

Faculté des Sciences Exactes et de l'Informatique
Département de Mathématiques et d'Informatique
Filière : Informatique

Option : **Ingénierie des Systèmes d'Information**

**Mémoire présenté pour l'obtention
Du diplôme de Master Académique**

Intitulé

Développement d'un Système de reconnaissance des
plaques d'immatriculation en utilisant
l'apprentissage automatique

Etudiant (e) : Mr. BOUALEM M'hamed

Mr. LASSAL Mawloud

Encadrant (e) : « KADEM Habib »

Année Universitaire 2022-2023



Résumé

Suite au trafic important de la circulation automobiles en Algérie, le contrôle des véhicules en circulation et la sécurité routiers sont devenus un enjeu stratégique majeur pour les services de sécurité (Police nationale et la Gendarmerie nationale, la douane), ainsi que les gestionnaires des parkings (Entreprises, Aéroports, SONATRACH, universités...) ont besoin à un mécanisme mobile et efficace qui traite en temps réel le contrôle d'accès d'entrées et de sorties de son propre parc automobile.

Comme tout système de reconnaissance automatique des plaques d'immatriculation, notre travail sera évidemment réalisé suite aux étapes suivantes. La première consiste à extraire à partir d'une image acquise au niveau de gris de la plaque d'immatriculation en se basant essentiellement sur une méthode de détection de contours. Après, on passe à la segmentation de l'image résultat (la plaque) en caractères afin d'être analysé par un système de Reconnaissance Optique de Caractère basé sur une technique d'apprentissage automatique. La finalité de notre travail consiste non seulement à tester le système, mais aussi de revoir le prétraitement des images acquises et des paramètres de la technique utilisée afin d'arriver à un meilleur résultat.

Mots clés : Traitement d'image, Détection de contour, Segmentation, d'apprentissage automatique.



Sommaire

Table des matières

Introduction générale.....	7
Chapitre I - Généralités sur le traitement d'images	
1. Introduction.....	10
2. Acquisition d'une image	10
3. Définition de l'image.....	10
4. Caractéristiques d'une image numérique.....	11
□ Dimension.....	11
□ Résolution	11
□ Bruit.....	12
□ Histogramme.....	12
5. Le codage des couleurs.....	13
5.1 Image noir et blanc (binaire).....	13
5.2 Images à niveaux de gris.....	13
5.3 Images en couleurs	14
6. La segmentation	17
6.1 Définition de la segmentation	17
6.2 Objectifs de la segmentation :.....	18
6.3 Le choix d'une technique de segmentation.....	18
6.4 Méthode de segmentation	18
7. La détection des contours	21
7.1 Objectifs de détection des contours.....	21
7.2 Le détecteur de contour Canny	22
8. La morphologie mathématique	22
8.1 L'érosion	22
8.2 La dilatation	23
8.3 L'ouverture	23
8.4 La fermeture.....	24
9. Conclusion	24
Chapitre II - La reconnaissance de plaques d'immatriculation	

1.	Introduction.....	26
2.	Reconnaissance des formes	26
2.1.	Intelligence Artificielle.....	26
2.2.	Définitions	27
2.3.	Méthodes de reconnaissance de formes.....	27
3.	Schéma général d'un système de Reconnaissance des formes	28
3.1.	Préparation des données	29
3.1.1.	Numérisation.....	29
3.1.2.	Prétraitement	29
3.1.3.	Calcul des représentations	29
3.1.4.	Apprentissage.....	29
3.1.5.	Classification.....	29
3.1.6.	Post traitement.....	30
4.	Etat de l'art sur les systèmes RAPI.....	30
5.	Méthodes de classification.....	30
5.1.	Machine à Vecteurs de Support (SVM)	31
5.2.	Définition de SVM.....	31
5.3.	Apprentissage automatique et SVM.....	32
5.4.	Principe de fonctionnement général de SVM.....	32
5.4.1.	Notions de base	32
	Linéarité et non-linéarité	34
5.4.2.	Cas linéairement séparable	34
5.4.3.	Cas non linéairement séparable.....	34
5.4.3.1.	Fonction noyau (kernel).....	35
5.4.3.2.	Condition de Mercer	36
5.5.	SVMs multi-classe	36
5.5.1.	Un contre tous	37
5.5.2.	Classification par pair.....	37
5.6.	Architecture générale d'une machine à vecteur support	38
6.	Conclusion	39
Chapitre III - Développement et expérimentation du système RPI		
1.	Introduction.....	41
2.	Les bibliothèques :.....	42
2.1.	OpenCV	42

2.2. Pytesseract	42
2.3. Matplotlib.....	42
2.4. Rembg.....	42
2.5. Keras.....	42
2.6. TensorFlow	43
2.7. Scikit-Learn.....	43
2.8. joblib.....	44
3. Plaque D'immatriculation Algérienne :.....	44
4. Le Système De RPI Propose.....	44
4.1. Acquisition :.....	45
4.2. niveau de gris :.....	46
4.3. Filtrage d'image :.....	46
4.4. Binarisation d'image :.....	47
4.5. Détection du contour :.....	47
4.6. Localisation de la plaque d'immatriculation :.....	48
4.7. Segmentation et étiquetage des caractères:.....	49
4.8. Reconnaissance des caractères :.....	50
5. Résultats Expérimental	52
6. Conclusion	53
Conclusion Générale	54

Liste de figures

Figure 1: Les pixels accessibles par leurs coordonnées (x, y), x pour les lignes et y pour.....	9
Figure 2: Image à 1 000 pixels de largeur et 750 pixels de hauteur, pour un total de.....	10
Figure 3: Histogramme et palette associés à une image	10
Figure 4: Image noir et blanc	11
Figure 5: Image en niveau de gris	12
Figure 6: Image en couleur	12
Figure 7: Illustration du modèle de couleurs RGB.....	13
Figure 8: Exemple d'ajout de couleur rouge, vert et bleu ensemble pour créer d'autres.....	13
Figure 9: Illustration du modèle de couleurs HSV.....	14
Figure 10: Illustration de l'espace couleurs L*a*b.	15
Figure 11: (a) image d'origine, (b) image segmentée.....	15
Figure 12: Application de la méthode Otsu pour une image niveau de gris.....	18
Figure 13: Modèles de contours.....	19
Figure 14: (a) image érodée, (b) image binaire.....	21
Figure 15: (a) image binaire, (b) image dilatée.....	21
Figure 16: Exemple d'ouvertures avec un élément structurant en croix	22
Figure 17: Exemple de fermetures avec un élément structurant horizontal.....	22
Figure 18: Schéma général d'un système de reconnaissance des formes.....	26
Figure 19: Exemple d'un hyperplan sépare deux ensembles de points.....	30
Figure 20: Exemple de vecteurs de support.....	31
Figure 21: Exemple de marge maximal (hyperplan valide).....	31
Figure 22: Schéma spectacles les différents modèles des SVM	32
Figure 23: a) Cas linéairement séparable, b) Cas non linéairement séparable	32
Figure 24: Exemple de changement de l'espace de données	33
Figure 25: Illustration de passage à R3.....	33
Figure 26: Trois classes séparées par la méthode d'un contre tous avec des séparateurs	35
Figure 27: Les surfaces rouges représente le « Reject decision ».....	35
Figure 28: Classification multi-classe par paire.....	35
Figure 29: Architecture d'une machine à vecteur support.....	36

Figure 30 : Plaque d'immatriculation algérienne	43
Figure 31 : Organigramme du système de RPI	45
Figure 32 : Image originale	45
Figure 33 : Conversion l'image en niveaux de gris.....	46
Figure 34 : Filtrage d'image, (a) : Image en niveau de gris, (b) : Image filtrée	46
Figure 35 : Binarisation d'image, (a) : Image originale, (b) : Image binaire	45
Figure 36 : Détection des contours	45
Figure 37 : Algorithme de localisation de la plaque	48
Figure 38 : Localisation de la plaque.....	49
Figure 39 : Extraction de plaque d'immatriculation	50
Figure 40 : Suppression les objets indésirables	50
Figure 41 : Segmentation de la plaque	50
Figure 42 : Redimensionnement des chiffres	51
Figure 43 : Algorithme de reconnaissance de caractères	52



Introduction générale

La reconnaissance automatique de plaques d'immatriculation des véhicules est le plus intéressant et stimulant sujet de recherche à partir de ces dernières années. Il est montré que les plaques d'immatriculation sont forme et de taille différente et ont également couleur différente dans différents pays.

Pour identifier un véhicule par un système de transport, les techniques de traitement automatique de l'image numérique est largement utilisés. Les systèmes de reconnaissance des plaques d'immatriculation ont large éventail d'applications telles que les maintenances de la circulation, le traçage des voitures volées ainsi que les systèmes électroniques automatiques de péage et beaucoup plus. Mais le but principal est de contrôler le système de gestion de trafic.

En Algérie, les plaques d'immatriculation ont la même forme (fond blanc ou jaune avec la couleur noire pour les caractères). Pour ces conditions, on va proposer un système qui se concentre sur deux phases principales : La localisation (prétraitement et la détection des plaques) et la reconnaissance des plaques d'immatriculation (la segmentation et la reconnaissance des chiffres).

La contribution principale de ce travail est de construire un système de reconnaissance automatique basée sur les réseaux de neurones appliqué aux plaques d'immatriculation algérien extraites en temps réel à partir d'une image de véhicule.

Ce manuscrit est organisé en trois chapitres :

❖ Chapitre 01 : Revue de littérature sur le traitement d'image

Présente quelques généralités sur la vision artificielle et le traitement d'images ainsi que des notions liées aux images, les caractéristiques d'une image numérique, la segmentation et la détection des contours. Il présente aussi les opérateurs morphe-mathématiques et se termine avec une conclusion.

❖ Chapitre 02 : Revue de littérature sur la reconnaissance de plaques d'immatriculation

Au début de ce chapitre nous allons présenter une vue générale sur la reconnaissance de forme RDF ensuite nous allons illustrer quelques travaux sur la détection et reconnaissance automatique des plaques d'immatriculation. A la fin, nous allons établir une évaluation des différents travaux dans le domaine en présentant leurs avantages ainsi que leurs inconvénients.

❖ Chapitre 03 : Méthode proposée

Ce chapitre est composé de trois parties, la première présente l'approche que nous avons développée pour l'identification et la reconnaissance des plaques d'immatriculation algériennes, dans la deuxième partie nous allons montrer l'implémentation de l'approche proposée sous forme de système complet en illustrant les principales interfaces d'entrées sorties. Concernant la troisième partie, présente l'environnement de travail ainsi que les résultats obtenus avec l'évaluation de performance de l'algorithme adopté ainsi que les résultats obtenus.

Et enfin, nous terminerons ce manuscrit avec une conclusion générale et nous proposerons des perspectives.

Chapitre I

Généralités sur le traitement d'images

1. Introduction

Le traitement d'images est un domaine très vaste qui a connu, et qui connaît encore, un développement important depuis quelques dizaines d'années. Nous ayons désigné par traitement d'images numériques l'ensemble des techniques permettant de modifier une image numérique afin d'améliorer ou d'en extraire des informations. Dans ce chapitre, nous abordons les notions de base nécessaires à la compréhension des techniques de traitement d'images. Ensuite, nous allons donner un aperçu sur les différentes techniques connues dans ce domaine.

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information. Généralement un système de traitement numérique d'images est composé de quatre parties essentielles :

- Acquisition de l'image
- Prétraitement
- Segmentation
- Interprétation

2. Acquisition d'une image

L'acquisition d'images constitue une des opérations essentielles de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation. Ces systèmes d'acquisition d'image, dénommés optiques, peuvent être classés en deux catégories principales :

- Les caméras numériques,
- Les scanners.

3. Définition de l'image

Une image est une représentation planaire d'une scène ou d'un objet situé en général dans un espace tridimensionnel (figure 1). Son élaboration résulte de la volonté de proposer une entité observable par l'œil humain. Ceci explique d'une part son aspect planaire et d'autre part

le fait que l'information élémentaire associée à chaque point de l'image soit transcrite en niveau de gris ou en couleur[1].

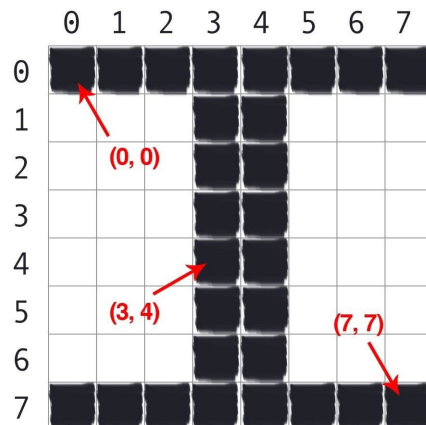


Figure 1: Les pixels accessibles par leurs coordonnées (x, y) , x pour les lignes et y pour les colonnes

4. Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants [2]:

➤ Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image (voir figure 1).

