CHORD DETECTION USING DEEP LEARNING

Samuel Guilluy
ENS Paris Saclay MVA
Audio Signal Processing
samuel.guilluy@ens-paris-saclay.fr

April 21, 2020

ABSTRACT

L'objectif de l'article [1] est de présenter une méthode de détection automatique d'accord basée sur un réseau convolutif. Ce réseau va nous permettre d'apprendre à détecter les accords à partir de features que l'on aura préalablement extrait de nos fichiers audio.

***K*eywords** Chord Detection · Deep learning · Convolutional Neural Networks

1 Introduction

Depuis Aristote avec la création des octaves dans son ouvrage "Notes sur les problèmes musicaux", la définition des hauteurs des notes ainsi que celle des accords a changé de nombreuses fois. La musique étant avant tout un art, la construction des accords passent d'abord par le ressenti des consonnances avant d'être formalisé mathématiquement.

De nos jours, il est possible à l'aide de cette analyse de reconnaitre automatiquement des accords. Cela a pour but de reconnaitre les accords joués à partir d'un fichier son composé de voix et de plusieurs instruments. Cette tâche est dite supervisée car elle nécessite dans un premier temps d'avoir un jeu de données annotés servant de référence à notre apprentissage.

La spécificité de cette exercice par rapport aux autres taches dont les meilleurs performances sont réalisées à partir d'algorithmes de Deep Learning (nottament : Speech Detection, Neural Machine Transcrition, Sentiment Analysis, ...) est le besoin de bien comprendre comment sont construit les accords afin d'adapter notre méthode aux contraintes liées à ce problème. De plus, il existe de nombreuses méthodes permettant l'extraction d'information à partir d'un fichier son, il sera donc nécessaire de les comparer afin de choisir celle qui convient le mieux à ce projet.

La première étape de ce projet est donc d'étudier comment fut construite la musique occidentale : du choix des différentes hauteurs à la construction des différents types d'accords.

Puis, nous allons donc étudier les différentes façon d'extraire des informations à partir de notre fichier audio afin de justifier notre choix.

Enfin, nous présenterons l'architecture de notre réseau convolutif : ses avantages et performance à travers de la réalisation d'expériences numériques.

2 Etude de la représentation des accords

2.1 Introduction à la musique occidentale

Tout son musical (ou note) possède une fréquence fondamentale (nombre de vibrations par seconde en hertz) correspondant à sa hauteur. Dans la musique occidentale, les catégories de hauteurs sont au nombre de douze. Sept d'entre elles sont considérées comme les principales et ont pour noms : do, ré, mi, fa, sol, la et si.

Deux notes dont les fréquences fondamentales ont un rapport qui est une puissance de deux (c'est-à-dire la moitié, le double, le quadruple...) donnent deux sons très similaires et portent le même nom. L'intervalle compris entre deux hauteurs dont la fréquence de l'une vaut le double (ou la moitié) de l'autre s'appelle une octave.

Cette première observation sur les notes nous permet de regrouper toutes les notes qui ont cette propriété dans la même catégorie de hauteur. Il existe ainsi plusieurs do mais à des octaves différentes.

Pour distinguer deux notes de même nom dans deux octaves différentes, on numérote les octaves et donne ce numéro aux notes correspondantes : par exemple, le la3 a une fréquence de 440 hertz dans la norme internationale. Le tableau 1 présente la fréquence des principales notes en fonction de leurs numérotation dans la gamme tempérée.

Fréquences des notes (en hertz) dans la gamme tempérée

Note/octave	-1	0	1	2	3	4	5	6	7	8	9
do ou *si*♯	16,35	32,70	65,41	130,81	261,63	523,25	1046,50	2093,00	4186,01	8 372,02	16 744,04
do♯ ou *ré*♭	17,33	34,65	69,30	138,59	277,18	554,37	1108,73	2217,46	4434,92	8 869,84	17 739,68
ré	18,36	36,71	73,42	146,83	293,66	587,33	1174,66	2349,32	4698,64	9 397,28	18 794,56
ré♯ ou *mi*♭	19,45	38,89	77,78	155,56	311,13	622,25	1244,51	2489,02	4978,03	9 956,06	19 912,12
mi ou *fa*♭	20,60	41,20	82,41	164,81	329,63	659,26	1318,51	2637,02	5274,04	10 548,08	21 096,16
fa ou *mi*♯	21,83	43,65	87,31	174,61	349,23	698,46	1396,91	2793,83	5587,65	11 175,30	22 350,60
fa♯ ou *sol*♭	23,13	46,25	92,50	185,00	369,99	739,99	1479,98	2959,96	5919,91	11 839,82	23 679,64
sol	24,50	49,00	98,00	196,00	392,00	783,99	1567,98	3135,96	6271,93	12 543,86	25 087,72
sol♯ ou *la*♭	25,96	51,91	103,83	207,65	415,30	830,61	1661,22	3322,44	6644,88	13 289,76	26 579,52
la	27,50	55,00	110,00	220,00	440,00	880,00	1760,00	3520,00	7040,00	14 080,00	28 160,00
la♯ ou *si*♭	29,14	58,27	116,54	233,08	466,16	932,33	1864,66	3729,31	7458,62	14 917,24	29 834,48
si ou *do*♭	30,87	61,74	123,47	246,94	493,88	987,77	1975,53	3951,07	7902,13	15 804,26	31 608,52

