Évaluation et comparaison de méthodes et modèles d'analyse d'expressions faciales

Rapport projet Traitements d'Images

Master M1

Rémy AULOY
Mohamed Amine DAHMOUNI
Boris DINH
Maxime DUPONT
Quentin ISOARD
Antony MADALENO
Christian TOMASINO

Année universitaire 2022 - 2023

Université de Bourgogne

Glossaire

CNN Convolutional neural network, réseau de neurones convolutif. 5, 6, 10

dataset Jeu de données utilisé par l'algorithme soit pour l'apprentissage, soit pour les phases de test. 5, 7–9, 14–16

FER Facial Expression Recognition, reconnaissance d'expressions faciales. 4, 5, 8

fine-tuning Ajuster un modèle d'IA pour répondre à un besoin spécifique. 4, 7, 9

IA Acronyme d'intelligence artificielle. 4, 8

LSTM Long Short Term Memory, réseaux récurrents à mémoire court et long terme. 6

Machine Learning Branche de l'intelligence artificielle spécialisée dans la création d'algorithmes basés sur l'apprentissage. 5

RNN Recurrent neural network, réseau de neurones récurrents. 5, 6

Table des matières

Remerciements

Nous tenons à remercier notre tuteur académique, M. Yannick Benezeth, pour son suivi pédagogique et son aide tout au long de la réalisation de ce projet.

Nous remercions également Mmes Céline Roudet et Stéphanie Bricq pour nous avoir donné accès au centre de calcul de l'Université de Bourgogne afin de profiter de la puissance de calcul pour l'entraînement de modèles.

Pour finir, nous tenons à remercier l'Imperial College London pour nous avoir donné accès à leur base de données d'expressions faciales.

Introduction

1 Contexte et objectif du projet

La reconnaissance faciale est un sujet d'étude qui ne cesse pas de s'améliorer. Depuis que l'intelligence artificielle est devenue un outil très populaire et surtout accessible, nombreuses sont les institutions et les entreprises qui ont commencé à intégrer ces technologies pour leurs besoins. Une des exploitations les plus récurrentes de la reconnaissance faciale est la détection des expressions et plus précisément des émotions. L'idée est de créer des algorithmes capables de reconnaître une émotion à partir d'une image ou d'une séquence d'images, donc une vidéo.

Il existe de nombreuses façons de détecter une émotion à l'aide de l'intelligence artificielle. Le choix d'implémentation dépend de différents facteurs qui varient en fonction des besoins et des problématiques à résoudre. Les contraintes ne sont évidemment pas les mêmes sur une image statique, sur une vidéo de quelques secondes ou encore sur un flux continu comme une webcam. Sur internet, il est possible de trouver de nombreux modèles capables de détecter des expressions faciales en fonction du contexte.

Pour ce projet, nous avons comme objectif d'implémenter un programme qui va prendre en entrée le flux vidéo de la webcam et détecter en temps réel l'émotion d'un visage capturé par le flux. Dans ce rapport nous allons vous expliquer la démarche qui nous a amené à choisir un modèle plutôt qu'un autre, vous expliquer comment nous avons appréhendé le modèle et fait du fine-tuning pour répondre à notre besoin. Nous vous présenterons notre analyse et des comparaisons de performance et d'efficacité. Puis, nous conclurons avec un ressenti sur cette première expérience avec l'intelligence artificielle et la reconnaissance faciale.

Avant de commencer, nous trouvons tout de même important de faire une introduction sur l'IA de manière générale et sur les méthodes de FER, pour nous assurer que quiconque lise ce rapport puisse s'y retrouver.

2 L'intelligence artificielle et le FER

L'intelligence artificielle est une branche du domaine scientifique et informatique qui permet de développer des algorithmes particuliers. En restant dans le domaine du traitement d'images, ces algorithmes sont capables d'analyser les images et de produire des conclusions probabilistes. Par exemple, déterminer si le sujet d'une image fait partie ou non d'une catégorie d'objets.

Pour réaliser cet exploit, il faut nécessairement identifier les informations permettant de faire des choix. Les informations à traiter varient selon le contexte. Dans le cas du traitement d'images, ce sont les pixels qui jouent le rôle de source d'informations. À travers les valeurs des pixels qui constituent l'image, il est possible de fournir une description qui va permettre de l'analyser, de la comparer à d'autres images et de la classifier.

La description peut se faire de différentes manières. En effet, identifier le bon descripteur pour décrire les images est une étape fondamentale, car cela va fortement influer sur la classification. Plus l'algorithme a une idée claire de comment décrire une image, plus il sera précis dans ses prédictions lors de la classification. Dans les algorithmes d'IA, les descripteurs sont souvent le fruit de ce que l'on appelle la phase

d'apprentissage. Dans ce cas, nous entrons dans le monde du Machine Learning, une branche bien connue de l'intelligence artificielle.

Il existe globalement deux familles d'apprentissage. L'apprentissage supervisé et l'apprentissage non supervisé. Dans les deux cas, l'algorithme a besoin d'un dataset avec lequel pouvoir s'entraîner. La différence entre les deux est que dans l'apprentissage supervisé, le dataset pour l'entraînement est accompagné d'une classification déjà effectuée, alors que dans le deuxième ce n'est pas le cas. Avec un apprentissage supervisé, on permet à l'algorithme de perfectionner ses paramètres tout au long de l'apprentissage en lui donnant la réponse que l'on souhaite avoir pour chaque image. Tandis qu'avec l'apprentissage non supervisé nous forçons l'algorithme à se perfectionner de façon autonome, erreur après erreur.

