

Promotion 2019/2020

Mémoire de Projet de Fin d'Etude

Pour l'obtention du

Diplôme de Master Académique en Génie Electrique

Spécialité : Electronique des systèmes embarques

THEME

Détection et classification des émotions des personnes

Présenté par :

GUEMIDI Abdelwahab

KAZITANI Noredine

Soutenu le / / 2020 devant le jury composé de :

Président : Mr Benabdellah Yagoubi

Examineur : Mr Sid Ahmed Henni

Encadreur : Mr Mohamed Daoud

Co-encadreur : Mr Mohamed Bentoumi

Année Universitaire 2019/2020

Table des matières

Table des matières	1
<i>Remerciement</i>	5
Résumé :	6
Abstract:	7
:الملخص	8
Liste des figures	9
Liste des tableaux	10
Liste des abréviations	10
INTRODUCTION GENERALE.....	11
Chapitre I : Notions de base des émotions et expression facial	13
I.1 Introduction :	14
I.2 Emotions :	14
I.2.1 Définition :.....	14
I.2.2 La différence entre expressions faciales et émotions :	15
I.2.3 Méthodes d'identification des émotions :.....	16
I.2.3.1 Par expression faciale (la méthode directe) :.....	16
I.2.3.2 L'acquisition des signaux (la méthode indirecte) :.....	16
I.3 Détection de visage :	16
I.3.1 Les techniques de détection du visage.....	17
I.3.2 Les problèmes de détection du visage	18
I.4 L'expression faciale :	18
I.4.1 Définition :.....	18
I.4.2 Description des six expressions faciales :.....	18
I.4.3 Les différentes méthodes pour la détection des caractéristiques [25] :	21
I.5 Conclusion :.....	22
Chapitre II : Apprentissage automatique (<i>Machine Learning</i>)	23

II.1 Introduction :	24
II.2 Définition de l'apprentissage automatique (MACHINE LEARNING) :	24
II.3 Les étapes de l'apprentissage automatique :	25
II.4 Types d'apprentissage automatique [13] :	25
II.4.1 Apprentissage supervisé :	25
II.4.2 Apprentissage semi-supervisé :	26
II.4.3 Apprentissage non supervisé :	26
II.4.4 Apprentissage par renforcement :	26
II.5 Algorithmes d'apprentissage automatique :	27
II.5.1 Machine à vecteurs de support (SVM) (apprentissage supervisé – classification) :	27
II.5.2 Méthode des k plus proches voisins :	29
II.5.3 Forêt des arbres décisionnels :	30
II.6 Conclusion :	30
Chapitre III : Planification du projet et préparation de l'environnement	31
III.1 Introduction :	32
III.2 Déroulement du projet :	32
III.3 Préparation d'environnement :	34
III.3.1 Python :	34
III.3.2 Anaconda :	34
III.3.3 Les Bibliothèques :	35
III.3.4 IDE :	36
III.4 Conclusion :	37
Chapitre IV : Détection faciale, Extraction et sélection des caractéristiques et classification	38
IV.1 Introduction :	39
IV.2 Base des données (<i>Database</i>) :	39
IV.3 Détection du visage :	39
IV.4 Extraction des points de repère du visage :	39
IV.5 Extraction des caractéristiques :	40
IV.6 Sélection des caractéristiques (<i>Feature Selective</i>) :	40
IV.6.1 Le classificateur <i>Random Forest</i> :	40
IV.7 Les techniques d'entraînement et de sélection du classificateur :	42

IV.7.1 La validation croisée (cross-validation) :.....	42
IV.8 Classification avec plusieurs modèles :.....	43
IV.8.1 Avec SVM linéaire :.....	43
IV.8.2 Avec SVM polynomial :.....	44
IV.8.3 Avec le foret aléatoire :.....	45
IV.8.4 Avec KNN :.....	45
IV.9 Sauvegarde du classificateur entraîné :.....	47
IV.10 Conclusion :.....	47
Chapitre V : Implémentation et résultats expérimentaux.....	48
V.1 Introduction :.....	49
V.2 Extraction et sélection des caractéristiques de l'exemple du test :.....	49
V.2.1 Lire l'image :.....	49
V.2.2 Extraction et sélection :.....	50
V.3 Les prédictions des probabilités du chaque classificateur :.....	51
V.3.1 Chargement du classificateur :.....	51
V.3.2 Obtention des probabilités :.....	51
V.4 Les résultats finaux :.....	51
V.5 Autres exemples de test :.....	53
V.6 Conclusion :.....	54
Conclusion générale :.....	55
Bibliographie.....	56

Remerciement

Je remercie tout d'abord Allah, que grâce à lui on a achevé ce modeste travail. Je tiens à remercier monsieur Mohamed Daoud et monsieur Mr Mohamed Bentoumi , pour nous avoir encadrés et fourni l'aide nécessaire qui a été crucial pour la réussite du projet.

Je souhaite aussi remercier mes parents qui ont toujours su être là dans tous les moments, leur soutien, leur générosité et leurs volontés ont toujours mérité mon plus profond respect.

Mes remerciements s'adressent aussi à monsieur Benabdellah Yagoubi et monsieur Sid Ahmed qui ont participé au jury de soutenance.

Résumé :

Avec l'apparition de l'apprentissage automatique plusieurs tâches humaines ont été apprises à la machine l'une d'elles est la reconnaissance des émotions, ces émotions sont issues des expressions faciales, la reconnaissance des émotions peut s'avérer très utile dans le développement de la communication homme-homme et la communication homme-machine. La reconnaissance des émotions commence avec une base de données (images) ensuite, on doit détecter le visage, ensuite on extrait les caractéristiques puis on sélectionne les meilleures caractéristiques grâce à un classificateur, après cela on a utilisé les différents classificateurs qui permettent d'avoir les meilleures performances (justesse et précision), on est arrivé à avoir plus de 85 % de précision et 85% de justesse dans l'apprentissage automatique, et on a fini avec quelques exemples pour évaluer nos classificateurs.

Abstract:

With the appearance of machine learning the machine one of them is the recognition of emotions learns several human tasks, these emotions are derived from facial expressions, the recognition of emotions can prove very useful in the development of human-human communication and human-machine communication. Emotion recognition begins with a database (images) after that we detect face in each image, then the characteristics are extracted from the face and the best features are selected by using a classifier, after we create the different classification that allow to have the best performance (precision and accuracy), we have succeeded to have more than 85% accuracy and 85% precision in the most of classifiers and we ended up with a few examples to evaluate our classifiers.

الملخص:

مع ظهور التعلم الآلي يتم تعلم العديد من المهام البشرية للألة مثل التعرف على العواطف، والتي مصدرها تعابير الوجه، والاعتراف بالعواطف يمكن أن تكون مفيدة جدا في تطوير العلاقة بين الإنسان والإنسان وبين الإنسان والآلة. التعرف على العاطفة يبدأ بقاعدة بيانات (صور)، ثم يجب التعرف على الوجه من خلال الصور، بعد ذلك يتم استخراج الخصائص من الوجه واختيار أفضلها باستخدام مصنف (الغابة العشوائية)، بعد ذلك يجب استخدام مصنفات التعلم الآلي المختلفة التي تسمح بأداء أفضل (دقة)، تمكنا من الحصول على أكثر من 85% دقة في تعلم الآلة من قاعدة البيانات، وسوف نقوم ببعض الأمثلة لتقييم المصنفات التي استخدمناها والمقارنة بينها.

Liste des figures

Figure I.1: famille Exprimant de la joie [1].....	14
Figure I.2: Représentation du système limbique[4].....	15
Figure I.3 : Application de HOG [6]	17
Figure I.4: Extraction et normalisation de l'image[7].....	17
Figure I.5 : Représentation des 68 points de repère	22
Figure II.1 : Représentation des noyaux	29
Figure II.2 : Architecture du modèle de Random Forest	30
Figure III.1 : Planification du projet	33
Figure III.2 : Logo de Python	34
Figure III.3 : Fenêtre d'accueil de ANACONDA NAVIGATOR.....	34
Figure III.4: Logo de OpenCV.....	35
Figure III.5 : Logo de Scikit-learn.....	35
Figure III.6: Logo de NumPy.....	36
Figure III.7: Logo de Dlib [11]	36
Figure III.8: Logo de VS Code.....	36
Figure IV.1 : Représentation des 68 points de repère	39
Figure IV.2 : Représentation de quelques distances (des exemples)	40
Figure IV.3 : Représentation des caractéristiques par rapport au seuillage d'importance	41
Figure IV.4: Explication de la méthode K-fold.....	42
Figure V.1 : Image de test	49
Figure V.2 : Exemple d'un filtre éliminateur.....	50
Figure V.3 : Représentation des caractéristiques sélection	50
Figure V.4 : Les probabilités regroupées sous formes des barres.....	52

Liste des tableaux

Tableau IV.1 : La précision et la justesse des classificateurs.....46

Liste des abréviations

BVP : Blood Volume Pulse

EEG : électroencéphalogramme

EMG : électromyographie

Fc : fréquence cardiaque

HOG : Histogram of Oriented Gradients

IDE : Integrated development environment

KNN : K nearest neighbors

OpenCV : Open Source Computer Vision Library

ovr : one vs rest

REF : Reconnaissance de l'expression faciale

RNA : réseau de neurones artificiels

SKC : Skin Conductance

SKT : Skin Temperature

SVM : support vector machine

VR : Volume respiratoire

INTRODUCTION GENERALE

L'informatique affective est une thématique pluridisciplinaire émergente qui regroupe des chercheurs du domaine de l'intelligence artificielle et du traitement du langage naturel jusqu'au domaine des sciences sociales et cognitives. La reconnaissance des émotions et de manière plus large l'informatique affective, a significativement évolué ces dernières années.

De nos jours la reconnaissance des activités du visage à partir d'images à attirer une grande attention dans le champ de vision par ordinateur. Parmi eux, la reconnaissance de l'expression qui a été un sujet de recherche active depuis la dernière décennie. Les expressions faciales sont parmi les formes les plus universelles du langage corporel. Elles traduisent l'état émotionnel d'un individu par rapport aux observateurs. En général, les chercheurs ont divisé les expressions faciales en six grandes catégories : la colère, le dégoût, la peur, la joie, la tristesse et la surprise ; qui sont aussi appelés émotions primaires.

La reconnaissance automatique des expressions faciales suit le schéma général d'un processus de la reconnaissance des formes classique qui est défini par les étapes suivantes :

- prétraitement
- extraction des caractéristiques et/ou suivie d'une étape de sélection de caractéristiques pertinentes
- classification et décision.

Plusieurs travaux ont été développés dans le domaine de l'apprentissage automatique sur la reconnaissance des expressions faciales en utilisant plusieurs algorithmes. C'est dans ce contexte qu'on a abordé notre travail. L'objectif de notre travail est de concevoir un système qui détecte les émotions en utilisant les expressions faciales d'une personne.

La description de notre mémoire est comme suit :

Dans le premier chapitre on va expliquer quelques notions de bases liées à notre projet : les émotions, la reconnaissance de visage et ses grandes difficultés. Également, on va aborder les expressions faciales en expliquant les différentes méthodes pour la détection des caractéristiques de ces expressions.

Le second chapitre est tout d'abord consacré à l'apprentissage automatique en général puis aux approches utilisées dans notre travail tels que SVM linéaire et polynomiale, le forêt des arbres décisionnels et méthode des k plus proches voisins KNN.

Dans le troisième chapitre, on présentera la méthodologie du travail en mettant en œuvre notre logiciel de programmation et les bibliothèques nécessaires pour l'implémentation de notre programme en python.

Dans le quatrième chapitre, on a vu comment détecter un visage à partir d'une image de la base de données, l'extraction et la sélection des caractéristiques faciales et d'utiliser les différents classificateurs.

Le dernier chapitre illustre l'implémentation et l'expérimentation de notre système, le test de nos classificateurs avec différents exemples en se basant sur la probabilité de la prédiction de l'état émotionnelle.

Enfin, nous terminerons ce mémoire par une conclusion générale et quelques perspectives.

Chapitre I : Notions de base des émotions et expression facial

Sommaire

I. 1 Introduction

I. 2 Emotions

I. 3 Détection de visage

I. 4 L'expression faciale

I. 5 Conclusion

I.1 Introduction :

Dans la complexité continue de la vie sociale les émotions jouent un rôle clé dans la définition et la régulation de nos relations avec les autres et plus généralement avec l'environnement qui nous entoure. Nos réactions émotionnelles à d'autres personnes influencent sur la façon dont ces autres personnes réagissent à nous et dans une certaine mesure

La plupart des systèmes d'analyse des expressions faciales tentent de classer les expressions en quelques grandes catégories émotionnelles, telles que la joie, la tristesse, la colère, la surprise, la peur et le dégoût.

