Université Sorbonne Paris Nord
Sup Galilée
Spécialité Informatique

Conduite et gestion de projet

Spécialité : INFO2

Livrable : Sprint 2

Interface de Génération Sonore par IA

Étudiants :
Seyfeddine JOUINI
Yosra SASSI
Sirine TLILI
Hasna ELGARANI
Kaoutar BRAHIMI

Client :
Vittascience

9 janvier 2025

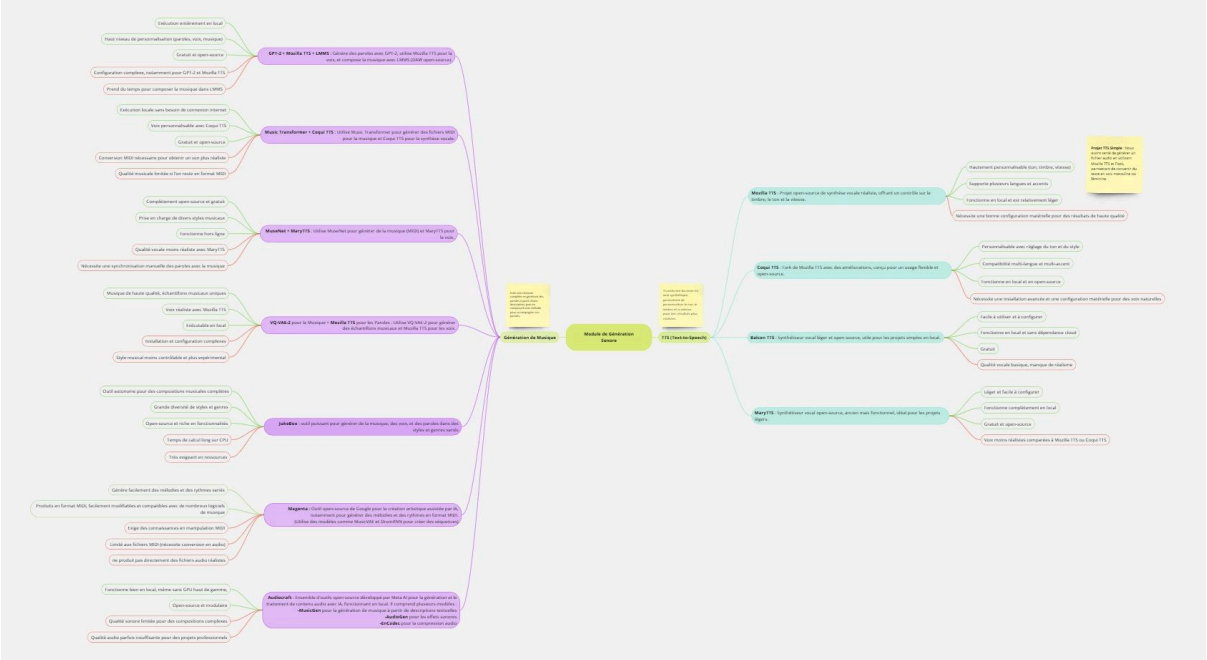
Table des matières

Introduction

Ce rapport présente les activités réalisées durant le Sprint 2 du projet de développement d'un module de génération sonore par l'intelligence artificielle. Tout d'abord, une recherche approfondie a été menée pour identifier et analyser les technologies adaptées à chaque cas d'utilisation. Ensuite, des descriptions détaillées des cas prioritaires ont été élaborées. Par la suite, les progrès réalisés dans ce sprint ont été présentés, et enfin, un prototype a été conçu pour valider les solutions techniques, accompagné d'une analyse des principaux défis rencontrés et des solutions apportées.

1. Recherche et analyse technologiques

Dans cette section, nous nous sommes concentrés sur l'identification et l'analyse des technologies possibles pour réaliser le projet de génération sonore par IA. Notre approche a consisté à regrouper les technologies adaptées à des cas d'utilisation, à en analyser les avantages et les inconvénients, et à construire une carte conceptuelle qui illustre clairement les options disponibles.



