# Classification des types d’instruments de musique

Avec Scikit learn

**Problématique** : Comment reconnaître le type d’un instrument de musique ?

# Le plan :

1. Introduction
2. L’origine des données
3. Exploration des données
4. Transformée de Fourier
5. Spectrogrammes
6. Conclusion

# 1. Introduction

Guitar

Flute

Bass

Keyboard

Brass

**Classification supervisée**

**Model-based learning**

Mallet

Organ

Reed

String

Synth

Vocal

**Seuil de satisfaction ?**

# 9.09%

11 types d’instruments :

$$1/11 \approx 0.0909 \approx 9.09\%$$

# 2. Origine des données

magenta Get Started Studio Demos Blog Research Talks Community

# The NSynth Dataset

Apr 5, 2017
• NSynth

A large-scale and high-quality dataset of annotated musical notes.

## Download

## Contents



source : https://magenta.tensorflow.org/datasets/nsynth

# 2. Origine des données

## Format

### Files

The NSynth dataset can be download in two formats:

- TFRecord files of serialized TensorFlow Example protocol buffers with one Example proto per note.
- JSON files containing non-audio features alongside 16-bit PCM WAV audio files.

The full dataset is split into three sets:

- **Train** [tfrecord | json/wav]: A training set with 289,205 examples. Instruments do not overlap with valid or test.
- **Valid** [tfrecord | json/wav]: A validation set with 12,678 examples. Instruments do not overlap with train.
- **Test** [tfrecord | json/wav]: A test set with 4,096 examples. Instruments do not overlap with train.

Below we detail how the note features are encoded in the Example protocol buffers and JSON files.

source : https://magenta.tensorflow.org/datasets/nsynth

# 3. Exploration des données

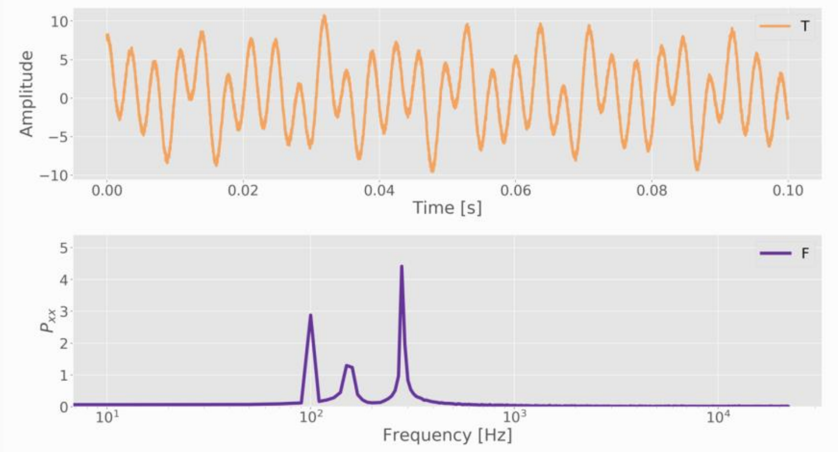
| Feature | Type | Description |
|---|---|---|
| note | `int64` | A unique integer identifier for the note. |
| note_str | `bytes` | A unique string identifier for the note in the format <instrument_str>-<pitch>-<velocity> . |
| instrument | `int64` | A unique, sequential identifier for the instrument the note was synthesized from. |
| instrument_str | `bytes` | A unique string identifier for the instrument this note was synthesized from in the format <instrument_family_str>-<instrument_production_str>-<instrument_name> . |
| pitch | `int64` | The 0-based MIDI pitch in the range [0, 127]. |
| velocity | `int64` | The 0-based MIDI velocity in the range [0, 127]. |
| sample_rate | `int64` | The samples per second for the `audio` feature. |
| audio* | `[float]` | A list of audio samples represented as floating point values in the range [-1,1]. |
| qualities | `[int64]` | A binary vector representing which sonic qualities are present in this note. |
| qualities_str | `[bytes]` | A list IDs of which qualities are present in this note selected from the sonic qualities list. |
| instrument_family | `int64` | The index of the instrument family this instrument is a member of. |
| instrument_family_str | `bytes` | The ID of the instrument family this instrument is a member of. |
| instrument_source | `int64` | The index of the sonic source for this instrument. |
| instrument_source_str | `bytes` | The ID of the sonic source for this instrument. |

Durée : 4 sec
Échantillonnage : 16 kHz

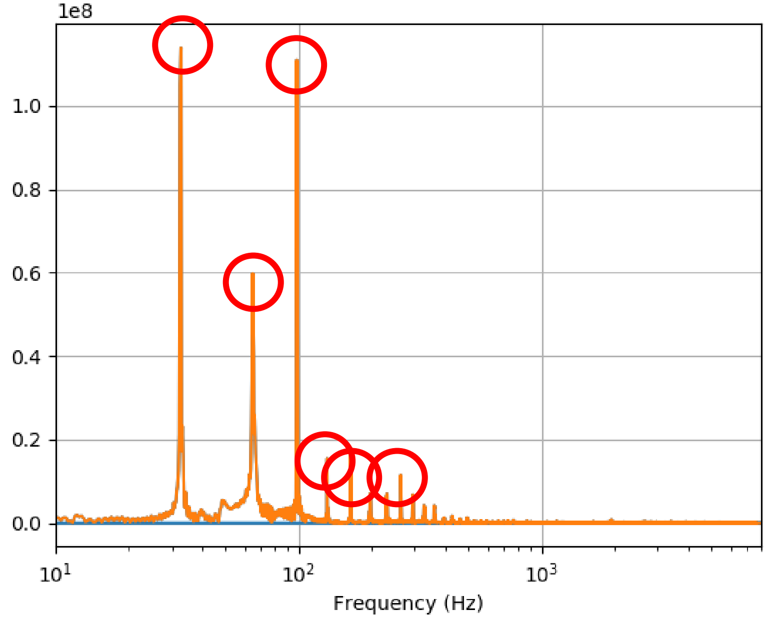
source : https://magenta.tensorflow.org/datasets/nsynth