I.2 Emotions :

I.2.1 Définition :

Les émotions, de façon générale, sont des états motivationnels qui sont constituées d'impulsions, de désirs ou d'aversion, elles comportent des changements de motivation. Ces émotions poussent l'individu à modifier sa relation avec un objet, un état du monde, un état de soi, ou à maintenir une relation existante malgré des obstacles ou des interférences



Figure I.1: famille exprimant de la joie [1]

D'un point de vue cérébral (médical), les émotions sont gérées par l'hippocampe (dans le système limbique), qui enregistre le contexte des événements, et par l'amygdale, qui retient la connotation émotionnelle qui leur est attachée [2], les émotions apparaissent comme des modes de fonctionnement du cerveau permettant de réagir rapidement face à une décision à prendre. Nous pouvons donc remarquer qu'un processus d'émotion a un module de réaction, un module de réflexion et un module d'apprentissage et de mémorisation [3].

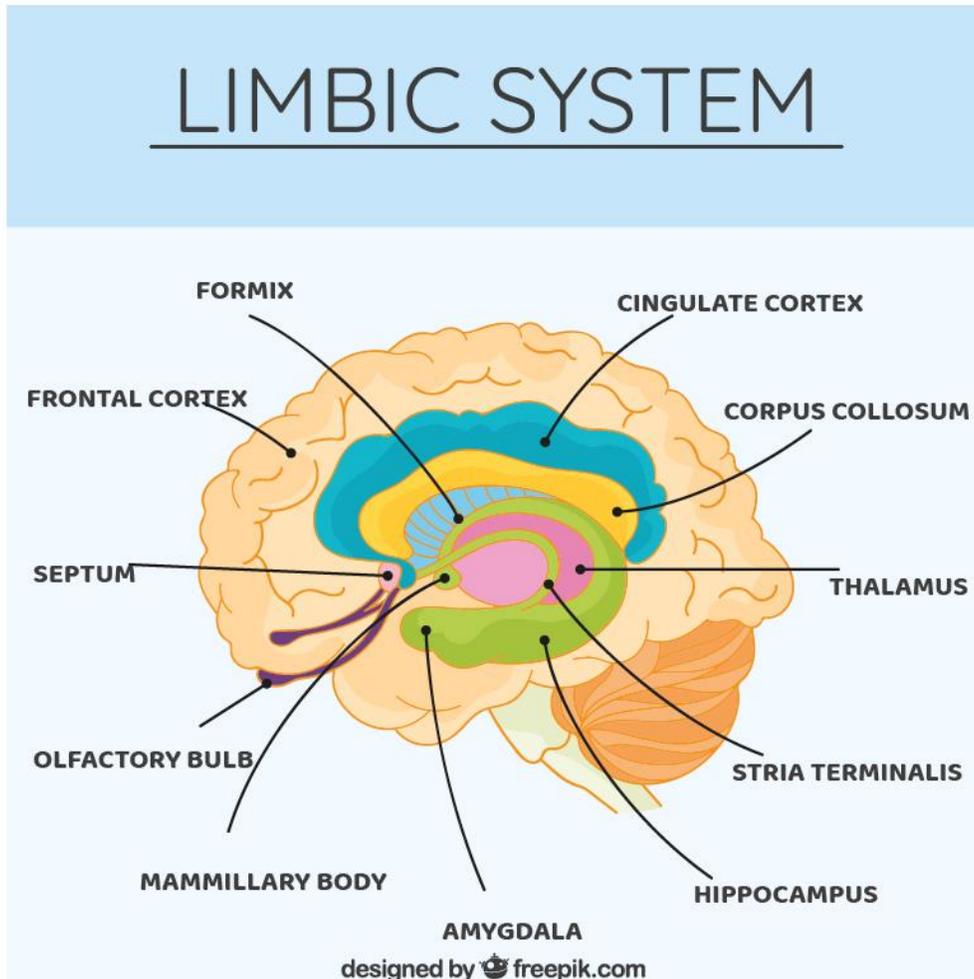


Figure I.2: Représentation du système limbique [4]

I.2.2 La différence entre expressions faciales et émotions :

Les expressions faciales émotionnelles sont des changements faciaux qui traduisent des états émotionnels internes, des intentions ou des communications sociales d'une personne. Il est important de souligner dès le début qu'il y a une distinction d'un point de vue de la vision par ordinateur entre la reconnaissance de l'expression faciale (*REF*) et la reconnaissance des émotions humaines.

L'expression faciale traite la classification du mouvement facial et la déformation des traits faciaux en classes abstraites basées sur des informations visuelles. L'émotion est le résultat de nombreux facteurs différents et peut être révélée sur plusieurs canaux, par exemple, la voix, la pose, les gestes, la direction du regard et l'expression faciale. En effet, la correspondance d'une expression faciale à une émotion implique la connaissance des catégories d'émotions humaines auxquelles des expressions faciales peuvent être attribuées.

I.2.3 Méthodes d'identification des émotions :

On peut reconnaître les émotions par 2 méthodes :

I.2.3.1 Par expression faciale (la méthode directe) :

L'expression faciale est un aspect important du comportement, l'expression faciale est déchiffré par interlocuteur en le lisant tout simplement sur le visage, mais pour la machine ce n'est pas aussi simple.

I.2.3.2 L'acquisition des signaux (la méthode indirecte) :

Aujourd'hui, il est reconnu que les changements physiologiques occupent une place importante lors d'expériences émotionnelles. Cependant, les capteurs physiologiques ne mesurent pas des expressions émotionnelles, mais suivent les marqueurs de l'état somatique qui varie avec l'état émotionnel en détectant les changements de ce dernier.

Signaux physiologiques pour la reconnaissance émotionnelle :

Les indices physiologiques qui sont couramment utilisés pour caractériser les deux composantes valences et intensité, d'une émotion, sont :

L'activité électrodermale (*SKC*) ;

La fréquence cardiaque (*Fc*) ;

Volume sanguin impulsionnel (*BVP*) ;

Le signal du volume respiratoire (*VR*) ;

L'activité électromyographie (*EMG*) ;

La température cutanée (*SKT*) ;

Le rythme électroencéphalogramme (*EEG*).

Dans notre projet on s'intéresse à l'analyse des expressions faciales.

I.3 Détection de visage :

La détection de visage est un domaine de la vision par ordinateur consistant à détecter un visage humain dans une image numérique. C'est un cas spécifique de détection d'objet, où l'on cherche à détecter la présence et la localisation précise d'un ou plusieurs visages dans une image, La forte activité de recherche en détection de visage a également permis de faire émerger des méthodes génériques de détection d'objet[5] .

I.3.1 Les techniques de détection du visage

I.3.1.1 HOG (Histogram of Oriented Gradients):

L'histogramme des gradients orientés (*HOG*) est un descripteur de caractéristiques utilisé pour détecter des objets en vision par ordinateur et en traitement d'image. La technique du descripteur HOG compte les occurrences d'orientation de gradient dans des parties localisées d'une fenêtre de détection d'image ou région d'intérêt (*ROI*). [6]



Figure I.3 : Application de HOG [6]

Image d'entrée et prétraitement :

Dans cette étape on va faire entrer une image, puisqu'une image peut prendre plusieurs tailles on va lui donner des tailles spécifiques (normalisation).

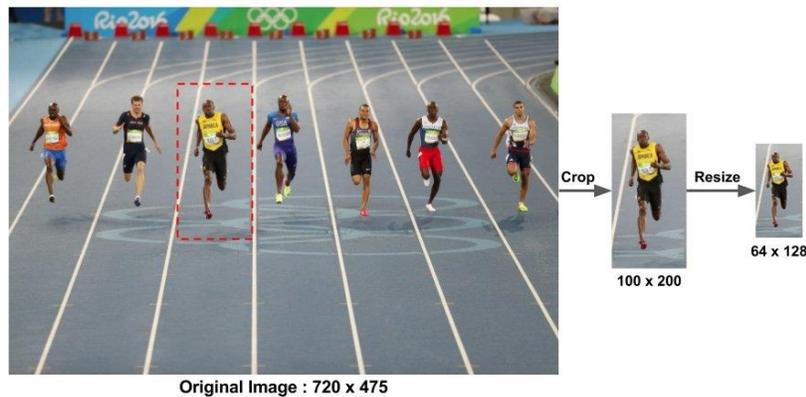


Figure I.4: Extraction et normalisation de l'image [7]

Calculer les gradients :

On va faire deux calculs, la convolution sur l'axe des X ; l'image convoluée avec le vecteur

$$\begin{matrix} -1 & 0 & 1 \end{matrix}$$

Et on calcule la convolution avec l'axe Y ; l'image convoluée avec vecteur

$$\begin{matrix} -1 \\ 0 \\ 1 \end{matrix}$$

Maintenant on calcule le module et la direction :

$$g = \sqrt{g_x^2 + g_y^2} \quad \text{I.1}$$

$$\theta = \tan^{-1}\left(\frac{g_y}{g_x}\right) \quad \text{I.2}$$

Après le calcul du l'image gradient on la divise à des cellules de taille (8x8) pixels pour la réduction du bruit, ensuite on calcule *HOG* de chaque cellule donc chaque histogramme contient 64 valeurs des deux informations (direction, module), les valeurs sont entre 0 et 180 regroupés dans 9 catégories.

Ces données (les histogrammes) sont dirigées vers classificateur linéaire à deux classes (détection du visage /pas de détection du visage).

I.3.2 Les problèmes de détection du visage

I.3.2.1 Influence des variations de la pose :

La position du visage est très importante car l'angle de l'inclinaison du visage et sa rotation influent sur sa détection surtout si le visage n'est pas visible dans son intégrité cela peut poser des problèmes.

I.3.2.2 Influence des changements d'éclairage :

Une mauvaise direction d'éclairage peut faire apparaître l'ombre ceci qui peut masquer quelques caractéristiques.

I.3.2.3 Influence des occultations :

Le port du chapeau et des lunettes peut empêcher la machine de détecter un visage,

*il faut un apprentissage très poussé pour contourner quelques occultations.

I.4 L'expression faciale :

I.4.1 Définition :

Une expression faciale est un ou plusieurs mouvements ou positions des muscles sous la peau du visage. Selon un ensemble de théories, ces mouvements transmettent l'état émotionnel d'un individu aux observateurs. Les expressions faciales sont une forme de communication non verbale.

I.4.2 Description des six expressions faciales :

Lors de la production d'une expression faciale, il apparait sur le visage un ensemble déformation au niveau des traits permanents du visage. [8] [9]

- a) **Neutre** : c'est l'état d'une personne ou les muscles du visage sont tous relaxés.
- b) **Joie** : Elle se caractérise par l'état d'une personne dans une condition de satisfaction intense. Elle est due par rapport au désir, à la réussite, au bien-être, et l'accomplissement, mais aussi l'approche.

Les caractéristiques typiques sur le visage sont :

- Les coins des lèvres tirés vers le haut.
- La bouche est ouverte et les dents sont visibles.
- Les joues sont soulevées.
- Apparition des rides sous la paupière inférieure ainsi qu'aux coins des yeux.

- c) **Tristesse** : c'est l'état d'une personne qui souffre moralement suite à une insatisfaction ou suite à des soucis. Elle est souvent due soit à une perte, ou un deuil, ou un obstacle, la personne se replie souvent sur soi.

Les caractéristiques typiques sur le visage sont :

- Les parties internes des sourcils sont abaissées.
- Les coins des lèvres s'abaissent

- d) **Dégoût** : c'est l'état d'une personne qui a une répugnance pour certains aliments ou à un manque d'appétit. Elle est souvent due soit à un rejet, ou contre quelqu'un, ou à une aversion. La plupart des personnes en ce moment préfèrent se retirer.

Les caractéristiques typiques sur le visage sont :

- La lèvre supérieure est levée, plus le dégoût est grand plus la dentition supérieure est visible.
- Des rides apparaissent sur le nez.

- e) **Colère** : c'est l'état d'une personne dans une réaction violente et agressive lors d'une contrariété. Elle est souvent due soit à une injustice, ou un dommage, atteinte au système de valeurs. La plupart des personnes en ce moment attaquent.

Les caractéristiques typiques sur le visage sont :

Les sourcils s'abaissent et se serrent ensemble, ce qui entraîne l'apparence des rides verticales entre les sourcils.

f) Surprise : c'est l'état d'une personne étonnée par quelque chose d'inattendu. Elle est souvent due soit à un danger immédiat, ou à un imprévu, ou à des inconnus. La plupart des personnes en ce moment préfèrent prendre la fuite ou elles sursautent.

Les caractéristiques typiques sur le visage sont :

- La remontée des parties interne et externe des sourcils.
- Des rides horizontales pouvant apparaître sur le front.
- L'ouverture de la bouche et des yeux.
- Les lèvres sont fermées hermétiquement ou doucement ouvertes (se préparant à crier), l'ensemble de ces caractéristiques préparent souvent le corps à une éventuelle attaque physique ou verbale.

g) Peur : c'est l'état d'une personne menacée par un danger réel ou imaginaire, elle est souvent due soit à une menace, ou à un danger, ou à des inconnus. La plupart des personnes en ce moment préfèrent prendre la fuite.

Les caractéristiques typiques sur le visage sont :

- Les yeux sont ouverts et les pupilles deviennent larges.
- Les sourcils sont soulevés et tirés vers l'intérieur.
- Les paupières supérieures sont soulevées, ce qui a pour effet d'afficher un *Sanpaku* supérieur (c.-à-d., un blanc des yeux visible).
- Les muscles de la bouche se tendent et abaissent souvent la lèvre inférieure.