Pour la reconnaissance d'expressions faciales, l'apprentissage se fait dans la plupart des cas en mode supervisé. Le modèle d'AI est entraîné donc sur un dataset ayant déjà l'expression du visage identifiée. Pour construire un modèle pour le FER, il est très courant d'utiliser des modèles dits "réseaux de neurones". Globalement, l'idée est de séparer les traitements en plusieurs couches consécutives. Chaque couche est composée de plusieurs "neurones", qui effectuent des opérations avec les résultats obtenus par les couches précédentes et les transmettent à la couche suivante.

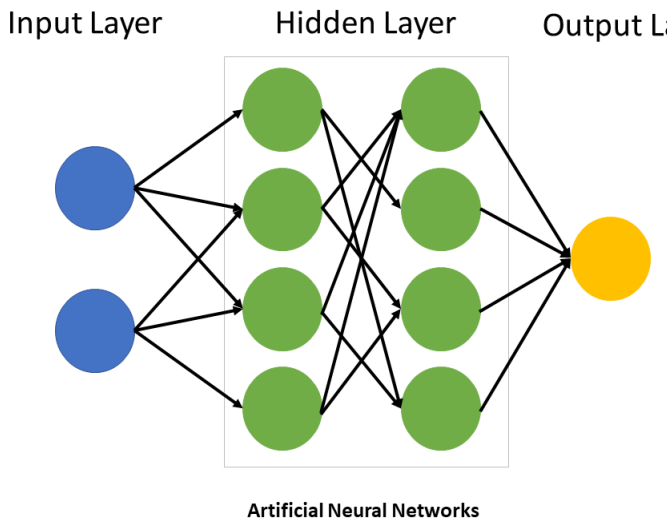

FIGURE 1 – Exemple simplifié d'un réseau de neurone

Avec les images, ou encore des flux d'images, avant d'arriver au réseau "classique", on utilise des CNN, un réseau de neurones à convolution, ou des RNN, un réseau de neurones récurrents. Les CNN sont des réseaux particuliers qui sont composés de plusieurs couches sur lesquelles deux phases se succèdent : la phase de convolution et la phase de mise en commun (polling). Dans la phase de convolution, l'image est traitée localement, donc chaque pixel va subir un traitement basé sur une fonction de convolution. La phase de polling qui succède la phase de convolution va permettre une mise en commun et de ce fait une reconstruction partielle ou totale de l'image. Il est possible d'enchaîner plusieurs couches de convolutions et de polling.

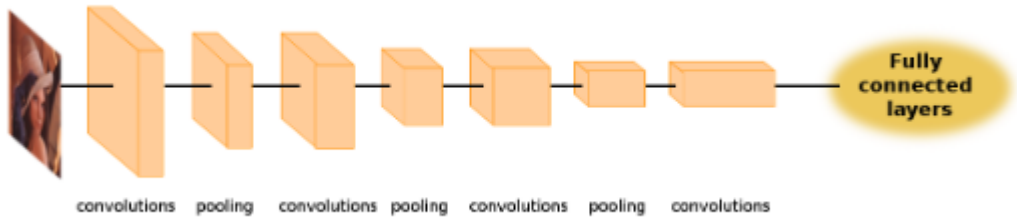

FIGURE 2 – Exemple d'architecture d'un réseau CNN

Les CNN permettent d'effectuer un prétraitement de l'image afin d'arriver au réseau de neurones s'occupant de la classification avec des données mieux adaptées aux calculs successifs. Pour les cas où on traite des images statiques, les CNN peuvent amplement suffire pour le modèle, et avec des bonnes architectures deviennent très efficaces aussi pour des cas avec des flux vidéos. Les réseaux RNN sont un type particulier de réseaux qui s'appuie sur des architectures LSTM, lesquelles, pour chaque neurone, associent un état qui joue le rôle de mémoire. Les réseaux RNN contiennent au moins un cycle dans leur structure, ce qui le rend en conséquence non linéaire.

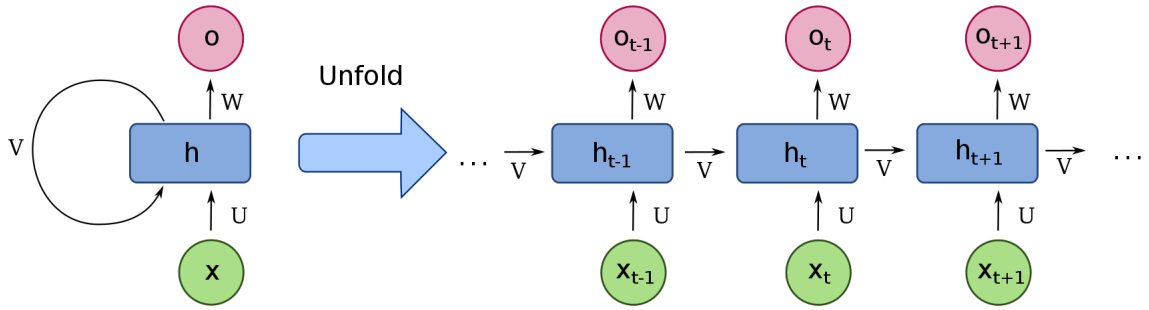

FIGURE 3 – Exemple d'un RNN et sa décomposition sur un réseau de neurones classique

Grâce à ces différents types de réseaux de neurones, il est possible de travailler à la fois sur des images statiques que sur des flux vidéo, ce qui rend la détection d'expression faciale applicable dans différents contextes. Cela reste tout de même un processus qui est assez complexe et avec cette introduction, nous n'en avons exploré que la pointe de l'iceberg. Pour la suite de ce rapport, nous entrerons davantage dans le détail afin de vous expliquer ces algorithmes plus en profondeur.

