



République Tunisienne

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Tunis El Manar

École Nationale d'Ingénieurs de Tunis

Département TIC

Rapport du Projet de Fin d'Année 2

Application Mobile De Reconnaissance Des Émotions

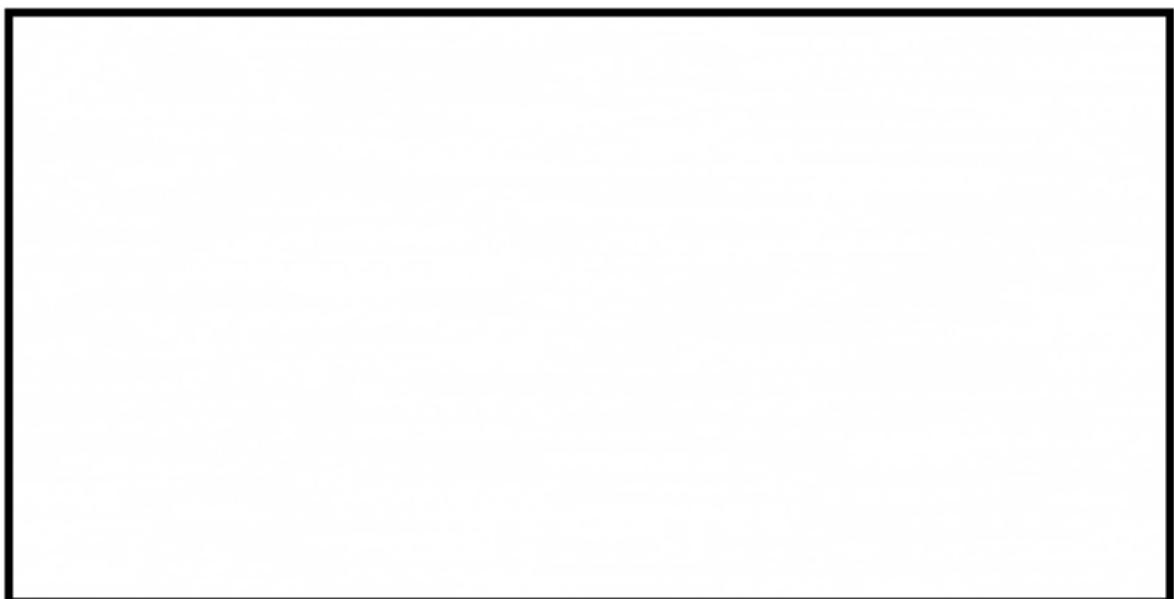
Réalisé par :

Hajtaher Chaima & Taktak Ranim

Classe : 2ATel 1

Année universitaire 2021/2022

Signatures



Remerciements

Avant d'entamer ce rapport, nous profitons de l'occasion pour remercier dans un premier temps toute l'équipe pédagogique de l'Ecole Nationale d'Ingénieurs de Tunis et les intervenants professionnels responsables de la formation Télécommunication.

Nous tenons aussi de remercier Madame Wafa Mefteh pour son encadrement tout au long de ce projet.

Finalement, nous tenons à remercier nos professeurs de nous avoir incité à travailler en mettant à notre disposition leurs expériences et leurs compétences.

Table des matières

Table des figures	vi
Introduction générale	1
Liste des tableaux	1
1 État De L'art	3
1.1 Introduction	3
1.2 Les solutions existant	3
1.2.1 Les solutions d'Affectiva	3
1.2.1.1 Affectiva Automotive AI	3
1.2.1.2 Affectiva Media Analytics	5
1.2.1.3 In Lab Biometric Solution	5
1.2.2 Amazon recognition	6
1.2.2.1 Présentation de la solution	6
1.2.2.2 Les avantages et Les inconvénients de la solution	7
1.2.3 Datakalab	7
1.2.3.1 Présentation de la solution	7
1.2.3.2 Les avantages et Les inconvénients de la solution	8
1.2.4 Kairos, le visage comme carte de fidélité	9
1.2.4.1 Présentation de la solution	9
1.2.4.2 Les avantages et Les inconvénients de la solution	10
1.2.5 Microsoft Azure API Visage, l'analyse du visage	10
1.2.6 Présentation de la solution	10
1.2.6.1 Les avantages et Les inconvénients de la solution	10
1.2.7 Vokaturi, l'émotion passe aussi par la voix	11
1.2.7.1 Présentation de la solution	11
1.2.7.2 Les avantages et Les inconvénients de la solution	11
1.2.8 Emotimeter - Emotion detector	11
1.2.8.1 Présentation de la solution	11
1.2.8.2 Les avantages et Les inconvénients de la solution	11
1.2.9 MoodMe	12
1.2.9.1 Présentation de la solution	12
1.2.9.2 Les avantages et Les inconvénient de la solution	12

1.3	Synthèse des solutions	12
1.4	Conclusion	13
2	Une Solution pour la Reconnaissance Intelligente des Émotions	14
2.1	Introduction	14
2.2	Présentation de la solution « XpressVision »	14
2.2.1	Présentation Générale	14
2.2.2	Architecture globale de la solution « XpressVision »	15
2.2.3	La détection du visage	17
2.2.4	L'extraction des caractéristiques	17
2.2.5	La classification	18
2.3	Architecture de l'application mobile	22
2.3.1	Présentation de l'Architecture 3-tiers	22
2.4	Choix de la Méthodologie de Développement	24
2.5	Conclusion	27
3	Analyse et Conception	28
3.1	Introduction	28
3.2	Analyse	28
3.2.1	Spécification des besoins	28
3.2.1.1	Les besoins fonctionnelles	28
3.2.1.2	Les besoins non fonctionnelles	29
3.2.2	Diagramme des cas d'utilisation	29
3.3	Conception	30
3.3.1	Diagramme de classes	30
3.3.2	Diagramme de séquences	32
3.4	Conclusion	33
4	Implémentation et Réalisation	34
4.1	Introduction	34
4.2	Environnement de travail	34
4.2.1	Représentations d'outils utilisées	34
4.2.1.1	La partie hardware	34
4.2.1.2	La partie software	35
4.2.2	La base de données utilisée	37
4.3	Implémentation de l'application	37
4.3.1	Module de détection visage	38
4.3.2	Module de reconnaissance d'émotion	38
4.3.3	Module de chargement des données	39
4.3.4	L'apprentissage de classificateur d'images de CNN avec Keras	41
4.3.5	Module d'apprentissage	42
4.3.6	Résultat de l'apprentissage	43
4.3.7	Présentation de l'application mobile	44
4.3.7.1	Présentation de l'environnement de travail	44
4.3.7.2	Les étapes de réalisation	46

4.3.8	Résultats de l'application	47
4.4	Conclusion	48
	Conclusion générale	50
	Références Bibliographiques	51

Liste des tableaux

1.1	Synthèse des solutions de reconnaissance des émotions	13
4.1	La partie hardware de l'environnement de travail	34

Table des figures

1.1	Affectiva Automotive AI	4
1.2	La solution In Lab Biometric	6
1.3	Amazon Rekognition	7
1.4	La solution de Datakalab	8
1.5	La solution de Kairos	9
2.1	Schéma global de la solution	16
2.2	Processus de détection du visage	17
2.3	Processus de l'étape d'extraction de caractéristiques du visage	18
2.4	Les fonctionnalités de MobileNetV2 [12]	19
2.5	L'architecture de MobileNetV2 [12]	20
2.6	La structure globale de MobileNetV2 [12]	21
2.7	Architecture 3-tiers simplifiée	23
2.8	Architecture 3-tiers Mobile	24
2.9	Modélisation du processus Cascade [13]	25
2.10	Modélisation du processus Cascade [13]	26
3.1	Diagramme des cas d'utilisation "Emotion Recognition"	30
3.2	Diagramme de classes	31
3.3	Diagramme de séquences	32
4.1	Détection de visage et dessin de rectangle	38
4.2	Module de reconnaissance d'émotion	39
4.3	Module de chargement des données	41
4.4	L'apprentissage de classificateur d'images de CNN	42
4.5	Résultat de l'apprentissage	44
4.6	Test d'un visage neutre	47
4.7	Test d'un visage en colère	48
4.8	Test d'un visage qui fait peur	48

Introduction générale

Que savez-vous de votre visage ? Avez-vous déjà utilisé ces expressions ? "Rire du fond du cœur", "être ennui à cause de quelque chose", "être en colère", "Rester bouche bée" ? Ces expressions sont souvent utilisées par tous les gens dans le monde pour faire passer une sensation, une émotion ou une idée par le biais des traits du visage.

A partir de cela, nous pouvons conclure le rôle puissant du visage dans la communication et l'interaction avec un interlocuteur. Donc, il n'est pas surprenant que l'expression faciale ait été un des domaines de recherche les plus important sur le comportement humain. Par conséquent, Les émotions sont indispensables et irremplaçables dans notre vie. Elles jouent un rôle vital, car elles nous permettent d'améliorer notre communication avec les individus, d'assurer une bonne réception du message véhiculé et s'adapter devant une situation précise. Les expressions faciales fournissent des informations aux observateurs sur l'expérience émotionnelle d'un individu. Par exemple, la joie émotionnelle est signalée aux observateurs en tirant les joues et en tirant les coins des lèvres vers le haut. Dans nos jours, la reconnaissance des émotions devienne un réel défi. D'une part, les représentations des émotions proposées par les théoriciens sont variées et d'autre part, les émotions sont multi componentielles « verbal, vocal et non verbal » et dans notre cas nous sommes focalisées dans la reconnaissance d'émotion non verbales : FACIALES en utilisant des images.

L'objectif ultime de ce projet de fin d'année est de concevoir et développer une application mobile pour la détection des émotions à partir des expressions faciales, en utilisant un montage simple et peu coûteux. Parmi les objectifs supplémentaires, il y a entre autres le choix des algorithmes et méthodes nécessaires pour effectuer les tâches de détection et de reconnaissance. De plus, différentes expérimentations doivent présenter les niveaux de

précision, de robustesse et d'efficacité des techniques sélectionnées. Ce projet vise l'exploration des différentes facettes de la vision numérique à partir de l'acquisition des images et de leur traitement, jusqu'à l'interprétation et la reconnaissance des expressions faciales. Il s'agit donc en fait de la conception d'un système complet de détection automatique des expressions faciales.

Pour réaliser ces objectifs ,ce travail, est donc organisé comme suit :Tout d'abord, le premier chapitre portera sur l'étude de l'art .c'est-à dire , la revue de littérature qui permet de décrire les principales approches de détection des émotions retrouvées dans la littérature, alors que le deuxième chapitre portera sur l'analyse et toutes les méthodes appliquées pour la réalisation des tâches de détection des émotions . Ensuite, le troisième chapitre présentera le fruit de nos expérimentations dans lesquelles le modèle de détection des émotions implémenté est testé avec des séquences d'images.

Finalement, on clôture notre projet par une conclusion.

Chapitre 1

État De L'art

1.1 Introduction

Au cours de la dernière décennie, la complexité des émotions a suscité l'intérêt de plusieurs chercheurs dans tout le monde car elles se manifestent par des réactions physiologiques aussi par des expressions gestuelles et faciales, et dans ce cadre nous sommes intéressées dans la reconnaissance faciale des émotions.

Ce chapitre est dédié à La présentation du recherche dans le domaine de reconnaissance des émotions et les solutions effectuées par les chercheurs ensuite une synthèse des travaux réalisés, et enfin une présentation de notre solution **XpressVision**.

1.2 Les solutions existant

1.2.1 Les solutions d'Affectiva

1.2.1.1 Affectiva Automotive AI

Affectiva Automotive AI est la première solution de détection multimodale en cabine pour comprendre ce qui se passe avec les passagers dans un véhicule a partir des expressions faciales et vocales, a été lancé le 21 mars 2018 [1].

Les constructeurs et fournisseurs automobiles peuvent utiliser cette solution pour suivre les humeurs et les comportements des occupants afin de créer une expérience de transport plus personnalisée tout en améliorant la sécurité routière grâce aux signaux faciaux

qui suggèrent la somnolence, le système prédit les niveaux de distraction et de fatigue du conducteur .Il se présente sous la forme d'un périphérique de type caméra ou d'une technologie conçue pour les systèmes embarqués.

La figure 1.1 représente la solution Affectiva Automotive AI.



FIGURE 1.1 – Affectiva Automotive AI

Parmi les avantages d'Affectiva Automotive AI, on peut citer la robustesse de cette solution. En effet, elle supporte l'habillement des masques chirurgicaux et les lunettes. De plus, elle suit la pose de la tête et les expressions faciales pour identifier les niveaux de somnolence et distraction. Ensuite, elle détecte les expressions de joie, surprise, colère et rire à partir des expressions faciales et vocales. Finalement, elle supporte les différentes positions de caméra.

En revanche, cette solution n'est pas adaptative aux conditions environnementales. En effet, le fait de modifier le style de conduite autonome, rend les passagers anxieux ou mal à l'aise. De plus, elle ne comprend pas quelque type d'émotions.

1.2.1.2 Affectiva Media Analytics

La solution Media Analytics d’Affectiva analyse les mouvements du visage pour mieux comprendre les émotions et les états cognitifs complexes et nuancés. La solution est élaborée à l'aide de logiciel Emotion AI, qui utilise la vision par ordinateur et l'apprentissage profond [2].

Elle vend également un outil qui mesure les réponses émotionnelles d'un groupe de personnes à une publicité, une vidéo, film et émission de télévision, ce qui permet aux marques d'établir des liens plus significatifs avec leurs clients. C'est une solution qui comprend le test des publicités « Add Testing », Test de contenu de divertissement « Entertainment Content Testing », et la Recherche qualitative « Qualitative Research ».

Cette solution a pour avantage la mesure de 7 émotions universelles et 20 expressions faciales et le test des voix-off et la compréhension de rôle émotionnel des marques. Ensuite, elle permet aux responsables du marketing de comprendre les moments clés pour les bandes annonces et le matériel de marketing. De plus, elle aide les studios de cinéma et leurs partenaires par exemple à comprendre et à optimiser le contenu pour qu'il ait un impact réel sur les publics visés. Finalement, elle améliore la commercialisation par le biais de bandes annonces. Par contre, cette solution a besoin de travailler sur la carence des types d'émotions « La précision ».

