**Music Genre Classification with SVM and VGG19**

Projet universitaire pour la classification des genres musicaux

* Ahmed Sidi Mohamed
* Ranim Benhamed
* Nom de l'université
* Nom du superviseur (si applicable)
* Date de soumission

**Résumé**

Ce projet vise à classer des fichiers audios en genres musicaux à l'aide de deux modèles : un SVM basé sur des caractéristiques audios extraites et un modèle VGG19 utilisant des spectrogrammes. L'application est déployée sous forme de micro-services avec une interface utilisateur intuitive.

**Introduction**

**Contexte**

La musique occupe une place essentielle dans la vie quotidienne, influençant nos émotions, nos cultures, et nos interactions sociales. Avec l'essor des plateformes de streaming comme Spotify et YouTube, le besoin de classer efficacement de vastes bibliothèques de morceaux en genres musicaux est devenu crucial. Cette classification permet aux utilisateurs de découvrir plus facilement de nouvelles chansons et artistes correspondant à leurs préférences. De plus, la reconnaissance automatique des genres musicaux est un outil clé pour les applications de recommandation, l'analyse musicale et les études en musicologie.

**Problématique**

La classification des genres musicaux est un défi complexe pour plusieurs raisons :

1. **Subjectivité des Genres** : Les frontières entre les genres sont souvent floues, et un même morceau peut appartenir à plusieurs genres selon l'interprétation.
2. **Variabilité des Données** : Les morceaux d'un même genre peuvent avoir des caractéristiques sonores très diverses, rendant la généralisation difficile.
3. **Dimensions Temporelles** : Les genres musicaux comportent des variations temporelles qui nécessitent une analyse fine pour capturer des patterns comme les rythmes ou les mélodies.
4. **Dépendance aux Données** : La qualité et la diversité du dataset utilisé influencent fortement les performances des modèles. Les datasets limités ou biaisés peuvent conduire à des prédictions incorrectes.

**Objectifs**

Ce projet vise à résoudre ces problématiques en combinant deux approches complémentaires :

1. **Développer une Application de Classification** :
   * Construire une application capable de recevoir un fichier audio et de prédire son genre musical.
   * Fournir des prédictions basées sur deux modèles distincts : un modèle **SVM** et un modèle **VGG19**.
2. **Explorer des Méthodes d'Apprentissage** :
   * Utiliser un **SVM** basé sur des caractéristiques audios extraites (e.g., MFCCs, spectrogrammes) pour une approche classique.
   * Appliquer un modèle **VGG19** basé sur des spectrogrammes transformés en images RGB pour une approche basée sur le Deep Learning.
3. **Créer une Interface Intuitive** :
   * Concevoir un frontend interactif permettant aux utilisateurs d'interagir facilement avec les modèles et de visualiser les résultats.
4. **Évaluer et Comparer les Modèles** :
   * Mesurer les performances des deux modèles en termes de précision et de fiabilité.
   * Identifier les avantages et limitations de chaque méthode.

En fin de compte, ce projet contribue à l’amélioration des systèmes de classification musicale tout en offrant une solution pratique et évolutive.

**Dataset**

**Description du Dataset**

Le projet utilise le **GTZAN Genre Dataset**, l'un des ensembles de données les plus couramment utilisés pour la classification des genres musicaux. Ce dataset est une référence dans le domaine de la recherche en apprentissage automatique pour les tâches liées à l'audio.

* **Nom** : GTZAN Genre Dataset.
* **Contenu** :
  + **Nombre total de fichiers** : 1 000 fichiers audio.
  + **Genres** : Le dataset est réparti de manière égale en 10 genres musicaux, chacun contenant 100 fichiers :
    - *Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, Rock*.
* **Format** :
  + Les fichiers sont au format WAV.
  + Chaque fichier dure 30 secondes et est échantillonné à une fréquence de 22 050 Hz.

Le dataset offre une base solide pour entraîner et évaluer des modèles de classification musicale, mais il présente également certains défis, comme la présence de fichiers similaires dans les ensembles d’entraînement et de test.

