Note méthodologique : preuve de concept

Dataset retenu

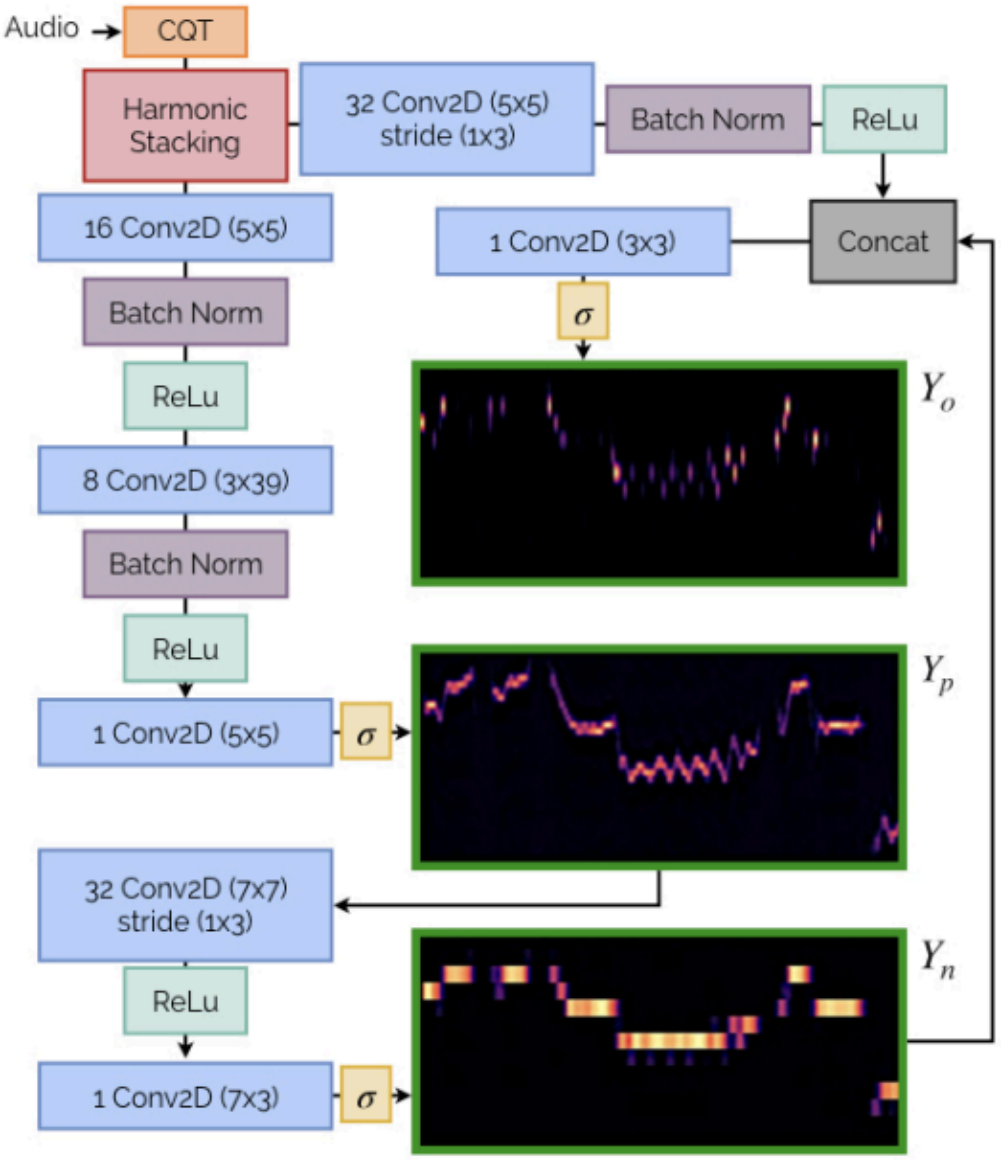
Le dataset choisi pour ce projet est **MAESTRO** (MIDI and Audio Edited for Synchronous Tracks and Organization). Ce jeu de données comprend environ 200 heures d'enregistrements de performances pianistiques virtuoses, capturées avec un alignement précis (~3 ms) entre les annotations MIDI et les formes d'onde audio. Les données ont été collectées sur une période de dix ans lors de l'International Piano-e-Competition, où les pianistes jouent sur des pianos Yamaha Disklavier. Ces instruments, en plus d'être des pianos acoustiques de concert, sont équipés d'un système intégré de capture et de reproduction MIDI de haute précision. Les fichiers MIDI incluent des informations détaillées telles que les vélocités des frappes de touches et les positions des pédales (sustain, sostenuto, una corda). Les fichiers audio et MIDI sont alignés avec une précision d'environ 3 ms et segmentés en pièces musicales individuelles, annotées avec le compositeur, le titre et l'année de la performance. Les enregistrements audio non compressés sont de qualité CD ou supérieure (44,1–48 kHz, 16-bit PCM stéréo). Une configuration de division en ensembles d'entraînement, de validation et de test est également proposée, garantissant qu'une même composition, même interprétée par plusieurs participants, n'apparaît pas dans plusieurs sous-ensembles. Le répertoire est principalement composé de musique classique, englobant des compositeurs du XVII^{e} au début du XX^{e} siècle. Pour plus d'informations sur la création de ce dataset et ses applications, veuillez consulter l'article où il a été introduit : Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. .

