L'IA rencontre les émotions : Détection des expressions faciales avec un réseau convolutif

Adrien LAIGLE
laigle@et.esiea.fr

Mehdi AIT HAMMA
aithamma@et.esiea.fr

Axel HEGO
hego@et.esiea.fr

Antoine BUISSON
buisson@et.esiea.fr

***Abstract*—La reconnaissance des expressions faciales (FER) suscite un intérêt croissant en raison de ses nombreuses applications, notamment dans les interactions homme-machine, l'analyse de la santé mentale et les systèmes de sécurité. Ce challenge de machine learning explore l'utilisation des réseaux de neurones convolutifs (CNN) pour détecter et classer automatiquement les émotions humaines à partir d'images faciales. En s'appuyant sur un ensemble de données d'expressions faciales labellisées, nous entraînons un modèle d'apprentissage profond capable d'identifier les caractéristiques clés associées à sept émotions fondamentales : colère, dégoût, peur, joie, tristesse, surprise et neutralité. Afin d'améliorer les performances du modèle, nous appliquons un prétraitement des images incluant la normalisation, la conversion en niveaux de gris et le redimensionnement, tout en expérimentant différentes architectures pour optimiser l'exactitude et réduire le surapprentissage. La méthode proposée montre des résultats prometteurs, soulignant le potentiel des CNN pour améliorer la précision et l'efficacité des systèmes de reconnaissance des émotions faciales.**

***Index terms*—Facial recognition, Convolutional Neural Network, Machine Learning, CNN**

I. Introduction

L'objectif de ce challenge était, à partir d'une base d'images et d'un ensemble de points caractéristiques liés au visage de personnes, de créer un modèle capable de détecter les émotions sur une image donnée. Pour ce faire, il nous a fallu mettre en place une pipeline permettant de réaliser la partie pré-processing des images, l'entraînement d'un réseaux de neurones convolutif pour extraire les features les plus importantes et également l'évaluation du modèle.

II. Analyse des données

Les données qui nous ont été fournies pour l'entraînement de notre modèle étaient composées d'une base d'images de différentes personnes exprimant des émotions basiques, d'un ensemble de points caractéristiques liés au visage de chaque personne et d'un label décrivant l'émotion sur l'image.

Table I: Répartition des données

Label	Total
Fear	150
Sad	149
Surprise	149
Anger	149
Disgust	149
Neutral	130
Happy	101
TOTAL	977

Les points caractéristiques sont répartis comme suit : 68 coordonnées X et 68 coordonnées Y.

Comme nous pouvons le remarquer sur la figure *Table1*, les données sont équitablement réparties mais sont trop peu nombreuses pour être appliquées à notre cas d'usage.

NB : On note tout de même un léger sous apport de l'émotion *Happy* dans notre dataset, mais sans impact puisque cette émotion est la plus "facile" à déterminer.

A. Data Augmentation

Afin d'entraîner un modèle de machine learning performant, il nous est indispensable de réaliser une *Data Augmentation*. C'est une technique qui génère artificiellement des données supplémentaires en appliquant des transformations aux données existantes (rotations, zooms, inversion…). Dans notre cas, nous avons notamment utilisé une méthode de rotation des images et des coordonnées des points caractéristiques afin d'augmenter notre dataset. En multipliant ce dernier par 6 (cf *TableII*), nous pouvons à présent entraîner notre modèle avec des risques réduits d'overfitting (surapprentissage) et permettre une meilleure généralisation sur les différentes émotions.

B. Description des features

Les 68 points caractéristiques [1] permettent de cerner les parties importantes du visage solicitées par les différentes émotions. Dans ce but, nous les avons utilisés afin d'améliorer notre dataset d'entraînement. Notamment en ajoutant les différents

TABLE II: RÉPARTITION DES DONNÉES SUR LE DATASET AUGMENTÉ

Label	Total
Fear	900
Sad	894
Surprise	894
Anger	894
Disgust	894
Neutral	780
Happy	606
TOTAL	5862

points sur l'image de base mais aussi en croppant les images en fonction de ces points.

Fig. 1: Transformation de l'image via la pipeline en 100x100

Avant d'être utilisé à travers notre modèle, nous appliquons un niveau de gris à toutes les images et les redimensionnons en 100x100.

