



Faculté des Sciences Exactes et d'Informatique

Département de Mathématiques et informatique

Filière : Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Intelligence Artificielle pour l'internet des objets**

(IA4IoT)

Présenté par :

«**REHAMNIA Younes**»

«**BOUKSARA Meriem**»

THEME :

Reconnaissance des expressions faciales

avec SVM/CNN

Année Universitaire 2024-2025

Résumé

L'expression faciale est un ensemble de mouvements des muscles du visage reflétant l'état émotionnel d'un individu. La reconnaissance automatique de ces expressions constitue un domaine clé de l'intelligence artificielle, avec des applications variées telles que l'interaction homme-machine, la vidéosurveillance et les environnements intelligents.

De nombreuses recherches ont exploré l'extraction des caractéristiques ainsi que l'utilisation de classifieurs afin d'améliorer la précision des systèmes de reconnaissance faciale. Toutefois, les performances restent limitées, notamment lors de l'utilisation de bases de données volumineuses et complexes.

Dans ce travail, nous proposons une approche combinant l'apprentissage automatique, à travers la Machine à Vecteurs de Support (SVM), et l'apprentissage profond (CNN) pour la classification des expressions faciales.

Des expérimentations ont été réalisées sur deux bases de données :

JAFFE : un taux de reconnaissance de 90,91 % avec SVM et 95,45 % avec CNN a été obtenu.

FER-2013 : bien que plus riche en données, cette base a nécessité un temps d'entraînement long (~10 heures), pour aboutir à une précision plus faible de 56 % avec CNN.

Malgré ces résultats perfectibles, cette approche montre un fort potentiel, et ouvre la voie à des améliorations futures à travers l'optimisation des hyperparamètres et l'intégration de techniques avancées d'extraction de caractéristiques.

Mots-clés : Reconnaissance des expressions faciales; Apprentissage automatique ,Machine à vecteurs de support; Traitement d'images ; SVM ; CNN; JAFFE .

Abstract

Facial expression is a set of facial muscle movements that reflect an individual's emotional state. Automatically recognizing these expressions is a key area of artificial intelligence, with diverse applications such as human-computer interaction and surveillance. Numerous studies have explored feature extraction and the use of classifiers to improve the accuracy of facial recognition systems. However, performance remains limited on large databases.

In this work, we propose an approach based on machine learning, specifically the Support Vector Machine (SVM) and CNN, for facial expression classification. Experiments were conducted on the JAFFE database, achieving a recognition rate of 90,91% .

Keywords: Facial expression recognition; Machine learning; Support vector machine; Image processing; SVM;CNN; JAFFE.

ملخص

تعبير الوجه هو مجموعة من حركات عضلات الوجه التي تعكس الحالة العاطفية للفرد. ويُعد التعرف التلقائي على هذه التعبيرات مجا ل رئيسيا في الذكاء الصنطاعي، مع تطبيقات متنوعة مثل التفاعل بين الإنسان والآلة، والمراقبة، والتعليم الذكي. لقد تناولت العديد من الدراسات تقنيات استخراج الميزات واستخدام المصنفات بهدف تحسين دقة أنظمة التعرف على تعابير الوجه. ومع ذلك، ل يزال الأداء محدودا عند التعامل مع قواعد بيانات كبيرة ومعقدة

، (SVM) في هذا العمل، نقتراح نه اجا يعتمد على التعلم الآلي، وبشكل أكثر تحديدا على آلة الدعم المتجه لتصنيف تعبيرات الوجه

معدل تعرف JAFFE ، حيث حققت قاعدة JAFFE و FER-2013 تم إجراء التجارب على قاعدتي البيانات بدقة قدرها 56% فقط FER-2013 ، في حين حققت CNN و 95.45% باستخدام SVM قدره 90.91% باستخدام. نظ ارا لأدائها

الأفضل وسهولة التعامل معها JAFFE بنا اء على هذه النتائج، تم اختيار قاعدة

ورغم إمكانية تحسين هذه النتائج، فإنها تُبرز إمكانيات هذا النهج، وتفتح آفاقا لتطويره من خلال تحسين

معلمات النموذج ودمج تقنيات متقدمة لاستخراج الميزات، مما قد يُسهم في تحسين دقة أنظمة التعرف على تعابير الوجه، خاصة عند التعامل مع بيانات أكثر تعقيدا وتنوعا

كلمات مفتاحية

التعرف على تعبيرات الوجه؛ الموجهة؛ الدعم آلة الصور؛ معالجة المتجه؛ الدعم آلة الآلي، التعلم

Dédicaces

Nous avons l'honneur de dédier ce mémoire fruit de nos efforts et notre parcours de formation à nos chers parents avec tout notre respect.

Nous dédions ce mémoire à :

- ✓ A nos familles.
- ✓ A tous ceux et ceux celle qui nous ont aidé de près ou de loin.
- ✓ A mon binôme, mes meilleurs ami(es).

Remerciements

Nous remercions tout d'abord **DIEU** qui nous a donné la force et le courage pour pouvoir réaliser ce travail.

Nous tenons à exprimer nos sincères remerciements et notre reconnaissance notre encadreur **Mme KAID SLIMANE Bouchra** d'avoir proposé ce sujet qui nous a mené vers la découverte du monde de l'expérimental, de nous avoir bien dirigé afin de pouvoir réaliser ce travail, nous n'oublions pas de la remercier pour sa disponibilité, ses conseils et son expérience.

Nous remercions également les membres du jury d'avoir accepté d'évaluer notre travail.

Sur le plan plus personnel, nous remercions vivement nos parents pour leurs encouragements et leur soutien pendant notre cursus, ainsi que tous les membres de nos chères familles et tous nos amis.

Enfin, on remercie tous ceux qui ont contribué de près ou de loin au bon déroulement de ce travail.

Liste des figures

Figure N°	Titre de la figure	Page
Figure 1.1	Les expressions faciales	10
Figure 1.2	Les 6 émotions faciales universelles	12
Figure 1.3	Méthodologie de reconnaissance d'expression	14
Figure 2.1	Échantillons d'images FER-2013 pour la reconnaissance faciale des émotion	22
Figure 2.2	Exemples d'images provenant de la base de données sur l'expression faciale DFAT-504 de Cohn-Kanade	23
Figure 2.3	Exemples d'images provenant de la base de données sur JAFFE	23
Figure 2.4	Exemples comment fonctionner L'ACP	25
Figure 2.5	Exemples comment fonctionner le HOG	25
Figure 2.6	Exemples comment fonctionner le LBP	26
Figure 2.7	Illustration du SVM Classifieur linéaire séparant les deux classes	27
Figure 2.8	Architecture d'un Réseau de Neurones Convolutifs (CNN) pour la Classification d'Images	32
Figure 2.9	Fonction d'Activation ReLU (Rectified Linear Unit) et ses Propriétés	34
Figure2.10	Diagramme de cas d'utilisation	36
Figure 2.11	Diagramme de classe	37
Figure 2.12	Diagramme de séquence pour envoyer l'image	38
Figure 3.1	les classes d'émotions utilisées dans la base de données	45

Figure 3.2	les images de la base de données 48x48 pixels	46
Figure 3.3	Capture du code d'installation des bibliothèques	47
Figure 3.4	Définition du chemin d'accès aux données et des classes d'expressions faciales	47
Figure 3.5	Prétraitement des images et extraction des caractéristiques HOG	48
Figure 3.6	Normalisation des données	48
Figure 3.7	Réduction de dimensionnalité avec l'analyse en composantes principales (PCA)	49
Figure 3.8	Entraînement d'un modèle SVM avec noyau RBF	49
Figure 3.9	Évaluation du modèle SVM par calcul de la précision	49
Figure 3.10	Sauvegarde des modèles et préprocesseurs	50
Figure 3.11	Fonction de prédiction d'expressions faciales	50
Figure 3.12	Configuration de base d'une application Flask pour le traitement d'images	51
Figure 3.13	Création des endpoints Flask pour la classification d'expressions faciales	51
Figure 3.14	Importation des bibliothèques pour un modèle de classification d'images CNN	52
Figure 3.15	Préparation des données pour l'entraînement du modèle	52
Figure 3.16	Architecture d'un modèle CNN pour la reconnaissance d'expressions faciales	53
Figure 3.17	Configuration de la sauvegarde automatique du modèle	53
Figure 3.18	Entraînement et évaluation du modèle CNN	54
Figure 3.19	Page d'accueil du système de reconnaissance d'expressions faciales	54

Figure 3.20	Interface utilisateur de prédiction d'émotions faciales	55
Figure 3.21	Exemple de prédiction d'expression faciale sur une image	56

Table des matières

Introduction Générale7

Chapitre 1 Contexte général du projet**Error! Bookmark not defined.**

1.1	Introduction	9
1.2	Expressions Faciales	Error! Bookmark not defined.
1.2.1	Définition	Error! Bookmark not defined.
1.2.2	Importance	Error! Bookmark not defined.
1.3	Catégories des expressions faciale	Error! Bookmark not defined.
1.4	Défis et enjeux de la reconnaissance des expressions faciales	Error! Bookmark not defined.
1.4.1	Variabilité des expressions	Error! Bookmark not defined.
1.4.2	Problèmes liés a l'éclairage et aux angles de prise de vue	12
1.4.3	Limites des bases de données existantes	13
1.4.4	Enjeux éthiques et sociaux	13
1.5	Système de la reconnaissance des expressions faciales	Error! Bookmark not defined.
1.5.1	Architecture Générale du Système	Error! Bookmark not defined.
1.5.2	Défis et Enjeux de la Reconnaissance des Expressions Faciales	Error! Bookmark not defined.
1.5.3	Améliorations et Perspective	Error! Bookmark not defined.
1.5.4	Applications de la Reconnaissance des Expressions Faciales	16
1.6	Application Web	Error! Bookmark not defined. 17
1.6.1	Définition d'une application Web.....	17
1.6.2	Pourquoi utiliser une application Web pour la FER ?.....	17
1.7	Conclusion	Error! Bookmark not defined.
Chapitre 2 Approches et techniques utilisées Error! Bookmark not defined. 9		
2.1	Introduction	Error! Bookmark not defined. 9
2.2	Apprentissage automatique	Error! Bookmark not defined. 9
2.3	Bases de données utilisées	Error! Bookmark not defined.
2.4	Techniques d'extraction des caractéristiques	Error! Bookmark not defined.
2.4.1	ACP (Analyse en Composantes Principales)	Error! Bookmark not defined.
2.4.2	Autres méthodes (HOG, LBP) si nécessaire.....	25

2.5	Classification des expressions faciales les Machines à Vecteurs de Support	Error!
Bookmark not defined.		
2.5.1	Principe des SVM	27
2.5.2	Utilisation des SVM pour la FER	30
2.5.3	Avantages et Limites des SVM dans la FER	Error! Bookmark not defined.0
2.5.4	Améliorations et Perspectives.....	31
2.6	Classification par Réseaux de Neurones Convolutifs (CNN)	31
2.6.1	Définition de CNN	32
2.6.2	Principe des CNN.....	32
2.6.3	Utilisation dans notre projet	33
2.7	Modélisation UML d'Application web	Error! Bookmark not defined.
2.7.1	Définition d'UML	Error! Bookmark not defined.
2.7.2	Diagramme de cas d'utilisation	35
2.7.3	Diagramme de classes	Error! Bookmark not defined.
2.7.4	Diagramme de séquence.....	37
2.8	Conclusion	Error! Bookmark not defined.
Chapitre 3 Expérimentations et résultats		
Error! Bookmark not defined.		
3.1	Introduction	41
3.2	Environnement de travail	Error! Bookmark not defined.
3.2.1	Environnement matériel (hardware)	Error! Bookmark not defined.
3.2.2	Environnement immatériel (software).....	40
3.2.3	Outils de développement	Error! Bookmark not defined.
3.2.4	Base de données	Error! Bookmark not defined.
3.3	Partie du programme	Error! Bookmark not defined.
3.3.1	Modèle SVM	Error! Bookmark not defined.
3.3.2	Modèle CNN	Error!
Bookmark not defined.		
3.4	Conclusion	Error! Bookmark not defined.
Conclusion Générale		
58		

Bibliographie.....	58
--------------------	----

Introduction Générale

La reconnaissance des expressions faciales est un domaine clé de l'intelligence artificielle et de la vision par ordinateur, elle permet d'analyser et d'interpréter les émotions humaines à partir d'images ou de vidéos. Cette technologie trouve des applications variées dans plusieurs secteurs, notamment la sécurité, la médecine, le marketing et les interactions homme-machine. Parmi les différentes approches de reconnaissance des expressions faciales, l'apprentissage automatique a émergé comme une solution efficace pour améliorer la précision des systèmes. Parmi les méthodes les plus utilisées dans ce contexte nous avons la Machine à Vecteurs de Support (SVM) et le deep learning (CNN), les deux méthodes permettent une classification robuste des expressions en s'appuyant sur des caractéristiques extraites des images.