Critères de choix technologique

Après discussion avec l'équipe et notre client, les critères suivants ont été fixés pour guider le choix des technologies :

- Fonctionner localement sans connexion Internet (offline).
- Être compatible avec JavaScript pour une intégration facile.
- Offrir des performances suffisantes pour des cas d'utilisation interactifs.
- Permettre la manipulation de paramètres sonores (exemple : hauteur, vitesse, etc..).

Cette analyse a permis d'identifier les technologies les plus adaptées pour chaque cas d'utilisation, tout en respectant les contraintes techniques et fonctionnelles définies. Ces choix guideront la conception et le développement des prototypes à venir.

2. Description détaillée des cas prioritaires

2.1. Cas d'utilisation 1 : Génération de speech à partir d'un prompt

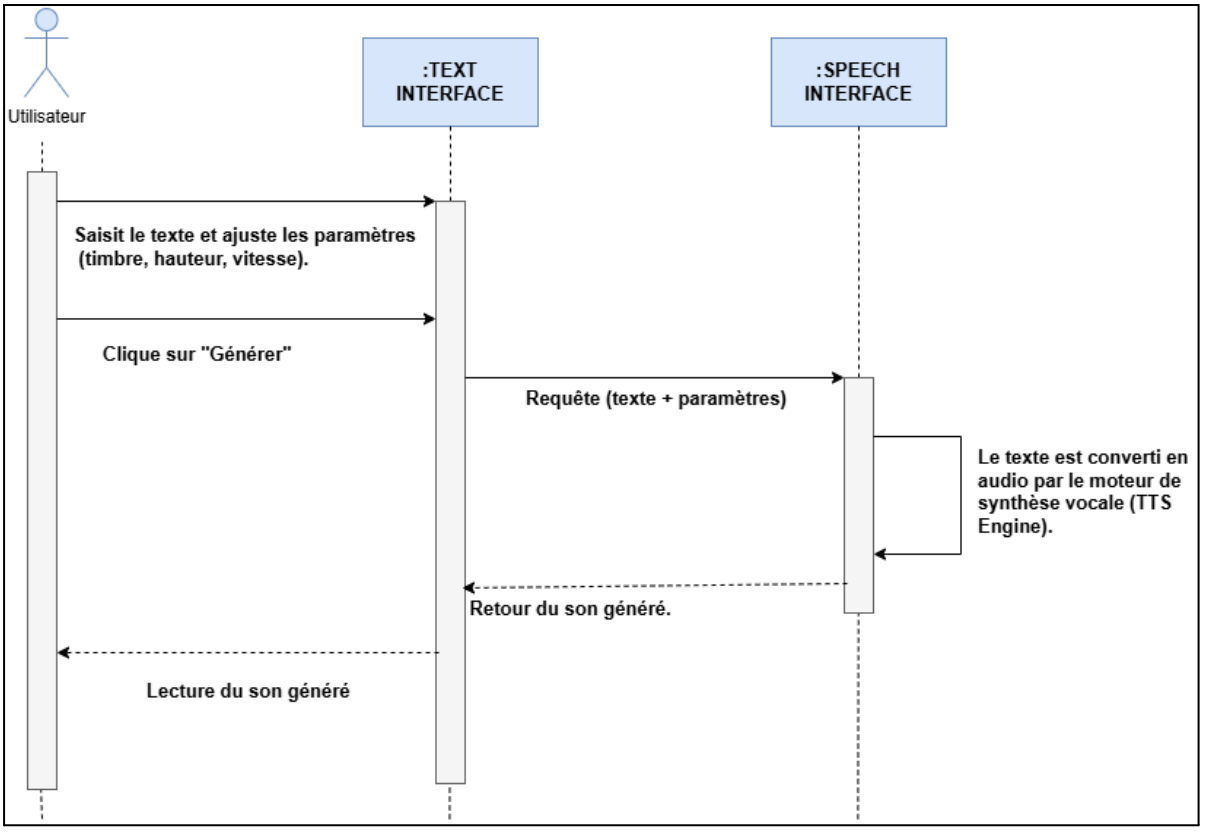
Objectif :

Ce cas d'utilisation vise à permettre à un utilisateur de générer un son synthétique réaliste à partir d'un texte saisi (Text-To-Speech), avec des paramètres personnalisables tels que le type de voix (homme ou femme), la hauteur et la vitesse. Cette fonctionnalité répond au besoin de transformer du texte en voix de manière fluide et intuitive, adaptée à des usages pédagogiques ou créatifs.

