# DEEP MEOW

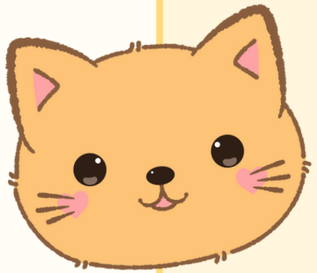
Antier Augustin
Causeur Lena
Prusiewicz-Blondin Louis

# OUTLINE

- Introduction
- Data
- Method : sound classification
  - Preprocessing
  - Audio Analysis
  - Why Mel Spectrogramm ?
  - CNN and Linear Classifier
- Results
- Discussion

# INTRODUCTION

**Goal : Sound classification to understand our cats**



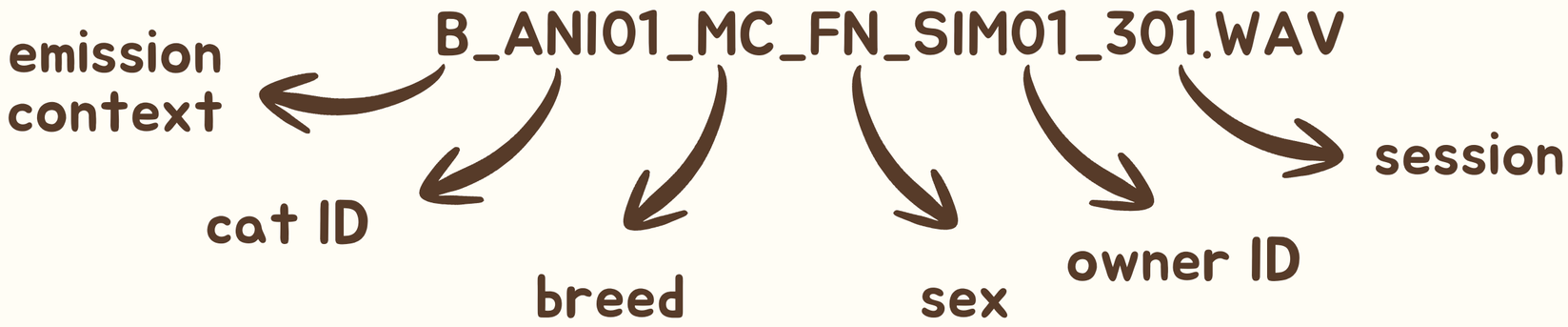
WHO? → For veterinarian or for cat owners'

WHY ? → Understanding our cats allows us to help them better and nurture a better relationship

HOW ? → Sound classification algorithms and a database of labeled meows. Each meows was emitted in a particular context.

# DATA

- 440 audio files provided by the university of Milan
- The metadata are contained in files name :




- Creation of a 'metadata' file

# DATA

- We are mainly focused on the first letter

B_ANI01_MC_FN_SIM01_301.WAV

3 emissions context

F = Waiting for food

I = Isolation in an unfamiliar environment

B = Brushing

# METHOD

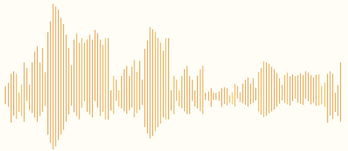
## What's a sound ?

