

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Systèmes informatiques**

Présenté par
Hakima KEBAILI

Thème

Etude des méthodes de détection et de reconnaissance des gestes de la main.

Mémoire soutenu publiquement le 15/juin/ 2016 devant le jury composé de :

Président : M Arezki HAMMACHE

Encadreur : M Hakim Radja

Examineur : M Samir SADOU

Examineur : M Youcef YACINE.

Remerciements

Je tiens à exprimer mes remerciements et ma vive gratitude :

A Monsieur Radja Hakim qui m'a offert, grâce à sa confiance et son soutien précieux, la possibilité de réaliser ce travail.

A Monsieur Sadou Samir, à Monsieur Hammache Arezki et à Monsieur Yacine Youcef pour avoir accepté de juger mon modeste travail.

Et enfin je tiens à remercier ma famille et en particulier mon père et ma mère.

Dédicaces

Je dédie ce modeste travail à :

Ma chère famille.

Tous mes amis et en particulier LAHZOUZ.

Table des matières

I	Communication gestuelle et langue des signes.	12
I.1	introduction	13
I.2	Communication :	13
I.3	Geste	13
I.4	Définition de la langue des signes :	17
I.5	Historique :	17
I.6	Séméiologie :	19
I.7	Quelques caractéristiques de la langue des signes :	20
I.8	Conclusion	26
II	Vision par ordinateur et état de l'art de la reconnaissance des gestes de la main	27
II.1	Introduction	28
II.2	Définition de la vision par ordinateur :	28
II.3	Les images numériques et leurs traitements :	28
II.4	Etat de l'art de la reconnaissance des gestes de la main	36
II.5	Conclusion	38
III	Méthodes de détection de la main et de reconnaissance des gestes	39
III.1	introduction :	42
III.2	Segmentation d'image :	42
III.3	Comparaison des trois approches de segmentation d'images :	61
III.4	Intelligence artificielle :	62
III.5	Apprentissage artificiel / automatique (Machine Learning) :	62
III.6	Types d'apprentissages :	62
III.7	Scénario de l'apprentissage supervisé :	64
III.8	Classification supervisée :	66
III.9	Algorithmes de classification directe :	69
III.10	Evaluation des performances d'un classifieur :	79
III.11	Conclusion :	80

IV Implémentation et résultats	82
IV.1 introduction	83
IV.2 Outils et environnement de développement :	83
IV.3 Approche utilisée dans le cadre de la réalisation de notre travail :	91
IV.4 Conclusion :	105

Table des figures

I.1	Geste déictique	15
I.2	Geste de manipulation	15
I.3	Geste sémaphorique désignant 'OK'	15
I.4	Continuum de Kendon	16
I.5	Alphabet dactylogique de la LSF	23
II.1	Vocabulaire d'une image.	29
II.2	Représentation d'un cercle avec une image vectorielle(a) et une image matricielle(b). . . .	30
II.3	Image binaire (a), en niveaux de gris(b) et en couleur(c).	30
II.4	Histogramme d'une image contenant 5 niveaux de gris.	32
II.5	Histogramme d'une image en couleur.	33
II.6	Application du filtre gaussien à l'image d'origine (a).	35
II.7	Matrice de convolution utilisée pour appliquer le filtre gaussien à une image.	35
II.8	L'image d'origine(d), segmentation fondée sur les contours de l'image (a), segmentation fondée sur la classification de l'image (b) , segmentation fondée sur les régions de l'image(c). .	36
III.1	Schéma récapitulatif d'une détection de contours par gradient.	45
III.2	Schéma récapitulatif d'une détection de contours avec le filtre de Canny.	46
III.3	Détection de contours avec l'opérateur de prewitt	48
III.4	Processus de progression	55
III.5	Seuillage d'une image.	56
III.6	Le scénario classique de l'apprentissage supervisé : L'environnement fournit des données x_i tirées aléatoirement suivant une distribution DX sur l'espace d'entrées X. Ces données sont étiquetées par un oracle qui utilise pour ce faire une fonction f. L'apprenant reçoit un échantillon d'exemples ou couples $(x_i, u_i) = (x_i, f(x_i))$, et à partir de cet échantillon, doit chercher à deviner f, ou au moins à en trouver une approximation h	65
III.7	Le principe de SVM.	71
III.8	Les hyperplans possibles pour séparer les données.	72
III.9	Projection de données non linéairement séparables dans l'espace de Hilbert.	74

III.10	Exemple d'arbre de décision pour la question : Ce thème est-il intéressant ?	76
III.11	Exemple d'ensemble d'entraînement pour l'arbre ci-dessus	76
III.12	Exemple de matrice de confusion	81
IV.1	Logo d'Open CV	84
IV.2	Logo de Python, Numpy, Scipy, Scikit Learn, Matplotlib	89
IV.3	Bureau d'ubuntu.	90
IV.4	Logo de Sublime Text.	90
IV.5	Schéma décrivant le processus d'apprentissage.	94
IV.6	Recherche du séparateur optimal dans chacun des 26 classifieurs	95
IV.7	Schéma qui montre l'un des 26 séparateurs SVM dans le feature space.	95
IV.8	Schéma illustrant la phase de construction de la forêt aléatoire.	96
IV.9	Schéma illustrant la phase de prédiction.	97
IV.10	Schéma montrant le surapprentissage	98
IV.11	Schéma montrant la division d'un dataset en deux : dataset d'entraînement et dataset de test.	99
IV.12	Capture d'écran du dataset que nous avons utilisé	99
IV.13	La lettre A de la langue des signes sous différents angles.	100
IV.14	Capture d'écran du taux de reconnaissance obtenu en utilisant SVM.	100
IV.15	Capture d'écran du taux de reconnaissance obtenu en utilisant les forêts aléatoires.	102

Résumé

Ce mémoire s'inscrit dans le domaine de la reconnaissance des gestes de la main, plus précisément de l'alphabet de la langue des signes. L'objectif de ce travail est de poser une base solide pour de futures travaux de la reconnaissance de la langue des signes dans la perspective d'aboutir dans quelques années à un produit fini qui va aider la communauté des sourds muets à mieux s'intégrer dans la société en lui permettant de communiquer et interagir avec son environnement aisement.

L'approche proposée ici se déroule en deux étapes consécutives :

La première consiste en l'acquisition du geste via une caméra d'un ordinateur portable en utilisant l'une des méthodes de détection de zones d'intérêts qui est la segmentation d'images.

Cette dernière propose différentes techniques, ce qui nous a poussé à faire une étude comparative pour en choisir la plus optimale.

La deuxième consiste en la reconnaissance du geste en utilisant deux algorithmes d'apprentissage différents que nous avons comparé par la suite à savoir l'algorithme Machine à vecteurs de supports(SVM) et celui des forêts aléatoires d'arbres de décision et cela en leurs soumettant les résultats obtenus lors de la phase de détection.

Abstract

This work(memoir) is related to the field of hand gestures recognition, More precisely to the sign alphabet recognition. The main purpose of this work is to lay a firm foundation for future research about sign language recognition in the light of ending up with a completed and finished material in few years. A product that we believe will help the deaf-mute community to interact efficiently and easily with the rest of the society.

The approach proposed here is conducted as a two-step procedure :

The first one consists on Acquiring human gestures through artificial vision using a web cam and one of the interests regions detection methods which consists on images segmentation method. The later offers a set of different techniques, that is the main reason that pushes us toward doing a comparative study to choose the most appropriate one to our work.

The second one consists of the gesture recognition using two different machine learning algorithms(support vector machines and random decision forests) . After feeding this algorithms with the results we obtained in the detection phases we will end up by comparing their results.

Introduction générale

Lorsque nous sommes dépourvus d'ouïe ou de parole voir les deux, le geste reste le seul moyen dont nous disposons pour communiquer et interagir avec notre environnement. De ce fait, la langue des signes est née, langue qui nous permet de nous exprimer par le biais des gestes de la tête (mimique) et de la main.

Cependant, son apprentissage ne concerne que les sourds muets et leur entourage restreint, les isolant du reste de la société et les privant ainsi d'une vie normale.

Avec l'avancement de la technologie, plusieurs systèmes ont été développés pour y remédier, tel que le gant numérique, mais son coût excessivement cher le rend inaccessible au grand public. Des recherches plus avancées sont également en cours, parmi lesquelles nous citons l'utilisation de la kinect qui est un capteur de mouvements développé par Microsoft.

Notre travail consistera à réaliser un système capable de reconnaître les gestes de la main, plus précisément l'alphabet de la langue des signes dans la perspective que ce dernier puisse reconnaître les mots voire des phrases de la LSF¹.

Le mémoire sera divisé en quatre chapitres. Dans le premier chapitre, nous parlerons de la communication gestuelle d'un point de vue général pour ensuite nous intéresser à celle réalisée avec les gestes de la main et enfin nous parlerons de la langue des signes.

Dans le second chapitre, nous détaillerons la vision par ordinateur qui sera le principal pilier dans la réalisation de notre travail puis nous présenterons les différentes techniques développées au cours des dernières années dans le domaine de la reconnaissance des gestes de la main.

Dans le troisième chapitre, je présenterais une étude comparative des différentes méthodes de détection de zones d'intérêts pour choisir la plus appropriée puis je détaillerais les méthodes de reconnaissance ainsi que les algorithmes rattachés à ces dernières.

Dans le quatrième chapitre, nous allons extraire la zone d'intérêt à partir d'images en utilisant la méthode choisie lors du précédent chapitre. Cette dernière contient les données requises par les algorithmes d'apprentissage afin d'effectuer la reconnaissance des gestes de la main.

1. Langue des Signes Française

Nous comparerons par la suite les résultats des différents algorithmes d'apprentissage pour que le plus optimal soit utilisé dans des travaux ultérieurs.

Chapitre I

Communication gestuelle et langue des signes.

Sommaire

I.1	introduction	13
I.2	Communication :	13
I.2.1	La communication verbale :	13
I.2.2	La communication non verbale ou gestuelle :	13
I.3	Geste	13
I.3.1	Définition du geste :	13
I.3.2	Types de geste :	14
I.4	Définition de la langue des signes :	17
I.5	Historique :	17
I.6	Séméiologie :	19
I.7	Quelques caractéristiques de la langue des signes :	20
I.7.1	Langue des signes française :	22
I.8	Conclusion	26

I.1 introduction

L'histoire de la communication est aussi ancienne que celle de l'humanité. En effet l'Homme a éprouvé le besoin de communiquer depuis les temps anciens, et alors mit au point des codes, des alphabets et des langages décrivant le message à transmettre. Pour se faire il utilisa principalement la parole, en l'appuyant du geste pour améliorer la compréhension.

Cependant les gens nés avec l'incapacité de parler ou d'entendre voir les deux, se sont trouvés obligés de développer un nouveau moyen de communication, basé seulement sur les gestes c'est là que la langue des signes a vu le jour.

Dans ce chapitre nous allons nous intéresser à la communication gestuelle plus précisément à la langue des signes afin de mieux nous familiariser avec le domaine de notre étude.

I.2 Communication :

La communication est l'action de communiquer, d'établir une relation avec autrui, de transmettre quelque chose à quelqu'un. Elle peut aussi se rapporter à l'ensemble des moyens et techniques permettant la diffusion d'un message auprès d'une audience plus ou moins vaste et hétérogène ou l'action pour quelqu'un ou pour une organisation d'informer et de promouvoir son activité auprès d'autrui, d'entretenir son image, par tout procédé médiatique [1]. Il existe deux types de communication :

I.2.1 La communication verbale :

La communication verbale passe par la parole. du point de vue ethnologique ¹, la communication verbale est un ensemble de sons émis dans le but d'établir une communication avec autrui.[2]

I.2.2 La communication non verbale ou gestuelle :

La communication non verbale (ou langage du corps) désigne dans une conversation tout échange n'ayant pas recours à la parole. Elle ne repose pas sur les mots, mais sur les gestes (actions et réactions), les attitudes, les expressions faciales (dont les micro-expressions) ainsi que d'autres signaux, conscients ou inconscients.[3]

I.3 Geste

I.3.1 Définition du geste :

Un geste est un signe manuel ou corporel qui permet d'illustrer les mots du langage, de les compléter ou de les appuyer. Le geste peut aussi être utilisé pour remplacer la parole. [4]

1. L'ethnologie est l'une des sciences humaines et sociales : elle relève de l'anthropologie et est connexe à la sociologie. Son objet est l'étude comparative et explicative de l'ensemble des caractères sociaux et culturels.

I.3.2 Types de geste :

Il existe plusieurs classifications de gestes parmi lesquels nous citons la :

a) Classification selon les parties du corps impliquées [5] :

On distingue généralement trois types de gestes :

a).1 les gestes de la main et du bras : Ils forment la principale catégorie de gestes interactifs. La main permet de réaliser des gestes précis et complexes. Les recherches autour de ces gestes concernent principalement la reconnaissance de positions de la main, l'interprétation du langage des signes et le développement d'interface homme-machine permettant la manipulation et l'interaction avec des données ou des éléments d'un environnement virtuel.

a).2 Les gestes de la tête et du visage : Peu de gestes de la tête ont une signification spécifique ; l'orientation de la tête est quant à elle très utile pour la détection du champ de vision. Les recherches dans ce domaine s'intéressent à la reconnaissance faciale comme moyen d'authentification biométrique, comme soutien à d'autres systèmes de reconnaissance tels que la reconnaissance de la parole ; l'analyse des gestes faciaux est également utile pour la réalisation d'avatars virtuels réalistes.

a).3 Les gestes impliquant tout le corps : Les recherches dans ce domaine s'intéressent à tout le corps en interaction avec son environnement (analyse des gestes d'un danseur afin de générer de la musique idoine ; analyse des gestes d'un athlète pour améliorer ses performances).

b) Classification selon Maria Karam et m.c. schraefel [6]

Karam et schraefel, présentent cinq styles de gestes, principalement de la main et du bras, synthétisant les types d'interaction décrits à travers la littérature scientifique :

b).1 Les gestes déictiques : Sont des gestes de pointage permettant d'identifier un objet ou son emplacement. Ces gestes sont typiquement utilisés dans des environnements virtuels. Ces gestes peuvent être considérés comme implicites dans d'autres formes de gestes (par exemple lorsque l'on pointe des doigts un objet à manipuler).



FIGURE I.1 – Geste déictique

b).2 Les gestes de manipulation : Dont le but est de contrôler une entité en appliquant une relation étroite entre le mouvement du geste et l'entité qui est manipulée.



FIGURE I.2 – Geste de manipulation

b).3 Les gestes sémaphoriques : Il font partie de tout système gestuel basé sur un catalogue conventionnel de gestes statiques ou dynamiques (par exemple : le geste (statique) pour le signe « ok » le signe (dynamique) de la main pour dire « au revoir »).



FIGURE I.3 – Geste sémaphorique désignant 'OK'

c) Classification selon Kendon : [7]

Le système de classification proposé par Kendon synthétise sous la forme d'un continuum², plusieurs autres réflexions portant sur le même sujet :

2. Un continuum est un ensemble d'éléments tels que l'on peut passer de l'un à l'autre de façon continue.

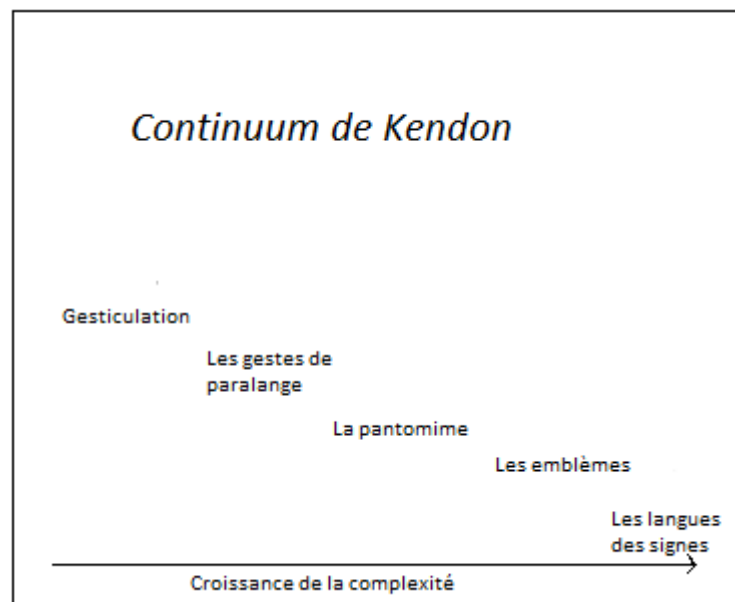


FIGURE I.4 – Continuum de Kendon

Ce continuum reflète une évolution croissante (lorsqu'on le parcourt de gauche à droite) de la complexité et de la richesse du message gestuel :

c).1 La gesticulation (Spontaneous Gesticulation) : Il faut entendre par cela l'ensemble des gestes co-verbaux (accompagnant la parole).

c).2 Les gestes de paralangage (Language slotted) : Résulte de la traduction de l'anglais language-like gestures. En réalité, ils ne constituent nullement un langage car, à la différence des langues des signes, ils apparaissent plutôt comme un simple système de codage au vocabulaire très restreint. Ce sont par exemple les gestes utilisés pour l'arbitrage de certains sports.

c).3 La pantomime : vise à créer une image gestuelle d'un objet, d'un événement, d'un état d'esprit. elle est fortement iconique.

Le geste iconique peut-être :

- Spatiographique : Montrent une relation spatiale entre les référents. Ex : le geste fait pour demander à quelqu'un de reculer.
- Pictographique : Évoquent la forme, la taille ou l'état de surface d'un objet, soit en dessinant celui-ci dans l'espace, soit grâce à la seule configuration de la main. Ex : Le geste fait pour décrire Le Zeppelin(ce dirigeable en forme de ballon de rugby).

— Kinégraphique : Représentent le mouvement de quelque chose ou quelqu'un. Bien souvent, la main joue le rôle d'un référent. Ex : le geste fait pour écrire.

c).4 Les emblèmes : (Emblems) sont des gestes complétant les mots ou utilisés à leur place. Ils sont utilisés au sein d'une société, dans laquelle ils ont une signification fixe et communément acceptée. Le froncement des sourcils est ainsi largement répandu pour exprimer le doute, le désaccord ou une difficulté de compréhension. En revanche, la représentation des chiffres avec les doigts de la main diffère selon les cultures.

c).5 Les langues des signes : (Signs) sont les seules à pouvoir être considérées comme de véritables langues. Elles possèdent certaines règles de formation et d'ordonnancement, et une combinatoire permettant de générer de nouveaux items lexicaux selon les besoins.

I.4 Définition de la langue des signes :

La langue des signes désigne l'une ou l'autre des langues gestuelles (produites par les mouvements des mains, du visage et du corps dans son ensemble) que les personnes atteintes de surdité ont développées pour communiquer. Elle assure toutes les fonctions remplies par les langues orales.[8]

I.5 Historique :

Bien qu'elle ne soit reconnue que depuis 2005 comme langue à part entière, la langue des signes est certainement apparue bien avant les langues orales, du temps où l'appareil phonatoire de l'Homme n'était pas développé. L'ironie a fait que la mère de toutes les langues, celle que nous pratiquons instinctivement, certes maladroitement et non codifiée, au sortir du ventre maternel, n'a trouvé sa légitimité que dans si peu de pays et si tardivement. Une histoire de la langue des signes mouvementée en quelque sorte et qu'on peut classer en sept grandes étapes : [9]

a) La langue des signes, un code gestuel : Dans l'antiquité, l'intelligence était étroitement liée à la parole. Aristote pensait que quelqu'un qui ne parle pas, ne peut pas penser. Les sourds, isolés, n'ont pas pu enrichir leurs langues signées et ont dû se contenter d'une gestuelle simpliste. De ce fait, ne disposant pas d'une langue élaborée et ne bénéficiant pas d'éducation, ils passaient pour des simples d'esprit.

b) A partir du 16^{ième} siècle : les précepteurs A partir du 16^{ième} siècle, des peintres sourds tels que Navarette ou Pinturicchio ont été reconnus. Par ailleurs, en Espagne, des enfants sourds issus de la noblesse ont été instruits par des précepteurs³. L'un d'entre eux, Pedro Ponce de Leon s'intéressa aux

3. est une personne chargée de l'éducation et de l'instruction d'un enfant issu d'une famille noble ou aisé qui reste dans sa famille et ne fréquente aucun établissement d'enseignement publique ou privé

codes gestuels existants tel que l'alphabet manuel et les utilisa pour enseigner auprès de ces enfants. D'autres précepteurs mirent plutôt l'accent sur l'apprentissage de la parole.

c) Le 18ème siècle : L'abbé de l'épée : L'abbé de l'épée fut, en 1760, le premier entendant connu à s'intéresser aux modes de communication des « sourds-muets » En observant un couple de jumelles sourdes communiquer entre elles par gestes il découvre l'existence d'une langue des signes. Il décide de s'appuyer sur cette langue pour instruire les enfants sourds. Il l'adapte en y ajoutant des notions grammaticales propres au français (par exemple, la conjugaison). C'est ce qu'il appelle les « signes methodistes » Par ailleurs, il regroupe les enfants sourds pour les instruire et ouvre une véritable école pour sourds qui deviendra l'institut national des jeunes sourds, aujourd'hui Institut Saint-Jacques, à Paris. L'abbé de l'épée est aujourd'hui une figure historique de l'histoire des Sourds.

d) 1880 : le congrès de Milan : Dans la même période, le courant « oraliste » s'amplifie. Les « oralistes » pensent que les sourds doivent apprendre à parler pour s'intégrer dans la société. Le congrès de Milan en 1880 - où l'immense majorité des participants est entendant et oraliste décrète :

« que la méthode orale pure doit être préférée » Trois raisons sont invoquées :

- la LSF (langue des signes française) n'est pas une vraie langue,
- elle ne permet pas de parler de Dieu,
- les signes empêchent les sourds de bien respirer ce qui favorise la tuberculose.

Cette « préférence » a eu des conséquences dramatiques pour les sourds : pendant 100 ans la langue des signes a été proscrite, méprisée et marginalisée aux seules associations de sourds. Dans les instituts de sourds, les élèves signent en cachette. La langue des signes s'est alors appauvrie.

e) Les années 80 : le réveil sourd : Durant les années 1980, se produit ce que les sourds appellent le « réveil sourd »

La langue des signes commence à reconquérir ses lettres de noblesse avec William Stokoe, linguiste, qui étudie la langue des signes comme une véritable langue. Des chercheurs en linguistique et en sociologie tels que Christian Cuxac et Bernard Mottez poursuivent ce travail et mettent en avant la culture sourde qui y est rattachée.

Par ailleurs, un travail culturel est mené par Jean Gremion (écrivain, journaliste et metteur en scène) et Alfredo Corrado (un artiste sourd américain). Ils créent en 1976, l'international Visual Theatre (IVT). Dès lors, ils travaillent à la requalification de la langue des signes. En parallèle, une réflexion est menée sur l'enseignement auprès des élèves sourds. La philosophie bilingue (LSF, langue des signes française) commence à germer dans les esprits. En 1980 est créée l'association « 2 Langues pour une Education », (2LPE⁴) Elle met en place des « stages d'été pour les parents » Ces stages rassemblent des parents d'enfants sourds,

4. Deux langues pour une seule éducation

des sourds, des interprètes. Ils oeuvrent ensemble à la création des premières classes bilingues dans un contexte législatif et sociologique difficile.

f) Les années 90 : le début de la médiatisation de la LSF : Dans les années 90, les sourds et la LSF commencent à avoir une renommée dans le grand public. En 1992, un numéro de la « La marche du siècle » est consacré aux sourds. Les gens découvrent alors cette communauté et cette langue à travers les témoignages de Victor Abbou et Joël Chalude. Puis Emmanuelle Laborit comédienne sourde, reçoit en 1993, le Molière de la révélation théâtrale pour son rôle dans Les Enfants du silence. Cette même année, le documentaire « Le pays des sourds » de Nicolas Philibert montre cet univers inconnu des entendant. Pendant ces années, de nombreuses associations de sourds ouvrent leurs portes aux entendant en leur proposant des cours de langue de signes. Ces formations, les films, le théâtre et l'engagement de plusieurs associations dans la sensibilisation pour la culture sourde, permet une meilleure reconnaissance des droits des sourds. Dans le même temps, le métier d'interprète en LSF, langue des signes français, se professionnalise et est validé par un diplôme. De son côté 2LPE, 2 Langues Pour une Education, continue à militer pour le bilinguisme dans l'éducation. En 1988, voit naître à Poitiers, 2LPE Centre Ouest qui oeuvre pour la mise en place de classes bilingues, la promotion et la reconnaissance de la Langue des Signes comme langue à part entière.

g) Les années 2000 : la LSF, langue à part entière : Progressivement les mentalités et les représentations évoluent. Yves Delaporte, ethnologue, se penche lui aussi sur la communauté sourde et la LSF. Il publiera en 2007 un « Dictionnaire étymologique et historique de la langue des signes française. » De son côté, 2 Langue Pour une Éducation, organise tous les ans des Universités d'été sur le thème du bilinguisme ou la LSF est la première langue. Les combats menés depuis 25 ans pour la reconnaissance de la langue des signes commencent à porter leurs fruits : la Loi numéro 2005 - 102 du 11 février 2005 reconnaît la LSF comme « langue à part entière ». En 2008, la LSF devient une option pour le Bac, comme n'importe quelle autre langue. En 2010, le CAPES, Certificat d'aptitude au professorat de l'enseignement du second degré, de LSF est créée. En 2012, c'est l'année du 300ème anniversaire de la naissance de l'Abbé de l'Épée. De multiples hommages lui ont été rendus par les Sourds.

I.6 Séméiologie :

Si la langue des signes est enseignée et diffusée, elle est conçue en tant que « reproduction » d'une langue qu'elle visualise et gestualise. Il faut attendre William Stokoe pour que la langue des signes soit observée comme une langue à part entière grâce à la description selon le principe de la double articulation que André Martinet développe pour le langage humain en général et atteste pour la langue des signes dans l'introduction à l'Essai de grammaire de la langue des signes française de Nève de Mévergnies. Ces descriptions, très souvent menées selon les critères d'analyse des langues orales, ont contribué à faire peu

à peu reconnaître à ces langues leur statut de langues naturelles à part entière. Cependant du fait que les langues des signes utilisent une modalité visuo-gestuelle et non audio-orale, elles mettent en place des structures spécifiques, bien différentes de celles des langues orales et nécessitent donc une description circonstanciée. Comme toute langue, une langue des signes nécessite un apprentissage mais il n'est pas nécessaire d'avoir une surdité pour apprendre ou communiquer en langue des signes. Pour exemple de nombreux entendants (enfants de sourds, partenaires, ou interprètes et autres professionnels en contact avec des sourds) parviennent à développer un haut degré de bilinguisme. « les langues des signes sont pour les sourds, le seul mode linguistique véritablement approprié, qui leur permette un développement cognitif et psychologique d'une manière équivalente à ce qu'il en est d'une langue orale pour un entendant. » On parle souvent quand on traite de la langue des signes d'une « pensée visuelle ». Elle remet en question ce que nous considérons habituellement comme appartenant au domaine de la linguistique. En effet, selon Christian Cuxac, dans une perspective sémiogénétique, le modèle de la langue des signes française propose une bifurcation⁵ de visée entre deux types de structures (fréquemment imbriquées dans le discours)[10] :

- les structures dites standards ou « signes standard », au caractère conventionnel :
- les structures de grande iconicité, à visée illustrative.

I.7 Quelques caractéristiques de la langue des signes :

La Langue des Signes est le moyen d'expression utilisé par les communautés de sourds ou malentendants pour communiquer entre eux. La Langue des Signes est une véritable langue à part entière (avec un lexique, une syntaxe...) et constitue de ce fait la forme la plus évoluée en communication gestuelle.