Dans ce mémoire, nous proposons deux approches basées sur l'Analyse en Composantes Principales (ACP) pour l'extraction des caractéristiques, combinée à un classificateur SVM pour reconnaître les expressions faciales humaines. Nous utilisons également le CNN comme deuxième approche, tout en exploitant les outils OpenCV pour le prétraitement des images et scikit-learn pour la mise en œuvre des modèles d'apprentissage. De plus, nous avons développé une interface avec Flask pour permettre une utilisation interactive du système.

Ce travail est structuré comme suit :

- Le premier chapitre présente le contexte général du projet, en détaillant les concepts de reconnaissance faciale, d'expressions faciales et les défis associés.
- Le deuxième chapitre décrit les approches et les techniques utilisées, notamment les bases de données exploitées, les méthodes d'extraction des caractéristiques et la modélisation UML du système.
- Le troisième chapitre est consacré aux expérimentations et aux résultats obtenus, avec une analyse des performances du modèle et des perspectives d'amélioration.

Enfin, nous conclurons ce mémoire par une discussion sur les limites de nos approches et les axes futurs de recherche pour améliorer la reconnaissance des expressions faciales.

Chapitre 1

Contexte général du projet

1.1 Introduction

La reconnaissance des expressions faciales est un domaine de recherche en pleine expansion, à la croisée de la vision par ordinateur, de l'intelligence artificielle et des neurosciences. Elle vise à analyser les émotions humaines à travers les expressions du visage, en identifiant des caractéristiques spécifiques liées aux mouvements musculaires. Cette technologie joue un rôle clé dans de nombreuses applications, notamment dans les interactions homme-machine, la surveillance, la psychologie et la médecine.

Mais comment un système peut-il détecter et interpréter automatiquement une émotion humaine à partir d'une simple image ? Quels sont les défis techniques qui limitent encore aujourd'hui l'efficacité des systèmes de reconnaissance des expressions faciales ?

Avec l'essor du Machine Learning et des techniques de Deep Learning, les performances des systèmes de reconnaissance faciale ont considérablement progressé. Cependant, la reconnaissance des expressions faciales présente encore plusieurs défis, notamment liés à la variabilité des visages, aux différences d'illumination, aux angles de prise de vue et aux émotions ambiguës.

Dans ce chapitre, nous présentons d'abord les concepts fondamentaux de la reconnaissance faciale et des expressions faciales. Ensuite, nous discuterons des défis et enjeux liés à cette

1.2 Les expressions faciales

1.2.1 Définition

Les expressions faciales font partie de notre langage corporel instinctif. Elles révèlent des informations supplémentaires, souvent subtiles, sur les choses que nous disons mais surtout sur

celles que nous ne disons pas. Par les expressions faciales, nous manifestons nos états émotionnels à l'extérieur, en changeant notre regard, en souriant ou en utilisant d'autres micro-expressions spécifiques à une émotion. Pensez à la façon dont vous ressemblez habituellement lorsque vous êtes heureux ou en colère. Très probablement, votre expression faciale lorsque vous êtes de bonne humeur comprend un large sourire, des rides autour des yeux, des joues surélevées et des coins de lèvres en diagonale. Ces expressions sont importantes dans vos interactions avec les autres, car elles permettent aux gens de saisir votre état émotionnel et de réagir de manière appropriée. [1]



Figure 1.1 – Les expressions faciales.

1.2.2 Importance

Les expressions faciales jouent un rôle clé dans divers domaines [2] :

- **Communication et interactions sociales :**
 - Elles permettent d'exprimer des émotions sans avoir besoin de mots.

- Elles aident à interpréter les intentions et les sentiments d'une personne.
- Elles renforcent ou nuancent un message verbal, rendant la communication plus efficace
- **Psychologie et neurosciences :**
 - Elles sont étudiées pour comprendre les émotions humaines et leur impact sur le comportement.
 - Elles sont utilisées dans la détection des troubles émotionnels et psychologiques, comme l'autisme ou la dépression.
- **Intelligence artificielle et reconnaissance faciale :**
 - Les systèmes de reconnaissance faciale analysent les expressions pour identifier les émotions en temps réel.
 - Utilisée en sécurité, marketing, santé mentale, et interactions homme-machine.
- **Applications médicales et bien-être :**
 - L'analyse des expressions faciales peut aider à détecter des signes de douleur ou de stress
 - Utilisée pour adapter les traitements en fonction des réponses émotionnelles des patients.

1.3 Catégories des expressions faciales

Pour Le psychologue Paul Ekman a identifié six émotions de base universelles, chacune associée à une expression faciale distincte [3]:

1. **Joie (Happiness)** : Caractérisée par un sourire, avec les coins de la bouche relevés et des rides autour des yeux.

2. **Tristesse (Sadness)** : Marquée par un abaissement des coins de la bouche et un regard souvent dirigé vers le bas.
3. **Colère (Anger)** : Se manifeste par des sourcils froncés, des yeux écarquillés et des lèvres serrées.
4. **Surprise (Surprise)** : Impliquant des sourcils levés, des yeux grands ouverts et une bouche légèrement ouverte.
5. **Peur (Fear)** : Caractérisée par des sourcils levés et rapprochés, des yeux grands ouverts et une bouche ouverte.
6. **Dégoût (Disgust)** : Se traduit par un froncement du nez et une lèvre supérieure relevée.



Figure 1.2 – les 6 émotions faciales universelles.

1.4 Défis et enjeux de la reconnaissance des expressions faciales

1.4.1 Variabilité des expressions

Les expressions faciales varient d'un individu à l'autre en raison de plusieurs facteurs :

- **Différences interindividuelles** : la manière dont une personne exprime une émotion peut différer en fonction de son âge, de son genre ou de sa culture.

- **Micro-expressions** : ces expressions brèves (moins de 0,5 seconde) sont difficiles à détecter par des systèmes automatiques.
- **Expressions composites** : certaines émotions peuvent être mélangées, rendant leur classification plus difficile (ex. : surprise + joie). [4]

1.4.2 Problèmes liés à l'éclairage et aux angles de prise de vue

- **Conditions d'éclairage** : la luminosité peut altérer la capture des traits du visage, affectant la précision des algorithmes.
- **Positions et angles** : un visage peut être partiellement caché ou incliné, compliquant l'extraction des caractéristiques faciales.
- **Occlusions** : les accessoires (lunettes, masques, barbes) peuvent masquer certaines parties du visage, réduisant la performance des modèles de reconnaissance.

1.4.3 Limites des bases de données existantes

- **Manque de diversité** : certaines bases de données ne couvrent pas toutes les variations culturelles et physiologiques des expressions faciales.
- **Expressions exagérées** : les jeux de données contiennent souvent des expressions artificielles qui ne reflètent pas toujours les émotions naturelles.
- **Données en conditions réelles** : la plupart des modèles sont entraînés sur des images statiques bien éclairées, mais la reconnaissance en environnement réel (vidéo, faible lumière) reste un défi.

1.4.4 Enjeux éthiques et sociaux

- **Vie privée et protection des données** : la reconnaissance faciale soulève des inquiétudes sur l'utilisation des données personnelles sans consentement.

- **Biais algorithmiques** : certains modèles sont plus performants sur certaines populations, ce qui peut entraîner des discriminations.
- **Utilisation dans la surveillance** : l'application de ces technologies en sécurité et surveillance peut poser des questions éthiques sur la liberté individuelle.

1.5 Système de la reconnaissance des expressions faciales

Un **système de reconnaissance des expressions faciales (FER - Facial Expression Recognition)** est un ensemble de techniques et d'algorithmes permettant d'identifier et de classer les émotions exprimées par un visage humain à partir d'images ou de vidéos. Ces systèmes jouent un rôle crucial dans divers domaines, notamment la psychologie, la sécurité et les interactions homme-machine [5].

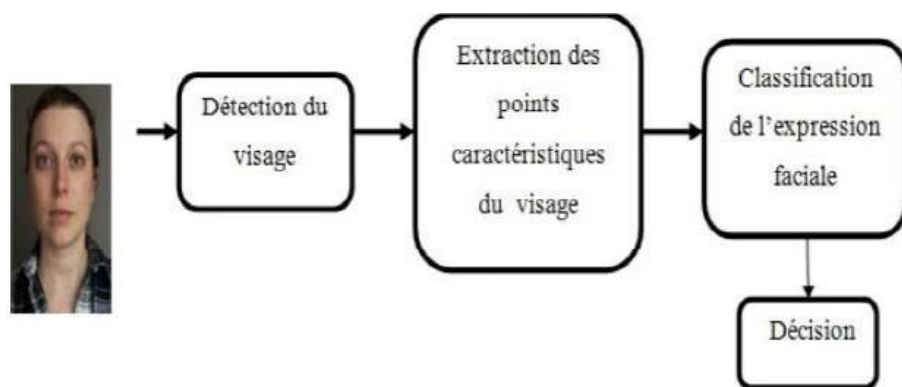


Figure 1.3 – Méthodologie de reconnaissance d'expression.

1.5.1 Architecture Générale du Système

Un **système FER** est généralement structuré en trois grandes phases [6] :

1.5.1.1 Acquisition et Prétraitement des Images

L'objectif de cette phase est d'obtenir une image de qualité exploitable en réduisant le bruit et en normalisant les données.

- **Capture des images** : Les données peuvent provenir de bases de données spécifiques (FER2013, JAFFE, CK+, etc.) ou être acquises en temps réel via une caméra
- **Détection et extraction du visage** : Les visages sont isolés à l'aide d'algorithmes spécialisés par exemple (**Haar Cascade**, **MTCNN** ou **HOG + SVM**).

- **Alignement et normalisation** : Correction des variations de rotation et d'échelle pour améliorer la qualité des images traitées.
- **Filtrage et réduction du bruit** : Suppression des artefacts inutiles et conversion en niveaux de gris si nécessaire.

1.5.1.2 Analyse et Traitement des Caractéristiques

Cette étape permet d'extraire les informations essentielles du visage. Les méthodes utilisées varient selon l'approche adoptée (statistique, géométrique ou basée sur l'apprentissage automatique). (Détails dans Chapitre 2)

1.5.1.3 Classification des Expressions Faciales

Une fois les caractéristiques extraites, un modèle de classification attribue une expression faciale à l'image analysée. Cette étape repose souvent sur des techniques de machine learning. (Détails dans Chapitre 2)

1.5.1.4 Affichage et Exploitation des Résultats

- Présentation des émotions détectées via une interface interactive.
- Analyse des tendances émotionnelles pour des applications avancées (ex. suivi psychologique, intelligence artificielle émotionnelle).

1.5.2 Défis et Enjeux de la Reconnaissance des Expressions Faciales

Malgré les avancées technologiques, plusieurs défis subsistent [7] :

- **Variabilité des expressions** : Un même individu peut exprimer une émotion de différentes manières, rendant la classification plus complexe.
- **Influence des conditions environnementales** : Les changements de lumière, d'angle de prise de vue et la présence d'occlusions (lunettes, barbe, masque) altèrent la reconnaissance.
- **Généralisation du modèle** : Un modèle entraîné sur une base de données spécifique peut avoir du mal à généraliser sur d'autres jeux de données.
- **Biais des bases de données** : Certaines bases de données sont déséquilibrées (ex. sous-représentation de certaines émotions), ce qui peut introduire un biais dans les prédictions.

- **Reconnaissance en temps réel** : L'optimisation du temps de calcul est essentielle pour les applications interactives.

1.5.3 Améliorations et Perspectives

Afin de pallier ces limites, plusieurs stratégies peuvent être mises en œuvre [8] :

- **Augmentation des données** : Utilisation de transformations d'images pour enrichir les bases de données (rotation, éclairage, occlusions artificielles).
- **Fusion de plusieurs techniques d'extraction** : Combinaison de **PCA + HOG + LBP** pour maximiser l'extraction des caractéristiques.
- **Approches multi-modales** : Combinaison de la reconnaissance faciale avec d'autres signaux (voix, posture, mouvement corporel) pour améliorer la précision.
- **Optimisation des modèles** : Réglage fin des hyperparamètres et utilisation de techniques comme **Transfer Learning** pour améliorer les performances.

1.5.4 Applications de la Reconnaissance des Expressions Faciales

Les avancées en reconnaissance faciale ont conduit à des applications dans divers domaines [9] :

1. Psychologie et Neurosciences

- Analyse des émotions dans les études cognitives et les troubles psychologiques (**détection de la dépression, autisme, etc.**).
- Applications en neuro-marketing pour analyser les réactions des consommateurs.

