Université Mohammed V - Rabat
École Nationale Supérieure d'Informatique et d'Analyse des Systèmes

Rapport du Projet du Traitement De l audio

Filière

Ingénierie de l'Intelligence Artificielle (2IA)

Sujet :

Séparation et Classification des Sources pour Instruments Musicaux dans un Chant Musical

---

Réalisé par :

MOUNIR LAMSAYAH

ZIYAD ABIDATE

HAMZA BENATHMANE

Encadré par :

Mme. Sanaa EL FKIHI

Année Universitaire 2024-2025

Table des matières

Chapitre 1

Séparation des Sources Audio par STFT, NMF et Clustering

1.1 Introduction

La séparation des sources audio est une tâche complexe dans le domaine du traitement du signal. Elle consiste à extraire des composantes sonores individuelles, comme des instruments de musique ou des voix, à partir d'un mélange audio unique. Cette problématique est courante dans des domaines tels que le karaoké, la restauration d'enregistrements anciens, et l'analyse de performances musicales.

Lorsqu'un signal audio est enregistré, il est souvent composé d'un mélange de plusieurs sources sonores superposées. Le défi est donc de pouvoir identifier et isoler ces sources distinctes de manière automatique. Cette tâche est difficile car les fréquences peuvent se chevaucher et les harmoniques des instruments peuvent interférer les unes avec les autres.

Ce rapport explore plusieurs techniques de séparation des sources audio :

— La Transformée de Fourier à Court Terme (STFT), utilisée pour représenter le signal dans le domaine temps-fréquence.

— La Décomposition en Matrice Non-Négative (NMF), qui permet de factoriser la matrice spectrale en composantes plus interprétables.

— Des méthodes de clustering appliquées aux matrices factorielles pour regrouper les éléments sonores similaires.

Les méthodes présentées ici offrent une approche combinée exploitant les représentations spectrales et des techniques d'apprentissage non supervisé pour améliorer la qualité de la séparation.

1.2 Transformée de Fourier à Court Terme (STFT)

La transformée de Fourier à court terme (STFT) est utilisée pour analyser un signal dans le domaine temps-fréquence. Elle permet de représenter l'évolution spectrale d'un signal audio sur des fenêtres de temps successives. Mathématiquement, la STFT d'un signal discret $y(n)$ est définie par :

$$\text{STFT}(y(t), \tau, f) = \sum_{n=-\infty}^{\infty} y(n) w(n - \tau) e^{-j2\pi f n}$$

où $w(n)$ est une fenêtre d'analyse centrée autour de l'instant τ et f représente la fréquence.

Cette transformation permet de générer un spectrogramme, qui est une représentation visuelle de l'énergie du signal dans le domaine temps-fréquence. Le spectrogramme est obtenu en prenant la magnitude du résultat de la STFT :

$$S(\tau, f) = |\text{STFT}(y(t), \tau, f)|^2$$

Avantages de la STFT :

— Représentation temps-fréquence permettant de suivre l'évolution des composantes spectrales au cours du temps.
— Identification des fréquences dominantes dans un signal audio.
— Préparation de données structurées pour l'application de techniques de séparation comme NMF et clustering.

Le code correspondant utilise la bibliothèque Python `librosa` pour effectuer la STFT avec des fenêtres de type Hamming et une taille de fenêtre de 1024 échantillons.

1.3 Décomposition en Matrice Non-Négative (NMF)

La Décomposition en Matrice Non-Négative (NMF) est une méthode de factorisation qui permet de décomposer une matrice V en deux matrices W et H, de manière à ce que toutes les valeurs des matrices résultantes soient non négatives. Cette approche est particulièrement utile pour la séparation de sources dans les signaux audio, où V est souvent une matrice représentant la magnitude d'un spectrogramme, avec des composantes temporelles et fréquentielles.

Mathématiquement, la NMF cherche à factoriser une matrice V de taille $F \times T$ en deux matrices W et H de dimensions respectives $F \times K$ et $K \times T$, de manière approximative :

$$V \approx WH$$

où :

— $V \in \mathbb{R}^{F \times T}$ est la matrice d'entrée représentant la magnitude du spectrogramme, avec F fréquences et T instants de temps.

— $W \in \mathbb{R}^{F \times K}$ est la matrice des bases fréquentielles, où K est le nombre de bases (ou composants) extraites.

