

MA-004-470-1

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique

Université SAAD DAHLEB-Blida

Faculté des Sciences

Département d'Informatique

Pour l'obtention du diplôme de Master
en Informatique

Option : Ingénierie des logiciels

THEME

Détection et Reconnaissance des plaques d'immatriculation à partir d'image

ORGANISME D'ACCUEIL :



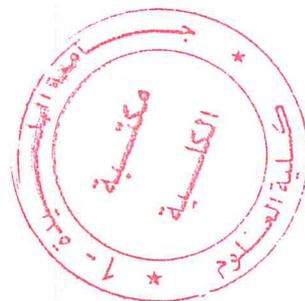
Réalisé par :

BAZERIA Sakina & MEGUENNI Ihcene

Promotrice : Mme ABED HAFIDA

Encadreur : Mme SETITRA Insaf

Président du jury: Mr Cherif Zahar Amine
2016/2017



MA-004-470-1

Remerciement

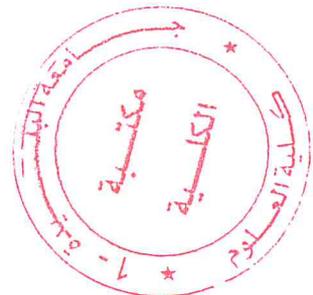
Avant tous et en premier lieu nous remercions « **ALLAH** » le tout puissant de nous avoir aidé et de nous avoir donné la force, le courage et la volonté à accomplir ce modeste travail.

Nous tenons à exprimer nos vifs remerciements à notre promotrice Madame **ABED Hafida** professeur à l'université de Blida, pour sa compréhension, ses précieux conseils, ainsi sa disponibilité et sa confiance en nous.

Nous remercions aussi vivement notre encadreur Madame **SETTRA Insaf**, ingénieur au **CERJST** pour son entière disponibilité, son amabilité et simplicité, ses conseils et encouragements. D'avoir tout mis en œuvre pour que nous puissions donner le meilleur de nous-mêmes et pour tous les moyens qu'elle a mis à notre disposition durant notre période de stage, nous la remercions sincèrement.

Qu'elle trouve ici l'expression de notre profonde gratitude d'avoir acceptés de diriger et de corriger ce travail.

En fin, nous remercions l'ensemble des enseignants et tout le personnel, du département d'informatique de l'université de Blida.



dedicace

Je dédie ce mémoire à :

Mes parents :

Ma mère, qui a œuvré pour ma réussite, par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, qu'elle reçoit à travers ce modeste travail, l'expression de mes sentiments et de mon éternelle gratitude.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.

Mes sœurs Madina et Kawther qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

A ma très chère la petite fille SERINE.

A ma promotrice Mme ABED et mon encadreur Mme SETITRA Insaf.

*A tous mes amis, plus particulièrement Khadidja, Fella, Rafik,
en Témoignage de mon amitié sincère.*

À celle qui a su m'écouter, supporter ma mauvaise humeur, mes angoisses et mon caractère insupportable, ma meilleure amie et mon binôme IHCENE, Je la remercie infiniment pour sa patience, sa gentillesse et sa grande générosité.

Sakina

dedicace

A mes très chers parents Meguenni Abd el Hakim et Bouchenafa Karima

qui depuis mon plus jeune âge ont toujours fait leur maximum,

en consacrant temps et argent, pour m'éveiller et

m'encourager dans mes passions. C'est grâce à vous et pour vous que

je me donne à fond dans mes études.

Aucun mot ne saurait exprimer ce que je vous dois, ni combien je vous aime.

Qu'Allah vous bénisse.

A ma sœur Maroua que j'adore.

A mes deux frères Mohamed Amine, et Zakaria.

A Ma promotrice Madame Abed Hafida qui m'a facilité le chemin de mes

Etudes, par ses précieux conseils et orientations bénéfiques.

*A mon encadreur Madame Setitra Insaf pour son aide précieux, sa patience, sa
disponibilité et sa compréhension.*

*A mon cher binôme et amie Sakina pour sa bonne compagnie et pour les meilleurs
moments que nous avons passés ensemble.*

*A tous mes amis, spécialement à ma chère Ilhem, Chahinez, ainsi qu'à Oussama,
sources de mon inspiration et mon courage, qui ont contribué à la réussite de ce
travail d'une manière indirecte, d'y avoir apporté tant d'humeur et d'amour et pour
tout le soutien moral prodigué dans les moments les plus difficiles.*

A tous mes collègues, plus particulièrement : Nacer et Omar

Je dédie ce travail

Ihcene

Abstract

This document describes an efficient technique of locating and extracting license plate and recognizes each segmented character of an image.

In this work, initially we proposed a method which is based on morphological operations, where different structuring elements are used to maximally eliminate non-plate region and enhance plate region. Next, the detected license plate is fed as input to the segmentation module; the character segmentation is done using adaptive thresholding and connected component analysis. Finally, the extracted characters output is trained using a classifier for recognition module.

This system is developed in JAVA (NetBeans 8.2) using OpenCV 3.2.0.

The proposed system is mainly applicable to Algerian license plate and can be extended for other countries.

Keywords: Locating and extracting license plate, morphological operations, structuring elements, character segmentation, adaptive thresholding, connected component, recognition.

Résumé

Ce document, présente une technique efficace pour la localisation et l'extraction des plaques d'immatriculation à partir d'images, ainsi que la segmentation et la reconnaissance des caractères de la plaque.

Dans ce travail, nous proposons une méthode basée sur des opérations morphologiques, ou différents éléments structurants sont utilisés pour éliminer des régions non désirer et améliorer la région de la plaque. La plaque d'immatriculation détectée sera utilisée comme entrée pour le module de segmentation, ce dernier était effectué en appliquant un seuil adaptatif et puis en segmentant chaque composant connecté. Enfin, nous entraînons un classifieur pour le module de reconnaissance des caractères.

Notre système est développé en JAVA (Netbeans 8.2) using OpenCV 3.2.0.

Ce système s'applique principalement aux plaques d'immatriculations algériennes, mais peut être étendu pour d'autres pays.

Mots-clés : Localisation et extraction des plaques d'immatriculations, opérations morphologiques, éléments structurants, segmentation, reconnaissance des caractères, seuil adaptatif, composants connecté.

ملخص :

هذه الوثيقة تقدم تقنية فعالة لتحديد واستخراج لوحة الترخيص والتعارف على كل رقم موجود داخل أجزاء الصورة.

في هذا العمل اقترحنا في البداية طريقة تقوم على العمليات الشكلية حيث يتم استخدام عناصر هيكلية مختلفة لتخلص بأقصى قدر من المناطق التي لا تمثل اللوحة واستخراج منطقة تواجد اللوحة. بعد ذلك يتم توجيه اللوحة إلى مرحلة التقسيم إلى أجزاء يستخدم التحليل و المكون المتصل وأخيراً يتم التعرف على الأرقام الموجودة في كل جزء باستخدام مصنف لمرحلة التعرف.

تم تطوير هذا النظام بالـ java في (NetBeans 8.2) باستخدام OpenCV 3.2.0 . وينطبق هذا النظام المقترح بشكل رئيسي على لوحة الترخيص الجزائرية ويمكن تمديده للبلدان الأخرى

كلمات البحث

العمليات الشكلية ، المكون المتصل ، عناصر هيكلية ، تحديد واستخراج لوحة الترخيص ، مرحلة التقسيم و مرحلة التعرف.

Table des matières

Introduction Générale

Chapitre 1 : Notions de base

1	Introduction :	1
2	Définition d'une image :	1
2.1	Images Analogique:	1
2.2	Images Numérique:	2
2.3	Pixel:	2
3	Espace de représentation d'une image :	3
3.1	Espace RGB:	3
3.2	Espace HSV :	4
3.3	Espace Lab:	5
3.4	Image en niveaux de gris :	6
3.5	Image Binaire :	6
4	Operations sur l'image :	7
4.1	Filtrage de l'image:	7
4.1.1	Processus:	7
4.1.2	Exemple de calcule :	8
4.2	Détection de contour:	9
4.2.1	Filtre Canny :	9
4.2.2	Filtre Sobel et Prewitt :	12
4.2.3	Filtre Gaussien:	14
4.2.4	Filtre Laplacien :	16
4.3	Binarisation d'une Image :	19
4.3.1	La méthode d'OTSU :	20
4.4	Opération Morphologique :	24
4.4.1	La dilatation:	24
4.4.2	L'Erosion :	25
4.4.3	L'Ouverture:	26
4.4.4	La fermeture :	26

4.5	Amincissement de l'image:	27
4.6	Analyse en composantes connexes:	29
	Fonctionnement d'Étiquetage en composantes connexes :	30
4.7	Clustering:.....	31
4.7.1	Mean shift:	31
4.7.2	K-means:	32
4.8	Descripteurs d'image:.....	33
4.8.1	Histogrammes de projection:	33
4.8.2	Projection horizontale:	34
4.8.3	Projection verticale:.....	35
4.9	Analyse de l'image:	35
4.9.1	Réseau de neurones :	36
4.9.2	Perceptron:	36
4.9.3	Classificateur en cascade:.....	37
5	Travaux liés:	40
5.1	Détection de la plaque d'immatriculation :	40
5.2	Segmentation des caractères:	44
5.3	Reconnaissance des caractères:	46
6	Conclusion:	47

Chapitre 2 : Conception

1-	Introduction.....	48
2-	Conception globale :	48
2.1	Description :	48
2.2	Organigramme global :	49
3.	Conception des différents modules :	49
3.1	Conception du module de détection de la plaque :.....	49
3.1.1	Description générale du module de détection de la plaque :	50
3.1.2	Algorithme du module de détection de la plaque :	51
3.2-	Conception du module segmentation des caractères :	52
3.2.1-	Description générale du module segmentation :	53

3.2.2-Algorithmme :.....	54
3.3- Conception du module reconnaissance des caractères :	54
3.3.1- Organigramme de reconnaissance :	55
4- Conclusion :	56

Chapitre 3 : Implémentation

1. Introduction :	57
2. Présentation d'environnement de travail :	57
2.1. Netbeans :	57
2.2. Java :	58
2.3 WampServer :	58
2.4. OpenCV :	58
3. Choix et paramétrage des Fonctions :	62
3.1. Fonction utilisées pour la détection de la plaque :	62
3.2. Segmentation des caractères:	64
4. Interfaces :	66
4.1. Interface d'authentification :	66
4.2. Interface d'accueil :	66
5. Conclusion :	69

Chapitre 4 : Tests et Résultats

1. Introduction.....	70
2. Résultats des tests de détection et d'extraction des plaques d'immatriculation :	70
3. Résultats des tests du module Segmentation :	75
4. Tests de détection et segmentation d'autres plaques de différents pays :	76
5. Conclusion :	82

Conclusion Générale

Liste des figures

Figure 1: Représentation de la lettre A comme un groupe de pixels	2
Figure 2 : Représentation de l'espace RGB (Red Green Blue)	3
Figure 3 : Cube présentant les couleurs	4
Figure 4 : Exemple de transformation de l'espace RGB à HSV	5
Figure 5 : La représentation d'une image binaire par pixels	6
Figure 6 : Calcul de la valeur médiane d'un quartier de pixels.	8
Figure 7 : Avant et après l'application du filtre médian [8].....	9
Figure 8: Matrices représentant les directions x et y	10
Figure 9 : Les étapes de Canny [9]	11
Figure 10: Masques de Sobel et Prewitt.....	12
Figure 11 : La norme du Gradient	12
Figure 12 : L'angle d'orientation du gradient	13
Figure 13 : Exemples des filtres Sobel et Prewitt.....	13
Figure 14 : La distribution gaussienne.....	14
Figure 15 : 1-D distribution gaussienne de moyenne 0 et $\sigma=1$	14
Figure 16 : Le produit de deux fonctions Gaussiennes 1D	15
Figure 17 : 2-D distributions gaussiennes de moyenne (0,0) et $\sigma=1$	15
Figure 18 : Exemple de l'utilisation du filtre gaussien [9].....	15
Figure 19 : Laplacien L (x, y) d'une image	16
Figure 20 : Noyaux utilisé approximations discrètes au filtre Laplacien.	16
Figure 21 : La fonction de log 2-D centrée sur zéro et avec écart type gaussienne	17
Figure 22 : La 2-D Laplacien de la fonction gaussienne (log).....	18
Figure 23 : Approximation discrète pour se connecter.....	18
Figure 24 : Fonction de binarisation	20
Figure 25 : Une image en niveaux de gris 6-niveau et son histogramme.	20
Figure 26 : La valeur seuil arrière-plan.....	21
Figure 27 : Les calculs pour trouver les écarts d'arrière-plan.....	21
Figure 28 : La valeur seuil premier plan	21
Figure 29 : Les calculs pour trouver les écarts de premier plan.....	22
Figure 30 : Calcule de la variance intra-classe	22
Figure 31 : Tableau représentant les résultats de ces calculs	23
Figure 32 : Résultat final de la méthode Otsu ave un seuil égale à 3.	23
Figure 33 : Un ensemble B de pixels" élément structurant "	24
Figure 34 : Exemple de la dilatation [7]	25
Figure 35 : Exemple de l'érosion [7]	26
Figure 36 : Exemple de l'ouverture [7].....	26
Figure 37 : Exemple de la fermeture [7]	27

Figure 38 : Amincissement binaire. (a) : image de départ, (b) un squelette curviligne de (a), (c) un squelette ultime de (a).....	28
Figure 39 : Composants Connectés	29
Figure 40 : Déroulement de l'algorithme Mean shift.....	31
Figure 41 : Séparation d'un ensemble de points en K clusters	32
Figure 42 : Définition de la mesure de distorsion	32
Figure 43 : Projection Horizontale [7]	34
Figure 44 : Projection Verticale [7]	35
Figure 45 : Perceptron a trois entrées, x_1 , x_2 , x_3	37
Figure 46 : Perceptron de 2 couches	37
Figure 47 : Type de fenêtre du descripteur de Haar-Like [31].....	38
Figure 48 : Différentes utilisations du descripteur de Haar-Like. [32].....	39
Figure 49 : Organigramme Globale	49
Figure 50 : Description du module "Détection de la plaque"	50
Figure 51 : Description du module "Segmentation des caractères"	53
Figure 52 : Description du module "Reconnaissance des caractères"	55
Figure 53 : Ajouter une Bibliothèque	60
Figure 54 : Créer une bibliothèque	60
Figure 55 : Personnaliser le chemin du fichier	61
Figure 56 : L'ajout de la bibliothèque.....	61
Figure 57 : Application du filtre gaussien.....	62
Figure 58 : Application du filtre Sobel	62
Figure 59 : Utilisation du seuil adaptatif gaussien.....	62
Figure 60 : Redimensionnement des images	62
Figure 61 : L'utilisation de la Dilatation.....	63
Figure 62 : L'utilisation de la fermeture	63
Figure 63 : Liste des Contours.....	63
Figure 64 : Fonction de vérification de l'aspect ratio	63
Figure 65 : Redimensionnement de l'image de la plaque d'immatriculation	64
Figure 66 : Conversion de l'image au niveau de gris	64
Figure 67 : Utilisation du filtre gaussien.....	64
Figure 68 : L'utilisation du seuil adaptatif gaussien pour l'image de la plaque.....	64
Figure 69 : L'utilisation de la fermeture	64
Figure 70 : Application de la Dilatation.....	65
Figure 71 : Trouver les différents contours	65
Figure 72 : Vérification des contours qui représente des caractères	65
Figure 73 : Interface d'authentification	66
Figure 74 : Interface d'accueil.....	67
Figure 75 : Menu Fichier.....	67
Figure 76 : Choisir une image	68
Figure 77 : Affichage de l'image choisie	68
Figure 78 : Résultats de segmentation	69

Figure 79 : Image d'entrée	70
Figure 80 : Image au niveau de gris.....	71
Figure 81 : Image après filtre Sobel.....	71
Figure 82 : Image au seuil adaptatif.....	72
Figure 83 : Image résultante des opérations morphologiques	72
Figure 84 : Image final avec les rectangles trouvés en surbrillance en vert	73
Figure 85 : Image résultante des opérations morphologiques après redimensionnement de l'image	74
Figure 86 : Image brute avec plaque d'immatriculation localisée et détectée en rouge .	74
Figure 87 : Extraction de la plaque d'immatriculation	74
Figure 88 : Image de la plaque d'immatriculation après le seuil adaptatif	75
Figure 89 : Image de la plaque d'immatriculation résultante des opérations morphologiques	75
Figure 90 : Image de la plaque d'immatriculation après segmentation, avec chaque caractère encadré en bleu	75
Figure 91 : Résultat de la Segmentation	75
Figure 92 : Détection de la plaque	76
Figure 93 : Segmentation des caractères par des rectangles bleus	76
Figure 94 : Résultat de la Segmentation des caractères	76
Figure 95 : Détection de la plaque	77
Figure 96 : Segmentation des caractères par des rectangles bleus	77
Figure 97 : Résultat de la Segmentation des caractères	77
Figure 98 : Détection de la plaque	78
Figure 99 : Segmentation des caractères par des rectangles bleus	78
Figure 100 : Résultat de la Segmentation des caractères	78
Figure 101 : Détection de la plaque	79
Figure 102 : Segmentation des caractères par des rectangles bleus	79
Figure 103 : Résultat de la Segmentation des caractères	79
Figure 104 : Détection.....	80
Figure 105 : Segmentation des caractères par des rectangles bleus	80
Figure 106 : Résultat de la Segmentation des caractères	80
Figure 107 : Détection de la plaque	81
Figure 108 : Segmentation des caractères par des rectangles bleus	81
Figure 109 : Résultat de la Segmentation des caractères	81

***Introduction
Générale***

Introduction Générale :

La reconnaissance automatique des plaques d'immatriculation des véhicules devient l'intérêt des chercheurs dans de nombreuses branches des technologies d'application, y compris les Systèmes de sécurités et les systèmes de surveillances.

Ce système devient crucial dans notre vie quotidienne en raison de l'augmentation illimitée des véhicules et des systèmes de transport qui rend impossible la gestion et la surveillance avec des opérations manuelles. Parmi les applications qui exigent réellement le système de reconnaissance sont la surveillance du trafic, le suivi des voitures volées, la gestion du péage de stationnement, l'application de la violation des feux rouges, l'entrée des parkings, l'entrée des bâtiments et les points de contrôle douaniers.

La reconnaissance automatique des plaques d'immatriculation est un système qui capture l'image des véhicules et extrait automatiquement la plaque afin de convertir chaque caractère trouvé dans l'image en caractère codé par la machine ; en procédant, selon 4 étapes principales :

- 1- L'acquisition de l'image
- 2- La détection et localisation de la plaque
- 3- La segmentation des caractères
- 4- La reconnaissance des caractères

Dans les pays développés, les attributs des plaques d'immatriculation sont strictement respectés, par exemple la taille de la plaque, la couleurs de la plaque, les polices de caractère l'espacement entre les caractères ... sont maintenus très précisément. En Algérie, les plaques d'immatriculation ne sont pas encore normalisées, ce qui rend extrêmement difficile la localisation et la reconnaissance des plaques d'immatriculation.

L'état de l'art comporte une grande panoplie d'approche de détection et de classification de plaque. Dans ce domaine, la localisation de la plaque se fait d'abord en utilisant des opérations de bas niveau en traitement d'images tel que l'élimination de bruit par filtrage [40], les détections de contour, la binarisation de caractères, ainsi que les opérations morphologiques [41], pour l'amélioration des caractères extraits.

Dans un deuxième temps, l'extraction et reconnaissance de caractères est faite en combinant des approches de bas niveau de traitement d'images ainsi que des approches de haut niveau de vision par ordinateur [42]. Dans ce contexte, des combinaisons entre les approches citées précédemment pour la cluterisation du caractère et des approches basées sur l'apprentissage machine ou machine learning en anglais [43] pour la reconnaissance du caractère sont réalisées. Ainsi, la reconnaissance basée sur l'apprentissage machine offre un ensemble d'algorithmes pour l'entraînement de caractère et leur reconnaissances. Nous citons à titre d'exemple le classifieur machine a vecteurs de supports ou support vector machine SVM en anglais [44], les KNN [45], les réseaux de neurones [46] et récemment l'apprentissage approfondi ou deep learning en anglais [47].

L'objectif de ce travail consiste à utiliser les connaissances issues du domaine de traitement d'images, de vision par ordinateur et d'apprentissage machine afin de concevoir et implémenter un système robuste de détection et de reconnaissance des plaques d'immatriculation à partir d'images.

La robustesse du système se mesurera selon son taux de reconnaissance de plaque d'immatriculation à titre d'exemple le changement d'illumination, la différence de résolution de la région de l'image contenant la plaque d'immatriculation et la présence du bruit.

Pour assurer une meilleure présentation du travail effectué et garantir la clarté du mémoire, outre cette introduction générale, ce manuscrit se compose de quatre chapitres. Chacun met en évidence une contribution particulière du travail :

- Dans le premier chapitre de ce mémoire, nous présentons les notions de base et les différentes méthodes et techniques utilisées pour la manipulation des images ;
- Le deuxième chapitre est consacré à l'étude conceptuelle de notre solution. Nous présentons les différents organigrammes et algorithmes utilisés ;
- Le troisième chapitre décrit l'implémentation de notre solution réalisée. Nous retrouvons la présentation des outils et des langages utilisés dans la programmation ainsi que le paramétrage des différentes fonctions, nous présentons aussi quelques interfaces de notre application ;
- Dans le quatrième chapitre, nous évaluons la performance du système proposés.

