



# Rapport projet de fin d'année

Ingénierie des Systèmes Intelligents

---

## Real Time Emotion Recognition from Facial Expressions

---

Réalisé :

- EL MAADOUDI Mohamed
- BOUFNICHEL Yassin

Encadré par:

MESBAH Abderrahim

## Table des matières

Remerciements . . . . .	4
Résumé . . . . .	5
Abstract . . . . .	6
<b>1 Introduction</b>	<b>7</b>
<b>2 Analyse et conception</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.2 Conception : . . . . .	10
2.2.1 Diagramme des cas d'utilisations: . . . . .	10
2.2.2 Diagramme de définitions des blocks : . . . . .	11
2.3 Conclusion : . . . . .	11
<b>3 Réalisation du projet</b>	<b>12</b>
3.1 Introduction . . . . .	12
3.2 Architecture du projet . . . . .	12
3.3 Structure du projet . . . . .	13
3.4 Outils et technologies utilisées . . . . .	14
3.4.1 Travail proposé . . . . .	14
3.4.2 Langage de développement et framework utilisés: . . . . .	15
3.4.3 Outils: . . . . .	16
3.4.4 Présentation du modèle . . . . .	18
3.4.5 Présentation de l'application Windows . . . . .	25
3.4.6 Présentation de l'application Mobile . . . . .	26

<b>4</b>	<b>Conclusion et Perspectives</b>	<b>28</b>
4.1	Conclusion . . . . .	28
4.2	Perspectives . . . . .	30
	Webographie . . . . .	31

# Remerciements

Tout d'abord nous remercions le bon Dieu, le clément de nous avoir donné la foi et le courage de terminer ce projet.

Ce projet est le fruit de conseils et critiques bienveillantes d'un grand nombre de personnes. Nous tenons à les remercier ici et leur faire part de toute notre gratitude pour avoir été toujours d'une aide précieuse.

Nous tenons donc à adresser plus particulièrement nos remerciements les plus sincères à tout notre encadrant Mr. MESBAH Abderrahim , pour nous avoir offert cette opportunité d'apprentissage, pour son encadrement et conseils.

Nous nous acquittons, également, volontiers d'un devoir de gratitude et de remerciements à Mr. BERRAHOU aissam responsable de la filière SSE, ainsi que tout le corps professoral de la filière SSE pour leur disponibilités et le savoir faire qui nous ont transmis.

Enfin, nous témoignons notre profonde gratitude aux membres du jury, nous ayant fait l'honneur de juger notre travail et l'Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes, pour le savoir qu'elle nous offre jour après jour. Qu'il nous soit également permis de remercier tout le corps professoral et administratif de l'école, ainsi que toute personne ayant contribué à bien mener ce projet.

# Résumé

La reconnaissance automatique des émotions basée sur l'expression faciale est un domaine de recherche intéressant, qui a été présenté et appliqué dans plusieurs domaines tels que la sécurité, la santé et les interfaces homme-machine. Les chercheurs dans ce domaine s'intéressent au développement de techniques pour interpréter, coder les expressions faciales et extraire ces caractéristiques afin d'avoir une meilleure prédiction par ordinateur.

Avec le succès remarquable du deep learning, les différents types d'architectures de cette technique sont exploités pour atteindre une meilleure performance.

Le but de ce projet est de faire une étude sur les travaux récents sur la reconnaissance automatique des émotions faciales FER via le deep learning. Nous soulignons sur ces apports traités, l'architecture et les bases de données utilisées et nous présentons les progrès réalisés en comparant les méthodes proposées et les résultats obtenus, l'objectif principal de notre projet consiste à développer un cadre en temps réel basé sur deep learning pour détecter et reconnaître les l'émotions du visages humains.

# Abstract

Automatic emotion recognition based on facial expression is an interesting research field, which has presented and applied in several areas such as safety, health and in human machine interfaces. Researchers in this field are interested in developing techniques to interpret, code facial expressions and extract these features in order to have a better prediction by computer.

With the remarkable success of deep learning, the different types of architectures of this technique are exploited to achieve a better performance.

The purpose of this project is to make a study on recent works on automatic facial emotion recognition FER via deep learning. We underline on these contributions treated, the architecture and the databases used and we present the progress made by comparing the proposed methods and the results obtained, the main objective of our project is to develop a real-time framework based on deep learning to detect and recognize the emotions of human faces.

# Chapter 1

## Introduction

Dans le monde d'aujourd'hui, il existe un certain nombre de techniques d'application utilisées pour la reconnaissance faciale, par exemple la reconnaissance faciale bio-matrice, le système de paiement, l'accès et la sécurité, l'identification criminelle, la publicité, les soins de santé.

Le problème posé par ces applications de reconnaissance des expressions faciales est encore une recherche en cours, tout d'abord dans la sécurité routière, détecter automatiquement de la colère ou de la tristesse pour diminuer le taux des accidents et avoir une voiture autonome qui connaitre les émotions de son chauffeur.

Dans ce projet, je présente le système d'un système artificiellement intelligent qui est capable de reconnaître les émotions à travers les expressions faciales. Ce système distinguant les sept émotions universelles : joie, dégoût, colère, peur, tristesse, surprise et neutre. En cela, il existe différentes couches utilisées dans le réseau de neurones, après quoi le réseau le plus performant est encore optimisé et renvoie finalement l'émotion de l'utilisateur au système de voiture pour aider à éviter des accidents de la route en réagissant avant qu'ils ne se produisent.

Dans le cadre de notre formation au sein de l'Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes et dans le cadre du projet fédérateur du 1er semestre du cursus ingénieur de la filière SSE, on est amené à réaliser un projet dans Nous proposons un modèle informatique pour la détection des émotions basé sur le comportement humain.