2. Sécurité et Surveillance

- Identification d'individus en situation de stress ou d'agression.
- Détection des comportements suspects dans les lieux publics (aéroports, banques).

3. Interactions Homme-Machine (IHM)

- Amélioration des assistants virtuels (ex. adaptation des réponses en fonction des émotions de l'utilisateur).
- Jeux vidéo et réalité virtuelle avec interactions émotionnelles en temps réel.

1.6 Application Web

1.6.1 Définition d'une application Web

Une application Web est un logiciel qui s'exécute dans votre navigateur Web. Les entreprises doivent échanger des informations et fournir des services à distance. Elles utilisent des applications Web pour se connecter aux clients de manière pratique et sécurisée. Les fonctionnalités de site Web les plus courantes telles que les paniers d'achats, la recherche et le filtrage de produits, la messagerie instantanée et les flux d'actualités sur les réseaux sociaux sont de par leur conception des applications Web, qui vous permettent d'accéder à des fonctionnalités complexes sans installer ni configurer de logiciel. [10]

1.6.2 Pourquoi utiliser une application Web pour la FER ?

L'utilisation d'une application Web pour la reconnaissance des expressions faciales (FER – **Facial Expression Recognition**) présente plusieurs avantages [11] :

- **Accessibilité et portabilité** : L'application peut être utilisée sur différents appareils (ordinateurs, tablettes, smartphones) sans installation supplémentaire. Cela facilite son déploiement et son utilisation.
- **raitement centralisé** : L'analyse des expressions faciales peut être effectuée sur un serveur, ce qui réduit la charge de calcul sur les appareils des utilisateurs et permet d'améliorer la rapidité et la précision du traitement.
- **Mise à jour et maintenance simplifiées** : Contrairement aux applications de bureau, une mise à jour unique sur le serveur suffit pour que tous les utilisateurs bénéficient des dernières améliorations.
- **ntégration avec d'autres services** : L'application peut facilement interagir avec des bases de données distantes, des services d'intelligence artificielle (APIe reconnaissance faciale, stockage cloud) et d'autres outils pour améliorer les performances et les fonctionnalités.
- **Scalabilité et flexibilité** : Une application Web permet d'ajouter de nouvelles fonctionnalités ou de supporter un plus grand nombre d'utilisateurs sans nécessiter de modifications majeures.

1.7 Conclusion

Dans ce premier chapitre, nous avons exploré les concepts fondamentaux liés à la reconnaissance des expressions faciales. Nous avons tout d'abord défini ce qu'est une expression faciale et souligné son importance dans la communication humaine. Ensuite, nous avons présenté les différentes catégories d'expressions faciales, dont nous nous sommes basé sur les travaux de Paul Ekman, qui a identifié six émotions de base. Nous avons également présenté des défis et des enjeux majeurs de la reconnaissance des expressions faciales, tels que la variabilité des expressions, les conditions d'éclairage et les angles de prise de vue, ainsi que les limites des bases de données existantes. Enfin, nous avons détaillé les étapes clés d'un tel système de reconnaissance, mettant en évidence les approches utilisées pour la détection, l'extraction des caractéristiques et la classification des émotions.

Ce chapitre a permis d'établir les bases théoriques nécessaires à la compréhension des systèmes de reconnaissance des expressions faciales. Dans le chapitre suivant, nous approfondirons les techniques spécifiques utilisées pour l'implémentation d'un Site web, en mettant l'accent sur les méthodes d'extraction des caractéristiques et les algorithmes de classification avec la présentation des concepts de base de langage de modélisation UML qui pourront nous aider à accomplir notre projet.

Chapitre 2

Approches et Techniques Utilisées

2.1 Introduction

La reconnaissance des expressions faciales repose sur une combinaison de techniques d'extraction de caractéristiques et de méthodes de classification. L'évolution des algorithmes de vision par ordinateur et d'apprentissage automatique a considérablement amélioré la précision des systèmes de reconnaissance faciale, permettant d'identifier avec plus de fiabilité les émotions exprimées par un individu.

Dans ce chapitre, nous présentons les principales approches utilisées dans notre projet. Nous commencerons par la définition de l'apprentissage automatique après une description des bases de données employées pour l'entraînement et l'évaluation du modèle. Ensuite, nous examinerons les techniques d'extraction de caractéristiques, notamment l'Analyse en Composantes Principales (ACP), ainsi que d'autres méthodes comme les Histogrammes de Gradients Orientés (HOG). Enfin, nous détaillerons l'approche de classification basée sur les Machines à Vecteurs de Support (SVM), qui constitue le cœur de notre système de reconnaissance des expressions faciales.

L'objectif est d'expliquer le choix des techniques adoptées, leurs avantages et leurs limites, afin de justifier leur pertinence dans le cadre de notre étude.

2.2 Apprentissage automatique

2.2.1 Définition

Le machine learning (ML) est une forme d'intelligence artificielle (IA) qui est axée sur la création de systèmes qui apprennent, ou améliorent leurs performances, en fonction des données qu'ils traitent. L'intelligence artificielle est un terme large qui désigne des systèmes

ou des machines simulant une forme d'intelligence humaine. Le machine learning et l'IA sont souvent abordés ensemble et ces termes sont parfois utilisés de manière interchangeable bien qu'ils ne renvoient pas exactement au même concept. [12]

2.2.2 Types d'apprentissage automatique

A-Machine learning supervisé

Les algorithmes de machine learning supervisé sont les plus couramment utilisés. Avec ce modèle, un data scientist sert de guide et enseigne à l'algorithme les conclusions qu'il doit tirer. Tout comme un enfant apprend à identifier les fruits en les mémorisant dans un imagier, en apprentissage supervisé, l'algorithme apprend grâce à un jeu de données déjà étiqueté et dont le résultat est prédéfini.

Comme exemples de machine learning supervisé, on peut citer des algorithmes tels que la régression linéaire et logistique, la classification en plusieurs catégories et les machines à vecteurs de support.

B-Machine learning Non supervisé

Le machine learning non supervisé utilise une approche plus indépendante dans laquelle un ordinateur apprend à identifier des processus et des schémas complexes sans un quelconque guidage humain constant et rigoureux. Le machine learning non supervisé implique une formation basée sur des données sans étiquette ni résultat spécifique défini.

Pour continuer avec l'analogie de l'enseignement scolaire, le machine learning non supervisé s'apparente à un enfant qui apprend à identifier un fruit en observant des couleurs et des motifs, plutôt qu'en mémorisant les noms avec l'aide d'un enseignant. L'enfant cherche des similitudes entre les images et les sépare en groupes, en attribuant à chaque groupe sa propre étiquette. Comme exemples d'algorithmes de machine learning non supervisé, on peut citer la mise en cluster de k-moyennes, l'analyse de composants principaux et indépendants, et les règles d'association. [13]

2.2.3 les principaux algorithmes de Machine Learning

En apprentissage automatique, différents algorithmes ont été conçus pour répondre à un large panel de besoins. S'il n'est pas évident de tous les maîtriser ou de les appréhender, il est

important de connaître les principaux qui sont d'ailleurs enseignés dans toute formation en data science digne du nom. Ils appartiennent tous à l'une des catégories d'algorithmes énoncées ci-dessus

- Perceptron
- L'arbre de décision
- La régression linéaire
- La régression logistique
- K-Means
- Support Vector Machine (SVM) (ce que nous utiliserons dans notre projet) [14]
- CNN

2.3 Bases de données utilisées

La performance d'un système de reconnaissance des expressions faciales dépend fortement de la qualité et de la diversité des bases de données utilisées pour l'entraînement et l'évaluation des modèles. Dans cette section, nous présentons les principales bases de données employées dans notre projet.

2.3.1 FER-2013 (Facial Expression Recognition 2013)

FER-2013 est l'une des bases de données les plus utilisées pour la reconnaissance des expressions faciales. Elle a été introduite lors de la compétition "Facial Expression Recognition Challenge" organisée dans le cadre de l'International Conference on Machine Learning (ICML) en 2013. Cette base contient environ 35 887 images en niveaux de gris de 48×48 pixels, réparties en sept classes d'émotions :

- Colère (Angry)
- Dégoût (Disgust)
- Peur (Fear)
- Bonheur (Happy)
- Neutre (Neutral)
- Tristesse (Sad)
- Surprise (Surprise)

Elle est particulièrement utile pour l'entraînement des modèles, mais présente certains défis liés à la qualité des images et au déséquilibre des classes[15]



Figure 2.1 – Échantillons d’images FER-2013 pour la reconnaissance faciale des émotions.

2.3.2 CK+ (The Extended Cohn-Kanade Dataset)

Le dataset CK+ est une référence incontournable dans la reconnaissance des expressions faciales. Il contient 593 séquences vidéo d’expressions faciales annotées, capturées à partir de 123 individus. Chaque séquence commence par une expression neutre et évolue progressivement vers une expression émotionnelle intense. Ce dataset est particulièrement utile pour l’analyse des micro-expressions et la détection des changements subtils dans les expressions faciales. [16]



Figure 2.2 – Exemples d’images provenant de la base de données sur l’expression faciale DFAT-504 de Cohn-Kanade.

2.3.3 JAFFE (Japanese Female Facial Expression Database)

JAFFE est une base de données qui contient 213 images de 10 femmes japonaises exprimant 7 émotions différentes (joie, tristesse, colère, surprise, peur, dégoût et neutre). Ce dataset est souvent utilisé pour évaluer la robustesse des algorithmes face aux variations culturelles.[17]



Figure 2.3 – Exemples d’images provenant de la base de données sur JAFFE (Japanese Female Facial Expression Database).

2.3.4 Autres Bases de Données

D’autres bases de données existent pour la reconnaissance des expressions faciales, comme AffectNet, RaFD (Radboud Faces Database) et MMI Facial Expression Database. Toutefois, pour notre projet, nous avons privilégié l’utilisation de FER-2013, qui est adaptée aux modèles basés sur l’apprentissage automatique et le Machine Learning.

L’utilisation de ces bases de données permet de garantir une bonne généralisation du modèle en tenant compte de la diversité des expressions et des variations interindividuelles

Dans notre projet on va utiliser la base de données FER-2013 (Facial Expression Recognition 2013).

2.4 Techniques d'Extraction des Caractéristiques

L'extraction des caractéristiques est une étape essentielle dans la reconnaissance des expressions faciales, permettant de transformer une image brute en un ensemble de descripteurs pertinents pour la classification. Plusieurs approches sont utilisées pour extraire les informations discriminantes des visages, notamment l'Analyse en Composantes Principales (ACP), ainsi que des méthodes comme les Histogrammes de Gradients Orientés (HOG) et les Local Binary Patterns (LBP).

2.4.1 Analyse en Composantes Principales (ACP)

L'analyse en composantes principales (ACP) est une technique statistique puissante utilisée pour simplifier des ensembles de données complexes tout en préservant les informations essentielles. Elle entre dans la catégorie de la réduction de la dimensionnalité, visant à diminuer le nombre de variables dans un ensemble de données pour le rendre plus facile à analyser et à modéliser. L'ACP y parvient en transformant les variables originales en un nouvel ensemble de variables appelées composantes principales. Ces composantes sont classées en fonction de la variance qu'elles capturent par rapport aux données d'origine, la première composante capturant la plus grande partie, la deuxième la plus grande partie suivante, et ainsi de suite.

Le principe est simple : L'ACP projette les images des visages dans un nouvel espace de caractéristiques basé sur leurs principales directions de variance, Cette transformation permet de supprimer les redondances et de réduire le bruit dans les images et En réduisant la dimensionnalité des données, L'ACP accélère l'entraînement des modèles et limite le risque de sur-ajustement.

Dans notre projet, L'ACP est appliqué après la détection et la normalisation des visages pour extraire les composantes principales qui décrivent au mieux les variations d'expression faciale [18]

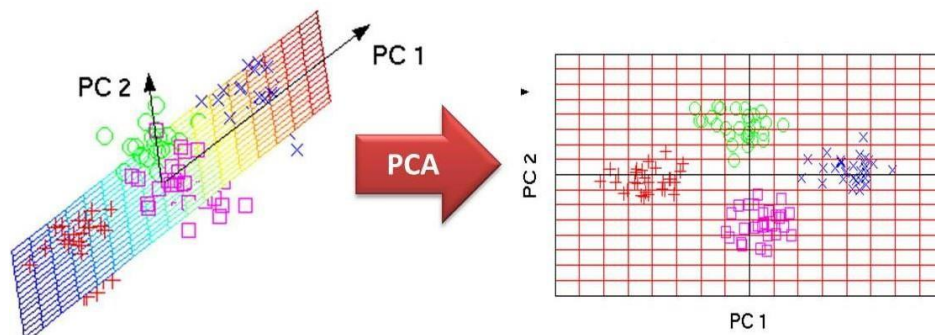


Figure 2.4 – Exemples comment fonctionner L'ACP.