Chapitre 1 :
Notions de base

1 Introduction :

Le traitement d'images et ses techniques associés sont destinés à exploiter au mieux l'information contenue dans une image.

Dans ce chapitre nous allons parler de quelques concepts concernant l'image et ses différents types, nous présentons aussi quelques techniques utilisés pour la manipulation et le traitement des images en citant les différents filtres et méthodes appliqués.

Ces techniques sont développées et mise en œuvre dans plusieurs travaux de reconnaissance de plaques d'immatriculation, dont nous citons quelques-uns à la fin de ce chapitre.

2 Définition d'une image :

Une image est une représentation plane d'une scène ou d'un objet situé en générale dans un espace tridimensionnel .Son élaboration résulte de la volonté de proposer une entité observable par l'œil humain. Ceci explique d'une part son aspect plane et d'autre part le fait que l'information élémentaire associée à chaque point de l'image soit transcrite au niveau de gris ou en couleur. [4]

Afin de pouvoir rendre une image exploitable par l'ordinateur, et la considérer comme une donnée informatique ; il est nécessaire de lui faire subir des transformations permettant sa numérisation, c'est-à-dire la convertir d'une image analogique en une image numérique.

2.1 Images Analogique:

En mathématique est un signal 2D Continu, même si l'image correspond à une scène 3D. [5] Donc l'image analogique est décrite par une fonction bidimensionnelle représentant l'intensité lumineuse à un point donné.

2.2 Images Numérique:

Sont des clichés électroniques d'une scène ou numérisés à partir de documents tels que photographies, manuscrits, textes imprimés et œuvres d'art.

L'image numérique est échantillonnée et mappée comme une grille de points ou éléments d'images (picture element ou pixel). A chaque pixel correspond une valeur tonale (noir, blanc, niveaux de gris ou couleurs), 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. [1]

2.3 Pixel:

Le nom de 'pixel', abrégé px, provient de l'expression anglaise 'Picture element', qui signifie 'élément d'image' ou 'point élémentaire', c'est une entité calculable qui peut recevoir une structure et une quantification. Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression.

La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure suivante :

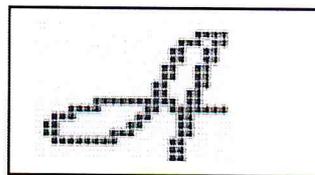


Figure 1: Représentation de la lettre A comme un groupe de pixels

La quantité d'information que véhicule chaque pixel donne des nuances entre images monochromes et images couleurs. [6]

3 Espace de représentation d'une image :

La représentation des images fixes est l'un des éléments essentiels des applications multimédias. Dans cette partie nous allons détailler les différents espaces de représentations des images.

3.1 Espace RGB:

L'espace RVB (Rouge Vert Bleu) ou RGB (Red Green Blue) est basé sur la synthèse additive. C'est l'espace le plus couramment utilisé pour représenter les couleurs. La télévision, les caméras et les moniteurs des ordinateurs utilisent ce système lors de l'affichage et de l'acquisition des images. Chaque couleur est représentée par 3 valeurs qui sont les quantités de rouge, de vert et de bleu qu'elle contient. Les valeurs sont comprises entre 0 et 1 (valeurs normalisées), ou 0 et 255 lors du codage des couleurs sur 8 bits. Par exemple : $(0,0,0)$ = noir $(1,1,1)$ = blanc $(1,0,0)$ = rouge $(0,1,0)$ = vert $(0,0,1)$ = bleu $(0,1,1)$ = cyan $(1,0,1)$ = magenta $(1,1,0)$ = jaune On représente souvent l'espace RVB comme un cube de dimensions $1 \times 1 \times 1$, placé à l'origine d'un repère à trois dimensions dont les axes indiquent la quantité de Rouge, de Vert et de Bleu.

Chaque couleur correspond à un point C de coordonnées (R_c, V_c, B_c) situé dans le cube. Les couleurs Rouge, Vert, Bleu, Cyan, Magenta, Jaune, Noir et Blanc sont situées aux sommets du cube.[2]

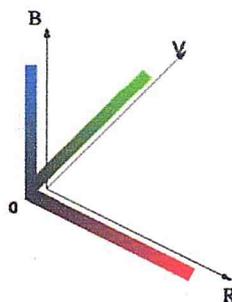


Figure 2 : Représentation de l'espace RGB (Red Green Blue)

Le Noir $(0,0,0)$ est situé à l'origine du repère.

Les couleurs primaires sont situées sur les 3 axes à la distance 1.

La diagonale principale du cube est une droite tracée entre l'origine $(0,0,0)$ et le point $(1,1,1)$. Les couleurs situées sur cette ligne ont des quantités égales de Rouge, de Vert, et de Bleu. Ce sont donc les différents niveaux de gris, variant du Noir en $(0,0,0)$ au Blanc en $(1,1,1)$. Le choix des 3 couleurs primaires R,V,B détermine les couleurs présentes dans le cube.

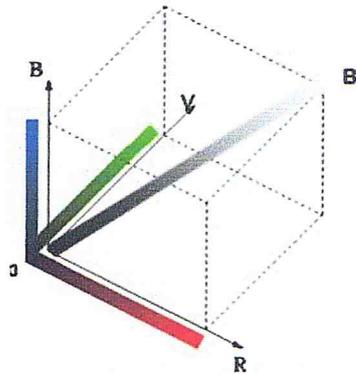


Figure 3 : Cube présentant les couleurs

3.2 Espace HSV :

Le principe de l'espace HSV est de caractériser les couleurs de façon plus intuitive, conformément à la perception naturelle des couleurs, en termes de :

- 1- **teinte** : intuitivement, c'est le nom qu'on utilisera pour désigner la couleur, "vert", "mauve", "orange", etc. Idéalement associé à une longueur d'onde, donc à une position sur le cercle de Newton.
- 2- **saturation** : c'est le taux de pureté de la couleur, qui doit varier entre la pureté maximale (couleur éclatante) et l'achromatisme (niveau de gris).
- 3- **valeur** : c'est la mesure de l'intensité lumineuse de la couleur, qui doit varier entre le noir absolu et le blanc.

Un point x de coordonnées (r,g,b) du cube RGB peut être décomposé en deux composantes (c,a) . Soit y la projection orthogonale de x sur l'axe achromatique.

- Une composante chromatique c : c'est le vecteur yx .
- Une composante achromatique a : c'est le vecteur Oy

Cette décomposition permet d'associer les composantes HSV à des grandeurs géométriques :

3.4 Image en niveaux de gris :

Est composée de pixels possédant plusieurs bits d'informations, allant en général de 2 à 8 bits, ou davantage.

Exemple : dans une image à 2 bits il existe 4 combinaisons possibles : 00, 01, 10 et 11. Si "00" représente le noir et "11" le blanc, alors "01" correspond au gris foncé et "10" au gris clair.

La profondeur de bits est de 2, mais le nombre de teintes pouvant être représentées est 2^2 , soit 4. A 8 bits, 256 valeurs (2^8) peuvent être assignées à chaque pixel. [1]

3.5 Image Binaire :

Est représentée par des pixels de 1 bit chacun, pouvant représenter deux teintes (d'habitude le noir et le blanc), en utilisant la valeur 0 pour le noir et 1 pour le blanc.

1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	1	0	1	1	1
1	1	0	1	1	1	0	1	1	1
1	1	0	1	1	1	0	1	1	1
1	1	0	0	0	0	0	0	1	1
1	1	0	1	1	1	0	1	1	1
1	1	0	1	1	1	0	1	1	1
1	1	0	1	1	1	0	1	1	1
1	0	0	0	1	1	0	0	0	1
1	1	1	1	1	1	1	1	1	1

Figure 5 : La représentation d'une image binaire par pixels

Ce point sera abordé dans la prochaine section avec la notion « Binarisation »

Le traitement d'image se déroule avec plusieurs opérations et techniques qu'on va les détaillées après cela. [1]

4 Opérations sur l'image :

Après avoir cité différentes notions concernant les images, la manipulation de celles-ci nécessite l'utilisation de diverses méthodes ; nous détaillons dans ce qui suit ces méthodes l'une après l'autre.

4.1 Filtrage de l'image:

Le filtrage d'une image consiste à faire passer une image par un filtre afin le plus souvent de réduire le bruit présent dans l'image. Le filtre médian et le filtre moyen sont utilisés pour réduire le bruit dans une image. Cependant, le filtre médian est souvent considéré meilleur que le filtre moyen pour la préservation des détails utiles de l'image. De ce fait, nous abordons dans ce qui suit spécialement le filtre médian tout en faisant référence au filtre médian lorsque des similarités sont observées. [7]

4.1.1 Processus:

Comme le filtre moyen, le filtre médian considère chaque pixel dans l'image et le compare à ses proches voisins pour décider s'il est ou non représentatif de son environnement. Au lieu de simplement remplacer la valeur de pixel par la moyenne des valeurs de pixels voisines comme il est le cas du filtre moyen, sa valeur est remplacée par la médiane de ces voisins. La médiane est calculée en triant d'abord toutes les valeurs de pixels du voisinage en ordre numérique, puis en remplaçant le pixel étant considéré avec la valeur de pixel intermédiaire. (Si le voisinage considéré contient un nombre pair de pixels, on utilise la moyenne des deux valeurs de pixels intermédiaires).

4.1.2 Exemple de calcul :

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Figure 6 : Calcul de la valeur médiane d'un quartier de pixels.

La figure 6 présente un exemple de filtrage par filtre médian. Comme on peut le voir sur la figure 1, la valeur du pixel centrale de 150 est peu représentative des pixels environnants. Celle-ci sera remplacée dans le filtre médian par la valeur médiane: 124 calculée à partir des valeurs : 124, 126, 127, 120, 125, 115, 119 et 123. Un quartier carré de 3×3 est utilisé ici. Des quartiers importants produiront un lissage plus sévère.

En calculant la valeur médiane d'un voisinage plutôt que le filtre moyen, le filtre médian présente deux avantages principaux par rapport au filtre moyen: la médiane est une moyenne plus forte que la moyenne et donc un seul pixel très peu représentatif dans un quartier n'affectera pas significativement la valeur médiane. Étant donné que la valeur médiane doit effectivement être la valeur d'un des pixels du voisinage, le filtre médian ne crée pas de nouvelles valeurs de pixel irréalistes lorsque le filtre chevauche un bord. Pour cette raison, le filtre médian est beaucoup mieux pour préserver les La figure 7 présente le résultat du filtre médian.

être considérée comme fausse. Cependant, la forme mathématique choisie pour le premier critère ne met pas en évidence cette condition : elle ne mesure que la capacité du filtre à distinguer le signal du bruit au centre du contour. Elle ne prend pas en compte le comportement du filtre dans son voisinage. Les auteurs choisissent donc d'écarter au maximum la distance entre les maxima de la réponse du filtre au bruit, pour limiter le nombre de réponses multiples lorsqu'il y a un unique contour réel. [11].

L'opérateur Canny travaille dans un processus en plusieurs étapes :

- 1) Filtrer le bruit : Lisser l'image par un filtre gaussien. L'augmentation de la largeur du noyau gaussien réduit la sensibilité du détecteur au bruit au détriment de perdre certains des détails les plus fins dans l'image. [12]
- 2) Calculer la magnitude du gradient en utilisant l'approximation des dérivées partielles. Pour cela ; il faut suivre une procédure analogue à Sobel.
 - a) Applique une paire de masques de convolution (dans les directions x et y)

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

Figure 8: Matrices représentant les directions x et y

- b) Trouver la magnitude de gradient et la direction avec

$$M(m, n) = \sqrt{G_x^2(m, n) + G_y^2(m, n)}$$

$$\theta = \tan^{-1}(G_y(m, n), G_x(m, n))$$

La direction est arrondie à l'un des quatre angles possibles (à savoir 0, 45, 90 ou 135)

- 3) Appliquer la suppression non-maximale : Cela permet de supprimer les pixels qui ne sont pas considérés comme faisant partie d'un bord. Par conséquent, seules les lignes minces (bords candidats) resteront. Pour ce faire, vérifier si chaque non zéro $M(m, n)$ est supérieur à ses deux voisins le long de la direction du gradient $\theta(m, n)$. Dans ce cas garder $M(m, n)$ inchangé, sinon le remplacer par 0. [10]

- 4) Hystérèse : l'utilisation de deux seuils t_1 et t_2 (supérieur et inférieur) (ou $t_1 < t_2$) pour obtenir deux images binaires T_1 et T_2 . [12]

Plus le seuil supérieur est élevé et le seuil inférieur est faible plus le résultat est bon

- a. Si une valeur de gradient de pixel est supérieure à t_2 , le pixel est reconnu comme un bord fort alors il est remplacé par 1 (ou 255).
- b. Si une valeur de gradient de pixel est en dessous de t_1 , le pixel sera rejeté.
- c. Si une valeur de gradient de pixel est entre les deux seuils, alors il sera considéré comme étant un bord faible et il ne sera accepté que s'il est connecté à un pixel qui est au- dessus du seuil supérieur. [15]

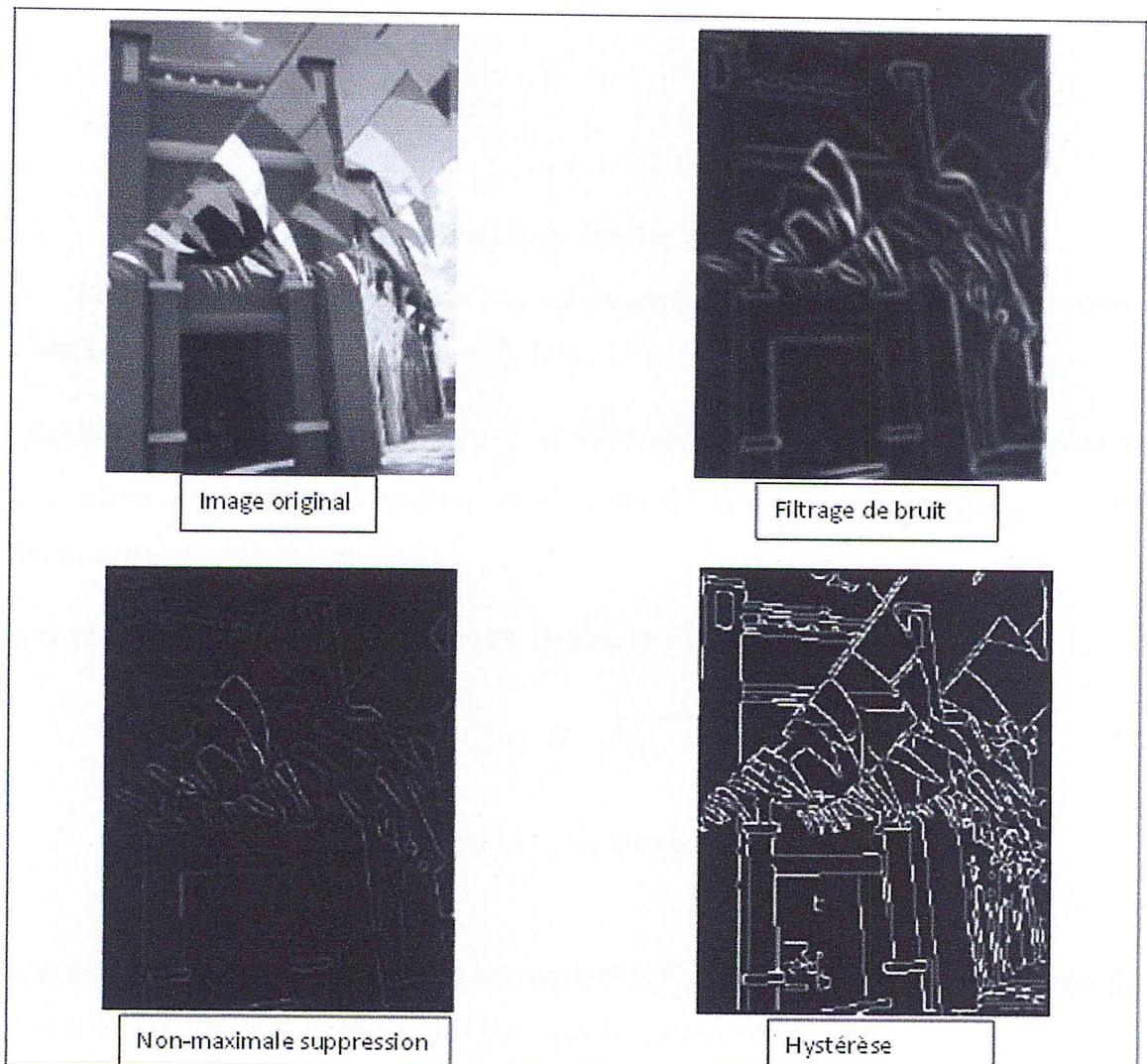


Figure 9 : Les étapes de Canny [9]

4.2.3 Filtre Gaussien:

Le filtrage gaussien est utilisé pour flouter les images et supprimer le bruit et les détails, il est effectué par convolution de chaque point de la matrice d'entrée avec un noyau gaussien, puis de les additionner pour produire tout le tableau de sortie. [17]

La distribution gaussienne en 1-D a la forme :[18]

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

Figure 14 : La distribution gaussienne

Où σ est l'écart - type de la distribution, la distribution est supposée avoir une moyenne de zéro (c'est-à-dire elle est centrée sur la ligne $x = 0$). Illustrée sur la figure 14 on voit la répartition gaussienne en forme de cloche :

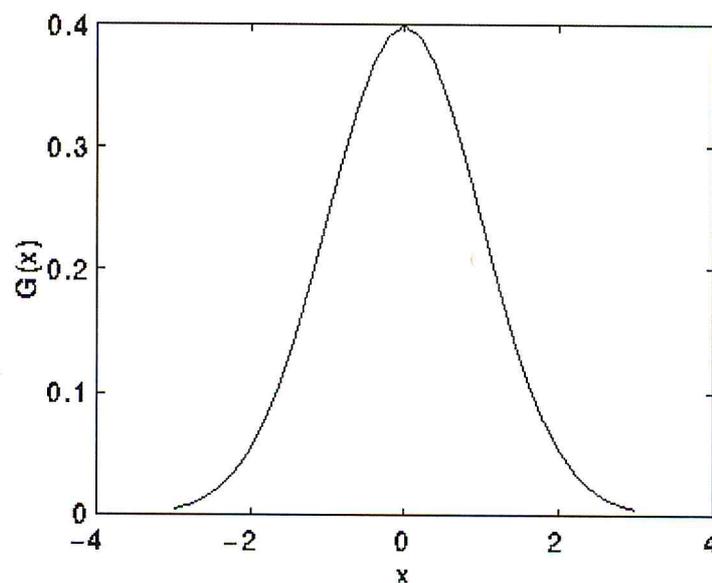


Figure 15 : 1-D distribution gaussienne de moyenne 0 et $\sigma=1$

Puisque nous travaillons avec des images, nous devons utiliser la fonction Gaussienne bidimensionnelle. Ceci est simplement le produit de deux fonctions Gaussiennes 1D (une pour chaque direction) et est donnée par :

4.2.4 Filtre Laplacien :

Laplacien est un 2-D isotrope (est celle qui applique aussi bien dans toutes les directions dans une image, sans sensibilité particulière ou polarisation vers un ensemble particulier de directions) mesure de la deuxième dérivée spatiale d'une image. Laplacien d'une image met en évidence les régions de changement d'intensité rapide et est donc souvent utilisé pour la détection de bord (voir détecteurs de front de passage par zéro). Le Laplacien est souvent appliquée à une image qui a d'abord été lissée avec quelque chose se rapprochant d'un filtre de lissage gaussien afin de réduire sa sensibilité au bruit, et par conséquent les deux variantes seront décrites ensemble ici. L'opérateur prend normalement une seule image de niveau de gris en entrée et produit une autre image en niveau de gris de sortie.

Laplacien $L(x, y)$ d'une image avec des valeurs d'intensité de pixels $I(x, y)$ est donnée par:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Figure 19 : Laplacien $L(x, y)$ d'une image

Cela peut être calculé à l'aide d'un filtre à convolution .

Étant donné que l'image d'entrée est représentée comme un ensemble de pixels discrets, nous devons trouver un noyau de convolution discrète qui peut rapprocher les dérivées secondes dans la définition du Laplacien. Deux petits noyaux couramment utilisés sont présentés dans la figure 20.

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

Figure 20 : Noyaux utilisé approximations discrètes au filtre Laplacien.

L'utilisation d'un de ces noyaux, le Laplacien peut être calculée en utilisant des méthodes de convolution standard.

Dans ce travail on va passer par plusieurs étapes pour atteindre nos objectifs parmi ces étapes la segmentation, segmenter une image en couleur peut être extrêmement coûteux. Traiter une image en couleurs de taille moyenne revient à caractériser quelques 500 000 points (800×600), et trouver la classe (ou le type) d'un point nécessite bien souvent la prise en compte de nombreux facteurs (l'histogramme, la couleur. . .). Afin de simplifier les traitements, de nombreuses recherches se sont orientées vers la binarisation de l'image. Binariser une image revient à segmenter l'image

4.3 Binarisation d'une Image :

La binarisation est souvent la première étape dans les systèmes de traitement et d'analyse d'images. Elle a pour but de diminuer la quantité d'informations présentes dans l'image, et de ne garder que les informations pertinentes. Les performances des étapes suivantes dans les systèmes de traitement et d'analyse d'images dépendent fortement du résultat obtenu lors de la binarisation. C'est pour cette raison que la méthode de binarisation utilisée doit être la plus adaptée possible. Elle doit, d'une part, conserver le maximum d'informations utiles présentes dans l'image de départ, et d'autre part, éliminer un maximum de bruit présent dans l'image.

