Animal Audio Classification

Goal

To classify animal sounds

Animal Classes

1. Dog
2. Rooster
3. Pig
4. Cow
5. Frog
6. Cat
7. Hen
8. Insects
9. Sheep
10. Crow

Dataset

Refers to the ECS dataset (https://github.com/karolpiczak/ESC-50/tree/master)

Exploratory Data Analysis (Descriptive Analysis)

(IPython File: EDA-Descriptive.ipynb)

First, we need to explore the data using descriptive analysis.

Dataset Labels

	filename	fold	target	category	esc10	src_file	take
0	1-100032-A-0.wav	1	0	dog	True	100032	A
1	1-100038-A-14.wav	1	14	chirping_birds	False	100038	A
2	1-100210-A-36.wav	1	36	vacuum_cleaner	False	100210	A
3	1-100210-B-36.wav	1	36	vacuum_cleaner	False	100210	B
4	1-101296-A-19.wav	1	19	thunderstorm	False	101296	A
5	1-101296-B-19.wav	1	19	thunderstorm	False	101296	B
6	1-101336-A-30.wav	1	30	door_wood_knock	False	101336	A
7	1-101404-A-34.wav	1	34	can_opening	False	101404	A
8	1-103298-A-9.wav	1	9	crow	False	103298	A
9	1-103995-A-30.wav	1	30	door_wood_knock	False	103995	A

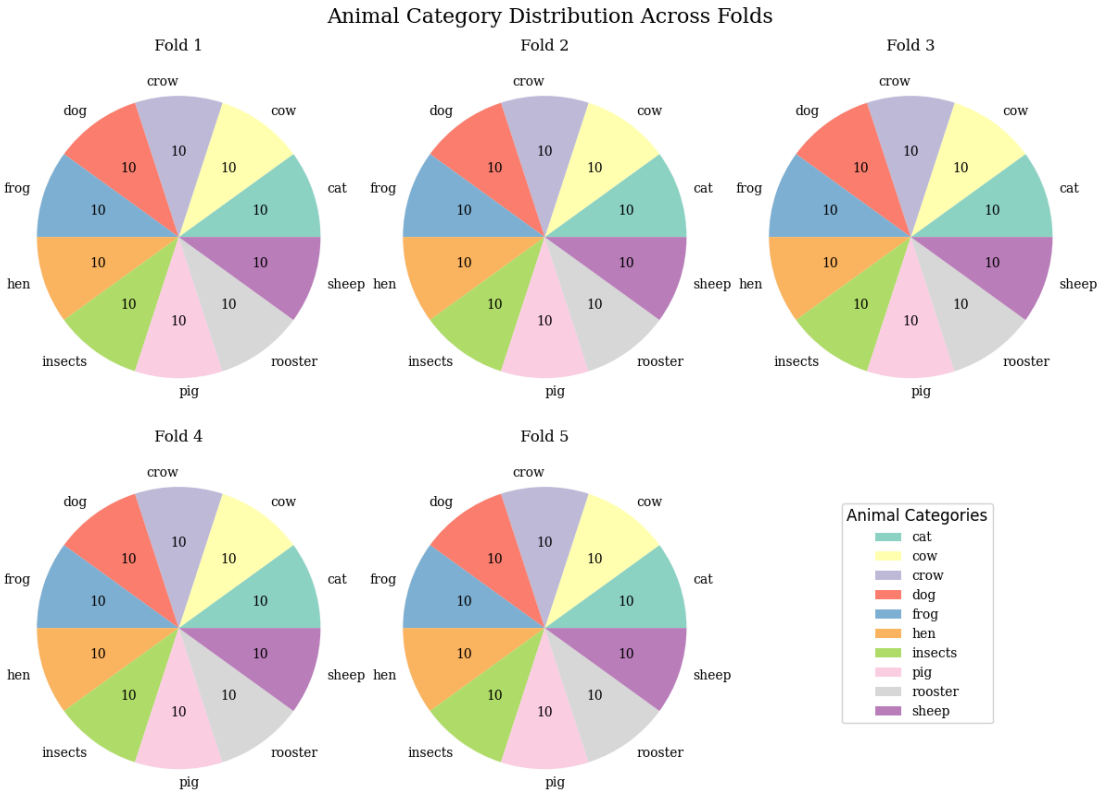
The label consists of the filename, folds, encoded target, and category or target name columns. Since we only need to train the animal category, as mentioned in the previous sections, we filtered the labels based on the **category** column.

