MU4RBI01 Advanced Audio Processing

Lab 4: traitement avancée des sons : Séparations de sources

KHAOUS Noureddine
hu.wenxuan@etu.sorbonne-universite.fr
ASSOUL Massin
achraf.el-messaoudi@etu.sorbonne-universite.fr

Supervisor
Alice COHEN-HADRIA
alice.cohen$_h$*adria*@*sorbonne* − *universite.fr*
Sorbonne University - Institute for Research and Coordination in Acoustics/Music

Academic year 2024-2025

Introduction

La séparation des sources musicales, et plus spécifiquement la séparation des voix chantées des pistes d'accompagnement, constitue un défi majeur en traitement du signal audio. Cette problématique, essentielle dans des domaines tels que la transcription automatique des paroles, l'identification des chanteurs ou encore les applications commerciales comme le karaoké, s'appuie sur des méthodes avancées d'apprentissage automatique pour atteindre des résultats significatifs. Dans le cadre de ce travail, nous nous focalisons sur l'implémentation et l'évaluation d'un modèle basé sur l'architecture U-Net.

Le rapport présente les choix d'implémentation retenus, les défis rencontrés, ainsi que les résultats obtenus lors de la séparation des sources. En complément, un notebook interactif et des exemples audio illustrent les performances de notre système.

0.1 Présentation de MUSDB18

Nous avons utilisé la base de données **MUSDB18**, disponible sur `sigsep.github.io/datasets/musdb.html` ou via la librairie `musdb` (`pip install musdb`).

Cette base contient 150 morceaux audio multipistes, avec des pistes isolées de *vocals*, *drums*, *bass*, etc. Pour notre tâche de séparation de la voix, nous n'avons besoin que de la piste *vocals* et du mix complet.

0.2 Comment l'obtenir

1. Télécharger les fichiers audio (MUSDB18).
2. Placer ces fichiers dans un répertoire, par exemple `data/musdb18`.
3. Charger la base (train/test) en Python:

```
import musdb
train_mus = musdb.DB(root="data/musdb18", subsets="train")
test_mus  = musdb.DB(root="data/musdb18", subsets="test")
```

0.3 Prétraitements

Nous convertissons les pistes en **mono** (moyenne stéréo) et **downsamplons** à 8192 Hz pour réduire la taille des spectrogrammes. Nous appliquons ensuite la *STFT* (Short-Time Fourier Transform) avec `n_fft = 1024` et `hop_length = 768`, puis **découpons** les spectrogrammes en patches temporels (`patch_size = 128`) pour faciliter l'entraînement en mini-lots (*batches*).

1 Architecture

Nous avons utilisé une **architecture U-Net** adaptée au traitement des spectrogrammes audios.

1.1 Structure générale

- **Encodeur (descendant)**: Convolutions 2D avec stride=2, réduisant la taille spatiale (512 $\rightarrow$ 256 $\rightarrow$ 128, etc.) tout en augmentant le nombre de canaux (16 $\rightarrow$ 32 $\rightarrow$ 64, etc.).
- **Décodeur (montant)**: Upsampling (facteur $\times 2$) + Convolutions 2D pour restaurer la résolution initiale.
- **Skip connections**: à chaque niveau, la sortie de l'encodeur est ajoutée (concaténée ou sommée) à la couche correspondante du décodeur, ce qui préserve les détails haute résolution.
- **Sortie**: un masque (entre 0 et 1) appliqué à la magnitude du mix, afin d'extraire la partie associée à la voix. On utilise en général une activation *sigmoïde* en sortie.

1.2 Paramètres clés

- **Taille en fréquence**: on tronque les spectrogrammes à 512 bins (sur 513 possibles pour `n_fft=1024`), pour obtenir une dimension multiple de 2. Cela évite les problèmes de dimension impaire lors des convolutions successives.
- **Profondeur**: 6 niveaux d'encodeur et 6 niveaux de décodeur, montant jusqu'à 512 canaux au niveau le plus profond.
- **Activation finale**: sigmoïde, retournant un masque à multiplier par `X_mag`.

2 Entraînement et résultats

2.1 Configuration de l'entraînement

- **Fonction de perte**:
$$\mathcal{L}(\hat{V}, Y) = \|\hat{V} \times X_{\text{mag}} - Y_{\text{mag}}\|_1$$
(L1 sur le spectrogramme reconstruit).
- **Optimiseur**: `Adam` avec `lr = 1e-3`.
- **Nombre d'époques**: Nous avons initialement entraîné le modèle sur 50 époques, mais nous avons observé qu'après 25 époques, il n'y avait plus de changement significatif dans les performances. Pour éviter le surapprentissage (overfitting), nous avons donc réduit la durée de l'entraînement à 25 époques.