➤ Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur ; plus grand est ce nombre, meilleure est la résolution (figure2).



Figure 2: Image à 1 000 pixels de largeur et 750 pixels de hauteur, pour un total de 750 000 pixels.

➤ **Bruit**

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur [04].

➤ **Histogramme**

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est répartie la majorité des niveaux de gris (couleur) dans le cas d'une image trop claire ou d'une image trop foncée.

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant.

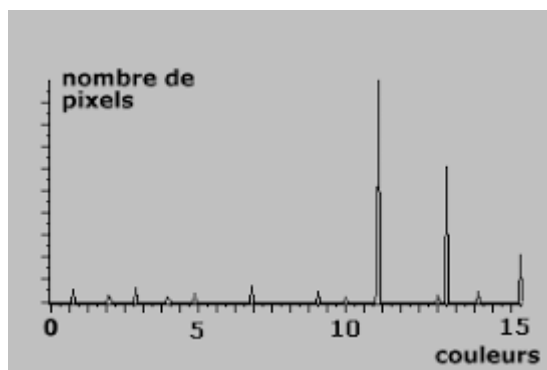


Figure 3: Histogramme et palette associés à une image.

5. Le codage des couleurs

D'une manière générale, une image apparaît comme une matrice où chaque case qui contient des nombres associés à une couleur. Usuellement on distingue trois grands types de couleurs pour une image numérique :

- Le noir et blanc (binaire)
- Le niveau de gris
- La couleur

5.1 Image noir et blanc (binaire)

Une image binaire est une matrice rectangulaire dont les éléments valent 0 ou 1. Lorsque l'on visualise une telle image, les zéros sont représentés par des noirs et les uns par des blancs, comme illustré la figure suivante :



Figure 4:Image noir et blanc

5.2 Images à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.

Le nombre de niveaux de gris dépend du nombre de bits utilisés pour décrire la " couleur " de chaque pixel de l'image. Plus ce nombre est important, plus les niveaux possibles sont nombreux.



Figure 5:: Image en niveau de gris

5.3 Images en couleurs

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités.

En effet, il faut tout d'abord choisir un modèle de représentation. On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs, ...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B.). La figure (6) Montre une image en couleurs[3].



Figure 6: Image en couleur

Il existe des nombreux modèles de couleurs populaires utilisés pour décrire une couleur, tels que RVB, CMJN, HSV, HSL, etc.

Le modèle de couleur RVB est le plus couramment utilisé dans les langages de programmation.

➤ **Modèle de couleur RVB**

Dans le modèle de couleur RVB, n'importe quelle couleur peut être générée en

mélangeant 3 couleurs primaires, à savoir, le rouge, le vert et le bleu (figure 7). Dans ce modèle, une couleur peut être décrite en spécifiant un groupe de 3 valeurs numériques (généralement allant de 0 à 255), Chacun spécifiant l'intensité des couleurs rouge, verte et bleue présentes dans une couleur donnée. Ces 3 couleurs sont également parfois appelées « canaux » ou « bandes ».

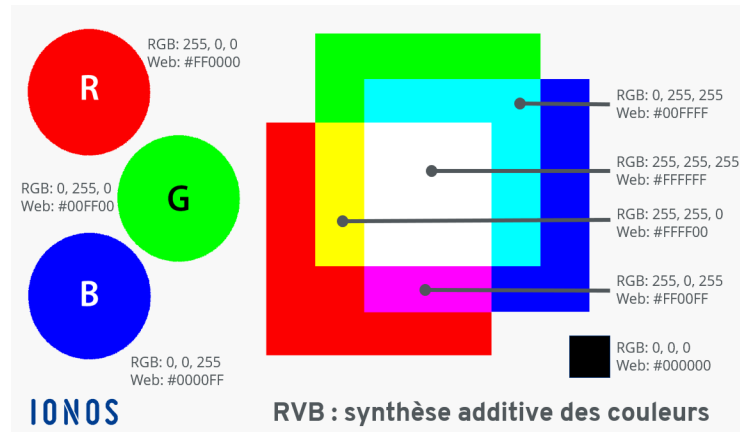


Figure 7: Illustration du modèle de couleurs RGB.

Dans la pratique, nous représentons une couleur sous la forme d'un tuple de 3 valeurs, correspondant au rouge, au vert, et les intensités bleues, dans cet ordre (sauf indication contraire). Les nombres de cette représentation, compris entre 0 et 255, sont représentés en codage décimal (base 10). Dans certains cas, ces nombres sont également réduits et spécifiés dans la plage de 0 à 1 (figure 8)[4].



Figure 8: Exemple d'ajout de couleur rouge, vert et bleu ensemble pour créer d'autres couleurs

➤ Le modèle de couleur HSV

Les modèles de couleurs HSI, HSL, HSV basés sur idée du système visuel humain [06]. La famille HSI (Hue Saturation Intensity) des modèles de couleurs utilisent des coordonnées cylindriques pour représentation des points RVB [07]. L'importance du HSI le modèle de couleur repose sur deux aspects principaux ; la composante I est séparée de la teinte H et de la saturation S qui sont les composants de chrominance, et deuxièmement ces composants de chrominance dépendent de la façon dont l'homme percevez ce spectre de couleurs [07]. Les espaces colorimétriques HSL et HSV (High Saturation Value) sont presque similaires sauf que

HSL attribue les valeurs de couleur élevées pour les couleurs qui s'approche de la couleur blanche avec un borné saturation [08]. Cette propriété augmenterait le degré de complication du modèle [08]. Modèle de couleur HSL est représenté dans un seul cône, tandis que HSI ou HSL est représenté en double cône [03] (figure 9).

L'espace de couleur HSV est largement utilisé dans les applications de vision par ordinateur, en particulier si nous sommes intéressés par le suivi de la couleur d'un objet dans une image. Il est beaucoup plus facile de définir une gamme de couleurs valides en utilisant HSV qu'en RGB.

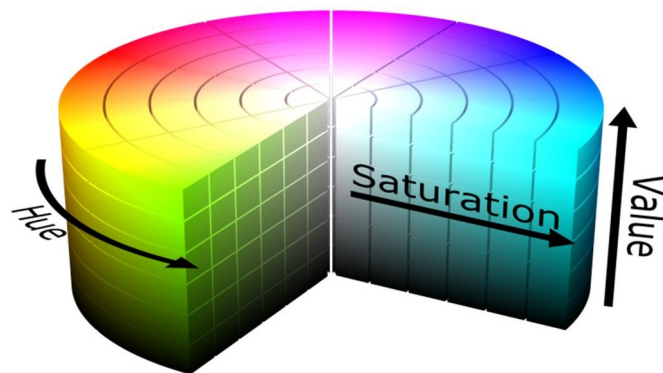


Figure 9: Illustration du modèle de couleurs HSV.

➤ Espace $L^*a^*b^*$

Le système Lab est issu du CIE XYZ. Il essaye de prendre en compte la réponse logarithmique de l'œil. Il possède le grand avantage d'être uniforme. Il est très utile dans le cas de mélanges de pigments, par exemple, pour l'industrie graphique ou du textile.

Les couleurs sont définies par trois valeurs : Luminosité (luminance) codée en pourcentages, a^* et b^* correspondent à l'information colorée (chrominance) où la couleur est définie à partir d'un mélange de vert à magenta (a^*) et un mélange de bleu à jaune (b^*). Les valeurs sont comprises entre -120 et +120 pour a^* et b^* .

Les écarts entre les couleurs sont plus proches de la perception humaine. Toutes les couleurs de même luminosité s'inscrivent dans un plan circulaire que traversent les axes a^* et b^* . Sur l'axe a^* , les valeurs positives correspondent aux rouges, les valeurs négatives aux verts. Sur l'axe b^* , les valeurs positives correspondent aux jaunes et les valeurs négatives aux bleus. La luminosité varie verticalement (figure 10).

Alors que l'espace $L^*a^*b^*$ est moins intuitif et pas aussi facile à comprendre que les espaces HSV et RGB, il est largement utilisé en vision par ordinateur. Cela est dû à la distance entre les couleurs ayant une signification perceptive réelle, nous permettant de surmonter divers problèmes de conditions d'éclairage. Il sert également de descripteur couleur d'image puissant.

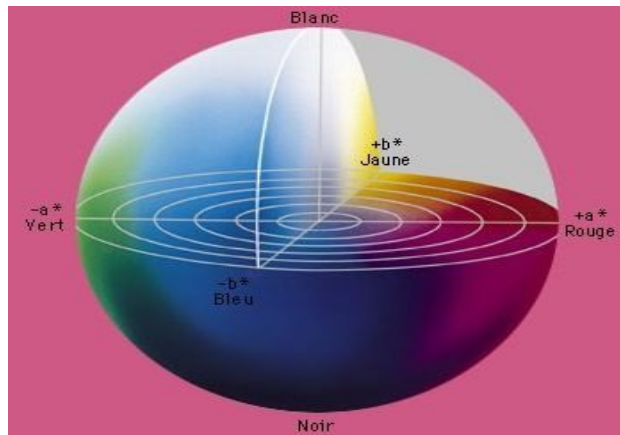


Figure 10: Illustration de l'espace couleurs $L^*a^*b^*$.

6. La segmentation

Dans le contexte de la vision par ordinateur, le traitement d'images se place après les étapes d'acquisition et de numérisation, assurant transformation est obtenue par le passage suivant trois étapes essentielles qui sont : le prétraitement, la segmentation et l'interprétation.

La segmentation d'images est l'opération la plus importante dans un système de traitement des images, car elle est située à l'articulation entre le traitement et l'analyse des images.

6.1 Définition de la segmentation

La segmentation est un traitement de bas niveau qui consiste à créer une partition de l'image en sous-ensembles appelés régions tels qu'aucune région ne soit vide, l'intersection entre deux régions soit vide et l'ensemble des régions recouvre toute l'image.

Une région est un ensemble de pixels connexes ayant des propriétés communes qui les différencient des pixels des régions voisines [6]. La figure (11) présente une image d'origine (a) et une image segmentée (b).

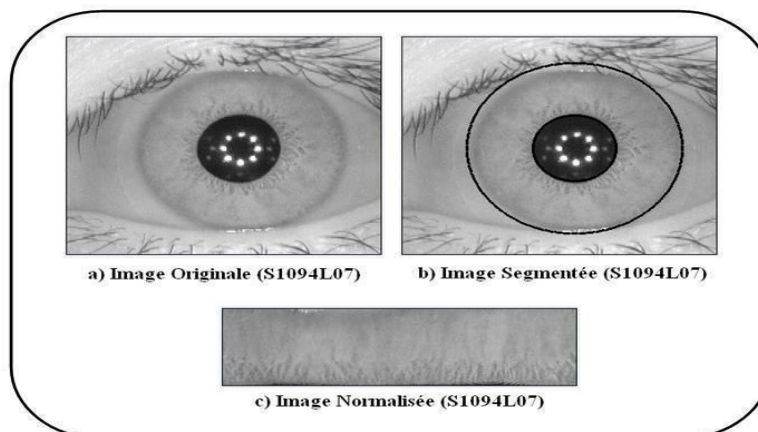


Figure 11: (a) image d'origine, (b) image segmentée

6.2 Objectifs de la segmentation :

Dans le domaine de l'imagerie médicale, la segmentation des images est une tâche cruciale, qui permet d'isoler dans l'image les objets significatifs sur lesquels doit porter l'analyse.

- Fournir des régions homogènes (selon un critère donné).
- Améliorer l'opération de la localisation de manière précise les contours des régions.
- L'étude et l'interprétation des structures anatomiques.
- La réduction de bruit.

6.3 Le choix d'une technique de segmentation

On appelle segmentation d'une image l'opération consistant à identifier les structures d'intérêt dans cette image. On distingue deux types d'approches pour la segmentation : contour ou région. Dans l'approche contour, on cherche à isoler le ou les contours des objets d'intérêt. Le choix est lié à :

- La nature de l'image à segmenter (éclairage, contours, texture, etc.).
- Aux opérations destinées par la segmentation (compression, reconnaissance des formes, Mesures, etc.).
- La forme des objets et aux primitives à extraire (droites, régions, textures, etc.).
- Aux contraintes d'exploitation (temps réel, espace mémoire, etc.).

6.4 Méthode de segmentation

La segmentation d'image ainsi définie est un domaine vaste où l'on retrouve de très nombreuses approches. Généralement on trouve :

- L'approche Contour : on considère que les primitives à extraire sont les lignes de contrastes séparant des régions de niveaux de gris différents et relativement homogènes, ou bien des régions de texture différentes qu'on peut regrouper en trois catégories : les méthodes dérivatives, les méthodes Analytiques et les méthodes déformables.
- L'approche régions : la segmentation d'image par l'approche région consiste à découper l'image en régions. Plusieurs techniques de cette approche sont à distinguer : Croissance de région, Segmentation par classification, Segmentation par division-fusion.
- Segmentation par seuillage

D'après les méthodes de segmentation largement utilisées, nous trouvons la segmentation par seuillage. Qui est de façon générale est la méthode la plus simple pour la segmentation des images.

