

Table des matières

| | | |
|----------|--|-----------|
| 1 | Vision par ordinateur | 11 |
| 1.1 | Introduction | 12 |
| 1.2 | Définition de la vision par ordinateur | 12 |
| 1.3 | Images | 13 |
| 1.3.1 | Les types d'images | 14 |
| 1.3.2 | Les caractéristiques d'une image numérique | 15 |
| 1.3.3 | Quelques propriétés d'image numérique | 17 |
| 1.3.4 | Interprétation d'image numérique | 18 |
| 1.3.5 | Traitement d'images numérique | 21 |
| 1.4 | Application et perspective de la vision par ordinateur | 23 |
| 1.5 | Conclusion | 24 |
| I | État de l'art sur la détection d'objets | 25 |
| 2 | Détection d'objet | 26 |
| 2.1 | Introduction | 27 |
| 2.1.1 | Segmentation d'images | 27 |
| 2.1.2 | Détection basée sur la différence inter-images | 33 |
| 2.1.3 | Détection des zones d'intérêt | 34 |
| 2.1.4 | Méthode de Viola et Jones | 38 |
| 2.2 | Détection de plaques d'immatriculation | 45 |
| 2.2.1 | introduction | 45 |
| 2.2.2 | Méthode de détection de plaques d'immatriculation | 45 |
| 2.3 | Conclusion | 49 |

| | | |
|-----------|--|-----------|
| II | État de l'art sur la reconnaissance d'objet | 50 |
| 3 | Reconnaissance visuelle par ordinateur | 51 |
| 3.1 | Introduction | 52 |
| 3.2 | Définition | 52 |
| 3.3 | Principes d'un algorithme de reconnaissance visuelle | 53 |
| 3.3.1 | Reconnaissance | 54 |
| 3.3.2 | Les étapes fondamentales de reconnaissance | 54 |
| 3.4 | Les différentes méthodes de reconnaissance | 55 |
| 3.4.1 | Méthodes géométrique | 55 |
| 3.4.2 | Méthode globale | 56 |
| 3.4.3 | Méthode locale | 57 |
| 3.5 | Conclusion | 58 |
| 3.6 | Reconnaissance optique des caractères | 58 |
| 3.6.1 | Introduction | 58 |
| 3.6.2 | Définition | 59 |
| 3.6.3 | Historique | 60 |
| 3.6.4 | Domaines d'application des OCR | 61 |
| 3.6.5 | Fonctionnement d'un OCR | 62 |
| 3.6.6 | Conclusion | 68 |
| 4 | Apprentissage et classification | 69 |
| 4.1 | Apprentissage | 70 |
| 4.1.1 | Introduction | 70 |
| 4.1.2 | Apprentissage naturel | 71 |
| 4.1.3 | Apprentissage artificiel (automatique) | 72 |
| 4.1.4 | L'environnement méthodologique de l'apprentissage | 72 |
| 4.1.5 | les types d'apprentissage | 74 |
| 4.1.6 | Conclusion | 78 |
| 4.2 | Classification | 78 |
| 4.2.1 | Introduction | 78 |
| 4.2.2 | Définition | 78 |
| 4.2.3 | Classification supervisée | 79 |

| | | |
|----------|--|-----------|
| 4.2.4 | La classification multi-labels | 81 |
| 4.2.5 | La classification binaire | 81 |
| 4.2.6 | Conclusion | 82 |
| 5 | Implémentation et Résultats | 83 |
| 5.1 | Outils et environnement de développement | 85 |
| 5.2 | OpenCV : Open source Computer Vision | 85 |
| 5.3 | Python | 86 |
| 5.3.1 | La bibliothèque standard math | 87 |
| 5.3.2 | NumPy | 88 |
| 5.3.3 | Latex | 88 |
| 5.4 | Système d'exploitation et outils utilisés | 89 |
| 5.5 | Outils | 89 |
| 5.5.1 | Sublime text | 89 |
| 5.5.2 | PyCharm | 90 |
| 5.5.3 | TeXnicCenter | 90 |
| 5.6 | Approche de détection implémentée | 91 |
| 5.6.1 | Caractéristique des plaques algériennes | 92 |
| 5.7 | Présentation de l'approche suivie | 93 |
| 5.7.1 | Prétraitement | 93 |
| 5.7.2 | Deuxième phase : Application des contraintes dimensionnelles | 95 |
| 5.7.3 | Troisième phase : Application de la contrainte de la couleur | 96 |
| 5.8 | Classification par caractéristiques de Haar | 98 |
| 5.8.1 | Apprentissage du classificateur | 98 |
| 5.8.2 | Extraction des caractéristiques d'une image | 99 |
| 5.8.3 | Élaboration du classificateur | 99 |
| 5.9 | Reconnaissance optique des caractères de la plaque | 101 |
| 5.10 | Tesseract | 101 |
| 5.10.1 | Motivations et historique | 101 |
| 5.10.2 | Architecture | 102 |
| 5.10.3 | Retrouver les lignes et les mots | 103 |
| 5.10.4 | Reconnaissance de mot | 105 |
| 5.10.5 | Classification | 106 |

| | | |
|------------|--|------------|
| III | Annexe | 107 |
| A | | 108 |
| A.1 | Opérations morphologiques | 109 |
| A.1.1 | Définition | 109 |
| A.1.2 | Ensemble caractéristique | 109 |
| A.1.3 | Élément structurant | 109 |
| A.1.4 | Dilatation | 110 |
| A.1.5 | Erosion | 112 |
| A.1.6 | Ouverture | 113 |
| A.1.7 | Fermeture | 113 |
| A.2 | Les filtres | 114 |
| A.2.1 | Filtres de lissage | 114 |
| A.2.2 | Filtrage global | 115 |
| A.2.3 | Filtrage local | 115 |
| A.2.4 | Filtre passe-bas | 115 |
| A.2.5 | Filtre passe-haut | 115 |
| A.2.6 | Filtre Gaussien | 116 |
| A.3 | Convolution | 116 |
| A.3.1 | Convolution spatiale | 117 |
| A.3.2 | Gestion des bords | 119 |
| A.4 | Gradient | 120 |
| A.4.1 | Direction du gradient | 120 |
| A.4.2 | Quelques opérateurs Gradient | 121 |

Table des figures

| | | |
|------|--|----|
| 1.1 | Image matricielle | 14 |
| 1.2 | Image vectorielle | 15 |
| 1.3 | Exemple d'une image pixellisée | 15 |
| 1.4 | Exemple d'image de définition (5*6) | 16 |
| 1.5 | Exemple illustrant Différentes résolutions d'image | 17 |
| 1.6 | Histogramme cumulé | 19 |
| 1.7 | Histogramme cumulé d'une image contenant tous les niveaux de gris possibles | 20 |
| 1.8 | Les trois histogrammes des trois couleurs formant l'image donnée | 20 |
| 1.9 | Spectre de la lumière blanche, couleurs de l'arc-en-ciel | 21 |
| 1.10 | Image représentant la synthèse additive | 21 |
| 1.11 | Illustration de la segmentation d'images (figure1 : image d'origine, figure2 : l'image segmentée) | 22 |
| 1.12 | Exemple de filtrage d'une image bruitée (à gauche l'image d'origine, à droite l'image filtrée) | 23 |
| 2.1 | Sélection du pic le plus signifiant de l'histogramme de l'image | 29 |
| 2.2 | Exemple illustratif de la définition des seuils | 30 |
| 2.3 | A gauche : une image réelle. A droite : histogramme des niveaux de gris | 30 |
| 2.4 | Segmentation de l'image précédente à l'aide de deux seuils : 50 et 110 | 31 |
| 2.5 | illustration d'une segmentation basée régions | 33 |
| 2.6 | Détection de contours | 35 |
| 2.7 | Différents types de points d'intérêt. La première image représente un coins, la seconde une jonction en un point T et en dernier des points de fortes variations de texture. | 36 |

| | | |
|------|---|----|
| 2.8 | Exemple illustratif de la manière de procéder au calcul de la similarité entre les zones adjacente. Il s'agit de calculer la dérivée de chaque pixel (point (x,y)) sur les huit directions (u,v) $\{ (1,0) (1,1) (0,1) (-1,1) (-1,0) (-1,-1) (0,-1) (1,-1) \}$ | 37 |
| 2.9 | Les points d'intérêt dans une forme géométrique | 37 |
| 2.10 | Exemple des caractéristiques de Haar à 2, 3 ou 4 rectangles | 39 |
| 2.11 | Calcul de la valeur du pixel à l'aide de l'image intégrale. La valeur du pixel, dont les coordonnées, sont (x, y) est égale à la somme de celui-ci avec les pixels à gauche et au-dessus de lui. | 40 |
| 2.12 | Utilisation des cascades de classificateurs pour la détection de voitures . . . | 43 |
| 2.13 | Fonctionnement de la cascade de classifier | 44 |
| 3.1 | Image initiale | 62 |
| 3.2 | Image après redressement | 63 |
| 3.3 | Image initiale | 63 |
| 3.4 | Image après redressement | 63 |
| 3.5 | Image affectée (présence de parasites bruits) | 64 |
| 3.6 | Image nettoyée (suppression des parasites) | 65 |
| 3.7 | Image passée à l'OCR | 65 |
| 3.8 | schéma des caractères reconnus | 65 |
| 3.9 | Image de plaque passée à l'OCR | 66 |
| 3.10 | Image transformée par l'OCR (segmentée) | 66 |
| 3.11 | Lecture des caractères (présence de quelques erreurs) | 66 |
| 3.12 | Détection des incohérences | 67 |
| 3.13 | Correction des incohérences | 67 |
| 3.14 | Détection erreurs du premier passage à l'OCR | 68 |
| 3.15 | Correction des erreurs détectées avec un traitement intelligent | 68 |
| 4.1 | Schéma récapitulatif de l'apprentissage supervisé | 75 |
| 4.2 | Exemple de comparaison de données : à gauche l'exemple d'entrés, à droite la nouvelle donnée (dans ce cas de figure le système doit retourner vrai) . . | 76 |
| 4.3 | Schéma récapitulatif de l'apprentissage non-supervisé | 76 |
| 4.4 | Schéma récapitulatif de l'apprentissage par renforcement | 77 |

| | | |
|------|--|-----|
| 5.1 | Image en niveau de gris | 93 |
| 5.2 | Image filtrée | 94 |
| 5.3 | Application de l'opérateur de Sobel | 94 |
| 5.4 | Obtention d'une image binaire | 95 |
| 5.5 | Fermeture | 95 |
| 5.6 | Dessin de tous les contours de l'image | 96 |
| 5.7 | Les rectangles en rouges satisfont les contraintes | 96 |
| 5.8 | Les différents masques générés | 97 |
| 5.9 | Les régions découpées | 98 |
| 5.10 | Cascades de classifier | 100 |
| 5.11 | Exemple d'une spline | 104 |
| 5.12 | Exemple d'une ligne de base incurvée. | 105 |
| A.1 | Exemples d'élément structurant | 109 |
| A.2 | Exemple Illustrant l'effet d'une dilatation sur une image | 111 |
| A.3 | Exemple Illustrant l'effet d'une érosion sur une image | 112 |
| A.4 | Illustration d'une ouverture | 113 |
| A.5 | Illustration d'une fermeture | 114 |
| A.6 | Produit de convolution | 118 |
| A.7 | Exemple illustratif du calcul du produit de convolution | 118 |
| A.8 | Exemple de filtre de convolution | 119 |
| A.9 | Schéma récapitulatif de la détection de contours par gradient | 121 |
| A.10 | Masques de convolution des operateurs de Prewitt (k=1) et de Sobel (k=2) | 122 |
| A.11 | Exemple d'application de l'opérateur de Sobel | 124 |

Résumé

Avec la nette ascension des ventes automobiles, il est devenu très difficile de contrôler le trafic routier chose qui fait ressentir le besoin de recourir à d'autres alternatives et de nouvelles technologies encore plus infaillibles. La lecture automatique des plaques d'immatriculation (LAPI) semble être un excellent procédé pour pallier à ce problème.

En effet la Lecture Automatique des Plaques d'Immatriculation (LAPI) ou plus couramment Automatic Number Plate Recognition system (ANPR) est une technique d'identification qui utilise des méthodes de traitement d'images et de vision par ordinateur pour lire les plaques minéralogiques des voitures et cela pour des fins bien diverses telles que la sécurité des transports, identification des véhicules volés, gestion des parkings (parkings intelligents) et bien d'autres usages .

Réellement c'est un système assez complexe qui se déroule en trois phases distinctes à savoir : la détection de la plaque minéralogique, la segmentation des caractères et enfin la reconnaissance des caractères, ces phases sont réalisées grâce à des algorithmes puissants rattachés à un vaste domaine qu'est l'intelligence artificielle plus précisément la vision par ordinateur.

Notre travail consiste à passer en revue l'état de l'art de la Lecture Automatique des Plaques d'Immatriculation (LAPI) et afin de réaliser une application capable de lire les plaques minéralogiques et traiter les informations amassées.

Abstract

With the net ascent of motorised sales, it became very difficult to control road traffic thing which makes feel the need to use other alternatives and new even more infallible technologies. The Automatic Number Plate Recognition (ANPR) seems to be a very good technique to palliate this problem.

This innovating system is an identification technology which uses methods of treatment of pictures and of vision by computer to read the mineralogical plates of cars and it's used for very various aims such as the transport safety, identification of the stolen vehicles, management of parkings (clever parkings) and many other manners.

Indeed it is a rather complex system which takes place in three distinct stages to be known : the detection of the mineralogical plate, the segmentation of characters and finally the recognition of characters.

These stages are realised thanks to powerful algorithms attached to a large artificial intelligence domain more precisely is vision by computer.

Our job consists in reviewing the state of the art of the Automatic Number Plate Recognition (ANPR) and accomplish an application capable of reading mineralogical plates and to treat accumulated information.

Introduction générale

Contexte et problématique

La sécurité de nos routes est un défi quotidien auquel se heurte la direction générale des transports.

En effet avec la nette croissance du marché automobile surtout à la fin des années 90, il est devenu très difficile de gérer ce gigantesque réseau, ce qui fait ressentir le besoin de se tourner vers des idées novatrices afin de combler le manque d'efficacité des moyens humains déjà à disposition.

Les techniques de lecture des plaques d'immatriculation semblent être un très bon moyen de subvenir aux besoins d'identification et de classification des véhicules empruntant les routes ce qui va simplifier par la suite la gestion du trafic et l'identification des automobilistes maladroits et imprudents.

Cependant, cette technique peut être mise à contribution d'autres secteurs d'activité pour qu'on puisse tirer profit de sa puissance et faire naître un gisement d'idées plus innovantes les unes que les autres.

Effectivement, les renseignements extraits des plaques minéralogiques peuvent être utilisés pour le contrôle d'accès, les parking intelligents, contrôle aux frontières, la mesure de temps de voyage etc .

Ce sont des domaines florissants qui vont inciter un bon nombre de pôles économiques à opter pour l'exploitation de ce gigantesque marché.

Chapitre 1

Vision par ordinateur

Contents

| | | |
|------------|---|-----------|
| 1.1 | Introduction | 12 |
| 1.2 | Définition de la vision par ordinateur | 12 |
| 1.3 | Images | 13 |
| 1.3.1 | Les types d'images | 14 |
| 1.3.2 | Les caractéristiques d'une image numérique | 15 |
| 1.3.3 | Quelques propriétés d'image numérique | 17 |
| 1.3.4 | Interprétation d'image numérique | 18 |
| 1.3.5 | Traitement d'images numérique | 21 |
| 1.4 | Application et perspective de la vision par ordinateur | 23 |
| 1.5 | Conclusion | 24 |

1.1 Introduction

La vision est une faculté dont l'être humain est doté dès sa naissance, cette faculté innée est, entre autres, le pouvoir de discerner et reconnaître les formes, les couleurs et les textures du monde qui nous entoure, mais la vision ne s'arrête pas là, on pourrait effectivement croire, à tort, que nous avons besoin uniquement de nos yeux pour pouvoir acquérir de l'information et interpréter ce qu'on voit, mais la réalité est toute autre, nos yeux ne servent qu'à capter le signal contenu dans les rayons lumineux qui viennent frapper nos rétines, pas à en extraire des informations, c'est le rôle du cerveau d'interpréter et de comprendre ce qu'il voit.

De nos jours la science éprouve de plus en plus le besoin de reproduire divers comportements ou facultés humaines et la vision est une des facultés qui captivent les chercheurs, et grâce aux différentes avancées technologiques, les chercheurs ont en effet pu offrir la possibilité d'avoir des aperçus du monde par le biais d'appareils d'acquisition d'images (appareils photo, caméras), cependant l'émergence de la vidéo comme média d'information s'accompagne d'un besoin croissant d'interprétations des informations contenues dans ces dernières à savoir de permettre à la machine (l'ordinateur) de comprendre ce qu'elle voit, c'est ce qui nous mène à la vision par ordinateur qui est une discipline dont les premières bases théoriques ont été énoncées vers les années 60.

Depuis, étant donné le spectre très large d'applications qui peut être envisagé, la vision par ordinateur a vite fait de dépasser le cadre restreint à des laboratoires de recherche et à gagner du terrain dans le domaine applicatif, en effet cette technologie est utilisée dans divers domaines : industriels, militaires, aérospatiales, et surtout dans le domaine de la sécurité et de la surveillance (surveillance du trafic routier par exemple) ce qui fera l'objet d'étude dans notre présent travail.

1.2 Définition de la vision par ordinateur

La vision par ordinateur est un champ du vaste domaine de l'intelligence artificielle qui inclut plusieurs méthodes afin de permettre à une machine d'analyser, de traiter et de comprendre une ou plusieurs images fournies par un système d'acquisition (appareils photo, caméras). Pour franchir ce cap et permettre cette compréhension à l'ordinateur, la vision par ordinateur se réfère à des algorithmes variés pour restaurer ou interpréter les

images digitales.

On peut distinguer deux niveaux de traitement d'images auxquels la vision par ordinateurs fait recours :

- la vision haut niveau dont le but est d'extraire les attributs symboliques (par exemple la reconnaissance des lettres écrites à la main).
- La vision bas niveau (ou vision pré-attentive) qui traite une grande quantité de pixels et les transforme en attributs nécessaires pour la vision haut niveau, on peut citer l'extraction des contours, détection des régions, texture, etc.

En effet, la vision par ordinateur fait recours aux techniques de traitement d'images vu que le principal paramètre d'entrée de cette dernière est l'image qui est une représentation visuelle d'un objet ou d'un concept du monde qui nous entoure, dans les séquences qui vont suivre nous allons nous pencher sur les détails et la complexité de cette technologie.

1.3 Images

Une image est la projection et la représentation d'un ensemble de rayons lumineux provenant du monde extérieur, elle est constituée d'un ensemble de points appelés pixels (PICTure Element). Le pixel représente ainsi le plus petit élément constitutif de cette dernière, en effet chaque point d'une image possède deux coordonnées (X, Y) de telle sorte que chaque couple de coordonnées (X, Y) représente l'emplacement d'un point dans cette image sur un plan (photo, écran). La représentation de ce couple correspond à la valeur du rayon lumineux capté.

Plus formellement, l'image est un cliché électronique d'une scène, elle est échantillonnée et mappée comme une grille de points ou éléments d'images où chaque point correspond à une valeur tonale (noir, blanc, niveaux de gris ou couleur), exprimée en code binaire (zéros et uns). Les chiffres binaires ("bits") de chaque pixel sont stockés dans une séquence mémoire 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.

1.3.1 Les types d'images

On peut distinguer deux types d'images, l'image en pixels (ou matricielle) et l'image vectorielle.

Image en pixels

L'image en pixels (matricielle) se présente sous forme d'une grille dotée de deux axes et constituée d'une multitude de points (pixels), ces pixels sont représentés sous forme de minuscules carrés, non perceptibles quand l'image est à sa taille normale et derrière chaque pixel se trouve un échantillon de couleur.



FIGURE 1.1 – Image matricielle

Image vectorielle

L'image vectorielle est dépourvue de matrices, elle est composée de lignes, de segments qui sont liés entre eux à l'aide de formules mathématiques, en d'autres termes, elle est composée d'objets géométriques individuels (segments de droites, polygones, arcs, cercles, etc). Grâce à la vectorisation, chaque élément a une place bien définie ce qui empêche la déformation et la perte de qualité de l'image.

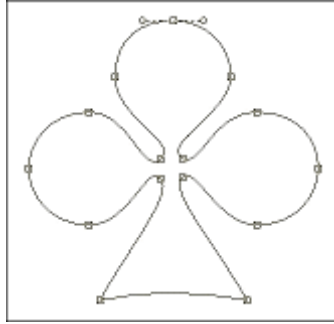


FIGURE 1.2 – Image vectorielle

1.3.2 Les caractéristiques d'une image numérique

Pixellisation d'une image

La pixellisation d'une image est le fait de convertir une image en intégralité de pixels ce qui permet de numériser ou transformer en une image numérique (exprimée par une matrice composée les valeurs numériques des pixels) de ce fait la pixellisation exprime le nombre de pixels contenu dans une image.



FIGURE 1.3 – Exemple d'une image pixellisée

Taille (définition) d'une image

On appelle définition le nombre de points (pixels) constituant une image, c'est le nombre de colonnes de l'image que l'on multiplie son nombre de lignes. Une image possédant 200

colonnes et 200 lignes aura une définition de 200x200

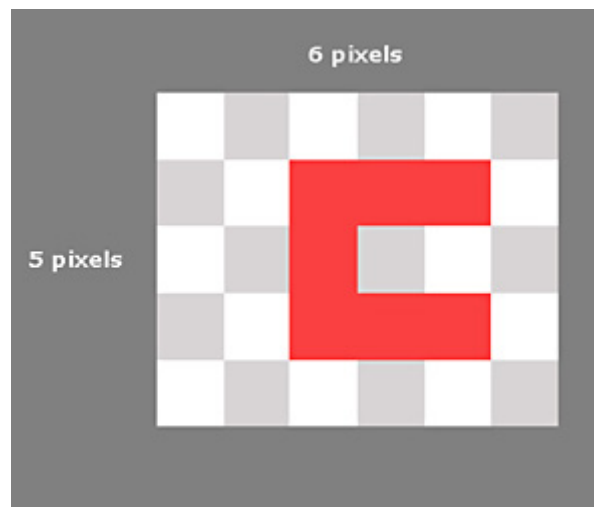


FIGURE 1.4 – Exemple d'image de définition (5*6)

Résolution d'une image

La résolution d'une image s'exprime elle en DPI (Dot per Inch ou Point par pouce où 1 pouce mesure 2,54 cm) et dépend donc de la taille. Selon la résolution choisie, la taille physique de l'image changera. La résolution d'une image est donc composée de points et définie par la densité des points par unité de surface. La résolution permet de définir la finesse de l'image. Plus la résolution est grande, plus la finesse de l'image est grande[14].

