Audio Classification Using Multiple Feature Representations

Taha Khaleel (s1158775), Laura Euverman (s1153730)

1 Introduction

The health of ecosystems in natural reserves is commonly assessed with diversity in species populations as a metric, which changes constantly [1]. The information acquired is used to make decisions about the conservation of these ecosystems and monitor their status. Birds are often chosen as the species of interest for this assessment, using point counts. Experts will document the species they hear and see during 5 to 10 minute intervals. This is very labor intensive, and its accuracy depends on the level of expertise of the one doing the point count[2]. Artificial intelligence could present a solution that allows for faster analysis and less variability in species identified.

The Cornell Lab of Ornithology have been hosting yearly Kaggle competitions to advance in this field. In the 2025 BirdCLEF+ competition, competitors are tasked with identifying understudied species based on their acoustic signatures. The species includes avians, amphibians, mammalia, and insecta [1]. Convolutional Neural Networks (CNNs) seem to provide the best performance, having an advantage over traditional methods for classification. Typically, approaches will transform the raw audio files into one feature representation only, training the models on the features extracted by the chosen method. However, the different feature representations focus on different aspects of the audio [3]. Therefore, using multiple feature representations could possibly increase the details captured by the model, in turn boosting the performance of the classification model.

Three examples of feature representations that could be of interest are mel-spectrogram, which is most commonly used in audio classification tasks, mel frequency cepstral coefficient (MFCC), and constant Q transform (CQT). Mel-spectrograms have been shown to perform well for acoustic event recognition and are typically used for human speech processing. However, the auditory capabilities of birds are different from human speech [2]. MFCCs are computed from mel-spectrograms, and simulate the masking effect found in human hearing. Humans are more sensitive to low-frequency sounds, and more sensitive to high loudness. CQT avoids the disadvantage of uniform time-frequency resolution. CQT has a higher temporal resolution at high frequencies, which can track rapidly changing overtones [3].

The current research aims to discover whether the use of different feature representations improves the performance of commonly used pre-trained models. We extract mel-spectrogram, MFCC and CQT features from the raw data of the BirdCLEF+ 2025 challenge on Kaggle. We aim to test these on three different pre-trained models, namely RegNetY, ResNet, and EfficientNet. These three models seem to be most commonly used in classification tasks, and are also seen used by other competitors in the challenge. It is hypothesized that an ensemble of models trained on different feature presentations will yield a higher score compared to models trained on one feature representation alone. In addition to this, we expect the mel-spectrograms to be the most important feature to include. This is due to mel-spectrogram conserving the most details in its representation.

2 Method

2.1 Data and submission limitations

The current study makes use of the data provided by the BirdCLEF+ 2025 Kaggle competition. This data includes training audio files with labels, and a training file containing unlabeled audio from the same locations. Besides this, metadata for training is provided and a taxonomy file containing information on the different species. The public test set is hidden in this competition, but will consist of approximately 700 recordings. Lastly, a sample submission file is provided [1].

As is typical for Kaggle competitions, the submitted model has to meet a list of conditions, for which the ones affecting the research most will be listed. This includes that the model is not allowed to make use of the GPU for more than a minute, and is limited to 90 minutes on CPU. Besides this, internet should be disabled. Due to the hidden test set, models could only be tested through submissions. The competition limited submissions to 4 per day, including both successful and failed submissions [1].

2.2 Baseline model

For the baseline model it was decided to take a existing notebook. The notebook used a pretrained **RegNetY_008** model trained on mel-spectrogram features [4]. The accuracy we got from running this model was 0.765 on the public leaderboard.

2.3 Feature extraction methods

To extract the three feature representations, the first step was to pre-process the audio files. Firstly, the audio files were cleaned by removing human voices using the Silero VAD library [5]. Some audio segments were found to include human voices. The library identifies the time frames where human noise is detected in the audio, and replaces the detected time frames with 0, indicating that audio will be turned off for these segments.

