Master 1 : Majeure Intelligence Artificielle

Rapport de projet d'application : Reconnaissance d'émotions faciales

Projet 2ème année Cycle Ingénieur, Majeure Intelligence Artificielle

Réalisé par Clément Reiffers, Yohan Cohen-Solal et Quentin Morel
Encadré par Larbi Boubchir

GitHub lm-Rises/emotion_recognition_cnn

Jupyter Notebook

Années Universitaires 2021-2022

Remerciements

Tout d'abord, nous tenons à remercier tout particulièrement et à témoigner toute notre reconnaissance aux personnes suivantes, pour leur dévouement et leur soutient dans la concrétisation de ce projet ingénieur :

- M. Larbi Boubchir, encadrant du projet qui nous aura guidé tout au long de ce semestre, et sans qui le projet n'aurait pas pu voir le jour.

- M. Madjid Maidi, responsable de la majeure Intelligence Artificielle qui nous aura appris le fonctionnement des systèmes ainsi que les notions importantes en Machine Learning.

Résumé

La reconnaissance d'émotions faciales est un enjeu majeur pour la compréhension d'autrui dans la société, et peut permettre d'élucider des mystères en analysant correctement la personne. Par exemple, analyser la compréhension d'un cours par des élèves ou encore déceler les mensonges d'une personne en garde à vue sont des sujets potentiels d'utilisation de cette reconnaissance d'émotions faciales.

Peu de temps après la fin de la Seconde Guerre Mondiale, des principes d'Intelligence Artificielle émergent. Les chercheurs se demandent alors : « Est-ce qu'une machine pourrait penser ? ». Des recherches ont eu lieu, mais ces dernières se sont retrouvées très vite limitées à cause des ordinateurs qui n'étaient pas assez puissants. A l'aube du XXIème siècle, nous avons aujourd'hui des outils assez évolués pour exécuter des algorithmes permettant d'automatiser cette tâche, en utilisant de l'Intelligence Artificielle.

Cette Intelligence Artificielle peut apprendre, grâce à des jeux de données qu'on lui fournit. Cet apprentissage est défini par des vecteurs et matrices, c'est ce qu'on appelle le Deep Learning.

Pour ce sujet, nous traiterons uniquement la reconnaissance d'émotions faciales par image. Nous utiliserons donc des réseaux de neurones implémentés avec Keras. Nous nous servirons d'une seule architecture, ResNet-50, aujourd'hui l'architecture la plus efficace dans les reconnaissances d'objets dans une image.

Mots clés : Réseaux de Neurones, Deep Learning, Keras, ResNet-50

Sommaire

I. Introduction 5
1. Problématique 5
2. Usages de la reconnaissance d'émotions faciales 5
3. Problèmes éthiques et légalité 6
a) Problèmes éthiques 6
b) Problèmes liés aux données 6
5. La reconnaissance d'émotions 7
6. Méthodes de reconnaissance émotionnelle 7
7. Le Deep Learning et Machine Learning 9
8. Les approches basées sur le Deep Learning 9
9. Applications existantes 9
II. Le Deep Learning et les différents modèles 11
1. Deep Learning ou Machine Learning 11
2. CNN / RNN / ANN 11
a) CNN 11
b) RNN 12
c) ANN 12
3. Tableau récapitulatif des réseaux de neurones 13
4. Le principe du CNN 13
a) La couche de convolution 13
b) La couche de pooling 14
c) La couche de correction ReLu 14
d) La couche fully-connected 14
5. Architectures existantes 14
a) LeNet-5 14
b) GoogleNet (inception) 14
c) ResNet 15
d) AlexNet 15
e) ZFNet 16
f) VGG 16
III. Méthodologie 16
1. Les librairies 16
a) Tensorflow/Keras 17
b) Scikit-learn 17
c) PyTorch 17
2. Le pré-traitement des images 17
3. Le modèle de réseaux de neurones 17

4. Méthodologie de mise en place d'un modèle 18
5. La data augmentation 18
6. Le Transfer Learning 20
7. La détection de visage 21
IV. Solution proposée 22
1. Le pré-traitement des images 22
2. ResNet50 22
3. L'Early Stopping 22
4. Data augmentation 23
5. Sélection de la meilleure base de données 24
V. Analyse, interprétation et discussion des résultats 26
1. Analyse 26
2. Résultats 27
VI. Conclusion et perspectives 28
VII. Bibliographie 29
1. Documents techniques 29
2. Ouvrages 29
3. Références électroniques 29
4. Images 29

I. Introduction

1. Problématique

Un système de reconnaissance d'émotions est un outil capable d'analyser les expressions du visage d'un individu et de comprendre l'émotion que celui-ci renvoie.

Paul Ekman et Wallace V Friesen sont les deux premiers scientifiques à imaginer six expressions du visage. Leurs expériences débutèrent dans les années 1960, en montrant des photos représentant plusieurs expressions faciales à diverses personnes, venant de différents pays. Ils ont constaté que ces derniers reconnaissaient chacun les mêmes expressions faciales :

- La joie
- La tristesse
- La surprise
- La peur
- Le dégoût
- La colère

De nombreuses bases de données ont vu le jour par la suite, en reprenant ce système.

L'émotion neutre est arrivée plus tard, ce qui a permis d'enrichir les bases de données et les rendre plus précises

Ces dernières sont constituées de nombreuses images venant de différentes personnes. Il peut y avoir plus ou moins d'images en fonction des bases de données et celles-ci reprennent tous les types d'expression existantes.

2. Usages de la reconnaissance d'émotions faciales

Le marché de la reconnaissance d'émotions est vaste, il est passé de 6.72 Milliards de dollars en 2016 à 36 milliards de dollars en 2021, soit une augmentation de 40 %.

Pour Orange, l'intérêt est de faciliter les interactions par une meilleure compréhension des clients. Avec les services téléphoniques, cela permet d'être sûr de se comprendre.

Pour les services d'application de la loi comme les polices, il y a une meilleure détection d'émotions venant des suspects, et donc pouvoir mieux les cerner. Ces algorithmes peuvent aussi déterminer si une personne se sent suspecte dans une foule afin de centrer nos observations sur elle, cela pourrait déterminer si une personne a des tendances terroristes ou autres.

Pour détecter des fraudes, une analyse du comportement de la personne selon sa voix, son visage, ce qu'elle peut dire est effectuée.

Pour McDonald's au Japon, la reconnaissance d'émotions est appliquée dans les caméras des restaurants pour voir quels sont les clients frustrés et surtout pour voir si les employés affichent un sourire tout le long de la commande pour ne pas frustrer les clients.

Optimisation des centres d'appel : cela permet de savoir si le client a été satisfait ou non. Il existe beaucoup de solutions comme les NLP ou les Speech Analytics qui permettent d'analyser la voix du client ou encore tout ce qu'il a dit en fonction de son contexte. Ces logiciels permettent aussi une analyse profonde de tous les termes utilisés afin de trier toutes les discussions en fonction de certains termes ou sujets.

Certaines écoles souhaitent appliquer des systèmes de reconnaissance d'émotions faciales pour voir si les élèves ont réellement compris un cours, afin d'optimiser le temps pour tout le monde.

Des jeux vidéo vous filment pour que ce dernier puisse s'adapter en fonction de vos sentiments.

La dépression et la démence peuvent également être diagnostiquées grâce à des reconnaissance d'émotions vocales.

De plus, les IA de reconnaissances d'émotions aident beaucoup les enfants autistes pour savoir ce que ressent leur entourage.

3. Problèmes éthiques et légalité

a) Problèmes éthiques

Les questions souvent posées sont les suivantes :

- Quelles sont les personnes fichées ?
- Qui peut consulter le fichier de données ?

Il faut pouvoir exploiter des données sans attaquer la liberté de chacun et la protection de la vie privée. En France, c'est la CNIL (Commission Nationale de l'Informatique et des Libertés) qui gère ces soucis et essaye de limiter les usages de ces technologies considérées comme trop intrusives dans les libertés et la vie privée de chacun.

b) Problèmes liés aux données

La RGPD (Règlement général sur la protection des données) oblige le consentement de chacun pour l'exploitation des données personnelles, comme les données biométriques tel qu'un visage, ce qui est pratique pour de la reconnaissance d'émotions faciales.

4. Liste de bases de données

Avec le développement de l'Intelligence Artificielle et le nombre de systèmes à reconnaissance faciale, le nombre de bases de données s'est grandement développé.