Le présent rapport a pour but de retracer le travail accompli durant ces 2 semestres en s'articulant autour de 4 grands chapitres. Il présentera en premier temps une Introduction générale. Puis se focalisera dans le deuxième chapitre sur notre analyse conceptuelle. Puis dans le troisième est dédiée à la réalisation et la mise en œuvre, nous allons y présenter les outils, langages, et frameworks utilisés ainsi que l'acheminement des interfaces réalisées. Enfin, le dernier chapitre est dédié à une conclusion générale.



## Chapter 2

# Analyse et conception

### 2.1 Introduction

La reconnaissance des émotions du visage laissent penser que pouvoir automatiquement détecter différents états émotionnels pourrait s'avérer particulièrement utile dans de nombreux domaines :

**Sécurité routière :** reconnaître automatiquement de la colère ou de la tristesse pourrait, par exemple, aider à éviter des accidents de la route en réagissant avant qu'ils ne se produisent.

**Santé :** des applications d'aide au diagnostic permettraient de détecter certains signaux de maladies comme la dépression.

**Jeux vidéos :** l'expérience d'un joueur pourrait être améliorée en adaptant automatiquement la difficulté du jeu en fonction du retour du joueur.

**E-learning :** lors de leçons à distance, la caméra face à l'apprenant pourrait permettre de détecter ses incompréhensions ou ses frustrations pour adapter automatiquement la méthode d'apprentissage.

## 2.2 Conception :

### 2.2.1 Diagramme des cas d'utilisations:

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en oeuvre.

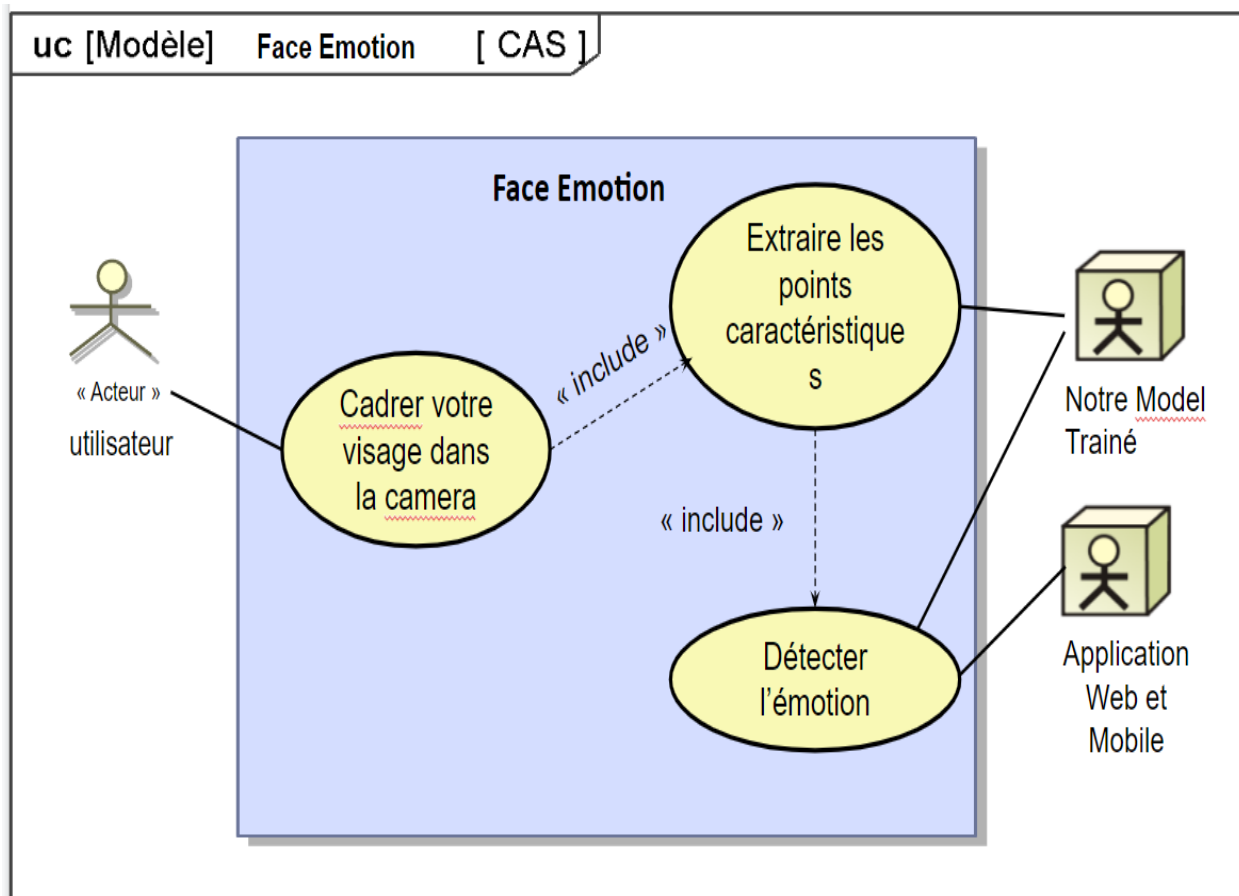


Figure 2.1 :Diagramme de cas d'utilisation générale

### 2.2.2 Diagramme de définitions des blocks :

Le diagramme de définition de blocs (bdd) permet de dresser la liste des différents constituants du système de façon hiérarchique.