— $H \in \mathbb{R}^{K \times T}$ est la matrice des activations temporelles, qui représente l'intensité de chaque base au fil du temps.

L'objectif de la NMF est de minimiser l'erreur de reconstruction entre V et WH tout en respectant la contrainte de non-négativité, c'est-à-dire que toutes les valeurs dans W et H doivent être positives ou nulles.

La formulation mathématique de l'objectif de la NMF peut être décrite comme un problème d'optimisation :

$$\min_{W,H} \|V - WH\|_F^2 \quad \text{sous les contraintes} \quad W \geq 0, H \geq 0$$

où $\|\cdot\|_F$ désigne la norme Frobenius, qui est définie par :

$$\|A\|_F = \sqrt{\sum_{i,j} |a_{ij}|^2}$$

Cette minimisation est réalisée en utilisant des algorithmes itératifs tels que la mise à jour alternée, où W et H sont mis à jour de manière itérative pour réduire l'erreur de reconstruction tout en maintenant les contraintes de non-négativité. Les mises à jour peuvent être effectuées en utilisant des règles de mise à jour basées sur des dérivées partielles ou des méthodes de type gradient.

Les mises à jour peuvent être exprimées de la manière suivante :

$$H \leftarrow H \circ \frac{W^T V}{W^T W H} \quad \text{et} \quad W \leftarrow W \circ \frac{V H^T}{W H H^T}$$

où $\circ$ désigne la multiplication élément par élément (Hadamard product). Ces mises à jour garantissent que les matrices W et H restent non négatives à chaque itération.

L'ingéniosité de la NMF dans ce contexte est qu'elle décompose le signal en motifs temporels et fréquentiels indépendants. Elle est particulièrement adaptée pour la séparation audio car elle extrait des motifs répétitifs correspondant souvent à des instruments distincts ou des sources sonores séparées.

Le code Python correspondant utilise la fonction `sklearn.decomposition.NMF` avec un paramètre K défini dynamiquement selon le chant ou le signal traité.

1.4 Clustering sur les Matrices Facteur (W et H)

Après la factorisation NMF, un **clustering** est effectué sur les colonnes de la matrice W. Chaque colonne de W représente un motif spectral indépendant. La matrice W est définie comme suit :

$$W = [w_1, w_2, \ldots, w_K]$$

où w_k représente le k-ième motif spectral dans l'espace des fréquences et K est le nombre de motifs extraits. Le paramètre K a été ajusté en fonction de la complexité spectrale du chant ou du signal traité.

Le clustering est effectué pour regrouper des motifs similaires, ce qui permet de mieux séparer les sources. En effet, les motifs spectraux associés à une même source ont tendance à être proches les uns des autres dans l'espace des caractéristiques. L'algorithme K-means est utilisé pour effectuer ce regroupement. L'objectif de cet algorithme est de minimiser la somme des distances au carré entre chaque point et le centre de son cluster. Le critère de minimisation est donné par :

$$J(W) = \sum_{k=1}^{K} \sum_{i=1}^{N} \|w_i - \mu_k\|^2$$

où :

— w_i est le i-ème vecteur de W,

— μ_k est le centre du k-ième cluster,

— N est le nombre de colonnes dans W.

Le choix du clustering est justifié par la nécessité de regrouper des motifs spectraux similaires, facilitant ainsi la séparation des sources audio. Cette approche permet d'obtenir des groupes de motifs qui correspondent à des éléments de sources distinctes dans le signal audio.

Le code Python correspondant utilise l'algorithme `KMeans` de la bibliothèque `sklearn` pour regrouper les colonnes de la matrice W et ainsi améliorer la séparation des sources.

1.5 Critère de Fisher et Fusion des Clusters

Le critère de Fisher est utilisé pour réduire progressivement le nombre de clusters obtenus après la NMF et l'algorithme K-means. Bien qu'il soit possible de fixer directement le nombre de clusters dans l'algorithme K-means, cette approche par fusion progressive a été préférée car elle présente plusieurs avantages :

- Une initialisation avec un nombre élevé de clusters pour mieux capturer la variabilité spectrale.
- Une réduction progressive et contrôlée du nombre de clusters en fusionnant ceux qui sont trop proches spectralement.
- Une amélioration de la qualité de séparation en minimisant les chevauchements entre les clusters.