Il en existe un grand nombre, toutes avec des caractéristiques différentes comme le nombre d'expressions faciales, le nombre de sujet, le nombre d'image ou vidéo, la colorimétrie (noir et blanc ou colorisé), la résolution et la pose de la personne, que ce soit de face ou de profil.

Liste de bases de données publiques d'images sur les expressions faciales :

Base d'images	Expressions faciales	Couleur/Niveau de gris	Résolution	Type
AffectNet	Neutre, Heureux, Triste, Surpris, Peur, Dégoût	Couleur	Divers	Cadre divers
Extended Cohn-Kanade Dataset (CK+)[5]	Neutre, Tristesse, Surprise, Bonheur, Peur, Colère, Mépris et Dégoût	Plutôt gris	640* 490	Pose avec sourires spontanés
FER-2013/FERPLUS	0=En colère, 1=Dégoût, 2=Peur, 3=Heureux, 4=Triste, 5=Surprise, 6=Neutre	Gris	48*48	Image frontale du visage
Google Facial Expression Comparison dataset	Chaque annotation est un entier de l'ensemble {1, 2, 3}.	Couleur	Divers	Image frontale du visage

Tableau 1 : Tableau comparatif des bases de données d'images

Les systèmes de reconnaissance d'émotions faciales sont utilisés à l'aide de bases de données d'images pour l'entrainement de l'IA.

On compte aujourd'hui de nombreuses bases de données regroupant plusieurs milliers d'images.

5. La reconnaissance d'émotions

La perception des émotions est la capacité à reconnaître et identifier les émotions d'autrui.

Elles sont généralement considérées comme ayant trois composantes :

- L'expérience subjective (connaissance d'une personne à reconnaître les émotions)
- Les changements physiques (la manière d'être physiquement, gestuelle, marche, etc)
- L'évaluation cognitive (La cognition est l'ensemble des processus mentaux qui se rapportent à la fonction de connaissance et mettent en jeu la mémoire, le langage, le raisonnement, l'apprentissage, l'intelligence, la résolution de problèmes, la prise de décision, la perception ou l'attention)

Les sens qui participent à la reconnaissance d'émotions sont la vue avec la vision du visage, l'ouïe avec la tonalité de la personne lorsqu'elle parle ou bien même via l'odorat, une personne stressée n'aura pas la même impression qu'une autre, l'aspect sensoriel peut aussi permettre d'identifier l'émotion d'une personne par exemple lorsqu'elle est inquiète. Pour un humain tous ces aspects sont plus ou moins innée, bien qu'on se fie majoritairement à la vue, les autres sens ont leur importance.

6. Méthodes de reconnaissance émotionnelle

La méthode de reconnaissance d'émotion qu'utilise le plus l'humain est la reconnaissance faciale. C'est pourquoi la plupart des systèmes de reconnaissances d'émotions sont basés dessus.

Ensuite, nous avons l'ouïe avec le timbre de voix d'une personne et en dernier l'odorat.

La plupart des systèmes de reconnaissances faciales sont donc basés sur le Deep Learning. Etant donné que l'expérience subjective d'une personne est un point très important, un système qui apprend à reconnaitre les émotions se base sur de vrais exemples.

De plus, avec un système de Machine Learning et un réseau de neurones artificiels, on imite le réseau de neurones humain. Sachant cela, une reconnaissance cognitive est imitée par notre système.

Qu'est-ce qu'un neurone et un réseau de neurone ?

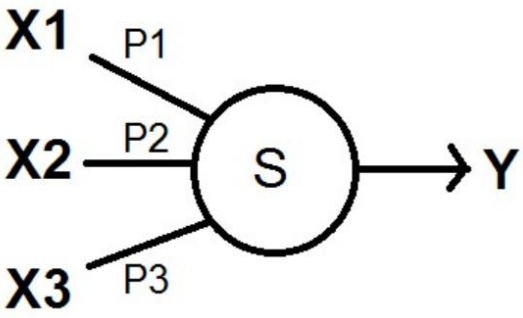


Figure 1 : Principe d'un neurone

Un neurone est un concept crée par Warren McCulloch et Walter Pitts en 1943. Il s'agit d'un système s'inspirant de neurones biologiques ayant des entrées et une sortie. Le neurone va fonctionner de la même façon qu'une fonction mathématique. Il va recevoir des paramètres d'entrées x1, x2, x3 qui vont faire partie d'une fonction de la forme f(x)=p1.x1+p2.x2+p3.x3, avec p1, p2, p3 des coefficients également appelés des poids. Lorsque le calcul de cette fonction sera effectué, son résultat sera comparé au seuil S.

- Si S > f(x) --> y=0
- Si S < f(x) -->y=1

Cependant, les résultats de ces neurones seront insuffisants pour avoir un système efficace. C'est pourquoi le réseau de neurones a été imaginée.

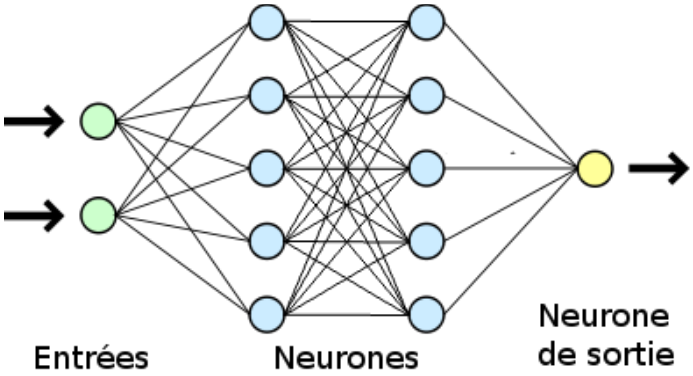


Figure 2 : Principe d'un réseau de neurones

Les réseaux de neurones consistent à récupérer les entrées et à créer des liens avec plusieurs neurones. En utilisant ce processus, les machines sont mieux entrainées et les systèmes deviennent bien plus performants

Il existe trois principales méthodes de reconnaissance :

- Haar classifier method
- AdaBoost method
- Contours

Une fois ces méthodes appliquées et que notre visage est à disposition nous pouvons appliquer une méthode d'extraction des données.

Il en existe plusieurs, les principales étant :

- Géométriques
- Basées sur l'apparence

L'approche géométrique va extraire les formes d'un visage sous forme géométrique et essayer d'analyser les formes afin d'en déduire une expression.

Les algorithmes basés sur l'apparence se concentrent sur les caractéristiques transitoires (rides, renflements, avant-plan) qui décrivent les changements dans la texture du visage, l'intensité, les histogrammes et les valeurs de pixel.

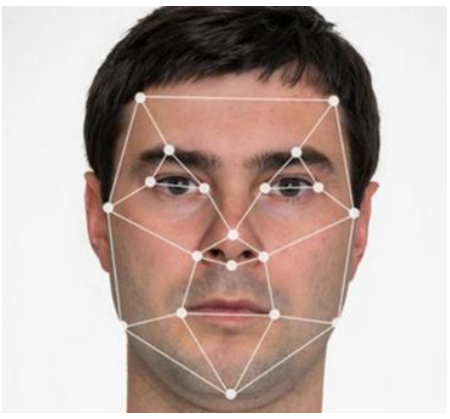


Figure 3 : Principie d'analyse faciale par un réseau de neurones

7. Le Deep Learning et Machine Learning

Dans le cadre de notre projet, nous avons préféré nous baser sur le Deep Learning.