Figure 1: Fréquences des notes en Hertz dans la gamme tempérée (source : Wikipédia)

La description des consonances est d'abord arithmétique. Un son vibrant à la fréquence F va aussi vibrer à la fréquence 2 F et 3 F ce qui entraine l'apparition de nouvelle note : on appelle la note vibrant à la fréquence 2 F (rapport 2/1) la note à l'**octave** supérieur et celle vibrant à la fréquence de rapport 3/2 une notte sonnant la **quinte**. De même on introduit aussi le rapport 4/3 sonnant la **quarte**.

Le nombre de 12 hauteurs différentes vient alors naturellement du à la cyclicité des notes suivant la quinte car la 13^{eme} hauteur d'une quinte est la première note à l'octave supérieur. Cela est du à la relation : $3^{12} \simeq 2^{19}$.

2.2 La construction des accords

Un accord est une combinaison de plusieurs notes jouées simultanément. Un accord est construit autour d'une note principale nommée racine de l'accord. Les accords ne sont pas des combinaisons d'harmoniques, mais des superpositions de notes, et chacune de celles-ci s'accompagne de ses propres harmoniques. Les consonances et les accords s'expliquent donc comme des concordances de plusieurs séries harmoniques entre elles.

L'objectif de la construction des accords a été de créer des combinaison de trois sons qui soient entièrement consonante (perçue comme agréable à l'oreille).

Comme une note de fréquence F sonne bien avec une sonna t à 5F et celle de 5F avec celle à 5/2 F, on introduit un nouveau rapport entre 2 notes : la **tierce majeur** de rapport 5/2. On s'aperçoit alors que cela correspond presque à la 5^{me} quinte car $5 \simeq (\frac{3}{2})^4$. De même, la tierce mineur correspond au rapport 6/5. En ajustant légèrement l'écart entre les quintes, ont parvient à avoir une meilleure approximation de la tierce majeur par une succesion de quinte 2 qui nous permet ensuite de construire des accords de tierce plus consonnant (accordage par tempérament égal).

Selon cette accordage, il existe différents types d'accords possibles dont les plus connues à 3 notes consonnant sont :

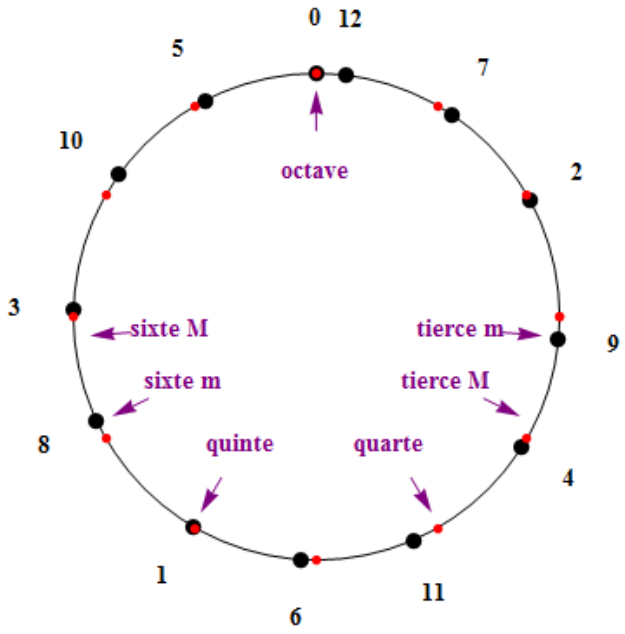

Figure 2: Quasi-cyclicité des quintes (source : `https://www.physinfo.org/chroniques/arithmetique.html`)