I - Base de données et modèle utilisés

Comme vous avez pu le comprendre dans le chapitre précédent, pour répondre aux besoins de la reconnaissance d'expressions faciales, il est primordial d'utiliser un bon modèle. Même s'il en existe un grand nombre disponibles sur internet, il est difficile d'en trouver un qui répond parfaitement à un besoin précis. L'efficacité d'un modèle dépend de nombreux facteurs et principalement du dataset avec lequel le modèle a été entraîné.

Pour notre projet, la première étape a été de trouver à la fois des bases de données intéressantes auxquelles nous pourrions accéder et des modèles pré-entraînés. Pour la recherche du modèle, nous avons fait un état des lieux de l'existant pour en trouver un qui pourrait, même si pas parfaitement, répondre à notre besoin. La raison pour laquelle on peut trouver un modèle qui ne répond que partiellement à notre besoin est qu'il est possible de faire du fine-tuning afin de perfectionner un modèle pour une tâche particulière.

Pour ajuster un modèle, nous avons évidemment besoin d'un dataset nous permettant d'entraîner le modèle à nouveau. C'est pourquoi nous avons cherché en parallèle du modèle différents jeux de données disponibles sur internet. La reconnaissance d'émotions étant un sujet de recherche assez répandu, il existe de nombreuses bases de données qui proposent des séquences d'images ou de vidéos de sujets en train de faire différentes expressions du visage.

Nous allons commencer par vous détailler notre démarche quant au choix d'une base de données et d'un modèle en vous expliquant quels critères nous avons défini.

I.1 Choix de la base de données

Trouver un jeu de données n'est pas une mince affaire. Quand on parle de dataset pour entraîner un modèle d'IA, il s'agit la plupart du temps d'un très grand nombre de données. La difficulté pour trouver des données cohérentes aux besoins du projet varie en fonction du contexte. Pour des images ou des vidéos, il est assez compliqué d'avoir accès à une grande quantité de données. D'autant plus que plus le besoin est précis, moins nous trouvons de base de données adaptées. Dans le contexte de ce projet, nous avons besoin de vidéos de sujets effectuant des expressions du visage et dégageant une série d'émotions bien précise.

Nous avions également besoin que les vidéos soient étiquetées, c'est-à-dire que la vidéo soit accompagnée de l'expression du visage. Les bases de données n'utilisent pas toutes les mêmes émotions, certaines en proposent plus que d'autres. Pour notre dataset nous avons convenu de travailler avec les six émotions d'Ekman : joie, tristesse, dégoût, peur, colère et surprise. Ce sont celles définies comme les six émotions primaires. Nous avons donc fait des recherches sur internet afin de trouver des dataset déjà construits et surtout labellisés. En parallèle de cette recherche de base de données, nous cherchions pareillement un modèle correspondant à notre objectif de reconnaissance d'expressions faciales. Modèle et dataset vont de pair, étant donné qu'un modèle est entraîné avec un dataset en particulier. Nous nous sommes rendus compte que la plupart fonctionnaient avec un apprentissage supervisé. Grâce aux articles présents dans la présentation des modèles et les datasets cités, nous avons pu faire une sélection de jeux de données candidats.

Le site Papers With Code, nous a grandement aidé pour la recherche de datasets avec les caractéristiques de notre choix. En effet, ce site répertorie des datasets pour machine learning de types différents

(image, vidéo, audio) et pour différentes tâches dont la reconnaissance d'expressions faciales. Chaque dataset est accompagné d'une liste de papiers portant sur ce dataset avec une évaluation, du nombre de papiers le citant, mais également des meilleurs modèles ayant été entrainés avec le dataset et des liens vers des implémentations. Ce site, très complet, nous a donc permis de sélectionner des bases de données et en plus les modèles fonctionnant avec.

Afin de trier les différentes bases de données et de pouvoir choisir la plus adaptée, nous avons établi une liste de critères permettant de classer les datasets. Pour évaluer la qualité des datasets, nous avons rempli un tableau avec, pour chacun : le nombre de citations, le nombre de vidéos, la résolution de la vidéo, le nombre et la diversité des sujets présents et la classe d'expression. Il était de même nécessaire de voir si les différentes bases étaient plus ou moins faciles à obtenir.

Les datasets sont souvent construits par des laboratoires de recherche universitaires ou des entreprises qui travaillent sur un modèle d'IA à entraîner. La construction des datasets peut être très coûteuse en termes de temps et d'argent. La plupart des fois, les laboratoires proposent des accès aux bases de données sous demande et dans des contextes bien spécifiques. Dans notre cas, lors d'un projet universitaire encadré par des enseignants chercheurs, nous étions dans une situation favorable à la requête pour obtenir des accès.

Parmi la liste des datasets retenus, il y en avait deux qui étaient cités par la plupart des articles en lien avec les modèles : CK+ et MMI. Ces deux datasets sont des références dans les projets FER. CK+ propose une grosse quantité d'images de sujet en train de faire des expressions faciales. MMI, quant à lui, propose des courtes vidéos, de quelques secondes, toujours avec des sujets en train de faire des expressions du visage.

FIGURE 4 – Exemple de séquences présentes dans le dataset MMI

Nous avons choisi MMI principalement, car les deux modèles retenus avaient été entraînés sur ce dataset. Ce qui nous permettait d'avoir une base solide sur laquelle partir pour perfectionner le modèle afin qu'il fonctionne sur un flux vidéo continu. La base de données contient 2900 vidéos comportant 75

sujets qui jouent les six émotions d'Ekman. Après avoir fait une demande aux responsables du datasets, nous y avons eu accès et nous avons pu télécharger les données afin de les exploiter pour le fine-tuning du modèle.