1.2.1.3 In Lab Biometric Solution

La solution In Lab Biometric d’Affectiva montre non seulement la réponse émotionnelle d'un individu, mais aussi sa réponse physiologique et cognitive [3].

Cela permettent de comprendre en profondeur la manière dont une personne vit le monde, car son comportement peut être mesuré et évalué de manière objective.

Ce processus est simple et facile à mettre en œuvre. En combinant les données sur les expressions faciales avec les mesures de l'éveil physiologique, l'activité cérébrale, le suivi des yeux et d'autres actes, cela permet d'obtenir une vue complète du comportement, de l'action et de la pensée humains.

La figure 1.2 présente la solution In Lab Biometric d’Affectiva.

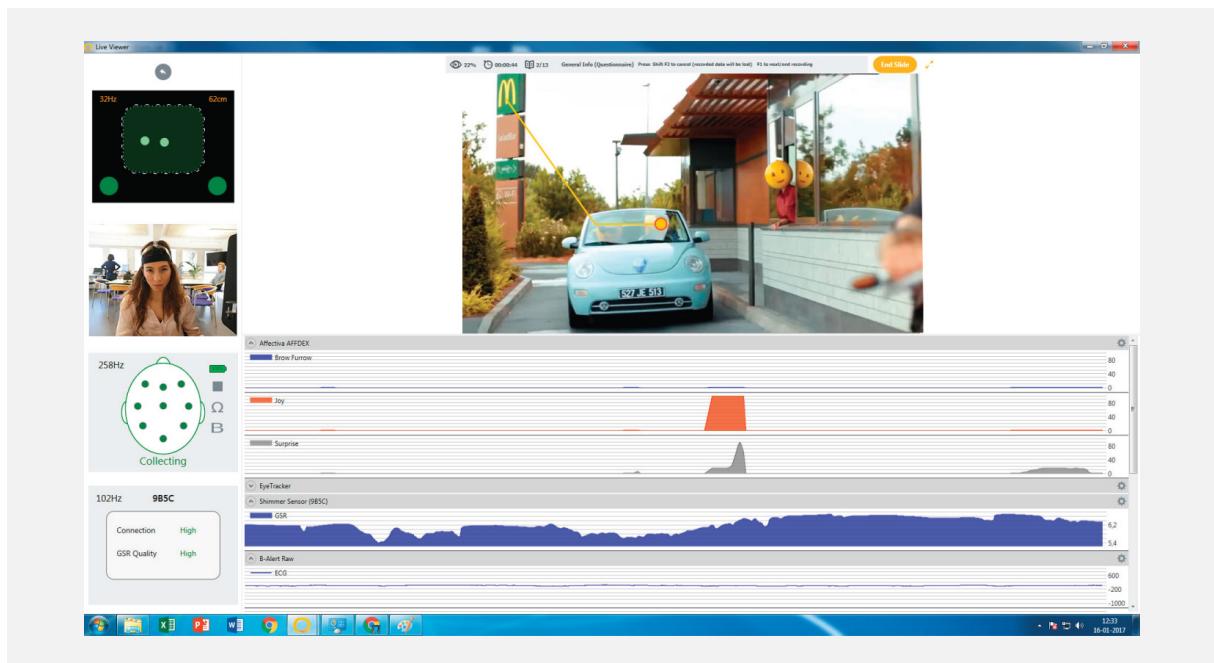


FIGURE 1.2 – La solution In Lab Biometric

Cette solution est **avantageux pour plusieurs secteurs spécialement pour la santé : c'est une solution qui comprend comportement humain** dans la santé et la maladie . De plus, elle est rapide et facile à mettre en œuvre.

1.2.2 Amazon recognition

1.2.2.1 Présentation de la solution

Amazon Rekognition est une interface de programmation d'application API fournie par Amazon Web Services. Qui s'applique à tout fichier image ou vidéo stockés sur Amazon S3, qui utilise l'apprentissage profond et permet de détecter plusieurs types d'émotions [4].

Rekognition est également développé pour la détection des émotions, l'identification des visages et le contrôle des contenus inappropriés. L'API détermine si une personne a les yeux ouverts, des lunettes ou une barbe, ainsi que sa tranche d'âge. Une méthode d'horodatage retrace l'évolution des sentiments du sujet au fil du temps dans le cas des vidéos.

La figure 1.3 représente la solution Amazon Rekognition.

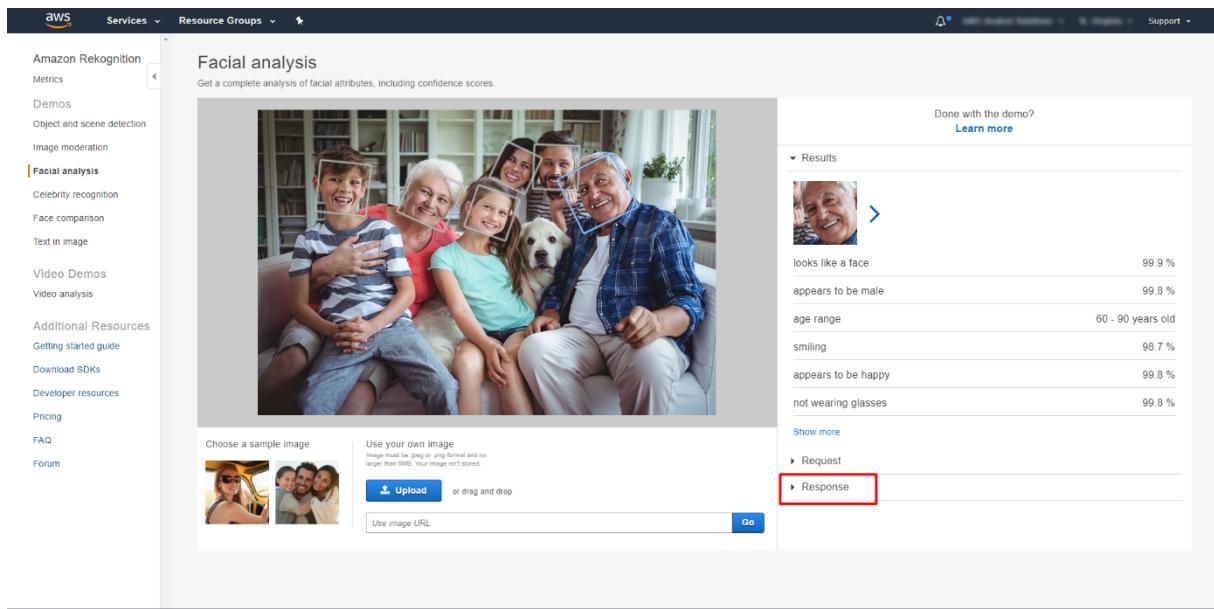


FIGURE 1.3 – Amazon Rekognition

1.2.2.2 Les avantages et Les inconvénients de la solution

Cette solution a pour avantage la détection de 8 émotions universelles telles que : la joie, la peur, la surprise, la tristesse, confuse, le dégoût et la colère. De plus, il s'agit d'une API qui détecte des types d'émotions un peu particulières : Calme, Peur. Aussi, elle détecte non seulement tous les visages d'une image. Mais, aussi, les émotions en temps réel ou en lots sans gestion d'infrastructure ou de modélisation. Par contre, cette solution a pour inconvénient : la carence des types d'émotions « La précision ».

1.2.3 Datakalab

1.2.3.1 Présentation de la solution

Datakalab est une start-up technologique française spécialisée dans la « face coding » qui propose aux sites E-commerce « les marques » de détecter les émotions des consommateurs et de croiser ces données "émotionnelles" aux données transactionnelles afin d'apporter de la chaleur à leur expérience client. Alors un E-commerçant peut piloter sa stratégie produit en intégrant les données émotionnelles à sa prise de décision [5].

Tealeaf c'est l'outil d'analyse de l'expérience client d'IBM, a été utilisé par Datakalab pour fournir une solution de reconnaissance des émotions. Pendant la visite d'un site web par un client, il évalue ses niveaux d'attention et d'engagement émotionnel. Cela suppose toutefois que l'utilisateur consente à être filmé en temps réel.

La figure 1.4 présente la solution de DataKalab.

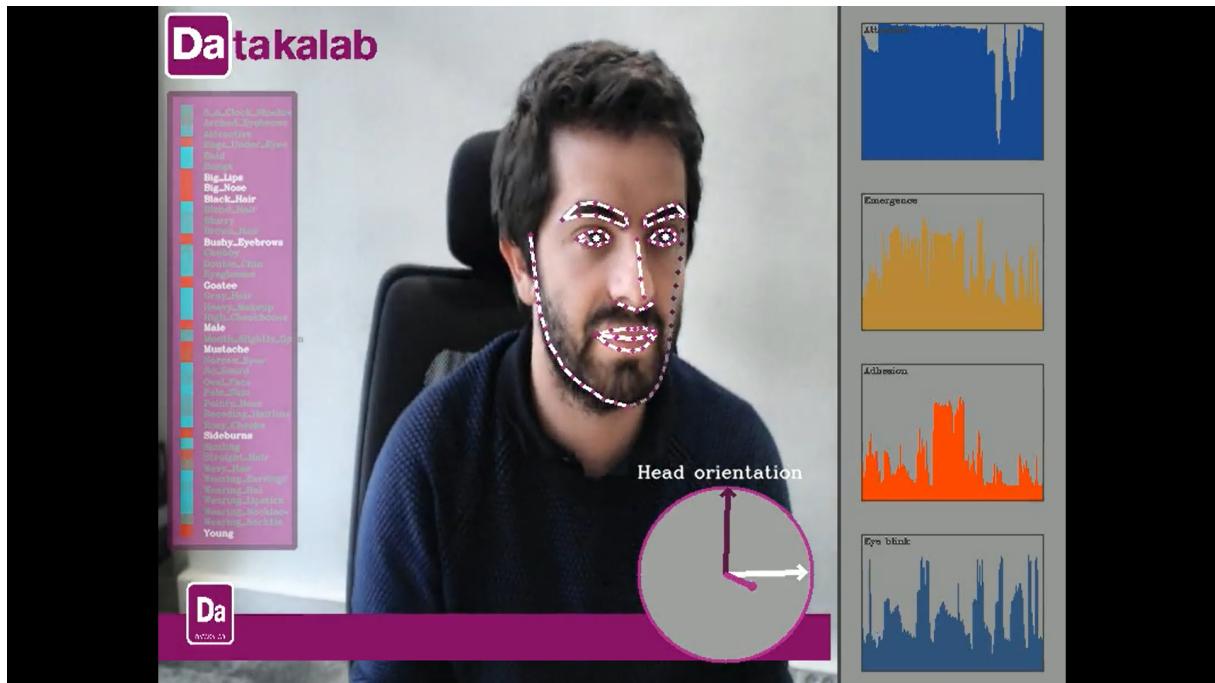


FIGURE 1.4 – La solution de Datakalab

1.2.3.2 Les avantages et Les inconvénients de la solution

Les avantages :

- Le logiciel de Datakalab est proposé en mode SaaS : software as a service => c'est-à-dire il est simple à utiliser.

Les inconvénients :

- Selon l'un des cofondateurs interrogé par les Echos, la démarche a un faible taux d'acceptation en France, environ 30%.

1.2.4 Kairos, le visage comme carte de fidélité

1.2.4.1 Présentation de la solution

Kairos est une entreprise française, qui a été fondée en 2012, travaille sur une API de reconnaissance faciale pour le marketing des marques. Cette entreprise basée à Paris propose notamment un moyen d'améliorer l'expérience client en magasin. Elle reconnaît les clients dès qu'ils entrent dans un magasin et prédit leur sexe, leur âge et leur état émotionnel afin de personnaliser leur expérience d'achat [6].

Kairos acquiert Emotion Reader pour mesurer la réaction émotionnelle des spectateurs à une vidéo et d'analyser cette réaction via un tableau de bord analytique et de prendre des décisions différentes concernant les dépenses médiatiques en fonction de la réaction des spectateurs.

La figure 1.5 présente la solution de Kairos.

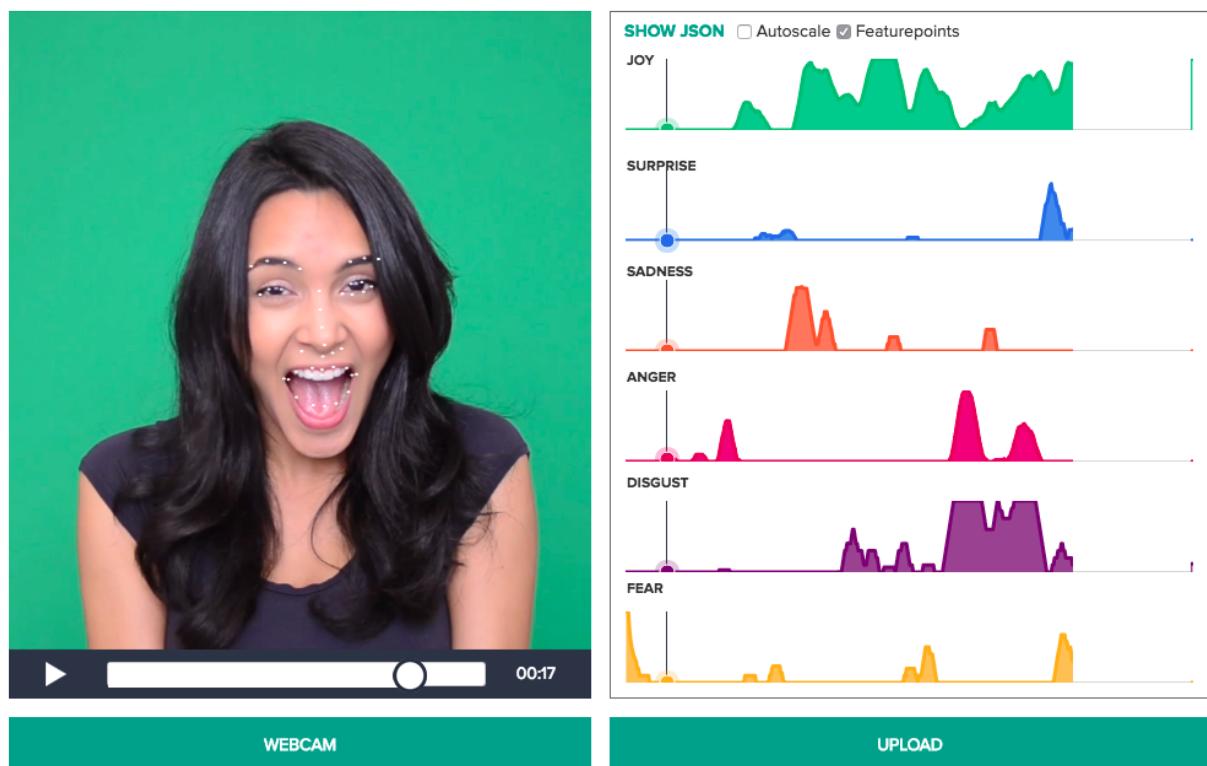


FIGURE 1.5 – La solution de Kairos

1.2.4.2 Les avantages et Les inconvénients de la solution

Les avantages

- Déetecte 6 types d'émotions :la joie, la peur, la surprise, la tristesse, dégouté, et en colère.
- Supporte 3 types de fichiers : Image, Vidéo, Audio.
- C'est un logiciel qui dispose d'un essai gratuit.
- C'est un logiciel SaaS : d'où simple.
- C'est une application Cloud sécurisé.