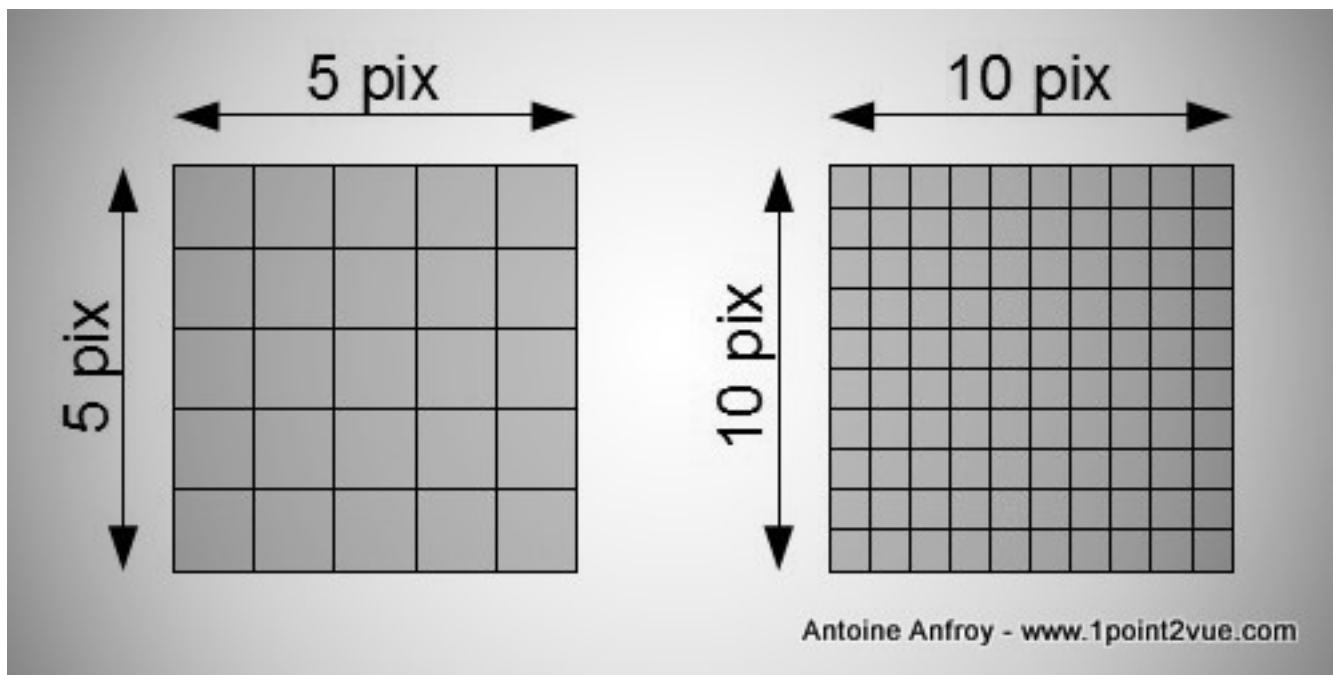


FIGURE 1.5 – Exemple illustrant Différentes résolutions d’image

Profondeur d’une image

La profondeur désigne la taille en “bit” occupé dans la mémoire pour chaque pixel de l’image.

1.3.3 Quelques propriétés d’image numérique

Photométrie

Mesure des grandeurs relatives aux rayonnements, évaluées selon l’impression visuelle produite par ceux-ci du point de vue de la perception de l’œil humain. Cette science étudie plusieurs grandeurs telles que l’intensité lumineuse, le flux lumineux, la luminance et l’éclairement, les unités correspondantes étant la candela, le lumen, la candela par mètre carré et le lux[27].

Flux lumineux

Il représente la fraction du flux énergétique correspondant à la lumière perceptible par l’œil humain, pondéré selon sa longueur d’onde.

Luminance

La luminance est tout simplement la sensation visuelle qu'on a de la lumière. De toutes les grandeurs photométriques qui caractérisent la lumière, la luminance est celle qui concerne le plus la vision humaine. Lorsque la vision est gênée dans une forte pénombre, ou lorsqu'un phare nous éblouit, c'est la luminance qui est en cause.

Intensité lumineuse

L'intensité lumineuse d'une source, dans une direction donnée, correspond au flux lumineux émis par unité d'angle solide. L'intensité est calculée avec la formule suivante :

$$I = \frac{d\phi}{d\Omega} \quad (1.1)$$

Où ϕ désigne le flux lumineux et Ω désigne l'angle solide d'émission.

1.3.4 Interprétation d'image numérique

Codage des couleurs de l'image numérique

Le pixel est l'unité de base permettant de mesurer la définition d'une image numérique matricielle, une image dépend du codage des couleurs du pixel elle peut donc être :

– **En noir et blanc :**

Dans cette représentation, un pixel de cette image n'a la possibilité d'avoir que deux valeurs distinctes 0 pour le noir et 1 pour le blanc.

– **En niveau de gris :**

Dans une image numérique, le niveau de gris représente la luminosité d'un pixel. Un pixel est codé généralement sur un octet et a donc 256 valeurs possibles ; il permet d'obtenir plus de nuances que le simple noir et blanc.

– **En couleurs :**

Il existe plusieurs modes de codages informatiques des couleurs, le plus utilisé pour le remaniement des image est l'espace colorimétrique rouge, vert, bleu (RVB ou en anglais RGB red, green, blue). Cet espace est basé sur la synthèse additive des couleurs, les couleurs sont obtenues en mélangeant les trois couleurs fondamentales,

à chaque couleur est associé un octet et chaque pixel « couleur » est codé sur 3 octets pour former 16 millions de couleurs différentes.

Histogramme cumulé d'une image

Histogramme cumulé en niveaux de gris d'une image

L'histogramme nous renseigne sur la luminosité d'une image, il se présente sous la forme d'un graphique à deux dimensions réalisé à partir de deux axes, il a pour fonction la représentation du mode de répartition des pixels d'une image en traçant le nombre de pixels correspondant à chaque niveau d'intensité de couleurs. On retrouve sur l'axe des abscisses les 256 valeurs de gris qui peuvent composer une image et qui sont représentées par un dégradé allant du noir jusqu'au blanc, l'axe des ordonnées quant à lui représente la quantité de ces valeurs de gris composant l'image.



FIGURE 1.6 – Histogramme cumulé

L'histogramme est ainsi un outil qui permet d'indiquer la quantité des pixels présents pour chacun des tons, il sert de repère visuel pour les tons dominants ou manquants, il varie donc, en fonction de l'image qu'il représente, on peut voir que pour une photographie terne, l'histogramme indique que la majorité des pixels se trouve dans les tons moyens et donc, une photographie exposée aura une prédominance des pixels sur les tons clairs, tant dis qu'une photographie sous-exposée aura une prédominance des tons foncés.

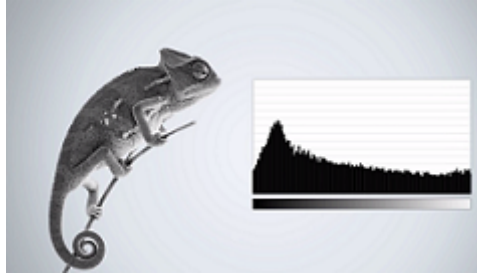


FIGURE 1.7 – Histogramme cumulé d’une image contenant tous les niveaux de gris possibles

Images en couleur

La représentation de la répartition des intensités sur une image en couleur (Rouge, Vert, Bleu) nécessite 3 histogrammes (chaque histogramme représentant la distribution de la luminance d’une couleur).

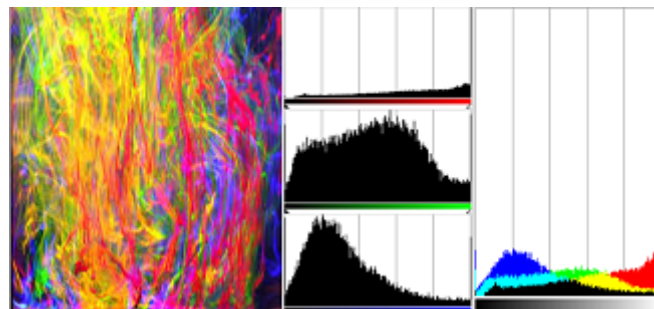


FIGURE 1.8 – Les trois histogrammes des trois couleurs formant l’image donnée

Qu’est-ce qu’une image en couleurs ?

La couleur est à la fois un phénomène psychophysique faisant intervenir la physique de la matière, notamment les interactions des ondes électromagnétiques avec les matériaux physiques. Mais c’est également un phénomène psychophysiologique par l’interprétation des phénomènes psychophysiques par le système visuel constitué notamment de l’œil et du cerveau. En 1666, Isaac Newton a réussi lors de la diffraction de la lumière blanche grâce à un prisme optique (un prisme droit en verre à base triangulaire) à mettre en évidence le spectre de la lumière blanche.

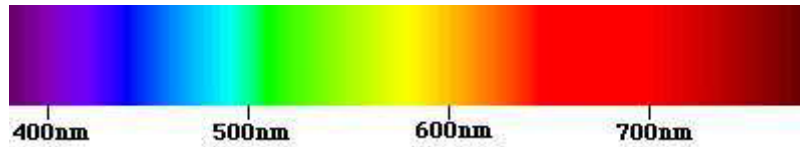


FIGURE 1.9 – Spectre de la lumière blanche, couleurs de l’arc-en-ciel

Le physicien Young fit le contraire de Newton. Il recomposa la lumière. Il fit converger les six couleurs du spectre et obtint la lumière blanche. Il a également démontré que les six couleurs du spectre pouvaient être réduites à trois, ce qui veut dire qu’avec ces trois couleurs il pouvait recomposer la lumière blanche. Parallèlement, Young a pu démontrer qu’en mélangeant ces trois couleurs (Rouge, Vert, Bleu) deux par deux, il obtenait les trois autres couleurs du spectre. Et c’est ainsi qu’on différençia les couleurs primaires des secondaires.

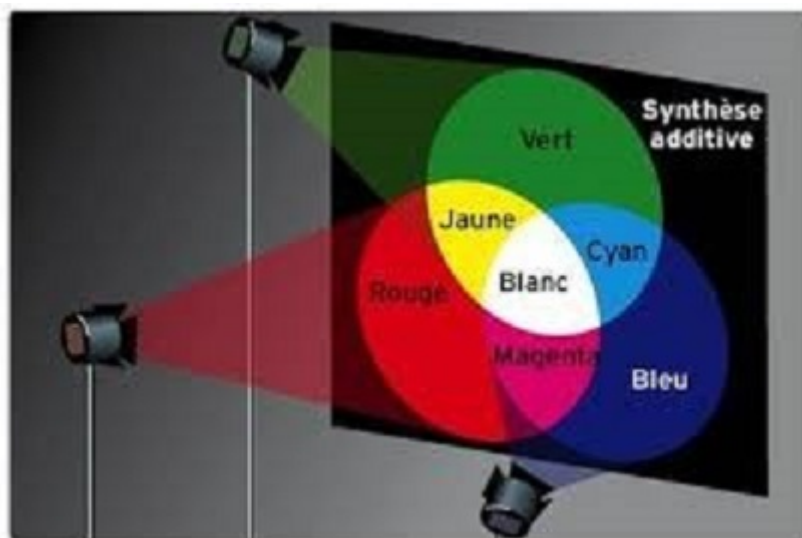


FIGURE 1.10 – Image représentant la synthèse additive

1.3.5 Traitement d’images numérique

Le traitement d’image est une discipline issue de deux domaines qui se croisent et qui sont l’informatique et les mathématiques appliquées, cette dernière étudie les caractéristiques et les propriétés des images numériques et leurs transformations dans le but soit

d'améliorer leur qualité ou d'en extraire de l'information. Dans ce qui va suivre, nous allons présenter deux opérations de traitement d'image visant à améliorer la qualité ou extraire de l'information de cette dernière.

Segmentation d'image

La segmentation consiste en la répartition de l'ensemble des pixels de l'image en groupe, chacun d'eux formera une région de l'image selon un critère d'homogénéité (chaque région est caractérisée par un critère d'homogénéité). Cette technique de traitement d'image vise à séparer le plus précisément possible les différents objets se trouvant dans l'image traitée et ainsi extraire l'information présente dans cette dernière. Il existe plusieurs méthodes et approches pour la segmentation d'images, celles-ci seront présentées dans les sections suivantes.



FIGURE 1.11 – Illustration de la segmentation d'images (figure1 : image d'origine, figure2 : l'image segmentée)

Filtrage

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

se servant des valeurs des pixels de l'image d'origine. Il existe plusieurs filtres effectuant différents traitements selon le besoin de l'analyse, on cite l'exemple de filtrage d'une image bruitée par un filtre opérant de telle sorte à supprimer les pixels bruits présents dans l'image.

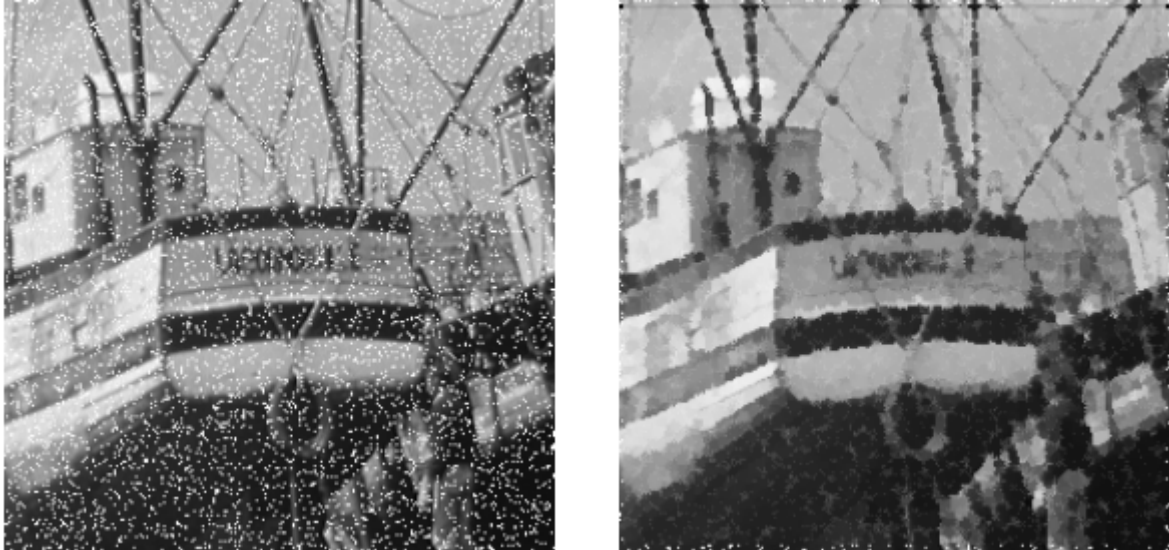


FIGURE 1.12 – Exemple de filtrage d'une image bruitée (à gauche l'image d'origine, à droite l'image filtrée)

1.4 Application et perspective de la vision par ordinateur

La vision par ordinateur est un domaine très jeune et en pleine croissance. Cette dernière recouvre les technologies centrales d'analyse automatique d'image utilisée dans divers domaines, en effet l'application de la vision par ordinateur va de la vision industrielle (application de la vision par ordinateur aux procédés industriels, effectuer un contrôle de qualité sur des produits industriels automatiquement) à la recherche dans le domaine de l'intelligence artificielle et des ordinateurs ou robots capables de comprendre le monde qui les entoure.

La vision par ordinateur à su s'immiscer et elle est en pleine expansion dans plusieurs domaines, on cite parmi ceux-ci quelques uns :

- **La robotique**, la vision par ordinateur est utilisée pour la fabrication des robots autonomes capables d'identifier et de manipuler des objets et d'interagir avec le monde extérieur.
- **La médecine**, l'imagerie médicale est très importante comme le démontre le rôle de l'IRM (Imagerie par Raisonance Magnétique) pour identifier les maladies et pour la recherche.
- Dans la **surveillance**, la reconnaissance faciale et la reconnaissance de personnes et leurs mouvements sont très utiles pour l'authentification et la sécurisation de divers organismes.
- Et en fin on cite la **gestion du trafic routier**, la reconnaissance automatique de véhicules de leurs types et de leurs plaques d'immatriculation permet de gérer et de sécuriser le trafic.

1.5 Conclusion

Dans ce précédent chapitre nous avons abordé l'un des piliers du vaste domaine de l'intelligence artificielle qui est la vision par ordinateur et quelques définitions et concepts reliés à cette dernière que nous manipulerons et qui seront très utiles pour la suite dans notre application qui consiste en la détection de plaque d'immatriculation algérienne dans des images et la reconnaissance des caractères la composant (les chiffres).

Dans ce qui va suivre, nous allons donc dans une première partie présenter un bref état de l'art sur les méthodes de détection d'objets et nous allons passer par la suite dans une seconde partie aux méthodes de reconnaissance de formes.

Première partie

État de l'art sur la détection d'objets

Chapitre 2

Détection d'objet

Contents

| | | |
|------------|---|-----------|
| 2.1 | Introduction | 27 |
| 2.1.1 | Segmentation d'images | 27 |
| 2.1.2 | Détection basée sur la différence inter-images | 33 |
| 2.1.3 | Détection des zones d'intérêt | 34 |
| 2.1.4 | Méthode de Viola et Jones | 38 |
| 2.2 | Détection de plaques d'immatriculation | 45 |
| 2.2.1 | introduction | 45 |
| 2.2.2 | Méthode de détection de plaques d'immatriculation | 45 |
| 2.3 | Conclusion | 49 |

2.1 Introduction

En vue d'obtention d'un système de reconnaissance de plaques d'immatriculations efficace, il est crucial de mettre en place un système de détection d'objets, afin de pouvoir détecter les véhicules et plus précisément les plaque d'immatriculation en question. En effet, les techniques de détection peuvent être considérées comme un prétraitement permettant de réduire la quantité d'informations à traiter et à analyser.

Dans ce chapitre, nous allons introduire de façon générale le cadre théorique de la détection de véhicule, et de façon plus précise la détection des plaques minéralogiques.

Afin de pouvoir détecter un objet notamment un véhicule dans une séquence d'images, il existe plusieurs méthodes sur lesquelles on peut s'appuyer, à savoir, les méthodes de détection de mouvement, détection de contours, segmentation des images, etc. Toutes ces méthodes visent à séparer ou segmenter l'objet recherché du reste de l'image, par exemple : un véhicule en mouvement est détecté si sa position change relativement à celle d'un ensemble d'objets statiques ou s'il est localisé dans une image acquise à l'instant T de la séquence à une position différente de celle qu'il occupait dans l'image précédente. Tandis que pour la détection des plaques d'immatriculations, on a recours à des méthodes et algorithmes plus précis qui commencent par utiliser une série de techniques de manipulation d'image pour détecter, normaliser et agrandir l'image de ces dernières, et enfin la reconnaissance optique de caractères pour extraire les caractères alphanumériques de la dite plaque. Toutes ces méthodes et algorithmes seront présentés plus en détail dans ce qui suit.

Les différentes approches de détection d'objets

2.1.1 Segmentation d'images

Définition

Dans un souci de détection précise et correcte, il est nécessaire de faire appel à certaines approches de traitement d'image, en effet, la segmentation d'image est l'un des principes les plus importants sur lequel se base divers algorithmes et méthodes de détection et de

reconnaissance dans une image numérique.

La segmentation est l'analyse d'une image dans le but d'extraire les informations contenues dans cette dernière et de fournir une description structurale que l'œil effectue naturellement, cette description peut prendre la forme de toute structure de données permettant la représentation des entités présentes dans l'image à analyser[3].

En d'autres termes, la segmentation consiste à partitionner l'image en un ensemble de régions connexes, dans lesquelles les pixels vérifient un certain critère d'homogénéité, tel que le niveau de gris ou la couleur, elle permet d'isoler dans l'image les objets sur lesquels doit porter l'analyse. L'obtention de ces régions ou partitions permet ensuite d'effectuer des manipulations et traitements visant l'extraction de caractéristiques de forme, de position, de taille, etc. Il existe de très nombreuses approches permettant ce partitionnement, dont l'efficacité dépend avant tout de l'image, il est donc très difficile de déterminer quelle est la segmentation idéale, pour ce faire deux propriétés principales sont dégagées :

1. Stabilité : la segmentation obtenue ne doit pas varier beaucoup lorsque les conditions d'acquisition varient légèrement (bruit, illumination, point de vue...)
2. Régularité : les régions obtenues doivent être simples à manipuler (taille suffisante, forme régulière) [4]

Dans ce qui suit, certaines méthodes de segmentation seront présentées.

Méthodes de segmentation

On peut distinguer plusieurs types de segmentation, la segmentation globale et la segmentation locale sont deux méthodes très utilisées dans ce domaine, dans la première approche "segmentation globale" les méthodes de segmentation par histogrammes sont implémentées, ces dernières sont caractérisées par la rapidité du calcul ainsi que le peu de sensibilité au bruit (parasites), mais elles intègrent peu d'informations géométriques sur les régions de l'image, car la décision d'appartenance d'un pixel à une région dépend de l'image en entier. Quant à la deuxième approche "segmentation locale" où l'étiquetage d'un pixel dépend essentiellement de son voisinage ce qui fournit plus d'informations

géométriques ou typologiques sur les régions, cette dernière implémente généralement les méthodes de segmentation par régions.

Segmentation par seuillage (par histogrammes)

Comme toutes les méthodes de segmentation, l'objectif principal de celle-ci est de partitionner une image en régions en recherchant les frontières entre les régions en se basant essentiellement sur les discontinuités en niveaux de gris ou sur les propriétés des couleurs. L'idée avec cette approche est de définir des partitions ou des intervalles entre lesquels les pixels de l'image étudiée doivent être pris en compte pour pouvoir décider si un pixel appartient ou non à la partition ou l'intervalle défini au préalable, ces intervalles représentent en pratique les objets ou les régions distinctes présentes dans l'image à analyser et ils sont tracés en se basant sur des propriétés de cette dernière (niveaux de gris, couleurs) et à l'aide d'un certain seuil qui représente la valeur de la partition ou de l'intervalle qui permet de définir des limites pour les intervalles possibles, ces seuils sont en effet obtenus à partir de l'histogramme de l'image en sélectionnant les pics bien définis dans ce dernier, l'objectif est donc la sélection des objets présents.

L'image qui suit illustre la manière d'extraction des seuils ou des couleurs dominantes dans une image :

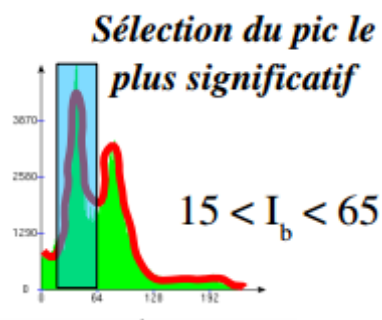


FIGURE 2.1 – Sélection du pic le plus significatif de l'histogramme de l'image

L'opération de l'extraction des pics est en effet répétée jusqu'à l'obtention de tous les pics du plus significatif (couleur dominante dans l'image) au moins significatif (couleur moins présente dans l'image) ainsi que l'obtention des intervalles pour la décision d'appartenance des pixels.

Afin d'éclaircir la notion du seuillage et de la définition des seuils, nous proposons un exemple d'une image en niveaux de gris contenant trois objets différents, de couleurs différentes un noir, un gris foncé et un gris clair :

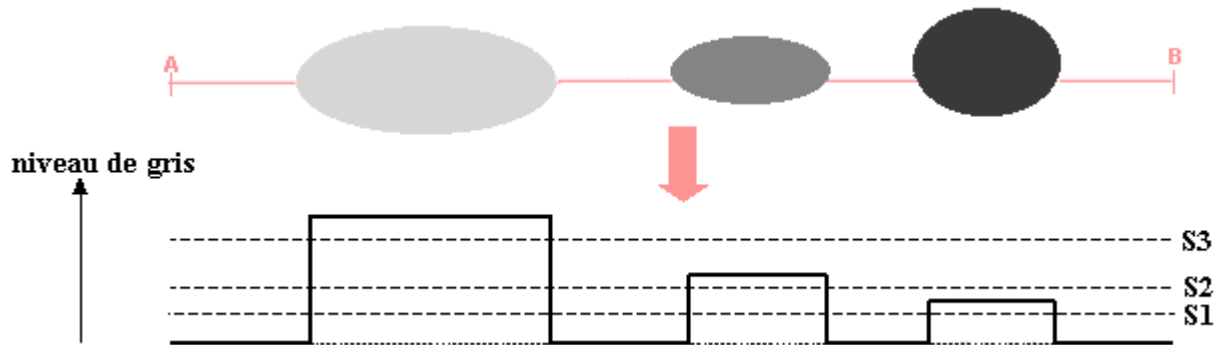


FIGURE 2.2 – Exemple illustratif de la définition des seuils

Dans cet exemple les seuils vont donc représenter des valeurs de niveau de gris qui permettent la partition de l'image en deux composantes, les objets qui sont définis par différents tons ou nuances d'une part et l'arrière-plan ou le fond de l'image d'une autre part. Cette opération retourne comme résultat une image binarisée, tous les pixels de l'image sont testés afin de pouvoir déterminer à quelle région ou intervalle ils appartiennent, les parties de l'image qui auront été sélectionné (appartiennent aux seuils définis) seront traduites en blanc et codées à 1 et le reste de l'image en noir codé à 0. Ce qui permet d'avoir une image segmentée et des objets séparés du fond de l'image[17].