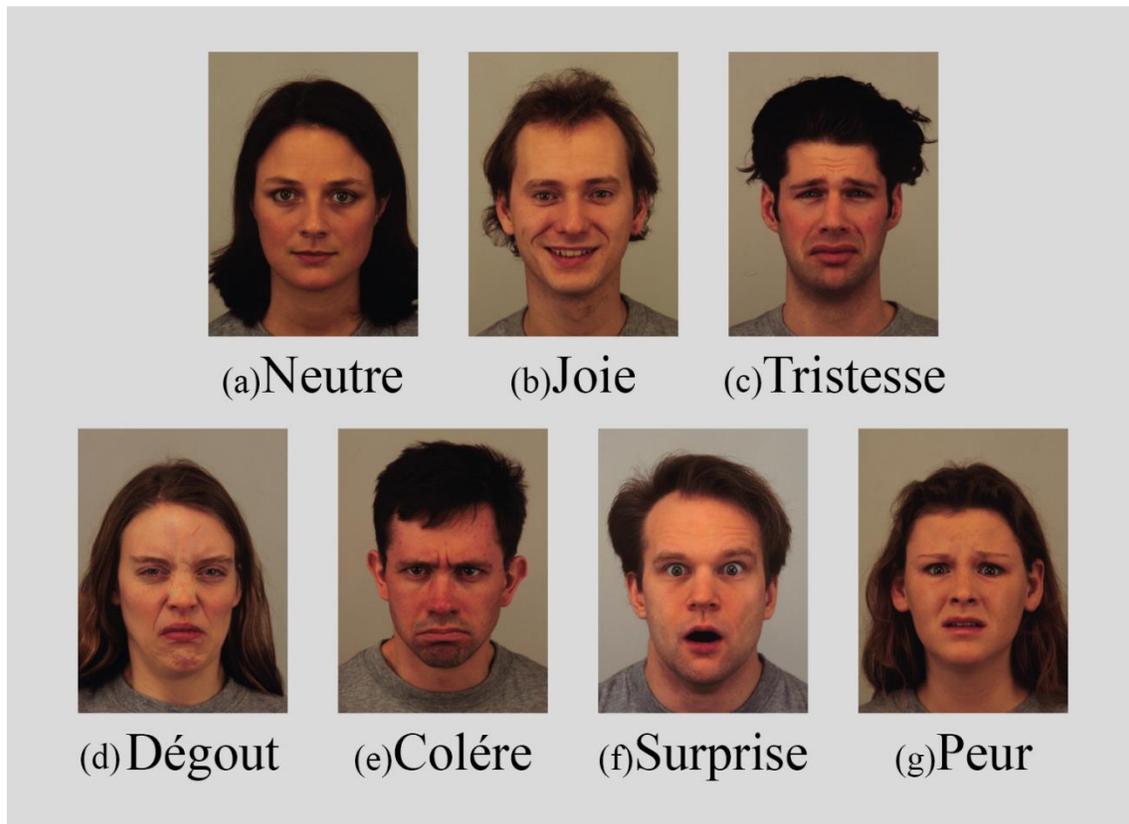


Figure : Les différentes expressions faciales [10]

I.4.3 Les différentes méthodes pour la détection des caractéristiques [25] :

I.4.3.1 Les méthodes globales :

Dans cette méthode tout le visage est considéré comme une caractéristique (un vecteur de pixels), à condition que l'image soit normalisée, l'avantage de cette méthode c'est le traitement et le calcul prennent beaucoup moins de temps, en revanche les problèmes de détection qu'on a cités auparavant ont beaucoup d'influence que les autres méthodes.

I.4.3.2 Les méthodes locales :

Ces méthodes reposent sur l'extraction des paramètres d'un visage (distances ou angles) ces méthodes prennent en compte la non linéarisation des données, et aussi permettent d'avoir une bonne précision et une bonne justesse dans l'apprentissage automatique, mais elles prennent beaucoup de temps dans les calculs.

ceci est un modèle pré-entraîné [11] qui estime la localisation des 68 coordonnées (x,y) ceci est décrit dans la figure suivante :

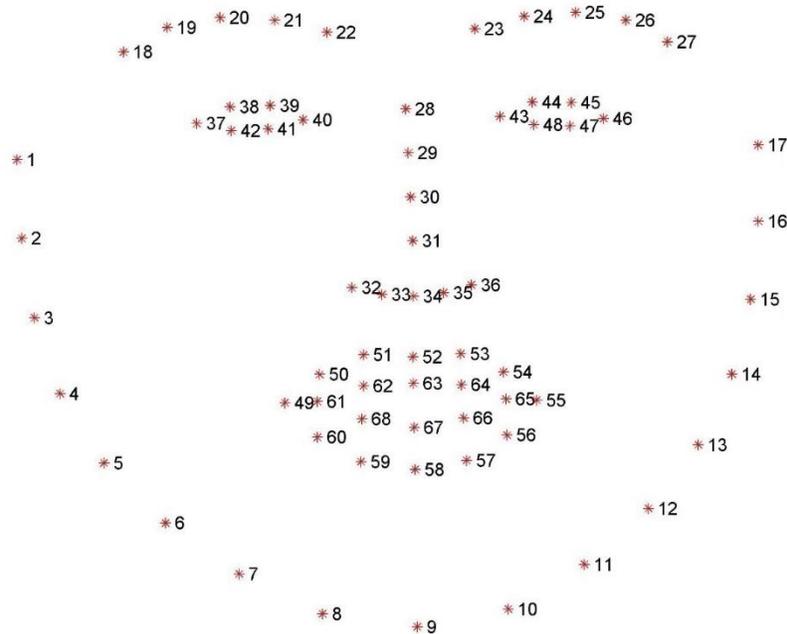


Figure I.5 : Représentation des 68 points de repère

Remarque : Ce modèle est entraîné sur une base des données (*i BUG 300-W face landmark dataset*) [12]

I.4.3.3 Les méthodes hybrides :

C'est une combinaison des méthodes globales et locales cette combinaison permet de faire face au problème de la variation de la pose.

I.5 Conclusion :

Maintenant qu'on a vu ce qu'est les émotions, l'expression faciale, les techniques et les problèmes de détection du visage, et enfin les méthodes de détection de caractéristiques, maintenant on va aborder l'apprentissage automatique (*Machine Learning*).

Chapitre II : Apprentissage automatique (*Machine Learning*)

Sommaire

II. 1 Introduction

II. 2 Définition de l'apprentissage automatique (*MACHINE LEARNING*)

II. 3 Les étapes de l'apprentissage automatique

II. 4 Types d'apprentissage automatique

II. 5 Algorithmes d'apprentissage automatique

II. 6 Conclusion

II.1 Introduction :

L'apprentissage est une technique d'intelligence artificielle permettant à une machine d'apprendre à partir d'exemples déjà disponibles, dans ce chapitre on des différentes étapes que doit passer l'apprentissage automatique, et aussi les types, les algorithmes, enfin on va voir la technique d'entraînement qu'on a utilisé avec la validation croisée et la classification avec plusieurs modèles.

II.2 Définition de l'apprentissage automatique (MACHINE LEARNING) :

L'apprentissage automatique est un élément important de l'intelligence artificielle que les deux termes sont souvent utilisés de manière interchangeable. En réalité, l'apprentissage automatique est un sous-ensemble de l'intelligence artificielle. L'intelligence artificielle comprend toutes les technologies, y compris l'apprentissage automatique, qui permettent à la machine d'imiter l'intelligence humaine. L'intelligence artificielle et l'apprentissage automatique sont des concepts différents, mais liés l'un à l'autre.

Pensons à l'intelligence artificielle et à l'apprentissage automatique de la manière suivante : l'intelligence artificielle est un problème, alors que l'apprentissage automatique fait partie des solutions possibles. Si le but est de faire en sorte qu'un ordinateur puisse résoudre un problème avec les capacités cognitives de l'intelligence (humaine), le traitement d'algorithmes au moyen de données fait partie des moyens d'y parvenir.

L'apprentissage automatique enseigne aux ordinateurs à faire ce qui est naturel chez un humain et chez un animal : apprendre de l'expérience. Les algorithmes d'apprentissage automatique se servent de méthodes de calcul pour « apprendre » directement à partir de données sans utiliser une équation prédéterminée comme modèle.

Les algorithmes améliorent leurs performances par l'adaptation à mesure que le nombre d'échantillons disponibles pour l'apprentissage augmente. La machine apprend directement à partir de données d'apprentissage (exemples) sans être dirigée par des instructions de programmation explicites. Par conséquent, plus les données sont nombreuses, meilleur est le résultat.

Les algorithmes d'apprentissage automatique recherchent des modèles naturels dans les données afin de produire des informations et de pouvoir prendre de meilleures décisions et faire de meilleures prédictions. Ils servent chaque jour à prendre des décisions critiques en matière de diagnostic médical, d'opérations d'actions, de prévisions des charges énergétiques, etc. [13]

On peut résumer la définition de la machine Learning dans cette citation :

« On dit qu'un programme apprend d'une expérience E pour une classe de tâches donnée T et pour une mesure de performance donnée P si sa performance à accomplir les tâches dans T, mesurée par P, s'améliore avec l'expérience E. » [13]

Tom Mitchell (Machine Learning 1997)

II.3 Les étapes de l'apprentissage automatique :

- a) **Acquisition des données** : La première étape consiste à obtenir des données pertinentes pour l'application à développer. Les données doivent être de grande qualité et détaillées.
- b) **Préparation des données** : Cette étape est également appelée nettoyage des données. Les données doivent être précises, propres et sécurisées.
- c) **Sélection de l'algorithme** : L'algorithme le plus approprié pour l'application à développer doit être choisi.
- d) **Entraînement du modèle** : Un modèle d'apprentissage automatique est une représentation mathématique d'un processus réel. L'algorithme retenu doit être entraîné sur les données pour créer le modèle. Le processus d'entraînement peut être supervisé, non supervisé ou renforcé. La description détaillée de ce processus est fournie dans d'autres sections de la présente monographie.
- e) **Évaluation** : Le modèle doit être évalué pour s'assurer que l'algorithme retenu est le mieux adapté.
- f) **Déploiement** : Il faut décider si le modèle doit être déployé dans le nuage informatique ou sur place.
- g) **Test** : Le modèle doit être testé avec des données nouvelles et pour faire des prédictions.
- h) **Évaluation** : La validité des prédictions établies par le modèle doit être évaluée, et le raffinement des données, du modèle et de l'algorithme doit être mis en œuvre selon qu'il convient.

II.4 Types d'apprentissage automatique [13] :

Il existe cinq types courants d'apprentissage automatiques :

II.4.1 Apprentissage supervisé :

Les algorithmes d'apprentissage supervisé font des prévisions en fonction des exemples, par exemple un historique de vente pour déterminer des prix futurs. Dans un tel cas, il y a une variable d'entrée composée de données d'entraînement étiquetées et d'une variable de sortie souhaitée. Un algorithme est utilisé pour analyser les données d'entraînement afin d'apprendre la fonction qui associe l'entrée à la sortie. Cette fonction permet de procéder à une mise en correspondance de nouveaux exemples en généralisant à partir des données d'entraînement pour anticiper les résultats de situations non connues. L'apprentissage supervisé est donc un apprentissage à partir d'exemples. L'apprenant (en général un programme informatique) reçoit deux ensembles de données, le premier pour l'entraînement et le second pour les tests. Le processus est le suivant : l'ensemble de données d'entraînement doit « apprendre » à partir d'un

ensemble d'exemples étiquetés afin qu'il puisse identifier des exemples non étiquetés dans l'ensemble de données de test avec la plus grande précision possible.

a. Classification : Lorsque les données servent à prédire une variable catégorielle, l'apprentissage supervisé est également appelé classification. C'est le cas par exemple lorsqu'une étiquette ou un indicateur (par exemple « chien » ou « chat ») est attribué à une image. Lorsqu'il n'y a que deux étiquettes, on parle de classification binaire. Lorsqu'il y a plus de deux catégories, on parle de classification en classes multiples lorsqu'il y a plusieurs catégories

b. Régression : Lorsqu'on procède à la prédiction de valeurs continues, on parle de régression.

c. Prévion : Il s'agit de faire des prédictions à partir de données passées et présentes. Ce type de processus sert le plus souvent à analyser des tendances, p. ex. estimer les ventes de l'année prochaine à partir des ventes de l'année en cours et des années précédentes.

II.4.2 Apprentissage semi-supervisé :

Dans l'apprentissage supervisé, l'étiquetage des données peut être long et coûteux. Si les étiquettes sont limitées, il est possible d'utiliser des exemples non étiquetés pour améliorer l'apprentissage supervisé. Étant donné que la machine n'est pas entièrement supervisée, on emploie le terme « semi-supervisé ». En ce qui concerne l'apprentissage semi-supervisé, on utilise des exemples non étiquetés et une petite quantité de données étiquetées pour améliorer la précision de l'apprentissage.