Les inconvénients

- Le carence des types d'émotions « La précision ».

1.2.5 Microsoft Azure API Visage, l'analyse du visage

1.2.6 Présentation de la solution

Microsoft Azure offre une variété de services cognitifs, parmi ceux-ci on trouve l'API visage qui est spécialisée dans la reconnaissance des émotions. Cet API utilise les algorithmes de reconnaissance faciale de pointe dans le Cloud pour objectif : détecter et identifier les visages humains à partir des images [7].

Par exemple, ce visage exprime le bonheur à 20% et l'étonnement à 80%. Azure permet aux utilisateurs de tester leur solution en ligne en téléchargeant une photo ou en saisissant une URL.

1.2.6.1 Les avantages et Les inconvénients de la solution

Les avantages

- Cette solution obtient un score de confiance élevé.
- L'API Vision de Google Cloud propose une méthode très similaire de reconnaissance des états émotionnels, ainsi qu'une démonstration en ligne.

Les inconvénients

- Le carence des types d'émotions « La précision ».

1.2.7 Vokaturi, l'émotion passe aussi par la voix

1.2.7.1 Présentation de la solution

Vokaturi est une entreprise danoise fondée en 2016, propose un logiciel qui mesure les aspects émotionnels de la parole indépendamment de son contenu sémantique [9].

Son algorithme est basé sur le Deep Learning, développé par un scientifique de l'Université d'Amsterdam, détermine le pourcentage de cinq états émotionnels : le bonheur, la tristesse, la peur, la colère, et un état d'esprit neutre [8].

1.2.7.2 Les avantages et Les inconvénients de la solution

Les avantages :

- Vokaturi propose une version open source sous licence GPL pour des raisons de test, en plus de ses versions premium qui offrent des fonctionnalités et des taux de reconnaissance améliorés.
- Donne le pourcentage d'état émotionnel.

Les inconvénients :

- Mesure que 5 états émotionnels : la joie, la peur, la colère, neutre et la tristesse d'où l'absence de précision.

1.2.8 Emotimeter - Emotion detector

1.2.8.1 Présentation de la solution

Emotimeter est une application mobile valable sur Google Play store depuis l'année 2020, qui permet aux utilisateurs de détecter les états émotionnels en temps réel à partir des expressions faciales obtenues par la caméra, en utilisant des technologies d'apprentissage automatique « Machine Learning » [9].

1.2.8.2 Les avantages et Les inconvénients de la solution

Les avantages :

- Supporte 2 types de fichiers : image et vidéo.

- Analyse les photos de la galerie d'un appareil mobile pour détecter les émotions de toutes les personnes qui s'y trouvent.
- Analyse les vidéos de votre galerie pour détecter les émotions de toutes les personnes qui s'y trouvent, image par image.
- Supporte les 7 émotions universelles.

Les inconvénients :

- la carence des émotions.

1.2.9 MoodMe

1.2.9.1 Présentation de la solution

MoodMe est une entreprise basée en Austin Texas fondée en 2015, spécialiste dans plusieurs secteurs comme l'IA, Deep Learning, Big Data et plusieurs autres domaines [10].

MoodMe met en disposition des utilisateurs une application mobile réalisée en 2019 qui leurs permettre de détecter les états émotionnels à partir des expressions faciales [11].

1.2.9.2 Les avantages et Les inconvénient de la solution

Les avantages :

- Analyse les photos de la galerie d'un appareil mobile pour détecter les émotions de toutes les personnes qui s'y trouvent.
- Supporte les 7 émotions : la joie, la tristesse, la neutralité, la surprise, la peur, la colère, le dégoût.

Les inconvénients :

- la carence des émotions.

1.3 Synthèse des solutions

Ce tableau-là donne une simple synthèse des solutions trouvées dans le domaine de la reconnaissance des émotions et à partir de ceci on va choisir notre méthodologie.

Les solutions	Le type de reconnaissance	Le domaine d'application	Algorithme	Le type de solution
Affectiva Automotive AI	Faciale et vocale	Automobile	Deep Learning	
Affectiva Media Analytics	Faciale et vocale	Media	Deep Learning	
In Lab Biometric Solution	Faciale et vocale	Biométrie	Deep Learning	
Amazon rekognition	Faciale	Embarqué		API
Datakalab	Faciale	Marketing		
Kairos	Faciale	Marketing		
Microsoft Azure API	Faciale	Pas de domaine précis		API
Vokaturi	Vocale	Pas de domaine précis	Deep Learning	
Emotimeter	Faciale	Destinés aux utilisateurs mobiles	Machine Learning	Application mobile
MoodMe	Faciale	Destinés aux utilisateurs mobiles		Application mobile

TABLE 1.1 – Synthèse des solutions de reconnaissance des émotions

En se basant sur notre recherche : on peut classifier les solutions de système de reconnaissance des «émotions à l'aide de plusieurs indicateurs comme le type de reconnaissance (faciale et/ou vocale), l'algorithme (Deep Learning ou machine Learning), le type de solution (API ou application mobile) et finalement le domaine d'application.

1.4 Conclusion

Dans ce chapitre, nous allons arriver à présenter les solutions effectuées par les chercheurs dans le domaine de reconnaissance des émotions.

En se basant sur ce chapitre : l'état de l'art nous allons décrire notre méthodologie pour réaliser notre application de détection intelligente des émotions.

Dans le deuxième chapitre nous allons présenter notre choix brièvement, en citant les différents outils pour réaliser notre solution.

Chapitre 2

Une Solution pour la Reconnaissance Intelligente des Émotions

2.1 Introduction

Dans ce chapitre, nous décrivons en détails notre approche proposée pour le développement d'un système de reconnaissance intelligente des émotions de base. Ainsi, nous allons présenter le choix de la méthodologie de développement.

2.2 Présentation de la solution « XpressVision »

2.2.1 Présentation Générale

Nous proposons une modélisation d'un système de reconnaissance intelligente des émotions de base à partir d'un visage **en temps réel** à base des réseaux de neurones convolutifs CNN. Ce système consiste à détecter un visage d'un individu à partir d'une image ou via une caméra pour connaître l'émotion avec un taux de précision associé aux sept expressions universelles.

2.2.2 Architecture globale de la solution « XpressVision »

Notre solution est structurée en trois modules principaux qui fonctionnent indépendamment :

- La détection de visage.
- L'extraction des caractéristiques.
- La classification.

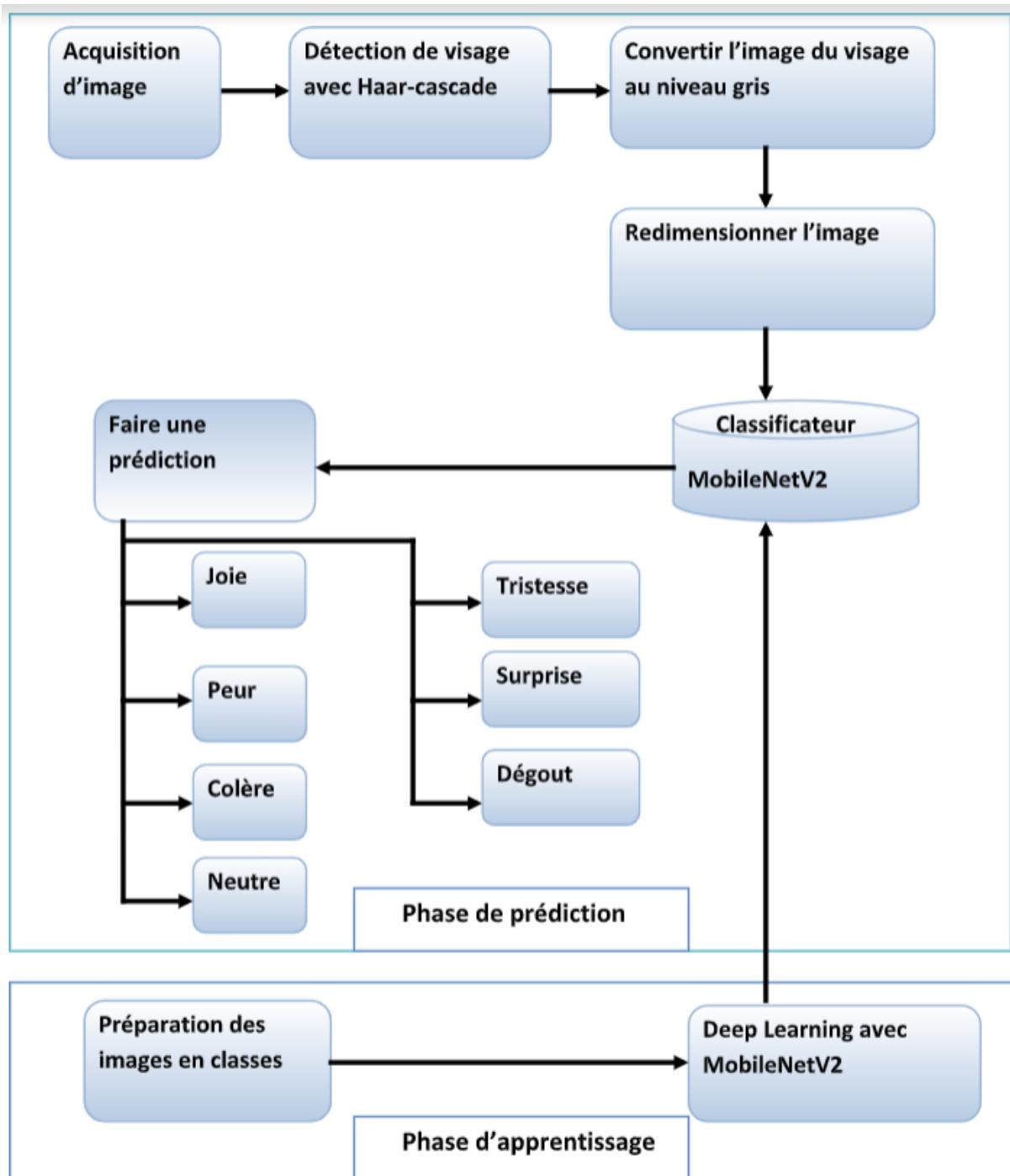


FIGURE 2.1 – Schéma global de la solution

2.2.3 La détection du visage

La détection des visages est l'identification de la présence de visage dans une image et la détermination des emplacements et des échelles des visages.

Dans notre solution, nous utilisons l'algorithme VIOLA-JONES pour détecter les diverses parties du visage comme la bouche, les yeux, le nez, les narines, les sourcils, les lèvres et les oreilles.



FIGURE 2.2 – Processus de détection du visage

2.2.4 L'extraction des caractéristiques

La deuxième étape de notre solution est le processus d'extraction des caractéristiques dont on va convertir les données des pixels à des représentations et configurations plus réduites et optimales afin que la représentation extraite soit utilisée dans le processus de classification ultérieurement .Cette étape permet de réduire les dimensions de l'image en entrée en gardant les données les plus utiles pour la classification.

L'étape d'extraction des caractéristiques représente le cœur de notre système de reconnaissance intelligente des émotions. En effet, à partir de l'image, on extrait les informations qui seront sauvegardées en mémoire pour être utilisées plus tard dans la phase de classification.



FIGURE 2.3 – Processus de l'étape d'extraction de caractéristiques du visage

Ce processus est basé sur la technologie de réseau de neurones convolutionel (CNN). Nous avons utilisé l'architecture de CNN, MobileNetV2, dans ce qui suit nous expliquons cette architecture.

2.2.5 La classification

Pour reconnaître l'expression faciale du visage détecté, nous avons exploité des réseaux de neurones convolutionel. En effet, les architectures de réseaux neuronaux sont des combinaisons de couches qui **se transmettent** les sorties les unes après les autres.