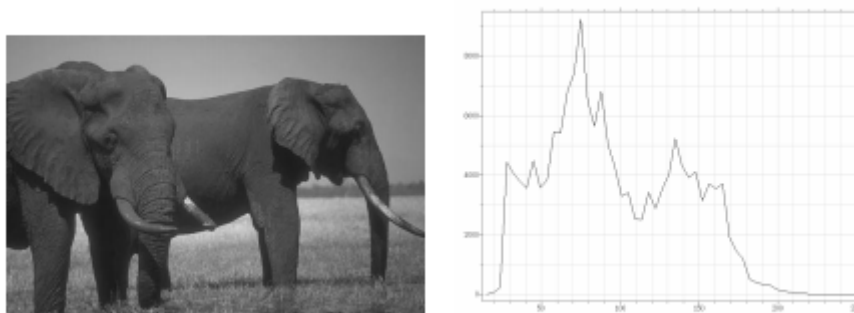


FIGURE 2.3 – A gauche : une image réelle. A droite : histogramme des niveaux de gris

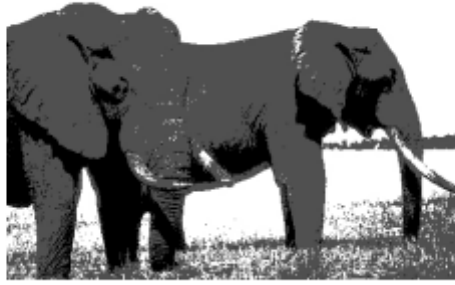


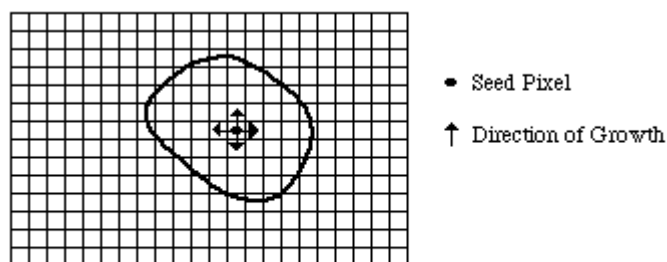
FIGURE 2.4 – Segmentation de l'image précédente à l'aide de deux seuils : 50 et 110

Segmentation par régions

Région Growing

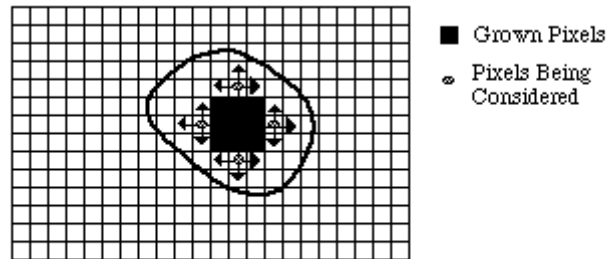
Au même titre que la méthode présentée plus haut, la segmentation par région vise également à séparer ou partitionner l'image en régions afin de distinguer les objets présents dans cette dernière, cette méthode permet de déterminer ou de séparer les différentes régions de l'image, elle est classée comme méthode de segmentation basée sur les pixels, en effet, le principe est de choisir un pixel de départ qui appartiendra à une région i et examiner par la suite les pixels voisins afin de déterminer si ces derniers appartiennent ou non à la région définie (la région i). Le processus de cette méthode se déroule suivant certaines étapes

- Une première série de petites zones sont choisis en se basant sur le critère de similarité.
- Choisir un pixel d'une manière arbitraire à l'intérieure de la zone choisit et le comparer par la suite aux pixels voisins.



(a) Start of Growing a Region

- Les pixels voisins sont analysés et ajoutés à la région si ils sont semblables (grande similarité), ce qui augmente la taille de la région.



(b) Growing Process After a Few Iterations

- Quand la croissance de la région s'arrête (plus de similarité entre les pixels), un autre pixel n'appartenant à aucune région est choisi et le même processus est répété.

L'ensemble de ces opérations sont répétées jusqu'à ce que tous les pixels de l'image appartiennent à une région dite i [15]

Plus formellement ces étapes sont implémentées par l'algorithme suivant :

- $\bigcup_{i=1}^n R_i = R$.
- R_i est une région connectée, $i=1,2,\dots,n$.
- $R_i \cap R_j = \emptyset$ pour tout $i=1,2,\dots,n$.
- $P(R_i) = \text{VRAIS}$ pour $i= 1,2,\dots,n$.
- $P(R_i \cup R_j) = \text{FALSE}$ pour toute région adjacente R_i et R_j .

Explication de l'algorithme

Sachant que $P(R_i)$ est une fonction quelconque appliquée aux pixels de la région.

- Signifie que la segmentation doit être terminée, tous les pixels doivent appartenir à une région i allant de 1 à n , l'union de toutes les régions R_i donne l'image R .

- (b) Exige que les points dans une région doivent être connectés dans un certain sens pré-défini (grande similarité entre les pixels).
- (c) Indique que les régions doivent être différentes l'une de l'autre (Pas de points en commun entre elles, les pixels appartenant à une région ne doivent pas figurer dans l'autre).
- (d) Etudie les propriétés de tous les pixels de l'image qui doivent correspondre à chaque région.
- (e) Illustre la différence entre la région R_i et R_j par rapport aux propriétés choisies pour chaque région (les propriétés choisies pour R_i diffèrent de celles choisies pour R_j)[9]



FIGURE 2.5 – illustration d'une segmentation basée régions

2.1.2 Détection basée sur la différence inter-images

La détection d'objet dans cette approche se base essentiellement sur le mouvement de l'objet à détecter (voiture) l'idée est en effet, d'effectuer une soustraction entre deux images successives fournies par un appareil d'acquisition numérique en temps réel. Afin de réaliser cette soustraction il est nécessaire de calculer l'intensité de l'image en question, ainsi, les pixels dont l'intensité résultante est proche de zéro sont assimilés comme étant les pixels du fond (arrière-plan de l'image) et donc l'analyse se focalise sur les zones

différentes du fond. De la même manière il est possible de soustraire à l'image courante une image de référence qui représentera uniquement le fond (aucun objet en mouvement n'est présent).

Ces approches basées sur le principe de l'inter-trame ou *image inter* qui désigne une image ou une trame appartenant à un flux qui a été encodé à partir d'un algorithme de prédiction. Elle dépend des images précédemment encodées afin de prédire la position des blocs dans l'image d'origine. L'objectif de cet algorithme est de définir un vecteur de mouvement qui traduit le déplacement d'un bloc dans une image déjà encodée dite de référence et sa position dans l'image courante. Cette prédiction est aussi appelée prédiction temporelle. La partie la plus délicate dans cette méthode est de définir le seuil global afin de pouvoir définir ou isoler l'arrière-plan de l'image (objets immobiles si on considère la caméra fixe), une fois ce dernier isolé l'analyse se focalise sur le reste des pixels de l'image et donne la possibilité de détecter différents objets à savoir, des véhicules.

2.1.3 Détection des zones d'intérêt

La détection de zones d'intérêt d'une image numérique consiste à relever ou mettre en évidence les blocs ou les zones d'une image jugés intéressants pour l'analyse ou pour des traitements ultérieurs, c'est une étape préliminaire de plusieurs processus de vision par ordinateur, ainsi, selon la méthode utilisée, les zones d'intérêt peuvent apparaître sous forme de points, de courbes continues (contours), etc.

Contours

Les contours dans une image sont définis à partir des points qui correspondent à des changements brutaux de l'intensité lumineuse, dans une image en niveaux de gris par exemple, un contour est caractérisé par un changement brutal de la valeur de l'intensité de gris. Ils incluent des discontinuités et traduisent en général des événements importants dans le cas d'objet en mouvement ou des changements dans les propriétés de l'image, en d'autres termes les contours peuvent être définis comme étant les frontières entre deux objets dans une image.

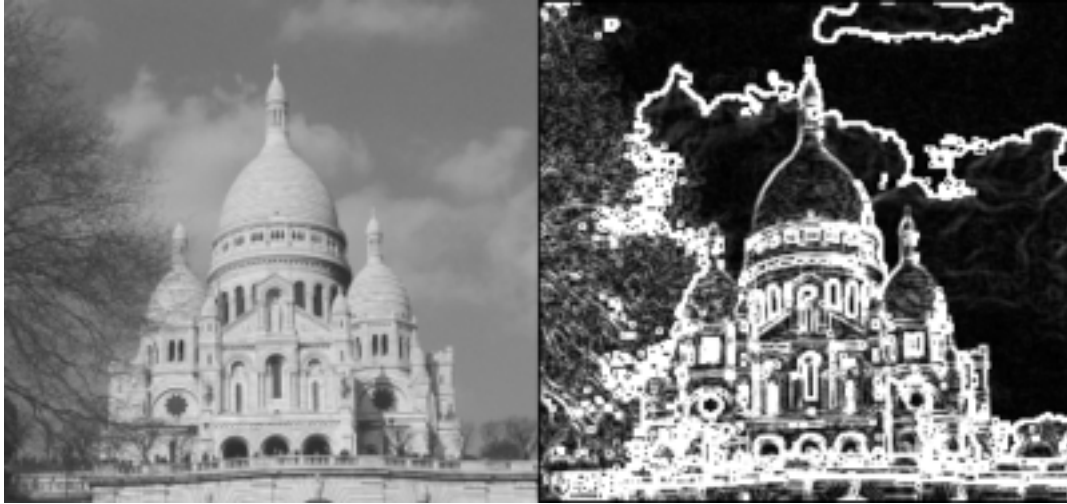


FIGURE 2.6 – Détection de contours

La détection des contours réduit de manière pertinente la quantité de données et élimine les informations qu'on peut juger moins distinctives, et ceci tout en préservant les propriétés structurelles de l'image. Ainsi, les contours constituent des indices riches pour toute interprétation ultérieure de l'image.

Concrètement, les contours proviennent des discontinuités de la fonction de rélectence¹ (texture, ombre) et des discontinuités de profondeur (bords de l'objet).

Le principe de détection repose essentiellement sur l'étude des dérivées de la fonction d'intensité de l'image : les extrémas locaux du gradient de la fonction d'intensité et les passages par zéro du laplacien, deux méthodes qui sont souvent mêlées malgré leur différence afin de produire un résultat idoine, cependant reste un problème qui réside dans la présence des bruits² dans les images ce qui nécessite un prétraitement qui consiste à effectuer un filtrage.

Points d'intérêt

Il existe plusieurs types de points d'intérêt pouvant se trouver dans une image, les coins, jonction en T et point de fortes variations de texture, l'image si après illustre les types cités plus haut :

1. Propriété de pouvoir réfléchir la lumière.

2. Phénomène parasite aléatoire dont les origines sont diverses (capteur, acquisition, lumière, ...)



FIGURE 2.7 – Différents types de points d'intérêt. La première image représente un coins, la seconde une jonction en un point T et en dernier des points de fortes variations de texture.

Les approches de détection de contour et de point d'intérêt peuvent se montrer complémentaires, en effet, les algorithmes de détection de points d'intérêt se basent généralement sur les résultats de la détection de contours ou des arêtes, ils se focalisent sur les points particuliers des contours. Ces point sont choisis selon des critères précis, ainsi les coins représentent les pixels de l'image où le contour (de dimension 1) change brutalement de direction (exemple des coins d'objets prenant des formes géométriques : carré, rectangle, triangle, etc.)

Plusieurs méthodes et algorithmes ont été implémentés pour la détection des zones d'intérêt, afin de pouvoir réduire considérablement le taux d'information à traiter et augmenter les performances, on cite par exemple : la méthode de Moravec qui consiste en une méthode de détection de coins qui se base sur l'analyse de chaque pixel de l'image et mesure par la suite les similarités entre différents patches voisins. La similarité est calculée en faisant la somme des différences entre les fenêtres choisies au carré, plus le résultat de la somme est faible plus la similarité est grande.

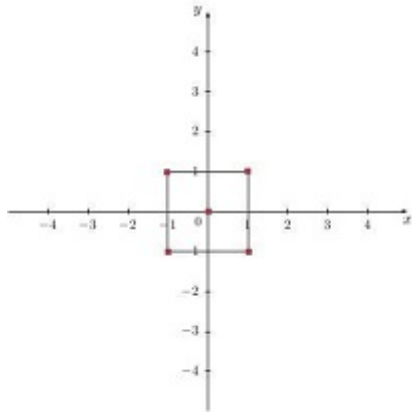


FIGURE 2.8 – Exemple illustratif de la manière de procéder au calcul de la similarité entre les zones adjacente. Il s'agit de calculer la dérivée de chaque pixel (point (x,y)) sur les huit directions (u,v) $\{ (1,0) (1,1) (0,1) (-1,1) (-1,0) (-1,-1) (0,-1) (1,-1) \}$

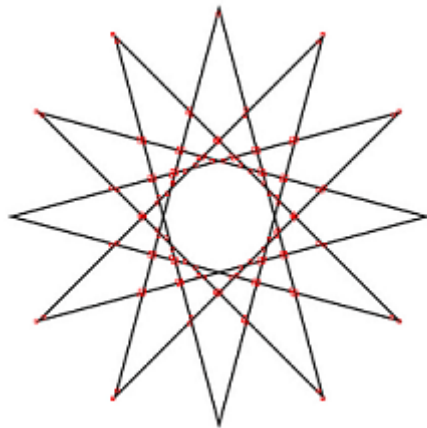


FIGURE 2.9 – Les points d'intérêt dans une forme géométrique

2.1.4 Méthode de Viola et Jones

Une avancée majeure dans le domaine de la détection a été réalisée par Viola et Jones en juillet 2001, en effet, leur méthode de détection d'objet dans une image numérique fait partie des toutes premières approches capable de détecter efficacement et en temps réel des objets dans une image. A l'origine cette dernière a été inventée pour détecter des visages, la simplicité de l'approche ainsi que son implémentation dans la bibliothèque d'OpenCV lui ont valu son succès et même son implantation dans la détection d'autres types d'objets (voiture : comme dans notre cas d'étude, avion, personne, etc.) La méthode de Viola et Jones est basée sur un apprentissage supervisé, des exemples de l'objet à recherché doivent être analysés à l'avance (phase d'entraînement de l'algorithme de détection avec un dataset), pour être classifiés. Par la suite, pendant l'analyse de l'image, des caractéristiques sont choisies par boosting, qui permet de classifier ces dernières, et de séparer les exemples positifs (l'objet qui doit être détecté) des exemples négatifs (les objets non recherchés dans l'image) par cascade de décision.

Caractéristiques de Haar

La technique de Viola et Jones permet d'analyser une image, sans avoir à étudier chaque pixel la constituant. Cette méthode est une approche basée sur l'apparence, qui consiste à parcourir l'ensemble de l'image en calculant un certain nombre de caractéristiques dans des zones rectangulaires qui se chevauchent, elle a la particularité d'utiliser nombreuses caractéristiques mais qui sont très simples (exemple : contours d'une voiture, forme d'une voiture, etc). Une caractéristique est une représentation synthétique et informative, calculée par la différence des sommes de pixels de deux ou plusieurs zones rectangulaires adjacentes et utilisée par la suite afin de pouvoir détecter l'objet recherché.

Ces caractéristiques sont illustrées sous forme de superposition de rectangles noirs et blancs, mais se présentent à vrai dire comme des nombres réels qui codent les variations du contenu pixellique à des positions données dans la fenêtre de détection, elles ont été établies pour repérer des coins précis de l'objet à détecter.

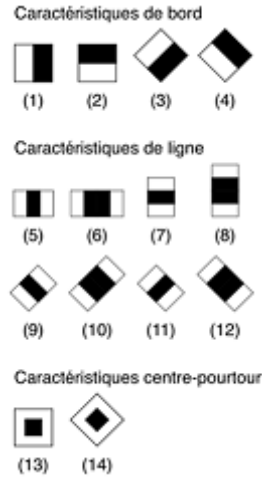



FIGURE 2.10 – Exemple des caractéristiques de Haar à 2, 3 ou 4 rectangles

Le résultat de l'application de chaque élément à une région particulière de l'image est donné par la somme des pixels qui se situent dans les rectangles noirs soustraite par la somme des pixels recouvrant partiellement les rectangles blancs. Les rectangles sont définis par leurs coordonnées en haut et à gauche (x,y) , leurs largeur W et leur hauteur H .

La somme des pixels qui se trouvent dans un rectangle R_i est représentée par la fonction $RecSum$ suivante :

$$Feature1 = \sum_{i=1}^N W_i * RecSum(r_i) \quad (2.1)$$

Les valeurs r_i , N et W_i sont choisies d'une manière arbitraire. Pour la caractéristique  avec une hauteur $H=2$ et une largeur $W=6$, l'issue de l'application caractéristique pour une région rectangulaire placée en (x,y) serait :

$$feature = -1 * RecSum(x, y, 6, 2) + \frac{6 * 2}{2 * 2} * RecSum(x + 2, y, 2, 2)$$

Avec $W_1 = \frac{Surface(r_0)}{Surface(r_1)}$

Image intégrale

Le calcul de caractéristiques d'une image numérique, parcourt tous les pixels couverts par les rectangles, par conséquent il requière une importante force de calcul ainsi qu'un temps d'acquisition conséquent, mais cet inconvénient a été contourné par viola et jones en intégrant la notion de l'image intégrale.

L'image intégrale est une représentation sous forme d'une image de la même taille que l'image d'origine, permettant de calculer rapidement et efficacement des sommes de valeurs dans des zones rectangulaires pour résumer les caractéristiques d'une image numérique, proposée initialement comme méthode d'infographie en 1984 par **Franklin C.Crow**, réutilisée par la suite par **viola et jones** dans leur technique de détection.

La valeur de l'image intégrale à un emplacement quelconque (x,y) dans une image est la somme de tous les pixels se trouvant au-dessus de lui et à sa gauche, y compris lui-même, dans ce qui suit des exemples de calcul de la valeur d'un pixel à partir d'une image intégrale sont illustrés :

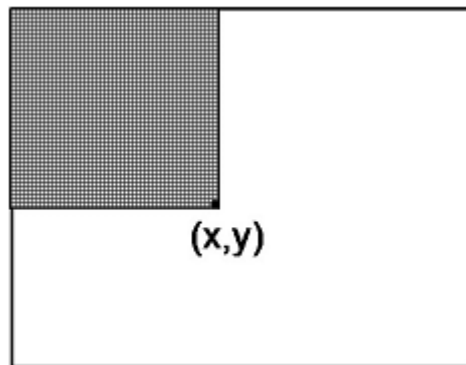
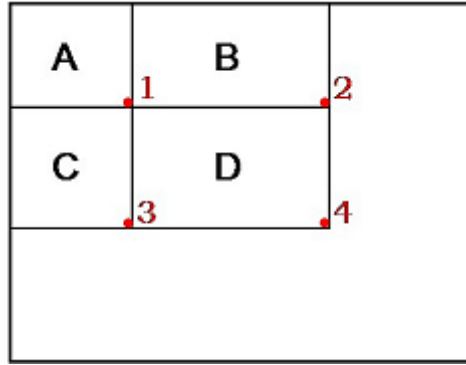


FIGURE 2.11 – Calcul de la valeur du pixel à l'aide de l'image intégrale. La valeur du pixel, dont les coordonnées, sont (x, y) est égale à la somme de celui-ci avec les pixels à gauche et au-dessus de lui.

Grâce à l'image intégrale, le calcul de la somme des valeurs des pixels appartenant à une zone rectangulaire de cette dernière s'effectue donc en accédant seulement à quatre pixels de celle-ci : soit le rectangle illustré dans la figure suivante :



Les valeurs, des pixels 1,2,3 et 4 sont calculés de la manière suivante :

$$ii1 = \text{sum}(A)$$

$$ii2 = \text{sum}(A) + \text{sum}(B)$$

$$ii3 = \text{sum}(A) + \text{sum}(C)$$

$$ii4 = \text{sum}(A) + \text{sum}(C) + \text{sum}(B) + \text{sum}(D)$$

D'une manière plus générale, une caractéristique pseudo-Haar à deux rectangles peut être calculée avec seulement 6 accès à l'image (deux points partagés) quant à une à trois rectangles est calculée avec seulement 8 accès, et ainsi de suite. Formellement l'image intégrale est définie à partir de l'image par :

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.2)$$

Boosting

Le calcul des caractéristiques de Haar nécessite une force de calcul conséquente, Viola et Jones ont introduit le concept de l'image intégrale, qui est une représentation intermédiaire de l'image d'entrée permettant de réduire le temps de calcul lié à l'application de ces filtres, cependant l'évaluation de l'ensemble des caractéristiques prendrait toujours une durée considérable ce qui pourrait sans nul doute réduire les performances et la robustesse de l'algorithme. Une solution envisageable est de choisir un nombre réduit de ces caractéristiques et qui vont jouer le rôle de classificateurs. Le boosting est une approche de l'apprentissage automatique basée sur la création d'une règle de prédiction très précise,

l'idée est de mettre en place des algorithmes afin de pouvoir sélectionner les caractéristiques les plus complémentaires et pouvoir produire des décisions précises, l'approche par boosting se présente comme suit :

1. Écrire un programme de prise de décision permettant de conclure à peu près correctement.
2. Appliquer ce programme a un échantillon des exemples d'apprentissages.
3. Trouver une première règle de décision.
4. Trouver une seconde règle de décision.
5. Répéter ces opérations T fois.

T représente le nombre nécessaire de décisions afin d'obtenir les résultats les plus précis. Les échantillons d'apprentissage sont choisis en se basant sur les exemples les plus difficiles (les exemples mal classés par les règles précédentes)

Les règles de décision sont combinées en utilisant un système de vote pondéré. Dans ce qui va suivre nous allons présenter brièvement l'un des algorithmes opérant par boosting

AdaBoost

L'algorithme AdaBoost est en effet une adaptation de la méthode qui fut, introduite par **Freund** et **Robert Schapire** en 1995, ce dernier offre la possibilité de traiter et de tirer des caractéristiques clés réduisant le taux de traitement, en d'autres termes, l'algorithme permet la sélection des caractéristique séparant le mieux les exemple positifs des exemples négatifs en prenant comme paramètres d'entrée toutes les caractéristiques des images (positives et négatives) étiquetées auparavant comme étant des images de voiture ou non.

L'algorithme traite l'ensemble de toutes les instances possibles communément appelé « espace d'instances » et noté χ . Ainsi une instance est une description d'un certain type qui est utilisée pour obtenir une classification.

Adaboost appelle un algorithme d'apprentissage faible d'une façon répétitive dans une série de tours $t=1...T$.

Une des idées principales de l'algorithme est de maintenir une distribution ou un ensemble de poids sur l'ensemble de la formation. Le poids de la distribution i au rond t est noté $D_t(i)$.

Dans un premier temps, tous les poids sont fixés d'une manière égale, mais sur chaque tour les poids des exemples mal classés sont augmentés de telle sorte que les classificateurs faibles se concentrent sur ces images mal classées à la prochaine itération. Le rôle des classificateurs faibles est de trouver une hypothèse faible $h_t : \mathcal{X} \rightarrow \{-1, +1\}$ appropriée à la distribution D_t et l'exactitude de cette hypothèse est mesurée par son erreur.

Cascade du classifieur

La méthode de Viola et Jones est une approche très complète et rigoureuse, qui se base sur le principe de la recherche exhaustive sur l'ensemble de l'image, elle teste en effet, la présence de l'objet en question (voiture) dans une fenêtre à toutes les positions et à plusieurs échelles en parcourant l'intégralité de l'image avec une fenêtre de 24x24 pixels, cette dernière change selon la taille de l'objet recherché.