Ces dernières décennies, un très grand nombre de techniques de binarisation ont été proposées avec chacune d'entre elles des caractéristiques différentes. Il est donc important de bien connaître l'image que l'on souhaite binariser afin d'obtenir le résultat souhaité.

La binarisation appelée aussi seuillage, est la technique de classification la plus simple. Les pixels de l'image sont partagés par un seuil T en deux classes. En général, ils sont représentés par une classe de pixels noirs et une autre classe de pixels blancs. L'image est alors séparée en deux classes, une classe représentant le fond de l'image et une autre classe représentant la scène de l'image (L'objet). La binarisation permet alors de conserver l'information comprise entre 0 et T ou entre $T+1$ et 255. Le reste des informations est par conséquent ignoré.

4.2.3 Filtre Gaussien:

Le filtrage gaussien est utilisé pour flouter les images et supprimer le bruit et les détails, il est effectué par convolution de chaque point de la matrice d'entrée avec un noyau gaussien, puis de les additionner pour produire tout le tableau de sortie. [17]

La distribution gaussienne en 1-D a la forme :[18]

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

Figure 14 : La distribution gaussienne

Où σ est l'écart - type de la distribution, la distribution est supposée avoir une moyenne de zéro (c'est-à-dire elle est centrée sur la ligne $x = 0$). Illustrée sur la figure 14 on voit la répartition gaussienne en forme de cloche :

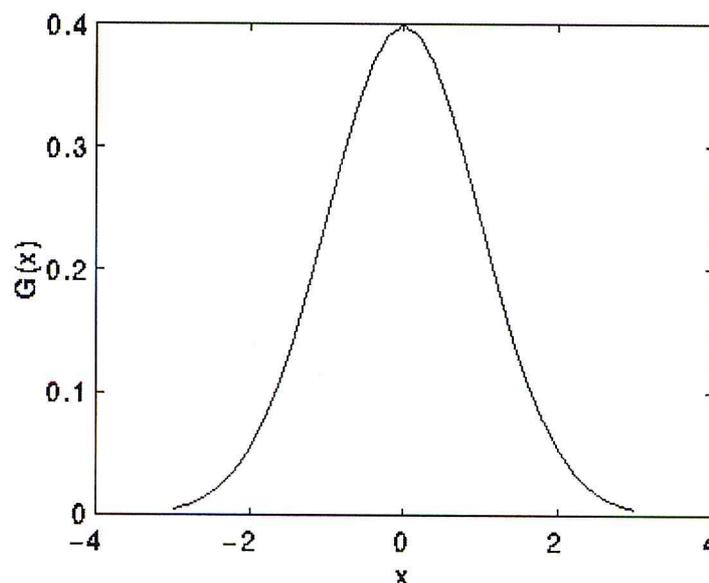


Figure 15 : 1-D distribution gaussienne de moyenne 0 et $\sigma=1$

Puisque nous travaillons avec des images, nous devons utiliser la fonction Gaussienne bidimensionnelle. Ceci est simplement le produit de deux fonctions Gaussiennes 1D (une pour chaque direction) et est donnée par :

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Figure 16 : Le produit de deux fonctions Gaussiennes 1D

Une représentation graphique de la distribution gaussienne 2D avec une moyenne (0,0) et $\sigma = 1$ est représentée sur la figure 16.

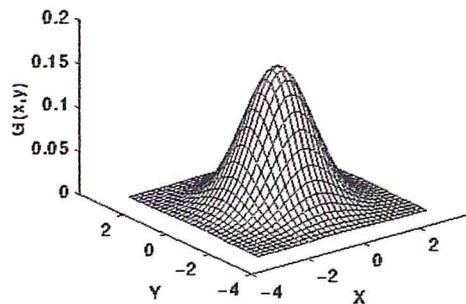


Figure 17 : 2-D distributions gaussiennes de moyenne (0,0) et $\sigma=1$

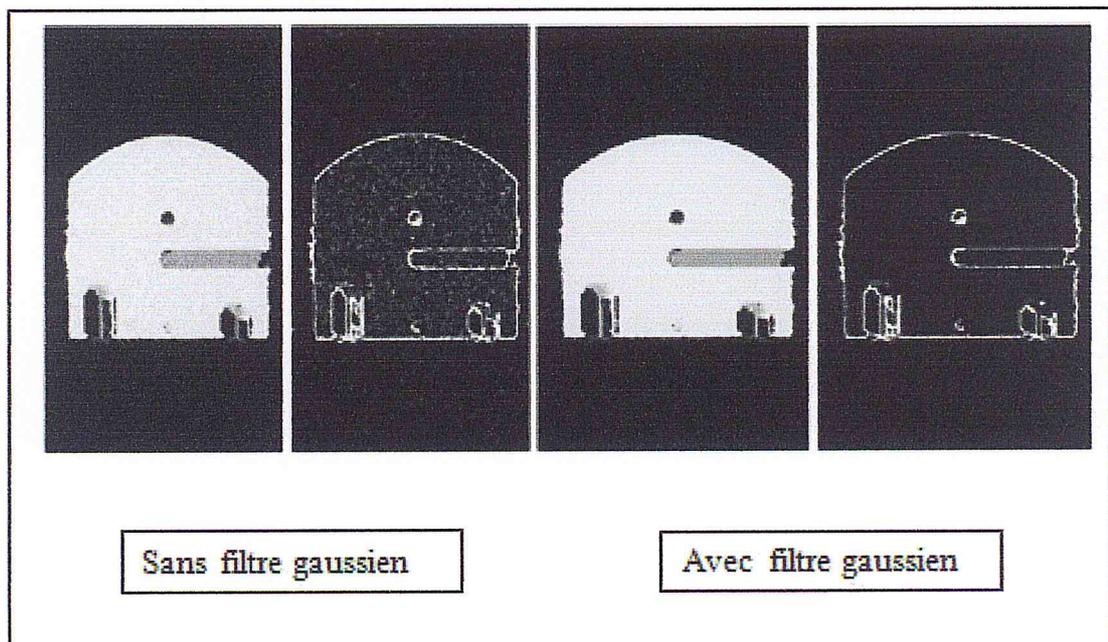


Figure 18 : Exemple de l'utilisation du filtre gaussien [9]

4.2.4 Filtre Laplacien :

Laplacien est un 2-D isotrope (est celle qui applique aussi bien dans toutes les directions dans une image, sans sensibilité particulière ou polarisation vers un ensemble particulier de directions) mesure de la deuxième dérivée spatiale d'une image. Laplacien d'une image met en évidence les régions de changement d'intensité rapide et est donc souvent utilisé pour la détection de bord (voir détecteurs de front de passage par zéro). Le Laplacien est souvent appliquée à une image qui a d'abord été lissée avec quelque chose se rapprochant d'un filtre de lissage gaussien afin de réduire sa sensibilité au bruit, et par conséquent les deux variantes seront décrites ensemble ici. L'opérateur prend normalement une seule image de niveau de gris en entrée et produit une autre image en niveau de gris de sortie.

Laplacien $L(x, y)$ d'une image avec des valeurs d'intensité de pixels $I(x, y)$ est donnée par:

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

Figure 19 : Laplacien $L(x, y)$ d'une image

Cela peut être calculé à l'aide d'un filtre à convolution .

Étant donné que l'image d'entrée est représentée comme un ensemble de pixels discrets, nous devons trouver un noyau de convolution discrète qui peut rapprocher les dérivées secondes dans la définition du Laplacien. Deux petits noyaux couramment utilisés sont présentés dans la figure 20.

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

Figure 20 : Noyaux utilisé approximations discrètes au filtre Laplacien.

L'utilisation d'un de ces noyaux, le Laplacien peut être calculée en utilisant des méthodes de convolution standard.

Parce que ces noyaux sont proches d'une seconde mesure sur l'image dérivée, ils sont très sensibles au bruit. Pour contrer cela, l'image est souvent gaussienne lissée avant d'appliquer le filtre Laplacien. Cette étape de pré-traitement permet de réduire les composantes de bruit à haute fréquence avant l'étape de différenciation.

En fait, étant donné que l'opération de convolution est associative, nous pouvons convoluer le filtre de lissage gaussienne avec le filtre Laplacien d'abord, et puis convoluer ce filtre hybride avec l'image pour obtenir le résultat nécessaire. Faire les choses de cette façon a deux avantages:

Étant donné que la gaussienne et les noyaux sont Laplacien généralement beaucoup plus petits que l'image, cette méthode nécessite généralement des opérations arithmétiques beaucoup moins.

Le rondin (Laplacien de gaussienne) noyau peut être précalculée à l'avance pour une seule convolution doit être effectuée au moment de l'exécution sur l'image.

La fonction de log 2-D centrée sur zéro et avec écart type gaussienne σ est de la forme:

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Figure 21 : La fonction de log 2-D centrée sur zéro et avec écart type gaussienne

et est représentée sur la figure 22.

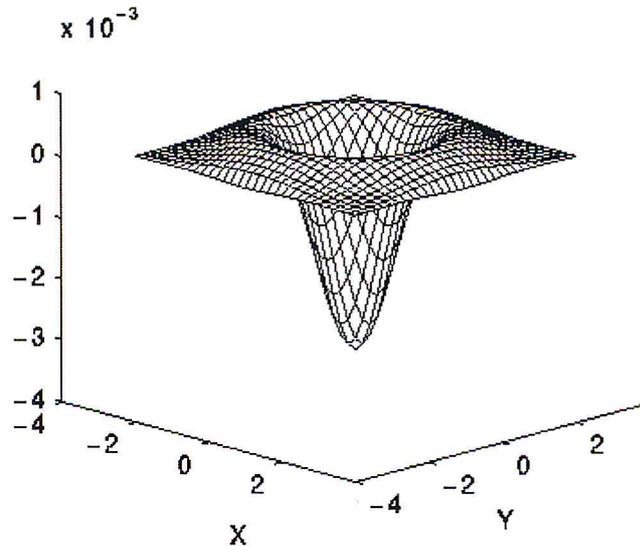


Figure 22 : La 2-D Laplacien de la fonction gaussienne (log)

Un noyau discret qui se rapproche de cette fonction (pour une gaussienne σ est représentée sur la Figure 23 = 1,4).

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

Figure 23 : Approximation discrète pour se connecter avec la fonction de Gauss $\sigma = 1,4$

Notez que lorsque le noyau de LoG gaussienne est fait de plus en plus étroite, devient la même que les noyaux simples Laplacien représenté sur la figure 1. En effet, le lissage avec une gaussienne (très étroite $\sigma < 0,5$ Les pixels) sur une grille discrète n'a aucun effet. Par conséquent sur une grille discrète, le simple Laplacien peut être considérée comme un cas limite de la grume pour gaussiennes étroites. [16]

Dans ce travail on va passer par plusieurs étapes pour atteindre nos objectifs parmi ces étapes la segmentation, segmenter une image en couleur peut être extrêmement coûteux. Traiter une image en couleurs de taille moyenne revient à caractériser quelques 500 000 points (800×600), et trouver la classe (ou le type) d'un point nécessite bien souvent la prise en compte de nombreux facteurs (l'histogramme, la couleur. . .). Afin de simplifier les traitements, de nombreuses recherches se sont orientées vers la binarisation de l'image. Binariser une image revient à segmenter l'image

4.3 Binarisation d'une Image :

La binarisation est souvent la première étape dans les systèmes de traitement et d'analyse d'images. Elle a pour but de diminuer la quantité d'informations présentes dans l'image, et de ne garder que les informations pertinentes. Les performances des étapes suivantes dans les systèmes de traitement et d'analyse d'images dépendent fortement du résultat obtenu lors de la binarisation. C'est pour cette raison que la méthode de binarisation utilisée doit être la plus adaptée possible. Elle doit, d'une part, conserver le maximum d'informations utiles présentes dans l'image de départ, et d'autre part, éliminer un maximum de bruit présent dans l'image.

Ces dernières décennies, un très grand nombre de techniques de binarisation ont été proposées avec chacune d'entre elles des caractéristiques différentes. Il est donc important de bien connaître l'image que l'on souhaite binariser afin d'obtenir le résultat souhaité.

La binarisation appelée aussi seuillage, est la technique de classification la plus simple. Les pixels de l'image sont partagés par un seuil T en deux classes. En général, ils sont représentés par une classe de pixels noirs et une autre classe de pixels blancs. L'image est alors séparée en deux classes, une classe représentant le fond de l'image et une autre classe représentant la scène de l'image (L'objet). La binarisation permet alors de conserver l'information comprise entre 0 et T ou entre $T+1$ et 255. Le reste des informations est par conséquent ignoré.

$$B(x, y) = \begin{cases} \text{Si } I(x, y) < T \text{ Alors } B(x, y) = 0 \\ \text{Sinon } B(x, y) = 1 \end{cases}$$

Figure 24 : Fonction de binarisation

La binarisation se fait par plusieurs méthodes, nous présentons dans ce qui suit l'une de ces méthodes.

4.3.1 La méthode d'OTSU :

La méthode d'Otsu, du nom de son inventeur Nobuyuki Otsu, est l'un des nombreux algorithmes utilisés pour la binarisation. Le principe consiste à itérer à travers toutes les valeurs de seuil possibles, et calculer une mesure de dispersion pour les niveaux de pixel de chaque côté du seuil, à savoir les pixels du premier plan ou d'arrière-plan.

L'objectif est de trouver la valeur seuil où la somme des écarts de premier plan et d'arrière-plan est à son minimum.

L'algorithme est démontré en utilisant une image simple 6x6 ci-dessous. L'histogramme de l'image est affiché à côté. Pour simplifier l'explication, seulement 6 niveaux sont utilisés en échelle de gris.

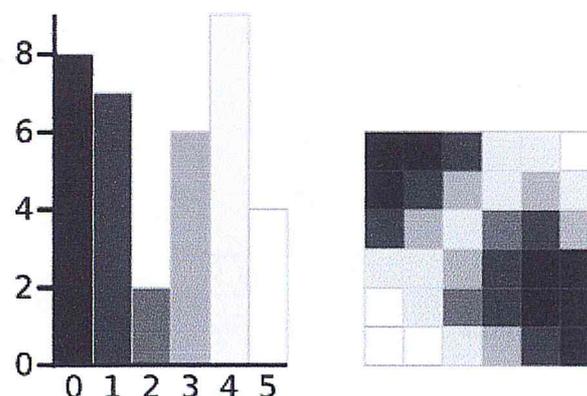


Figure 25 : Une image en niveaux de gris 6-niveau et son histogramme.

$$\begin{aligned}
 \text{Weight } W_f &= \frac{6 + 9 + 4}{36} = 0.5278 \\
 \text{Mean } \mu_f &= \frac{(3 \times 6) + (4 \times 9) + (5 \times 4)}{19} = 3.8947 \\
 \text{Variance } \sigma_f^2 &= \frac{((3 - 3.8947)^2 \times 6) + ((4 - 3.8947)^2 \times 9) + ((5 - 3.8947)^2 \times 4)}{19} \\
 &= \frac{(4.8033 \times 6) + (0.0997 \times 9) + (4.8864 \times 4)}{19} \\
 &= 0.5152
 \end{aligned}$$

Figure 29 : Les calculs pour trouver les écarts de premier plan

L'étape suivante consiste à calculer la « variance intra-classe ». Ceci est simplement la somme des deux variances multipliées par leurs poids associés :

$$\begin{aligned}
 \text{Within Class Variance } \sigma_W^2 &= W_b \sigma_b^2 + W_f \sigma_f^2 = 0.4722 * 0.4637 + 0.5278 * 0.5152 \\
 &= 0.4909
 \end{aligned}$$

Figure 30 : Calcul de la variance intra-classe

Cette valeur finale est la « somme des écarts pondérés » pour la valeur de seuil 3. Ce même calcul a été effectué pour toutes les valeurs de seuil possible entre 0 et 5. [20]

Le tableau ci-dessous présente les résultats de ces calculs :

Seuil	T=0	T=1	T=2	T=3	T=4	T=5
Poids, arrière-plan	$W_b = 0$	$W_b = 0.222$	$W_b = 0.4167$	$W_b = 0.4722$	$W_b = 0.6389$	$W_b = 0.8889$
Moyenne, Fond	$M_b = 0$	$M_b = 0$	$M_b = 0.4667$	$M_b = 0.6471$	$M_b = 1.2609$	$M_b = 2.0313$
Écart, Fond	$\sigma_b^2 = 0$	$\sigma_b^2 = 0$	$\sigma_b^2 = 0.2489$	$\sigma_b^2 = 0.4637$	$\sigma_b^2 = 1.4102$	$\sigma_b^2 = 2.5303$
Poids, Premier plan	$W_f = 1$	$W_f = 0.7778$	$W_f = 0.5833$	$W_f = 0.5278$	$W_f = 0.3611$	$W_f = 0.1111$
Moyenne, Premier plan	$M_f = 2.3611$	$M_f = 3.0357$	$M_f = 3.7143$	$M_f = 3.8947$	$M_f = 4.3077$	$M_f = 5.000$
Écart, Premier plan	$\sigma_f^2 = 3.1196$	$\sigma_f^2 = 1.9639$	$\sigma_f^2 = 0.7755$	$\sigma_f^2 = 0.5152$	$\sigma_f^2 = 0.2130$	$\sigma_f^2 = 0$
Variation de la classe	$\sigma_W^2 = 3.1196$	$\sigma_W^2 = 1.5268$	$\sigma_W^2 = 0.5561$	$\sigma_W^2 = 0.4909$	$\sigma_W^2 = 0.9779$	$\sigma_W^2 = 2.2491$

Figure 31 : Tableau représentant les résultats de ces calculs

Nous remarquons que le seuil égal à 3, à la somme la plus faible des écarts pondérés. Ce seuil est donc le dernier sélectionné. Tous les pixels avec un niveau inférieur à 3 sont des arrières plans, et ceux ayant un niveau égal ou supérieur à 3 sont des premiers plans. [20]

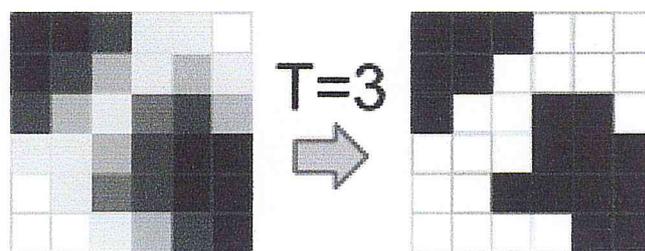


Figure 32 : Résultat final de la méthode Otsu avec un seuil égale à 3.

Dans l'image on peut trouver des structures qui ne respectant pas certains critères, alors on va parler des opérations morphologique qui nous permettre de les éliminer.

4.4 Opération Morphologique :

Les opérations morphologiques sont des filtres non-linéaires qui peuvent s'appliquer tant aux images binaires qu'à celles à niveaux de gris. Contrairement à la plupart des mises en œuvre pratiques de filtres, la fenêtre $W(p)$ associée à un pixel p n'est pas nécessairement de forme rectangulaire ou circulaire. Au contraire, le choix de sa forme dépend du but recherché, plus particulièrement du type de structure qu'on veut mettre en évidence dans l'image. On choisit donc un ensemble B de pixels, appelé élément structurant, et la fenêtre $W(p)$ associée à un pixel p sera le translaté de B positionné sur p , noté B_p . [21]

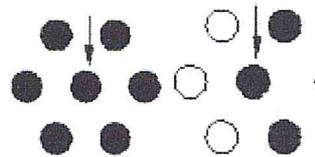


Figure 33 : Un ensemble B de pixels" élément structurant "

- : Pixel devant appartenir à l'image initiale.
 - : Pixel ne devant pas appartenir à l'image initiale.
 - . : Pixel indifférent.
- ↓ : Origine.

On note \tilde{B} la transposée de B (symétrie par rapport à l'origine). [22]

Afin d'arranger les formes détectés par la binarisation ou les formes d'une image au niveau de gris nous utilisons les opérations dilatation, érosion, ouverture et fermeture.

4.4.1 La dilatation:

Pour chaque position de B sur l'image X , si un, au moins, des pixels de B fait partie de X , alors l'origine de B appartient à l'image générée. [22]

Notation : $Y = X \oplus B$

L'effet de la dilatation est d'abord d'élargir la figure, la hauteur et largeur de la figure dilatée seront les sommes respectivement des hauteurs et largeurs de la figure originelle et de l'élément structurant. Si l'élément structurant est décentré, la dilatation décalera la figure dans le même sens. Enfin les coins convexes de la figure seront déformés en fonction de l'élément structurant (par exemple si celui-ci est un disque, les coins convexes seront arrondis). [21]

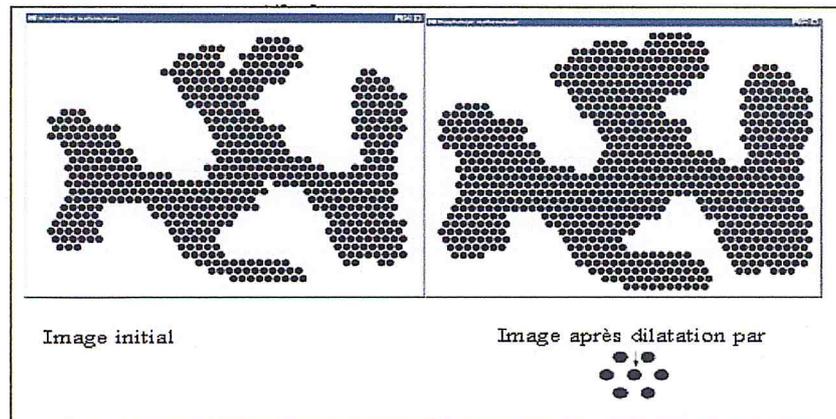


Figure 34 : Exemple de la dilatation [7]

4.4.2 L'Erosion :

Pour chaque position de B sur l'image X, si tous les pixels de B font partie de X, alors l'origine de B appartient à l'image générée.