D'après les recherches nous avons trouvé une architecture CNN qui **appelée Mobile-NetV2** a plusieurs fonctionnalités comme illustre la figure 2.4 :

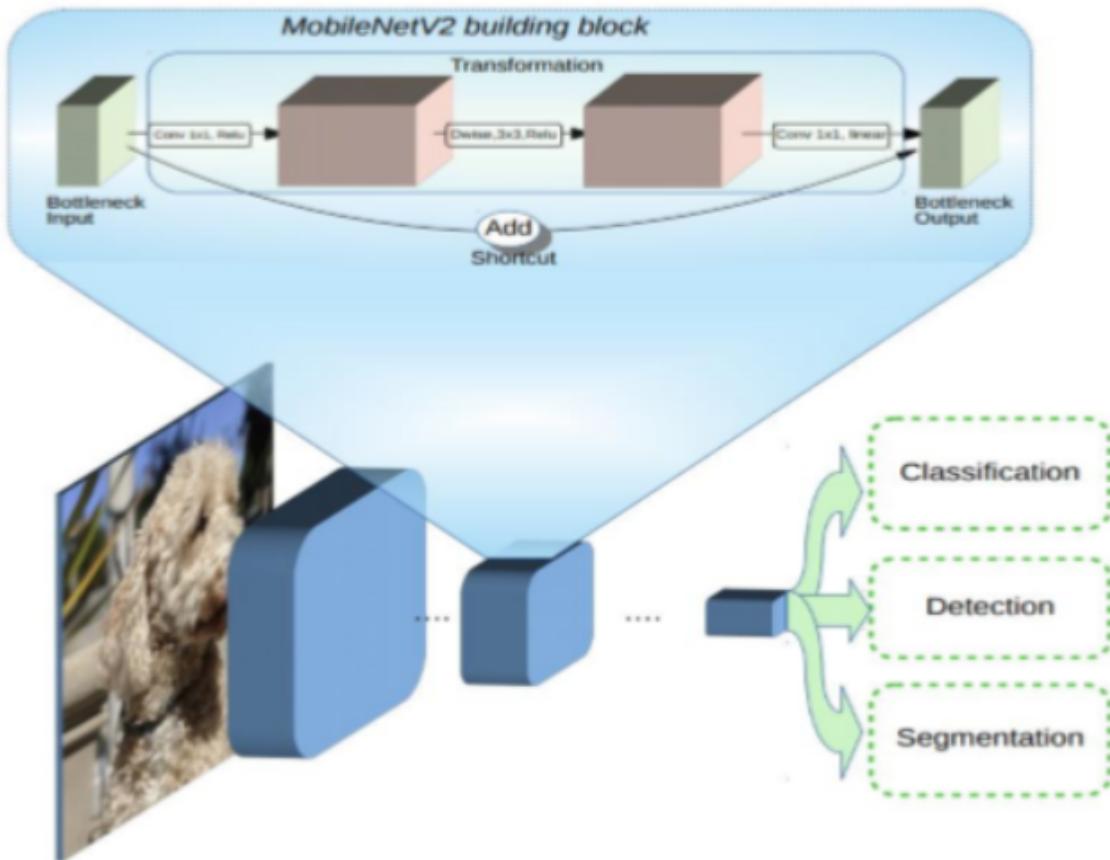


FIGURE 2.4 – Les fonctionnalités de MobileNetV2 [12]

MobileNetV2 est une structure moderne publiée en 2018, dans le cadre de la bibliothèque de classification d’images Tensorflow, par des chercheurs de Google. Ce modèle améliore les performances de modèle MobileNet précédent (MobileNetV1). Le modèle introduit des nouvelles structures résiduelles inversées avec des connexions de raccourci entre les couches de goulot d’étranglement minces.

MobileNetV2 est une architecture de réseau neuronal convolutionnel qui cherche à bien fonctionner sur les appareils mobiles. Il est basé sur une structure résiduelle inversée où l’entrée et la sortie du bloc résiduel sont de minces couches de goulot d’étranglement, opposées aux modèles résiduels traditionnels qui utilisent des représentations étendues dans l’entrée.

Un MobileNetV2 utilise des convolutions légères dans le sens de la profondeur pour filtrer les entités dans la couche d’expansion intermédiaire. Dans son ensemble, l’architecture de MobileNetV2 contient la couche initiale entièrement à convolution avec 32 filtres,

suivie de 19 couches de goulot d'étranglement résiduelles.

Dans MobileNetV2, il existe deux types de blocs. Un est un bloc résiduel avec une foulée de 1. Et l'autre est un bloc avec foulée de 2 pour la réduction des effectifs, la figure 2.5 explique cela.

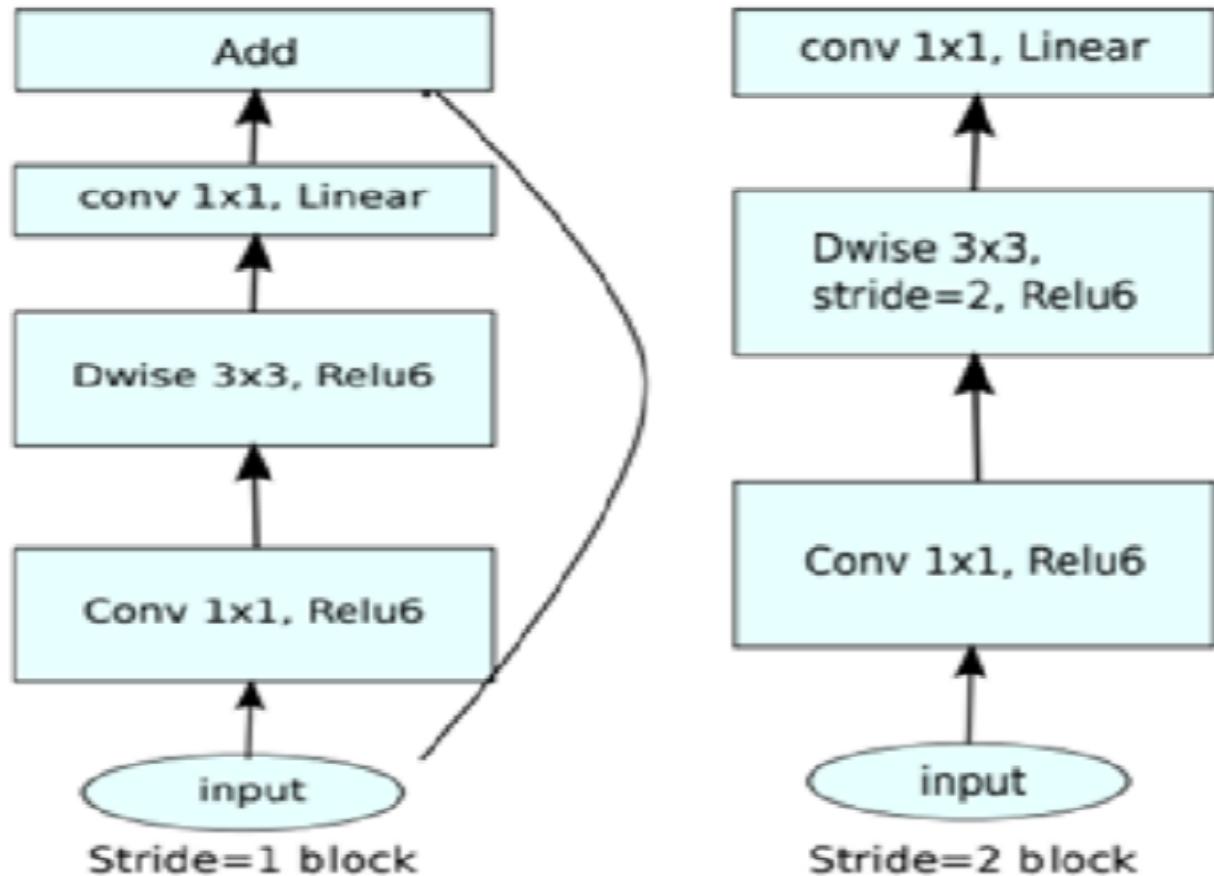


FIGURE 2.5 – L'architecture de MobileNetV2 [12]

Il y a trois couches pour les deux types de blocs :

- La première couche est la convolution 1×1 avec ReLU.
- La deuxième couche est la convolution en profondeur.
- La troisième couche est une autre convolution 1×1 mais sans aucune non-linéarité.

La structure globale de mobilenetv2 est présentée dans la figure suivante :

Input	Operator	<i>t</i>	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	-

FIGURE 2.6 – La structure globale de MobileNetV2 [12]

Avec :

- t : facteur d’expansion.
- c : nombre de canaux de sortie.
- n : nombre répétitif.
- s : foulée.

Chaque ligne décrit une séquence de 1 ou plusieurs couches identiques, répétées n fois.

Toutes les couches de la même séquence ont le même nombre c de canaux en sortie. La première couche de chaque séquence a une foulée s et toutes les autres utilisent la foulée 1. Toutes les convolutions spatiales utilisent des noyaux 3 X 3. Le facteur d’expansion t est toujours appliqué à la taille d’entrée.

Nous avons choisi cette architecture pour la construction de notre système de reconnaissance d’émotion pour la classification. Parmi les avantages de cette architecture :

- Réduire l’encombrement mémoire.

- Augmenter la vitesse de calcul.
- Taille de modèle plus petite : Moins de paramètres.
- Plus petite complexité.

2.3 Architecture de l'application mobile

2.3.1 Présentation de l'Architecture 3-tiers

L'architecture d'une application mobile utilisant une base de données distante pourrait se schématiser comme suivant :

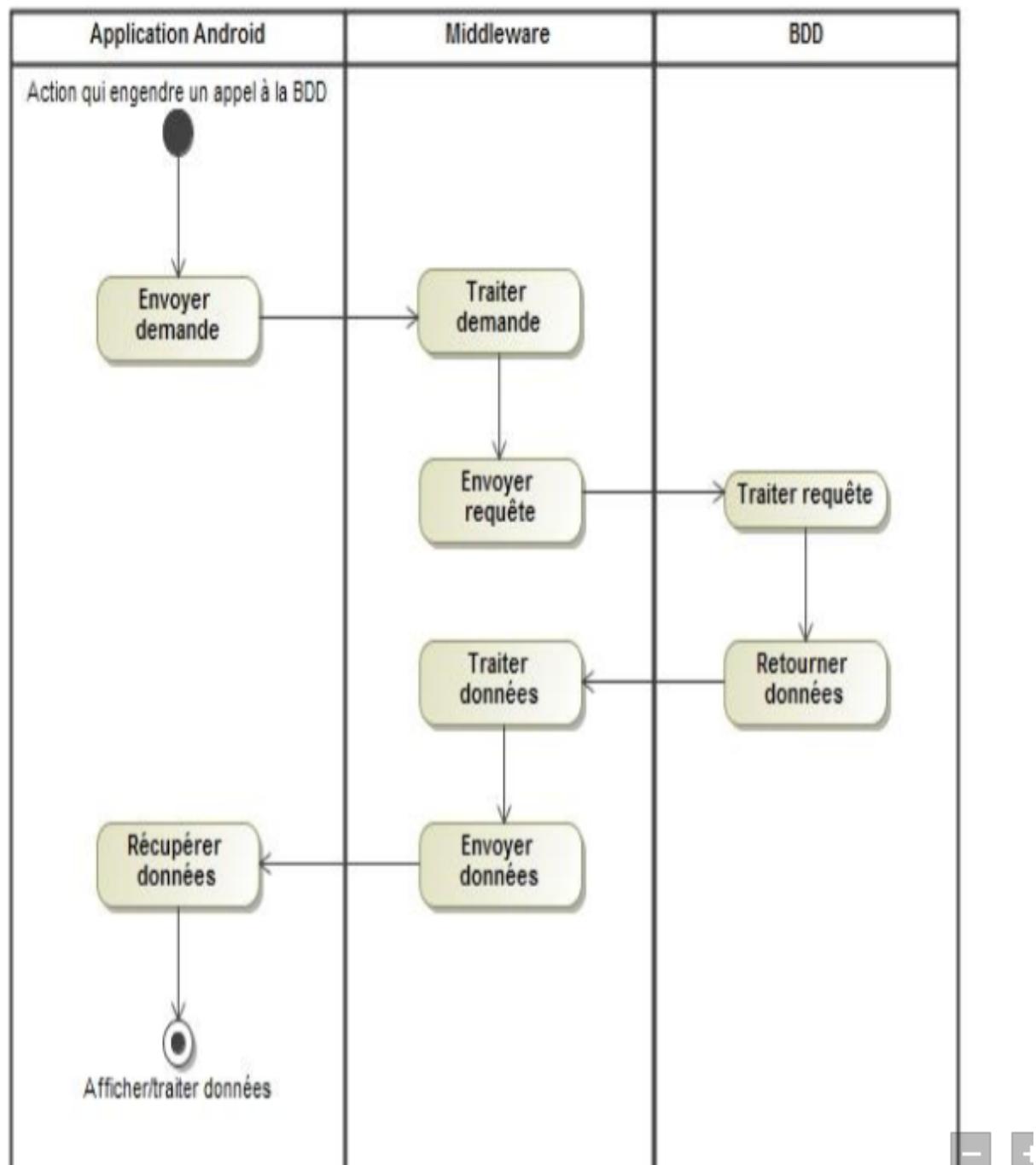


FIGURE 2.7 – Architecture 3-tiers simplifiée

L’application Android représente la couche client qui envoie des requêtes à une couche intermédiaire (Middleware). Par le biais d’un ou plusieurs web service REST, la communication entre la couche Client et la couche données contenant la base de données peut être établie.

Une architecture 3-tiers est une architecture client-serveur où les couches accès aux

données, interface utilisateur et traitement des données sont séparées et maintenue sur des plateformes distantes. C'est une architecture 3 niveaux qui permettent chacune des couches d'être changée ou remplacée de manière indépendante, sans toucher aux autres couches.

Les trois tiers d'une architecture 3-tiers sont :

- « **Présentation** » : ou la couche client, demandeuse de ressources. Elle est l'interface utilisateur.
- « **Application** » : ou middleware. C'est la couche « logique », qui contient les fonctionnalités et chargée de fournir la ressource en faisant appel à un autre serveur.
- « **Données** » : la couche contenant les bases de données et toutes les datas.