La différence entre le Machine Learning et le Deep Learning tient du fait que dans le cas du Machine Learning on va extraire les données manuellement afin de les envoyer en entrée de notre classificateur. Alors que dans le cas du Deep Learning notre système n'aura pas besoin d'intervention humaine :

- Extraction des données
- Traitement des données

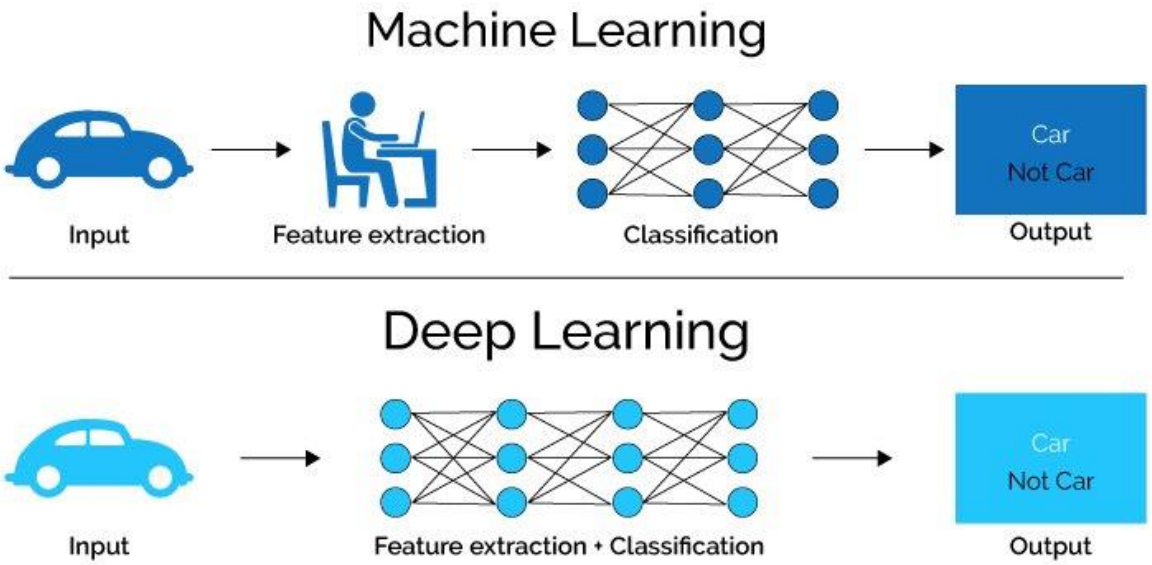


Figure 4 : Différence entre Machine et Deep Learning

8. Les approches basées sur le Deep Learning

La reconnaissance d'émotions faciales passe donc par notre réseau de neurones. De ce fait les systèmes de reconnaissances d'émotions faciales sont donc basés dessus avec trois types de réseaux de neurones principaux :

- CNN
- RNN
- ANN

Chacun avec ses avantages et inconvénients, nous allons utiliser l'un ou l'autre en fonction du type d'analyse (visuel, auditif, etc…).

Pour répondre à notre problématique nous utiliserons et détaillerons pourquoi nous utiliserons des réseaux de neurones CNN.

9. Applications existantes

De nos jours, avec le grand développement de l'IA, la reconnaissance faciale ainsi que la reconnaissance d'émotions faciales se sont beaucoup développées.

Les grandes entreprises comme Google, Apple, Samsung ont notamment beaucoup investis dans le domaine des smartphones.

Des fonctionnalités comme la reconnaissance d'un visage pour déverrouiller son téléphone ou tout simplement dans la galerie d'images pour reconnaître des personnes sur les photos. La plupart sont en fait aujourd'hui directement intégrées dans nos appareils.

Les applications les plus connues sont les suivantes :

- Vision Framework (Apple)
- AI Vision (Google)
- Samsung AI Vision (Samsung)

II. Le Deep Learning et les différents modèles

1. Deep Learning ou Machine Learning

Les deux domaines les plus utilisés de l'IA sont aujourd'hui le Machine Learning et le Deep Learning.

Malgré leur diverse grande ressemblance, chacun est plus approprié pour un domaine ou un autre. Dans notre cas, nous avons décidé de faire une approche de type Deep Learning.

Le Machine Learning est une application de l'IA comprenant des algorithmes qui analysent les données, apprennent de ces données et appliquent ce qu'ils ont appris pour prendre des décisions informées.

Le Deep Learning est une discipline du Machine Learning qui structure les algorithmes en couches pour créer un « réseau neuronal artificiel », capable d'apprendre et de prendre des décisions intelligentes tout seul.

Etant donné que l'analyse d'images et la classification d'émotions se basent sur un grand nombre de paramètres, il est beaucoup plus intéressant de se baser sur un réseau de neurones qui réfléchira comme un humain pour tenter d'analyser notre émotion.

2. CNN / RNN / ANN

Dans le cadre du Deep Learning, il existe un nombre de 3 réseaux de neurones, CNN, RNN et ANN. Chacun a ses avantages et inconvénients en fonction des cas.

a) CNN

Le réseau CNN signifie "Convolutional Neural Network" (Réseau de neuronal convolutif). Les CNN ont pour principe de ne traiter uniquement une entrée partie par partie, en utilisant des suites d'étages convolutifs. A la fin de de ces étages convolutifs, nous aurons alors un étage de décision qui nous donnera le résultat.

Avantages :

- Très haute précision dans les problèmes de reconnaissance d'images.
- Détecte automatiquement les fonctionnalités importantes sans aucune supervision humaine.
- Partage des poids.

Désavantages :

- CNN n'encode pas la position et l'orientation de l'objet.
- Manque de capacité à être spatialement invariant par rapport aux données d'entrée.
- De nombreuses données d'entraînement sont nécessaires.

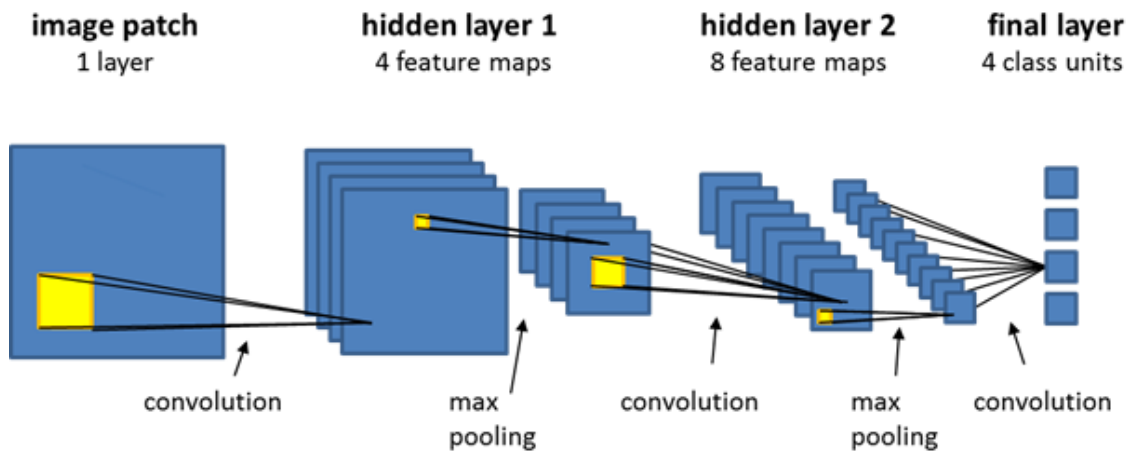


Figure 5 : exemple de couches CNN

b) RNN

Le réseau RNN signifiant “Recurrent Neural Network” (Réseau de neurones récurrents).

Avantages :

- Un RNN se souvient de chaque information à travers le temps. Il est utile dans la prédiction de séries chronologiques uniquement en raison de la fonctionnalité permettant de mémoriser également les entrées précédentes. C’est ce qu’on appelle la mémoire à long terme.
- Les réseaux de neurones récurrents sont même utilisés avec des couches convolutives pour étendre le voisinage de pixels effectif.

Désavantages :

- Problèmes de disparition et d’explosion de gradient.
- Former un RNN est une tâche très difficile.
- Il ne peut pas traiter de très longues séquences si vous utilisez tanh ou relu dans les fonctions d’activation.

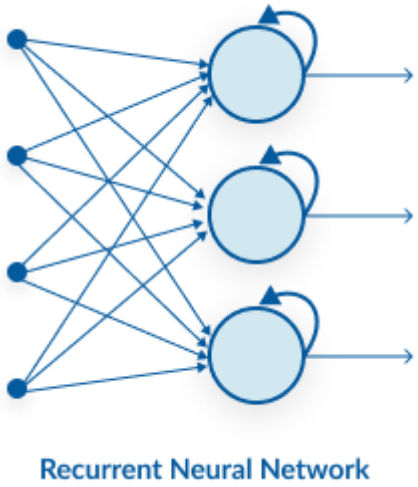


Figure 6 : exemple de RNN

c) ANN

Le réseau ANN signifiant “Artificial Neural Network” (Réseau de neurones artificiels) également connu sous le nom de réseau de neurones Feed-Forward est un réseau de neurones où les entrées ne sont traitées que dans un seul sens.

Avantages :

- Stockage des informations sur l’ensemble du réseau.
- Capacité à travailler avec des connaissances incomplètes.
- Tolérance aux pannes.
- Mémoire distribuée.

Désavantages :

- Dépendance matérielle.
- Comportement inexpliqué du réseau.
- Détermination de la structure appropriée du réseau.

