: Label Encoding

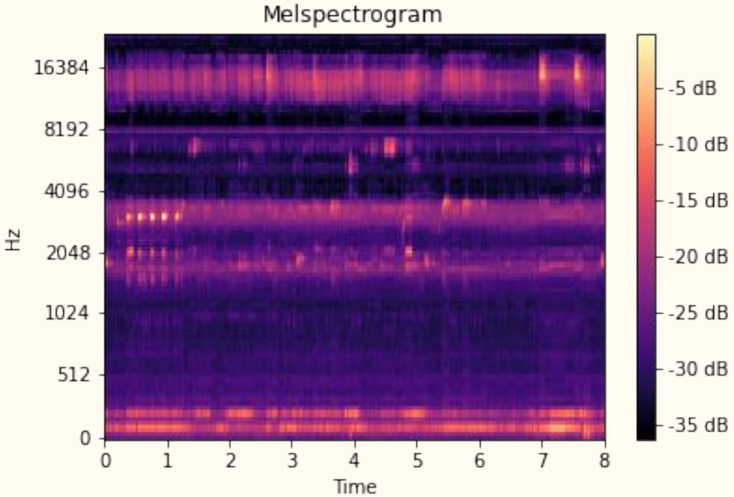


Audio to Image

- Spectrogram
- Melspectrogram
- Chromagram

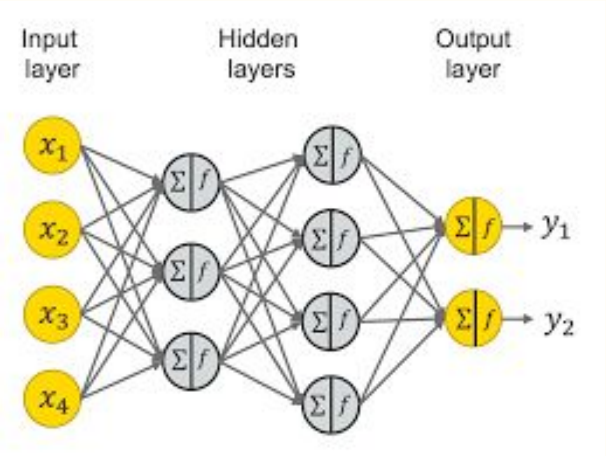






Neural Networks and Transfer Learning

Neural networks use a set of algorithms modeled after the human brain to identify patterns in data.

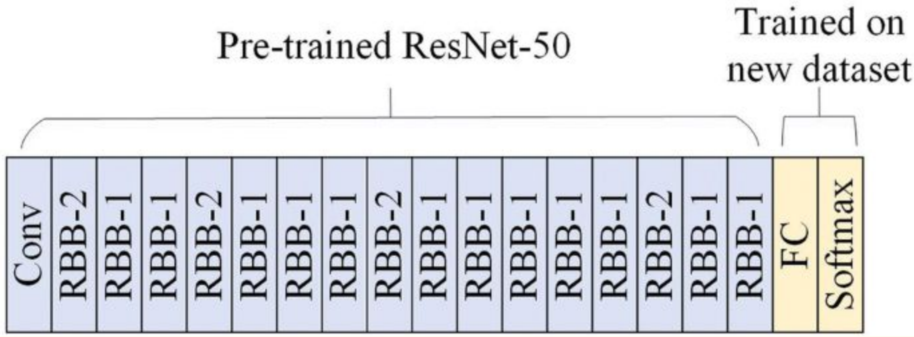


https://www.knime.com/blog/a-friendly-introduction-to-deep-neural-networks

Transfer learning uses weights from neural networks trained on other datasets

https://builtin.com/data-science/transfer-learning

Model 1 - ResNet50



50 layer deep convolutional neural network which is pretrained on millions of images.

- ResNet is the most popular image classification model

```
model = models.resnet50(pretrained=True)
model.fc = nn.Sequential(
    nn.Linear(2048, 1024),
    nn.ReLU(),
    nn.Dropout(p=0.2),
    nn.Linear(1024, 1024),
    nn.ReLU(),
    nn.Dropout(p=0.2),
    nn.Linear(1024, num_species)
)
```

Optimal Hyperparameters and Model Components:

Learning rate = .001

Optimizer = SGD

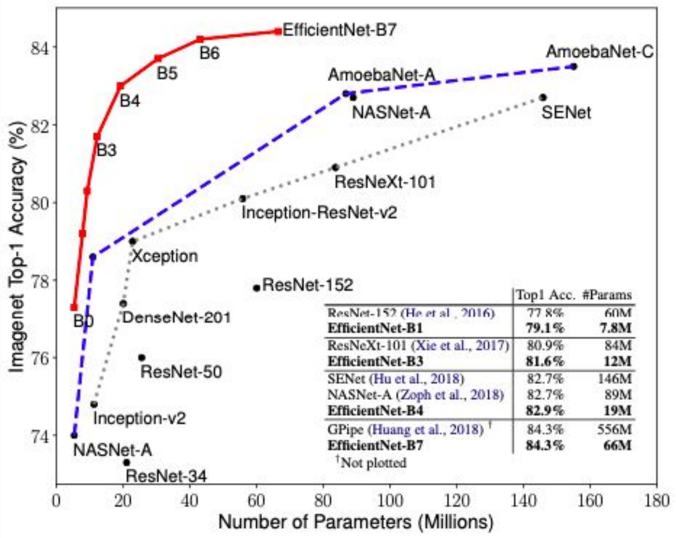
Momentum = .9

Loss = BCE w/Logits Loss

https://link.springer.com/article/10.1007/s00521-019-04097-w

Model 2 - EfficientNet7

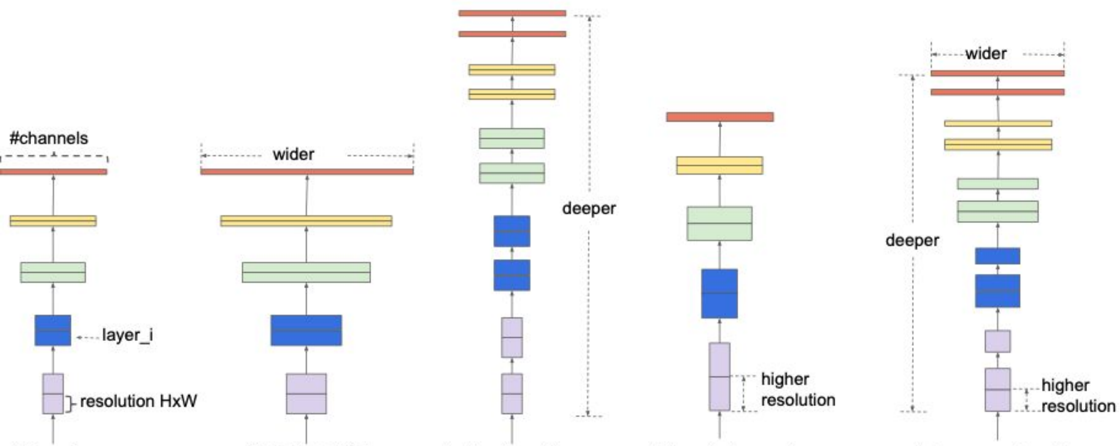
- Scales width, depth, and resolution using equal ratios



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

†Not plotted

ResNet



(a) baseline (b) width scaling (c) depth scaling (d) resolution scaling (e) compound scaling

EfficientNet

Optimal Hyperparameters and Model Components:

Learning rate = .001

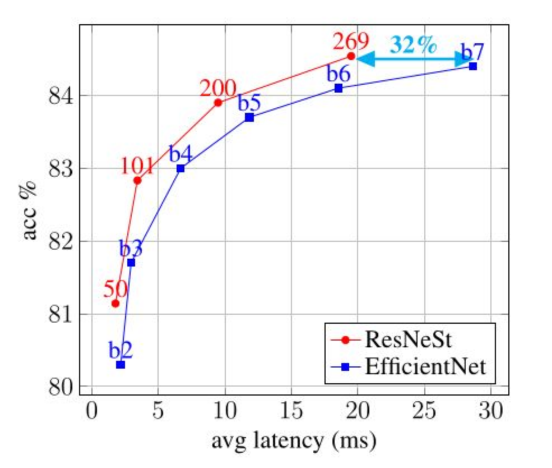
Optimizer = SGD

Momentum = .9

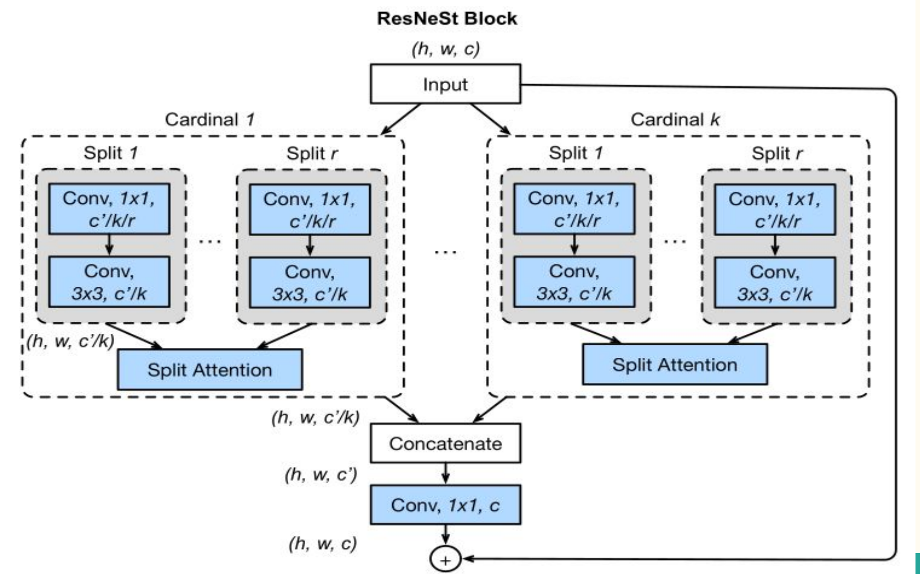
Loss = BCE w/Logits Loss

https://arxiv.org/pdf/1905.11946v5.pdf

Model 3 - ResNeSt



- Split-Attention NN designed to improve on EfficientNet and ResNet
  - “Individual salient attributes for different visual features”
  - *Correlated* vs *Independent* visual features
- Expects 3-channel RGB images, normalized using mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]



Optimal Hyperparameters and Model Components:

Learning rate = .001

Momentum = .7

Optimizer = SGD

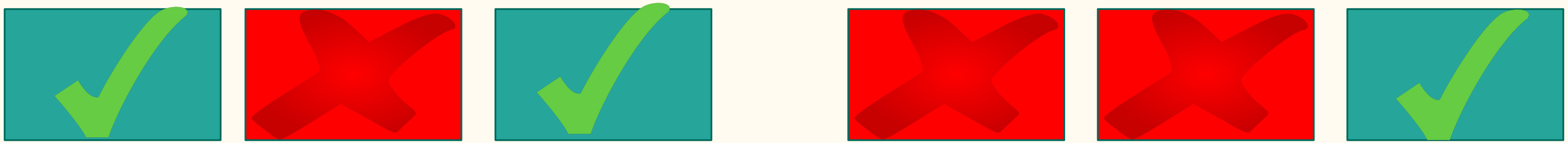
Loss = Cross Entropy

Metric

Label-weighted label-ranking average precision (LWLRAP)

- Generalization of mean reciprocal rank measure
- Each label receives equal weight
- Each test observation is weighted differently based on number of true labels