II.4.3 Apprentissage non supervisé :

Dans l'apprentissage non supervisé, la machine reçoit des données non étiquetées. On lui demande de découvrir les schémas qui sous-tendent les données, p. ex. une structure en grappes, une variété en basses dimensions, ou un arbre et un graphique de faible densité.

II.4.4 Apprentissage par renforcement :

L'apprentissage par renforcement permet d'analyser et d'optimiser le comportement d'un agent en fonction du retour d'informations de l'environnement. Les machines essaient différentes situations pour déterminer les actions les plus avantageuses, plutôt que de simplement recevoir des instructions sur les actions à entreprendre. Ce qui distingue l'apprentissage par renforcement des autres techniques, ce sont l'apprentissage par essais et erreurs et la récompense différée. L'apprentissage par renforcement est un modèle d'apprentissage comportemental. L'algorithme reçoit des informations grâce à l'analyse des données, de sorte que l'utilisateur est orienté vers le meilleur résultat. L'apprentissage par renforcement diffère des autres types d'apprentissage supervisé, car le système n'est pas entraîné à partir d'un ensemble de données : il apprend par

essais et erreurs. Par conséquent, une série de décisions a pour effet de « renforcer » le processus, car celui-ci convient le mieux pour résoudre le problème.

II.5 Algorithmes d'apprentissage automatique :

II.5.1 Machine à vecteurs de support (SVM) (apprentissage supervisé – classification) :

C'est un classificateur très puissant dans la reconnaissance faciale. La machine à vecteurs de support peut être utilisée à la fois pour les tâches de régression et les tâches de classification. Il est cependant largement utilisé à des fins de classification. L'algorithme de la machine à vecteurs de support vise à trouver un hyperplan dans un espace à n-dimensions (N = nombre de caractéristiques) qui classe distinctement les points de données. [13]

Principe :

SVM est parmi les méthodes de classification les plus connues. Il est basé sur la théorie d'apprentissage statistique [14]. Étant donné une donnée d'entraînement étiquetée

$$y_i, i = 1, \dots, p, y_i \in \{-1, 1\}, x_i \in R \quad \text{II.3}$$

L'algorithme SVM construit un hyperplan avec la plus grande marge qui sépare les exemples positifs et négatifs. Tous les points situés d'un côté de l'hyperplan sont étiquetés comme 1, et tous les points situés de l'autre côté sont étiquetés comme -1. Les points les plus proches du plan de séparation de données sont appelés vecteurs de support. SVM peut alors être défini comme :

$$f(x_i, W, b) = x_i * W + b \quad \text{II.4}$$

Où :

W : Matrice des poids ; x_i vecteur d'entre ; b vecteur de biais

On utilise la fonction de perte pour ajuster les poids W , et les biais b selon l'équation suivante :

$$L = \underbrace{\frac{1}{N} \sum_i \sum_{j \neq y_i} \max(0, f_j - f_{y_i} + 1)}_{\text{data loss}} + \lambda \underbrace{\sum_k \sum_l W_{k,l}^2}_{\text{regularization loss}} \quad \text{II.5}$$

Où :

N : Nombre des exemples dans la base des données

λ : Hyper paramètre qui contrôle le taux de la régularisation

Utilisation des noyaux :

Le fait d'admettre mal-classification de certains exemples, ne peut pas toujours donner une bonne généralisation pour un hyperplan même si ce dernier est optimisé

La détermination d'une fonction parfaite non linéaire est très difficile voire impossible. Pour cela les données sont amenées dans un espace où cette fonction devient linéaire (figure 2.5), cette astuce permet de garder les mêmes modèles de problèmes d'optimisation vus dans les sections précédentes, utilisant les *SVMs* basés essentiellement sur le principe de séparation linéaire. Cette transformation d'espace est réalisée souvent à l'aide d'une fonction $F = \{\phi(x)|x \in X\}$ appelé "*Mapping function*" et le nouvel espace est appelé espace de caractéristiques "*Features space*".

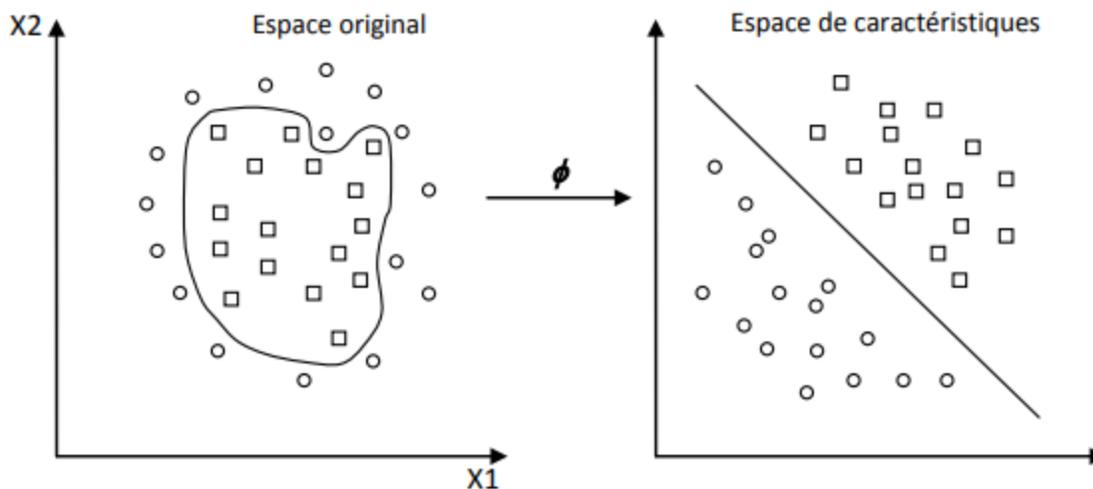


Figure 3.1 : Transformation d'espace

Dans le calcul de l'optimum de la fonction (2.28), on utilise une astuce appelée "Noyau" (*Kernel*), au lieu de calculer $\phi(x_i)$, $\phi(x_j)$ et leur produit scalaire, on calcule plutôt une fonction $K(x_i, x_j)$ qui représente à la fois les deux transformations (qui peuvent être inconnues) et leur produit scalaire. Cette fonction permet de surmonter le problème de détermination de la transformation ϕ et permet d'apprendre des relations non linéaires par des machines linéaires. La fonction $K(x_i, x_j)$ peut être vue comme une matrice $G [n, n]$ dite de Gram [15] qui représente les distances entre tous les exemples :

$$\begin{pmatrix} K(x_1, x_1) & \cdots & K(x_1, x_n) \\ \vdots & \ddots & \vdots \\ K(x_n, x_1) & \cdots & K(x_n, x_n) \end{pmatrix} \quad \text{II.6}$$

Les noyaux les plus utilisés sont :

Noyau linéaire :

Si les données sont linéairement séparables, on n'a pas besoin de changer d'espace, et le produit scalaire suffit pour définir la fonction de décision : [16]

$$K(x_i, x_j) = x_i^T x_j \quad \text{II.7}$$

Noyau polynomial :

Le noyau polynomial élève le produit scalaire à une puissance naturelle d : [16]

$$K(x_i, x_j) = (x_i^T x_j)^d \quad \text{II.8}$$

Si $d = 1$ le noyau devient linéaire. Le noyau polynomial dit non homogène $K(x_i, x_j) = (x_i^T x_j + C)^d$ est aussi utilisé

Noyau RBF :

Les noyaux RBF (Radial Basis fonctions) sont des noyaux qui peuvent être écrits sous la forme : [19]

$$K(x_i, x_j) = f(d(x_i, x_j)) \quad \text{II.9}$$

Où d est une métrique sur X et f est une fonction dans :

$$K(x_i, x_j) = e\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right) \quad \text{II.10}$$

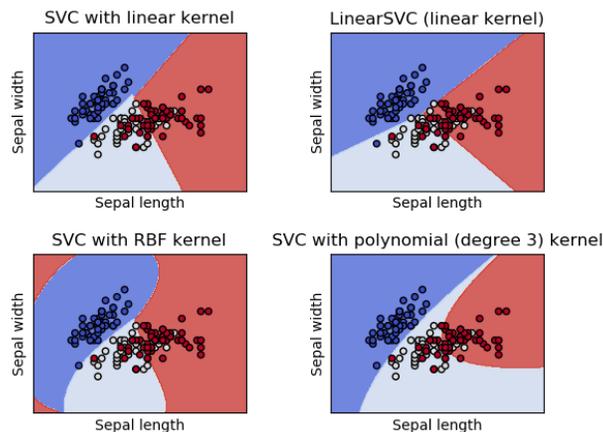


Figure II.1 : Représentation des noyaux

II.5.2 Méthode des k plus proches voisins :

En intelligence artificielle, plus précisément en apprentissage automatique, la méthode des k plus proches voisins est une méthode d'apprentissage supervisé. En abrégé k -NN ou KNN, de l'anglais *k-nearest Neighbors*.

Dans ce cadre, on dispose d'une base de données d'apprentissage constituée de N couples « entrée-sortie ». Pour estimer la sortie associée à une nouvelle entrée x , la méthode des k plus proches voisins consiste à prendre en compte (de façon identique) les k échantillons d'apprentissage dont l'entrée est la plus proche de la nouvelle entrée x , selon une distance à définir.

Par exemple, dans un problème de classification, on retiendra la classe la plus représentée parmi les k sorties associées aux k entrées les plus proches de la nouvelle entrée x [17]

II.5.3 Forêt des arbres décisionnels :

On a déjà expliqué ceci dans le chapitre 2, cette équation ci-dessous permet de mieux décrire le principe de fonctionnement :

$$H(x) = \operatorname{argmax} \sum_{i=1}^k I(h_i(x) = Y) \quad \text{II.11}$$

Où $H(x)$ est le classificateur combiné final, k est le nombre d'arbres de décision, $h_i(x)$ représente un arbre de décision, Y est l'étiquette de classe, $I(h_i(x) = Y)$ indique x appartient à la classe Y [21].

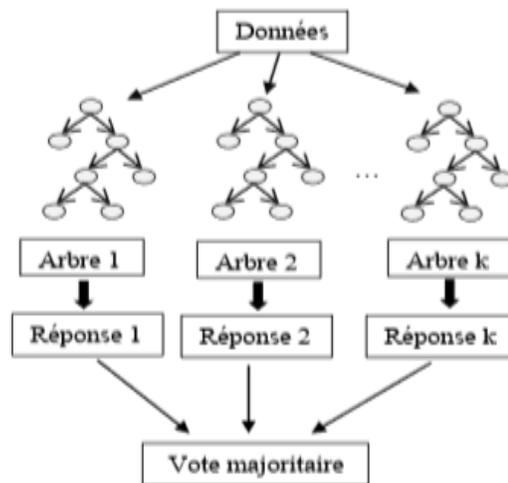


Figure II.2 : Architecture du modèle de Random Forest

II.6 Conclusion :

Après avoir vu les classificateurs d'apprentissage automatique utilisé dans notre travail (SVM, KNN, RFC), en va entamer l'avant-projet qui consiste à la planification du projet et préparation de l'environnement.

Chapitre III : Planification du projet et préparation de l'environnement

Sommaire

III. 1 Introduction

III. 2 Déroulement du projet

III. 3 Préparation d'environnement

III. 4 Conclusion

III.1 Introduction :

Une bonne planification et une bonne préparation sont deux choses qui dépendent de la réussite d'un projet et de la qualité du travail et facilite la vision, dérouler les étapes avant le commencement permet d'identifier la majorité des difficultés rencontrées

III.2 Déroulement du projet :

Les étapes et Le déroulement et la planification de notre projet est expliqué dans la figure 4 :

a) La base de données : c'est là où la qualité du travail se joue c'est un pilier essentiel de notre projet, une mauvaise base de données va causer des dégâts au niveau des résultats

b) Extraction des points de repère du visage : on va extraire l'emplacement des yeux, du nez, de la bouche et des sourcils dans le visage

Extraction des caractéristiques : prendre toutes les combinaisons possibles entre tous les points de repère

c) Sélection automatique des caractéristiques : sélection des meilleures caractéristiques avec le classificateur forêt aléatoire c'est important pour un bon apprentissage de la machine

d) Validation croisée K-FOLD : division des données en l'entraînement et le test avec la méthode K-FOLD

e) Choix du classificateur : choisir un ou plusieurs classificateurs qui s'adaptent à nos données

f) Visualisation de la précision et de la justesse : visualiser la qualité de l'apprentissage

Exemple : Evaluation des différents modèles

Obtenir le résultat de l'exemple : résultat des données sous la forme d'un tableau comparant les différents classificateurs avec le pourcentage de prédiction de chacun d'eux