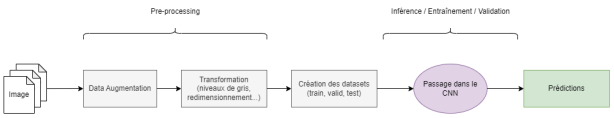

Fig. 2: Pipeline de notre modèle permettant le pre-processing, la création de datasets et l'inférence

Après transformation et augmentation des données, nous sommes prêts à créer et entrainer notre modèle.

III. CRÉATION D'UN MODÈLE

Pour notre modèle, nous avons étudié différents cas d'usages, notamment en utilisant dans un premier temps uniquement les coordonnées des points caractéristiques et des modèles type arbres de décision, FFNN (Feed-Forward Neural Network)… Comme nous pouvons le constater sur la *Table III*, le meilleur score en accuracy que nous ayons obtenu est 0.71 avec un LinearSVC.

TABLE III: ACCURACY DE DIFFÉRENTS MODÈLES SUR LES CARACTÉRISTIQUES

Model	Accuracy
LightGBM	0.54
RandomForest	0.50
LinearSVC	**0.71**
KNeighbors	0.31
SVC	0.30
MLPClassifier	0.13
FFNN	0.20

Ces modèles étant très peu performants sur nos données, nous nous sommes penchés sur la création d'un réseau de neurones convolutif [2] utilisant notre base d'images d'une part et les caractéristiques d'autre part.

A. *Architecture du modèle*

Les CNN, ou Convolutional Neural Network, sont conçus pour exploiter les relations spatiales locales dans les images en utilisant des filtres convolutifs. Cela signifie qu'ils peuvent détecter des motifs et des features importantes dans différentes parties de l'image de manière local et global.

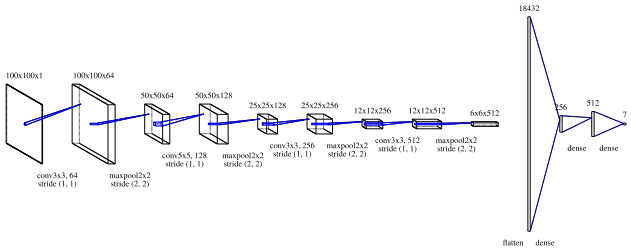

Fig. 3: Architecture du modèle: on retrouve 4 couches de convolution, du max-pooling et des couches Dense à la sortie du réseau.

Pour notre modèle, nous avons utilisé 4 couches de convolution, accompagnées de Max-Pooling, avec des images d'entrée de taille 100 x 100. Notre CNN comporte également des couches Linéaires qui permettent de combiner toutes les caractéristiques extraites par les couches de convolution pour décider de la classe finale. Elles utilisent des poids appris pour donner plus ou moins d'importance à certaines caractéristiques dans la décision de classification et proposer une classe dominante parmi les 7 sorties (7 émotions).

Le choix d'utiliser 4 couches de convolution s'est illustré par la nécessité d'extraire les features et détails les plus importants sur l'image. En effet, avec moins de couches, les détails étaient plus difficilement cernable par le modèle.

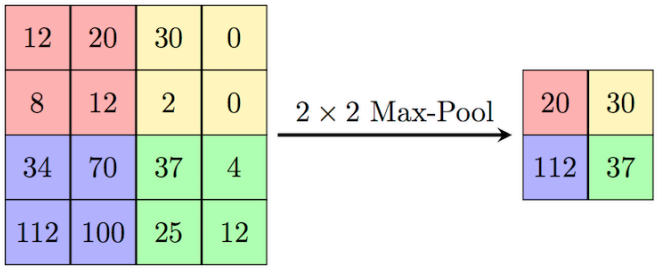

Fig. 4: Max Pooling s'appliquant sur l'ensenble de la feature map (matrice des pixels de l'image après le passage d'un filtre)

Le modèle applique donc des filtres convolutifs sur les images d'entrée pour extraire les caractéristiques importantes, comme les contours, les formes et les textures associées aux expressions faciales. Le Max-Pooling (cf *Fig. 4*) permet de réduire la taille des cartes de caractéristiques, tout en conservant les informations essentielles.

B. *Datasets*

Pour l'entraînement de notre modèle, nous sommes partis dans un premier temps sur une répartition 90/10 de notre dataset principal en train et validation.