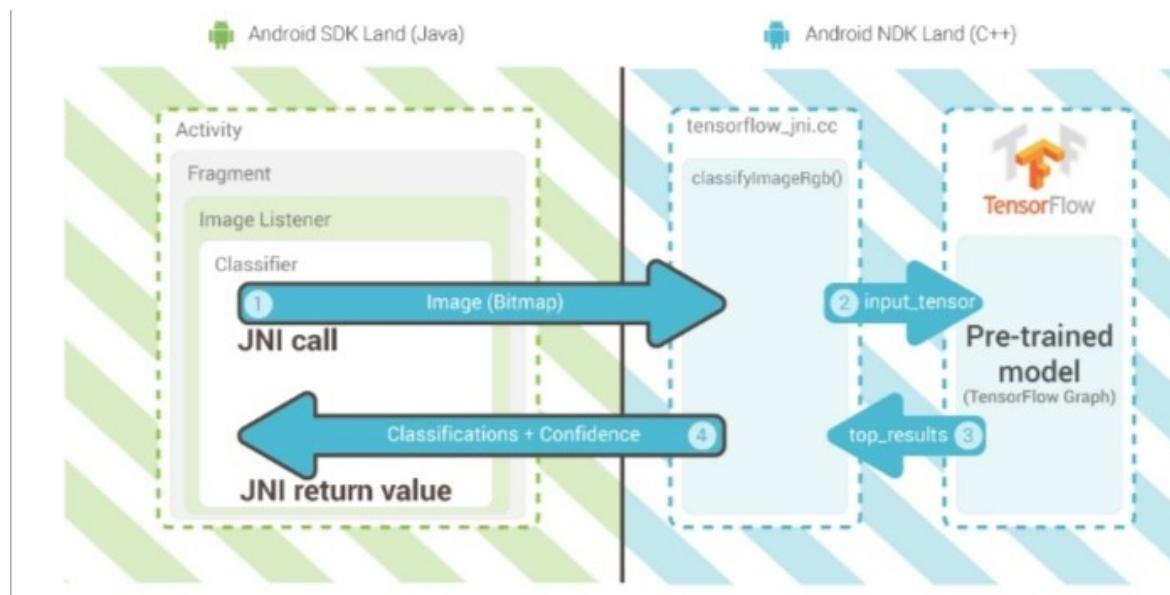


FIGURE 2.8 – Architecture 3-tiers Mobile

2.4 Choix de la Méthodologie de Développement

Présentation du processus Cascade

- Chaque étape doit être terminée avant que ne commence la suivante.
- A chaque étape, production d'un document qui est basé de l'étape suivante.

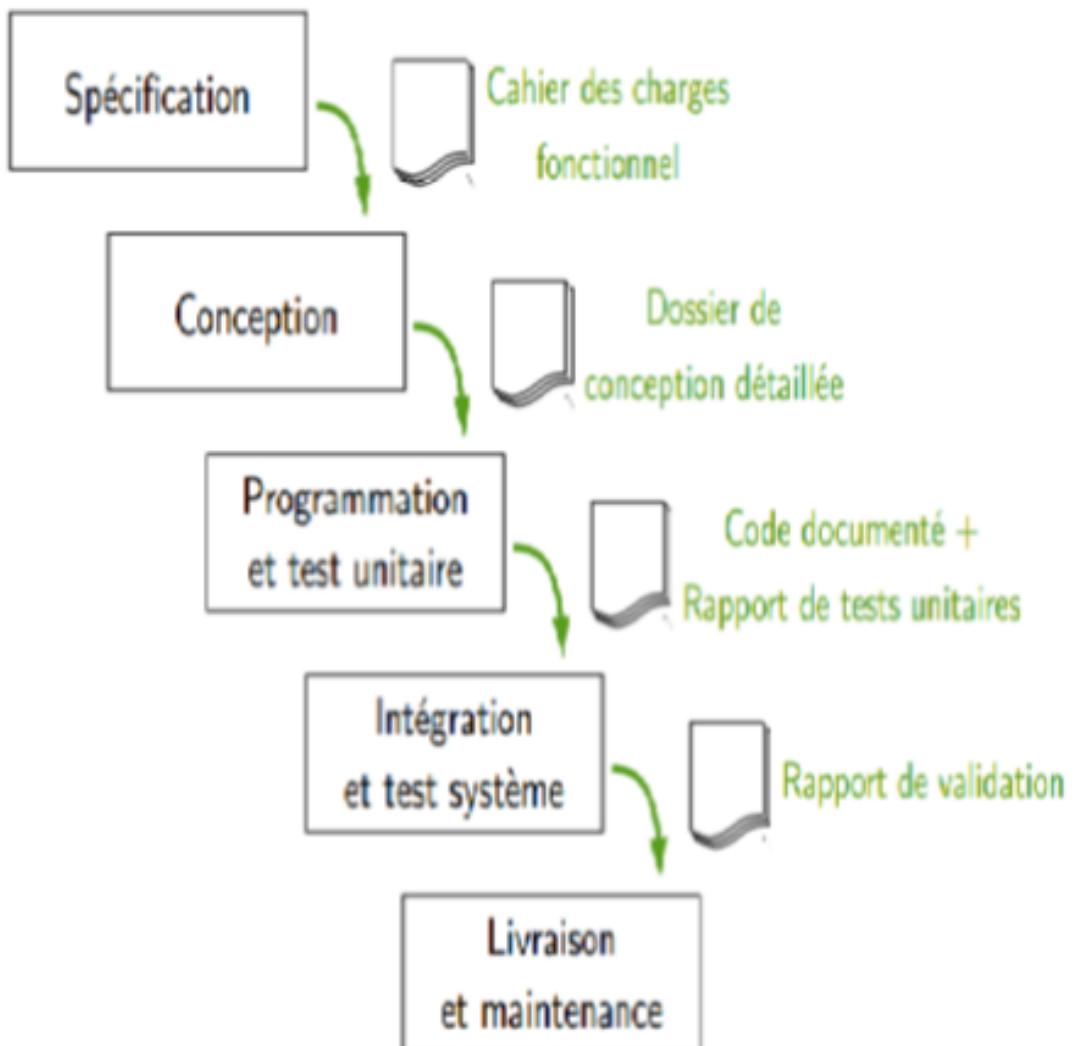
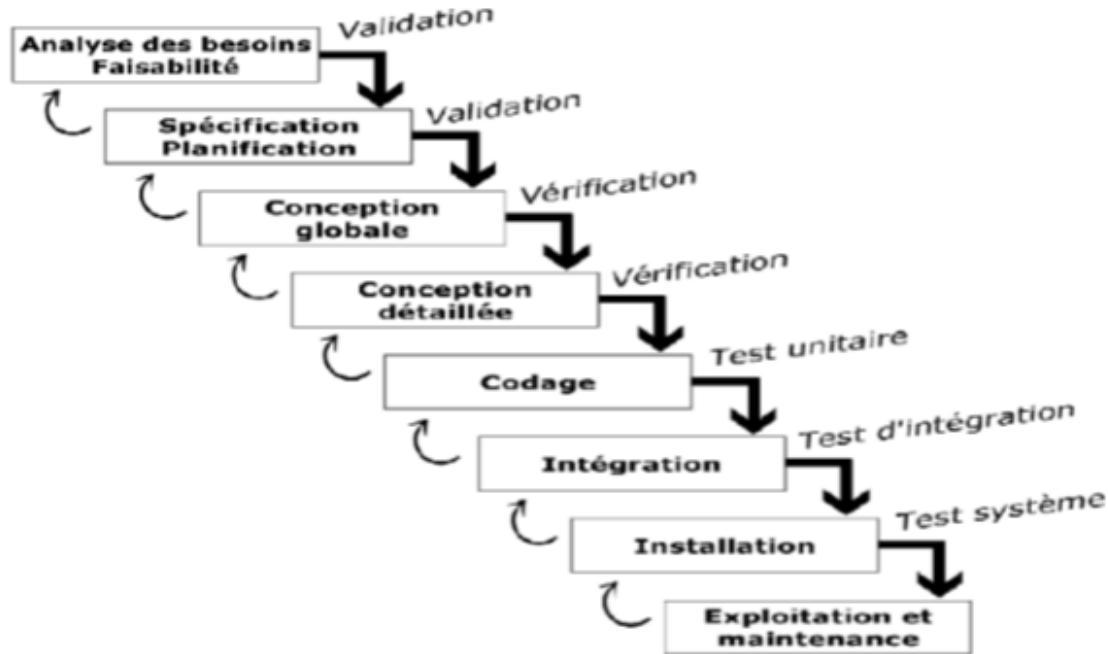


FIGURE 2.9 – Modélisation du processus Cascade [13]

Les versions actuelles de ce modèle font apparaître la validation-vérification à chaque étape.

Processus en cascade



73

Cours GL 2AInfo
Chiraz jelassi

FIGURE 2.10 – Modélisation du processus Cascade [13]

Avantages

- Simple à mettre en œuvre.
- La documentation est produite à chaque étape.

Inconvénients

- Difficulté d'avoir toutes les spécifications du client.
- Preuve tardive du bon fonctionnement.
- Cout de modification d'une erreur importante.
- Pas transparent au client lors du développement.

Notre choix a été porté sur le processus Cascade de fait qu'elle nous permet de faire ,d'une part , une conception qui est plus pragmatique et détaillée.et d'autre part ,ce dernier est utilisée pour les projets de petites taille comme le cas de notre projet de PFA2.

2.5 Conclusion

Dans ce chapitre, nous avons présenté la présentation de notre système de reconnaissance intelligente des émotions de base. En utilisant le modèle CNN, afin de révéler les performances d'un système FER 2013 par apprentissage profond. Puis, nous avons terminé par présentation de notre choix pour la méthodologie de développement de notre application.

Chapitre 3

Analyse et Conception

3.1 Introduction

Ce chapitre se subdivise en deux sections. La première section propose la spécification et la planification des fonctionnelles et non fonctionnelles du client et le diagramme des cas d'utilisation, la deuxième section représente une conception détaillé de la solution proposée en faisant l'appel du langage de modélisation UML.

3.2 Analyse

3.2.1 Spécification des besoins

3.2.1.1 Les besoins fonctionnelles

Les fonctionnalités principales de notre système pour la reconnaissance intelligente des émotions de résument en :

- Récupérer l'image à analyser.
- Déetecter le visage.
- Déetecter les différentes caractéristiques du visage.
- Enregistrer chaque image (avec le visage encadré).
- Calculer les distances caractéristiques des images cibles et celles du test.
- Comparer l'image cible avec les images de test.

- Retourner le résultat de reconnaissance faciale.
- Retourner le taux de ressemblance.

3.2.1.2 Les besoins non fonctionnelles

Notre système de reconnaissance des émotions doit satisfaire certains critères :

- Rapidité du calcul.
- Organisation et indexation de la base.
- La gestion des erreurs : Le système doit gérer mieux ses exceptions pour l'apparition d'un message d'erreur qui permet de filtrer les données et de ne prendre en considération que les données qui correspondent aux types adéquats.

3.2.2 Diagramme des cas d'utilisation

Dans notre cas d'utilisation, l'utilisateur insère l'image à analyser (l'image à insérer directement depuis la webcam). Ensuite, on lance la reconnaissance des émotions faciale, Finalement, le système effectue la reconnaissance. Il existe deux cas possibles :

- Emotion non reconnue (l'émotion n'appartient pas aux 7 émotions de base).
- Emotion reconnue (l'émotion appartient aux 7 émotions de base) Pour ces deux cas, l'application devrait retourner l'image avec la prédiction.

La figure 3.1 représente de cas d'utilisation de système de reconnaissance des émotions.

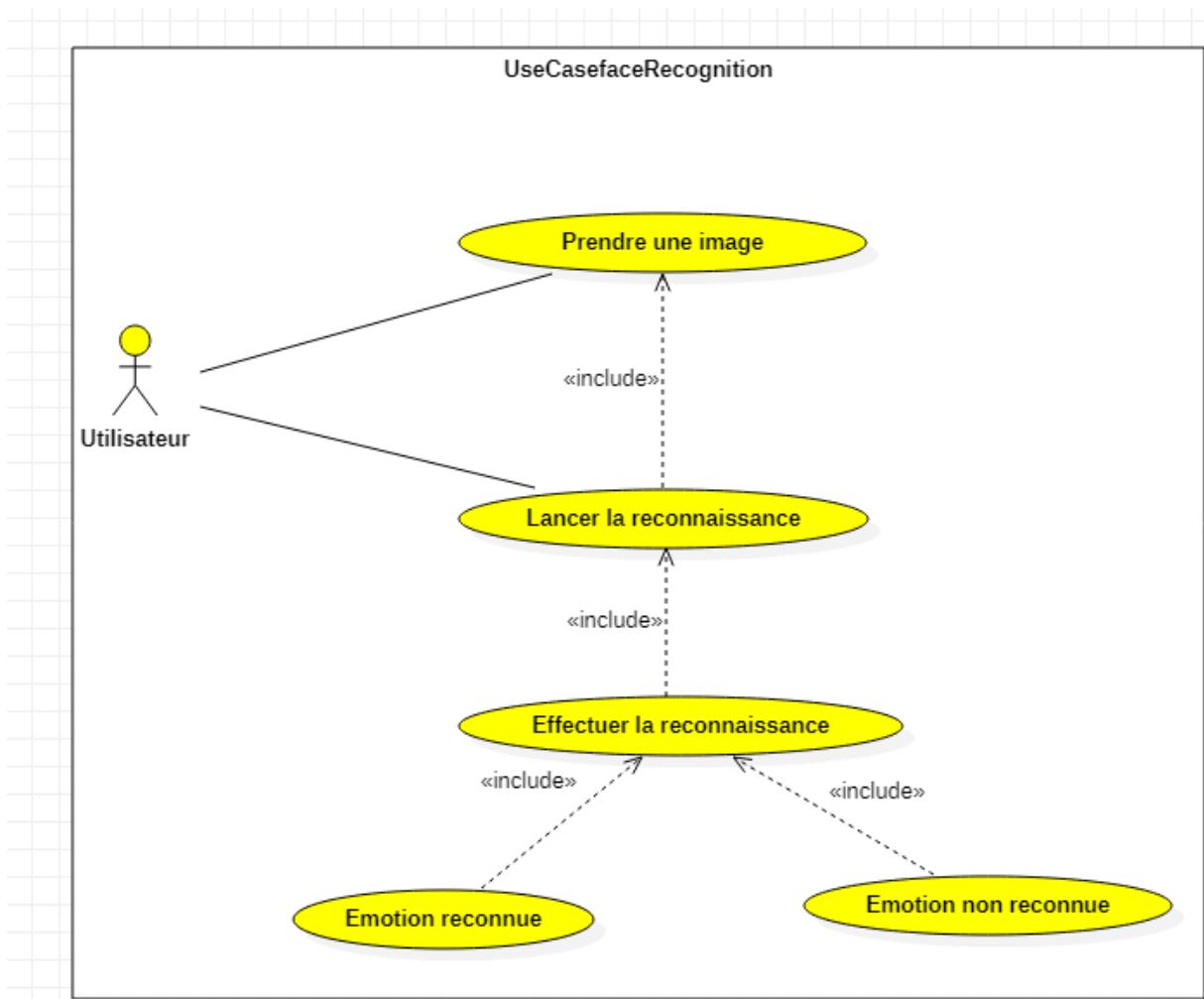


FIGURE 3.1 – Diagramme des cas d'utilisation "Emotion Recognition"

3.3 Conception

3.3.1 Diagramme de classes

Le diagramme de classe de notre système de reconnaissance des émotions comportent deux 5 classes comme suit :

- Image : la classe image est identifiée par son nom, sa résolution en pixel et son path ou Link.
- Image Cam : la classe Image Cam est hérité de la classe Image et identifiée par son

emplacement.