After the filtering process, we found only 400 data points with the animal sound category, which is significantly small for machine learning modelling.

	filename	fold	target	category	esc10	src_file	take
0	1-100032-A-0.wav	1	0	dog	True	100032	A
8	1-103298-A-9.wav	1	9	crow	False	103298	A
14	1-110389-A-0.wav	1	0	dog	True	110389	A
29	1-121951-A-8.wav	1	8	sheep	False	121951	A
45	1-15689-A-4.wav	1	4	frog	False	15689	A

Data Distribution Across the Classes

We have to ensure our data distribution is balanced enough before training. To achieve this, we plot the number of animal classes for each fold.



Our animal classes already balance for each fold. Conclusion: No data imbalance handling methods required.

Split Dataset

We split the dataset into training, validation, and testing with the following proportions: 70%, 20%, and 10%. We this the data splitting before feature engineering **to avoid information leaking**.

```
Training set:   280 samples (70.0%)
Validation set:  80 samples (20.0%)
Testing set:     40 samples (10.0%)
Total:          400 samples (100.0%)

Distribution by category:
Category   Train  Val    Test   Total
----------------------------------------
cat        28     8      4      40
cow        28     8      4      40
crow       28     8      4      40
dog        28     8      4      40
frog       28     8      4      40
hen        28     8      4      40
insects    28     8      4      40
pig        28     8      4      40
rooster    28     8      4      40
sheep      28     8      4      40

Training samples: 280
Validation samples: 80
Testing samples: 40
```

