

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITÉ DE 8 MAI 1945 - GUELMA -  
FACULTÉ DES MATHÉMATIQUES, D'INFORMATIQUE ET DES SCIENCES DE LA MATIÈRE

Département d'Informatique



Mémoire de Fin d'études Master

*Filière* : Informatique

*Option* : Sciences et Technologies d'Information et de Communication

*Thème* \_\_\_\_\_

**Conception et implémentation d'un service web pour  
la reconnaissance d'émotion faciale**

---

**Encadré Par :**

DR. CHOHRRA Chemseddine

**Présenté par :**

AMIAR Nor el houda

Juillet 2022

# Remerciements

Alhamdoulillah qui m'a facilité mon périple, et qui m'a fait sortir des ténèbres de l'ignorance à la lumière de la science et de la connaissance. Alhamdoulillah que j'ai pu finalement arriver ici, et que j'ai pu aboutir à un de mes objectifs dans la vie. Arriver ici n'était pas vraiment facile pour moi, mais grâce à l'aide des bonnes personnes que j'ai la chance de les avoir dans ma vie, je suis finalement ici.

Je commence par remercier sincèrement mon encadreur Dr. Chemseddine CHOHRA pour ses efforts géants. Ce jeune docteur me motivait, m'encourageait, et même influençait ma personne avec ses pensées et ses principes. Je lui remercie infiniment.

Mon encadreur n'était pas le seul à m'aider dans mon parcours, j'avais de la chance d'avoir des enseignants respectés, certains d'entre eux m'ont même donné plus que la simple information, leurs conseils étaient éclairants à plusieurs stages de mon parcours, et je suis certain que ces conseils vont encore m'aider dans ma vie poste-universitaire.

Je tiens à remercier aussi le staff administratif, pour tous qu'ils font pour garder la forteresse du département.

Le meilleur pour la fin! Mes parents la raison de mon existence, ma flamme inextinguible, ma source qui ne draine jamais. Je leurs dédie mon succès et tous mes accomplissements.

AMIAR CHERIF ,AMIAR SOUAD Soyez fières.

## RÉSUMÉ

Pour les êtres humains, l'expression du visage est l'un des plus puissants et naturels moyens pour communiquer leurs émotions et leurs intentions. Un être humain est capable de reconnaître les expressions du visage sans effort, mais, pour une machine, cette tâche est très difficile. A travers notre projet nous allons concevoir et implémenter un service web pour la reconnaissance d'émotion faciale. Ce dernier est destiné à être utilisé par un développeur qui pourrait envoyer des requêtes via un client HTTP en utilisant n'importe quel langage de programmation.

<b>TABLE DES MATIÈRES</b>
---------------------------

<b>Liste des figures</b>		<b>vii</b>
<b>Liste des tableaux</b>		<b>viii</b>
<b>1 Reconnaissance d'émotions Faciales</b>		<b>1</b>
1.1 Introduction . . . . .		1
1.2 Reconnaissance d'émotions . . . . .		2
1.3 Expression faciale . . . . .		2
1.4 Descriptions des expressions faciales avec les caractéristiques . . . . .		3
1.5 Domaines d'application de la reconnaissance d'émotions . . . . .		7
1.6 Méthodes de reconnaissance des émotions de base . . . . .		8
1.7 Méthodologie de la reconnaissance d'expression faciale . . . . .		12
1.8 Travaux connexes . . . . .		13
1.9 Conclusion . . . . .		18
<b>2 Les Services Web</b>		<b>19</b>
2.1 Introduction . . . . .		19
2.2 Définition d'un service web . . . . .		19
2.3 Le fonctionnement d'un service web . . . . .		20
2.4 Les protocoles de service web . . . . .		21

2.5	API . . . . .	33
2.6	Services web et IA . . . . .	37
2.7	Conclusion . . . . .	38
<b>3</b>	<b>Architecture du Système</b>	<b>39</b>
3.1	Introduction . . . . .	39
3.2	Architecture du Système . . . . .	39
3.3	Méthodologie de Conception . . . . .	40
3.3.1	Le langage UML . . . . .	40
3.3.2	Les Différents diagrammes UML . . . . .	40
	Vue statique . . . . .	41
	Vue dynamique . . . . .	41
3.4	Conception . . . . .	42
3.4.1	Identification des classes et fonctionnalités . . . . .	42
3.4.2	Représentation de diagramme de cas d'utilisation . . . . .	43
	Description textuelle cas d'utilisation . . . . .	44
3.4.3	Représentation de diagramme de classe . . . . .	46
	Description textuelle des classes . . . . .	47
3.4.4	Diagrammes de Séquences . . . . .	48
	La création d'un compte . . . . .	49
	Envoyer une image . . . . .	49
	Consulter l'historique . . . . .	50
3.5	Conclusion . . . . .	51
<b>4</b>	<b>Résultat et Discussions</b>	<b>52</b>
4.1	Introduction . . . . .	52
4.2	Outils de Développement . . . . .	52
4.2.1	Python . . . . .	53
4.2.2	FastAPI . . . . .	53
4.2.3	Bibliothèque FER (Facial Emotion Recognition) . . . . .	53

4.2.4	MTCNN . . . . .	54
4.2.5	OpenCV (Open Computer Vision) . . . . .	54
4.2.6	TensorFlow . . . . .	54
4.2.7	Wampserver . . . . .	55
4.2.8	Uvicorn . . . . .	55
4.2.9	Postman . . . . .	55
4.3	Présentation de l'application . . . . .	55
4.3.1	Page d'inscription . . . . .	56
4.3.2	Connexion . . . . .	56
4.3.3	Tester notre API . . . . .	57
4.4	Exemples d'utilisation . . . . .	59
4.4.1	Premier exemple (PHP) . . . . .	59
4.4.2	Deuxième exemple (Javascript) . . . . .	61
4.5	Conclusion . . . . .	61

TABLE DES FIGURES
-------------------

1.1	Les expressions faciales . . . . .	3
1.2	les 6 émotions faciales universelles . . . . .	4
1.3	Méthodologie de reconnaissance d'émotions . . . . .	12
2.1	Le fonctionnement d'un service web . . . . .	20
2.2	Échange entre le client et le serveur à l'aide du protocole http . . . . .	22
3.1	l'architecture de service web pour la détection d'émotions . . . . .	40
3.2	Les différents diagrammes du langage UML . . . . .	42
3.3	Diagramme de cas d'utilisation . . . . .	44
3.4	Diagramme de classe . . . . .	46
3.5	Diagramme de séquence pour la création d'un compte . . . . .	49
3.6	Diagramme de séquence pour l'envoi d'une requête . . . . .	50
3.7	Diagramme de séquence pour la gestion d'historiques . . . . .	51
4.1	Page d'inscription . . . . .	56
4.2	Connexion . . . . .	57
4.3	Interface Graphique de Postman . . . . .	57
4.4	Remplir un Formulaire dans Postman . . . . .	58
4.5	Résultat affiché par Postman . . . . .	59

4.6	Code PHP pour utiliser notre service web . . . . .	60
4.7	Résultat du code PHP . . . . .	60
4.8	Code Javascript pour utiliser notre service web . . . . .	61

LISTE DES TABLEAUX
--------------------

1.1	Description et déformations des expressions faciales . . . . .	7
2.1	Différence entre les différentes méthodes HTTP . . . . .	24
2.2	Les méthodes verbes HTTP . . . . .	25
2.3	Les codes de la classe d'information . . . . .	27
2.4	Les codes de la classe succès . . . . .	28
2.5	Les codes de la classe redirection . . . . .	29
2.6	Les codes de la classe "erreur de client" . . . . .	32
2.7	Les codes de la classe "erreur de serveur" . . . . .	33
3.1	description textuelle de classe "Utilisateur" . . . . .	47
3.2	description textuelle de classe "Images" . . . . .	48

## INTRODUCTION GÉNÉRALE

L'informatique est la science qui permet de traiter automatiquement et rationnellement les informations à l'aide d'un ordinateur. Un ordinateur est un appareil électronique qui traite les informations dans une unité centrale selon un programme enregistré en mémoire. Un programme est une suite ordonnée d'instructions traduites dans un langage de programmation et capable de résoudre un problème donné. L'informatique affective est le fait que l'ordinateur soit intelligent à comprendre, reconnaître et interpréter l'émotion humaine. Savoir si une machine pourrait un jour ressentir des émotions est une question posée et discutée depuis longtemps, sans qu'une réponse définitive n'ait été apportée. L'expression de l'émotion constitue une dimension importante en communication. Elle permet de définir une relation particulière avec le sentiment d'un individu, de la mettre à jour, de telle sorte qu'une personne peut, de façon fiable, à partir de comportements verbaux ou non verbaux (tels que l'attitude corporelle, les expressions faciales) de déduire l'état affectif actuel d'une autre personne avec laquelle elle interagit.

L'objectif de ce travail consiste à réaliser un service web pour la reconnaissance d'émotion faciale. Dans le but de faciliter l'utilisation de cet outil dans les différents domaines d'application sans la nécessité de l'implémenter à chaque fois.

Ce mémoire se compose de 4 chapitres :

— **Chapitre 01 (Reconnaissance d'émotions Faciales)** : Dans ce premier chapitre

nous allons introduire quelques notions (terminologies) et méthodes de reconnaissance d'émotion faciale.

- **Chapitre 02 (Les Services Web)** : Dans le deuxième chapitre, nous définissons ce que c'est un service web, et nous mettons le point sur les protocôles de communications, les standards et les formats utilisés.
- **Chapitre 03 (Architecture du Système)** : Ce chapitre est consacré à l'étude conceptuelle de notre système.
- **Chapitre 04 (Résultat et Discussions)** : Dans ce dernier chapitre nous montrons les langages et les outils utilisés pour le développement du service web, ainsi que la description détaillée de notre solution, avec des exemples d'utilisation dans quelques langages de programmation.

# CHAPITRE 1

## RECONNAISSANCE D'ÉMOTIONS FACIALES

### 1.1 Introduction

Au cours des dernières années, l'application de l'intelligence artificielle dans le domaine de la reconnaissance des expressions faciales a évolué rapidement. La reconnaissance des expressions faciales est considérée comme un sujet de recherche actif vu ses diverses applications dans le monde réel, notamment l'interaction homme-machine, la sécurité, les jeux, l'animation, éducation, domaine médical ··· etc.

Dans ce premier chapitre, nous allons présenter quelques notions sur la reconnaissance des expressions faciales, les descriptions avec les caractéristiques, les méthodes, les domaines d'application, la méthodologie de système et les algorithmes utilisés pour optimiser les résultats.

## 1.2 Reconnaissance d'émotions

La reconnaissance des émotions est le processus d'identification des émotions humaines, souvent à travers des expressions faciales qui représentent l'état psychologique d'une personne. Le sentiment ne devient une émotion que lorsque ce changement physique est évalué cognitivement.

Il y a Trois domaines principaux où les systèmes de reconnaissance d'émotions se sont développés [13] :

- La reconnaissance des émotions à partir de la parole (langage parlé).
- La reconnaissance des émotions à partir de signaux du système nerveux (EEG).
- La reconnaissance des émotions à partir d'images fixes ou de vidéos (Analyse d'expressions faciales).

Nous nous intéressons dans ce mémoire à la reconnaissance d'émotions par l'analyse des expressions faciales dans une image. La Figure 1.1 montre les expressions faciales correspondante à quelques émotions.

## 1.3 Expression faciale

L'expression faciale est une mimique faciale chargée de sens, elle est définie par un ensemble des signes du visage qui traduisent le sentiment d'une émotion. Dû à l'activation (volontaire ou non) ou mouvements de l'un ou de plusieurs muscles composant le visage, Ces mouvements transportent l'état émotionnel de l'individu aux observateurs [32].



FIGURE 1.1 – Les expressions faciales

## 1.4 Descriptions des expressions faciales avec les caractéristiques

Les émotions sont au centre de l'interprétation des expressions faciales, qui sont l'un des éléments les plus importants du processus de communication. D'après la recherche du psychologue Paul Ekman en 1960 sur les reconnaissances des émotions, il y a 6 expressions faciales dites universelles (universellement reconnues) [32] :

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

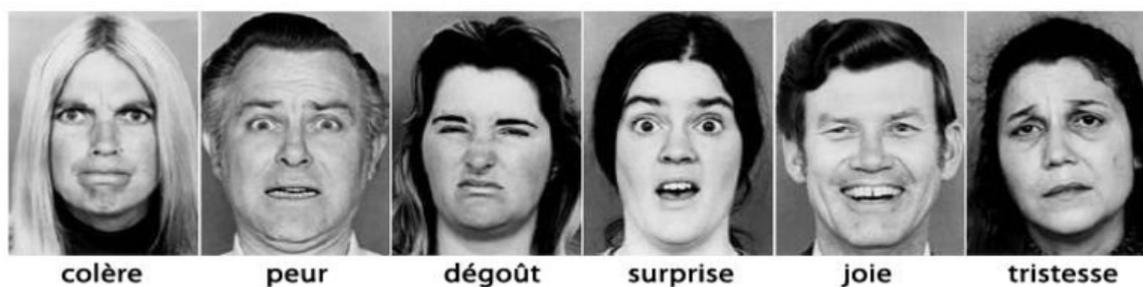
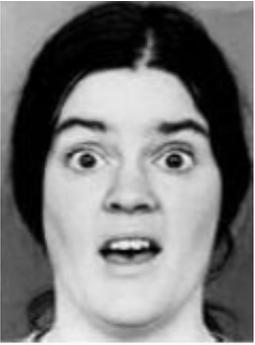
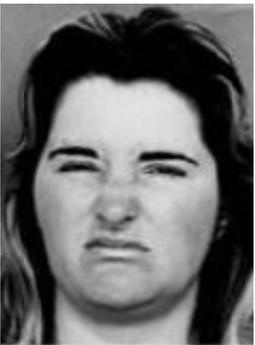


FIGURE 1.2 – les 6 émotions faciales universelles

Les expressions faciales émotionnelles se basent principalement sur le mouvement des muscles. Ces mouvements sont détectables par le changement de l'aspect visuel qui se résume à des changements de la forme des caractéristiques faciales et des déformations de la peau pour former des rides et des sillons. Nous expliquons dans le Tableau 1.1. la description et les déformations correspondant à des émotions faciales .