Les concepts de l’algorithme récent

L'algorithme récent choisi pour cette étude est **Basic Pitch**, développé par Spotify. Il s'agit d'un modèle léger de transcription musicale automatique, capable de traiter des signaux polyphoniques et de généraliser sur une large gamme d'instruments, y compris la voix humaine. Le modèle est conçu pour prédire simultanément les onsets, les hauteurs multiples (multipitch) et les activations de notes, ce qui améliore la précision de la transcription. Cette approche multi-sortie permet d'obtenir une meilleure estimation des notes au niveau des frames. De plus, Basic Pitch est optimisé pour fonctionner en temps réel sur des appareils à faibles ressources, grâce à sa faible consommation de mémoire et à son efficacité de traitement. Les résultats expérimentaux montrent que, malgré sa simplicité, Basic Pitch offre des performances comparables aux systèmes de transcription musicale spécialisés plus complexes. Cette combinaison de légèreté, de polyvalence instrumentale et

de haute précision rend Basic Pitch particulièrement adapté aux applications de transcription musicale en temps réel et sur des plateformes aux ressources limitées.

Architecture du modèle



Basic Pitch utilise une architecture de **réseau de neurones convolutifs (CNN)** optimisée pour la transcription musicale. Son pipeline de traitement se divise en plusieurs étapes clés :

1. **Extraction des caractéristiques audio** : Utilisation de spectrogrammes de type Constant-Q Transform (CQT) pour capturer les caractéristiques musicales.
2. **Modélisation multi-tâches** : Le modèle prédit simultanément trois sorties :

Présence de notes : Détecte si une note est jouée.

Détection d’onsets : Identifie les débuts des notes.

Suivi de la hauteur tonale : Suit l'évolution des fréquences.

3. **Post-traitement** : Filtrage et ajustement des sorties pour produire une représentation MIDI plus propre et fidèle.

La modélisation

Méthodologie

1. **Prétraitement des données** :

   Chargement du dataset MAESTRO et extraction des fichiers audio et MIDI.

   Création d'un dataframe depuis maestro-v3.0.0.csv afin d'avoir le chemin de tous les fichiers cibles.

   Conversion des fichiers audio en spectrogrammes avec librosa.

   Normalisation et segmentation en fenêtres temporelles.

2. **Application des modèles** :

   Basic Pitch :

   Utilisation de la fonction *predict(audio_path)*, qui génère des sorties MIDI à partir des fichiers audio.

   Génération et enregistrement des fichiers MIDI prédits.

   Enregistrement des chemins des nouveau fichier sur le dataframe

   Comme chaque export est traiter en itérant dans le dataframe, et qu’on ne prédit pas un fichier déjà existant, on peut facilement stopper et reprendre plus tard

   Melodia (baseline) :

   Extraction des pitches dominants via une analyse spectrale heuristique utilisant une commande système qui charge le plugin Melodia, et un logiciel de traitement audio (sonic-annotator equivalent command line de audacity).

   Conversion en format MIDI sans ajustement temporel avancé.

   Enregistrement des chemins des nouveau fichier sur le dataframe

   Comme chaque export est traiter en itérant dans le dataframe, et qu’on ne prédit pas un fichier déjà existant, on peut facilement stopper et reprendre plus tard

Extraction des notes dans un df

Calcul de score a partir des mes dataframe imbriquer comprenant un dataframe pour chaque midi et une colonne par metric calculer à partir de ceux-ci.