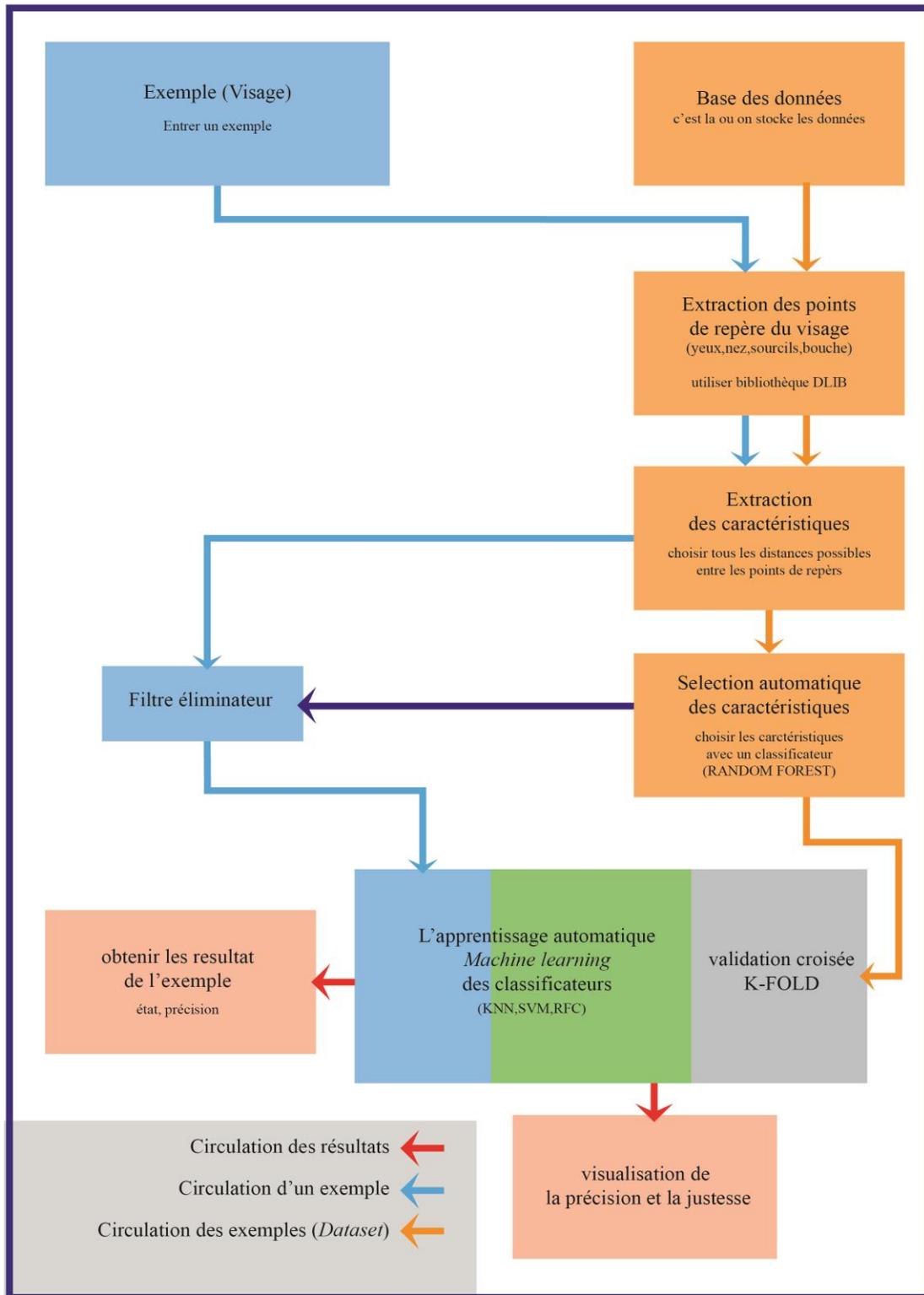


Figure III.1 : Planification du projet

III.3 Préparation d'environnement :

III.3.1 Python :

Python est un langage de programmation interprété, multi-paradigme et multiplateformes

Il favorise la programmation impérative structurée, fonctionnelle et orientée objet [18].



Figure III.2 : Logo de Python

III.3.2 Anaconda :

Anaconda est une distribution gratuite et open-source des langages de programmation Python et R pour le calcul scientifique, parmi ces applications : science des données, applications d'apprentissage automatique, traitement de données à grande échelle, analyse prédictive, etc.

Anaconda facilite la création des environnements sur python et continue plusieurs *Ides* [19]

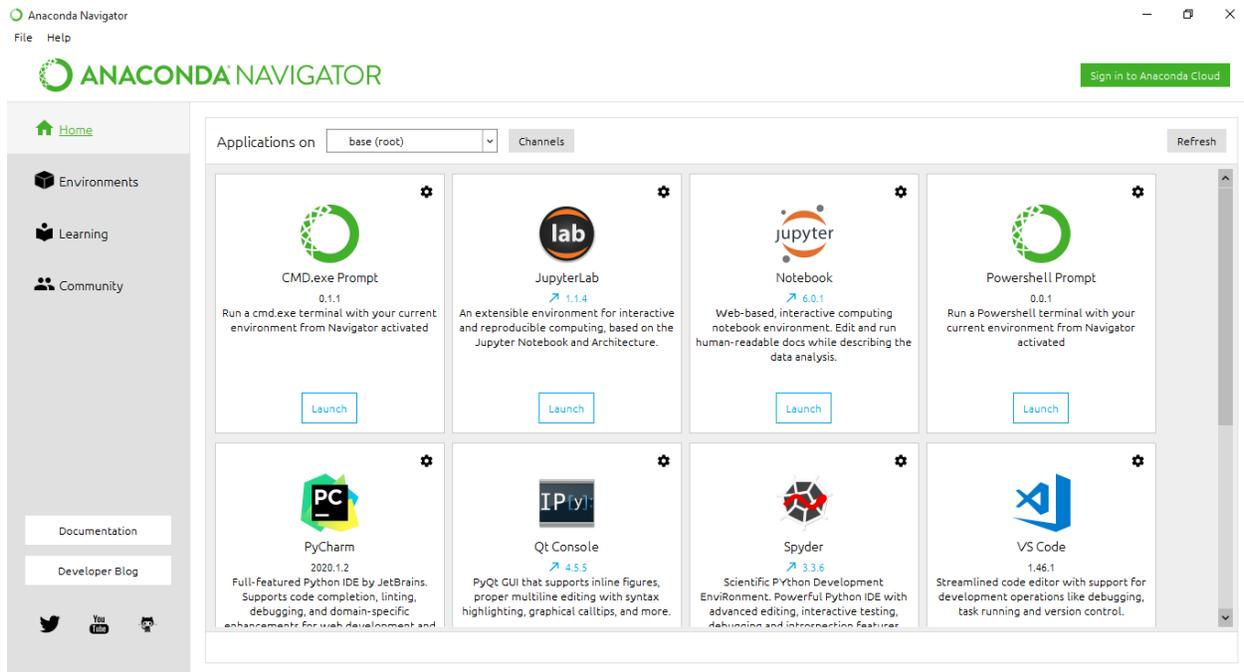


Figure III.3 : Fenêtre d'accueil d'ANACONDA NAVIGATOR

III.3.3 Les Bibliothèques :

III.3.3.1 Open CV :

Open CV est parmi les premières bibliothèques de traitement d'image avec une variété fonctions qui a beaucoup d'avantages sur la méthode de programmation [20].

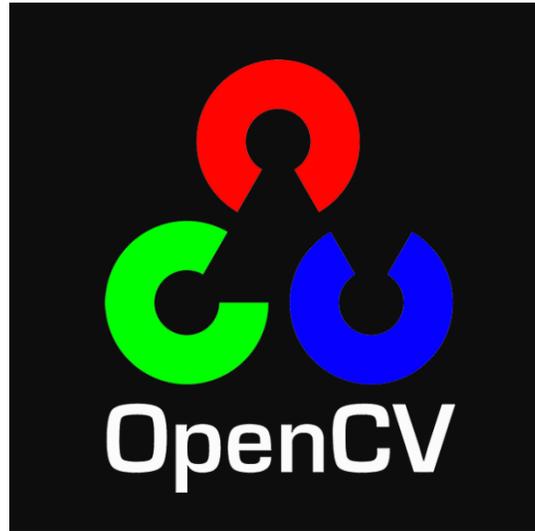


Figure III.4: Logo d'Open CV

III.3.3.2 Scikit-learn :

Scikit-learn est une bibliothèque libre Python destinée à l'apprentissage automatique (Machine Learning). Elle est développée par de nombreux contributeurs notamment dans le monde académique. Elle comprend notamment des fonctions pour estimer des forêts aléatoires, des régressions logistiques, des algorithmes de classification, et les machines à vecteurs de support. Elle est conçue pour s'harmoniser avec d'autres bibliothèques libres Python, notamment NumPy et SciPy. [21]



Figure III.5 : Logo de Scikit-learn

III.3.3.3 NumPy :

C'est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux de nombres, des fonctions sophistiquées (diffusion) [22].



Figure III.6: Logo de NumPy

III.3.3.4 Pickle :

Le Pickle module implémente des protocoles binaires pour sérialiser et dé-sérialiser une structure d'objet Python. «*Pickling*» est le processus par lequel une hiérarchie d'objets Python est convertie en un flux d'octets, et «*unpickling*» est l'opération inverse, par laquelle un flux d'octets (à partir d'un fichier binaire ou d'un objet semblable à des octets) est reconverti en une hiérarchie d'objets. Le décapage (et le décapage) est également appelé «sérialisation», «trriage», 1 ou «aplatissement» ; cependant, pour éviter toute confusion, les termes utilisés ici sont «décapage» et «décapage». On utilise cette pour sauvegarder notre le classificateur après apprentissage. [23]

III.3.3.5 DLIB :

Dlib est une bibliothèque logicielle multiplateforme à usage général écrite dans le langage de programmation C ++. Sa conception est fortement influencée par les idées de la conception par contrat et de l'ingénierie logicielle basée sur les composants. Il s'agit donc avant tout d'un ensemble de composants logiciels indépendants.



Figure III.7: Logo de Dlib [11]

III.3.4 IDE :

III.3.4.1 Visual Studio Code :

VS Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. [24]

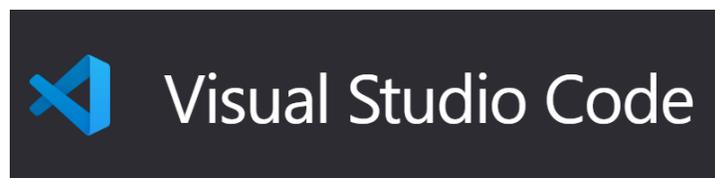


Figure III.8: Logo de VS Code

III.4 Conclusion :

Après avoir entamé la préparation et la planification. Dans le prochain chapitre on va voir comment détecter un visage dans une image et la procédure de l'extraction et la sélection des caractéristiques.

Chapitre IV : Détection faciale, Extraction et sélection des caractéristiques et classification

Sommaire

IV. 1 Introduction

IV. 2 Base des données (*Database*)

IV. 3 Détection du visage

IV. 4 Extraction des points de repère du visage

IV. 5 Extraction des caractéristiques

IV. 6 Sélection des caractéristiques (*Feature Selective*)

IV. 7 Les techniques d'entraînement et de sélection du classificateur

IV. 8 Classification avec plusieurs modèles

IV. 9 Sauvegarde du modèle entraîné

IV. 10 Conclusion

IV.1 Introduction :

La détection automatique des expressions faciales par vision numérique est très complexe, car on doit détecter le visage, après cette opération on doit extraire les caractéristiques, enfin on doit sélectionner les bonnes caractéristiques qui seront utiles pour la suite du travail

IV.2 Base des données (*Database*) :

La base de données est l'ensemble des images stockées dans un support permettant de l'exploiter afin de retrouver des résultats, ces images expriment les diverses émotions, notre base de données s'appelle KDEF [12] cette base de données se compose de 140 images dans chacune des 7 états d'émotions, ce qui fait un total de 980 images

IV.3 Détection du visage :

La fonction introduite dans la bibliothèque *Dlib* :

```
dlib.get_frontal_face_detector ()
```

Intègre le principe de détection du visage avec la méthode *HOG*.

IV.4 Extraction des points de repère du visage :

La fonction pour importer le modèle suivant

```
dlib.shape_predictor ("dlib predictor/shape_predictor_68_face_landmarks.dat")
```

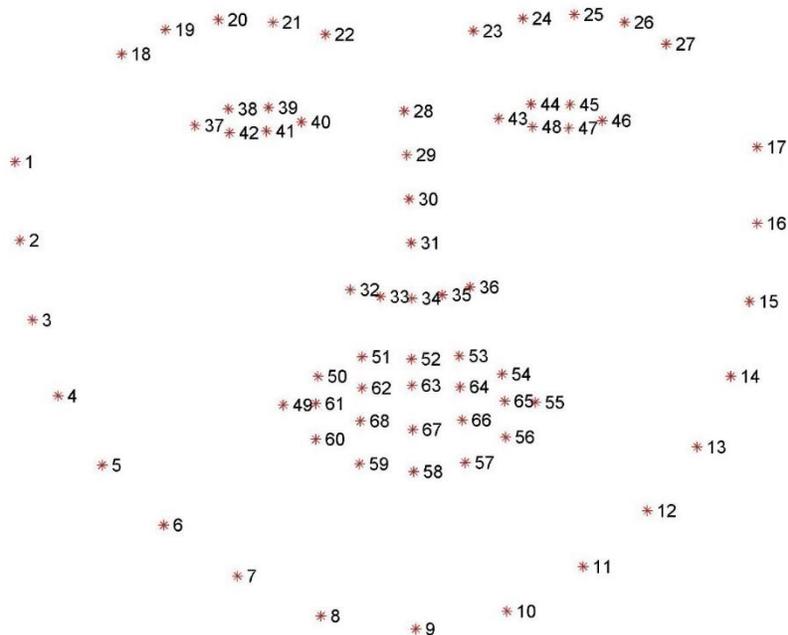


Figure IV.1 : Représentation des 68 points de repère

IV.5 Extraction des caractéristiques :

Après avoir tiré les points de repère du visage on va maintenant calculer tous les distances euclidiennes possibles entre le point 18 et le point 68.