Les méthodes les plus classiques des approches régions sont les méthodes de seuillage. Ces méthodes essaient de trouver une valeur d'intensité appelée seuil, à partir de l'histogramme de l'image pour séparer les différentes régions. Elles sont applicables avec efficacité sur des images qui présentent un fort contraste entre les régions.

Il existe de très nombreuses méthodes de seuillage d'un histogramme. La plupart de ces méthodes s'appliquent correctement si l'histogramme contient réellement des pics séparés. De plus, ces méthodes ont très souvent été développées pour traiter le cas particulier de la segmentation en deux classes (i.e. passage à une image binaire) et leur généralité face aux cas multi-classes n'est que très rarement garantie. L'une des méthodes de référence dans cette classe est la méthode d'Otsu [11].

Le seuillage est une technique qui repose sur une mesure quantitative d'une grandeur. Il permet de classer les pixels selon un seuil fixé. Si un pixel a une valeur supérieure au seuil fixé, il prendra la valeur 1 (Blanc), et si sa valeur est inférieure au seuil fixé, il prendra la valeur zéro (Noir). Les méthodes de seuillage peuvent être réparties en deux catégories : Seuillage global et seuillage local ou adaptatif.

➤ Méthode d'Otsu

L'une des méthodes les plus classiques des approches régions est la méthode d'Otsu. Cette méthode essaie de trouver une valeur d'intensité appelée seuil, à partir de l'histogramme de l'image pour séparer les différentes régions. Elle est applicable avec efficacité sur des images qui présentent un fort contraste entre les régions c'est à dire l'histogramme contient réellement des pics séparés.

La méthode Otsu a été développée pour traiter le cas particulier de la segmentation en deux classes (passage à une image binaire) et leur généralité face aux cas multi-classes n'est que très rarement garantie.

Mathématiquement le calcul de la nouvelle image en utilisant un seuil déterminé par la méthode Otsu est comme suite :

$$\forall i, j \in N \times M, I(i, j) = \begin{cases} 1 & \text{si } f(i, j) > S, \\ 0, & \text{sinon} \end{cases} \quad (1.1)$$

Avec :

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

- I : Image binarisée
- ff : Fonction de l'image d'origine
- S : seuil de binarisation.

Les algorithmes locaux impliquent généralement plus de calcul et donc ils sont plus lents lors de l'exécution puisqu'ils travaillent sur chaque pixel ou un petit groupe de pixels en choisissant un seuil adaptatif pour chacun.

$$S(i, j) = \frac{\max(i, j) + \min(i, j)}{2} \quad (1.2)$$

Avec :

- (ii, jj) : Seuil à appliquer pour le point (ii, jj) ;
- $\text{Max}(ii, jj)$: Valeur du niveau de gris maximal dans une fenêtre centrée en (ii, jj) de taille $N \times M$;
- $\text{Min}(ii, jj)$: Valeur du niveau de gris minimal dans une fenêtre centrée (ii, jj) de taille $N \times M$;

La figure (12) montre le résultat de l'application Otsu sur une image en niveau de gris :

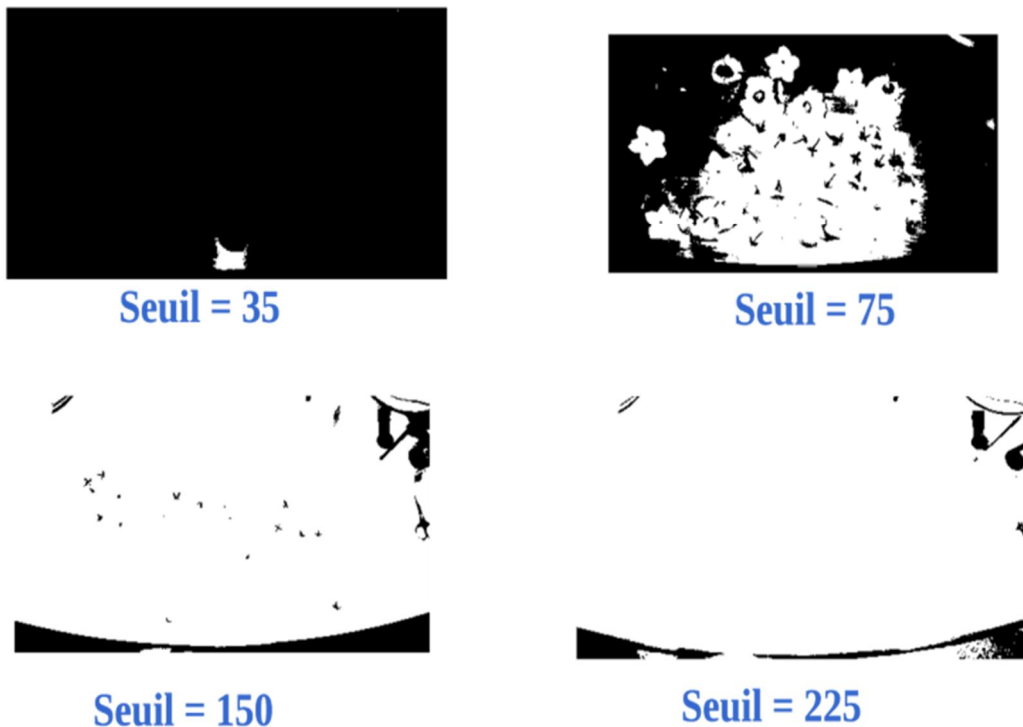


Figure 12: Application de la méthode Otsu pour une image niveau de gris.

7. La détection des contours

Un contour est un ensemble des points d'une image numérique qui correspond à un changement brutal de l'intensité lumineuse [12].

La détection de contour est une étape incontournable à de nombreuses applications de l'analyse d'images. Car elle permet une réduction importante de la quantité d'information relative à une image, tout en préservant des informations structurelles comme les contours et les frontières des images. Les contours constituent en effet des indices riches, au même titre que les points d'intérêts, pour toute interprétation ultérieure de l'image. Les contours dans une image proviennent des :

- Discontinuités de la fonction de réflectance (texture, ombre),
- Discontinuités de profondeur (bords de l'objet) et sont caractérisés par de discontinuités de la fonction d'intensité dans les images.

Ils existent plusieurs modèles de contours : escalier, rampe et toit (figure 13), le plus utilisé est celui en marche d'escalier.



Figure 13: Modèles de contours.

7.1 Objectifs de détection des contours

Le but de la détection de contours est de repérer les points d'une image numérique qui correspondent à un changement brutal de l'intensité lumineuse ; en effet un contour se matérialise par une rupture d'intensité dans l'image suivant une direction donnée. Plusieurs méthodes existent pour détecter cette rupture, les unes plus ou moins complexes, les autres plus ou moins gourmandes en calcul.

Elles s'appliquent en deux étapes :

- La première étape permet de localiser les contours à partir d'un calcul de Gradient ou de Laplacien dans des directions privilégiées tout en quantifiant l'importance du contour.
- La seconde étape va permettre d'isoler les contours du reste de l'image à partir d'un seuillage judicieux.

Plusieurs méthodes permettent de déterminer le Gradient ou le Laplacien d'une image.

Il en est de même des techniques de seuillage. Il existe donc un grand nombre de combinaisons de calcul Gradient et seuillage conduisant à la mise en évidence des contours [13].

7.2 Le détecteur de contour Canny

Le détecteur de contour Canny est sans doute le détecteur de contour le plus connu et le plus utilisé dans tous les domaines de la vision par ordinateur et du traitement d'images.

L'algorithme de détection de contours de Canny, est un processus en plusieurs étapes consistant à :

- Application d'un lissage gaussien à l'image pour aider à réduire le bruit.
- Calcul des gradients d'images G_x et G_y à l'aide du noyau Sobel.
- Application de la suppression du non-maxima pour ne conserver que les maxima-locaux des pixels de magnitude du gradient qui pointent dans la direction du gradient.
- Définition et application des seuils $T_{supérieur}$ et $T_{inférieur}$ pour le seuillage d'hystérésis.

8. La morphologie mathématique

La morphologie mathématique est une théorie essentiellement non linéaire, utilisée en particulier en analyse d'images, dont le but est l'étude des objets en fonction de leur forme, de leur taille, des relations avec leur voisinage (en particulier topologiques), de leur texture, et de leurs niveaux de gris ou de leur couleur. Par les transformations qu'elle propose, elle se situe à différents niveaux du traitement d'images (filtrage, segmentation, mesures, analyse de texture) et fournit ainsi des outils pour la reconnaissance des formes. La morphologie mathématique, développée à l'origine pour l'étude des matériaux poreux, trouve maintenant ses applications dans de nombreux domaines du traitement d'images, aussi bien 2D que 3D, en biologie et cytologie quantitative, en imagerie médicale, en imagerie aérienne et satellitaire, en robotique et vision par ordinateur.

8.1 L'érosion

Lors de cette transformation (figure 16) :

- Les objets de taille inférieure à celle de l'élément structurant vont disparaître,
- Les autres seront ôtés d'une partie correspondant à la taille de l'élément structurant,
- S'ils existent des trous à l'intérieur des objets, ils seront accentués.
- Les objets reliés entre eux vont être séparés.

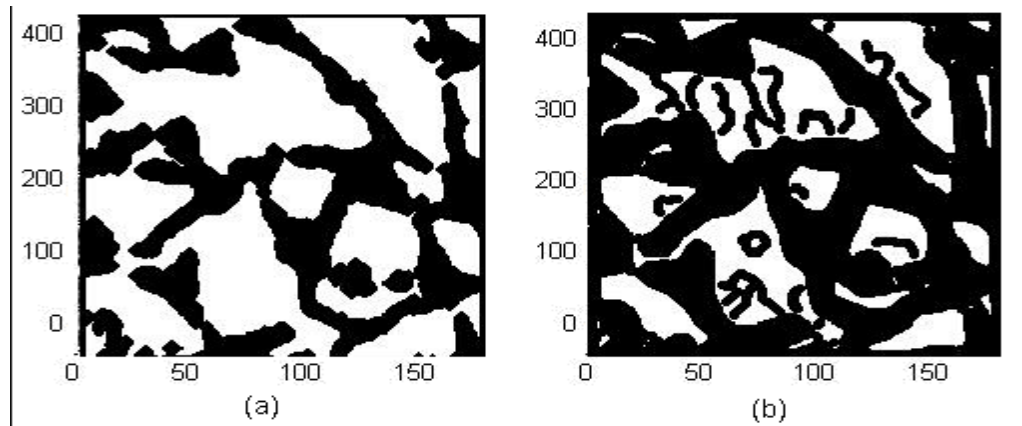


Figure 14: (a) image érodée, (b) image binaire.

8.2 La dilatation

- La dilatation est l'opération duale (ou inverse) de l'érosion. Lors d'une dilatation (figure I.17) :
- Tous les objets vont "grossir" d'une partie correspondant à la taille de l'élément structurant
- S'il existe des trous dans les objets, c'est à dire des "morceaux" de fond à l'intérieur des objets, ils seront comblés.
- Si des objets sont situés à une distance moins grande que la taille de l'élément structurant, ils vont fusionner.

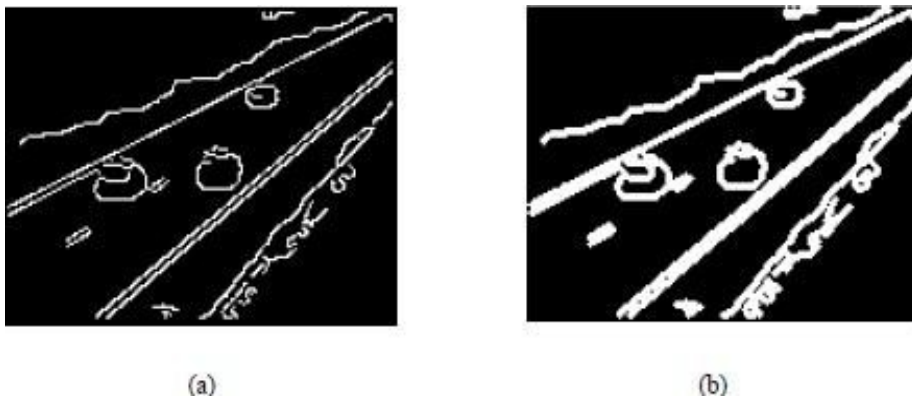


Figure 15: (a) image binaire, (b) image dilatée.

8.3 L'ouverture

Une érosion suivie d'une dilatation s'appelle une ouverture. L'ouverture a pour propriété d'éliminer toutes les parties des objets qui ne peuvent pas contenir l'élément structurant, La figure suivant donne un exemple.