Scénario utilisateur :

1. L'utilisateur ouvre l'interface de génération.
2. Il saisit un texte dans le champ prévu à cet effet.
3. L'utilisateur ajuste les paramètres de la voix (par exemple : hauteur, vitesse, etc..).
4. Il clique sur le bouton "Générer".
5. Le système traite la requête et génère un fichier audio correspondant au texte saisi.
6. Le fichier audio est restitué à l'utilisateur, qui peut l'écouter directement.

Diagramme de séquence:



Technologies :

1. **API Web Speech (SpeechSynthesis)** :
    - **Description** : L'API Web Speech est une fonctionnalité native des navigateurs modernes qui permet de synthétiser de la voix à partir de texte sans dépendre d'une connexion Internet ou de serveurs externes.
    - **Avantages** :
        - Fonctionne localement (offline) sur les navigateurs supportant cette API.
        - Permet de personnaliser les paramètres vocaux (hauteur, vitesse, etc..).
        - Facilité d'intégration dans le projet du notre client..
    - **Inconvénients** :
        - Support limité selon le navigateur utilisé (incompatibilité avec certains navigateurs obsolètes).
        - Options de voix dépendantes des voix disponibles sur le système de l'utilisateur.
2. **JavaScript** :
    - **Description** : Utilisé pour développer la logique de l'application et interagir avec l'API Web Speech.
3. **HTML et CSS** :
    - **Description** : Utilisés pour construire l'interface utilisateur et améliorer l'expérience utilisateur.

2.2. Cas d'utilisation 2 : Lipsync à partir d'un audio et d'une image

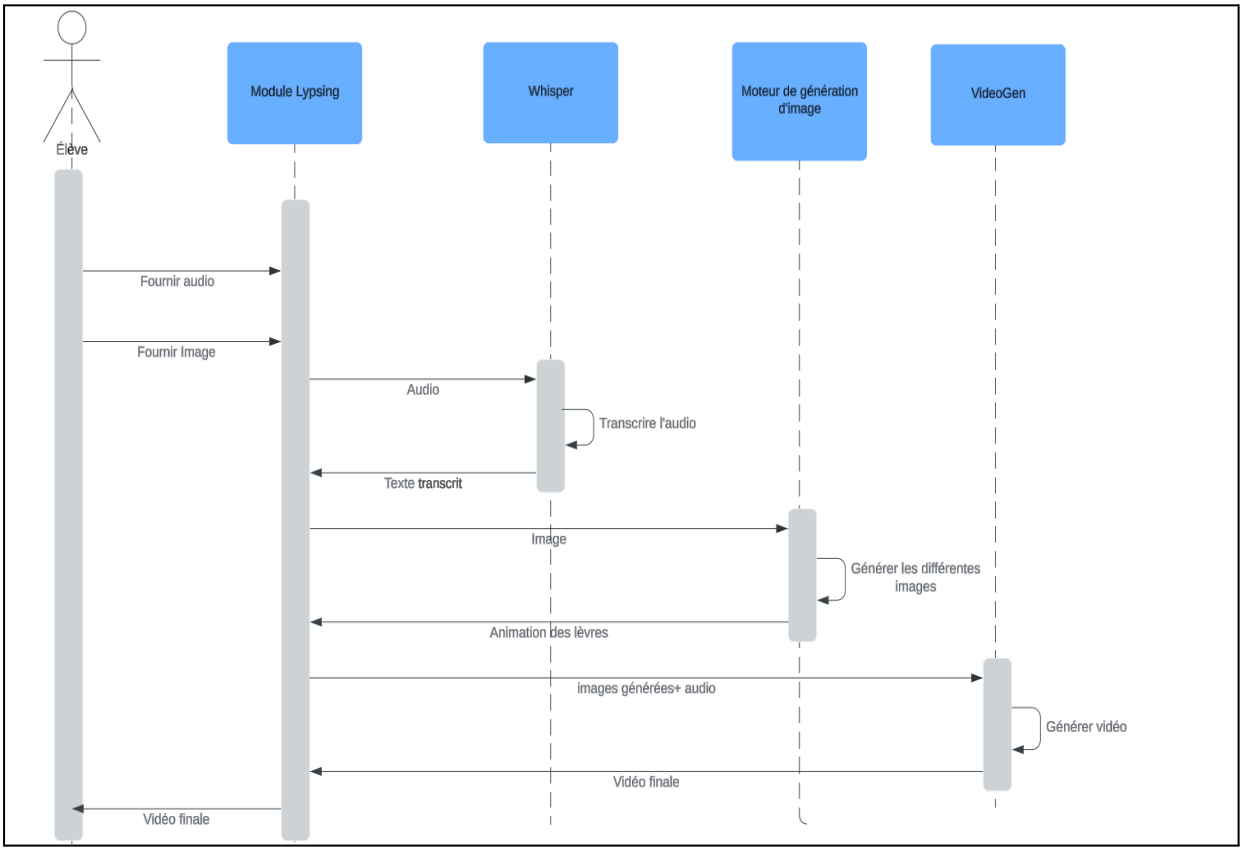
Objectif :