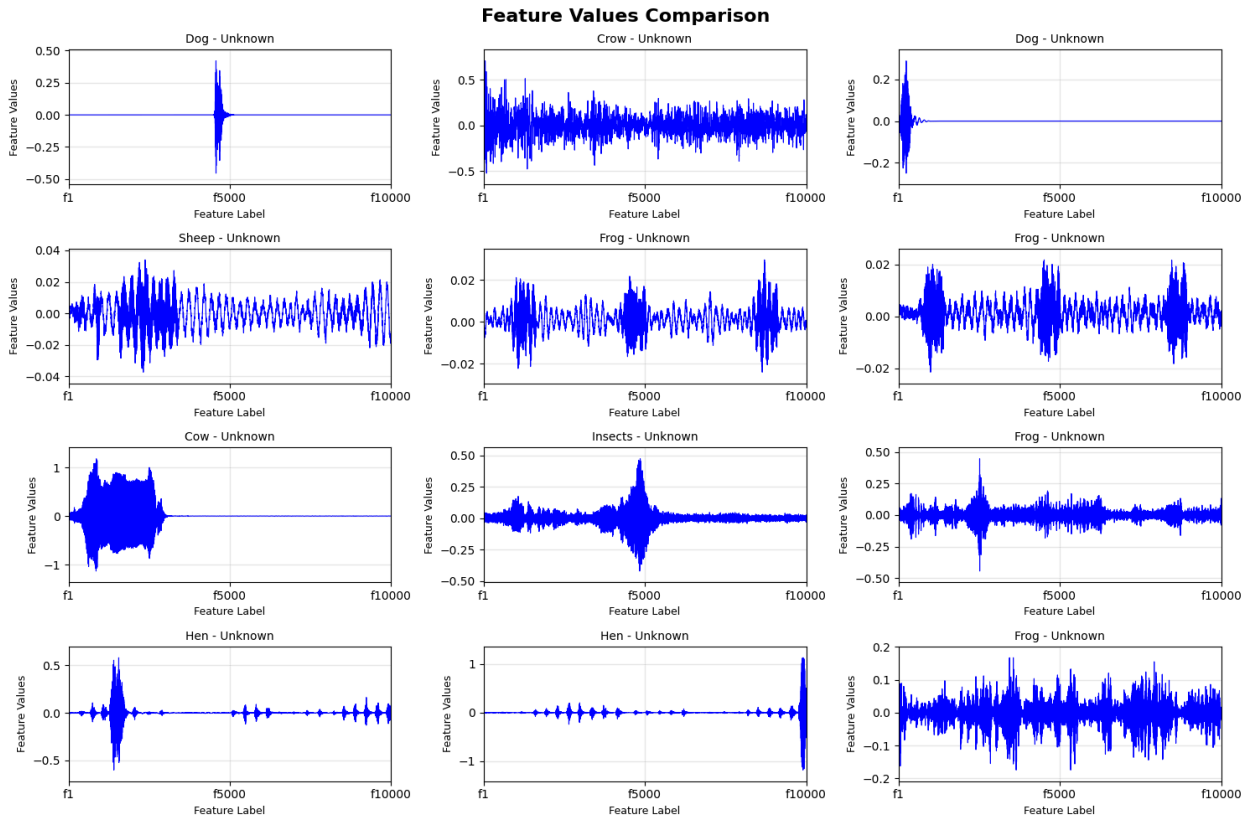
Load Audio data

We load audio data for training, validation, and testing based on the data splitting explained in the previous section. We performed audio data loading using sr=2000, consequently resulting in 10.000 audio features.

```
1 train_features.head()
```

f5	f6	f7	f8	f9	f10	...	f9994	f9995	f9996	f9997	f9998	f9999	f10000	category	target	path_file
.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	dog	0	data/ESC-50-master/audio/1-100032-A-0.wav
.115159	-0.058397	-0.028185	-0.022023	-0.024017	-0.028899	...	0.084428	0.076987	0.046018	0.023764	0.022066	-0.011075	-0.007280	crow	9	data/ESC-50-master/audio/1-103298-A-9.wav
.000630	-0.000509	-0.001278	-0.001702	-0.001951	-0.002169	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	dog	0	data/ESC-50-master/audio/1-110389-A-0.wav
.005998	-0.006433	-0.006854	-0.004552	-0.003868	-0.004532	...	-0.017308	-0.017288	-0.017264	-0.016457	-0.016510	-0.014393	-0.015810	sheep	8	data/ESC-50-master/audio/1-121951-A-8.wav
.001749	0.001854	0.002096	0.001321	0.001141	0.000947	...	-0.001159	-0.001116	-0.000818	-0.000031	-0.000539	-0.000733	-0.001524	frog	4	data/ESC-50-master/audio/1-15689-A-4.wav

Plot Audio Data Samples



Inspect Feature Ranges

In order to determine whether we need data scaling or not, we inspected the data range for each audio feature. Ideal feature values for machine learning model training are around -1 to 1 or 0 to 1.

We found that **0.05% of training feature values are outside the range**, **0.01% for validation data**, and **0.0% for test data**. Hence, **we concluded that 99% of our data lay inside the range**.

For the feature values mean is -0.00028 for the training data, -0.00018 for validation, and -0.00018 for data testing. Consequently, we said that the mean of our data is nearly 0.

Based on these two pieces of evidence, we skipped the data scaling.

Save Features.