Notation : $Y = X \ominus B$. [22]

L'effet de l'érosion est d'abord de rétrécir la figure, la hauteur et largeur de la figure érodée seront les différences respectivement des hauteurs et largeurs de la figure originelle et de l'élément structurant (en particulier si l'élément structurant est plus large ou plus haut que la figure, l'érosion de celle-ci sera vide). Si l'élément structurant est décentré, l'érosion décalera la figure en sens inverse.

Enfin les coins concaves de la figure seront déformés en fonction de l'élément structurant (par exemple si celui-ci est un disque, les coins concaves seront arrondis). [21]

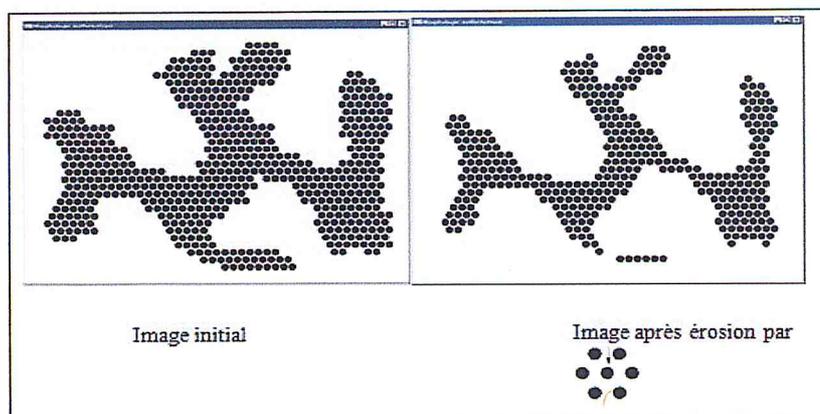


Figure 35 : Exemple de l'érosion [7]

4.4.3 L'Ouverture:

On définit l'ouverture par B comme la composition de l'érosion par B suivie de la dilatation par la transposée de B :

$$Y = X_B = (X \ominus B) \oplus \tilde{B}$$

Le but de l'ouverture est d'isoler les surfaces présentes dans l'image, lisser les contours.[22]

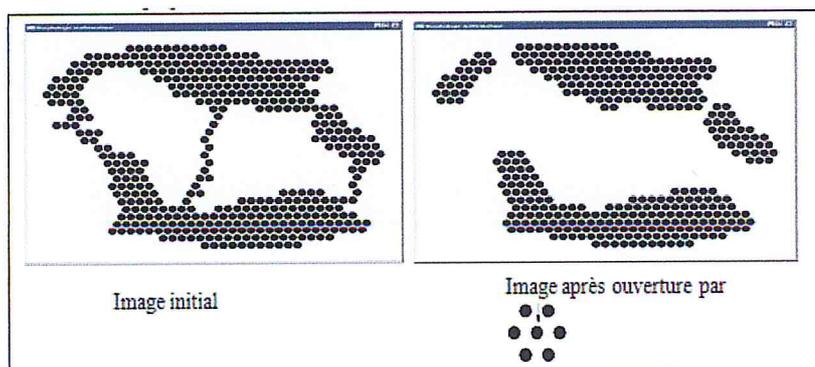


Figure 36 : Exemple de l'ouverture [7]

4.4.4 La fermeture :

On définit la fermeture par B comme la composition de la dilatation par B suivie de l'érosion par la transposée de B

$$Y = X^B = (X \oplus B) \ominus \tilde{B}$$

Le But de la fermeture est de : Recoller des morceaux de surfaces proches de manière à fermer des contours disjoints, lisser les contours. [22]

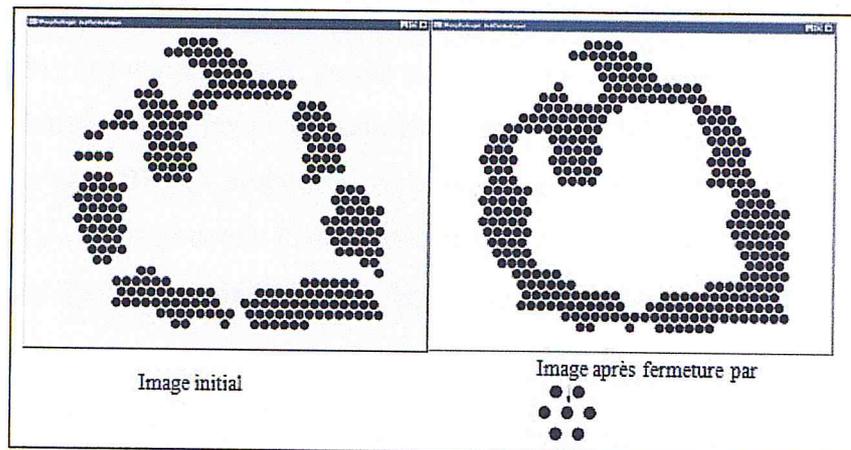


Figure 37 : Exemple de la fermeture [7]

Les deux opérations morphologiques : érosion et ouverture sont des cas particuliers de l'amincissement de l'image.

4.5 Amincissement de l'image:

L'amincissement est un opérateur utilisé en reconnaissance de formes. Un algorithme d'amincissement supprime des points d'une image binaire, en préservant la topologie, pour obtenir un squelette. Pour beaucoup d'applications il est utile que le squelette possède, en plus de l'homotopie avec l'image de départ, des caractéristiques non topologiques : chaque forme allongée dans l'image d'entrée doit être représentée par un segment de courbe. Nous distinguons deux types de squelettes :

Les squelettes curvilignes, constitués seulement de points non simples et d'extrémités de courbes et **les squelettes ultimes**, constitués uniquement de points non simples. La figure 1 montre des exemples de ces types de squelettes.

Il existe un grand nombre d'algorithmes d'amincissement, chacun produisant des squelettes avec des propriétés géométriques spécifiques, souvent liées à une application

Fonctionnement d'Étiquetage en composantes connexes :

Étiquetage en composantes connexes fonctionne par balayage d'une image, pixel par pixel (de haut en bas et de gauche à droite) afin d'identifier des régions connectées de pixels, à *savoir* les régions de pixels adjacents qui partagent le même ensemble de valeurs d'intensité V . (Pour une image binaire $V = \{1\}$, mais dans une image niveau de gris V va prendre une plage de valeurs, par exemple: $V = \{51, 52, 53, \dots, 77, 78, 79, 80\}$.

Travaux d'étiquetage des composants connectés sur images binaires ou images niveau de gris et différentes mesures de connectivité sont possibles. Cependant, pour ce qui suit, nous supposons images binaires d'entrée et δ connectivités. L'opérateur d'étiquetage des composantes connexes balaye l'image en déplaçant le long d'une ligne jusqu'à ce qu'il arrive à un point p (où p désigne le pixel à étiqueter, à tout stade du processus de balayage) pour lequel $V = \{1\}$. Lorsque cela est vrai, il examine les quatre voisins de p qui ont déjà été rencontrés dans l'analyse (c. -à- voisins (i) à la gauche de p , (ii) ci - dessus, et (iii et iv) les deux termes diagonaux supérieurs). Sur la base de ces informations, l'étiquetage des p se produit comme suit:

- Si les quatre voisins sont 0, attribuer une nouvelle étiquette à p , autre
- Si un seul voisin a $V = \{1\}$, attribuer l'étiquette à p , sinon
- Si plus d'un des voisins ont $V = \{1\}$, attribuer l'une des étiquettes à p et faire une note des équivalences.

Après avoir terminé l'analyse, les paires d'étiquettes équivalentes sont classés en classes d'équivalence et une étiquette unique est attribuée à chaque classe. Comme étape finale, un second balayage est effectué à travers l'image, au cours de laquelle chaque étiquette est remplacée par l'étiquette attribuée à ses classes d'équivalence. Pour l'affichage, les étiquettes peuvent être différentes gray Levels ou couleurs. [25]

4.7.2 K-means:

K-means est un algorithme de quantification vectorielle (clustering en anglais). K-means est un algorithme de minimisation alternée qui, étant donné un entier K, va chercher à séparer un ensemble de points en K clusters

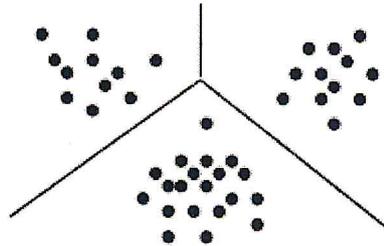


Figure 41 : Séparation d'un ensemble de points en K clusters

Notations, Mesure de distorsion :

On utilise les notations suivantes :

- Les $x_i \in \mathbb{R}^p$, $i \in \{1, \dots, n\}$ sont les points à séparer.
- Les z_i^k sont des variables indicatrices associées aux x_i telles que $z_i^k = 1$ si x_i appartient au cluster k , $z_i^k = 0$ sinon. z est la matrice des z_i^k .
- μ est le vecteur des $\mu_k \in \mathbb{R}^p$, où μ_k est le centre du cluster k . On définit de plus la mesure de distorsion $J(\mu, z)$ par :

$$J(\mu, z) = \sum_{i=1}^n \sum_{k=1}^n z_i^k \|x_i - \mu_k\|^2$$

Figure 42 : Définition de la mesure de distorsion

Algorithme:

Le but de l'algorithme est de minimiser $J(\mu, z)$, il se présente sous la forme d'un algorithme de minimisation alternée :

- Etape 0 : “choisir le vecteur μ ”
- Etape 1 : on minimise J par rapport à z : $z_i^k = 1$ pour $k \in \arg \min |x_i - \mu_k|$, ie on associe à x_i le centre μ_k le plus proche.
- Etape 2 : on minimise J par rapport à μ : $\mu_k = \frac{\sum_i z_i^k x_i}{\sum_i z_i^k}$
- Etape 3 : retour à l'étape 1 jusqu'à convergence.

Convergence et initialisation:

On peut montrer que cet algorithme converge en un nombre fini d'opérations. Cependant la convergence est locale, ce qui pose le problème de l'initialisation.

Une méthode classique consiste à lancer plusieurs fois l'algorithme en prenant les moyennes μ_k aléatoirement à chaque fois, puis on compare leur mesure de distorsion. On choisit la ré-partition qui possède la distorsion minimale. Dans le pire des cas, cet algorithme peut se révéler arbitrairement mauvais, mais dans la pratique, il réalise de très bons résultats.

Choix de K :

Le choix de K n'est pas universel, on remarque que si on augmente K, la distorsion diminue, et s'annule lorsque chaque point est centre de son cluster. Pour pallier à ce phénomène il est possible de rajouter un terme en fonction de K dans l'expression de J, mais là encore son choix est arbitraire. [27]

Afin de partitionner les données des images, on utilise Des descripteurs d'image.

4.8 Descripteurs d'image:

4.8.1 Histogrammes de projection:

Les histogrammes de projection sont généralement utilisés pour détecter l'orientation du texte dans des images binaires, dans le cas des plaques d'immatriculation les histogrammes sont utilisés pour détecter l'inclinaison de la plaque dans leurs images

4.9.1 Réseau de neurones :

La définition la plus simple d'un réseau de neurones, plus correctement appelé un réseau de neurones artificiel ou ANN pour Artificial Neural Network en anglais, est fourni par l'inventeur de l'un des premiers ordinateurs neuronaux, le Dr Robert Hecht-Nielsen. Il définit un réseau de neurones comme:« (...) *un système informatique composé d'un certain nombre d'éléments de traitement simples, fortement interconnectés, qui traitent l'information par leur réponse d'état dynamique d'une source externe* » dans "Neural Network Primaire: Partie I" par Maureen Caudill, AI Expert, février 1989.

Les ANNs sont des dispositifs de traitement (algorithmes ou matériel réel) qui sont vaguement calqués sur la structure neuronale du cortex cérébral Mamalian mais sur des échelles beaucoup plus petites. Un grand ANN pourrait avoir des centaines ou des milliers d'unités de traitement, alors qu'un cerveau Mamalian a des milliards de neurones avec une augmentation correspondante de la grandeur de leur interaction globale et un comportement émergent. Bien que les chercheurs sur les ANN ne se soient généralement pas préoccupés de savoir si leurs réseaux ressemblent avec précision aux systèmes biologiques, certains ont. Par exemple, les chercheurs ont simulé avec précision la fonction de la rétine et modélisé l'œil plutôt bien. [35]

4.9.2 Perceptron:

Un perceptron est un type de neurone artificiel, il a été développé dans les années 1950 et 1960 par le scientifique Frank Rosenblatt, inspiré par le travail de Warren McCulloch et Walter Pitts apparu plus tôt. Un perceptron prend plusieurs entrées binaires, $x_1, x_2, \dots, x_n, x_{n+1}, \dots$, Et produit une sortie binaire unique: y .

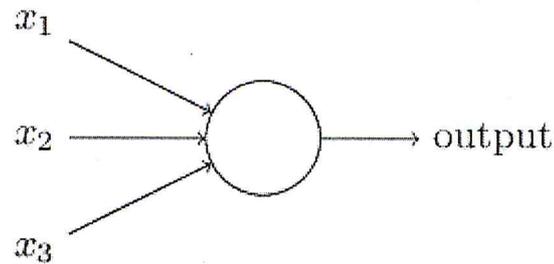


Figure 45 : Perceptron a trois entrées, x_1 , x_2 , x_3 .

L'exemple illustré dans la figure 44 présente un perceptron a trois entrées, x_1 , x_2 , x_3 .

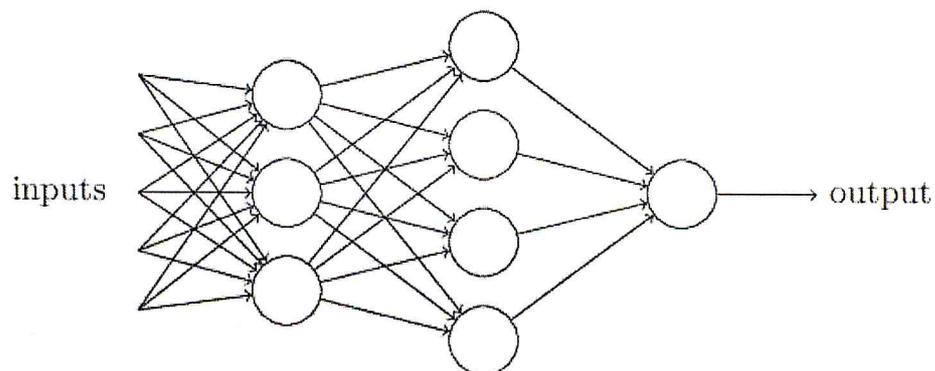


Figure 46 : Perceptron de 2 couches

Dans le réseau la figure 45, la première colonne de perceptrons - ce que nous appellerons la première *couche* de perceptrons - fait trois décisions très simples, en pesant les éléments de preuve d'entrée. Chacun de ces perceptrons de la deuxième couche prend une décision en utilisant les résultats de la première couche pour la prise de décision. De cette façon, un perceptron dans la deuxième couche peut prendre une décision à un niveau plus complexe et plus abstrait que perceptrons dans la première couche. Et de même des décisions plus complexes peuvent être faites par le perceptron dans la troisième couche. De cette façon, un réseau de nombreuses couches de perceptrons peut être engagé dans une prise de décision sophistiquée. [35]

4.9.3 Classificateur en cascade:

La détection d'objets en utilisant des classificateurs en cascade basés sur les caractéristiques de Haar est une méthode efficace proposée par *Paul Viola* et *Micheal*

Jones dans leur document intitulé « *Rapid object detection using a boosted cascade of simple features* » en 2001.[33]

Cette méthode donne une information sur la distribution de niveaux de gris entre deux régions voisines dans une image, les points de Haar consistent en une somme pondérée des pixels situés autour du point d'intérêt.

Ce classificateur est composé d'un ensemble de zones rectangulaires mises à l'échelle d'une taille précise, tel que les pixels situés dans les zones claires sont formé avec une centaine de vues d'échantillons d'un objet particulier (exemple image de plaques), appelés des exemples positifs, et ceux des zones sombres sont des exemples négatifs : des images arbitraires (exemple des images ne contenant pas de plaque).

Pour obtenir la valeur (sortie) d'un filtre appliqué à une région de l'image, la somme des pixels dans le rectangle blanc est soustraite de la somme des pixels dans le rectangle gris [34]. Si la somme excède un seuil fixé θ , le point est positif.

Les descripteurs de Haar sont classifiés généralement en 3 sortes : 2 rectangles, 3 rectangles et 4 rectangles descripteurs (figure 47).

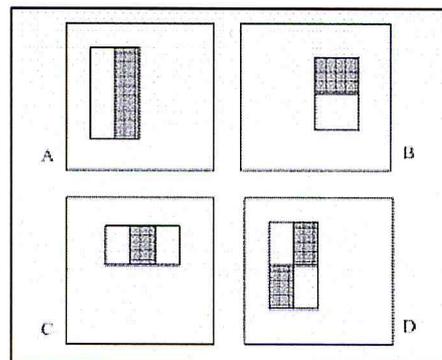


Figure 47 : Type de fenêtre du descripteur de Haar-Like [31]

Ce genre de descripteur est valable pour plusieurs détections d'objet. La figure 48 présente d'autres types de fenêtre pour la détection d'autres formes d'objets.

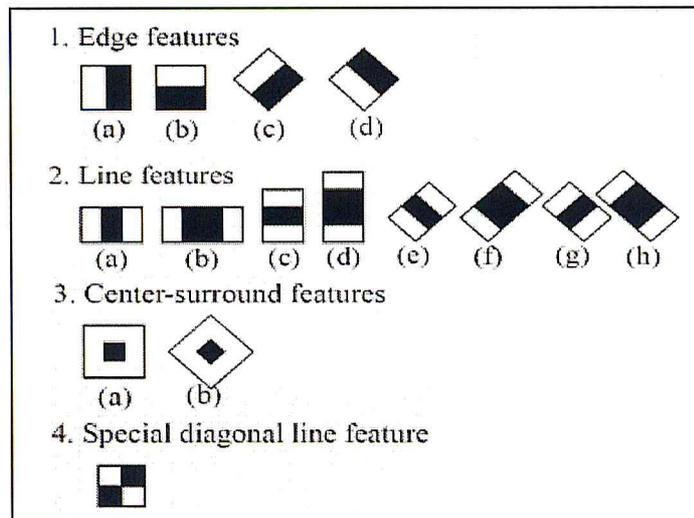


Figure 48 : Différentes utilisations du descripteur de Haar-Like. [32]

5 Travaux liés:

Diverses méthodes sont développées et mises en œuvre pour les processus des systèmes de reconnaissance des plaques d'immatriculations.

Dans ce contexte nous allons citer quelques travaux qui ont contribué au succès du système :

5.1 Détection de la plaque d'immatriculation :

De nombreuses méthodes sont utilisées pour la détermination de la région de la plaque, regroupées généralement en quatre approches principales : approches basées sur le contour, approches basées sur la couleur, approches basées sur la texture et approches basées sur le caractère.

Les méthodes basées sur le contour effectuent généralement des opérations sur des images de niveaux de gris ; elles sont basées sur le principe de la recherche des pixels avec des changements brusques de luminosité, qui correspondent aux frontières des objets dans une image.

Dans leur travail « *İbrahim Türkyılmaz et Kirami Kaçan [48]* », ont adopté l'approche basée sur le contour pour la localisation de la plaque ; développée en 5 étapes : les méthodes basées sur le contour sont affectées par le bruit pour cela ils ont commencé par une amélioration de l'image en éliminant le bruit par un filtre moyen de taille 3*3, ils ont utilisé par la suite la méthode du gradient pour détecter les contours verticaux ; ils ont d'abord convolué l'image par un filtre Sobel de taille 3*3 afin de trouver l'amplitude approximative du gradient pour chaque point de l'image ; ensuite ils ont détecté les points de bords avec seuillage.

Malgré l'utilisation de la procédure d'amélioration d'image antérieure au processus de détection de bord, dans certains cas des bords verticaux en dehors de la région de la plaque ; peuvent être plus intenses, dans de telles situations le système de reconnaissance de plaque d'immatriculation peut détecter de manière incorrecte d'autres régions comme étant une plaque. Pour cette raison, « *İbrahim Türkyılmaz et Kirami Kaçan[48]* », ont procédé à l'élimination de ces bords en utilisant la méthode d'élimination proposée par « *Zheng et al [49]* », cette dernière met comme hypothèse

que la longueur des bords verticaux dans la zone de la plaque change dans la plage de 5 à 28 pixels et que les bords verticaux dont la longueur ne se trouvant pas dans cette plage doivent être éliminées.

Ils ont calculé par la suite les densités de bordures verticales en divisant la matrice de l'image en blocs locaux de taille 16*16 ; le nombre de pixels blancs dans chaque bloc sera stocké dans une nouvelle matrice de taille 30*40. Les points blancs qui ont la valeur la plus élevée dans la matrice seront les points de la région de la plaque, et les autres points blancs indésirables ont été éliminés en posant un seuil.

