CLASSIFICATION MODEL TO DETERMINE MUSIC TRACKS MOOD USING PYTORCH

5th, February 2020

- Aicha BOKBOT
- Eva FRANCOIS
- Arthur KRIEFF
- Corentin SENE

PROJECT DESCRIPTION

Goal

- Classify the mood of a given song into 4 classes:
  - **Happy**
  - **Angry**
  - **Sad**
  - **Relaxed**

Challenges

- Mood can be subjective
- Can be hard to define even for a human being
- Possible to have a song that fits into none of the classes

Sources

- Freely-available collection of audio tags and metadata for a million contemporary popular music tracks
- Spotify's Web API to download the MP3 file (30s)

Models

CNN based on spectrogram images

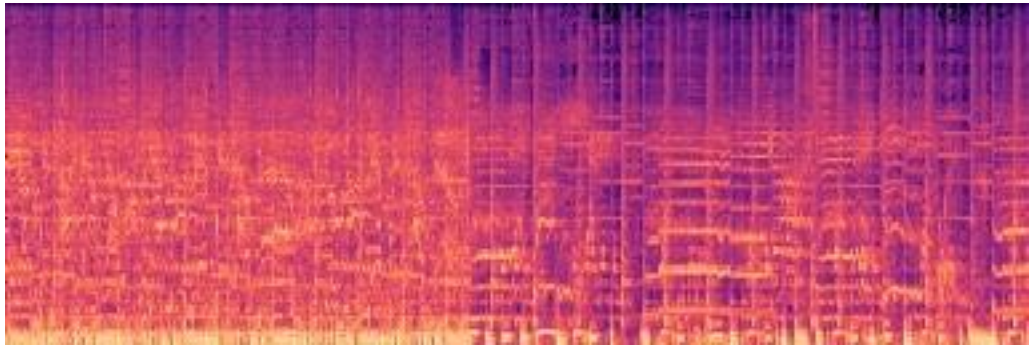
PyTorch

DATA COLLECTION AND PREPARATION

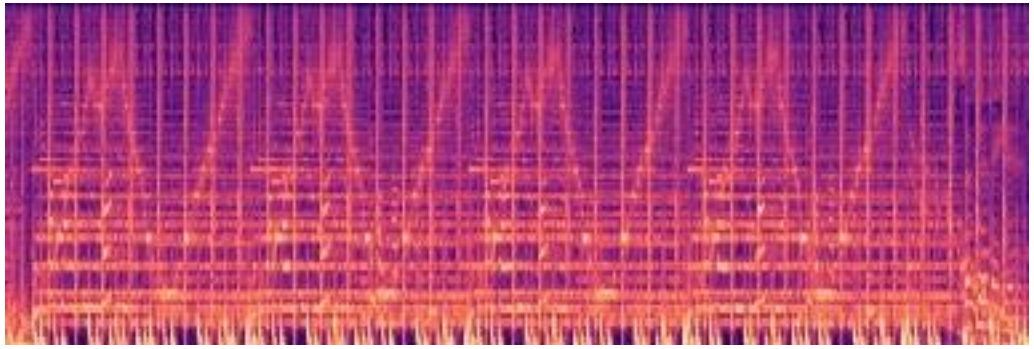
Data Preparation

	Step 1	Step 2	Step 3
Input	last.fm J-son encoded text files Keys: artist, title, timestamp, tags Train and test set provided	Spotify Spotify developer API + Train and test csv files	Mp3 files
Steps	1. Download last.fm dataset 2. Filter songs with moods of interest 3. Convert J-son to Dataframe	1. Create Spotify client credentials 2. Fetch url of songs from train and test csvl files 3. Download mp3 30s-preview	1. Convert mp3 to WAV 2. Use python library Librosa to open WAV files and display spectrograms 3. Save spectrograms as JPG images
Output	Train and Test csv files Columns: ["artist", "timestamp", "tags", "track_id", "title", "tags_list"]	Mp3 30s-preview files	Spectrograms as JPG files