L'expression de phrases en LS, Langue des signes, ne se réduit pas aux gestes produits par les deux mains, c'est le corps tout entier qui peut être mis à contribution pour exprimer une phrase. On peut distinguer trois principales parties qui interviennent : les mains, la tête (mimique et regard) et le buste. Dans cette étude on se restreint uniquement aux mains, c'est pourquoi les autres parties du corps ne seront pas abordées par la suite. Chaque phrase est constituée d'une suite de gestes des mains que l'on appelle signes et qui sont agencés suivant une syntaxe régie par une logique spatiale et temporelle.[11]

a) Composition d'un signe : Chaque geste d'une main peut être décomposé en quatre paramètres qui sont indépendants et peuvent être aussi bien dynamiques qu'invariants durant l'émission du signe.

- La configuration : forme de la main (exemples de noms de configurations : les noms de l'alphabet dactylogique, bec de canard, index)
- L'orientation : l'orientation de la main est la position de la paume de la main par rapport au signeur.

Il y a 6 directions de la paume de la main et une direction pour les deux paumes. Ces directions

5. est l'étude de certains aspects des systèmes dynamiques. Une bifurcation intervient lorsqu'un petit changement d'un paramètre physique produit un changement majeur dans l'organisation du système.

peuvent varier durant l'exécution du signe.

La paume peut être orientée vers : Le haut, Le bas, La droite, La gauche Vers soi Devant soi.

- Le mouvement : Il peut y avoir mouvement sur les doigts, la main, le bras, le corps et la tête en jouant sur l'ampleur, la vitesse ou la répétition.
- L'emplacement : L'emplacement de la main représente l'endroit où l'action est effectuée ou un endroit générique. Elle se trouve en générale dans l'espace, devant le signeur, dans la zone d'à côté de la tête du signeur et sur le corps du signeur.

Chacun de ces paramètres est porteur d'information et contribue au sens d'un signe.

b) Interaction entre les deux mains : Un signe peut aussi bien faire intervenir une main que deux mains et les deux mains ont différentes façons d'interagir. Lorsque les deux mains sont impliquées dans un signe, deux cas se présentent. Dans le premier cas on voit apparaître un rôle pour chaque main. Une main est dite dominante et a pour rôle de décrire "l'action", tandis que l'autre main qui est appelée main dominée sert de référence à cette action. En général la main dominante se déplace au cours du geste tandis que la main dominée reste statique. Dans le deuxième cas, les deux mains sont complètement synchronisées : leurs paramètres sont identiques ou symétriques.

c) Différentes classes de signes : En LS, langue des signes, on peut distinguer différentes classes de signes, chacune correspondant à un usage particulier. L'ensemble des signes dits « standards » correspond à des mots (nom, verbe, adjectif...) ayant un sens bien établi. On peut leur associer une correspondance relativement précise dans les langues orales. Du fait qu'il y a une correspondance « directe », ce sont les signes qui ont été le plus étudiés en reconnaissance de geste. Il existe deux autres classes de signes qui eux ne permettent aucune correspondance directe avec des mots du langage oral. Il s'agit des spécificateurs de forme et de taille, et des classificateurs. Ces signes ont des aspects fortement iconiques et font partie de ce que l'on appelle la grande iconicité. Les spécificateurs permettent de décrire un objet, un animal, une scène. C'est à l'aide de la forme des mains, de leur orientation et de leur mouvement que le signeur décrit une forme et les dimensions d'un objet. Les classificateurs sont similaires aux spécificateurs dans le sens où ils représentent également un objet (ou personne, animal...) et donc la forme de la main est en rapport avec celle de l'objet ou de sa fonction. Mais ils ont un tout autre rôle car ils servent en quelque sorte de pronom. Lorsqu'un objet a été cité dans une phrase à l'aide d'un signe du vocabulaire standard (ou de spécificateurs), un classificateur peut ensuite être utilisé pour représenter cet objet dans le reste de la phrase. On peut avec ce classificateur préciser la position de l'objet ou décrire une trajectoire qu'il a emprunté.

d) Espace-temps : Il faut également relever l'utilisation particulière de l'espace par la langue des signes. En effet, alors que les langues vocales utilisent de préférence des structures syntaxiques linéaires pour le marquage temporel ou encore les relations entre différents éléments de la phrase, la langue des signes utilise de préférence des structures syntaxiques spatiales : le temps peut par exemple se dérouler selon un axe arrière-avant dans l'espace du signeur ou encore selon un axe gauche-droite.

e) Marqueur pronominal : L'espace de signation (là où la personne signe) peut aussi servir à créer des repères, des marqueurs auxquels on se réfère tout au long du discours (par ex. un repère pour l'école, un pour la maison, un autre pour un personnage). Il suffit alors de pointer du doigt ou du regard l'endroit pour « l'activer » et y faire référence dans le discours. C'est en quelque sorte un usage spatial du pronom.

I.7.1 Langue des signes française :

La langue des signes française (LSF) est la langue des signes utilisée par les sourds francophones et leurs proches ainsi que certains malentendants pour communiquer et par des entendants. La LSF est une langue à part entière et un des piliers de l'identité de la culture sourde.[12]

a) Reconnaissance en France : Le décret de la loi no 2005-102 du 11 février 2005 reconnaît la langue des signes française comme « langue à part entière » dans le code de l'éducation : La langue des signes française est reconnue comme une langue à part entière. Tout élève concerné doit pouvoir recevoir un enseignement de la langue des signes française. Le Conseil supérieur de l'éducation veille à favoriser son enseignement. Il est tenu régulièrement informé des conditions de son évaluation. Elle peut être choisie comme épreuve optionnelle aux examens et concours, y compris ceux de la formation professionnelle. Sa diffusion dans l'administration est facilitée.

b) Alphabet dactylologique : L'alphabet dactylologique est la façon de signer l'alphabet latin. Il est utilisé pour épeler les noms propres ou les mots n'existant pas encore en LSF. La dactylologie de la LSF se fait d'une seule main, alors que les langues de la famille de la langue des signes britannique se pratiquent avec les deux mains.

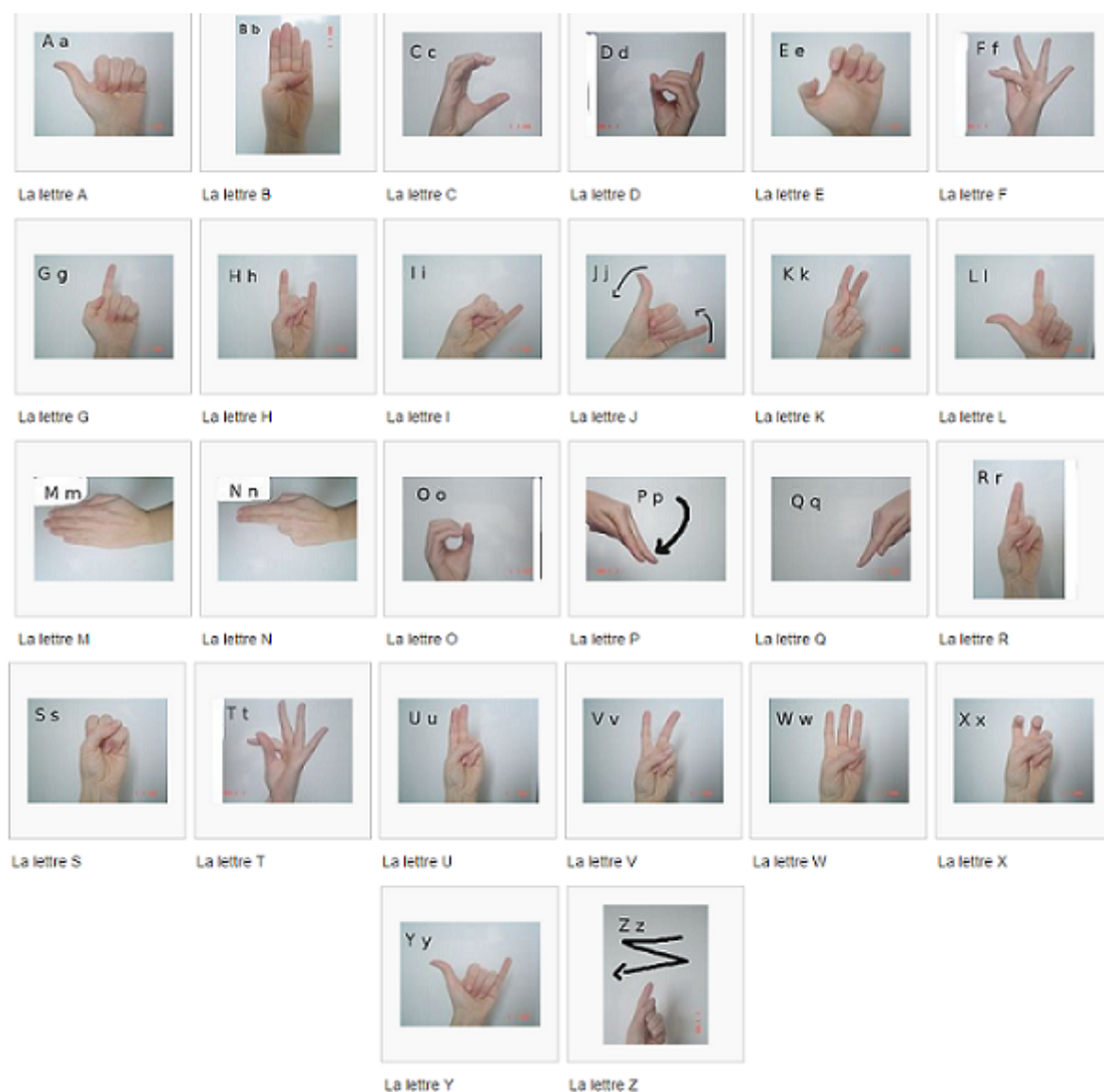


FIGURE I.5 – Alphabet dactylogique de la LSF

c) Grammaire : La grammaire de la LSF est « en 3D », c'est à dire qu'il est possible d'exprimer plusieurs idées simultanément, ce qui la différencie de la grammaire française des entendant. Par exemple : le francophone va dire : « Hier je me suis super bien amusé à la fête » en mettant les mots dans cet ordre, le signeur va signer sur la ligne du temps que c'était « hier », signer le mot « la fête » et qu'il s'est « super bien amusé » en utilisant les intensifs du visage et des gestes. La langue des signes française a une grammaire différente du « français signé » (la syntaxe est différente et elle utilise des signes pour les mots).

c).1 Clefs de la grammaire LSF :

— Expressions du visage : pour indiquer le sens de la phrase. Par exemple, pour poser une question

totale (qui sollicite une réponse par oui ou non), le locuteur aura les sourcils froncés pendant sa phrase. En revanche, une question ouverte comme « Où vas tu en vacances ? » sera posée avec un haussement des sourcils, également utilisé pour les intensifs par exemple très ou beaucoup..

- La ligne du temps : il n'existe pas de conjugaison en LSF, il suffit au signeur de situer l'action sur la ligne du temps (perpendiculaire à lui : derrière son épaule pour le passé, au niveau de son corps pour le présent et devant lui pour le futur).
- Transferts : des gestes pronominaux montrant le rôle ou la forme du référent. Pour faire référence à un homme, le locuteur utilisera son index vers le haut, mais pour parler d'une voiture, il utilisera la main à plat. Notez que les noms pour homme et voiture sont différents des pronoms décrits. Les verbes de mouvement peuvent également être signés pour indiquer la direction : « La voiture tourne à gauche » sera le classificateur de voiture tournant à gauche. Autre exemple, « Un homme monte l'escalier » se fera par deux doigts imitant les jambes montant un escalier, dans un mouvement pouvant être différent selon qu'il s'agit d'un escalier classique ou en colimaçon ...
- Pour raconter : pour raconter quelque chose et parler de personnes absentes le signeur définit sa place et celle des autres dans l'espace, à la manière d'une mise en scène théâtrale, il est ainsi plus facile de comprendre et de suivre visuellement de qui il s'agit et quels sont les rapports entre les personnages.

d) Verbes uni et pluridirectionnels :

- Les verbes pluridirectionnels sont utilisés pour désigner le sens de la communication , par exemple Pour signer « Je te téléphone », le locuteur fera le signe téléphone de lui vers son interlocuteur. En revanche, « Tu me téléphones » se fera en signant téléphone de l'interlocuteur vers le locuteur.
- Les verbes unidirectionnels : qu'on utilise le « je », le « tu » ou le « il » le verbe aura la même forme comme les verbes « aller », « manger ».

e) L'ordre des mots : tout d'abord le temps, puis le lieu, ensuite le sujet et enfin l'action. Ce qui est logique puisque la pensée visuelle des sourds entraîne une mise en scène systématique de ce qui se dit : le décor est tout d'abord planté, les acteurs entrent ensuite en scène et l'action peut enfin débiter...

f) Vocabulaire et genèse des signes : Le lexique des signes est toujours en perpétuel mouvement et s'enrichit encore aujourd'hui. En effet, au fur et à mesure que le monde des sourds découvre et accède à des milieux spécialisés (milieu étudiant ou professionnel), le besoin de créer de nouveaux signes se fait davantage sentir.

f).1 Des signes venus du mime : (dits iconiques⁶) : beaucoup de signes peuvent être faciles à retenir même pour un entendant car ils font partie du mime pour des actions (manger, dormir, parler...), des objets (pomme de terre, poupée...), des lieux ou paysages (école, maison, montagne...), des animaux (vache, escargot, éléphant). Ce sont ces signes culturels que sourds et entendant ont en commun dans leur imaginaire collectif qui sont la base de la communication entre eux. ces signes créent souvent une complicité et un sens de l'humour commun. Tout cela fait des échanges entre sourds et entendant un moment agréable, voire une découverte, pour les entendant, d'un univers poétique qui allie visuel et pensée.

f).2 Des signes arbitraires : l'alphabet dactylographique est un des meilleurs exemples de signes arbitraires (bien que certains signes aient des ressemblances de formes avec la graphie de la lettre) créés afin de faire lire les mots français aux sourds. Il permet aux sourds d'épeler des mots à des entendants qui ne connaissent pas le signe correspondant, mais le plus souvent c'est pour épeler leur nom ou celui d'une ville dont le signe n'est pas encore connu. Il s'agit là d'un pont non négligeable entre les deux langues

f).3 Des signes influencés par la langue française : en côtoyant le monde des entendants la LSF a aussi intégré des signes directement en relation avec le français et souvent les premières lettres des mots sont associées à des mouvements plus ou moins arbitraires, par exemple le v de vert, vrai ou vacances, le r de rêve ou de raison, le s de soeur ou le f de frère...

f).4 Des signes sans cesse inventés : ceux par exemple pour nommer quelqu'un (orthosignes) : c'est la première chose que font les sourds lorsqu'une personne nouvelle arrive et qu'elle n'a pas de signe, ils en trouvent un en fonction du physique ou du caractère de la personne...

g) Le français signé : est l'utilisation de signes de la LSF ordonnés selon la syntaxe de la langue française. Ce compromis naît de la nécessité de communiquer, il est utilisé par des entendants de langue maternelle française qui ont d'ailleurs parfois une bonne connaissance des signes mais ne maîtrisent pas la syntaxe de la langue des signes française. Par exemple, en LSF, la phrase « J'aime cette voiture. » sera signée voiture cette aimer. Dans le français signé, le locuteur utilisera l'ordre : aimer cette voiture. Dans l'enseignement aux jeunes sourds, le problème qui se pose est que leurs enseignants sont souvent des entendants et qu'ils n'utilisent pas naturellement la syntaxe de la LSF, mais plutôt naturellement celle du français signé, ainsi les jeunes sourds n'ayant pas de parents sourds calquent leur façon de signer sur leurs enseignants entendants.

6. qui se rapporte à l'icône, à l'image en tant que signe.

I.8 Conclusion

L'apparition de la langue des signes a certes contribué à rapprocher les sourds et mal-entendants entre eux mais par la même occasion , elle les a isolé de la société, car son apprentissage s'est limité à ces derniers.

Pour palier ce problème plusieurs technologies se basant sur la vision par ordinateur ont vus le jour. Dans le chapitre qui suit nous allons en parler.

Chapitre II

Vision par ordinateur et état de l'art de la reconnaissance des gestes de la main

Sommaire

II.1	Introduction	28
II.2	Définition de la vision par ordinateur :	28
II.3	Les images numériques et leurs traitements :	28
II.3.1	Définition d'une image :	28
II.3.2	Types d'images :	29
II.3.3	Codage des couleurs d'une image numérique :	30
II.3.4	Caractéristiques d'une image numérique : [17]	31
II.3.5	Format d'une image numérique :	31
II.3.6	Interprétations d'une image :	32
II.3.7	Traitement de l'image numérique :	33
II.4	Etat de l'art de la reconnaissance des gestes de la main	36
II.4.1	Acquisition du geste :	36
II.4.2	Reconnaissance du geste :	37
II.5	Conclusion	38

II.1 Introduction

La vision par ordinateur est une discipline en plein essor. Elle est utilisée dans de nombreuses applications nécessitant l'analyse et le traitement automatique des images numériques. Elle est la base de plusieurs travaux de reconnaissance, mais ceux liés à la reconnaissance de la Langue des signes sont relativement récents ; comme nous allons le voir dans la partie : état de l'art ; du fait que cette dernière n'a été officiellement reconnue qu'en 2005.

II.2 Définition de la vision par ordinateur :

La vision est la faculté d'extraire de l'information à partir des signaux contenue dans les rayons lumineux qui frappe nos rétines et que captent nos yeux ; en d'autres termes, c'est le processus de perception de ce qui nous entoure à travers ce que l'oeil recueille.[13]

La vision par ordinateur, pour sa part, est un domaine de l'intelligence artificielle qui vise à fournir une modélisation algorithmique et une automatisation de la vision biologique ; d'une autre manière, c'est la capacité d'une machine (ordinateur) à extraire des informations des images et des flux vidéo, de manière à percevoir, analyser et comprendre le monde qu'elle observe.

On considère comme sous-domaine de la vision par ordinateur, la reconnaissance d'objets qui est une méthode permettant de détecter la présence d'une instance ou classe dans une image numérique. Pour de meilleurs résultats lors de la détection, le traitement d'image est nécessaire. Ce dernier se place après l'étape d'acquisition, assurant les transformations d'images permettant d'aller vers une interprétation des images traitées.

II.3 Les images numériques et leurs traitements :

II.3.1 Définition d'une image :

Une image est la projection et la représentation d'un ensemble de rayons lumineux provenant du monde extérieur, elle est constituée d'un ensemble de points appelés pixels (PICTure Element). Le pixel représente ainsi le plus petit élément constitutif de cette dernière, en effet chaque point d'une image possède deux coordonnées (X, Y) de tel sorte que chaque couple de coordonnées (X, Y) représente l'emplacement d'un point dans cette image sur un plan et dont l'image de ce couple correspond à la valeur du rayon lumineux capté. Plus formellement, l'image est un cliché électroniques d'une scène, elle est échantillonnée et mappée comme une grille de points ou éléments d'images ou chaque point correspond à une valeur tonale (intensité lumineuse en noir et blanc, niveaux de gris ou couleur), exprimée en code binaire (zéros et uns). Les chiffres binaires ("bits") de chaque pixel sont stockés dans une séquence par l'ordinateur, et

souvent réduits à une représentation mathématique (compressés). Les bits sont alors réinterprétés et lus par l'ordinateur afin de délivrer une version analogique en vue d'être affichée ou imprimée.

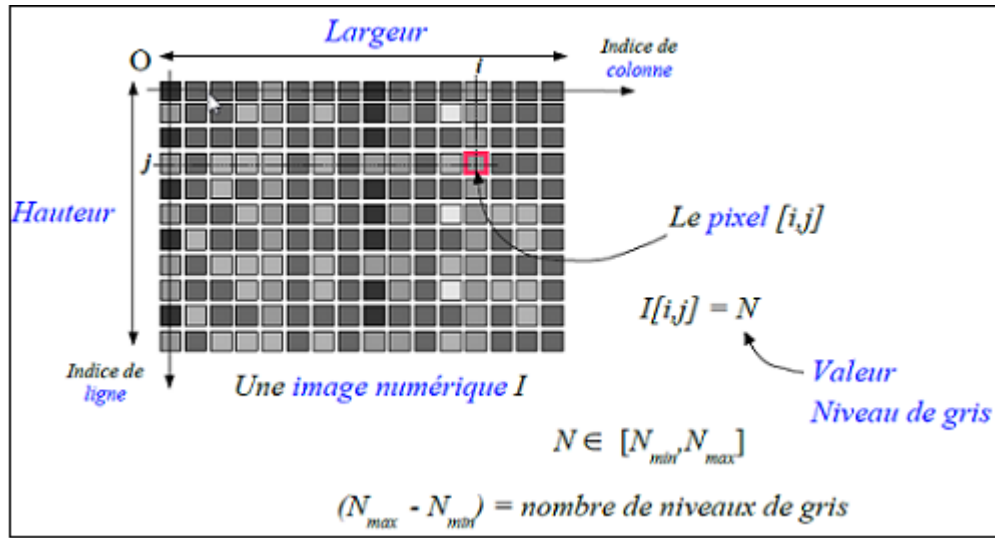


FIGURE II.1 – Vocabulaire d'une image.

II.3.2 Types d'images :

Il existe deux types d'image numérique, selon les éléments avec lesquels elles sont représentées :

- Image vectorielle : Les images vectorielles sont des représentations d'entités géométriques telles qu'un cercle, un rectangle ou un segment. Ceux-ci sont représentés par des formules mathématiques. Ce qui les rend très flexibles sans perte de qualité.[14]
- Image matricielle : Il s'agit d'images pixellisées, c'est-à-dire un ensemble de points (pixels) contenus dans un tableau, chacun de ces points possédant une ou plusieurs valeurs décrivant sa couleur.[15]

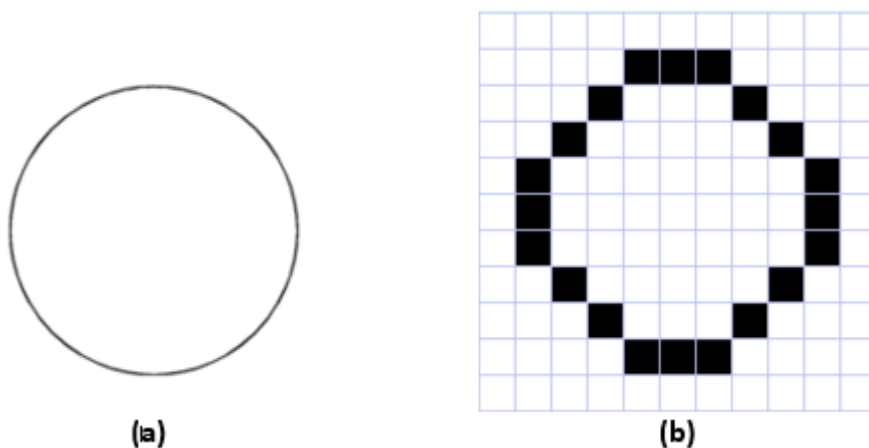


FIGURE II.2 – Représentation d'un cercle avec une image vectorielle(a) et une image matricielle(b).

II.3.3 Codage des couleurs d'une image numérique :

Comme dit ci-dessous, chaque point (pixel) de l'image possède une valeur qui décrit sa couleur, l'image peut être alors :[16]

- En noir et blanc (Images binaires) : Dans cette représentation un pixel peut prendre uniquement la valeur noir ou blanc.
- En niveaux de gris : Dans cette représentation, un pixel est noir, blanc, ou a un niveau de gris entre les deux. Il est généralement codé sur 1 octet (8 bit), ce qui lui donne 256 valeurs possibles, dont le blanc(255) et le noir(0).
- En couleur : Dans cette représentation, la couleur d'un pixel est obtenue en mélangeant les trois couleurs fondamentales à savoir le Rouge, Vert, Bleu. Chaque pixel est codé sur 3 octets, ce qui lui donne $((2^3)^3)$ valeurs possibles .



FIGURE II.3 – Image binaire (a), en niveaux de gris(b) et en couleur(c).

II.3.4 Caractéristiques d'une image numérique : [17]

- Définition (Taille) : La définition d'une image est définie par le nombre de pixel la composant. En image numérique, cela correspond au nombre de pixels qui composent l'image en hauteur (axe vertical) et en largeur (axe horizontal) : 200 pixels par 450 pixels par exemple (200x450).
- Résolution : La résolution d'une image est définie par un nombre de pixels par unité de longueur de la structure à numériser (classiquement en pixel par pouce). Plus le nombre de pixels par unité de longueur de la structure à numériser est élevé, plus la quantité d'information qui décrit cette structure est importante et plus la résolution est élevée. La résolution d'une image numérique définit le degré de détail de l'image.
- Profondeur : La profondeur est le nombre de bits utilisés pour représenter la couleur d'un pixel dans une image.

II.3.5 Format d'une image numérique :

Le format d'image est une représentation informatique de l'image, associée à des informations sur la façon dont l'image est codée et fournissant éventuellement des indications sur la manière de la décoder et de la manipuler.

Parmi les formats les plus répandus, nous trouvons : [18]

.SVG (Scalable Vector Graphics) : est un format de données conçu pour décrire des ensembles de graphiques vectoriels et basé sur le langage XML.

.JPG (Joint Photographic Experts Group) : Un fichier JPG est un fichier d'image compressé qui ne supporte pas un fond transparent. Le niveau de compression de fichiers JPG peut varier dans la résolution de haute qualité pour l'impression de documents commerciaux, de qualité moyenne pour l'affichage web et de faible qualité pour le courrier électronique. Lorsqu'il est comprimé à plusieurs reprises la qualité globale d'une image JPG est réduite.

.PNG (Portable Network Graphics) : Le format PNG est le plus couramment utilisé pour une utilisation en ligne et sur les sites en raison de leur faible résolution. Les fichiers PNG sont des images bitmap(vectoriel) qui emploient la compression sans perte de données, les fichiers PNG peuvent être créés avec un fond transparent.

.GIF (Graphic Interchange Format) : Les fichiers GIF sont des fichiers de basse résolution les plus couramment utilisés à des fins web et pour les e-mail. Presque tous les navigateurs peuvent soutenir l'utilisation des fichiers GIF, qui utilisent un système de compression pour maintenir la taille du fichier. Les fichiers GIF peuvent être créés avec un fond transparent.

II.3.6 Interprétations d'une image :

a) Histogramme d'une image : [19]

Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse. Par convention un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé (à gauche) au plus clair (à droite).

Ainsi, l'histogramme d'une image en 256 niveaux de gris sera représentée par un graphique possédant 256 valeurs en abscisses, et le nombre de pixels de l'image en ordonnées.

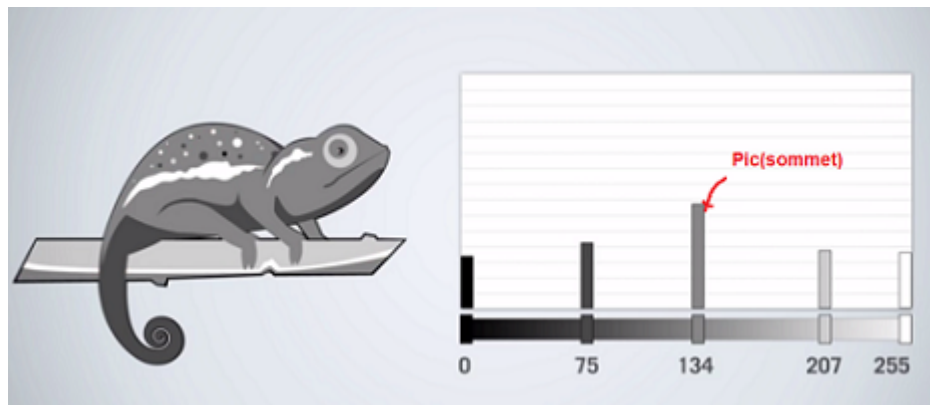


FIGURE II.4 – Histogramme d'une image contenant 5 niveaux de gris.