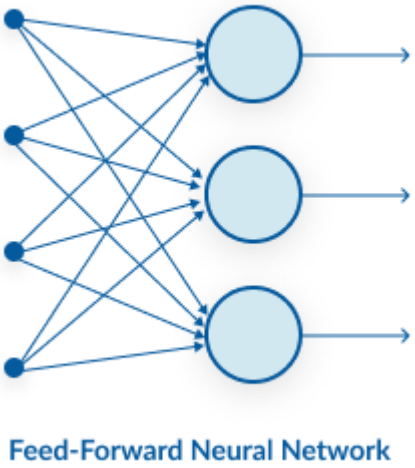


Figure 7 : exemple de Feed Forward Neural Network

Dans le cadre de notre projet, nous avons décidé avec notre encadrant attitré, d'utiliser le modèle de réseau de neurones CNN pour la raison principale de sa grande efficacité sur le traitement d'images. Les autres réseaux de neurones étant plus spécialisés pour d'autres cas.

3. Tableau récapitulatif des réseaux de neurones

Réseau de neurone	CNN	RNN	ANN
Types de données traitées	Données textuelles	Images	Séquence de données
Avantages	Possibilité de travailler avec des données incomplètes	Grande précision dans l'analyse d'image	Se rappelle toutes les informations passées au préalable
Inconvénients	De nombreuses données d'entraînement sont nécessaires.	Former un RNN est une tâche très difficile.	Comportement inexpliqué du réseau.

4. Le principe du CNN

Le CNN est composé de diverses couches :

- Convolution
- Pooling
- Correction ReLu
- fully-connected.

Toutes ces couches ont un but précis.

a) La couche de convolution

Son but est de repérer la présence d'un ensemble de caractéristiques dans les images reçues en entrée.

Le principe est de faire "glisser" une fenêtre représentant la caractéristique sur l'image, et de calculer le produit de convolution entre la caractéristique et chaque portion de l'image balayée. On effectue une sorte de filtrage sur l'image.

b) La couche de pooling

L'opération de pooling consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.

Celle-ci permet de réduire le nombre de paramètres et de calculs dans le réseau.

c) La couche de correction ReLu

La couche de correction ReLU remplace toutes les valeurs négatives reçues en entrées par des zéros.

d) La couche fully-connected

La couche fully-connected constitue toujours la dernière couche d'un réseau de neurones. Elle permet de classifier l'image en entrée du réseau.

5. Architectures existantes

Une architecture est utilisée comme outil de modélisation pour le traitement des données des différents programmes nécessaires à la conception d'une Intelligence Artificielle.

Afin de trier les architectures les plus intéressantes, nous allons parler d'un concours, celui qui s'appelle ILSVRC « ImageNet Large Scale Visual Recognition Challenge ». L'intérêt est de donner un dataset d'images, où l'homme a un taux d'erreur de 5% en moyenne, et voir si la machine est capable de battre l'homme.

a) LeNet-5

C'est une architecture créée par Yann Le Cun et al en 1998 qui permet de classifier des numéros, il est constitué de 7 étages de réseaux de convolution et est constitué de 60 000 paramètres.

L'architecture est constituée de 2 étages qui s'occupent de faire des convolutions et des moyennes, suivie d'un étage qui s'occupe de faire un aplatissement convolutif, puis 2 autres couches et enfin n classificateurs softmax.

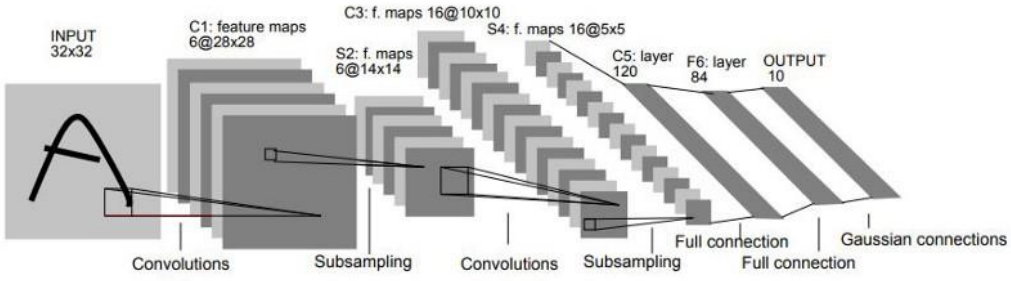


Figure 8 : décomposition d'un réseau LeNet

b) GoogleNet (inception)

Cette architecture est aussi appelée Inception V1 et a été créée par Google. C'est l'architecture qui a battu tous ses concurrents en 2014 avec un taux d'erreur de 6.67%, ce qui est très proche du niveau humain (qui tourne autour de 5%). Google a réutilisé l'architecture de LeNet mais en y ajoutant plusieurs éléments comme la normalisation par

lots, la distorsion d'images et le RMSprop. L'architecture est constituée de 22 couches mais elle réduit le nombre de paramètres de 60 millions (chez alexNet) à 4 millions.

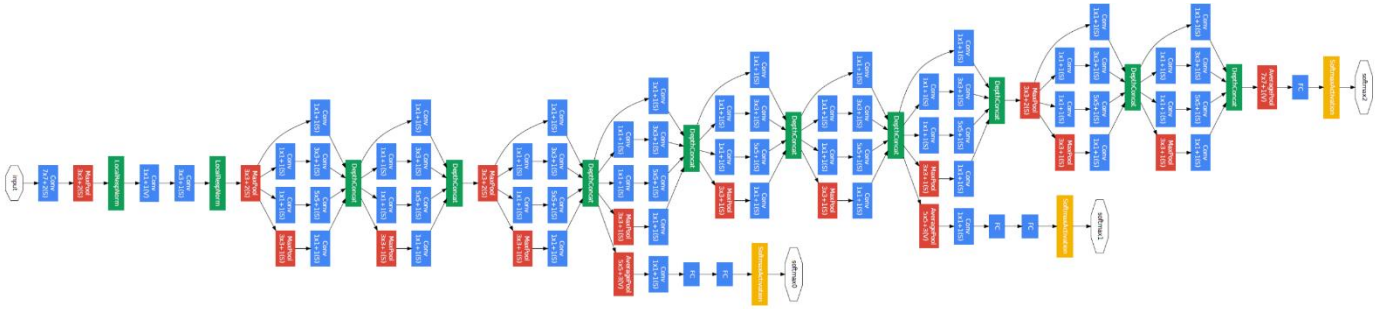


Figure 9 : décomposition du réseau Inception : Understanding GoogLeNet Model - CNN Architecture - GeeksforGeeks

c) ResNet

ResNet signifie "Residual Neural Network", car cette architecture constituée de 152 couches de convolution utilise de la normalisation par lots et des sauts de connexion. Cette architecture a beaucoup de similitudes avec les réseaux RNN. En 2015, elle a battu l'homme en donnant une performance de 3.57% de taux d'erreur.

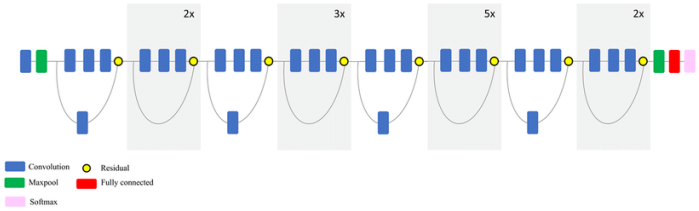


Figure 10 : décomposition d'un réseau ResNet : Quatre modèles pré-entraînés en vision par ordinateur - ResNet50 (Réseau résiduel)_Ascend Forum_HUAWEI Forum CLOUD (huaweicloud.com)

d) AlexNet

Architecture très similaire à celle de LeNet créée par Yann Le Cun mais avec encore plus de filtres. En 2012, il a dépassé tous ses concurrents en réduisant le taux d'erreur de 26 à 15.3%. Cette architecture est constituée de 60 millions de paramètres.