Feature Engineering

(IPython File: Feature-engineering.ipynb)

Create New Artificial Training and Validation Data

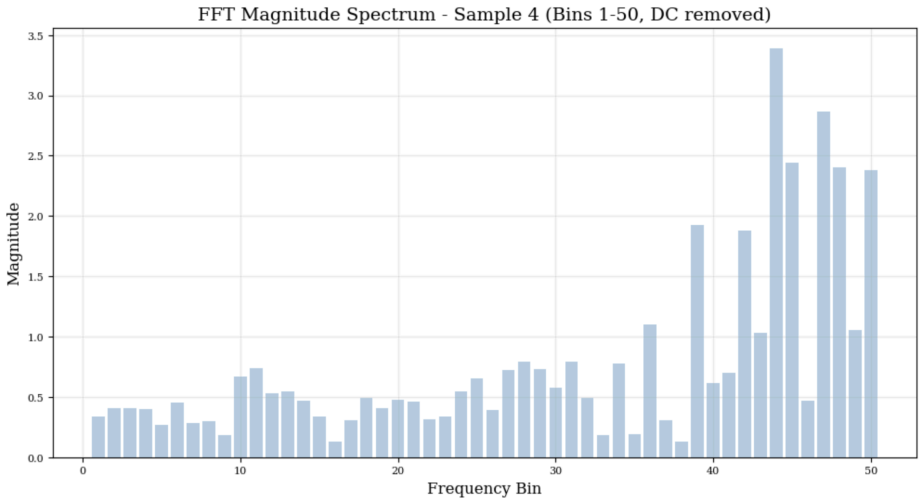
Our dataset is extremely small. Here we will create a new artificial dataset with the following techniques.

1. Add a new dataset with noise reduction (Fourier-based)
2. Add a new dataset with background sound removal
3. Shift the main audio component dataset (not executed yet, due to time limit)
4. Blend audio feature with the same animal category (not executed yet, due to time limit)

Subsequently, now we have more datasets, consisting of the original dataset, the noise reduction dataset, and the background removal dataset. The main idea of this approach is to encourage the model to recognize the main features or form of animal audio, whether it is in a noisy or clear condition.

Fourier-Based Filtering

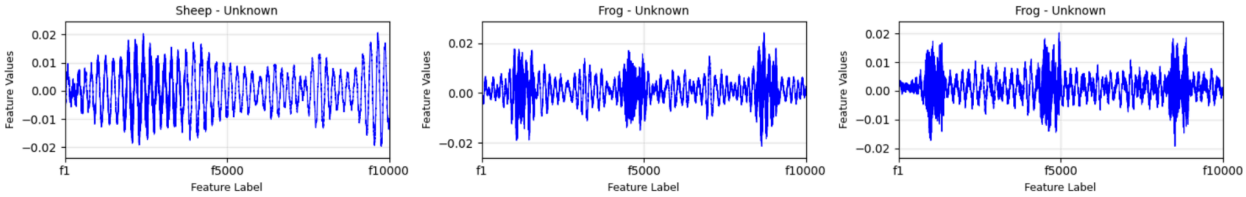
We transformed our audio (time-domain) into the frequency domain using the Fourier transform. Then we remove high-frequency levels to reduce noise. Noise is usually concentrated around high frequencies.



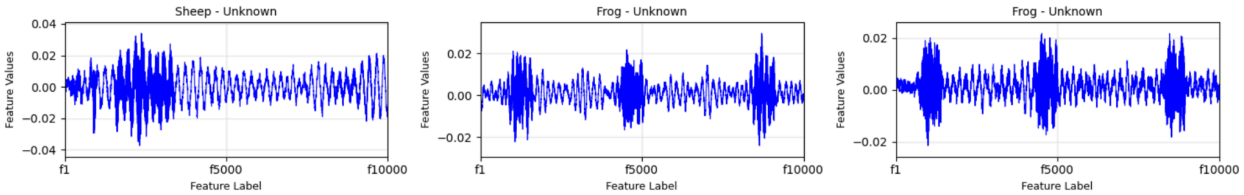
Frequency-based audio data

We removed high frequency at this step. Subsequently, we did an Inverse Fourier transform, after high-frequency removal, to convert the audio data into the time domain.