Expression faciale	Description	Déformation remarquable
 <p><b>colère</b></p>	<p>La colère est une réaction émotionnelle forte et peut également être une émotion dangereuse car elle pourrait provoquer de la violence. La source de la colère a de nombreuses raisons : les obstacles, les menaces verbales, les réclamations ... etc.</p>	<p>Le visage est contracté : froncement des sourcils, mâchoires serrées, narines pincées.</p> <ul style="list-style-type: none"> <li>— Il peut y avoir rougeur ou pâleur du visage.</li> <li>— La personne a des gestes de nervosité.</li> </ul>

 <p><b>tristesse</b></p>	<p>La tristesse apparaît quand une personne souffre. L'origine de la tristesse est typiquement une perte de quelque chose. Cette émotion calme et non impulsive, est souvent accompagnée de larmes.</p>	<ul style="list-style-type: none"> <li>— Remontée de la partie interne des sourcils.</li> <li>— Abaissement et rapprochement des sourcils.</li> <li>— Abaissement des coins externes des lèvres, qui signe la tristesse.</li> <li>— Les sourcils ont une position oblique.</li> <li>— Le regard est absent, les yeux sont brillants, humides, parfois surviennent des pleurs.</li> <li>— Des rides apparaissent sur le front.</li> </ul>
 <p><b>joie</b></p>	<p>La joie est un sentiment positif souvent associé à un sourire. Des sentiments de joie apparaissent lors de la réalisation des objectifs, ou du succès.</p>	<ul style="list-style-type: none"> <li>— Les coins des lèvres tirés vers le haut.</li> <li>— La bouche peut être ouverte et les dents peuvent être visibles.</li> <li>— Les joues peuvent être soulevées.</li> <li>— Les rides sous la paupière inférieure ainsi que les rides aux coins des yeux peuvent apparaître.</li> </ul>

 <p><b>surprise</b></p>	<p>La surprise est une émotion soudaine. Cela vient sans réfléchir et ne dure que peu de temps. Le début est une situation inattendue ou fausement attendue. Par conséquent, une émotion surprenante peut être positive ou négative.</p>	<ul style="list-style-type: none"><li>— La remontée des parties interne et externe des sourcils.</li><li>— Des rides horizontales pouvant apparaître sur le front.</li><li>— L'ouverture de la bouche et des yeux.</li></ul>
 <p><b>dégoût</b></p>	<p>Le dégoût est une émotion négative généralement évoquée par l'odorat, le goût ou la vision. Contrairement aux autres émotions, les objets évoquant le dégoût ne sont pas universels, mais culturels ou personnels, par exemple aliments. La réponse physiologique extrême est le vomissement.</p>	<ul style="list-style-type: none"><li>— La lèvre supérieure est levée, plus le dégoût est grand plus la dentition supérieure est visible.</li><li>— Les rides apparaissent sur le nez.</li></ul>

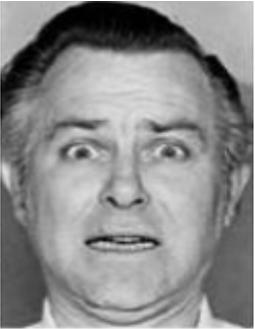
 <p><b>peur</b></p>	<p>La peur est induite par des situations dangereuses ou stressantes. On peut ressentir la peur des événements futurs, par exemple peur de la misère. Pour éprouver la peur, le corps de la personne se prépare à une évasion ou à une défense contre toute attaque possible.</p>	<ul style="list-style-type: none"> <li>— Les yeux sont ouverts et les pupilles deviennent larges.</li> <li>— Les sourcils sont soulevés et tirés vers l'intérieur.</li> <li>— Les paupières supérieures sont soulevées, ce qui a pour effet d'afficher un</li> <li>— Sanpaku supérieur (c-à-d, un blanc des yeux visible).</li> <li>— Les muscles de la bouche se tendent et abaissent souvent la lèvre inférieure.</li> <li>— La personne semble fixer quelque chose comme si elle ne pouvait s'en détacher.</li> </ul>
--	---	--

TABLE 1.1 – Description et déformations des expressions faciales

## 1.5 Domaines d'application de la reconnaissance d'émotions

La reconnaissance des émotions se développe ces dernières années en raison de différents domaines d'application comme :

- **Marketing** : Applications pour mesurer la satisfaction des clients et prédire les produits qui les intéressent ou qu'ils utilisent le plus et qu'ils aiment, Ainsi que des applications utilisées pour détecter l'émotion des clients. Après avoir reconnu l'émotion d'un client en colère ou satisfait, le système peut donner la

priorité aux clients en colère pour satisfaire leurs besoins afin d'améliorer les ventes. Du côté marketing, le processus de reconnaissance des émotions a plus d'impact sur la publicité pour soutenir les décisions d'achat.

— **Santé :**

Les applications qui aident à détecter certaines maladies mentales, d'apprentissage émotionnel des enfants autistes, aide aux enfants souffrant de problèmes psychologiques d'élocution. Des systèmes sont développés aussi pour réduire le stress, la dépression et l'anxiété chez les patients à partir de leurs émotions.

— **Éducation :**

Dans le cadre d'apprentissage à distance des systèmes de tutorat en ligne sont conçus pour prédire les émotions des étudiants et modifier la méthodologie d'enseignement en conséquence. Par exemple, dans le système de production MOOC (Massive Open Online Course), le tuteur est capable de prédire l'humeur de l'étudiant et le doute peut être clarifié. Le système de tutorat reconnaît l'émotion et adopte une attitude positive pour maximiser le processus d'apprentissage. Ainsi, il fournit un moyen d'apprentissage plus interactif et efficace, puis donne un retour d'information et guide vers une solution au problème des élèves (Leony et al., 2015).

— **Interaction Homme-Machine :**

robot d'accompagnement, voiture intelligente, la maison intelligente, la cuisine intelligente, le miroir interactif intelligent, le climatiseur intelligent et le système de musique pour offrir une meilleure ambiance. Ces applications peuvent reconnaître l'émotion de l'utilisateur par l'expression faciale.

## 1.6 Méthodes de reconnaissance des émotions de base

Elles peuvent être divisées globalement en quatre approches principales qui sont : l'approche basée modèles, l'approche basée règles, l'approche basée géométrie et l'approche basée réseaux de neurones [24].

**Approche Basée Modèles :**

Cette approche consiste à voir le visage comme un tout. L'analyse consiste alors à mesurer la ressemblance du visage observé à un modèle (connu ou appris). Les méthodes de cette approche font généralement appel à des méthodes de mise en correspondance de modèles. L'intérêt de ces méthodes est qu'elles peuvent être appliquées de manière plus locale. Ainsi la majorité de ces méthodes qui sont utilisées pour le visage sont aussi utiles pour n'importe quel objet et plus particulièrement pour les composantes du visage.

**Les avantages :**

- Peuvent s'adapter à beaucoup de problèmes.
- Moins sensibles au bruit par rapport aux méthodes classiques.
- En plus elles sont robustes par rapport à l'occultation pour des visages de la base d'apprentissage et même sur des visages non connus (hors de la base d'apprentissage).
- Ces méthodes sont bien adaptées à l'extraction d'informations sur des composantes dont certains états sont difficiles à déterminer (le gonflement de la joue par exemple).

**Inconvénients :**

- Difficile de construire une base d'apprentissage.
- La taille de la matrice d'apprentissage est fixée donc les images doivent être redimensionnées.
- Ces méthodes ne donnent pas des mesures mais seulement des paramètres, par exemple il est difficile d'implémenter une tâche qui indique quel est le degré, en pixel, de la bouche, tant qu'il est facile d'implémenter une tâche qui décide si la bouche est ouverte ou fermée.

**Approche Géométrique**

Depuis peu de temps, les informaticiens utilisent l'analyse géométrique pour représenter l'information faciale car elle donne des mesures précises qui ne donnent pas les approches basées modèle. Dans cette approche, les mouvements sont détectés

à l'aide des calculs des distances entre les points caractéristiques entre deux images l'une à l'état neutre et l'autre expressive. Lien et al. proposent une méthode hybride basée sur deux étapes :

1. Suivi des points faciaux (points autour des contours des yeux, des sourcils, du nez et de la bouche manuellement localisés sur la première image de la séquence vidéo).
2. Le flux optique (mouvement apparent des points faciaux).
3. Détection des rides afin d'extraire des informations sur l'expression.

Tian et al. utilisent deux réseaux de neurones séparés afin de reconnaître six unités d'actions de la partie supérieure du visage et dix de la partie inférieure, leur méthode est basée sur les traits permanents du visage ainsi que les transitoires. Les traits faciaux sont initialisés manuellement sur la première image et suivis dans le reste des images de la séquence vidéo. La reconnaissance des expressions faciales est réalisée par la combinaison des unités d'actions des deux parties supérieure et inférieure du visage.

Cohen et al. ont développé un système basé sur un algorithme de suivi des traits faciaux afin d'extraire les mouvements locaux des composantes faciales. Ces mouvements forment les entrées d'un réseau bayésien utilisé pour reconnaître les six expressions faciales. La représentation basée composantes exige une détection efficace et fiable des composantes faciales pour faire face à la variation de l'illumination, du mouvement principal significatif et de la rotation.

#### **Inconvénients :**

Le problème majeur des méthodes de géométrie est la détection non exacte des points caractéristiques du visage sur lesquels des mesures géométriques seront calculées à cause des variations de l'illumination, de l'occlusion et de beaucoup d'autres facteurs.

#### **Approche Basée Règles :**

Elle a été proposée par Pantic et Rothkrant. La méthode réalise une codification automatique en unités d'action. Un multi détecteur automatique de composantes du

visage est appliqué. Depuis les contours des composantes, des modèles de composantes sont extraits. La différence est ensuite calculée entre le modèle détecté et le modèle correspondant du visage à l'état neutre. Les règles de production classifient le modèle détecté dans la classe appropriée. La méthode proposée a été testée sur un ensemble de 49 vues et le taux de reconnaissance étaient de 92% pour les unités d'actions de la partie supérieure du visage et 86% pour celles du bas du visage. La classification finale dans une des six catégories correspondantes aux six expressions universelles est réalisée à l'aide de la description linguistique donnée par Eckman. La performance globale du système a été testée sur l'ensemble de 265 vues représentant six expressions faciales affichées par huit sujets. Le taux de reconnaissance réalisé était de 91%.

**Inconvénients :**

Le problème majeur de ces méthodes est qu'aucune méthode n'arrive à reconnaître les 46 unités d'actions définies dans FACS.

**Approches Basées Réseaux de Neurones :**

Un réseau de neurones peut être vu comme une fonction ayant un certain nombre d'entrées et un certain nombre de sorties. Le principe de l'apprentissage est de donner en entrée du réseau un certain nombre d'exemples et de fixer la sortie à la valeur désirée. Une méthode d'apprentissage permet alors aux neurones de s'adapter au mieux pour qu'ils affichent la même sortie quand on lui donne des données proches des celles d'apprentissage.

**Avantages :**

Les réseaux de neurones sont généralement utilisés pour leur faible sensibilité au bruit (robustesse au bruit) et leur capacité d'apprentissage.

**Inconvénients :**

La plupart de ces méthodes sont testées uniquement sur des images utilisées au cours de l'apprentissage, c'est pourquoi on ne peut prévoir le comportement de ces méthodes dans le cas de sujets inconnus. En outre, la plupart de ces méthodes exigent une intervention manuelle.

## 1.7 Méthodologie de la reconnaissance d'expression faciale

Un système qui effectue une reconnaissance automatique des expressions faciales est généralement composé de trois modules principaux, comme illustre dans la Figure 1.3 : [20]

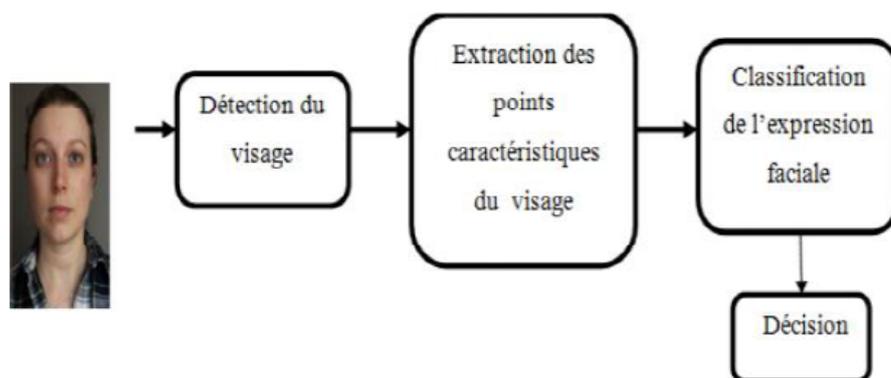


FIGURE 1.3 – Méthodologie de reconnaissance d'émotions

1. Le premier module consiste à détecter et enregistrer la région du visage dans les images ou les séquences d'images d'entrée. Il peut s'agir d'un détecteur de visage dans chaque image. La localisation du visage dans l'image se fait à l'aide de diverses techniques et algorithmes.
2. Le deuxième module consiste à extraire des points caractéristiques du visage et représenter les changements faciaux causés par les expressions faciales. Plusieurs techniques existent pour l'extraction de points caractéristiques. Les exemples incluent la transformation de caractéristiques invariantes d'échelle (SIFT), l'analyse discriminante linéaire (LDA), les ondelettes de Gabor, les caractéristiques robustes accélérées par moments (SURF) et bien d'autres. Les ondelettes de Gabor sont fortes et puissantes contre les influences photométriques perturbatrices tout en donnant un vecteur de caractéristiques avec une mesure énorme. SURF inclut uniquement l'accent sur les deux fonctionnalités les plus ancrées.

3. le dernier module détermine une similarité entre l'ensemble des caractéristiques extraites et un ensemble de caractéristiques de référence. Il existe différents classificateurs pour classer les émotions. Certains des classificateurs populaires sont les classificateurs discriminants linéaires (LDC), le k-plus proche voisin (k-NN), le modèle de mélange gaussien (GMM), les machines à vecteurs de support (SVM), les réseaux de neurones artificiels (ANN), les algorithmes d'arbre de décision et le Markov caché. Modèles (HMM).