Voici les différentes couches :

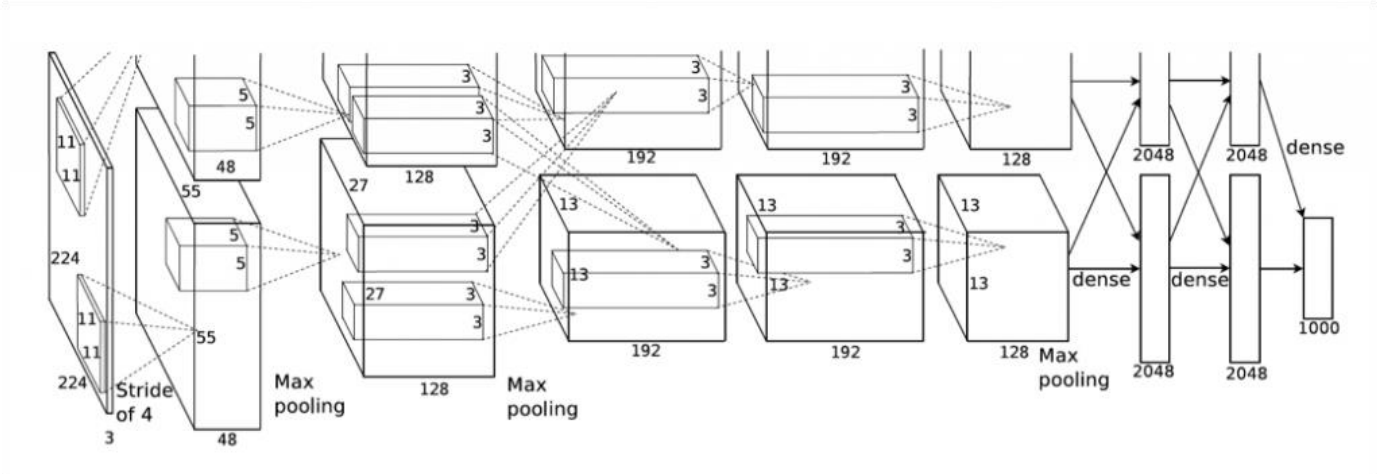


Figure 11 : décomposition d'un réseau AlexNet : Une visite d'AlexNet — Programmation Informatique — DATA SCIENCE

e) ZFNet

Cette architecture a permis de réduire le taux d'erreur jusqu'à 14.8% en 2013. Il réutilise AlexNet, en ajustant correctement ses hyperparamètres et en y ajoutant encore quelques filtres.

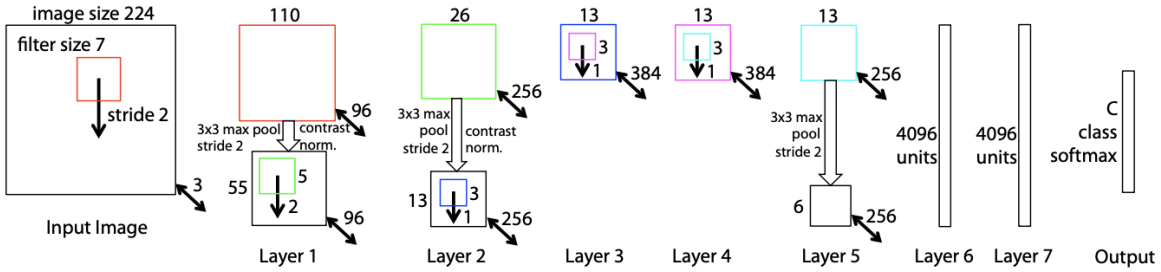


Figure 12 : décomposition d'un réseau ZFNet : [Key Deep Learning Architectures - ZFNet – Max Pechyonkin]

f) VGG

C'est une architecture constituée de 16 couches de convolution mais elle contient 138 millions de paramètres, ce qui la rend moins intéressante que ses concurrents

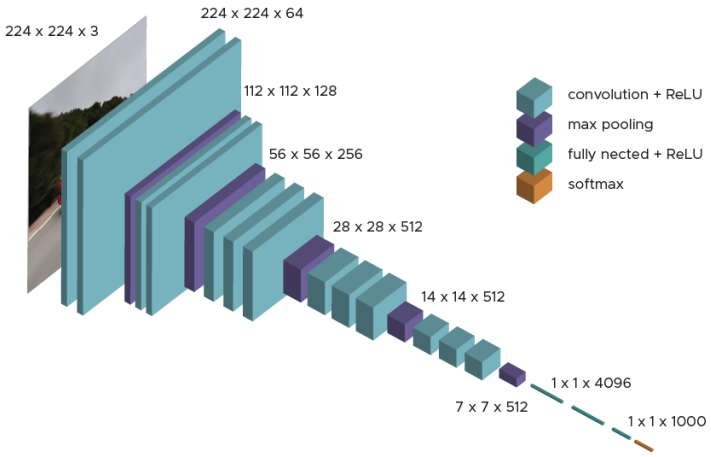


Figure 13 : décomposition d'un réseau VGG : [VGG : en quoi consiste ce modèle ? Daniel vous dit tout ! (datascientest.com)]

III. Méthodologie

Notre projet se basant sur l'IA, nous avons en premier lieu effectué des recherches sur le langage ainsi que les librairies à utiliser. Par la suite, nous avons organisé la mise en place de modèles d'IA pour répondre à notre problématique.

1. Les librairies

Notre projet repose en grande partie sur l'utilisation de librairies pour le Deep Learning.

Notre choix s'est fait entre les suivantes :

Librairie	Languages	Machine Learning	Deep Learning
Tensorflow/Keras	C++, Python, Java	Oui	Oui
Scikit-learn	Python	Oui	Non
PyTorch	Python	Oui	Oui

a) Tensorflow/Keras

Tensorflow et Keras sont des libraires développées pour le Machine Learning et le Deep Learning.

Elle peut être utilisée dans beaucoup de langages différents, que ce soit Python, C++, Java, etc…

b) Scikit-learn

Scikit-learn est une librairie pour le Machine Learning disponible pour Python.

c) PyTorch

PyTorch est une libraire pour le Machine Learning mais surtout développée pour le Deep Learning.

Etant donné que Scikit-learn n'est pas une librairie spécialisée pour le Deep Learning, notre avons décidé de faire un choix entre PyTorch et Tensorflow/Keras.

Tensorflow et Keras étant les libraires avec le plus de modèles pré-implémentés et sa grande popularité sur internet. Nous avons ainsi décidé de commencer avec tensorflow et keras.

2. Le pré-traitement des images

Dans notre cas, l'étude des émotions ne se basent pas sur quelques couleurs. Nous avons donc dans un premier temps un traitement à faire sur les images pour toutes les passées en niveau de gris (NDG).

Nous permettant de ce fait de diviser par trois le nombre de paramètre de l'image à prendre en compte par nos modèles.

Pour ce faire, on utilise tout simplement la formule :

$$NDG = \frac{R + G + B}{3}$$

La formule est appliquée sur chaque pixel de chaque image.

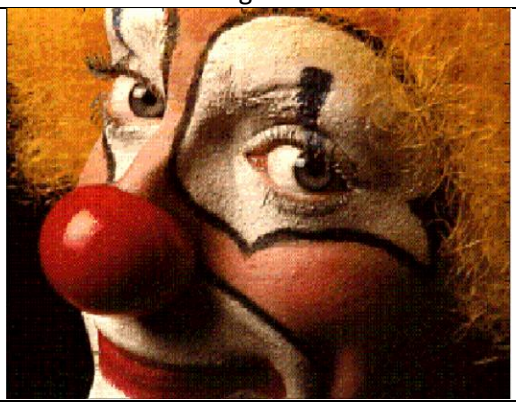
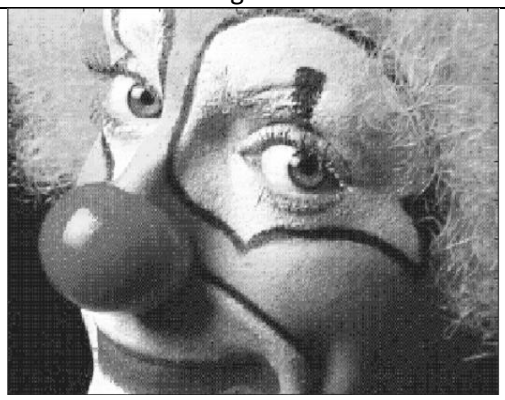
Image RGB	Image NDG
	

Tableau 2 : Comparaison d'image en RGB et NDG

3. Le modèle de réseaux de neurones

Le choix du type de réseau de neurones s'est orienté vers le CNN.

Comme présenté dans la partie II, il s'agit du réseau de neurones spécialisé pour l'IA dans le domaine de l'image.

Notre but avec ce type de réseau de neurones va donc être de d'extraire les caractéristiques de chaque image, comme la bouche, le nez, les yeux qui pourraient nous indiquer l'état émotionnelle de la personne sur l'image.

Se référer à la partie II.4. pour les détails sur les différents réseaux de neurones.

4. Méthodologie de mise en place d'un modèle

Dans un premier temps nous avons décidé de mettre en place un CNN assez simple dont on modifierait les paramètres ainsi que sa structure au fur et à mesure des tests pour essayer d'obtenir des résultats optimums.

Le modèle VGG, est une architecture créée pour la détection de caractéristiques dans des images.