Cette fonctionnalité permet à l'utilisateur d'animer une image contenant un visage en synchronisant les mouvements des lèvres avec un dialogue ou une chanson d'un audio fourni, pour créer l'illusion réaliste que le personnage parle ou chante.

Scénario utilisateur :

1. Fournir une image contenant un visage et un audio.
2. Le système transcrit l'audio.
3. Le système génère les différentes formes du visage (bouche ouverte/ bouche ronde …).
4. Le système créer la vidéo en faisant la correspondance entre les lettres du texte transcrit et les images.
5. La vidéo est restituée à l'utilisateur.

Diagramme de séquence:



Technologies :

1. **Python :**
   - **Description :** Utilisé pour le développement de l'intégralité de la fonctionnalité (tout fonctionne en off-line)
   - **Remarque :** Dans cette partie il y a eu des changements par rapport à l'autre fois afin d'améliorer les performances du système.
   - **Changement :** Remplacement du modèle **Vosk** par la bibliothèque **Whisper** .

Bibliothèques utilisées :

1. **Whisper:** C'est une bibliothèque Python qui permet la transcription d'un audio.
2. **Numpy:** C'est une bibliothéque utilisée pour le traitement d'images qui nous a permis la génération des différentes images.

Remarque : Les fichiers audio peuvent être de type : mp3, mp4, mpeg, mpga, m4a, wav et webm.

3. Avancement sur les cas prioritaires

3.1. Cas d'utilisation 1 : Génération de speech à partir d'un prompt

Activités réalisées

- Création d'une application pour tester une API Text-to-Speech (TTS) fonctionnant localement sans dépendre d'une connexion Internet.
- Développement d'une interface utilisateur intuitive permettant de saisir un texte, d'ajuster les paramètres (hauteur, vitesse, volume) et de sélectionner la langue et la voix(homme/femme).

Résultat

- Application fonctionnelle, testée avec succès sur les navigateurs modernes prenant en charge l'API Web Speech.
- Les paramètres ajustables fonctionnent correctement, offrant une personnalisation vocale précise.

3.2. Cas d'utilisation 2 : Lipsync à partir d'un audio et d'une image

Activités réalisées

- Exploration des outils et technologies capables de synchroniser des mouvements labiaux avec un fichier audio.
- Recherche sur les bibliothèques disponibles pour le traitement audio et l'analyse des phonèmes, notamment pour mapper les phonèmes de l'audio avec des animations faciales réalistes.
- Transcription d'audio réussite en utilisant Whisper.
- Synchronisation des images avec l'audio dans une vidéo.