# 4. Transformée de Fourier

Exemple de transformée de Fourier :



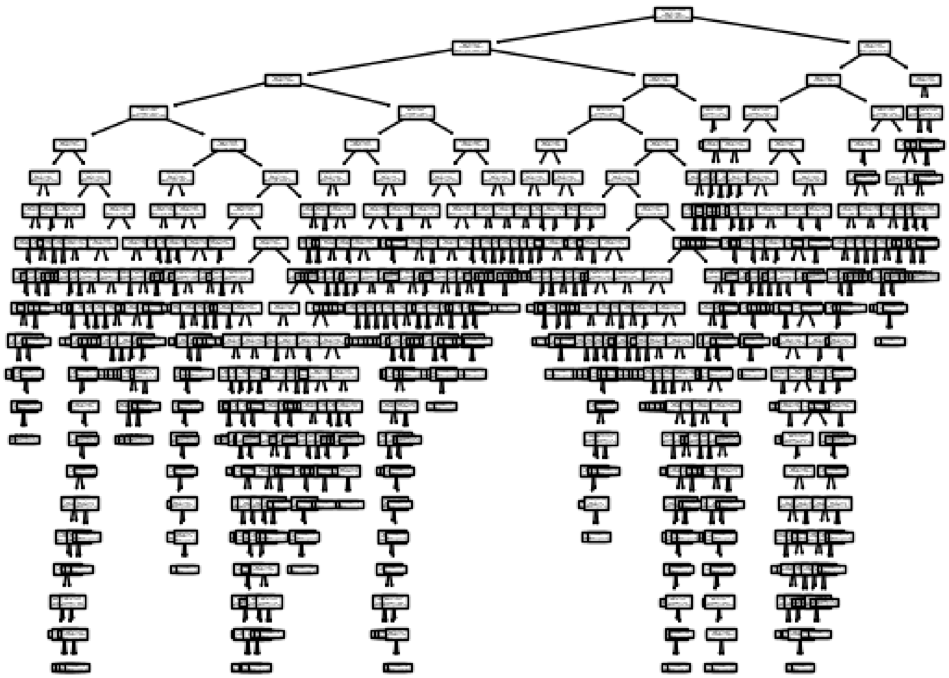

Transformée de Fourier pour une bass :




**La transformation de Fourier** est une opération qui transforme une fonction intégrable et continue sur $\mathbb{R}$ en une autre fonction, décrivant le spectre fréquentiel de cette dernière.

# 4. Transformée de Fourier

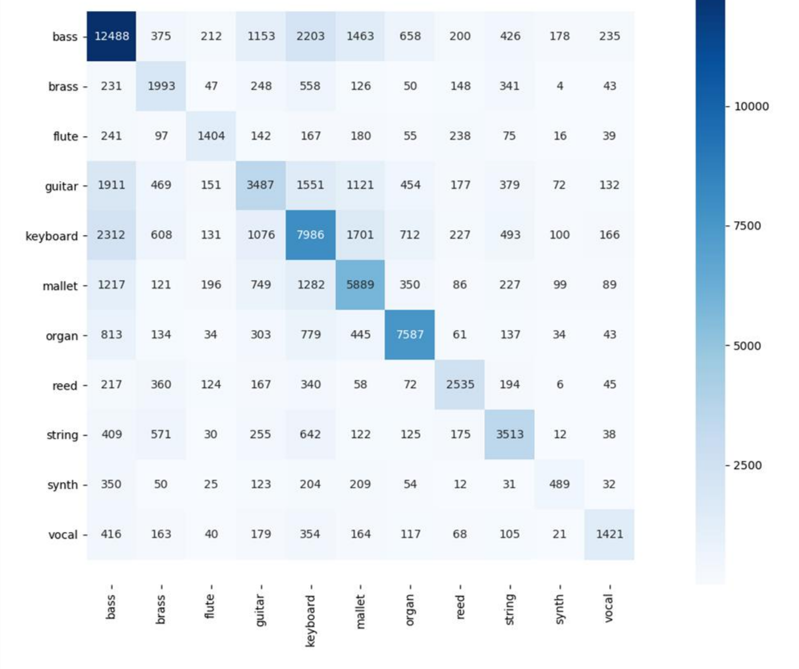
Arbre de décision :