- Superposer une tierce majeure entre les notes 1 et 2 et une tierce mineure entre les notes 2 et 3 ; la distance entre les notes 1 et 3 est d'une quinte le résultat est l'accord parfait **majeur**, comme do-mi-sol ;
- Procéder de même mais en inversant l'ordre des tierces: mineure entre 1 et 2, majeure entre 2 et 3, produisant toujours une quinte entre 1 et 3 – le résultat est l'accord parfait **mineur**, comme do-mi bémol-sol ;
- Superposer une tierce majeure et une quarte la distance entre les notes 1 et 3 est une sixte majeure – l'accord qui en résulte s'appelle logiquement accord de **sixte majeure**, comme do-mi-la ;
- Procéder de même avec une tierce mineure ; la distance entre les notes 1 et 3 est une sixte mineure – l'accord est celui de **sixte mineure**, comme do-mi bémol-la bémol

On a donc construit un ensemble d'accord dont les noms vont être basés sur la note racine de l'accord. L'ensemble des définitions des composants des différents types d'accord se trouve dans le tableau 3.

2.3 Présentation des annotations de notre base de données

Le site `http://isophonics.net/` propose une base de donnée publique d'annotation de musique afin d'aider la recherche.

Les annotations fournisent par le site ont été réalisées sur la base des travaux de thèse de Chris Harte [3] ainsi que par l'équipe du "center of digital music" de Queen Mary [2], University of London qui propose de nombreux articles de recherche sur le sujet sur son site : `http://c4dm.eecs.qmul.ac.uk/`

La liste des albums composants notre base de données est :

- Carole King: Tapestry, Ode Records, 4931802003
- Queen: Greatest Hits I, Parlophone, 0777 7 8950424
- Queen: Greatest Hits II, Parlophone, CDP 7979712
- Queen: Greatest Hits III, Parlophone, 7243 52389421
- The Beatles: Please Please Me, CDP 7 46435 2
- The Beatles: With the Beatles, CDP 7 46436 2
- The Beatles: A Hard Day's Night, CDP 7 46437 2
- The Beatles: Beatles For Sale, CDP 7 46438 2

Table 2: Shorthand definitions for common chords

Chord Type		Shorthand Notation	Components List
Triad Chords:			
	Major	maj	(3,5)
	Minor	min	(b3,5)
	Diminished	dim	(b3,b5)
	Augmented	aug	(3,#5)
Seventh Chords:			
	Major Seventh	maj7	(3,5,7)
	Minor Seventh	min7	(b3,5,b7)
	Seventh	7	(3,5,b7)
	Diminished Seventh	dim7	(b3,b5,bb7)
	Half Diminished Seventh	hdim7	(b3,b5,b7)
	Minor (Major Seventh)	minmaj7	(b3,5,7)
Sixth Chords:			
	Major Sixth	maj6	(3,5,6)
	Minor Sixth	min6	(b3,5,6)
Extended Chords:			
	Ninth	9	(3,5,b7,9)
	Major Ninth	maj9	(3,5,7,9)
	Minor Ninth	min9	(b3,5,b7,9)
Suspended Chords:			
	Suspended 4th	sus4	(4,5)

Figure 3: Lien entre le nom des accords et ses composants (source : [2])

- The Beatles: Help!, CDP 7 46439 2
- The Beatles: Rubber Soul, CDP 7 46440 2
- The Beatles: Revolver, CDP 7 46441 2
- The Beatles: Sgt. Pepper's Lonely Hearts Club Band, CDP 7 46442 2
- The Beatles: Magical Mystery Tour, CDP 7 48062 2
- The Beatles: The Beatles (the white album), CDS 7 46443 8
- The Beatles: Abbey Road, CDP 7 46446 2
- The Beatles: Let It Be, CDP 7 46447 2
- Zweieck: Zwielicht (contact for info or to purchase)

L'objectif de la préparation des labels est de créer pour chaque frame un array binaire ayant une valeur 1 à l'indice de la note si elle compose l'accord et 0 aux autres indices.

Pour cela il faut parvenir à décomposer les accords selon leurs notes et ensuite ordonnées les notes selon leurs fréquences. La librairie Python Pychord permet d'ordonnéer les notes en leurs attribuants un entier entre 0 à 11 (respectant l'ordre des hauteurs : do-ré-mi-fa-sol-la-si). Cette librairie permet aussi de donner la composition des principaux accords : min, maj,7. Pour les autres, il faudra utiliser le tableau 3.