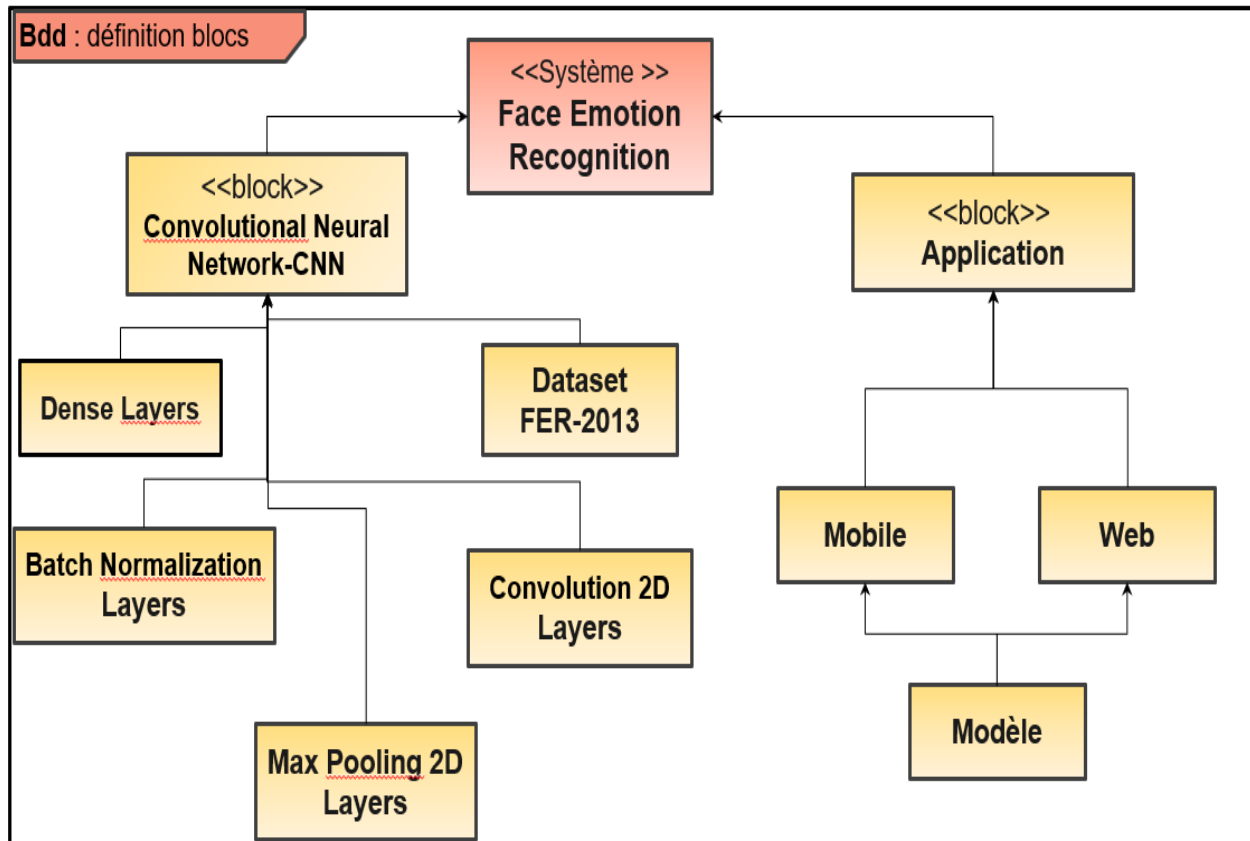


Figure 2.2 :Diagramme de définitions des blocks

### 2.3 Conclusion :

Dans ce chapitre, nous avons décrit la phase d'analyse et conception de notre projet. Et nous avons présenté quelques diagrammes du formalisme UML, relatifs à notre projet afin d'illustrer son fonctionnement. Le chapitre suivant est dédié à la phase de réalisation de notre application.

## Chapter 3

# Réalisation du projet

### 3.1 Introduction

Ce chapitre est consacré à la phase réalisation du projet. Tout d'abord, on va présenter quelques outils et frameworks utilisés dans la mise en œuvre de l'analyse de la série et la réalisation concrète du modèle ainsi que la création de l'application mobile et l'architecture adaptée.

### 3.2 Architecture du projet

Dans le domaine du deep Learning, il existe un ensemble spécifique de méthodes et de techniques particulièrement bien adaptées pour atteindre une meilleur performance. Dans ce projet, nous aborderons la méthode CNN.

Ces dernières années, l'apprentissage en profondeur a été une approche très réussie et efficace grâce au résultat obtenu avec ses architectures qui permettent l'extraction automatique de caractéristiques et la classification telles que la convolution réseau de neurones CNN.

### 3.3 Structure du projet

Ce projet comporte cinq volets:

**Emotion-Detection.py** : Cela contient le code en python du projet.

**App-face-emotion.py** : Cela contient une application pour tester la validation du model.

**haarcascade-frontalface-default.xml** : Il s'agit d'un fichier XML contenant un détecteur de visages en cascade Haar sérialisé (algorithme Viola-Jones) dans la bibliothèque OpenCV.

**model.h5** : Contient le code du modèle d'apprentissage automatique pour détecter les émotions du visage humain en temps réel.