Nous avons observé une **baisse régulière de la loss** sur le jeu d'entraînement, accompagnée d'une baisse similaire (légèrement plus haute) sur le jeu de validation, sans remontée significative. Cela indique **peu de sur-apprentissage**.

2.2 Interprétation des résultats

Qualité d'écoute: À 8192 Hz, on conserve la *bande de fréquences* principale de la voix (0–4 kHz), mais on perd une partie des hautes fréquences. Malgré cela, la séparation est **convaincante** dans la plupart des cas: la voix est audible et relativement propre, même si quelques artefacts ou un léger *bleed* instrumental subsistent.

Visualisation des spectrogrammes: En traçant les spectrogrammes (sortie estimée vs. original), on constate que les **zones fréquentielles dominantes de la voix** sont correctement captées par le masque du U-Net.

Stabilité et convergence: Les courbes de loss (0.004–0.006) traduisent un apprentissage cohérent. L'utilisation d'un **U-Net profond** et la **réduction à 512 bins** (en plus du downsampling) ont permis un *entraînement relativement rapide* sans explosions de mémoire ni blocages de convergence.

2.3 Gestion du recouvrement entre les patches

2.3.1 Méthode Utilisée

Pour préparer les données, nous avons adopté une approche différente de celle proposée dans le fascicule du TP. Plutôt que de sélectionner des segments d'audio de 7 secondes et de générer des spectrogrammes avec un fort taux de recouvrement, nous avons opté pour une méthode plus efficace et représentative de notre base de données.

2.3.2 Description de la Méthode

1. **Préparation des Spectrogrammes** : Nous avons appliqué la transformation STFT (Short-Time Fourier Transform) sur l'ensemble du signal audio, plutôt que sur des segments de 7 secondes. Cette transformation a été effectuée avec les paramètres d'analyse spécifiés dans l'article de référence.

2. **Division en Patches** : Une fois les spectrogrammes obtenus, nous les avons divisés en patches de 128 trames sans recouvrement (overlap). Chaque patch contient des informations de magnitude et de phase pour les fréquences et les trames correspondantes.

3. **Création du Dataset** : Nous avons répété ce processus pour tous les fichiers audio de notre base de données. Les patches obtenus ont été stockés dans une liste, créant ainsi un dataset complet de patches.

4. **Sélection Aléatoire pour l'Entraînement** : Pour l'entraînement, nous avons utilisé un `DataLoader` avec l'option `shuffle=True`. Cela permet de sélectionner aléatoirement des patches pour chaque batch, garantissant ainsi une bonne représentativité de l'ensemble de la base de données et évitant le problème de recouvrement excessif des générateurs naïfs.

2.3.3 Avantages de la Méthode

- **Évitement du Recouvrement Excessif** : Contrairement à la méthode proposée dans le fascicule, notre approche évite le problème de recouvrement excessif entre les spectrogrammes de 128 trames. Cela permet de mieux représenter la diversité des données dans chaque batch.

- **Représentativité du Batch** : En utilisant un `DataLoader` avec `shuffle=True`, nous nous assurons que chaque batch est représentatif de l'ensemble de la base de données. Cela améliore la généralisation du modèle et réduit le risque de surapprentissage (overfitting).

Analyse des métriques calculées

Les métriques (*SDR*, *SIR*, *SAR*) calculées sont très faibles ou négatives pour la majorité des extraits, et la valeur de *SAR* apparaît souvent infinie. Cela peut être interprété comme suit :

- ***SDR* négatif :** l'erreur de séparation est plus importante que la voix elle-même, suggérant que le modèle n'isole pas correctement la voix (souvent recouverte par l'accompagnement ou réduite au silence).

- ***SIR* négatif :** l'interférence (instruments résiduels) domine la voix séparée, indiquant que la piste vocale estimée contient trop d'autres sons.

- ***SAR* infini :** dans de nombreuses fenêtres, la partie « artefact » détectée est nulle ou insignifiante, ce qui peut être la conséquence d'une sortie presque silencieuse sur certaines zones (ou d'un calcul extrême lié au fenêtrage).