Pour l'étape de correction d'inclinaison de la plaque, ils ont utilisé la transformée de radon avec détermination de l'angle d'inclinaison θ .

et en fin, la projection horizontale et verticale a permis l'élimination des champs inutiles autres que les caractères de la plaques.

D'autre part « *Sulaiman et al [50]* », ont réalisé un projet de développement et d'intégration du matériel et logiciel.

Notons que pour la partie logiciel, ils ont convertit l'image RVB en image ζ niveaux de gris, puis ils ont amélioré cette dernière en appliquant le filtre Laplacien afin d'éliminer le bruit. Pour la détection de contours, ils ont utilisé un filtre à chapeau mexicain de taille 9*9, par conséquent toute l'image sera recouverte en noir et seulement les bords des objets seront misent en surbrillance en blanc.

L'étape de reconnaissance des caractères était brièvement décrite dans cet article en mentionnant seulement qu'ils ont utilisé un filtre adapté et une technique de reconnaissance optique de caractère.

Leur système a été testé sur 25 véhicules, dans deux conditions : à l'intérieur et à l'extérieur ; sachant que la distance entre la camera et la plaque était d'environ 1.5 à 2 mètres, avec une position de camera aligné ou perpendiculaire. Les résultats à l'intérieur étaient satisfaisants à 60%, quant à l'extérieur les tests étaient réussis à 76%. Ce qui indique que le facteur de lumière joue un rôle essentiel dans l'amélioration de la précision de la reconnaissance des plaques d'immatriculations.

« *Ganesh R.Jadhav et Kailash J.Karande [51]* », ont localisé la plaque d'immatriculation en utilisant un algorithme basé sur la combinaison des caractéristiques de la texture de la plaque et des opérations morphologiques. Il s'agit d'un algorithme de meilleure performance pour une image de plaque d'immatriculation avec un arrière-plan compliqué.

La plaque d'immatriculation se compose de plusieurs bords verticaux, qui regroupent les bordures de la plaque, les caractères et les chiffres. Pour détecter ces bords verticaux, le masque Sobel été appliqué.

Ils ont analysé après l'image résultante en utilisant la décomposition par ondelettes (Wavelet decomposition), par la suite et après avoir convertie l'image en binaire, des opérations morphologiques telles que l'érosion et la dilatation étaient effectuée afin de trouver l'emplacement de la PI.

« *Suproakash Dey et al [52]* », ont réussi à localiser la plaque en utilisant les méthodes morphologiques.

Tout d'abord ils ont procédé au prétraitement de l'image, tout en la convertissant en image à échelle de gris. Puis ils ont supprimé le bruit en utilisant le filtre médian de taille 3*3, ce dernier permet de remplacer la valeur grise d'un pixel par la médiane des valeurs grises de ses voisins.

La localisation morphologique de la plaque présentée dans ce travail, consiste à déterminer en premier l'élément structurant, qui a été présenté sous forme de disque de rayon 1, ensuite ils ont fait la dilatation de l'image suivi par une érosion en utilisant toujours l'élément structurant défini au paravent.

Ils ont amélioré les arêtes en utilisant le gradient morphologique, une conversion de la classe en double ainsi qu'une convolution de l'image ont été faites afin d'éclaircir les bords, l'intensité était mis dans une échelle entre 0 et 1, par la suite ils ont binarisé l'image en sélectionnant un seuil, ce dernier ne permet pas seulement de réduire le bruit mais permet aussi de distinguer clairement entre la cible et le fond conduisant à réduire le temps de calcul.

L'élimination des bords horizontaux indésirable était considéré avant de calculer le centroïde (x,y) de chaque région connecté. Après avoir fait un remplissage de toutes les zones de l'image, « *Suproakash Dey et al [52]* », ont décidé d'éclaircir l'image pour assurer l'isolation des caractères.

Comme dernière étapes, les régions ayant des pixels supérieurs à 100 étaient sélectionnées afin d'obtenir l'emplacement exact de la plaque.

« *Sneha G. Patel [53]* », ont utilisé les méthodes morphologiques pour la localisation de la plaque, en commençant par la conversion de l'image RVB en image à niveaux de gris, ils ont appliqué par la suite le filtre médian afin d'éliminer le bruit. Afin d'extraire de la plaque ils ont utilisé la projection horizontale et verticale, suivi par un lissage des courbes résultantes, ils ont recherché aussi leurs pics afin de trouver le rectangle précis qui inclut la plaque.

En fin l'ouverture et la fermeture ont été utilisées comme méthode de lissage.

« *P. ANISHIYA et al [54]* », ont converti l'image en niveaux de gris à fin de faciliter l'analyse, ils ont procédé aussi à l'élimination du bruit en appliquant le filtre médian.

L'utilisation des méthodes morphologiques pour l'extraction de la région de la plaque dans ce travail, était la suivante : une dilatation avec un élément structurant de taille 3*3, suivi par la sélection des rectangles de taille maximale. Une érosion a été réalisée par la suite avec un élément structurant rectangulaire de taille 10*5. Chaque pixel de la plaque qui touche un pixel d'arrière-plan serait transformé en un pixel d'arrière-plan. Cela rend les objets plus petits. La plaque d'immatriculation serait extraite en soustrayant l'image originale de l'image érodée.

Le critère de couleur pour la reconnaissance des objets par la perception humaine est une caractéristique très puissante, mais les valeurs de couleur dans l'image peuvent montrer des sensibilités différentes en raison de diverses conditions d'éclairage et de la qualité des systèmes d'acquisition d'image. Ces valeurs de couleur représentent différentes caractéristiques dans des conditions différentes ce qui peut affecter négativement les performances du système. L'utilisation d'une méthode basée sur la couleur devient donc très limitée [59]

« *Shi et al. [59]* » ont considéré la transformation du modèle de couleur et ont déterminé la région de plaque en catégorisant les combinaisons de couleurs de plaques d'immatriculations chinoises. [48]

« *Zimic et al. [60]* » ont utilisé la théorie des ensembles flous et ont identifié la zone de la plaque d'immatriculation en identifiant des règles floues fondées sur la perception humaines des couleurs. [48]

Les méthodes de niveaux de gris effectuent des opérations sur des images de niveaux de gris pour la détermination de la région LP. Ces méthodes sont préférables et plus largement utilisées en comparaison avec les méthodes basées sur la couleur, car elles ont une meilleure performance. Cependant, dans certains cas, elles présentent des inconvénients tels qu'une charge de traitement élevée et un temps de calcul long. Ainsi, il est presque impossible d'obtenir des caractères significatifs à partir d'images de haute résolution. [48]

« *Hsieh et al. [58]* », ont tenté de déterminer la région de plaque en identifiant les caractéristiques du contraste sur l'image à l'aide de la transformée Wavelet. [48]

« *Zunino et al. [57]* », ont indiqué qu'il y avait des densités de pixels sombres sur le fond blanc de la région LP et ont essayé d'identifier cette région en utilisant la quantification vectorielle adaptative. [48]

Dans leurs article « *Z.C. Zhang et Y.Y Tang [56]* ». Ont utilisé en entrée une image en échelle de gris, puis ils ont appliqué les opérations morphologiques : érosion et dilatation. Ensuite ils ont réussi à extraire La plaque en utilisant la projection verticale et horizontale. [52]

Dans les travaux de « *P. Kanani et al. [55]* », une décomposition Haar wavelet de l'image en 2-Niveau 2-D est utilisée pour valider les coordonnées de la plaque d'immatriculation dans l'image de la voiture. [52]

5.2 Segmentation des caractères:

La segmentation est le processus de partitionnement de l'ensemble des pixels de l'image en différents groupes. Chaque groupe est supposé correspondre à un "objet" de l'image. [63]

C'est l'étape la plus importante dans les systèmes de reconnaissances des plaques d'immatriculation, car toutes les autres étapes dépendent de son succès, si cette dernière échoue le système obtient des résultats erroné.

Selon « *Sneha G. Patel [53]* », Il existe de nombreux facteurs qui rendent la tâche de segmentation de caractères difficile, comme le bruit de l'image, le cadre de la plaque, le rivet, et la variance de rotation et d'illumination.

Pour segmenter les caractères de la plaque, « *brahim Türkyılmaz et Kirami Kaçan [48]* », ont utilisé, la méthode de projection vertical : tout en réduisant d'abord l'image à échelle de gris en une image en noir et blanc (0 et 1) sur la base d'un seuil. Le succès des opérations de seuillage dépend de la valeur du seuil sélectionné, Afin de trouver la valeur de seuil optimal dans ce travail, la méthode de seuil Otsu été utilisée. Après la binarisation, ils ont effectué un processus d'amélioration de l'image, en utilisant un filtre moyen.

De cette manière, les coordonnées de chaque caractère seront déterminées et les caractères seront séparés les uns des autres à l'aide de ces coordonnées.

D'un autre côté, et afin l'isoler des caractères dans la composante d'image, «*Suproakash Dey et al [52]* » ont utilisé l'algorithme d'Étiquetage de Composant Connecté (CCL), ou Connected Component Analysis (CCA) , qui est une application algorithmique de la théorie des graphes, où des sous-ensembles de composants connectés sont marqués de manière unique sur la base d'une heuristique donnée. Le résultat de l'algorithme CCL est un ensemble de segments d'avant-plan qui sont supposés être les caractères ou les chiffres dans la plaque d'immatriculation.

« *Shi et al. [59]* » ont séparé les caractères les uns des autres en utilisant la projection verticale au niveau binaire. [48]

« *Nomura et al. [61]* » ont proposé une nouvelle méthode morphologique adaptative basée sur l'égalisation de l'histogramme pour la segmentation des caractères, permettant de séparé les caractères les uns des autres recherchant des points de

segmentation naturels dans un histogramme de la projection et combinant ces points avec les parties appropriées du même caractère.[48]

5.3 Reconnaissance des caractères:

Afin de produire des résultats significatifs, les modèles d'entrée traités par la machine sont référés à la reconnaissance de caractères.

Une procédure de reconnaissance de caractères contient les étapes de prétraitement, d'extraction de caractéristiques et de classification. La classification est le processus consistant à identifier chaque caractère et à l'affecter à la classe de caractères appropriée.

« *Suproakash Dey et al [52]* », ont choisi l'adaptation de modèle pour la reconnaissance des caractères, cette dernière est une méthode classique de reconnaissance de formes : les résultats obtenus de la segmentation seront reconnus grâce à une correspondance de modèle. Dans leur modèle proposé, les valeurs de pixels du modèles de caractères (A-Z, 0-9), seront stockés dans un vecteur, tel que l'emplacement 1 stocke la valeur du caractère A, 2 pour B .. Tout d'abord, l'échantillon est classé et ensuite les caractères reconnus sont normalisés par la taille du modèle dans la base de données de caractères et en fin il sera comparé à tous les modèles en calculant la mesure de similarité.

« *Sneha G. Patel [53]* », ont utilisé les réseaux de neurones artificiel dans des application OCR , ce qui a permis considérablement de simplifier le code et d'améliorer la qualité et la performance de reconnaissance ; les points du réseau neuronal ont été ajustés à l'aide d'un algorithme de rétro propagation.

« *J.Cowell et F.Hussain [62]* », Cet article décrit la reconnaissance des caractères latins, leur méthode identifie les caractères en fonction du nombre de lignes et de colonnes de pixels noir du caractère, les valeurs obtenus seront comparer à un ensemble de signature dans la base de données.

Dans le travail de « *İbrahim Türkyılmaz et Kirami Kaçan [48]* », les plaques d'immatriculation étaient reconnues en construisant un réseau de neurones artificiels avancés à trois couches ; utilisant un algorithme d'apprentissage de rétro propagation.

6 Conclusion:

Dans ce chapitre, nous avons défini quelques notions de base sur le traitement d'images, y compris différents filtres, méthodes et opérateurs parmi ces derniers, nous allons citer les méthodes choisies pour la réalisation de notre système.

Nous avons opté l'utilisation des opérations morphologiques pour la détection de la plaque d'immatriculation, étant une méthode efficace avec des résultats satisfaisants :

- Sachant que les images RGB sont très difficiles à exploiter, nous avons converti l'image au niveau de gris afin de faciliter les traitements.
- Le filtre Sobel était utilisé, afin de trouver l'amplitude approximative du gradient pour les différents points de l'image.
- Le but de l'utilisation du seuillage était pour détecter les différents points de l'image.
- La dilatation et fermeture choisies pour faciliter la sélection des rectangles dans l'image binaire.

Pour la partie Segmentation des caractères nous avons utilisé le seuillage et puis l'extraction des composants connectés, étant une méthode efficace simple à implémenter.

Pour l'étape de reconnaissance, le choix d'utilisation d'un classifieur est justifié par résultats des différents travaux cités en état de l'art.

Chapitre 2 :
Conception

1- Introduction

Ce chapitre consiste au développement de l'activité de conception, qui représente une étape importante dans l'élaboration de notre projet de fin d'étude.

Nous allons présenter une vision descriptive des différentes étapes et technique de traitement d'images utilisées pour la détection de la plaque, la segmentation et la reconnaissance des caractères.

2- Conception globale :

2.1 Description :

La détection et la reconnaissance des plaques d'immatriculation est une branche de vision par ordinateurs, qui permet la compréhension d'une scène ou d'un objet à partir d'image ; liant intimement perception, comportement et contrôle.

Après avoir accéder à notre application et choisir une image, et afin d'arriver au résultat demandé, notre système consiste à suivre trois étapes principales, qui sont la détermination de la plaque, la segmentation et la reconnaissance des caractères. Dans chacune de ces étapes l'image subit des traitements qui sont un ensemble d'opérations qui la transforme en d'autres images facile à exploiter.

2.2 Organigramme global :

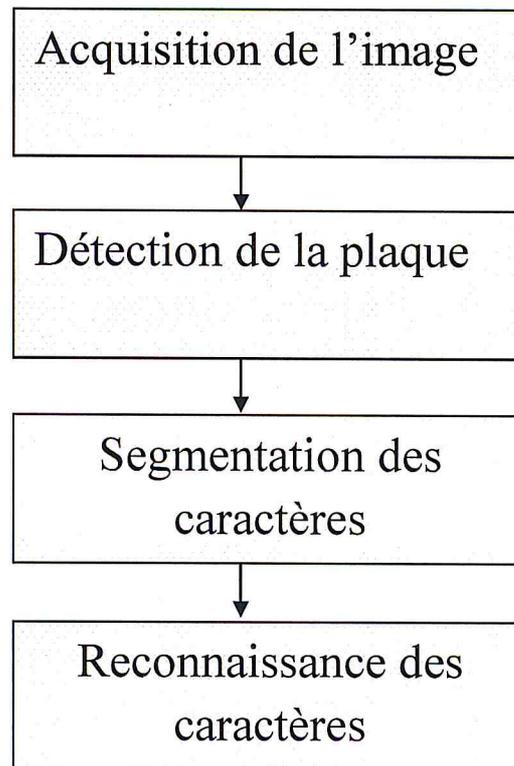


Figure 49 : Organigramme Globale

3. Conception des différents modules :

Sur la base de l'organigramme cité ci-dessus nous allons détailler les étapes les plus importantes:

3.1 Conception du module de détection de la plaque :

A ce stade, notre but est de déterminer la région de la plaque d'immatriculation à partir de l'image d'origine en utilisant des techniques de traitement d'images basées sur des opérations morphologiques. Une image avec un fond complexe contient habituellement des bruits qui conduisent à des résultats indésirables. Pour traiter ce problème et éliminer les champs inutiles nous procédons à l'amélioration de l'image.

3.1.1 Description générale du module de détection de la plaque :

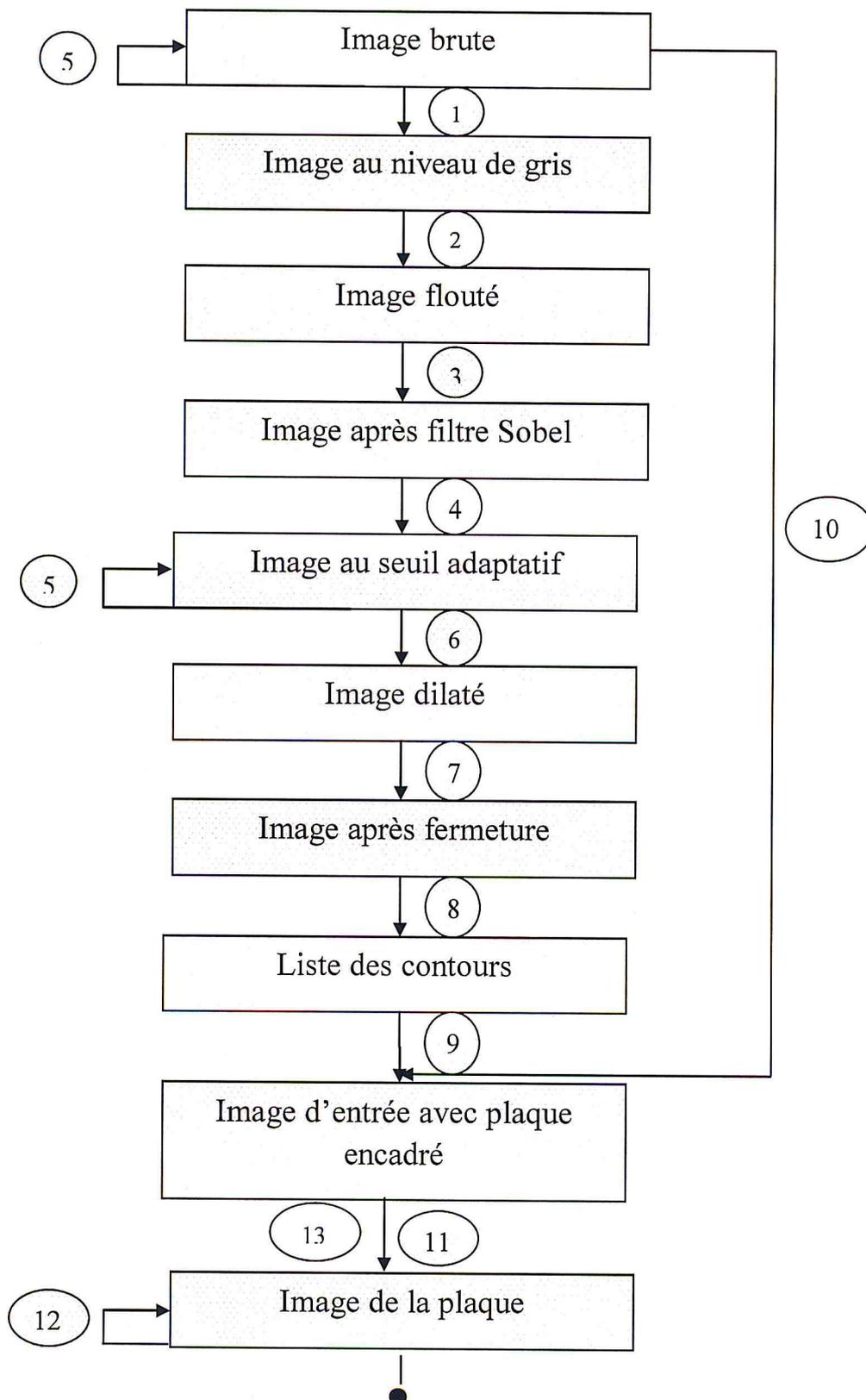


Figure 50 : Description du module "Détection de la plaque"

3.1.2 Algorithme du module de détection de la plaque :

Début

Acquérir une image

- 1 : Convertir l'image au niveau de gris
- 2 : Appliquer filtre gaussien
- 3 : Appliquer le filtre Sobel
- 4 : Utiliser un seuil adaptatif gaussien afin de binariser l'image
- 5 : Redimensionner les images
- 6 : Dilater l'image binaire avec un élément structurant rectangulaire, cette opération morphologique permet de combiner deux ensembles en utilisant l'addition vectorielle de l'élément structurant
- 7 : Fermeture de l'image dilatée avec un élément structurant, la fermeture permet de fusionner les ruptures étroites et d'éliminer les petits trous et de combler les lacunes dans le contour.
- 8 : Chercher les contours dans l'image résultante des opérations morphologiques et les placer dans la liste: « Contours »
- 9 : Si check Ration (d'un élément de la liste) est vrai
- 10 : Encadrer l'élément dans l'image brute en rouge
- 11 : Extraire l'élément comme image de la plaque



12 : Redimensionner l'image de la plaque

13 : Si check Ratio retourne faux encadrer l'élément de la liste en vert dans l'image brute c'est-à-dire région indésirable.

Afficher l'image brute avec les différents contours trouvés

Afficher l'image de la plaque.

Fin

3.2- Conception du module segmentation des caractères :

Segmenter une image, c'est la partitionner en plusieurs région ou ensemble de pixels. La segmentation d'objet est une tâche essentielle dans la vision et la reconnaissance automatiques d'objet.

Puisque le système conçu reconnaitra chaque caractère à part, les caractères situés dans la région de la plaque d'immatriculation doivent être séparés les uns des autre.

La segmentation est tout à fait essentielle à l'extraction des caractères d'images et à la classification ultérieure des caractéristiques résultantes. C'est l'une des étapes les plus importantes dans les systèmes de reconnaissance automatique des plaques d'immatriculation, car toutes les autres étapes dépendent de son succès, si cette dernière échoue le système obtient immédiatement des résultats erronés.