TABLE IV: RÉPARTITION DES DONNÉES POUR NOTRE DATASET TRAIN / VALID

Label	Train	Valid	Total
Fear	820	80	900
Sad	801	93	894
Surprise	814	80	894
Anger	810	84	894
Disgust	791	103	894
Neutral	690	90	780
Happy	549	57	606
TOTAL	5275	587	5862

IV. RÉSULTATS

Cette répartition nous a permis d'obtenir de bons résultats sur ce dataset au fur et à mesure de l'amélioration de notre modèle. Nous avons, en effet, eu l'occasion d'enlever des couches de convolution ou d'ajouter des filtres en plus selon les performances. Ce qui nous a amené à obtenir un score d'accuracy en training de **0.98** et en validation de **0.94** sur ce dataset.

Le modèle a été entrainé sur 36 époques avec un batch size de 32 sur un Intel(R) Core(TM) i7-9750H CPU avec 32Go de RAM et un GPU RTX 2060.

Dans un second temps, nous avons pu utiliser le dataset de test fourni en vue de ce challenge afin de le soumettre à notre modèle. Après labellisation manuelle, nous avons récolté les résultats qui étaient en-dessous de nos attentes : **0.77** en score d'accuracy.

Nous avons remarqué, via une matrice de confusion, que les classes les plus problématiques étaient principalement *Sad*, *Anger* et *Disgust* qui possèdent des caractéristiques similaires.

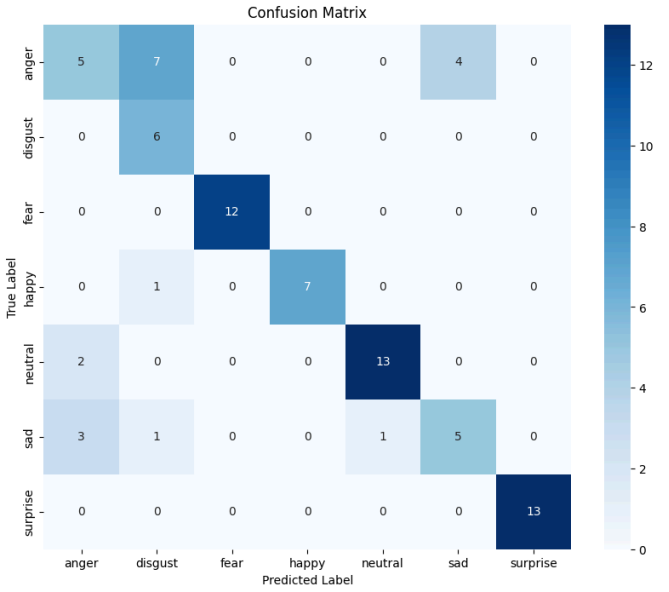

Fig. 5: Matrice de confusion du modèle sur l'extrait du jeu de test fourni pendant le challenge

Nous avons donc ajouté des exemples de type *Sad*, *Anger* et *Disgust* dans notre dataset d'entraînement en mergeant le prédent dataset de train et validation en un seul dataset de train afin de l'enrichir.

De plus, pour avoir plus de détails sur la manière de notre modèle de performer sa classification, nous avons fait appel à **l'explicabilité** via **GradCAM**. GradCAM est une technique populaire de visualisation qui est utile pour comprendre comment un réseau neuronal convolutif a été conduit à prendre une décision de classification et nous permet de générer une carte de chaleur. Cette heatmap nous donne énormément d'informations sur la façon dont notre modèle traite et analyse les images. Avec ces informations, nous savons s'il est nécessaire d'alimenter notre dataset d'entraînement avec des cas supplémentaires que nous n'aurions pas identifier, si le modèle identifie les zones correctes relatives à la classe recherchée…

Nous nous sommes donc aperçus que certaines zones des images (cf *Fig. 6*), hors visage, étaient utilisées pour la classification. Pour y remédier, nous avons du accentuer le centrage de l'image sur le visage et les points caractéristiques.