The result of noise reduction,

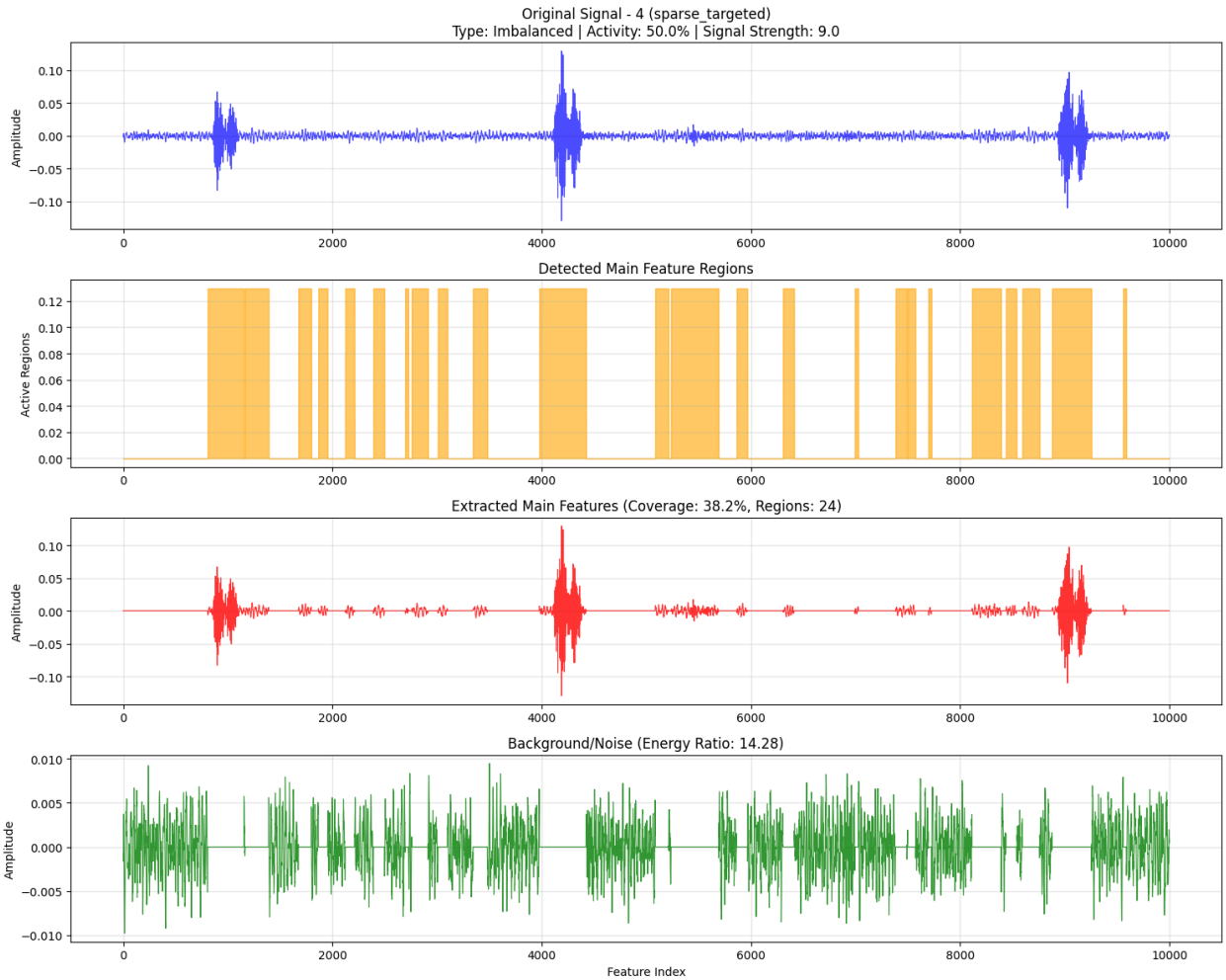


Compared to the original signals



Background Sound Removal

We performed background removal using adaptive thresholding.