2.4.2 Autres Méthodes d'Extraction des Caractéristiques

Outre L'ACP, il y a d'autres méthodes peuvent être utilisées pour extraire des descripteurs discriminants des visages :

a-Histogrammes de Gradients Orientés (HOG) :

Les HOG sont des descripteurs de forme qui comptabilisent les occurrences des orientations de gradient dans des portions localisées d'une image. Ils sont principalement utilisés pour la détection d'objets, mais se révèlent également efficaces pour modéliser la forme des muscles faciaux à travers une analyse des contours. [19]

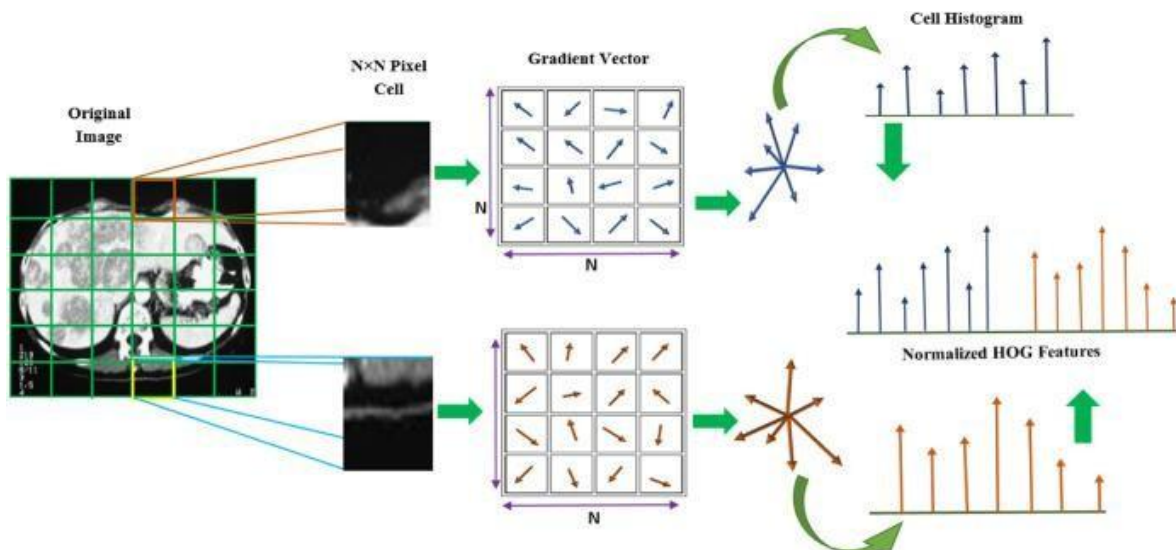


Figure 2.5 – Exemples comment fonctionner LE HOG.

b- Local Binary Patterns (LBP) :

Les LBP sont des descripteurs non paramétriques visant à résumer efficacement les structures locales des images. Ils ont suscité un intérêt croissant dans de nombreux domaines du traitement d'images et de la vision par ordinateur, démontrant leur efficacité dans diverses applications, notamment l'analyse des images faciales. [20]

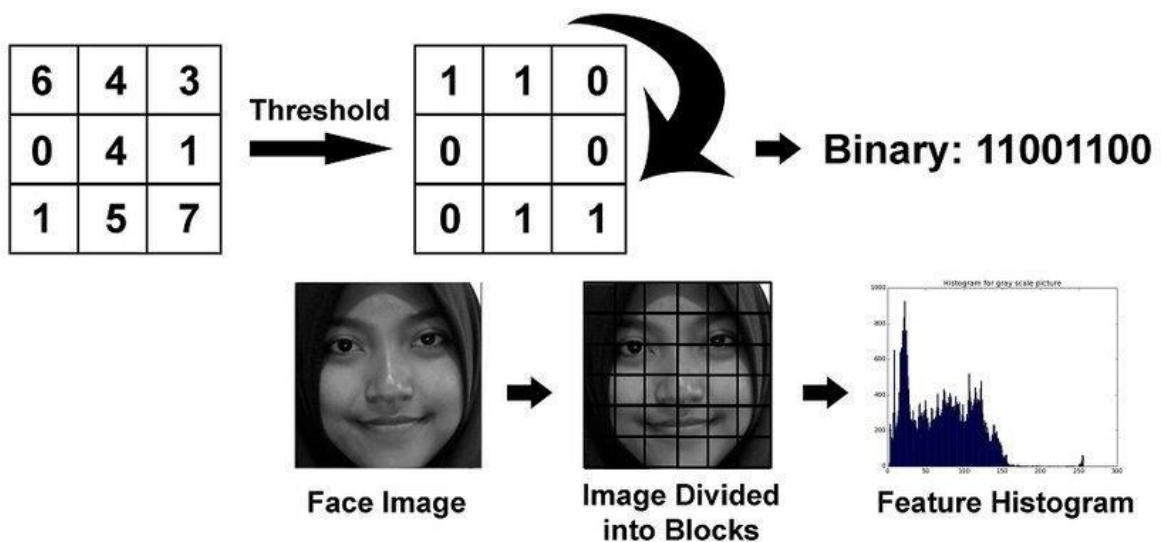


Figure 2.6 – Exemples comment fonctionner LE LBP.

2.5 Classification des expressions faciales les Machines à Vecteurs de Support (SVM)

La classification des expressions faciales est une étape essentielle dans un système de reconnaissance des expressions faciales (FER). Parmi les nombreux algorithmes de classification, les Machines à Vecteurs de Support (SVM - Support Vector Machines) se distinguent par leur efficacité, en particulier pour les problèmes de classification avec des ensembles de données de taille modérée. [21]

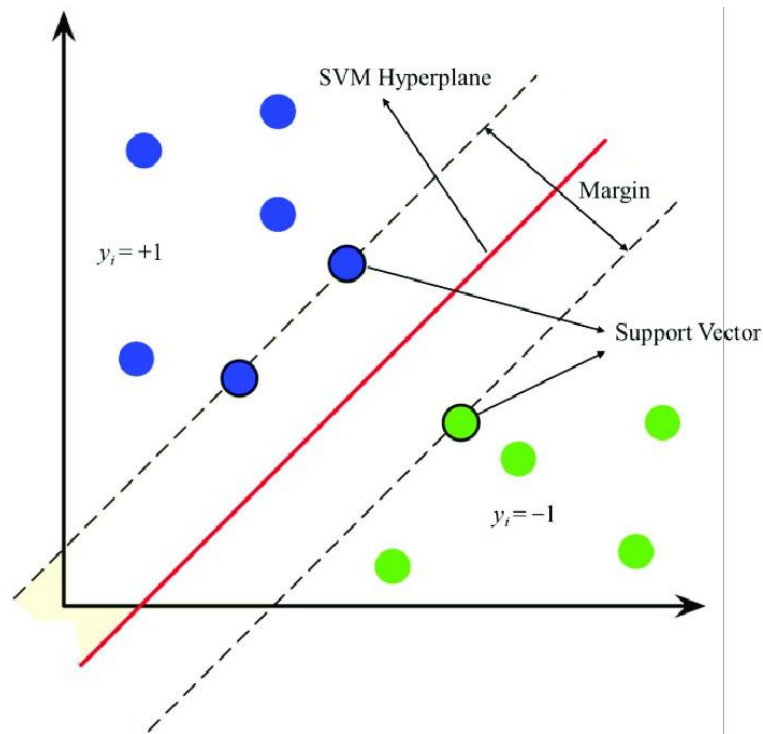


Figure 2.7 – Illustration du SVM Classifier linéaire séparant les deux classes.

2.5.1 Principe des SVM

Les machines à vecteurs de support (SVM, pour Support Vector Machines) sont des algorithmes d'apprentissage supervisé utilisés pour des tâches de classification et de régression. Elles excellent particulièrement dans la résolution de problèmes de classification binaire, où l'objectif est de séparer des données en deux catégories distinctes.

Pourquoi utiliser SVM pour la reconnaissance des expressions faciales ?

- **Efficacité dans les espaces de faible dimension** : SVM fonctionne bien avec des données réduites en dimensionnalité, ce qui est utile lorsqu'on applique des techniques comme PCA (Analyse en Composantes Principales).
- **Robustesse aux données bruitées** : Grâce à l'utilisation d'un paramètre de régularisation, SVM gère bien le bruit et les variations subtiles d'expression.

- **Capacité à gérer des classes non linéairement séparables** : Avec le **truc du noyau (Kernel Trick)**, les SVM peuvent transformer les données en un espace de dimension supérieure où une séparation linéaire est possible.

2.5.2 Utilisation des SVM pour la FER

L'application des SVM dans la reconnaissance des expressions faciales suit plusieurs étapes :

A-Extraction des caractéristiques :

Avant d'entraîner un modèle SVM, il est nécessaire d'extraire des caractéristiques pertinentes des images faciales. Les méthodes couramment utilisées incluent :

- **PCA (Principal Component Analysis)** : Réduction de la dimensionnalité tout en conservant les informations essentielles.
- **LBP (Local Binary Patterns)** : Analyse des textures du visage pour identifier les variations d'expressions.
- **HOG (Histogram of Oriented Gradients)** : Capture des contours et des gradients du visage.

B-Entraînement du modèle SVM :

L'entraînement d'un SVM pour la reconnaissance des expressions faciales repose sur la capacité du modèle à séparer différentes classes d'émotions en fonction des caractéristiques extraites des images. Selon la distribution des données dans l'espace des caractéristiques, trois cas principaux se présentent :

1. Linéairement séparable
2. Linéairement non séparable
3. Non linéaire

1. Linéairement Séparable

Lorsque les données sont parfaitement linéairement séparables alors seulement nous pouvons utiliser SVM linéaire. Parfaitement séparable linéairement signifie que les points de données peuvent être classés en 2 classes en utilisant une seule ligne droite (si 2D)

Principe :

Le **SVM linéaire** cherche l'**hyperplan optimal** qui maximise la **marge** entre les points des deux classes.

La fonction de décision est :

$$f(x)=w \cdot x+b$$

où :

- w est le vecteur normal à l'hyperplan,
- b est le biais qui ajuste la position de l'hyperplan.

L'objectif est de maximiser la **marge** ($2/\|w\|$) pour améliorer la généralisation du modèle.

2. Linéairement Non Séparable

Dans les ensembles de données non séparables, le modèle SVM linéaire est étendu en incorporant des caractéristiques polynomiales ou en utilisant des fonctions noyau pour projeter les données dans un espace de dimension supérieure où une séparation linéaire devient possible. Cependant, même avec ces extensions, une séparation parfaite peut ne pas être réalisable en raison du bruit et des chevauchements dans la distribution des données. Pour y remédier, des variables d'assouplissement (ξ_i) sont introduites pour chaque point de données afin de mesurer le degré de mauvaise classification

Principe :

On introduit un terme de relaxation pour autoriser quelques erreurs :

$$\min ||w||^2 + C \sum \xi_i$$

où C contrôle le compromis entre maximisation de la marge et réduction des erreurs.

3. Non Linéaire

Lorsque les données ne sont pas linéairement séparables, on peut utiliser une SVM non linéaire. Cela signifie que, lorsque les points de données ne peuvent pas être séparés en deux classes par une ligne droite (en 2D), on utilise des techniques avancées comme le **kernel trick** pour les classer. Dans la plupart des applications réelles, les points de données ne sont pas linéairement séparables, c'est pourquoi on utilise le **kernel trick** pour résoudre ce problème

Principe :

On applique une transformation non linéaire pour projeter les données dans un espace de plus haute dimension, où elles deviennent linéairement séparables.

exemples de noyaux non linéaires :

- **Noyau Polynomial :**

$$K(x_i, x_j) = (x_i \cdot x_j + c)^d$$

- **Noyau RBF :**

$$K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$$

où γ contrôle l'influence des points proches dans l'espace transformé

C- Classification des expressions faciales avec SVM

Une fois le modèle SVM entraîné, il est utilisé pour classifier une nouvelle image en fonction des expressions faciales prédéfinies, telles que la joie, la tristesse, la colère, la surprise, la peur, le dégoût et l'expression neutre. Ce processus repose sur plusieurs étapes essentielles, allant du prétraitement de l'image jusqu'à l'affichage des résultats.