L'histogramme d'une image en couleur aura 3 courbes, chacune représentant la distribution d'intensité de l'une des couleurs Rouge, Vert ou Bleu, les autres couleurs présentes dans l'image seront les points d'intersection de ces trois courbes.

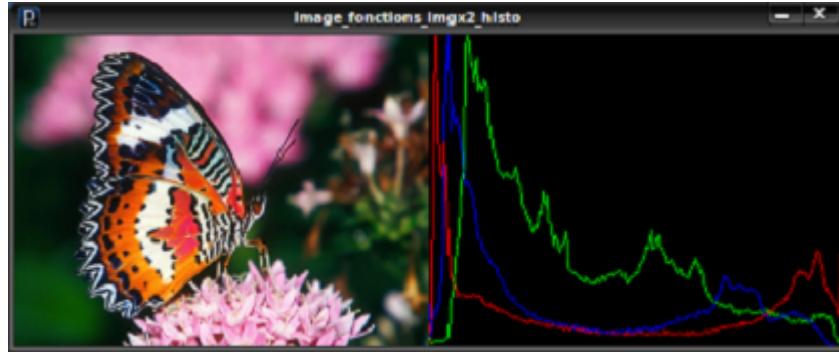


FIGURE II.5 – Histogramme d’une image en couleur.

II.3.7 Traitement de l’image numérique :

Le traitement d’images est une discipline de l’informatique qui étudie les images numériques et leurs transformations, dans le but d’améliorer leur qualité (Convolution et Filtrage) ou d’en extraire de l’information (Segmentation).

a) Convolution d’une image : [20]

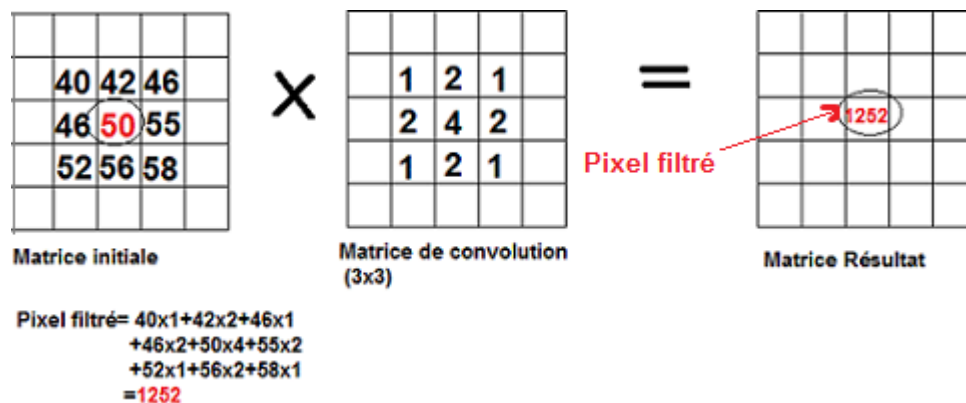
La convolution de l’image est un traitement d’une matrice par une autre appelée une matrice de convolution (ou noyau).

La plus part des filtres de traitement d’image utilisent des matrice de convolution.

Le filtre ou la matrice de convolution utilise une première matrice qui est l’image, c’est à dire, une collection de pixel en coordonnées rectangulaire 2D, et un noyau variable selon l’effet souhaité.

Les matrices de convolution les plus utilisées sont de la taille 3x3 ou 5x5 .

Le filtre étudie successivement chacun des pixels de l’image et pour chaque pixel que nous appelon "pixel initial", il multipli la valeur de ce pixel et de chacun des pixel qui l’entourent par la valeur correspondante dans le noyau. Il additionne l’ensemble des résultats et le pixel initial prend alors la valeur du résultat final.



- A gauche, se trouve une partie de la matrice de l'image. Le pixel initial est encadré en rouge.
- Au centre, se trouve le noyau (la matrice de convolution) qui diffère selon le type de filtrage que l'on souhaite appliquer à l'image. Certains de ces filtres permettant la convolution portent le nom de ceux qui les ont créés comme le filtre de Sobel que nous allons voir par la suite.
- A droite, se trouve le résultat de la convolution.

b) Filtrage d'une image : [21]

Le filtrage est utilisé pour éliminer les bruits (parasites) sur l'image (lissage) pour des utilisations et applications ultérieures.

Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement dans le but d'améliorer son aspect. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine et d'une matrice de convolution.

Nous distinguons deux types de filtrage :

- Filtrage global : Dans le filtrage global, chaque pixel de la nouvelle image est calculé en prenant en compte la totalité des pixels de l'image de départ. Dans cette catégorie on trouve, par exemple, les opérations sur les histogrammes.
- Filtrage Local : Dans le filtrage local, chaque pixel de la nouvelle image est calculé en prenant en compte seulement un voisinage du pixel correspondant dans l'image d'origine. Il est d'usage de choisir un voisinage carré et symétrique autour du pixel considéré. Ces voisinages sont donc assimilables à des tableaux à deux dimensions (matrices) de taille impaire. Voisinage 3x3 du pixel x,y :

$$\begin{pmatrix} P_{x-1,y-1} & P_{x,y-1} & P_{x+1,y-1} \\ P_{x-1,y} & P_{x,y} & P_{x+1,y} \\ P_{x-1,y+1} & P_{x,y+1} & P_{x+1,y+1} \end{pmatrix}$$

Parmi les filtres les plus usuels dans le traitement d'image :

Le filtre Gaussien(Gaussian Blur) :

Il appartient au deuxième type de filtrage à savoir le filtrage local. Son principe est le suivant :

Calculer la moyenne pondérée des valeurs du voisinage. Les pixels du voisinage qui sont proches du pixel central ont un poids plus fort (= plus d'influence) que ceux qui sont plus éloignés.

Il utilise une matrice de convolution(aussi appelée noyau) pour effectuer le filtrage.

Exemple :

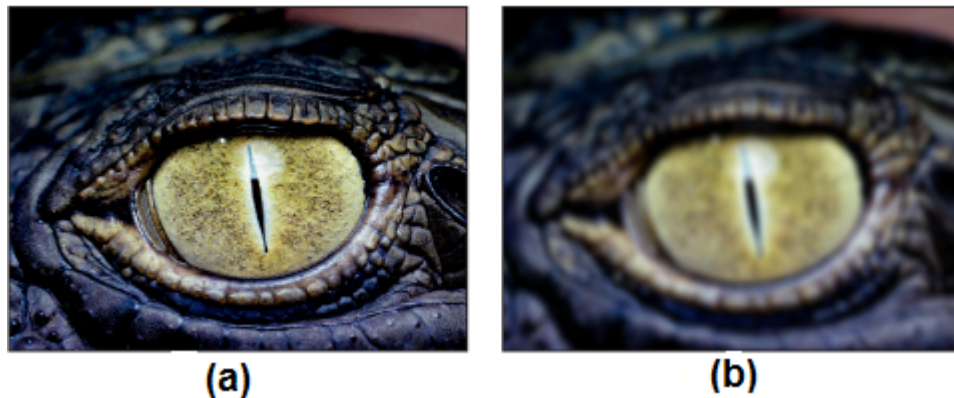


FIGURE II.6 – Application du filtre gaussien à l'image d'origine (a).

0.00000067	0.00002292	0.00019117	0.00038771	0.00019117	0.00002292	0.00000067
0.00002292	0.00078634	0.00655965	0.01330373	0.00655965	0.00078633	0.00002292
0.00019117	0.00655965	0.05472157	0.11098164	0.05472157	0.00655965	0.00019117
0.00038771	0.01330373	0.11098164	0.22508352	0.11098164	0.01330373	0.00038771
0.00019117	0.00655965	0.05472157	0.11098164	0.05472157	0.00655965	0.00019117
0.00002292	0.00078633	0.00655965	0.01330373	0.00655965	0.00078633	0.00002292
0.00000067	0.00002292	0.00019117	0.00038771	0.00019117	0.00002292	0.00000067

FIGURE II.7 – Matrice de convolution utilisée pour appliquer le filtre gaussien à une image.

c)Segmentation de l'image :[22]

La segmentation d'image est une opération de traitement d'images qui a pour but de rassembler des pixels entre eux suivant des critères pré-définis. Les pixels sont ainsi regroupés en régions, qui constituent un pavage ou une partition de l'image. Il peut s'agir par exemple de séparer les objets du fond. Si le nombre de classes est égal à deux, elle est appelée aussi binarisation. La segmentation est une étape primordiale en traitement d'image. À ce jour, il existe de nombreuses méthodes de segmentation, que l'on

peut regrouper en trois principales classes :

- La segmentation fondée sur les régions (en anglais : region-based segmentation). la segmentation de régions sépare les zones homogènes de l'image en étudiant la proximité colorimétrique entre un pixel et ses voisins ; les pixels qui se rapprochent sont alors regroupés. (voir l'image c).
- La segmentation fondée sur les contours (en anglais : edge-based segmentation) : Comme son nom l'indique, elle est responsable de la détection de contour des objets ; cette approche est basée sur le changement rapide d'intensité lumineuse ou de couleur dans cette partie de l'image. (voir l'image a)
- La segmentation fondée sur la classification ou le seuillage des pixels en fonction de leur intensité (en anglais : classification ou thresholding). C'est le procédé le plus simple, l'approche consiste à dire que tout objet se différencie de l'arrière-plan. Elle consiste à tester pour chaque pixel de l'image si sa valeur est supérieure ou inférieure à un certain seuil déterminé au préalable, et retourne une image binarisée. (voir l'image b)

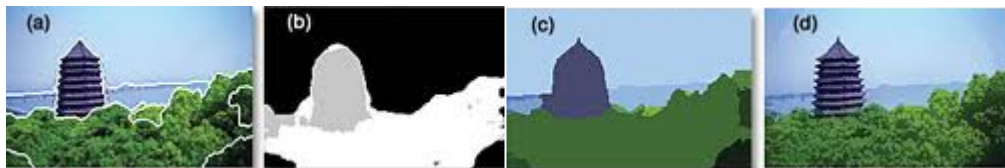


FIGURE II.8 – L'image d'origine(d), segmentation fondée sur les contours de l'image (a), segmentation fondée sur la classification de l'image (b) , segmentation fondée sur les régions de l'image(c).

II.4 Etat de l'art de la reconnaissance des gestes de la main

II.4.1 Acquisition du geste :

L'acquisition du geste par vision artificielle a été le sujet de plusieurs travaux de recherches, ces derniers peuvent être classés en deux principales catégories, d'une part l'analyse des aspects d'une image et d'autre part la modélisation 3D de leur contenu.

Les méthodes basées sur la 3D tirent avantage de la connaissance préalable de la forme d'un modèle et de la possibilité d'appliquer des transformations géométriques afin de le déformer, il est alors nécessaire d'utiliser la stéréovision¹ ou plus de deux caméras pour acquérir l'information ce qui constitue un procédé

1. méthode de mesure qui consiste à se servir de la prise d'images (photographiques ou numériques) sous différents angles de vue pour déterminer les dimensions, les formes ou les positions d'objets.

lourd et couteux.

Les méthodes basées sur la 2D quant à elles reconnaissent un nombre limité de gestes après un procédé d'apprentissage.

II.4.2 Reconnaissance du geste :

Les études réalisées dans le domaine de la reconnaissance automatique des gestes sont relativement récentes, les premiers étant liés à l'apparition des gants numérique en 1987 avec le DataGlove de VPL ². par la suite, nombre d'entre elles se sont d'abord attachées à la reconnaissance de gestes dans le cadre d'interaction homme-machine. D'autres ont porté sur la classification par MMC, Modèle de Markov Caché, de mouvement de tennis en s'appuyant sur des technique de quantification vectorielle dans des séquences d'images binaires.

Dans le domaine plus spécifique de la reconnaissance du langage des sourds, braffort a présenté un système pour la langue des signes française basé sur l'acquisition de paramètres à partir de gant instrumentés, les mots y sont analysé et caractérisé selon plusieurs critères, le système de reconnaissance étant basé sur des MMC, Modèle de Markov Caché, divisée en deux modules, l'un d'une part pour classifier les signes conventionnels et le second d'autre part pour les signes variable et non conventionnels, les taux de reconnaissance obtenus sont respectivement de 96% et de 92% parmi un dictionnaire de 44 phrase et un vocabulaire de 7 signes .

Starner et Pentland, ont développé un système basé sur la vidéo pour la reconnaissance des phrases de la langues des signes américaine, à partir d'un vocabulaire de 40 signes, les gestes sont modélisés par un MMC, Modèle de Markov Caché, à 4 états, une seule caméra est utilisé pour l'enregistrement des images, le taux de reconnaissance est compris entre 75% et 99% .

Vogler et Metaxas ont décrit un système basé sur les MMC et destiné à la reconnaissance de l'ASL, Américain Sign Langage, d'un vocabulaire de 53 signes, 3 caméra sont utilisé pour l'acquisition de paramètres 3D des mouvement des bras et des mains des signeurs, 97 phrase de test ont été utilisé, les résultats obtenus varies entre 92.1% et 95.8% .

La majorité des travaux sur la reconnaissance du geste utilise les méthodes de markov caché, basé sur l'apprentissage de Baum-Wlech et l'algorithme de viterbi pour approximer la vraisemblance d'une observation étant donnée le modèle . [23]

2. est l'une des première compagnie à avoir développer et vendu des produits de réalité virtuelles.

Dans notre travail nous allons adopter une approche qui combine la stéréovision et les méthodes d'apprentissage direct (contrairement au MMC) à savoir les séparateur à vaste marges et les forêts aléatoires d'arbres de décision.

II.5 Conclusion

Afin de pouvoir mener à bien notre projet,nous avons fait le tour des différentes notions de la vision par ordinateur, nécessaires à la réalisation de ce dernier.

Puis nous avons fait une revue de l'existant dans la reconnaissance des gestes de la main pour positionner notre approche dans ce dernier.

Dans le chapitre qui suit, nous allons détailler les méthodes que nous allons utiliser dans mon approche.

Chapitre III

Méthodes de détection de la main et de reconnaissance des gestes

Sommaire

III.1 introduction :	42
III.2 Segmentation d'image :	42
III.2.1 Définition :	42
III.2.2 Différentes approches de segmentation d'images :	43
A.Approche contours :	43
B.Approche régions :	50
C.Approche seuillage :	55
III.3 Comparaison des trois approches de segmentation d'images :	61
III.4 Intelligence artificielle :	62
III.5 Apprentissage artificiel / automatique (Machine Learning) :	62
III.6 Types d'apprentissages :	62
III.6.1 Selon la nature des données traitées :	62
A.Apprentissage symbolique :	63
B.Apprentissage numérique :	63
III.6.2 Selon l'intervention ou non d'un expert :	63
A.Apprentissage non supervisé :	63
B.Apprentissage supervisé :	64
III.7 Scénario de l'apprentissage supervisé :	64
III.8 Classification supervisée :	66
III.8.1 Définition de la classification :	66
III.8.2 Classification supervisée :	66
III.8.3 Classifieur :	66

III.8.4	Processus de la classification automatique :	66
III.8.5	Formalisation de la classification automatique :	67
III.8.6	Types de la classification supervisé :	67
	A.Classification binaire :	67
	B.Classification multi-classes :	67
	C.Classification multi-label :	67
	D.Méthodes de classification probabiliste :	68
	E.Méthodes de classification directe :	68
III.8.7	Problèmes liés à l'utilisation des méthodes probabilistes :	69
III.9	Algorithmes de classification directe :	69
III.9.1	Séparateur à vaste marge (SVM) : [27]	69
	A.Le concept de linéarité :	69
	B.Le principe des SVM :	70
	C.Le problème de classification :	71
	D.Récapitulatif :	75
III.9.2	Les arbres de décisions :	75
	A.Le principe des arbres de décision	75
	B.Construction d'un arbre :	76
	C.Utilisation :	77
	D.Formalisation :	77
	E.Récapitulatif :	78
III.9.3	Les forêts d'arbres décisionnels (Random decision forest) :	78
III.10	Evaluation des performances d'un classifieur :	79
III.10.1	Taux de bonne classification sans coûts :	79
	A.Taux de classification : (Accuracy score)	80
	B.taux d'erreurs :	80
	C.Matrice de confusion :	80
III.11	Conclusion :	80

Notations utiles pour ce chapitre

I : une image.

R : une région de l'image.

$R_i \cap R_j$: L'intersection des deux régions de l'image, R_i et R_j .

\emptyset : L'ensemble vide.

P : Un prédicat.

f : La fonction f qui représente l'image I .

$\overrightarrow{Grad}f$: le gradient de la fonction f .

$\vec{\nabla} f$: Le symbole du gradient de la fonction f .

∂ : Symbole de la dérivée.

$\frac{\partial f}{\partial x}$: La dérivée de la fonction f par rapport à x .

$\frac{\partial f}{\partial y}$: La dérivée de la fonction f par rapport à y .

G_x : Image après l'application d'un masque de convolution

G_y : Image après l'application du même masque de convolution qui a subi une rotation de $\frac{\pi}{2}$

G : La norme du gradient.

α_0 : La direction du gradient.

$\nabla^2 f$: La deuxième dérivée de la fonction f et qui représente la valeur de Laplacien.

RVB : Rouge Vert Bleu ; qui est un système colorimétrique.

S : Un niveaux de gris représentant un seuil.

$M \times N$: Dimension de l'image.

σ_w^2 : La variance intra-classe.

III.1 introduction :

La détection de zones d'intérêt d'une image numérique (feature detection en anglais) consiste à mettre en évidence des zones de cette image jugées « intéressantes » pour l'analyse, c'est-à-dire présentant des propriétés locales remarquables. De telles zones peuvent apparaître, selon la méthode utilisée.

Il existe plusieurs méthodes de la détection de zone d'intérêt, parmi lesquelles on trouve la segmentation d'image qui s'avère être très efficace. En effet elle nous permet de détecter de telles zones sous la forme de points, de courbes continues, ou encore de régions connexes et qui constituent le résultat de la détection. Ce résultat est ensuite utilisé dans la phase de reconnaissance qui s'appuie sur plusieurs algorithmes d'apprentissage que nous verrons dans la deuxième partie de ce chapitre.

III.2 Segmentation d'image :

III.2.1 Définition :

La segmentation d'images consiste à regrouper les pixels des images qui partagent une même propriété pour former des régions connexes. Il existe plusieurs définition de la segmentation parmi lesquelles nous trouvons celle de Zucker [24], qui l'a définit comme le partitionnement de l'ensemble des pixels d'une image I en n sous ensembles appelés régions R_i , $i=1..n$: $I = R_1, R_2, R_3, \dots, R_n$

Tel que :

- Aucune région ne doit être vide.
- L'intersection entre deux régions doit être un ensemble vide.
- L'ensemble des régions doit recouvrir toute l'image.

Ce qui se traduit par l'algorithme :

$$\bigcup_{i=1}^n R_i = I \quad \text{Avec } R_i \cap R_j = \emptyset \quad i \neq j$$

$$\text{ET : } \begin{cases} P(R_i) = \text{vrai} & \forall i = 1..n \\ P(R_i \cap R_j) = \text{faux} & R_i \text{ Adjacente à } R_j \end{cases}$$

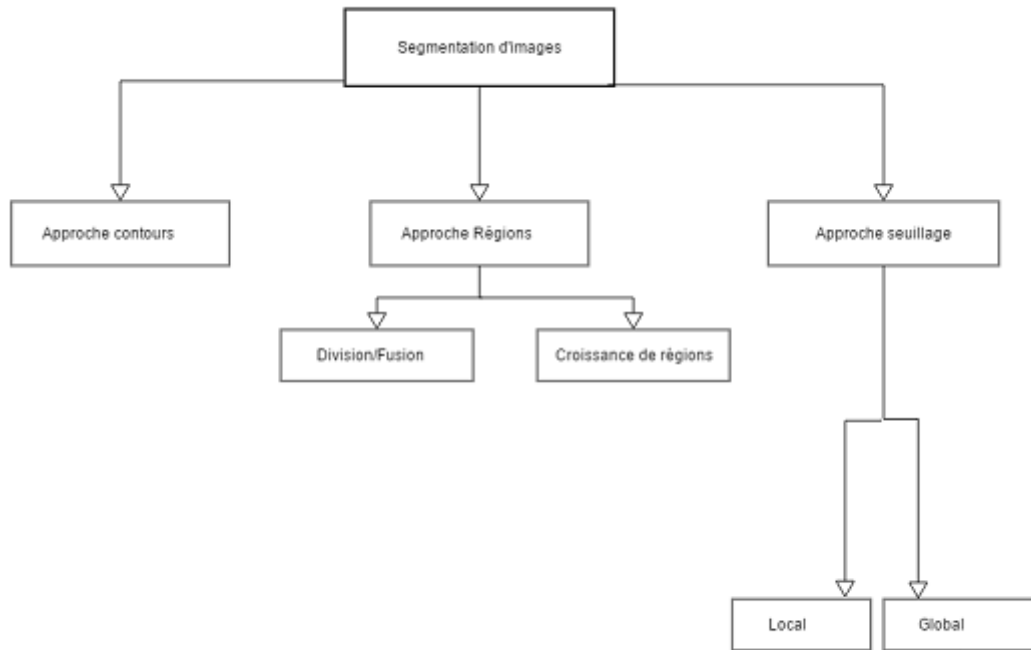
Avec :

- L'union de toutes les régions R_i , avec $i=1..n$, forme l'image I .
- L'intersection de deux régions R_i et R_j est un ensemble vide pour $i \neq j$.
- Soit P un prédicat d'homogénéité, soient R_i et R_j deux régions adjacentes de l'image. Si la région R_i satisfait le prédicat P , l'intersection de R_i et R_j ne le satisfera pas.

III.2.2 Différentes approches de segmentation d'images :

La segmentation d'image est un domaine vaste où l'on trouve de très nombreuses approches.

Toutes ces approches visent à extraire des indices visuels et cela de différentes façons, le schéma suivant en présente les plus connues :



A.Approche contours :

Les contours sont les lieux de variation significative de l'information niveau de gris.

La notion de contours étant relié à celle de la variation, il est évident qu'une telle variation nous amène vers une évaluation de la variation en chaque point. Ce qui se traduit par le calcul de deux grandeurs qui sont le gradient¹ et laplacien (la deuxième dérivé du gradient).

Les contours sont alors repérés là où le gradient présente un maximum local ou lors des passages à zéro de laplacien.

Dans ce qui suit on va détailler les deux méthodes :

1. En physique et en analyse vectorielle, le gradient est une grandeur vectorielle indiquant la façon dont une grandeur physique varie dans l'espace. En mathématiques, le gradient est un vecteur représentant la variation d'une fonction par rapport à la variation de ses différents paramètres.

A.1 Méthode du gradient :

En considérant l'image dans un repère orthogonal (Oxy) tel que (Ox) désigne l'axe horizontal et (Oy) l'axe vertical, le gradient de l'image en tout point ou pixel de coordonnées (x,y) est un vecteur à deux dimensions dont les coordonnées sont les dérivées de la fonction f (f étant la représentation de l'image dans ce repère (Oxy)) selon la direction horizontale et verticale. En effet le calcul de cette grandeur en chaque pixel nous indiquera la direction de la plus forte variation du clair au sombre, ainsi que le taux du changement dans cette direction. Il est désigné par :

$$\vec{Grad}f = \vec{\nabla}f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}.$$

Pour calculer la valeur du gradient, des filtres comme celui de Sobel sont appliqués à l'image grâce à une matrice de convolution (comme nous allons le voir dans ce qui suit). Cette matrice, généralement de taille 3x3, nous permet de réaliser une convolution avec l'image pour calculer des approximations de la dérivée horizontale et verticale de la fonction d'intensité de l'image.

Soit A l'image source, Gx et Gy deux images qui en chaque point contiennent des approximations respectivement de la dérivée horizontale et verticale de chaque point. Ces images sont calculées comme suit :

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \times A \quad G_y = \begin{bmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \times A$$

En chaque point, les approximations des gradients horizontaux et verticaux peuvent être combinés comme suit pour obtenir une approximation de la norme du gradient :

$$G = \sqrt{G_x^2 + G_y^2}$$

On peut également calculer la direction du gradient comme suit :

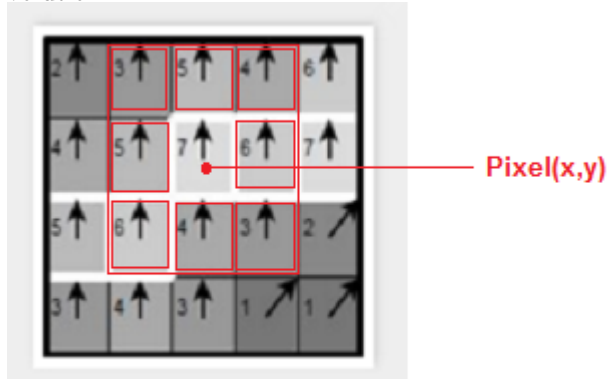
$$\alpha_0 = \arctan(G_y, G_x)$$

Le principe de la détection de contours par l'utilisation du gradient consiste à :

Calculer le gradient de l'image dans les deux directions orthogonales puis du module du gradient. Une forte intensité dans la valeur du module du gradient indique une forte probabilité de présence d'un

contour. Toutefois cette intensité ne suffit pas à décider si un point appartient au contour. Seuls les points correspondants à des maximums locaux sont considérés comme appartenant au contour.

Exemple : La matrice qui suit représente une partie d'une image après l'application d'un filtre de convolution.



Le pixel (x,y) appartient au contour car son gradient présente un maximum local comparé au 8 voisins qui l'entourent. Le pixel prend alors la couleur blanche.

Par la suite il faudra effectuer une sélection des contours les plus marqués, c'est-à-dire les points des plus forts contrastes par un seuillage adéquat.

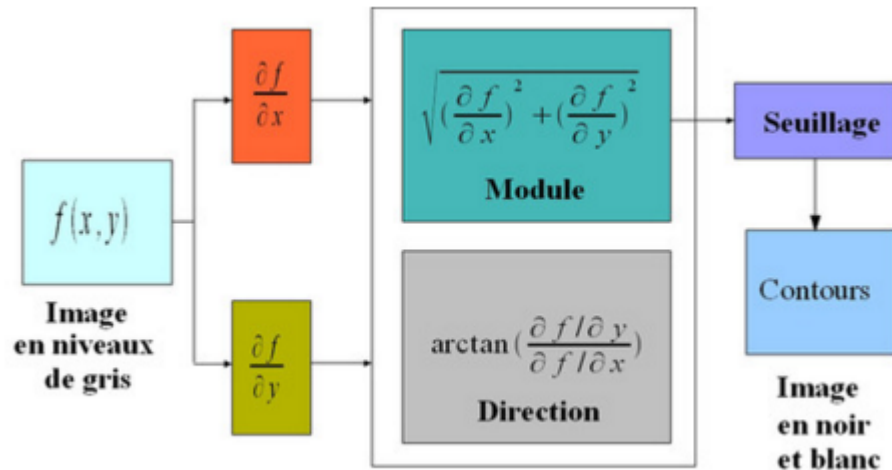


FIGURE III.1 – Schéma récapitulatif d'une détection de contours par gradient.

Exemple : Détection de contours avec le filtre de Canny :

L'objectif est bien sûr de calculer le module du gradient de l'image analysée. Souvent avant d'appliquer le filtre de Canny, un filtrage préalable est opéré sur l'image au moyen d'un filtre gaussien. Les différentes étapes sont les suivantes :

- Convolution de l'image initiale avec un filtre passe-bas² gaussien.
- Convolution de l'image résultante avec le filtre de Canny.
- Calcul du module du gradient à partir des deux images représentant le gradient de l'image filtré passe-bas dans les directions horizontale et verticale.