**main.dart et home.dart** : Contients le code de l'application mobile (android) réalisée à l'aide de flutter.

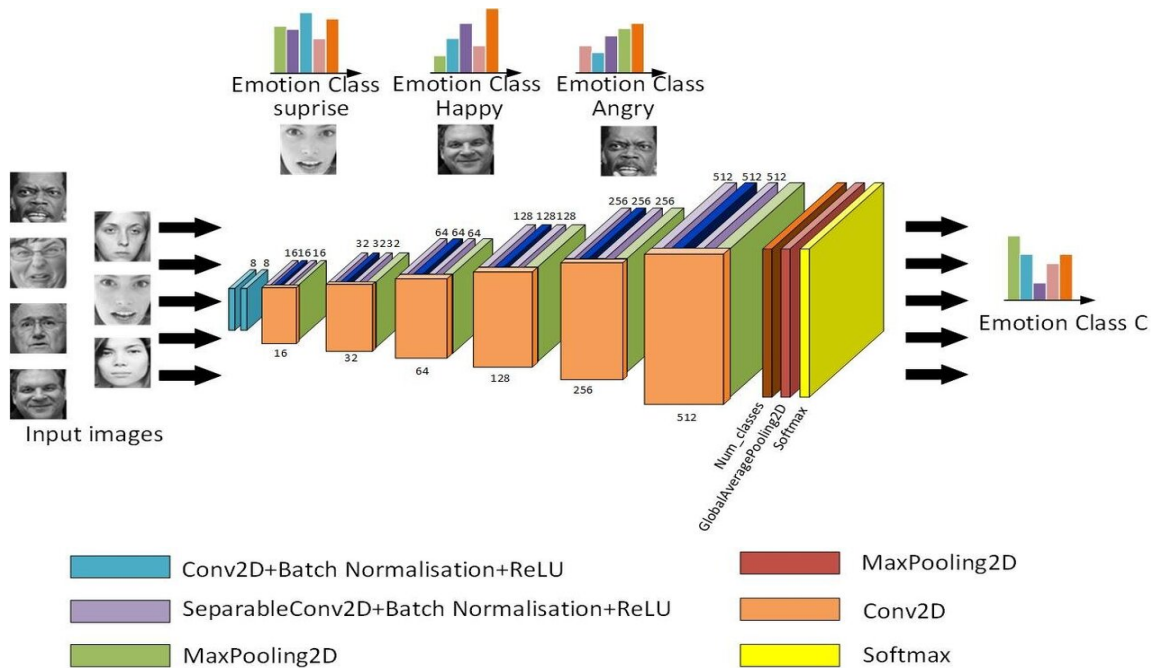


Figure 3.1 : Facial Expression Recognition Using Deep Convolutional Neural Networks

### 3.4 Outils et technologies utilisées

#### 3.4.1 Travail proposé

- J'utilise la base de données de visages FER-2013, qui est un ensemble de 28 709 photos de personnes affichant 7 expressions émotionnelles en colère, dégoûté, craintif, heureux, triste, surpris et neutre.
- Extraire les caractéristiques parmi la base de données au dessus.
- Former mon modèle en utilisant les techniques CNN et certaines fonctions pré-définies.

- Enregistrer mon modele et l'utilisé dans notre application windows et mobile.
- Utilisant flutter et android studio pour développer une application mobile, qui se base sur notre modèle.

### 3.4.2 Langage de développement et framework utilisés:

**Python:** est le langage de programmation open source le plus employé par les informaticiens.

Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens.



**Flutter:** est un kit de développement logiciel d'interface utilisateur open-source créé par Google. Il est utilisé pour développer des applications pour Android, iOS, Linux, Mac, Windows, Google Fuchsia et le web à partir d'une seule base de code.



### 3.4.3 Outils:

**Jupyter** : JupyterLab est un environnement de développement interactif basé sur le Web pour les blocs-notes, le code et les données Jupyter. JupyterLab est flexible : configurez et organisez l'interface utilisateur pour prendre en charge un large éventail de flux de travail en science des données, en calcul scientifique et en apprentissage automatique. JupyterLab est extensible et modulaire : écrivez des plugins qui ajoutent de nouveaux composants et s'intègrent à ceux existants.



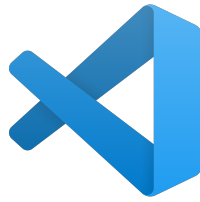
**Google Colab** : Colaboratory, souvent raccourci en "Colab", est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine learning, à l'analyse de données et à l'éducation. En termes plus techniques, Colab est un service hébergé de notebooks Jupyter qui ne nécessite aucune configuration et permet d'accéder gratuitement à des ressources informatiques, dont des GPU.



**Visual Studio Code** : Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code,



les snippets, la refactorisation du code et Git intégré.



**Android Studio** : Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux.



**Overleaf LaTeX** : est un éditeur LaTeX en ligne. LaTeX lui est un langage et un système de composition de documents. LaTeX permet de rédiger des documents dont la mise en page est réalisée automatiquement en se conformant du mieux possible à des normes typographiques.



### 3.4.4 Présentation du modèle

#### 1. Dataset:

- Nous allons utiliser une base de données intitulée FER2013, se compose d'images de visages en niveaux de gris de 48x48 pixels.
- Les visages ont été automatiquement enregistrés afin que le visage soit plus ou moins centré et occupe à peu près la même quantité d'espace dans chaque image.
- La tâche consiste à classer chaque visage en fonction de l'émotion montrée dans l'expression faciale dans l'une des sept catégories (0 = en colère, 1 = dégoût, 2 = peur, 3 = heureux, 4 = triste, 5 = surprise, 6 = neutre) . L'ensemble de train se compose de 28 709 exemples et l'ensemble de test public se compose de 3 589 exemples.

#### (a) Problèmes de la base de données:

- Problème de déséquilibre.
- Variation intra-classe du FER.
- Occlusion.
- Variation de contraste.
- Eyeglasses.

Dataset							
FER2013	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Training	3995	436	4097	7215	4830	3171	4965
Validation	958	111	1024	1774	1247	831	1233

## 2. Convolutional Neural Network-CNN

- Il est maintenant temps de concevoir le modèle CNN pour la détection des émotions avec différentes couches. Nous commençons par l'initialisation du modèle suivie d'une couche de normalisation par lots (Batch Normalization) puis de différentes couches de convolutions avec ReLu comme fonction d'activation, de couches de MaxPool et dropouts pour effectuer un apprentissage efficace. Vous pouvez également modifier l'architecture en initiant les couches de vos choix avec différents nombres de neurones et de fonctions d'activation.

