

# UE Signal Apprentissage Multimedia - séance 3

Valentin Emiya

M2 IAAA

16 décembre 2019

1

## TP

- Le projet
- Les données
- Aujourd'hui : modalités séparées, features standards

# Outline

- 1 TP
  - Le projet
  - Les données
  - Aujourd'hui : modalités séparées, features standards

# Les grandes lignes du projet

- Tâche : classification de genre musical sur données multimodales
- Données : audio, images (pochettes d'album)
- Solutions : développer plusieurs systèmes
  - sur chaque modalité séparément<sup>1</sup> / en multimodal
  - en extrayant des descripteurs standards<sup>1</sup> / en apprenant des représentations
- Prise en main des outils pour travailler sur des données brutes
- Références

Oramas, Barbieri, Nieto, and Serra, *Multimodal Deep Learning for Music Genre Classification*, Trans. of ISMIR, 2018.

Oramas, Nieto, Barbieri, and Serra, *Multi-label Music Genre Classification from audio, text and images using Deep Features*. In Proc. of ISMIR Conf, 2017.

# Aperçu des jeux de données

Les jeux à votre disposition :

- Un jeu de sons pour la classification de compositeur
- Un jeu d'image de pochettes pour la classification de genre (single-label)
- Un jeu d'images de pochettes pour la classification multilabel de genre musical

Toutes les données sont téléchargeables ici :

<https://amubox.univ-amu.fr/s/jM5maPc3ymY7SB6>

## msdi\_img

Classification multiclasse, single-label

- Données : 16804 images couleurs  $200 \times 200$
- Labels : 15 genres ('Metal', 'Rock', 'Reggae', 'Jazz', 'Latin', 'Folk', 'Electronic', 'Pop', 'RnB', 'Rap', 'World', 'Country', 'Punk', 'Blues', 'New Age')

Référence : <https://zenodo.org/record/1240485#.XfKfb-tCf0Q>

# mumu\_img

## Classification multiclasse, multi-label

- Données : 31471 images couleurs  $200 \times 200$
- Labels : 446 genres ; 191345 labels soit en moyenne 6 genres/exemples

Référence : <https://zenodo.org/record/1236906#.XfQCYutCf0Q>

## maps\_composers

Remarque : les données brutes audio pour la classification de genre ne sont pas encore accessibles ; on travaille sur une tâche similaire en attendant.

Classification multiclasse, single-label

- Données : 141 enregistrements de musique de piano de 30s
- Labels : 4 compositeurs (Beethoven, Chopin, Liszt, Mozart)

Référence : <http://www.tsi.telecom-paristech.fr/aao/en/2010/07/08/>

maps-database-a-piano-database-for-multipitch-estimation-and-automatic-transcripti



# Classifieurs sur les sons

## Description du système

- prendre des extraits de morceaux de 10 à 15s : un extrait = un exemple
- segmenter les sons par trames de 100 ms environ
- calculer des features sur chaque trame, par exemple (au choix)
  - MFCC, et éventuellement les  $\Delta$ MFCC et les  $\Delta\Delta$ MFCC
  - Chroma, et éventuellement leur  $\Delta$ Chroma et les  $\Delta\Delta$ Chroma
- agréger les features calculés dans chaque trame pour obtenir des features au niveau de l'extrait, par exemple (au choix)
  - calculer la moyenne des features sur toutes les trames
  - calculer un histogramme des features sur toutes les trames
- normaliser les features
- utiliser un classifieur de votre choix

## Outils Python

- librosa pour la segmentation et l'extraction de features
- sklearn pour les classifieurs

# Classifieurs sur les images

## Description du système

- calculer des features sur chaque image, par exemple (au choix)
  - sous-échantillonnage 3x3
  - HOG 3x3
- normaliser les features
- utiliser un classifieur de votre choix

## Outils Python

- coder soi-même le sous-échantillonnage 3x3 en comparant plusieurs solutions :
  - sous-échantillonnage naïf (`image[:, :p, ::q, :]`)
  - moyenne sur chaque block (`skimage.transform.downscale_local_mean`)
  - sous-échantillonnage propre (`scipy.signal.resample`)
- HOG : utiliser `scikit-image` (`skimage.feature.hog`)
- `sklearn` pour les classifieurs

# Résumé du travail attendu pour ce TP

- prendre en main les données : ce sont des données brutes, vous pouvez écouter les sons, regarder les images !
- construire un classifieur de compositeur pour l'audio, évaluer les performances sur `maps_composers`
- construire un classifieur de genre simple label pour les images, évaluer les performances sur `msdi_img`
- optionnel : construire un classifieur de genre multilabel pour les images, évaluer les performances sur `mumu_img`
- exhiber des exemples mal classés et essayer d'interpréter les erreurs

Vous pouvez travailler à deux et vous répartir les tâches.

# Conseils d'architecture logicielle

Séparez les parties d'extraction de features et d'apprentissage

- extraction de features : créez des fonctions qui lit chaque fichier son/image, calcul le vecteur de descripteur et le stocke dans un ou plusieurs fichiers  
*soit un fichier par fichier d'entrée, portant le même nom ; soit un fichier unique contenant la matrice  $n \times d$  des features de taille  $d$  pour les  $n$  exemples*
- normalisation et classification : travaillez sur les vecteurs de features directement, faites un pipeline sklearn.