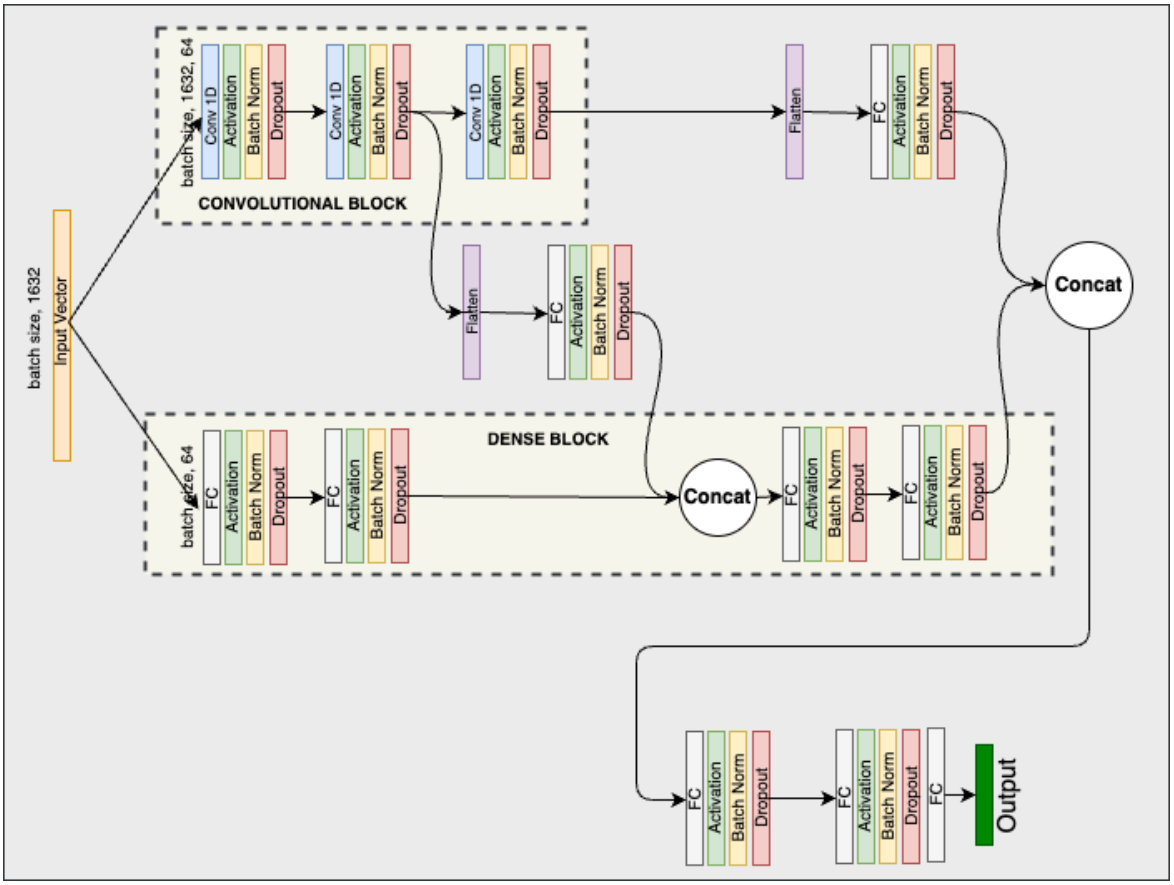
Red feature -> main features / main form of the animal sound, extracted for training synthetic data.

Consequently, we have an 840-dataset for training, and 240 for validation, three times more than the initial data.

For testing data, we didn’t perform data augmentation.

Modeling

Proposed Architecture



We specifically designed a neural network architecture to analyze the sound features, using the PyTorch library. We combined convolution and dense operations. Convolutions are excellent for sequential data analysis, like sound features data sequences.