Nous avons donc décidé de nous baser sur cette dernière en commençant par un système très simple contenant une seule couche VGG à laquelle nous en rajouterons de nouvelles à chaque test d'entrainement de notre modèle.

Une couche VGG est composée de deux couches de convolution qui se suivent avec ensuite une couche de max pooling.

Dans le cas d'images de dimensions 224x224x3, notre modèle VGG1 ressemblerait au schéma suivant :

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0

5. La data augmentation

Pour notre entrainement, nous utilisons dans un premier temps des dataset très simple qui contiennent des images de personnes de face affichant diverses expressions :

Figure 14 : émotions types pouvant être trouvées dans les bases de données

Afin de rendre notre modèle plus performant, la mise en place de la Data Augmentation est un des points les plus importants pour notre modèle.

Celle-ci nous permet d'augmenter artificiellement le nombre d'images de nos dataset et de rendre notre réseau de neurones encore plus intelligent. Nous pourrons obtenir de meilleurs résultats étant donné que la fiabilité d'une IA dépend énormément de la quantité d'élément pour notre entrainement notre modèle. De plus, lui faire apprendre des images différentes de celles habituelles (légèrement penché, image plus flou, zoom), va faire apprendre notre modèle à mieux réfléchir et à apprendre ce qu'est un visage plutôt que des pixels toujours aux mêmes endroits d'une image.

Figure 15 : augmentation de données sur une image

6. Le Transfer Learning

Le « Transfer Learning » a pour principe de prendre un modèle pré-entrainé, de rajouter quelques couches (donc un nouveau classifieur) pour correspondre à nos attentes, et de réentraîner une partie de ce réseau.

Il existe 3 types de Transfer Learning :

Fine-tuning total : on réentraîne tout le réseau avec nos datasets.

Extraction de caractéristique : on n'entraine que le nouveau classifieur mais on ne touche pas au réseau pré-entraîné.

Fine-tuning partiel : c'est un mélange des deux façons de faire précédentes, c'est-à-dire qu'on ne réentraîne que le classifieur et un certain nombre de dernières couches du réseau pré-entrainé.

Dans notre cas, l'utilisation du Transfer Learning est cohérente, car il existe des modèles capables de reconnaître des tas d'objets, mais aussi des humains sur une photo. Il suffit alors de profiter de leurs connaissances pour la reconnaissance d'humains pour recentrer leurs analyses sur les émotions de ceux-ci à la toute fin.

Pour faire une analogie à la vie courante, on apprend d'abord à un enfant à reconnaître différents animaux avant d'être plus précis et de leur apprendre différentes races de chien. Dans ce cas-là, cet enfant ne prendra plus en compte les autres animaux mais il arrivera à reconnaître différentes races de canidés.

Dans la bibliothèque Keras, nous avons la possibilité d'utiliser des modèles pré-entraînés directement, nous avons donc utilisé une architecture ResNet pré-entraînée sur ImageNet, une bibliothèque de reconnaissance d'objet.

Après plusieurs tests, nous avions remarqué qu'un fine-tuning partiel était le choix le plus optimum, en entrainant seulement les 5 dernières couches du réseau.

Une fois notre architecture implémentée avec son fine-tuning, il nous manquait plus que d'avoir assez d'images pour l'entraîner, ce qui sera vu dans la partie qui suit.

Dans notre cas, l'utilisation du Transfer Learning est cohérente, car il existe des modèles capables de reconnaître des tas d'objets, mais aussi des humains sur une photo. Il suffit alors de profiter de leurs connaissances pour la reconnaissance d'humains pour recentrer leurs analyses sur les émotions de ceux-ci à la toute fin.

7. La détection de visage

La dernière étape de notre projet. Une fois de bons résultats de reconnaissance d'émotion obtenus, cela va être la détection de visage dans une image/vidéo en temps réel.

Il existe divers modèles pour la détection de visages :

1) VGG-Face
2) Google FaceNet
3) OpenFace
4) Facebook DeepFace
5) DeepID
6) Dlib
7) ArcFace

Cette partie n'étant pas incluse dans notre projet, nous avons décidé d'utiliser des librairies parmi les suivantes pour la recherche de visage.

8) OpenCV
9) Dlib
10) SSD
11) MTCNN
12) RetinaFace

Nous avons décidé d'utiliser OpenCV. Sa simplicité d'utilisation et sa grande présence dans Python nous permet de gagner énormément de temps sur la mise en place de notre modèle.

Image	Détection du visage	Nouvelle image
		

Tableau 3 : Détection d'un visage sur une image

IV. Solution proposée

1. Le pré-traitement des images

Les images pour nos divers modèles sont toutes transformés en niveau de gris. Nous permettant de focaliser notre modèle sur l'apprentissage que de données les plus importantes.

2. ResNet50

Nous avons essayé plusieurs modèles, tel que VGG qui était notre deuxième meilleure architecture, mais nous stagnions directement à 50% en utilisant le Transfer Learning. Nous nous sommes penchés alors sur ResNet, qui nous offrait de bien meilleurs résultats.

Nous avons essayé plusieurs ResNet, et le modèle qui nous donnait la meilleure précision était la version avec 50 couches, soit ResNet50

En faisant un simple entrainement sur FER2013, ResNet50 nous offrait une précision de 55%, nous nous sommes alors penchés sur l'entrainement de celui-ci afin d'atteindre un meilleur score.

Afin d'affiner les résultats de ResNet50, nous avons optimisé les choix de plusieurs hyperparamètres :

- Le nombre de dernières couches à entrainer pour un fine-tuning : en faisant plusieurs tests, nous avions remarqué que dans notre cas, si on n'entraînait que les 5 dernières couches, cela donnait une meilleure précision.
- Le learning rate de l'optimizer dans l'entraînement : nous avions remarqué que le choix optimium de ce learning rate était de 0,001.

Et nous avons fait en sorte d'utiliser un maximum de fonction qui permettent d'améliorer au fur et à mesure les résultats :

- Les callbacks (early stopping présenté dans la partie suivante)
- La data augmentation (présenté après le early stopping)
- Les fonctions de preprocessing d'image implémenté par Keras pour avoir une image parfaite à utiliser lors de l'entrainement du modèle. Chaque modèle a sa propre fonction de preprocessing.

Et enfin nous nous sommes penchés sur comment récupérer un maximum de données afin d'améliorer nos résultats au maximum (présenté après l'early stopping)

3. L'Early Stopping

La mise en place d'une régularisation avec l'early stopping va nous permettre d'arrêter l'entrainement de notre modèle lorsqu'il commence à surapprendre.

Lors de la détection d'une baisse d'apprentissage sur la courbe de précision sur le test, notre modèle va immédiatement arrêter l'apprentissage.

Le résultat nous donnant un système tout aussi précis sur les images de test et plus du tout surentraîné.

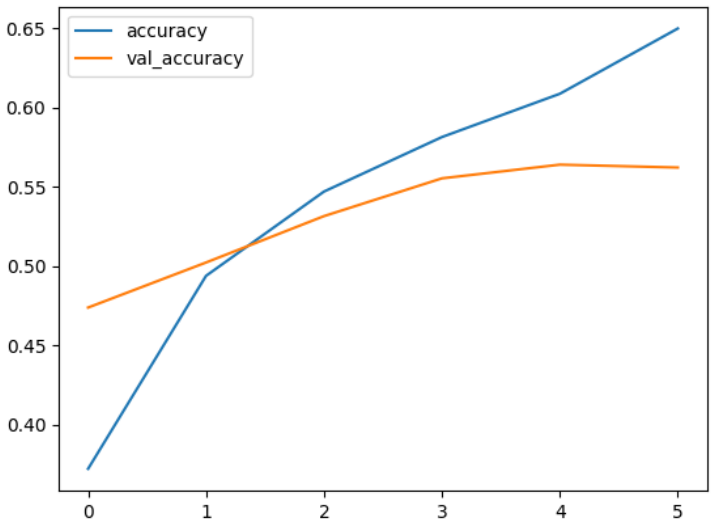


Figure 16 : exemple d'un entrainement sans early stopping

Ici lors de l'apprentissage, dès que les courbes commencent à s'éloigner, notre EarlyStopping s'active et arrête l'entrainement empêchant du surapprentissage.

4. Data augmentation

Avec de la Data Augmentation, nous sommes arrivés à des résultats un peu plus fiables, nous sommes passés à une précision de 56%.

Mais où ce système est intéressant est bien la courbe d'évolution de nos entrainements.