Since every feature expects the input to be the same size, we compute target samples by multiplying the target audio duration by the sampling rate. If the denoised audio is shorter than the target samples, tiling is applied until the audio is as long as the target sample. This method was chosen instead of padding, since padding will just create silence. Tiling applies repeated audio of the animal, which gives more data for models to learn from. If the denoised audio is longer than the target samples, the center part of the audio is selected. This method was chosen because high frequencies mainly occur in the middle part of an audio segment.

After audio pre-processing steps, the three different feature representations were extracted, namely mel-spectrogram, MFCC and CQT. For this, the Librosa library was used [6]. All three feature representations are extracted via a similar method and separately stored as a dataset in a pickle file. Table 1 shows the parameters used for feature extractions. Except for CQT and MFCC parameters, which were randomly selected, we adopted all the other audio-related hyperparameters from the baseline model, because it gave us a good accuracy of around 0.7. When we applied the same parameters to our pretrained model, it boosted the accuracy to achieve around 0.8. Moreover, we did experiment with target duration to explore its influence on the performance of the model.

Parameter	**Value**
Sampling rate	32000 Hz
FFT window size	1024
Hop length	512
Number of MEL bands	128
Lowest frequency	50 Hz
Highest frequency	14000 Hz
Target duration	5 s
Output array size	256×256
Mel-spectrogram power exponent	2
MFCC coefficients	40
CQT bins	84
CQT octaves	12

Table 1: Audio feature parameters

Mel-Spectrogram. For the mel-spectrogram, first a power-scaled mel-spectrogram is computed. This is converted to log-scale, and normalized to fit decibel (dB) units.

MFCC. Extraction of the MFCC feature representations starts with the same steps as for the mel-spectrogram features. However, after normalization, the MFCC features are calculated for each time frame of the extracted mel-spectrogram features.

CQT. For extraction of the CQT features, the audio features are first converted to the number of frequency bins specified before hand. These are equally spaced by the number of octaves, from which the real power values are calculated by taking the square of the magnitude. The last step is to also normalize these features to fit dB units.

2.4 Approach

For training the models we used 5-folds stratified cross-validation on the primary labels, so the model preserve the class distribution. Due to the large size of the dataset and the associated training time, either 5 or 10 epochs were selected. For optimizer it was decided on *AdamW* and for the loss function *BCEWithLogitsLoss*,

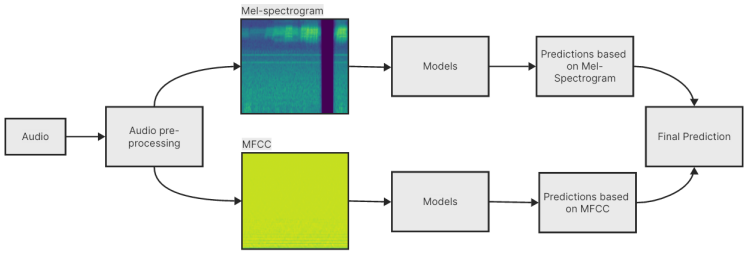

Figure 1: Pipeline

as the challenge was a multi-label classification problem. Furthermore, the learning rate for the model was optimized using *CosineAnnealingLR* as a scheduler. After each fold the best performing model was saved.

As mentioned earlier, it was decided to use pre-trained models for this research. The models are taken from the timm library [7]. The selected models are as following: **RegNetY_008 [8], EfficientNetB0 [9], ResNet50**[**10**], and **RegNet_y_040 [11]**. The **RegNet_y_040** yielded the best results on the mel-spectrogram feature representation, and thus was decided to be primarily used for increasing performance. For the ensemble, we use models trained on different feature representations, and ensemble them at the end for the final prediction. The full implementation of this pipeline can be seen in Figure 1.

2.5 Inference

The last step in the pipeline is inference. Some pre-processing steps were also implemented for the test audio, to explore whether it would improve performance. Firstly, it was experimented with removing human noise from the test audio as well. However, the performance of the model did not change when removing human noise, from which the conclusion was drawn that the test data did not include human noise. Besides this, the test data consisted of audio files with a 1 minute duration. We therefore had to ensure that the audio file was split in 5 second segments, indicated with the **window size** variable. The remainder of the pre-processing steps used in training are not implemented on the test set data. This was decided on because of the length difference between test and train audio files, which made it unnecessary for the test data.