3.2.1-Description générale du module segmentation :

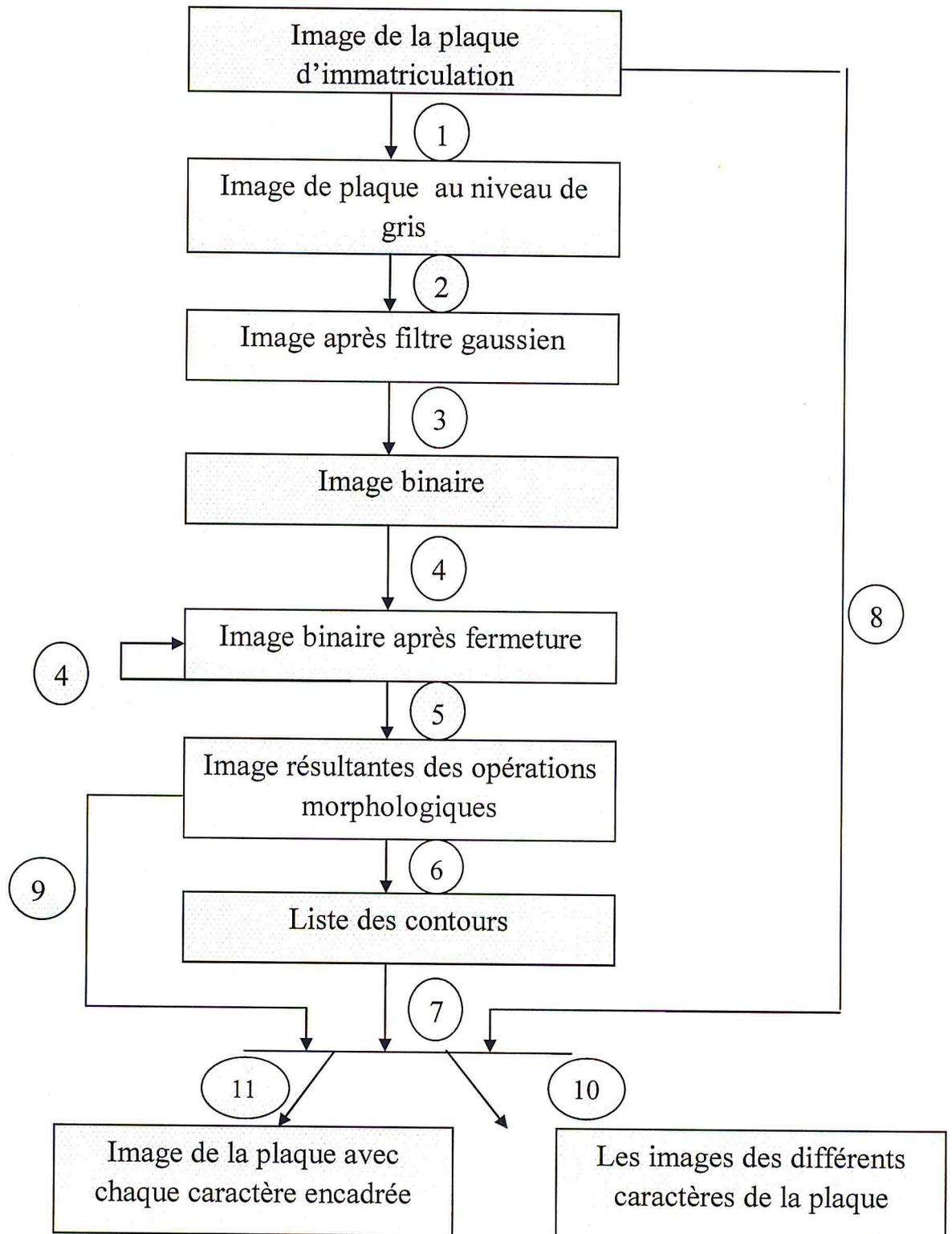


Figure 51 : Description du module "Segmentation des caractères"

3.2.2-Algorithmme :

Récupérer l'image de la plaque d'immatriculation

- 1 : Convertir l'image au niveau de gris
- 2 : Appliquer filtre gaussien
- 3 : Utiliser un seuil adaptatif gaussien afin de binariser l'image
- 4 : Fermeture de l'image binaire avec un élément structurant
- 5 : Dilater l'image résultante avec un élément structurant
- 6 : Chercher les contours dans l'image résultante des opérations morphologiques, et les mettre dans la liste « Contour Caract>>
- 7 : Si ((la longueur d'un élément de la liste est entre 16 et 50) et (sa largeur entre 12 et 50))
- 8 : Encadrer l'élément en bleu dans l'image de la plaque brute.
- 9 : Extraire l'élément de l'image résultante des opérations morphologiques
- 10 : Redimensionner les images des caractères extraits.
Retourne l'image de la plaque les caractères encadrés
- 11 : Retourne l'image de chaque caractère de la plaque.

Fin.

3.3- Conception du module reconnaissance des caractères :

Une procédure de reconnaissance des caractères contient des étapes de prétraitement, d'extraction de caractéristique et de classification. La classification est un processus qui consiste à identifier chaque caractère et à l'affecter à la classe des caractères appropriés.

Les différentes approches de classification, sont utilisées pour la reconnaissance des caractères, lorsque la description du caractère peut être représentée numériquement dans un vecteur de caractéristique. Ainsi, dans notre modèle proposé, la reconnaissance des caractères est effectuée en entraînant un classifieur sur une base d'apprentissage d'images. Afin de pouvoir classifier les caractères il faut qu'ils aient une taille normalisée, c'est pour cette raison que nous avons procédé auparavant au redimensionnement des images des caractères résultantes de la phase "segmentation".

3.3.1- Organigramme de reconnaissance :

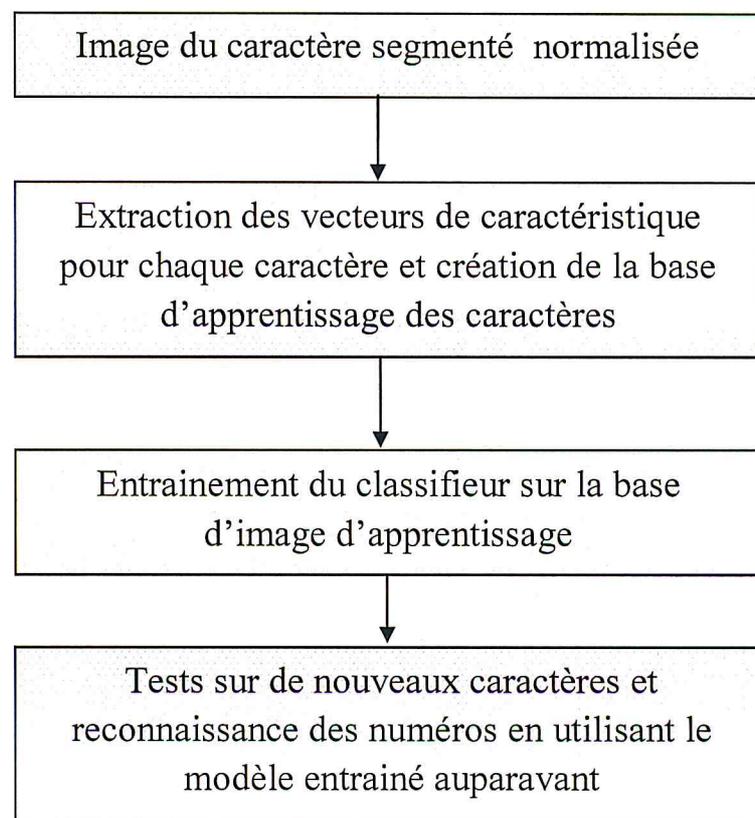


Figure 52 : Description du module "Reconnaissance des caractères"

4- Conclusion :

Dans ce chapitre, nous avons montré en détails l'architecture de notre système, nous avons présenté ainsi la conception de l'application à travers différents organigrammes et algorithmes utilisés pour son développement.

Chapitre 3 :
Implémentation

1. Introduction :

Nous allons présenter dans ce chapitre, les outils de développement que nous avons utilisés pour la réalisation de notre application, cette dernière était programmée en Java, sous Netbeans 8.2 en utilisant la bibliothèque OpenCV 3.2.0.

Nous allons aborder aussi les paramétrages des différentes fonctions et filtres utilisés, et en fin quelques interfaces et scénarios d'utilisations seront présentés afin d'illustrer les différentes fonctionnalités de l'application réalisée.

2. Présentation d'environnement de travail :

2.1. Netbeans :

Netbeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, Groovy, PHP et HTML de façon native ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web). Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java. NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plateforme. [36]



2.2. Java :

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp). [37]



2.3 WampServer :

C'est une plateforme de développement web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL. Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un trayicon (icône près de l'horloge de Windows). [30]



2.4. OpenCV :

Pour ce qui concerne la partie de traitement d'images, nous avons utilisé une bibliothèque OpenCV (Open Source Computer Vision), comme son nom l'indique, OpenCV est une bibliothèque dédiée totalement à la vision par ordinateur, ce qui signifie traitement des images, matrices, vidéos... C'est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de vision par ordinateur, accessibles au travers d'API pour les langages C, C++, et Python. Elle est distribuée sous une



licence BSD (libre) pour les plate-formes Windows, GNU/Linux, Android et MacOS. Initialement écrite en C il y a 10 ans par des chercheurs de la société Intel, OpenCV est aujourd'hui développée, maintenue, documentée et utilisée par une communauté de plus de 40 000 membres actifs. C'est la bibliothèque de référence pour la vision par ordinateur, aussi bien dans le monde de la recherche que celui de l'industrie.[38]

Afin de mieux présenter son étendue et ce qu'elle permet de faire, les principaux modules accessibles au travers de son API C sont présentés dans ce qui suit :

- *Core* : les fonctionnalités de base.

Cette bibliothèque permet de manipuler les structures de base, réaliser des opérations sur des matrices, dessiner sur des images, sauvegarder et charger des données dans des fichiers XML...

- *Imgproc* : traitement d'image.

Nous entrons dans le cœur du sujet. Les fonctions et structures de ce module ont trait aux transformations d'images, au filtrage, à la détection de contours, de points d'intérêt...

- *Features2d* : descripteurs.

Ce module concerne principalement l'extraction de descripteurs selon deux approches courantes (SURF et StarDetector), que nous aborderons lorsque nous nous intéresserons à la caractérisation d'images.

- *Video* : traitement de flux vidéo.

Ces fonctions servent à segmenter et suivre les objets en mouvement dans une vidéo.

- *Highgui* : entrées-sorties et interface utilisateur.

OpenCV intègre sa propre bibliothèque haut-niveau pour ouvrir, enregistrer et afficher des images et des flux vidéo. Celle-ci contient aussi un certain nombre de fonctions permettant de réaliser des interfaces graphiques très simples, mais largement suffisantes pour tester nos programmes.

- *Calib3d* : calibration, estimation de pose et stéréovision.

Ce module contient des fonctions permettant de reconstruire une scène en 3D à partir d'images acquises avec plusieurs caméras simultanément.[38]

- **Configuration de OpenCV3.2.0 :**

Les étapes suivantes montrent la configuration de la version Netbeans8.2 avec la version OpenCV3.2.0.

Tout d'abord, Il faut créer un projet sur Netbeans.

- Etape 1 : Nous avons Sélectionné Librairies du projet Netbeans, après un clic droit sur Librairies un menu contextuel s'affiche, Nous avons choisi "Add Library" :

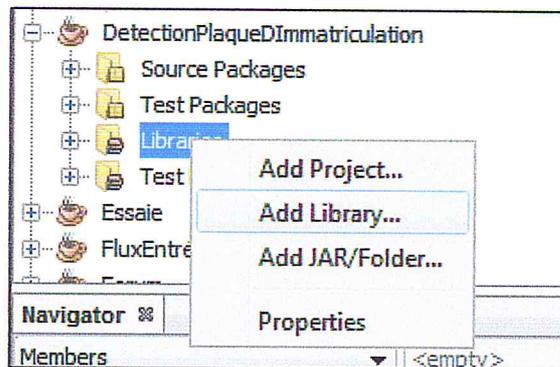


Figure 53 : Ajouter une Bibliothèque

- Etape 2 : Nous avons Cliqué sur le bouton "Créer" pour créer une nouvelle bibliothèque et nous avons écrit le nom de la bibliothèque "OpenCV3.2.0", et nous avons Cliqué sur "OK".

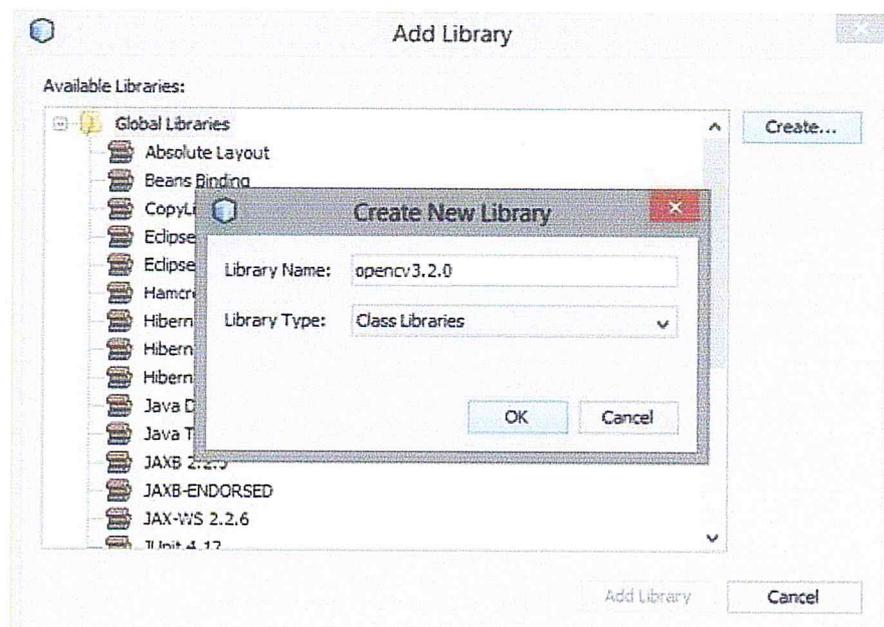


Figure 54 : Créer une bibliothèque

- Etape 3 : Nous avons Sélectionné l'onglet Classpath pour ajouter le chemin du fichier. Nous avons cliqué sur "Add JAR / Folder" pour ajouter le chemin. Après avoir sélectionner le chemin du fichier " c: \opencv \ build \ java \ opencv-320.jar " Nous avons Cliqué sur le bouton "OK".

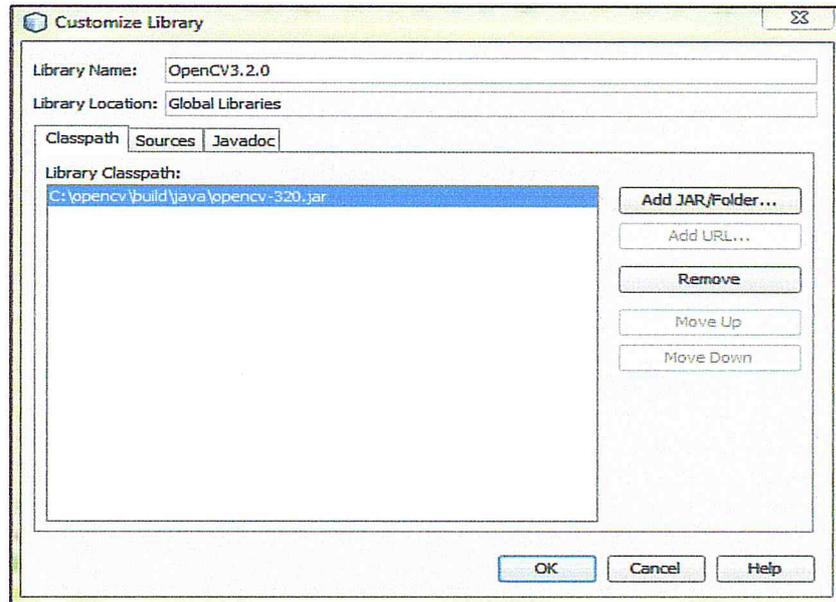


Figure 55 : Personnalisé le chemin du fichier

- Etape 4 : Dans cette étape nous avons sélectionné la bibliothèque configurée "opencv3.2.0", et nous avons cliqué sur le bouton "Add Library" pour ajouter la bibliothèque à l'espace de travail du projet.

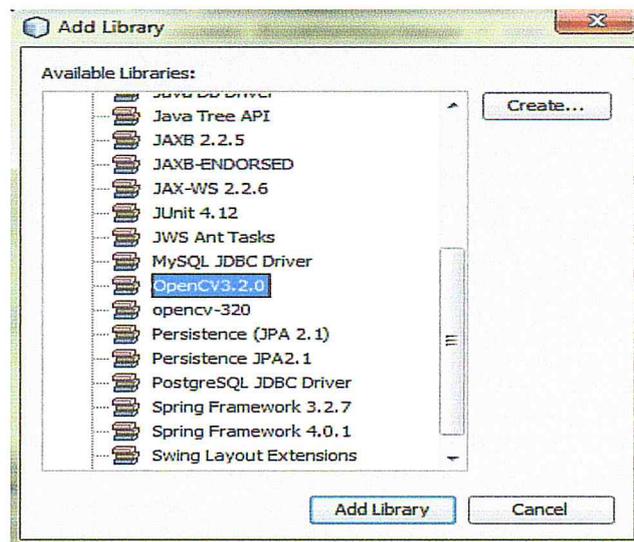


Figure 56 : L'ajout de la bibliothèque

3. Choix et paramétrage des Fonctions :

La plupart des fonctions utilisées sont prédéfinies sur OpenCV3.2.0, mais leurs rendements dépendent des valeurs des paramètres configurés, après plusieurs tests nous avons choisi les paramètres suivants :

3.1. Fonction utilisées pour la détection de la plaque :

Après avoir converti l'image brute en image au niveau nous avons procédé comme suit :

- *Filtre gaussien :*

```
Imgproc.GaussianBlur(imgGray, imgGaussianBlur, new Size(7, 7), 0);
```

Figure 57 : Application du filtre gaussien

- *Filtre Sobel :*

```
Imgproc.Sobel(imgGaussianBlur, imgSobel, -1, 1, 0);
```

Figure 58 : Application du filtre Sobel

- *Seuillage adaptatif gaussien :*

```
Imgproc.adaptiveThreshold(imgSobel, imgAdaptiveThreshold, 255,
    CV_ADAPTIVE_THRESH_GAUSSIAN_C, CV_THRESH_BINARY_INV, 75, 35);
```

Figure 59 : Utilisation du seuil adaptatif gaussien

- *Redimensionnement des Images :*

```
Size s= new Size(295,175);
Imgproc.resize(imgAdaptiveThreshold, imgAdaptiveThreshold, s);
Imgproc.resize(img, img, s);
```

Figure 60 : Redimensionnement des images

- *Opération morphologique : Dilatation*

```
Mat dilate = Imgproc.getStructuringElement(Imgproc.MORPH_RECT,new Size(4, 4));
Imgproc.morphologyEx(imgAdaptiveThreshold, imgMoprhological, Imgproc.MORPH_DILATE, dilate);
```

Figure 61 : L'utilisation de la Dilatation

- *Opération morphologique : Fermeture*

```
Mat element = getStructuringElement(MORPH_RECT, new Size(17, 3));
Imgproc.morphologyEx(imgMoprhological, imgMoprhological, CV_MOP_CLOSE, element);
```

Figure 62 : L'utilisation de la fermeture

- *Chercher les contours :*

```
imgContours = imgMoprhological.clone();
Imgproc.findContours(imgContours,
    contours,
    new Mat(),
    Imgproc.RETR_LIST,
    Imgproc.CHAIN_APPROX_SIMPLE);
```

Figure 63 : Liste des Contours

- *Vérifier l'aspect ratio de la plaque d'immatriculation, Si retourne vrai, extraire la plaque.*

```
public static boolean checkRatio(RotatedRect image) {
    double error = 0.3;
    double aspect = 6;
    int min = (15 * (int)aspect * 15)-900;
    int max = (125 * (int)aspect * 125);
    double rmin= aspect - aspect * error -1.8;
    double rmax= aspect +50 + aspect*error ;
    double area= image.size.height * image.size.width;

    float r= (float)image.size.width / (float)image.size.height;
    if(r<1)
        r= 1/r;
    if(( area < min || area > max ) || ( r < rmin || r > rmax )){
        return false;
    }else{
        return true;
    }
}
```

Figure 64 : Fonction de vérification de l'aspect ratio

- *Après localisation et extraction de la plaque, nous avons procédé au redimensionnement de l'image résultante :*

```
Size s2= new Size(400,70);
Imgproc.resize(imgDetectedPlateCandidate, imgDetectedPlateCandidate, s2);
```

Figure 65 : Redimensionnement de l'image de la plaque d'immatriculation

3.2. Segmentation des caractères:

- *Conversion de l'image de la plaque d'immatriculation extraite du module précédent, au niveau de gris :*

```
Imgproc.cvtColor(imgDetectedPlateCandidate, imgDetectedPlateCandidateAmeliorated,
                Imgproc.COLOR_BGR2GRAY);
```

Figure 66 : Conversion de l'image au niveau de gris

- *Filtre Gaussien :*

```
Imgproc.GaussianBlur(imgDetectedPlateCandidateAmeliorated, imgDetectedPlateCandidateAmeliorated
                    , new Size (5,5), 0);
```

Figure 67 : Utilisation du filtre gaussien

- *Seuillage adaptatif gaussien :*

```
Imgproc.adaptiveThreshold(imgDetectedPlateCandidateAmeliorated, imgThreshold1, 250,
                        CV_ADAPTIVE_THRESH_GAUSSIAN_C, CV_THRESH_BINARY_INV, 15, 9);
```

Figure 68 : L'utilisation du seuil adaptatif gaussien pour l'image de la plaque

- *Opération morphologique : Fermeture*

```
Mat element2 = getStructuringElement(MORPH_RECT, new Size(3, 3));
Imgproc.morphologyEx(imgThreshold1, imgMorphologicalCaract, CV_MOP_CLOSE, element2);
```

Figure 69 : L'utilisation de la fermeture

- *Opération morphologique : Dilatation*

```
Mat dilate2 = Imgproc.getStructuringElement(Imgproc.MORPH_RECT, new Size(2, 1));
Imgproc.morphologyEx(imgThreshold1, imgMorphologicalCaract, Imgproc.MORPH_DILATE, dilate2);
```

Figure 70 : Application de la Dilatation

- *Chercher les Contours :*

```
imgContoursCaract = imgMorphologicalCaract.clone();
Imgproc.findContours(imgContoursCaract,
    contoursCaract,
    new Mat(),
    Imgproc.RETR_EXTERNAL,
    Imgproc.CHAIN_APPROX_NONE);
```

Figure 71 : Trouver les différents contours

- *Vérifier l'aspect ratio du caractère, si acceptable l'encadrer et l'extraire :*

```
imgMinAreaRect2 = imgDetectedPlateCandidate.clone();
int b = contoursCaract.size();
//System.out.println(b);
int i=1;
if (contoursCaract.size() > 0) {

    for (MatOfPoint matOfPoint : contoursCaract) {

        MatOfPoint2f points = new MatOfPoint2f(matOfPoint.toArray());
        RotatedRect box3 = Imgproc.minAreaRect(points);
        //System.out.println(i);
        if (box3.size.height > 12 && box3.size.width >12 &&
            box3.size.width <50 && box3.size.height <50){

            Rect m = box3.boundingRect();
            imgDetectedCaractCandidate = new MatOfRect(box3.boundingRect());
            Mat mm = new Mat(imgMorphologicalCaract, m);
            imgDetectedCaractTrue = mm.clone();
            Imgproc.rectangle(imgMinAreaRect2, box3.boundingRect().tl(),
                box3.boundingRect().br(), new Scalar(255, 0, 0));

            Imgcodecs.imwrite("C:\\Users\\PFE\\Documents\\NetBeansProjects\\NewEssaiePlate\\src\\"
                + "newessaieplate\\Caract\\New1\\c"+i+"17_imgDetectedCaractTrue.png",
                imgDetectedCaractTrue);

        }

        i=i+1;

    }

}
```

Figure 72 : Vérification des contours qui représente des caractères

4. Interfaces :

Dans ce qui suit nous allons présenter les différentes interfaces de notre application.