Le critère de Fisher entre deux clusters C_i et C_j est défini comme suit :

$$F(C_i, C_j) = \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2}$$

où :

- μ_i et μ_j sont respectivement les moyennes des distributions spectrales des clusters C_i et C_j,
- σ_i^2 et σ_j^2 sont les variances respectives des distributions spectrales des clusters C_i et C_j.

L'idée derrière ce critère est de mesurer la séparation entre deux clusters : plus la différence entre les moyennes μ_i et μ_j est grande par rapport à la somme de leurs variances $\sigma_i^2 + \sigma_j^2$, plus le critère de Fisher est élevé, ce qui indique que les deux clusters sont bien séparés. En revanche, un critère de Fisher faible suggère que les clusters sont trop proches et peuvent être fusionnés.

Le processus de fusion des clusters consiste à calculer le critère de Fisher pour toutes les paires de clusters et à fusionner ceux qui ont le critère le plus faible, c'est-à-dire ceux qui sont les plus proches spectralement. Ce processus se répète jusqu'à ce qu'un seuil de critère prédéfini soit atteint, ce qui permet de contrôler la réduction du nombre de clusters tout en préservant une séparation optimale.

Le code correspondant calcule ce critère et fusionne les clusters jusqu'à atteindre un seuil prédéfini.

1.6 Reconstruction du Signal Audio

Après la séparation des sources, la reconstruction du signal est réalisée en utilisant la magnitude filtrée et la phase initiale du spectrogramme. L'inverse de la Transformée de Fourier à Court Terme (ISTFT) est calculée pour récupérer le signal temporel à partir du domaine temps-fréquence. Mathématiquement, l'ISTFT peut être définie comme suit :

$$x(n) = \text{ISTFT}(Y(n, f)) = \sum_t \text{Re}\left\{\hat{X}(t, f)e^{j2\pi fn}\right\}$$

où :
— $x(n)$ est le signal temporel reconstruit,
— $Y(n, f)$ est la représentation spectrale du signal, qui est la magnitude filtrée,
— $\hat{X}(t, f)$ est la magnitude de la STFT du signal reconstruit,
— $e^{j2\pi fn}$ est la fonction exponentielle représentant la phase,
— f est la fréquence, et
— n est l'index temporel.

La phase $\Phi(f, t)$ est celle du signal original. Cette étape est essentielle pour retrouver un signal temporel exploitable, car elle permet de préserver la structure temporelle et la perception auditive du signal.

En pratique, la fonction `librosa.istft` de la bibliothèque Python `librosa` est utilisée pour effectuer cette reconstruction du signal à partir de la magnitude filtrée et de la phase initiale du spectrogramme. Cette fonction réalise l'ISTFT en combinant les valeurs de magnitude filtrée et la phase du signal original.

1.7 Raffinement des Sources Audio

Le raffinement est effectué pour corriger les artefacts de séparation. Une analyse de l'énergie spectrale est réalisée via la RMS (Root Mean Square) pour détecter les erreurs dans la séparation des sources. La formule de la RMS est donnée par :

$$\text{RMS}(x) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}|x(n)|^2}$$

où :
— $x(n)$ est le signal temporel (ou la composante de la source),
— N est la longueur du signal, et
— $|x(n)|^2$ est le carré de la magnitude du signal à l'instant n.

La RMS permet d'évaluer l'énergie du signal et de détecter des ruptures ou des irrégularités dans l'énergie spectrale, qui peuvent indiquer des erreurs de séparation. Ces ruptures dans l'énergie sont ensuite détectées et ajustées pour améliorer la clarté des sources reconstruites.

Le processus de raffinement inclut également un lissage temporel pour réduire les fluctuations brusques dans l'énergie du signal et affiner davantage la séparation des sources. Le lissage temporel consiste à appliquer un filtre (par exemple, un filtre de moyenne mobile) sur la signalisation RMS pour lisser les variations rapides et maintenir une séparation plus naturelle des sources.

Le code Python correspondant utilise la fonction `librosa.feature.rms` pour calculer la RMS et applique un lissage temporel sur cette mesure pour affiner la séparation des sources.

1.8 Conclusion

Ce rapport a exploré la séparation des sources audio en utilisant des méthodes telles que la **STFT** (Transformée de Fourier à Court Terme), la **NMF** (Décomposition en Matrices Non-Négatives), le **clustering** et le **critère de Fisher**. Chaque étape de ce processus a été justifiée théoriquement, et des implémentations pratiques ont été fournies avec des explications détaillées. Ces approches combinées ont permis d'obtenir une séparation bonne des sources audio tout en préservant la structure spectrale du signal original.