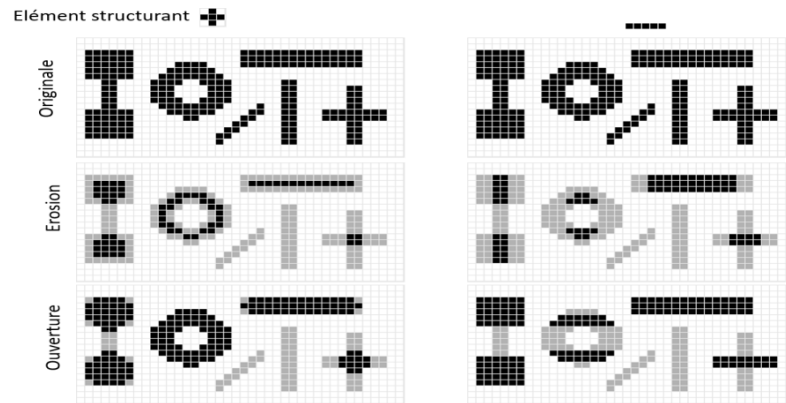


Figure 16: Exemple d'ouvertures avec un élément structurant en croix.

8.4 La fermeture

Une dilatation suivie d'une érosion s'appelle une fermeture. Il a pour propriété de combler tout ce qui est de taille inférieure à l'élément structurant. La figure suivante montre un exemple.

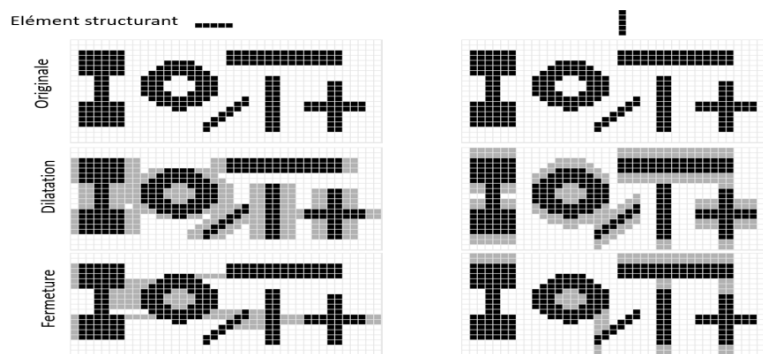


Figure 17: Exemple de fermetures avec un élément structurant horizontal.

9. Conclusion

Nous avons introduit dans ce chapitre « généralités du traitement d'image », les notions de base qui représente l'image et ses caractéristiques puis la compréhension de différentes techniques de traitement d'images et leurs opérateurs. En effet, L'un des buts de ce domaine est la segmentation des images pour l'extraction des caractéristiques et pour détecter la présence de certains objets sous les différentes formes.

Chapitre II

La reconnaissance de plaques d'immatriculation

1. Introduction

L'un des principaux objectifs de l'ordinateur est d'automatiser les tâches habituellement effectuées par l'homme. La plupart des activités humaines reposent sur une faculté importante de notre cerveau : pour voir distinguer entre les formes. Par exemple, reconnaître une pomme d'une orange ; un bon diagnostic d'un mauvais, ou encore un sous-marin d'une baleine. Pour pouvoir automatiser ce genre de tâches, l'ordinateur doit obligatoirement acquérir la faculté de reconnaître les formes et de classer les objets.

La reconnaissance ou classification de forme est une étape qui consiste à reconnaître un objet à partir de sa forme, en attribuant un même label à un ensemble d'objets identiques, cette étape est basée sur l'utilisation d'un descripteur de forme pour faire cette étape, il faut passer sur l'extraction des caractéristiques. Pour cela, il est souvent nécessaire de passer par une phase d'apprentissage qui permet au classifieur, c.à.d. au système chargé de classer les objets, de connaître et d'apprendre les caractères de la plaque d'immatriculation. La reconnaissance de caractère est l'un des problèmes les plus difficiles en reconnaissance de forme.

Dans ce chapitre, nous présentons d'une manière général le domaine de la reconnaissance de formes par la machine et un état de l'art sur quelques systèmes de la lecture automatique de plaques d'immatriculation.

2. Reconnaissance des formes

2.1. Intelligence Artificielle

La reconnaissance d'image, sous-catégorie de la Computer Vision et de l'Intelligence Artificielle, représente un ensemble de méthodes de détection et d'analyse d'images pour permettre l'automatisation d'une tâche spécifique. Il s'agit d'une technologie qui est capable d'identifier des lieux, des personnes, des objets et plusieurs autres types d'éléments au sein d'une image et d'en tirer des conclusions en les analysants.

La reconnaissance photo ou vidéo peut être réalisée à des degrés de précision différents, par rapport au type d'information ou de concept recherché. En effet, un modèle ou algorithme est capable de détecter un élément spécifique, tout comme il peut simplement attribuer une image à une grande catégorie.

Il existe donc différentes "tâches" que la reconnaissance d'image peut effectuer :

- **La classification.** Il s'agit de l'identification de la "classe", autrement dit la catégorie, à laquelle une image appartient. Une image ne peut avoir qu'une seule classe.

- **Le tagging.** Aussi appelé “labellisation”, il s’agit d’une tâche de classification mais à un degré de précision plus élevé. Cela va permettre de reconnaître la présence de plusieurs concepts ou objets au sein d’une image. Il est par conséquent possible d’attribuer un ou plusieurs tags à une image en particulier.
- **La détection.** Cela est nécessaire lorsque l’on souhaite situer un objet dans une image. Une fois la localisation réussie, un rectangle, appelée en anglais bounding box, va encadrer l’objet en question.
- **La segmentation.** Elle représente également une tâche de détection. La segmentation peut situer au pixel près un élément sur une image. Car dans certains cas, la précision ne peut être négligée, tel que pour le développement des voitures autonomes.

2.2. Définitions

La reconnaissance de formes (RDF) est un domaine qui a fortement évolué ces dernières décennies si bien que ses contours sont devenus aujourd'hui plus flous. Il n’est pas simple d’en trouver une définition dans laquelle tous les chercheurs en RDF se reconnaîtront. Si l’on s’intéresse à l’histoire moderne de la RDF, disons depuis les années 1980, et à ses développements récents liés bien entendu au développement des machines informatiques, on constate que le domaine de la RDF est attaché à l’automatisation de tâches perceptives. Donc la reconnaissance des formes est un ensemble de techniques visant à détecter la présence de formes, de motifs ou de régularités dans des données. Cette branche de l’intelligence artificielle fait largement appel aux techniques d'apprentissage automatique et aux statistiques.

2.3. Méthodes de reconnaissance de formes

Un système de reconnaissance de formes peut être réalisé via l’une des méthodes citées ci-dessus :

- Mise en correspondance de graphes.
- Méthode Bayésienne.
- Estimation Paramétrique.
- Réseau de neurones.
- Support Vector Machine (SVM).
- ...

La reconnaissance de motifs peut être effectuée au moyen de divers algorithmes d'apprentissage automatique tels :

- Un réseau de neurones.
- Une analyse statistique.
- L'utilisation de modèles de Markov cachés.
- Une recherche d'isomorphisme de graphes ou sous-graphes.

Les formes recherchées peuvent être des formes géométriques, descriptibles par une formule mathématique, telles que :

- Cercle ou ellipse.
- Courbes de Bézier, splines
- Droite.

Elles peuvent aussi être de nature plus complexe :

- Lettre
- Chiffre.
- Empreinte digitale.

Les algorithmes de reconnaissance peuvent travailler sur des images en noir et blanc, avec en blanc les contours des objets se trouvant dans l'image. Ces images sont le fruit d'algorithmes de détection de contours. Ils peuvent aussi travailler sur des zones de l'image prédéfinies issues de la segmentation de l'image

3. Schéma général d'un système de Reconnaissance des formes

La majorité des systèmes de Reconnaissance des formes ont le schéma de fonctionnement suivant (voir figure 18).

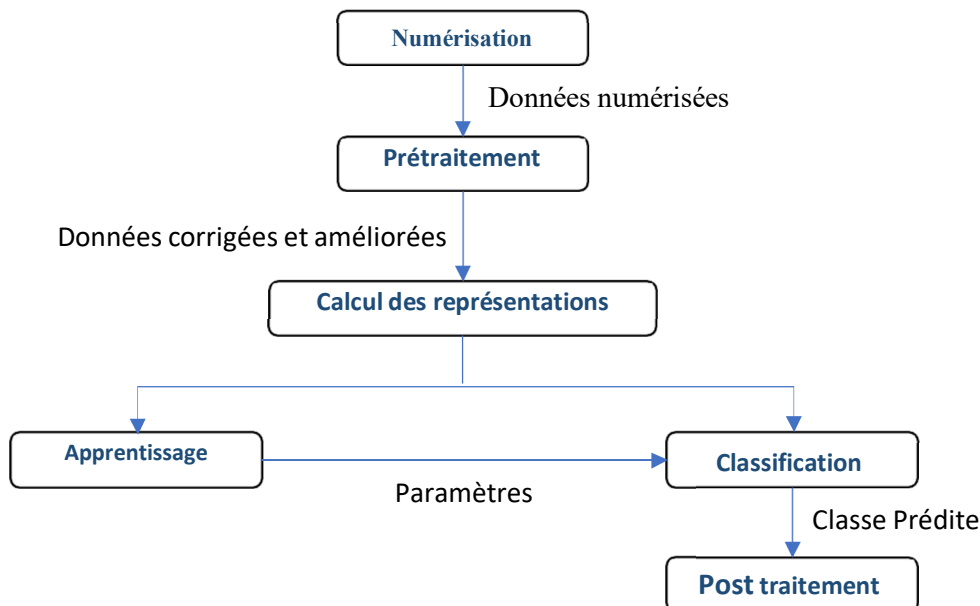


Figure 18: Schéma général d'un système de reconnaissance des formes

3.1. Préparation des données

3.1.1. Numérisation

A partir des informations du monde physique, construire une représentation des données directement manipulable par la machine. Des capteurs (microphone, caméra, instruments de mesure) convertissent les signaux reçus du monde réel en une représentation numérique discrète. L'espace résultant, appelé "espace de représentation", a une dimension r très grande lui permettant de disposer du maximum d'informations sur les formes numérisées.

3.1.2. Prétraitement

Consiste à sélectionner dans l'espace de représentation l'information nécessaire au domaine d'application. Cette sélection passe souvent par l'élimination du bruit, la normalisation des données, ainsi que par la suppression de la redondance. Le nouvel espace de représentation a une dimension r' très inférieure à r mais demeure un espace de grande dimension et contient des informations encore assez primitives.

3.1.3. Calcul des représentations

Il s'agit de la phase finale de la préparation des données. Elle fournit un certain nombre de "caractéristiques" ou "paramètres" (fameux attributs) en utilisant des algorithmes de sélection et/ou d'extraction d'attributs. Les attributs étant limités en nombre, "l'espace des paramètres" ainsi obtenu est de dimension p très petite par rapport à r' .

3.1.4. Apprentissage

L'apprentissage ou entraînement, est une partie importante du système de reconnaissance. Le classificateur étant généralement une fonction paramétrique, l'apprentissage va permettre d'optimiser les paramètres du classificateur pour le problème résoudre, en utilisant des données d'entraînement. Lorsque les données d'entraînements ont préalablement classé, l'apprentissage est dit "supervisé", sinon il est "non supervisé"

3.1.5. Classification

Cette phase est le noyau de la Reconnaissance des formes. En utilisant les modèles (paramètres) obtenus lors de l'apprentissage, le classificateur assigne à chaque forme inconnue sa ou ses formes les plus probables.

3.1.6. Post traitement

Cette phase a pour but de corriger les résultats de la classification en utilisant des outils spécifiques au domaine d'application. Par exemple pour un système de reconnaissance de textes manuscrits, le classificateur se charge de classer chaque caractère séparément, alors que le post traitement applique un correcteur orthographique sur tout le texte pour valider et éventuellement corriger le résultat de la classification. Bien que facultative, cette phase permet d'améliorer considérablement la qualité de la reconnaissance.

4. Etat de l'art sur les systèmes RAPI

De nos jours, il existe de nombreux systèmes de la reconnaissance de plaques d'immatriculation, tels que : AutoVu Est le système de reconnaissance de plaques d'immatriculation (RAPI) sur IP de Security Center, la plate-forme de sécurité unifiée de Genetec (Synergis). [14]. LAPI ENGINE est un produit autonome dédié principalement à la traçabilité de véhicules et pouvant s'adapter à un large éventail d'applications. Par exemple, il s'intègre facilement à des solutions "métier" = de contrôle d'accès, de pesée industrielle ou des logiciels de vidéosurveillance, de gestion de quais, péage, stations-services, etc..., LAPI ENGINE intègre de nombreuses innovations algorithmiques [15]. SeeTec Est un module d'extension de SeeTec Cayuga qui permet la reconnaissance automatique de plaques d'immatriculation de véhicules à l'arrêt ou en mouvement [16]. Asia Vision Technology Limited (AVT) Est le premier fournisseur mondial de solutions technologiques et de solutions de gestion intelligente des véhicules et des conteneurs. AVT a été un pionnier dans le développement et la fourniture de la technologie de reconnaissance optique de caractères [17].