The output consists of 10 neurons (same with animal classes)

The configurations of our neural networks are as follows,

Parameter	Nilai	Note
Optimizer	Adam optimizer	
Regularization	Dropout	
Dropout rates	0.25	
Variable initialization	Kaiming initialization	
Conv 1D kernel size	1 x 3	
Num of dense layer neurons	64	each layer
Num of the 1D convolution kernel	64	each layer
Learning rates	0.0001	
Learning rates decay	5%	
Learning rates patience	10	
Hidden layer activation	Leaky-ReLu	
Output activation	Softmax	

Loss Function

We applied softmax cross-entropy

Regularization

We applied dropout (25% rate)

Performance Metrics
We used Accuracy as a performance metric

Learning Result

```
Starting training...
New best validation loss: 0.325149 (previous: inf)
Epoch 1/2000 | Train Loss: 0.4867 | Val Loss: 0.3251 | Train Acc: 11.07% | Val Acc: 13.64% | LR: 0.000100 | 20:09:24***
Epoch 2/2000 | Train Loss: 0.4190 | Val Loss: 0.3364 | Train Acc: 17.32% | Val Acc: 10.91% | LR: 0.000100 | 20:12:31
Epoch 3/2000 | Train Loss: 0.3778 | Val Loss: 0.3454 | Train Acc: 24.48% | Val Acc: 9.55% | LR: 0.000100 | 20:14:42
Epoch 4/2000 | Train Loss: 0.3559 | Val Loss: 0.3561 | Train Acc: 26.95% | Val Acc: 10.23% | LR: 0.000100 | 20:17:18
Epoch 5/2000 | Train Loss: 0.3336 | Val Loss: 0.3764 | Train Acc: 29.82% | Val Acc: 10.45% | LR: 0.000100 | 20:20:00
Epoch 6/2000 | Train Loss: 0.3033 | Val Loss: 0.3883 | Train Acc: 35.81% | Val Acc: 13.86% | LR: 0.000100 | 20:22:44
Epoch 7/2000 | Train Loss: 0.2876 | Val Loss: 0.3829 | Train Acc: 37.50% | Val Acc: 13.86% | LR: 0.000100 | 20:25:21
Epoch 8/2000 | Train Loss: 0.2848 | Val Loss: 0.3663 | Train Acc: 40.76% | Val Acc: 15.91% | LR: 0.000100 | 20:27:56
Epoch 9/2000 | Train Loss: 0.2520 | Val Loss: 0.3551 | Train Acc: 44.92% | Val Acc: 18.64% | LR: 0.000100 | 20:30:20
Epoch 10/2000 | Train Loss: 0.2426 | Val Loss: 0.3578 | Train Acc: 49.22% | Val Acc: 19.55% | LR: 0.000100 | 20:33:14
Epoch 11/2000 | Train Loss: 0.2297 | Val Loss: 0.3749 | Train Acc: 52.73% | Val Acc: 20.23% | LR: 0.000100 | 20:35:47
Epoch 12/2000 | Train Loss: 0.2155 | Val Loss: 0.3790 | Train Acc: 53.91% | Val Acc: 21.82% | LR: 0.000100 | 20:37:10
Epoch 13/2000 | Train Loss: 0.2116 | Val Loss: 0.3529 | Train Acc: 56.12% | Val Acc: 23.86% | LR: 0.000100 | 20:39:00
New best validation loss: 0.307802 (previous: 0.325149)
Epoch 14/2000 | Train Loss: 0.1995 | Val Loss: 0.3078 | Train Acc: 57.16% | Val Acc: 29.77% | LR: 0.000100 | 20:42:07***
```

The model training losses are gradually decreasing across epochs. Subsequently, the training accuracies are increasing relatively fast. It achieved 52% accuracy only in the 11th epoch.
This evidences have some meanings:

1. There are distinctive features for each animal's sound
2. Our model can capture it

But our validation performances are struggling **(but slowly increasing)**, indicating the model is hard in generalizing the pattern. This evidence indicates that

1. Our dataset is too small; instead of generalizing the pattern, our model becomes memorized it.
2. Our dataset lacks variance.

To overcome this, we have to fine-tune,

1. Dropout rate
2. Model complexity (reduce the number of layers, or hidden neurons)
3. Increase (again) the dataset number and variations (by augmentation, or new human labelling)

Unfortunately, we do not have enough time to continue the training process for thousands of epochs.