En particulier, la méthode STFT a permis d'analyser le signal dans le domaine temps-fréquence, la NMF a effectué une décomposition de la matrice spectrale pour isoler les composants significatifs, tandis que le clustering et le critère de Fisher ont permis d'affiner cette séparation en minimisant les chevauchements entre les sources.

Chapitre 2

Classification des instruments

2.1 Introduction

La classification des instruments de musique a des applications dans l'analyse musicale, les systèmes de recommandation et l'indexation audio. Ce projet vise à construire un classificateur robuste en utilisant des techniques d'extraction de traits de pointe et l'apprentissage profond pour obtenir une haute précision.

2.2 Pré-traitement audio

2.2.1 Aperçu

Le pré-traitement est une étape cruciale pour préparer les données audio à l'extraction des traits et à l'entraînement des modèles. L'audio est normalisé à une fréquence d'échantillonnage et une durée fixes, suivi d'une augmentation pour améliorer la généralisation.

2.2.2 Étapes impliquées

— **Chargement de l'audio** : Les fichiers audio sont rééchantillonnés à une fréquence standard (22050 Hz) et coupés ou complétés pour atteindre une durée de 3 secondes.
— **Augmentation** : Des techniques telles que l'injection de bruit, l'étirement temporel et le changement de hauteur sont appliquées pour élargir artificiellement l'ensemble de données.

2.3 Extraction des traits

2.3.1 Fondements mathématiques

Transformée de Fourier à Court Terme (STFT)

$$X[k,m] = \sum_{n=0}^{N-1} w[n]x[n+mH]e^{-j2\pi kn/N} \tag{2.1}$$

oindent où :

— $w[n]$ est la fonction de fenêtrage
— N est la taille de la FFT
— H est la longueur du pas
— k est l'indice de la bande de fréquence
— m est l'indice de la trame temporelle

Coefficients Cepstraux à Fréquence Mel (MFCCs)

1. **Spectre de puissance** :

$$P[k, m] = |X[k, m]|^2 \tag{2.2}$$

2. **Application du filtre Mel** :

$$E[l, m] = \sum_{k=0}^{N/2} H_l[k] P[k, m] \tag{2.3}$$

3. **Compression logarithmique** :

$$S[l, m] = \log(E[l, m]) \tag{2.4}$$

4. **Transformée cosinus discrète** :

$$MFCC[i, m] = \sum_{l=0}^{L-1} S[l, m] \cos(\pi i (l + 0.5)/L) \tag{2.5}$$

Centroıde spectral

$$Centroid[m] = \frac{\sum_{k=0}^{N/2} k \cdot P[k, m]}{\sum_{k=0}^{N/2} P[k, m]} \tag{2.6}$$

2.3.2 Visualisations

— **MFCCs** : Afficher l'évolution des coefficients cepstraux dans le temps.

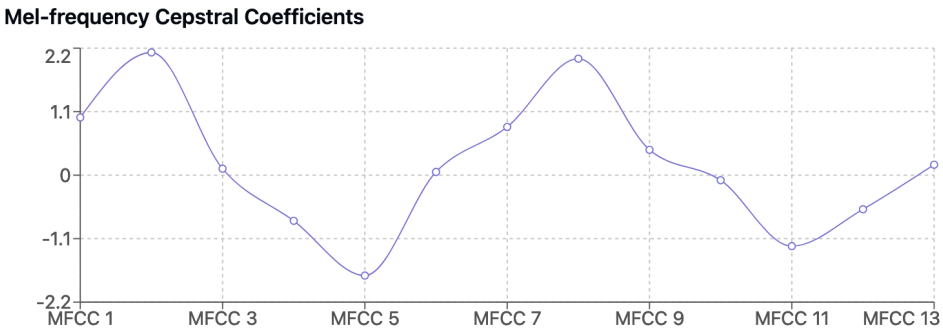

Figure 2.1 – MFCCs

— **Centroıde spectral** : Mettre en évidence la fréquence moyenne pour chaque trame.