FIGURE 2.12 – Utilisation des cascades de classificateurs pour la détection de voitures

Cette approche, bien qu'elle soit efficace, elle pose néanmoins quelques inconvénients, à savoir, les coûts élevés au niveau des calculs et du temps d'exécution ce qui réduit ses performances, en effet, en moyenne seulement 0.1% de toutes les sous-fenêtres (24x24) traitées sont positives (présence d'une voiture), pour remédier à cela une méthode a été mise en place, elle consiste en l'organisation de l'algorithme de détection en une cascade de classificateurs.

Un classificateur (à savoir une cascade de classificateurs boostés travaillant avec des fonctionnalités de haar-like) est formé avec quelques centaines d'images échantillons d'un objet particulier (voiture), appelé exemples positifs, qui sont mis à l'échelle à la même taille (par exemple, 20x20) et des exemples négatifs - images arbitraires de la même taille. Cette phase est appelée également phase d'entraînement de l'algorithme qui lui permet de décider si la zone explorée représente ou non l'objet (ou une partie de l'objet) recherché.

Viola et Jones ont donc mis ce principe de cascade de classificateurs en pratique de la manière suivante : au lieu de faire qu'avec un seul classificateur fort, ils ont voulu en faire plusieurs qui se succèdent du plus large au plus précis. En d'autres termes c'est la formation d'un arbre de classificateurs de telle sorte que chaque fenêtre d'une image prête à être testée est passée au premier classificateur de l'arbre (premier étage) et ce dernier prend donc une décision d'acceptation (1) si la fenêtre passée en paramètre présente bien l'objet recherché (présence d'une voiture), cette fenêtre est alors passée au prochain étage, ou de rejet (0) dans le cas d'exemple négatif (pas de voiture).

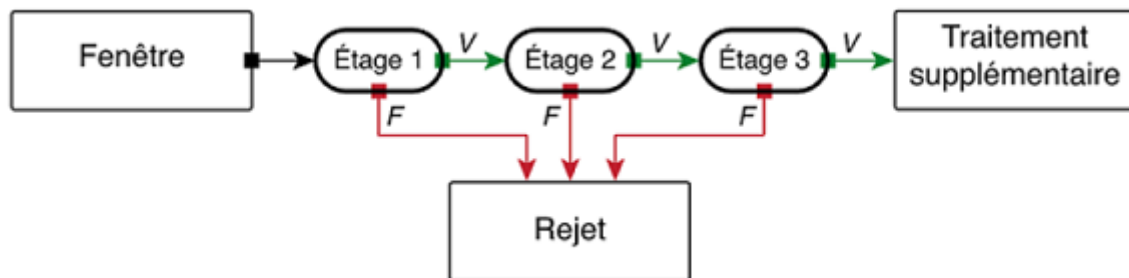


FIGURE 2.13 – Fonctionnement de la cascade de classifieur

Cette figure illustre la méthode de fonctionnement de la cascade de classifieur, une fenêtre positive (voiture) doit donc parcourir tous les étages définis par Viola et Jones à savoir 38 étages dont lesquels environs 6000 caractéristiques sont testées, ces dernières sont réparties sur les étages d'une manière progressive (les cinq premiers ont respectivement 1, 10, 25, 25 et 50 caractéristiques) à l'issue du 36^{ème} étage la fenêtre est donc considérée comme une image de voiture est donc encadrée.

2.2 Détection de plaques d'immatriculation

2.2.1 introduction

Dans la précédente partie, nous avons présenté la méthode de Viola et Jones ainsi que plusieurs concepts et approches utilisés pour la détection d'objet, cela nous a permis de comprendre et d'analyser différents algorithmes permettant la compréhension de la détection d'objets dans une image numérique en général et des voitures en particulier, cette approche bien qu'elle soit efficace et très utilisée dans le domaine de la détection d'objet, elle présente néanmoins quelques inconvénients.

En effet, comme illustré précédemment cette méthode repose essentiellement sur le principe des exemples positifs et négatifs (présence ou non de l'objet recherché dans l'image fournie), en d'autres termes, elle reçoit comme paramètre d'entrée des images étiquetées au préalable et effectue ensuite divers traitements afin de séparer les exemples positifs des exemples négatifs. Ces traitements requièrent généralement une assez bonne qualité de l'image à savoir une bonne résolution un fond clair néanmoins elle présente une contrainte assez importante : les images d'entraînement doivent toutes être de la même taille, par conséquent la plaque d'immatriculation pourra être détectée que quand elle est bien visible, ce qui nous conduit à explorer d'autres approches plus robustes afin de pouvoir extraire la plaque minéralogique d'une image de voiture avec un fond complexe et relativement mauvaise qualité.

La détection de plaque d'immatriculation est une étape cruciale, voire indispensable dans divers systèmes de sécurité routière, de surveillance de parking, contrôle routier ainsi que plusieurs autres applications et à propos, beaucoup de travaux ont été réalisés dans ce domaine.

Dans la section qui suivra on se propose d'effectuer une étude de certains travaux antérieurs, avant de présenter la méthode de détection de plaque adoptée dans cette thèse.

2.2.2 Méthode de détection de plaques d'immatriculation

Détection de texte dans une image L'extraction de l'information textuelle d'une image est une application du vaste domaine de la vision par ordinateur, elle peut s'avérer très

utile pour la compréhension du contenu de l'image ainsi que l'acquisition de données pouvant être utilisées pour diverses fins. En effet, la reconnaissance des caractères présents dans une plaque d'immatriculation fait recourir à des systèmes d'extraction de texte à partir d'une image.

Un système d'extraction de texte est un système qui reçoit comme paramètre d'entrée une image quelconque (en niveaux de gris, couleurs, compressées, etc.) et fournit en sortie la même image avec des rectangles englobant les textes présents dans le paramètre d'entrée en prenant en compte la typologie et les propriétés du texte (alignement du texte, la taille, le style, orientation, etc.).

Ce processus peut être décomposé en trois étapes essentielles :

- Détection : sert à déterminer pour chaque image si oui ou non un texte est présent dans cette dernière, cette étape fournit une réponse booléenne "1" si du texte est présent dans l'image "0" sinon.
- Localisation : cette étape sert à isoler les zones ou les parties de l'image où le texte ou les caractères sont présents et les regroupe dans des blocs rectangulaires.
- Extraction : l'objectif est d'analyser les blocs fournis par l'étape localisation pour isoler chaque caractère afin de pouvoir les faire passer un par un dans un système de reconnaissance, pour la lecture de ces derniers.

Dans ce contexte on distingue plusieurs approches et plusieurs travaux déjà réalisés dans ce sens, en effet, les premières recherches effectuées dans le domaine ont été effectuées par [Lienhart (1996)] qui propose une méthode basée sur la couleur, la texture, le contraste et la géométrie du texte, ce qui a donné naissance à d'autres travaux de recherche et d'autres approches, à savoir le travail de (Li et al., 2002) qui utilise des méthodes d'apprentissage pour la détection du texte. Le travail de (Wolf et al., 2001) qui présente un schéma de détection s'appuyant sur la mesure du gradient directionnel cumulé[32].

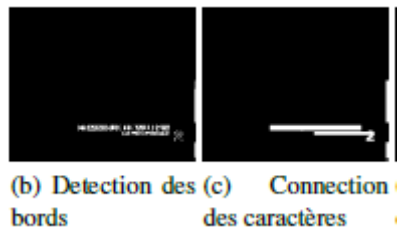
Dans ce qui suit, nous allons présenter une approche qui s'adapte le mieux aux plaques d'immatriculations, en effet, cette approche ne détecte que les caractères écrits horizontalement, pour ce faire, la technique présentée se base sur différentes caractéristiques de l'image comme la texture, la couleur, le contraste, etc.

La détection se fait suivant les étapes suivantes :

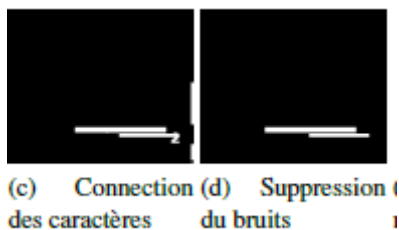
- Appliquer un filtre Sobel à l'image d'entrée, qui est un filtre qui prend une image en entrée et donne en sortie une image avec les contours en blanc sur un fond noir, dans notre cas ce filtre servira à détecter les principaux contours de l'image et les met en blanc et ignore le reste de l'image (fond noir).



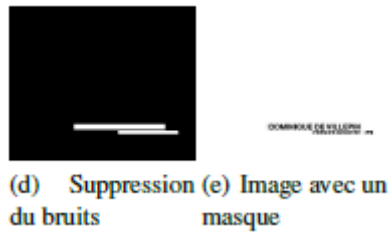
- Traiter l'image obtenue en appliquant une dilatation et une érosion qui permet de détecter les composantes connexes, et avoir une vue plus claire des caractères en question et les connecter entre eux.



- Après connexion des caractères, un autre filtre est appliqué à l'image afin de supprimer les bruits présents dans cette dernière.



- La suppression des bruits permet d’obtenir des composantes connexes (des boites rectangulaires) comportant le texte recherché, un dernier filtre (masque) est appliqué à ces boites, afin d’avoir des caractères plus claire et facilement lisible.



Limites de l’approche présentée

Les étapes illustrées précédemment permettent en effet une détection correcte du texte (caractère dans une image), néanmoins dans le cadre de notre travail quelques inconvénients se sont présentés à savoir :

- La mobilité du texte, en effet cette approche effectue une détection correcte que lorsque le texte en question est fixe.
- La complexité du fond : la présence d’un fond complexe pour les images de plaque d’immatriculation rend la tâche particulièrement difficile pour les méthodes de segmentation et du filtre appliqué, des erreurs de précision peuvent être générées dans ces cas de figure.
- La taille des caractères : la variation de la taille des caractères du texte en question pose en effet problème lors de la connexion des caractères entre eux, et cela peut générer des erreurs de détection.
- Le temps de calcul : l’apprentissage et la classification des caractères requièrent un temps de calcul important surtout quand la taille de l’image et celle des caractères est importante[29].

2.3 Conclusion

Cet état de l'art nous a permis de comprendre le cadre théorique général de la détection d'objets, et étudier des concepts et des propriétés reliés à ce domaine et qui nous seront très utiles lors de l'implémentation de notre application, en effet nous avons clôturé cette partie avec une étude critique d'une méthode existante utilisée pour la détection de plaque qui est "la détection de texte dans une image" il est donc nécessaire d'envisager une méthode plus adéquate, plus efficace et plus robuste, cette dernière sera donc présentée en détail dans le dernier chapitre de ce mémoire intitulé "implémentation et résultats".

Deuxième partie

État de l'art sur la reconnaissance d'objet

Chapitre 3

Reconnaissance visuelle par ordinateur

Contents

| | | |
|------------|---|-----------|
| 3.1 | Introduction | 52 |
| 3.2 | Définition | 52 |
| 3.3 | Principes d'un algorithme de reconnaissance visuelle | 53 |
| 3.3.1 | Reconnaissance | 54 |
| 3.3.2 | Les étapes fondamentales de reconnaissance | 54 |
| 3.4 | Les différentes méthodes de reconnaissance | 55 |
| 3.4.1 | Méthodes géométrique | 55 |
| 3.4.2 | Méthode globale | 56 |
| 3.4.3 | Méthode locale | 57 |
| 3.5 | Conclusion | 58 |
| 3.6 | Reconnaissance optique des caractères | 58 |
| 3.6.1 | Introduction | 58 |
| 3.6.2 | Définition | 59 |
| 3.6.3 | Historique | 60 |
| 3.6.4 | Domaines d'application des OCR | 61 |
| 3.6.5 | Fonctionnement d'un OCR | 62 |
| 3.6.6 | Conclusion | 68 |

3.1 Introduction

Les plaques d'immatriculation sont l'un des principaux traits utilisés pour l'identification de véhicules ainsi que l'acquisition de diverses informations concernant ces derniers (propriétaire, antécédents, wilaya d'appartenance...) ce qui permet de faciliter et de gérer plusieurs tâches à savoir la gestion du trafic routier, la vérification automatiques et rapide d'antécédent d'un véhicule, etc.

Par le passé, les tâches ou les fonctionnalités citées plus haut étaient gérées manuellement par le biais de l'humain et la capacité du son cerveau à reconnaître d'une manière quasi parfaite d'un simple coup d'œil un nombre immense d'objets, de caractères et plusieurs autres aspects du monde extérieur, même si ces derniers peuvent se trouver parfois à une grande distance, dans des conditions de luminosité extrême, a captivé l'attention des chercheurs et des développeurs, ce qui les a poussé à œuvrer afin de reproduire certaines de ces fonctionnalités pour diverses applications qui réduisent considérablement la quantité de travail pour l'être humain avec un gain de temps immense.

Dans la suite de cette partie, nous allons passer en revue les différentes méthodes de reconnaissance ainsi que l'approche retenue pour la reconnaissance des caractères de nos éventuelles plaques d'immatriculation.

3.2 Définition

Dans le cadre de la reconnaissance d'objets en vision par ordinateur, le principal souci est de différencier les catégories d'objets (associées aux **Formes** : objets carrés, rectangulaires) et les instances d'objets (associées aux **Choses** : visage, lettre, chiffre).

Un cerveau humain adulte est capable de reconnaître plus 10000 catégories d'objets (forme géométrique, personnes, caractères) et une quantité bien plus importante d'instances d'objets (rectangle, visage, lettres et chiffres), cette reconnaissance s'effectue sans effort et avec robustesse même dans des conditions de visualisations dégradées parfois, pour cela le cerveau passe par une étape appelée apprentissage, pour l'initier à une nouvelle catégorie d'objets, par exemple il est nécessaire de faire appel à un nombre significatif

d'images représentant des instances de cette catégorie, en revanche l'initiation à une nouvelle instance d'objet peut s'effectuer en faisant appel à une seule image illustrant cette dernière.

L'intérêt de cette démarche est de justement programmer un ordinateur de telle sorte à lui apprendre à reconnaître des catégories et des instances d'objets, après la phase d'apprentissage, l'ordinateur doit être capable d'identifier si une telle catégorie ou instance d'objet est présente ou non dans l'image.

La reconnaissance visuelle par ordinateur, consiste à déterminer à l'aide d'un algorithme quel objet se trouve dans une image, l'algorithme en question prend en paramètre d'entrée une image, et fourni en sortie la classe de l'objet présent dans cette dernière s'il s'agit de reconnaissance de catégorie d'objet ou la référence d'un objet précis dans le cas de reconnaissance d'instance d'objet.

D'après la définition de la reconnaissance présentée plus haut, on peut en effet facilement confondre entre les deux principaux concepts de notre travail à savoir la reconnaissance et la détection, mais la tâche de reconnaissance diffère de celle de la détection, où cette dernière consiste à déterminer ou localiser précisément l'emplacement ou la position de l'objet recherché quant à la première consiste à trouver les correspondances point à point entre deux objets présentés dans telle sorte à pouvoir les classifier.

3.3 Principes d'un algorithme de reconnaissance visuelle

Il existe plusieurs méthodes et algorithmes de reconnaissance d'objets qui adoptent des approches et des principes très différents pour arriver à la reconnaissance voulue, néanmoins, ces derniers partagent une trame commune qui sera présentée dans ce qui suit.

3.3.1 Reconnaissance

Au même titre que la détection, la reconnaissance des formes ou de motifs est une branche de l'intelligence artificielle qui est un ensemble de techniques et méthodes visant à identifier une catégorie ou une instance d'objet informatique à partir d'une image, de telle sorte à pouvoir prendre une décision à propos de la catégorie d'appartenance de ce dernier (code barre, lettre, chiffre, visage, empreintes digitales, etc).

Le mot reconnaissance se décompose naturellement en deux parties, à savoir le radicale du mot qui est « connaître » et le préfixe « re ». La partie connaître représente en pratique la première partie du traitement qui est une phase d'acquisition de connaissances, il s'agit d'apprendre à identifier ou caractériser l'objet en question de façon à pouvoir répondre à quelques questions comme : que-ce qui fait que je puisse reconnaître un caractère ? Que-ce qui le différencie des autres objets ? Etc. la deuxième phase re consiste à utiliser les connaissances acquises. Il s'agit d'analyser d'autres objets afin de pouvoir décider s'ils appartiennent ou pas à la catégorie apprise dans la première phase du traitement.

Cette technologie nous sera utile dans la suite de notre travail afin de pouvoir reconnaître les lettres et les chiffres composant les plaques d'immatriculation détectées précédemment.

3.3.2 Les étapes fondamentales de reconnaissance

Comme introduit dans les sections précédentes, il existe plusieurs méthodes et approches de reconnaissance, toutes avec des algorithmes différents, mais suivent toutes une logique et un processus quasi-similaire. Trois étapes fondamentales de ce raisonnement sont en effet dégagées.

La première représente la phase d'acquisition, en d'autres termes l'entraînement de l'algorithme, cela se fait en présentant un nombre très important d'images représentant l'objet qu'on veut reconnaître à l'algorithme, afin que ce dernier puisse relever le maximum de caractéristiques possibles lui permettant de reconnaître l'objet voulu.

La seconde, apparaît dans la notion de mesure de similarité qui s'occupe de mesurer à quel point un objet présent dans l'image donnée en paramètre d'entrée correspond à un modèle considéré (modèle d'entraînement).

La troisième et dernière étape, consiste en la reconstitution de l'image.

Dans les deux premières étapes le contenu original de l'image (les pixels) est ignoré pour des faims de reconnaissance, on peut par exemple choisir d'ignorer les pixels au profit des droites présentes dans l'image ou encore au profit des gradients, etc. Le choix du modèle à prendre en considération (droite ou gradients de l'image) aux dépend des pixels dépend de la catégorie de l'objet à reconnaître. La dernière étape du processus est donc consacrée à la reconstitution de l'image d'origine.

3.4 Les différentes méthodes de reconnaissance

3.4.1 Méthodes géométrique

Les méthodes géométriques furent parmi les premières méthodes de reconnaissance d'instances d'objets implémentés, ces dernières se basent sur un modèle 3D de l'objet ou de l'instance recherchée. Elle consiste à représenter les objets de référence par leurs contours et d'étudier par la suite les propriétés de ces derniers.

Le principe est d'utiliser des méthodes d'alignement de primitives géométriques qui disposent d'un modèle 3D de l'objet recherché.

Les primitives (des droites par exemple) composant ce modèle 3D sont alignées sur les primitives détectées au préalable dans l'image, et la qualité de l'alignement détermine si l'objet correspond ou non à l'objet recherché. On peut donner l'exemple de reconnaissance d'un chiffre : la méthode d'alignement possède un modèle précis de cette dernière (l'alignement des droites composant le chiffre en question) pour pouvoir reconnaître si un chiffre détecté dans une image inconnue correspond ou non au gabarit connu dans la base de données, les primitives détectées sont alignées sur le modèle existant et la qualité de ce dernier détermine si oui ou non le caractère détecté correspond au chiffre connue au préalable.

Le calcul des correspondances entre les primitives des modèles existants et ceux détectés, peut s'avérer incomplet dans certains cas, ces méthodes font donc appel à des heuristiques afin de réduire la complexité du problème.

Les méthodes géométrique prennent en compte ou considère que les contours des objets, et donc l'intérieur de ces derniers est ignoré et pas pris en compte ce qui fait que ces

méthodes obtiennent une grande robustesse face aux changements d'illumination et de texture des objets. Une autre force de ces dernières réside dans l'utilisation d'un modèle 3D (toutes les primitives de l'objet sont représentées sous différents angles) cela permet en effet, de gérer toutes les occultations ainsi qu'un fond chargé d'une manière très élégante. Cependant ces méthodes souffrent de deux problèmes majeurs. Le premier, est que certaines catégories d'objet comme les arbres présentent des contours très complexes, de ce fait ces derniers ne peuvent être définis que par leurs contours mais ils ont besoins d'autres paramètres pour être correctement décrits.

Comme deuxième point, avec des conditions réelles d'acquisition d'image, il est souvent impossible d'obtenir les contours et les tailles précises des objets photographiés (cela dépend de l'angle de l'appareil d'acquisition, de la distance entre l'appareil et l'objet photographié, etc.) ces méthodes ne sont donc pas robustes à une détection imprécise des contours.

3.4.2 Méthode globale

Les méthodes de reconnaissance globale utilisent l'information extraite d'une image prise dans sa globalité (en entier), cette information appelé également signature d'une image peut représenter divers paramètres de l'image comme la distribution des couleurs dans l'image ou encore les différentes nuances de gris présentes si l'image est représentée en niveau de gris, la signature peut également être la suite des pixels mis les uns à côté des autres dans un vecteur colonne.

En d'autres termes, cette approche extrait un paramètre de l'image de tous les pixels présents dans cette dernière et le représente sous forme d'un vecteur de grande dimension utilisé par la suite comme repère de reconnaissance sur d'autres images.

Le principe de la méthode est en effet, de calculer la signature d'une image inconnue (image passée en paramètre d'entrée à l'algorithme) et procéder par la suite à la comparaison avec les signatures calculées au préalable d'autres images dites de références représentant des instances d'objets, l'objet reconnu ou prédit est alors l'objet de référence le plus similaire à la signature de l'image inconnue.

Ce qui fait la force des méthodes globales c'est leur simplicité et surtout leur robustesse face à certaines difficultés de reconnaissance comme les modifications des conditions

d'illumination et de contraste. Quant au point négatif de l'approche c'est les conditions à remplir afin que cette méthode soit robuste. En effet, pour que cette approche puisse reconnaître des objets malgré un changement significatif de luminosité, il faut mettre à sa disposition une base de références contenant un très grand nombre d'images sous différentes conditions d'illumination.

Un autre inconvénient peut être ajouté à cela, en effet, la signature intègre l'influence ou les propriétés de tous les pixels de l'image ce qui impose du traitement en plus (procède au traitement de toutes les régions de l'image même celles qui ne sont pas intéressantes à l'analyse), ce qui fait que cette méthode donne de mauvais résultats en présence de fond chargé (les pixels de fond sont tous pris en considération ce qui fausse la reconnaissance de l'objet voulu lors de la comparaison des signatures).

3.4.3 Méthode locale

Les méthodes locales sont venues principalement pour pallier les principaux défauts des méthodes globales, ces dernières se basent sur les caractéristiques locales dans une image contrairement aux approches globales qui prennent l'image dans sa globalité, elles ne considèrent plus l'image comme un tout mais comme une collection de régions locales, qui sont le plus souvent des parties d'images carrées ou rectangulaires, pouvant contenir une centaine de pixels ou bien plus. Ces régions locales s'affranchissent des difficultés rencontrées sur des images entières, en effet, les modifications d'illumination sont localement uniformes (dans de petites régions les changements de luminosité sont insignifiants et négligeables), et la luminosité et le contraste peuvent donc être normalisés. Le problème du fond chargé peut également être pallié avec les régions locales découpées, effectivement, quand ces régions sont découpées à l'intérieur des objets ou prennent l'objet entier sans l'arrière-plan, elles ne subissent pas d'influence du fond chargé, et en sont donc indépendantes.

Le principe des méthodes locales est de représenter les images de référence ou d'entraînement sous forme d'un ensemble de régions locales et de relever pour chacune de ces dernières des descripteurs (des caractéristiques propres à l'objet à reconnaître) et stocker ces descripteurs dans une base de données. Lors de la reconnaissance, les descripteurs des régions locales sélectionnées dans l'image sont recherchés dans les images inconnues, ce

qui permet de voter si oui ou non l'image dite inconnue contient l'objet recherché en se basant sur la similarité entre les descripteurs des images de référence et ceux des images inconnues.

En raison de leur rapidité, de leur bonne gestion du fond chargé ainsi que leurs robustesse face aux modifications de luminosité, les méthodes locales ont aujourd'hui la préférence de la communauté de la reconnaissance visuelle d'instances d'objets.