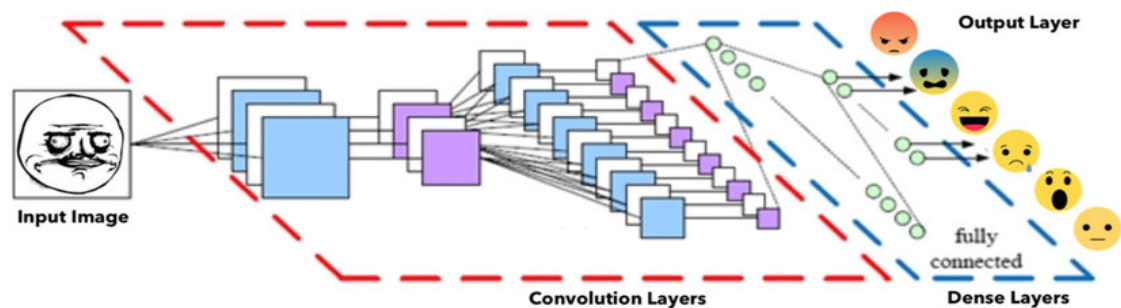


Figure 3.2 :CNN

- **Augmentation des données:**

Plus de données sont générées à l'aide de l'ensemble d'apprentissage en appliquant des transformations. Il est nécessaire si l'ensemble d'apprentissage n'est pas suffisant pour apprendre la représentation. Les données d'image sont générées en transformant les images d'entraînement réelles par rotation, recadrage, décalages, cisaillement, zoom, retournement, réflexion, normalisation, etc.

```
batch_size = 128

datagen_train = ImageDataGenerator()
datagen_val = ImageDataGenerator()

train_set = datagen_train.flow_from_directory(folder_path+"train",
                                              target_size = (picture_size,picture_size),
                                              color_mode = "grayscale",
                                              batch_size=batch_size,
                                              class_mode='categorical',
                                              shuffle=True)

test_set = datagen_val.flow_from_directory(folder_path+"validation",
                                           target_size = (picture_size,picture_size),
                                           color_mode = "grayscale",
                                           batch_size=batch_size,
                                           class_mode='categorical',
                                           shuffle=False)
```

Found 28709 images belonging to 7 classes.  
Found 7178 images belonging to 7 classes.

Figure 3.3 : Data Augmentation

## • Convolution 2D :

La convolution 2D est une opération assez simple au fond : vous commencez avec un noyau, qui est simplement une petite matrice de poids. Ce noyau "glisse" sur les données d'entrée 2D, effectuant une multiplication élément par élément avec la partie de l'entrée sur laquelle il se trouve actuellement, puis résumant les résultats en un seul pixel de sortie.

```
model.add(Conv2D(64,(3,3),padding = 'same',input_shape = (48,48,1)))
model.add(Conv2D(128,(5,5),padding = 'same'))
model.add(Conv2D(512,(3,3),padding = 'same'))
model.add(Conv2D(512,(3,3), padding='same'))
```

Figure 3.4 : Conv2D

- **Batch Normalization :**

La normalisation par lots réduit la quantité de décalage des valeurs unitaires cachées (décalage de covariance). On applique cette fonction pour les 4 couches de notre model.

```
model.add(BatchNormalization())
```

Figure 3.5 : Batch Normalize

- **Max Pooling 2D :**

In case of Max Pooling, a kernel of size  $n \times n$  is moved across the matrix and for each position the max value is taken and put in the corresponding position of the output matrix.

```
model.add(MaxPooling2D(pool_size = (2,2)))
```

Figure 3.6 : Max Pooling 2D

- Après cela, nous compilons le modèle en utilisant Adam comme optimiseur, la perte comme entropie croisée catégorielle et les métriques comme précision, comme indiqué dans le code ci-dessous.

```
opt = Adam(lr = 0.0001)
model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

Model: "sequential\_2"

Figure 3.7 : Compilés notre modèle

La fonction de perte utilisée est categorical\_crossentropy. La fonction de perte mesure simplement la différence absolue entre notre prédiction et la valeur réelle.

L'optimiseur utilisé est l'optimiseur Adam(), signifie Adaptive Moment Estimation, est une autre méthode qui calcule les taux d'apprentissage adaptatifs pour chaque paramètre.

- Après avoir compilé le modèle, nous ajustons ensuite les données pour la formation et la validation. Ici, nous prenons batch size égale à 128 avec 48 epoch.