Nous avons trouvé beaucoup systèmes RAPI, et la plupart de ces systèmes sont basés sur des propriétés différentes. Des techniques utilisent des règles simples basées sur des méthodes de classification pour la détection et la reconnaissance des plaques d'immatriculation

5. Méthodes de classification

Les méthodes de classification ont pour but d'identifier les classes auxquelles appartiennent des objets à partir de certains paramètres descriptifs. Elles s'appliquent à un grand nombre d'activités humaines et conviennent en particulier au problème de la prise de décision automatisée. La procédure de classification sera extraite automatiquement à partir d'un ensemble d'exemples.

Parmi les méthodes qui ont été développées dans ce contexte (machine d'apprentissage), nous distinguons par exemple les réseaux de neurones et les machines à vecteurs du support

(SVM : Support Vector Machines, en anglais) ainsi que K plus proches voisins (K Nearest Neighbors)

Les SVM ont été utilisés dans différents domaines d'expertise, tels que les diagnostics médicaux, le traitement d'image, le traitement du signal, etc. Dans ce chapitre, nous présentons les aspects théoriques de la méthode SVM.

5.1. Machine à Vecteurs de Support (SVM)

L'algorithme des machines à vecteurs de support a été développé dans les années 90 par le russe *Vladimir Vapnik*. Initialement, les SVM ont été développés comme un algorithme de classification binaire supervisée. Il s'avère particulièrement efficace de par le fait qu'il peut traiter des problèmes mettant en jeu de grands nombres de descripteurs, qu'il assure une solution unique (pas de problèmes de minimum local comme pour les réseaux de neurones) et il a fourni de bons résultats sur des problèmes réels.

L'algorithme sous sa forme initiale revient à chercher une frontière de décision linéaire entre deux classes, mais ce modèle peut considérablement être enrichi en se projetant dans un autre espace permettant d'augmenter la séparabilité des données. On peut alors appliquer le même algorithme dans ce nouvel espace, ce qui se traduit par une frontière de décision non linéaire dans l'espace initial.

5.2. Définition de SVM

Les SVM sont un ensemble de techniques d'apprentissage supervisé de nouvelle génération basés sur les progrès récents de la théorie de l'apprentissage statistique, ils sont basés sur le principe de la maximisation de la marge (séparation des classes [11]). Principalement conçues pour résoudre ces problèmes.

- Des problèmes de discrimination, permettant de décider à quelle classe appartient un échantillon.
- Cependant, ils peuvent aussi résoudre des problèmes de régression visant à prédire la valeur numérique d'une variable. C'est-à-dire de prédiction d'une variable continue en fonction d'autres variables, Comme c'est le cas par exemple dans de la prédiction de consommation électrique en fonction de la période de l'année, de la température, etc.

La résolution de ces deux problèmes se base sur la construction d'une fonction h qui, à un vecteur d'entrée x , fait correspondre une sortie y ($y = h(x)$) [11].

5.3. Apprentissage automatique et SVM

La notion d'apprentissage étant importante, nous allons commencer par effectuer un rappel. L'apprentissage par induction permet d'arriver à des conclusions par l'examen d'exemples particuliers. Il se divise en apprentissage supervisé et non supervisé. Le cas qui concerne les SVM est l'apprentissage supervisé [12].

Les exemples particuliers sont représentés par un ensemble de couples d'entrée/sortie. Le but est d'apprendre une fonction qui correspond aux exemples vus et qui prédit les sorties pour les entrées qui n'ont pas encore été vues. Les entrées peuvent être des descriptions d'objets et les sorties la classe des objets donnés en entrée [12].

5.4. Principe de fonctionnement général de SVM

5.4.1. Notions de base

5.4.1.1. Hyperplan

On appelle hyperplan séparateur un hyperplan qui sépare les deux classes, en particulier il sépare leurs points d'apprentissage. Comme il n'est en générale pas possible d'en trouver un, on se contentera donc de chercher un hyperplan discriminant qui est une approximation au sens d'un critère à fixer (maximiser la distance entre ces deux classes) [11].

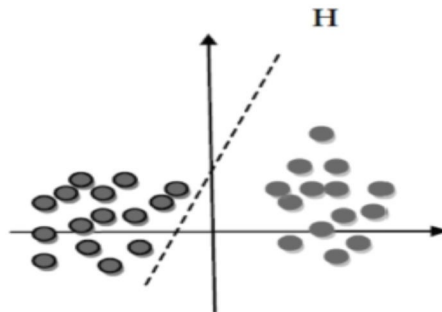


Figure 19: Exemple d'un hyperplan sépare deux ensembles de points

5.4.1.2. Vecteurs de support

Ce sont les points de la frontière entre les deux classes des données parmi l'ensemble total d'apprentissage, ces points sont appelés vecteurs de support [11].

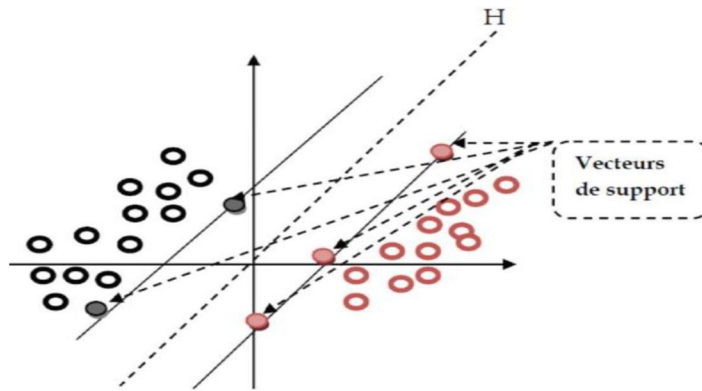


Figure 20: Exemple de vecteurs de support.

5.4.1.3. Marge

Il existe une infinité d'hyperplans capable de séparer parfaitement les deux classes d'exemples. Le principe des SVM est de choisir celui qui va maximiser la distance minimale entre l'hyperplan et les exemples d'apprentissage (i.e. la distance entre l'hyperplan et les vecteurs de support), cette distance est appelée la marge [11].

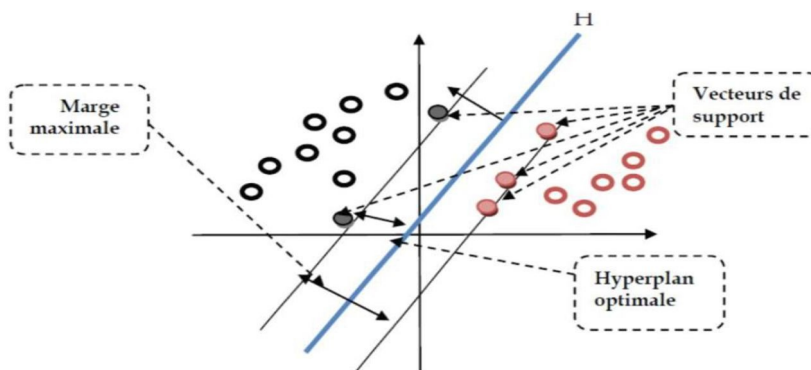


Figure 21: Exemple de marge maximale (hyperplan valide)

Linéarité et non-linéarité

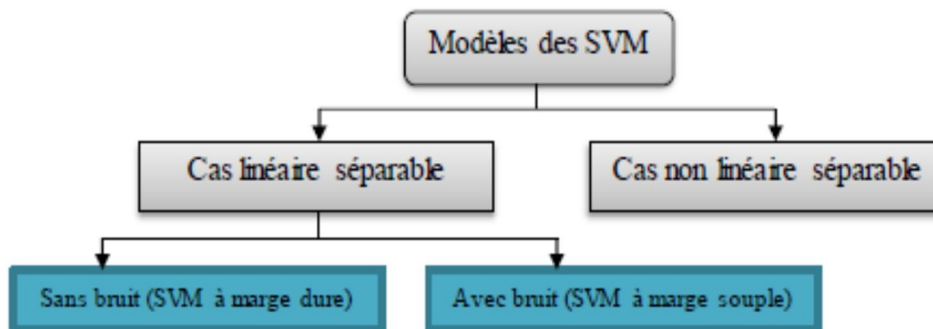


Figure 22: Schéma spectacles les différents modèles des SVM.

Parmi les modèles des SVM, on constate les cas linéairement séparables et les cas non linéairement séparables. Les premiers sont les plus simples de SVM car ils permettent de trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels il n'y a pas de séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables

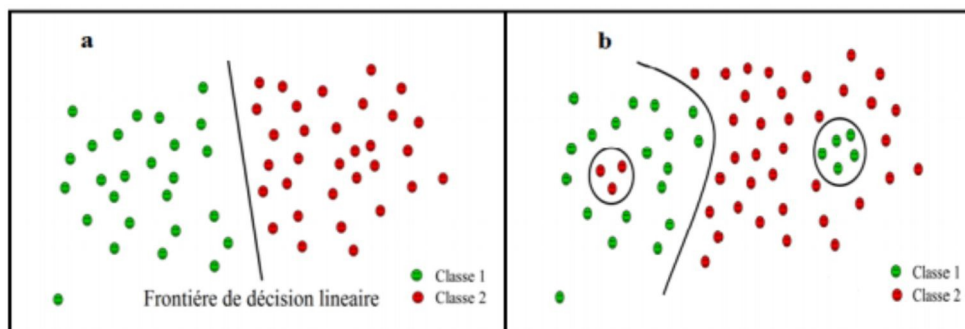


Figure 23: a) Cas linéairement séparable, b) Cas non linéairement séparable

5.4.2. Cas linéairement séparable

Ces cas sont les plus simples de SVM car ils permettent de trouver facilement le classificateur linéaire et dans un modèle linéaire, on a $f(x)=w.x+b$.

- L'hyperplan séparateur (frontière de décision) a donc pour équation $w.x+b=0$
- La distance d'un point au plan est donnée par $d(x)=\frac{|w.x+b|}{\|w\|}$
- L'hyperplan optimal est celui pour lequel la distance aux points les plus proches (marge) est maximale.

5.4.3. Cas non linéairement séparable

Il existe néanmoins des cas où on ne peut pas classer les entrées de façon linéaire [10].

Pour surmonter les inconvénients des cas non linéairement séparables, l'idée des SVM est de changer l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace. On va donc avoir un changement de dimension. Cette nouvelle dimension est appelé « espace de ré-description ».

En effet, intuitivement, plus la dimension de l'espace de ré-description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée. Ceci est illustré par le schéma suivant.

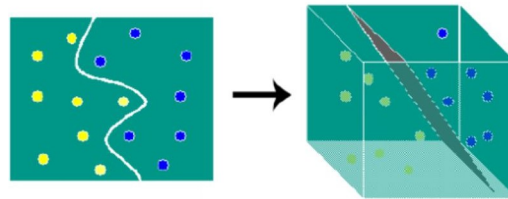


Figure 24: Exemple de changement de l'espace de données

On a donc une transformation d'un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de ré-description de plus grande dimension. Cette transformation non linéaire est réalisée via une fonction noyau.

En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants : polynomiale, gaussien, sigmoïde et laplacien[09].

5.4.3.1. Fonction noyau (kernel)

Dans les cas linéaires, on pouvait transformer les données dans un espace où la classification serait plus aisée. Dans ce cas, l'espace de re-description utilisé le plus souvent est \mathbb{R} (ensemble des nombres réels). Il se trouve que pour des cas non linéaires, cet espace ne suffit pas pour classer les entrées. On passe donc dans un espace de grande dimension

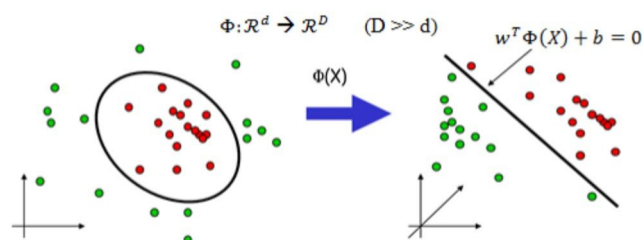


Figure 25: Illustration de passage à \mathbb{R}^3

Le passage dans $F = \mathbb{R}^3$ (i.e. $D=3$). Rend possible la séparation linéaire des données. On doit donc résoudre :

$$\left\{ \begin{array}{l} \max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \varphi(x_i) \varphi(x_j) \\ \forall i, 0 \leq \alpha_i \leq c \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{array} \right.$$

Et la solution à la forme : $f(x) = \sum \alpha_i y_i \varphi(x_i) \cdot \varphi(x) + b$

Le problème et sa solution ne dépendent que du produit scalaire $\varphi(x_i)\varphi(x')$. Plutôt que de choisir la transformation non-linéaire : $X \rightarrow F$, on choisit une fonction K (nombres réels) appelés fonction noyau.