- Eléments.
- Comparaison : la classe comparaison est identifiée par son image cible et l'image source.

La figure 3.2 représente le diagramme de classes de système de reconnaissance des émotions.

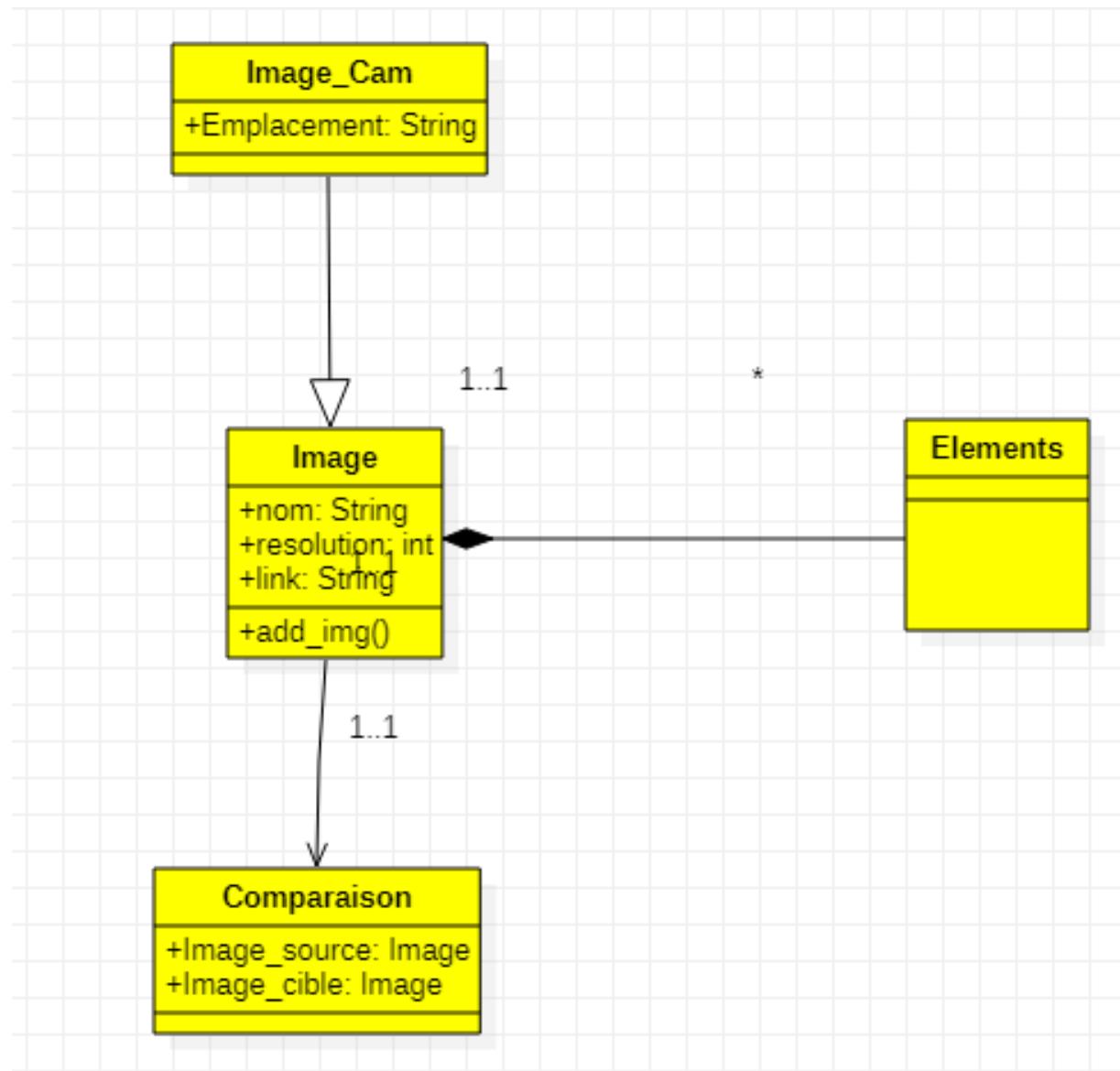


FIGURE 3.2 – Diagramme de classes

3.3.2 Diagramme de séquences

Dans notre diagramme de séquence, l'utilisateur introduit l'image à analyser depuis le « User Interface ». Puis, il se déroule le traitement de l'image insérer sur le serveur d'application .Si l'image est traitée correctement, le serveur lance l'algorithme d'apprentissage en se basant sur la base de données FER2013 et l'application décide si le présent soit heureux, triste, en colère, surpris, etc.

La figure 3.3 représente le diagramme de classes de système de reconnaissance des émotions.

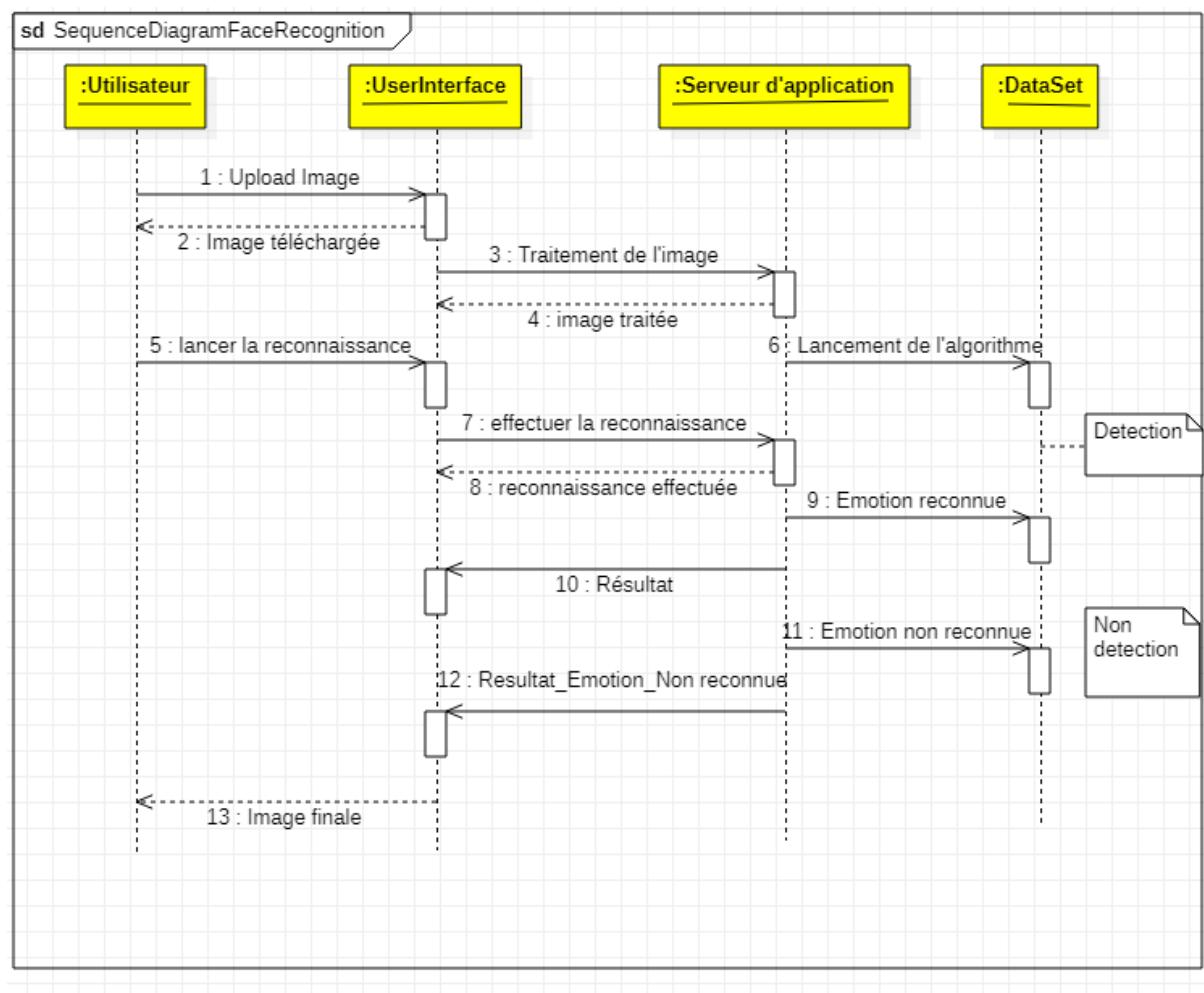


FIGURE 3.3 – Diagramme de séquences

3.4 Conclusion

Dans ce chapitre, nous avons présenté les besoins fonctionnelles et non fonctionnelles pour la réalisation du système de reconnaissance des émotions .Ainsi, les différents diagrammes UML qui nous aident à la contribution de notre application mobile.

Chapitre 4

Implémentation et Réalisation

4.1 Introduction

Ce chapitre se concentre sur l'environnement de travail, suivi des outils utilisés dans la mise en œuvre (matériel et logiciel), puis des résultats obtenus et enfin la présentation d'un exemple de notre solution «XpressVision».

4.2 Environnement de travail

4.2.1 Représentations d'outils utilisées

4.2.1.1 La partie hardware

Pour notre projet, nous avons utilisé un ordinateur portable MSI avec les caractéristiques suivantes :

Processeur (CPU)	Intel core-I5
Mémoire (RAM)	8.00 Go
Système d'exploitation	Windows 10, 64 bits
Ecran	15.6

TABLE 4.1 – La partie hardware de l'environnement de travail

4.2.1.2 La partie software

1. Le langage de programmation utilisé :

Après des recherches approfondies, il a été conclu que le langage de programmation le plus adapté à notre projet était Python, qui appartient au domaine de l'intelligence artificielle et, plus précisément, de la reconnaissance intelligente des émotions.

En fait, Python est un langage de script de haut niveau, structuré et open source. Il est multi paradigme et multi-usage et polyvalent. Ce langage de programmation présente de nombreuses caractéristiques intéressantes, telles que :

- Gratuit : Python est gratuit, mais il peut être utilisé pour des projets commerciaux sans restriction.
- Simplicité : Python est relativement facile à apprendre.
- Python convient pour des scripts d'une douzaine de lignes ou pour des projets complexes de plusieurs dizaines de milliers de lignes.
- Python gère ses ressources (mémoire, descripteurs de fichiers) sans l'intervention d'un programmeur [18].
- Python est largement utilisé en intelligence artificielle et, plus généralement, en analyse de données. Toutes ces caractéristiques font de Python l'outil idéal pour la mise en œuvre de nos applications [16].

2. Les bibliothèques utilisées :

Afin d'arriver à réaliser notre application, nous avons utilisées plusieurs bibliothèques. Parmi d'eux, on peut citer :

OpenCV

OpenCV (Open Source Computer Vision Library) est une bibliothèque gratuite de logiciels de vision et d'apprentissage automatique. La bibliothèque compte plus de 2500 algorithmes d'optimisation, dont un ensemble complet d'algorithmes classiques et de pointe de vision par ordinateur et d'apprentissage automatique [14].

La bibliothèque offre un large éventail de fonctionnalités telles que :

Traitements des images, Traitement vidéo, Algorithmes d'apprentissage.

Tensorflow

Tensorflow est une bibliothèque d'apprentissage automatique, une boîte à outils permettant de résoudre facilement des problèmes mathématiques extrêmement complexes.

La bibliothèque peut être utilisée pour former et exécuter des réseaux neuronaux pour la classification de chiffres manuscrits, la reconnaissance d'images, la suffixation de mots, les réseaux neuronaux récursifs, les modèles de séquence à séquence pour la traduction automatique et le traitement du langage naturel [15].

Numpy

Numpy est la bibliothèque de base pour le calcul scientifique en Python. Il fournit un objet tableau multidimensionnel très puissant et des outils pour travailler avec ces tableaux. La bibliothèque est fortement optimisée pour les opérations numériques et sa syntaxe est proche de celle de MATLAB.

Le principe de base de Numpy est de fournir des tableaux multidimensionnels. En outre, NumPy peut être considéré comme la base du calcul matriciel en Python, qui est conçu à des fins mathématiques et scientifiques [17].

Il contient, entre autres, un certain nombre de choses [18] :

Un puissant objet de type tableau à N dimensions, Fonctions complexes (diffusion), Outils d'intégration de code C/C++ et Fortran, Des capacités utiles d'algèbre linéaire.

Matplotlib

Matplotlib est une bibliothèque du langage de programmation Python permettant de tracer et de visualiser des données sous forme de graphiques. Il peut être combiné avec les bibliothèques de calcul scientifique Python NumPy et SciPy. Il fournit également une API orientée objet pour l'intégration de graphes dans des applications utilisant des outils d'interface graphique courants tels que Tkinter, wxPython, Qt ou GTK.

Plusieurs points rendent cette bibliothèque intéressante :

- La possibilité d'utiliser de nombreuses matrices (PNG, JPEG...) et vecteurs (PDF, SVG...) Formats de sortie.
- Documentation en ligne très complète, de nombreux exemples peuvent être trouvés

sur Internet.

- Une communauté forte et active.
- interface pylab : reproduction fidèle de la syntaxe MATLAB.
- Bibliothèques de haut niveau : idéales pour les calculs interactifs.

4.2.2 La base de données utilisée

Pour améliorer la précision et les performances de notre modèle, il est préférable d'entraîner le réseau avec de nombreux échantillons d'images.