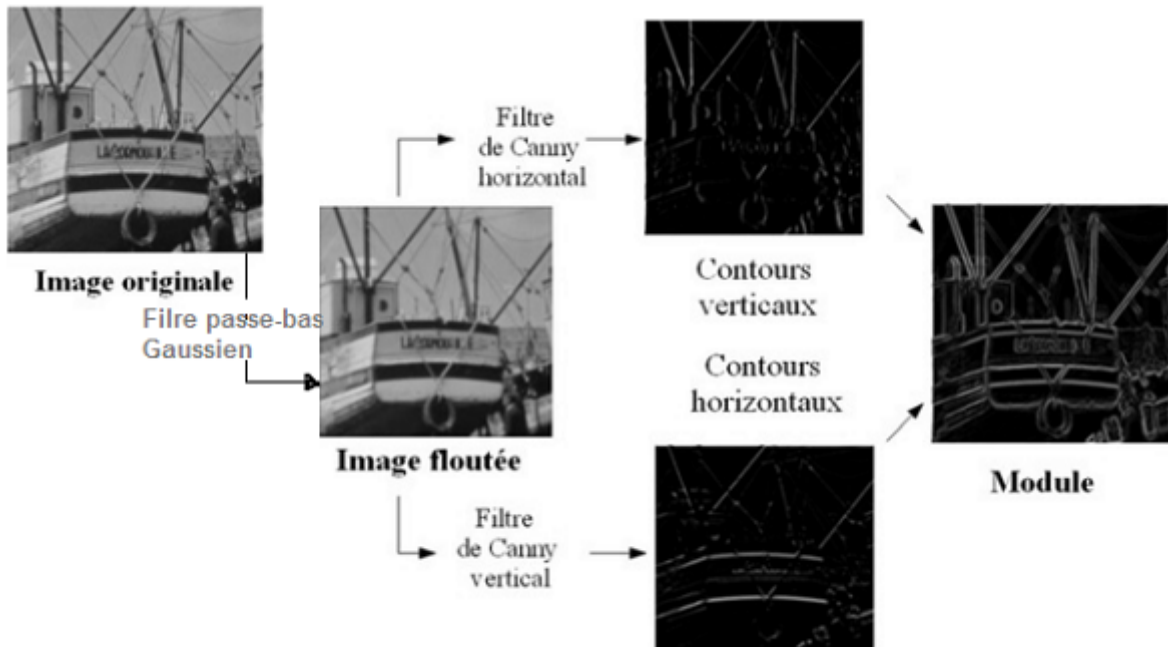


FIGURE III.2 – Schéma récapitulatif d'une détection de contours avec le filtre de Canny.

A.2 Méthode de Laplacien :

La méthode de gradient vue précédemment exploite le fait qu'un contour dans une image correspond au maximum du gradient dans la direction orthogonale au contour. La méthode de laplacien consiste à calculer le passage par zéro de la dérivée seconde du gradient :

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial y} \right)$$

2. Un filtre passe-bas est un filtre qui laisse passer les basses fréquences et qui atténue les hautes fréquences.

Dans le cas d'une approche dérivée seconde, on dispose de la valeur de laplacien en chaque point de l'image soit de la fonction f .

On considère que les points contours sont localisés aux passages par zéro de laplacien. Cela revient à sélectionner les points M tels que $f(M)=0$.

Une dernière étape de seuillage est encore nécessaire afin d'éliminer les points de trop faible gradient.

Cette opération de calcul de laplacien peut alors être appliquée à une image par l'intermédiaire d'un filtre avec le masque 3*3 suivant :

0	1	0
1	-4	1
0	1	0

D'autres filtres peuvent être utilisés.[3]

Après le filtrage de l'image au moyen de l'un de ces filtres, il s'agit de détecter le passage par zéro.

Exemple : Détection de contours avec le filtre de Prewitt :

Prewitt est l'un des premiers algorithmes de détection de contours dans le domaine du traitement d'images. Il s'inspire fortement de la méthode de Laplacien que nous venons de voir. Il s'agit d'une approximation du gradient par convolution de l'image avec des masques de convolutions.

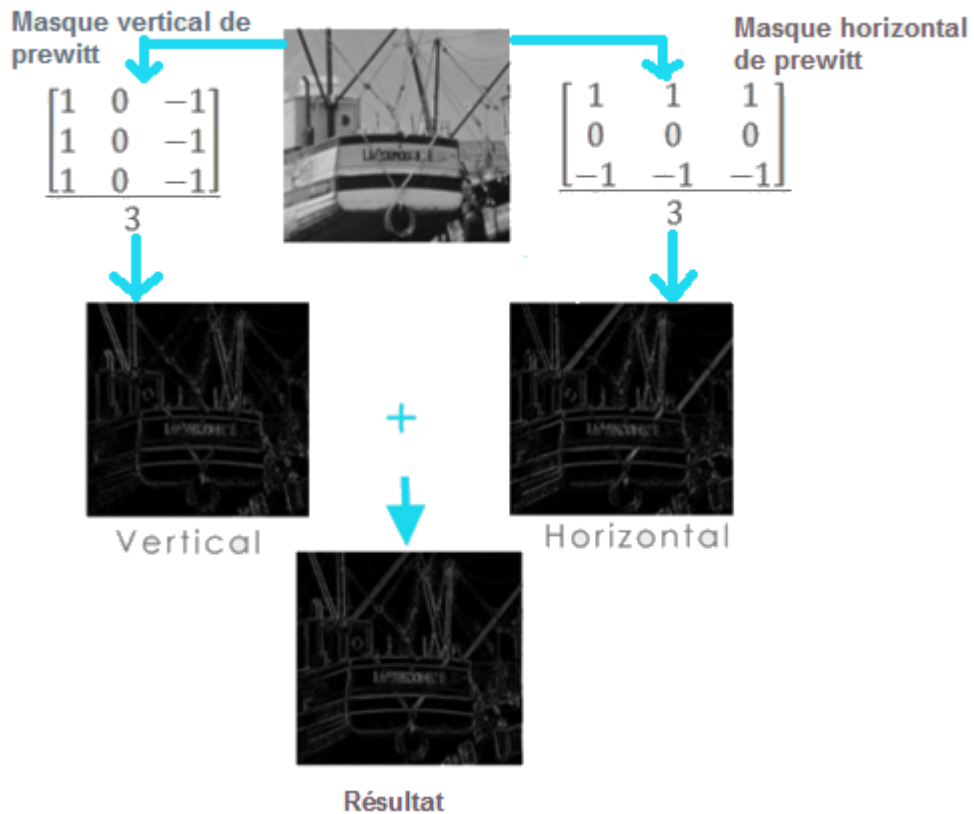


FIGURE III.3 – Détection de contours avec l'opérateur de prewitt

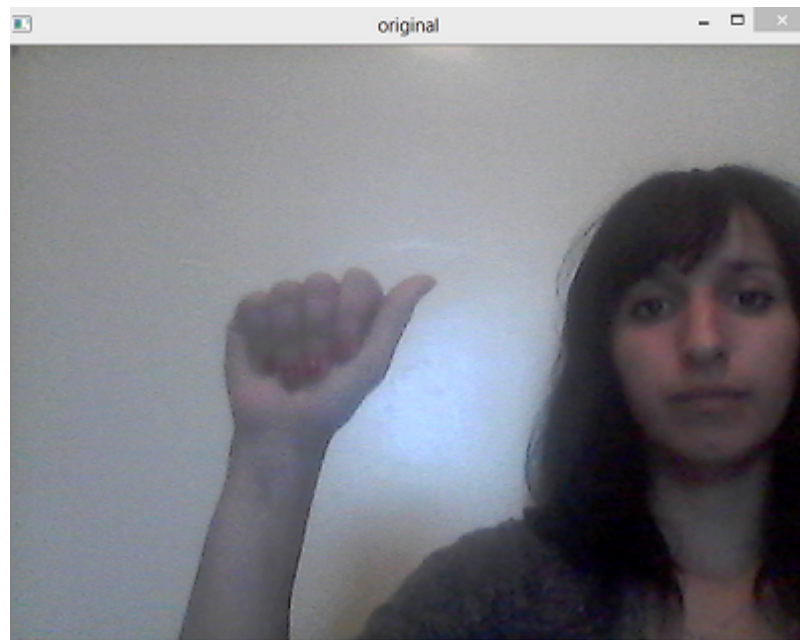
A.3 Test de l'approche contours en utilisant le filtre de Sobel :

Le filtre de Sobel détecte séparément les bords horizontaux et verticaux sur une image en niveaux de gris. Les images en couleurs RVB sont transformées en niveaux de gris. L'application du filtre aboutit à une image noir avec de fines lignes blanches et quelques restes de couleurs, comme avec le filtre Laplace, mais les lignes sont plus épaisses et plus noires.

Résultats du test :

s

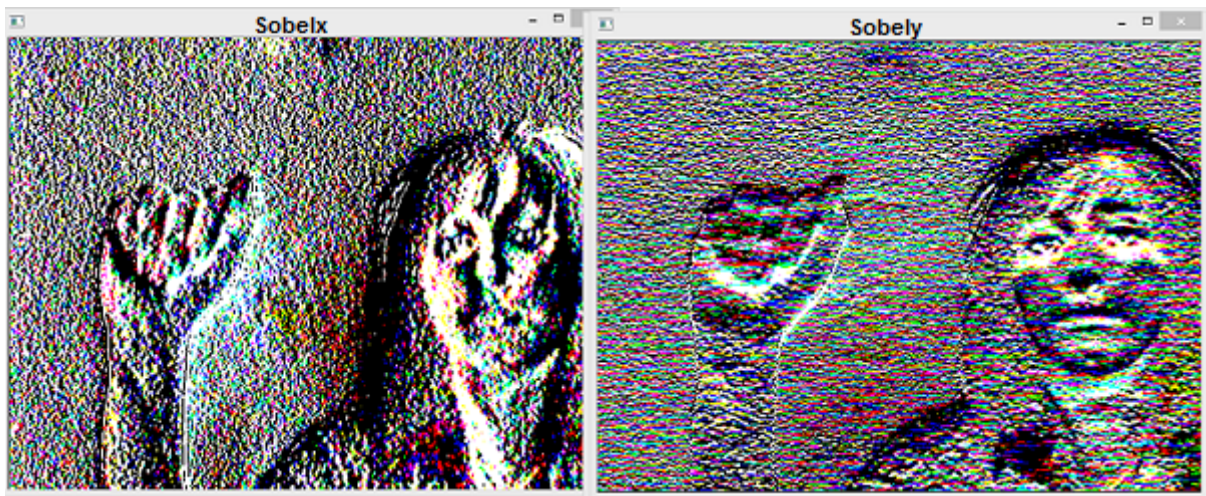
-Image d'origine :



-Application de l'opérateur Laplacien :



-Application du filtre de Sobel, horizontal(Sobelx) et vertical(Sobely) :



-Image contenant les contours recherchés :



B.Approche régions :

Les méthodes appartenant à cette famille manipulent directement des régions.

Soit elles partent d'une première partition de l'image, qui est ensuite modifiée en divisant ou regroupant des régions, et on parle alors de méthodes de **division/fusion** (ou split and merge en anglais) ; soit elles partent de quelques régions, qui sont amenées à croître par incorporation de pixels jusqu'à ce que toute l'image soit couverte, et on parle alors de méthodes par **croissance de régions**.

B.1 Division/Fusion :

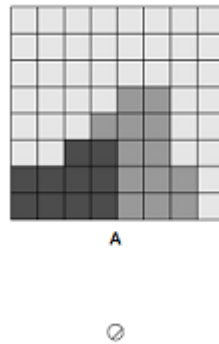
L'idée des algorithmes « split merge » est de produire automatiquement une partition initiale de petites régions (Split), qui vont ensuite croître en se regroupant (Merge).

La partition initiale est réalisée en divisant récursivement l'image en régions de tailles identiques lorsqu'un certain critère d'homogénéité n'est pas satisfait (un seuil par exemple).

Lors de cette phase (Split) le graphe d'adjacence, ou Region Adjacency Graph (RAG) est créé : à chaque région est associé un sommet du graphe, et des arêtes relient les sommets correspondants à deux régions qui se touchent.

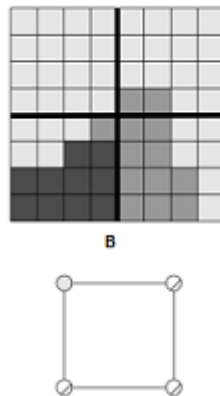
La phase de regroupement (Merge) utilise le RAG pour modifier la partition initiale : pour chaque sommet R du RAG, on cherche s'il existe un sommet R' voisin dans le RAG et de valeur suffisamment proche, et si c'est le cas, on les fusionne.

1. Image d'origine :



Au départ, On a un seul sommet dans le graphe d'adjacence (RAG) représentant l'image entière.

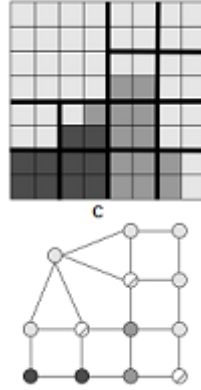
2. Première division (Split) :



On prend l'image pixel par pixel et on vérifie si il satisfait un critère d'homogénéité (ici un seuil représentant la couleur), si ce n'est pas le cas, on procède à une division de l'image. Nous obtenons un graphe

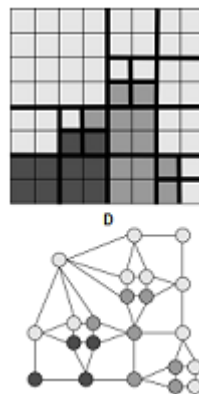
d'adjacence avec 4 sommets dont chacun représente une partition.

3. Deuxième division :



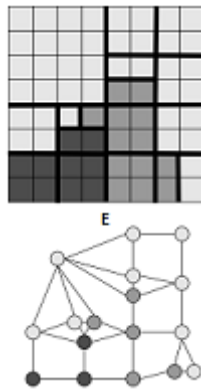
On prend les quatre partitions obtenues précédemment et on vérifie pour chacune si elle satisfait le critère d'homogénéité, si non, on procède à une autre division de l'image, cette fois ci on obtient un graphe d'adjacence avec 13 sommets dont chacun représente une nouvelle partition.

4. Troisième division :



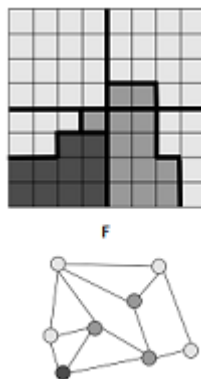
On prend les 13 partitions obtenues précédemment et on vérifie pour chacune si elle satisfait le critère d'homogénéité, si non, on procède à une autre division de l'image, cette fois ci on obtient un graphe d'adjacence avec 22 sommets dont chacun représente une nouvelle partition.

5. Premier regroupement :



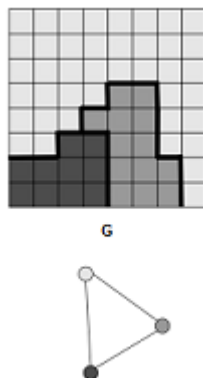
On remarque que dans chacune des partitions de l'image obtenues précédemment (L'image D), le critère d'homogénéité est satisfait (chacune à une couleur bien déterminée). On peut alors procéder au premier regroupement qui consiste à concatener deux partitions adjacentes si leurs valeurs (ici on parle de la couleur) sont assez proches. Le nombre de sommets du graphe d'adjacence diminue par conséquent.

6. Deuxième regroupement :



De la même manière, on regroupe les partitions adjacentes dont les valeurs sont assez proches .

7. Troisième regroupement :



On procède à un dernier regroupement des partitions adjacentes dont les valeurs sont assez proches et

on obtient une partition de l'image en 3 régions homogènes qui est le résultat visé lors de l'utilisation de cette approche de segmentation d'image.

B.2 Croissance de régions :

Cette technique consiste à faire progressivement grossir les régions autour de leur point de départ.

L'algorithme est le suivant :

Créer une liste S de points de départ.

Pour chaque pixel P dans la liste S

 Si le pixel P est déjà associé à une région, alors prendre le pixel P suivant

 Créer une nouvelle région [R]

 Calculer la valeur moyenne de [R]

 Créer la liste [N] des pixels voisins du pixel P

 Pour chaque pixel 'Pn' dans la liste [N]

 Si ('Pn' n'est pas associé à une région et R+Pn est homogène) alors

 Ajouter le pixel Pn dans la région R

 Ajouter les pixels voisins de 'Pn' dans [N]

 Recalculer la valeur moyenne de [R]

 Fsi

 Fin pour

 Fsi

Fin pour.

Les points de départ (seeds en anglais) :

Le choix des points de départ est l'étape la plus critique de l'algorithme, En effet, l'étape de croissance va utiliser une mesure de similarité pour choisir les pixels à agglomérer. Si le point de départ est situé dans une zone non homogène, la mesure de similarité va produire de fortes variations et la croissance va s'arrêter très tôt.

Par conséquent, il convient de choisir des points de départ dans des zones les plus homogènes possibles.

Pour trouver des zones, on peut réutiliser le principe de décomposition utilisé dans l'algorithme split/merge. Comme la décomposition finale nous donne une liste de blocs homogènes, il suffit de choisir le centre des plus gros blocs pour avoir de bons points de départ.

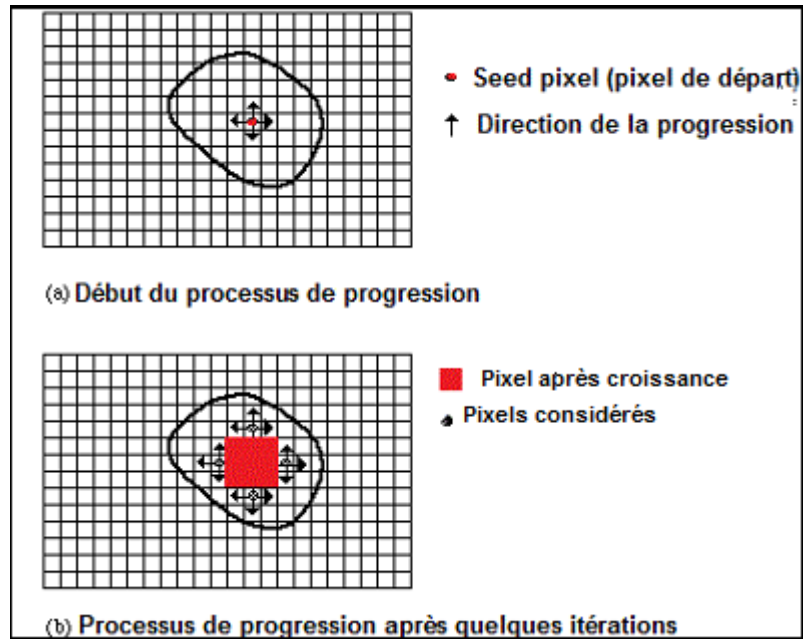


FIGURE III.4 – Processus de progression

B.3 Inconvénients de l'approche régions :

- Les méthodes de cette approches sont récursives donc ont des risques de débordements .
- Difficulté de définir les critères de fusions ou division des pixels(dans le cas Divion/Fusion).
- Influence de la position initiale des points de départs.(Dans le cas de Croissance de régions).
- Temps de calcul important.
- Tendance des algorithmes à trouver un nombre trop important de régions par rapport au nombre d'objets présents dans l'image.
- Méthodes sensibles au bruit.

C.Approche seuillage :

Le seuillage est une technique de segmentation très populaire à cause de sa facilité de mise en oeuvre et sa rapidité. Elle permet d'extraire les objets du fond de l'image. Dans le cas le plus classique, les pixels de l'image sont classés en deux classes par l'intermédiaire d'un niveau de gris S appelé seuil. La première classe regroupe les pixels du fond et la deuxième classe les pixels de l'objet. Cette opération s'appelle binarisation.

Soit l'image $I(M \times N)$, supposons que $f(x,y)$ représente le niveau de gris d'un pixel de coordonnées (x,y) , $0 \leq x \leq M$, $0 \leq y \leq N$

et S le seuil choisi arbitrairement ou calculé automatiquement par l'une des méthodes qu'on verra par la suite. Les pixels de l'objet sont ceux dont le niveau de gris est inférieur au seuil S et les pixels dont le niveau de gris est supérieur au seuil S appartiennent au fond de l'image. L'image segmentée G est définie pour chaque pixel de coordonnées (x,y) par :

$$g(x,y) = \begin{cases} 1 & \text{si } f(x,y) > S \\ 0 & \text{si } f(x,y) \leq S \end{cases}$$

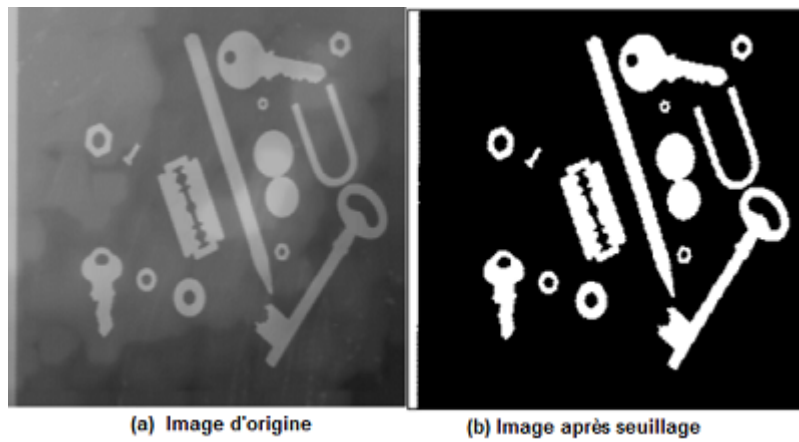


FIGURE III.5 – Seuillage d'une image.

Le seuillage peut être local ou global :

C.1 Seuillage local (Dynamique) :

Pour ce seuillage, la classification d'un pixel dépend non seulement de son niveau de gris mais aussi de ses informations locales c'est-à-dire des niveaux de gris de ses voisins. On définit alors pour chaque pixel un seuil selon sa position. Le premier à proposer une technique donnant de bons résultats fut Bernsen.[25]

Mathématiquement le seuil se calcule comme suit :

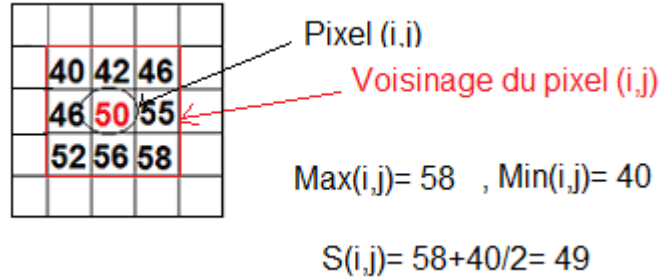
$$S(i,j) = (\max(i,j) + \min(i,j))/2$$

Avec :

- $S(i,j)$: seuil à appliquer pour le point i,j (pixel (i,j)) ;
- $\max(i,j)$: valeur du niveau de gris maximal dans une fenêtre centrée en (i,j) de taille $N \times M$

- $\min(i,j)$: valeur du niveau de gris minimal dans une fenêtre centrée en (i,j) de taille $N \times M$
- N et M appartenant à N .

Dans l'exemple qui suit, nous allons calculer le seuil local qui permettra de classer le pixel (i,j) :



Cependant, ce filtre est très sensible au bruit du fond.

C.2 Seuillage global :

Le principe du seuillage global est d'utiliser une valeur seuil à partir de laquelle on peut choisir à quelle classe le pixel appartient. Les méthodes utilisant ce type de seuillage reposent sur l'exploitation de l'histogramme de toute l'image qui caractérise la distribution des niveaux de gris. En général une méthode de seuillage consiste à déterminer la valeur optimale du seuil S en se basant sur certains critères.

Mathématiquement :

$$\forall i, j \in N \times M \quad I(i, j) = \begin{cases} 1 & \text{si } f(i, j) > S \\ 0 & \text{sinon.} \end{cases}$$

avec :

$N \times M$: Nombre de colonnes et de lignes de l'image

I : Image Binarisée

f : valeur fonction de l'image d'origine

S : Seuil de binarisation.

Cependant, l'approche seuillage pose un problème qui est le choix du seuil. On pourrait le choisir arbitrairement mais les résultats risquent de ne pas être optimaux. Pour de meilleurs résultats la grande majorité des techniques utilisent l'histogramme des niveaux de gris pour choisir le seuil à appliquer, ce qui est le cas de la méthode d'Otsu que nous allons présenter dans ce qui suit.

Exemple : Méthode d'Otsu :

La méthode d'Otsu est utilisée pour effectuer un seuillage automatique à partir de la forme de l'histogramme de l'image, ou la réduction d'une image à niveaux de gris en une image binaire. L'algorithme suppose alors que l'image à binariser ne contient que deux classes de pixels, (c'est-à-dire le premier plan et l'arrière-plan) puis calcule le seuil optimal qui sépare ces deux classes afin que leur variance intra-classe³ soit minimale.

Dans la méthode d'Otsu, le seuil qui minimise la variance intra-classe est recherché à partir de tous les seuillages possibles :

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t)$$

Où :

w_i : La probabilité d'être dans la ième classe.

σ_i^2 : Les variances de ces classes.

t : Le seuil qui varie de 1 à 255.

-La probabilité d'être dans la ième classe se calcule comme suit :

$$w_i = \frac{N_{ci}}{N_t}$$

Où N_{ci} est le nombre de pixel dans la classe i et N_t le nombre total des pixel de l'image.

-La variance de chaque classe se calcule comme suit :

$$\sigma_i^2 = \frac{1}{N_{ci}} \sum_{i=1}^n (x_i - m)^2$$

Où x_i est un élément de la classe i (un pixel) et m la moyenne des niveaux de gris dans la classe i .

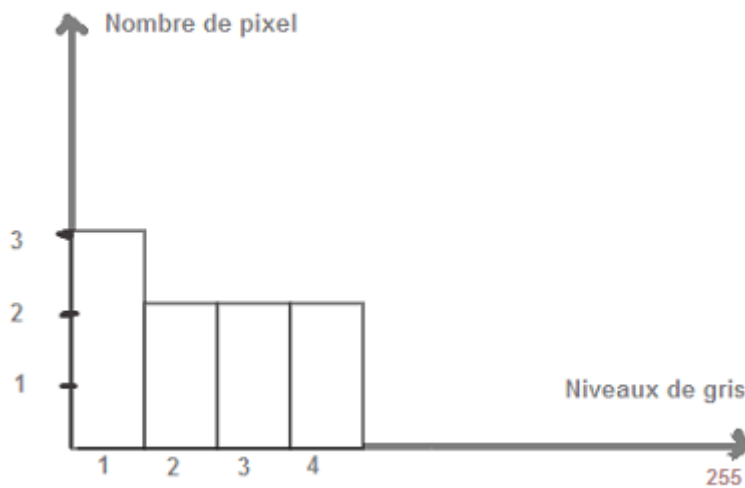
3. Variabilité au sein d'une même classe

> Prenons un exemple afin de mieux comprendre en quoi consiste cette méthode :

- Soit la matrice suivante représentant les niveaux de gris d'une image :

1	3	1
4	2	2
1	4	3

- Soit l'histogramme correspondant à la matrice :



- Calcul de la variance intra-classe pour le premier seuil :

Pour $t=1$:

$$\sigma_w^2(1) = w_1(1)\sigma_1^2(1) + w_2(1)\sigma_2^2(1)$$

- Calcul des probabilités d'être dans les classes 1 et 2 :

$$w_1(1) = \frac{N_{c1}}{N_{total}} = \frac{3}{9} = \frac{1}{3}$$

$$w_2(1) = \frac{N_{c2}}{N_{total}} = \frac{6}{9} = \frac{2}{3}$$

- calcul des moyennes dans les deux classes :

$$m_1 = \frac{1 \cdot 3}{3} = 1$$

$$m_2 = \frac{2 \cdot 2 + 3 \cdot 2 + 4 \cdot 2}{6} = 3$$

-Calcul de la variance pour chacune des classes :

$$\sigma_1^2(1) = \frac{1}{3}((1-1)^2 + (1-1)^2 + (1-1)^2) = 0$$

$$\sigma_2^2(1) = \frac{1}{6}((2-3)^2 + (2-3)^2 + (3-3)^2 + (3-3)^2 + (4-3)^2 + (4-3)^2) = \frac{4}{6} = \frac{2}{3}$$

-Calcul de la variance intra-classe :

$$\sigma_w^2(1) = \frac{1}{3} * 0 + \frac{2}{3} * \frac{2}{3} = \frac{4}{9}$$

- L'algorithme calcule de la même manière toutes les variances intra-classe pour chaque seuil t , avec $t \in [1..256]$ jusqu'à trouver le seuil optimal qui est celui qui minimise cette variance intra-classe.