D'autres filtres ou modules de pré traitement de données peuvent être utilisés entre ces modules principaux pour améliorer les résultats de détection, d'extraction de caractéristiques ou de classification.

## 1.8 Travaux connexes

Afin de réaliser l'expression des émotions faciales, des algorithmes de caractérisation qui examinent les images de manière plus progressive sont nécessaires qu'une simple identification à actualiser par un système informatique utilisant des appareils donnés ou planifiés sans personne d'autre. Un nombre considérable d'outils et d'algorithmes ont été créés ou utilisés pour y parvenir.

- Goal et al. (2016) ont suggéré le mélange de modèles experts pour la prédiction des émotions dans les films. C'est une manière dynamique de prédire l'émotion dans les films par l'expression faciale et l'audio (musique). L'histogramme de la zone du visage et la compression vidéo sont utilisés pour reconnaître l'expression du visage. Les premiers formats audio uniquement et vidéo uniquement sont utilisés séparément pour la détection des émotions. La méthode de fusion ultérieure combinant l'audio et la vidéo rend le mélange dans ce système multi-modèles. Il est incapable de prédire les sous types d'émotions [6].
- Chavan et al. (2013) présentent une réponse dynamique pour les émotions de base telles que la joie, la tristesse, la peur, le dégoût, la surprise, la neutralité et la colère en utilisant des expressions faciales. L'utilisation de la courbe de

- Bézier reconnaît l'émotion de manière dynamique. Il fournit les meilleurs résultats [6].
- Mistry et al. (2016) ont proposé l'algorithme micro génétique intégré à l'optimisation des essaims de particules. Il a utilisé différents classificateurs pour détecter les sept dimensions de l'état émotionnel et l'opérateur hvnLB proposé effectue une comparaison horizontale et verticale des pixels de voisinage pour récupérer les traits faciaux discriminatoires initiaux. Il donne de meilleures performances par rapport aux autres algorithmes classiques présents dans cet article [6].
  - Malassiotis et Tsalakanidou (2011) présentent une méthode de détection d'émotion par reconnaissance d'expression faciale dans la séquence d'image 2D et 3D. Les émotions sont heureuses, tristes, dégoûtées, surprises et neutres. Unité d'action faciale et reconnaissance des expressions faciales par capteurs 3D. Technique de modèle de forme active utilisée dans le suivi de visage 3D. La séquence d'images 3D est plus précise pour détecter l'émotion par différentes variations de pose. Cela prend en charge la classification des émotions en temps réel [6].
  - Lee et al. (2014) utilise une caméra mobile pour détecter le système de reconnaissance des émotions d'expression faciale. Pondéré La classification floue k-NN est utilisée pour comparer le résultat. La triangulation de Delaunay forme des maillages triangulaires pour cartographier vecteurs dans l'image du visage. Le modèle d'apparence active est utilisé dans cet article pour extraire l'ensemble de fonctionnalités d'entrée. C'est dynamique processus de prise d'image en séquence de vidéo. Seulement heureux, les émotions tristes et neutres peuvent être reconnues par ce méthode [6].
  - Liu et Yin (2015) ont proposé des soins du visage spontanés analyse des expressions. Dans cette étude, la vidéo thermique infrarouge descripteur est utilisé afin d'améliorer la spontanéité reconnaissance des émotions. Il représente l'image sous forme de vidéo thermique clips. Il est principalement basé sur la

- température générée par mouvement du corps et mouvement musculaire qui produit Chaffer. Il capture l'émotion par la caméra thermique FLIR. Cette méthode est utilisée pour identifier la lombalgie comme réelle ou faux. Il est utile de détecter les changements spontanés de émotions [6].
- Zen et al. (2016) proposent le modèle personnalisé pour la reconnaissance faciale. Transfert de paramètres transductifs (TPT), un cadre de construction personnalisé modèles de classification et définir une application. Il est utilisant les données visuelles et les mouvements gestuels pour détecter l'émotion de l'utilisateur. Il est indépendant de l'utilisateur. Il fournit précision à faible coût de calcul. Reconnaissance gestuelle et l'expression faciale sont supposées "taille unique". Ce est principalement conçu pour la personne moyenne applicable pour unité d'action, détection de la douleur et geste basé sur une montre intelligente reconnaissance [6].
  - Ramachandran Vedantham et ses collègues ont proposé un système de reconnaissance des expressions faciales en temps réel dans une vidéo en utilisant le classificateur du plus proche voisin. La méthode proposée comporte cinq étapes, La première étape est d'acquérir de la vidéo en temps réel à partir d'une webcam en utilisant les Progiciels d'acquisition d'images existants de MATLAB. Ensuite détecter le visage dans les images en utilisant la méthode Viola Jones avec les caractéristiques du Haar. La troisième étape consiste à trouver les caractéristiques importantes dans un visage en utilisant le filtre du Gabor. Finalement, ils ont utilisé la méthode des K plus proche voisin (KNN) pour la classification. La méthode a été testée sur la base de données Cohn Kanade qui contient six expressions de base telles que la colère, la joie, le dégoût, la peur, la tristesse et la surprise. Cette méthode a enregistré un taux de reconnaissance moyen de 96,67% [35].
  - Qi-rong Mao, Xin-yu Pan, Yong-zhao Zhan et Xiang-jun Shen proposent une approche de reconnaissance des émotions en temps réel basée sur les caractéristiques d'expression faciale 2D et 3D capturées par les capteurs Kinect. Pour

capturer la déformation du maillage 3D pendant l'expression faciale, ils combinent les caractéristiques des unités d'animation (AUs) et des positions des points de caractéristique (FPPs) suivies par Kinect. Un algorithme de fusion basé sur des profils émotionnels améliorés (PEI) et une confiance maximale sont proposés pour reconnaître les émotions avec ces caractéristiques d'expression faciale en temps réel. Des expériences sur un ensemble de données d'émotions et une vidéo en temps réel montrent les performances supérieures de leur méthode [19].

- Matthew Ratliff et Eric Patterson discutent d'un cadre pour la classification des états émotionnels, basé sur des images fixes du visage. La technique qu'ils présentent implique la création d'un modèle d'apparence active (AAM) formé sur des images de visage à partir d'une base de données accessible au public pour représenter la clé de variation de forme et de texture à la reconnaissance d'expression. Les paramètres de l'AAM sont utilisés comme caractéristiques pour un schéma de classification capable d'identifier avec succès les visages liés aux six émotions universelles. Les résultats de leur étude démontrent l'efficacité des AAM pour capturer l'importante structure faciale pour l'identification de l'expression et aident également à suggérer un cadre pour le développement futur [26].
- Miss Sneha et Dr Desai proposent une technique de reconnaissance des émotions faciales, cette technique utilise un LBP modifié, qui effectuent une comparaison horizontale et verticale des pixels de voisinage, pour générer une faciale initiale discriminante représentation. Ensuite, un PSO modifié est proposé pour effectuer l'optimisation des fonctionnalités. Le système de reconnaissance des émotions faciales se compose de trois étapes : 1) Extraction de caractéristiques ; 2) Optimisation des fonctionnalités ; 3) Reconnaissance des émotions. Premièrement, ils utilisent des motifs binaires locaux modifiés (LBP), c'est-à-dire des voisinages horizontaux et verticaux. Comparaison LBP, pour extraire la représentation faciale initiale. Ensuite, l'algorithme PSO proposé est utilisé

pour identifier les caractéristiques discriminantes et significatives pour différencier les expressions faciales distinctes. SVM (Machine à vecteur de soutien) et Les classificateurs SVM multiples sont utilisés pour reconnaître six expressions faciales [16].

- Hamid Sadeghi et Abolghasem-A. Raie ont présenté un système de reconnaissance des expressions faciales qui comprend trois étapes essentielles, la première étape consiste à détecter le visage avec une méthode points de repères (landmarking). Ensuite l'extraction des caractéristiques est la deuxième étape, dans cette étape l'image du visage détecté est convoluée avec le filtre de Gabor puis les matrices de convolution obtenues sont correctement codées sur la base des réponses maximales et minimales. A la fin de cette étape d'extraction, le vecteur de caractéristique est obtenu en calculant l'histogramme de ces codes. Enfin pour la classification des caractéristiques les auteurs ont utilisé le classificateur bien connu SVM à l'aide de la boîte à outils LIBSVM dans Matlab. Les auteurs ont testé la méthode sur ces ensembles de données CK+, SFEW et MMI et ont obtenu respectivement des taux de reconnaissance de 95.72%, 36.92% et 71.88% [29].
- Mahesh M. Goyani, a essayé d'accroître la précision et la robustesse globales du système de reconnaissance d'expressions faciales. D'abord la première étape consiste à détecter le visage par la méthode de Haar Cascade. Puis l'extraction de deux types de caractéristiques en utilisant le filtre de Gabor pour les détails structurels et le Modèle directionnel local (LDP) pour les détails texturaux qui se sont fusionnées à l'aide de l'analyse de corrélation canonique (CCA). Enfin, l'utilisation de l'analyse en composantes principales (ACP) et de l'analyse discriminante linéaire (LDA) pour obtenir un vecteur de caractéristique optimal. La technique est testée sur un ensemble de données de référence comme CK, JAFFE, TFEID et CASIA-VIZ et sur deux ensembles de données d'expression spontanée appelés SFEW (standard) et WESFED (interne), dans des conditions

différentes. Les résultats sont comme suit : dans la base de données CK la précision égale à 99.1% et dans la base de données JAFFE la précision égale à 100% par contre avec la base de CASIA-VIZ la précision est limitée égale à 92% et à la base SFEW la précision est seulement de 72.1% [28].

## **1.9 Conclusion**

Dans ce premier chapitre, nous avons présenté quelques notions et des concepts pour la Reconnaissance des expressions faciales. Nous avons eu à définir la reconnaissance, les expressions faciales, la citation de 6 émotions avec ses descriptions et ses déformations, les domaines de la reconnaissance, et ses méthodes. A la fin nous avons présenté quelques travaux dans le domaine pour illustrer les différentes méthodes de reconnaissance d'émotions qu'on peut trouver dans la littérature.

## CHAPITRE 2

## LES SERVICES WEB

### 2.1 Introduction

De nombreuses interactions se passent entre l'homme et la machine. Mais il s'en passe tout autant entre les machines. Des requêtes et des réponses sont constamment envoyées entre le client et le serveur. Cet échange est rendu possible grâce aux Web services ou bien le service Web en français. Dans ce chapitre, nous présentons quelques notions sur le service web, leur fonctionnement, les protocoles utilisées et API et quelques exemples d'utilisation des services web.

### 2.2 Définition d'un service web

Les services Web sont une nouvelle génération d'applications Web. Ce sont des applications modulaires autonomes, auto-descriptives qui peuvent être publiées, localisées et appelées sur le Web. Les services web exécutent des fonctions, qui peuvent aller de simples requêtes à des processus métier complexes. Une fois qu'un service web est déployé, d'autres applications (et d'autres services Web) peuvent découvrir et appeler le service déployé. Lorsqu'un service Web est utilisé, un client envoie une

requête à un serveur et déclenche ainsi une action auprès de ce serveur. Le serveur renvoie ensuite une réponse au client souvent dans un langage semi-structuré\* (JSON est le format le plus utilisé actuellement, mais il est également possible d'utiliser XML).

## 2.3 Le fonctionnement d'un service web

Le fonctionnement d'un service peut être résumé d'une 3 étapes très simples de la manière suivante [12] :

1. Le client ( un utilisateur sur un ordinateur) effectue une requête dans un des langages suivants : XML, JSON ou HTTP.
2. Cette requête est transmise à un serveur distant via les protocoles SOAP, REST ou HTTP.
3. La réponse est ensuite délivrée sous le même format que sa demande : XML, JSON ou HTTP.



FIGURE 2.1 – Le fonctionnement d'un service web

Et ce grâce à un modèle en couche. La première couche est l'invocation : elle décrit la structure des messages échangés par le client et le serveur. Elle s'effectue à l'aide de standard SOAP (Simple Object Access Protocol) ou l'approche REST. La deuxième est la découverte, C'est la phase de recherche et de localisation des données demandées par le client. La troisième est la description qui stipule les paramètres des fonctions

et les types de données des services web utilisés. Son protocole standard est le WSDL qui repose sur la notation XML [12].

## 2.4 Les protocoles de service web

Comment nous l'avons déjà expliqué, le service web est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit d'une technologie s'appuyant sur un ensemble de protocoles Internet très répandus (SOAP, XML, JSON, HTTP, REST, RESTFUL, WSDL, URL...), afin de communiquer.

Les services web sont normalisés car ils utilisent les standards XML et HTTP pour transférer des données et ils sont compatibles avec de nombreux autres environnements de développement.

### **XML :**

Pour Extensible Markup Language, désigne un langage informatique (ou métalangage pour être plus précis) utilisé, entre autres, dans la conception des sites web et pour faciliter les échanges d'informations sur Internet. Ce langage de description a pour mission de formaliser des données textuelles. Il s'agit, en quelque sorte, d'une version améliorée du langage HTML avec la création illimitée de nouvelles balises. Le XML se classe dans la catégorie des langages de description (il n'est ni un langage de programmation, ni un langage de requêtes). Il est donc naturellement utilisé pour décrire des données en s'appuyant sur des balises et des règles personnalisable [38].