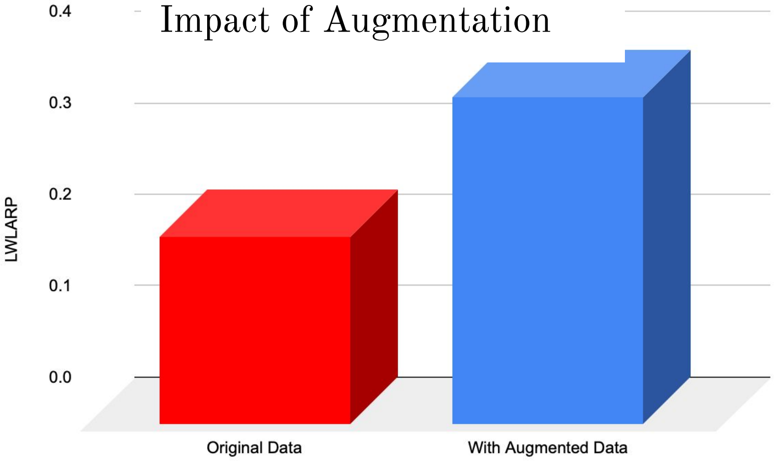
Test Prediction 1 | Test Prediction 2

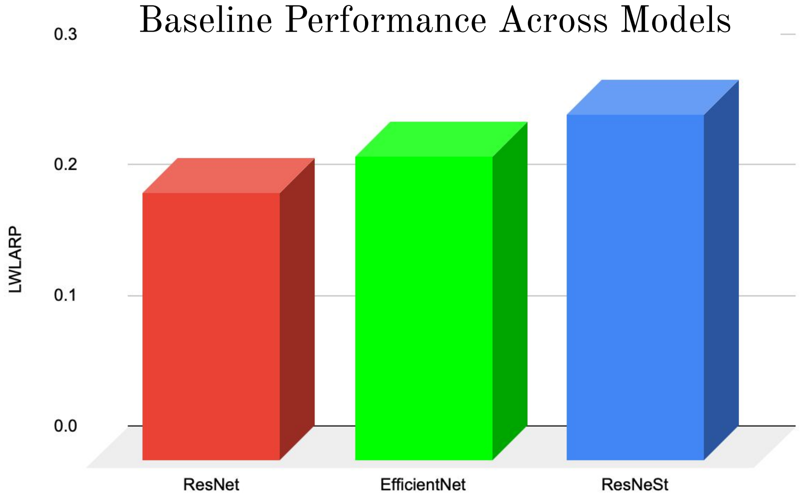


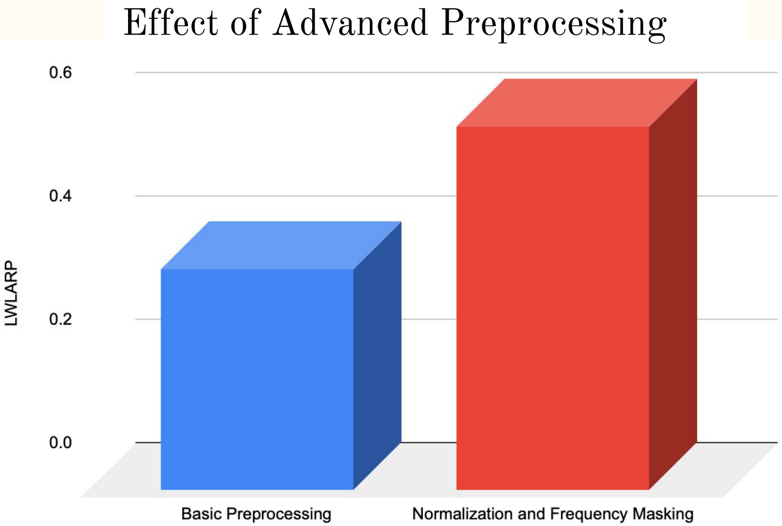
1/1 | 2/3 | 1/3

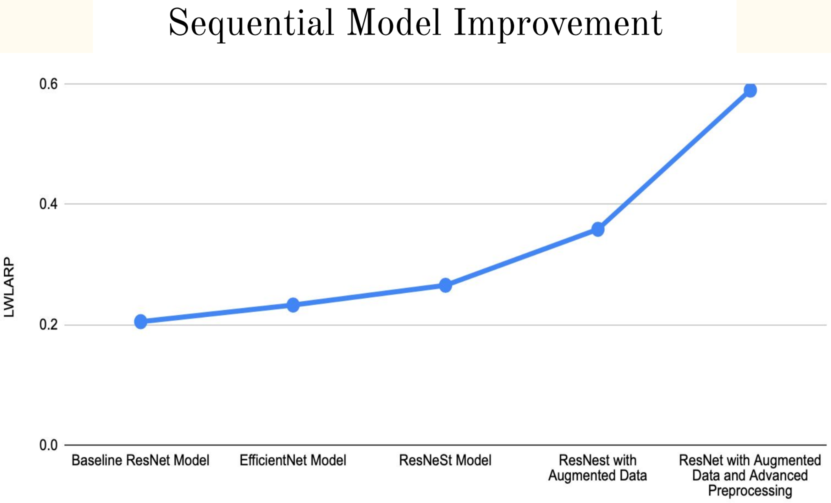
$$\text{LWLRAP} = (1 + .67 + .33)/3$$
$$= 0.67$$

Results


Impact of Augmentation
LWLARP
0.4
0.3
0.2
0.1
0.0
Original Data
With Augmented Data


Baseline Performance Across Models
LWLARP
0.3
0.2
0.1
0.0
ResNet
EfficientNet
ResNeSt


Effect of Advanced Preprocessing
LWLARP
0.6
0.4
0.2
0.0
Basic Preprocessing
Normalization and Frequency Masking


Sequential Model Improvement
LWLARP
0.6
0.4
0.2
0.0
Baseline ResNet Model
EfficientNet Model
ResNeSt Model
ResNest with Augmented Data
ResNet with Augmented Data and Advanced Preprocessing

Conclusions

- Converting audio classification into a image classification problem is effective
- Data augmentation is useful to increase training data size
- For this problem, preprocessing is more important than model fine-tuning