Prétraitement de l'image d'entrée

Lorsqu'une nouvelle image est fournie au modèle, elle doit d'abord être traitée afin d'être compatible avec les caractéristiques utilisées lors de l'entraînement. La première étape consiste à détecter et extraire le visage de l'image à l'aide d'algorithmes de détection tels que Haar Cascade ou MTCNN. Ensuite, le visage est redimensionné et normalisé afin d'assurer une uniformité entre les différentes images en entrée. De plus, les mêmes techniques d'extraction de caractéristiques appliquées durant la phase d'entraînement, comme PCA (Analyse en Composantes Principales), HOG (Histogram of Oriented Gradients) ou LBP (Local Binary Patterns), sont utilisées pour convertir l'image en un vecteur de caractéristiques exploitable par le classificateur SVM.

Passage de l'image au modèle SVM et classification

Après l'extraction des caractéristiques, le vecteur obtenu est introduit dans le modèle SVM, qui évalue à quelle catégorie d'expression faciale il appartient. Le SVM repose sur la notion

d'hyperplan séparateur, permettant de distinguer différentes classes d'émotions. Grâce aux données apprises lors de l'entraînement, le modèle analyse la position du vecteur dans l'espace des caractéristiques et attribue une étiquette correspondant à l'émotion la plus probable.

Dans le cas de la classification des expressions faciales, le problème est multi-classe, ce qui signifie que le modèle doit être capable de différencier plus de deux catégories. Les SVM étant initialement conçus pour des problèmes binaires, il existe deux stratégies couramment utilisées pour gérer la classification multi-classe :

- **One-vs-One (OvO)** : Cette approche consiste à entraîner un classificateur SVM pour chaque paire d'émotions, et la classe finale est déterminée par un vote majoritaire.
- **One-vs-All (OvA)** : Ici, un classificateur est entraîné pour chaque émotion, en l'opposant à toutes les autres. L'émotion assignée est celle ayant le score le plus élevé.

Prédiction de l'émotion et affichage des résultats

Une fois que l'image a été classifiée, l'émotion prédite est retournée par le modèle. Si le système est implémenté dans une application interactive, comme une interface graphique basée sur Flask et OpenCV, l'expression reconnue peut être affichée en temps réel sur l'écran. Pour les applications en analyse comportementale, les résultats peuvent être enregistrés et étudiés sur une période prolongée afin de suivre l'évolution des émotions d'un individu.

2.5.3 Avantages et Limites des SVM dans la FER

2.5.3.1 Avantages :

- Bonne généralisation sur des petits ensembles de données.
- Résistant au sur-apprentissage (overfitting) grâce à la régularisation.
- Compatible avec différentes méthodes d'extraction comme PCA, LBP, et HOG.

2.5.3.2 Limites :

- Moins efficace pour les grandes bases de données : Le temps de calcul devient élevé lorsque le nombre d'échantillons augmente.
- Choix du noyau complexe : Trouver le bon noyau et régler les hyper paramètres (C, gamma) peut être difficile.

- Moins performant que le Deep Learning pour des tâches complexes comme la reconnaissance d'expressions faciales en temps réel.

2.5.4 Améliorations et Perspectives

Pour surmonter ces limites, plusieurs approches peuvent être envisagées :

- **Combinaison PCA + SVM** : Réduire la dimensionnalité avec PCA avant d'entraîner un SVM améliore l'efficacité.
- **Fusion de caractéristiques** : Utiliser une combinaison de LBP, HOG et PCA pour une meilleure robustesse.
- **Hybride SVM + CNN** : Intégrer SVM en couche de classification finale après un réseau de neurones convolutionnel (CNN) pour tirer parti du deep learning.

2.6 Classification par Réseaux de Neurones Convolutifs (CNN)

2.6.1 Définition de CNN

Les Convolutional Neural Networks (CNN), ou réseaux de neurones convolutifs, sont une classe d'architectures de deep learning particulièrement adaptées à l'analyse d'images. Ils ont montré des performances remarquables dans des tâches de classification d'images, y compris la reconnaissance des expressions faciales. [22]

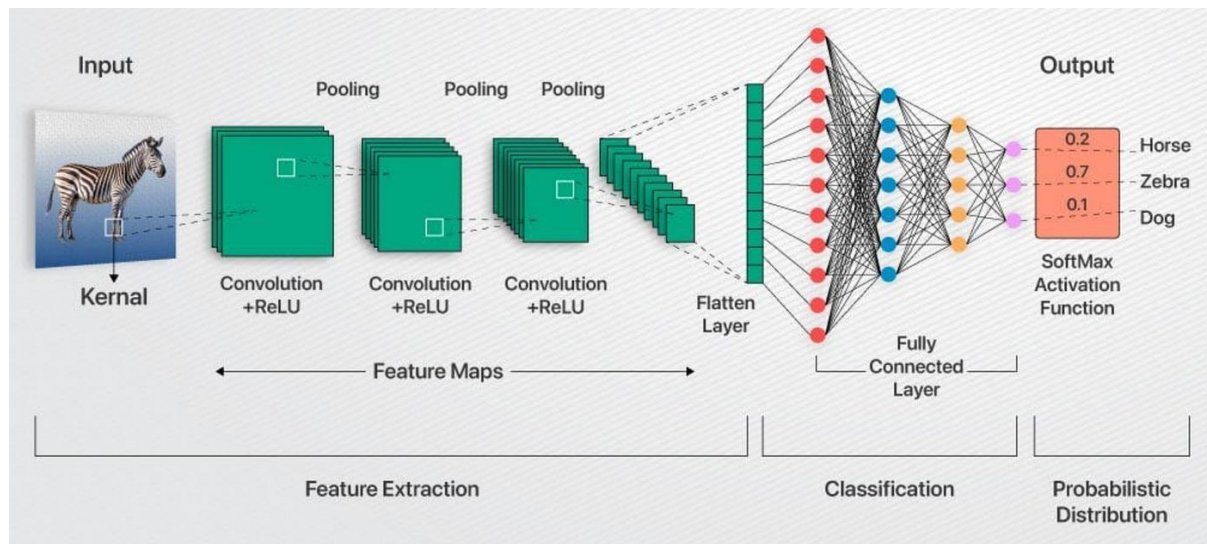


Figure 2.8 – Architecture d'un Réseau de Neurones Convolutifs (CNN) pour la Classification d'Images

2.6.2 Principe des CNN

Le CNN est composé de plusieurs types de couches :

1- Couche de convolution

La couche de convolution est la composante clé des réseaux de neurones, convolutifs, elle constitue toujours au moins leur première couche. Les couches de convolution sont formées de ce qu'on appelle des filtres. Les filtres sont des tableaux de valeurs appelées feature maps. Chaque couche de convolution prend en entrée une image et produit une feature map. Chaque feature map est obtenue en appliquant le filtre à l'image. Par exemple, si l'image est de taille 5x5 et que le filtre est de taille 3x3, la feature map sera de taille 3x3. La couche de convolution reçoit donc en entrée plusieurs images et calcule la convolution de chacune d'entre elles avec chaque filtre. Les filtres correspondent exactement aux features que l'on souhaite retrouver dans les images.

2- Couche de pooling

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling. Une couche de pooling, agit comme une couche de réduction. Elle divise l'image en blocs et ne garde que le maximum de chaque bloc. Cela permet de réduire la dimension de l'image tout en conservant les caractéristiques les plus importantes. On obtient en sortie le même nombre de feature maps qu'en entrée, mais celles-ci sont bien plus petites.

3- La couche de correction ReLU

ReLU (Rectified Linear Units) désigne la fonction réelle non-linéaire définie par $\text{ReLU}(x) = \max(0, x)$

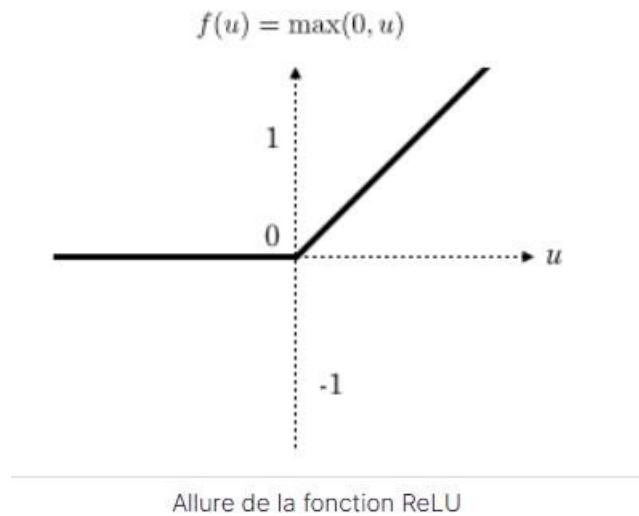


Figure 2.9 – Fonction d'Activation ReLU (Rectified Linear Unit) et ses Propriétés.

4- Couche fully-connected

Les CNN's sont généralement formés de plusieurs couches de convolution et de pooling, suivies par une couche fully-connected qui combine les features extraites par les couches précédentes pour classifier l'image, elle renvoie un vecteur de taille N, où N est le nombre de classes dans notre problème de classification d'images. Chaque élément du vecteur indique la probabilité pour l'image en entrée d'appartenir à une classe.

Par exemple, s'il s'agit bien d'un problème de classification de pommes et d'orange, le vecteur final sera de taille 2 : chaque élément donne la probabilité d'appartenir soit à la classe pomme, soit à la classe orange. Ainsi, le vecteur [0.8 , 0.2] signifie que l'image à 80% de chances de représenter une pomme.

2.6.3 Utilisation dans notre projet

Dans notre projet, nous avons expérimenté l'utilisation d'un modèle CNN (2 à 3 couches convolutives, suivies de couches de pooling et de couches entièrement connectées), afin de comparer ses performances à celles du SVM pour la classification des expressions faciales.

Comparaison entre CNN et SVM

Critère	svm	cnn
Extraction des caractéristique	Manuelle (ex. HOG, LBP, SIFT) nécessite expertise	Extraction des caractéristiques
Prétraitement des données	Nécessaire(redimensionnement normalisation,extraction manuelle)	Moins nécessaire – le réseau apprend directement à partir d’images
Capacité de généralisation	Bonne si bien régularisé, mais dépend des features extraits	Très bonne si bien entraîné (surtout avec dropout/batch norm)
Complexité de mise on œuvre	Relativement simple (avec sklearn,ect)	Plus complexe (besoin de frameworks comme TensorFlow)
Temps d'entraînement	Plus long sur grands datasets (pas optimisé pour GPU)	Rapide avec GPU
Exigences matérielles	Faibles – fonctionne sur CPU sans problème	Élevées – GPU recommandé pour entraînement
Précision sur JAFFE	90,91 %	95.45%
Taille du dataset nécessaire	Peut fonctionner avec petits jeux si features sont bons	Nécessite souvent plus de données pour bien généraliser
Interprétabilité	Relativement interprétable (on voit les marges, support vectors)	Moins interprétable (modèle boîte noire)
Performance avec grandes données	Limitée, overfitting possible sur grands jeux sans réduction de dimension	Excellente – plus le dataset est grand, plus la performance augmente

2.7 Modélisation UML d'Application web

La modélisation UML (Unified Modeling Language) est une étape essentielle pour structurer et organiser l'application Web dédiée à la reconnaissance des expressions faciales (**FER**). Elle permet de représenter graphiquement les différents composants du système, les interactions entre eux et le comportement attendu.

Dans cette section, nous allons présenter UML et les trois diagrammes essentiels pour notre application Web :

2.7.1 Définition de l'UML

UML (langage de modélisation unifié) est un langage de modélisation graphique à base de pictogrammes, il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ».

L'UML est couramment utilisé dans les projets logiciels, il est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet.

UML offre un standard de modélisation pour représenter l'architecture logicielle. [23]

2.7.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux Utilisateurs du système. C'est le premier diagramme du modèle UML au niveau duquel s'assure la relation entre l'utilisateur et les objets que le système met en œuvre.[24]

Principaux acteurs :

- **Utilisateur** : Interagit avec l'application en téléchargeant une image ou en utilisant la caméra pour analyser les expressions faciales.
- **Site web** : Traite l'image en appliquant l'algorithme SVM pour reconnaître l'expression faciale.

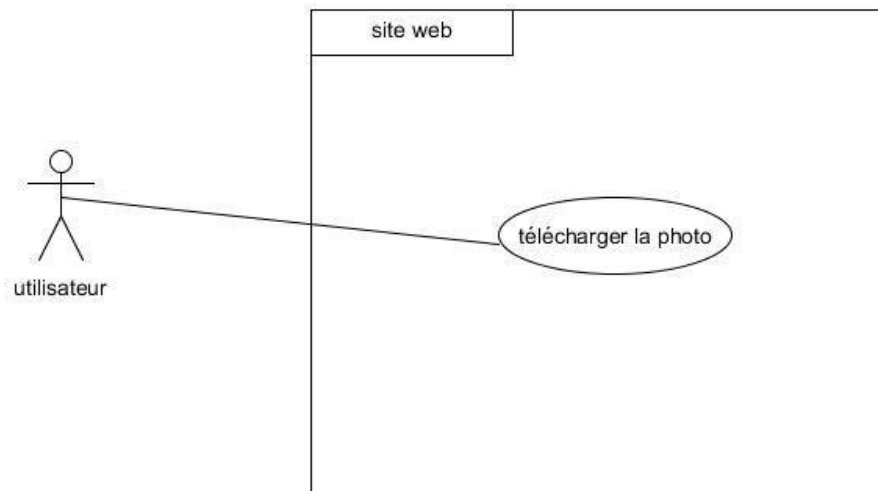


Figure 2.10 – Diagramme de cas d'utilisation

2.7.3 Diagramme de classe et de conception

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels.