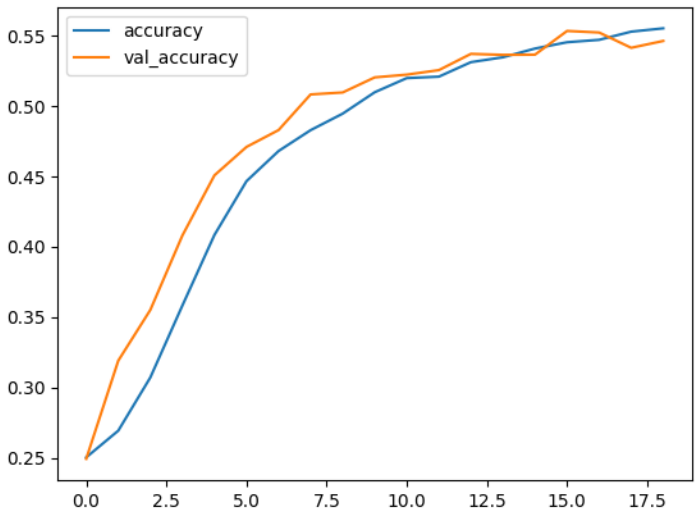


Figure 17 : exemple d'un entrainement avec Early Stopping & Data Augmentation

On remarque bien que l'évolution de notre précision de l'entrainement et notre précision sur le test se fait parfaitement en parallèle, elles arrivent à des données très similaires. Ce qui veut dire que nous n'avons pas de surapprentissage et avons donc rendu notre modèle plus intelligent au lieu de lui donner un énorme mémoire et apprendre sans réfléchir.

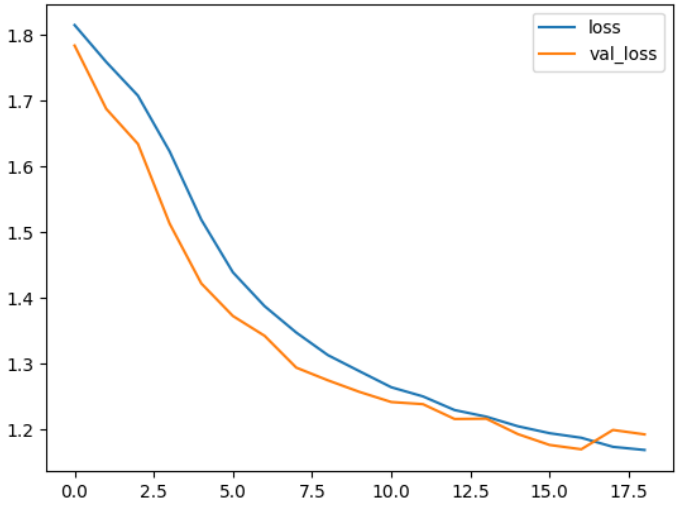


Figure 18 : pertes lors de l'apprentissage

5. Sélection de la meilleure base de données

Après divers tests, nous nous sommes vites rendu compte qu'avec FER-2013 nous obtenions une fiabilité beaucoup plus grande lors des tests qu'avec CK+.

Sur le graphe ci-dessous, l'évolution de la précision pour les images d'entrainements et de test sont totalement confondus pourtant un grand surapprentissage a lieu avec le dataset CK+.

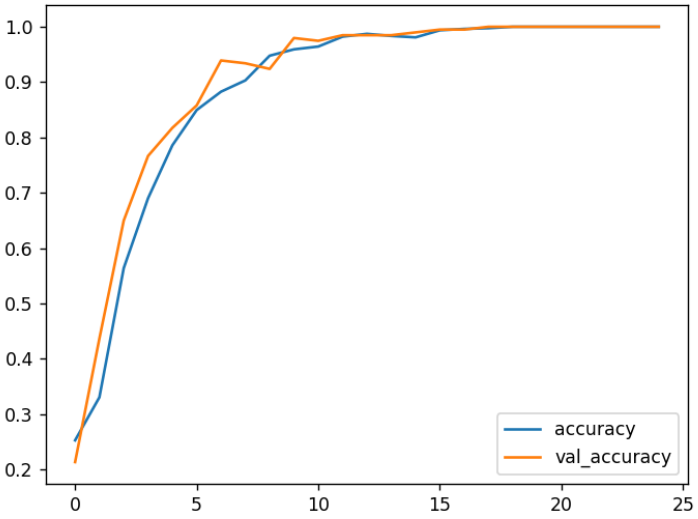


Figure 19 : entrainement avec CK+

Bien que Fer et CK+ soit des bases de données très similaires. Les images ont la même résolution et sont en niveau de gris.

Il se trouve que CK+ est remplis d'images identiques. On peut par exemple retrouver plusieurs fois la même image d'une personne faisant la même expression ce qui fait que notre modèle aura tendance à surapprendre par rapport à ces images plutôt qu'apprendre à reconnaître les émotions faciales.

Image de FER-2013	Image de CK+
	

Tableau 4 : Comparaison des images de FER-2013 et CK+

C'est pour cela que par la suite nous avons décidé d'utiliser le dataset FER-2013, mais nous voulions améliorer encore plus les résultats.

La précision sur le test et le l'entrainement sont des mesures qui nous permettent de vérifier que notre modèle est bon, et augmentent exponentiellement en fonction du nombre de données, nous sommes alors partis à la recherche de supplément à FER-2013.

Nous sommes alors tombés sur FERPLUS, proposé par Microsoft : microsoft/FERPlus: This is the FER+ new label annotations for the Emotion FER dataset. (github.com), qui rajoute beaucoup de d'images à FER2013 et qui peuvent donc améliorer les résultats.

Avec ce rajout de données, nous sommes passés d'une précision de 62% à 68%.

V. Analyse, interprétation et discussion des résultats

1. Analyse

Le résultat final est de 68.3% de précision sur les données test, pour des pertes à 0.87 (voir la photo ci-dessous pour les résultats précis)

```
Test loss: 0.8792112469673157
Test accuracy: 0.6832560300827026
```

Figure 20 : meilleurs scores que nous avons pu obtenir

Nous dessinons alors les courbes qui illustre l'entrainement :

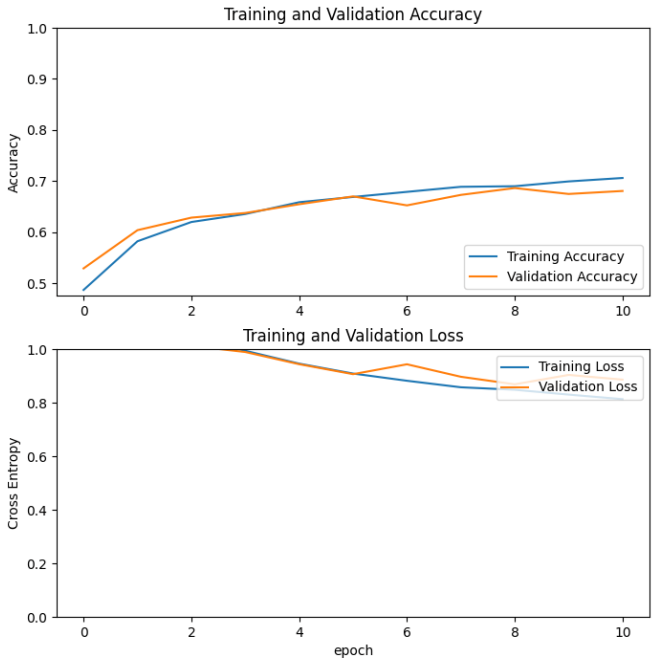


Figure 21 : courbes relatant cet entraînement

Et nous remarquons que nous n'avons pas beaucoup d'écart entre la précision de l'entrainement et la précision des données de validation, ce qui prouve que nous ne sommes pas en surapprentissage.

Comment améliorer ces résultats ?

On pourrait utiliser un ensemble de vote : on donne une image à plusieurs modèles, et la majorité décidera de quelle émotion il s'agit. On pourrait alors bénéficier des points forts de chacun des modèles et donc sensiblement améliorer les résultats

On pourrait aussi utiliser un cascading : on extrait des caractéristiques par un modèle et on les analyse par un autre modèle, ce qui peut aussi apporter des précisions.

2. Résultats

Nous avons implémenté une IHM qui récupère notre entrée caméra, qui analyse chaque frame à la recherche d'un visage, et s'il en trouve un, essaye de prédire son émotion.