Métriques d’évaluation

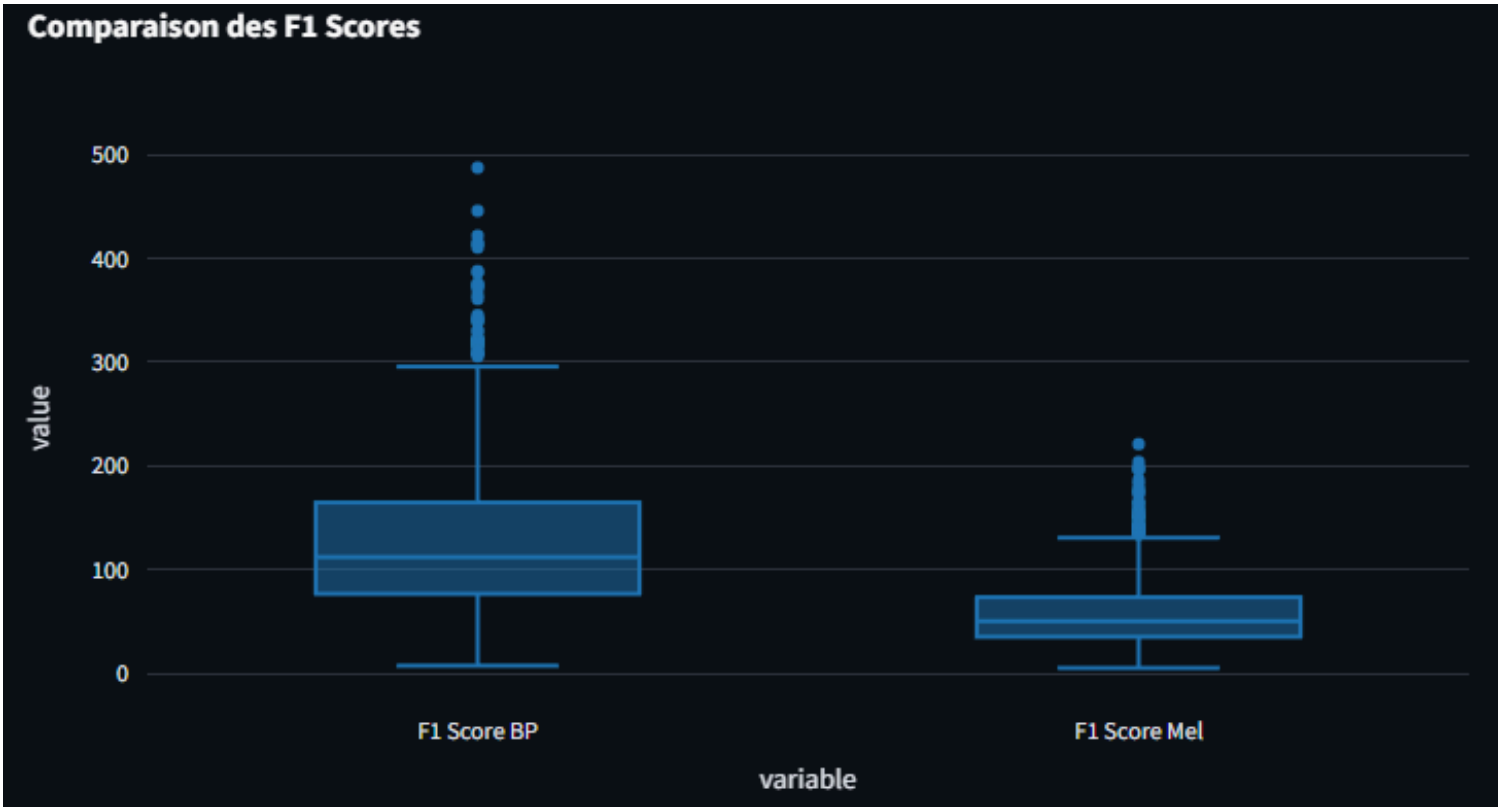
Les performances des modèles ont été comparées à l'aide des métriques suivantes :

F1-score sur l'alignement MIDI entre la prédiction et la vérité terrain.