### **HTTP :**

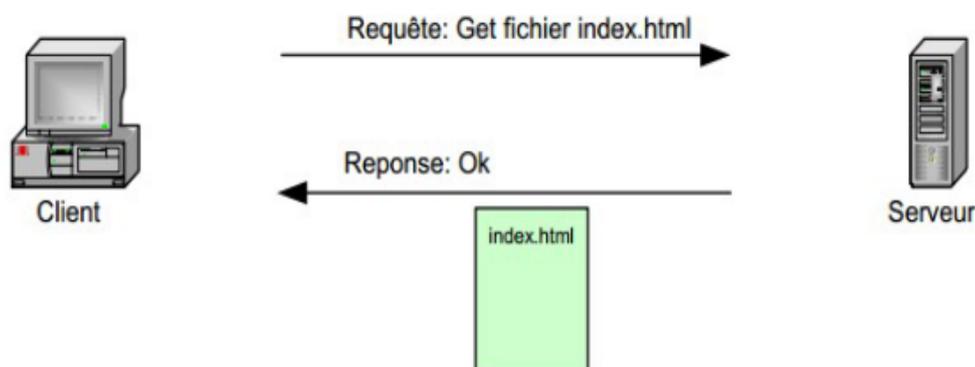


FIGURE 2.2 – Échange entre le client et le serveur à l’aide du protocole http

pour HyperText Transfer Protocol, désigne dans le langage informatique un protocole de communication entre un client et un serveur pour le World Wide Web. On le traduit littéralement en français par protocole de transfert hypertexte. Le protocole http établit une liaison entre un ordinateur (client) et un serveur Web. Le premier, via un navigateur web (ou autre client http), envoie une requête au second qui lui apporte une réponse presque instantanée. Autrement dit, le protocole de communication http est ce qui permet à un internaute d’accéder à un contenu (une page web, un fichier CSS, . . . etc.), en commandant au serveur d’effectuer une action [15].

#### les méthodes de HTTP :

http définit un ensemble de méthodes de requête qui indiquent l’action que l’on souhaite réaliser sur la ressource indiquée. Bien qu’on rencontre également des noms (en anglais), ces méthodes sont souvent appelées verbes http. Chacun d’eux implémente une sémantique différente mais certaines fonctionnalités courantes peuvent être partagées par différentes méthodes (exp : une méthode de requête peut être sûre (safe), idempotente ou être mise en cache (cacheable)) [22].

Nous listons ci-dessous des verbes http qu’on peut utiliser pour les requêtes http.

— **GET** : La méthode GET demande une représentation de la ressource spécifiée.

Les requêtes GET doivent uniquement être utilisées afin de récupérer des données.

- **HEAD** : La méthode HEAD demande une réponse identique à une requête GET pour laquelle on aura omis le corps de la réponse (on a uniquement l'entête).
- **POST** : La méthode POST est utilisée pour envoyer une entité vers la ressource indiquée. Cela entraîne généralement un changement d'état ou des effets de bord sur le serveur.
- **PUT** : La méthode PUT remplace toutes les représentations actuelles de la ressource visée par le contenu de la requête.
- **DELETE** : La méthode DELETE supprime la ressource indiquée.
- **CONNECT** : La méthode CONNECT établit un tunnel vers le serveur identifié par la ressource cible.
- **OPTIONS** : La méthode OPTIONS est utilisée pour décrire les options de communications avec la ressource visée.
- **TRACE** : La méthode TRACE réalise un message de test aller/retour en suivant le chemin de la ressource visée.
- **PATCH** : La méthode PATCH est utilisée pour appliquer des modifications partielles à une ressource.

Nous résumons la différences entre les méthodes cités dans les Tableaux 2.1 et 2.2.

Méthodes	Corps requête	Corps réponse	safe	idempotente	cache
GET	Optionnel	oui	oui	oui	oui
HEAD	non	non	oui	oui	oui
POST	oui	oui	non	non	oui
PUT	oui	oui	non	oui	non
DELETE	non	oui	non	oui	non
CONNECT	oui	oui	non	non	non
OPTIONS	Optionnel	oui	oui	oui	non
TRACE	non	oui	oui	oui	non
PATCH	oui	oui	non	non	non

TABLE 2.1 – Différence entre les différentes méthodes HTTP

Méthodes	Descriptions
GET	Permet de demander un document.
HEAD	Permet de ne recevoir que les lignes d'entête de la réponse, sans le corps de document.
POST	Permet de transmettre des données (d'un formulaire par exemple) à l'URL spécifié dans la requête. l'URL désigne en général un script perl, PHP ... etc.
PUT	Permet de transmettre au serveur un document à enregistrer a l'URL spécifié dans la requête.
DELETE	Permet d'effacer la ressource spécifiée.
CONNECT	Permet de se connecter à un proxy ayant la possibilité d'effectuer du tunneling.
OPTIONS	Permet au client de connaître les options du serveur utilisables pour obtenir une ressource.
TRACE	Permet de signaler au serveur qu'il doit renvoyer la requête telle qu'il la reçue.
PATCH	permet de faire une mise a jour partielle d'une ressource.

TABLE 2.2 – Les méthodes verbes HTTP

### les codes de HTTP :

Le protocole de communication informatique HTTP définit une liste de codes. Ces codes sont des numéros de 3 chiffres. La plupart des codes correspondent chacun à un type d'erreur, quelques-uns correspondent à un type de succès. Ces codes permettent aux logiciels client HTTP de déterminer automatiquement si une requête a réussi, et sinon de connaître le type d'erreur.

Tous les codes d'état de réponse HTTP sont séparés en cinq classes ou catégories. Le premier chiffre du code d'état définit la classe de réponse (informations, succès, redirection, erreur client et erreur serveur), tandis que les deux derniers chiffres

n'ont aucun rôle de classification ou de catégorisation. Il existe cinq classes définies par la norme [18] :

1. 1xx.Réponse d'informations (100 - 199) : La demande a été reçue, poursuite du processus.
2. 2xx.Réponse de Succès (200 - 299) : La demande a été reçue, comprise et acceptée avec succès.
3. 3xx.Réponse de Re-direction (300 - 399) : D'autres mesures doivent être prises afin de compléter la demande.
4. 4xx.Réponse d'Erreur de client http (400 - 499) : La demande contient une mauvaise syntaxe ou ne peut pas être satisfaite.
5. 5xx.Réponse d'Erreur de serveur (500 - 599) : Le serveur n'a pas réussi à répondre à une demande apparemment valide.

#### **les codes les plus courants :**

Ci-dessous la liste des codes les plus utilisés dans les applications et services web actuellement.

- 200 : succès de la requête.
- 301 et 302 : Re-direction, respectivement permanente et temporaire.
- 401 : utilisateurs non authentifié.
- 403 : accès refusé.
- 404 : ressource non trouvée.
- 500 et 503 : erreur de serveur.
- 504 : le serveur n'a pas répondu.

Certains autres codes ne sont pas encore utilisés, mais sont prévus pour une utilisation future. D'autres codes n'entraînent aucun affichage spécifique pour l'utilisateur, mais sont sous-entendus (par exemple, les codes 200 ou 304, jamais vus par le client car ils concernent la majorité des requêtes réussies).

#### **Liste des codes d'état :**

Nous listons dans dans les tables ci-dessous, tous les codes d'état appartenant à chaque classe. Les codes de la table ci-dessous marqués WebDAV ne sont pas spécifiés par le protocole HTTP mais par le protocole WebDAV qui est une extension de HTTP.

Code	Message	signification
100	Continue	Attente de la suite de la requête.
101	Switching Protocols	Acceptation du changement de protocole.
102	Processing	Traitement en cours (évite que le client dépasse le temps d'attente limite).
103	Early Hints	(Expérimental) Dans l'attente de la réponse définitive, le serveur retourne des liens que le client peut commencer à télécharger.

TABLE 2.3 – Les codes de la classe d'information

Code	Message	Signification
200	OK	Requête traitée avec succès. La réponse dépendra de la méthode de requête utilisée.
201	Created	Requête traitée avec succès et création d'un document.
202	Accepted	Requête traitée, mais sans garantie de résultat.
203	Non-Authoritative Information	Information retournée, mais générée par une source non certifiée.
204	No Content	Requête traitée avec succès mais pas d'information à renvoyer.
205	Reset Content	Requête traitée avec succès, la page courante peut être effacée.
206	Partial Content	Une partie seulement de la ressource a été transmise.
207	Multi-Status	Reponse multiple.

208	Already Reported	Le document a été envoyé précédemment dans cette collection.
210	Content Different	La copie de la ressource côté client diffère de celle du serveur (contenu ou propriétés).
226	IM Used	Le serveur a accompli la requête pour la ressource, et la réponse est une représentation du résultat d'une ou plusieurs manipulations d'instances appliquées à l'instance actuelle.

TABLE 2.4 – Les codes de la classe succès

Code	Message	Signification
300	Multiple Choices	L'URI demandée se rapporte à plusieurs ressources.
301	Moved Permanently	Document déplacé de façon permanente.
302	Found	Document déplacé de façon temporaire.
303	See Other	La réponse à cette requête est ailleurs.
304	Not Modified	Document non modifié depuis la dernière requête.
305	Use Proxy (depuis HTTP/1.1)	La requête doit être ré-adressée au proxy.
306	Switch Proxy	Code utilisé par une ancienne version de la RFC 2616, à présent réservé. Elle signifiait « Les requêtes suivantes doivent utiliser le proxy spécifié ».

307	Temporary Redirect	La requête doit être redirigée temporairement vers l'URI spécifiée sans changement de méthode.
308	Permanent Redirect	La requête doit être redirigée définitivement vers l'URI spécifiée sans changement de méthode.
310	Too many Redirects	La requête doit être redirigée de trop nombreuses fois, ou est victime d'une boucle de redirection.

TABLE 2.5 – Les codes de la classe redirection

Code	Message	Signification
400	Bad Request	La syntaxe de la requête est erronée.
401	Unauthorized	Une authentification est nécessaire pour accéder à la ressource.
402	Payment Required	Paiement requis pour accéder à la ressource.
403	Forbidden	Le serveur a compris la requête, mais refuse de l'exécuter. Contrairement à l'erreur 401, s'authentifier ne fera aucune différence. Sur les serveurs où l'authentification est requise, cela signifie généralement que l'authentification a été acceptée mais que les droits d'accès ne permettent pas au client d'accéder à la ressource.
404	Not Found	Ressource non trouvée.
405	Method Not Allowed	Méthode de requête non autorisée.

406	Not Acceptable	La ressource demandée n'est pas disponible dans un format qui respecterait les en-têtes « Accept » de la requête.
407	Proxy Authentication Required	Accès à la ressource autorisé par identification avec le proxy.
408	Request Time-out	Temps d'attente d'une requête du client, écoulé côté serveur. D'après les spécifications HTTP : « Le client n'a pas produit de requête dans le délai que le serveur était prêt à attendre. Le client peut répéter la demande sans modifications à tout moment ultérieur »
409	Conflict	La requête ne peut être traitée en l'état actuel.
410	Gone	La ressource n'est plus disponible et aucune adresse de redirection n'est connue.
411	Length Required	La longueur de la requête n'a pas été précisée.
412	Precondition Failed	Préconditions envoyées par la requête non vérifiées.
413	Request Entity Too Large	Traitement abandonné dû à une requête trop importante.
414	Request-URI Too Long	URI trop longue.
415	Unsupported Media Type	Format de requête non supporté pour une méthode et une ressource données.
416	Requested range unsatisfiable	Champs d'en-tête de requête « range » incorrect.

417	Expectation failed	Comportement attendu et défini dans l'en-tête de la requête insatisfaisante.
418	I'm a teapot	« Je suis une théière » : Ce code est défini dans la RFC 2324. Certains sites web utilisent ce code de réponse pour les requêtes qu'ils ne souhaitent pas traiter (par exemple des requêtes automatiques).
421	Bad mapping / Misdirected Request	La requête a été envoyée à un serveur qui n'est pas capable de produire une réponse (par exemple, car une connexion a été réutilisée).
422	Unprocessable entity	L'entité fournie avec la requête est incompréhensible ou incomplète.
423	Locked	L'opération ne peut avoir lieu car la ressource est verrouillée.
424	Method failure	Une méthode de la transaction a échoué.
425	Too Early	Le serveur ne peut traiter la demande car elle risque d'être rejouée.
426	Upgrade Required	Le client devrait changer de protocole.
428	Precondition Required	La requête doit être conditionnelle.
429	Too Many Requests	Le client a émis trop de requêtes dans un délai donné.
431	Request Header Fields Too Large	Les entêtes HTTP émises dépassent la taille maximale admise par le serveur.
449	Retry With	Code défini par Microsoft. La requête devrait être renvoyée après avoir effectué une action.

450	Blocked by Windows Parental Controls	Code défini par Microsoft. Cette erreur est produite lorsque les outils de contrôle parental de Windows sont activés et bloquent l'accès à la page.
451	Unavailable For Legal Reasons	Ce code d'erreur indique que la ressource demandée est inaccessible pour des raisons d'ordre légal
456	Unrecoverable Error	Erreur irrécupérable.

TABLE 2.6 – Les codes de la classe "erreur de client"

Code	Message	Signification
500	Internal Server Error	Erreur interne du serveur.
501	Not Implemented	Fonctionnalité réclamée non supportée par le serveur.
502	Bad Gateway ou Proxy Error	En agissant en tant que serveur proxy ou passerelle, le serveur a reçu une réponse invalide depuis le serveur distant.
503	Service Unavailable	Service temporairement indisponible ou en maintenance.
504	Gateway Time-out	Temps d'attente d'une réponse d'un serveur à un serveur intermédiaire écoulé.
505	HTTP Version not supported	Version HTTP non gérée par le serveur.
506	Variant Also Negotiates	Erreur de négociation.

507	Insufficient storage	Espace insuffisant pour modifier les propriétés ou construire la collection.
508	Loop detected	Boucle dans une mise en relation de ressources
509	Bandwidth Limit Exceeded	Utilisé par de nombreux serveurs pour indiquer un dépassement de quota.
510	Not extended	La requête ne respecte pas la politique d'accès aux ressources HTTP étendues.
511	Network authentication required	Le client doit s'authentifier pour accéder au réseau. Utilisé par les portails captifs pour rediriger les clients vers la page d'authentification.

TABLE 2.7 – Les codes de la classe "erreur de serveur"

## 2.5 API

Le terme API désigne l'acronyme Application Programming Interface. En français, on traduit cette expression par interface de programmation d'application. Une API est un ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications. Elle est parfois considérée comme un contrat entre un fournisseur et un utilisateur d'informations, qui permet de définir le contenu demandé au consommateur (l'appel) et le contenu demandé au producteur (la réponse), c'est à dire qu'il s'agit d'une interface codée par laquelle plusieurs applications qui ne sont pas sensées fonctionner ensembles peuvent interagir entre elles [1].

