INF8801A – Applications multimédias

Automne 2021

Mini-rapport de projet

Groupe 01

1954495 – Philippe Savard

2166333 – Gaspard Petitclair

Soumis à : Lama Séoud

22 Octobre 2021

Présentation du sujet

Les émotions jouent un rôle essentiel dans la communication des êtres humains. Intuitivement, le cerveau nous permet de capturer les émotions des autres et ainsi nous adapter en conséquence. Cette habileté innée chez l'Homme n'est toutefois pas une tâche triviale pour une machine. Malgré l'aide de l'intelligence artificielle, les implémentations modernes de la détection d'expression faciale ne sont pas parfaites et ont des difficultés à donner des réponses décisives [1]. L'article choisi, *Real-Time Facial Emotion Recognition System With Improved Preprocessing and Feature Extraction*, présente une méthode améliorée de détection des émotions (expression faciale) en temps réel. Celle-ci combine l'approche traditionnelle par réseaux de neurones convolutifs avec les descripteurs de points clés du visage et les histogrammes d'orientation de gradients. Notre objectif pour ce projet sera de réimplémenter intégralement la méthodologie présentée dans cet article. L'application présentée à la fin du projet devra être en mesure de détecter avec bonne précision l'émotion ressentie par l'usager devant la caméra.

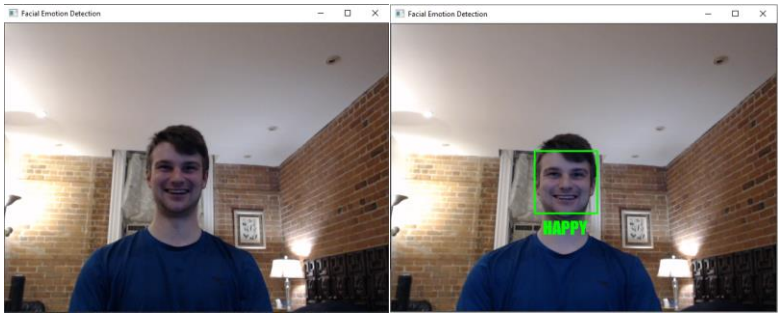

Figure 1.1 - Exemple de détection de la joie

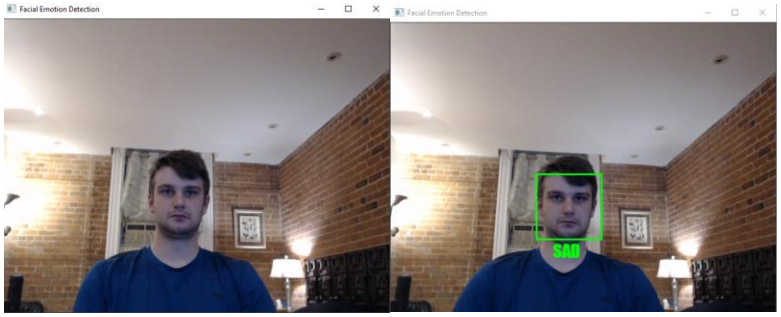

Figure 1.2 - Exemple de détection de la tristesse

Les figures 1.1 et 1.2 sont des exemples de résultats attendus une fois l'application complétée.

Méthodologie

La méthodologie présentée dans l'article est décomposée en quatre étapes de traitement. Dans cette section, nous présenterons sommairement ces étapes et donnerons une représentation schématique du processus (pipeline).

1. Détection du visage

La première étape du traitement consiste à détecter les visages devant la caméra. L'application prend en entrée un flux vidéo continu en provenance d'une caméra vidéo (webcam). Ce flux est alors traité afin de détecter le ou les visages qui apparaissent dans la scène filmée. La détection est effectuée à l'aide d'un classifieur en cascade LBP (Local Binary Patterns) considérant l'image comme une composition de micromotifs [2].