Classes principales :

- **Utilisateur** (nom, email, rôle)
- **Image** (id, chemin)

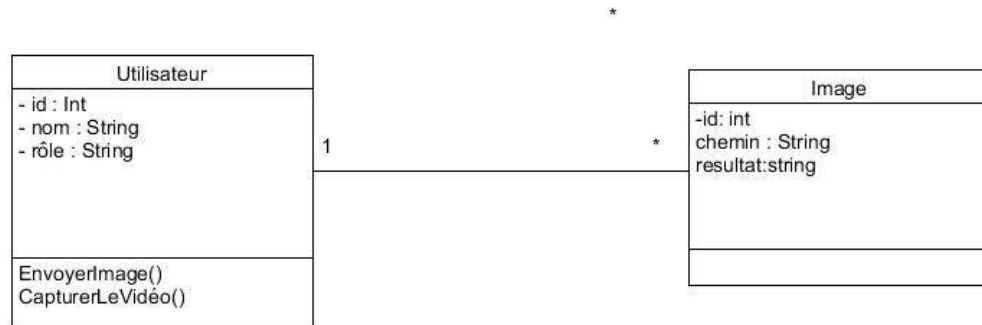


Figure 2.11 – Diagramme de classe

2.7.4 Diagrammes de séquences

Les diagrammes de séquence servent à illustrer les cas d'utilisations décrits précédemment. Ils permettent de représenter la succession chronologique des opérations réalisées par un acteur et qui font passer d'un objet à un autre pour représenter un scénario. Dans cette partie, nous allons décrire les scénarios les plus importants ainsi que leurs représentations par les diagrammes de séquence.

Envoyer une image

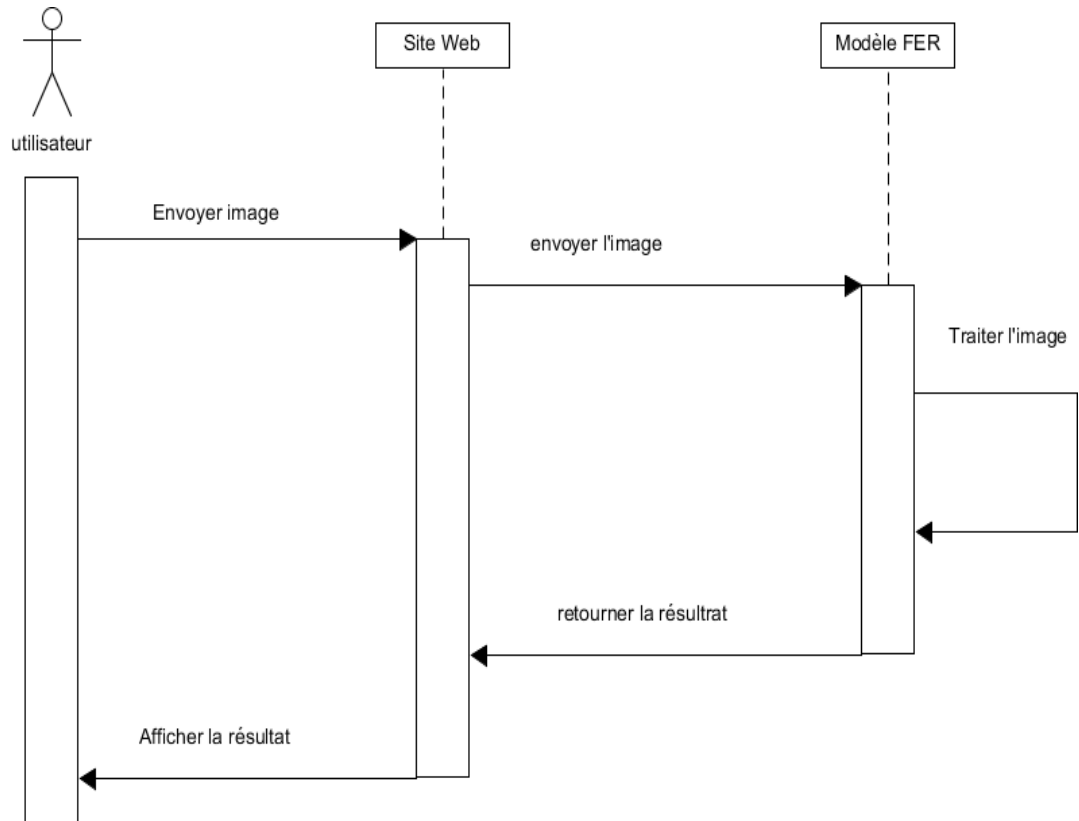


Figure 2.12 – Diagramme de séquence pour envoyer l'image.

2.8 Conclusion

Dans ce chapitre, nous avons exploré les différentes approches et techniques utilisées pour la reconnaissance des expressions faciales. Nous avons d'abord présenté les bases de données utilisées, qui jouent un rôle crucial dans l'entraînement et l'évaluation des modèles. Ensuite, nous avons détaillé les techniques d'extraction des caractéristiques, notamment l'Analyse en Composantes Principales (ACP), ainsi que d'autres méthodes comme HOG et LBP, permettant de transformer les images faciales en des représentations exploitables par les algorithmes de classification.

Nous avons également mis l'accent sur l'utilisation des Machines à Vecteurs de Support (SVM) et le CNN, en expliquant leurs fonctionnements et leurs importances dans ce domaine. De plus, nous avons introduit la modélisation UML de notre application web, ce qui nous a permis de mieux structurer et organiser les interactions entre les différents acteurs et composants du système, le chapitre suivant concerne la conception de nos modèles.

Chapitre 3

Expérimentations et résultats

1.2 Introduction

Ce chapitre est consacré à l'implémentation de notre site Web qui s'appuie sur la modélisation présentée dans le chapitre précédant, nous allons présenter les différents outils et langage utilisés lors de l'implémentation puis nous présentons notre site Web à travers ses différentes interfaces.

3.2 Environnement de développement

3.2.1 Environnement matériel

(hardware):

Un ordinateur avec les caractéristiques suivantes :

- **Processeur** : Processeur : Intel i5 8th Gen
- **RAM** : 8go
- **Disque Dur** : 512 go SSD

3.2.2 Environnement immatériel (software) :

- **Système d'exploitation** : Windows 10 64bit
- **Logiciel** : python 3.7 64bit

3.2.3 Outils d'implémentation :

❖ PYTHON [25]

Python est un langage de programmation qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. Il est cependant particulièrement utilisé comme langage de script pour automatiser des tâches simples mais fastidieuses, comme un script qui récupérerait la météo sur Internet ou qui

s'intégrerait dans un logiciel de conception assistée par ordinateur afin d'automatiser certains enchaînements d'actions répétitives (voir la section Adoption). On l'utilise également comme langage de développement de prototype lorsqu'on a besoin d'une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau. Il est particulièrement répandu dans le monde scientifique, et possède de nombreuses bibliothèques optimisées destinées au calcul numérique.

❖ **Scikit-Learn [26]**

Scikit-Learn a été créée en 2007 par l'ingénieur David Cournapeau sous le nom de Scikit-Learn, pour un projet Google Summer of Code (GSoC). Développé et maintenu par une communauté de bénévoles, le projet compte aujourd'hui plus de 30 contributeurs actifs. Son lancement public a eu lieu le premier février 2010, ce qui lui a permis de bénéficier du soutien financier de l'INRIA, Google, Tinyclues et Python Software Foundation.

Scikit-Learn est une bibliothèque open source d'apprentissage automatique en Python. Aussi connue sous son nom abrégé `sklearn`, elle fournit une vaste gamme d'outils, d'algorithmes et de fonctionnalités permettant de faciliter le développement de modèles de machine learning. Scikit-Learn couvre les principaux algorithmes de machine learning généralistes : classification, régression, clustering, gradient boosting.

❖ **NumPy [27]**

NumPy (Numerical Python) est une bibliothèque Python open source utilisée pour le calcul scientifique. Elle permet de manipuler des tableaux multidimensionnels (arrays) et fournit de nombreuses fonctions mathématiques efficaces pour effectuer des opérations sur ces tableaux. Elle est très utilisée dans les domaines de la science des données, de l'intelligence artificielle, du traitement d'images, etc.

❖ **OS [28]**

La bibliothèque `os` de Python est l'une des bibliothèques les plus importantes et les plus utilisées de la bibliothèque standard Python. Elle offre une multitude de fonctions permettant d'interagir avec le système d'exploitation sous-jacent, notamment l'accès et la manipulation du système de fichiers, des variables d'environnement et même des processus.

❖ **Joblib [29]**

Joblib est une bibliothèque open source pour le langage de programmation Python, qui facilite le traitement parallèle, la mise en cache des résultats et la distribution des tâches. Elle a été conçue pour simplifier les tâches gourmandes en ressources de calcul en permettant aux développeurs de paralléliser les opérations, d'accélérer les calculs et de réduire le temps d'exécution global.

❖ **Scikit-image [30]**

`scikit-image` est une bibliothèque open source pour le traitement d'images en Python. Elle est construite sur NumPy, SciPy et matplotlib. Elle fournit un ensemble riche de fonctions pour lire, modifier, analyser et traiter des images (filtres, détection de contours, segmentation, transformation, etc.). Elle est utilisée dans des domaines variés comme la vision par ordinateur, la recherche scientifique, ou encore l'imagerie médicale.

❖ **Tensorflow [31]**

TensorFlow est une bibliothèque open source de machine learning et d'intelligence artificielle, développée par Google. Ce framework permet de concevoir, entraîner et déployer des modèles de deep learning (réseaux de neurones) de manière flexible et optimisée. Compatible avec Python et d'autres langages, TensorFlow est utilisé pour des tâches variées : vision par ordinateur, traitement du langage naturel, analyse de données, et bien plus.

❖ **OpenCV [32]**

OpenCV (Open Source Computer Vision Library) est une bibliothèque open source spécialisée dans la vision par ordinateur et le traitement d'images. Elle est compatible avec plusieurs langages, dont Python. Avec OpenCV, on peut capturer des vidéos, détecter des visages, appliquer des filtres, faire de la reconnaissance d'objets, du suivi de mouvement, etc. Elle est très utilisée dans des domaines comme la surveillance, la robotique, l'intelligence artificielle, et la réalité augmentée.

❖ **Flask [33]**

Flask est un framework web Python compact et léger qui fournit des outils et fonctionnalités utiles pour faciliter la création d'applications web en Python. Il offre une grande flexibilité aux développeurs et est plus accessible aux nouveaux développeurs, car il permet de créer rapidement une application web à partir d'un seul fichier Python.

❖ **HTML [34]**

Le langage de balisage hypertexte ou HTML ("HyperText Markup Language" en anglais) est le langage de balisage standard pour les documents conçus pour être affichés dans un navigateur Web. Il est souvent assisté par des technologies telles que les feuilles de style en cascade (CSS) et des langages de script tels que JavaScript.

Les navigateurs Web reçoivent des documents HTML d'un serveur web ou d'un stockage local et les restituent sous forme de pages Web multimédias. Le HTML décrit sémantiquement la structure d'une page Web et comportait à l'origine des indications sur son apparence.

❖ **CSS [35]**

Les feuilles de style en cascade ou CSS ("Cascading Style Sheets" en anglais) sont un langage de feuille de style utilisé pour décrire la présentation d'un document écrit dans un langage de balisage tel que HTML ou XML (y compris les dialectes XML tels que SVG,

MathML ou XHTML). CSS est une technologie de base du World Wide Web, au même titre que HTML et JavaScript.

CSS est conçu pour permettre la séparation du contenu et de la présentation, y compris la mise en page, les couleurs et les polices. Cette séparation peut améliorer l'accessibilité du contenu, fournir plus de flexibilité et de contrôle dans la spécification des caractéristiques de présentation, permettre à plusieurs pages Web de partager le formatage en spécifiant le CSS pertinent dans un fichier ".css" séparé.