IL existe 2 types d'API :

1. Les API SOAP (Simple Object Access Protocol)
2. Les API REST (Representational State Transfer)

Pour standardiser l'échange des informations entre les API toujours plus nombreuses, il a fallu développer un protocole : le « Simple Object Access Protocol », plus connu sous le nom de SOAP. Les API conçues d'après le protocole SOAP utilisent le format XML pour leurs messages et reçoivent des requêtes via HTTP ou SMTP. SOAP a pour objectif de simplifier l'échange des informations entre les applications qui s'exécutent dans des environnements différents ou qui ont été écrites dans des langages différents. SOAP est un protocole d'échange d'informations dans une implémentation de services web en XML. REST est une autre tentative de normalisation, c'est un style d'architecture de services web qui opère comme une chaîne de communication entre différents systèmes et Internet. On utilise principalement le REST pour mettre en œuvre des API modernes [1].

#### **REST :**

Le protocole REST (REpresentational State Transfer) constitue un style architectural et un mode de communication fréquemment utilisé dans le développement de services Web. Le recours à REST est souvent privilégié par rapport au style SOAP, plus lourd, car REST ne consomme pas autant de bande passante, ce qui rend son utilisation plus pratique sur Internet. En effet, l'approche SOAP exige l'écriture ou l'utilisation d'un programme serveur fourni (destiné à servir des données) et d'un programme client (qui demande des données). Alliant architecture découplée et communications légères entre producteur et consommateur, REST est un mode apprécié de création d'interfaces de programmation d'applications (API) dans le Cloud, du type de celles que fournissent Amazon, Microsoft et Google. L'architecture REST implique la lecture d'une page Web désignée qui contient un fichier XML. Ce fichier décrit et contient le contenu souhaité. Les consommateurs peuvent accéder à l'interface dès qu'elle est définie dynamiquement. L'architecture REST, généralement exécutée sur le protocole HTTP, obéit à plusieurs contraintes [27] :

- Elle dissocie les consommateurs des producteurs.

- Elle est sans état.
- Elle peut tirer parti d'un cache.
- Elle utilise un système de couches.
- Son interface est uniforme.

REST est souvent utilisé dans les applications mobiles, les sites Web des réseaux sociaux, les outils composites (mashup) et les processus métier automatisés. Dans le style REST, les interactions entre clients et services sont améliorées par le recours à un nombre limité d'opérations (verbes). L'affectation aux ressources (noms) de leurs propres identifiants URI (Universal Resource Identifiers) uniques autorise une grande souplesse. Etant donné que chaque verbe possède une signification spéciale (GET, POST, PUT et DELETE), REST permet d'éviter toute ambiguïté. Les services Web qui utilisent l'architecture REST sont appelés API RESTful ou API REST.

#### **RESTFUL :**

les API sont RESTful tant qu'elles respectent les six contraintes de conception d'un système RESTful [1] :

- **Architecture client-serveur** : une architecture REST est composée de clients, de serveurs et de ressources et elle traite les requêtes via le protocole HTTP.
- **Serveur stateless** : le contenu du client n'est jamais stocké sur le serveur entre les requêtes. Les informations sur l'état de la session sont, quant à elles, stockées sur le client.
- **Mémoire cache** : la mise en mémoire cache permet de se passer de certaines interactions entre le client et le serveur.
- **Système à couches** : des couches supplémentaires peuvent assurer la médiation dans les interactions entre le client et le serveur. Ces couches peuvent remplir des fonctions supplémentaires, telles que l'équilibrage de charge, le partage des caches ou la sécurité.
- **Code à la demande (facultatif)** : un serveur peut étendre les fonctionnalités d'un client en lui transférant du code exécutable.

- **Interface uniforme** : cette contrainte est capitale pour la conception des API RESTful et couvre quatre aspects différents :
  - **Identification des ressources dans les requêtes** : les ressources sont identifiées dans les requêtes et sont séparées des représentations retournées au client.
  - **Manipulation des ressources par des représentations** : les clients reçoivent des fichiers qui représentent les ressources. Ces représentations doivent contenir suffisamment d'informations pour être modifiées ou supprimées.
  - **Messages autodescriptifs** : tous les messages renvoyés au client contiennent assez d'informations pour décrire la manière dont celui-ci doit traiter les informations.
  - **Hypermédia comme moteur du changement des états applicatifs** : après avoir accédé à une ressource, le client REST doit être en mesure de découvrir toutes les autres actions disponibles par des hyperliens.

Ces contraintes peuvent sembler difficiles à appliquer, mais dans les faits, elles le sont moins qu'un protocole. C'est pour cette raison que les API RESTful prennent progressivement le pas sur les API SOAP.

- Les API web utilisent en général le protocole HTTP pour leurs messages de requête et fournissent une définition de la structure des messages de réponse. Les messages de réponse se présentent la plupart du temps sous la forme d'un fichier XML ou JSON. Ces deux formats sont les plus courants, car les données qu'ils contiennent sont faciles à manipuler pour les autres applications.
- Une des principales fonctionnalités qu'on retrouve quand on utilise une API est l'exposition de données. Cela signifie que les contenus d'un site (pages, articles) sont accessibles grâce à des endpoints, dans un format de données structurées. Concrètement, en se rendant à une certaine URL vous aurez les données d'un site au format JSON. De nombreux sites font cela de façon native, parfois sans que vous le sachiez.

**JSON :**

JSON (JavaScript Objet Notation) est un langage léger d'échange de données textuelles. Pour les ordinateurs, ce format se génère et s'analyse facilement. Pour les humains, il est pratique à écrire et à lire grâce à une syntaxe simple et à une structure en arborescence. JSON permet de représenter des données structurées (comme XML par exemple).

Fondé sur un sous-ensemble de Javascript, JSON est un format texte totalement indépendant de tout langage. Pourtant, les conventions utilisées ne surprendront pas les codeurs familiers aux langages descendant du C tels que Javascript, Python, Pearl ou d'autres. Autrement dit, il fonctionne un peu comme le XML (mais en moins développé) et facilite la structuration des informations présentes dans un document informatique. Comme il sert simplement à fluidifier l'échange de données, il n'est pas supposé contenir de commentaires, par exemple, ce qui le distingue d'un langage informatique à part entière. Toutefois, certaines bibliothèques en acceptent, s'ils sont écrits en JavaScript [17].

## 2.6 Services web et IA

De nos jours, fournir une fonctionnalité d'intelligence artificielle via un service est une idée déjà existante. Plusieurs plateformes en ligne fournissent des services web très intelligents et intéressants. Nous pouvons lister par exemple les services de google comme "google translate API [5]" qui comme son nom l'indique, peut recevoir des textes dans une langue, et retourne un résultat traduit dans une autre. Toujours fourni par google, le service "AI Natural Language [4]" permet d'analyser un texte écrit en langage naturel, et en extraire la sémantique, l'émotion, les entités ... etc. Malheureusement, les services de google sont payants et l'accès même à une version d'évaluation nécessite de fournir les coordonnées d'une carte bancaire, ce qui nous a empêcher de les tester.

D'autres géants de l'industrie digitale comme microsoft et amazon proposent également des service web fournissant des fonctionnalités d'intelligence artificielle via leur plateformes "Microsoft Azure [2]" et "AWS AI Services [30]" respectivement. D'autres petits acteurs (comparativement) offre également des services web dans le même contexte, nous citons ici OpenAI [23] et DeepAI [3].

## 2.7 Conclusion

Dans ce deuxième chapitre, nous avons présenté quelques notions et des concepts pour les services web. Nous avons eu à parler de les étapes de fonctionnement d'un service web, la citation de ses protocoles avec ses définitions et ses descriptions, les domaines de la reconnaissance, et ses méthodes. A la fin nous avons parler de L'API.

## CHAPITRE 3

# ARCHITECTURE DU SYSTÈME

### 3.1 Introduction

L'analyse et la conception sont les étapes les plus importantes de la mise en œuvre n'importe quel projet. Ce chapitre est consacré à la représentation de la structure globale et l'architecture détaillée de notre service web et son utilisation opérationnelle à l'aide des diagrammes UML (diagramme cas d'utilisation , diagramme de classe et de séquence).

### 3.2 Architecture du Système

Comme le montre la figure 3.1, notre service web est l'intermédiaire entre l'utilisateur (souvent un développeur d'applications) et un modèle de reconnaissance de visage et d'émotion et une base de données relationnelle. Le service web reçoit une image de la part de l'utilisateur, puis utilise le modèle pour le reconnaissance, et la base de données pour le stockage des résultats et des informations de chaque utilisateur. Les interactions entre les différents composants et acteurs de notre système

seront expliquées avec plus de détails prochainement en utilisant des diagrammes de séquence.

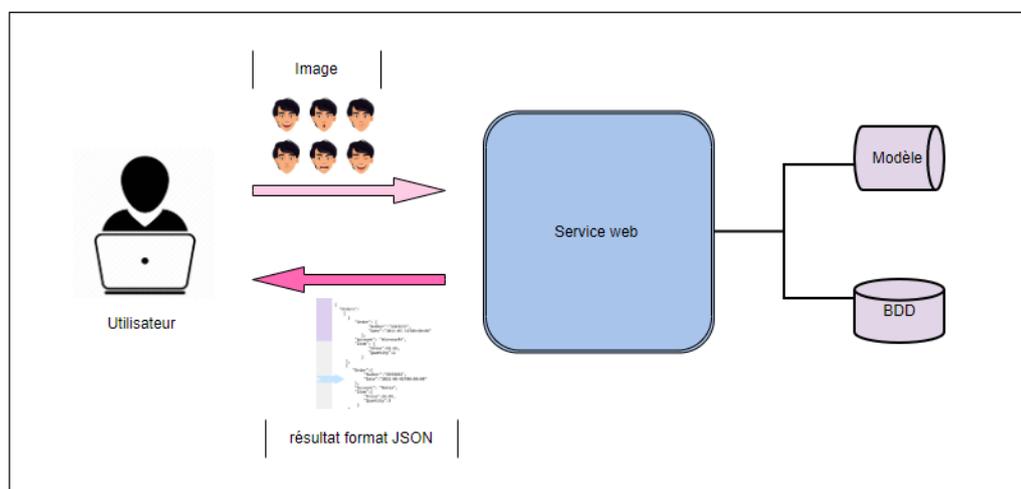


FIGURE 3.1 – l'architecture de service web pour la détection d'émotions

## 3.3 Méthodologie de Conception

### 3.3.1 Le langage UML

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié), est un langage visuel constitué d'un ensemble de schémas, appelés diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel ... etc [10].

### 3.3.2 Les Différents diagrammes UML

UML est un Langage constitué d'un ensemble de diagrammes composés par des vues [33] :

**Vue statique**

- Diagramme de cas d'utilisation.
- Diagramme de classe.
- Diagramme de déploiement.
- Diagramme d'architecture.
- Diagramme de composants.
- Diagramme d'objets.

**Vue dynamique**

- Diagramme de séquence.
- Diagramme d'états-transitions.
- Diagramme d'activités.
- Diagramme de collaborations.

Nous résumons dans la figure 3.2 la répartition des différents diagrammes UML. Dans la suite de ce chapitre, nous allons utiliser le diagramme de classes pour montrer les différentes classes de notre système ainsi que les relations entre elles, et le diagramme de séquence pour expliquer les scénarios d'exécution des fonctionnalités les plus importantes de notre système.

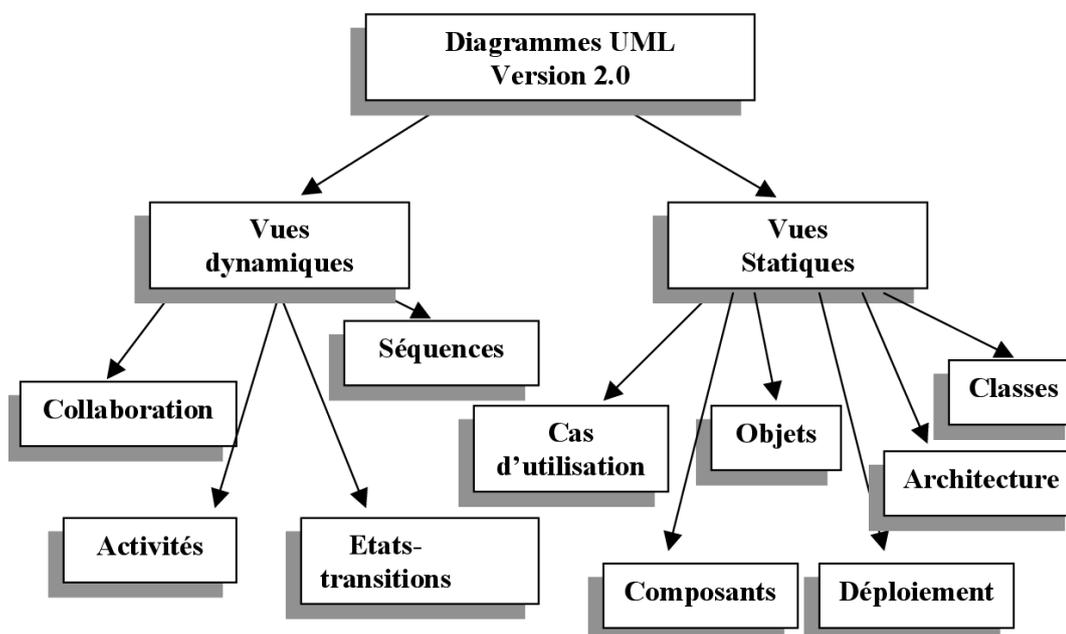


FIGURE 3.2 – Les différents diagrammes du langage UML

## 3.4 Conception

### 3.4.1 Identification des classes et fonctionnalités

Nous utilisons seulement deux classes :

- Classe Utilisateurs.
- classe Images.

**Chaque utilisateur peut avoir plusieurs scénarios d'utilisations :**

- inscrivez ou connectez au service service web.
- Envoyez une image et recevez une réponse sous format JSON, contient les pixels qui constituent chaque visage si l'image contient plusieurs visages , et la probabilité de Chaque émotion et l'émotion dominante. L'utilisateur reçoit aussi Un lien pour l'image originale qu'il l'a envoyée, et un autre lien pour l'image générée qu'elles sont toujours stockée sur nos service , (des exemples seront retrouvées dans le prochain chapitre). Cette fonctionnalité est disponible

aux utilisateurs enregistrés ainsi qu'aux utilisateurs non enregistrés. Cependant Il est fortement conseillé aux utilisateurs de s'inscrire pour avoir accès leur historique de transactions.