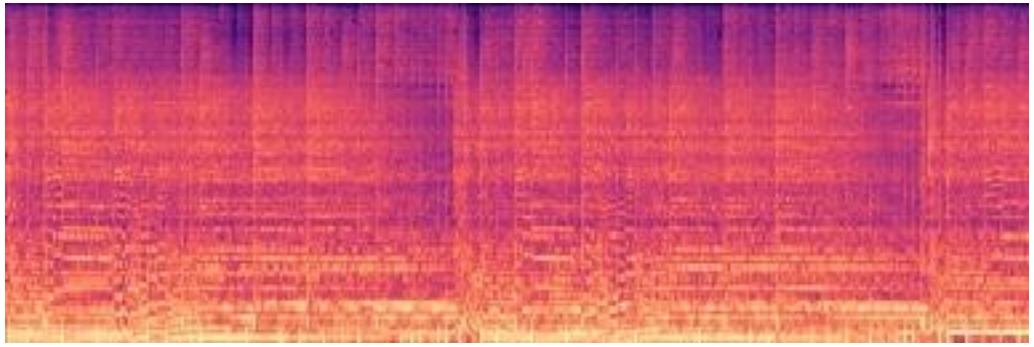
EXAMPLES OF SPECTROGRAMS



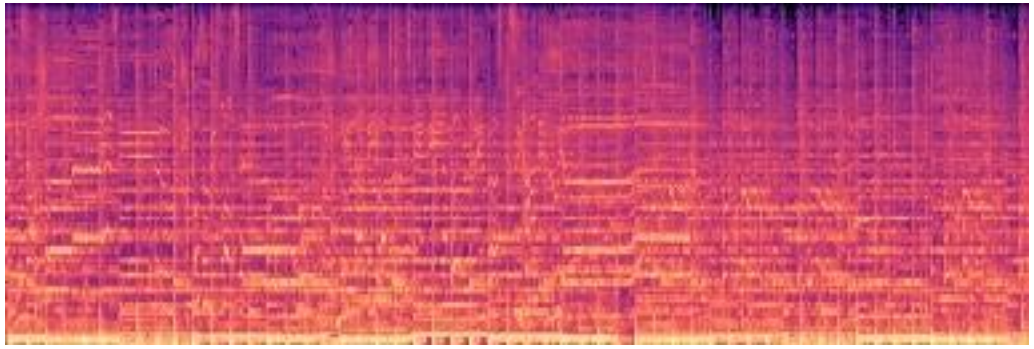
Angry



Happy



Relaxed



Sad

FINAL MODELS

	Model 1:	Model 2:
Model	Resnet18 – No Pretrained Small dataset balanced (~8000 spectrograms) **4 labels** (angry, happy, sad, relaxed)	Resnet18 – Pretrained Large dataset imbalanced (~23000 spectrograms) **3 labels** (angry, happy, sad)
Parameters	• Epochs: 20 • Batch size: 32 • Optimizer: SGD (lr = 0.001, momentum = 0.9)	• Epochs: 15 • Batch size: 64 • Optimizer: SGD (lr = 0.001, momentum = 0.9)
Score	• Train accuracy: 0.85 • **Test accuracy: 0.60**	• Train accuracy: 0.73 • **Test accuracy: 0.74**
Evaluation	+ • Mood tags exhaustive • Balanced dataset — • Overfitting • Low accuracy • Mixing sad & relaxed	+ • Robust model • Pretty good accuracy • Fast (transfer learning) — • Labels non-exhaustive • Imbalanced dataset

FINAL MODELS

Modelling

Model 1:

Model

Resnet18 – No Pretrained
Small dataset balanced (~8000 spectrograms)
4 labels (angry, happy, sad, relaxed)

Parameters

- Epochs: 20
- Batch size: 32
- Optimizer: SGD (lr = 0.001, momentum = 0.9)