2. Prétraitement de l'image

La seconde étape du traitement regroupe toutes les manipulations effectuées à l'image avant l'extraction des caractéristiques (voir étape 3). Le prétraitement consiste à rogner uniquement les visages de l'image, les redimensionner à la dimension désirée et normaliser les intensités [3]. Les détails de l'implémentation ne seront pas expliqués ici, mais l'essentiel est que nous souhaitons retirer le plus d'informations superflues possible présentes dans l'image initiale.

3. Extraction des caractéristiques

La troisième étape du traitement est l'extraction des caractéristiques du visage à partir de l'image en sortie de l'étape précédente. L'extraction des caractéristiques se fait en deux étapes : obtenir les histogrammes d'orientations du gradient et calculer les points clés du visage. Puisque les HOG opèrent sur des régions locales, ils sont invariants aux transformations géométriques (rotation, mise à l'échelle, etc.). Ils sont également différents pour chaque expression que nous souhaitons détecter. Les points clés permettent ensuite de réduire l'étendue des données traitées en sélectionnant les portions du visage essentielles à la détection de l'expression. Une fois l'extraction effectuée, nous obtiendrons une image similaire à celle présentée dans l'article à la figure 2.1. Il est à noter que les points clés sont mesurés à l'aide de la librairie DLib disponible sous Python.

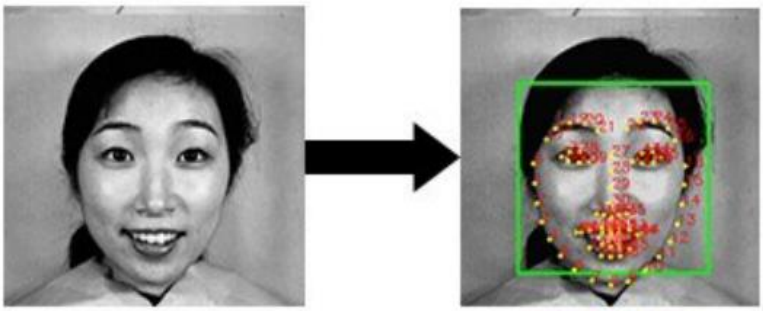

Figure 2.1 - Sortie du traitement de DLib [4]

4. **Classification**

La quatrième et dernière étape consiste à entraîner un réseau de neurones convolutifs (CNN). Celui-ci prend en entrée les caractéristiques d'une image (extraites à la partie 3) et renvoie en sortie une classe correspondant à une des 7 émotions utilisées pour cette expérience. Les 7 émotions sont les suivantes : la joie, la peur, la colère, le dégoût, la tristesse, la surprise et la neutralité. Toutes les images d'entraînement pour le CNN seront tirées de la base de données publiques FER2013 (https://www.kaggle.com/msambare/fer2013).

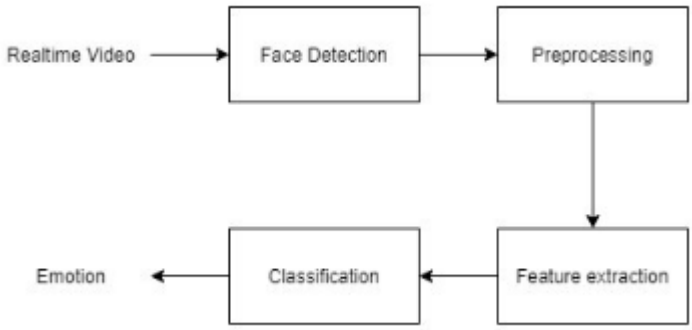

Figure 2.2 - Pipeline de la méthodologie implémentée [5]

Détails d'implémentation

L'objectif de ce projet est de réimplémenter la méthodologie présentée à la section précédente. Nous allons donc implémenter la détection du visage, le prétraitement, l'extraction de caractéristiques et la classification afin d'être en mesure de répondre aux attentes illustrées aux figures 1.1 et 1.2. Dans cette section nous discutons des détails de l'implémentation réalisée ainsi que les particularités de celle-ci.

1. **Multi-block Local Binary Patterns (MB-LBP)**

Puisque l'implémentation de la détection du visage par classifieur en cascade LBP n'est pas spécifiée, nous allons utiliser le cadriciel de Skimage pour la détection d'objets. Celui-ci utilise un classifieur en cascade avec les fichiers de Skimage.data entraînées en utilisant MB-LBP. Cela nous permettra de faire la détection de(s) visage(s) captés par la webcam.