- Consulter l'historique des opérations, cette fonctionnalité est naturellement disponible seulement pour les utilisateurs inscrits sur notre base de données. A l'inscription chaque utilisateur reçoit un code d'accès unique, ce code lui permet par la suite de consulter son historique via un "endpoint" dans notre service web. L'historique expose les même informations qui ont été communiquées à l'envoi initial de chaque requête (image, zone de délimitation, émotion ... etc).

### 3.4.2 Représentation de diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel [8].

La figure 3.3 le diagramme de cas d'utilisation de chaque acteur (utilisateur inscrit, utilisateur non inscrit).

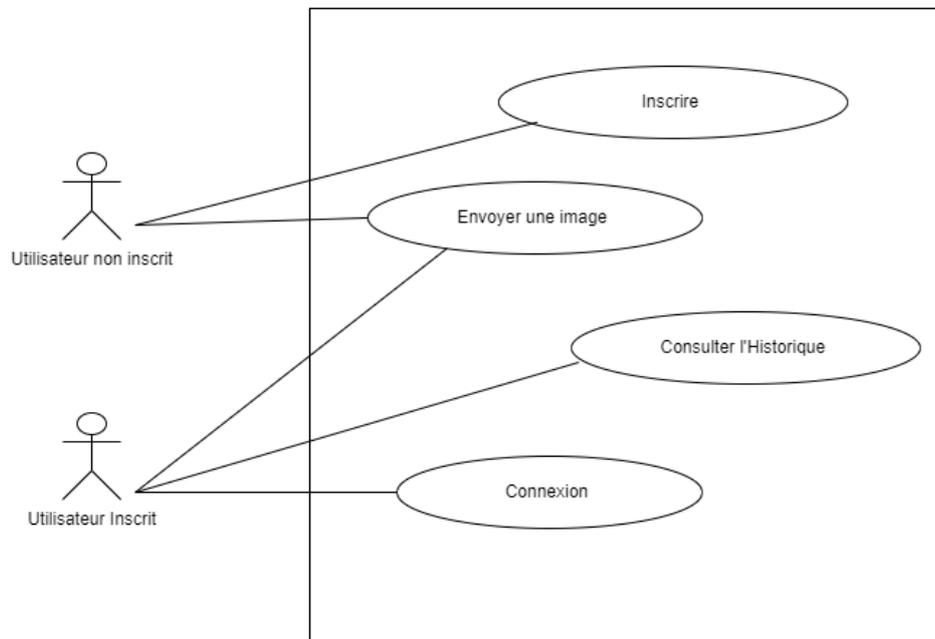


FIGURE 3.3 – Diagramme de cas d'utilisation

### Description textuelle cas d'utilisation

#### Utilisateur non inscrit

Cas no 1
<b>Nom</b> : S'Inscrire .
<b>Acteur (s)</b> : Utilisateur non inscrit.
<b>Description</b> : Cette fonction contient une seule opération (remplir le formulaire).
<b>Pré-condition</b> : Au début l'utilisateur non inscrit doit accéder a l'interface.

Cas no 2
<b>Nom</b> : Envoyer une image.
<b>Acteur (s)</b> : Utilisateur non inscrit.
<b>Description</b> : L'utilisateur envoie une image en utilisant n'importe quel client HTTP.
<b>Pré-condition</b> : Avoir un client HTTP.

### Utilisateur inscrit

Cas no 1
<b>Nom</b> : Connexion .
<b>Acteur (s)</b> : Utilisateur inscrit.
<b>Description</b> : Cette fonction contient une seule opération (remplir le formulaire de connexion) .
<b>Pré-condition</b> : Au début l'utilisateur avoir un compte.

Cas no 2
<b>Nom</b> : Envoyer une image.
<b>Acteur (s)</b> : Utilisateur inscrit.
<b>Description</b> : L'utilisateur envoie une image en utilisant n'importe quel client HTTP.
<b>Pré-condition</b> : L'utilisateur doit avoir un code d'identification (sinon il sera considéré comme non-inscrit).

Cas no 3
<b>Nom</b> : Consulter l'historique .
<b>Acteur (s)</b> : Utilisateur inscrit.
<b>Description</b> : Cette fonction contient une seule opération envoyer une requête au bon url avec le code d'identification comme paramètre.
<b>Pré-condition</b> : L'utilisateur doit avoir un code d'identification.

### 3.4.3 Représentation de diagramme de classe

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objet [9].

Nous montrons dans la figure 3.4 le diagramme de classe de notre système

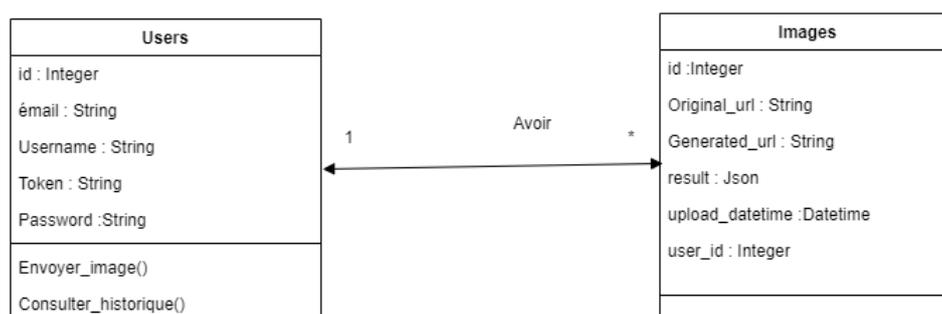


FIGURE 3.4 – Diagramme de classe

**Description textuelle des classes****Classe Utilisateur**

Attribut	Type	Description
id	Integer	l'identificateur de l'utilisateur.
email	String[150]	l'email de l'utilisateur utilisé pour la création du compte .
username	String[150]	nom d'utilisateur utilisé pour la création du compte, il peut aussi être utilisé pour se connecter à via l'interface web.
password	String[150]	Le mot de passe de l'utilisateur.
token	String[150]	Code unique associé à l'utilisateur lui permettant de s'identifier lorsqu'il utilise le service web.

TABLE 3.1 – description textuelle de classe "Utilisateur"

**Classe Image**

Attribut	Type	Description
id	Integer	l'identificateur de l'image.
original_url	String[150]	Un url vers l'image original enregistré sur nos serveurs.
generated_url	String[150]	Un url vers une image générée qui affiche zone de délimitation avec l'émotion dominante pour chaque visage.
result	Text	Le résultat qui a été renvoyé à l'utilisateur après la réception de l'image, ce résultat est gardé dans l'historique pour permettre de consulter les anciennes images sans les envoyer à nouveau.
upload_datetime	datetime	Date et temps de la requête envoyée.
user_id	Integer	Une clé étrangère contenant l'identifiant de utilisateur qui a envoyé cette image.

TABLE 3.2 – description textuelle de classe "Images"

### 3.4.4 Diagrammes de Séquences

Un diagramme de séquence montre comment un ensemble d'objets interagissent dans un processus dans le temps. Il affiche les messages qui passent entre des participants et des objets dans le système, ainsi que l'ordre dans lequel ils se produisent [7].

Dans cette partie nous allons voir des diagrammes de séquences des différents cas tels que : la création d'un compte, envoyer une image, consulter l'historique.

### La création d'un compte

Ce Diagramme de séquence représente le scénario de création d'un nouveau compte utilisateur, pour que l'utilisateur puisse se connecter au service, il doit remplir un formulaire avec son email, nom d'utilisateur, mot de passe puis le système crée son compte et lui attribue un code d'accès lui permettant de s'authentifier lorsqu'il utilise le service web. La figure 3.5 représente le diagramme de séquence pour la création d'un compte

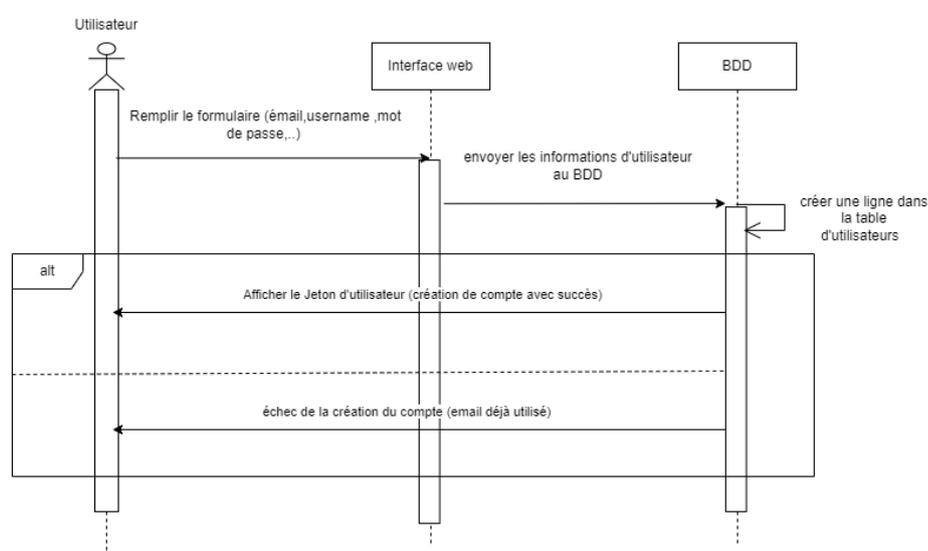


FIGURE 3.5 – Diagramme de séquence pour la création d'un compte

### Envoyer une image

Ce diagramme représente le scénario de l'envoi d'une requête contenant une image au service pour la traiter. Noter que cette requête peut être envoyée de n'importe quel application en utilisant n'importe quel langage de programmation. Notre service web va traiter l'image après il retourne le résultat sous le format JSON qui comporte les zone de délimitation de chaque visage ainsi que l'émotion dominante. La figure 3.6 représente le diagramme de séquence pour l'envoi d'une image

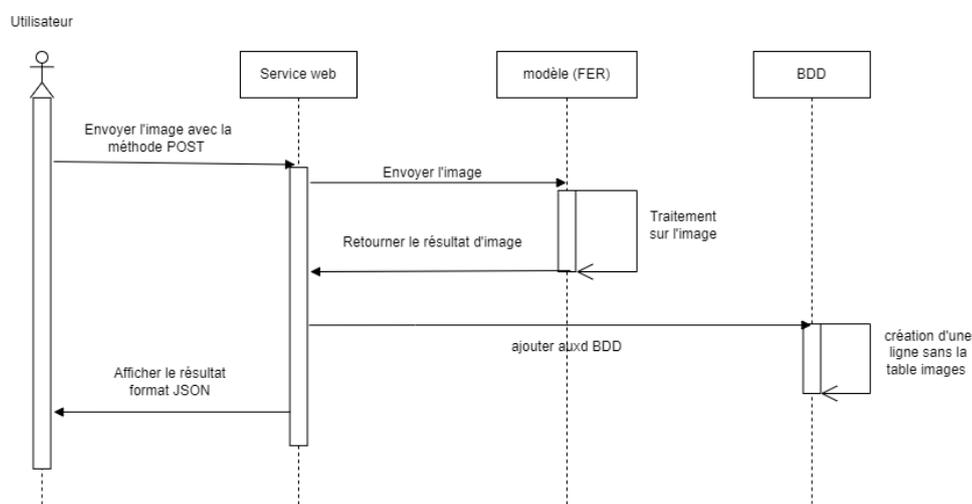


FIGURE 3.6 – Diagramme de séquence pour l'envoi d'une requête

### Consulter l'historique

Ce diagramme montre les interactions effectuées par l'utilisateur avec notre système pour consulter l'historique de ses actions. Chaque compte utilisateur a le droit de consulter seulement les résultats des requêtes qu'il a envoyées. L'utilisateur doit connaître son token (affiché sur l'interface web lorsqu'il se connecte à son compte), car c'est à travers ce dernier que notre système peut l'identifier, et lui renvoyer ses anciens résultats. La figure 3.7 représente le diagramme de séquence pour la gestion d'historiques.

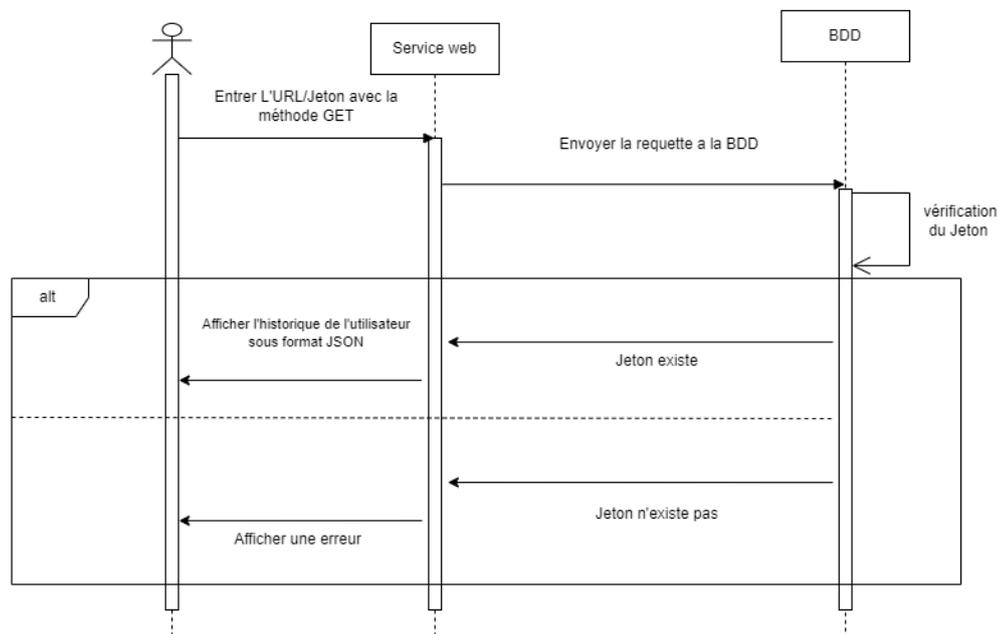


FIGURE 3.7 – Diagramme de séquence pour la gestion d'historiques

### 3.5 Conclusion

Dans ce projet, nous avons décrit les spécification de notre projet, puis nous avons détaillé l'architecture de ce dernier en utilisant des schémas et des diagrammes UML. Nous avons utilisé le diagramme de classe pour spécifier les classes et les relations entre elles, ce qui nous permet d'avoir une idée très claire sur la structure de la bases de données à utiliser. Et nous avons utilisé aussi le diagramme de séquence pour expliquer le déroulement des fonctionnalités les plus importantes de notre système. Dans le prochain chapitre nous allons passer à l'implémentation de ce projet.