3.5 Conclusion

La reconnaissance d'objets est un domaine très utilisé et mis en pratique dans diverses applications d'intelligence artificielle, dans le cadre de notre application on vise à reconnaître de petites occurrences à savoir les lettres et les chiffres composant une plaque d'immatriculation, en effet, les méthodes présentées précédemment, en particulier les méthodes locales ont justement lancé et ouvert le domaine à beaucoup d'autres travaux de recherche concernant les petites occurrences (lettre et chiffres), ce qui a donné naissance à la Reconnaissance Optique des Caractères (ROC ou OCR en anglais Optical Character Recognition) ces derniers seront présenté dans la section qui suit.

3.6 Reconnaissance optique des caractères

3.6.1 Introduction

Dans la précédente partie de notre application s'intitulant "lecture automatique des plaques d'immatriculation", nous avons présenté la base théorique et pratique d'un détecteur de plaque d'immatriculation ce qui nous à permit d'isoler dans des images les zones intéressantes pour la suite de l'application, qui sont dans notre cas d'étude les plaques d'immatriculation.

La seconde partie du travail, consiste en la lecture de ces dernières, en d'autres termes transformer l'image qui représente un ensemble de pixels en un fichier texte modifiable et exploitable, cela fait appel à une technologie appelée OCR (Optical Character Recognition) ou ROC (Reconnaissance Optique des Caractères).

Le principe de cette technologie est en effet, de lire un document (une image) et détecter les formes présentes dans ce dernier pour les comparer ensuite à une bibliothèque intégrée contenant des milliers d'échantillons possibles pour chaque forme, cette comparaison vise à faire correspondre chaque forme détectée à une forme de la bibliothèque pour en fournir le caractère alphanumérique correspondant. Dans le cas où des erreurs surviennent lors de la reconnaissance, l'OCR fait recours à un autre procédé qui compare alors le mot entier au contenu d'un dictionnaire intégré pour en déduire l'équivalence la plus proche et ainsi corriger le caractère mal lu.

La technologie de reconnaissance optique de caractères (OCR) est la clé permettant d'accéder aux informations papier imprimées et les modifier pour une exploitation antérieure[19].

3.6.2 Définition

L'OCR ou reconnaissance optique des caractères encore appelé vidéo-codage, est une technologie qui désigne les procédés informatiques permettant de reconnaître dans une image les caractères alphanumériques (lettre ou chiffre) composant un texte ou autre information textuelle, elle permet de convertir différents types de documents tels que les documents papiers scannés, les fichiers PDF ou encore des images ou photos numériques, vers des formats modifiables et exploitables.

L'OCR procède en effet à l'extraction des données contenues dans une image en restituant chaque caractère de de l'image afin de reconstruire le mot ou l'information en entier, cela en passant par une étape appelée apprentissage où des échantillons connus pour chaque caractère sont collectés pour pouvoir faire correspondre aux caractères détectés les formes existantes dans la mémoire, en effet la précision de reconnaissance dépend du nombre d'échantillons collectés pour chaque caractère (plus le nombre d'échantillons est important, plus la précision de reconnaissance augmente) aujourd'hui, il est courant de trouver des systèmes « intelligents » qui peuvent reconnaître la plupart des polices avec un haut niveau de précision[2].

En d'autres termes, le procédé OCR fournit aux systèmes d'imagerie et de numérisation (appareil photo, scanner) la capacité de transformer des images de caractères imprimés à la machine en caractère lisible et compréhensible par la machine.

3.6.3 Historique

Avec l'apparition des appareils numériques, plusieurs données et document sont stockés dans différentes machines sous format numérique (image), ce qui a fait ressentir le besoin d'un dispositif permettant de traduire ces documents numériques en des fichiers textes modifiables et faciles à manipuler, de ce fait l'idée première des OCR peut être attribuée à l'origine à la création de dispositifs de lecture pour les aveugles, en effet en 1914 **Emanuel Goldberg** a développé une machine qui lit les caractères et les convertit en code télégraphique standard. En même temps, **Edmund Fournier** d'Albe a développé le "Optophone" qui est un scanner portable qui lorsqu'il est déplacé sur une page imprimée, produit des tonalités qui correspondent à des lettres ou des caractères spécifiques. De ces idées une première machine OCR fut créée par l'ingénieur allemand appelé **Gustav Tauschek**, en 1929 cette machine est baptisée sous le nom de "Reading Machine" (machine de lecture) qui est un dispositif mécanique qui utilise un modèle correspondant à un photo-détecteur (détecteur photosensible qui pointait une lumière sur un mot lorsqu'il correspondait à un gabarit ou un modèle contenu dans sa mémoire), cette machine a été suivie et améliorée par **Paul Handel** qui a obtenu un brevet américain sur L'OCR en 1933.

En 1974 **Ray Kurzweil** a démarré l'entreprise « Kurzweil Computer Products » qui a poursuivi le développement d'OCR arrivant à une machine capable de reconnaître le texte imprimé dans pratiquement toutes les polices, ce dernier après l'obtention de ces résultats sur les OCR a décidé que la meilleure application de cette technologie serait de créer une machine de lecture pour les aveugles, ce qui permettrait aux aveugles d'avoir un ordinateur pouvant lire le texte à partir de documents imprimé à haute voix, le 13 Janvier 1976, le produit fini avec succès a été dévoilé lors d'une conférence dirigée par Kurzweil et les dirigeants de la Fédération nationale des aveugles, deux ans plus tard le produit a commencé à être commercialisé et vendu, ce qui a ouvert le domaine à plusieurs chercheurs et innovateurs, ce qui a considérablement élargi les domaines d'application de ce dernier.

Dans les années 2000, l'OCR a été mis en ligne en tant que service (WebOCR), dans des applications mobiles comme la traduction en temps réel des signes en langue étrangère

sur un smartphone.

Par la suite, divers systèmes OCR été disponibles pour les systèmes d'écriture les plus courants, y compris le latin, cyrillique, arabe, hébreu, Indic, chinois, japonais et diverses applications ont été développés dans ce sens comme la reconnaissance des chiffres et des caractères spéciaux comme dans notre cas d'étude.

3.6.4 Domaines d'application des OCR

L'utilité et l'évolution des OCR leur ont valu leurs mises au point et leur application dans divers domaines, on cite parmi ceux-là quelques domaines d'application :

- La saisie des données pour les documents commerciaux : comme les chèques saisis à la main, les relevés bancaires, les factures, etc. Les OCR permettent une lecture automatique de ces derniers ce qui réduit considérablement la quantité de travail et offre un gain de temps.
- La lecture d'image en temps réel, avec le développement des applications mobiles diverses informations sont stockées sous format numérique, ainsi le besoin de conversion d'image comportant du texte se fait souvent ressentir d'où l'utilité de cette technologie d'OCR.
- L'aide aux utilisateurs aveugles ou malvoyants, en effet les OCR permettent de lire et fournir les lettres et les mots d'un document imprimé, ce qui permet leurs manipulations, de telle sorte à produire des documents sous format typhlofil¹ ou encore produire des sons prononçant les lettres ou les mots reconnus.
- Reconnaissance automatique des plaques d'immatriculation, pour diverses utilisations antérieures principalement pour des faims de sécurité, en effet, les OCR permettant la lecture des caractères d'une plaque ce qui permet l'identification du véhicule et par la suite de la personne possédant ce dernier ce qui permet la résolution de plusieurs fonctionnalités, à savoir la gestion automatique du trafic routier, des parkings, etc [8].

1. écriture pour aveugle

3.6.5 Fonctionnement d'un OCR

Il existe plusieurs logiciels et implémentations d'OCR, ces derniers ont chacun une technologie et un raisonnement particulier, néanmoins ces derniers partagent tous un raisonnement quasi-identique et suivent des étapes d'une manière consécutive en utilisant leurs différentes technologies, ces étapes seront présentées dans ce qui suit :

Les prétraitements

Redressement de l'image

la première étape de reconnaissance par OCR consiste à numériser l'information destinée à la lecture, cette étape permet de représenter les données à lire sous la forme d'une image numérique.

C'est une étape très importante dans la mesure où elle s'occupe de l'acquisition et de la préparation des informations nécessaires, ainsi que du format de stockage des images, la qualité des résultats obtenus découlent en effet de cette dernière, il est donc nécessaire de prendre toutes les précautions permettant une bonne numérisation de la donnée, mais malgré tous les soins apportés, il arrive parfois que l'image soit de travers et ceci pour multiple raisons à savoir : l'angle de prise de l'appareil d'acquisition, les conditions dans lesquelles se trouvait l'objet à numériser (dans le cas d'une plaque d'immatriculation si le véhicule est en mouvement l'image peut-être de travers).

Le traitement à effectuer consiste à redresser l'image à l'aide d'algorithmes de correction d'orientation afin de rendre la lecture des caractères plus facile.

Si dessous, des exemples illustratifs du redressement d'un document scanné et d'une image.

Document texte :



FIGURE 3.1 – Image initiale



FIGURE 3.2 – Image après redressement

Exemple de redressement d'une plaque d'immatriculation :



FIGURE 3.3 – Image initiale



FIGURE 3.4 – Image après redressement

Nettoyage du bruit

La première étape du prétraitement avant le passage à l'OCR est la numérisation ou l'acquisition de l'image, celle-ci engendre généralement l'apparition de bruit numérique qui est un phénomène constitué de pixels parasites de couleur, qui se superposent à l'image. Le bruit est gênant pour les traitements postérieurs dans la mesure où il fait perdre du détail à l'image, il peut aller jusqu'à la consteller de points colorés très visibles. Ces parasites qui nuisent à l'image peuvent avoir plusieurs sources, telles que les conditions dans lesquelles se trouve la plaque à photographier, notamment des tâches ou de la poussière présente sur la plaque ou un fond complexe présentant plusieurs couleurs, ainsi que les conditions dans lesquelles se trouve l'appareil d'acquisition comme les différents réglages applicables sur ce dernier (zoom, résolution, etc) ajouter à ces raisons les conditions naturelles comme les variations de la lumière.

Le bruit cause des ennuis considérables à la reconnaissance, en effet à cause de ce dernier des caractères peuvent être reconnus à la place d'autres ou encore un pixel noir peut être assimilé comme une virgule et bien d'autres contraintes.

Dans un souci d'optimisation des résultats de la reconnaissance, un traitement de réduction voir suppression des parasites "bruit" est nécessaire afin d'améliorer la netteté de l'image à traiter.

Si-suivant un exemple illustratif de la suppression des parasites bruit dans des images affectées

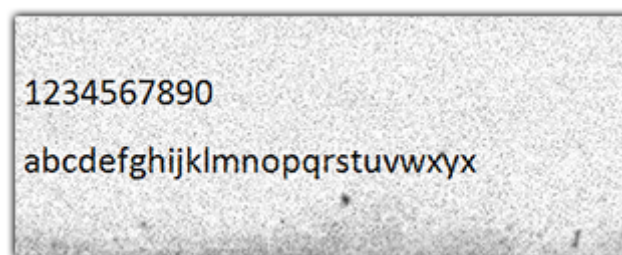


FIGURE 3.5 – Image affectée (présence de parasites bruits)

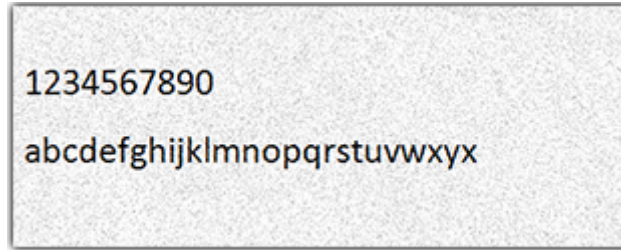


FIGURE 3.6 – Image nettoyée (suppression des parasites)

Le passage à l'OCR

A ce moment du traitement, l'OCR a pour mission de transformer l'image en la segmentant (découper l'image) en un ensemble de sous-ensembles de pixels (point noir ou blanc) et il garde en mémoire la position de chaque sous-ensemble dans le document (l'image traitée). Par la suite il passe à la reconnaissance du caractère correspondant à chaque ensemble de pixels et forme donc des lettres ou des chiffres.

Exemple d'un document passé à un OCR :

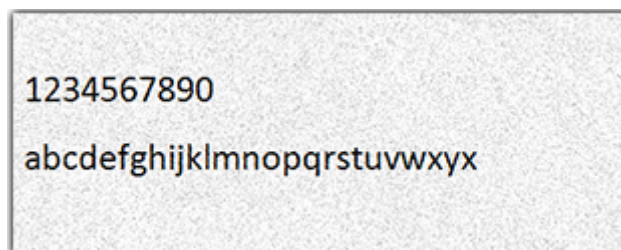


FIGURE 3.7 – Image passée à l'OCR

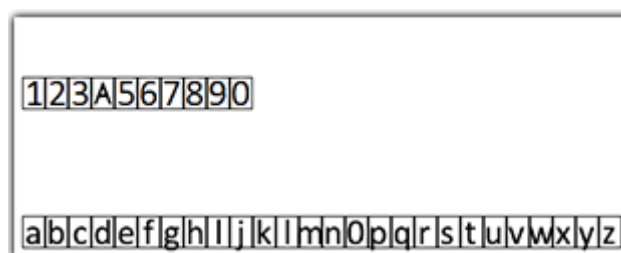


FIGURE 3.8 – schéma des caractères reconnus

Exemple d'une image de plaque d'immatriculation passée à l'OCR :



FIGURE 3.9 – Image de plaque passée à l'OCR



FIGURE 3.10 – Image transformée par l'OCR (segmentée)



FIGURE 3.11 – Lecture des caractères (présence de quelques erreurs)

Retraitement de l'OCR

La ressemblance et la quasi-similarité entre certains chiffres et lettres pose généralement un souci de reconnaissance à l'OCR, il n'arrive pas toujours à faire la différence entre certaines lettres et certains chiffres qui ont des formes et traits similaires.

Dans ce qui suit, quelques exemples seront cités :

- Les formes rondes : elles peuvent représenter plusieurs caractères différents : le “o” lettre minuscule, le “O” lettre majuscule, “Q” la lettre “q”, “°” l'exposant ainsi que le chiffre zéro “0”.
- Les bâtons : ils peuvent représenter également plusieurs caractères différents à savoir la lettre “I” en majuscule et en minuscule, la lettre “L” en minuscule, le chiffre “1”, l’opérateur de division “/”, etc.

En raison des contraintes citées plus haut, un retraitement intelligent est opéré dans le but d’optimiser et de corriger les résultats.

Exemples de traitement intelligent :

- Si dans une série de chiffres se glisse un “I” majuscule ou minuscule ou encore la lettre L minuscule, il sera remplacé par le chiffre le plus similaire, le chiffre “1”.
- Si dans une série de lettres se glisse le chiffre “0” il sera remplacé par la lettre “o”.

Exemples illustratifs :

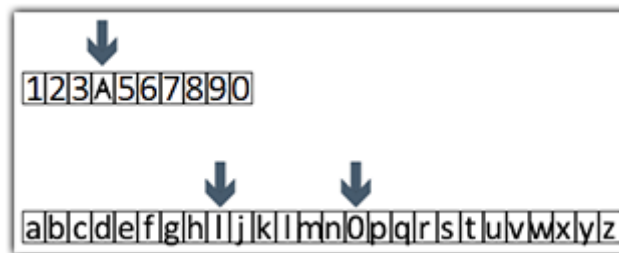


FIGURE 3.12 – Détection des incohérences

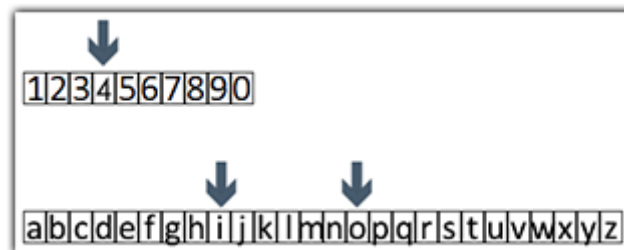


FIGURE 3.13 – Correction des incohérences

Exemple avec une image d'une plaque :

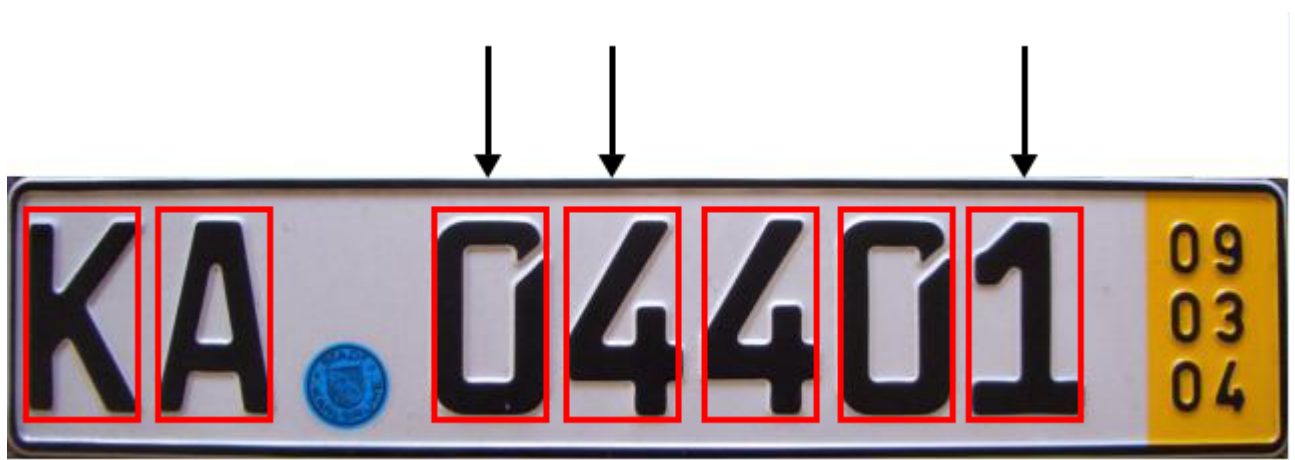


FIGURE 3.14 – Détection erreurs du premier passage à l'OCR



FIGURE 3.15 – Correction des erreurs détectées avec un traitement intelligent

3.6.6 Conclusion

Les étapes illustrées précédemment, résument brièvement des généralités ainsi que le fonctionnement d'un OCR. Les OCR présentent des taux de reconnaissance allant de 80 à 100% (reconnaissance parfaite sans erreurs) si on considère l'image de bonne qualité et avec de correcte prétraitements.

Ces chiffres peuvent considérablement baisser avec des conditions moins favorables (des tâches sur l'image, présence de beaucoup de pixels parasite, etc.).

De ce fait il est donc recommandé de prendre les précautions nécessaires d'acquisitions ainsi que d'effectuer les prétraitements nécessaires à l'image avant de passer à la lecture afin d'obtenir une reconnaissance correcte et optimale.

Chapitre 4

Apprentissage et classification

Contents

| | | |
|------------|---|-----------|
| 4.1 | Apprentissage | 70 |
| 4.1.1 | Introduction | 70 |
| 4.1.2 | Apprentissage naturel | 71 |
| 4.1.3 | Apprentissage artificiel (automatique) | 72 |
| 4.1.4 | L'environnement méthodologique de l'apprentissage | 72 |
| 4.1.5 | les types d'apprentissage | 74 |
| 4.1.6 | Conclusion | 78 |
| 4.2 | Classification | 78 |
| 4.2.1 | Introduction | 78 |
| 4.2.2 | Définition | 78 |
| 4.2.3 | Classification supervisée | 79 |
| 4.2.4 | La classification multi-labels | 81 |
| 4.2.5 | La classification binaire | 81 |
| 4.2.6 | Conclusion | 82 |

4.1 Apprentissage

4.1.1 Introduction

Depuis la naissance de la science et de la technologie l'être humain ne s'est jamais contenté des travaux et progrès réalisés jusque-là, ses grandes ambitions l'ont toujours poussé à aller vers la nouveauté et l'exploration de nouveaux horizons scientifiques et technologiques. En s'inspirant de la nature les entourant et surtout des comportements et fonctionnalités accomplis par l'être humain les chercheurs ont en effet, réussi à reproduire divers comportements de ce dernier avec la création de machines et robots pouvant effectuer des tâches physiques habituellement effectuées par l'homme, mais la science ne s'arrête pas là, depuis bientôt un demi-siècle, plusieurs chercheurs se sont mis à étudier l'éventualité de pouvoir reproduire un comportement typique et propre à l'être humain qui est l'intelligence, un domaine appelé "intelligence artificielle".

Les chercheurs dans le domaine de l'intelligence artificielle visent à programmer des machines capables d'effectuer des tâches qui requièrent de l'intelligence, en d'autres termes, rendre des machines intelligentes.

Les travaux d'intelligence artificielle ont été adaptés et testés dans plusieurs domaines, on peut citer "l'aide à la décision" utilisée dans le domaine médical notamment dans l'aide aux diagnostics, "la prédiction" et on peut citer la prédiction de la consommation électrique, "le contrôle de processus" utilisé dans la conduite des procédés industriels, "la conduite de robot" robots pouvant s'adapter aux changements de conditions de leurs environnement, on cite également le domaine de la "reconnaissance des formes" utilisée dans plusieurs applications comme la reconnaissance de geste pour l'aide aux aveugles ou malvoyant par exemple ainsi que pour des faims de sécurité comme la reconnaissance faciale ainsi que la reconnaissance automatique des plaques d'immatriculation. En outre, programmer des machines capables de s'adapter à toutes les situations et contraintes de l'environnement extérieur et éventuellement évoluer en fonction de nouveaux paramètres est difficile.

De ce fait, il est donc nécessaire de contourner cette difficulté en dotant la machine de capacités d'apprentissage lui permettant de tirer profit de son expérience. C'est pourquoi

parallèlement aux recherches et aux travaux sur le raisonnement automatique, se sont développées des recherches sur l'apprentissage des machines.

4.1.2 Apprentissage naturel

L'apprentissage est un ensemble de mécanismes menant à l'acquisition de savoirs ou de connaissances, autrement dit, l'apprentissage consiste à acquérir ou à modifier une représentation d'un environnement de façon à permettre avec celui-ci des interactions efficaces ou de plus en plus efficaces prodiguant ainsi l'accroissement de son répertoire de connaissances[31].

Les modalités d'apprentissage naturel sont multiples : apprentissage par cœur comme le cas des connaissances innées acquises dès la naissance de l'enfant (l'odeur de sa mère, sa voix, la coordination de ses perceptions comme sa vue et son toucher avec ces mouvements, il apprend ensuite à segmenter des catégories de son et les associer à des significations .), apprentissage par instructions qui est acquis par le biais de l'éducation parentale ou de l'enseignement (apprendre à différencier entre le bien et le mal par exemple à l'aide des instructions parentales, apprendre à lire et à écrire en suivant des instructions précises), apprentissage par généralisation, cas de l'apprentissage au calcul ou la reconnaissance de formes (l'apprentissage pour l'opération d'addition par exemple se fait sur deux ou trois chiffres, le concept est ensuite généralisé à tous les chiffres, quant à l'exemple de l'apprentissage aux formes géométriques, il suffit de montrer quelques objets d'une certaine forme pour ensuite effectuer la généralisation sur tous les objets de la même forme) apprentissage impliquant la catégorisation voire la formation de théories faisant appel à un certain degré d'imagination, apprentissage autonome par découverte(fournir des points de vue personnels à propos de différents phénomènes sociaux, scientifiques).

Après l'étude des diverses formes d'apprentissage, il est à se demander si pour chacune d'elle, s'il existerait une contre partie lors de l'apprentissage artificiel ou pas des machines, dans ce qui suit une définition de l'apprentissage artificiel et des différentes formes utilisées.

4.1.3 Apprentissage artificiel (automatique)

L'apprentissage automatique est un champ d'études qui donne à des machines la capacité d'apprendre sans être explicitement programmées. L'apprentissage machine explore l'étude et la construction d'algorithmes qui peuvent apprendre et faire des prédictions à partir de données existantes, ces algorithmes ne fonctionnent pas comme les algorithmes classiques qui suivent des instructions strictement statiques, mais en construisant un modèle de la réalité à partir des entrées d'exemple afin de produire des prédictions ou des décisions basées sur les données d'entrée fournies comme sorties ou résultat de l'algorithme.