```
Epoch 1/48
224/224 [=====] - ETA: 0s - loss: 1.7868 - accuracy: 0.3202WARNING:tensorflow:Can save best model only with val_acc available, skipping.
224/224 [=====] - 827s 4s/step - loss: 1.7868 - accuracy: 0.3202 - val_loss: 1.6450 - val_accuracy: 0.3765 - lr: 0.0010
Epoch 2/48
224/224 [=====] - ETA: 0s - loss: 1.4310 - accuracy: 0.4487WARNING:tensorflow:Can save best model only with val_acc available, skipping.
224/224 [=====] - 860s 4s/step - loss: 1.4310 - accuracy: 0.4487 - val_loss: 1.3747 - val_accuracy: 0.4595 - lr: 0.0010
Epoch 3/48
224/224 [=====] - ETA: 0s - loss: 1.2693 - accuracy: 0.5149WARNING:tensorflow:Can save best model only with val_acc available, skipping.
224/224 [=====] - 788s 4s/step - loss: 1.2693 - accuracy: 0.5149 - val_loss: 1.2868 - val_accuracy: 0.5022 - lr: 0.0010
Epoch 4/48
224/224 [=====] - ETA: 0s - loss: 1.1867 - accuracy: 0.5488WARNING:tensorflow:Can save best model only with val_acc available, skipping.
224/224 [=====] - 877s 4s/step - loss: 1.1867 - accuracy: 0.5488 - val_loss: 1.2355 - val_accuracy: 0.5280 - lr: 0.0010
Epoch 5/48
224/224 [=====] - ETA: 0s - loss: 1.1289 - accuracy: 0.5725WARNING:tensorflow:Can save best model only with val_acc available, skipping.
224/224 [=====] - 1264s 6s/step - loss: 1.1289 - accuracy: 0.5725 - val_loss: 1.1599 - val_accuracy: 0.5590 - lr: 0.0010
Epoch 6/48
224/224 [=====] - ETA: 0s - loss: 1.0710 - accuracy: 0.5935WARNING:tensorflow:Can save best model only with val_acc available, skipping.
224/224 [=====] - 1257s 6s/step - loss: 1.0710 - accuracy: 0.5935 - val_loss: 1.1962 - val_accuracy: 0.5555 - lr: 0.0010
Epoch 7/48
224/224 [=====] - ETA: 0s - loss: 1.0258 - accuracy: 0.6156WARNING:tensorflow:Can save best model only with val_acc available, skipping.
224/224 [=====] - 1208s 5s/step - loss: 1.0258 - accuracy: 0.6156 - val_loss: 1.1526 - val_accuracy: 0.5569 - lr: 0.0010
Epoch 8/48
224/224 [=====] - ETA: 0s - loss: 0.9840 - accuracy: 0.6293WARNING:tensorflow:Can save best model only with val_acc available, skipping.
ng.
```

Figure 3.8: Entraînement du modèle

```
224/224 [=====] - 1072s 5s/step - loss: 0.9840 - accuracy: 0.6293 - val_loss: 1.1621 - val_accuracy: 0.5691 - lr: 0.0010
Epoch 9/48
224/224 [=====] - ETA: 0s - loss: 0.9378 - accuracy: 0.6487WARNING:tensorflow:Can save best model only with val_acc available, skipping.
ng.
224/224 [=====] - 1123s 5s/step - loss: 0.9378 - accuracy: 0.6487 - val_loss: 1.1668 - val_accuracy: 0.5756 - lr: 0.0010
Epoch 10/48
224/224 [=====] - ETA: 0s - loss: 0.8994 - accuracy: 0.6579WARNING:tensorflow:Can save best model only with val_acc available, skipping.
ng.
224/224 [=====] - 1161s 5s/step - loss: 0.8994 - accuracy: 0.6579 - val_loss: 1.1130 - val_accuracy: 0.5838 - lr: 0.0010
Epoch 11/48
224/224 [=====] - ETA: 0s - loss: 0.8503 - accuracy: 0.6812WARNING:tensorflow:Can save best model only with val_acc available, skipping.
ng.
224/224 [=====] - 1059s 5s/step - loss: 0.8503 - accuracy: 0.6812 - val_loss: 1.0626 - val_accuracy: 0.6053 - lr: 0.0010
Epoch 12/48
224/224 [=====] - ETA: 0s - loss: 0.8135 - accuracy: 0.6963WARNING:tensorflow:Can save best model only with val_acc available, skipping.
ng.
224/224 [=====] - 756s 3s/step - loss: 0.8135 - accuracy: 0.6963 - val_loss: 1.2558 - val_accuracy: 0.5519 - lr: 0.0010
Epoch 13/48
224/224 [=====] - ETA: 0s - loss: 0.7659 - accuracy: 0.7128WARNING:tensorflow:Can save best model only with val_acc available, skipping.
ng.
224/224 [=====] - 774s 3s/step - loss: 0.7659 - accuracy: 0.7128 - val_loss: 1.2426 - val_accuracy: 0.5608 - lr: 0.0010
Epoch 14/48
224/224 [=====] - ETA: 0s - loss: 0.7292 - accuracy: 0.7259Restoring model weights from the end of the best epoch: 11.
WARNING:tensorflow:Can save best model only with val_acc available, skipping.