## CHAPITRE 4

# RÉSULTAT ET DISCUSSIONS

### 4.1 Introduction

L'objectif de la phase d'implémentation ou bien la phase des résultats et discussions est d'aboutir à un produit final, exploitable par les utilisateurs de notre service web. Nous allons consacrer ce chapitre à la réalisation et la mise en oeuvre de notre projet pour la détection des émotions a partir des images.

Dans cette partie, nous allons présenter premièrement les outils utilisés pour le développement de notre service, par la suite nous allons montrer un petit manuel d'utilisation de notre application, avec des exemples de consommation de notre service en utilisant quelques autres langages de programmation.

### 4.2 Outils de Développement

Dans cette partie, nous allons présenter les langages et les bibliothèques utilisés pour le développement de notre projet :

### 4.2.1 Python

Avec la complexification du web et l'accumulation des données, Python a pris une place majeure dans le développement informatique car il peut être utilisé dans des situations variées, dans tous les secteurs d'activité. 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. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages. Techniquement, ce langage servira surtout pour le scripting et l'automatisation (interaction avec les navigateurs web) [37].

### 4.2.2 FastAPI

C'est un framework Web moderne et rapide (haute performance) pour la création d'API avec Python 3.6+. FastAPI est un framework très performant, intuitif et très facile à apprendre et utiliser. FastAPI repose principalement sur deux bibliothèques pythons qui sont très largement utilisées, Starlette et Pydantic, mais aucune connaissances préalable sur ces deux derniers n'est nécessaire pour utiliser FastAPI [37].

### 4.2.3 Bibliothèque FER (Facial Emotion Recognition)

La bibliothèque de reconnaissance des expressions faciales est développée par Justin Shenk. Cette bibliothèque nécessite l'installation des dépendances : **OpenCV**, **MTCNN** et **Tensorflow**. Les visages sont détectés à l'aide du classificateur Haar Cascade d'OpenCV ou un réseau de neurones convolutifs implémenté dans MTCNN selon les paramètres données par le programmeur [11].

#### 4.2.4 MTCNN

C'est une bibliothèque développée comme solution pour la détection et l'orientation du visage [21]. Elle est disponible sous forme de package "pip" et peut être installée très facilement. Dans le cadre de notre travail, MTCNN est utilisé en transparence par la bibliothèque FER pour détecter les visages avant de lancer la tâche de reconnaissance d'émotion.

#### 4.2.5 OpenCV (Open Computer Vision)

Est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société de robotique Willow Garage assure le support de cette bibliothèque depuis 2010. Cette bibliothèque est distribuée sous licence BSD. La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques, parmi ces fonctionnalités [14] :

- Traitement d'images.
- Traitement de vidéos.
- Calculs matriciels.

Ce qui nous intéresse dans notre projet c'est la partie de traitement d'images, Elle propose la plupart des opérations classiques en traitement bas niveau des images comme lecture, écriture, affichage, binarisation d'une image . . . etc.

#### 4.2.6 TensorFlow

TensorFlow est une plate-forme Open Source de bout en bout dédiée au machine learning. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine du machine learning, et aux développeurs de créer et de déployer facilement des applications qui exploitent cette technologie [31].

### 4.2.7 Wampserver

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données. Dans notre cas, nous allons utiliser MySQL pour stocker les données de notre bases de données relationnelle, nous avons également utilisé PHP pour créer un petite application qui illustre le fonctionnement de notre service web, mais ce n'est pas la partie essentielle de notre travail [36].

### 4.2.8 Uvicorn

Uvicorn est une implémentation de serveur Web ASGI (Asynchronous Server Gateway Interface) pour Python. La spécification ASGI signifie que nous sommes désormais en mesure de créer un ensemble commun d'outils utilisables dans tous les frameworks asynchrones. Uvicorn prend actuellement en charge HTTP et WebSockets [34]

### 4.2.9 Postman

Postman est une application permettant de tester des API. Postman simplifie chaque étape du cycle de vie des API et rationalise la collaboration afin que nous puissions créer des API plus rapidement. Il sert à exécuter des appels HTTP directement depuis une interface graphique [25].

## 4.3 Présentation de l'application

Cette section est consacré à la présentation de notre application, nous allons d'abord commencer par la présentation des interfaces web, puis nous montrons comment utiliser notre service à en donant des exemples avec postman.

### 4.3.1 Page d'inscription

L'interface web de notre application est consacré seulement à l'inscription des utilisateurs. Il suffit de remplir le formulaire d'inscription visible sur la figure 4.1 pour créer un compte.



FIGURE 4.1 – Page d'inscription

### 4.3.2 Connexion

Sur la même page d'accueil, les utilisateurs inscrits ont le choix de se connecter à leur comptes existants pour consulter le code code d'identification leur permettant d'accéder à toutes les fonctionnalités du service web. Nous montrons le formulaire de connexion dans la figure 4.2.

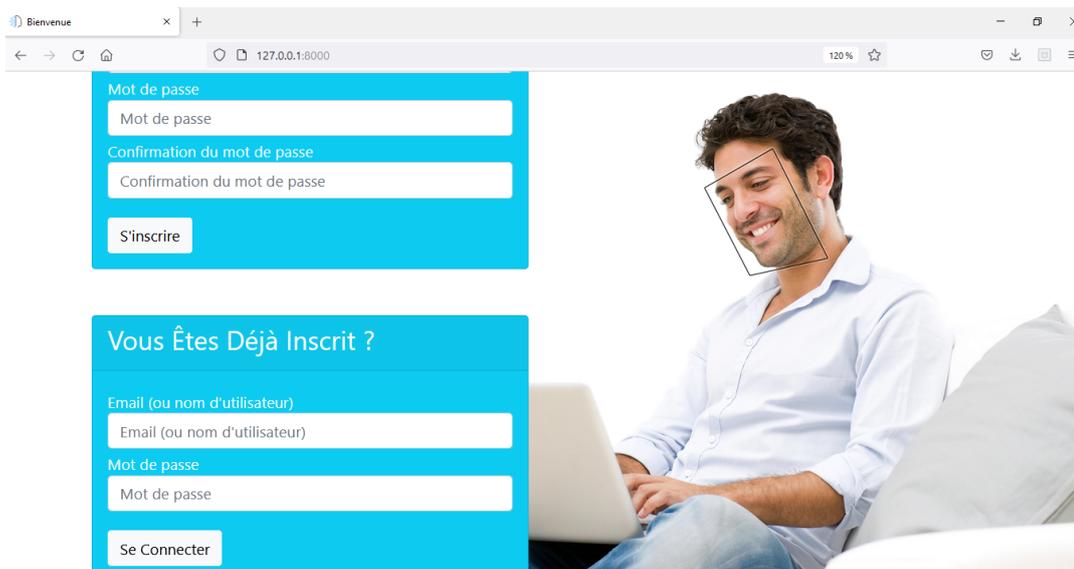


FIGURE 4.2 – Connexion

### 4.3.3 Tester notre API

Nous allons montrer dans cette section comment nous avons utilisé Postman pour tester le bon fonctionnement de notre service web, ce qui permet de montrer clairement quelles sont les entrées et les sorties de notre API.

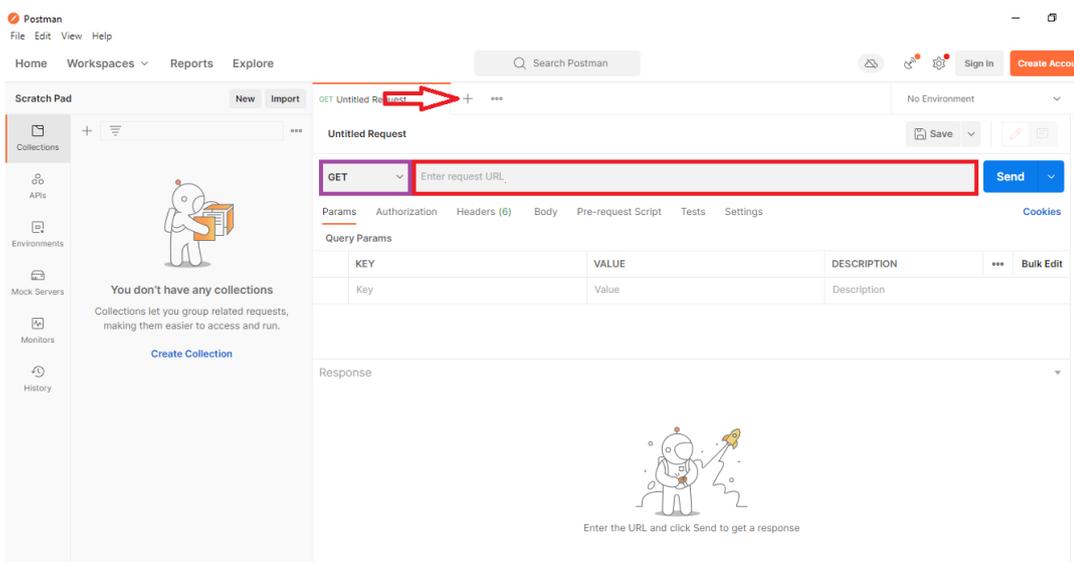


FIGURE 4.3 – Interface Graphique de Postman

La figure 4.3 montre l'interface de Postman, comme sur un navigateur web, pour tester un service, il faut mettre utiliser son url, mais en plus, nous pouvons spécifier la méthode utilisée POST ou GET dans notre cas, alors que les navigateurs n'étant pas destinés aux développeurs, utilisent toujours la méthode GET par défaut.

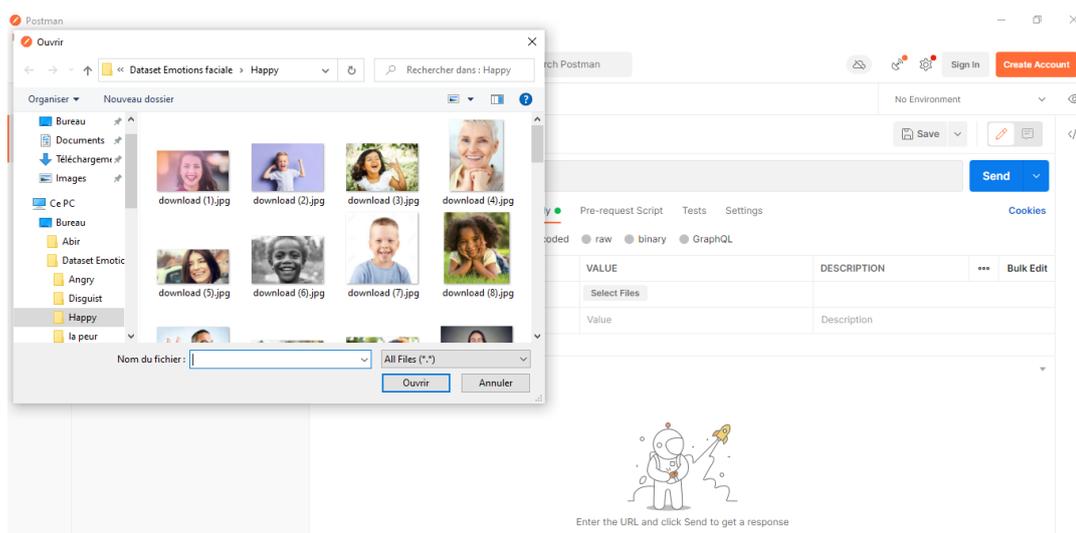


FIGURE 4.4 – Remplir un Formulaire dans Postman

Nous avons aussi la liberté de spécifier les données à envoyer dans le formulaire avec le nom et la valeur de chaque champs. La figure 4.4 montre comment importer une image dans le formulaire de Postman.

Finalement, après l'envoi de la requête, Postman reçoit la réponse, et il nous l'affiche comme le montre la figure 4.5. Il est possible de voir le statut de la réponse, les erreurs données par le serveur (en cas d'erreur) ou le résultat dans cet exemple lorsque tout marche bien.

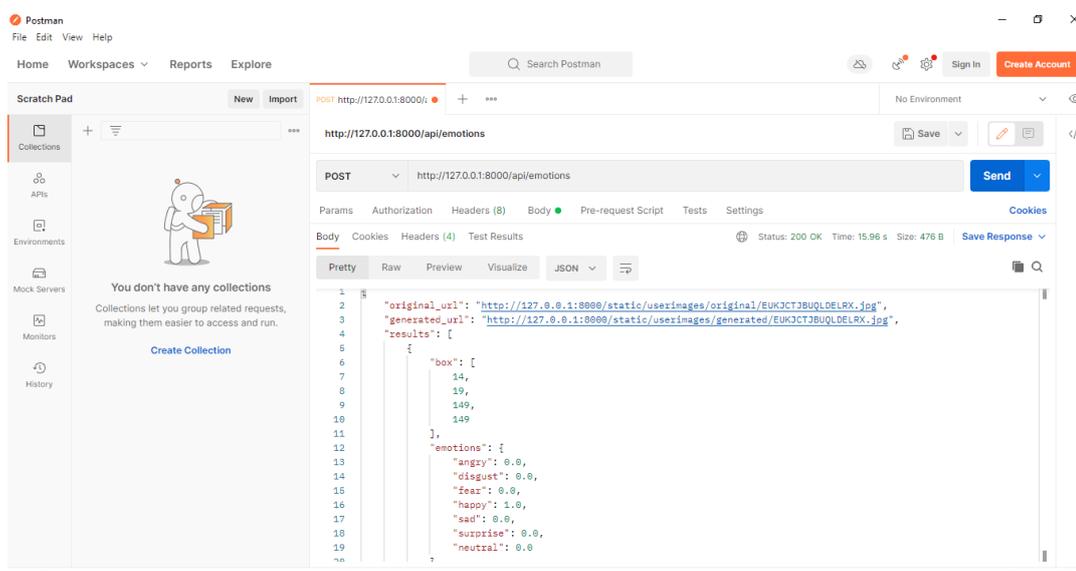


FIGURE 4.5 – Résultat affiché par Postman

Nous rappelons encore une fois que Postman ne sert qu'à tester le bon fonctionnement de notre API, en pratique, un service web est destiné à être utilisé par un développeur qui pourrait envoyer des requêtes via un client HTTP en utilisant n'importe quel langage de programmation. Nous montrons plus d'exemples d'utilisation de notre API dans la section suivante.