Elle représente un produit scalaire dans l'espace de représentation intermédiaire. Du coup k est linéaire (ce qui nous permet de faire le rapprochement avec le cas linéaire des paragraphes précédents). Cette fonction traduit donc la répartition des exemples dans cet espace $k(x, x') = \varphi(x) \cdot \varphi(x')$. Lorsque k est bien choisie, on n'a pas besoin de calculer la représentation des exemples dans cet espace pour calculer φ .

Exemples de noyaux :

- Linéaire $K(x, x') = x \cdot x'$.
- Polynomial $K(x, x') = (x \cdot x')^d$ ou $(c + x \cdot x')^d$.
- Gaussien $K(x, x') = e^{-\|x - x'\|^2 / \sigma}$.
- Laplacien $K(x, x') = e^{-\|x - x'\| / \sigma}$.

5.4.3.2. Condition de Mercer

Une fonction k symétrique est un noyau si $(k(x_i, x_j))_{i,j}$ est une matrice définie positive. Dans ce cas, il existe un espace F et une fonction \square tels que $k(x, x') = \square(x) \times \square(x')$

Problèmes :

- Cette condition est très difficile à vérifier.
- Elle ne donne pas d'indication pour la construction de noyaux.
- Elle ne permet pas de savoir comment est.

5.5. SVMs multi-classe

La plupart des problèmes de classification sont à multi classe, les SVMs ont été conçus initialement pour résoudre des problèmes de classification à deux classes, en revanche d'autres méthodes ont permis l'extension des approches SVM pour traiter ce type de problème.

5.5.1. Un contre tous

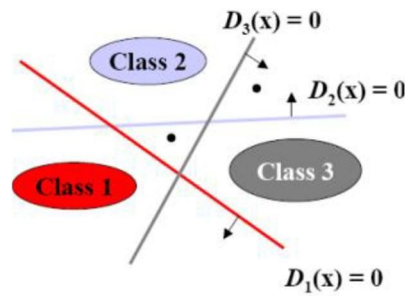


Figure 26: Trois classes séparées par la méthode d'un contre tous avec des séparateurs linéaires

Pour obtenir des classifieurs pour les M classes, on définit un ensemble de classifieur binaire (à deux classes) $f_1 \dots f_m$, chaque classifieur sépare la classe i de toute les autres classes produisant ainsi M fonction de décision de type sgn , la valeur $+1$ est attribué aux donnée appartenant à la classe i et la valeur -1 et attribué aux données appartenant aux classes restantes:

$$f^j(x) = sgn(g^j(x))$$

$$\text{Avec } g^j(x) = \sum_{i=1}^m y_i \alpha_i^j k(x, x_i) + b^j$$

La valeur maximale de $f(x)$ désigne l'argument j de f comme classe d'un exemple x_i , si la différence entre deux grand $g^j(x)$ est inferieur a un seuil θ lors de la classification d'un exemple x , cet exemple sera rejeté est ne sera affecté à aucune classe, cet écartement de l'exemple est appelé *Reject decision*. (figure 23)

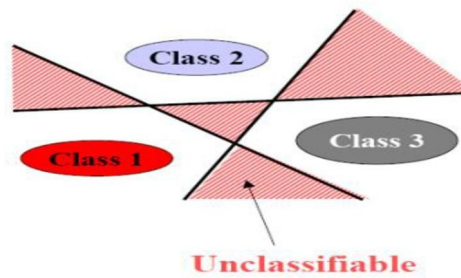


Figure 27: Les surfaces rouges représente le « Reject decision »

5.5.2. Classification par paire

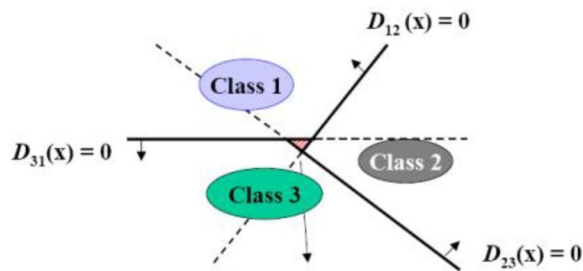


Figure 28: Classification multi-classe par paire

Connu sous le nom de classification un contre un (one versus one), dans cette méthode on détermine un classifieur pour chaque pair de classe, pour M classe on aura $\frac{M.(M-1)}{2}$ classifieurs binaire (le nombre de fonctions de décision est égale aux nombre de classifieurs), le nombre des séparateurs est habituellement plus grand a celui des séparateurs sur la méthode un contre tous, pour un $M=10$ on a besoin de 45 classifieurs binaires, chaque classe effectue un vote pour affecter un point x , la classe majoritaire après vote sera celle à qui le point est affecté.

D'après (figure 24) on remarque que la région des points non classifiable existe toujours.

5.6. Architecture générale d'une machine à vecteur support

Une machine à vecteur support, recherche à l'aide d'une méthode d'optimisation, dans un ensemble d'exemples d'entraînement, des exemples, appelés vecteurs support, qui caractérisent la fonction de séparation. La machine calcule également des multiplicateurs associés à ces vecteurs.

Les vecteurs supports et leurs multiplicateurs sont utilisés pour calculer la fonction de décision pour un nouvel exemple. Le schéma de la relation de (figure 29) résume l'architecture générale d'une SVM dans le cas de la reconnaissance des chiffres manuscrits

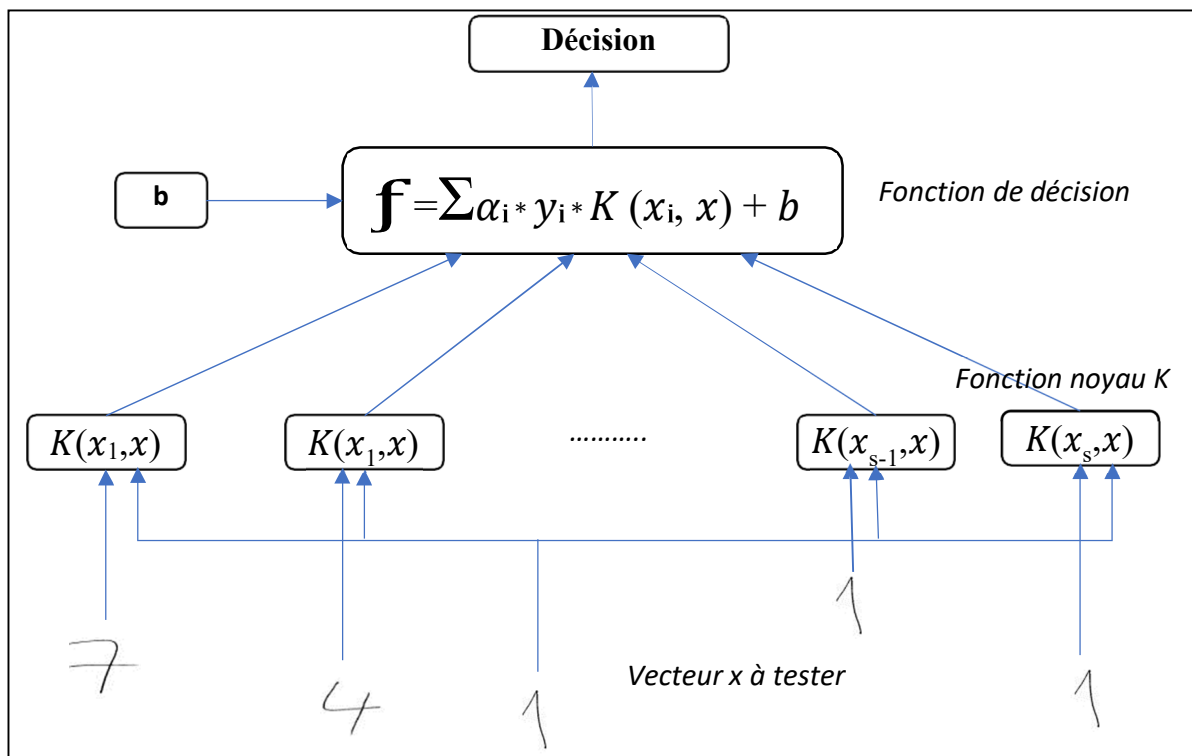


Figure 29: Architecture d'une machine à vecteur support

6. Conclusion

Dans ce chapitre, nous avons tenté de présenter de manière générale la reconnaissance d'image (formes/motifs) par machine, après nous exposons le concept de système d'apprentissage introduit par Vladimir Vapnik , les Machines à Vecteurs de Support. Nous avons donné une vision générale et une vision purement mathématique des SVM.

Cette méthode de classification est basée sur la recherche d'un hyperplan qui permet de séparer au mieux des ensembles de données. Nous avons exposé les cas linéairement séparables et les cas non linéairement séparables qui nécessitent l'utilisation de fonction noyau (kernel) pour changer d'espace. Cette méthode est applicable pour des tâches de classification à deux classes, mais il existe des extensions pour la classification multi classe. Nous nous sommes ensuite intéressés aux différents domaines d'application. Il existe des extensions que nous n'avons pas présentées, parmi lesquelles l'utilisation des SVM pour des tâches de régression, c'est-à-dire de prédiction d'une variable continue en fonction d'autre variable.

Chapitre III

Développement et expérimentation du système RPI

1. Introduction

L'un des principaux objectifs de la détection et de la reconnaissance automatique des plaques d'immatriculation des voitures est de réduire la main d'œuvre, les coûts et les délais et de rendre le processus rapide et hautement disponible. Il est nécessaire d'identifier et de lire le numéro de plaque d'immatriculation du véhicule à partir d'une image capturée par les appareils de surveillance.

Il est nécessaire de convertir l'image au format texte avant que d'autres parties du système puissent l'utiliser, sans intervention humaine.

a. Environnement matériel :

Afin de mener à bien ce projet, il a été mis à notre disposition un ordinateur Dell avec les caractéristiques suivantes :

- Processeur : Intel(R) Core(TM) i3-3217U CPU @ 1.80GHz 1.80 GHz
- RAM : 8,00 Go
- Disque Dur : 1To
- Type de système : système d'exploitation 64 bit.
- OS : Microsoft Windows 10

b. Environnement logiciel :

- Python

Il existe un très grand nombre de langages de programmation, chacun avec ses avantages et ses inconvénients. Il faut bien en choisir un. D'après les langages utilisés (orienté) pour le traitement d'images on trouve le MATLAB, C++, JAVA, PYTHON...etc.

Nous avons finalement décidé d'adopter Python dans notre projet, langage très moderne à la popularité grandissante.

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

Les domaines d'application naturels de Python incluent entre autres:

- L'apprentissage de la programmation objet.
- Les scripts d'administration système ou d'analyse de fichiers textuels.
- Tous les développements liés à L'internet et en particulier au Web: scripts CGI, navigateurs Web, moteurs de recherche, agents intelligents, objets distribués...

- L'accès aux bases de données (relationnelles).
- La réalisation d'interfaces graphiques utilisateurs .
- Le traitement des images (filtrage, segmentation ...etc.)

2. Les bibliothèques :

2.1. OpenCV

Pour ce qui concerne la partie de traitement d'images nous avons utilisé une bibliothèque appelée OpenCv (Open Source Computer Vision), comme son nom le suggère, OpenCv est une bibliothèque dédiée totalement à la vision par ordinateur ce qui signifie traitements des images, matrices, vidéos, ..., Données visuelles.

2.2. Pytesseract

Python-tesseract est un outil de reconnaissance optique de caractères (OCR) pour Python. Autrement dit, il reconnaîtra et « lira » le texte intégré dans les images.

Python-tesseract est un wrapper pour le moteur Tesseract-OCR de Google. Il est également utile en tant que script d'invocation autonome pour tesseract, car il peut lire tous les types d'images pris en charge par les bibliothèques d'imagerie Pillow et Leptonica, notamment jpeg, png, gif, bmp, tiff et autres. De plus, s'il est utilisé comme script, Python-tesseract imprimera le texte reconnu au lieu de l'écrire dans un fichier.

2.3. Matplotlib

Matplotlib est la troisième bibliothèque que nous couvrons dans une série sur les bibliothèques de science des données. Bienvenue dans le cinquième article de la série d'articles sur les bibliothèques de science des données en Python.

Dans cet article, nous vous donnerons un aperçu rapide de la bibliothèque Matplotlib permettant de visualiser des données à l'aide du langage Python, et nous vous donnerons une explication simplifiée en plus de quelques exemples pratiques.