Autrement dit, l'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes et d'algorithmes qui permettent à une machine d'évoluer grâce à un processus d'apprentissage (apprendre à une machine des modèles de la réalité de telle sorte à ce qu'elle puisse suivre l'évolution de ces derniers), l'objectif est en effet, d'extraire et d'exploiter l'information présente dans un jeu de données[21].

L'apprentissage de la machine est utilisé dans une gamme de tâches informatiques où la conception et la programmation explicite se montrent insuffisantes, en effet, dans certaines applications comme la détection et la reconnaissance de formes très variées et changeantes au fil du temps (visages, lettres) la programmation explicite suivant des instructions fixes s'avère très limitée, c'est pourquoi l'apprentissage artificiel vise à collecter et construire des modèles, des règles ainsi que des connaissances pouvant œuvrer pour la prise de décision automatique.

4.1.4 L'environnement méthodologique de l'apprentissage

Comme décrit dans la section précédente, l'apprentissage opère avec deux paramètres principaux, à savoir la description des objets d'entrée avec un dictionnaire de données ainsi qu'un espace d'hypothèses. L'objectif étant de choisir un principe inductif permettant de sélectionner l'hypothèse la plus optimale, en effet un algorithme d'apprentissage est dit efficace si avec l'espace de données, il tente d'identifier une hypothèse optimale ou du moins s'en rapprocher.

Dans ce qui suit, nous allons passer en revue la façon de représentation des données pour l'apprentissage ainsi que leurs utilisations.

L'espace des données d'apprentissage

L'apprentissage se base sur des données représentant des objets qu'il faut décrire de la meilleure façon possible afin d'entraîner l'algorithme à suivre l'évolution de ces derniers, en effet, les types de données représentatives diffèrent et chaque objet réclame une certaine catégorie spécifique.

La représentation des objets d'apprentissage Afin de représenter des objets d'apprentissage, des connaissances (forme, taille, couleurs) concernant ces derniers sont regroupées dans un espace de représentation de données, ces dernières sont symbolisées de telle sorte à pouvoir décrire l'objet en question. Pour une meilleure perception du concept de la représentation d'objets à partir de données, on donne l'exemple d'un objet qui est une plaque d'immatriculation, en effet cette dernière pourrait être décrite selon des critères spécifiques la concernant, une plaque peut donc être décrite suivant deux traits principaux sa taille (hauteur et largeur sachant que ces derniers sont standardisés et représentés avec des valeurs numériques) et son niveau de gris (représentant également une valeur numérique allant de 0 à 255). Par conséquent chaque plaque d'immatriculation peut être représentée par trois valeurs numériques (hauteur, largeur, niveau de gris), ces représentations ne sont pas toujours complètes et efficace on peut donc ajouter d'autres critères devant être spécifiques à l'objet devant être représenté.

Il y'a un autre cas de représentation courant où les données sont représentées par des vecteurs binaires. Ceci correspond au cas où l'on décrit les objets à l'aide d'une série de tests où chaque objet est soit "vrais" soit "faux" selon chaque test effectué. La complexité de l'objet à représenter est proportionnelle au nombre de tests effectués, en effet, plus l'objet en question est complexe plus le nombre de descripteurs binaires est grand, on peut parfois avoir recours à des centaines voire des milliers de descripteurs binaires.

Les exemples cités plus haut nous donnent un bref aperçu de la manière dont des données manipulables (données numériques, booléennes) sont associées à des objets, dans ce qui suit nous allons voir comment les hypothèses d'apprentissage sont représentées.

L'espace des hypothèses d'apprentissage

Selon la complexité des objets à représenter pour l'apprentissage, l'espace de données à manipuler s'agrandit proportionnellement. De ce fait est né le problème de gestion de toutes ces données, en effet, manipuler ou traiter des nombres énormes de données requière beaucoup de ressources ainsi qu'un temps d'exécution, il est donc nécessaire de créer un espace d'hypothèses qui doit être recherché par le système, cela permet de guider à la fois la détermination de l'algorithme à utiliser, mais aussi les données qui seront nécessaires pour que l'apprentissage soit possible. En effet, la description d'un espace d'hypothèses permet d'obtenir une description plus précise et plus compacte, et permet également d'établir naturellement des liens avec les autres connaissances disponibles.

4.1.5 les types d'apprentissage

Les algorithmes d'apprentissage peuvent se catégoriser selon le mode d'apprentissage qu'ils emploient :

Apprentissage supervisé

Dans l'apprentissage supervisé, l'intervention d'un expert est nécessaire, dans la mesure où ce dernier est employé pour étiqueter correctement les exemples passés en entrées, le système doit alors trouver ou approximer la fonction qui permet d'affecter la bonne étiquette à ces exemples. Il s'agit donc d'apprendre des relations entre les entrées et les sorties, l'expert donne des exemples exprimant ces relations. Pour illustrer cela, on cite quelques exemples :

- Pour la détection de visage, les données = des images + des labels
- Pour la reconnaissance vocale, les données = son + texte
- Pour la reconnaissance de lettres et de chiffres, les données = des images + texte[16]

D'après ces exemples, c'est donc l'association de données relatives à chaque catégorie d'objet permettant au système d'associer ces données à tout objet similaire à l'exemple traité.

On parle alors d'apprentissage supervisé si les classes d'exemple sont prédéterminées, et un expert étiquette préalablement les exemples. Ce processus se déroule en deux étapes. Lors de la première phase (hors ligne, dite d'apprentissage), il s'agit de déterminer un modèle des données étiquetées. La seconde phase (en ligne, dite de test) consiste à prédire l'étiquette d'une nouvelle donnée, connaissant le modèle préalablement appris[10].

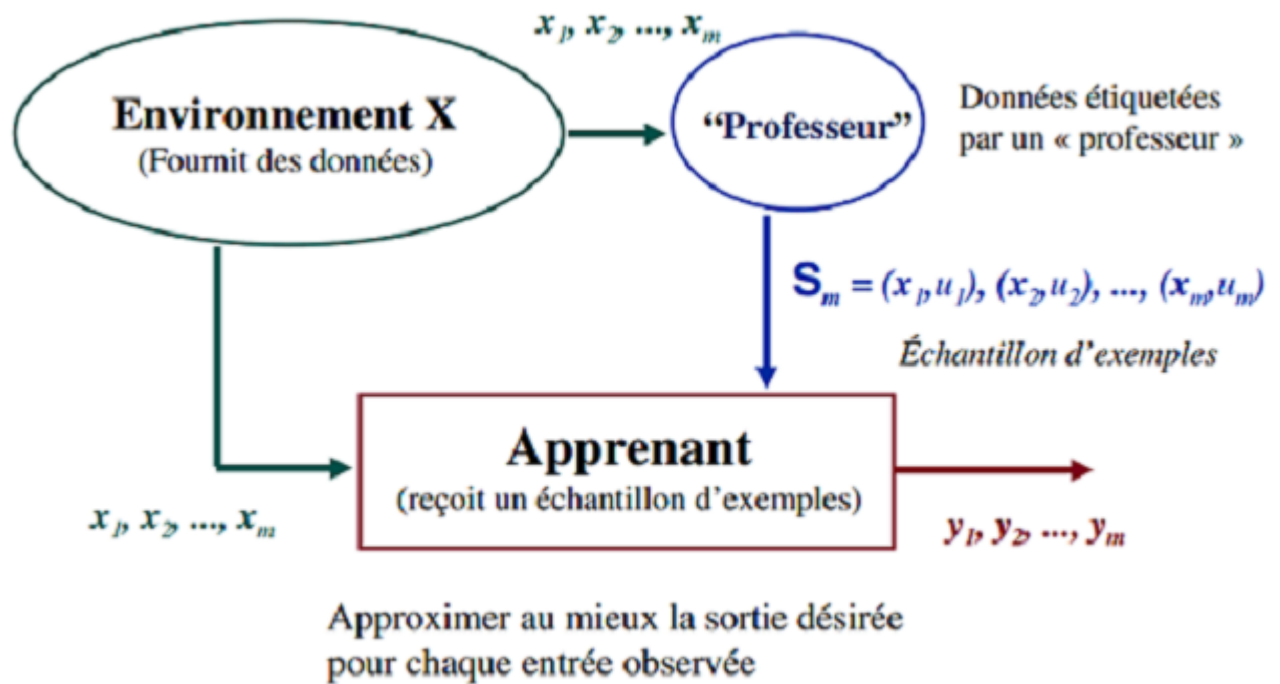


FIGURE 4.1 – Schéma récapitulatif de l'apprentissage supervisé

Apprentissage non-supervisé

A l'encontre de l'apprentissage supervisé celui-ci ne nécessite aucune intervention d'un expert, le système doit découvrir par lui-même la structure des données se basant sur des informations acquises. Il s'agit donc d'apprendre une structure dans un ensemble de données, on peut citer comme exemple la comparaison entre deux objets, en effet, à partir des données du premier exemple le système établit un lien de comparaison, et fournit l'hypothèse ou la décision automatique si oui ou non la nouvelle donnée correspond à l'exemple d'entrée.



FIGURE 4.2 – Exemple de comparaison de données : à gauche l'exemple d'entrées, à droite la nouvelle donnée (dans ce cas de figure le système doit retourner vrai)

Quand le système ou l'opérateur ne disposent que d'exemples, et pas d'étiquettes, et que le nombre de classes et leur nature n'ont pas été prédéterminés, on parle alors d'apprentissage non-supervisé. Le système doit ici dans l'espace de description (la somme des données) cibler les données selon leurs attributs disponibles, pour les classer en groupe homogène d'exemples. Autrement dit, à partir d'un échantillon d'apprentissage non étiqueté, le système cherche des régularités sous-jacentes qu'il représente sous forme d'une fonction ou d'un modèle complexe qu'il utilise par la suite pour résumer et comprendre le modèle en question[16].

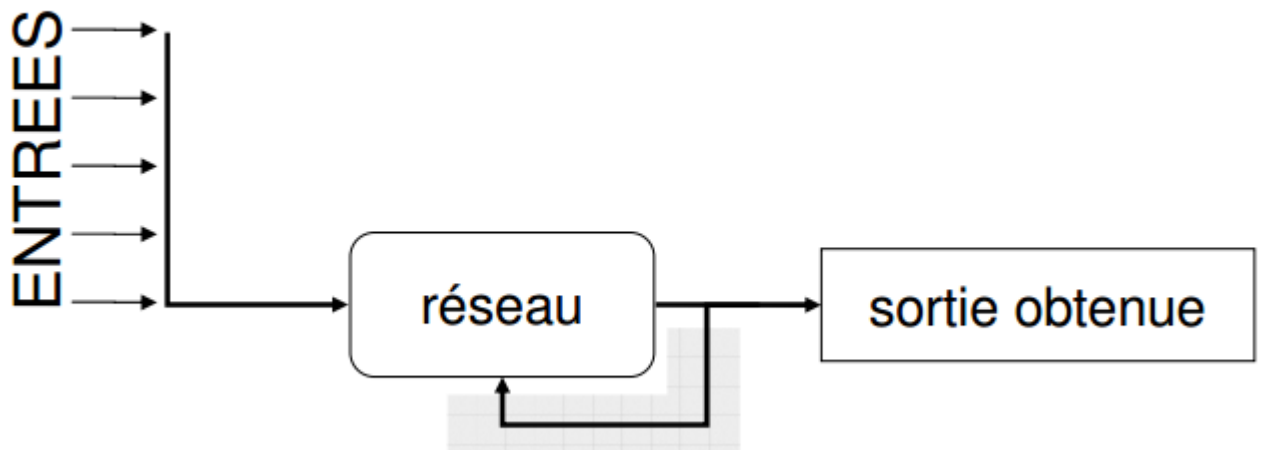


FIGURE 4.3 – Schéma récapitulatif de l'apprentissage non-supervisé

Apprentissage par renforcement

Dans ce type d'apprentissage le processus est dit en ligne ou interactif, l'algorithme apprend un comportement étant donné une observation, en d'autres termes, il apprend à se comporter et prend des décisions séquentielles en se basant sur un modèle interactif (changement) observé. L'algorithme interagit avec l'environnement extérieur, en effet, l'action de l'algorithme sur ce dernier produit une valeur de retour qui guide le système d'apprentissage pour la suite de ces actions[30].

On peut citer quelques exemples :

- Le contrôle d'accès : selon les documents (données) présentés à l'entrée de l'algorithme, celui-ci décide si oui ou non l'accès est valide.
- Jeux vidéo : selon l'action effectuée par le joueur (données d'entrée) l'algorithme décide quelle action effectuer.
- Robotique : les robots interagissent avec l'environnement selon la tâche qu'ils doivent exécuter.[4]

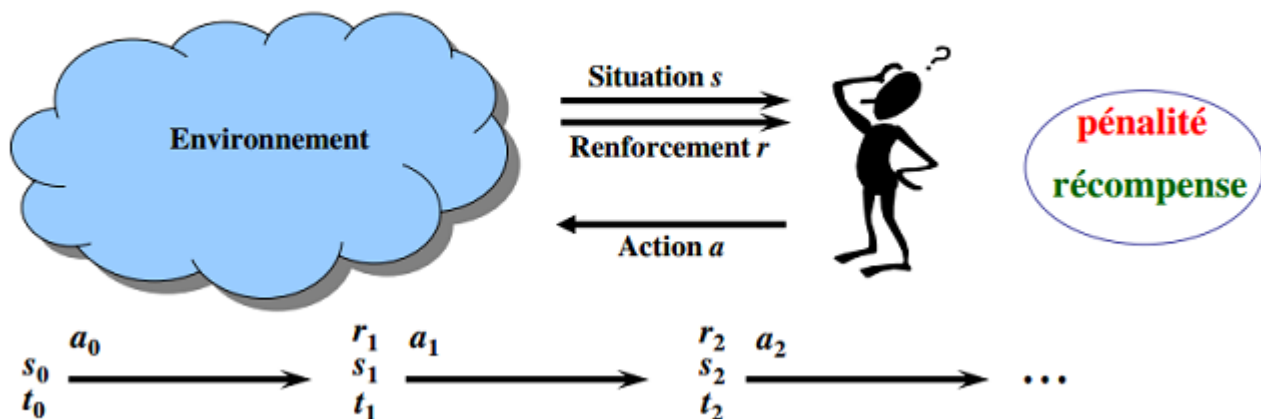


FIGURE 4.4 – Schéma récapitulatif de l'apprentissage par renforcement

L'agent apprend à se rapprocher d'une stratégie comportementale optimale par des interactions répétitives avec l'environnement, les décisions sont prises séquentiellement à

des intervalles de temps discrets.

4.1.6 Conclusion

Dans cette partie nous avons présenté le concept général de l'apprentissage automatique, pour les faims dans notre application la machine doit être capable d'apprendre les modèles et les caractéristiques des plaques d'immatriculation ainsi que ceux des caractères (chiffre et lettres) afin de pouvoir les localiser et les reconnaître en toute circonstance. Mais afin d'obtenir des résultats robustes en plus de l'apprentissage, on a recours à la classification des données. Dans ce qui suit, nous allons présenter les différentes méthodes de classification et choisir la méthode la plus adaptée a notre application.

4.2 Classification

4.2.1 Introduction

Lors de l'apprentissage de la machine, des données et des caractéristiques relatives à chaque objet sont présentés à cette dernière afin qu'elle puisse tirer des modèles et des descripteurs qu'elle utilisera par la suite pour la prédiction ou la décision automatique. Dans la majorité de ces cas, le nombre de données est tellement grand qu'il est nécessaire d'adopter des techniques et des méthodes afin de créer des groupes ou ce qu'on appelle des classes regroupant des données et des caractéristiques selon certains critères, ce processus facilite considérablement le traitement des masses de données volumineuses et donne un aspect plus clair et plus lisible aux données à traiter.

4.2.2 Définition

La classification est une méthode mathématique d'analyse de données visant à faciliter l'étude d'une base de données d'un volume très important, c'est en effet, la catégorisation d'objets selon des critères déterminés. Il s'agit d'opérer des regroupements en classes homogènes d'un ensemble de données de telle sorte à ce que les objets d'une même classe soient les plus semblables possible et que les classes soient les plus distinctes possible, pour ce faire, il existe diverses façons de procéder et qui peuvent conduire à des résultats différents, quoi que ces dernières procèdent toute suivant un même principe, elles visent

à attribuer une classe ou une catégorie à chaque objet, en se basant sur des données statistiques. La classification fait couramment appel à l'apprentissage automatique et est largement utilisée en reconnaissance de formes.

4.2.3 Classification supervisée

La classification supervisée vise à produire à partir d'un ensemble ordonné de données un modèle ou une procédure de classification, cette dernière opère en sachant les classes existantes et en disposant de données déjà classées (dispose d'éléments déjà classés). L'objectif de la classification supervisée est principalement de définir des règles permettant de classer des objets dans des classes à partir de variables qualitatives ou quantitatives caractérisant ces objets.

Formellement : Soit $D = d_1, d_2, \dots, d_i, \dots, d_n$ un ensemble de documents représentés chacun par une description noté ds_1, ds_2, \dots, ds_n et $C = C_1, C_2, \dots, C_k, \dots, C_c$ un ensemble de classes, la classification supervisée opère par deux fonctions. La première fait correspondre à tout individu d_i une classe C_k elle est définie au moyen de couples (d_i, C_k) donnés comme exemples au système. La deuxième fait correspondre à tout individu d_i sa description ds_i . La classification supervisée consiste alors à déterminer une procédure de classification :

$$C(f) : ds_i \rightarrow C_k$$

À partir de la description de l'élément, on détermine sa classe avec le plus faible taux d'erreurs. La performance de la classification dépend notamment de l'efficacité de la description. De plus, si l'on veut obtenir un système d'apprentissage, la procédure de classification doit permettre de classer efficacement tout nouvel exemple (pouvoir prédictif) [18]. On peut également caractériser les étapes d'une classification supervisée comme suit :

- Localisation de parcelles ou de caractéristiques représentatives de chaque catégorie d'objets se trouvant dans l'espace de données à traiter (appelées parcelles d'entraînement).
- Vectorisation des polygones autour de chaque parcelle avec assignation d'un identifiant unique pour chaque classe.
- Analyse des propriétés de chaque objet inclus dans les parcelles d'entraînement et création d'un fichier de signatures pour chaque catégorie d'objets.

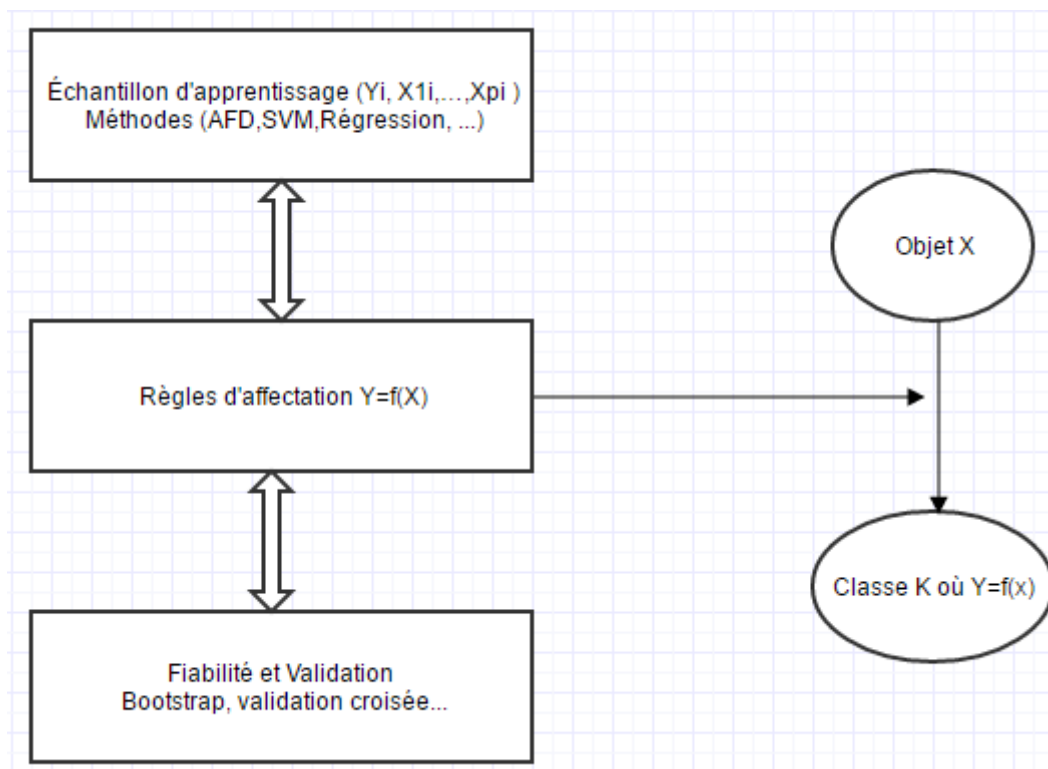
- Classification de l'ensemble des objets par comparaison de la signature de chacun par rapport à celle du fichier des signatures témoins.

La classification supervisée regroupe plusieurs méthodes, parmi celles-ci :

La classification multi-classes

La classification mutli-labels

La classification binaire (qui est l'approche utilisée pour notre méthode de classification)



Classification multi-classes

La classification multi-classes est utilisée quand on a recours à plus de deux catégories d'objets (plus de deux classes), cette dernière opère de telle sorte que chaque point de formation appartienne à l'une des N classes différentes [5]. L'objectif est de construire une fonction qui, étant donné un nouveau point de données, permettra de prédire correctement la classe à laquelle le nouveau point appartient. Par exemple, classer un ensemble d'images de fruits qui peuvent être des oranges, des pommes,

ou des poires. La classification multi-classes fait l'hypothèse que chaque échantillon (objet) est affecté à une et une seule étiquette : un fruit peut être une pomme ou une poire, mais pas les deux en même temps [22].

4.2.4 La classification multi-labels

Au même titre que les autres types de classification l'objectif est toujours de construire une fonction permettant une classification des données présentées, la différence avec les autres approches réside donc dans l'étiquetage des données, en effet dans cette dernière, nous manipulons des données ayant plus d'une seule étiquette (chaque objet possède plusieurs références le décrivant). Il s'agit donc de déterminer l'appartenance ou non d'un objet à une classe donnée. Effectivement, dans cette méthode on assigne à chaque échantillon un ensemble d'étiquettes cibles. Cela peut être considéré comme la prédiction des propriétés d'un point décrivant des données qui ne sont pas mutuellement exclusives, telles que les propriétés reliées au point de donnée, document par exemple. Ce dernier peut-être un texte religieux, un texte politique, ou encore un document financier et éducatif à la fois comme il peut être d'aucun de ces cas.

4.2.5 La classification binaire

La classification binaire est la tâche de classifier un ensemble d'éléments (données) en deux groupes sur la base d'une règle de classification, cette dernière est mise en application dans plusieurs domaines comme le domaine médical (prédiction d'après des symptômes si oui ou non un patient porte une maladie) ou encore dans le domaine industriel (contrôle de qualité d'un produit) Dans le cadre de notre application le type de classification le plus adaptés à nos faims est la classification binaire, en effet cette classification est utilisée dans le cas de présence de deux classes uniquement, comme c'est le cas dans notre travail, le but est de classifier des images comme étant oui ou non des plaques d'immatriculation ou définir si oui ou non une image représentant un caractère (lettre ou chiffre) correspond à un certain modèle, ce qui permet la reconnaissance du chiffre ou de la lettre en question.

4.2.6 Conclusion

Ce chapitre nous a permis de prendre connaissance de certains concepts des deux vastes domaines de l'apprentissage et de la classification qui se chevauchent généralement dans plusieurs domaines notamment dans le domaine de la détection et la reconnaissance des formes, en effet, comme nous l'avons vu dans ce présent chapitre, l'apprentissage permet d'apprendre de nouvelles catégories d'objets à une machine quant à la classification, elle permet de catégoriser et de séparer ces catégories d'objets pour une meilleure compréhension.

Dans ce travail de mémoire, nous avons recours à ces deux notions afin d'apprendre à notre machine les caractéristiques d'une plaque d'immatriculation afin qu'elle puisse les localiser en toutes circonstances en utilisant la méthode de classification par les caractéristiques de Haar qu'on présentera dans le prochain chapitre intitulé "implémentation et résultats".