2. **Recadrage et redimensionnement**

Pour cette partie, l'article ne détaille pas la méthode employée. Nous allons donc nous baser sur le module OpenCV de Python qui est capable de placer des marqueurs sur les

différentes zones du visage (contour, bouche, yeux et nez). Une fois que nous aurons accès à ces positions, nous utiliserons ces dernières normaliser le recadrage.

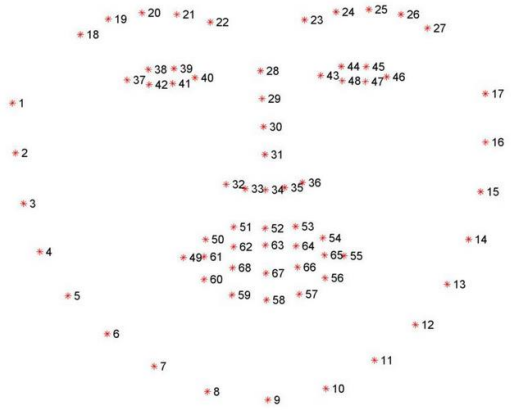

Figure 3.1 - Points clés d'un visage selon DLib [6]

Nous avons imaginé cette méthode sans la tester. Si nous voyons que cette méthode ne fonctionne pas, nous chercherons une autre façon de recadrer correctement l'image autour du visage.

3. Normalisation d'intensité

Pour la normalisation d'intensité, les auteurs de l'article mentionnent avoir utilisé la normalisation MinMax. Nous allons donc nous baser sur cette même méthode.

4. Extraction du HOG

Pour l'extraction du HOG, nous allons utiliser la librairie Skimage prévue à cet effet.

5. Extraction des facial landmarks

L'extraction des marqueurs faciaux se fera avec la librairie DLib de OpenCV, comme mentionné dans la partie 2 ci-dessus [7].

6. Classification

Tel que mentionné ci-haut, toutes les images d'entraînement pour le CNN seront tirées de la base de données publique FER2013. Nous allons faire dans un premier temps une classification entre les émotions "happy" et "sad" afin de simplifier le problème, et nous allons ensuite essayer d'augmenter le nombre d'émotions pour utiliser les 7 émotions de la

base de données. Pour ce qui est du réseau de neurones, nous allons utiliser un réseau de neurones CNN tel que paramétré dans l'article, avec comme paramètres les données brutes de l'image, les données extraites de HOG et les facial landmarks.

Échéancier

Le projet sera réalisé sur une période de 5 semaines (excluant la semaine de rédaction du mini-rapport). Dans cette section nous détaillons la distribution des tâches sur le temps alloué.

Semaine	Objectifs
1	Détection du visage à l'aide de MB-LBP. Normalisation des intensités.
2*	Recadrage et extraction des caractéristiques du visage (points clés et HOG).
3	Classification à deux états (création du CNN et début de l'entraînement)
4	Classification à sept états. Ajout des fonctionnalités manquantes (interface, etc.).
5	Analyse des résultats et préparations de la présentation orale.

** Puisque l'implémentation du recadrage n'est pas spécifiée dans l'article, le risque associé à notre objectif est plus élevé que celui des autres semaines. Par conséquent, un débordement de la semaine 2 sur la semaine 3 est envisageable.*

Références

[1] [2] [3] [4] [5] A. John, A. MC, A. S. Ajayan, S. Sanoop and V. R. Kumar, "Real-Time Facial Emotion Recognition System With Improved Preprocessing and Feature Extraction," *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2020, pp. 1328-1333, doi: 10.1109/ICSSIT48917.2020.9214207.

[6] [7] Adrian Rosebrock. (Avril 2017). *Detect eyes, nose, lips, and jaw with dlib, OpenCV, and Python.* pyimagesearch. https://www.pyimagesearch.com/2017/04/10/detect-eyes-nose-lips-jaw-dlib-opencv-python/