Score

- Train accuracy: 0.85
- **Test accuracy: 0.60**

Evaluation

+
- Mood tags exhaustive
- Balanced dataset

−
- Overfitting
- Low accuracy
- Mixing sad & relaxed

Model 2:

Resnet18 – Pretrained
Large dataset imba ctrograms)

- Epoch
- Batch
- Optim 9)

- Train a
- **Test a**

+
- Robust model
- Pretty good accuracy
- Fast (transfer learning)

−
- Labels non-exhaustive
- Imbalanced dataset

Cross-table

pred	angry	happy	relaxed	sad
true				
angry	352	50	21	34
happy	65	284	49	56
relaxed	15	68	245	112
sad	43	61	146	202

FINAL MODELS

	Model 1:	Model 2:
Model	Resnet18 – No Pretrained Small dataset balanced (~8000 spectrograms) **4 labels** (angry, happy, sad, relaxed)	Resnet18 – Pretrained Large dataset imbalanced (~23000 spectrograms) **3 labels** (angry, happy, sad)
Parameters	• Epochs: 20 • Batch size: 32 • Optimizer: SGD (lr = 0.001, momentum = 0.9)	• Epochs: 15 • Batch size: 64 • Optimizer: SGD (lr = 0.001, momentum = 0.9)
Score	• Train accuracy: 0.85 • **Test accuracy: 0.60**	• Train accuracy: 0.73 • **Test accuracy: 0.74**
Evaluation	+ • Mood tags exhaustive • Balanced dataset – • Overfitting • Low accuracy • Mixing sad & relaxed	+ • Robust model • Pretty good accuracy • Fast (transfer learning) – • Labels non-exhaustive • Imbalanced dataset

WEB APPLICATIONPP DEMONSTRATION

Reporting

Mood Detection Algorithm

Goal is to detect the mood of a given song

Music Detection

Select

- (•) Spotify Search
- ( ) Upload Song

Enter Artist

LIMITATIONS & NEXT STEPS

Limitations

- **Imbalanced dataset** for the 2nd model
- **Spectrograms do not contain all the information** concerning the mood/emotion of a song
- A song can have **several mood tags**
- **Models' accuracy** is not very high
- Works only on **30s audio**

Next steps

- **Design a bimodal model** that combines NLP (Bert) for lyrics and CNN for spectrograms
- **Improve the model performance** (with more data, more layers and a better hyperparameters tuning)
- **Extends the number of moods** to be more exhaustive
- Have the model work on >30s audio

GIT PRESENTATION

Name	Last commit	Last update
app	slight update	10 minutes ago
data	removed sad	6 days ago
notebooks	rename	1 hour ago
src	print update	39 minutes ago
.gitignore	fix	1 week ago
Makefile	modif makefile	58 minutes ago
README.md	Update README.md	5 minutes ago
requirements.txt	add image in notebook	1 hour ago
setup.py	add setup	3 hours ago

README.md

Music Mood Classification

Project Description

The objective is to build a music mood classification system on Pytorch using Deep Learning. The model would classify music tracks into 4 different labels: Happy, Sad, Angry and Relaxed. The model would learn patterns for each type of music to be able to detect the mood of new music tracks based on Spectograms.

Usage

Virtual environment setup

APPENDIX

Model 1

Split based on sample(2500) for each category then train_test_split

TRAIN: 5,400 spectograms
- Happy: 1,359
- Sad: 1,355
- Angry: 1,370
- Relaxed: 1,316

TEST: 1,803 spectograms
- Happy: 454
- Sad: 452
- Angry: 457
- Relaxed: 440

Model 2

Split based on last fm datasets

TRAIN: 20,843 spectograms
- Happy: 8;875
- Sad: 5,561
- Angry: 1,652
- Relaxed: 4,755

TEST: 2,559 spectograms
- Happy: 1,153
- Sad: 603
- Angry: 169
- Relaxed: 634