I.2 Choix du modèle

Comme indiqué précédemment, nous avons cherché en parallèle un modèle pour fonctionner avec le dataset. Nous avons utilisé une approche similaire pour la recherche de modèle en établissant un certain nombre de critères. Pour évaluer un modèle, il faut par exemple prendre en compte la précision, les temps de traitements, les temps d'apprentissage.

Les modèles sont souvent accompagnés par des articles scientifiques écrits par les concepteurs des modèles. Cela permet de montrer les capacités et d'expliquer leur démarche de développement. Un modèle dont le papier est très cité par d'autres articles est un modèle qui potentiellement fonctionne bien et qui est apprécié à la fois par les pairs que par les utilisateurs. En partant de ce principe, il est possible de faire des recherches d'articles avec des services tels que Google Scholar par exemple, qui donne accès à un moteur de recherche sur des articles scientifiques répertoriés.

En plus de Google Scholar, il existe également des sites qui répertorient des dépôts de projet en lien avec l'intelligence artificielle. C'est le cas par exemple de Hugging Face, une sorte de GitHub centré principalement sur les modèles d'IA. Ce site permet donc aux développeurs de partager leurs modèles avec la communauté tout en donnant très souvent accès au code et aux articles correspondants.

Après avoir fait plusieurs recherches et sélectionné différents modèles, nous avons exploré les codes sources des projets et lu les articles afin de restreindre la liste des candidats pour répondre à notre besoin. Deux modèles ont été retenus, DeXpression et FMPN (Facial Motion Prior Networks for facial expression recognition). Le choix du modèle et du dataset s'est fait conjointement. Ces modèles ont utilisé le dataset MMI.

II - Algorithmes

II.1 Fonctionnement du modèle choisi : DeXpression

Le modèle DeXpression a été créé avec un réseau de neurones de convolutions profond, un CNN. Il est composé de plusieurs couches qui se répètent à plusieurs reprises, formant ainsi le réseau complet.

Parmi les différentes couches qui composent le réseau, il y a :

— **Couche de convolution (Convolutional Layer)** : Les couches de convolutions permettent d'effectuer un traitement à l'image. L'idée est d'appliquer un filtre, ou plusieurs, à chaque pixel basé sur son voisinage n x m. L'utilisation de plusieurs filtres permet de calculer une représentation plus riche et diversifié de l'image d'entrée. Un aspect intéressant de cette implémentation est que les neurones partagent des coefficients (poids) avec des neurones voisins afin de rendre l'apprentissage plus performant.
Voici la formalisation de la sortie de ces couches de convolution :

$$C(X_{u,v}) = \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{m}{2}}^{\frac{m}{2}} f_k(i,j) X_{u-i,v-j}$$

— **Couche de mise en commun maximal (Max Pooling Layer)** : Ces couches recherchent la valeur maximale en utilisant un filtre de taille m centré sur l'entre xi. L'opération permet de réduire la dimension des données tout en préservant les caractéristiques importantes. Cela permet également de réduire les ressources de calculs tout en préservant une invariance translationnelle, c'est-à-dire la capacité de reconnaître certains motifs indépendamment de leur position exacte dans l'image.
Avec k et l les limites de voisinage à considérer, l'opération peut être formalisée ainsi :

$$M(x_i) = \max\{x_{i+k,i+l} \mid |k| \leq \frac{m}{2}, |l| \leq \frac{m}{2} \;\; k,l \in N\}$$

— **Unité linéaire redressé (Rectified Linear Unit - ReLU)** : Il s'agit d'une fonction d'activation très utilisée dans les réseaux de neurones. Elle permet d'introduire une non-linéarité en favorisant l'activation de neurones qui ont une valeur positive dans leur sortie, tout en désactivant les neurones qui ont une valeur négative. De cette manière, le réseau est capable d'apprendre des relations complexes entre les caractéristiques d'entrée. Ces unités sont plus précises par rapport aux unités binaires qui retournent uniquement des zéros ou des uns, et elles sont plus performantes par rapport aux unités qui utilisent une sigmoïde avec des calculs exponentiels. La formalisation de cette fonction est :

$$R(x) = \max(0, x)$$

— **Couche entièrement connecté (Fully Connected Layer)** : Ces couches, qui sont également appelées "couche de perceptron multicouche", permettent de relier chaque neurone de la couche précédente à chaque neurone de sa propre couche. Si la couche a en entrée un vecteur x de taille k, et la couche à l neurones, nous obtenons une matrice W de taille $l * k$. Cette implémentation utilise une fonction d'identité comme activation, qui est appliquée au résultat de $W * x$, où W contient les poids des neurones, et qui donne en sortie un vecteur de taille l.
Utiliser la fonction d'identité comme activation donne en sortie de la couche la somme pondérée des entrées, sans aucune transformation non linéaire. Voici la formalisation :

$$F(x) = \sigma(W * x)$$

— **Couche de sortie (Output layer)** : Cette couche est responsable de la représentation de la classe de l'image d'entrée. Elle est représentée sous forme d'un vecteur de dimension égale au nombre de classes du problème. Ayant un vecteur de sortie x de taille n, où n est le nombre de classes, nous pouvons déterminer la classe correspondante à ce vecteur. Pour cela, nous recherchons l'indice i tel que xi est le plus grand parmi tous les éléments de x. Pour le dire simplement, on cherche l'indice de l'élément le plus élevé dans le vecteur.
En formalisant, nous obtenons :

$$C(x) = \{i \mid \exists i \forall j \neq i : x_j \leq x_i\}$$

— **Couche softmax** : Il s'agit d'une fonction d'activation couramment utilisée dans les réseaux de neurones pour les problèmes de classification multiclasse. L'objectif de cette couche est de normaliser les valeurs du vecteur d'entrée de manière à ce qu'elles puissent être interprétées comme des probabilités. La probabilité est comprise entre 0 et 1.
Ayant en entrée un vecteur x de dimension n, pour chaque composante du vecteur, avec e la base du logarithme naturel ($2,71828$), la sortie est calculée ainsi :

$$S(x)_j = \frac{e^{x_j}}{\sum_{i=1}^{N} e^{x_i}}$$

Cette couche permet d'ajuster les poids du réseau tout en minimisant l'erreur globale lors de l'apprentissage. Elle est effectuée lors de la rétro-propagation de l'erreur, les nouvelles valeurs des poids sont calculées en utilisant la dérivée de cette fonction.