2.4. Rembg

Le module rembg en Python est une bibliothèque utilisée pour supprimer l'arrière-plan des images. Il est basé sur l'algorithme Rembg, qui utilise un réseau neuronal pour effectuer la tâche de suppression de l'arrière-plan. L'algorithme est entraîné pour identifier et séparer les objets du premier plan de leur arrière-plan dans les images, ce qui donne des images avec un arrière-plan transparent.

2.5. Keras

Keras est une bibliothèque Python open source gratuite, puissante et facile à utiliser,

permettant de développer et d'évaluer des modèles d'apprentissage en profondeur.

Il fait partie de la bibliothèque TensorFlow et vous permet de définir et d'entraîner des modèles de réseaux de neurones en quelques lignes de code seulement.

Dans ce tutoriel, vous découvrirez comment créer votre premier modèle de réseau neuronal d'apprentissage profond en Python à l'aide de Keras.

Démarrez votre projet avec mon nouveau livre Deep Learning With Python, comprenant des didacticiels étape par étape et les fichiers de code source Python pour tous les exemples.

2.6. TensorFlow

TensorFlow est une bibliothèque open source qui utilise le langage Python pour le calcul numérique, ce qui rend l'apprentissage automatique plus rapide et plus facile. En général, l'apprentissage automatique est un système complexe, mais la mise en œuvre de modèles d'apprentissage automatique est devenue moins difficile qu'auparavant, grâce à l'apprentissage automatique. des cadres d'apprentissage - tels que TensorFlow de Google - qui facilitent la tâche. Le processus d'obtention de données et de modèles de formation sert aux prédictions et contribue à améliorer les résultats futurs.

2.7. Scikit-Learn

Scikit-Learn (également appelée sklearn) est une bibliothèque open source en Python dédiée à l'apprentissage automatique (machine learning). Elle offre une large gamme d'outils et d'algorithmes pour la classification, la régression, le clustering, la préparation des données, la sélection de modèles, et bien plus encore. Scikit-Learn est largement utilisée dans la communauté de l'apprentissage automatique en raison de sa simplicité et de sa cohérence, ce qui en fait un choix populaire pour les projets d'apprentissage automatique.

La bibliothèque Scikit-Learn offre plusieurs fonctionnalités telles que :

- Classification et régression : Scikit-Learn propose une variété d'algorithmes pour la classification (par exemple, SVM, régression logistique, arbres de décision) et la régression (par exemple, régression linéaire, régression Lasso, régression Ridge).
- Clustering : Vous pouvez effectuer la segmentation des données en utilisant des algorithmes de clustering tels que K-Means, DBSCAN, et hierarchic clustering.
- Prétraitement des données : Scikit-Learn fournit des outils pour la normalisation, la

standardisation, la gestion des valeurs manquantes et la sélection de caractéristiques.

- Évaluation des modèles : Vous pouvez évaluer la performance de vos modèles en utilisant diverses métriques telles que l'exactitude, la précision, le rappel, le F-score, etc.
- Validation croisée : Scikit-Learn propose des outils pour effectuer une validation croisée afin d'estimer la performance de votre modèle.
- Réglage des hyperparamètres : Vous pouvez utiliser la recherche d'hyperparamètres pour optimiser les paramètres de votre modèle.

2.8. joblib

Joblib est une bibliothèque Python permettant d'exécuter des tâches gourmandes en calcul en parallèle. Il fournit un ensemble de fonctions permettant d'effectuer des opérations en parallèle sur de grands ensembles de données et de mettre en cache les résultats de fonctions coûteuses en calcul.

3. Plaque D'immatriculation Algérienne :

La plaque d'immatriculation en Algérie est composée de dix chiffres, voir figure 30



Figure 30 : Plaque d'immatriculation algérienne

Nous commençons par la droite :

« 19 » Correspond du code de la wilaya d'immatriculation.

« 15 » Correspond a l'année de mise en marche de véhicule.

« 1 » Correspond au type de véhicule, ici c'est 1 car c'est une véhicule léger, « 3 » pour le poids lourd.

« 01813 » Correspond au numéro de dossier du véhicule.

4. Le Système De RPI Propose

Le système conçu dans le cadre de ce projet est composé de les étapes suivante : étape

d'acquisition d'image, étape de prétraitement de l'image comprenant la conversion d'un format d'image en un autre, le filtrage et la binarisation d'image, la troisième étape est la localisation et l'extraction de la plaque d'immatriculation, ainsi l'étape suivante est la segmentation et étiquetage des caractères de la PI. L'étape dernière consiste à identifier les caractères.

L'organigramme du système de RPI est illustré à la figure 31.

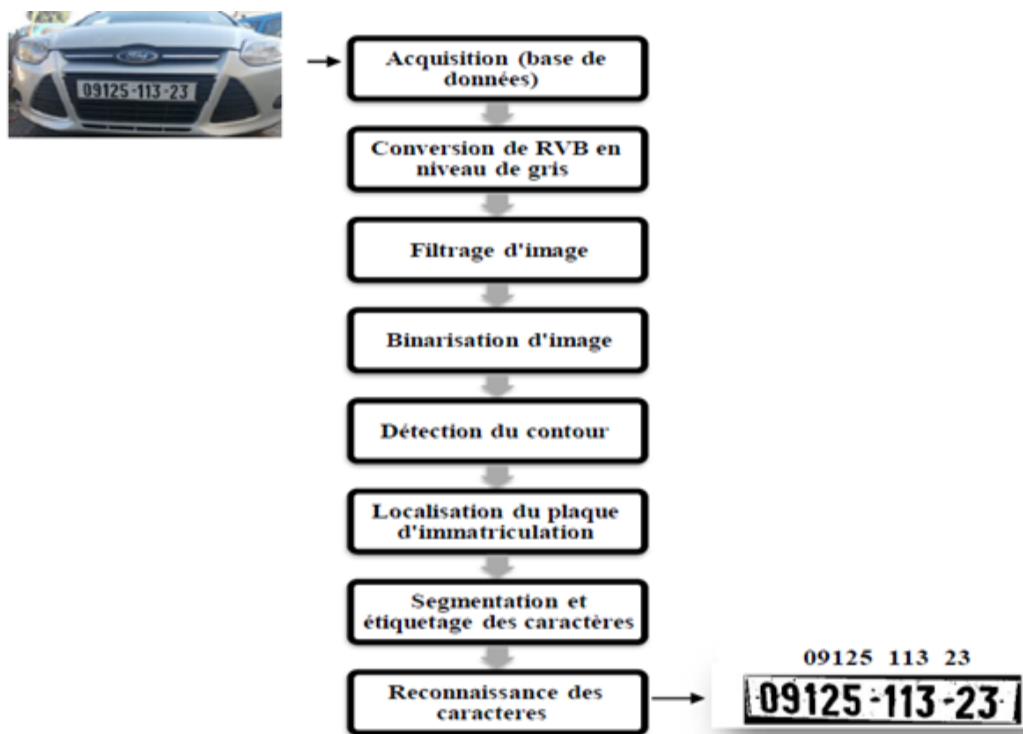


Figure 31 : Organigramme du système de RPI

4.1. Acquisition :

La première étape consiste à capturer une image à l'aide d'un appareil photo, ces images seront prises et stockées au format JPEG, l'image d'entrée est de taille 960 x 720.



Figure 32 : Image originale

4.2. niveau de gris :

Afin de réduire la luminance de l'image couleur, la conversion en niveaux de gris est appliquée (on utilisant l'équation 31. Cela permet de réduire le bruit de l'image dans une certaine mesure et facilite également le traitement ultérieur de l'image. La figure 33 montre l'image d'un véhicule à laquelle a été appliqué le filtre en niveaux de gris.



Figure 33 : Conversion l'image en niveaux de gris, (a) : Image original, (b) : Image en niveaux de gris

4.3. Filtrage d'image :

Une fois l'image convertie, elle est soumise à un processus de lissage afin d'éliminer le bruit. Dans ce cas, le filtrage médian a été utilisé. La taille du voisinage utilisé est de 5×5 . Le résultat de ce processus est une image avec moins de bruit et un meilleur lissage. Le résultat est illustré à la figure 34.



Figure 34 : Filtrage d'image, (a) : Image en niveau de gris, (b) : Image filtrée

4.4. Binarisation d'image :

Cette étape consiste à convertir l'image en réduisant les informations contenues dans l'image de 256 nuances de gris à 2 : noir et blanc, cela permet de faciliter la reconnaissance des caractères.



Figure 35 : Binarisation d'image, (a) : Image originale, (b) : Image binaire

4.5. Détection du contour :

Une fois l'image convertie en niveaux de gris et lissée à l'aide d'un filtre numérique médian, un algorithme de détection des contours a été utilisé pour rechercher les contours d'une image en appliquant la méthode de prewitt. Elle produit les bords aux points où le gradient d'une image est le maximum. Cet algorithme donne le détecteur de contour optimal avec un taux d'erreur très faible. La figure (9) montre l'image après effet la détection de contour prewitt.



Figure 36 : Détection des contours

4.6. Localisation de la plaque d'immatriculation :

C'est l'une des sections importantes. Comme mentionné précédemment, l'image est soumise à plusieurs processus. La figure 37. montre l'algorithme de localisation la région de la plaque d'immatriculation

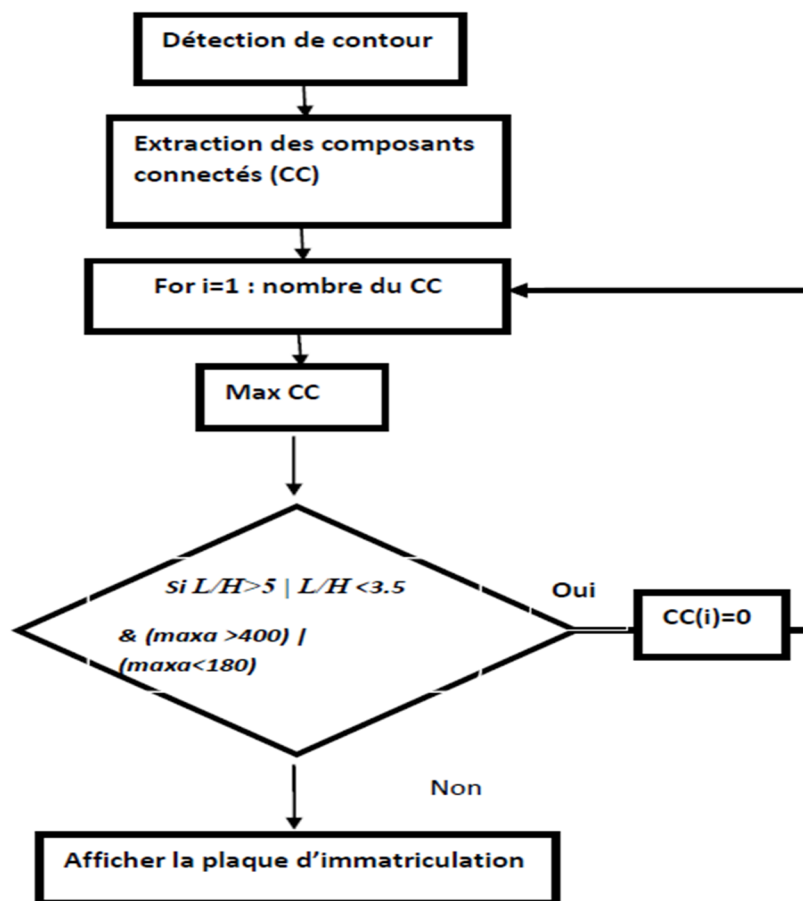


Figure 37 : Algorithme de localisation de la plaque

Ou L : Largeur de la PI, H : Hauteur de la PI, maxa : Surface de la PI

Après la détection des contours de l'image, l'étape suivante consiste à étiqueter les régions connectées. Dans notre cas, nous indiquons également que les régions connectées

doivent avoir au moins 8 objets connectés. Et classées les avec ses coordonnées.



Figure 38 : Localisation de la plaque, (a) : Détection de contour, (b) : Localisation la région de la plaque

Comme nous l'avons vu dans l'image précédente, la région de la plaque d'immatriculation est celle qui a la plus grande surface.



Figure 39: Extraction de plaque d'immatriculation

4.7. Segmentation et étiquetage des caractères:

Après avoir localisé la région de la plaque d'immatriculation, vient l'étape de la segmentation qui est le processus par lequel nous séparons chaque caractère de la plaque d'immatriculation afin de les traiter un par un.

- Différentes méthodes peuvent être utilisées pour réaliser le processus de segmentation, notre approche consiste à créer des cadres de sélection autour de chaque caractère.
- Afin d'exclure les objets connectés qui sont soit trop petits soit trop grands, une condition relative à la taille minimale et maximale d'un cadre de sélection est établie.
- Après avoir créé ces boîtes englobantes, des sous-images sont générées à partir de l'image en noir et blanc correspondant à chaque caractère, comme illustré dans la figure 41.