L'article [1] se limite aux accords majeur et mineur triade en négligeant les 7 et classe les autres accords dans une catégories N "not found". Dans le cadre de notre étude, nous allons tenter d'élargir les accords considérés en conservant tous les accords présents dans la liste 4 afin de diminuer le nombre d'accord inconnu.

L'analyse de notre base de données nous permet déjà d'identifier les inégalités du nombre d'apparition des différents accords. En effets, les accords de la triade : do,ré,mi,fa,sol,la,si,do apparaissent pour certains plus de 1000 fois tandis que d'autres accords comme Ab:maj9 apparaissent qu'une seule fois. La tableau 5 illustre cela en représentant le nombre d'accord en fonction du nombre de fois où on les trouve dans la base de données. Il y a notamment 140 accords qui apparaissent que entre 0 et 50 fois.

```
dict_keys(['N', 'D:min', 'A', 'G:7', 'B:min', 'G', 'F', 'Eb', 'C', 'E:min', 'C:7', 'D', 'G:sus4', 'A:min', 'Ab:aug', 'D:7', 'C
#:min', 'F#:min', 'B', 'E:7', 'D:maj', 'A:7', 'F#', 'E:aug', 'E', 'F:7', 'Bb:7', 'B:9', 'D#', 'E:9', 'A:aug', 'D:min7', 'G#:au
g', 'Bb', ':', 'D:maj6', 'A:maj', 'E:sus4', 'A:min7', 'D#:hdim7', 'G#', 'A:9', 'D:dim', 'G#:7', 'D:dim7', 'D:min9', 'C:sus4',
'F:maj7', 'B:dim', 'Gb', 'Ab', 'E:maj6', 'F#:min7', 'B:maj6', 'G:min7', 'G#:min7', 'G:9', 'C:maj7', 'E:min7', 'C#', 'D#:dim',
'C:maj6', 'B:sus2', 'E:dim', 'F:maj9', 'B:7', 'Eb:min', 'Db', 'Bb:min', 'F:dim', 'G:min', 'D:9', 'G#:min', 'B:aug', 'A:sus4',
'C:9', 'C:maj9', 'F:min', 'F#:sus4', 'D#:min', 'D#:7', 'C#:7', 'G:maj6', 'D:sus4', 'C:min7', 'C:min9', 'Eb:maj7', 'Ab:maj7',
'F:min7', 'F:6', 'F:9', 'Bb:maj7', 'D:maj9', 'G:maj7', 'D:6', 'G:6', 'D:maj7', 'A:6', 'E:sus', 'E:7sus', 'B:min7', 'B:7sus',
'C:min', 'Eb:7', 'E:4', 'Bb:dim', 'F#:dim', 'Db:6', 'Eb:9', 'G:dim', 'F#:7', 'A:min6', 'F#min7', 'C#:min7', 'Bb:min7', 'Ab:6',
'Ab:7', 'Gb:7', 'Db:maj6', 'Db:maj7', 'Bb:maj6', 'Bb:maj', 'Eb:maj', 'C#:dim', 'F:maj', 'Ab:min', 'Ab:maj', 'Eb:dim', 'Db:maj',
'A:dim', 'E:maj', 'B:maj', 'Gb:maj', 'Bb:9', 'Ab:dim', 'C:maj', 'G:maj', 'Bb:aug', 'D:aug', 'F#:maj', 'Bb:sus4', 'Db:maj9', 'D
b:7', 'Gb:min', 'Cb', 'G:min9', 'E:min9', 'Ab:maj6', 'G:maj9', 'F#:hdim7', 'E:min6', 'D#:aug', 'A:maj6', 'A:sus2', 'F#:min9',
'G:aug', 'Eb:min7', 'F:maj6', 'G#:dim7', 'A:dim7', 'B:hdim7', 'F#:aug', 'D#:dim7', 'G#:hdim7', 'E:maj7', 'F#:9', 'A:maj7', 'C:m
in6', 'F#:minmaj7', 'B:maj7', 'Db:min', 'Ab:min7', 'F:min6', 'C#:maj7', 'Eb:maj6', 'B:dim7', 'B:sus4', 'Eb:sus4', 'C#:sus4',
'F:aug', 'C:aug', 'E:dim7', 'G:minmaj7', 'Bb:dim7', 'C:dim7', 'Bb:sus2', 'Ab:maj9', 'Eb:aug', 'A:min9', 'F#:min6', 'E:sus2', 'C
#:hdim7', 'G:dim7', 'Db:dim', 'Gb:maj7', 'D:min6'])
```