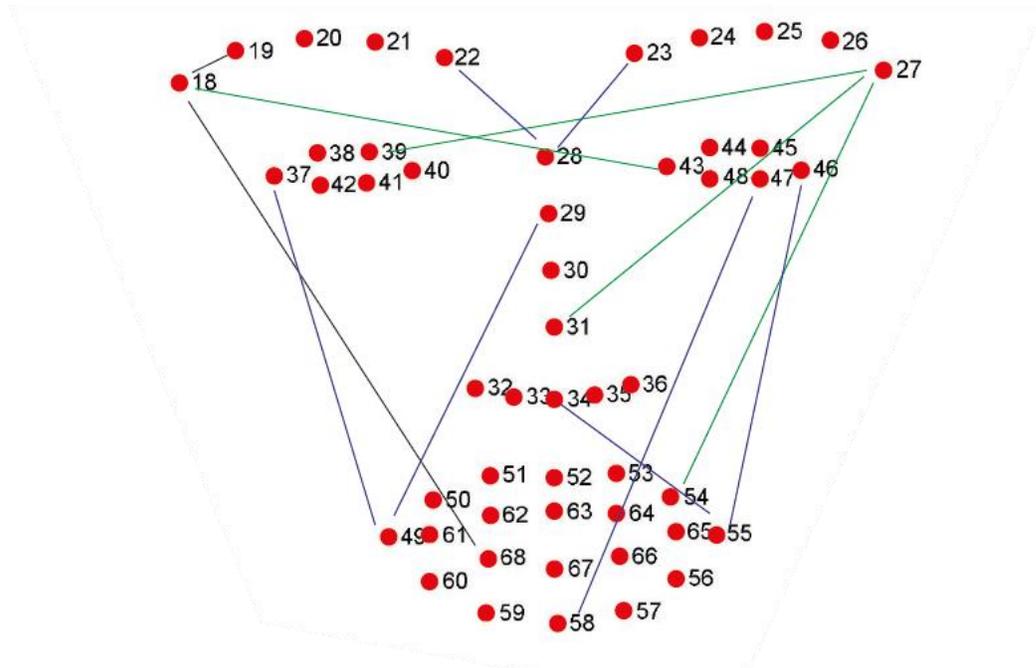


Figure IV.2 : Représentation de quelques distances (des exemples)

En combinant tous les points qu'on décrit précédemment on a obtenu 1275 distances, la plupart de ces distances sont redondants et cette redondance diminue la précision et la justesse du classificateur, c'est pour cela qu'on va procéder à la phase suivante.

IV.6 Sélection des caractéristiques (*Feature Selective*) :

On a utilisé le classificateur *Random Forest* pour la sélection automatique des caractéristiques les plus pertinents.

IV.6.1 Le classificateur *Random Forest* :

C'est un classificateur basé sur des centaines d'arbres de décision, chaque arbre est construite sur l'extraction aléatoire des caractéristiques et l'observation du *dataset*, ces arbres de décision ne visualise pas tous les caractéristiques ce qui garantit que les arbres sont dé-corrélés pour éviter le sur-ajustement, chaque arbre a une décision à prendre (oui ou non) sur une ou une combinaison de caractéristique.

```
modèle_tree = RandomForestClassifier()  
modèle_tree.fit(X_train, y_train)
```

```
threshold=[0.003,0.0025,0.002,0.0015,0.001,0.0005,0.0003]  
sel_modèle_tree = SelectFromModèle(estimator=modèle_tree, prefit=True, threshold=  
threshold[0])
```

Dans cette partie du script on a donné à notre modèle toutes les caractéristiques qu'on a extraites précédemment pour l'entraîner, une fois l'entraînement fini, chaque caractéristique a un chiffre qui représente l'importance par rapport aux autres caractéristiques, pour cela on introduit plusieurs seuils pour fixer plusieurs nombres de caractéristiques, le but est d'évaluer les modèles d'apprentissage.

Voici les différents niveaux de caractéristiques choisissés par seuillage d'importance :

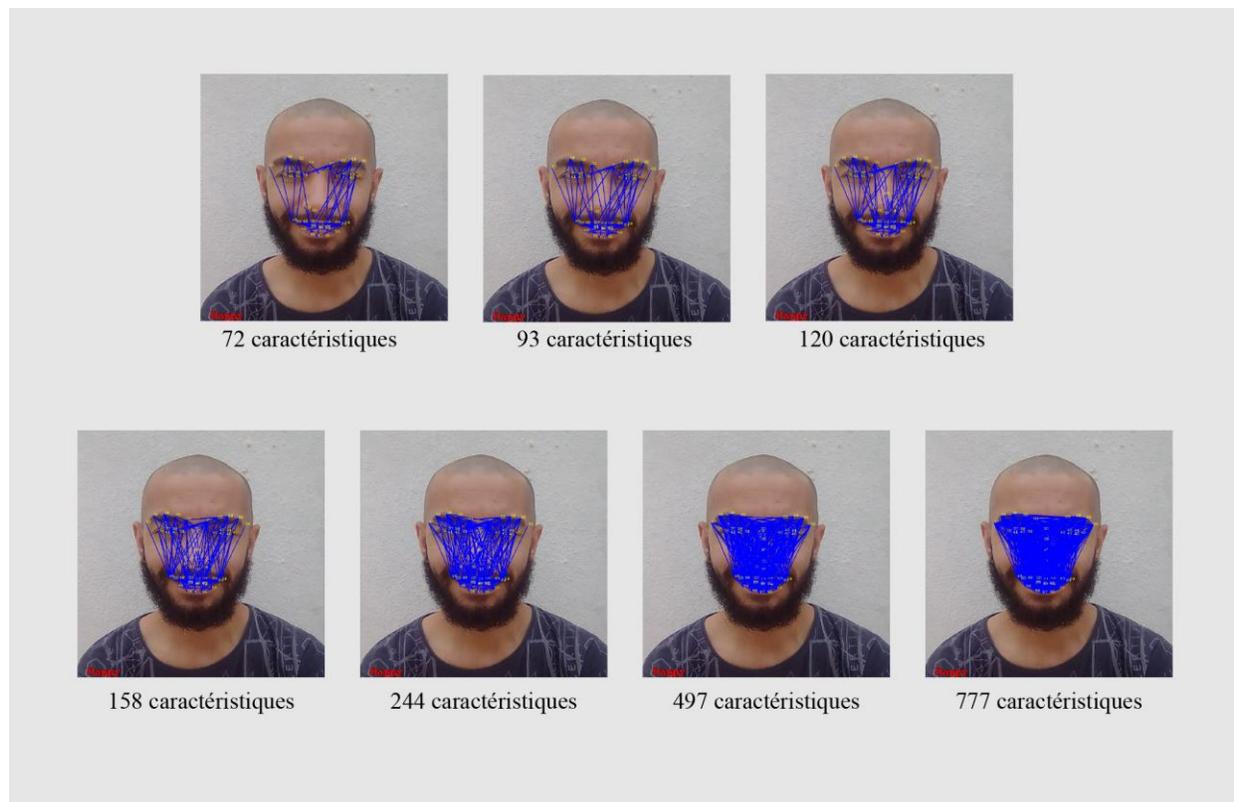


Figure IV.3 : Représentation des caractéristiques par rapport au seuillage d'importance

IV.7 Les techniques d'entraînement et de sélection du classificateur :

IV.7.1 La validation croisée (cross-validation) :

Dans l'apprentissage automatique on a besoin des données d'entraînement pour l'apprentissage du modèle et on a besoin des données de test pour évaluer cet apprentissage pour cela, on va diviser les caractéristiques dans chaque état, pour avoir les meilleurs résultats on a divisé les caractéristiques dans chaque état comme suit selon la méthode de K -fold ($K=10$) :

- 90 % des données comme données d'entraînement
- 10 % des données comme données de test

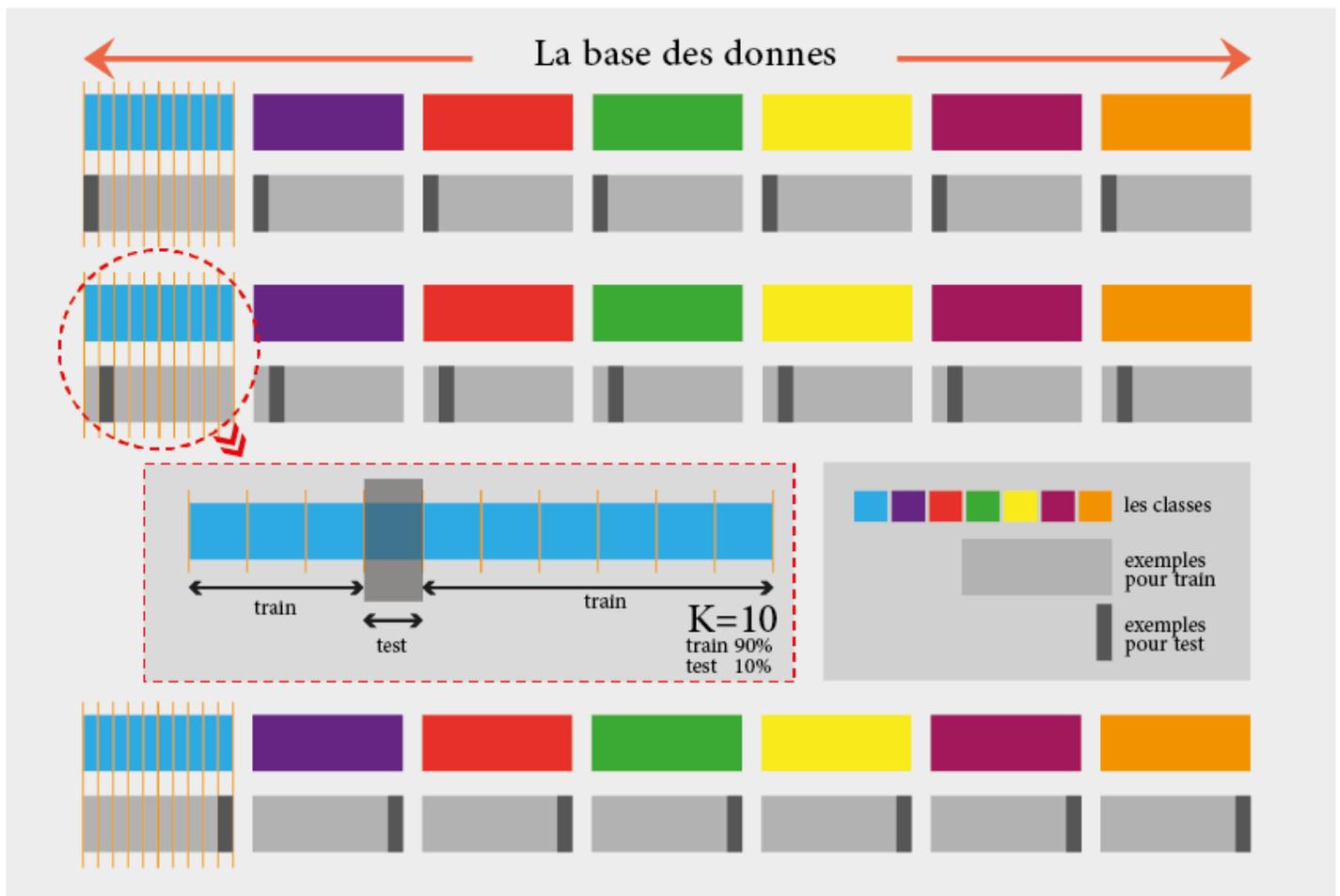


Figure IV.4: Explication de la méthode K-fold

On a répété cette procédure k fois pour obtenir k modèles, le but de cette technique est d'obtenir le modèle le plus performant selon deux critères : précision et justesse.