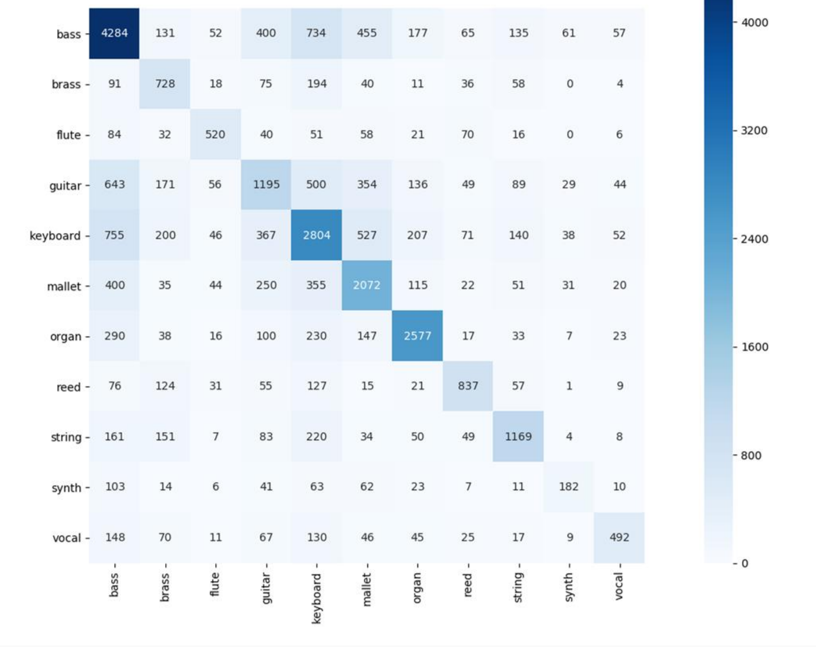
# 4. Transformée de Fourier

## a. Exploration initiale - Decision Tree

**Tous instruments 70% train, 30% test**
**Tous les échantillons :** 289 205
**Précision :** 56%

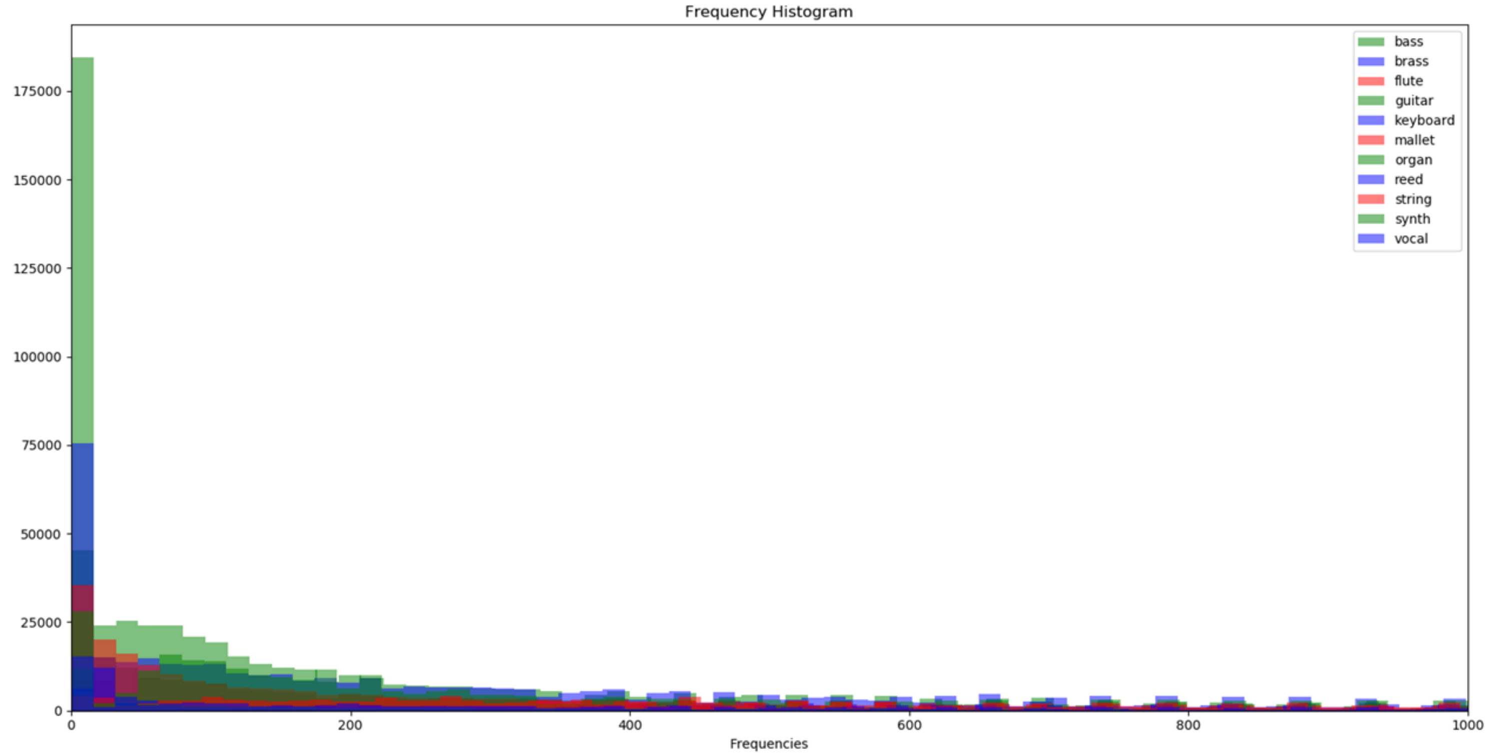



**Tous instruments 90% train, 10% test**
**Tous les échantillons :** 289 205
**Précision :** 58.29%



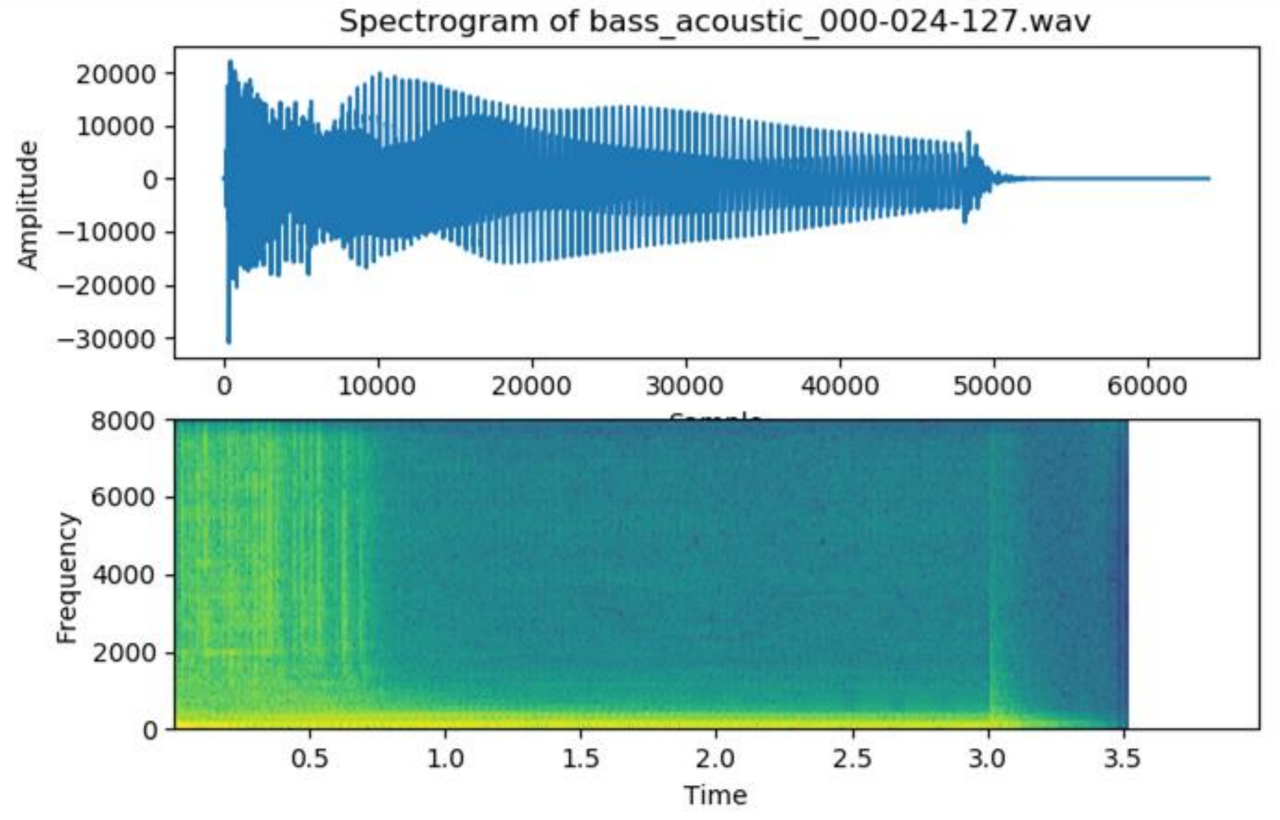
# 4. Transformée de Fourier

## a. Exploration initiale



# 5. Spectrogrammes

## Qu’est-ce qu’un spectrogramme audio ?

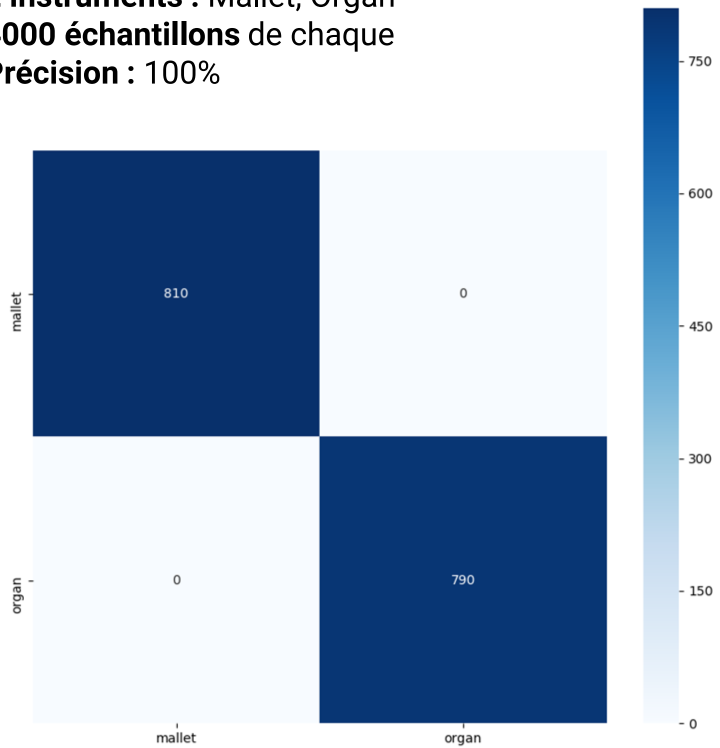


# 5. Spectrogrammes

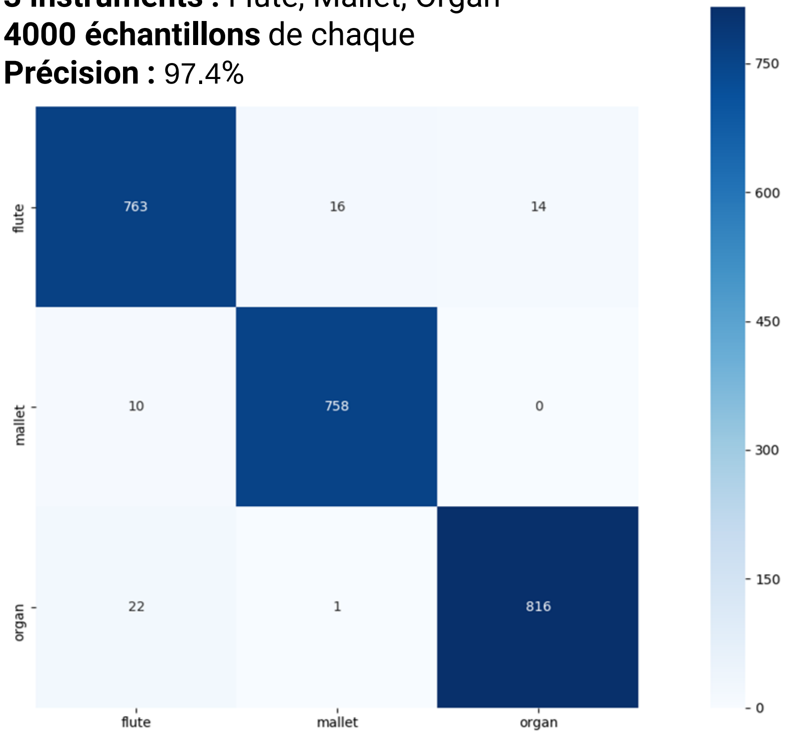
*80% train, 20% test, sur toutes les diapositives suivantes*