Figure 4: Liste des accords présents dans nos fichiers test

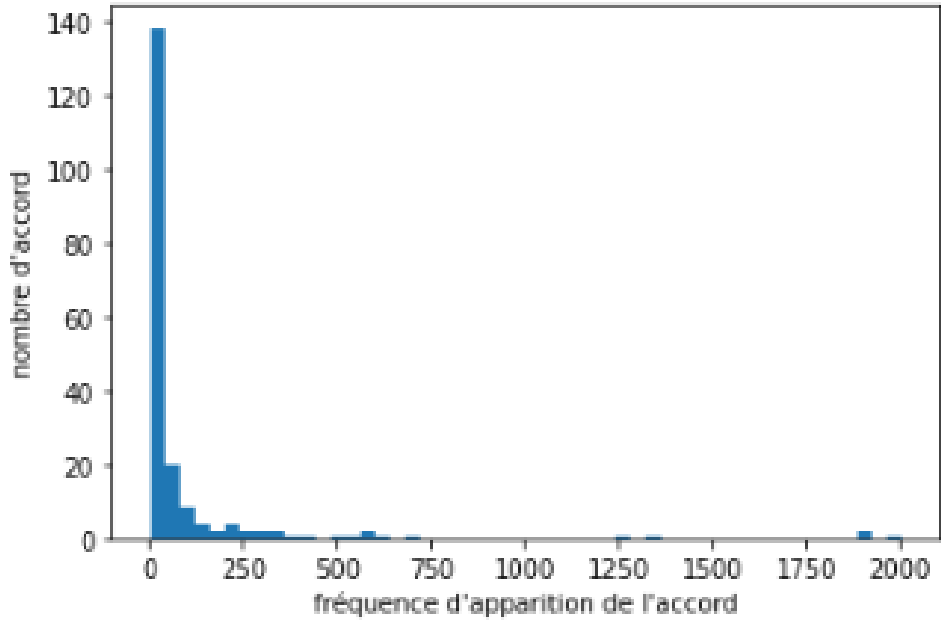

Figure 5: fréquence d'apparition des accords

3 Présentation des méthodes d'extraction de features.

Il existe différentes méthodes d'extraction de features à partir un fichier son.

3.1 Transformée de Fourier à fenêtre glissante

Cette méthode basée sur la transformée de Fourier de notre signal nous permet d'extraire des informations relatives à la phase et l'amplitude du signal étudié.

Cette méthode consiste a découpé le signal en segment plus court de longueur égale et d'appliquer à chacun une transformée de Fourier on a ainsi le spectre de fourier de chaque segment.

3.2 Constant Q Transform

Cette transformation est une série de filtre f_k espacé en fréquence de manière logarithmique. La bande passe du filtre d'indice k est un multiple de la bande passante de celle du filtre précédent. Mathématiquement, cela revient à :

$$\delta f_k = 2^{1/n} \delta f_{k-1} \tag{1}$$

où n est le nombre de filtre par octave.