IV.8 Classification avec plusieurs modèles :

On a pris un exemple de 244 caractéristiques et ont visualisé les résultats (la justesse et précision)

IV.8.1 Avec SVM linéaire :

```
from sklearn.svm import SVC
```

Grâce a la bibliothèque *scikit-learn* on peut créer un modèle SVM linéaire avec des hyper paramètres qui permettent d'adapter le modèle au *dataset* et caractéristiques

```
Modèle = SVC(kernel='linear',C=0.2,probability=True, class_weight='balanced',  
             decision_function_shape='ovr')
```

$C=0.2$: c 'est un paramètre de Régularisation, λ le taux de larégularisation est est inversement proportionnelle à C , le choix du chiffre 0.2 est issu du test et évaluation du modèle.

class_weight = 'balanced' : initialise la matrice du poids de façon aléatoire à condition que leurs moyenne est égal à zéro.

decision_function_shape='ovr' : chaque classe qui ciblé par l'apprentissage est opposé au reste des classes , le nombre de combinaisons possibles est égale au nombres des classes

résultat d'apprentissage :

```
training SVM linear 0  
accuracy: 0.8877551020408163 | precision: 0.8892464678178963  
training SVM linear 1  
accuracy: 0.8469387755102041 | precision: 0.8580786160618094  
training SVM linear 2  
accuracy: 0.8163265306122449 | precision: 0.8416811258916521  
training SVM linear 3  
accuracy: 0.826530612244898 | precision: 0.8346320346320346  
training SVM linear 4  
accuracy: 0.7346938775510204 | precision: 0.740834165834166  
training SVM linear 5  
accuracy: 0.8571428571428571 | precision: 0.8628851540616245  
training SVM linear 6  
accuracy: 0.8367346938775511 | precision: 0.8406087609869123  
training SVM linear 7  
accuracy: 0.7857142857142857 | precision: 0.8107142857142857  
training SVM linear 8  
accuracy: 0.8469387755102041 | precision: 0.8553854875283445  
training SVM linear 9  
accuracy: 0.826530612244898 | precision: 0.8393939393939395  
Mean accuracy SVM: 0.826530612244898  
Mean precision SVM: 0.8373460037922665  
Best SVM modèle 0 with accu of : 88 and preci of :88
```

On choisit le meilleur modèle d'après ces résultats dans ce cas c'est le premier modèle de $K=0$ qui a plus de 88% de justesse et de 88% de précision.

IV.8.2 Avec SVM polynomial :

Dans ce cas ont changé de noyau polynomial avec un degré de six

```
clf = SVC(kernel='poly',C=0.2,degree=6,probability=True,gamma='auto' ,class_weight='balanced',decision_function_shape='ovr')
```

Résultat de l'apprentissage :

```
training SVM poly 0
accuracy: 0.8469387755102041 | precision: 0.8422619047619047
training SVM poly 1
accuracy: 0.8469387755102041 | precision: 0.8575030012004803
training SVM poly 2
accuracy: 0.8061224489795918 | precision: 0.8311797225331061
training SVM poly 3
accuracy: 0.8163265306122449 | precision: 0.8297234045133205
training SVM poly 4
accuracy: 0.7755102040816326 | precision: 0.8046043501012445
training SVM poly 5
accuracy: 0.8367346938775511 | precision: 0.8434407096171801
training SVM poly 6
accuracy: 0.8163265306122449 | precision: 0.8164708307565451
training SVM poly 7
accuracy: 0.8163265306122449 | precision: 0.8325368240781774
training SVM poly 8
accuracy: 0.8061224489795918 | precision: 0.8166874442084526
training SVM poly 9
accuracy: 0.8367346938775511 | precision: 0.8451441102756893
Mean accuracy SVM: 0.8204081632653061
Mean precision SVM: 0.83195523020461
Best SVM modèle 0 with accu of : 84 and preci of :84
```

On choisit le meilleur modèle d'après ces résultats dans ce cas c'est le premier modèle de $K=0$ qui a plus de 84% de justesse et de précision de 84%.

IV.8.3 Avec le forêt aléatoire :

```
clf = RandomForestClassifier(criterion="gini")
```

On a créé un autre modèle cette fois c'est la forêt aléatoire avec la même bibliothèque dans le but de confirmer le résultat des autres classificateurs ou donner un meilleur résultat (précision et justesse).

criterion="gini" : permet de mesurer la fréquence à laquelle un élément choisi au hasard dans l'ensemble serait incorrectement étiqueté

```
accuracy: 0.8571428571428571 | precision: 0.8647340754483612
training RFC 1
accuracy: 0.7244897959183674 | precision: 0.7384853241996099
training RFC 2
accuracy: 0.7653061224489796 | precision: 0.7861175589116765
training RFC 3
accuracy: 0.7448979591836735 | precision: 0.7573609348045437
training RFC 4
accuracy: 0.7551020408163265 | precision: 0.7575232724668814
training RFC 5
accuracy: 0.8571428571428571 | precision: 0.8658971280820018
training RFC 6
accuracy: 0.7244897959183674 | precision: 0.731283422459893
training RFC 7
accuracy: 0.7551020408163265 | precision: 0.7806659505907627
training RFC 8
accuracy: 0.7755102040816326 | precision: 0.7978355714755538
training RFC 9
accuracy: 0.8163265306122449 | precision: 0.8382859621581426
Mean accuracy RFC: 0.7775510204081632
Mean precision RFC: 0.7918189200597426
Best SVM modèle 0 with accu of : 85 and preci of :86
```

On choisit le meilleur modèle d'après ces résultats dans ce cas c'est le premier modèle de $K=0$ qui a plus de 77% de justesse et de précision de 79%.

IV.8.4 Avec KNN :

```
knn = neighbors.KNeighborsClassifier(n_neighbors=10)
```

On va importer le classificateur *KNN* avec la même bibliothèque *Scikit learn*

```
knn = neighbors.KNeighborsClassifier(n_neighbors=6)
```

n_neighbors : c'est le nombre des classes voisins au a la classe qui est en face d'apprentissage

```

training knn 0
accuracy: 0.7653061224489796 | precision: 0.7892006802721089
training knn 1
accuracy: 0.7857142857142857 | precision: 0.7905045351473924
training knn 2
accuracy: 0.6836734693877551 | precision: 0.6939849624060149
training knn 3
accuracy: 0.7142857142857143 | precision: 0.7534409689689193
training knn 4
accuracy: 0.7551020408163265 | precision: 0.7707142857142857
training knn 5
accuracy: 0.7857142857142857 | precision: 0.8075139146567717
training knn 6
accuracy: 0.7448979591836735 | precision: 0.7563258636788049
training knn 7
accuracy: 0.6938775510204082 | precision: 0.7194414607948442
training knn 8
accuracy: 0.6632653061224489 | precision: 0.6960389610389611
training knn 9
accuracy: 0.7653061224489796 | precision: 0.8210600907029478
Mean accuracy knn: 0.7357142857142858
Mean precision knn: 0.759822572338105
Best SVM modèle 1 with accu of : 78 and preci of :79

```

On choisit le meilleur modèle d'après ces résultats dans ce cas c'est le premier modèle de K=0 qui a plus de 73% de justesse et de précision de 75%.

Classificateur/caractéristiques	72	93	120
SVM linéaire (justesse/précision)	80,9/82,1	81,7/82,9	82/83,1
SVM polynomiale	81/82,4	82/83,4	82,4/83,7
KNN	73,7/75,4	74,5/76,7	74,7/76,8
RFC	80,4/81,7	80,7/81,7	80,3/81,7

Classificateur/caractéristiques	158	244	497	777
SVM linéaire (justesse/précision)	81/82,3	82,7/83,7	81,8/83,3	82,3/83,8
SVM polynomiale	82,9/84,3	83,3/84,8	83,4/84,4	83/84,3
KNN	74,5/77,2	74,7/77,5	75,3/77,6	73/74,8
RFC	80,1/81,3	80,4/81,8	81,2/82,4	80,6/82

Tableau IV.1 : La précision et la justesse des classificateurs

Ce tableau contient la justesse et la précision moyenne en prenant les différents nombres des caractéristiques selon plusieurs modèles on a constaté que le nombre de caractéristiques 497 donne de meilleur précision et justesse.

IV.9 Sauvegarde du classificateur entraîné :

Après l'apprentissage il faut sauvegarder le modèle, il y a plusieurs méthodes pour cette opération par exemple Pickle simple qui simple à utiliser :

```
filename = 'modèles/modèleSVM.sav'  
pickle.dump(clf, open(filename, 'wb'))
```

Les modèles sont sauvegardés dans des fichiers sous format (.sav, modèleSVM_linear.sav)

IV.10 Conclusion :

Dans ce chapitre on a vu comment faire une détection du visage grâce à une fonction de la bibliothèque *DLIB*, ensuite on a vu comment avoir les points de repère avec HOG, et enfin l'extraction des caractéristiques grâce à un code python et la sélection des caractéristiques avec le classificateur de forêt aléatoire, maintenant on va tester et comparer nos classificateurs avec quelques exemples.

Chapitre V : Implémentation et résultats expérimentaux

Sommaire

V. 1 Introduction

V. 2 Extraction et sélection des caractéristiques de l'exemple du test

V. 3 Les prédictions des probabilités du chaque classificateur

V. 4 Les résultats finaux

V. 5 Autres exemples de test

V. 6 Conclusion

V.1 Introduction :

Maintenant on a terminé la phase de la réalisation du projet on va maintenant tester nos classificateurs avec des exemples

V.2 Extraction et sélection des caractéristiques de l'exemple du test :

V.2.1 Lire l'image :

Avec l'aide de la bibliothèque *Open CV* on va lire une image (figure ci-dessous) :

```
path='exemple/Nouveau dossier/'  
filen='test'  
typee='.jpg'  
face = cv2.imread(path+filen+typee)
```

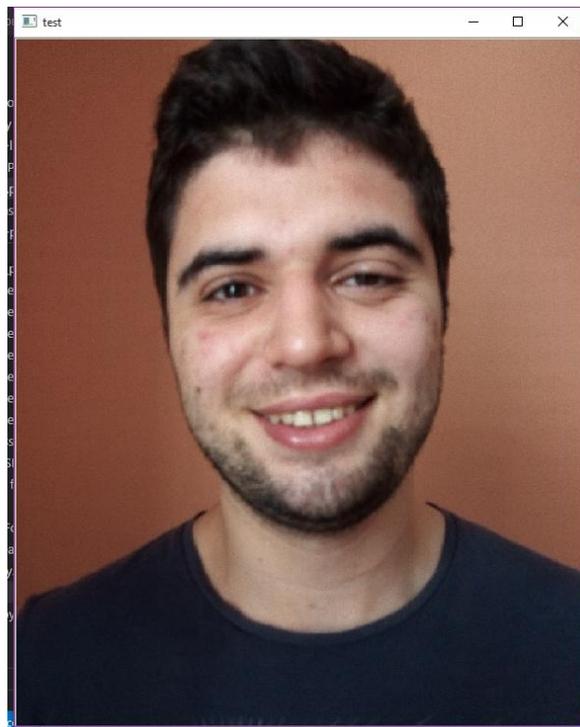


Figure V.1 : Image de test

V.2.2 Extraction et sélection :

On va appliquer la fonction vue précédemment (chapitre III) avec la bibliothèque *DLIB* pour l'extraction des points de repère et l'extraction des caractéristiques avec la même procédure que la base de données.

Dans cette étape on va entrer les caractéristiques dans un filtre éliminateur, pour atteindre les meilleurs caractéristiques, ce filtre est construit avec la méthode *Feature Selective* (chapitre III).



Figure V.2 : Exemple d'un filtre éliminateur

Après avoir appliqué le filtre on a obtenu les caractéristiques représenté dans la figure suivante :

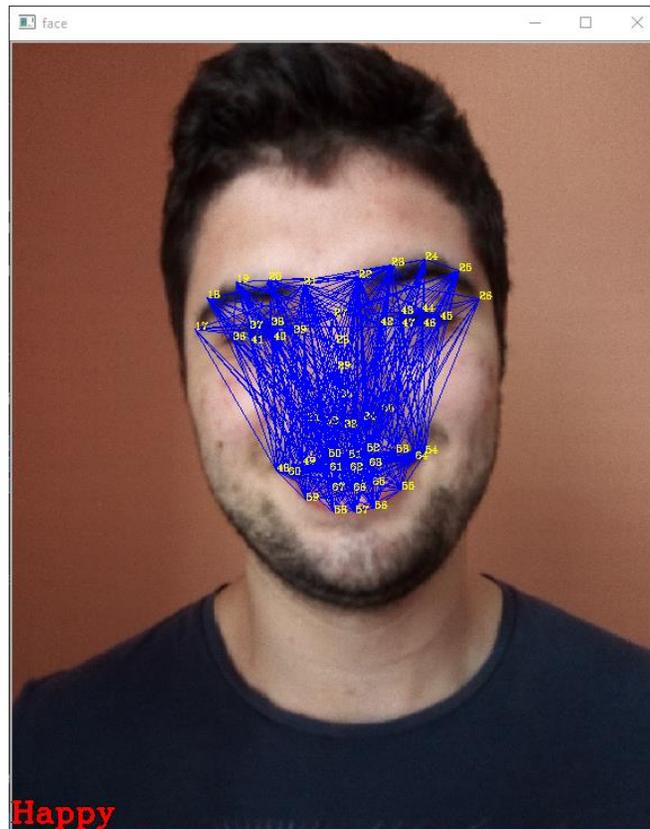


Figure V.3 : Représentation des caractéristiques sélectionnées

V.3 Les prédictions des probabilités du chaque classificateur :

V.3.1 Chargement du classificateur :

On va charger les quatre classificateurs (sauvegardés précédemment) avec bibliothèque *Pickle*

```
#filenameModèle : l'emplacement du modèle
modèle = pickle.load(open(filenameModèle, 'rb'))
```

V.3.2 Obtention des probabilités :

Les probabilités de chaque émotion avec les classificateurs suivants : *KNN*, *SVMLinear*, *SVMPoly*, *RFC* concernant l'image test représenté dans le tableau ci-dessous :

Etat	RFC	SVmlinear	SVMpoly	KNN
Neutral	0	0	0	0
Happy	89	97	97	100
Sad	0	0	0	0
Surprise	0	0	0	0
Discust	9	0	0	0
Angry	0	0	0	0
Fear9	0	1	1	0

*La probabilité est sous forme de pourcentage.