Résultat

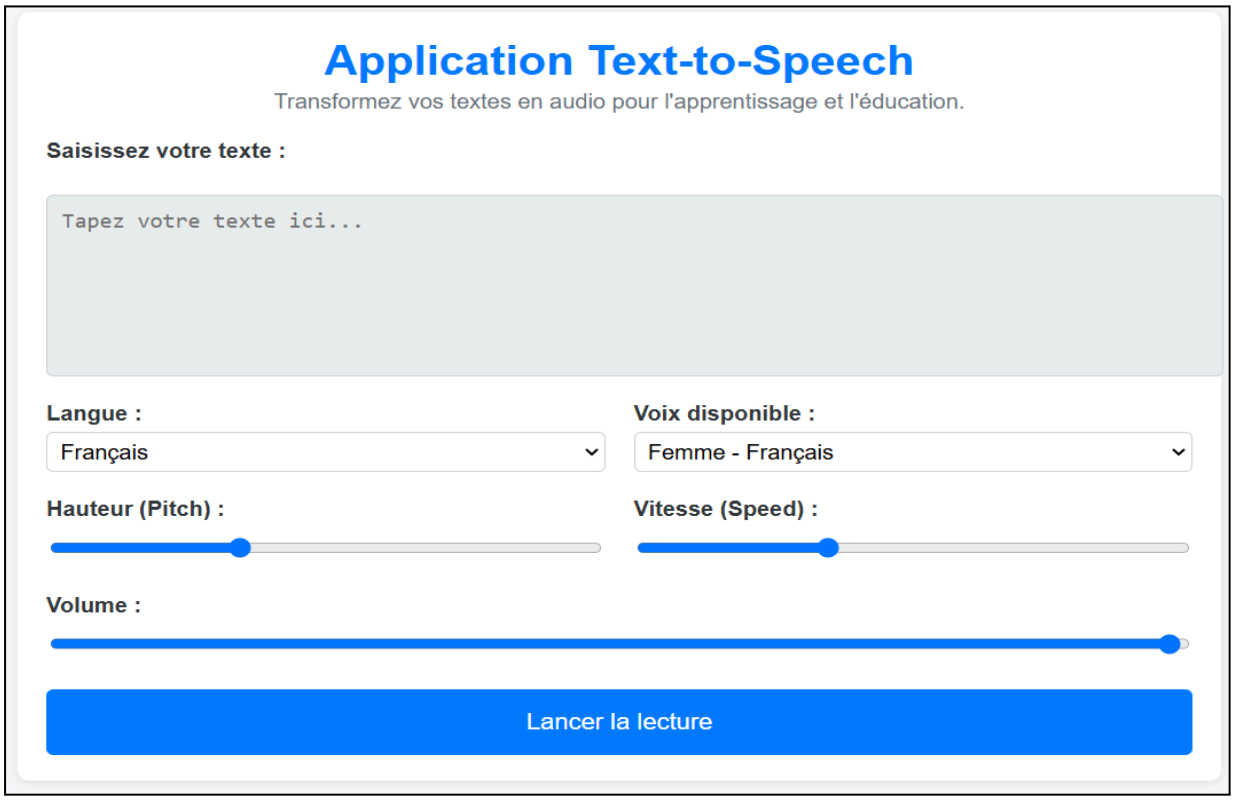
- Fonctionnalité testée permettant d'obtenir une vidéo animée avec les lèvres synchronisées avec l'audio.
- Les images générées montrent un changement au niveau des lèvres cependant ce n'est pas encore au point.

4. Prototype

4.1. Cas d'utilisation 1 : Génération de speech à partir d'un prompt

Fonctionnalités techniques de l'application :

- **Personnalisation des voix** :
  - L'utilisateur peut choisir parmi différentes voix (masculines et féminines) disponibles sur son système.
  - Les voix sont filtrées selon la langue sélectionnée (Français ou Anglais).
- **Paramètres ajustables** :
  - **Hauteur (pitch)** : Contrôle le timbre de la voix pour des variations émotionnelles.
  - **Vitesse (rate)** : Permet de ralentir ou d'accélérer la lecture.
  - **Volume** : Ajuste l'intensité sonore de la voix générée.
- **Compatibilité linguistique** :
  - Support pour le Français et l'Anglais grâce à un menu de sélection des langues.
  - Résultats obtenus ou problèmes identifiés lors de cette conception.