4.1. Interface d'authentification :

Cette interface s'affiche au lancement de l'application et exige à l'utilisateur de renseigner les informations nécessaires afin de lui permettre de se connecter à la plateforme. Les informations demandées sont le nom d'utilisateur et le mot de passe.



Figure 73 : Interface d'authentification

4.2. Interface d'accueil :

Après avoir été authentifié, l'utilisateur doit choisir une photo. Une fois la photo choisie, il aura le choix entre détecter la plaque, segmenter ou reconnaître les caractères.



Figure 74 : Interface d'accueil

Pour choisir une image, il faut cliquer sur le menu Fichier puis sur Parcourir :



Figure 75 : Menu Fichier

En cliquant sur Parcourir un FileChooser s'affiche pour choisir une image :

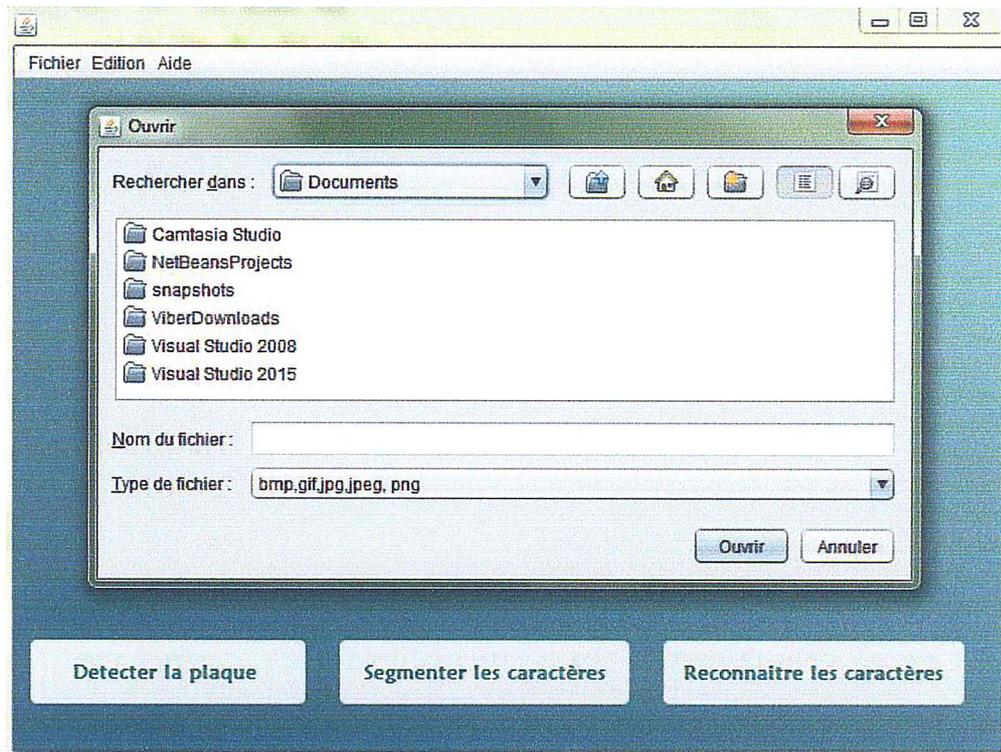


Figure 76 : Choisir une image

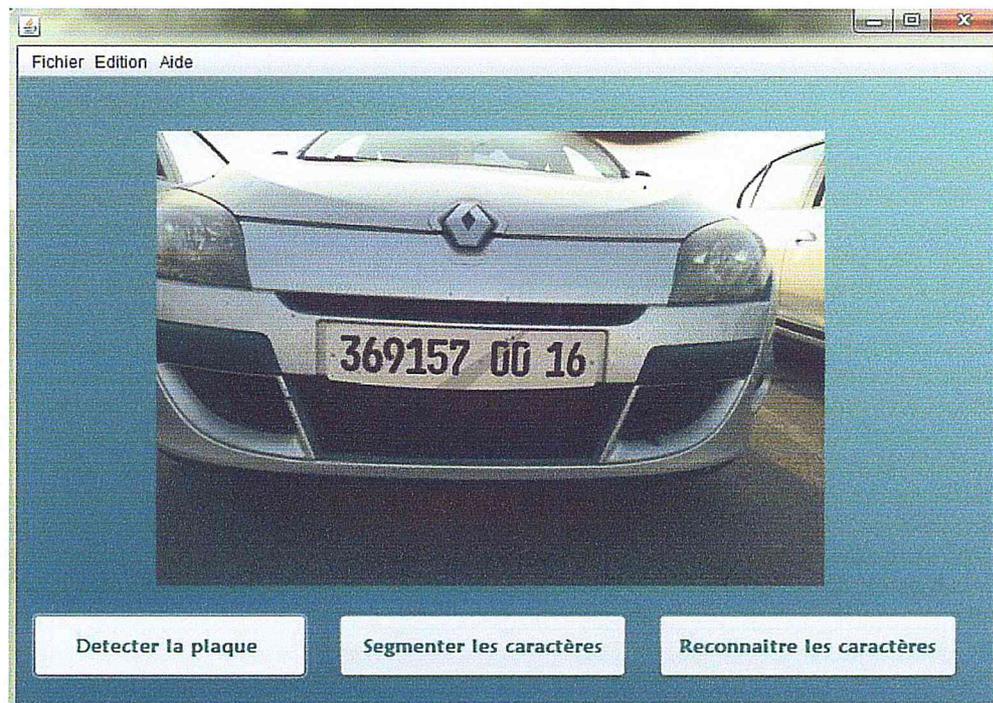


Figure 77 : Affichage de l'image choisie

Une fois l'image affichée, l'utilisateur aura le choix entre détecter la plaque, segmenter les caractères ou reconnaître les caractères.



Figure 78 : Résultats de segmentation

Quelques images du traitement sont affichées afin de montrer le déroulement du système.

5. Conclusion :

Nous avons présenté dans ce chapitre les détails techniques liés à la mise en œuvre de notre application; nous avons commencé par présenter les différentes technologies utilisées pour la réalisation de l'application. Ensuite nous avons cité les fonctions et filtres appliqués, la dernière partie été consacré à la présentation des différentes interfaces.

Chapitre 4 :
Tests et Résultats

1. Introduction

Afin d'évaluer l'efficacité de notre système, plus de 45 échantillons d'images en couleurs de véhicules algériens étaient testés. Toutes ces images étaient d'une taille d'environ 960*720 pixels, capturées par une caméra de téléphone d'une capacité de 13Mpixels.

Ces images étaient prises par l'avant ainsi que par l'arrière des véhicules, dans diverses conditions d'éclairages (temps nuageux, temps ensoleillés, journée, nuit). Prenant en considération : la distance entre la camera et la plaque d'immatriculation d'environ 1.5 à 3 mètres, ainsi que la position parallèle de la camera par rapport à la plaque.

Nous présentons dans ce qui suit, les différents résultats des tests effectués sur ces images.

2. Résultats des tests de détection et d'extraction des plaques d'immatriculation :



Figure 79 : Image d'entrée



Figure 80 : Image au niveau de gris

Après l'application du filtre gaussien et sobel nous obtenons l'image suivante :

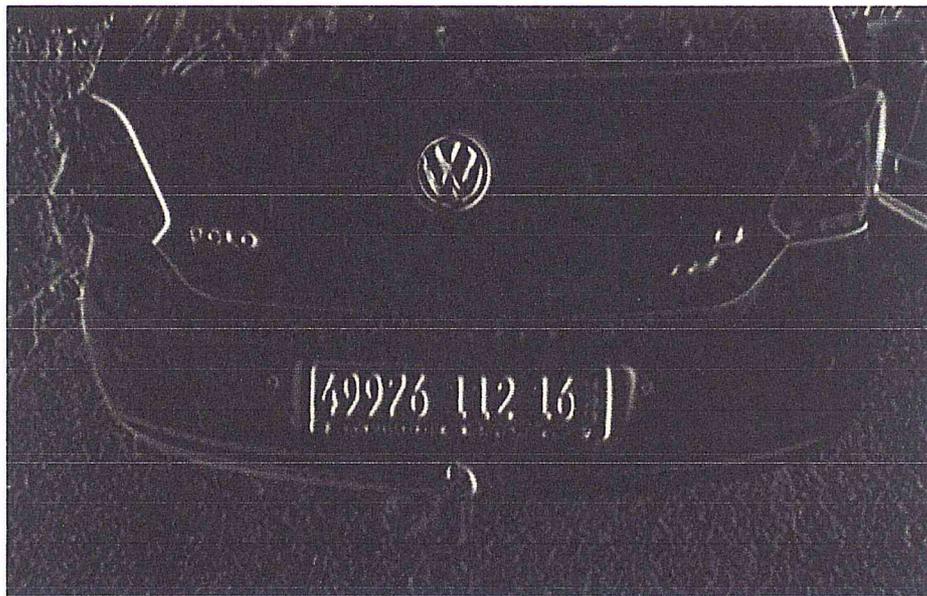


Figure 81 : Image après filtre Sobel

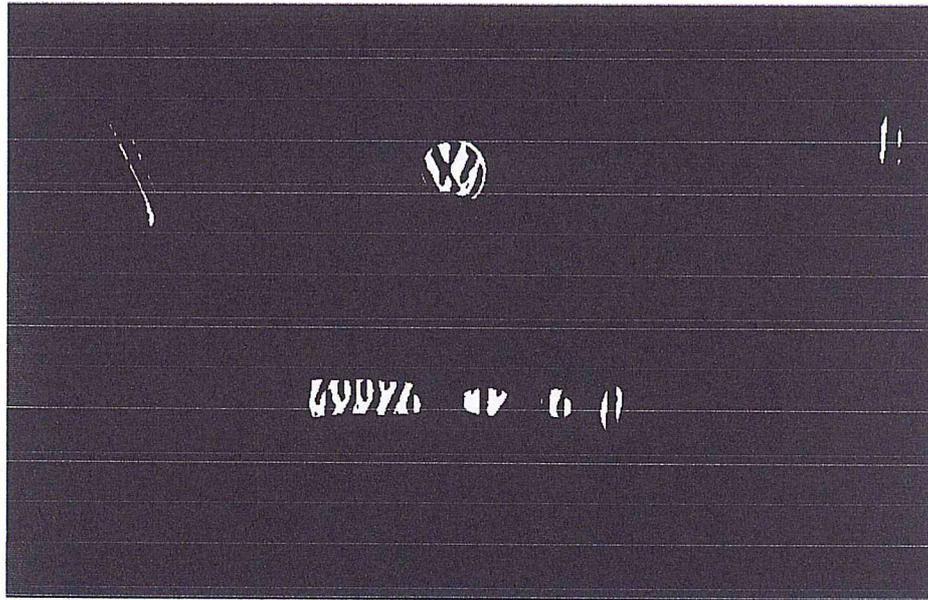


Figure 82 : Image au seuil adaptatif

Après avoir appliqué le seuil adaptatif, l'étape suivante revient à l'utilisation des opérations morphologiques, telle la dilatation et la fermeture.

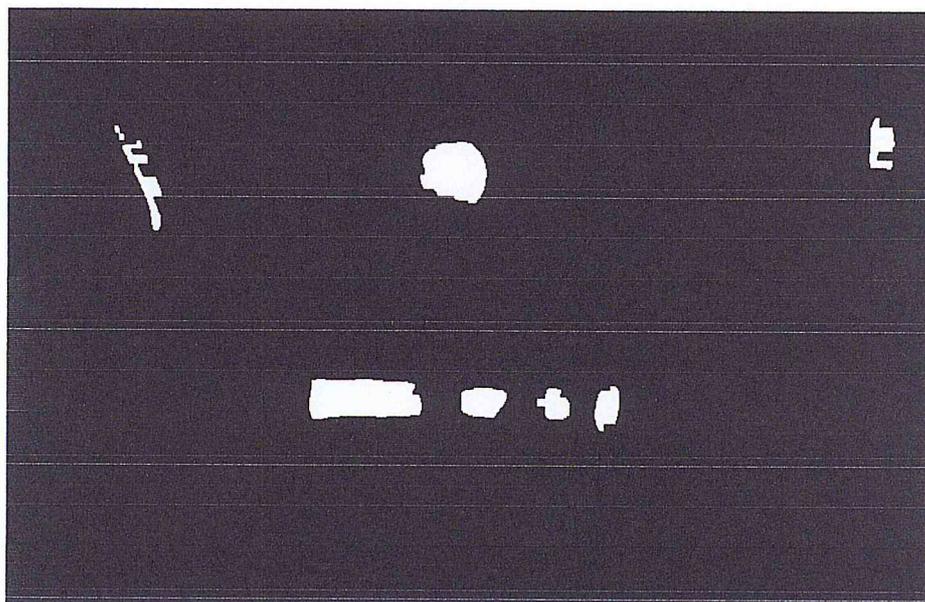


Figure 83 : Image résultante des opérations morphologiques



Figure 84 : Image final avec les rectangles trouvés en surbrillance en vert

Dans la figure ci-dessus, le système ne localise pas la plaque d'immatriculation, à cause des espaces entre les caractères de la plaque, la figure des opérations morphologiques présentée précédemment montre ces espaces clairement.

Pour que le système détecte la plaque, il faut qu'elle soit toute couverte en blanc dans l'image résultante des opérations morphologiques, pour ce faire, et pouvoir minimiser les espaces entre les caractères, nous avons procédé au redimensionnement de l'image après application du seuil adaptatif, les résultats étaient les suivants :

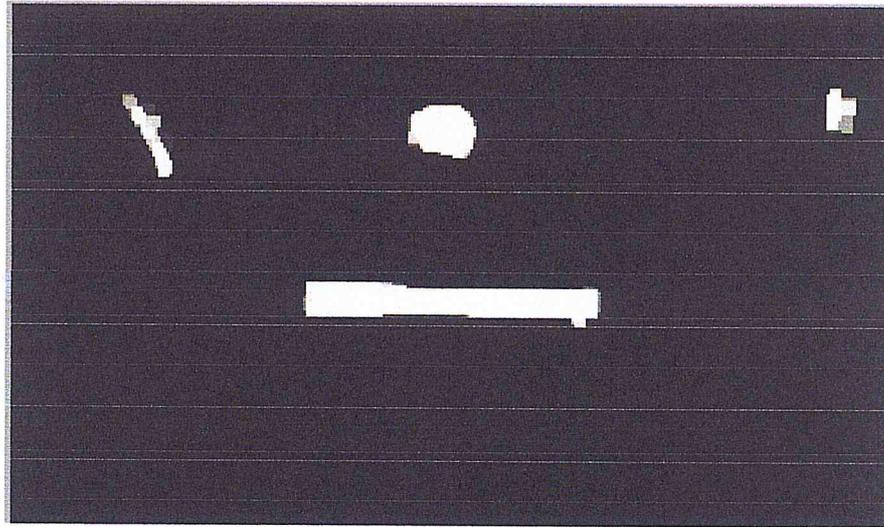


Figure 85 : Image résultante des opérations morphologiques après redimensionnement de l'image



Figure 86 : Image brute avec plaque d'immatriculation localisée et détectée en rouge



Figure 87 : Extraction de la plaque d'immatriculation

3. Résultats des tests du module Segmentation :

Les tests suivant, étaient faites sur la même plaque d'immatriculation de la figure 87.

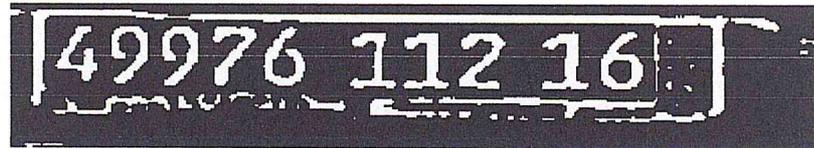


Figure 88 : Image de la plaque d'immatriculation après le seuil adaptatif



Figure 89 : Image de la plaque d'immatriculation résultante des opérations morphologiques

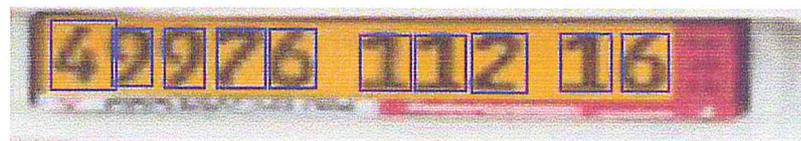


Figure 90 : Image de la plaque d'immatriculation après segmentation, avec chaque caractère encadré en bleu

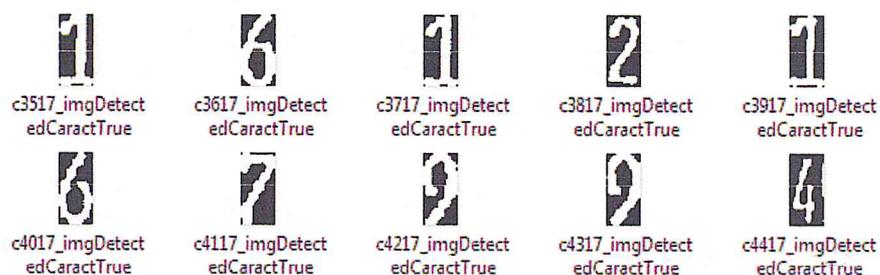


Figure 91 : Résultat de la Segmentation

4. Tests de détection et segmentation d'autres plaques de différents pays :

Les différents images des véhicules étrangers présentées ci-dessous, étaient prise depuis le moteur de recherche Google.

