

Rapport intermédiaire

DAHMANI Sonia, SANTOS DUARTE Elsa

18.02.2022

1 Présentation générale

Le but de ce projet est de convertir des images, plus précisément des tableaux et des peintures en mélodies. Les tableaux dont il est question pour le moment, sont de types abstraits. Au fil de l'avancement du projet, nous envisageons d'ajouter des tableaux de types variés, notamment des portraits. Pour ce faire, le code est principalement divisé en deux parties. D'une part, du code python avec la librairie OpenCV, permettant l'analyse d'images et leur traduction en données formelles. Et d'autre part, du code java, récupérant ces données formelles et les transformant en code *Sonic-pi* pour la création de mélodies.

1.1 Analyse d'images et production de données formelles

Cette partie concerne l'analyse de peintures, pour le moment de types abstraits. On utilisera des algorithmes de Machine Learning ainsi que des bibliothèques de traitement d'images afin d'extraire des informations qui seront ensuite utilisées dans la deuxième partie : la traduction des données en mélodie. Les données auxquelles on s'intéresse sont les couleurs ainsi que les formes. Chacune ayant une signification différente dans la production de la mélodie. On pourra mettre en place un modèle de réseaux de neurones qui sera entraîné à reconnaître des objets dans l'image.

1.2 Traduction de données formelles en mélodie

Dans cette partie on souhaite convertir les données formelles telles que les entiers rgb, les formes ou encore leur taille en mélodie. On doit donc établir une correspondance de l'une à l'autre. Les couleurs servent à la hauteur de note : la gamme est traduite depuis la couleur dominante, les accords successifs depuis les couleurs des formes, et le rythme depuis les différentes formes.

2 Réalisé jusqu'ici

Pour l'analyse d'image, les librairies utilisées sont: Sklearn pour la partie Machine Learning, Numpy pour la représentation vectorielle, Pandas, matplotlib.pyplot..etc

pour le traitement d'images en python. On traite l'image sous la forme d'un vecteur, par exemple: `array(2567, 4608, 3)` les deux premières valeurs représentent le pixel sur l'image et la troisième est la couleur. On appliquera au vecteur l'algorithme de partitionnement en k-moyennes: c'est une méthode de partitionnement de données qu'on utilisera pour la segmentation de l'image et la définition d'une couleur principale pour chaque segment. Le partitionnement se repose sur le regroupement des pixels les plus proches en couleurs. Pour appliquer l'algorithme, on devra convertir d'abord notre image en image RVB. Elle est sous forme tridimensionnelle, on la remodèle en tableau bidimensionnel. La prochaine étape est donc d'identifier les couleurs de l'image et d'afficher les couleurs principales sous forme de liste ou de vecteur. On utilise une représentation en HEX des valeurs RVB afin de construire comme un dictionnaire de couleurs que l'algorithme utilisera ensuite pour identifier les couleurs extraites du tableau (un vecteur de string). Ensuite, pour l'extraction de la couleur, la méthode prend en entrée l'image ainsi que le nombre maximum de couleurs que l'on veut en extraire. L'algorithme K-partitionnement attend en entrée un tableau bidimensionnel. L'algorithme formera à la fin des clusters qui seront donc nos couleurs principales. A la sortie, les valeurs extraites dans les clusters sont comparées à celle du dictionnaire pour identifier les couleurs. Quant à la traduction des données formelles en mélodie, la première étape a été d'établir une correspondance couleur/note. Dans un premier temps, et pour faciliter les correspondances, les couleurs ont été réduites en intervalle rgb tel que $r, g, b \in [0, 2]$. Ainsi, si r, g ou b sont compris entre 0 et 85, ils seront interprétés comme 0, s'ils sont compris entre 85 et 170, ils sont interprétés comme 1 et s'ils sont compris entre 170 et 255, ils sont interprétés comme 2. Cette première correspondance donne un squelette pour de potentielles améliorations d'analyse. On détermine ainsi la gamme mineure ou majeure selon la froideur ou la chaleur de la couleur dominante, et le sample de fond est pour l'instant déterminé sur les formes reconnues. La longueur du sample de fond est décidée selon la taille des formes qui se succèdent dans l'image.

3 Difficultés rencontrées

Le projet étant étendu sur deux principales parties, les premières difficultés sont rapidement venues de pair. D'une part, la difficulté d'analyse d'images.

La première étape était de trouver une bibliothèque d'images suffisamment vaste pour pouvoir reconnaître les données nécessaires au projet, pour l'instant les couleurs. La seconde difficulté était de trouver un moyen efficace d'analyser les images : soit en réduisant le nombre de leurs couleurs, soit en réduisant directement leur qualité. D'autre part de l'analyse d'image, il y a eu les difficultés liées à la production musicale.

La première d'entre elles était de trouver un langage de sortie permettant l'écriture de la musique. Dans un premier temps, les bibliothèques de traitement midi sous java paraissaient une bonne solution. Puis, au fur et à mesure, le langage *Sonic-pi* nous est apparu comme plus intuitif et adapté à la traduction de

données formelles en musique. La seconde difficulté concernant la traduction de données formelles en musique, a été et est toujours de trouver une représentation du rythme. Ce qu'il nous paraissait à priori le plus logique était de se servir des formes reconnues lors de l'analyse d'image et de les convertir en rythme.

4 Etapes futures

Les prochaines étapes sont de pouvoir identifier des formes en plus des couleurs, ainsi que leurs tailles, pour établir les proportions. Et le but serait, finalement, de traiter des tableaux d'un autre type comme les portraits. On pourrait même rajouter, plus tard si l'on a le temps, une partie où l'algorithme devra reconnaître lui-même de quel type de tableau il s'agit.

Il faudra également établir une traduction du rythme qui soit cohérente aux règles musicales à partir de données formelles, quitte à y ajouter une part d'aléa. Et de façon générale, d'étendre la part d'utilisation des données formelles dans la production musicale. Aussi bien dans le rythme, que dans les variations musicales : les notes, l'écho, la réverbération, ou la longueur de notes.