C.4 Test de l'approche seuillage :

- Application d'un seuillage basique (le seuil est choisi arbitrairement) :



- Application d'un seuillage en utilisant la méthode d'Otsu :



III.3 Comparaison des trois approches de segmentation d'images :

Après avoir étudié les trois approches de segmentation d'images, et testé quelques unes, on a vu que :

- **Dans l'approche contours :** Comme le test l'a révélé, les résultats ne sont pas très satisfaisants. La main n'est détectée que partiellement.
- **Dans l'approche régions :** Les inconvénients sont nombreux, le plus contrariant étant la lenteur des algorithmes utilisés dû au nombre important d'itérations qu'effectuent ces derniers.
- **Dans l'approche seuillage :** Comme le test l'a révélé, les résultats du seuillage basique (avec un seuil choisi arbitrairement) ne sont pas très satisfaisants, tandis que ceux trouvés en utilisant le seuillage par la méthode d'Otsu sont intéressants.

Dans notre travail, nous allons alors utiliser l'approche seuillage pour segmenter nos images et pouvoir détecter la région d'intérêt qui est dans notre cas la main.

III.4 Intelligence artificielle :

L'intelligence artificielle, l'une des branches les plus récentes dans le domaine de l'informatique vise à programmer des machines capables d'effectuer des tâches qui requiert de l'intelligence.

En effet, les programmes de l'intelligence artificielle sont aujourd'hui capable de reconnaître des commandes vocales, d'analyser automatiquement des photos satellite, d'assister des experts pour prendre des décision dans des environnements complexes et évolutifs par exemple dans les diagnostics médicaux etc..

Pour réaliser ces tâches, les machines doivent être capables de s'adapter à toutes les situations et éventuellement d'évoluer en fonction de nouvelles contraintes. C'est pour cela qu'elles ont été dotées de la capacité d'apprentissage qui leurs permet de tirer partie de leurs expériences.

La plupart des programmes d'intelligence artificielle possèdent un module d'apprentissage et tous les programmes de reconnaissance des formes sont fondés sur des algorithmes d'apprentissage, c'est pour cela que nous allons nous intéresser dans notre travail à cet aspect particulier de l'intelligence artificielle.

III.5 Apprentissage artificiel / automatique (Machine Learning) :

L'apprentissage artificiel englobe toute méthode permettant de construire un modèle de la réalité à partir de données, soit en améliorant un modèle partiel ou moins général, soit en créant complètement le modèle, ainsi l'apprentissage artificiel a pour but l'élaboration de modèles, de règles et de connaissances. Ce qui va servir lors du processus de la prise de décision automatique.

L'apprentissage artificiel vise à développer des techniques pour que les programmes puissent s'entraîner à partir d'exemples et qu'ils soient capables de s'améliorer automatiquement au fil du temps. [26]

III.6 Types d'apprentissages :

Selon le critère pris en compte, on peut distinguer plusieurs types d'apprentissage automatique :

III.6.1 Selon la nature des données traitées :

On distingue deux types d'apprentissage ; l'apprentissage symbolique et l'apprentissage numérique.

A.Apprentissage symbolique :

L'apprentissage symbolique comme son nom l'indique, manipule des symboles. Il fonctionne grâce à la mise en place de réactions entre ces symboles par le biais de jugements.

L'idée est donc d'élaborer des méthodes permettant d'extraire des connaissances structurelles ou décisionnelles à partir d'instances peu structurées. La procédure d'apprentissage ainsi produite peut être écrite sous forme de règles sémantiquement significatives. Ceci représente l'avantage principale de l'apprentissage symbolique. Un expert qui analyse le système apprenant va pouvoir comprendre la façon dont fonctionne celui-ci.

B.Apprentissage numérique :

L'apprentissage numérique ne manipule pas de symboles, il traite uniquement des valeurs numériques quantitative qui vont être manipulées afin de réaliser l'apprentissage.

L'avantage de ces méthodes, c'est leurs grandes adaptabilité, d'ailleurs elles sont parfois nommées méthodes adaptatives. Contrairement aux méthodes symboliques elles produisent des procédures de classification en boîte noire, en plus, il est très difficile voir impossible de comprendre comment le système apprend.

Dans notre travail nous allons utiliser des données numériques ; les résultats obtenus lors de la phase de segmentation ; c'est pour cela que nous allons nous intéresser qu'à l'apprentissage numérique dans ce qui va suivre.

III.6.2 Selon l'intervention ou non d'un expert :

En apprentissage artificiel, le fait qu'intervienne (ou non) un expert afin d'aider l'apprentissage, en fonction des informations dont il dispose, définit le type d'apprentissage, d'où la distinction entre l'apprentissage supervisé et non supervisé.

A.Apprentissage non supervisé :

Dans l'apprentissage non supervisé, le système ne reçoit aucune information concernant les résultats attendus. Le système ou l'algorithme doit découvrir lui même la structure plus ou moins cachée des données et la corrélation existante entre ces dernières.

Le système doit ici, dans l'espace de description(l'ensemble des données), cibler les données selon leurs attributs disponibles, pour les classer en groupes homogènes d'exemples. La similarité est généralement

calculée selon une fonction de distance entre paires d'exemples. C'est ensuite à l'opérateur d'associer ou déduire du sens pour chaque groupe et pour les motifs (patterns en anglais) d'apparition de groupes.

B.Apprentissage supervisé :

L'apprentissage supervisé étudie comment construire automatiquement des connaissances manipulables par la machine à partir d'exemples fournis par l'homme.

Le terme supervisé vient du fait que l'ensemble des classes et la classe associée a chaque objet sont connu. Un expert est employé pour étiqueter correctement des exemples.

Ce type d'apprentissage consiste ainsi à inférer un modèle de prédiction à partir d'un ensemble d'apprentissage, c'est à dire plusieurs couples de la forme observation, étiquette, où chaque étiquette dépend de l'observation pour laquelle elle est choisie. Un algorithme d'apprentissage supervisé a pour but de déterminer une fonction s'approchant au mieux de la relation liant les observations et les étiquettes à partir de l'ensemble d'apprentissage uniquement. Cette fonction doit par ailleurs posséder de bonnes propriétés de généralisation et ainsi être capable d'associer une étiquette adéquate à une observation qui n'est pas dans l'ensemble d'apprentissage .

Dans notre travail nous allons utiliser l'apprentissage supervisé, c'est pour cela que nous allons le détailler dans ce qui suit.

III.7 Scénario de l'apprentissage supervisé :

Le système apprenant reçoit des données de l'environnement dans lequel il est placé. Nous ne nous interrogeront pas ici sur ce qui permet de définir et de segmenter ces données.

Dans le cas de l'apprentissage supervisé, chacune de ces données prend la forme d'un couple dans lequel on distingue d'une part la description d'une situation ou encore d'une observation, et d'autre part une réponse, que l'on appelle aussi « sortie désirée », qui est supposée être fournie par un oracle ⁴.

Une donnée Z_i est ainsi définie par un couple (observation, sortie désirée) que nous noterons (x_i, u_i) .

Dans le scénario de l'apprentissage supervisé, la tâche de l'apprenant est d'essayer d'approximer au mieux la sortie désirée u_i pour chaque entrée observée x_i .

Dans le cas idéal, l'apprenant devient capable après un certain temps d'apprentissage, de prédire exactement pour chaque entrée x_i , la sortie désirée u_i .

Nous supposons que l'environnement, qu'il soit mesuré par les senseurs d'un robot ou qu'il s'exprime

4. Un expert dans un domaine qui va s'occuper d'étiqueter les exemples qui vont être appris par la machine.

sous la forme d'une base de données, fournit un ensemble de données x_i défini sur l'espace d'entrées x et tirés aléatoirement suivant une distribution de probabilité noté D_x ; on parle de tirage indépendant et identiquement distribué ou tirage (IID).

On peut ainsi imaginer qu'une camera prend des images à intervalles réguliers d'une main , et que les formes x_i mesurées correspondent à l'alphabet de la langue des signes.

Dans le cas d'un apprentissage supervisé nous supposons également qu'un oracle étiquette les formes x_i grâce à une fonction inconnue de l'apprenant, que nous appelons fonction cible noté f .

l'apprenant reçoit donc un ensemble d'exemples :

$$S = \{x_i, u_i\}_{1 \leq i \leq m} = \{x_i, f(x_i)\}_{1 \leq i \leq m}$$

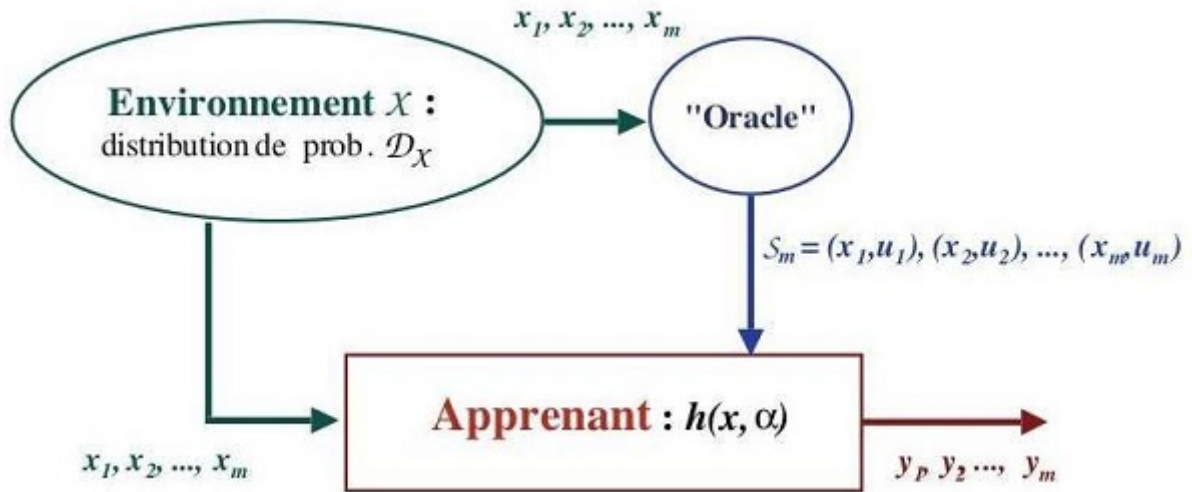


FIGURE III.6 – Le scénario classique de l'apprentissage supervisé : L'environnement fournit des données x_i tirées aléatoirement suivant une distribution D_X sur l'espace d'entrées X . Ces données sont étiquetées par un oracle qui utilise pour ce faire une fonction f . L'apprenant reçoit un échantillon d'exemples ou couples $(x_i, u_i) = (x_i, f(x_i))$, et à partir de cet échantillon, doit chercher à deviner f , ou au moins à en trouver une approximation h .

Remarque :

On distingue en général trois types de problèmes auxquels l'apprentissage supervisé est appliqué : la classification supervisée, la régression et les séries temporelles. Ces trois types de problèmes se différencient en fonction du type d'étiquettes fournit par l'oracle. Dans le cadre de notre travail, nous ne nous intéresserons qu'à la classification supervisée.

III.8 Classification supervisée :

III.8.1 Définition de la classification :

Une classification consiste à identifier les classes auxquelles appartiennent des objets à partir de traits descriptifs (attributs, caractéristiques).

Intuitivement une règle de classification est un acte cognitif ou une procédure permettant d'affecter à un objet la famille à laquelle il appartient autrement dit de le reconnaître.

III.8.2 Classification supervisée :

Appelé aussi classement ou classification inductive, a pour objectif d'apprendre par l'exemple. Elle consiste à inférer à partir d'un échantillon de données classées, une procédure de classification c.à.d. les classes sont connues et l'on dispose d'exemples de chaque classe (On dispose d'éléments déjà classés).

Donc la classification supervisée est un modèle inductif où l'apprenant considère un ensemble d'exemples, et infère l'appartenance d'un objet à une classe en considérant les similarités entre l'objet et les éléments de la classe étiquetée préalablement.

III.8.3 Classifieur :

Résoudre le problème de classification, c'est donc trouver une application de l'ensemble des objets à classer dans l'ensemble des classes. L'algorithme ou la procédure qui réalise cette application, est appelée classifieur. Un classifieur est donc un outil automatique de classification.

III.8.4 Processus de la classification automatique :

Le processus de classification automatique peut être décomposé en 4 principales étapes :

1. Faire classer un échantillon d'individus par un expert. Cet échantillon est désigné par le nom de base d'apprentissage.
2. Utiliser l'algorithme appelé classifieur qui parvient à reproduire la classification de l'échantillon d'apprentissage.
3. Évaluer la qualité du classifieur en l'appliquant à un ensemble d'individus classés par l'expert mais qui n'ont pas été utilisés au cours de la phase d'apprentissage.
4. Si le test est satisfaisant, appliquer la méthode de la 2ème étape à l'ensemble des objets à classer.

III.8.5 Formalisation de la classification automatique :

Ω : est la population.

D : est l'ensemble des descriptions.

L'ensemble des classes est $1, \dots, c$.

$X : \Omega \rightarrow D$ est la fonction qui associe une description à tout élément de la population.

$Y : \Omega \rightarrow 1, \dots, c$ est la fonction de classement qui associe une classe à tout élément de la population.

$C : D \rightarrow 1, \dots, c$ est une fonction qui sera appelée fonction de classement ou procédure de classification.

Le but de l'apprentissage est de rechercher une procédure de classification C , $C(X)=Y$.

De manière plus réaliste, telle que $C(X)$ soit une bonne approximation de Y .

III.8.6 Types de la classification supervisé :

Selon le nombre de classes et d'étiquettes on peut distinguer 3 types de classification automatique :

A. Classification binaire :

La classification binaire est le problème de classer des éléments dans deux catégories selon une loi de classification, ici le nombre total de classes est de deux.

B. Classification multi-classes :

Le problème multi-classe revient à classer une instance dans l'une des nombreuses classes. Dans ce cas l'ensemble de classes est de $1..n$.

C. Classification multi-label :

Dans la classification multi-label chaque instance peut appartenir à une ou plusieurs classes simultanément.

Remarque :

Alors que certains classifieurs permettent naturellement la classification multi-classe, d'autres sont de nature des classifieurs binaires, cependant, il existe plusieurs stratégies pour transformer un classifieur binaire en un classifieur multi-classes.

Une deuxième distinction est à faire entre les méthodes estimant des probabilités pour en déduire la

fonction de classement et les méthodes estimant directement la fonction de classement sans passer, du moins explicitement, par les probabilités.

D.Méthodes de classification probabiliste :

Dans l'approche probabiliste la fonction de classement se base sur une fonction de probabilité, fonction de prédiction suivant la règle de maximum de vraisemblance ⁵, pour une observation d , on choisit la classe pour laquelle cette observation est la plus probable. Cette vraisemblance associée à d la classe k telle que $P(d/k)$ soit max.

Parmi les approches probabilistes, les plus utilisées se distinguent suivant la nature du modèle sous-jacent :

- La méthode de Bayes naïve.
- Processus gaussiens, relevance vector machines ou Bayes point machines. Le formalisme des modèles graphiques a été développé pour représenter ce type de relation. En particulier si l'on s'intéresse à la prise en compte des contraintes temporelles, les différents modèles graphiques employés sont :
 - Les réseaux bayésiens dynamiques (RBD).
 - Modèles de Markov caché (MMC).
 - Les champs aléatoires conditionnels (CAC ou CRF en anglais).

Le problème typique partiellement résolu grâce à l'utilisation des modèles de Markov est celui du traitement de la parole.

E.Méthodes de classification directe :

Une manière de présenter les méthodes de classification supervisée dites « directes » consiste à les classer suivant la nature de leur ensemble d'hypothèses. Conceptuellement, cet ensemble d'hypothèses peut être vu comme l'ensemble de combinaisons linéaires de fonctions de référence. On distingue trois catégories selon la nature des fonctions de références.

- Soit elles sont fixées indépendamment de l'échantillon en prenant une base (d'ondelettes par exemple).
- Soit elles ne dépendent que des observations x_i et pas des étiquettes. C'est le cas des méthodes à base de noyaux et notamment des séparateurs à vaste marge (SVM), des k plus proches voisins, des arbres de décisions, des méthodes additives et celles d'agrégation comme le boosting.
- Enfin les fonctions de références peuvent dépendre à la fois des observations et des étiquettes comme pour les réseaux de neurones de type « perceptron multicouche » ou « fonctions de base radiales ».

5. est une méthode statistique courante utilisée pour inférer les paramètres de la distribution de probabilité d'un échantillon donné \mathcal{D} . Si j'observe d , je choisie la classe pour laquelle cette observation est la plus probable. Cette vraisemblance associée à d la classe k telle que $P(d/k)$ soit max.

Les méthodes du deuxième type conduisent à des algorithmes d'apprentissage rapides et souvent bien posés alors que les réseaux de neurones sont connus pour être associés à des algorithmes d'apprentissage au comportement chaotique. En revanche ils donnent souvent des modèles plus concis que les algorithmes de la deuxième catégorie. Une des problématiques de recherche dans le domaine de l'apprentissage est donc de trouver des modèles parcimonieux facilement identifiables. C'est le cas des SVM, ce qui explique en partie leur succès actuel. Une autre explication est leur efficacité dans le traitement de problèmes concrets comme celui de la reconnaissance de caractères manuscrits. Sur ce problème, les SVM donnent de très bons résultats, et surtout offrent un compromis performance/facilité de mise en oeuvre très intéressant.

III.8.7 Problèmes liés à l'utilisation des méthodes probabilistes :

Une première critique peut être adressée aux méthodes probabilistes. S'il est vrai qu'elles permettent de prendre en compte les connaissances à priori dont on dispose sur la nature des dépendances entre les variables considérées, ce que l'on cherche à apprendre c'est une fonction de décision et non pas une loi de probabilité.

La seconde critique est liée au problème de variabilité intra et inter utilisateur. Si pour un utilisateur donné, dans une condition d'utilisation donnée, ce type d'approche est très efficace, les performances se dégradent notablement, comme dans le cas de la reconnaissance de la parole, lorsque l'on modifie les conditions d'utilisation et que l'on change de sujet. En effet, ils font l'hypothèse que la distribution de probabilité du signal sous-jacent est stationnaire, ce qui n'est pas le cas lorsque l'on considère différents utilisateurs. Il y a là une autre question difficile posée au domaine de l'apprentissage : Comment apprendre sur certains individus (avec une loi de probabilité donnée) et généraliser sur d'autres (la loi de probabilité ayant changé).

C'est pour cela que nous avons opté pour l'utilisation des méthodes de classification directe dans notre travail.

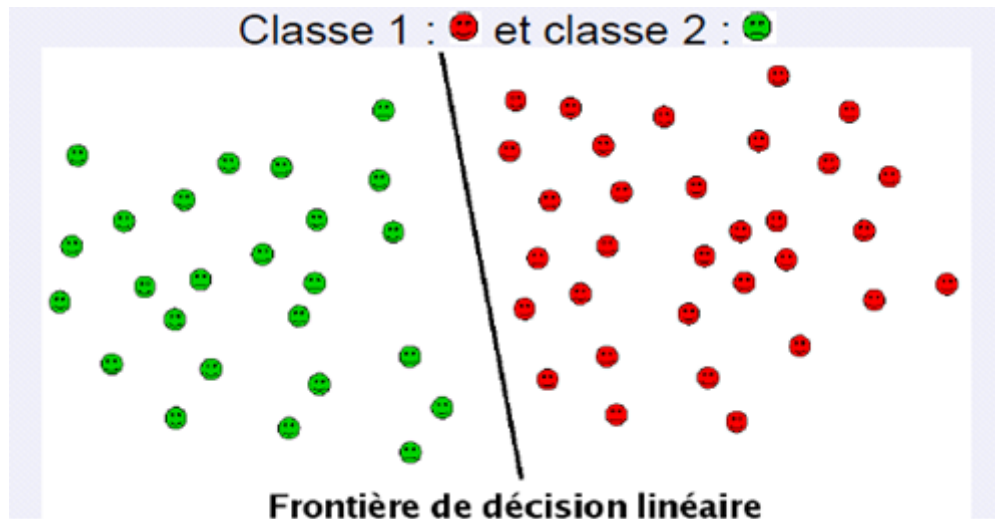
III.9 Algorithmes de classification directe :

III.9.1 Séparateur à vaste marge (SVM) : [27]

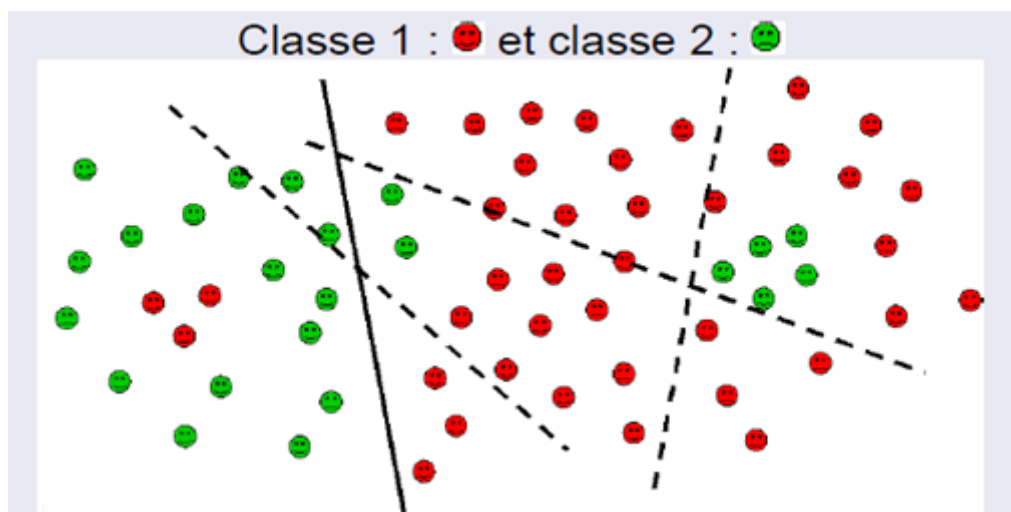
A. Le concept de linéarité :

Les données peuvent être :

- Séparables linéairement : Si tous les points associés aux données peuvent être séparés correctement par une frontière linéaire (hyperplan).



- Non séparables linéairement : Si tous les points associés aux données ne peuvent être séparés correctement par une frontière linéaire.



B. Le principe des SVM :

Les SVM ont été créés pour résoudre les problèmes de classification. Le but est de séparer l'espace des données en classes. A chaque réalisation des variables de base est associée une classe.

Le principe de l'algorithme est de créer un hyperplan qui sépare les données en deux classes (classification binaire),

En utilisant le principe de marge maximale : considérons des points d'apprentissage appartenant aux classes -1 et $+1$, l'hyperplan maximisant la marge sépare les deux classes des points d'apprentissage de telle façon que la distance entre les points les plus proches de la marge et l'hyperplan soit maximisée.

La première version de l'algorithme a été proposée par Vladimir Vapnik en 1963. Elle s'appliquait aux

séparateurs linéaires (on parle de classes linéairement séparables).

Puis, en 1992, Bernhard Boser, Isabelle Guyon et Vladimir Vapnik proposent la création d'un séparateur non linéaire en appliquant le Kernel trick (méthode à noyaux) au principe de la marge maximale.

L'algorithme de base n'est pas modifié, seul le produit scalaire classique est remplacé par une fonction noyau non linéaire. L'algorithme de marge maximale est alors utilisé dans un espace transformé appelé espace support (feature space⁶). La transformation peut être non linéaire et l'espace support de très grande dimension : cela permet d'obtenir un séparateur qui est un hyperplan dans l'espace support, mais une fonction non linéaire dans l'espace de départ.

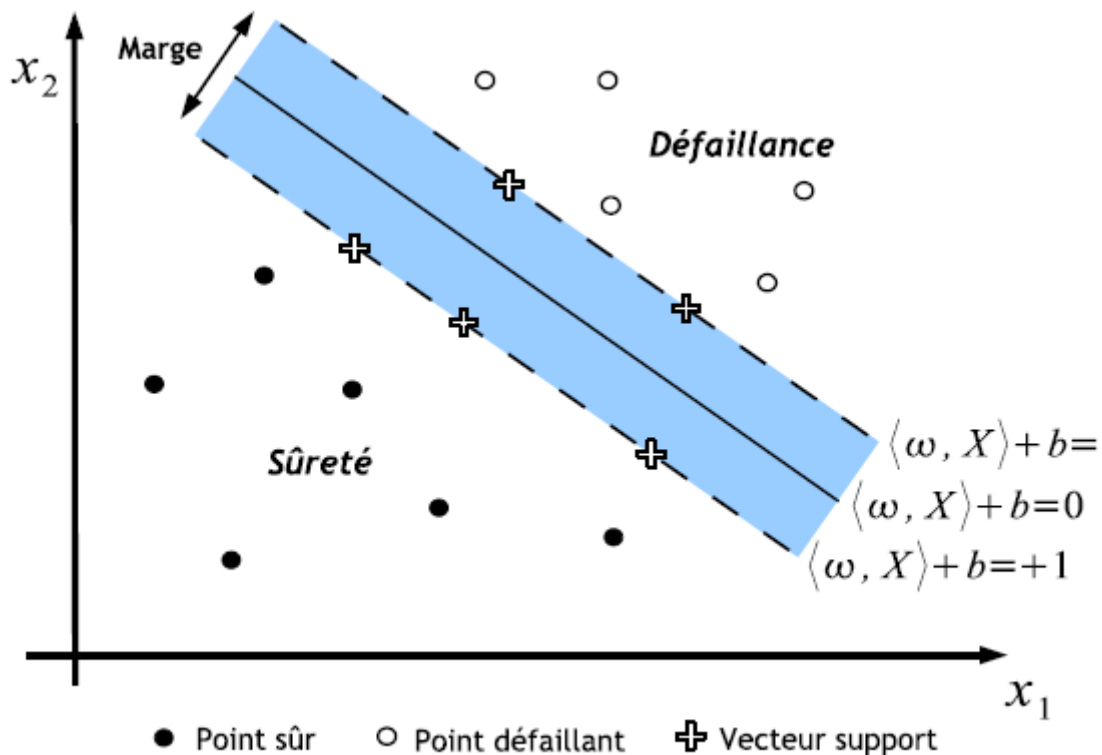


FIGURE III.7 – Le principe de SVM.