L'architecture proposée par les concepteurs du modèle est composée de quatre parties. La première partie va pré-traiter les données. Cela commence avec une première couche de convolution qui va appliquer 64 filtres différents. Cette couche est suivie d'une fonction d'activation ReLU qui va préparer la première couche de mise en commun, le pooling. Le pooling permet de sous-échantillonner les images qui vont être ensuite normalisées par une fonction d'activation LRN (Local Response Normalisation) qui permet de compenser si un trop grand nombre d'activations de neurones est effectué lors du ReLU. Cela est nécessaire afin de passer aux couches successives des données qui ne surestiment pas la description de l'entrée.

Les deux étapes successives sont des blocs de FeatEx (Parallel Feature Extraction Block). Ces deux blocs ont le rôle de descripteurs, ils permettent donc d'extraire des données depuis les images pour pouvoir permettre une classification adaptée par la couche qui les suit. Les blocs de FeatEx se composent de couches de convolution, de pooling et de ReLU. La première couche de convolution réduit la dimension en effectuant une convolution avec un filtre de taille 1×1. Elle est améliorée par une couche ReLU, qui permet d'envoyer à la couche suivante uniquement les neurones activés après la ReLU. Ensuite, une autre convolution est effectuée avec un filtre de taille 3×3. En parallèle, une autre couche de mise en commun est utilisée pour réduire les informations avant d'appliquer le CNN de taille 1×1. L'utilisation de filtres de tailles différentes permet de tenir compte des différentes échelles auxquelles les visages peuvent être présentés. Pour continuer, le premier bloc FeatEx crée deux chemins parallèles de caractéristiques avec différentes échelles, qui sont ensuite combinés dans une couche qui permet de concaténer les différents résultats. Le deuxième bloc FeatEx affine la représentation des caractéristiques et il diminue également la dimension de l'entrée.

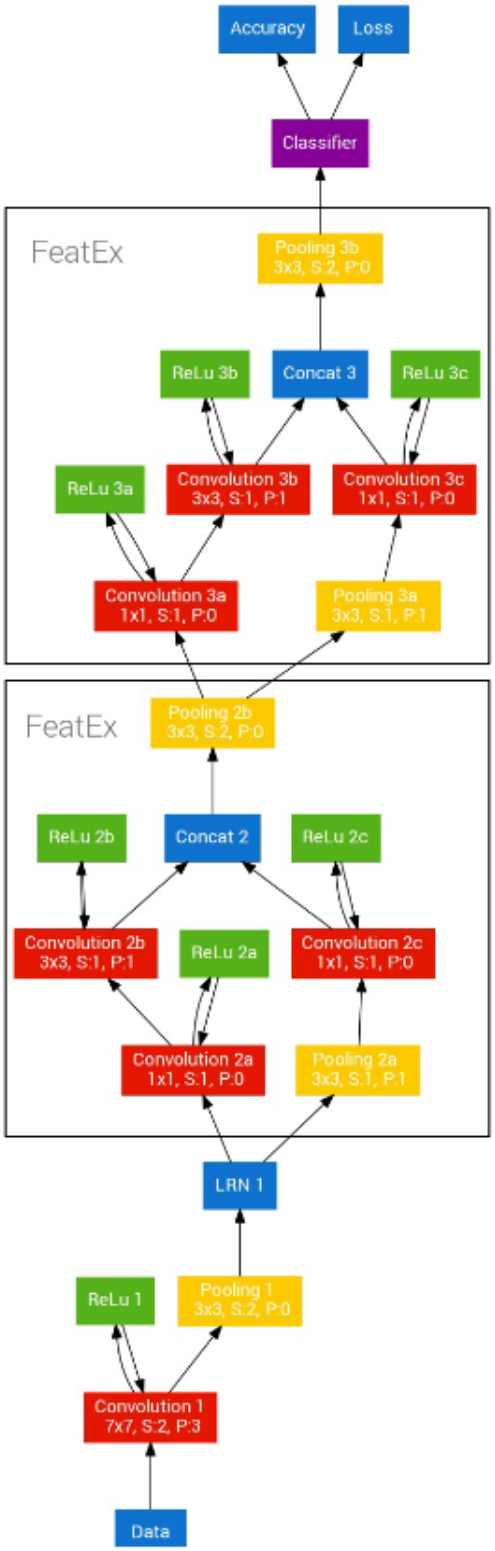

FIGURE 5 – Schéma de l'architecture du modèle DeXpression

II.2 Présentation de la méthode de reconnaissance

Le modèle a été conçu pour fonctionner principalement avec des images. Mais l'architecture proposée est également adaptée à des situations de temps réel, grâce à la compacité de l'architecture. Cela permet d'obtenir des résultats dans un temps acceptable à la fois lors de la phase d'entraînement que durant la prédiction dans un contexte de temps réel.

Pour faire fonctionner le modèle avec un flux en temps réel, nous avons eu besoin de réfléchir à la façon de transmettre l'entrée, donc l'image, au modèle pour qu'il puisse effectuer la prédiction. L'objectif étant d'obtenir une prédiction du modèle en fonction des expressions du visage issues d'un flux webcam continu. Pour ce faire, nous avons décidé de récupérer le flux vidéo et de le découper en temps réel en images. Les images récupérées sont ainsi insérées dans une queue qui va être lue par le modèle pour qu'il puisse prédire l'émotion.