Epoch 14: ReduceLROnPlateau reducing learning rate to 0.000200000000949949026.
224/224 [=====] - 768s 3s/step - loss: 0.7292 - accuracy: 0.7259 - val_loss: 1.1402 - val_accuracy: 0.5957 - lr: 0.0010
Epoch 14: early stopping
```

Figure 3.9: Entraînement du modèle(suite)

### 3. Tracer la précision et la perte :

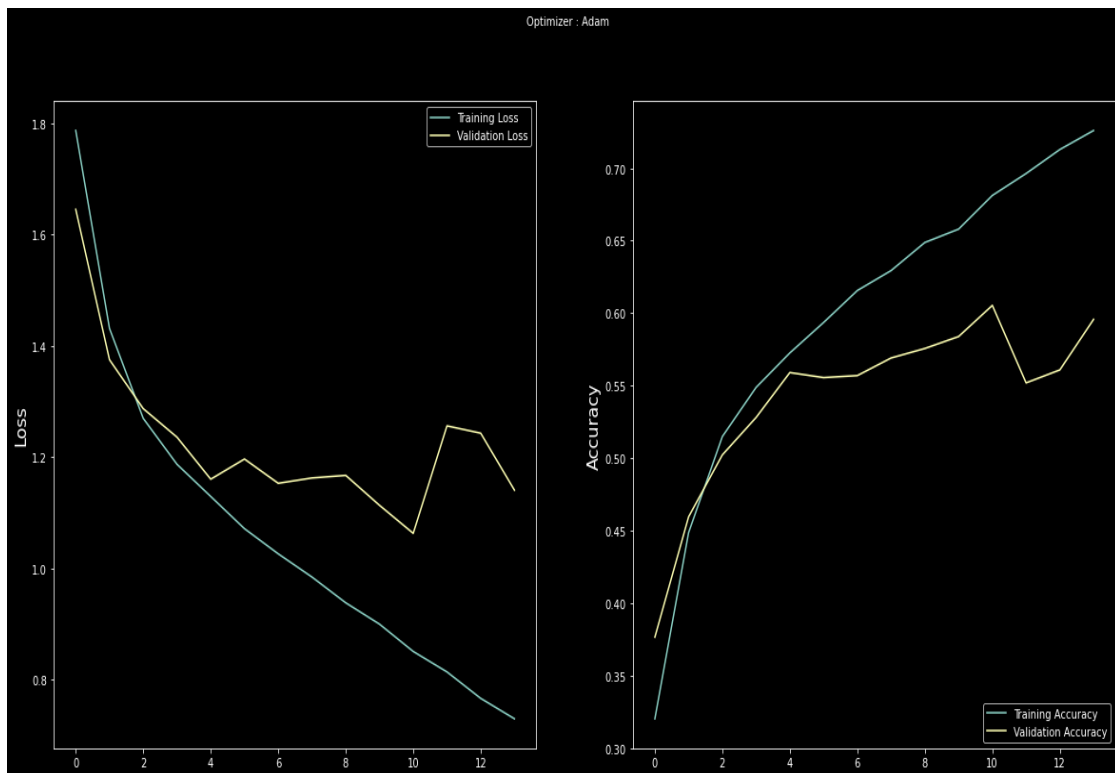


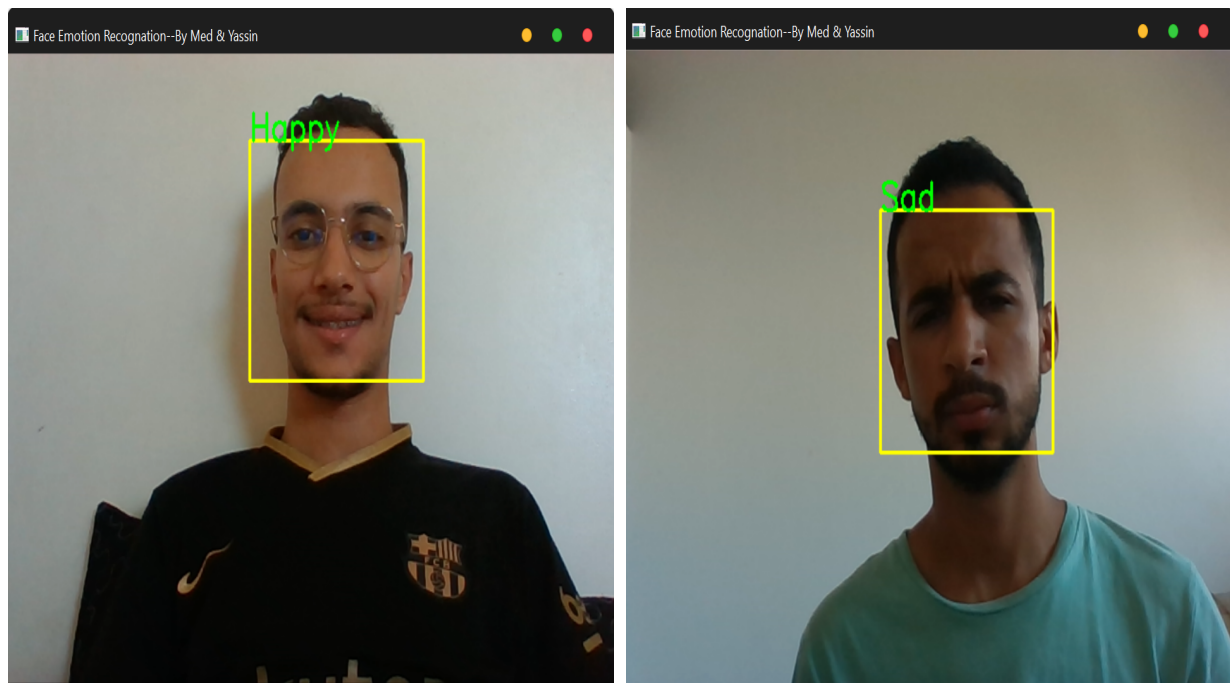
Figure 3.10 : Courbes de la precision et l'erreur

Comme on peut le voir, pour l'accuracy on obtient, pour la validation 0.65, et pour Training 0.7259.



### 3.4.5 Présentation de l'application Windows

Nous allons présenter maintenant notre application qui sert à reconnaître l'émotion d'une personne en utilisant la caméra de notre ordinateur à partir d'un modèle intégré dans notre application web. Alors cette interface fait un rectangle au tour du visage et donne directement l'émotion détectée.



### 3.4.6 Présentation de l'application Mobile

Nous allons présenter maintenant notre application qui sert à reconnaître l'émotion d'une personne en utilisant la caméra de notre smartphone à partir d'un modèle intégré dans notre application mobile. Alors cette interface détecte le visage de la personne qui utilise la caméra, après la classification et la détection de l'émotion, elle lui affiche au dessous.

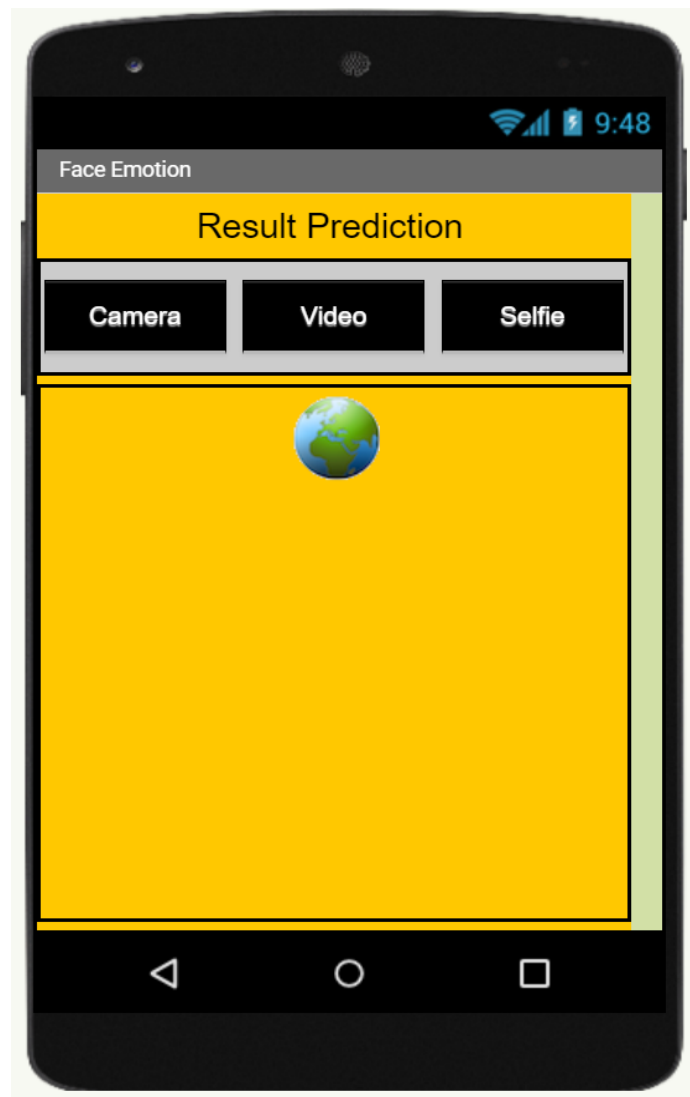


Figure 3.11: Interface graphique

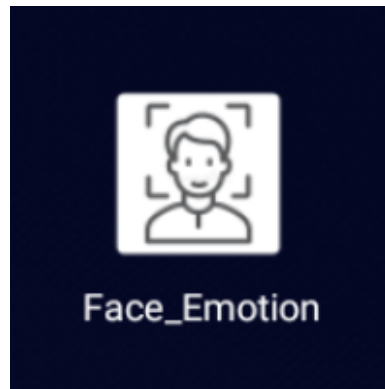


Figure 3.12: Logo

## Chapter 4

# Conclusion et Perspectives

### 4.1 Conclusion

Nous vous avons présenté tout au long de ce rapport la démarche que nous avons suivie pour mettre au point ce modèle et cette application : en commençant par la présentation du sujet et du cahier de charge, pour ensuite aborder la partie conception du modèle et de l'application et enfin présenter la réalisation finale.

L'objectif de notre travail était de développer un cadre en temps réel basé sur deep learning pour détecter et reconnaître les l'émotions du visages humains dans une application Web et Mobile.

Nous sommes arrivées à développer la totalité des fonctionnalités que nous avons prévu d'accomplir dans les premiers chapitres, nos objectifs nous permettant d'assouvir les besoins des utilisateur de notre application sont donc atteints. Nous aspirons tout de même à développer encore plus ces fonctionnalités dans le futur.