Chapitre 5

Implémentation et Résultats

Contents

| | |
|--|-----------|
| 5.1 Outils et environnement de développement | 85 |
| 5.2 OpenCV : Open source Computer Vision | 85 |
| 5.3 Python | 86 |
| 5.3.1 La bibliothèque standard math | 87 |
| 5.3.2 NumPy | 88 |
| 5.3.3 Latex | 88 |
| 5.4 Système d'exploitation et outils utilisés | 89 |
| 5.5 Outils | 89 |
| 5.5.1 Sublime text | 89 |
| 5.5.2 PyCharm | 90 |
| 5.5.3 TeXnicCenter | 90 |
| 5.6 Approche de détection implémentée | 91 |
| 5.6.1 Caractéristique des plaques algériennes | 92 |
| 5.7 Présentation de l'approche suivie | 93 |
| 5.7.1 Prétraitement | 93 |
| 5.7.2 Deuxième phase : Application des contraintes dimensionnelles | 95 |
| 5.7.3 Troisième phase : Application de la contrainte de la couleur | 96 |
| 5.8 Classification par caractéristiques de Haar | 98 |
| 5.8.1 Apprentissage du classificateur | 98 |

| | | |
|-------------|---|------------|
| 5.8.2 | Extraction des caractéristiques d'une image | 99 |
| 5.8.3 | Élaboration du classificateur | 99 |
| 5.9 | Reconnaissance optique des caractères de la plaque . . | 101 |
| 5.10 | Tesseract | 101 |
| 5.10.1 | Motivations et historique | 101 |
| 5.10.2 | Architecture | 102 |
| 5.10.3 | Retrouver les lignes et les mots | 103 |
| 5.10.4 | Reconnaissance de mot | 105 |
| 5.10.5 | Classification | 106 |

5.1 Outils et environnement de développement

Afin de réaliser notre travail, nous avons eu recours à plusieurs outils de développement notamment la bibliothèque de vision par ordinateur Open CV 3.1.0 et le langage de programmation python version 3.4, dans ce présent chapitre nous allons présenter les différentes bibliothèques et modules auxquels nous avons fait appel afin de concevoir un système de lecture automatique de plaques d'immatriculation fonctionnel et qui nous a permis de gagner un temps et considérable et ainsi nous focaliser sur la partie la plus importante qui est la détection et la lecture des plaques d'immatriculation.

5.2 OpenCV : Open source Computer Vision



OpenCV (pour Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société de robotique Willow Garage assure le support de cette bibliothèque depuis 2008.[12]

Cette dernière propose 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.

Afin de mieux présenter son étendue et ce qu'elle permet de faire, nous allons jeter un œil à certains modules accessibles au travers de son API C.

- **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.
- **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.

5.3 Python



Python est un langage objet interprété de haut niveau¹, créé au début des années quatre-vingt-dix par Guido Van Rossum au Centrum voor Wiskunde à Informatica, Amsterdam. En 1995, Rossum poursuit le développement de Python à la Corporation for National Research Initiatives de Reston (Virginie). Et en 2000, Rossum créa l'équipe BeOpen PythonLabs qui, en octobre de la même année est incorporée à Zope Corporation puis à la société Digital Creations. En 2001, la PSF (Python

1. Un langage de haut niveau est un langage de programmation orienté autour du problème à résoudre, qui permet d'écrire des programmes en utilisant des mots usuels des langues naturelles (très souvent de l'anglais) et des symboles mathématiques familiers.

Software Foundation) est créée.

Il s'agit d'une organisation à but non lucratif détenant les droits de propriété intellectuelle de Python. Elle est sponsorisée en particulier par Zope Corporation. Python est distribué sous forme de logiciel libre.[1]

Python est un langage de programmation interprété, ce qui veut dire que les instructions qu'on lui envoie sont « transcrites » en langage machine au fur et à mesure de leur lecture. D'autres langages (comme le C / C++) sont appelés « langages compilés » car, avant de pouvoir les exécuter, un logiciel spécialisé se charge de transformer le code du programme en langage machine. On appelle cette étape la « compilation ». À chaque modification du code, il faut rappeler une étape de compilation. Les avantages d'un langage interprété sont la simplicité (on ne passe pas par une étape de compilation avant d'exécuter son programme) et la portabilité.

Cependant, un langage interprété se révélera un peu plus lent qu'un langage compilé (la traduction à la volée du programme ralentit l'exécution), bien que cette différence tende à se faire de moins en moins sentir au fil des améliorations. De plus, il faudra installer Python sur le système d'exploitation utilisé pour que l'ordinateur puisse comprendre le code.

En contrepartie, un langage compilé se révélera bien plus rapide qu'un langage interprété (la traduction à la volée de votre programme ralentit l'exécution), bien que cette différence tende à se faire de moins en moins sentir au fil des améliorations. De plus, il faudra installer Python sur le système d'exploitation que vous utilisez pour que l'ordinateur puisse comprendre votre code. Dans notre travail, nous avons vu et utilisé plusieurs bibliothèques et modules python, nous allons les passer en revue dans ce qui suit.

5.3.1 La bibliothèque standard math

Pour disposer des fonctions mathématiques usuelles, la librairie d'origine du python se nomme math. On peut alors d'importer juste les fonctions nécessaires par **from math import cos, log** ou toutes les fonctions mathématiques par **from math import ***. Dans le premier cas l'inconvénient est qu'il faut savoir à l'avance les fonctions utilisées par la suite, dans le deuxième cas on risque de surcharger inutilement

la mémoire.

5.3.2 NumPy



NumPy est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.[11] Elle fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices

5.3.3 Latex



LaTeX est un langage et un système de composition de documents créé par Leslie Lamport en 1983. Plus exactement, il s'agit d'une collection de macro-commandes destinées à faciliter l'utilisation du « processeur de texte » TeX de Donald Knuth. Depuis 1993, il est maintenu par le LATEX3 Project team. La première version utilisée largement, appelée LaTeX2.09, est sortie en 1984. Une révision majeure, appelée LaTeX2, est sortie en 1991.

Le nom est l'abréviation de Lamport TeX. On écrit souvent LATEX, le logiciel permettant les mises en forme correspondant au logo.

Du fait de sa relative simplicité, il est devenu la méthode privilégiée d'écriture de documents scientifiques employant TeX. Il est particulièrement utilisé dans les

domaines techniques et scientifiques pour la production de documents de taille moyenne ou importante (thèse ou livre, par exemple). Néanmoins, il peut être aussi employé pour générer des documents de types variés (par exemple, des lettres, ou des transparents).

Le moteur actuel (2014) de LATEX est PdfTeX mais ses limites, notamment concernant le traitement des textes Unicode, ont entraîné l'adoption officielle de LuaTeX comme futur successeur. Lorsque le développement en sera stabilisé, LuaLaTeX sera probablement l'implémentation standard de LaTeX[6].

5.4 Système d'exploitation et outils utilisés

Afin de rendre notre application fonctionnelle sur plusieurs plateformes, nous avons dû la tester sur deux systèmes d'exploitation différents :

- Elementary Os Freya pour le développement et les tests.



5.5 Outils

5.5.1 Sublime text



Sublime Text est un éditeur de texte générique, mais clairement orienté programmation, développé par Jon Skinner, en C++ et Python. Le logiciel a été conçu tout d'abord comme une extension pour Vim (ligne de commande), il est ensuite devenu un logiciel à part entière, il dispose de toutes les fonctionnalités classiques d'un éditeur de texte, comme la coloration syntaxique, et l'auto-complétion de code, ainsi que plusieurs autres fonctionnalités qui font sa force comme la gestion des fichiers et des projets, le minimap, les plungins, etc.

5.5.2 PyCharm



PyCharm est un environnement de développement intégré (abrégé EDI en français ou en anglais : IDE (Integrated Development Environment)) utilisé pour programmer en Python.

5.5.3 TeXnicCenter



TeXnicCenter est un logiciel libre sous la licence open source GNU GPL, qui se présente sous la forme d'un environnement de développement intégré pour le langage

LaTeX sous Windows. Il est basé sur les distributions MiKTeX ou TeXLive (en). Il permet à l'utilisateur d'éditer un document LaTeX et de le compiler dans les formats PDF, DVI ou PS. Un menu permet un accès simple et rapide aux éléments usuels (formules, symboles, sections). Il permet également la création de projets pour organiser et accéder aux sections et environnements des documents, et pour insérer une bibliographie (avec BibTeX) ainsi qu'un index (avec MakeIndex). Cependant, TeXnicCenter n'est pas un éditeur WYSIWYG²[18].

5.6 Approche de détection implémentée

Ce présent et dernier chapitre est consacré à la présentation de la méthode choisie et implémentée afin d'aboutir à notre but ultime qui est la réalisation d'un système de reconnaissance automatique de plaques d'immatriculation algériennes. Comme introduit précédemment, afin de réaliser un tel système deux étapes primordiales sont nécessaires, en premier lieu nous procédons à la détection des plaques d'immatriculation ce qui a pour but la localisation de l'emplacement exacte de cette dernière, ensuite la seconde étape consiste à reconnaître les caractères composant la plaque, en d'autres termes transformer l'image qui est un ensemble de pixels en un fichier texte utilisable. Dans ce qui va suivre nous allons procéder à la présentation des étapes à suivre afin de réaliser la détection, ces étapes permettront de fournir des images contenant uniquement la plaque minéralogique, ces dernières seront ensuite passées à un système de reconnaissance de caractères (OCR) appelé "Tesseract" qu'on présentera en second lieu dans ce chapitre.

Avant d'entamer les détails de l'approche nous allons faire un tour d'horizons sur les caractéristiques des plaque d'immatriculation algériennes.

2. WYSIWYG : What You See Is What You Get

5.6.1 Caractéristique des plaques algériennes

Dimensions

Les plaques minéralogiques doivent avoir 52 centimètres de longueur et 11 centimètres de largeur.

Couleur

1. Plaque avant : Chiffre arabes noirs sur fond réfléchorisé blanc-gris.
2. Plaque arrière : Chiffre arabe noirs sur fond réfléchorisé jaune.

Composition du numéro

Le numéro d'immatriculation est composé (en partant de la droite vers la gauche) :

- a. 02 chiffres représentant la wilaya d'immatriculation.
- b. Groupe de 03 chiffres arabes, caractérisant l'année de mise en circulation(les deux premiers chiffre à partir de la droite) et la catégorie du véhicule(le troisième chiffre)

Les chiffre arabes représentant la catégorie des véhicules sont indiqué ci-après :

- (1) Véhicules de tourisme.
 - (2) Camions.
 - (3) Camionnettes.
 - (4) Autocar et autobus.
 - (5) Autres tracteurs.
 - (6) Véhicules spéciaux.
 - (7) Remorques et semi-remorques.
- c. Groupe de 05 chiffres arabes représentant le numéro d'ordre chronologique d'immatriculation du véhicule dans la catégorie, l'année de mise en circulation et la wilaya considérée.

Maintenant que nous connaissons les différentes caractéristiques des plaques algériennes nous pouvons aborder la démarche qu'on a suivie afin de réaliser un système de détection et de reconnaissance fonctionnel qui justement se base sur l'exploitation de ces propriétés.

5.7 Présentation de l'approche suivie

5.7.1 Prétraitement

Conversion en gris

La majorité des traitements effectués, dans l'algorithme proposé, travaillent sur des images en niveaux de gris. Il faut donc appliquer une conversion depuis les images sources RGB.

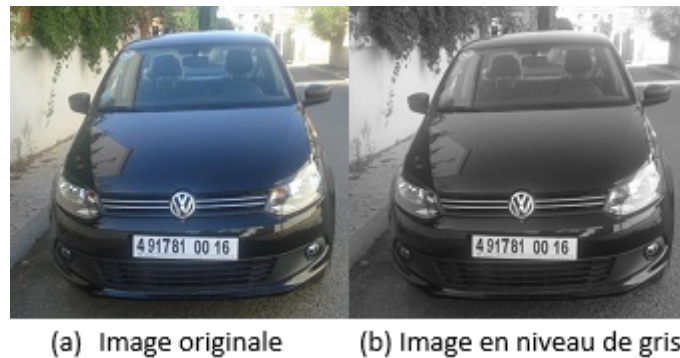


FIGURE 5.1 – Image en niveau de gris

Filtrage

Application un filtre gaussien 5×5 pour éliminer les bruits verticaux.



FIGURE 5.2 – Image filtrée

Opérateur de Sobel

Trouver le filtre de Sobel avec un noyau 3*3



FIGURE 5.3 – Application de l'opérateur de Sobel

Seuillage

Après le filtre de Sobel, on applique un seuillage afin d'obtenir une image binaire avec un seuil obtenu automatiquement grâce à la méthode d'Otsu.



FIGURE 5.4 – Obtention d’une image binaire

Fermeture

La fermeture sert à supprimer les espaces vides entre chaque ligne de contour vertical, et connecter toutes les régions qui ont un nombre élevé de contours.

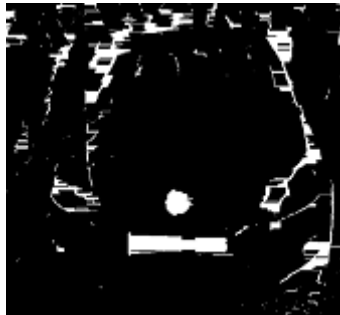


FIGURE 5.5 – Fermeture

5.7.2 Deuxième phase : Application des contraintes dimensionnelles

Dans la première phase de l’application, nous avons passé en revue les prétraitements nécessaires à appliquer à l’image afin de faciliter la détection et la localisation de la plaque. Dans cette présente phase nous allons montrer les différentes opérations qui vont nous permettre de relever la plaque minéralogique.

1. Trouver et récupérer tous les contours de l’image.

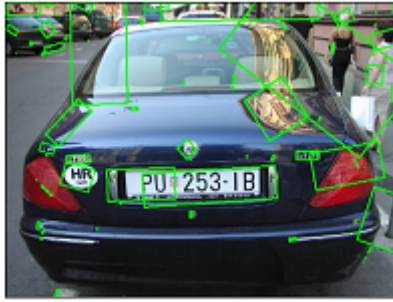


FIGURE 5.6 – Dessin de tous les contours de l'image

2. Récupérer tous les rectangles (bounding contours) de l'image qui correspondent au critères suivants :
 - (a) La surface est supérieure ou égale à 100 px.
 - (b) Le ratio qui est égale à la longueur sur la hauteur doit être compris entre 3 et 8.

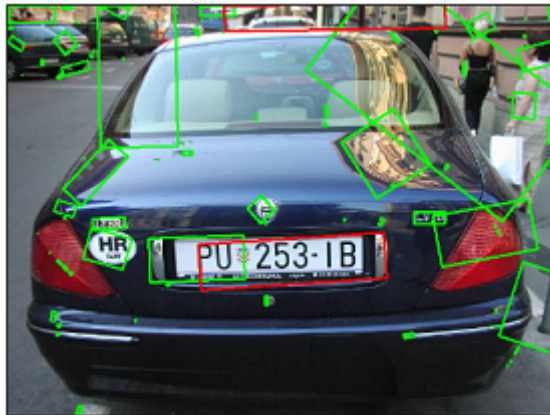


FIGURE 5.7 – Les rectangles en rouges satisfont les contraintes

5.7.3 Troisième phase : Application de la contrainte de la couleur

Comme nous l'avons vu dans la section concernant les caractéristiques des plaques algériennes, elles optent pour deux couleurs (blanc-gris et jaune), nous allons donc

essayer de tirer profit de cette spécificité. Nous générerons des masques avec l'algorithme "floodfill" qui est un algorithme de remplissage par diffusion qui prend 03 paramètres pour une image donnée : la position du pixel de départ (appelé aussi germe ou grain), la couleur ciblée et la couleur de remplacement. L'algorithme recense tous les pixels de l'image qui sont connectés au germe par un chemin de la couleur ciblée et substitue à celle-ci la couleur de remplacement. Dans notre cas, au départ, on n'a aucune idée comment choisir un germe spécifique parmi les rectangles validés par la 2^{ème} étape ce qui pousse à générer des nombres aléatoires en espérant qu'au moins un d'entre eux réussisse dans l'exploitation d'une couleur blanche (jaune) et qu'il tire une partie de cette couleur.

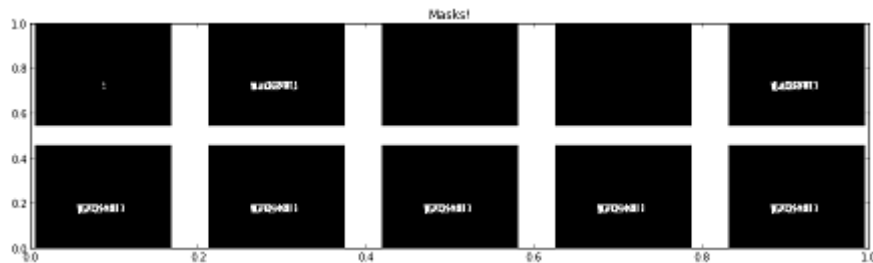


FIGURE 5.8 – Les différents masques générés

La plupart des masques probables sont maintenant ici. Mais il peut y avoir ces cas où presque les mêmes masques sont répétés. C'est possible puisque les graines étaient au hasard et il peut toujours y avoir plus de 1 graine produisant le même masque, et c'est pour cela que nous avons utilisé l'algorithme **Root Mean Square** qui mesure la similarité de deux images, si les deux images sont exactement identiques il retourne une valeur nulle ce qui nous a permis de faire des tests et d'éliminer les images identiques. Maintenant que le processus de segmentation est fini et nous avons des régions valides, nous pouvons découper la région qui nous intéresse, changer les dimensions de l'image et égaliser la lumière de régions découpées.



FIGURE 5.9 – Les régions découpées

5.8 Classification par caractéristiques de Haar

En 2001 Les chercheurs Paul Viola et Michael Jones ont mit au point une méthode de détection d'objets dans des images numériques. Pour ce faire ils ont mis au point un classificateur appelé « classificateur de haar » ce dernier requière une étape préliminaire très importante qui est l'apprentissage du classificateur. Il s'agit d'entraîner le classificateur afin de le sensibiliser à ce que l'on veut détecter (plaques d'immatriculation dans notre cas) afin de pouvoir effectuer ensuite une classification binaire dans une banque d'images et déterminer dans chacune d'elles si oui ou non il s'agit d'une plaque d'immatriculation[26] .

5.8.1 Apprentissage du classificateur

Dans cette partie de l'algorithme l'objectif est d'entraîner et d'apprendre à la machine la forme et les caractéristiques relatives à l'objet recherché, pour ce faire, deux étapes fondamentales sont nécessaires, la première est la présentation d'une banque d'images dites positives (image de plaque d'immatriculation) à l'algorithme, la seconde est l'inverse de la première où une quantité énorme d'images dites négatives (images ne contenant pas de plaque d'immatriculation) sont également présentées à l'algorithme. Ces deux étapes visent à extraire le maximum de caractéristiques

d'une plaque (taille, couleurs,) afin de pouvoir détecter cette dernière dans de nouvelles images.

5.8.2 Extraction des caractéristiques d'une image

Afin de relever les caractéristiques d'une image Viola et Jones proposent dans leur méthode d'apprentissage d'utiliser une représentation synthétique et informative des propriétés de l'image, calculée à partir des valeurs des pixels (utilisation des valeurs de chaque pixel de l'image afin de définir et de relever des caractéristiques communes à deux images contenant le même objet). Viola et Jones définissent des caractéristiques très simples, les caractéristiques pseudo-Haar définies dans le détail dans la partie « Etat de l'art sur la détection d'objet », elles sont donc calculées par la différence des sommes de pixels de deux ou plusieurs zones rectangulaires adjacentes. Afin de faciliter le calcul de ces dernières le principe de l'image intégrale est utilisé, ce qui permet de relever plus facilement des caractéristiques communes à un même objet, et de les utiliser par la suite de le processus de classification).

5.8.3 Élaboration du classificateur

Dans cette partie le principe est d'assimiler une ou plusieurs caractéristiques relatives à l'objet recherché (plaque d'immatriculation) à un classificateur, ce dernier considère ces caractéristiques comme règle de classification afin de pouvoir générer la décision automatique ou classifier l'objet en question. La sélection des caractéristiques se fait par boosting, il s'agit de construire un classificateur « fort » à partir d'une combinaison pondérée de classificateur « faibles ». Après avoir calculé toutes les caractéristiques provenant de chacune des formes de la fonction de Haar, il est à présent nécessaire de définir un seuil selon lequel on décide si l'image en traitement est une plaque d'immatriculation ou non. Afin de calculer ces seuils chaque paramètre ou caractéristique passe sur toutes les images intégrales calculées à base de la banque d'images prise en considération et relève donc un seuil de validité pour chaque paramètre, en d'autres termes, pour chaque paramètre, les résultats des calculs par image intégrale de tous les échantillons dit positifs sont comparés et un seuil ou une moyenne est relevée, ces derniers seront ensuite classés dans un fichier

qui contiendra les poids (les seuils à atteindre pour une réponse positive) de tous les classificateurs de haar. Afin de catégoriser chaque classificateur formé à base des calculs des fonctions de harr pour toutes les images positives comme un bon ou un mauvais classificateur, des paramètres qui sont la moyenne des valeurs obtenues, l'écart type, le minimum et le maximum doivent être calculés. Une fois ces derniers calculés, on définit si le classificateur est bon ou mauvais par une distribution gaussienne des paramètres calculés (on commence à partir de la valeur moyenne et on élargit selon l'écart-type, plus l'écart-type est grand, plus la reconnaissance est optimal). Le même traitement est effectué pour les images négatives. Le principe du boosting doit être respecté par les classificateurs, il s'agit d'ordonner les classificateurs du plus fort au plus faible afin de former une chaîne en cascades, chaque classificateur, ayant une distribution gaussienne, doit vérifier 3 conditions qui sont les suivantes :

1. Obtenir une reconnaissance de l'objet de près de 100% (presque toutes les images contenant l'objet à détecter sont détectées).
2. Obtenir une reconnaissance sans objets très faible (la plupart des images sans objets ne sont pas détectées).
3. Erreur totale pour le classificateur inférieure à 50%.

Selon ces trois critères les classificateurs sont ordonnés.

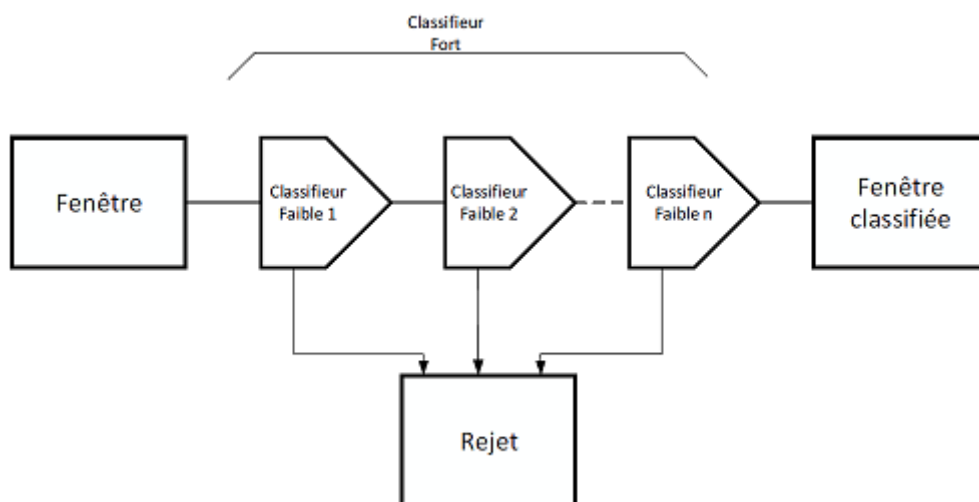


FIGURE 5.10 – Cascades de classifier

La formation du classificateur faible consiste donc à trouver la valeur du seuil de la caractéristique qui permet de mieux séparer les exemples positifs des exemples négatifs, en d'autres termes former des couples de la forme (caractéristique, seuil) afin de procéder à une classification binaire et classer les images dans deux groupes (image positive contenant l'objet recherché, et négative ne contenant pas l'objet en question). Un classificateur faible ne suffit pas pour une détection, c'est donc la combinaison des "n" classificateurs faibles qui forme un classificateur fort capable d'effectuer une détection quasi-parfaite sans erreurs.