C.Le problème de classification :

C.1 Dans le cas de classe linéairement séparable :

Prenons tout d'abord le cas de classes linéairement séparables. Nous considérons deux classes +1 et -1. Soit X_1, \dots, X_n un échantillon de réalisations des variables de conception. Soit $C_i \in \{-1, 1\}$, $i = 1, \dots, n$ les classes associées à chacune des réalisations. Nous cherchons un hyperplan qui sépare les deux classes

6. Feature vector dans le domaine de la reconnaissance des formes et de l'apprentissage artificiel, est un vecteur à n dimension d'attributs qui représentent un objet quelconque. En d'autres termes c'est une représentation numérique des objets que les algorithmes d'apprentissage utilisent. L'espace vectoriel associé à ces vecteurs est appelé feature space.

de la forme :

$$G(X) = (\omega, X) + b$$

Où ω est un vecteur de paramètres, vecteur définissant la normale à l'hyperplan ; et b est appelé le seuil. La fonction de classification s'écrit alors :

$$C(X) = \text{signe}(G(X)) = \text{signe}((\omega, X) + b)$$

On demande à cet hyperplan de maximiser la distance entre l'hyperplan et les deux classes, ce qui constitue une condition d'optimalité de position de l'hyperplan. Le problème de classification se résume à déterminer le vecteur de paramètres ω et le seuil b , en respectant les conditions d'optimalité de position de l'hyperplan.

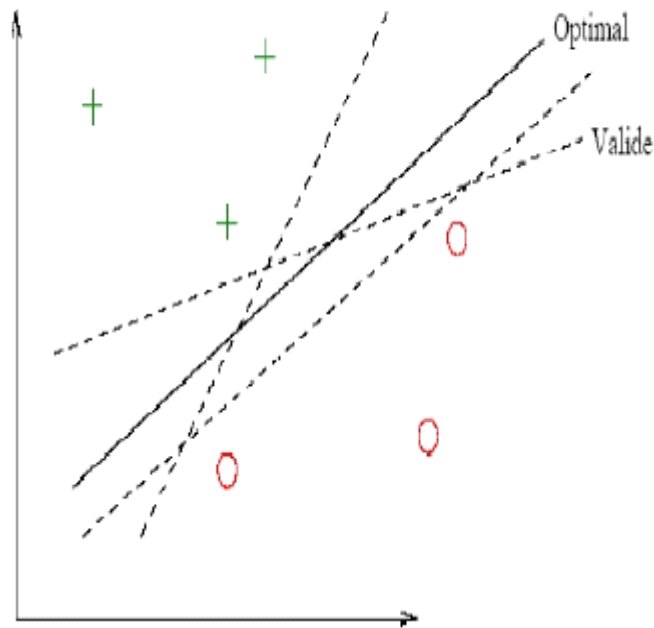


FIGURE III.8 – Les hyperplans possibles pour séparer les données.

Le problème d'optimisation :

Le problème d'optimisation à résoudre pour trouver les paramètres de l'hyperplan est alors le suivant :

$$\max_{\omega, b}, \min_i \{ \|X - X_i\| | (\omega, X) + b = 0, i = 1..n \}$$

Où :

$$\min_i \{X - X_i | \omega, X + b = 0, i = 1..n\}$$

Représente la distance minimale entre l'hyperplan et les points d'apprentissage.

L'écriture d'une classe par un signe (+1 ou -1) facilite l'écriture de l'optimisation.

γ_i désigne la marge de chaque réalisation, marge qui s'écrit :

$$\gamma_i = ci(\omega, X_i + b), i = 1, \dots, n$$

Le signe positif de γ_i montre une bonne classification. En effet, si le produit de C_i par $\omega, X_i + b$ est positif, cela signifie que le signe de C_i et le signe de l'évaluation du séparateur au point X_i sont identiques, et que le point X_i est classé du bon côté du séparateur.

Il y a une infinité de vecteurs ω qui répondent au problème à un facteur multiplicateur près. Il s'agit alors d'imposer une condition supplémentaire pour obtenir une solution unique. La condition suivante est imposée : la distance minimale entre l'hyperplan et la réalisation la plus proche est fixée, soit :

$$\min_{i=1..n} |(\omega, X_i) + b| = 1$$

Cela implique que la marge soit égale à deux fois la distance minimale, soit :

$$\gamma = 2 \frac{|(\omega, X_i) + b|}{\|\omega\|} = \frac{2}{\|\omega\|}$$

Maximiser la marge correspond donc à minimiser la norme de ω , ce qui conduit au problème d'optimisation suivant : $\min \frac{\|\omega\|^2}{2}$ sous la contrainte : $ci((\omega, X_i) + b) = 1, i = 1..n$

Où les n contraintes imposent la bonne classification des points d'apprentissage par rapport au séparateur recherché.

La solution de l'optimisation va donner un nombre S de points. Les points les plus proches de l'hyperplan optimal sont appelés les vecteurs supports.

Pour définir la frontière entre les domaines il suffit de s'appuyer sur les points proches de cette frontière. Cette propriété est essentielle pour la suite : pour affiner la séparation, seules les réalisations dans la marge sont utiles, les autres peuvent être classées sans calcul supplémentaire à partir de la connaissance du séparateur SVM.

C.2 Dans le cas de classe non séparable linéairement :(méthode à noyaux)

Le principe est le suivant : en utilisant un projecteur non linéaire ϕ , une transformation de l'espace de départ est réalisée. Cette transformation effectue une projection de l'espace de départ dans l'espace dit support, espace qui présente la particularité d'avoir une dimension supérieure à l'espace de départ. Le Kernel Trick ou L'espace support permet de retrouver le cas précédent des classes linéairement séparables, la formulation du séparateur reste la même, mais le produit scalaire (fonction noyau linéaire) est remplacé par une fonction noyau K plus générale.

Cela se traduit dans les équations de la manière suivante : en remplaçant le produit scalaire par une fonction noyau, on obtient l'expression suivante :

$$c(X) = \text{sgn}(G(X)) = \text{signe}(\sum_{i=1}^n \alpha_i c_i (K(X_i, X) + b))$$

$$\text{Où : } K(X_1, X_2) = (\Phi(X_1), \phi(X_2))$$

$\Phi()$ est la fonction de projection. Cette fonction n'a pas besoin d'être explicitée, nous utilisons dans la pratique uniquement une fonction noyau K . La construction d'une fonction noyau permet de travailler dans un espace de Hilbert⁷ de dimension infinie. Cette dimension permet d'obtenir une classification correcte en utilisant toujours un hyperplan.

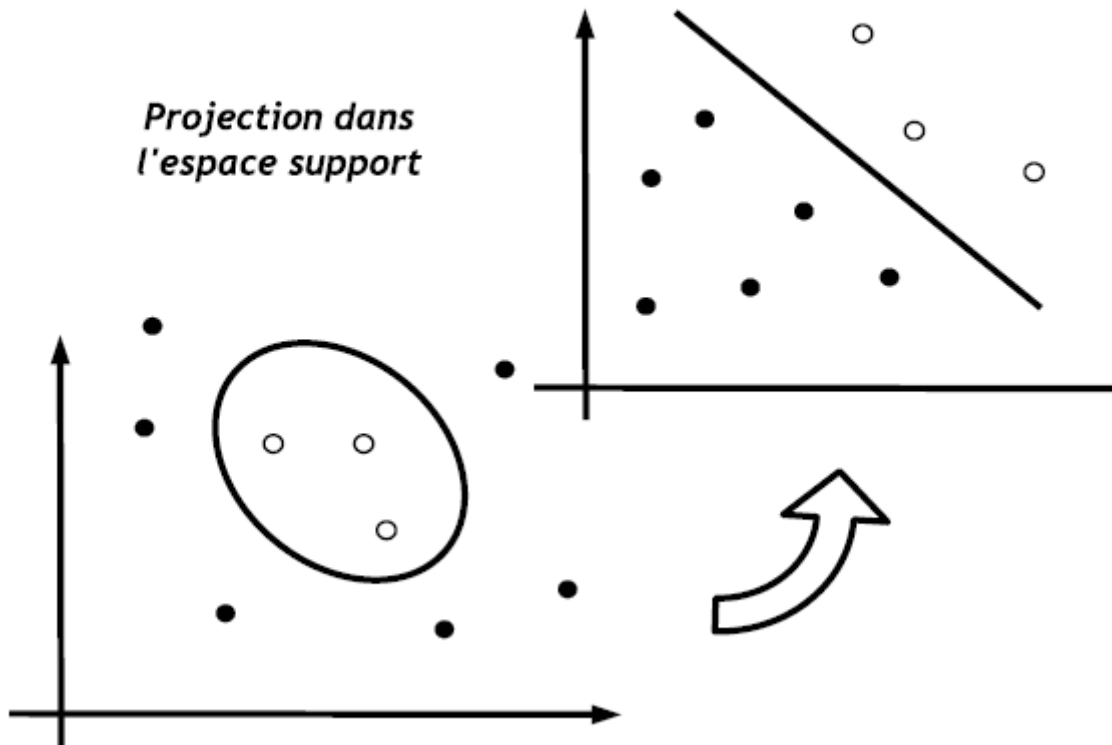


FIGURE III.9 – Projection de données non linéairement séparables dans l'espace de Hilbert.

Quelques fonctions noyau couramment utilisées :

7. Le concept mathématique d'espace de Hilbert, nommé d'après David Hilbert, généralise la notion d'espace euclidien. Il étend les méthodes de l'algèbre linéaire et de l'analyse des espaces euclidiens classiques (plan, de dimension deux, et espace à trois dimensions) à des espaces de dimension quelconque.

Radial Basic Function(RBF) : $K = \exp(-\frac{1}{2\sigma^2}\|X_1 - X_2\|^2)$

Polynomiale : $K = ((X_1, X_2) + b)^d$

Fourrier : $K = \frac{(m+\frac{1}{2})(X_1-X_2)}{2\sin\frac{X_1-X_2}{2}}$

Sigmoidale : $K = \tanh(-k(X_1 - X_2) + \theta)$

Selon la fonction choisie, un ou plusieurs paramètres sont à ajuster par rapport au problème étudié.

D.Récapitulatif :

L'algorithme se base principalement sur 3 astuces pour obtenir de bonnes performances tant en qualité de prédiction qu'en complexité de calcul.

- On cherche l'hyperplan comme solution d'un problème d'optimisation sous-conainte. La fonction à optimiser intègre un terme de qualité de prédiction et un terme de complexité du modèle.
- Le passage à la recherche de surfaces séparatrices non linéaires est introduit en utilisant un noyau kernel qui code une transformation non linéaire des données.
- Numériquement, toutes les équations s'obtiennent en fonction de certains produits scalaires utilisant le noyau et certains points de la base de données (ce sont les Support Vectors).

III.9.2 Les arbres de décisions :

A.Le principe des arbres de décision

Un arbre de décision permet de représenter les objets étudiés sous une forme arborescente, en modélisant une hiérarchie de test sur les valeurs d'un ensemble de variables appelé attributs. Ces méthodes sont populaires pour la présentation synthétique des données qu'elles fournissent, ainsi que pour la clarté des explications concernant la décision rendue.

Comme toute méthode d'apprentissage supervisé, les arbres de décision utilisent des exemples. Si l'on doit classer des éléments dans des catégories, il faut construire un arbre de décision par catégories. Pour déterminer à quelle catégorie appartient un nouvel élément, on utilise l'arbre de décision de chaque catégorie auquel on soumet l'élément à classer. Chaque arbre répond oui ou non (il prend une décision) concrètement, chaque nœud d'un arbre de décision contient un test (un IF THEN) et les feuilles ont les valeurs oui ou non. Chaque test regarde la valeur d'un attribut de chaque exemple. Par exemples, l'arbre de la décision ci-dessous, décide une réponse booléenne (classification dans l'ensemble {oui, non}), en fonction des valeurs discrètes des attributs {difficile, durée, motivation, surprenant }.

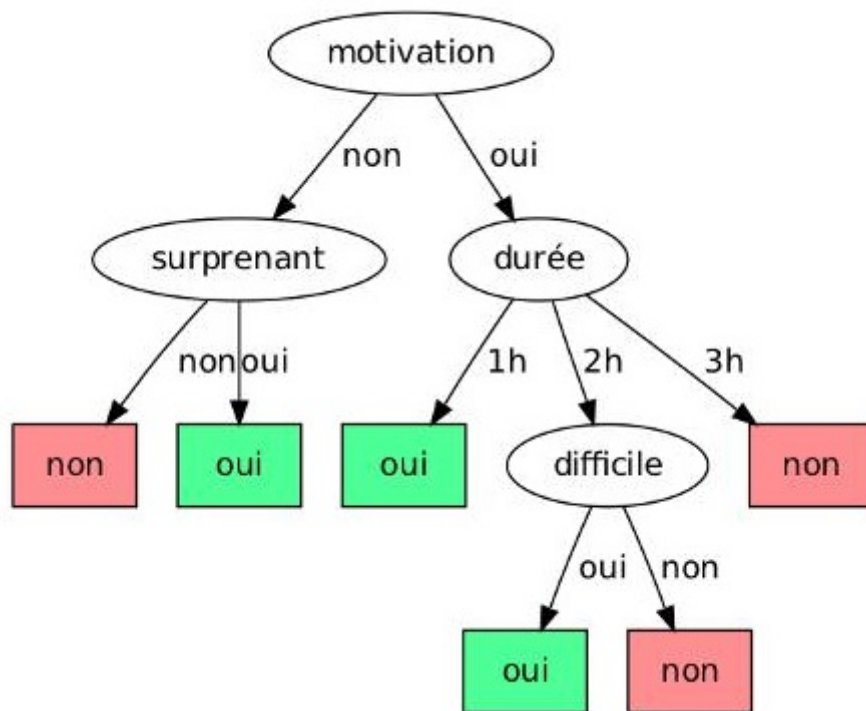


FIGURE III.10 – Exemple d'arbre de décision pour la question : Ce thème est-il intéressant ?

Un ensemble de valeurs pour les différents attributs est appelé une « instance », que l'on note généralement (x, y) où y est la valeur de l'attribut que l'on souhaite prédire (dans le cas de la classification l'attribut que l'on souhaite prédire est une classe) et $x = x_1, \dots, x_m$ désignent les valeurs des m autres attributs. L'apprentissage d'un arbre de décision se fait sur un ensemble d'instances $T = \{(x, y)\}$ appelé « ensemble d'entraînement ».

date	motivation	durée (h)	difficile	surprenant	température (K)
11/03	oui	1	non	oui	271
14/04	oui	4	oui	oui	293
03/01	non	3	oui	non	297
25/12	oui	2	oui	non	2911
⋮	⋮	⋮	⋮	⋮	⋮

FIGURE III.11 – Exemple d'ensemble d'entraînement pour l'arbre ci-dessus

B.Construction d'un arbre :

La génération d'un arbre de décision se fait en deux phases :

B.1.construction de l'arbre : au début tous les tuples (toutes les instances d'apprentissage) se trouvent à la racine de l'arbre. Partitionner les tuples récursivement en se basant à chaque fois sur un attribut sélectionné qui sépare mieux les instances (la sélection des attributs est basée sur une heuristique ou une mesure statistique).

B.2.Simplification de l'arbre (élagage) : Identifier et supprimer les branches qui correspondent à des exceptions ou de bruits.

Construire un arbre le plus petit possible.

C.Utilisation :

Utilisation :Tester les attributs du tuple par rapport à l'arbre pour trouver la branche qu'il satisfait et donc sa classe.

D.Formalisation :

D.1.Construction de l'arbre :

À partir d'un ensemble d'observations $T = \{(x, y)\}$, on souhaite construire un arbre de décision prédisant l'attribut y en fonction de nouvelles instances x . Pour ce faire, les arbres de décision suivent le paradigme diviser-pour-régner, que l'on peut schématiser (dans le cas d'attributs à valeurs discrètes) par le pseudo-code suivant :

```
1:  ArbreDecision(T)
2:      si "condition d'arret"
3:          retourner feuille(T)
4:      sinon
5:          choisir le "meilleur" attribut i entre 1 et m
6:          pour chaque valeur v de l'attribut i
7:               $T[v] = \{(x, y) \text{ de } T \text{ tels que } x_i = v\}$ 
8:               $t[v] = \text{ArbreDecision}(T[v])$ 
9:          fin pour
10:         retourner noeud(i, {v -> t[v]})
11:     fin si
```

Où $\text{noeud}(i, \{v \rightarrow tv\})$ désigne le constructeur d'un noeud qui teste l'attribut i et possède un descendant tv pour chaque valeur possible. Les parties entre guillemets correspondent à des choix heuristiques propres à chaque algorithme

Condition d'arrêt :

Elle influe sur la profondeur et la précision du prédicteur produit. Par exemple, la condition $|T| = 1$ produira des arbres très précis sur l'ensemble d'entraînement (i.e. pour une instance $(x, y) \in T$, la prédiction pour x sera exactement y) mais également très profonds, donc longs à calculer, et qui risquent de surapprendre les données d'entraînement.

Meilleur attribut :

Il s'agit d'évaluer localement quel attribut apporte « le plus d'information » (ou encore « est le plus corrélé ») au résultat à prédire.

D.2. l'Utilisation de l'arbre pour classifier de nouvelles instances :

Le raisonnement pour classifier une instance est le suivant :

```

1: Classifier(x, t)
2:   si t = feuille(Tf)
3:     retourner la classe majoritaire de Tf
4:   sinon si t = noeud(i, v, t_left, t_right)
5:     si x[i] <= v
6:       retourner Classifier(x, t_left)
7:     sinon, x[i] > v
8:       retourner Classifier(x, t_right)
9:   sinon, t = noeud(i, {v -> t[v]})
10:  retourner Classifier(x, t[x[i]])
11: fin si

```

Si toutes ou la grande majorité des instances de T_f ont la même classe $c \in \{1, \dots, C\}$, c apparaît comme la meilleure prédiction possible, et la fraction des instances de classe c dans T_f est un indicateur de la « sûreté » de cette prédiction. Dans le cas contraire, prédire la classe la plus fréquente reste une option, mais la sûreté de la prédiction n'en sera que moins bonne.

E. Récapitulatif :

L'arbre de décision est un arbre associé à DX tel que :

- Chaque noeud interne est étiqueté par un attribut (test sur un attribut) ;
- Chaque arête est étiquetée par un prédicat pouvant s'appliquer à l'attribut du noeud parent (résultat d'un test (valeur de l'attribut)) ;
- Chaque feuille est étiquetée par une classe C_j ;
- Un noeud est terminal s'il n'y a plus d'attribut disponible.

III.9.3 Les forêts d'arbres décisionnels (Random decision forest) :

[28] L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents.

Les arbres de décision peuvent se montrer instables de part la taille de l'échantillon. Le grand principe des forêts aléatoires est alors de chercher à tirer profit de cette instabilité en agrégeant les arbres de décision entre eux et stabiliser le tout au travers d'une méthode de vote majoritaire.

La méthode d'approche est différente et plus coûteuse que celle des arbres décisionnels. Elle consiste en l'utilisation du hasard pour améliorer les performances d'algorithmes de faibles qualités de classification.

Une part d'aléatoire est ainsi ajouté au cours de la construction d'un arbre. De cette façon on construit plusieurs arbres décisionnels aléatoires qui vont être agrégés ensemble pour former une forêt.

Les forêts aléatoires se construisent en concevant un arbre sur un sous-échantillon tiré aléatoirement. A chaque noeud de l'arbre il faut tirer uniformément un nombre de variables et chercher la meilleure coupure uniquement pour ces variables sélectionnées au sein de l'arbre décisionnel en cours de conception pour la forêt.

L'objectif de cette approche est de rendre les arbres construits plus indépendants entre eux ce qui permet de meilleures performances lors de l'agrégation en forêt. Elle a l'avantage d'être très fructueuse en grande dimension et d'être simple à mettre en oeuvre.

III.10 Evaluation des performances d'un classifieur :

L'évaluation des performances d'un classifieur est une tâche d'une grande importance, et les critères d'évaluation de ces dernières diffèrent selon le domaine auquel appartiennent les données utilisées par le classifieur.

En effet, selon le domaine ou la classification est faite, les conséquences des erreurs de classification seront différentes. Certaines erreurs ont un coût plus important que d'autres, par exemple dans les diagnostics médicaux, un mauvais diagnostic ou traitement peut avoir des coûts différents selon le type d'erreurs commises.[28]

Dans notre travail nous nous contenteront d'évoquer les méthodes qui permettent d'évaluer les performances d'un classifieur où les erreurs de classification n'auront pas des résultats catastrophique ou mortel.

III.10.1 Taux de bonne classification sans coûts :

L'objectif d'une méthode de classification est de prédire correctement la classe d'un nouvel objet c'est-à-dire d'objets non utilisés lors de la phase d'apprentissage.

Pour évaluer la précision d'un classifieur, plusieurs méthodes existent. Souvent, on utilise un échantillon de test indépendant c'est-à-dire qui contient des objets distincts de ceux de l'échantillon d'apprentissage. La base de données d'objets disponible est donc séparée en deux échantillons disjoints. On calcule le taux d'erreur par le calcul du nombre d'objets classés erronément dans l'échantillon test par application du modèle induit à partir de l'échantillon d'apprentissage.

Quelques outils utilisés :

A. Taux de classification : (Accuracy score)

Le taux de classification sans coût est l'indicateur le plus simple pour évaluer les performances d'un système de classification, on calcule le pourcentage des éléments correctement classés dans un échantillon de test.

$tcc = \text{nbr d'instance correctement classé} / \text{nombre de classes total}$

B. taux d'erreurs :

$te = 1 - tcc$

avec :

te : taux d'erreurs

tcc : taux de classification correcte

C. Matrice de confusion :

La matrice de confusion est un outil servant à mesurer la qualité d'un système de classification. Chaque colonne de la matrice représente le nombre d'occurrences d'une classe estimée, tandis que chaque ligne représente le nombre d'occurrences d'une classe réelle (ou de référence). Les données utilisées pour chacun de ces groupes doivent être différentes.

Un des intérêts de la matrice de confusion est qu'elle montre rapidement si le système parvient à classer correctement. En effet un système de classification sera d'autant meilleur que sa matrice de confusion s'approchera d'une matrice diagonale.

L'évaluation quantitative: la matrice de confusion

		Classes tests		
Classes affectées	classe	1	2	3
	1			
	2			
	3			

FIGURE III.12 – Exemple de matrice de confusion

III.11 Conclusion :

Dans ce chapitre, nous avons détaillé les approches les plus utilisées par la segmentation d'images qui est une méthode de détection des zones d'intérêt. Puis, nous avons effectué une étude comparative de ces différentes approches et nous avons opté pour la méthode d'Otsu.

Par la suite, nous avons fait une étude générale de l'apprentissage artificiel et énoncé les raisons qui nous ont poussé à choisir les méthodes directes (SVM, Forêts aléatoires) au lieu des méthodes probabilistes (MMC) pour effectuer la reconnaissance.

Une fois le choix fait, nous avons détaillé les algorithmes de ces méthodes et présenté quelques outils de comparaison de performances de ces derniers.

Dans le dernier chapitre, nous allons implémenter mon approche de reconnaissance des gestes de la main.

Chapitre IV

Implémentation et résultats

Sommaire

IV.1 introduction	83
IV.2 Outils et environnement de développement :	83
IV.2.1 Open CV :	83
IV.2.2 Python :	84
IV.2.3 PyQT	86
IV.2.4 Ubuntu 14.04 :	89
IV.2.5 Sublime Text :	90
IV.3 Approche utilisée dans le cadre de la réalisation de notre travail :	91
IV.3.1 Détection de la main et extraction des données nécessaires à l'apprentissage et la reconnaissance des gestes :	91
A.Capture des frames (images) et leur conversion en niveaux de gris :	91
B.Application du filtre gaussien aux images prises (Gaussian Blur) :	92
C.Segmentation des images par la méthode d'Otsu de l'approche seuillage :	92
D.Dessiner les contours de la main et trouver la coque convexe (qui entoure la main) :	93
IV.3.2 Apprentissage et reconnaissance des gestes de la main :	93
A.Le processus de l'apprentissage peut être décomposé en deux phases :	94
B.Comment opèrent les algorithmes d'apprentissage qu'on a utilisé :	94
C.Test des algorithmes d'apprentissage et de reconnaissance :	97
C1.Test de l'approche en utilisant SVM :	100
C2.Test de l'approche en utilisant les forêts aléatoires (random décision forest) :	101
IV.4 Conclusion :	105

IV.1 introduction

Dans ce chapitre nous allons présenter les outils ainsi que l’environnement de développement qui nous ont permis de concrétiser notre travail. Nous décrirons ensuite l’implémentation de notre approche avec des captures d’écrans et des explications.

Nous finirons par une comparaison des résultats de l’approche en utilisant deux algorithmes d’apprentissage différents. section Outils et environnement de développement :

Pour réaliser notre travail, nous avons utilisé la bibliothèque de vision par ordinateur Open CV et le langage de programmation python et certains de ses modules et bibliothèques :

IV.2 Outils et environnement de développement :

IV.2.1 Open CV :

OpenCV (pour Open Computer Vision) est une bibliothèque graphique libre multiplateforme, initialement développée par Intel, spécialisée dans le traitement d’images en temps réel. La société de robotique Willow Garage assure le support de cette bibliothèque depuis 2008. La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu’à la création d’interfaces graphiques basiques. Elle propose la plupart des opérations classiques en traitement bas niveau des images :

- Lecture, écriture et affichage d’une image ;
- Calcul de l’histogramme des niveaux de gris ou d’histogrammes couleurs ;
- Lissage, filtrage ;
- Seuillage d’image (méthode d’Otsu, seuillage adaptatif).

Elle possède plus de 2500 algorithmes optimisés, elle comprend un ensemble complet d’algorithmes d’apprentissage classique et les dernières innovations en vision par ordinateur. Ces algorithmes peuvent être utilisés pour détecter et reconnaître des visages, les gestes de la main, identifier des objets, classer les actions humaines dans les vidéos, suivre les mouvements de caméra, suivre des objets en mouvement, extraire des modèles 3D d’objets, produire des nuages de points 3D à partir de caméras stéréo, assembler des images pour produire des images haute résolution d’une scène, trouver des images similaires à partir d’une base de données d’image, supprimer les yeux rouges des images prises au flash, suivre les mouvements des yeux, reconnaître des paysages et établir des marqueurs pour superposer des réalités augmentées, etc [29]



FIGURE IV.1 – Logo d'Open CV

IV.2.2 Python :

Python est un langage portable, dynamique, extensible, gratuit, qui permet une approche modulaire et orientée objet de la programmation. Python est développé depuis 1989 par Guido van Rossum et de nombreux contributeurs bénévoles.[30]

Caractéristiques du langage :

Détaillons un peu les principales caractéristiques de Python, plus précisément, du langage et de ses deux implantations actuelles :

Python est **portable**, non seulement sur les différentes variantes d'UNIX, mais aussi sur les OS propriétaires : MacOS, BeOS, MS-DOS et les différentes variantes de Window. Un nouveau compilateur, baptisé JPython, est écrit en Java et génère du bytecode Java.

Python est **gratuit**, mais on peut l'utiliser sans restriction dans des projets commerciaux. Python convient aussi bien à des scripts d'une dizaine de lignes qu'à des projets complexes de plusieurs dizaines de milliers de lignes.

La **syntaxe de Python est très simple** et, combinée à des types de données évolués (listes, dictionnaires,...), conduit à des programmes à la fois très compacts et très lisibles. A fonctionnalités égales, un

programme Python (abondamment commenté et présenté selon les canons standards) est souvent de 3 à 5 fois plus court qu'un programme C ou C++ (ou même Java) équivalent, ce qui représente en général un temps de développement de 5 à 10 fois plus court et une facilité de maintenance largement accrue.

Python gère ses ressources (mémoire, descripteurs de fichiers...) sans intervention du programmeur, par un mécanisme de **comptage de références** (comptage de références (proche, mais différent, d'un garbage collector)).

Il n'y a **pas de pointeurs** explicites en Python.

Python est (optionnellement) **multi-threadé**.

Python est **orienté-objet**. Il supporte l'héritage multiple et la surcharge des opérateurs. Dans son modèle objets, et en reprenant la terminologie de C++, toutes les méthodes sont virtuelles.