## a. Exploration initiale - Decision Tree

**2 instruments :** Mallet, Organ
**4000 échantillons** de chaque
**Précision :** 100%



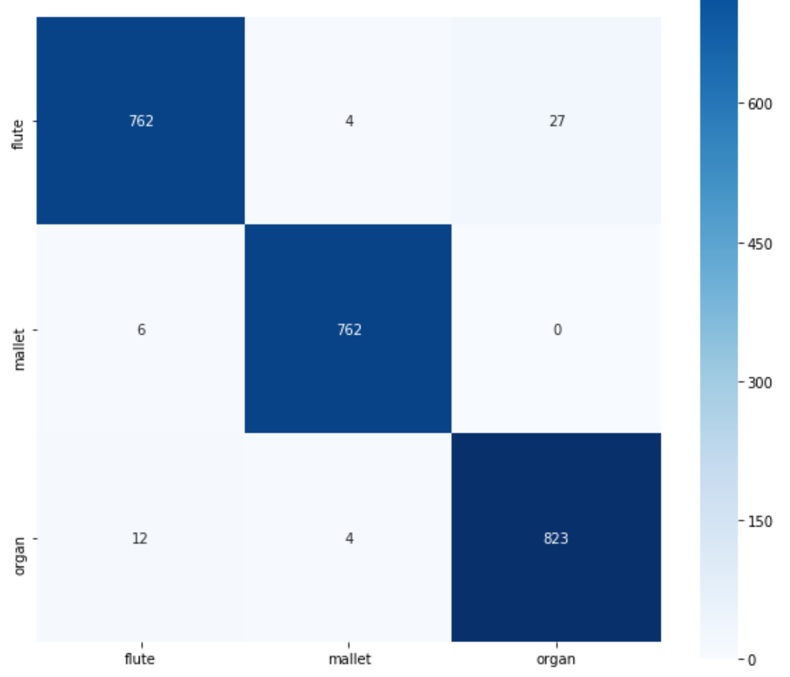

**3 instruments :** Flute, Mallet, Organ
**4000 échantillons** de chaque
**Précision :** 97.4%



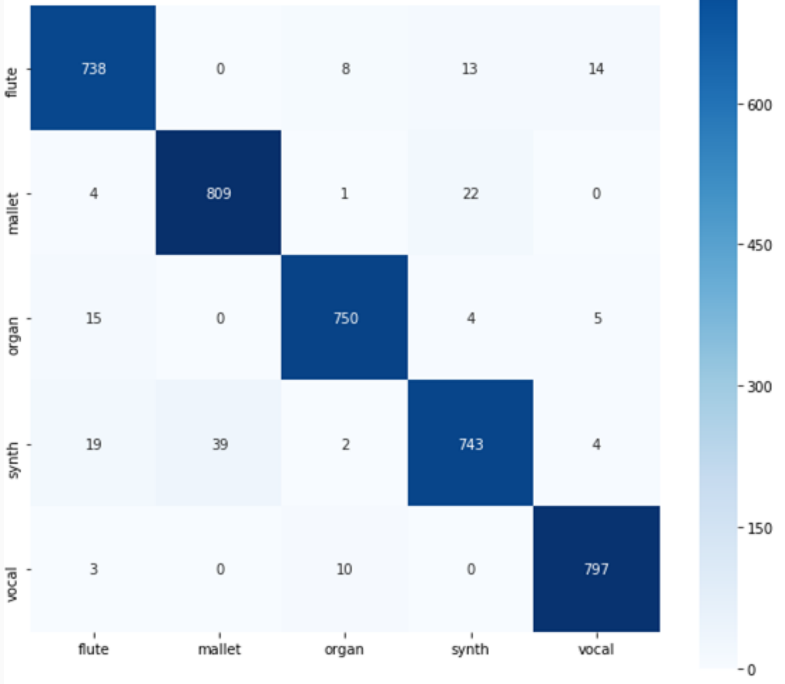
# 5. Spectrogrammes

## b. Avec **normalisation** des spectrogrammes - Decision Tree

**3 instruments :** Flute, Mallet, Organ
**4000 échantillons** de chaque
**Précision :** 97.87%



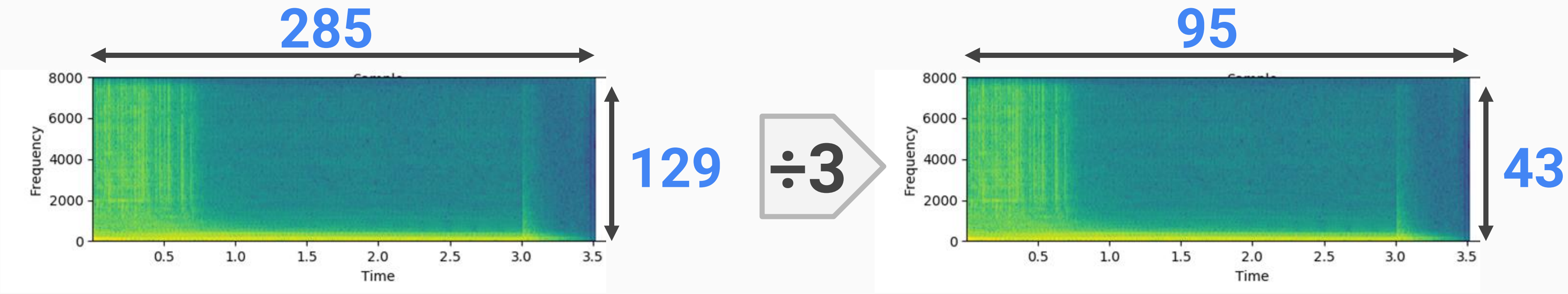

**5 instruments :** Flute, Mallet, Organ, Synth, Vocal
**4000 échantillons** de chaque
**Précision :** 95.92%



# 5. Spectrogrammes

**Pourquoi redimensionner les spectrogrammes ?