La base de données **FER 2013** a été utilisée comme base de données d'images d'expressions faciales. Cette dernière comprend un total de 35887 images en niveaux de gris, 28709 images d'entraînement et 7178 images de test pré-cadrées de 48x48 pixels, chacune étiquetée avec l'une des 7 catégories d'émotions suivantes :

- 0 : 3995 images → Colère
- 1 : 436 images → Dégoût
- 2 : 4097 images → Peur
- 3 : 7215 images → Bonheur
- 4 : 4965 images → Neutre
- 5 : 4830 images → Tristesse
- 6 : 3171 images → Surprise

Toutes ces images ont été résolues par le site web kaggle et chaque image a une taille de (48x48 pixels).

4.3 Implémentation de l'application

Cette section décrit les différentes étapes de la réalisation de notre système « de reconnaissance intelligente des émotions ».

4.3.1 Module de détection visage

Pour détecter les visages, nous utilisons une méthode populaire proposée par Paul Viola et Michael Jones dans leur article de 2001 intitulé « Fast object detection using simple function cascades ». Dans notre système, nous utilisons (haarcascade frontalface alt2.xml) de la bibliothèque OpenCV, qui fournit la méthode Haar-Cascade ou d'autres méthodes de Viola et Jones.

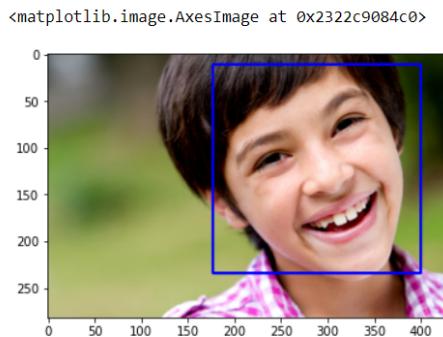


FIGURE 4.1 – Détection de visage et dessin de rectangle

4.3.2 Module de reconnaissance d'émotion

Notre modèle se compose de 3 groupes de couches.

- Le groupe initial de convolution complète se compose de 32 filtres qui effectuent une opération de normalisation par lots et une fonction d'activation ReLU.

- Le deuxième groupe représente la couche de profondeur comme un type de convolution où un seul filtre de convolution est appliqué par canal d'entrée. En d'autres termes, dans l'approche en profondeur, contrairement à la convolution normale (bidimensionnelle), la convolution profonde ne mélange pas les informations provenant de différents canaux d'entrée. Ce groupe effectue également une opération de normalisation par lots et une fonction d'activation ReLU.

- Le troisième groupe, représentant la couche de projection, consiste à réduire le nombre de canaux et à projeter les données d'une figure (canal) de haute dimension vers un tenseur de figure de basse dimension. Il effectue également une opération de normalisation des lots.

On a ajouté pour la sortie du réseau une couche de dense avec une fonction Softmax qui nous renvoi la classification.

Layer (type)	Output Shape	Param #	Connected to
input_5 (InputLayer)	[(None, 224, 224, 3 0)]		[]
Conv1 (Conv2D)	(None, 112, 112, 32 864)		['input_5[0][0]']
bn_Conv1 (BatchNormalization)	(None, 112, 112, 32 128)		['Conv1[0][0]']
Conv1_relu (ReLU)	(None, 112, 112, 32 0)		['bn_Conv1[0][0]']
expanded_conv_depthwise (Depth wiseConv2D)	(None, 112, 112, 32 288)		['Conv1_relu[0][0]']

FIGURE 4.2 – Module de reconnaissance d’émotion

4.3.3 Module de chargement des données

Dans la base de données Fer2013, les images sont traitées de manière à ce que les visages soient presque centrés et que chaque visage occupe approximativement la même espace dans chaque image.

Chaque image est regroupée dans l’une des sept catégories d’émotions faciales. C’est-à-dire :

0=fâché, 1=dégouté, 2=craintif, 3=heureux, 4=neutre, 5=malade, 6=surpris. Dans notre projet, nous avons utilisé 28709 images pour la formation, car il y a quelques images mal classées qui conduiraient à une formation incorrecte.

Les images sont répertoriées dans sept catalogues, un pour chaque émotion :

0 : Colère (3995 images).

1 : Dégoût (436 images).

2 : Peur (4097 images).

3 : Joie (7215 images).

4 : Neutre (4965 images).

5 : Tristesse (4830 images).

6 : Surprise (3171 images).

Pour ce faire, la fonction `create training data()` charge une image dans un tableau à deux colonnes [image, emotion index], chaque ligne contenant une image et sa catégorie. Après les avoir redimensionnées à (224*224).

Nous séparons les deux colonnes d'images du tableau `training data []` dans un tableau `X []` et les annotons dans un autre tableau `Y[]`, puis nous remodelons le tableau `X []` en ajoutant une quatrième dimension, puis nous convertissons les deux tableaux en type numpy, et enfin nous normalisons nos données entre 0 et 1 en les divisant par 255.

```

##### read how much images
print(len(training_Data))

28709

```

```

import tensorflow as tf
import cv2
import os
import matplotlib.pyplot as plt
import numpy as np

```

```

Datadirectory = 'C:/Users/MSI I5/Documents/training' #trainingdataset

classes = ["0","1","2","3","4","5","6"] #list of classes

for category in classes:
    path = os.path.join(Datadirectory,category)
    for img in os.listdir(path):
        img_array = cv2.imread(os.path.join(path,img))
        plt.imshow(cv2.cvtColor(img_array,cv2.COLOR_BGR2RGB))
        plt.show()
        break
    break

create_training_Data()
print("loaded")

```

```

loaded

X=[]
y=[]
for features , label in training_Data:
    X.append(features)
    y.append(label)

X=np.array(X).reshape(-1, img_size , img_size , 3)
X=X.astype('float16')

```

```

X.shape
(28709, 224, 224, 3)

type(y)
list

```

FIGURE 4.3 – Module de chargement des données

4.3.4 L'apprentissage de classificateur d'images de CNN avec Keras

Après la préparation des données, le classificateur a été entraîné en utilisant l'architecture CNN simple de MobileNetV2 et une classe appelée training.py a été utilisée pour mettre en œuvre le CNN pour l'architecture MobileNetV2. Nous avons commencé à utiliser l'apprentissage profond pour entraîner le classificateur d'images en utilisant Tensorflow.

deep learning model for training mobilenetv2 architecture

```
Entrée [148]: import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

Entrée [149]: temp = tf.keras.applications.mobilenet_v2.MobileNetV2() #####pretrained model

Entrée [150]: temp.summary() #mobilenet accepts 224
Model: "mobilenetv2_1.00_224"
+-----+-----+-----+-----+
| Layer (type) | Output Shape | Param # | Connected to |
+-----+-----+-----+-----+
| input_5 (InputLayer) | [(None, 224, 224, 3  0)] | [] |           |
| Conv1 (Conv2D) | (None, 112, 112, 32  864) | ['input_5[0][0]'] |           |
| bn_Conv1 (BatchNormalization) | (None, 112, 112, 32  128) | ['Conv1[0][0]'] |           |
| Conv1_relu (ReLU) | (None, 112, 112, 32  0) | ['bn_Conv1[0][0]'] |           |
| expanded_conv_depthwise (DepthwiseConv2D) | (None, 112, 112, 32  288) | ['Conv1_relu[0][0]'] |           |
+-----+-----+-----+-----+
base_input = temp.layers[0].input

base_output = temp.layers[-2].output

final_output = tf.keras.layers.Dense(128)(base_output) ##### adding new layer , after the output of global pooling layer
final_output = tf.keras.layers.Activation('relu')(final_output) #####activation function
final_output = tf.keras.layers.Dense(64)(final_output)
final_output = tf.keras.layers.Activation('relu')(final_output)
final_output = tf.keras.layers.Dense(7, activation='softmax')(final_output) #####my classes are 7, classification layer
#####

final_output
<KerasTensor: shape=(None, 7) dtype=float32 (created by layer 'dense_14')>

new_model = tf.keras.Model(inputs = base_input , outputs = final_output)

new_model.summary()
Model: "model_4"
+-----+-----+-----+-----+
| Layer (type) | Output Shape | Param # | Connected to |
+-----+-----+-----+-----+
| input_5 (InputLayer) | [(None, 224, 224, 3  0)] | [] |           |
| Conv1 (Conv2D) | (None, 112, 112, 32  864) | ['input_5[0][0]'] |           |
| bn_Conv1 (BatchNormalization) | (None, 112, 112, 32  128) | ['Conv1[0][0]'] |           |
| Conv1_relu (ReLU) | (None, 112, 112, 32  0) | ['bn_Conv1[0][0]'] |           |
| expanded_conv_depthwise (DepthwiseConv2D) | (None, 112, 112, 32  288) | ['Conv1_relu[0][0]'] |           |
+-----+-----+-----+-----+
```

FIGURE 4.4 – L'apprentissage de classificateur d'images de CNN

4.3.5 Module d'apprentissage

Après avoir créé le CNN, l'étape suivante consiste à l'entraîner. Pour entraîner le réseau, nous utilisons la fonction "fit()" de notre modèle, avec les paramètres suivants. Les 42

données d'apprentissage (x) sont les données d'image, (y) sont les étiquettes, et le nombre d'époques.

4.3.6 Résultat de l'apprentissage

Nous savions que le processus d'apprentissage de la compilation prendrait 18 heures, mais nous avons trouvé une solution très astucieuse utilisant la mémoire virtuelle et le temps d'apprentissage est devenu très rapide.

Une fois le modèle formé, nous allons essayer de lui attribuer certains paramètres.

- Epochs : fait référence au nombre d'époques (le nombre de fois que le modèle est exécuté sur les données).
- Perte : désigne le taux d'erreur.
- Précision : fait référence au taux de précision.

La précision de l'apprentissage et du test augmente à mesure que l'époque augmente, ce qui reflète le fait que plus d'informations sont apprises par modèle d'époque. Si la précision diminue, alors nous avons besoin de plus d'informations pour que notre modèle apprenne, et nous devons donc augmenter le nombre d'époques, et vice versa.

Là encore, l'erreur d'apprentissage et la précision diminuent lorsque le nombre d'époques augmente.

```

Epoch 1/25
898/898 [=====] - 2466s 3s/step - loss: 1.2392 - accuracy: 0.5334
Epoch 2/25
898/898 [=====] - 2375s 3s/step - loss: 1.0663 - accuracy: 0.6016
Epoch 3/25
898/898 [=====] - 2403s 3s/step - loss: 0.9990 - accuracy: 0.6267
Epoch 4/25
898/898 [=====] - 2365s 3s/step - loss: 0.9429 - accuracy: 0.6493
Epoch 5/25
898/898 [=====] - 2367s 3s/step - loss: 0.8943 - accuracy: 0.6688
Epoch 6/25
898/898 [=====] - 2493s 3s/step - loss: 0.8454 - accuracy: 0.6844
Epoch 7/25
898/898 [=====] - 2637s 3s/step - loss: 0.7971 - accuracy: 0.7062
Epoch 8/25
898/898 [=====] - 2357s 3s/step - loss: 0.7435 - accuracy: 0.7267
Epoch 9/25
898/898 [=====] - 2336s 3s/step - loss: 0.6895 - accuracy: 0.7447
Epoch 10/25

Epoch 11/25
898/898 [=====] - 2356s 3s/step - loss: 0.3321 - accuracy: 0.8807
Epoch 12/25
898/898 [=====] - 2355s 3s/step - loss: 0.3035 - accuracy: 0.8920
Epoch 13/25
898/898 [=====] - 2359s 3s/step - loss: 0.2735 - accuracy: 0.9050
Epoch 14/25
898/898 [=====] - 2358s 3s/step - loss: 0.2665 - accuracy: 0.9069
Epoch 15/25
898/898 [=====] - 2360s 3s/step - loss: 0.2318 - accuracy: 0.9187
Epoch 16/25
898/898 [=====] - 2359s 3s/step - loss: 0.2289 - accuracy: 0.9196
Epoch 17/25
898/898 [=====] - 2361s 3s/step - loss: 0.2038 - accuracy: 0.9289
Epoch 18/25
898/898 [=====] - 2361s 3s/step - loss: 0.2000 - accuracy: 0.9302
Epoch 19/25
898/898 [=====] - 2363s 3s/step - loss: 0.1942 - accuracy: 0.9308

```

Out[158]: <keras.callbacks.History at 0x22d1a2f5c10>

FIGURE 4.5 – Résultat de l’apprentissage

La précision de l’apprentissage et du test augmente avec le nombre d’époque, ce qui reflète le fait que le modèle apprend plus d’informations par modèle d’époque. Si la précision diminue, nous avons besoin de plus d’informations pour que notre modèle puisse apprendre, et nous devons donc augmenter le nombre d’heures calendaires, et vice versa.

Là encore, l’erreur d’apprentissage et la précision diminuent à mesure que le nombre d’époques augmente.

4.3.7 Présentation de l’application mobile

4.3.7.1 Présentation de l’environnement de travail

1. Présentation de la plateforme Android

Android est un système d’exploitation open source pour les appareils mobiles, conçu par Google, une start-up américaine qui a été rachetée par Google en juillet 2005. Ce système d’exploitation se distingue de ses concurrents principalement par le fait qu’il est ouvert.

Le modèle commercial de Google semble indiquer que les fabricants seront plus prompts à adopter Android. Les fabricants seront plus rapides parce que le système est gratuit pour les fabricants. C'est pourquoi il est logique pour nous de travailler sur cette plateforme. Une autre raison est que les applications Android sont développées en Java, un langage de programmation que nous avons appris en classe (en théorie et en pratique).

Android dispose également d'un ensemble de bibliothèques. Il comprend la plupart des fonctionnalités que l'on trouve dans Java, ainsi que des fonctionnalités supplémentaires pour gérer l'interface graphique. Par exemple, les fonctions utilisées pour gérer l'interface graphique, il existe plusieurs versions de cette plateforme mobile construite autour du noyau Linux 2.6 et son architecture est divisée en plusieurs couches qui permettent le développement d'applications innovantes.

- Noyau Linux.
- Une bibliothèque graphique.
- Une machine virtuelle (la machine virtuelle Dalvik).
- Un cadre d'application.
- Applications.