V.4 Les résultats finaux :

Pour minimiser les erreurs on a utilisé des coefficients pour chaque classificateur, ces coefficients dépendent de la justesse.

```
def moy(a,b,c,d):
    x=77*a+88*b+88*c+77*d
    return int(x/(77+88+88+77))
```

```
p=[moy(int(p[0][0]*100),int(p[1][0]*100),int(p[2][0]*100),int(p[3][0]*100)),
.
.
.
moyenne(int(p[0][6]*100),int(p[1][6]*100),int(p[2][6]*100),int(p[3][6]*100))]
```

Les moyennes sont introduites dans un vecteur et visualisées dans la figure ci-dessous :

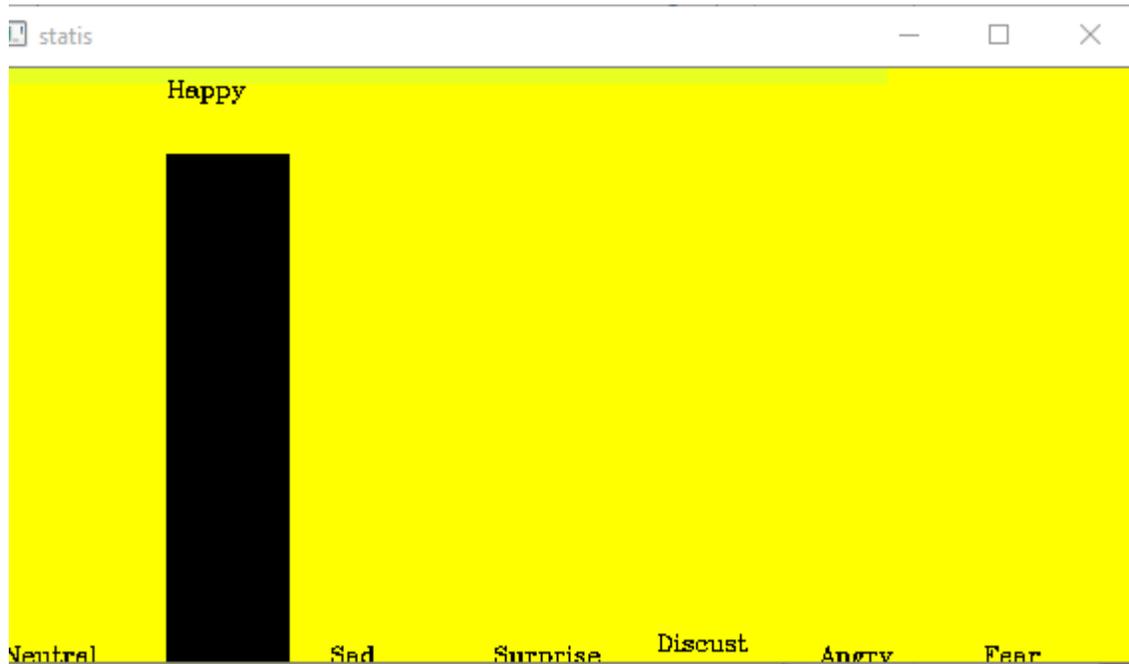


Figure V.4 : Les probabilités regroupées sous formes des barres

Etat	moyenne de chaque modèle
Neutral	0
Happy	95
Sad	0
Surprise	0
Discust	2
Angry	0
Fear	0

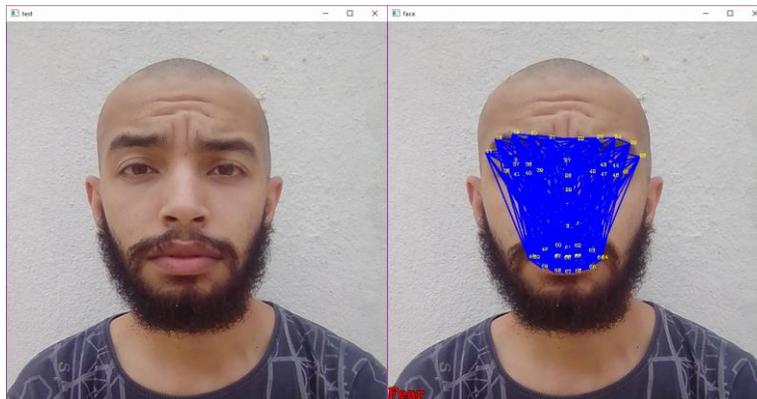
La prédiction de l'état est dédiée de la moyenne la plus haute de l'ensemble des classificateurs, dans notre exemple, l'état joie (*happy*) est la plus haute moyenne donc elle est pris comme résultat final.

Happy	precision moyen par modèle	95%
-------	----------------------------	-----

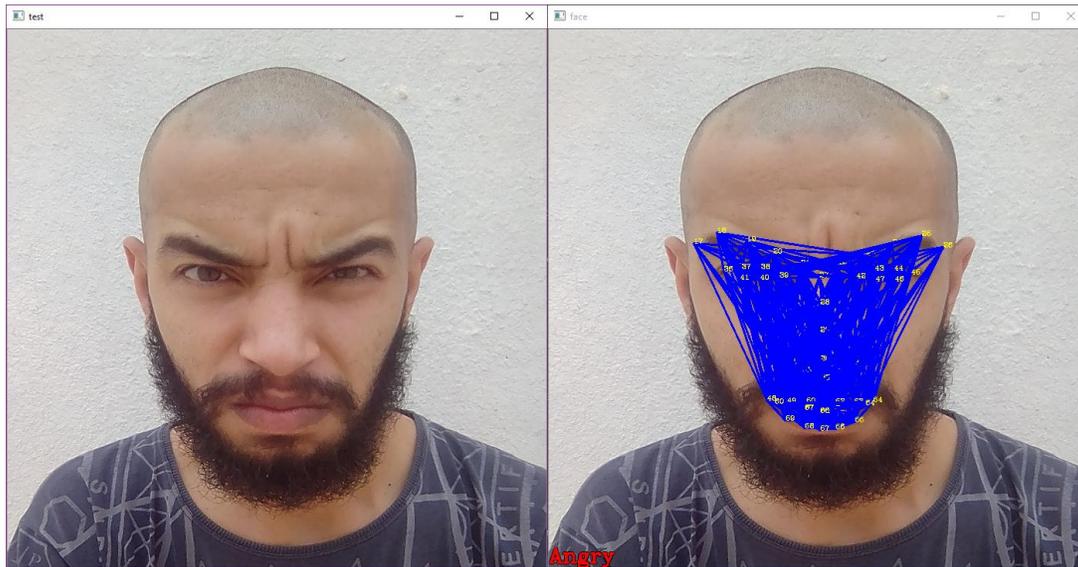
V.5 Autres exemples de test :



Etat	RFC	SVMlinear	SVMpoly	KNN	moyenne de chaque modèle
Neutral	0	0	0	0	0
Happy	100	98	99	100	99
Sad	0	0	0	0	0
Surprise	0	0	0	0	0
Discust	0	0	0	0	0
Angry	0	0	0	0	0
Fear	0	1	0	0	0
Happy	précision moyenne par modèle 99 %				



Etat	RFC	SVMlinear	SVMpoly	KNN	moyenne de chaque modèle
Neutral	9	4	5	19	8
Happy	0	1	1	0	0
Sad	39	17	26	50	32
Surprise	0	6	1	9	3
Discust	9	0	0	0	2
Angry	0	0	0	0	0
Fear	39	69	64	19	49
Fear	précision moyenne par modèle 49 %				



Etat	RFC	SVMlinear	SVMpoly	KNN	moyenne de chaque modèle
Neutral	0	0	0	0	0
Happy	0	1	0	0	0
Sad	9	0	0	0	2
Surprise	0	0	0	0	0
Discust	19	7	0	9	8
Angry	70	90	98	89	87
Fear	0	0	0	0	0
Angry	précision moyenne par modèle			87 %	

V.6 Conclusion :

Dans ce chapitre on a testé les performances de nos classificateur (probabilité de la prédiction de l'état émotionnel) avec on a eu de bonnes performance avec la forêt aléatoire et le Machine à vecteurs de support (linéaire et polynomial) mais le K- des voisins les plus proche était moins précis que les autres classificateurs.

Conclusion générale :

Notre travail consistait à mettre en œuvre une procédure de traitement pour la reconnaissance d'émotion à partir d'images. La procédure développée est fondée sur l'analyse des expressions faciales. Pour l'extraction des paramètres, on a utilisé une méthode basée sur les caractéristiques locales, et pour la sélection des meilleures caractéristiques, on a utilisé un classificateur basé sur la méthode « la forêt aléatoire » avec filtre éliminateur qu'on a conçu, pour représenter efficacement les images de visages.

Pour la classification et la prédiction, on a utilisé le classificateur SVM (polynomiale et linéaire), les classifieurs KNN et RFC qui ont été très performants au niveau de la reconnaissance des états émotionnels à partir des exemples. Pour certains exemples nos classificateurs ont atteint un taux de précision de 95%.

Ce projet nous a permis d'entrer dans le monde de l'intelligence artificielle, réaliser certains classifieurs en les paramétrant et comparer leurs performances sur une base de données.

Ce travail nous a permis de mettre en pratique nos connaissances sur les différents outils de l'apprentissage automatique tels que SVM qui entre dans le domaine d'intelligence artificielle ainsi qu'on a utilisé un nouvel outil python, on a touché aussi le domaine de traitement d'images et d'en acquérir d'autres. Il nous a servi d'une bonne initiation à la recherche.

Perspectives :

Comme perspectives de recherches futures, nous envisageons :

1. de tester sur notre modèle d'autre grande base de données telle que Fer2013 [26]
2. d'utiliser d'autres outils de sélection de caractéristiques pour augmenter plus la précision des classificateurs.
3. de tester d'autres noyaux de SVM.
4. d'utiliser le classificateur CNN.

Bibliographie

[1] Image by Brad Dorsey from Pixabay.com

[2] L'affective computing ; rôle adaptatif des émotions dans l'interaction homme-machine, Travail d'Étude et de Recherche, 2002-2003

[3] Yang, M.-H., Kriegman, D. J., and Ahuja, N. (2002). Detecting faces in images: A survey. IEEE Transactions on pattern analysis and machine intelligence, 24(1) :34–58.

[4]<https://www.freepik.com/free-photos-vectors/infographic>

[5] https://fr.wikipedia.org/wiki/D%C3%A9tection_de_visage consultée le 18 juin 2020

[6] <https://maelfabien.github.io/tutorials/face-detection/> consultée le 21 juin 2020

[7] <https://www.learnopencv.com/histogram-of-oriented-gradients/>

[8] Padgett, C., & Cottrell, G. W. (1997). Representing face images for emotion classification. In Advances in neural information processing systems (pp. 894-900).

[9] Lanitis, A., Taylor, C. J., & Cootes, T. F. (1997). Automatic interpretation and coding of face images using flexible modèles. IEEE Transactions on Pattern Analysis and machine intelligence, 19(7), 743-756.

[10] <https://kdef.se/download/KDEF-.zip>

[11] <http://dlib.net/> consultée le 19 juin 2020

[12] <https://ibug.doc.ic.ac.uk/resources/300-W/>

[13] <https://www.cpanewbrunswick.ca/-/media/provinces/new-brunswick/pdfs/intro-to-machine-learning-fr-web.pdf>

[14] thèse Khadija LEKDIOUI, Reconnaissance d'états émotionnels par analyse visuelle du visage et apprentissage machine, page 69

[15] B. Scholkopf and A.J. Smola. Learning with Kernels Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, 2002.

[16] http://www.abdelhamid-djeffal.net/web_documents/thesedjeffal.pdf page16-17 [20] <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html> consultée le 10 juin 2020

[17] <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html> consultée le 10 juin 2020

[18] [https://fr.wikipedia.org/wiki/Python_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage)) consultée le 22 juin 2020

- [19] [https://fr.wikipedia.org/wiki/Anaconda_\(Python_distribution\)](https://fr.wikipedia.org/wiki/Anaconda_(Python_distribution)) consultée le 8 juin 2020
- [20] <https://opencv.org/about/> consultée le 8 juin 2020
- [21] <https://fr.wikipedia.org/wiki/Scikit-learn> consultée le 8 juin 2020
- [22] <https://numpy.org/> consultée 8 juin 2020
- [23] <https://docs.python.org/3/library/pickle.html> consultée 10 juin 2020
- [24] https://fr.wikipedia.org/wiki/Visual_Studio_Code consultée le 8 juin 2020
- [25] Thèse Diallo Nene Adama dian, La reconnaissance des expressions faciales, page 12-14
- [26] <https://www.kaggle.com/deadskull7/fer2013>