Avant d'insérer chaque image dans la queue, nous avons la possibilité d'effectuer certains traitements dans le but d'améliorer l'image avant de laisser le modèle prédire l'émotion. Étant donné que les flux webcam ont souvent une faible qualité et sont issus d'un capteur de lumière peu performant, il est nécessaire d'effectuer des prétraitements afin d'améliorer l'image. De plus, nous pouvons rendre l'image plus légère et mieux adaptée pour l'analyse du modèle. Nous pouvons donc rehausser le contraste avec une égalisation d'histogramme ou encore effectuer un filtre gaussien pour atténuer certains bruits.

Ensuite, nous détectons le ou les visages présents dans l'image dans le but de d'extraire cette partie et de ne passer au modèle que l'information qui lui est nécessaire pour prédire l'émotion. Ce prétraitement est très important, notamment pour donner au modèle une plus faible quantité de données, ce qui conduit à de meilleures performances. Une fois ces traitements effectués, nous pouvons passer chaque image au modèle pour qu'il puisse faire sa prédiction.

Il est important de noter que cette démarche nous permet de gérer le flux et les données passées au modèle comme nous le souhaitons. En effet, si nous nous rendons compte que prédire chaque image issue du flux est lent et coûteux, nous pouvons limiter le nombre d'images étant analysées par le modèle. Cela a une grande influence sur les performances du programme.

III - Résultats

III.1 Analyse du modèle trouvé : DeXpression

L'implémentation du modèle DeXpression que nous avons trouvé a été entraînée sur le dataset CK+. Cependant, nous n'avons pas eu accès aux poids des neurones du modèle, nous avons donc dû entraîner le modèle nous même, en utilisant le dataset MMI. Pour ce faire, nous avons eu accès au CCUB, le centre de calcul de l'université de Dijon. L'entraînement a été fait avec trois mille images triées et labellisées depuis la base de données MMI. Pour chaque vidéo nous avons extrait les images et récupéré celles où l'expression était bien présente. Lors de l'entraînement nous avons eu des résultats plutôt satisfaisants, avec une précision de 97% à la fin de dix époques. Cette précision n'est pas forcément le signe d'un modèle très performant. Il est fort probable que le modèle soit uniquement capable de prédire l'émotion sur des visages issus de ceux de l'entraînement, et pas sur des nouveaux visages. En effet, nous n'avons pas une très grande diversité de visages pour l'entraînement. Pour cette raison, nous avons constaté que le modèle a du mal à détecter certaines émotions, notamment lors d'un flux vidéo provenant d'une webcam. Tandis que si la détection se fait depuis une image issue du dataset d'entraînement, elle est toujours correcte.

De plus, le modèle tend à confondre certaines émotions qui pourraient avoir des traits similaires. Par exemple, avec un visage énervé qui montre les dents, le modèle peut l'interpréter comme de la joie étant donné qu'en souriant, nous montrons également nos dents. Même chose pour la surprise et la peur, qui sont des émotions qui peuvent facilement être confondue entre elle, même par un être humain. En revanche, le modèle est très rapide et efficace même sur des flux vidéos. Cela s'explique par le fait que l'architecture n'est pas très conséquente, elle comporte un nombre de couches peu élevé ce qui engendre un nombre de paramètres convenable pour du temps réel. Voici un tableau montrant les paramètres pris en entrée par chaque couche :

Couche	Paramètres en sortie
Input	48×48
Convolution_1	$64 \times 21 \times 21$
Pooling_1	$64 \times 21 \times 21$
LRN	$64 \times 10 \times 10$
Convolution_2a	$96 \times 10 \times 10$
Convolution_2b	$64 \times 8 \times 8$
Pooling_2a	$64 \times 56 \times 56$
Convolution_2c	$208 \times 8 \times 8$
Concat	$272 \times 8 \times 8$
Pooling_2b	$272 \times 6 \times 6$
Convolution_3a	$208 \times 6 \times 6$
Convolution_3b	$64 \times 6 \times 6$
Pooling_3a	$272 \times 28 \times 28$
Convolution_3c	$64 \times 28 \times 28$
Concat	$272 \times 6 \times 6$
Pooling_2b	$272 \times 6 \times 6$
Classifier	$11 \times 1 \times 1$

TABLE 1 – Tableau des paramètres de sortie de chaque couche du modèle deXpression

III.2 Comparaison et analyse avec des modèles déjà entraînés

Pour tester d'autres modèles et les comparer avec le modèle DeXpression, nous en avons trouvé deux ainsi que leurs poids respectifs, dont le célèbre modèle VGGFace développé par l'université d'Oxford. En utilisant ces deux autres modèles sur un flux de webcam, nous avons pu comparer les trois afin de mettre en évidence des points forts et des points faibles de notre implémentation de DeXpression.

Tout d'abord, nous avons comparé leur vitesse d'exécution, c'est-à-dire le temps nécessaire à l'algorithme pour prédire une image de flux webcam. Étant donné que le nombre de paramètres diffère ainsi que les couches des architectures, ils sont plus ou moins adaptés à du temps réel. Nous avons pu constater que DeXpression, tel que nous l'avons implémenté, est très rapide à traiter une image. Il est aussi rapide que le modèle VGGFace.

