Classifying music genres

Thomas Schaller | Dorian Guyot | Jonathan Péclat

SUMMARY

- Task description & Motivation
- Challenges
    - Data preparation
- State of the art (prior work)
- Our solution
- Results

Task description & Motivation

- Classify a music in a genre :
  - Blues, Classical, Country, Disco, Hip Hop
  - Jazz, Metal, Pop, Reggae, Rock.

- Genre
  - quickly describe music
  - quite consensual among people

- Our solution may help
  - find music
  - uninitiated people

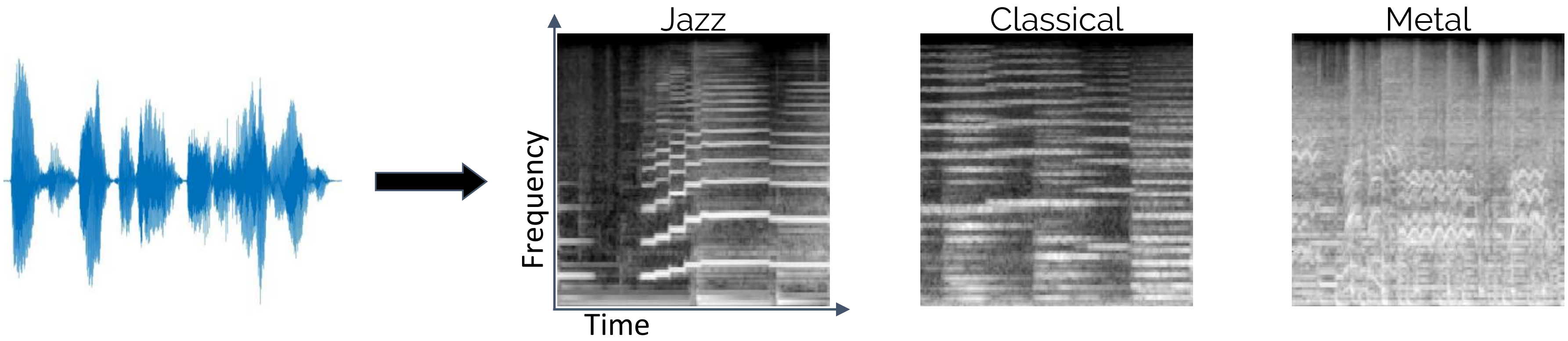
Challenges

- Understand audio waveform
  - Use FFT

- Rather small dataset (1000 sounds)
  - Data augmentation
  - Transfer learning

Data preparation

- Manual feature extraction: spectrograms (log scale)
  - Reduced dimensionality (600k to 45k)
  - Closer to human representation of music



Cherry-picked examples

Data augmentation

- GTZAN dataset is small (100 30-seconds samples / genre, 10 genres)

Solutions:

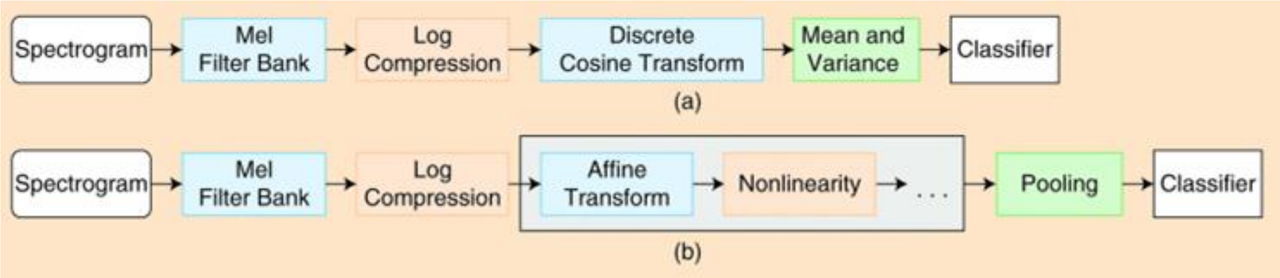
- Split the samples in smaller samples (human still able to guess genre)
- Rolling window
- Color jittering on spectrograms (~white noise)

Further improvements:

- Data augmentation directly on audio (pitch and tempo shifting, noise)

State of the art (prior work)

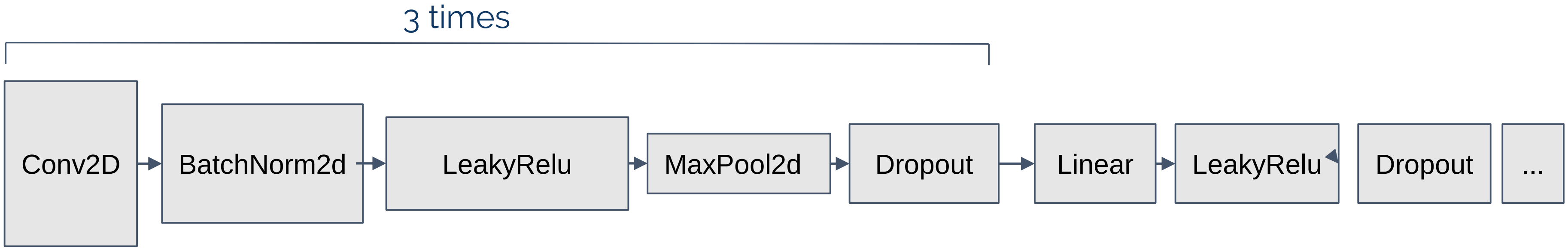
- Feature extraction and conventional ML (~2010)



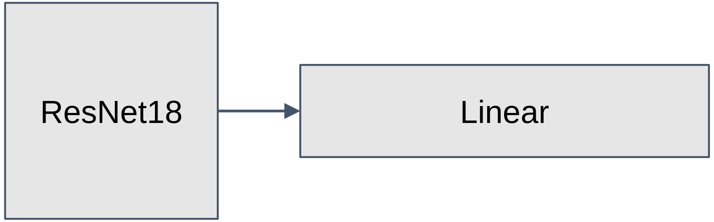
- Deep CNN on spectrogram (Recently)
- Deep CNN directly on Waveform (Most recently)
  - Sample-level CNN
    - First end-to-end with state of the art performance
    - 90% accuracy on MTAT and 88% on MSD dataset
  - Recurrent CNN are also promising

Reference: NAM, Juhan, CHOI, Keunwoo, LEE, Jongpil, *et al.* Deep Learning for Audio-Based Music Classification and Tagging: Teaching Computers to Distinguish Rock from Bach. *IEEE Signal Processing Magazine*, 2019, vol. 36, no 1, p. 41-51.

Solution 1: custom network



Solution 2: transfer learning



- ➔ Optimizer: Adam
- ➔ Loss: Cross Entropy
- ➔ Learning rate: 0.001
- ➔ Learning Rate reducer
- ➔ Number of epochs: 50
- ➔ Keep the smallest loss
- ➔ Input: 216x216
- ➔ Output: 10

Results and analysis: Fully trained model

Before data augmentation	13% on test set
Before data augmentation	32%
Too much data augmentation	15%
Good data augmentation	70%
After adding brightness and contrast	75% with a loss of 1.13

Conv2D with kernel size of (5,1)

Results and analysis: Transfer learning

Fixed features	68% with a loss of 0.92
Finetuning	**95% with a loss of 0.16**

Conclusions

- Pretty happy about the final result
- Frustrated to think about a model for a long time to finally get better result with transfer learning
  - But it's normal, a model with 175 layers trained on a lot of images
- Data augmentation was really important in this project

QUESTIONS ?