Avantages de cette approche technologique :

- **Exécution locale (offline)** : Aucune connexion Internet n'est requise, ce qui améliore la confidentialité des données et la rapidité d'exécution.
- **Simplicité d'utilisation** : Une interface intuitive permet à l'utilisateur de générer du son en quelques clics.
- **Flexibilité et personnalisation** : Les paramètres ajustables offrent une expérience utilisateur adaptée à des besoins variés (éducation, narration, etc.).

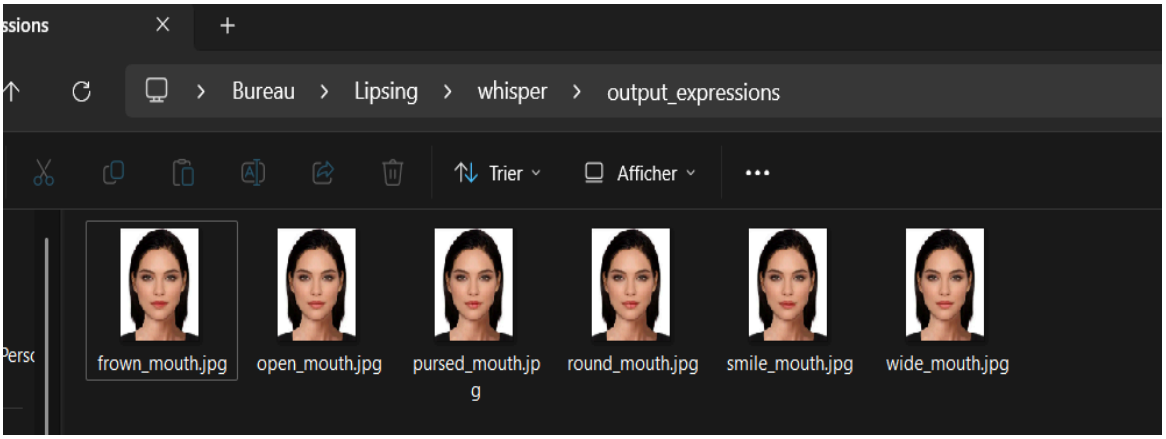
- **Open source et extensibilité** : Le code basé sur JavaScript et l'API Web Speech peut être facilement étendu pour inclure des fonctionnalités supplémentaires (exemple : nouvelles langues, nouvelles voix).

4.2. Cas d'utilisation 2 : Lipsync à partir d'un audio et d'une image

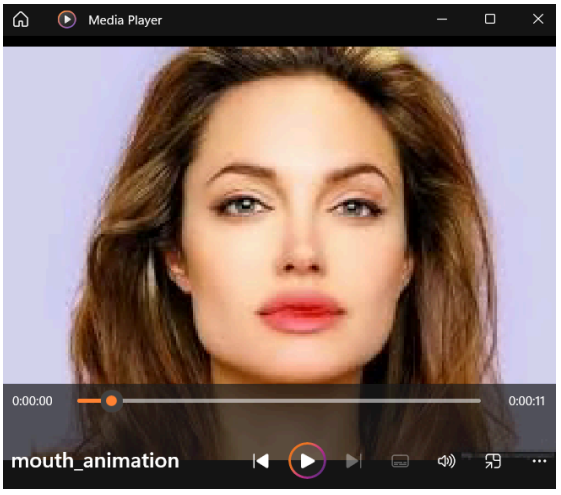
Fonctionnalités techniques de l'application :

- **Transcription de l'audio :**
  La figure ci-dessous illustre la transcription d'un audio.

- **Génération des expressions du visage :**



- **Animation de l'image:**



Avantages de cette approche technologique :

- **Exécution locale (hors ligne)** : Fonctionne sans connexion Internet, garantissant une meilleure confidentialité des données et une exécution plus rapide.
- **Facilité d'utilisation** : Une interface intuitive permet de créer du son en seulement quelques clics.