**

**Taille d’origine : 129x285 = 36765 valeurs**
**Taille corrigée : 95x43 = 4085 valeurs**



**Pourquoi redimensionner les spectrogrammes ?**

**Taille d’origine : 129x285 = 36765 valeurs**
**Taille corrigée : 95x43 = 4085 valeurs**

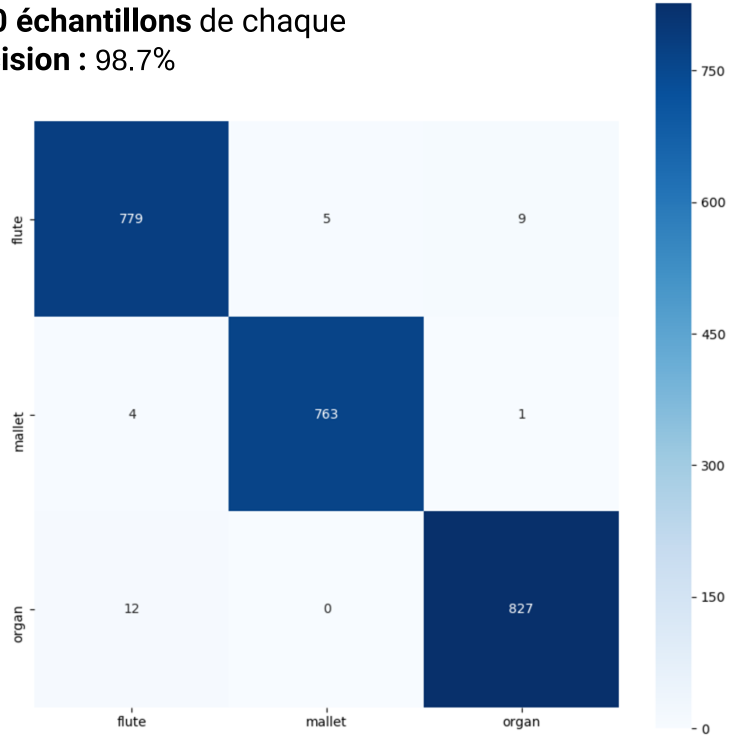
# 90%

**de valeurs en moins**

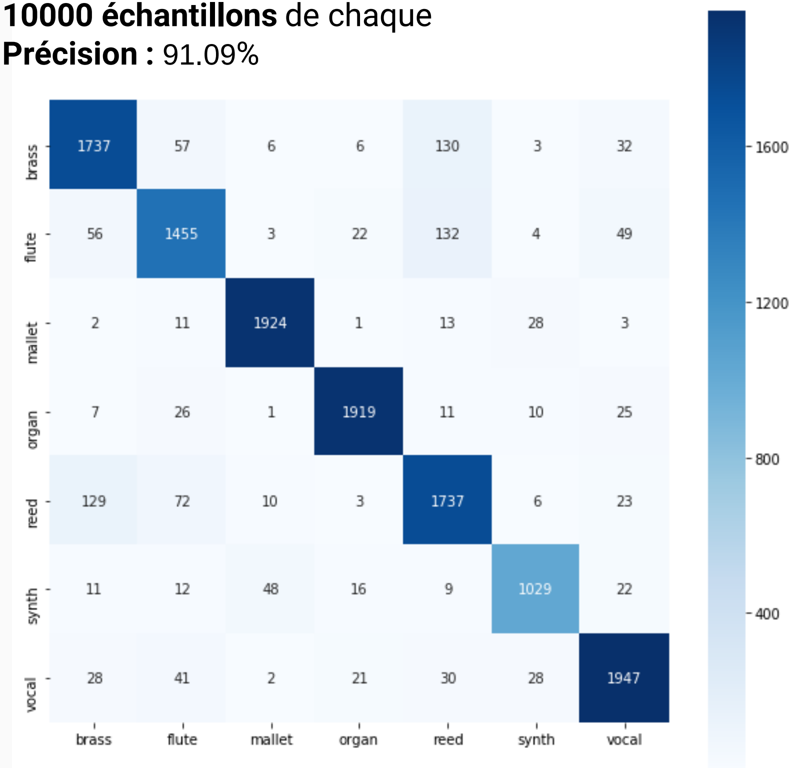
# 5. Spectrogrammes

## c. Ajout de la réduction de la résolution - Decision Tree

**3 instruments :** Flute, Mallet, Organ
**4000 échantillons** de chaque
**Précision :** 98.7%



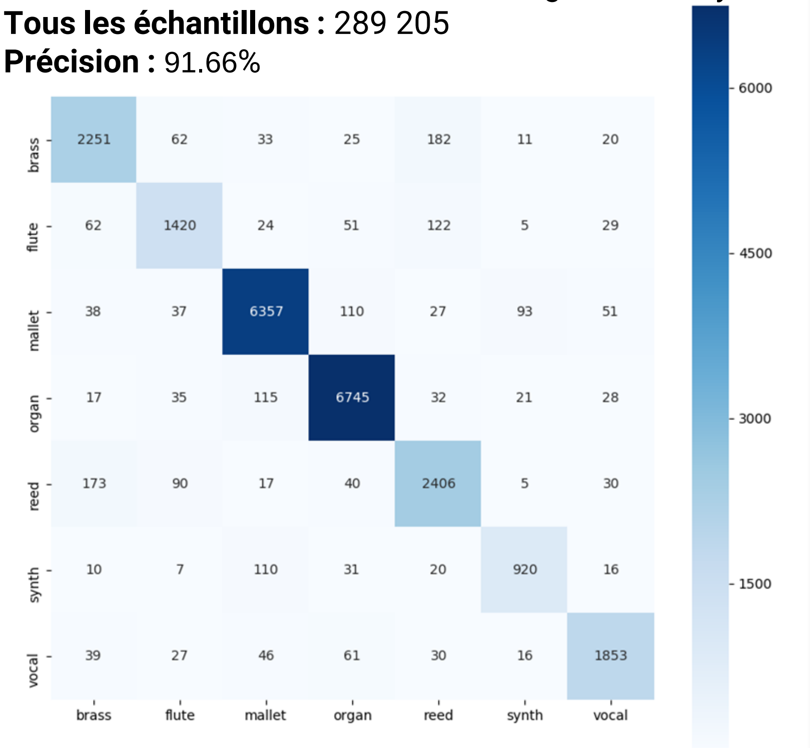

**7 instruments :** Brass, Flute, Mallet, Organ, Reed, Synth, Vocal
**10000 échantillons** de chaque
**Précision :** 91.09%