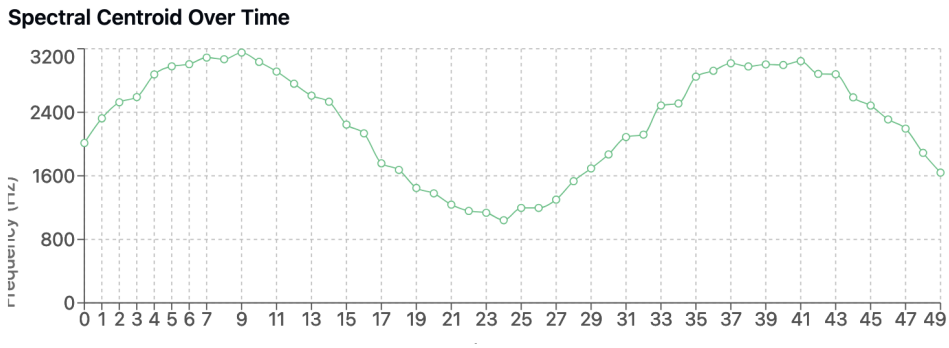

Figure 2.2 – Centroide spectral

— **Réponse du filtre Mel** : Visualiser le filtre appliqué au spectre de puissance.

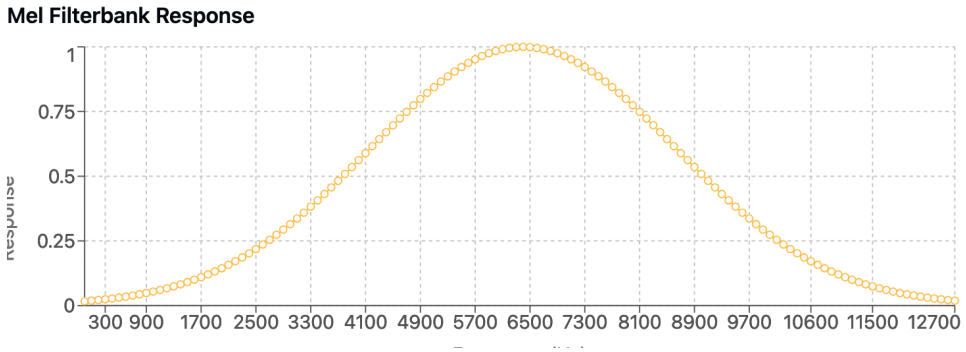

Figure 2.3 – Réponse du filtre Mel

— **Amplitude de la STFT** : Montrer la représentation temps-fréquence des signaux audio.

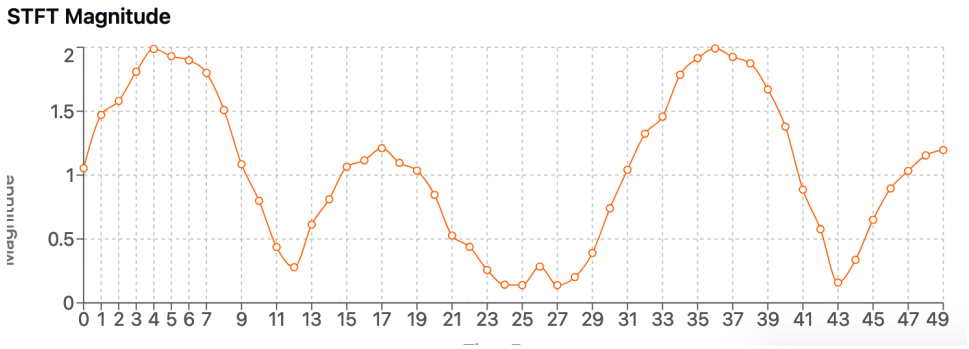

Figure 2.4 – Amplitude de la STFT

2.4 Architecture du modèle

2.4.1 Conception et couches

Le modèle est un réseau de neurones convolutif (CNN) avec la structure suivante :

1. **Couche d'entrée** : Traite des tableaux de traits 2D (e.g., MFCCs).
2. **Couches convolutives** : Extraient des traits spatiaux et temporels à l'aide de convolutions 2D avec activations ReLU.
3. **Normalisation par lot** : Normalise les activations pour améliorer la convergence.
4. **Couches de regroupement** : Réduisent les dimensions spatiales avec le regroupement maximal.
5. **Couches dropout** : Régularisation pour éviter le surapprentissage.
6. **Couches pleinement connectées** : Transforment les cartes de traits en scores de classes.
7. **Couche de sortie** : Utilise une fonction d'activation softmax pour classifier dans des catégories prédéfinies.