Une application Android est une collection de composants liés entre eux par des profils et basés sur un cycle de vie, ses composants sont :

- Les vues, qui sont comme les éléments d'une interface graphique et contiennent des composants organisés selon différentes dispositions.
- Contrôles Boutons, champs de saisie, cases à cocher, etc.
- Les activités peuvent être assimilées à un écran constitué d'un ensemble de vues et de contrôles. Il comporte deux composants, le composant métier et la gestion du cycle de vie implémentés en JAVA dans la classe qui hérite de l'Activité, et le composant interface défini en XML ou en JAVA.

Dans le cadre de la mise en œuvre de ce projet, nous avons dû utiliser un ensemble de technologies, nous allons décrire ces techniques dans les sections suivantes.

2. Langage de programmation utilisés.

- **Langage JAVA**

Le langage de programmation utilisé dans le cadre de ce projet Android est Java, qui est très courant dans le développement informatique actuel. Ce langage fournit un ensemble de bibliothèques faciles à développer et disposant d'une bonne documentation.

- **Langage XML**

Afin d'implémenter l'interface graphique de l'application, nous avons utilisé le langage XML qui est un langage informatique principalement utilisé pour stocker ou transférer des données textuelles structurées en champs arborescents.

Ce langage est caractérisé comme étant extensible car il permet à l'utilisateur de définir des balises (tags) pour faciliter la navigation dans un fichier et donc la lecture de son contenu du fichier et ainsi lire les informations, il est couramment utilisée et facile à apprendre.

4.3.7.2 Les étapes de réalisation

1. Entraînement et sauvegarde du modèle

- L'Entraînement de modèle CNN Keras sur le célèbre dataset kaggle FER 2013. Les données consistent en des images de visages en niveaux de gris de 48x48 pixels. Il est composé d'environ 28K images d'entraînement.
- La conversion et l'enregistrement du modèle dans un seul fichier protobuf (.pb). Cela peut être fait en utilisant Tensorflow comme backend.

2. Étapes impliquant Android Studio

- La configuration de opencv dans Android studio et quelques applications de base (comme l'ouverture de la caméra opencv).
- L'apprentissage à détecter les visages en temps réel en utilisant le classificateur opencv Haar-cascade.
- Le déploiement du modèle pré-entraîné dans notre application Android qui reçoit des données spécifiques en entrée et prédit les résultats. Ceci est réalisé en utilisant le backend TensorFlow.
- Le téléchargement des fichiers .AAR pour le support de l'interface d'inférence de TensorFlow.

- Le chargement du modèle. Ensuite, on prend la partie visage de l'humain à partir de l'image de la caméra et on la convertit en tableau flottant parce que le modèle ne prend en entrée qu'un tableau 2D de taille 48x48. Tout cela se passe en cours d'exécution.
- La réception du modèle et l'impression du résultat ayant la probabilité maximale.

4.3.8 Résultats de l'application

Dans cette partie, nous avons testé notre application mobile pour la reconnaissance des émotions « XpressVision » sur notre cousine.

Ci-joint quelques captures qui montrent le bien fonctionnement de notre application :

La figure 4.6 présente la probabilité d'un visage neutre :

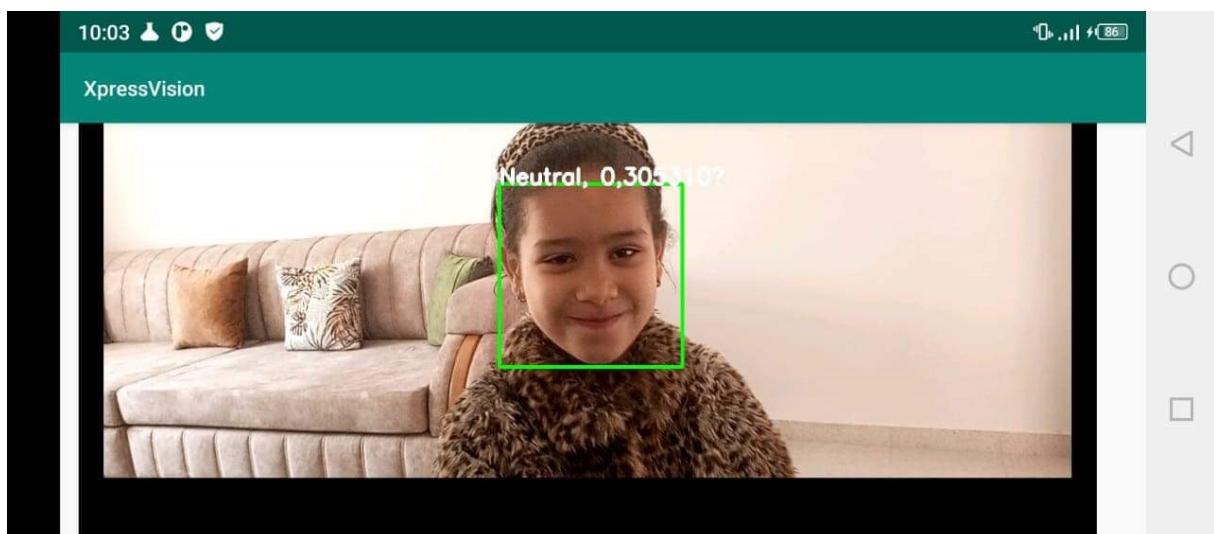


FIGURE 4.6 – Test d'un visage neutre

La figure 4.7 présente la probabilité d'un visage en colère :

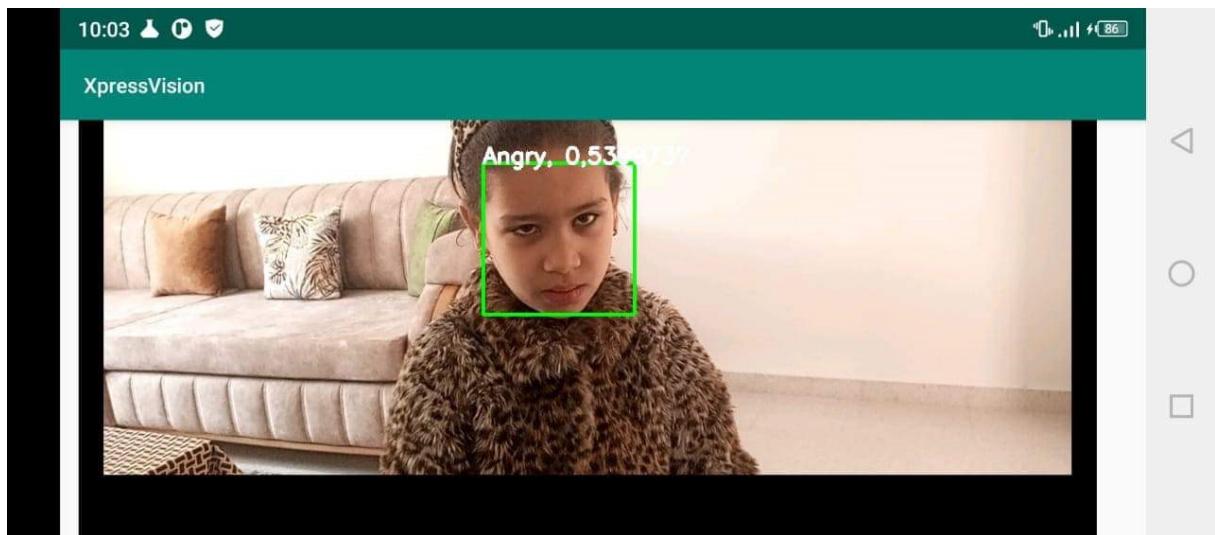


FIGURE 4.7 – Test d'un visage en colère

La figure 4.8 présente la probabilité d'un visage qui fait peur :

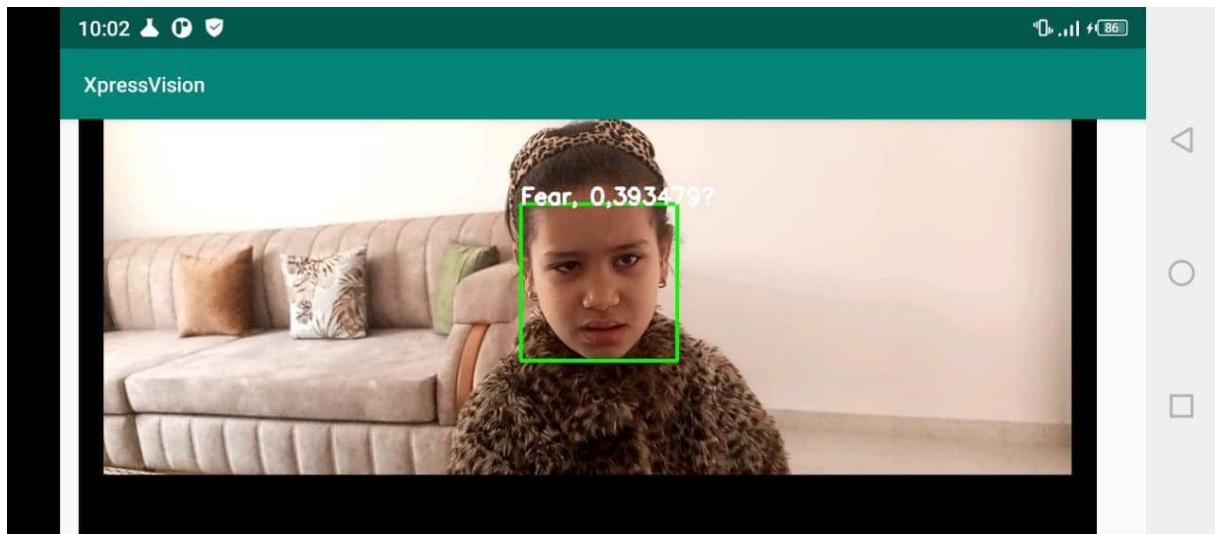


FIGURE 4.8 – Test d'un visage qui fait peur

4.4 Conclusion

Dans ce chapitre, nous avons présenté notre système de base de reconnaissance des émotions. Un modèle a été utilisé pour révéler les performances du système de détection des émotions à apprentissage profond. Nous avons ensuite élucidé l'environnement de

travail, les aspects matériels et les aspects logiciels pour mettre en œuvre notre système. Nous avons vu les résultats obtenus et présenté les performances de notre système.

Conclusion générale

L'objectif de ce projet est le développement d'une application mobile pour la reconnaissance intelligente des émotions basé principalement sur le modèle MobileNetV2. Le système a été entraîné sur la base de données Fer2013. Il est capable de reconnaître les émotions à partir d'une vidéo en temps réel. Vu la quantité de logiciels potentiels (sécurité, Réseaux sociaux,...) pouvant se baser sur cette application, celle-ci doit répondre à des exigences de rapidité et de robustesse des résultats.

Notre travail, comme tout travail de recherche, a une certaine reconnaissance et est considéré comme un travail futur. En premier lieu, on peut tester notre modèle sur d'autres bases de données plus importantes. De plus, on pourra Ajoutez des données d'entraînement supplémentaires à la base de données fer2013. En dernier lieu, on doit Inclure davantage de classes d'émotions secondaires pour reconnaître les micro-expressions.

Au final, ce projet nous a permis d'acquérir de nouvelles connaissances. Dans ce travail, nous avons pu découvrir de nouveaux concepts tels que la notion d'émotion et la manipulation des expressions faciales, et nous avons appris les difficultés de confondre émotion et expressions faciales, notamment dans la découverte de l'interaction homme-machine lors de la reconnaissance des émotions.

Références Bibliographiques

- [1] [https://go.affectiva.com/auto.](https://go.affectiva.com/auto)
- [2] [https://go.affectiva.com/affdex-for-market-research.](https://go.affectiva.com/affdex-for-market-research)
- [3] [https://www.affectiva.com/product/individual-product-page-imotions/.](https://www.affectiva.com/product/individual-product-page-imotions/)
- [4] [https://aws.amazon.com/fr/rekognition/.](https://aws.amazon.com/fr/rekognition/)
- [5] [https://business.lesechos.fr/entrepreneurs/idees-de-business/0600478877092-datakalab-transforme-les-emotions-en-donnees-326118.php.](https://business.lesechos.fr/entrepreneurs/idees-de-business/0600478877092-datakalab-transforme-les-emotions-en-donnees-326118.php)
- [6] [https://www.kairos.com/blog/kairos-commitment-to-your-privacy-and-facial-recognition-regulations.](https://www.kairos.com/blog/kairos-commitment-to-your-privacy-and-facial-recognition-regulations)
- [7] [https://azure.microsoft.com/fr-fr/pricing/details/cognitive-services/face-api/.](https://azure.microsoft.com/fr-fr/pricing/details/cognitive-services/face-api/)
- [8] [https://vokaturi.com/.](https://vokaturi.com/)
- [9] [https://play.google.com/store/apps/details?id=com.reaimagine.josem.emotimeter_facialemotionrecognizer&hl=fr&gl=US.](https://play.google.com/store/apps/details?id=com.reaimagine.josem.emotimeter_facialemotionrecognizer&hl=fr&gl=US)
- [10] [https://lu.linkedin.com/company/moodme.](https://lu.linkedin.com/company/moodme)
- [11] [https://play.google.com/store/apps/details?id=com.moodme.android_demo.](https://play.google.com/store/apps/details?id=com.moodme.android_demo)
- [12] [https://paperswithcode.com/method/mobilnetv2.](https://paperswithcode.com/method/mobilnetv2)
- [13] [https://fr.wikipedia.org/wiki/Mod%C3%A8le_en_cascade.](https://fr.wikipedia.org/wiki/Mod%C3%A8le_en_cascade)
- [14] [https://fr.wikipedia.org/wiki/OpenCV#Traitement_d'images.](https://fr.wikipedia.org/wiki/OpenCV#Traitement_d'images)
- [15] [https://www.lebigdata.fr/tensorflow-definition-tout-savoir.](https://www.lebigdata.fr/tensorflow-definition-tout-savoir)

- [16] RAOUNAK LILIA DAHAH. « La Détection de la colère chez le conducteur en utilisant le Deep Learning ». In : (2020).
- [17] Rebahi Ghediri IMANE, Semri KHAWLA et Belhouchette KENZA. « La Reconnaissance des émotions de base par Les réseaux de neurones : application de deep Learning ». In : (2021).
- [18] Foued NACER. « Reconnaissance d'expression faciale à partir d'un visage réel ». In : (2019).