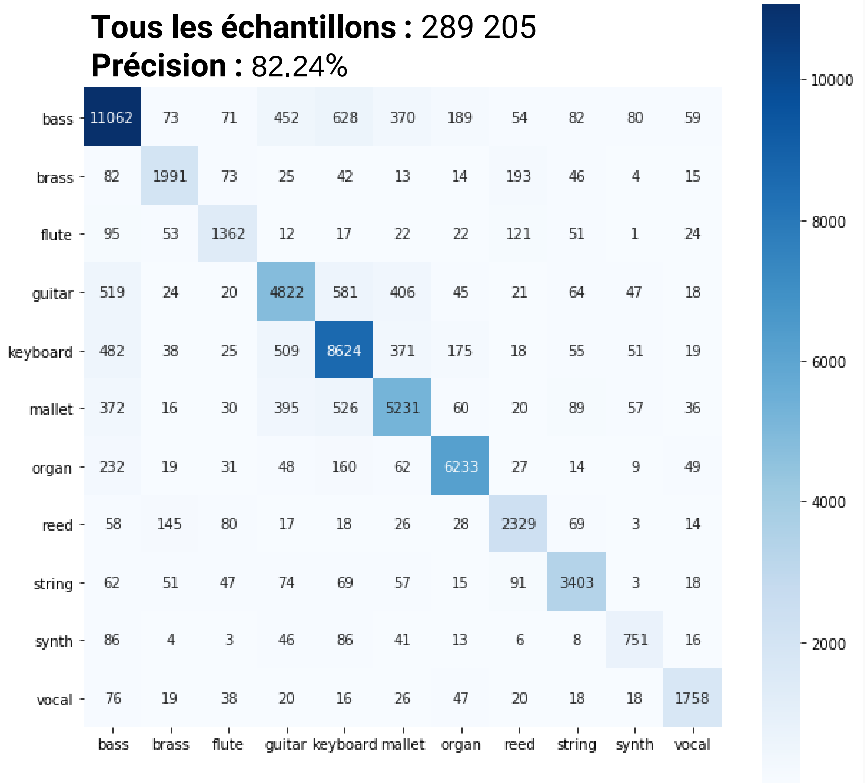
# 5. Spectrogrammes

## c. Ajout de la réduction de la résolution - Decision Tree

**7 instruments :** Brass, Flute, Mallet, Organ, Reed, Synth, Vocal
**Tous les échantillons :** 289 205
**Précision :** 91.66%




**Tous les instruments**
**Tous les échantillons :** 289 205
**Précision :** 82.24%



# 5. Spectrogrammes

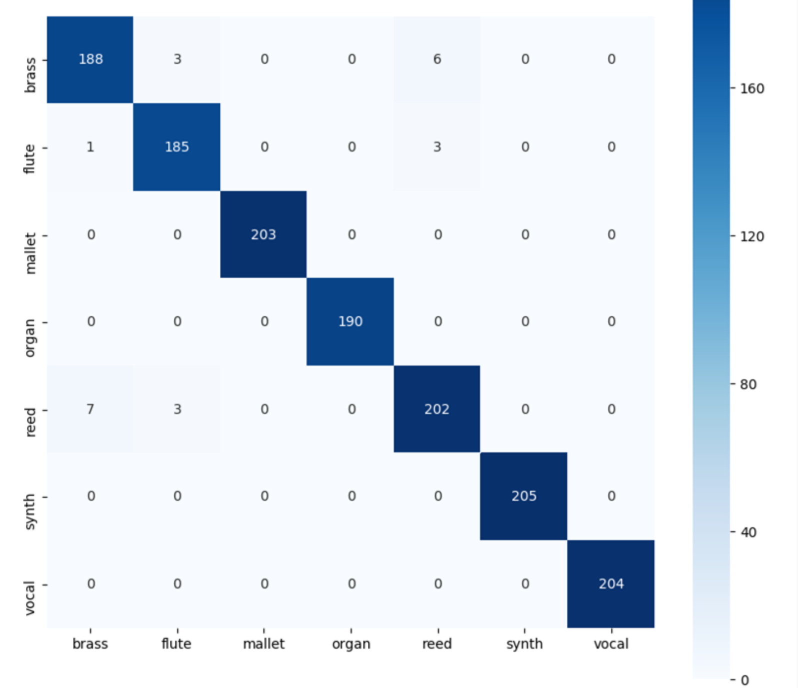
## c. Ajout de la réduction de la résolution - Random Forest

Decision Tree vs. Random Forest

**7 instruments :** Brass, Flute, Mallet, Organ, Reed, Synth, Vocal
**10000 échantillons** de chaque

**Précision :** 91.09%


**Précision :** 98.36%



# 5. Spectrogrammes

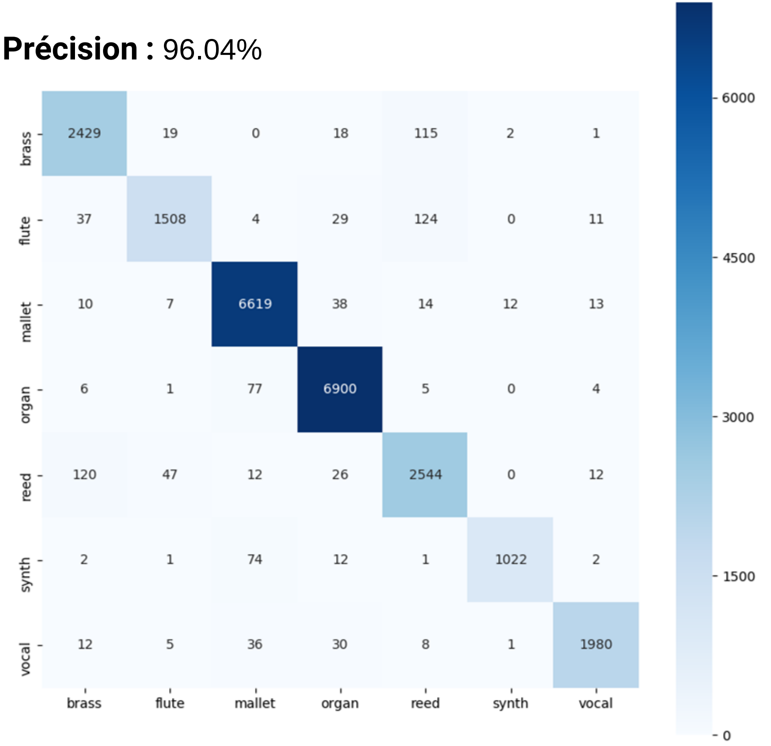
c. Ajout de la réduction de la résolution - Random Forest