❖ **Bootstrap [36]**

Bootstrap est une bibliothèque HTML, CSS et JS qui se concentre sur la simplification du développement de pages web informatives (par opposition aux applications web). Le but principal de l'ajouter à un projet web est d'appliquer les choix de couleur, taille, police et mise en page de Bootstrap à ce projet. En tant que tel, le principal facteur est de savoir si les développeurs en charge trouvent ces choix à leur goût. Une fois ajouté à un projet, Bootstrap fournit des définitions de style de base pour tous les éléments HTML. Le résultat est une apparence uniforme pour la prose, les tableaux et les éléments de forme dans tous les navigateurs web. En outre, les développeurs peuvent profiter des classes CSS définies dans Bootstrap pour personnaliser davantage l'apparence de leur contenu. Par exemple, Bootstrap a prévu des tableaux de couleur claire et foncée, des titres de page, des guillemets plus importants et du texte avec un sur-lignage.

❖ **JavaScript [37]**

JavaScript (JS) est un langage de programmation léger, interprété ou compilé en flux tendu, doté de fonctions de premier ordre. Bien qu'il soit surtout connu comme langage de script pour les pages Web, de nombreux environnements autres que les navigateurs l'utilisent également, tels que Node.js, Apache CouchDB et Adobe Acrobat. JavaScript est un langage dynamique, basé sur des prototypes, multiparadigme et à un seul fil, qui prend en charge les styles orienté objet, impératif et déclaratif (par exemple, la programmation fonctionnelle).

3.2.4 Base de données

Pour l'entraînement et l'évaluation de notre système de reconnaissance des expressions faciales, nous avons utilisé une base de données spécialisée contenant des images annotées représentant différentes émotions humaines. La base de données joue un rôle essentiel dans la performance du modèle, car elle doit couvrir un large éventail d'expressions, de sujets, de variations d'angles, et de conditions d'éclairage.

Dans notre projet, nous avons choisi d'utiliser la base de données JAFFE , car elle est largement utilisée dans la recherche scientifique et offre un bon compromis entre diversité, taille et qualité des annotations.

Nombre d'images : environ 213 images faciales annotées

Nombre de classes : 7 classes d'émotions : Colère (Angry) , Dégoût (Disgust) , Peur (Fear), Heureux (Happy) , Triste (Sad) , Surprise (Surprise) , Neutre (Neutral)

Nom	Modifié le	Type	Taille
angry	29/01/2025 20:17	Dossier de fichiers	
disgust	29/01/2025 20:17	Dossier de fichiers	
fear	29/01/2025 20:17	Dossier de fichiers	
happy	29/01/2025 20:17	Dossier de fichiers	
neutral	29/01/2025 20:17	Dossier de fichiers	
sad	29/01/2025 20:17	Dossier de fichiers	
surprise	29/01/2025 20:17	Dossier de fichiers	

Figure 3.1 – les classes d'émotions utilisées dans la base de données.

Format des images : 256×256 pixels, en niveaux de gris (grayscale).

Répartition entraînement/test : 80 % pour l'entraînement, 20 % pour le test.

Cette base a permis d'entraîner efficacement notre modèle SVM après extraction des caractéristiques par ACP. Les performances observées sur l'ensemble de test montrent une bonne capacité du modèle à généraliser à de nouvelles données.



Figure 3.2 – les images de la base de données.

3.3 Programme SVM

1. Modèle SVM

- ◆ Dans cette partie on importe les bibliothèques installées (comme OpenCV, scikit-learn, scikit-image) ainsi que certaines bibliothèques préinstallées (comme os et numpy) nécessaires au traitement d'images, à l'extraction de caractéristiques, et à l'entraînement du modèle SVM.

```

import cv2
import numpy as np
import os
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from skimage.feature import hog
from joblib import dump, load
#from sklearnex import patch_sklearn
✓ 0.0s

```

Figure 3.3 – Capture du code d’installation des bibliothèques.

- ◆ Dans cette partie, on définit le chemin vers le dossier contenant les images d’entraînement (data_dir) et on précise la liste des expressions faciales à reconnaître (expressions), comme angry, happy, sad, etc. Chaque dossier dans train doit correspondre à l’une de ces expressions et contenir les images associées.

```

data_dir = "C:/Users/lenovo/Desktop/BDDD/train/train"
expressions = ["angry", "disgust", "fear", "happy", "neutral", "sad", "surprise"]
✓ 0.0s

```

Figure 3.4 – Définition du chemin d'accès aux données et des classes d'expressions faciales.

- ◆ Dans cette partie, on prétraite chaque image en niveaux de gris, on applique un rehaussement de contraste (CLAHE), puis on extrait des caractéristiques HOG afin de transformer les images en vecteurs utilisables pour l'entraînement du modèle SVM.

```

def preprocess_image(img_path):
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
    if img is None:
        return None

    img = cv2.resize(img, (128, 128))
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8, 8))
    img = clahe.apply(img)

    features = hog(img, orientations=9, pixels_per_cell=(8, 8),
                   cells_per_block=(2, 2), feature_vector=True, block_norm='L2-Hys')
    return features.astype(np.float32)

X, y = [], []
for label, expression in enumerate(expressions):
    folder_path = os.path.join(data_dir, expression)
    for img_name in os.listdir(folder_path):
        img_path = os.path.join(folder_path, img_name)
        features = preprocess_image(img_path)
        if features is not None:
            X.append(features)
            y.append(label)

X = np.array(X, dtype=np.float32)
y = np.array(y)
print(f"Data shape: X={X.shape}, y={y.shape}")

```

Figure 3.5 – Prétraitement des images et extraction des caractéristiques HOG.

- ◆ Dans cette étape, on applique une normalisation des données à l'aide de StandardScaler, ce qui permet de centrer et de réduire les caractéristiques pour améliorer les performances du modèle SVM.

```

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
✓ 37.9s

```

Figure 3.6 – Normalisation des données.

- ◆ A ce niveau, on applique une réduction de dimensionnalité à l'aide de PCA avec 100 composantes principales, ce qui permet de transformer les données tout en conservant l'essentiel de leur variance pour optimiser les performances et l'efficacité du modèle.

```
pca = PCA(n_components=100)
X_train = pca.fit_transform(X_train)
X_test = pca.transform(X_test)
```

✓ 25.9s

Figure 3.7 – Réduction de dimensionnalité avec l'analyse en composantes principales (PCA).

- ◆ On entraîne un modèle SVM en utilisant un noyau RBF avec des hyper-paramètres $C=150$ et $\gamma=0.0001$.

```
model = SVC(kernel='rbf', C=150, gamma=0.0001)
model.fit(X_train, y_train)
```

Figure 3.8 – Entraînement d'un modèle SVM avec noyau RBF.

- ◆ Ce code correspond à l'évaluation des performances du modèle SVM en prédisant les étiquettes de l'ensemble de test (X_{test}) et en calculant la précision (accuracy). La précision, exprimée en pourcentage, mesure le taux de classification correcte du modèle.

```
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy * 100:.2f}%")
```

Py

Accuracy: 90.91%

Figure 3.9 – Évaluation du modèle SVM par calcul de la précision.

- ◆ Cette partie du code sauvegarde les trois éléments clés du pipeline (le modèle SVM entraîné, le normaliseur StandardScaler et le transformateur PCA) dans des fichiers séparés à l'aide de Joblib, permettant ainsi de les réutiliser ultérieurement pour des prédictions sans avoir à refaire l'entraînement ou le prétraitement des données.

```
dump(model, 'svm_model.joblib')
dump(scaler, 'scaler.joblib')
dump(pca, 'pca.joblib')
```

Figure 3.10 – Sauvegarde des modèles et préprocesseurs.

- ◆ Cette fonction prédit l'expression faciale d'une image en la prétraitant, et en appliquant les transformations (normalisation et PCA), puis en utilisant le modèle SVM entraîné, tout en gérant les cas d'images invalides.

```
def predict_expression(image_path):
    features = preprocess_image(image_path)
    if features is None:
        print(f"Warning: Could not read test image {image_path}")
        return "Invalid image"

    features = scaler.transform([features])
    features = pca.transform(features)

    prediction = model.predict(features)[0]
    return expressions[prediction]
```

Figure 3.11 – Fonction de prédiction d'expressions faciales.

- ◆ Ce code configure une application Flask en créant un dossier pour stocker les images uploadées et en préparant le serveur à recevoir des fichiers.

```

app = Flask(__name__)
UPLOAD_FOLDER = "uploads"
os.makedirs(UPLOAD_FOLDER, exist_ok=True)
app.config["UPLOAD_FOLDER"] = UPLOAD_FOLDER

```

Figure 3.12 – Configuration de base d'une application Flask pour le traitement d'images.

- ◆ Le code ci-dessous crée une API Flask avec deux routes : une pour afficher une page web de téléchargement d'image, et une autre pour recevoir l'image, prédire l'expression faciale, et renvoyer le résultat.

```

@app.route('/')
def home():
    return render_template("upload.html")

@app.route("/upload", methods=["POST"])
def upload():
    if "file" not in request.files:
        return jsonify({"error": "No file uploaded!"})

    file = request.files["file"]
    if file.filename == "":
        return jsonify({"error": "No file selected!"})

    file_path = os.path.join(app.config["UPLOAD_FOLDER"], file.filename)
    file.save(file_path)

    predicted_expression = predict_expression(file_path)
    return jsonify({"expression": predicted_expression})

if __name__ == "__main__":
    app.run(debug=True)

```

Figure 3.13 – Création des endpoints Flask pour la classification d'expressions faciales.

2. Modélé CNN :

Une importation des librairies est nécessaire pour créer et entraîner un modèle de réseau de neurones convolutif (CNN) capable de classer des images.

```

import os
import numpy as np
import cv2
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
from tensorflow.keras.callbacks import ModelCheckpoint

```

Figure 3.14 – Importation des bibliothèques pour un modèle de classification d'images CNN.

- ◆ Ce code charge des images d'expressions faciales, les redimensionne et les normalise pour préparer l'entraînement d'un modèle de reconnaissance d'émotions.

```

X, y = [], []
for label, expression in enumerate(expressions):
    folder_path = os.path.join(data_dir, expression)
    for img_name in os.listdir(folder_path):
        img_path = os.path.join(folder_path, img_name)
        img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
        if img is None:
            continue
        img = cv2.resize(img, (48, 48))
        img = img.astype('float32') / 255.0
        X.append(img)
        y.append(label)

X = np.array(X).reshape(-1, 48, 48, 1)
y = to_categorical(y, num_classes=len(expressions))

```

Figure 3.15 – Préparation des données pour l'entraînement du modèle.

- ◆ Ce code crée un réseau de neurones pour reconnaître les émotions sur les visages, avec des couches spécialisées pour extraire les caractéristiques et éviter les erreurs.

```

from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=(48, 48, 1)),
    BatchNormalization(),
    MaxPooling2D((2, 2)),
    Dropout(0.25),

    Conv2D(64, (3, 3), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D((2, 2)),
    Dropout(0.25),

    Conv2D(128, (3, 3), activation='relu', padding='same'),
    BatchNormalization(),
    MaxPooling2D((2, 2)),
    Dropout(0.5),

    Flatten(),
    Dense(256, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),

    Dense(len(expressions), activation='softmax')
])

```

Figure 3.16 – Architecture d'un modèle CNN pour la reconnaissance d'expressions faciales.

- ◆ Ce paramétrage permet de sauvegarder automatiquement la meilleure version du modèle pendant son entraînement, en se basant sur sa performance en validation.

```

checkpoint = ModelCheckpoint(
    'cnn_model.keras',
    save_best_only=True,
    monitor='val_accuracy',
    mode='max'
)

```

Figure 3.17 – Configuration de la sauvegarde automatique du modèle.

- ◆ Cette partie du code, permet le lancement de l'entraînement du modèle CNN pendant 30 cycles complets (epochs), en sauvegardant automatiquement la meilleure version, puis évalue sa précision finale sur des images de test pour mesurer ses performances réelles.

```
model.fit(X_train, y_train, epochs=30, batch_size=32, validation_split=0.1, callbacks=[checkpoint])

loss, acc = model.evaluate(X_test, y_test)
print(f"Test accuracy: {acc * 100:.2f}%")
```

Figure 3.18 – Entraînement et évaluation du modèle CNN.

3.2.3 Démonstrations des interfaces

Notre application contient les interfaces suivantes :

- **Page d'accueil**

C'est la première page qui s'affiche lorsque l'utilisateur entre dans notre site Web. L'utilisateur peut choisir la prédiction les expression faciales en sélectionnant une image ou en ouvrant la camera.