2.6 Models

Details	Folds	Pretrained model	Target Duration	Denoised human voice (train)	Denoised human voice (infer)	Features Used	Epochs
Baseline	5	RegNetY008	5 s	No	No	Mel-spectrogram	5
Model 2	5	ResNet50	5 s	No	No	Mel-spectrogram	10
Model 3	5	EfficientNetB0	5 s	No	No	Mel-spectrogram	10
Model 4	5	RegNety040	5 s	No	No	Mel-spectrogram	10
Model 5	5	RegNety040	5 s	Yes	Yes	Mel-spectrogram	10
Model 6	5	RegNety040	10 s	Yes	No	Mel-spectrogram	10
Model 7	2	RegNety040 EfficientNetB0	5 s	No	No	Mel-spectrogram	10
Model 8	2	RegNety040	5 s	Yes	No	Mel-spectrogram MFCC	10
Model 9	5	RegNety040	5 s	Yes	No	MFCC	10

Table 2: Experimental setup for models

Initially, we used the baseline model with RegNetY008. After seeing it's performance, we wanted to experiment with different pre-trained models (e.g., ResNet50, EfficientNetB0, RegNety040). After experimenting with these, we observed that ResNet50 performed the worst despite increasing the epochs (*Model 2*). The accuracy of RegNety040 (*Model 3*) was higher than EfficientNetB0 (*Model 4*), despite having the same parameters. Still, we wanted to check their performance as ensembles, thus we selected the best performing fold from each pretrained model: EfficientNetB0 and RegNety040 (*Model 7*). Still, we did not observe a good accuracy in comparison to using all the folds from RegNety040 alone. Thus, we used RegNety040 for the next set of experiments. Next, we experimented with removing human noise for our model to learn only from the animals voices provided, and we did the same for inference (*Model 5*). However, this did not change the performance, from which we concluded that the test dataset is already without human noise. Therefore, when increasing the duration time to 10 seconds, we did not add denoised human functionality during inference (*Model 6*). Lastly, we utilized MFCC features, and did not get any good results from it (*Model 8*), but when using it with Mel-spectrogram as an ensemble (*Model 9*), there was an increase in the score. However, we could not utilize all the folds from the Mel-spectrogram and MFCC trained on RegNety040 due to time limitation.

3 Results

Below you will find a table including our results for all 9 models. We have provided the model name, the public leaderboard scores, and the private leaderboard scores. Furthermore, we provide the difference between the public

and private scores, to demonstrate which models typically improve performance. Our best performing model (*Model 5*) landed us in place 1139 on the leaderboard.

Details	**Baseline**	**Model 2**	**Model 3**	**Model 4**	**Model 5**	**Model 6**	**Model 7**	**Model 8**	**Model 9**
Public score	0.765	0.753	0.789	0.791	0.805	0.798	0.788	0.786	0.458
Private score	0.774	0.761	0.788	0.798	0.807	0.796	0.785	0.778	0.443
Difference	+ 0.009	+ 0.008	- 0.001	+ 0.007	+ 0.002	- 0.002	- 0.003	- 0.008	- 0.015

Table 3: Results

4 Discussion

With a score of 0.807 on the private leaderboard, our best performing model uses RegNetY-040 on mel-spectrogram feature representation. We used two types of ensembles, one using two model on mel-spectrogram, while the other used the one model on mel-spectrogram and MFCC. For both ensembles submission limitations limited the full use of the models. For both we could not take all 5 folds for the ensemble. Both these models performed less on the private test set compared to the public test set. We propose that different usage of both models and feature representations could possibly yield better results, for instance via knowledge distillation.