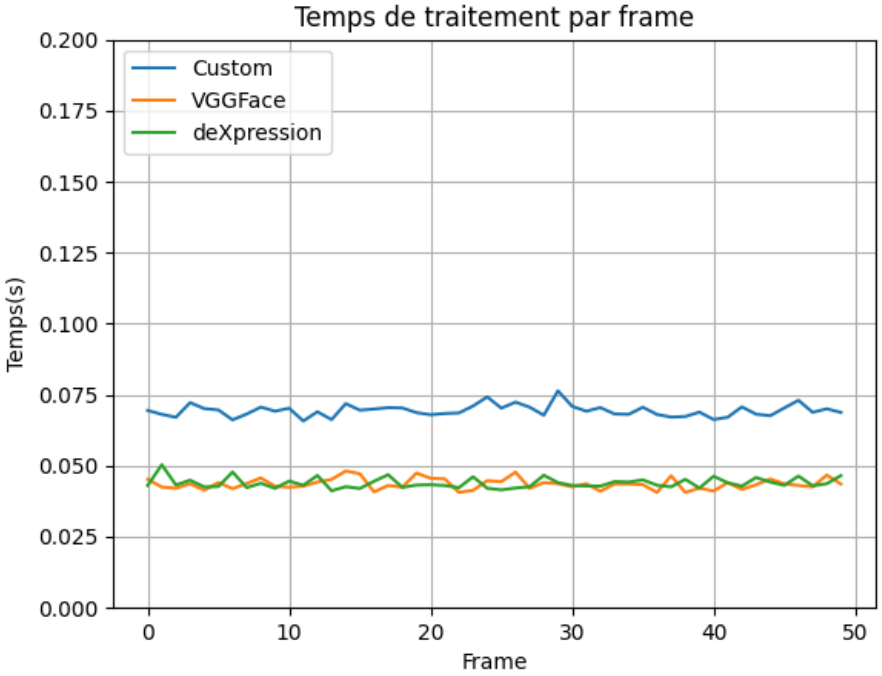

FIGURE 6 – Graphique du temps d'exécution moyen par image

Dans le graphique d'inférence (figure 6), nous pouvons voir que DeXpression est capable de tenir tête à un algorithme connu comme celui de VGGFace. Il faut tout de même dire que nous avons adapté DeXpression pour travailler avec des images de taille 48×48, ce qui rend le traitement plus rapide étant donné qu'il y a moins de paramètres à traiter.

En plus de la rapidité d'exécution, il aurait été intéressant de tester la précision du modèle pour une prédiction. Pour pouvoir analyser ce résultat, il nous aurait fallu un autre jeu de données labellisé avec lequel tester l'algorithme. Étant donné que notre programme fonctionne sur un flux vidéo, il n'est pas possible pour nous de dire à l'algorithme si ça prédiction est juste ou non par rapport à l'expression courante présente dans le flux vidéo. Si nous avions eu accès à un dataset différent de celui ayant servi à l'entraînement du modèle, nous aurions pu lancer l'algorithme avec et mesurer la justesse des prédictions. Cela nous aurait donné une idée de la capacité de l'algorithme à prédire les émotions sur des données qui n'étaient pas dans le dataset d'origine.

III.3 Éléments d'amélioration des techniques mises en place

Comme nous l'avons expliqué précédemment, le modèle de reconnaissance d'expressions faciales est très dépendant du dataset sur lequel a été réalisé l'apprentissage. Ainsi, les expressions faciales reconnues seront celles présentes dans le dataset et avec les mêmes caractéristiques. La diversité du genre humain en termes d'expressions faciales est large et les réactions à une émotion telles que la colère, la joie, l'angoisse ne sont pas les mêmes chez tout le monde. Même s'il a été démontré que dans le monde, les humains utilisaient les mêmes expressions faciales primaires, ce qui confirme la théorie de Darwin affirmant que leur expression est universelle, les traits sur le visage ne sont pas les mêmes. Une bouche formant un sourire, un sourcil froncé, des yeux écartés, on retrouve ces différents éléments dans une expression faciale, mais suivant le visage, la taille et l'inclinaison ne seront pas les mêmes par exemple. Par conséquent, le modèle est conditionné à des analyses qu'il a convenues en ce qu'était un sourire ou une mine boudeuse.

Une des premières limites de notre programme est donc le dataset que nous avons utilisé. Dans la base de données MMI, nous ne retrouvons pas forcément la même diversité de personnes, c'est-à-dire de tout âge, de toute origine, avec autant de femmes que d'hommes. Nous sommes dépendants de cette base de données et si elle contient des biais, ils peuvent se transmettre à notre programme. Nous connaissons tous les polémiques qui ont eu lieu par le passé avec les méthodes de reconnaissance de visage accusé de discrimination et même de racisme.

Nous sommes également conscients que les expressions faciales des sujets dans la base de données sont surjoués pour justement permettre un apprentissage le plus juste du modèle. Dans la vraie vie, les expressions faciales ne sont pas toujours si franches et un visage peut même en mêler plusieurs. Il en résulte que ce n'est pas toujours simple pour un humain de les distinguer, et encore plus dur pour un algorithme de trancher. D'autant plus que dans cette base le nombre de vidéos labellisées n'était pas si important. En effet, 236 vidéos étaient accompagnées d'une étiquette précisant l'émotion. Cela peut paraître beaucoup et à la fois peu pour un apprentissage précis et complet.

Pour finir, la reconnaissance d'expressions faciales est effective que si le sujet est clairement discernable, éclairé avec une bonne luminosité, sans mouvements et bien en face de la caméra. Cela fait partie des travaux en cours, pour reconstruire si besoin une partie masquée du visage. Dans le cadre de notre projet, sur l'algorithme que nous avons créé, nous appliquons un premier traitement pour recadrer l'image et l'éclairer à nouveau si besoin.

IV - Organisation du groupe et méthode de travail

Maintenant que nous vous avons présenté les grandes étapes de comment nous avons répondu à la problématique, nous allons revenir sur notre organisation au sein du groupe et nos méthodes de travail.