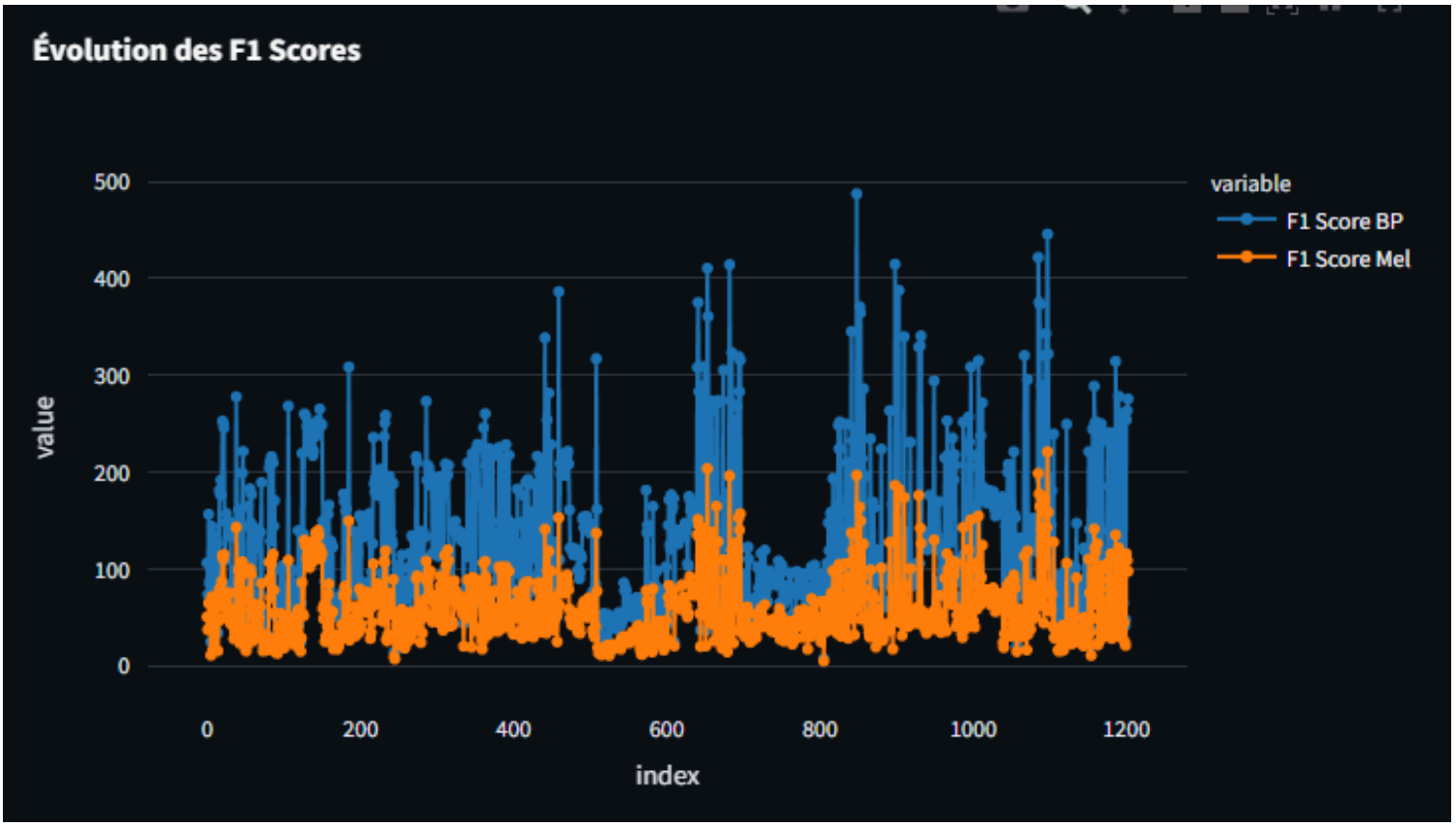
Notes Correct, mesurant la précision de l'extraction des notes.

Une synthèse des résultats

On voit une nette amélioration de manière générale au niveau du score F1, cependant on constate que le minimum est toujours le même. On peut se poser la question suivante, est-ce que le modèle est mauvais sur certaines musiques, ou ces musiques sont juste trop courtes pour avoir un score élevé, et l'augmentation de la médiane nous suffit a dire que le modèle est meilleur, pour aucun morceau basic pitch est moins bon.



Pour cela on peut comparer chaque morceau unitairement, sans passer par un globalisation via une boîte à moustache, mais comparer des courbes



On peut donc voir que chacune des courbes suit à peu près le même chemin avec une amélioration globale pour basic pitch.

Maintenant comme cela reste de la musique, et que nous voulons juger la musique, le mieux reste encore d'écouter certains individu afin de voir si on peux ressentir l'interprétation via les variance de vélocité dans les notes

L'analyse de la feature importance globale et locale du nouveau modèle

Pour ce projet nous analysons des fréquences audio provenant d'une musique, ce qui veut dire que nous ne pouvons pas réellement faire de la sélection de feature, chaque fréquence est importante car elle relève de l'enregistrement audio, et en sélectionner c'est perdre une partie des notes et donc perdre en cohérence.

Les limites et les améliorations possibles

Les limites de ce type de modèle reste la complexité du son. Même si le modèle est très performant pour la polyphonie notamment face a melodia qui est incapable de traiter plus d'une note a la fois, si on commence a lui fournir une cacophonie, il ne pourra certainement pas en extraire toutes les notes. Même si cela reste une limite technique, est-ce que nous voulons donner une mélodie ignoble a un modèle qui a pour but de retranscrire des partition pour des musiques écoutables.

Autres point, le respect des temps est parfois suffisamment proche pour être juste d'un point de vue metrics, mais extrêmement peux décaler pour que cela choque l'oreil et que cela sonne faux

Un second modèle spécialisé dans le recalage minime des notes dans le temps permettra de changer de quelques microsecondes les notes et permettra un amélioration significative des résultats sans que les metrics soit impacté.