## 4.4 Exemples d'utilisation

Nous allons dans cette section montrer des exemples d'utilisation de notre service web à partir de deux petites applications en utilisant les langages PHP et Javascript. Nous illustrons ici des exemples simples seulement pour montrer que notre service marche correctement, plusieurs autres scénarios d'utilisation plus avancés peuvent avoir lieu en pratique.

### 4.4.1 Premier exemple (PHP)

Nous avons écrit un script PHP très simple qui reçoit un formulaire contenant une image, puis envoie l'image reçue à notre service web puis reçoit le résultat et affiche

l'image générée par le service (avec les visages détectés et les zone de délimitation bien visibles pour chaque visage).

```
28 $args['file'] = new CurlFile(__DIR__ . DIRECTORY_SEPARATOR . $newFileName);
29 $results = CallAPI(
30     'POST',
31     'http://127.0.0.1:8000/api/emotions/SIMABZXNBNDURNCYMSQMXHPZUQFEYLSZ',
32     $args
33 );
34 $results = json_decode($results, true);
35 echo "<img style='width: 100%' src='" . $results["generated_url"] . "' class='img-fluid">
```

FIGURE 4.6 – Code PHP pour utiliser notre service web

Dans la figure 4.6 nous montrons le code que nous avons écrit en PHP pour faire appelle à notre service web et recevoir et exploiter le résultat. A la ligne "29" du script nous faisons appelle au service web et obtenont le résultat. La ligne "35" convertit le résultat en un tableau PHP, et la dernière ligne affiche l'image à l'url dans la variable "generated\_url". Le résultat d'exécution de ce script est montré dans la figure 4.7.

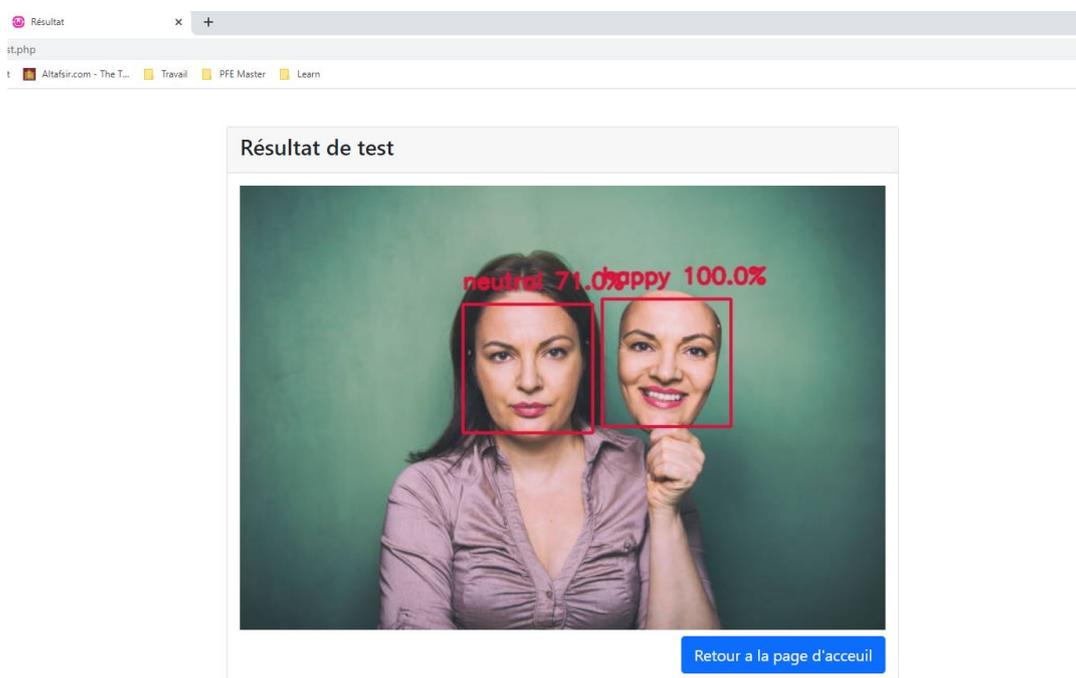


FIGURE 4.7 – Résultat du code PHP

### 4.4.2 Deuxième exemple (Javascript)

Dans Javascript, plusieurs clients HTTP très puissants et faciles à utiliser sont disponibles. En plus, les résultats JSON sont convertis automatiquement en objets dans Javascript. Le code dans l'image 4.8 permet similairement l'exemple précédent d'envoyer une image à notre service web, puis recevoir et afficher le résultat.

```
JS script.js > sendForm
1  function sendForm() {
2    fetch('http://127.0.0.1:8000/api/emotions/SIMABZXNBNDURNCYMSQMXHPZUQFEYLSZ', {
3      method: 'POST',
4      ContentType: 'multipart/form-data',
5      body: new FormData(document.getElementById('form'))
6    })
7    .then(response => response.json())
8    .then(data => {
9      document.getElementById('result').src = data.generated_url;
10     $('#resultModal').modal('show');
11   })
12 }
```

FIGURE 4.8 – Code Javascript pour utiliser notre service web

## 4.5 Conclusion

Dans ce dernier chapitre, nous nous sommes intéressés aux détails techniques et l'implémentation de notre service web. Nous avons commencé par la présentation des différents outils utilisés, puis nous avons montré l'interface web et l'API qui et comment on peut les utiliser, et finalement, nous avons écrit des petits scripts en langages de programmations différents pour illustrer comment notre service doit être utilisé en pratique. Ce service pourrait être utilisé par les différentes applications et développeurs dans le besoin d'utiliser l'émotion d'un utilisateur, patient ou client afin de personnaliser l'expérience de ce dernier, ce qui va engendrer un gain de temps énorme, surtout sur des frameworks ou langages de programmation qui ne sont pas conçus à la base pour gérer des tâches d'intelligence artificielle et traitement d'image.

## CONCLUSION GÉNÉRALE

Tout au long de la préparation de notre projet de fin d'études, nous avons essayé de mettre en pratique les connaissances acquises durant nos études universitaires et cela dans le but de réaliser un service web pour la reconnaissance de l'émotion faciale. Notre application est destinée principalement à être utilisée par un développeur qui pourrait envoyer des images via un client HTTP en utilisant n'importe quel langage de programmation et recevoir en réponse toutes les informations nécessaires sur les visages présents dans l'image ainsi que les émotions exprimées.

Au cours de ce mémoire, nous avons étudié les différentes solutions disponibles pour la reconnaissance de l'émotion faciale, ainsi que les outils techniques adaptés pour offrir cette fonctionnalité derrière un service web facile à utiliser. Nous avons choisi d'utiliser le langage python car c'est l'un des langages les plus utilisés dans le domaine de l'intelligence artificielle. Plusieurs bibliothèques sont disponibles pour la reconnaissance de l'émotion faciale, la bibliothèque "FER (Facial Emotion Recognition)" est celle que nous avons utilisée derrière un service web développé à l'aide du framework "FastAPI".

Nous avons montré de le quatrième chapitre de ce mémoire une petite démonstration technique du fonctionnement de notre projet, premièrement en utilisant l'outil "Postman" pour bien voir les échanges entre le client http et notre service. Par la suite nous avons montré des exemples d'utilisation d'un autre langage de programmation

pour consommer notre service. Ce travail à principalement pour but de faciliter l'accès à la fonctionnalité de reconnaissance de l'émotion faciale dans les différents domaines d'application, ce qui va engendrer un gain de temps et d'énergie très importants pour l'avancement des différents projets.

## BIBLIOGRAPHIE

- [1] *APIs*. Accessed : 20-04-2022. URL : <https://www.redhat.com/fr/topics/api/what-are-application-programming-interfaces/>.
- [2] *Azure Cognitive Services*. Accessed : 12-06-2022. URL : <https://azure.microsoft.com/en-gb/services/cognitive-services/>.
- [3] *Build with AI | DeepAI*. Accessed : 12-06-2022. URL : <https://deepai.org/apis>.
- [4] *Cloud Natural Language*. Accessed : 12-06-2022. URL : <https://cloud.google.com/natural-language>.
- [5] *Cloud Translation*. Accessed : 12-06-2022. URL : <https://cloud.google.com/translate>.
- [6] D DAVID et M SAMRAJ. « A comprehensive survey of emotion recognition system in facial expression ». In : *Artech J. Eff. Res. Eng. Technol* 1 (2020), p. 76-81.
- [7] *définition de cas de séquece*. Accessed : 01-05-2022. URL : <https://support.microsoft.com/>.
- [8] *définition diagramme de cas d'utilisationn*. Accessed : 01-05-2022. URL : <https://esi2cisuml.fandom.com/fr/wiki/Accueil>.
- [9] *définition diagramme de classe*. Accessed : 01-05-2022. URL : <https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml/>.

- 
- [10] *définition uml*. Accessed : 01-05-2022. URL : <https://pdfcoffee.com/uml-cest-quoi-object-2-pdf-free.html/>.
- [11] *fer*. PyPI. Accessed : 09-06-2022. URL : <https://pypi.org/project/fer/>.
- [12] *fonctionnalités de service web*. Accessed : 20-04-2022. URL : <https://www.oracle.com/fr/cloud/definition-web-service/>.
- [13] Hamza HAMDI. « Plate-forme multimodale pour la reconnaissance d'émotions via l'analyse de signaux physiologiques : application à la simulation d'entretiens d'embauche ». Thèse de doct. Université d'Angers, 2012.
- [14] *Home - OpenCV*. Accessed : 09-06-2022. URL : <https://opencv.org/>.
- [15] *HTTP*. Accessed : 20-04-2022. URL : <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203581-http-hypertext-transfert-protocol-definition-traduction/>.
- [16] 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).
- [17] *JSON*. Accessed : 20-04-2022. URL : <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445308-json-definition-et-presentation-de-ce-format-de-donnees/>.
- [18] *les codes de HTTP*. Accessed : 20-04-2022. URL : [https://fr.wikipedia.org/wiki/Liste\\_des\\_codes\\_HTTP/](https://fr.wikipedia.org/wiki/Liste_des_codes_HTTP/).
- [19] Qi-rong MAO et al. « Using Kinect for real-time emotion recognition via facial expressions ». In : *Frontiers of Information Technology & Electronic Engineering* 16.4 (2015), p. 272-282.
- [20] Manoj MOOLCHANDANI et al. « A survey on : Facial Emotion Recognition and Classification ». In : *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE. 2021, p. 1677-1686.

- [21] *Multi-task Cascaded Convolutional Networks (MTCNN) for Face Detection and Facial Landmark Alignment*. Accessed : 09-06-2022. URL : <https://medium.com/@iselagradilla94/multi-task-cascaded-convolutional-networks-mtcnn-for-face-detection-and-facial-landmark-alignment-7c21e8007923>.
- [22] *méthodes de HTTP*. Accessed : 20-04-2022. URL : <https://developer.mozilla.org/fr/docs/Web/HTTP/Methods/>.
- [23] *OpenAI API*. Accessed : 12-06-2022. URL : <https://openai.com/api/>.
- [24] Hassiba OUAKKAF et M BERKANE. « Reconnaissance automatique des expressions faciales par support vector machine ». In : (2017).
- [25] *Postman API Platform*. Accessed : 09-06-2022. URL : <https://www.postman.com/>.
- [26] Matthew S RATLIFF et Eric PATTERSON. « Emotion recognition using facial expressions with active appearance models ». In : *Proc. of HRI*. Citeseer. 2008.
- [27] *REST*. Accessed : 20-04-2022. URL : <https://www.lemagit.fr/definition/REST/>.
- [28] I Michael REVINA et WR Sam EMMANUEL. « Face expression recognition with the optimization based multi-SVNN classifier and the modified LDP features ». In : *Journal of Visual Communication and Image Representation* 62 (2019), p. 43-55.
- [29] Hamid SADEGHI et Abolghasem-A RAIE. « Human vision inspired feature extraction for facial expression recognition ». In : *Multimedia Tools and Applications* 78.21 (2019), p. 30335-30353.
- [30] *Services d'intelligence Artificielle*. Accessed : 12-06-2022. URL : <https://aws.amazon.com/fr/machine-learning/ai-services/>.
- [31] *TensorFlow*. Accessed : 09-06-2022. URL : <https://www.tensorflow.org/>.
- [32] Matthew TURK et Alex PENTLAND. « Eigenfaces for recognition ». In : *Journal of cognitive neuroscience* 3.1 (1991), p. 71-86.
- [33] *Types UML*. Accessed : 01-05-2022. URL : <http://uml.free.fr/cours/p9.html>.

- 
- [34] *Uvicorn*. Accessed : 09-06-2022. URL : <https://www.uvicorn.org/>.
- [35] Ramachandran VEDANTHAM, Lavanya SETTIPALLI et E Sreenivasa REDDY. « Real time facial expression recognition in video using nearest neighbor classifier ». In : *International Journal of Pure and Applied Mathematics* 118.9 (2018), p. 849-854.
- [36] *WAMPSEVER*, plate-forme de développement Web sous Windows. Accessed : 09-06-2022. URL : <https://www.wampserver.com/>.
- [37] *Welcome to Python.org*. Accessed : 09-06-2022. URL : <https://www.python.org/>.
- [38] *XML*. Accessed : 20-04-2022. URL : <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203615-xml-extensible-markup-language-definition-traduction//>.