RegNetY uses the squeeze-and-excitation block, which is designed to improve representational power [12]. We expect that this aspect of the model might play a role in its good performance. ResNet50 underperformed a lot compared to the other two models, which we suggest would be due to it being an older model and thus might not be as efficient and accurate compared to the other two. The good performance of especially the RegNetY-040 model is shown when using the mel-spectrogram feature representation, however, its performance is not translated to the MFCC features. It is proposed that the details lost when transforming the features to MFCC explain the low performance more than the model. Besides this, it could be the case that the test sets did not include species with low frequency sounds and high loudness. The choice was made to not train the MFCC data on EfficientNet and ResNet due to time limitations. We do not expect its performance to be comparable to the performance on mel-spectrogram features. An effort was made to improve the performance of the RegNetY-040 model using the first and second derivative of the MFCC features. However, this effort did not yield results due to submission errors.

The current research aimed to use three different feature presentations and pre-trained models. We aimed to explore which feature representation yielded the best results. Unfortunately, due to circumstances, we were not able to include the CQT feature representation. We do propose future research to explore this feature representation combined with MFCC and mel-spectrogram in either an ensemble or knowledge distillation.

Furthermore, switching the training and inference of the models from using mel-spectrogram to MFCC took more time than expected. Due to submission constraints and not being able to test the models before submission on the public test set, the majority of models failed to submit, even though it yielded no errors when testing before submission. Combined with the submission limit of 5 submissions a day, lots of time was lost on getting a submission successful. Therefore, most of the models were trained with the mel-spectrogram feature representation only.

5 Conclusion

The current research aims to discover whether using multiple feature representations for classification could improve performance compared to using only one feature representation in audio classification. We aimed to test this on three different feature representations, namely mel-spectrogram, MFCC, and CQT. CNNs were used for training and classification. For this study we chosen RegNetY, ResNet, and EfficientNet, as these seem to perform best on the public test set. The results show that RegNety indeed outperforms both ResNet and EfficientNet when trained on the mel-spectrogram features. Especially ResNet yielded unexpectedly low results. Besides this, training MFCC on a RegNetY model yielded low results as well. We expect this to be mainly due to the lost details when using MFCC feature representation compared to mel-spectrogram. However, in an ensemble of both these models using RegNetY, the results are lower than models using solely mel-spectrogram. We expect this result to be due to either only being able to use one fold from each trained model, or ensembling to not make use of the full potential of both feature representations. Therefore, future research is expected to be able to get better results using, for instance, knowledge distillation, as this method will make better use of both feature representations.

References

[1] Holger Klinck et al. *BirdCLEF+ 2025*. https://kaggle.com/competitions/birdclef-2025. Kaggle. 2025.

[2] Stefan Kahl et al. "BirdNET: A deep learning solution for avian diversity monitoring". In: *Ecological Informatics* 61 (2021), p. 101236.

[3] Liang Gao et al. "Multi-representation knowledge distillation for audio classification". In: *Multimedia Tools and Applications* 81.4 (2022), pp. 5089–5112.

[4] Shionoa. *BirdCLEF+ 2025*. https://www.kaggle.com/code/shionao7/bird-25-submission-regnety008-v1. Kaggle Notebook, Accessed: 2025-04. 2025.

[5] Silero Team. *Silero VAD: pre-trained enterprise-grade Voice Activity Detector (VAD), Number Detector and Language Classifier*. https://github.com/snakers4/silero-vad. GitHub repository. 2024.

[6] Brian McFee et al. *librosa: Audio and music signal analysis in Python*. https://github.com/librosa/librosa. GitHub repository. 2015.

[7] PyTorch. *PyTorch Image Models*. https://github.com/rwightman/pytorch-image-models. 2019. DOI: 10.5281/zenodo.4414861.

[8] Hugging Face. *timm/regnety_008.pycls_in1k: Pretrained RegNetY-800MF Model*. https://huggingface.co/timm/regnety_008.pycls_in1k. Accessed: 2025-06-05. 2025.

[9] Hugging Face. *google/efficientnet-b0: Pretrained EfficientNet-B0 Model*. https://huggingface.co/google/efficientnet-b0. Accessed: 2025-06-05. 2025.

[10] Hugging Face. *microsoft/resnet-50: Pretrained ResNet-50 Model*. https://huggingface.co/microsoft/resnet-50. Accessed: 2025-06-05. 2025.

