

République Tunisienne Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Université de Tunis 	Mini Projet 1 ^{er} SEMESTRE A-U : 2025-2026	Date de création :
Classe : Master Info Matière : Séminaire de recherche Deep Learning		
Date de remise : 11/12/2025	Enseignant : Dr. Mehrez Boulares	Nbre de pages : 03

Projet : VoiceGAN-Transformation – Conversion de voix

A→B Contexte

On souhaite développer un système capable de prendre la voix d'une personne A et de la retranscrire dans le style vocal d'une personne B.

Exemple : une phrase dite par A → en sortie, la même phrase mais avec le timbre, l'intonation et la prosodie (l'ensemble des traits oraux d'une expression verbale d'un locuteur) de B.

Applications possibles :

- doublage personnalisé (cinéma, jeux vidéo, avatars virtuels),
- aides médicales (patients ayant perdu leur voix),
- anonymisation ou personnalisation vocale.

Objectif

Construire un modèle de conversion vocale A→B :

1. A dit une phrase (contenu).
2. On conserve le contenu linguistique.
3. On change le style vocal pour qu'il corresponde à B.
4. On génère un audio final réaliste et fluide.

Approche technique

1. Prétraitement audio

- Entrées : voix de A (source) et voix de B (référence).
- Transformation en melspectrogrammes.

1

2. Encodeurs

- Content Encoder (CNN + Transformer) :
 - Capture le contenu de la voix de A (les mots prononcés).
- Style Encoder (CNN) :
 - Capture le timbre et les caractéristiques vocales de B.

3. Générateur (G)

- Fusionne contenu (A) + style (B).
- Génère un nouveau spectrogramme correspondant à "contenu A mais style B".

4. Discriminateur (D, basé CNN)

- Vérifie si le spectrogramme ressemble à une vraie voix de B.
- Force le générateur à produire des voix naturelles.

5. Vocoder

- Convertit le spectrogramme généré en fichier audio (.wav).
- Ex : HiFi-GAN ou MelGAN.

Fonction de perte

- Perte de reconstruction (L1) : spectrogramme généré \approx spectrogramme réel de B.
- Perte adversariale (GAN) : assure un son réaliste.
- Perte d'identité vocale : garantit que la sortie ressemble bien à B.
- Perte de contenu : garantit que le texte prononcé reste celui de A.

Étapes du projet

1. Collecte des données : voix de A et voix de B (ex. VCTK, LibriTTS).
2. Prétraitement : melspectrogrammes normalisés.
3. Implémentation :
 - Content Encoder (CNN + Transformer),
 - Style Encoder (CNN),
 - Générateur CNN,
 - Discriminateur CNN.

4. Entraînement du GAN avec pertes multiples.
5. Inference :
 - o Input = audio de A + référence audio de B,
 - o Output = voix transformée (A→B).
6. Évaluation :
 - o Objective (Mel Cepstral Distortion, Similarité cosinus),
 - o Subjective (écoute humaine MOS).
7. Démo : interface interactive (ex. Streamlit) → l'utilisateur charge la voix de A et choisit une voix de B.

Livrables

- Code (prétraitement, modèle, entraînement, inference).
- Rapport détaillé (~12-15 pages) : architecture, résultats, comparaisons CNN vs CNN+Transformer+GAN.
- Exemples audio (avant/après conversion).
- Schéma du pipeline A→B.

Pipeline simplifié

1. Voix A (contenu) → CNN+Transformer → Encodage du contenu.
2. Voix B (style) → CNN → Encodage du style.
3. Fusion contenu+style → Générateur → Spectrogramme A→B.
4. Vocoder → Audio final (A qui parle avec la voix de B).

La logique A→B : on prend ce qu'A dit et on le fait dire avec la voix de B, en combinant CNN (local), Transformer (global) et GAN (réalisme).