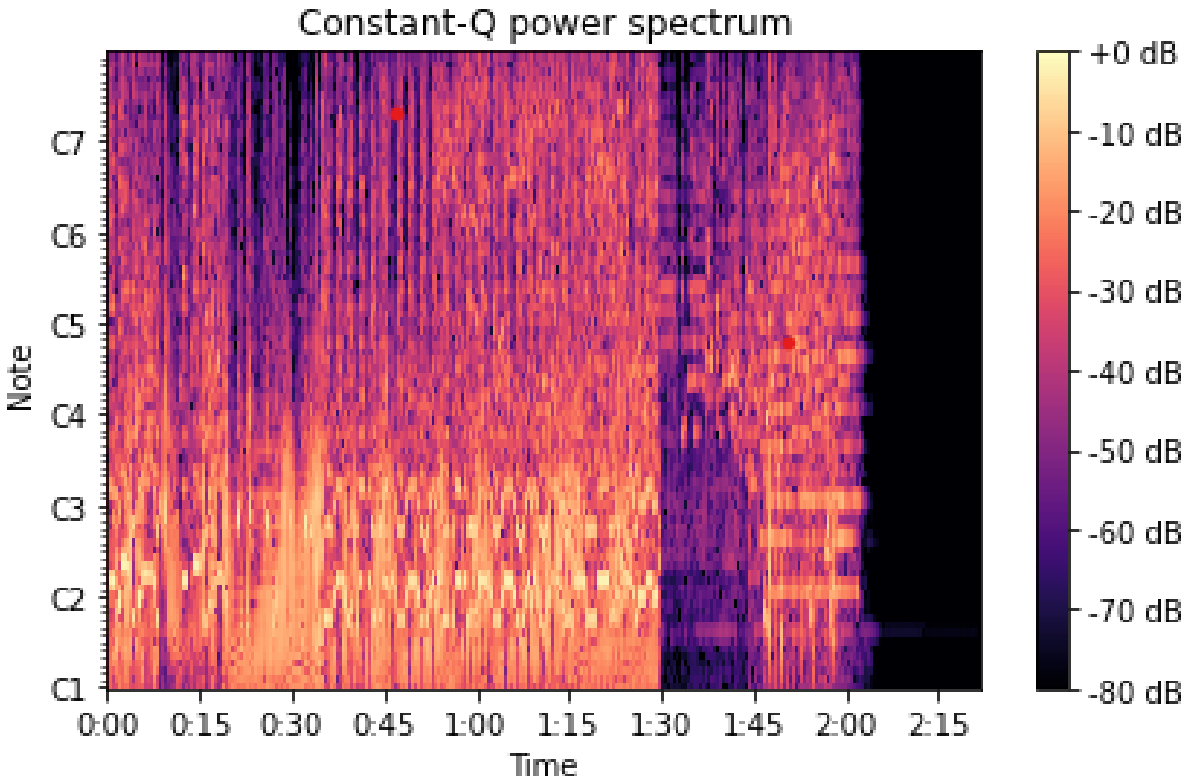

Figure 6: Représentation de la transformation CQT

Cette transformation est plus adaptée à l'étude de données musical que la transformée de Fourier à fenêtre glissante pour plusieurs raisons :

- Comme le résultat de la CQT est une représentation amplitude en fonction de la log fréquence, cette méthode nécessite moins de fenêtre de fréquence afin de couvrir une certaine taille de fréquence.
- Cette transformation présente une réduction de la résolution pour les hautes fréquences, cela est adaptés à l'oreille humaine et à l'étude de la musique.
- Les harmoniques des notes de musiques formant un motif, il est possible d'identifier à travers cela l'instrument joué.

3.3 Représentation chromatique

L'objectif principale de ces méthodes est d'associer un son à sa hauteur dans la gamme tempérée. Le principe est représenté dans la figure 7 montrant en (b) l'association théorique entre note et le spectre et dans (d) les résultats expérimentaux. Il existe différentes méthodes afin d'obtenir une représentation chromatique d'un fichier audio. L'article [4] présente plusieurs méthodes dont une qui est une variante de la Constant-Q transform présentée précédement consistant à ajuster les fréquences bins aux classes des 12 hauteurs définissant la gamme tempérée. Une autre méthode peut consisté à utiliser la représentation MFCC du signal et l'adapté pour correspondre aux hauteurs souhaitées.

Une amélioration du chromagramme peut être effectuée si l'on synchronise la représentation avec le tempo de la musique. La figure 8 présente l'amélioration effectuée par synchronisation avec le tempo de la musique.

Beaucoup de méthodes de détection d'accords ont utilisées la représention chromatique, combinée à l'utilisation d'un modèle de Markov Caché. Un des objectifs de l'article étudié est d'évaluer si un réseau de neuronnes peut apprendre une représentation plus efficaces de nos données afin de mieux identifier les accords.