Figure 39 : Suppression des objets indésirables, (a) : Avant la suppression, (b) : Après la suppression



Figure 40 : Segmentation de la plaque

Après cela, les caractères segmentés seront redimensionnés à la taille du modèle, soit 42 x 24 pixels. Le but du redimensionnement est de rendre la taille des caractères comparable à celle des modèles :



Figure 41 : Redimensionnement des chiffres

4.8. Reconnaissance des caractères :

La figure III.15 montre les étapes de reconnaissance des caractères .

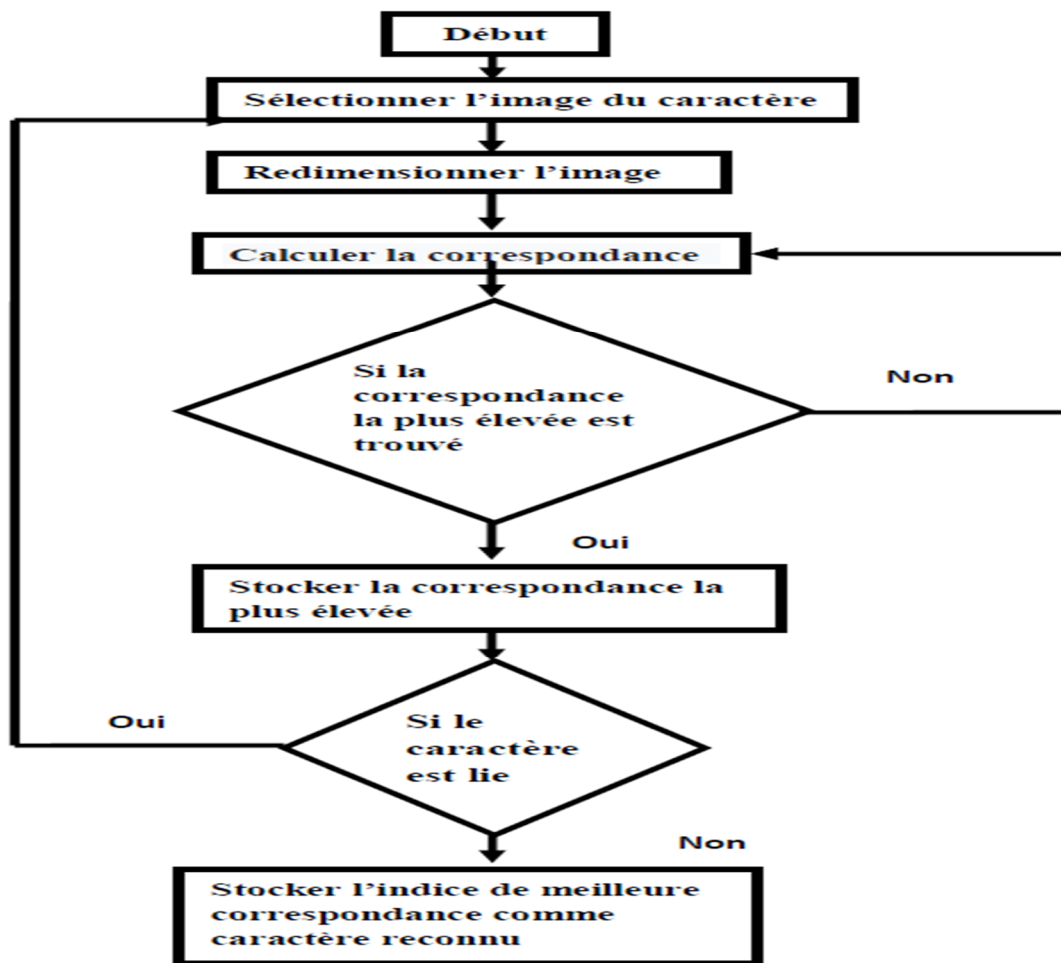


Figure 42 : Algorithme de reconnaissance de caractères

La reconnaissance des caractères est effectuée à l'aide d'une correspondance de modèle.

Les modèles sont normalisés à 42 x 24 pixels et stockés dans la base de données. Les modèles doivent tenir dans une fenêtre sans espaces blancs des quatre côtés. Une fois les caractères segmentés, nous commençons la reconnaissance des caractères par corrélation à 2 dimensions. Nous corrélons chaque caractère segmenté avec les modèles de chiffres de la base de données, puis nous choisissons la valeur de chaque caractère en fonction du résultat de la corrélation.

Le résultat de la reconnaissance des caractères est sélectionné en fonction des valeurs maximales de corrélation pour chaque caractère. Seuls les composants connectés affichant les valeurs de corrélation maximales seront sélectionnés, les cases restantes seront ignorées.



Figure 43 : Modèles utilisés pour la correspondance de modèles

5. Résultats Expérimental

Les résultats de la correspondance de modèle pour la reconnaissance des caractères sur certaines des plaques d'immatriculation algériennes à partir d'images statiques Sont illustrés au tableau 30. La comparaison des autres résultats expérimentaux est illustrée au tableau suivant :







Image	Plaque réelle	Plaque prédite	Caractères non identiques	Taux de réussite
	15416 112 24	15416 112 24	0	100%
	00087 119 24	0087 119 24	0	100%
	695195 00 16	695795 00 76	2	80%
	01943 118 24	01943 118 24	0	100%
	008951 314 16	00895 3 4 6	3	70%
	01813 115 19	01813 11519	0	100%

Tableau: Exemple de reconnaissance des plaques d'immatriculation.

- Le système proposé pour cette recherche a été testé sur quelques images car il n'y a pas de base de données de plaques d'immatriculation (PI) algérienne,

Le temps moyen pour la détection PI est de 2 secondes alors que le temps moyen pour la segmentation et la reconnaissance est de 10 secondes. Ainsi, le temps moyen total de capture de l'image pour reconnaître la plaque d'immatriculation est de 12 secondes.

6. Conclusion

Ce chapitre expose la méthodologie adoptée tout au long du développement de ce projet. Il décrit en détail le processus de développement de notre système de reconnaissance automatique des plaques d'immatriculation, en explicitant la mise en œuvre de chacune des différentes étapes du système.

Conclusion Générale

Ce mémoire a pour but de développer est de concevoir un Système de Reconnaissance Automatique de Plaque d'Immatriculation (RPI). Ce système reçoit comme input une image de véhicule, la soumet à un traitement et à une analyse à l'aide de diverses techniques de vision par ordinateur et un algorithme d'apprentissage artificiel SVM de la bibliothèque Scikit-Learn et, . Finalement, il est en mesure d'identifier la plaque d'immatriculation du véhicule présente dans l'image.

Ce système a atteint son objectif grâce à l'excellente conception et à l'implémentation réussie de toutes les étapes qui le composent. Ces étapes englobent l'acquisition d'images, la segmentation d'images, la détection de plaques d'immatriculation, la segmentation des plaques d'immatriculation et la reconnaissance des chiffres.

Dans un premier temps, il a été nécessaire de se familiariser avec les concepts fondamentaux relatifs aux images et à leurs caractéristiques, ainsi que de comprendre diverses techniques de traitement d'images et leurs opérations pour la détection et la segmentation des plaques d'immatriculation."

Après avoir abordé quelques notions de base, nous nous sommes penchés sur les étapes essentielles pour la création du système de reconnaissance de plaques d'immatriculation, en mettant particulièrement l'accent sur l'apprentissage artificiel et les algorithmes couramment utilisés dans la reconnaissance des chiffres.

En dernière étape, nous avons décrit la mise en œuvre des différentes phases constituant le système, en détaillant les résultats obtenus à chaque étape. Cette mise en œuvre a été réalisée en utilisant la base de données de chiffres manuscrits nommée "Digits", intégrée dans la bibliothèque Scikit-Learn.

Les résultats obtenus ont mis en évidence l'efficacité des méthodes employées à chaque étape du processus, que ce soit pour la détection de la plaque grâce à des techniques de traitement d'images simples, ou pour la segmentation des chiffres sur la plaque, allant jusqu'à l'identification des chiffres composant le numéro d'immatriculation. Cela a été accompli grâce à l'extraction de caractéristiques à partir des chiffres segmentés.

Les résultats de la reconnaissance mettent en évidence l'influence de plusieurs facteurs cruciaux pour obtenir une localisation précise de la plaque et une identification précise des chiffres qui la composent. Parmi ces facteurs, on peut mentionner les suivants : l'importance d'un bon éclairage au moment de la capture, le respect des normes de construction de la plaque, ainsi que le choix judicieux des paramètres et des seuils appropriés pour les algorithmes de segmentation et de détection de contours.

Nous avons fait face à des difficultés et à des obstacles tout au long des différentes étapes de notre travail, qui ont quelque peu ralenti notre progression vers une version améliorée et une mise en application optimale du système. Parmi ces problèmes, nous pouvons citer les suivants:

- Des objets qui cachent une partie de la plaque, bien souvent, une barre de remorquage, ou la saleté sur la plaque ou des chiffres altérés.
- La mauvaise résolution de l'image, soit parce que la plaque est trop loin ou parfois résultant de l'utilisation d'une caméra de mauvaise qualité.
- Nous avons été confrontés à une limitation liée au fait qu'il n'y avait pas de bases de données disponibles contenant des images de chiffres de plaques d'immatriculation de véhicules. De plus, en raison de contraintes de temps, nous n'avons pas pu créer notre propre ensemble de données pour entraîner notre modèle SVM, ce qui aurait été essentiel pour obtenir des résultats de classification et de reconnaissance des chiffres de plaques d'immatriculation de haute qualité.

Ce travail s'est avéré extrêmement bénéfique pour nous, car il nous a permis de maîtriser des outils essentiels de ces dernières années, tels que Python et OpenCV. Il a également constitué une excellente occasion de réaliser un travail concret, avec des objectifs clairs et bien définis.

À l'issue des travaux effectués dans le cadre de ce mémoire, nous citons quelques perspectives potentielles à explorer dans de futures recherches et projets.

- L'exploration de méthodes plus performantes, telles que le deep-learning, pour la localisation et la reconnaissance des plaques d'immatriculation.
- L'extension de l'application du système en temps réel, notamment pour la vidéo.

Références

- [01] : L. Laouamer, « Approche Exploratoire Sur La Classification Appliquée Aux Images », Université Du Québec,2006,pp 10.
- [02] : Philippe Bolon, Jean-Marc Chassery, Jean-Pierre Cocquerez, Didier Demigny, Christine Grafigne, et al... « Analyse d'images : Filtrage et segmentation ». Jean-Pierre Cocquerez; Sylvie Philipp. MASSON, 1995, Enseignement de la physique.
- [03] : T. Mohamed, D. Djafar, « la découverte des concepts sémantiques cachés avec plusieurs niveaux d'abstraction pour la recherche d'images », Université kasdi merbalh ouargla.2013.
- [04]: C. Taouciffi, « Implémentation d'un Environnement Parallèle pour la Compression d'Images à l'aide des Fractales », Université Mentouri Constantine, 2005.
- [05] : S. Benfriha & A. Hamel, « Segmentation d'image par Coopération région-contours », Université KasdiMerbah-Ouargla.2016
- [06] : Che-Yen Wen, Chun-Ming Chou, « Color Image Models and its Applications to Document Examination», Forensic Science Journal. pp. 23- 32 .Vol. 3(1). 2004.
- [07] : K.N. Plataniotis and A.N. Venetsanopoulos, « Color Image Processing and Applications», Springer- Verlag, February 18, 2000.
- [08] : <http://www.colorbasics.com/ColorSpace/>
- [9] H.Mohamadally , B.Fomani : « SVM machine à vecteurs de support ou séparateur à vaste marge », BD Web, ISTEY3, Versailles St Quentin, France, janvier 2006.
- [10] P. Mahé, L. Ait-Ali : « Projet d'apprentissage statistique SVM pour l'apprentissage non supervisé», DEA MVA, Février 2003.
- [11] h.ouakkaf belgasmik«reconnaissance automatique des expressions faciales par support vector machine», université larbi ben m'hidi oum el bouaghi faculté des sciences exactes et des sciences de la nature et de la vie...06 juin 2017.
- [12] M. ZAIZ Faouzi « Les Supports Vecteurs Machines (SVM) pour la reconnaissance des caractères manuscrits arabes » Université Mohamed Khider – BISKRA...15/07/2010.
- [13] B.Scholkopf, .A.J. Smola « Apprentissage avec les noyaux,, Support Vector Machines, Regularization, Optimization, and Beyond», the MIT Press 2002
- [14] <https://securite2000technique.net/produits/controle-dacces-genetec/autovu/>
- [15] <http://www.alphanumericvision.com/>
- [16] <https://www.cgv-systeme.fr/merchant/product/module-seetec-7773610>
- [17] <https://www.parking.net/parking-industry/asia-vision-technology-ltd->