Python intègre, comme Java ou les versions récentes de C++, un système **d'exceptions**, qui permettent de simplifier considérablement la gestion des erreurs.

Python est **dynamique** (l'interpréteur peut évaluer des chaînes de caractères représentant des expressions ou des instructions Python), orthogonal (un petit nombre de concepts suffit à engendrer des constructions très riches), réflexif (il supporte la métaprogrammation, par exemple la capacité pour un objet de se rajouter ou de s'enlever des attributs ou des méthodes, ou même de changer de classe en cours d'exécution) et introspectif (un grand nombre d'outils de développement, comme le debugger ou le profiler, sont implantés en Python lui-même).

Comme Scheme ou SmallTalk, Python est dynamiquement typé. Tout objet manipulable par le programmeur possède un type bien défini à l'exécution, qui n'a pas besoin d'être déclaré à l'avance.

Python possède actuellement deux implémentations. L'une, **interprétée**, dans laquelle les programmes Python sont compilés en instructions portables, puis exécutés par une machine virtuelle (comme pour Java, avec une différence importante : Java étant statiquement typé, il est beaucoup plus facile d'accélérer l'exécution d'un programme Java que d'un programme Python). L'autre, génère directement du bytecode Java.

Python est **extensible** : comme Tcl ou Guile, on peut facilement l'interfacier avec des bibliothèques C existantes. On peut aussi s'en servir comme d'un langage d'extension pour des systèmes logiciels complexes.

La **bibliothèque standard** de Python, et les paquetages contributifs, donnent accès à une grande variété de services : chaînes de caractères et expressions régulières, services UNIX standard (fichiers, pipes, signaux, sockets, threads...), protocoles Internet (Web, News, FTP, CGI, HTML...), persistance et bases de données, interfaces graphiques.

Python est un langage qui continue à **évoluer**, soutenu par une communauté d'utilisateurs enthousiastes et responsables, dont la plupart sont des supporters du logiciel libre. Parallèlement à l'interpréteur principal, écrit en C et maintenu par le créateur du langage, un deuxième interpréteur, écrit en Java, est en cours de développement.

IV.2.3 PyQT

PyQt est un module libre qui permet de lier le langage Python avec la bibliothèque Qt distribué sous deux licences : une commerciale et la GNU GPL. Il permet ainsi de créer des interfaces graphiques en Python. Une extension de QtDesigner (utilitaire graphique de création d'interfaces Qt) permet de générer le code Python d'interfaces graphiques.

Parmi les bibliothèques et modules de python utilisés :

- **NumPy** est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux. Plus précisément, cette bibliothèque logicielle open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes. NumPy est la base de SciPy, regroupement de bibliothèques Python autour du calcul scientifique.[31]
- **SciPy** est un projet visant à unifier et fédérer un ensemble de bibliothèques Python à usage scientifique. Scipy utilise les tableaux et matrices du module NumPy. Cette distribution de modules est destinée à être utilisée avec le langage interprété Python afin de créer un environnement de travail scientifique très similaire à celui offert par Scilab, GNU Octave, Matlab voire R. Il contient par exemple des modules pour l'optimisation, l'algèbre linéaire, les statistiques, le traitement du signal ou encore le traitement d'images. Il offre également des possibilités avancées de visualisation grâce au module matplotlib. Afin d'obtenir d'excellentes performances d'exécution (point faible des langages interprétés), la plupart des

algorithmes de SciPy et NumPy sont codés en C. Le module NumPy permet d'appliquer des opérations simultanément sur l'ensemble d'un tableau permettant d'écrire un code plus lisible, plus facile à maintenir et donc plus efficace.

Le projet suit la même philosophie que le langage Python à savoir clarté du code, facilité d'apprentissage et extensibilité.[32]

- **Matplotlib** est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques. Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy. Matplotlib est distribuée librement et gratuitement. Sa version actuelle (la 1.4.3 en 2015) est compatible avec la version 3 de Python.[33]
- **Scikit-learn** est une bibliothèque Python dédiée à l'apprentissage automatique. Elle est développée par une équipe de l'Inria. Elle comprend notamment des fonctions pour estimer des forêts aléatoires et des régressions logistiques.[34]
- **Pickle** (Python Object serialisation) est un module python qui implémente un algorithme puissant pour la sérialisation et la dé-sérialisation d'une structure d'objet python.
Pickle est le processus par lequel une hiérarchie d'objet python est convertie en un flot doctet et **unpickeling** est le processus inverse.[35]

Exemple d'utilisation de pickle :

Sauvegarde de données : python pickle

Utiliser pickle pour sauvegarder et récupérer des données de types complexes (listes, dictionnaires, tuples et les classes d'objets).

```
1 >>> import pickle #inclure pickle
2 >>> f = open('save_file.p', 'wb') #Ouvrir un fichier de type pickle
3 >>> ex_dict = {'hello': 'world'} #Affectation de données au dictionnaire
4 >>> pickle.dump(ex_dict, f) #Sauvegarder le dictionnaire dans f
5 >>> f.close() #Fermer f
```

```
1 >>> import pickle #inclure pickle
2 >>> f = open('save_file.p', 'rb') #Ouvrir un fichier de type pickle
3 >>> loadobj = pickle.load(f) #charger le fichier dans loadobj
4 >>> print loadobj['hello'] #Afficher le contenu du fichier loadobj
5
```

- **Metrics** : est un package du Python Package Index, qui produit des métriques pour les programmes python. Dans notre travail on l'utilisera des outils d'évaluation de performance des algorithmes d'apprentissage tel que la confusion matrix et l'Accuracy score.[36]

— **Confusion Matrix** (Matrice de confusion) :

on utilise cet outil qui se trouve dans le package metrcis comme suit :

```
sklearn.metrics.confusion_matrix(y_true, y_pred, labels=None)
```

où :

y_true : est un vecteur contenant les données d'entraînement.

y_pred : est un vecteur conteant les données de test.

labels : est un vecteur contenant la liste des labels (chaque classe a un label défini).

Cette fonction retourne une matrice de confusion.

— **Accurancy score** (pourcentage de classification correcte) :

on utilise cet outil qui se trouve dans le package metrics comme suit :

```
sklearn.metrics.accuracy_score(y_true, y_pred, normalize=True, sample_weight=None)
```

où :

y_true : est un vecteur contenant les données d'entraînement

y_pred : est un vecteur contenant les données de test

normalize : si faux, retourne le nombre d'échantillons classés correctement.

Cette fonction retourne un pourcentage des échantillons correctement classés par rapport au nombre d'échantillons testés.



FIGURE IV.2 – Logo de Python, Numpy, Scipy, Scikit Learn, Matplotlib

Nous avons utilisé comme système d'exploitation Ubuntu 14.04, et comme éditeur de texte Sublime text :

IV.2.4 Ubuntu 14.04 :

Ubuntu est un système d'exploitation open source basé sur la distribution Linux Debian. Son nom provient d'un ancien mot bantou qui signifie « je suis ce que je suis grâce à ce que nous sommes tous ». Dans le même ordre d'esprit, les utilisateurs sont encouragés à étudier son fonctionnement, le modifier, l'améliorer et enfin de le redistribuer.[37]

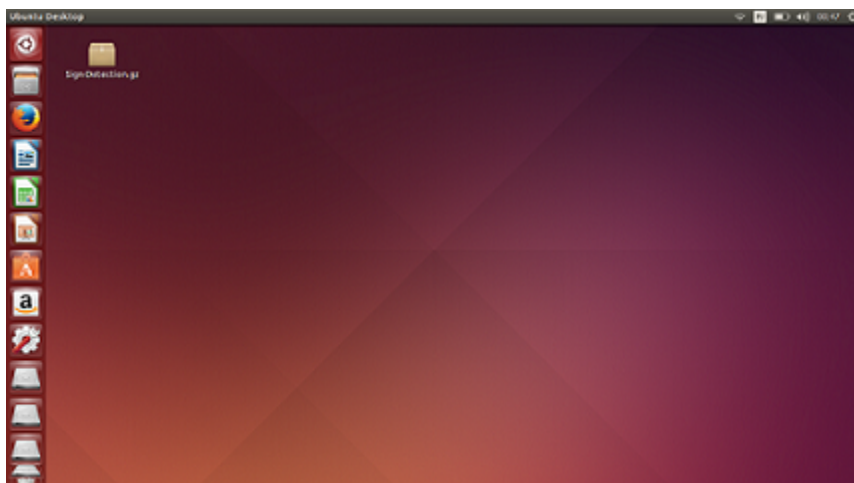


FIGURE IV.3 – Bureau d'ubuntu.

IV.2.5 Sublime Text :

Sublime Text est un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour Vim, riche en fonctionnalités.

Depuis la version 2.0, sortie le 26 juin 2012, l'éditeur prend en charge 44 langages de programmation majeurs (parmi lesquels nous trouvons python), tandis que des plugins sont souvent disponibles pour les langages plus rares.[38]



FIGURE IV.4 – Logo de Sublime Text.

IV.3 Approche utilisée dans le cadre de la réalisation de notre travail :

Pour réaliser notre travail, nous avons utilisé une approche qui peut être divisée en deux parties : dans la première, nous avons extrait des informations à partir des images en opérant sur elles différents traitements, dans la deuxième, nous avons utilisé les informations extraites lors de la première partie pour reconnaître les gestes de la main.

Dans ce qui suit, nous allons détailler les deux parties de l'approche :

IV.3.1 Détection de la main et extraction des données nécessaires à l'apprentissage et la reconnaissance des gestes :

La détection de la main et l'extraction des données dont on a besoin pour la reconnaissance se basent sur la détection des contours et de la coque convexe (Convex Hull en anglais) qui entoure la main.

Pour ce faire, nous suivons les étapes suivantes :

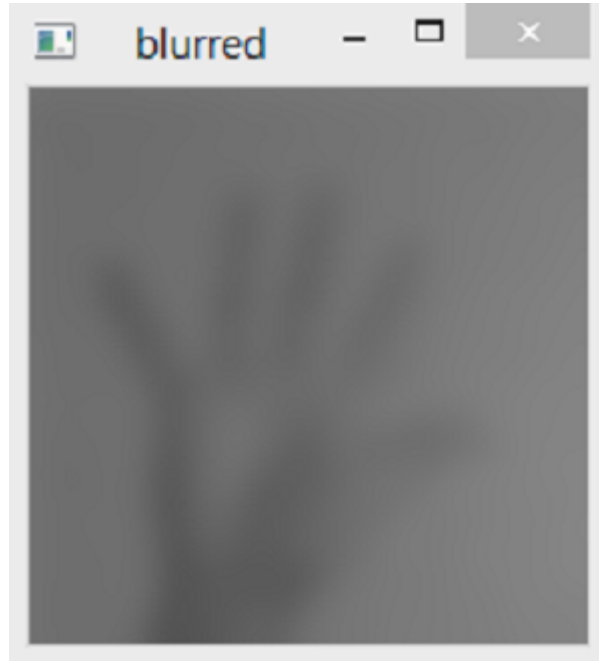
A.Capture des frames (images) et leur conversion en niveaux de gris :

Notre région d'intérêt est la main, donc on prend des images successives de la main (frame en anglais) et on les convertis en niveaux de gris.



B. Application du filtre gaussien aux images prises (Gaussian Blur) :

On utilise le filtre gaussien pour éliminer les bruits et réduire les détails dans les images.



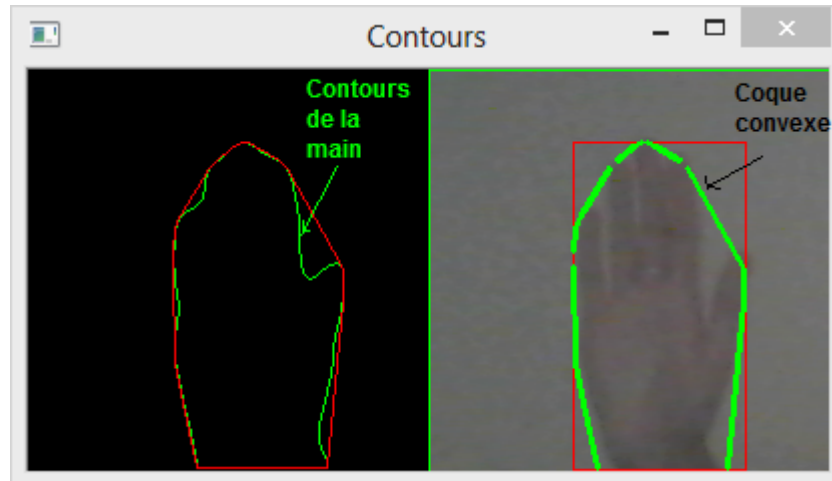
C. Segmentation des images par la méthode d'Otsu de l'approche seuillage :

En termes très simples, ce seuillage permet d'avoir dans l'image seulement deux classes de couleur, à savoir le noir et le blanc ; le blanc représentant la région d'intérêt.



D. Dessiner les contours de la main et trouver la coque convexe (qui entoure la main) :

Cette étape permet de dessiner les contours de la main pour ensuite trouver la coque convexe.



Après la fin de cette première partie, nous avons obtenu une coque convexe entourant la main. Nous avons pu par la suite extraire sous forme de vecteur, les points qui représentent généralement les bouts des doigts et qui ont permis de dessiner cette dernière.

Les coordonnées de ces points-là seront les données qu'utilisera l'algorithme d'apprentissage pour reconnaître les gestes de la main.

IV.3.2 Apprentissage et reconnaissance des gestes de la main :

L'apprentissage supervisé consiste à apprendre le lien entre deux types de données :

- Des données observées x : dans notre cas ces données sont le vecteur de coordonnées d'un geste de la main .
- Des variables externes y : qu'on essaie de prédire, aussi appelées étiquette, dans notre cas, les étiquettes sont l'alphabet de la langue française tel que, chaque étiquette correspond à une lettre de l'alphabet.

Remarque :

Pour réaliser la partie apprentissage et reconnaissance, on a utilisé la bibliothèque d'apprentissage artificiel de python (scikit learn), cette dernière impose le format de données utilisé par les classifieurs qu'elle implémente.

Dans scikit learn, les classifieurs travaillent avec des données en forme de vecteurs d'où l'extraction des coordonnées des points de la coque convexe en forme de vecteurs (les données x).

Quant aux étiquettes elles sont placées dans un vecteur de type string puisqu'il va comporter les lettres de l'alphabet.

A. Le processus de l'apprentissage peut être décomposé en deux phases :

1. **La phase d'entraînement :** Les classifieurs qu'on va utiliser implémentent la fonction $\text{fit}(x,y)$, qui va leur permettre de trouver un lien, exprimé avec une fonction, entre les données x et y .
2. **La phase de test :** avec la fonction $\text{predict}(x)$, le classifieur va deviner l'étiquette des données x qu'on lui fournit, en utilisant la fonction induite lors de la phase d'entraînement.

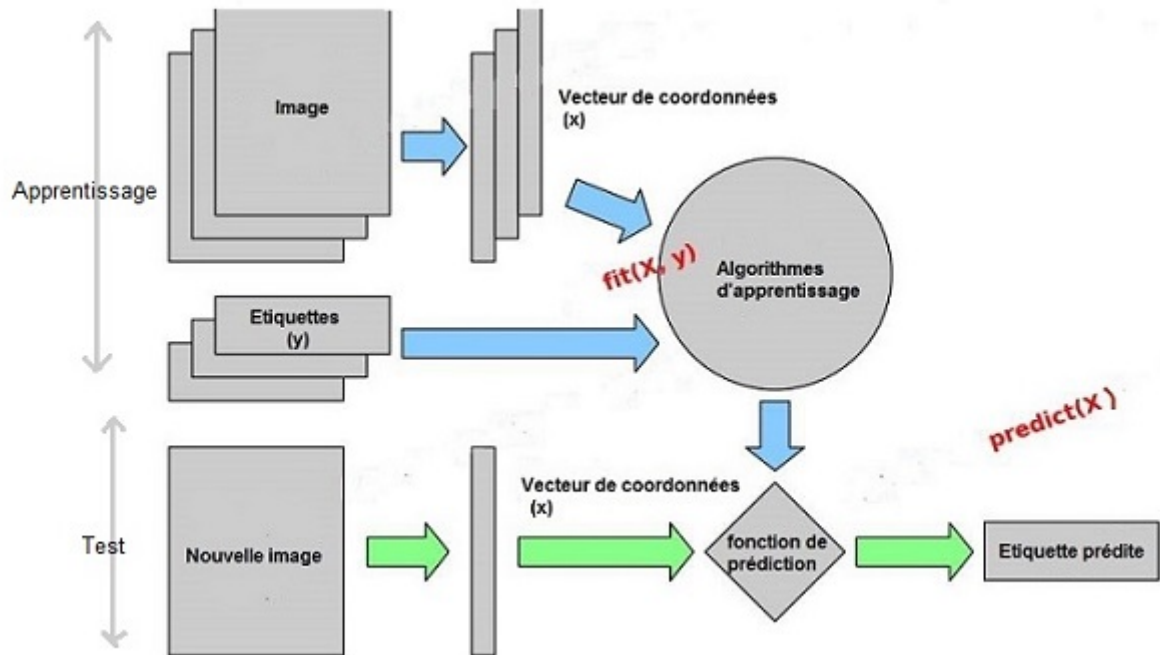


FIGURE IV.5 – Schéma décrivant le processus d'apprentissage.

B. Comment opèrent les algorithmes d'apprentissage qu'on a utilisé :

Les algorithmes utilisés dans la partie apprentissage sont des algorithmes qui opèrent en boîte noire, c'est à dire qu'ils utilisent les données qu'on leur fournit en entrées, et nous retournent le résultat sans montrer les détails du traitement. Dans ce qui suit nous allons expliquer brièvement comment ces algorithmes procèdent réellement. **Dans le cas de support vector machine (SVM) :**

SVM est à l'origine, un classifieur binaire, mais dans notre travail il joue le rôle d'un classifieur multi-classes. Comme on a 26 étiquettes et comme chaque classe est étiquetée par une étiquette, on aura un total de 26 classes.

SVM utilise la méthode one against all : pour pouvoir résoudre ce problème de classification multi-classes, SVM utilise la méthode one against all qui est une méthode simple qui consiste à créer un classifieur binaire pour chacune des 26 classes.

SVM est entraîné pour dire, pour chaque classe(i) des 26 classes, si l'élément appartient à la classe (i) ou pas, et il distingue ainsi les éléments qui appartiennent à la classe (i) des éléments de toutes les autres classes.

Chaque séparateur SVM des 26 utilisés, cherche un séparateur optimal qui maximise la marge entre les deux classes.

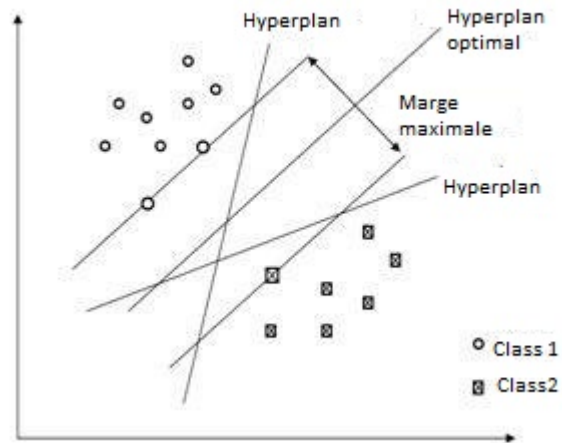


FIGURE IV.6 – Recherche du séparateur optimal dans chacun des 26 classifieurs

SVM utilise le kernel trick : comme les données de l'apprentissage (le vecteur de coordonnées qu'on fournit à l'algorithme) ne sont pas séparable linéairement, svm utilise le kernel trick, et pour cela il se sert d'une fonction de transformation polynomiale de degré 10.

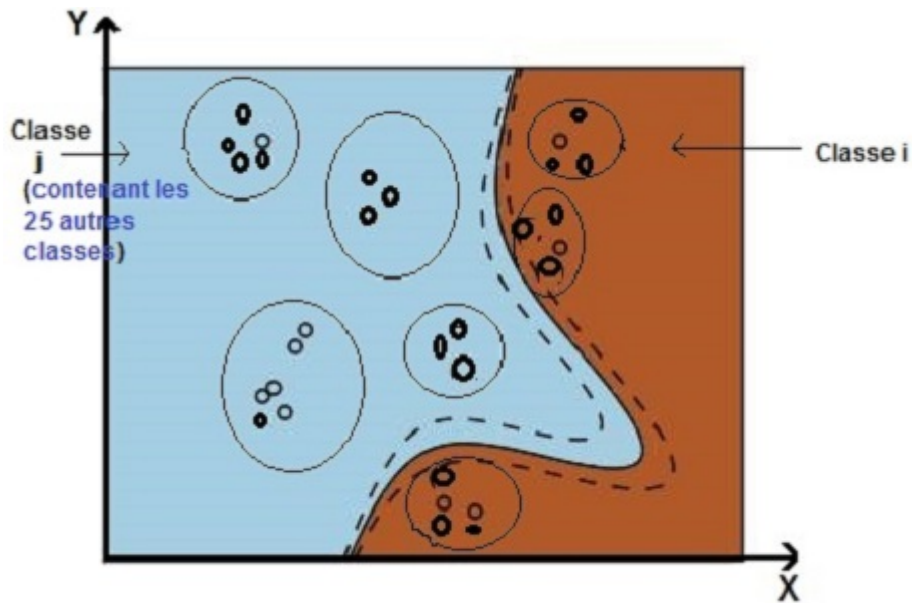


FIGURE IV.7 – Schéma qui montre l'un des 26 séparateurs SVM dans le feature space.

Dans le cas des forêts aléatoires (Random decision forest) :

Lors de la phase d'entraînement l'algorithme des forêts aléatoires construit plusieurs arbres de décision qui, à la fin de leurs construction auront 26 feuilles chacun , chaque feuille correspond à l'une des 26 classes. Rappelons que chaque classe est étiquetée par une lettre de l'alphabet de la langue française.

La disposition des variables dans les noeuds de chaque arbre est différente des autres arbres, cette disposition est effectuée aléatoirement.

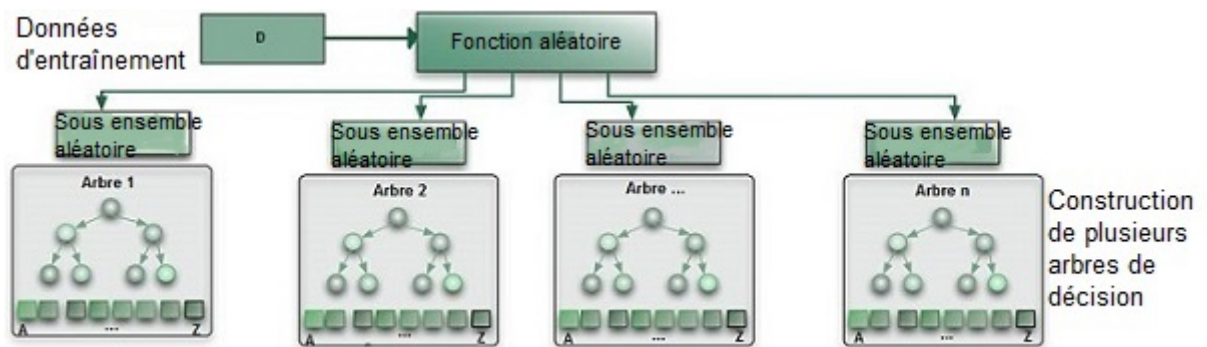


FIGURE IV.8 – Schéma illustrant la phase de construction de la forêt aléatoire.

Lors de la phase de prédiction, l'algorithme des forêts aléatoires parcourt chaque arbre de décision et

enregistre le résultat de la prédiction obtenue, vers la fin il compare le résultat de tout les arbres, et la prédiction faite par le plus grand nombre d'arbres sera retenue.

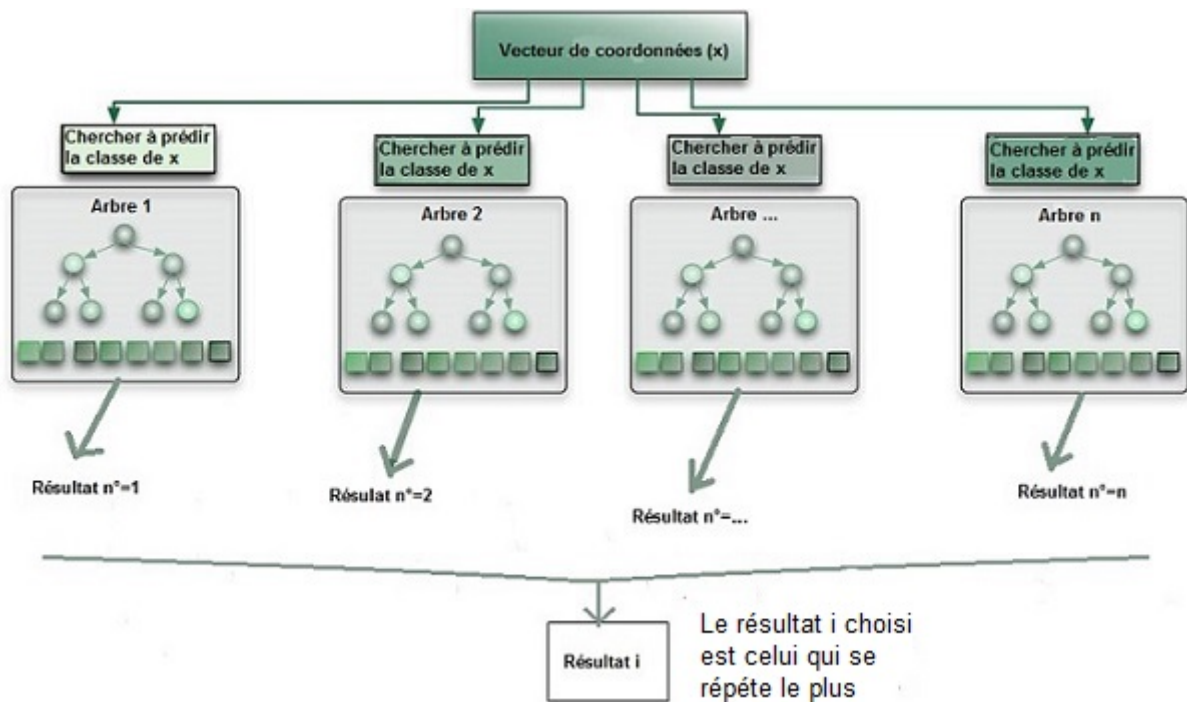


FIGURE IV.9 – Schéma illustrant la phase de prédiction.

C. Test des algorithmes d'apprentissage et de reconnaissance :

Afin d'évaluer les performances des algorithmes d'apprentissage on peut utiliser deux différentes procédures :

La première procédure nous permet de calculer le taux de réussite de l'entraînement (training accuracy) en mesurant à quel point le modèle classe correctement les données de l'entraînement. Cependant cette procédure peut entraîner le surapprentissage.

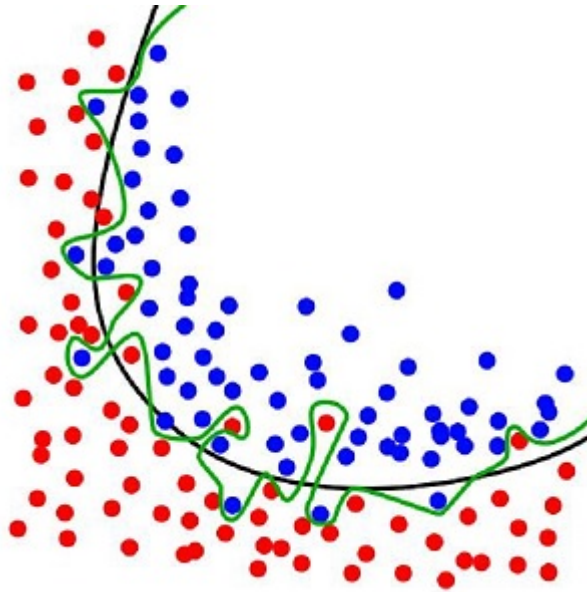


FIGURE IV.10 – Schéma montrant le surapprentissage

Le surapprentissage, comme celui présenté dans l'image ci dessus, est la situation où le modèle apprend le bruit dans les données (la ligne verte) au lieu d'apprendre le signal (la ligne noire), dans ce cas le modèle ne pourra reconnaître que les données déjà apprises lors de la phase d'apprentissage.