5. Conception des cas d'utilisation restants

5.1. Speech-to-Text

Ce cas d'utilisation consiste à convertir rapidement et avec précision un contenu vocal en texte. Il peut être utilisé pour la transcription d'audios ou la reconnaissance vocale dans des applications interactives.

Activités réalisées

- Exploration des technologies et modèles de reconnaissance vocale tels que Google Speech-to-Text API et Mozilla DeepSpeech.

Conception

- Définition des paramètres clés, incluant la langue, le taux de précision attendu, et la gestion des accents ou variations linguistiques.

Résultat attendu

- Développement d'un prototype capable de convertir des enregistrements audio simples en texte avec un niveau de précision élevé et des performances optimisées.

5.2. Speech-to-Speech (Transformation vocale)

Ce cas étend les fonctionnalités du système en permettant de transformer une voix enregistrée en une autre tout en conservant un rendu naturel.

Activités réalisées

- Enregistrement audio depuis le microphone.
- Transcription de l'audio en texte.
- Estimation du genre de la voix (masculin/féminin).
- Synthèse vocale du texte transcrit.

Conception

- Module d'enregistrement et transcription
  - Utiliser <sounddevice> pour capturer l'audio.
  - Emploie Vosk pour la transcription en temps réel.
- Module d'analyse vocale
  - Estime le genre avec l'analyse de fréquence fondamentale.
- Module de synthèse vocale
  - Utilise Mozilla TTS pour la conversion Texte-Parole.

Résultat attendu

- Mise en place d'un prototype initial capable de transformer une voix enregistrée en un style ou un timbre différent tout en conservant un rendu réaliste.

5.3. Génération de Musique

Ce cas propose une extension créative du système, permettant la génération automatique de compositions musicales personnalisées en fonction des paramètres définis par l'utilisateur, tels que le style, la durée et le tempo.

Activités réalisées

- Recherche approfondie sur des outils IA comme Magenta (Google), Jukebox (OpenAI), et VQ-VAE-2 pour la composition musicale.

Conception

- Définition des paramètres utilisateur personnalisables (style musical, tonalité, durée).
- Développement d'un modèle IA permettant de générer une composition musicale cohérente à partir des préférences de l'utilisateur et, si nécessaire, d'un texte fourni.
- Validation des résultats via des tests d'écoute et ajustement des paramètres en fonction des retours.

Résultat attendu

- Génération d'un fichier musical simple, conforme aux paramètres choisis par l'utilisateur, et adapté à des styles préconfigurés.

6. Défis et solutions

6.1. Text-to-Speech

Un des principaux défis rencontrés lors de ce sprint concerne la compatibilité limitée de l'API Web Speech avec certains navigateurs. Cette restriction peut entraîner des difficultés d'accès pour certains utilisateurs. Pour y remédier, l'application a été optimisée et testée sur les navigateurs les plus largement utilisés, en veillant à garantir une expérience fluide et cohérente tout en restant entièrement basée sur JavaScript.

6.2. Lipsync

Un des principaux défis rencontrés est la transcription exacte des audios qu'on faisait au début avec vosk qui faisait une transcription pas totalement exacte en plus de consommer beaucoup de temps et d'espace mémoire. Pour résoudre ce défis on a recherché d'autres solutions plus légères ce qui nous a mené à Whisper.

Au niveau de l'animation des images, au début on a utilisé la bibliothèque Rhubarb qui était supposé faire l'animation des images en les synchronisant avec l'audio mais qui n'a pas voulu fonctionner. Du coup, on a opté pour une solution plus simple, c'est de générer les images des différentes formes des lèvres et puis les utiliser.

Conclusion

Ce rapport a présenté les travaux réalisés durant le Sprint 2, mettant en lumière les avancées majeures sur les cas prioritaires, notamment la création d'une application fonctionnelle pour le Text-to-Speech, la recherche approfondie sur les technologies adaptées, et le développement de prototypes pour valider certaines approches. Les bases techniques établies lors de ce sprint permettront de poursuivre le développement des fonctionnalités restantes dans les prochains sprints, tout en consolidant les acquis pour offrir un système performant et conforme aux attentes.