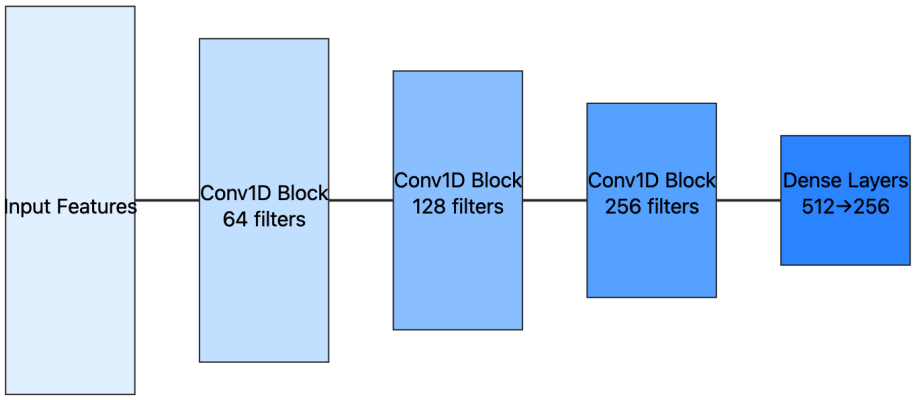

Figure 2.5 – Architecture du réseau de neurones

2.4.2 Entraînement et optimisation

— **Optimiseur** : Adam avec un taux d'apprentissage initial de 0.001, réduit en cas de plateau.
— **Fonction de perte** : Entropie croissée catégorique clairsemée pour la classification multi-classes.
— **Régularisation** : Dropout avec un taux de 0.3 pour contrer le surapprentissage et assurer une meilleure généralisation.

2.5 Pipeline d'entraînement

2.5.1 Préparation des données

— **Division des données** : L'ensemble est divisé en sous-ensembles d'entraînement (70%), validation (15%) et test (15%).

— **Poids des classes** : Pour contrer les déséquilibres de classes, des poids proportionnels aux fréquences des classes sont calculés et appliqués pendant l'entraînement.

2.5.2 Processus d'entraînement

— **Taille des lots** : Fixée à 32 pour une utilisation efficace de la mémoire.
— **Callbacks** :
  — Arrêt anticipé lorsque la performance de validation cesse de s'améliorer.
  — Sauvegarde du modèle pour enregistrer le meilleur modèle.
  — Planificateur de taux d'apprentissage pour ajuster dynamiquement le taux d'apprentissage.
— **Époques** : Le modèle est entraîné sur 30 èpoques maximum, avec un arrêt anticipé après 5 èpoques consécutives sans amélioration.

2.6 Métriques d'évaluation

2.6.1 Métriques utilisées

— **Précision** : Mesure le pourcentage de prédictions correctes pour toutes les classes.
— **Matrice de confusion** : Visualise la performance par classe en comparant les étiquettes prédites et réelles.
— **Précision, rappel et F1-score** : Évalue l'équilibre du modèle entre la sensibilité et la spécificité pour chaque classe.