i) France

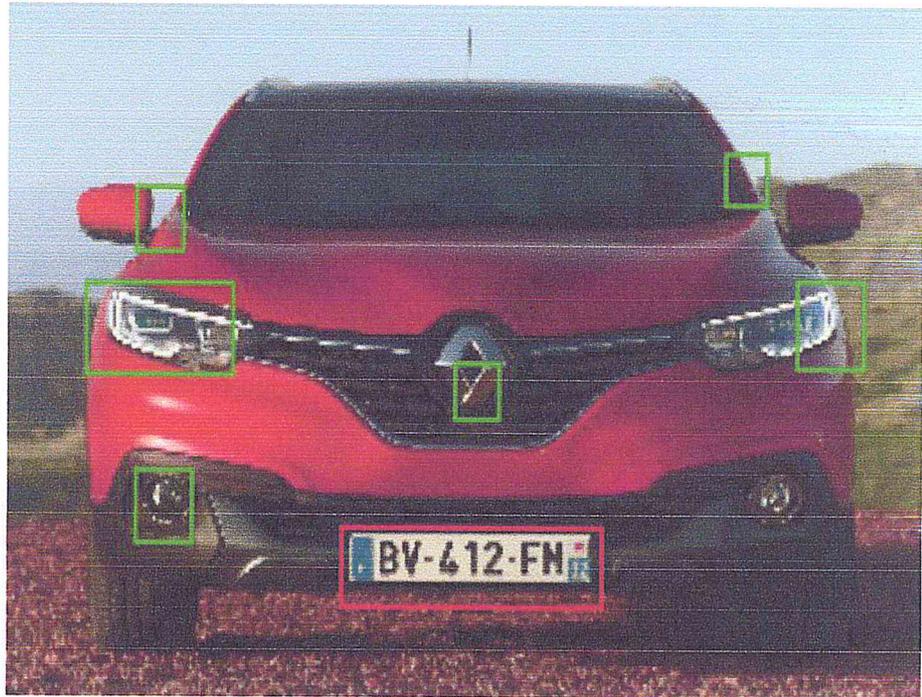


Figure 92 : Détection de la plaque



Figure 93 : Segmentation des caractères par des rectangles bleus

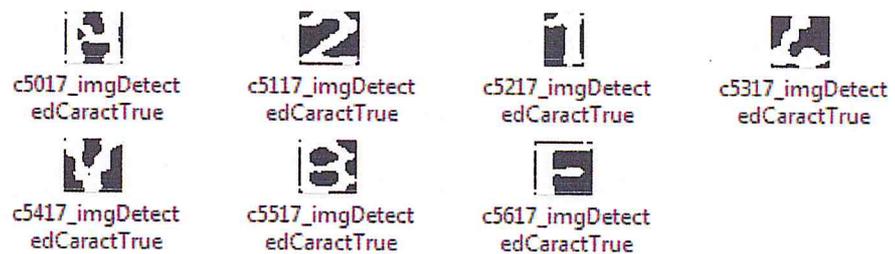


Figure 94 : Résultat de la Segmentation des caractères

ii) Grande Bretagne :



Figure 95 : Détection de la plaque

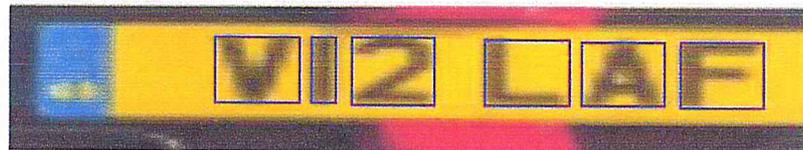


Figure 96 : Segmentation des caractères par des rectangles bleus

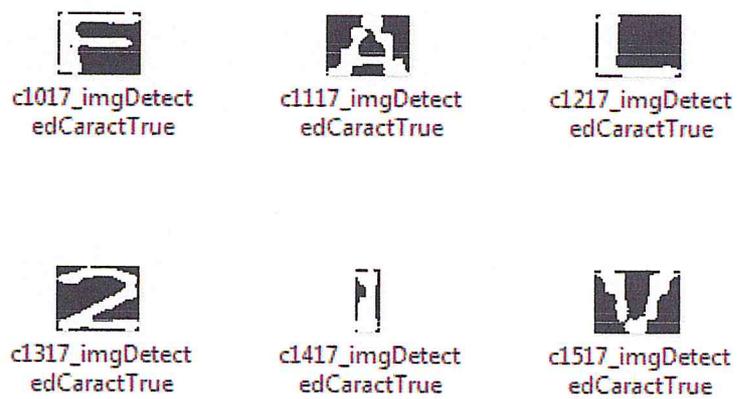


Figure 97 : Résultat de la Segmentation des caractères

iii) Allemagne :



Figure 98 : Détection de la plaque



Figure 99 : Segmentation des caractères par des rectangles bleus

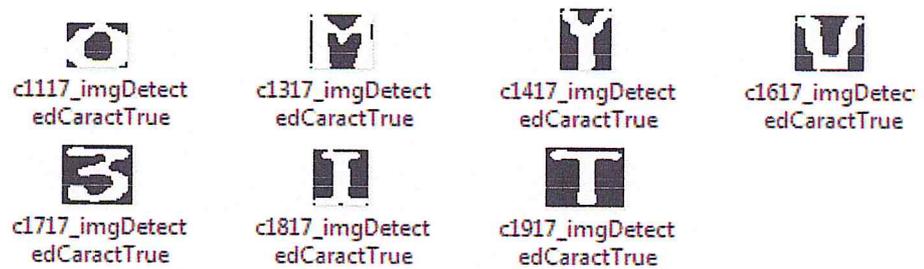


Figure 100 : Résultat de la Segmentation des caractères

iv) Inde :



Figure 101 : Détection de la plaque



Figure 102 : Segmentation des caractères par des rectangles bleus

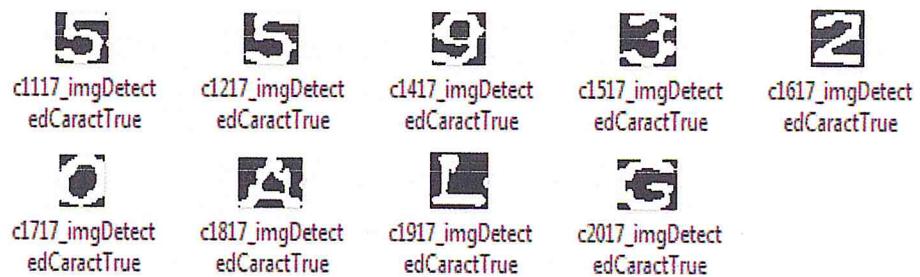


Figure 103 : Résultat de la Segmentation des caractères

v) *Autres Plaques Algériennes :*

(1)



Figure 104 : Détection



Figure 105 : Segmentation des caractères par des rectangles bleus

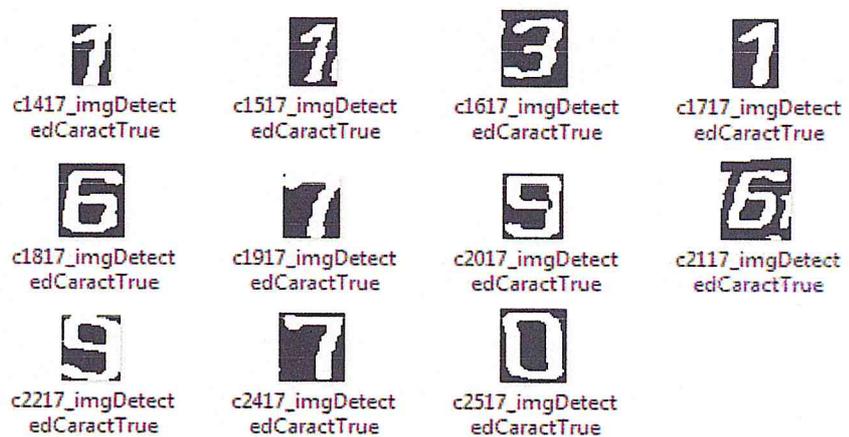


Figure 106 : Résultat de la Segmentation des caractères

(2)



Figure 107 : Détection de la plaque

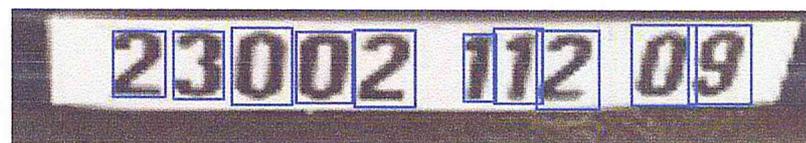


Figure 108 : Segmentation des caractères par des rectangles bleus

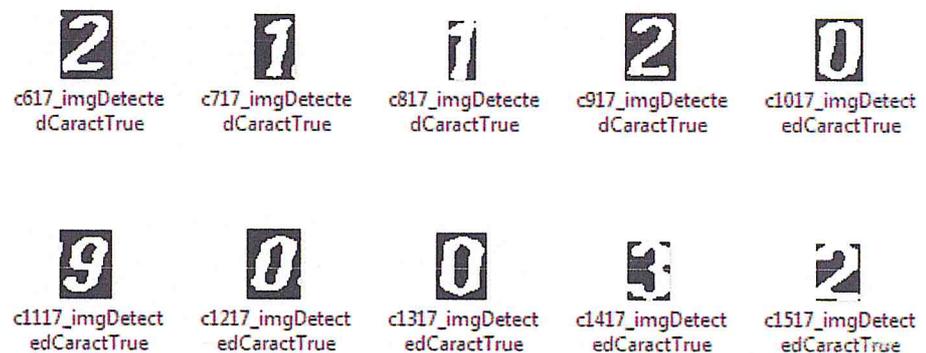


Figure 109 : Résultat de la Segmentation des caractères

5. Conclusion :

Après les différents tests effectués, le taux de réussite du module détection de la plaque est de 95%, le système a réussi à localiser et extraire 43 plaques sur 45. Quant au module de segmentation des caractères le taux de réussite était de 88%, tel que le système a segmenté 38 plaques sur 43.

Le temps d'exécution du système pour les deux modules détection de la plaques et segmentation des caractères varie entre 3 à 7 secondes.

Sur la base de ces résultats, on peut en déduire que les algorithmes proposés sont efficaces pour la détection et la segmentation des plaques d'immatriculation ayant un aspect ratio équivalent au plaques d'immatriculation algériennes.

***Conclusion
Générale***

Conclusion Générale :

Dans ce mémoire, nous avons présenté un système conçu pour la détection et la reconnaissance des plaques d'immatriculation, qui s'articule essentiellement autour de trois parties principales :

Tout d'abord, nous avons réussi à localiser et extraire la plaque d'immatriculation, en effectuant des opérations morphologiques ainsi que des tests de critères de zones telles que l'aspect ratio. Ensuite, les caractères de la plaque étaient segmentés en utilisant un seuil adaptatif, chaque caractère apparaît clairement segmenté de l'arrière-plan dans l'image binaire, ce qui facilite leurs extractions.

Afin de réussir la partie reconnaissance des caractères, un classifieur doit être utilisé, pour pouvoir fonctionner ce dernier doit être réglé par apprentissage. L'apprentissage consiste à donner au classifieur des exemples d'images en lui indiquant à chaque fois la classe de l'image, qui est un ensemble de données étiquetées que l'on nomme base d'entraînement ce qui revient à apprendre la description des objets à partir d'une base d'exemples.

Pour les deux parties qui sont déjà implémentées et testées : « Détection de la plaque » et « Segmentation des caractères ». Notre système a donné des résultats satisfaisant pour des plaques d'immatriculation différentes de plusieurs pays (ayant un aspect ration identique à celui des plaques algériennes) et dans diverses conditions d'éclairages (temps ensoleillé, temps nuageux, journée, nuit).

Bien que l'algorithme proposé concerne les plaques d'immatriculation des voitures algériennes de nombreuses parties des algorithmes sont facilement étendues pour être utilisées pour des plaques d'immatriculations d'autres pays.

Bibliographie

Références Bibliographiques :

- [1] : De la théorie a la pratique [en ligne]
<https://www.library.cornell.edu/preservation/tutorial-french/intro/intro-01.html>.
[Consulté le 02/02/2017].
- [2] : Université Rennes2 [en ligne] https://www.sites.univ-rennes2.fr/arts-spectacle/cian/image_numFlash/pdf/chap3_cours33.pdf [consulté le 17/02/2017]
- [3] : Antoine MANZANERA «Traitement Et Reconnaissance d'Images » Paris 6
- [4] : J-P COCQUEREZ et S.PHILIPP « Analyse d'images : filtrage et segmentation »
Edition MASSON
- [5] : Elise Arnaud – Edmond Boyer « Cours de traitement d'images » Université Joseph Fourier.
- [6] : Antoine MANZANERA - Cours de TI / ENSTA D9-2
- [7] : Filtre median [en ligne]<http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm>
[Consulté le 02/03/2017]
- [8] : Filtre median [en ligne]<http://stackoverflow.com/questions/18427031/median-filter-with-python-and-opencv> [consulté le 02/03/2017]
- [9] : Cornelia Fermüller& Marc Pollefeys, « EdgeDetection»,
<http://www.ics.uci.edu/~majumder/DIP/classes/EdgeDetect.pdf> [consulté mars 2017].
- [10]: Ruye Wang « Filtre Canny», 2013 [enligne]
<http://fourier.eng.hmc.edu/e161/lectures/canny/node1.html> [consulté le 05/02/2017]
- [11]: SandrineAugais - Richard Chevalier - Vincent Domurado, « Comparaison des critères de Canny, Spacek, Tagare et Figueiredo pour l'évaluation des performances des détecteurs de contours. » Projet Méthodes de Base d'Analyse des Images, mars 2001
- [12] : Fisher, S. Perkins, A. Walker and E. Wolfart, « image processing learning», 2003
[en ligne] http://homepages.inf.ed.ac.uk/rbf/HIPR2/hipr_top.htm [consulté le 06/01/2017].

[13]: Edouard Ritz, Tomasz Marszal « Epaisseur et localisation des contours des détecteurs par dérivation (Sobel, Prewitt, Roberts) ou par gradient morphologique ». Cree le 18 mars, 1998

[14] : Eric Lormeau, « Localisation et epaisseur des contours des detecteurs par derivation (Sobel, Prewitt, Roberts) ou par Gradient Morphologique. » Projet de MTI Master IST (98)

[15] : Canny Edge Detector [en ligne]
http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html [Consulté le 10 janvier 2017]

[16] R. Fisher, S. Perkins, A. Walker et E. Wolfart. « Filtre laplacien », [enligne]
<http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm> [consulté le 12/02/2017]

[17]: « Gaussian Filtering », the university of auckland, new zealand, 25/05/2010

[18] : Espace RGB [enligne] https://www.sites.univ-rennes2.fr/arts-spectacle/cian/image_numFlash/pdf/chap3_cours33.pdf [consulté le 03/03/2017]

[19]: « implémentation d'algorithmes efficaces de seuillage d'images. », [enligne]
<https://sites.google.com/site/lizantchristopher/home> [consulté le 03/03/2017]

[20]: Otsu Thresholding [enligne]
<http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html> [consulté le 15/03/2017]

[21] : Opérations morphologiques de base : dilatation, érosion, ouverture et fermeture binaires [enligne] <https://dpt-info.u-strasbg.fr/~cronse/TIDOC/MM/deof.html>

[consulté le 20/03/2017]

[22] : « le traitement d'image » [enligne], <http://raphaello.univ-fcomte.fr/IG/TraitementImages/TraitementImages.htm#Introduction> [Consulté le 20 fevrier 2017]

[23] : Amincissement [enligne]
<http://www.tsi.telecomparistech.fr/pages/enseignement/ressources/beti/topolo/Chapitre3-6.htm> [consulté le 30/03/2017]

[24] : Amincissement [enligne] <http://alexandre.alapetite.fr/dess-irr/timage/aphelinux/>
[consulté le 30/03/2017]

[25] : Connected components [enligne]
<http://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm> [consulté le 13/04/2017]

[26]: Mean Shift [enligne] <http://xphilipp.developpez.com/articles/meanshift/>

[Consulté le 25/02 /2017]

[27] : K-means [enligne] <https://www.di.ens.fr/~fbach/courses/fall2010/cours3.pdf>

[consulté le 25/02/2017]

[28]: Xiangjian He, Lihong Zheng, QiangWu, WenjingJia, BijianSamali et MarimuthaPalaniswamin, "Segmentation of Characters on car Licence plates",

[29]: Muhammad H Dashtban, Zahra Dashtban, Hassan Bevrani, "A Novel Approach for Vehicle License Plate Localization and Recognition", International Journal of Computer Applications (0975 – 8887) Volume 26– No.11, July 2011.Conference: International Workshop on Multimedia Signal Processing, MMSP 2008, October 8-10, 2008, Shangri-la Hotel, Cairns, Queensland, Australia

[30] : PALUKU VAGHENI Elois. Conception Et Réalisation d'un Système d'inscription En Ligne Dans Les Institutions Universitaires : Cas De l'UCBC /Beni[thèse du Diplôme de Grade en Sciences Appliquées, département de Génie Informatique] Congo: université Chretienne Bilingue du Congo;2014

[31] : Shuji ZHAO , « Apprentissage et Recherche par le Contenu Visuel de Catégories Sémantiques d'Objets Vidéo », Master 2 Informatique - parcours Images, Université Paris Descartes, Juillet 2007

[32] : Cascade Classification, HaarFeature-based Cascade Classifier for Object,OpenCv,

[Enligne] http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html,

[Consulté le 10 mars 2017]

[33] : Détection des visages à l'aide Haar Cascades, [enligne] http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html [Consulté le 11 mars 2017]

[34] : « A cascade of generative and discriminative classifiers for vehicledetection », [enligne] http://www.isir.upmc.fr/files/RFIA2008_Negri.pdf [consulté le 02/04/2017]

- [35] :Neural Networks [enligne]
<http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html> [consulté le 05/04/2017]
- [36] : NetBeans .[En ligne]. <https://doc.ubuntu-fr.org/netbeans>. [Consulté le 10/05/2017].
- [37] : java. [enligne]. <http://www.futura-sciences.com/tech/definitions/internet-java-485/> [Consulté le 14/05/2017].
- [38] : Opencv [enligne] <https://openclassrooms.com/courses/introduction-a-la-vision-par-ordinateur/avant-de-commencer-8> . [Consulté le 13/05/2017].
- [39] : How to Use OpenCV with Java under NetBeans [enligne]
<https://www.codeproject.com/Tips/717283/How-to-Use-OpenCV-with-Java-under-NetBeans-IDE>. [Consulté le 13/05/2017].
- [40] : BENCHENIEF Abderrazak. Réduction du bruit d'un signal monodimensionnel par le filtrage adaptatif récursif[Thèse D'INGENIEUR D'ETAT EN ELECTRONIQUE]. BISKRA : UNIVERSITE MOHAMED KHEIDER BISKRA FACULTE DES SCIENCES ET DE LA TECHNOLOGIE.
- [41] : les opération morphologique [en ligne]
<https://dptinfo.ustrasbg.fr/~cronse/TIDOC/MM/deof.html> [consulté le 16/05/2017]
- [42] : Radu HORAUD CNRS , Olivier MONGA INRIA . Vision par ordinateur : outils fondamentaux .2nd ed. Editions Hermes.
- [43] : apprentissage machine [en ligne] <https://docs.microsoft.com/fr-fr/azure/machine-learning/machine-learning-what-is-machine-learning> [consulté le 16/05/2017]
- [44] : support vector machine [en ligne] <http://pageperso.univ-lr.fr/arnaud.revel/MesPolys/SVM.pdf> [consulté le 16/05/2017]
- [45] : KNN [enligne] <http://webia.lip6.fr/~rifqi/COURS2001-2002/IA/knn.pdf>
[consulté le 16/05/2017]
- [46] : C. Touzet, Les Réseaux de Neurones Artificiels : Introduction au Connexionnisme, 1992
- [47] : deep learning [en ligne] <http://deeplearning.net/> [consulté le 16/05/2017]
- [48] : Ibrahim T, Kirami K. A New License Plate Recognition System Using Artificial Neural Networks, 7 June 2016, 1508-0766 .

- [49]: Zheng D, Zhao Y, Wang J. An Efficient Method of License Plate Location. *Pattern Recognition Letters*. 2005;(26): 2431 – 2438.
- [50] N Sulaiman¹, AF Razali¹, MS Jadin², et al. Development of Online Vehicle Plate Recognition System. *Jan 2016*;6(1).
- [51] Ganesh RJ, Kailash JK. Extraction of Vehicle Number Plate, *IJIET*, 2 Déc 2015;(6).
- [52] Suproakash D, Amitava C, Joydeep M. An Efficient Technique to Recognize License Plate using Morphological Edge Detection and Character Matching Algorithm, *IJOCA*, Sep 2014;101(15):0975 – 8887.
- [53] Sneha G, Patel. VEHICLE LICENSE PLATE RECOGNITION USING MORPHOLOGY AND NEURAL NETWORK, Institute of Technology, Vasad, *IJCI*, Feb 2013;2(1) 388 306.
- [54] ANISHIYA P, PROF S, MARY J. Number Plate Recognition for Indian Cars Using Morphological Dilation and Erosion with the Aid Of Ocrs, *IACSIT*, 2011;(4).
- [55] Kanani P, A. Gupta, D. Yadav, et al. Vehicle license plate localization using wavelets, *IEEE*, 11-12April, 2013.
- [56] Zhang ZC, Tang YY. License plate recognition algorithm based on derived kernel, *IEEE*, 2012;238-243
- [57] Zunino R, Rovetta S. Vector quantization for license-plate location and image coding, *IEEE*, Feb2000;47(1):159–167.
- [58] Hsieh CT, Juan YS, Hung KM. Multiple license plate detection for complex background, in *Proc Int Conf AINA*, 2005;2: 389–392.
- [59] Shi X, Zhao W, Shen Y. Automatic License Plate Recognition System Based on Color Image Processing, *LNCS 3483*, 2005; 1159–1168.
- [60] Zimic N, Ficzko J, Mraz M et al. The fuzzy logic approach to the car number plate locating problem, *IIS*, 1997; 227– 230.

[61] Nomura S, Yamanaka K, Katai O et al. A novel adaptive morphological approach for degraded character image segmentation, Pattern Recognition, 11 nov 2015; 38(11): 1961–1975.

[62] Cowell J , Hussain F. A Fast Recognition System for Isolated Arabic characters, IEEE , 2002; 650-654

[63] : «Quelques méthodes de segmentation d'images », Agnès Desolneux, MAP5, Université Paris 5, ANR Mipomodim, 10 novembre 2006.