Pour éviter ce problème on a recours à une autre procédure de test qui sépare les données en données d'entraînement et en données de test.

Cette procédure permet de calculer le taux de réussite des test (test accuracy) qui permet une meilleure estimation des données futures .

Dans cette procédure l'idée est de :

- Diviser le dataset en deux parties :
 - Un dataset d'entraînement.
 - Un dataset de test.
- On entraîne le modèle sur le dataset d'entraînement puis on teste avec les données du deuxième dataset c'est à dire le dataset de test ,et on évalue les résultats grâce à la matrice de confusion et le taux de reconnaissance (Accuracy score).

	Attribut 1	Attribut 2	Réponse	
X_train	1	2	2	y_train
	3	4	12	
X_test	5	6	30	y_test
	7	8	56	
	9	10	90	

FIGURE IV.11 – Schéma montrant la division d'un dataset en deux : dataset d'entrainement et dataset de test.

L'idée est que puisque nous avons évalué le modèle sur des données qui n'ont pas été utilisées pour entraîner le modèle, on a mieux simuler comment le modèle va réagir avec des données futures.

Dans notre travail nous avons utilisé la deuxième procédure d'évaluation afin d'éviter le surapprentissage. Pour cela nous avons utilisé un dataset contenant des images de l'alphabet de la langue des signes :



FIGURE IV.12 – Capture d'écran du dataset que nous avons utilisé

Pour de meilleurs résultats on a introduit la notion de stéréovision et cela en prenant chaque lettre des différents angles possibles :

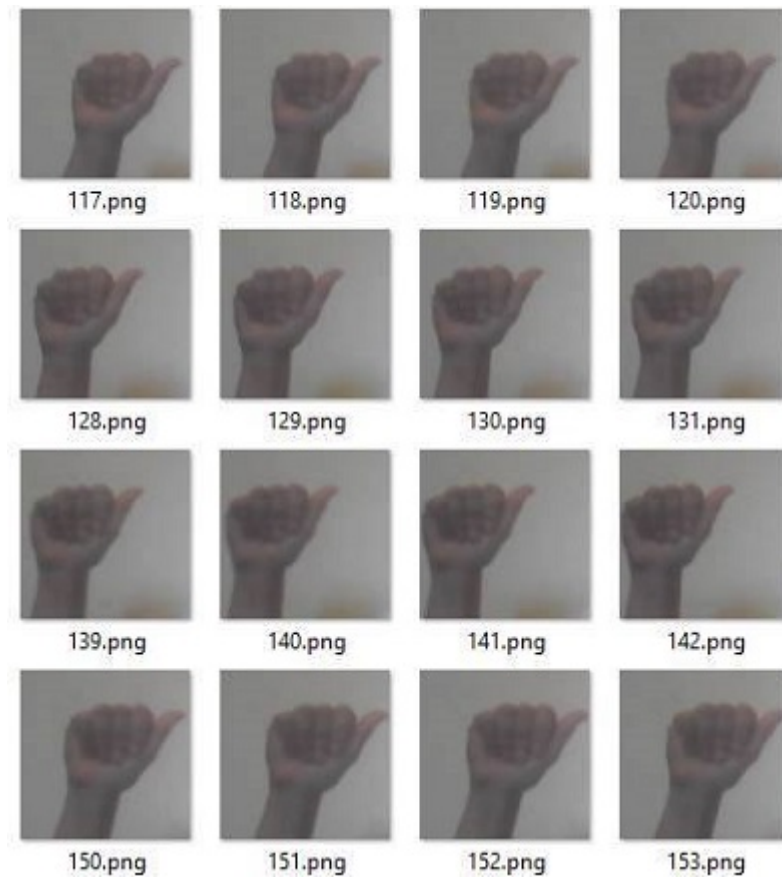


FIGURE IV.13 – La lettre A de la langue des signes sous différents angles.

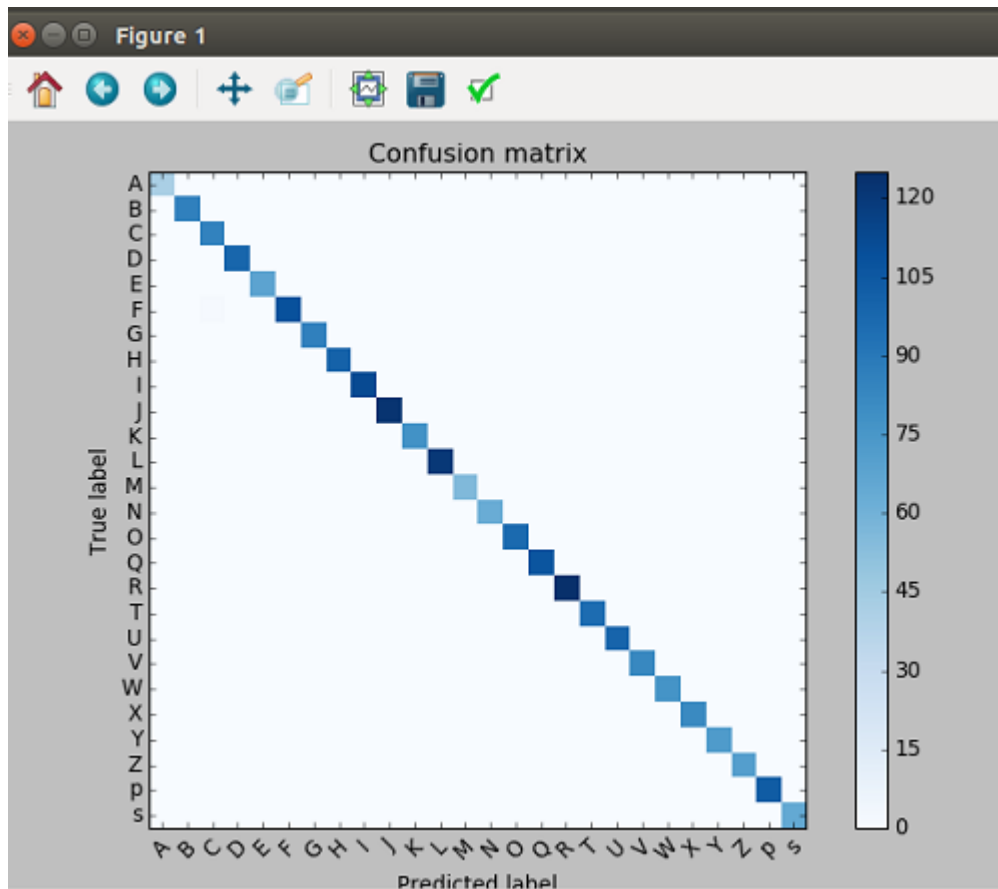
C1. Test de l'approche en utilisant SVM :

On entraîne le modèle sur le train-dataset puis on teste le test-dataset et on obtient les résultats suivants :
le **taux de reconnaissance(testing accuracy)** : en utilisant l'algorithme d'apprentissage SVM.

```
Accuracy: 0.999567099567
init done
opengl support available
[Finished in 23.9s ]
```

FIGURE IV.14 – Capture d'écran du taux de reconnaissance obtenu en utilisant SVM.

La matrice de confusion(confusion matrix) : en utilisant l'algorithme d'apprentissage SVM



La matrice de confusion retournée ici est représenté sur un axe orthonormé dont l'axe des abscisses contient les classes prédites et l'axe des ordonnées contient les classes déjà classées correctement (Dans notre cas chaque classe est une lettre de l'alphabet de la langue des signes)

La diagonale obtenue indique alors que les classes prédites et celles déjà classées sont identiques, c'est à dire que la reconnaissance a été bonne.

la variation de la couleur de la diagonale revient au fait que dans une classe, la prédiction peut être correcte avec toutes les instances de cette classe ou elle peut être correcte mais pas avec la totalité des instances de la classe : la couleur foncée indique une prédiction correcte sur la totalité des instances de la classe, et plus la couleur s'atténue plus la prédiction n'est correcte que sur certaines instances de la classe.

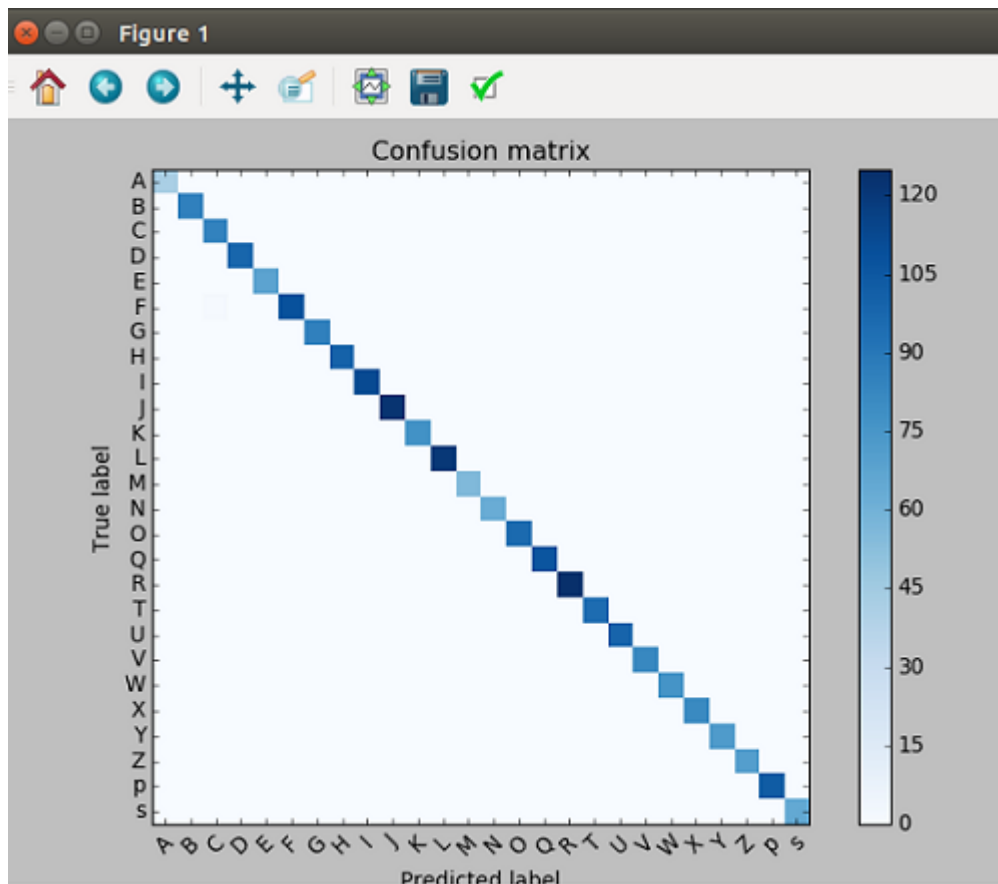
C2.Test de l'approche en utilisant les forêts aléatoires (random décision forest) :

Le taux de reconnaissance (testing accuracy) : en utilisant l'algorithme d'apprentissage des forêts aléatoires

```
Accuracy: 0.999567099567
init done
opengl support available
[Finished in 14.2s]
```

FIGURE IV.15 – Capture d’écran du taux de reconnaissance obtenu en utilisant les forêts aléatoires.

La matrice de confusion(confusion matrix) : en utilisant l’algorithme d’apprentissage des forêts aléatoires.



La matrice de confusion retournée ici est représenté sur un axe orthonormé dont l’axe des abscisses contient les classes prédites et l’axe des ordonnées contient les classes déjà classées correctement (Dans notre cas chaque classe est une lettre de l’alphabet de la langue des signes)

La diagonale obtenue indique alors que les classes prédites et celles déjà classées sont identiques, c’est à dire que la reconnaissance a été bonne.

la variation de la couleur de la diagonale revient au fait que dans une classe, la prédiction peut être correcte avec toutes les instances de cette classe ou elle peut être correcte mais pas avec la totalité des instances de la classe : la couleur foncée indique une prédiction correcte sur la totalité des instances de

la classe, et plus la couleur s'atténue plus la prédiction n'est correcte que sur certaines instances de la classe.

Remarque :

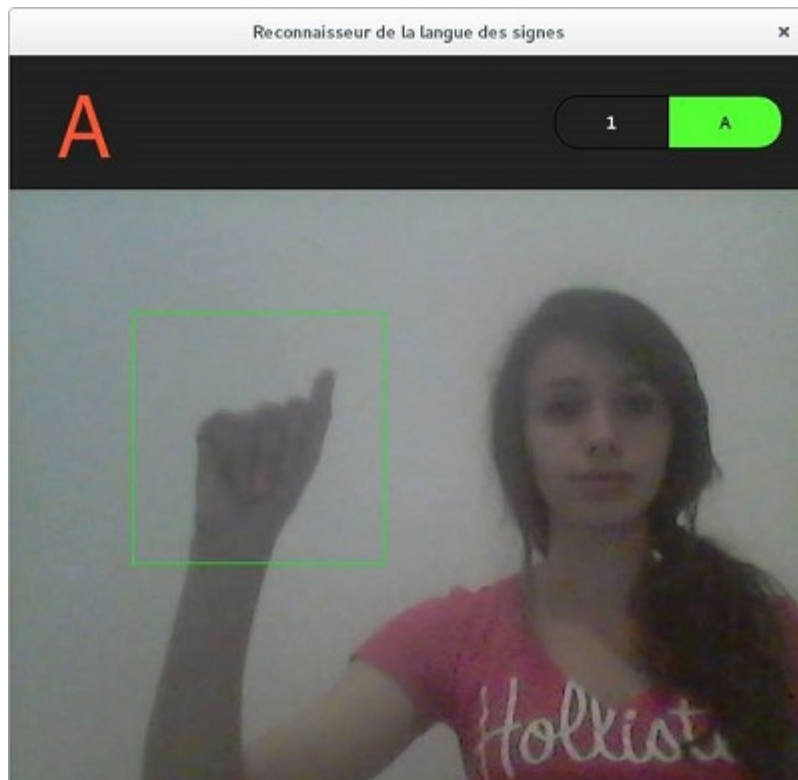
La phase test montre que les forêts aleatoires donnent des résultats aussi satisfaisants que ceux donnés par SVM. Par contre nous remarquons que le temps d'exécution en utilisant les forêts aléatoires est plus court que lors de l'utilisation de SVM, c'est pour cela que nous allons utiliser les forêts aléatoires(Random Decision Forest) dans notre approche de reconnaissance des gestes de la main.

-L'application de l'algorithme Random Decision Forests sur des données (images) n'appartenant pas au dataset de test ou d'entraînement mais sur des données acquises en temps réel(capture d'images en temps réel) :

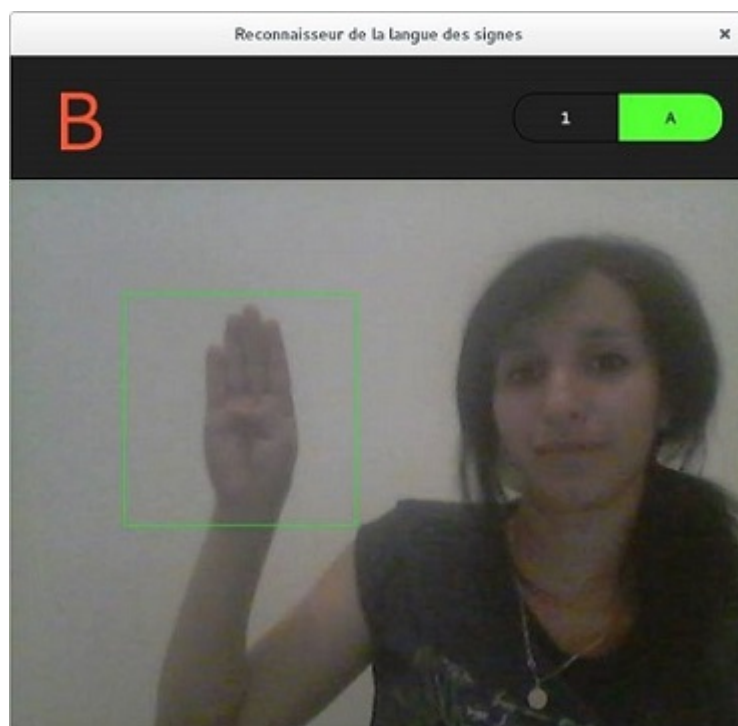
-Sur cette interface, nous pouvons distinguer deux modes, le mode alphabet et le mode comptage :

Le mode alphabet :

La lettre A en langue des signes et son équivalent en langage écrit (à gauche de l'image)

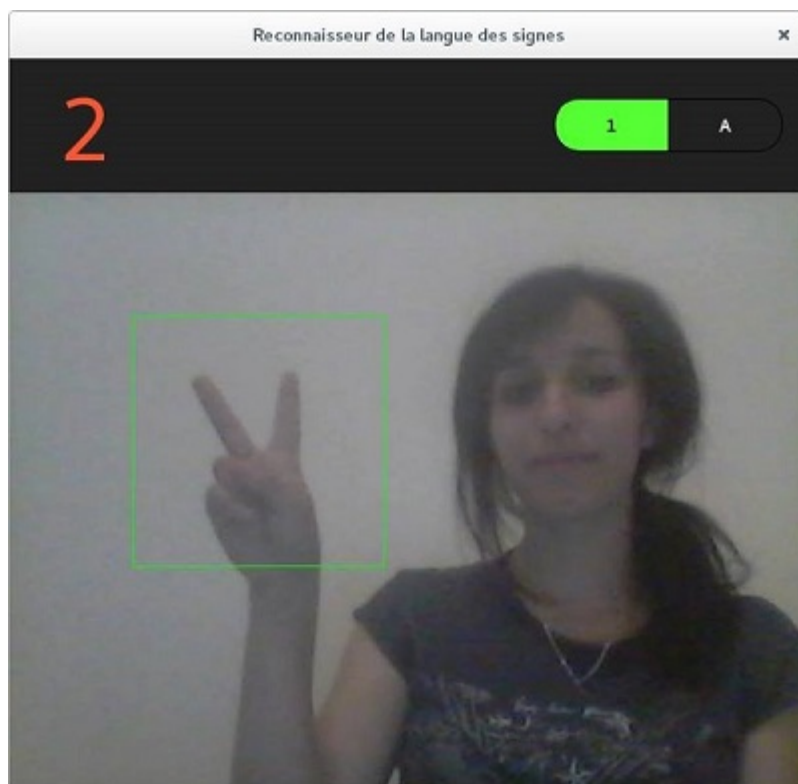


La lettre B en langue des signes et son équivalent en langage écrit (à gauche de l'image)

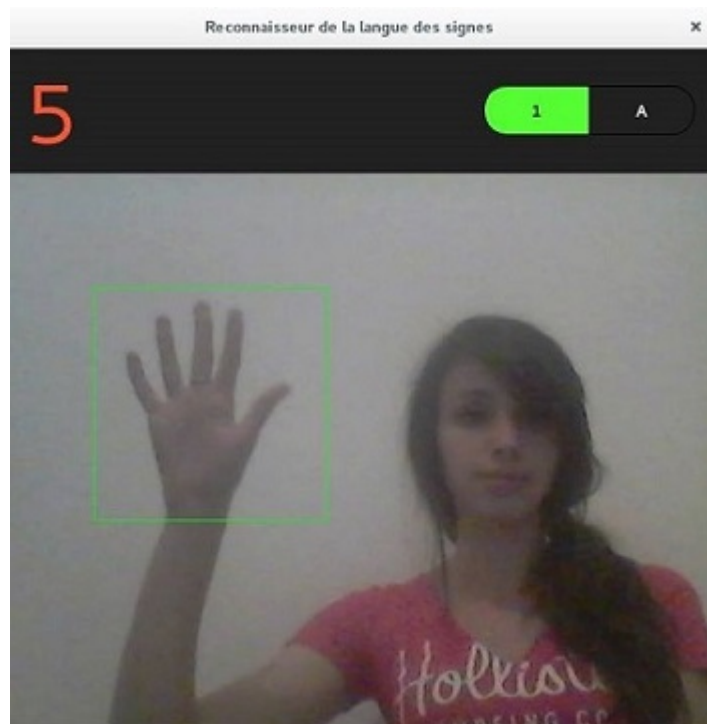


Le mode comptage :

Retranscription du chiffre 2 en langage écrit (à gauche de l'image)



Retranscription du chiffre 5 en langage écrit (à gauche de l'image)



IV.4 Conclusion :

Dans ce chapitre, nous avons passé en revue le langage de programmation, les modules, les bibliothèques ainsi que l'environnement de travail qui nous ont permis de mener ce projet à son terme.

Ensuite nous avons fait une description de notre approche, qui consiste à détecter la zone d'intérêt qui est dans notre cas la main dans une image, d'en extraire des données sous forme de vecteur contenant les coordonnées des points de la coque convexe qui entoure la main et cela pour chaque geste correspondant à une lettre de l'alphabet de la langue des signes, nous avons introduit par la suite, la notion de stéréovision en enrichissant notre dataset avec des images du même geste sous différents angles, après cela, nous avons fournit ces données en entrée à deux algorithmes d'apprentissage à savoir SVM et les Forêts aléatoires afin de les entraîner dessus et cela dans le but qu'ils puissent reconnaître de nouvelles instances.

Nous avons finit par tester les deux algorithmes d'apprentissage sur de nouvelles instances, puis comparé leur résultats grâce à des outils d'évaluation de performance : La matrice de confusion et le taux de bonne reconnaissance et nous avons conclu que les deux algorithmes sont appropriés pour la réalisation de notre travail.

Conclusion générale

Il était question, dans ce présent mémoire, de la réalisation d'un système capable de détecter et de reconnaître les gestes de la main en se basant sur différents algorithmes conçus à cet effet à savoir : L'algorithme de la méthode d'Otsu pour la phase de détection, l'algorithme SVM, celui des arbres de décisions et des forêts aléatoires dans la phase de reconnaissance.

Le système que nous avons obtenu permet la reconnaissance de l'alphabet de la langue des signes. Par ailleurs, nous avons rajouté à ce dernier la possibilité de reconnaître les gestes permettant de compter jusqu'à cinq.

La réalisation de ce système nous a permis de renforcer nos connaissances acquises tout au long de notre cursus universitaire avec de nouveaux concepts théoriques, que nous avons pu mettre en pratique par la suite. En effet, de par l'immensité du domaine de la reconnaissance du geste de la main, qui se trouve à la croisée des chemins entre de nombreux champs d'études tels que l'analyse des données ou encore le traitement d'images, les différentes étapes qui nous ont menées à la réalisation d'un tel système nous ont permis de nous confronter à des domaines très divers tels que l'intelligence artificielle et la vision par ordinateur.

Etant donnée la complexité de la langue des signes, notre premier travail a été de reconnaître l'alphabet comme cité ci-dessus. Ce travail là a posé les bases d'un projet d'intérêt général. Dans des travaux que nous envisageons de faire plus tard, nous citons :

- L'amélioration de la reconnaissance des gestes de la main.
- La reconnaissance de mots et de phrases de la langue de signes et leur retranscription en langage écrit.
- Le processus inverse qui consiste à traduire des mots et des phrases en langue des signes.
- Le déploiement de l'application sur mobiles et tablettes et sa mise en disponibilité sur le marché algérien.

Bibliographie

- [1] <https://fr.wikipedia.org/wiki/Communication>.
- [2] <http://www.etre-bien-au-travail.fr/glossaire/c/communication-verbale>.
- [3] <https://fr.wikipedia.org/wiki/Communication-non-verbale>.
- [4] <https://fr.wikipedia.org/wiki/Geste>.
- [5] Julien Thomet. Une vue d'ensemble de la reconnaissance de geste.
- [6] Maria Karam and m. c. schraefel. A taxonomy of gestures in human computer interactions. 2005.
- [7] Olivier Losson. Modelisation du geste communicatif et realisation d'un signeur virtuel de phrases en langue des signes francaise. 2000.
- [8] <https://fr.wikipedia.org/wiki/Langue-des-signes>.
- [9] <http://www.signesetformations.com/cours-langue-des-signes/histoire-de-la-langue-des-signes/>.
- [10] <https://fr.wikipedia.org/wiki/Langue-des-signes>.
- [11] Bossard Bruno. Problemes poses par la reconnaissance de gestes en langue des signes. 2004.
- [12] <https://fr.wikipedia.org/wiki/Langue-des-signes-francaise>.
- [13] et al. Radu Horaud, Olivier Monga. Vision par ordinateur : outils fondamentaux. 1995.
- [14] www.commentcamarche.net/contents/video-numerique-20934723001217/.
- [15] www.commentcamarche.net/contents/video-numerique-20934723001217/.
- [16] http://ww2.ac-poitiers.fr/sc_phys/IMG/pdf/Images_numeriques.pdf.
- [17] <https://fr.wikipedia.org/wiki/Image-numerique>.
- [18] <http://www.webmarketing-com.com/2012/11/06/16580-quels-sont-les-9-formats-differents-pour-une-imagehash.n6zZbDDd.dpuf> 2012.
- [19] <http://www.commentcamarche.net/contents/1216-traitement-d-images>.
- [20] <https://docs.gimp.org/fr/plugin-convmatrix.html>.
- [21] <http://xphilipp.developpez.com/articles/filtres> 2012.
- [22] <https://fr.wikipedia.org/wiki/Segmentation-d-image>.
- [23] Arnaud DESLANDES. Reconnaissance du geste humain par vision artificielle : Application a la langue des signes. 2002.

- [24] s w zucker. Region growing : childhood and adolescence. computer vision graphics and image processing. 1967.
- [25] john Bernsen. Dynamic thresholding of grey level images. 1986.
- [26] Antoine Cornuejols Lorent Micelet. Apprentissage artificiel, concepts et algorithmes. 2003.
- [27] [http ://scikit-learn.org/stable/modules/svm.html](http://scikit-learn.org/stable/modules/svm.html).
- [28] [https ://lemakistatheux.wordpress.com/2015/09/03/les-forets-aleatoires/](https://lemakistatheux.wordpress.com/2015/09/03/les-forets-aleatoires/).
- [29] [https ://fr.wikipedia.org/wiki/OpenCV](https://fr.wikipedia.org/wiki/OpenCV).
- [30] [http ://www.grappa.univ-lille3.fr/polys/python2/apb.html](http://www.grappa.univ-lille3.fr/polys/python2/apb.html).
- [31] [https ://fr.wikipedia.org/wiki/NumPy](https://fr.wikipedia.org/wiki/NumPy).
- [32] [https ://fr.wikipedia.org/wiki/SciPy](https://fr.wikipedia.org/wiki/SciPy).
- [33] [https ://fr.wikipedia.org/wiki/Matplotlib](https://fr.wikipedia.org/wiki/Matplotlib).
- [34] [https ://fr.wikipedia.org/wiki/Scikit-learn](https://fr.wikipedia.org/wiki/Scikit-learn).
- [35] [https ://docs.python.org/2/library/pickle.html](https://docs.python.org/2/library/pickle.html).
- [36] [https ://pypi.python.org/pypi/metrics](https://pypi.python.org/pypi/metrics).
- [37] [https ://fr.wikipedia.org/wiki/Ubuntu](https://fr.wikipedia.org/wiki/Ubuntu).
- [38] [https ://fr.wikipedia.org/wiki/Sublime-Text](https://fr.wikipedia.org/wiki/Sublime-Text).