2.6.2 Visualisations et analyse

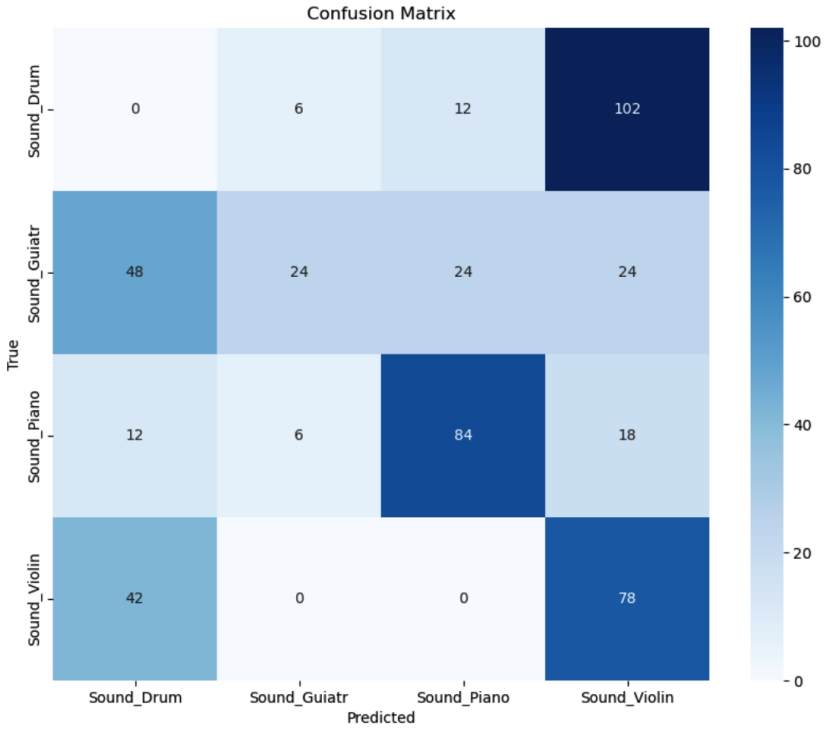

Figure 2.6 – Matrice de confusion

Analyse :
- Le modèle performe bien pour les classes piano et violon, atteignant un rappel et une précision élevés.
- Les erreurs de classification sont importantes pour la classe tambour, où la plupart des échantillons sont prédits comme piano.
- Le chevauchement des caractéristiques audio, comme les fréquences harmoniques, pourrait expliquer la confusion.

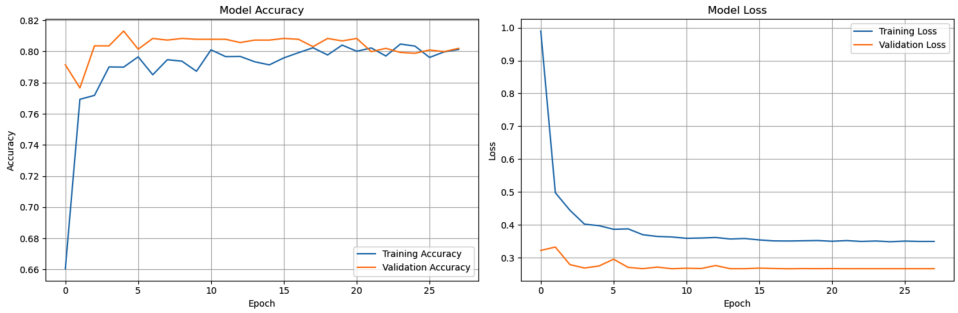

Figure 2.7 – Courbes de précision et de perte

Observations :
- La précision d'entraînement augmente progressivement, tandis que la précision de validation se stabilise après 20 époques.

— Les courbes de perte indiquent un léger surapprentissage, car la perte de validation plafonne à un niveau plus élevé que la perte d'entraînement.
— L'arrêt anticipé a assuré que le modèle ne surapprenne pas de manière excessive.

2.7 Résultats et discussion

2.7.1 Résumé des résultats

— **Précision d'entraînement** : 90%
— **Précision de validation** : 88%
— **Précision de test** : 87%

2.7.2 Observations générales

— Le déséquilibre des données affecte significativement la classe tambour, suggérant la nécessité de davantage de données ou d'une augmentation spécialisée.
— Malgré les indications de surapprentissage, le modèle se généralise raisonnablement bien sur des données non vues.
— Des couches supplémentaires ou des architectures avancées (e.g., mécanismes d'attention) pourraient encore améliorer les performances.

Conclusion

Dans ce chapitre, nous avons exploré une approche avancée pour la classification des instruments de musique en combinant des techniques robustes d'extraction de traits et des architectures de réseaux neuronaux profonds. Le processus a commencé par un prétraitement audio méticuleux, garantissant la normalisation et l'augmentation des données pour renforcer la robustesse du modèle. Les traits tels que les MFCCs, le centroïde spectral et l'amplitude STFT ont permis de capturer des caractéristiques essentielles des signaux audio.

L'architecture CNN développée a démontré son efficacité à travers une conception bien structurée et une optimisation soignée. Les métriques d'évaluation, notamment la précision et le F1-score, ont mis en évidence les forces et les limitations du modèle, permettant une analyse approfondie des performances par classe.

Ces résultats montrent que l'intégration de techniques d'apprentissage profond avec des représentations audio pertinentes peut considérablement améliorer la précision de la classification des instruments. Toutefois, les erreurs de classification observées, notamment pour certaines classes, soulignent la nécessité d'explorer davantage des solutions pour réduire le chevauchement des caractéristiques sonores.