## It's a wave with some parameters:

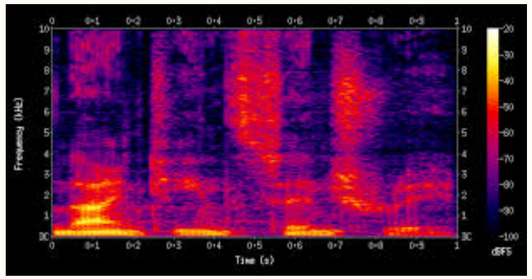
- Frequency
- Amplitude
- Sample rate

# PIPELINE

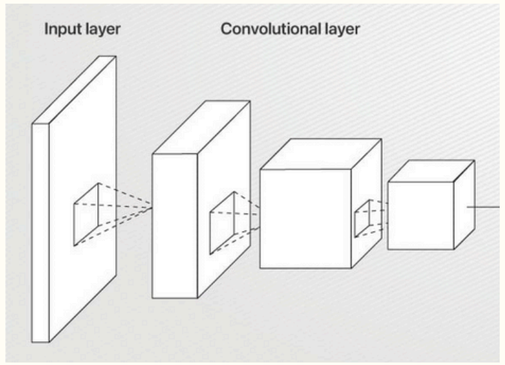
Audio wave



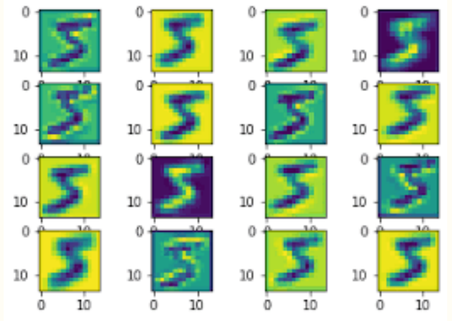
Mel Spectrogram



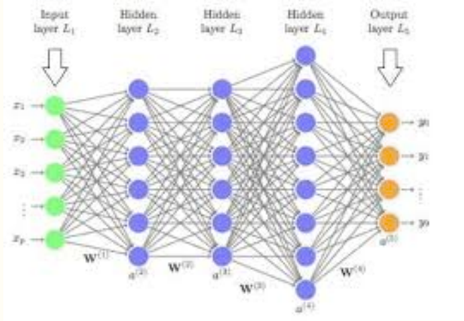

CNN Architechture




Feature Maps



Linear Classifier



# PREPROCESSING

## Convert to Mono

- All items must have the same dimension
- We convert mono files into stereo files

## Sample rate

- Standardizing sample rate
- Once again, all items must have the same dimension

## Length

- Truncate or fill audio to make them 4 000 ms long

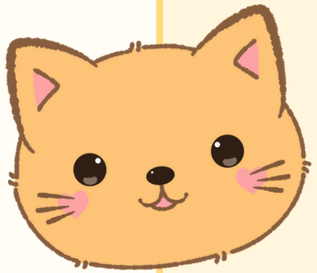
## Data Augmentation

- Time shift : the signal is shifted randomly
- Create more data improve the model

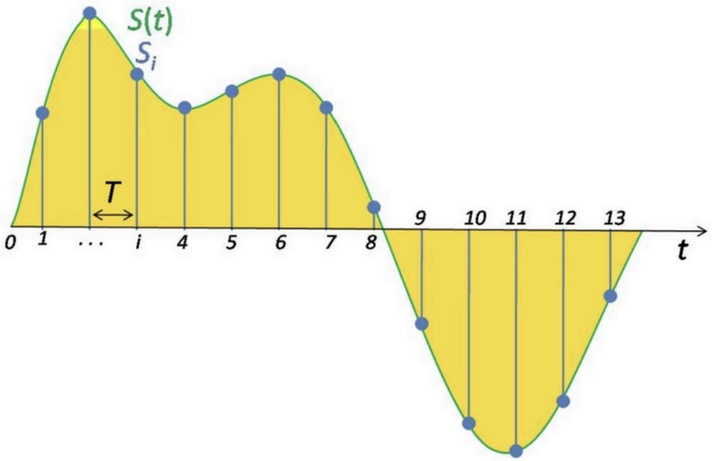
# AUDIO ANALYSIS

## How audio is represented digitally

Idea : We must transform the sound signal into a series of number to input it into our model



How : We measure the amplitude of the sound at fixed intervals of times. Each measurement is called a sample.