**Prétraitement**

Avant d’entraîner les modèles, les fichiers audios subissent plusieurs étapes de prétraitement afin d’extraire des informations pertinentes et de garantir une analyse efficace :

1. **Normalisation Audio** :
   * Les fichiers audios sont normalisés pour s'assurer que les variations d'amplitude n'affectent pas la classification.
   * Cette étape garantit que les modèles se concentrent sur les patterns sonores et non sur les différences de volume.
2. **Extraction des Caractéristiques** :
   * **MFCCs (Mel-Frequency Cepstral Coefficients)**:
     + Capturent les caractéristiques spectrales essentielles de l’audio.
     + Utilisés par le modèle SVM pour l’apprentissage.
   * **Spectrogrammes** :
     + Générés à partir de l'énergie fréquentielle de l’audio.
     + Transformés en images RGB pour alimenter le modèle VGG19.
   * **Autres Caractéristiques** :
     + *Chroma* : Analyse des harmonies.
     + *Spectral Centroid*: Représente le barycentre des fréquences.
     + *Zero-Crossing Rate (ZCR)* : Compte le nombre de changements de signe dans le signal.

Ces étapes permettent de réduire la complexité des données audio tout en capturant les informations les plus pertinentes pour la classification.

**Architecture**

**Diagramme d'Architecture**

Le diagramme ci-dessous illustre l'architecture du projet, qui repose sur une approche **micro-services**. Il présente les trois principaux services et leurs interactions :

scss

Copy code

+-------------------+ +------------------+ +--------------------+

| | | | | |

| Frontend | <----> | SVM Service | <----> | VGG19 Service |

| (Streamlit) | | (Flask + SVM) | | (Flask + VGG19) |

| | | | | |

+-------------------+ +------------------+ +--------------------+

| | |

| | |

User Uploads WAV File Classify based on Classify based on

| Audio Features (MFCCs) Spectrograms

| | |

+-------------------> JSON Prediction JSON Prediction

(Genre, Confidence) (Genre, Confidence)

**Description des Composants**

1. **Frontend (Streamlit)** :
   * **Rôle** : L'interface utilisateur est construite avec **Streamlit**. Elle permet à l'utilisateur de télécharger un fichier audio au format WAV.
   * **Fonctionnalités** :
     + **Téléchargement de Fichiers** : L'utilisateur peut facilement télécharger un fichier WAV depuis son appareil.
     + **Affichage des Résultats** : Après le traitement du fichier audio, le frontend affiche les résultats des deux modèles de classification (SVM et VGG19), y compris les genres prédits et les scores de confiance.
     + **Visualisation** : Le frontend génère également un graphique représentant l'onde sonore de l'audio téléchargé.
2. **SVM Service** :
   * **Rôle** : Ce service est basé sur un modèle **Support Vector Machine (SVM)** qui prédit le genre musical à partir des caractéristiques extraites de l'audio.
   * **Fonctionnement** :
     + **Extraction des Caractéristiques** : Les caractéristiques comme les **MFCCs**, **Chroma**, et **Spectrogrammes** sont extraites à partir du fichier audio.
     + **Prédiction** : Le modèle SVM utilise ces caractéristiques pour prédire le genre du morceau de musique.
   * **Technologies** :
     + **Flask** : Pour l'API REST.
     + **Scikit-learn** : Pour l'implémentation du modèle SVM.
     + **Librosa** : Pour le prétraitement audio.
3. **VGG19 Service** :
   * **Rôle** : Ce service utilise un modèle **VGG19** pré-entraîné pour effectuer des classifications en fonction des spectrogrammes d'audio transformés en images RGB.
   * **Fonctionnement** :
     + **Création de Spectrogrammes** : À partir des fichiers audios, des spectrogrammes sont générés en utilisant **Librosa** et convertis en images de type RGB.
     + **Prédiction** : Le modèle VGG19 prédit ensuite le genre musical en utilisant ces spectrogrammes.
   * **Technologies** :
     + **Flask** : Pour l'API REST.
     + **TensorFlow/Keras** : Pour le modèle VGG19.
     + **Librosa** : Pour la génération des spectrogrammes.

**Flux de Données entre les Services**

1. L'utilisateur télécharge un fichier audio au format WAV via le **Frontend**.
2. Le fichier est encodé en **Base64** et envoyé simultanément aux services **SVM** et **VGG19**.
3. Chaque service reçoit les données, effectue la classification, et retourne un **JSON** avec :
   * **Genre prédit**.
   * **Score de confiance**.
4. Le **Frontend** affiche les résultats pour les deux services afin que l'utilisateur puisse comparer les prédictions.