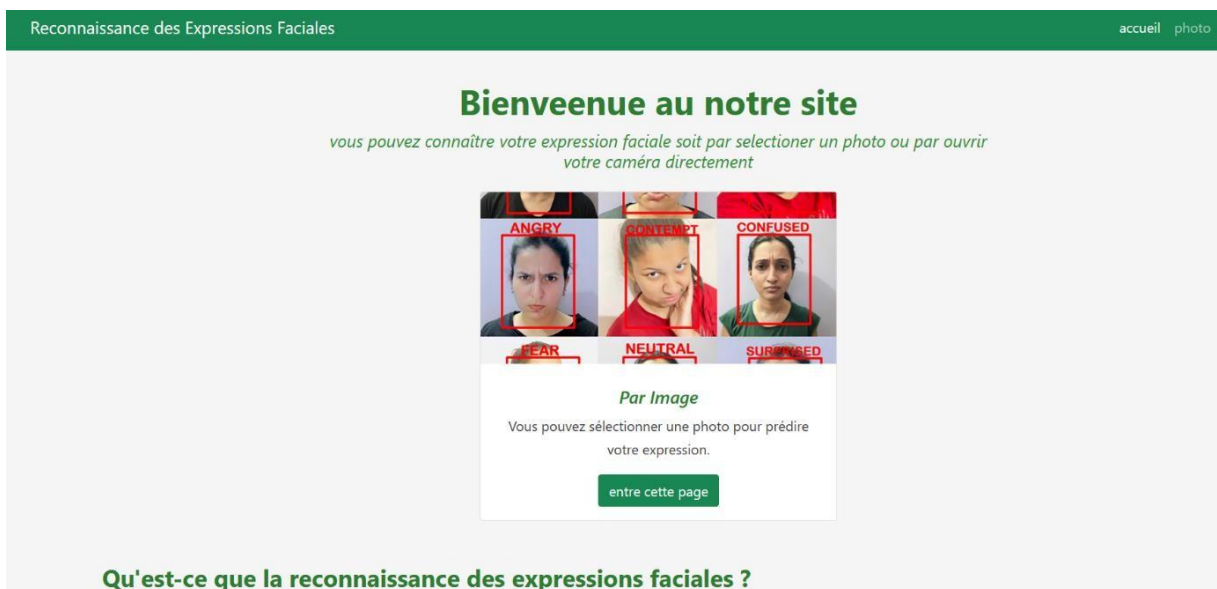


Figure 3.19 – Page d'accueil du système de reconnaissance d'expressions faciales.

- **Page de prédiction de la photo**

Dans cette page l'utilisateur peut sélectionner une photo et cliquer sur le bouton « prédire », ce dernier lui affiche le résultat.

The screenshot shows a web application titled 'Reconnaissance des Expressions Faciales'. It features a green header bar with the title and navigation links 'accueil', 'photo', and 'camera'. The main content area has a light gray background. At the top, it says 'Choisissez une photo pour prédire votre expression faciale'. Below this is a file selection interface with a button labeled 'Choisir un fichier' and a text area that says 'Aucun fichier n'a été sélectionné'. At the bottom center, there is a green button labeled 'Prédire'.

Figure 3.20 – Interface utilisateur de prédiction d'émotions faciales.

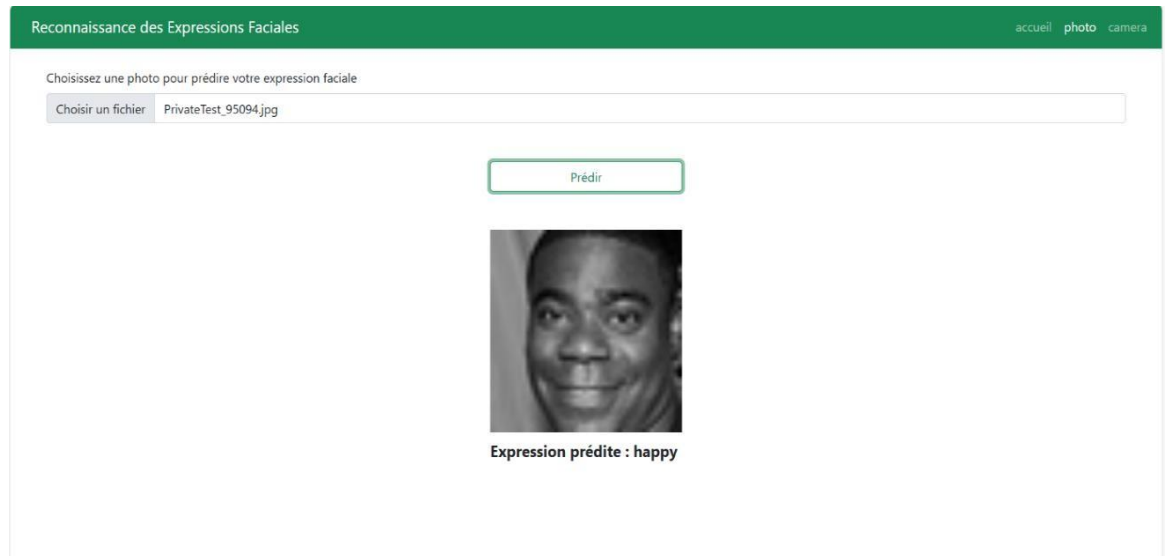


Figure 3.21 – Exemple de prédiction d'expression faciale sur une image.

3.4 Conclusion

Dans ce chapitre, nous avons décrit le processus de réalisation de notre application en spécifiant les outils et les langages de développement associés à notre système. En effet, nous avons achevé l'implémentation de nos deux classifieurs (SVM et CNN).

Conclusion Générale

L'objectif de notre projet était de concevoir une application capable de reconnaître automatiquement les expressions faciales humaines à partir d'images et vidéo, en utilisant des techniques de traitement d'image et d'intelligence artificielle. Pour ce faire nous avons commencé par une phase d'analyse et de recherche sur les expressions faciales, les bases de données disponibles, ainsi que les méthodes d'extraction des caractéristiques comme LBP, HOG, ACP. Ensuite, nous avons implémenté nos deux systèmes de reconnaissance et qui sont SVM et CNN, en utilisant les outils et bibliothèques adaptés tels que Python, OpenCV, scikit-learn.

Ce projet nous a permis de renforcer nos connaissances en apprentissage automatique, en traitement d'images, ainsi qu'en modélisation logicielle, et d'acquérir une expérience pratique dans la mise en œuvre d'un système intelligent basé sur l'analyse du visage humain.

Dans nos perspectives, nous envisageons de continuer à apprendre et à approfondir nos compétences dans le domaine de l'intelligence artificielle, notamment en explorant les réseaux de neurones profonds et les techniques de reconnaissance en temps réel, afin de développer des systèmes encore plus performants et utiles dans différents domaines comme la santé, la sécurité et l'interaction homme-machine.

Bibliographie

- [1] EIA Groupe Expressions faciales et micro-expressions | EIA Group (FR)

- [2] Communication Theory Importance of Facial Expressions in Communication

- [3] UNIVERSAL EMOTIONS Universal Emotions | What are Emotions? | Paul Ekman Group

- [4] Challenges and Emerging Trends for Machine Reading of the Mind from Facial Expressions (PDF) Challenges and Emerging Trends for Machine Reading of the Mind from Facial Expressions

- [5] Li, S., & Deng, W. (2020). Deep Facial Expression Recognition: A Survey. IEEE Transactions on Affective Computing, 13(3), 1195-1215. Deep Facial Expression Recognition: A Survey | IEEE Journals & Magazine | IEEE Xplore

- [6] Feature extraction and classification methods of facial expression .(PDF) Feature extraction and classification methods of facial expression: a surey

- [7] Evaluation and analysis of visual perception using attention Evaluation and analysis of visual perception using attention-enhanced computation in multimedia affective computing - PMC

- [8] Reconnaissance automatique des émotions Reconnaissance automatique des émotions — Wikipédia

- [9] Qu'est-ce qu'une application Web ? Qu'est-ce qu'une application Web ? – Les applications Web expliquées – AWS

[10] What are the application areas of facial expression recognition
<https://www.tencentcloud.com/techpedia/106380?>

[12] Qu'est-ce que le machine learning ? Qu'est-ce que le machine learning ? | Oracle Afrique

[13] Machine Learning : Définition, fonctionnement, utilisations Machine Learning :
Définition, fonctionnement, utilisations

[14] les 9 types d'algorithmes les plus pertinents en entreprise Machine Learning : les 9 types
d'algorithmes les plus pertinents en entreprise | LeMagIT

[15] Representation Learning: A report on three machine learning contests. In Neural Networks
(Vol. 64, pp. 59–63). Elsevier. <https://doi.org/10.1016/j.neunet.2014.09.005>

[16] The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and
emotion-specified expression. The Extended Cohn-Kanade Dataset (CK+): A complete dataset
for action unit and emotion-specified expression | IEEE Conference Publication | IEEE Xplore

[17] Coding facial expressions with Gabor wavelets. Coding facial expressions with Gabor
wavelets | IEEE Conference Publication | IEEE Xplore

[18] Eigenfaces for recognition. Journal of Cognitive Neuroscience. Eigenfaces for
Recognition | Journal of Cognitive Neuroscience | MIT Press

[19] Qu'est-ce que c'est : Histogramme de dégradés orientés Qu'est-ce que c'est : Explication
de l'histogramme des dégradés orientés

[20] Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A comparative study of texture
measures with classification based on featured distributions. *Pattern Recognition*, 29(1), 51–
59.

- [21] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- [22] Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2017). AffectNet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1), 18–31
- [23] “Livre sur UML” du module “Génie Logicielle”.
- [24] Les types des diagrammes UML <https://www.lucidchart.com/blog/fr/types-de-diagrammes-UML>
- [25] Python (langage) Python (langage) — Wikipédia
- [26] scikit-learn Scikit-learn — Wikipédia
- [27] Numpy What is NumPy? — NumPy v2.4.dev0 Manual
- [28] Python os Module Python os Module
- [29] Joblib Joblib : Quelle est cette bibliothèque Python ? Comment l'utiliser ?
- [30] Scikit-image Scikit-image — Wikipédia
- [31] TensorFlow TensorFlow : le framework de Machine Learning de Google
- [32] OpenCV OpenCV : tout savoir sur le principal outil de Computer Vision
- [33] Flask Flask (framework) — Wikipédia

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة عبد الحميد بن باديس-مستغانم-
كلية العلوم الدقيقة والإعلام الآلي

تصريح شرقي خاص بالالتزام بقواعد النزاهة العلمية
لإنجاز البحث

أنا الممضي أدناه،

الطالب(ة): Bouksara Meriem رقم التسجيل الجامعي: 202037029835

الحامل لبطاقة التعريف الوطنية رقم: 43191588 والصادرة بتاريخ: 25.10.2024

عن: Mostaganem المسجل بـ: كلية العلوم الدقيقة والإعلام الآلي / قسم 2:

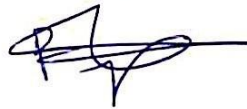
شعبة: Informatique / التخصص: Intelligence artificielle

والمكلف بإنجاز مذكرة ماستر بعنوان: Reconnaissance des expressions faciales avec SVM\CNN

أصرح بشرفي أنني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات العلمية والنزاهة الأكاديمية
المطلوبة في إنجاز البحث ، وأتحمل المسؤولية الشخصية عن كل المحتوى المتضمن في البحث المذكور أعلاه .

التاريخ: 28.06.2025

إمضاء المعني



* ملحق القرار الوزاري رقم 933 المؤرخ في 28 جويلية 2016 الذي يحدد القواعد المتعلقة بالوقاية من السرقة العلمية ومكافحتها.

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة عبد الحميد بن باديس - مستغانم -
كلية العلوم الدقيقة والإعلام الآلي

تصريح شرفي خاص بالالتزام بقواعد النزاهة العلمية
لإنجاز البحث

أنا الممضي أدناه،

الطالب(ة): رجاء منية بن نسي رقم التسجيل الجامعي: 2037031360
الحامل لبطاقة التعريف الوطنية رقم: 403335443 والصادرة بتاريخ: 07/13/2020
عن بلدية عين تادلس

المسجل بـ: كلية العلوم الدقيقة والإعلام الآلي / قسم: رياضيات وعلوم إعلامية آلي
شعبة: إعلام آلي / التخصص: الدكتوراه في الرياضيات والأشياء
والمكلف بإنجاز مذكرة ماستر بعنوان:

Reconnaissance des expressions faciales avec SVM/CNN

أصرح بشرفي أنني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات العلمية والنزاهة الأكاديمية
المطلوبة في إنجاز البحث ، وأتحمل المسؤولية الشخصية عن كل المحتوى المتضمن في البحث المذكور أعلاه .

التاريخ: 2020/06/28

إمضاء المعني

R. Younes

* ملحق القرار الوزاري رقم 933 المؤرخ في 28 جويلية 2016 الذي يحدد القواعد المتعلقة بالوقاية من السرقة العلمية ومكافحتها.