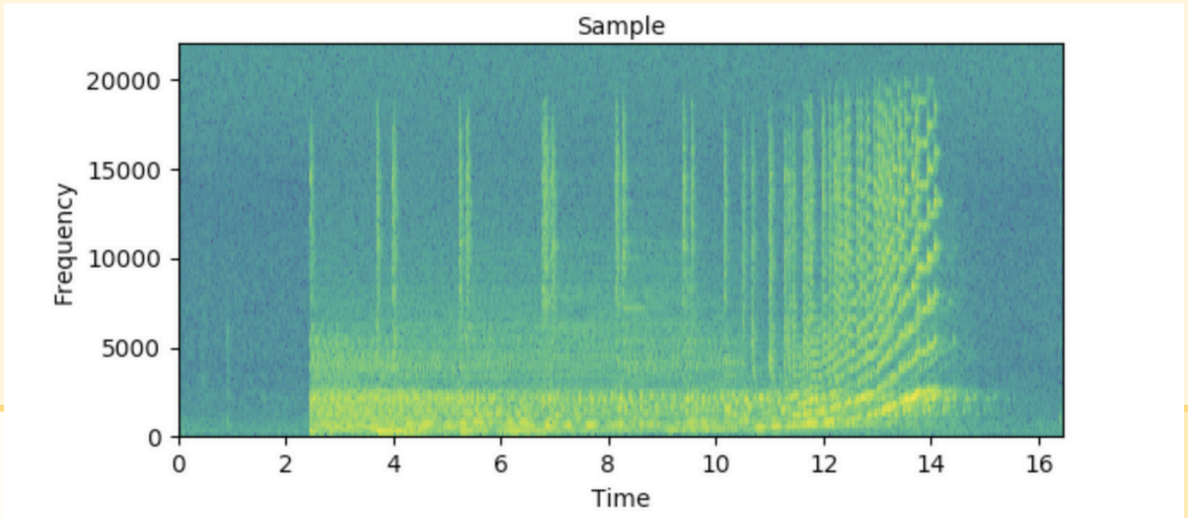



# AUDIO ANALYSIS

## How audio is represented digitally

To go further: convert the audio data into images as input for our CNN architecture

How : We generate a Spectrogram from the audio file



# AUDIO ANALYSIS

## How to generate a spectogram

Any sound signal is made up of multiple frequencies, each representing how many times the signal completes one full cycle per second. A spectrum shows which frequencies are present in the signal and the amplitude of each one.

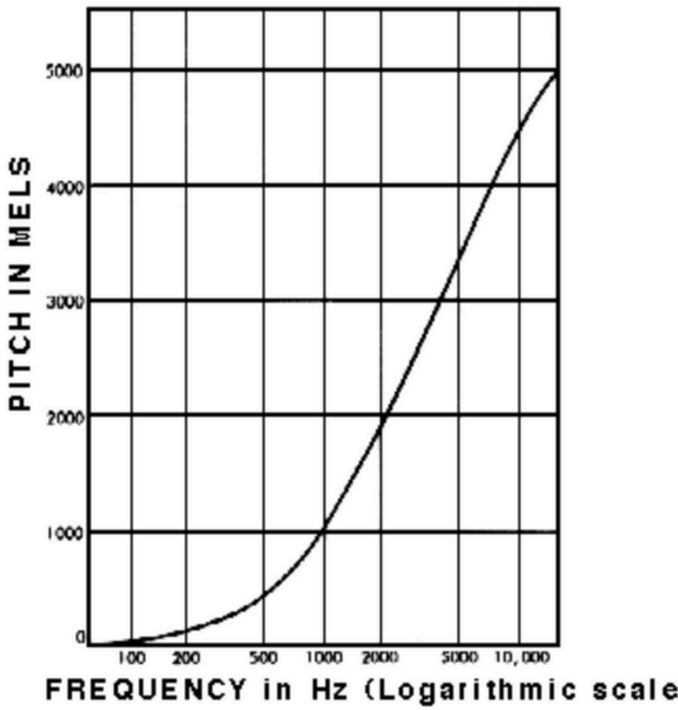
A spectrogram extends this idea by showing how the spectrum changes over time. It is like a 'photograph' of the signal, with time on the x-axis, frequency on the y-axis, and amplitude represented by color or intensity. Spectrograms are created using Fourier Transforms, which break the signal into its constituent frequencies at each moment in time.