**Résultats**

**Précision des Modèles**

Les modèles **SVM** et **VGG19** ont été évalués sur l'ensemble de test du **GTZAN Genre Dataset**. Les résultats montrent des performances intéressantes pour chaque modèle, avec des différences notables en termes de précision et de fiabilité.

1. **SVM (Support Vector Machine)** :
   * **Précision** : Le modèle SVM, entraîné sur des caractéristiques extraites comme les **MFCCs** et les **spectrogrammes**, a obtenu une précision moyenne de **70%** sur l'ensemble de test.
   * **Avantages** : Rapidité de traitement, moins coûteux en termes de ressources, et adapté aux jeux de données plus petits.
   * **Limites** : Peut être limité par la qualité des caractéristiques extraites et peut rencontrer des difficultés avec des genres musicaux très similaires.
2. **VGG19 (Convolutional Neural Network)** :
   * **Précision** : Le modèle VGG19, utilisant des spectrogrammes transformés en images, a atteint une précision moyenne de **85%** sur l'ensemble de test.
   * **Avantages** : Excellente capacité de généralisation grâce à l'utilisation d'un modèle de Deep Learning et des spectrogrammes qui capturent mieux les informations complexes des morceaux.
   * **Limites** : Consommation de ressources plus élevée et nécessite un matériel adéquat pour le traitement, notamment un GPU pour un entraînement et une inférence plus rapide.

**Comparaison des Deux Modèles (SVM vs VGG19)**

|  |  |  |
| --- | --- | --- |
| **Critère** | **SVM** | **VGG19** |
| **Précision** | 70% | 85% |
| **Vitesse de Prédiction** | Rapide (environ 0.5s par fichier) | Plus lent (1-2s par fichier) |
| **Complexité** | Faible (modèle classique) | Élevée (réseau de neurones profond) |
| **Utilisation des Ressources** | Faible | Élevée (utilisation du GPU recommandée) |

**Conclusion** : Le modèle **VGG19** surpasse le modèle **SVM** en termes de précision, ce qui est attendu en raison de la capacité de VGG19 à capturer des informations plus complexes à partir des spectrogrammes. Cependant, le modèle **SVM** reste compétitif pour des déploiements plus légers et rapides, avec des ressources limitées.

**Exemple Visuel**

Voici un exemple de l'interface **Frontend** où les utilisateurs peuvent voir les résultats des prédictions des deux modèles :

*L'exemple ci-dessus montre les résultats d'une chanson testée. Les utilisateurs peuvent comparer les genres prédits et les scores de confiance des deux modèles.*

Dans cet exemple :

* Le **SVM** a prédit le genre **Rock** avec un score de confiance de **72%**.
* Le **VGG19** a également prédit le genre **Rock** mais avec un score de **91%**.

Le frontend fournit ainsi une interface interactive permettant à l'utilisateur de vérifier la fiabilité des prédictions de chaque modèle et de choisir celui qui lui convient le mieux.

A screenshot of a computer

Description automatically generated

**Conclusion**

**Résumé des Contributions du Projet**

Ce projet a pour objectif principal de développer une application de **classification des genres musicaux** en utilisant deux modèles d'apprentissage automatique distincts : un modèle **Support Vector Machine (SVM)** et un modèle **VGG19** basé sur le Deep Learning. L'application est composée de trois services principaux :

1. **Frontend** : Interface utilisateur basée sur **Streamlit** permettant le téléchargement de fichiers audio (format WAV) et l'affichage des résultats.
2. **SVM Service** : Service Flask qui prédit les genres musicaux à partir des caractéristiques extraites des fichiers audio (telles que les **MFCCs**, les **chroma** et les **spectrogrammes**).
3. **VGG19 Service** : Service Flask utilisant un modèle de deep learning **VGG19** pour classer les genres musicaux à partir de spectrogrammes transformés en images.

Les résultats des deux modèles ont été comparés sur un ensemble de test et ont montré que le modèle **VGG19** offrait une meilleure précision, mais que le modèle **SVM** était plus rapide et plus léger, offrant ainsi un compromis utile pour des environnements à faible consommation de ressources.