La réalisation de ce projet nous a offert une excellente opportunité d'appliquer et d'approfondir les connaissances qui nous ont été inculquées au cours de ce semestre et d'augmenter en compétence.

Ce fut pour nous une expérience enrichissante et bénéfique qui nous a permis de faire face à plusieurs contraintes et développer les techniques de recherche d'information. De plus, nous avons pu acquérir l'esprit du travail en groupe et ainsi écouter les autres, défendre nos idées, partager le travail et prendre des initiatives.

Comme cela est assez évident après de nombreuses recherches et délibérations, avoir un aperçu de ce qu'une personne peut ressentir est très précieux pour de nombreuses raisons. La portée future de ce domaine est visualisée comme étant pratiquement illimitée, avec des applications plus futuristes visibles à l'horizon, en particulier dans le domaine de la sécurité, en particulier la sécurité routière.

Notre objectif en formant un tel système est de former un protocole standard qui peut être utilisé comme composant dans de nombreuses voitures à pour but d'éviter des accidents de la route en réagissant avant qu'ils ne se produisent.

## 4.2 Perspectives

Bien que les objectifs de notre projet soient atteints, l'application que nous avons développée pourrait être enrichie par d'autres fonctionnalités avancées et d'autres améliorations peuvent être envisagées.

notre vision dans un avenir proche est de mettre dans chaque voiture une caméra connectée à l'ensemble du système de sécurité de la voiture avec un taux de réaction entre les composants très élevé, c'est à dire lorsque la caméra detecte un expression de : tristesse ou colère les freins de la voiture commencés à diminuer la vitesse automatiquement, ça c'est un simple exemple de fonctionnement, et ils existent beaucoup.

# Webographie

1. <https://www.slideshare.net/AshwinRachha/facial-emotion-recognition-a-deep-learning-approach>
2. <https://www.slideshare.net/Doozyg/facial-expression-recognition-via-python>
3. <https://www.youtube.com/watch?v=R-zgTJCBfDu0>
4. <https://www.youtube.com/watch?v=fkgpvkqcoJct=318s>
5. <https://www.sciencedirect.com/science/article/pii/S1877050920318019>
6. <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0177239>