Now we can use our spectrogram as an input into our CNN

# MEL SPECTROGRAM

## In our pipeline, we will be using a Mel spectrogram instead of a Spectrogram but why ?

- The Mel scale represents frequency in a way that match human pitch perception. Therefore it compresses high frequencies and expands low frequencies.
- This makes meaningful spectral pattern more distinct in the representation
- Since deep-learning models learn from these patterns, better separation of perceptually relevant features usually leads to better classification accuracy than using a standard linear spectrogram.




A Mel-spectrogram often lead to better classification because they produce a representation where important audio features are more clearly separated

# CNN

- Our CNN comes from a pre-existing architecture. It consists of 4 layers of convolutions, of different sizes, with different strides and padding.

```
# First Convolution Block with Relu and Batch Norm. Use Kaiming Initialization
self.conv1 = nn.Conv2d(2, 8, kernel_size=(5, 5), stride=(2, 2), padding=(2, 2))
self.relu1 = nn.ReLU()
self.bn1 = nn.BatchNorm2d(8)
init.kaiming_normal_(self.conv1.weight, a=0.1)
self.conv1.bias.data.zero_()
conv_layers += [self.conv1, self.relu1, self.bn1]
```

# LINEAR CLASSIFIER

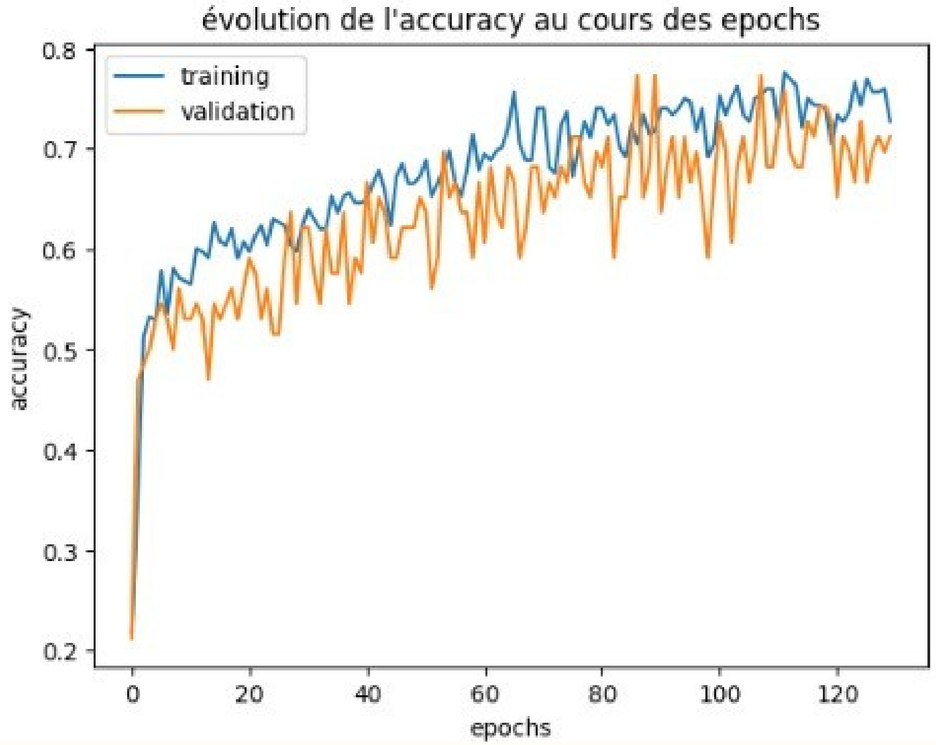
- The Linear Classifier also comes from a pre-existing architecture. It only consists of one linear Layer, with an out features of 3

```
# Linear Classifier
self.ap = nn.AdaptiveAvgPool2d(output_size=1)
self.lin = nn.Linear(in_features=32, out_features=3)
```