[11] Hugging Face. *facebook/regnet-y-040: Pretrained RegNet-Y-040 Model*. https://huggingface.co/facebook/regnet-y-040. Accessed: 2025-06-05. 2025.

[12] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-excitation networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.

6 Code Notebooks

Feature extraction notebooks for both mel-spectrogram and MFCC:
Mel-spec: https://www.kaggle.com/code/khaleel0123/transforming-audio-to-mel-spec-birdclef-25?scriptVersionId=244406082
MFCC: https://www.kaggle.com/code/lauraeuverman/transform-audio-to-mfcc-birdclef-25?scriptVersionId=244406057

Training and inference notebooks model 5:
Training: https://www.kaggle.com/code/khaleel0123/training-regnety040-comments
Inference:https://www.kaggle.com/code/lauraeuverman/fork-of-regnet-pytorch-inference-birdcl-b3d782?scriptVersionId=244410195

Training and inference notebooks ensemble features:
Training mel-spectrogram: https://www.kaggle.com/code/khaleel0123/training-regnety040-comments. (This is the same notebook as the training file for model 5 listed above)
Training MFCC:https://www.kaggle.com/code/lauraeuverman/fork-of-regnety-040-pytorch-mfcc-train-birdcle?scriptVersionId=244408853
Inference: https://www.kaggle.com/code/khaleel0123/melspec-mfcc-regnety040?scriptVersionId=244427239

Training and inference notebooks third model:
Training:https://www.kaggle.com/code/lauraeuverman/fork-of-regnety-040-pytorch-mfcc-train-birdcle?scriptVersionId=244408853
Inference:https://www.kaggle.com/code/lauraeuverman/fork-of-regnet-pytorch-inference-mfcc-birdclef?scriptVersionId=244409739

7 Generative A.I. Acknowledgement

No generative AI was used for writing this report.

8 Author Contributions

Laura was tasked with extracting the three features separately. Besides this, she was responsible for the MFCC training and inference files. She made an effort to create the mel-spectrogram/MFCC ensemble file, but Taha took over due to time constraints. Furthermore, she tried to improve the performance on the MFCC RegNetY-040 model by adding the first and second derivatives, but this did not yield an result due to submission errors. In regards to the report, she wrote the introduction, the data and submission limitations, the results, discussion and conclusion.

Taha did the majority of the coding work. She created a feature extraction file that would extract all features at once, but that unfortunately failed. Due to GPU limitations on Kaggle, Laura took the extraction over. Furthermore, she created the baseline models, and was responsible for the mel-spectrogram models. For the report, she created the tables and figures, and wrote the methods except from the data and submission limitations. Both Laura and Taha cleaned the notebooks before submitting. Besides this, both members were involved in rewriting the report before finalizing.

Yunfei wrote the script for the literature part of the presentation.

9 Evaluation of the Process

We first started with literature research, getting familiar with audio classification. Besides this, we looked into successful models from the BirdCLEF 2024 competition. When we found the idea of using multiple feature representations, we started to explore the three features of choice. We made a plan in regards to who would be responsible for which feature representation, in which it was decided Taha would be responsible for the mel-spectrogram features, as she had already delivered the most work and these would need the least amount of adapting the code. Laura had taken MFCC upon herself, and Yunfei would be responsible for CQT. As adapting the code for the different features took more time and effort than expected, we could not realize our full plan anymore. We therefore prioritized the ensemble of the features over, for example, training on other models. Unfortunately, the third member, Yunfei, did

not prioritize this project. This has caused a setback for the remaining members as well. Besides this, it meant we could not test on the CQT features at all. As there had been no unique contribution made by him a couple of days before the deadline, it was decided to exclude him from the project. This report can be confirmed to have no work contributed by him.

10 Evaluation of the Supervision

For the most part, supervision was helpful. We were made aware by the professors that implementing knowledge distillation might have been too complex for this project, and thus to rather focus on ensembles instead. Besides this, the professors made an effort to help us two weeks before the deadline to make a plan for Yunfei to make a meaningful contribution.