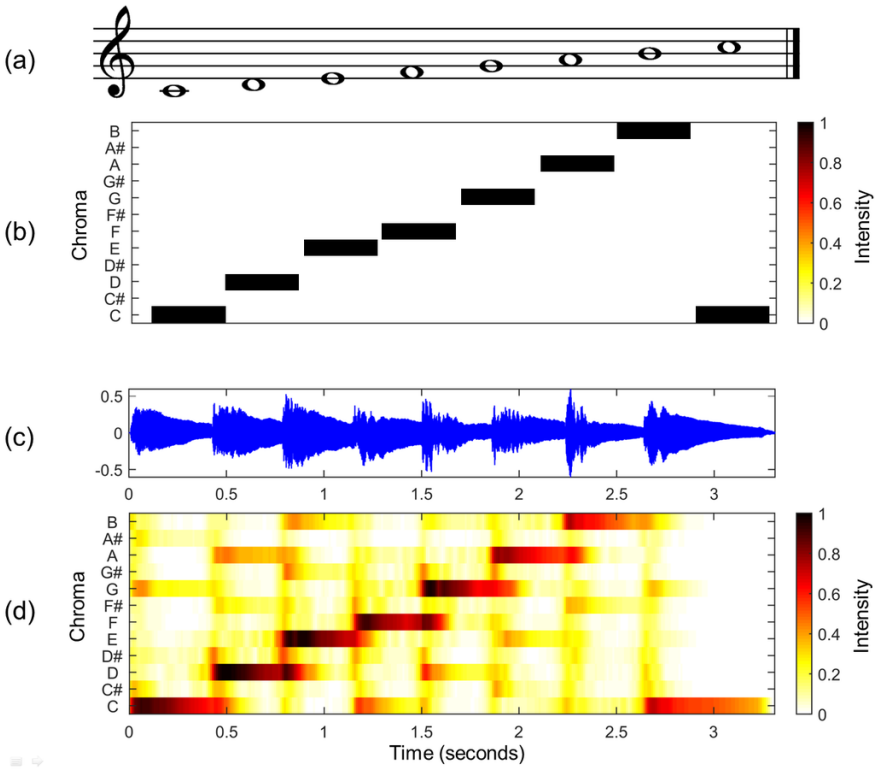

Figure 7: (a) Note de musique de la clé de Do (b) Chromagramme obtenu à partir du score (c) Enregistrement audio de la gamme de Do au piano (d) Chromagramme obtenu à partir de l'enregistrement audio (source : Wikipédia)

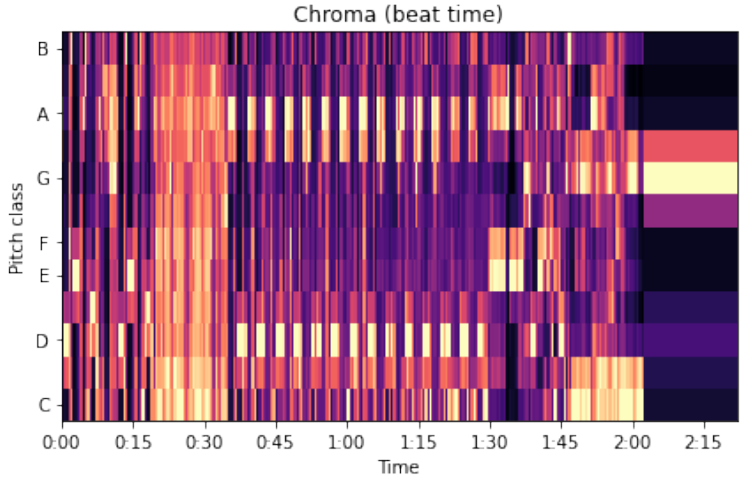

Figure 8: Chromagramme avec synchronisation du tempo

4 Présentation de l'architecture de la solution

Dans cette partie, nous allons présenter l'architecture de la solution étudiée puis les résultats expérimentaux associés.

4.1 Préparation des données

Comme indiqué dans la partie 2, les labels proviennent du site isophonics et des travaux de Chris Harte. Cependant, les fichiers son associés étant sous copyright, il est nécessaire d'obtenir les données par un autre moyen. J'ai opté pour

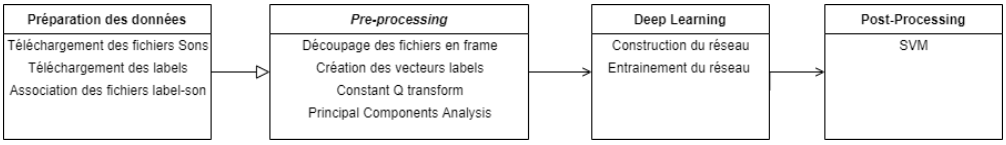

Figure 9: Architecture de la solution

le téléchargement des fichiers à partir du logiciel NoteBurner qui permet de télécharger les 3 premières minutes des musiques présentent sur Spotify lorsque l'on a un compte payant.

Cette méthode nous oblige a vérifier que l'on a la bonne version de l'album qui a été plusieurs fois ré-édité. Il suffit ensuite d'adapter le nom des différents fichiers afin d'associer les musiques aux labels.

4.2 Pre-processing

Les différentes étapes du pré-processing consistent à :

- Découper les fichiers en frame de taille égale
- Time Splicing : méthode consistant à considérer en entrée la frame précédente et suivante à celle en cours.
- Constant Q transform : présenté en partie 3
- Analyse en Composante Principale : méthode de réduction de dimension des données dont l'objectif est de conserver uniquement les composantes de la série temporelle dans les directions conservant au mieux la covariance de la série.

4.3 Modèle de réseau de neuronnes convolutif

L'avantage des réseaux convolutifs est l'utilisation d'un poids unique associé aux signaux entrant dans tous les neurones d'un même noyau de convolution. Cette méthode permet une invariance du traitement par translation.