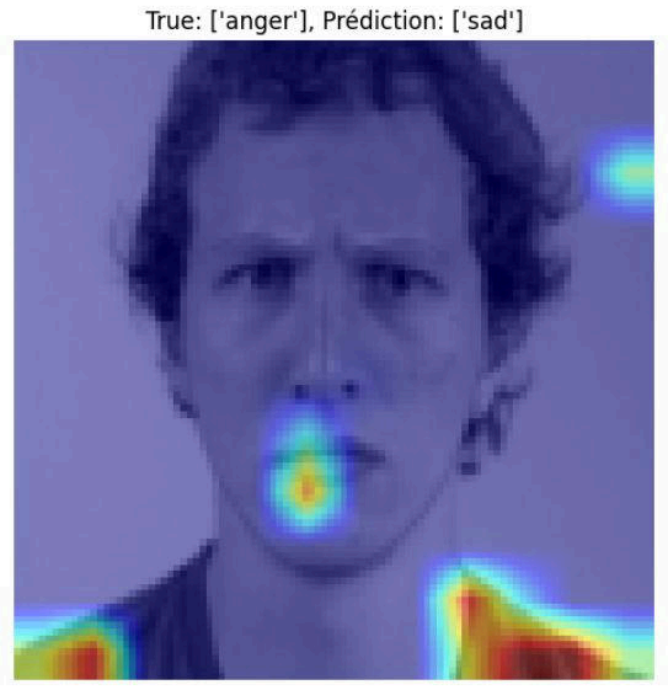

Fig. 6: Carte de chaleur, calculée via GradCAM, représentant les features les plus utilisées par le modèle pour faire sa classification. On voit que la prédiction est incorrecte et que le modèle utilise des élements hors visage.

Voici la courbe qui montre l'évolution de notre modèle après ces différentes modifications :

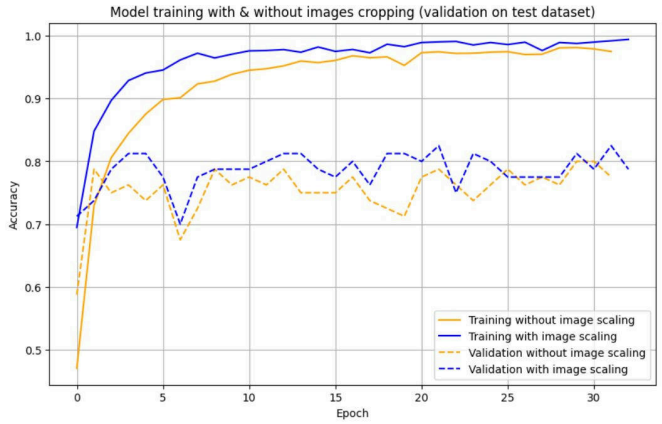

Fig. 7: Courbes représentant l'accuracy du modèle pendant son entraînement et sa validation avec et sans le recadrage des images lié aux points caractéristiques des visages.

TABLE V: ACCURACY DU MODÈLE

Model	Train	Valid
Optimized model	**0.98**	**0.81**
Model without optimization	0.95	0.70

Malheureusement, le jeu de test que nous avions à disposition lors de ce challenge était réduit ce qui biaise légèrement les résultats. Cependant, en nous basant sur la dernière version du jeu de test publié complet et en labellisant ce dernier manuellement, nous serions proche des **0.90** d'accuracy avec notre dernier modèle optimisé.

V. CONCLUSION

Pour conclure, ce challenge nous a permis d'analyser un cas d'usage pertinent sur l'utilisation de l'intelligence artificielle à travers la reconnaissance d'émotions. Nous avons pu créer un réseau de neurones convolutif puis le perfectionner jusqu'à atteindre un niveau de performance et de précision correct. L'utilisation de l'explicabilité a aussi été un plus dans l'objectif d'améliorer notre dataset et notre modèle.

Pour des projets futurs, nous pouvons imaginer développer notre dataset d'entraînement avec des millions d'images et utiliser notre modèle pour de la classification instantanée via une caméra ou autre.

REFERENCES

[1] A. M. Pascual *et al.*, "Light-FER: A Lightweight Facial Emotion Recognition System on Edge Devices," *Sensors (Basel, Switzerland)*, vol. 22, no. 23, p. 9524–9525, Dec. 2022, doi: 10.3390/s22239524.

[2] R. Gill and J. Singh, "A Deep Learning Approach for Real Time Facial Emotion Recognition," Dec. 2021, pp. 497–501. doi: 10.1109/SMART52563.2021.9676202.