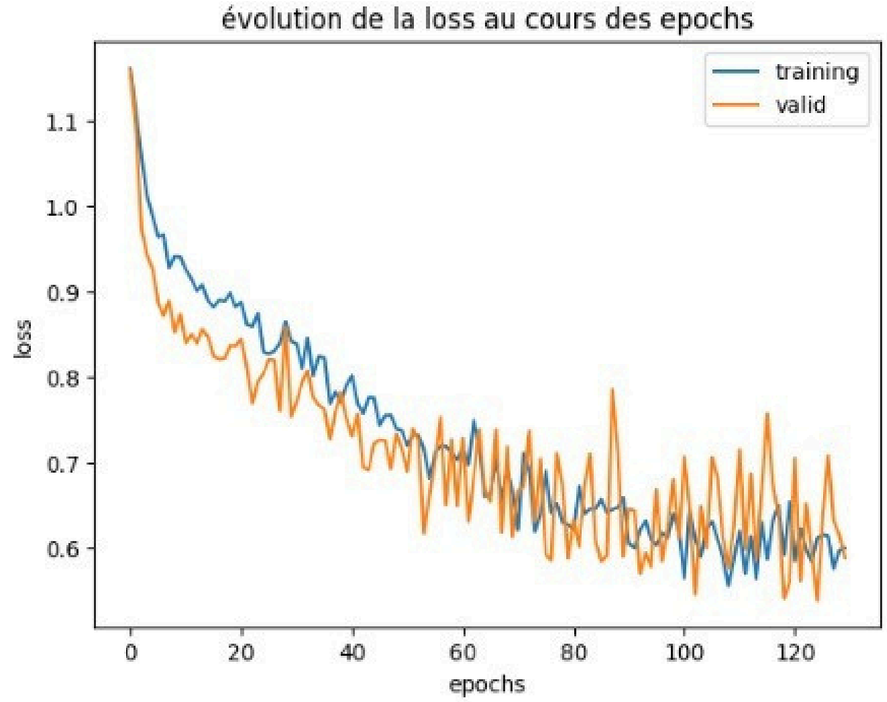
# RESULTS

## Evolution of accuracy overs epochs

...




## Evolution of loss over epochs




```
la performance à la dernière itération (validation) est : 0.71
0.67. 0.71
```

# DISCUSSION

## Opening

- Our model could be used in a different context, like in an agronomical context where we would, listen to the sounds of cows, goats or pigs and interpret it
- A RNN architecture could have also been relevant.
- We could have try to predict not only the emission context of the meow, but the species of the cat that meows and the emission context.

# DISCUSSION

## Limits

- The dataset contains only 440 meows produced by 21 cats only across 2 species
- The paper that was published with the dataset isn't open source
- Cats practically never meow in the wild only domestic cats meow, and these meows are based on human education which is  diverse
- Even though we can labelise easily cat meows', means of communications can vary amongsts different species and our model mignht not work on different species
- Because of the stochastic nature of our model, we sometimes find results that are  abnormal (it happens once every 4 runs)

# SOURCES


Doshi K (2021a) Audio Deep Learning Made Simple (Part 1): State-of-the-Art Techniques. Towards Data Science

Doshi K (2021b) Audio Deep Learning Made Simple (Part 2): Why Mel Spectrograms perform better. Towards Data Science

Doshi K (2021c) Audio Deep Learning Made Simple (Part 3): Data Preparation and Augmentation. Towards Data Science

Doshi K (2021d) Audio Deep Learning Made Simple: Sound Classification, step-by-step. TDS Archive

Ludovico LA, Ntalampiras S, Presti G, Cannas S, Battini M, Mattiello S CatMeows: A Publicly-Available Dataset of Cat Vocalizations. doi: https://doi.org/10.5281/zenodo.4008297