De plus, l'auteur a choisi d'utiliser une architecture du réseau sous forme de "bottleneck" (voir figure 10) afin de réduire encore une fois la dimension de notre jeu de données et ainsi éviter l'overfitting.

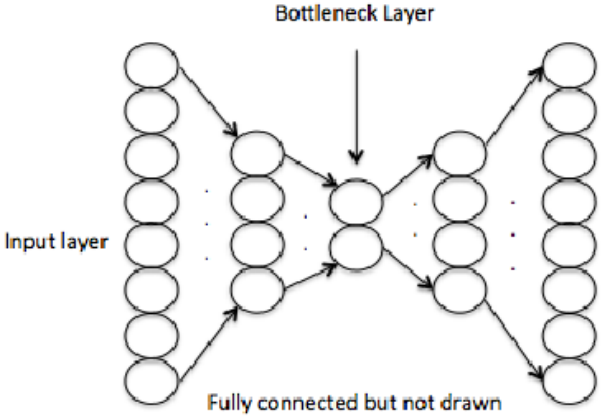

Figure 10: Architecture CNN de type "bottleneck " (source [1])

4.4 Résultats expérimentaux

La structure du réseaux convolutif employé est résenté en figure 11 Afin d'entrainer ce modèle, il a été nécessaire de choisir un critère capable de prendre en compte les multi-label.

Le critère BCE With Logits Loss de Pytorch permet d'ajouter à notre modèle une fonction Sigmoid layer ainsi qu'un critère Binary Cross Entropy adapté à nos données.

```
Net(
  (conv1): Conv2d(77, 20, kernel_size=(1, 1), stride=(1, 1))
  (conv2): Conv2d(20, 10, kernel_size=(1, 1), stride=(1, 1))
  (conv3): Conv2d(10, 20, kernel_size=(1, 1), stride=(1, 1))
  (conv4): Conv2d(20, 50, kernel_size=(1, 1), stride=(1, 1))
  (fc): Linear(in_features=500, out_features=12, bias=True)
)
```

Figure 11: Composants du réseau convolutif utilisé

L'article [1] suggère qu'il est plus difficile de procéder à une classification multi-label, nous arrivons à une loss moyenne après 10 epochs de 0.53, (somme de la loss BCE sur les 12 composantes) et qui décroit que très peu au cours de l'apprentissage ce qui suggère un modèle peu efficace.

Le code utilisé afin d'obtenir ces résultats est disponible sur github au lien suivant : `https://github.com/SamuelGuilluy/Audio-Signal-Processing`.

5 Conclusion

Nous avons à travers cette étude une méthode utilisant un réseau de neurones convolutifs associé à une CQT afin de détecter automatiquement les accords d'un fichier son.

Cette étude nous a obligé dans un premier temps à étudier la construction des accords et à comprendre la difficulté d'effectuer une classification des accords.

Les résultats d'une classification multi-label sont inférieurs à ceux d'une classification plus simple ne considérant pas toute la richesse de la construction des accords.

De plus, ce modèle datant de 2016 pourrais être améliorer à l'aide de progret récent en réseau de neuronnes.

Pour conclure, voici les trois axes qu'il semblerait être intéressant d'approfondir pour aller plus loin :

- Utiliser une base de donnée plus homogène du point de vue fréquence d'apparition des accords et style de musique : rock / classique.
- L'utilisation d'un modèle language combiné à un model accoustic [5] permet de prendre en compte les probabilités conditionelles d'apparitions succesives des accords au sein d'une même musique.
- L'utilisation d'un modèle de réseaux de neurones utilisant le mécanisme d'attention comme utilisé au sein du Transformer [6] pour les cas d'usage NLP permet de prendre en compte différents aspects

References

[1] Xinquan Zhou and Alexander Lerch. CHORD DETECTION USING DEEP LEARNING. page 7, 2015.

[2] Christopher Harte, Mark Sandler, Samer Abdallah, and Emilia Gomez. SYMBOLIC REPRESENTATION OF MUSICAL CHORDS: A PROPOSED SYNTAX FOR TEXT ANNOTATIONS. page 6.

[3] Christopher Harte. Towards Automatic Extraction of Harmony Information from Music Signals. page 283.

[4] Sebastian Ewert. Computer Science III University of Bonn. *Poster Session*, page 6, 2011.

[5] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. An End-to-End Neural Network for Polyphonic Piano Music Transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):927–939, May 2016.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. arXiv: 1810.04805.