Nous avons alors essayé plusieurs émotions :

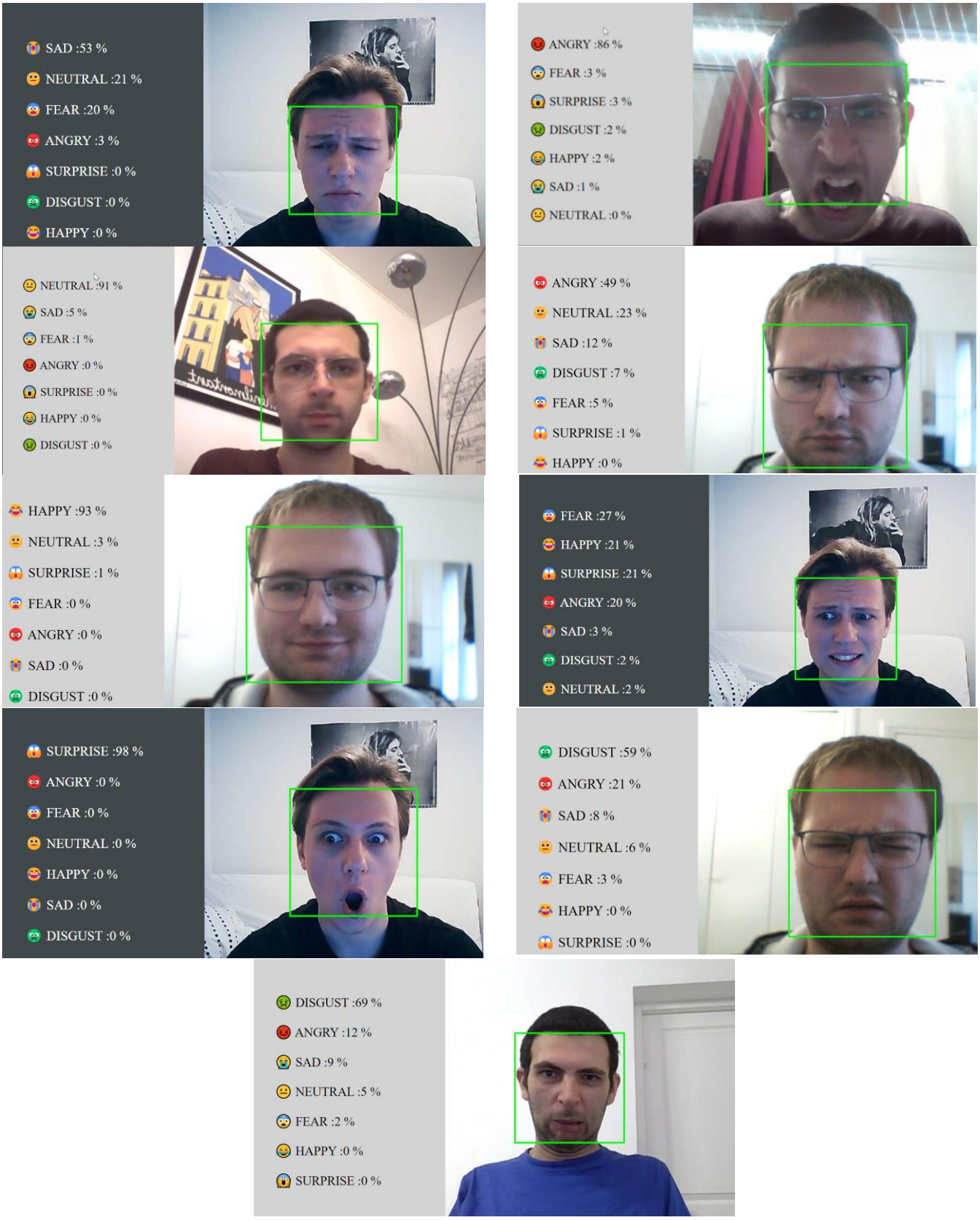


Figure 22 : Utilisation du modèle avec notre interface graphique.

VI. Conclusion et perspectives

Conclusion :

Notre projet consiste en l'utilisation du Deep Learning, qui est un type d'Intelligence Artificielle. Ce dernier relève les caractéristiques toutes seules. En effet, relever des caractéristiques manuellement sur le thème de la reconnaissance d'émotions faciales peut être très compliqué à gérer par voie humaine, et donc à implémenter dans un algorithme de Machine Learning. L'utilisation du Deep Learning est donc essentielle dans notre cas.

Aujourd'hui le Deep Learning est un domaine accessible, grâce à des bibliothèques comme Keras, PyTorch et autres, qui fournissent des modèles pré-entraînés qu'on n'a plus qu'à modifier si besoin. Ces bibliothèques vont nous fournir des architectures très connues comme ResNet, AlexNet, VGG… avec à chaque fois plusieurs versions avec un nombre différent de couche, selon les besoins. Ces architectures sont des réseaux de neurones, tous présentés lors de l'ImageNet Large Scale Visual Recognition Challenge (ILSVRC), un concours qui met à l'épreuve toutes les architectures d'analyse d'image.

Nous avons alors essayé plusieurs architectures pré-entraînées sur ImageNet. Grâce aux connaissances de base de ces architectures, en appliquant du Transfer Learning, nous avons pu concentrer leurs analyses sur les émotions des personnes. Nous avons alors entraîné ces architectures sur de petits lots de données pour les comparer entre eux, et nous avons alors choisi de travailler avec ResNet50 qui nous offrait les meilleurs résultats. La suite consistait alors de choisir les meilleurs paramètres pour notre architecture correspondant à notre problème mais surtout récupérer un maximum de données.

Nous nous sommes donc mis à la recherche de beaucoup de données car elles jouent un rôle très important dans la précision de l'algorithme. Celle-ci va dépendre du nombre d'images présentes dans la base. Plus nous avons de données d'entrainement, plus la précision augmente exponentiellement.

Afin de toujours améliorer les résultats, nous avons utilisé de l'augmentation de données, chaque image sera dupliquée avec à chaque fois des modifications. Le but est de tordre les données d'entrainement pour que notre réseau de neurone ne soit pas sensible à des paramètres comme le fait d'analyser une tête penchée de 20° par exemple.

L'ajout de données en plus dans notre algorithme et l'utilisation de l'augmentation de données, nous a permis de sensiblement augmenter la précision de l'algorithme, passant de 60 à 68.3%. Ce qui prouve que le nombre de données d'entrainement est une part extrêmement importante, l'Intelligence Artificielle ayant besoin d'un maximum d'exemples.

Les systèmes de reconnaissance d'émotions faciales seront certainement de plus en plus utilisés à l'avenir, certains voudront les utiliser pour régler des affaires judiciaires en analysant les micro-expressions de certains suspects, ou encore analyser les expressions des élèves pour dresser un profil psychologique de ceux-ci ou voir s'ils ont correctement compris ce que l'on essaye de leur apprendre. Bien sûr, dans un cas idéal, lorsque nous essayons d'avoir un maximum de pensées saines, cela pourrait potentiellement aider, mais une limite doit être définie.

Est-ce que l'usage de cette technologie est considéré comme une pratique éthique ?

Perspectives :

Etant donné que nous sommes arrivés à obtenir des résultats plutôt satisfaisants. Chaque émotion peut être identifiée par notre IA sur de nouvelles têtes. L'analyse en temps réel fonctionne également avec une réactivité plutôt acceptable. Nous sommes maintenant en train de réfléchir pour faire un vrai site web de ce programme, accessible par tout le monde et sans installation de programmes externes pour récupérer la caméra.

VII. Bibliographie

1. Documents techniques

Documentation de Tensorflow : https://www.tensorflow.org/api_docs

Documentation de Keras : https://keras.io

Documentation sur OpenCV : https://opencv.org

2. Ouvrages

Deep Learning avec Keras et Tensorflow (Documentation de O'Reilly, Aurélien Géron)

3. Références électroniques

Datacorner (Deep Learning basis) : https://www.datacorner.fr/image-processing-7/

Datacorner (transfer learning) : https://www.datacorner.fr/vgg-transfer-learning/

Vie-publique : https://www.vie-publique.fr/

How to Develop a CNN : https://machinelearningmastery.com/

FEC dataset : https://research.google/tools/datasets/google-facial-expression/

CNN, ANN, RNN : https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/

4. Images

Schéma ResNet50 : https://docs.ecognition.com/Resources/Images/ECogUsr/UG_CNN_scheme.png

Machine Learning ou Deep Learning : https://lawtomated.com/wp-content/uploads/2019/04/MLvsDL.png

Schéma Neurone simple : https://img.blogduwebdesign.com/articles/11366/images/neuroneart.png

Schéma Réseau de Neurones : https://www.juripredis.com/upload/actualites/Machine_learning/reseaux_neurones_feed_forwarded_2.png