## Decision Tree vs. Random Forest

**7 instruments :** Brass, Flute, Mallet, Organ, Reed, Synth, Vocal
**Tous les échantillons :** 289 205

**Précision :** 91.66%


**Précision :** 96.04%



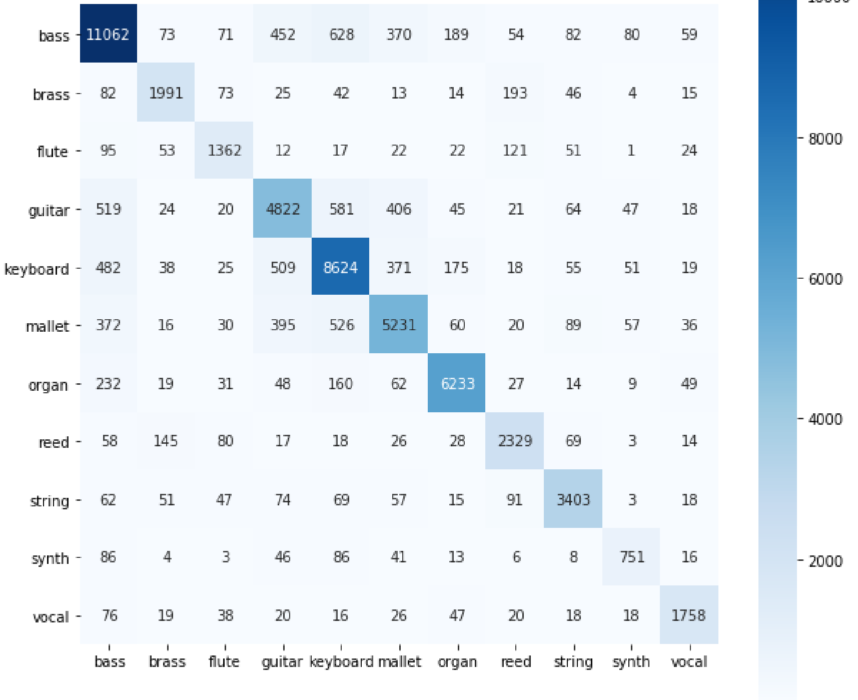
# 5. Spectrogrammes

## c. Ajout de la réduction de la résolution - Random Forest

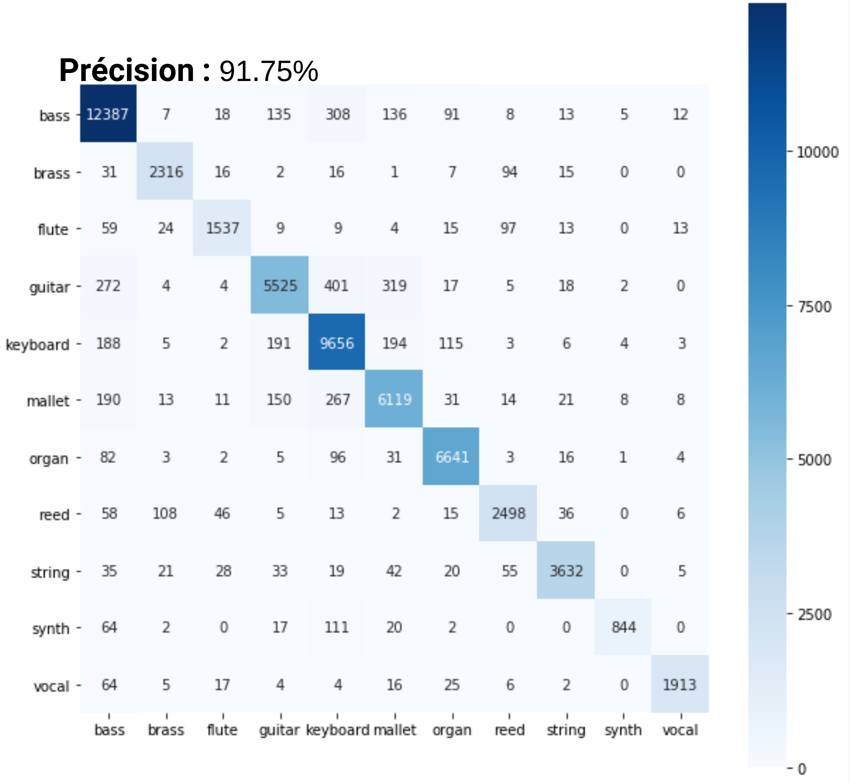
Decision Tree vs. Random Forest

**Tous les instruments**
**Tous les échantillons**

**Précision :** 82.24%




**Précision :** 91.75%



# 6. Conclusion

## Première approche - Transformée de Fourier :

- **Exploitation des données incomplète (manque l'intensité sur les fréquences sélectionnées)**
- **Notre algorithme actuel est très long pour extraire les fréquences (≈ 7h)**

## Seconde approche - Spectrogrammes :

- **Plus fiable : meilleure précision sur nos tests**
- **Rapidité, grâce à la normalisation et la réduction de la résolution. (90% économie)**

**Decision Tree : (82.24%)**

- **Bons résultats, mais faussés à cause de la faible quantitée de certains instruments.**

**Random Forest : (91.75%)**

- **Comble la faiblesse due aux écarts dans le nombre d'échantillons de chaque classe en faisant une moyenne à partir de nombreux tirages (bootstrap).**