Tout d'abord, nous avons réalisé des réunions hebdomadaires organisées avec notre tuteur sur Microsoft Teams pour échanger sur l'avancée du projet. Cela nous a permis de lui faire part de notre travail, de lui poser des questions et parfois d'être recadré quand nous partions sur différentes pistes. Même quand le tuteur n'était pas disponible ou nous laissait plus de temps pour avancer, nous réalisons des réunions entre nous pour connaître le travail de chacun. Nous avons également rédigé des compte-rendus à l'issue des réunions pour mettre par écrit les conseils et méthodes donnés par notre tuteur.

Nous nous sommes créés un serveur Discord pour échanger entre les différents membres du groupe. Nous avons ainsi pu utiliser différents salons textuels selon l'échange désiré et un salon vocal pour discuter de vive voix tout en partageant un écran si besoin.

Pour ce qui est de la répartition du travail, nous avons commencé par faire un tour de table pour connaître les aptitudes et l'aisance de chacun sur ce domaine de l'IA. Certaines personnes plus à l'aise avec l'IA et l'utilisation de modèles avaient envie de s'essayer à l'entraînement de modèles tandis que d'autres se sentaient plus à même d'effectuer des recherches pour répondre à la problématique. Nous avons donc organisé notre travail de la sorte. Ce projet étant plutôt orienté recherche, une partie conséquente du projet s'est joué dans la recherche de la base de données, de modèles, de méthodes pour parvenir à nos fins. Ce qui nous a également permis de nous initier au fonctionnement de réseaux neuronaux et l'entraînement de modèles.

Conclusion

Durant la réalisation de ce projet, nous avons rencontré quelques difficultés et imprévus. Étant donné notre manque d'expérience sur l'utilisation de modèles d'intelligence artificielle, nous avons eu besoin d'un temps d'adaptation. Le fait d'être suivi par notre tuteur a été pour nous très important lors des moments où nous étions dans une impasse, sans savoir comment s'y prendre pour effectuer une tâche.

Pour commencer, le fait de sélectionner un dataset et un modèle convenables à notre projet était une tâche difficile pour nous. Nous ne savions pas quelles caractéristiques rendaient un modèle plus intéressant qu'un autre, même chose pour les bases de données. Mais grâce à des recherches et aux conseils de notre tuteur, nous avons su trouver des réponses nous permettant de faire des choix.

La base de données MMI, même si elle est très connue dans les projets de FER, a eu besoin d'un tri et d'un filtrage. En effet, même s'il est possible de télécharger uniquement les vidéos accompagnées d'un label pour l'émotion, il est nécessaire de filtrer les frames où l'émotion est jouée. Cela a été un grand problème, mais qui a pû être surmonté pour éviter de rencontrer des biais lors de l'entraînement.

L'exploitation du modèle deXpression ne s'est pas révélée être une tâche simple non plus. L'implémentation que nous avions trouvé n'expliquait pas dans le détail les structures de données qui sont passées au modèle pour l'entraîner. Nous avons eu beaucoup de mal à savoir comment entraîner ce modèle. C'est seulement à force d'essais que nous avons réussi à l'entraîner.

Finalement, ce projet a été surtout une façon de s'apercevoir du grand nombre d'aspects qu'il faut prendre en compte pour pouvoir entraîner et exploiter un modèle d'intelligence artificielle. Nous avons pu trouver de nombreuses implémentations très efficaces qui permettent de détecter des émotions dans un visage en temps réel. Nous avons pu les tester et comparer donc les résultats de chaque modèle en prenant en compte leurs particularités : paramètres, architecture ou encore les performances.

Nous sommes contents d'avoir pu explorer le monde de l'intelligence artificielle et de nous initier aux traitements d'images qui visent à faire de la reconnaissance d'expressions faciales. Merci pour l'attention que vous porterez à notre rapport. Nous espérons avoir été clairs dans nos explications et dans les démarches que nous avons effectuées.

Bibliographie

Mmi facial expression database. a. URL https://mmifacedb.eu/.

A simple convolutional neural network, reproduced from rebala et al. b. URL https://www.researchgate.net/figure/A-simple-convolutional-neural-network-reproduced-from-Rebala-et-al-18_fig1_367786775.

Papers with code - mmi dataset. c. URL https://paperswithcode.com/dataset/mmi.

Supervised and unsupervised machine learning. june 2018. URL https://commons.wikimedia.org/wiki/File:Supervised_and_unsupervised_machine_learning.webp.

Neural network explain. november 2021. URL https://commons.wikimedia.org/wiki/File:Neural_network_explain.png.

Réseau de neurones récurrents. mai 2023. URL https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_r%C3%A9currents#.

Peter Burkert, Felix Trier, Muhammad Zeshan Afzal, Andreas Dengel, and Marcus Liwicki. Dexpression : Deep convolutional neural network for expression recognition. URL https://arxiv.org/pdf/1509.05371v2.pdf.

Alan S. Cowen, Dacher Keltner, Florian Schroff, Brendan Jou, Hartwig Adam, and Gautam Prasad. Sixteen facial expressions occur in similar contexts worldwide. *Nature*, 2020. URL https://www.nature.com/articles/s41586-020-3037-7.epdf?sharing_token=5D6wk0dzyf3VOXZkVTxke9RgN0jAjWel9jnR3ZoTv0MoNbV4Dp3UAuVQXWMsVp07Qxj-0ijbaQHL5JGuHNwPG7LDs0VFvGeIg3D&tracking_referrer=ici.radio-canada.ca.

Michel F. Valstar and Maja Pantic. Induced disgust, happiness and surprise : an addition to the mmi facial expression database. URL http://www.lrec-conf.org/proceedings/lrec2010/workshops/W24.pdf#page=73.