5.9 Reconnaissance optique des caractères de la plaque

Comme introduit dans les sections précédentes, afin de réaliser un système de reconnaissance automatique de plaques d'immatriculation, il est crucial de recourir à deux étapes essentielles, la partie détection que nous avons illustrée plus haut ainsi que la partie reconnaissance qui vise à lire les caractères composant la plaque en transformant l'image fournie par la partie détection en un ensemble de caractères utilisables pour diverses faims pouvant un but de sécurité (contrôle du trafic routier) ou encore de la gestion (Classification des véhicules selon la wilaya d'appartenance, ou selon leur catégories...). Afin de réaliser cette opération, on fait appel à la technologie OCR (Reconnaissance optique de caractères) que nous avons précédemment introduit. Dans cette partie nous allons présenter l'OCR retenu pour notre application et qui s'intitule TESSERACT

5.10 Tesseract

5.10.1 Motivations et historique

Tesseract est un moteur OCR open source, développé par l'entreprise HP³ entre 1984 et 1994, ce dernier a brillé et surpris par ces résultats en 1995 lorsqu'il fût pu-

3. HP : Hewlett-Packard

blié dans une revue scientifique. Le projet Tesseract a commencé comme un travail de recherche pour une thèse de doctorat dans les laboratoires du géant informatique HP, après quelques années de recherche et de travail acharné, ce projet a gagné de l'élan comme étant une éventuelle machine ou un logiciel pour les lignes HP, ceci fût animé par la motivation du marché commercial des machines OCR⁴ existantes à cette époque, en effet, les OCR étaient dans leur phase de développement, et leurs résultats étaient voués à l'échec sauf dans le cas d'une qualité d'impression ou de photographie quasi-parfaite. Après un certain têt d'avancement du projet, les laboratoires Bistol d'HP se sont alliés aux divisions de scanner d'HP se trouvant au Colorado, ce qui a valu une avancée significative au logiciel mais qui n'arrivait toujours pas à être un produit fini à cette époque. La prochaine étape du développement de ce dernier fût le retour aux laboratoires HP Bistol, où le travail était concentré sur l'amélioration et la précision du niveau de base existant. Le développement de Tesseract s'est entièrement arrêté en fin de l'année 1994, et il fût enfin un logiciel à part entière qui a été testé et approuvé en 1995 lors des tests annuels de UNLV (University of Nevada, Las Vegas). A la fin de l'année 2005 HP a publié et commercialisé Tesseract pour l'open source, il est actuellement disponible et utilisable.

5.10.2 Architecture

Tesseract suppose que son paramètre d'entrée est une image contenant des régions de texte polygonales⁵. Après l'acquisition de son image d'entrée, pour son traitement Tesseract suit une méthode traditionnelle appelée step-by-step pipeline cependant certaines étapes de cette dernière étaient inhabituelles à cette époque et le reste probablement de nos jours.

La première étape du processus est une analyse des composantes connexes dans laquelle les grands traits des composants sont stockés. Néanmoins, Ce fut une décision de conception informatiquement chère à l'époque, mais qui avait un avantage significatif : par l'inspection de l'imbrication des contours, et le nombre d'enfants et de petits-enfants décrits, il est simple de détecter le texte inverse et le reconnaître aussi facilement que du texte noir sur blanc.

4. OCR : Optical Character Recognition

5. Figure géométrique plane, formée d'une suite cyclique de segments consécutifs

A ce stade, les contours sont réunis, purement par emboîtement, dans des Blobs.⁶ La reconnaissance procède ensuite comme un processus à deux passages.

Dans le premier passage, une tentative est faite pour reconnaître chaque mot à son tour ainsi chaque mot approuvé est transmis à un classificateur adaptatif en tant que données d'entraînement. Le classificateur adaptatif obtient alors une chance de reconnaître de façon plus précise le texte plus bas dans la page. Il arrive que le classificateur adaptatif reconnaisse quelque chose d'utile mais un peu trop tard pour apporter une contribution en haut de la page, un second passage est exécuté sur la page et ainsi les mots qui n'ont pas été reconnus assez bien sont reconnus à nouveau. Une phase finale résolu des espaces flous, et vérifie des hypothèses alternatives pour localiser le texte[33].

5.10.3 Retrouver les lignes et les mots

Retrouver une ligne

L'algorithme destiné à retrouver les lignes du document ou de l'image à traiter est conçu de telle sorte à pouvoir reconnaître même les pages faussées ou endommagées sans avoir à ajouter des traitements pouvant fausser cette dernière, ce qui permet également un gain considérable en matière de qualité de l'image ou de la page traitée. Les parties clés de l'algorithme sont la détection des blobs (contours) avec des filtres ainsi que la construction des lignes. En supposant que les prétraitements effectués ainsi que l'analyse de mise en page fournissent des zones de texte ayant des tailles quasi-uniformes. Un filtre dit Filtre à hauteur centile procède à la division de l'image en plusieurs petites régions, chaque pixel de cette dernière est remplacé par la valeur du point qu'il représente dans l'histogramme des pixels voisins qui correspond à un centile⁷. Ces traitements sont effectués par ce filtre afin de pouvoir supprimer tout

6. Le BLOB, pour Binary Large Object, est un type de donnée permettant le stockage de données binaires (le plus souvent des fichiers de type image, son ou video) dans le champ d'une table d'une base de données.

7. Un centile en statistique représente chacune des 99 valeurs qui divisent des données triés en 100 parts égales de sorte à ce que chaque partie représente 1/100 de l'échantillon total.

effet décoratif sur les lettres ainsi que leurs empattements⁸ il procède également à la suppression de tout dépassement vertical des lettres en considérant que la hauteur moyenne se rapproche de la taille du texte dans la région étudiée, il est donc sans danger de supprimer certains pixels qui sont inférieur à une certaine fraction de la hauteur médiane qui peuvent être la ponctuation, un bruit ou encore des tâches, sans toucher aux blobs (contours) principaux. Après ces traitements, les blobs obtenus sont plus susceptibles de former des modèles de lignes sans chevauchements, des lignes parallèles. Le tri et le traitement de ces derniers avec des coordonnées de chaque lettre ou chiffre donne la possibilité d'assigner chaque contours à une ligne de texte unique, ce traitement permet également le suivi des pentes (les lettres contenant des lignes inclinées allant du bas vers le haut), réduisant au maximum les erreurs éventuelles (affecter une ligne incorrecte a une lettre ou à un chiffre). Une fois les contours assignés aux lignes, des carrés correspondant à chaque modèle de lettre sont utilisés afin d'estimer et d'étudier les contours de bas de chaque caractère ainsi les contours considérés comme hors des lignes définies sont de nouveau assigné aux lignes approprié.

Lignes de base

Une fois les lignes du texte trouvées, les lignes de base sont mises en évidence d'une manière plus précise en utilisant des splines. Une spline est une fonction qui définit des morceaux de droite avec des polynômes.



FIGURE 5.11 – Exemple d'une spline

8. Les empattements sont les petites extensions qui forment la terminaison des caractères dans certaines polices d'écriture ou ce qui lie entre les caractères pour former un mot

Cette technologie est une grande première pour les systèmes OCR, c'est elle qui a permis à tesseract de manipuler des pages avec des lignes de base courbées. Les lignes de base sont montrées en partitionnant les blobs ou les contours en groupes avec un raisonnable déplacement continu suivant les lignes de base d'origine du texte traité.



FIGURE 5.12 – Exemple d'une ligne de base incurvée.

L'avantage des splines est que le calcul des fonctions est raisonnablement stable ce qui offre des résultats assez satisfaisants, ces dernières présentent néanmoins un inconvénient qui se présente lorsque le texte présente beaucoup de discontinuités ce qui requière l'utilisation de plusieurs segments de spline, des erreurs de reconnaissance surviennent donc, dans ce cas de figure.

Détection des hauteurs fixe

Tesseract évalue les lignes de textes pour déterminer si elles sont de hauteur fixe, ainsi les zones où la hauteur est fixe il découpe les mots en caractères.

5.10.4 Reconnaissance de mot

Une partie du processus de reconnaissance pour tout moteur de reconnaissance de caractères est d'identifier comment un mot doit être segmenté en caractères. La sortie de segmentation initiale de la phase de détection des lignes est classifiée en premier, le reste de l'étape de reconnaissance de texte s'applique uniquement au texte à taille non fixe.

5.10.5 Classification

La classification se fait en deux étapes, dans la première étape, un "élagueur de classe" crée une liste de classes de caractères auxquelles les caractères inconnus pourront correspondre.

Chaque fonction récupère, à partir d'une table de consultation en 3 dimensions un vecteur binaire de classe, les vecteurs de bits sont additionnés sur toutes les caractéristiques ainsi les classes avec la grande somme deviennent alors les classes de caractères auxquelles les caractères inconnus pourront correspondre. Chaque caractéristique de l'inconnu regarde un vecteur de bits de prototypes de la classe étant donné que cela pourrait correspondre, puis la similitude réelle entre eux est calculée. Chaque classe de caractère "prototype" est représentée par une expression logique de somme de produit avec chaque terme appelé une "configuration", le processus de calcul de la distance conserve un enregistrement de la preuve totale de similitude de chaque fonction dans chaque configuration.

Troisième partie

Annexe

Annexe A

Contents

| | |
|--------------------------------------|------------|
| A.1 Opérations morphologiques | 109 |
| A.1.1 Définition | 109 |
| A.1.2 Ensemble caractéristique | 109 |
| A.1.3 Élément structurant | 109 |
| A.1.4 Dilatation | 110 |
| A.1.5 Erosion | 112 |
| A.1.6 Ouverture | 113 |
| A.1.7 Fermeture | 113 |
| A.2 Les filtres | 114 |
| A.2.1 Filtres de lissage | 114 |
| A.2.2 Filtrage global | 115 |
| A.2.3 Filtrage local | 115 |
| A.2.4 Filtre passe-bas | 115 |
| A.2.5 Filtre passe-haut | 115 |
| A.2.6 Filtre Gaussien | 116 |
| A.3 Convolution | 116 |
| A.3.1 Convolution spatiale | 117 |
| A.3.2 Gestion des bords | 119 |
| A.4 Gradient | 120 |
| A.4.1 Direction du gradient | 120 |
| A.4.2 Quelques opérateurs Gradient | 121 |

A.1 Opérations morphologiques

A.1.1 Définition

Les opérations morphologiques sont des opérations de traitement d'image qui vise à modifier des propriétés et des caractéristiques d'une image d'origine, en particulier la structure ou la forme des objets présents dans l'image dans le but d'améliorer séparer l'information présente dans l'image, ces opérations peuvent s'appliquer tant bien sur des images en couleurs qu'à celles en niveau de gris.

Formellement, la transformation morphologique modifie la valeur d'un pixel en fonction de la valeur de ses voisins, pour ce faire, on utilise un élément structurant qui est un masque binaire qui permet de prendre en compte le voisinage du pixel. On peut distinguer plusieurs opérations de base, parmi elles la dilatation, l'érosion, l'ouverture et la fermeture binaire.

A.1.2 Ensemble caractéristique

Une image binaire peut être caractérisée par l'ensemble de ses pixels de valeur 1 (pixels noirs). Cet ensemble est souvent appelé ensemble caractéristique de l'image.[6]

A.1.3 Élément structurant

Un élément structurant est un masque binaire (constitué de pixels blancs et noirs) muni d'un point d'ancrage. Voici, à titre d'exemple, quatre éléments structurants, le point d'ancrage étant marqués par un point jaune :

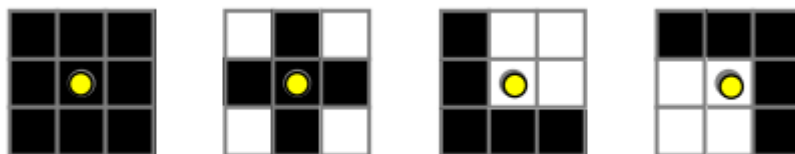


FIGURE A.1 – Exemples d'élément structurant

Soit (x,y) les coordonnées d'un pixel d'une image binaire X et M un élément structurant. Alors $M(x,y)$ représente l'ensemble des pixels noirs de X qui coïncident avec les pixels noirs de M lorsque le point d'ancrage est superposé au pixel de coordonnées (x,y) . Les pixels "blancs" peuvent être interprétés comme "transparents".

Formellement :

Dans ce qui va suivre posons un ensemble d'éléments $E = Z^2$ (l'ensemble des entiers relatifs) qui représente une modélisation du support des images binaires à deux dimensions. Soit B (l'élément structurant) un sous-ensemble de E , si x est un élément de E alors nous noterons B_x l'ensemble B translaté de x

$$B_x = \{b + x/b \in B\} \quad (\text{A.1})$$

L'élément structurant peut être défini comme une sorte de sonde ou un ensemble dont on se donne la forme et la taille, et dont on particularise un point (l'origine) qu'on appelle le centre (qui n'est pas forcément le centre géométrique). Il parcourt toute l'image à traiter et à chaque position de cet élément on étudie sa relation avec l'image binaire en considérons que cette dernière est un ensemble, Ces relations peuvent être du type "est inclut dans l'ensemble" ou "touche à l'ensemble" [7].

Exemple d'un élément structurant :

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

Dans cet exemple l'élément structurant placé au centre, permet de prendre en compte le voisinage du pixel et d'étudier les relations entre ces derniers.

A.1.4 Dilatation

La dilatation a comme premier but d'élargir la figure ou l'objet en question. 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, en d'autres terme c'est l'ajout de la valeur de l'élément structurant à la hauteur et à la largeur de la figure d'origine. Si l'élément structurant est décentré, la dilatation décalera la figure dans le même sens. Enfin les coins de la figure seront déformés en fonction de l'élément structurant.

Formellement :

Soit X une figure, un ensemble de pixels.

Pour un élément structurant B , la dilatation de X par B est l'ensemble obtenu en remplaçant chaque pixel p de X par sa fenêtre B_p [2] :

$$Dil_B(X) = \cup\{B_p \mid p \in X\} \quad (\text{A.2})$$

Autre définition :

Soit X un sous-ensemble de E La dilatation morphologique avec l'élément structurant B est définie comme la somme de **Minkowski**¹

$$Dil_B(X) = X \oplus B = \{x + b \mid b \in B, x \in X\} = \cup_{x \in X} B_x \quad (\text{A.3})$$

Exemple en image de dilatation d'une image :

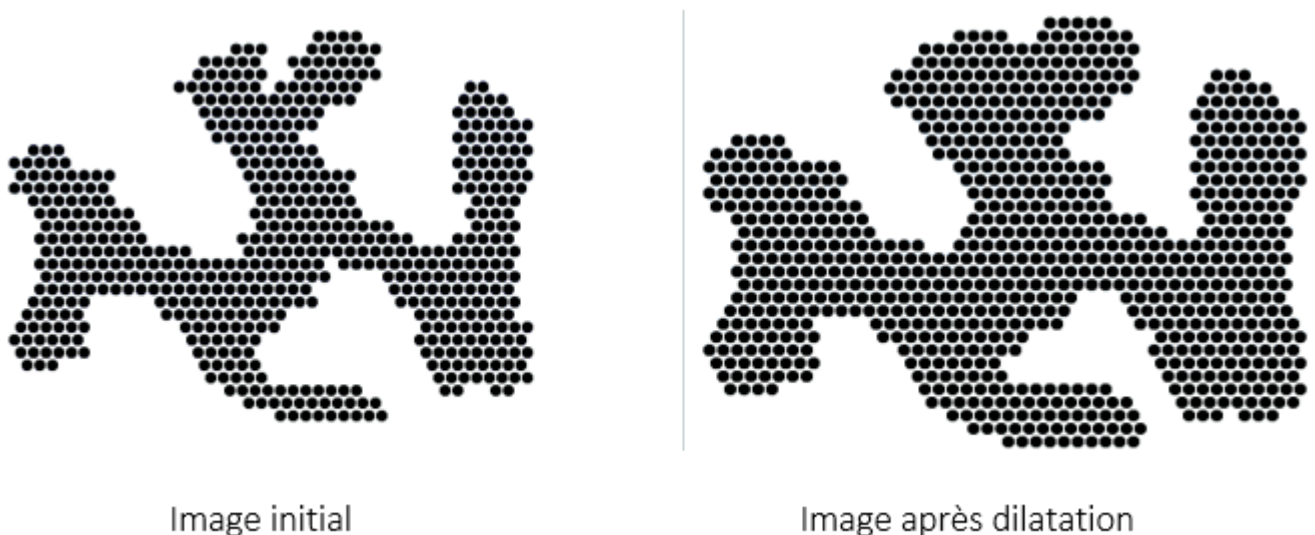


FIGURE A.2 – Exemple Illustrant l'effet d'une dilatation sur une image

1. Opération sur les parties d'un espace vectoriel à deux parties A et B elle associe leur ensemble (somme), formé des sommes d'éléments de A et de B. On note \oplus

A.1.5 Erosion

L'effet de l'érosion est contraire à celui de la dilatation, il a pour objectif de rétrécir la figure, la hauteur et largeur de la figure érodée seront les différences respectivement des hauteurs et largeurs de la figure originale 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.

Formellement :

Soit X une figure, et B un élément structurant.

L'érosion de X par B est l'ensemble des pixels p tels que la fenêtre B_p est incluse dans X [3] :

$$Eros_B(X) = \{ p \mid B_p \subseteq X \} \quad (\text{A.4})$$

Exemple illustratif d'érosion d'une image :

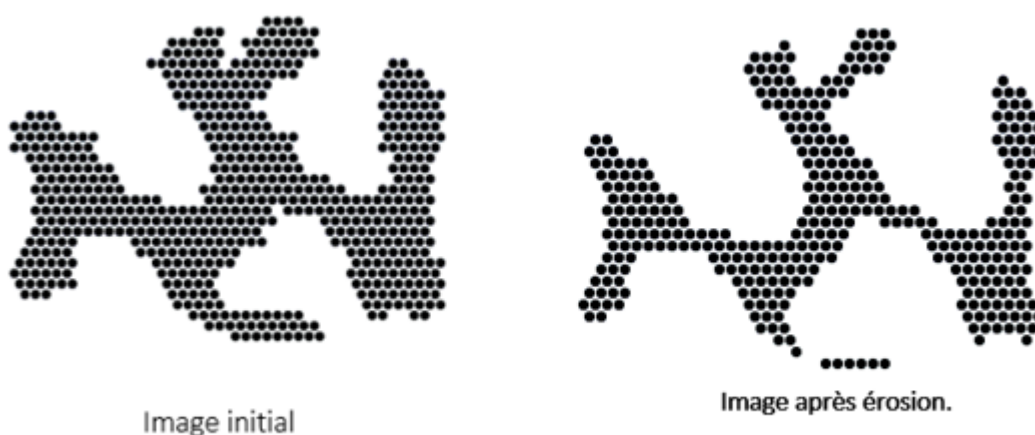
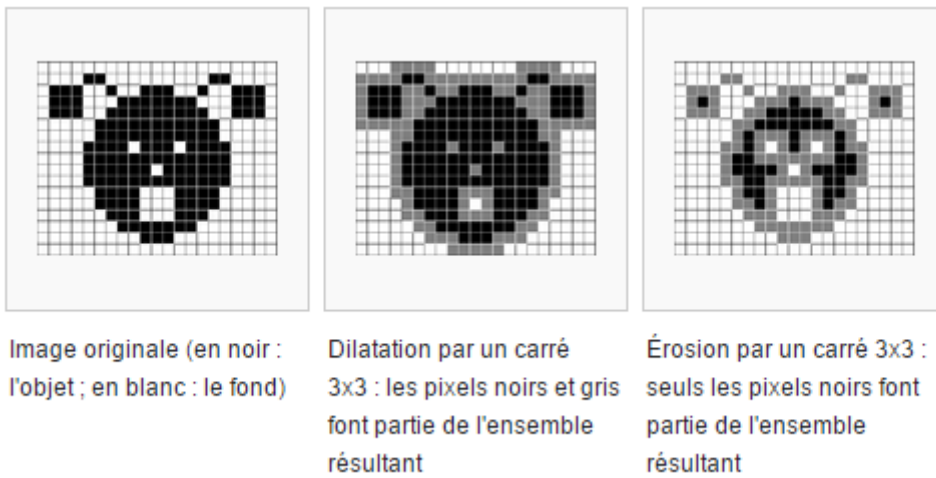


FIGURE A.3 – Exemple Illustrant l'effet d'une érosion sur une image

Exemple récapitulatif :



A.1.6 Ouverture

Soit une image binaire X et un élément structurant M . L'ouverture de X par M est une image binaire définie par : $Ouv_M(X) = Dil_M^-(Ero_M(X))$ où M^- représente l'élément structurant symétrique de M par rapport au point d'ancrage.

A.1.7 Fermeture

Soit une image binaire X et un élément structurant M . La fermeture de X par M est une image binaire définie par $Fer_M(X) = Ero_M^-(Dil_M(X))$

Illustration d'une ouverture

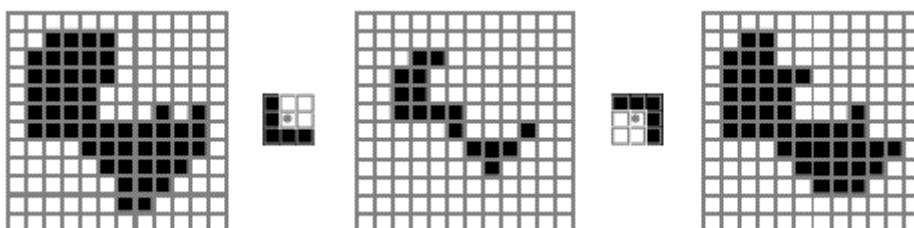


FIGURE A.4 – Illustration d'une ouverture

L'image de droite représente l'ouverture de l'image de gauche par un élément structurant en "L"; elle est obtenue par la composition d'une érosion et d'une dilatation.

Illustration d'une fermeture

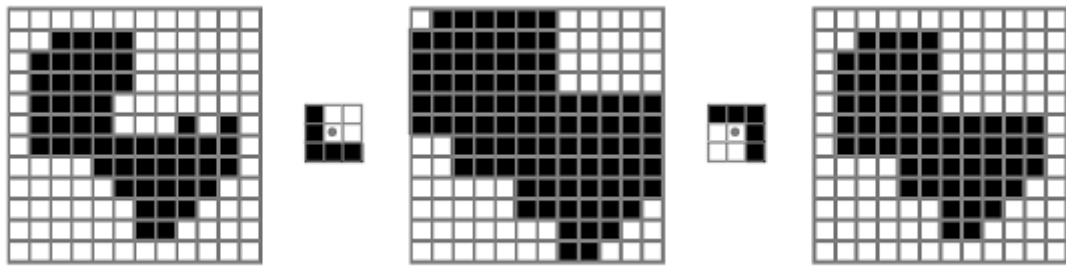


FIGURE A.5 – Illustration d'une fermeture

L'image de droite représente la fermeture de l'image de gauche par un élément structurant en "L"; elle est obtenue par la composition d'une dilatation et d'une érosion.

A.2 Les filtres

A.2.1 Filtres de lissage

Un système d'enregistrement d'image ne restitue pas l'image de manière parfaite. Le principe du filtrage est de modifier la valeur des pixels d'une image, généralement, dans le but d'améliorer son aspect[25]. En pratique, il s'agit de créer une nouvelle image en se servant des valeurs des pixels de l'image d'origine en éliminant les éléments non significatifs (perturbateurs) soit pour améliorer leur visualisation, soit pour les simplifier dans le but d'un traitement postérieur.

A.2.2 Filtrage global

Dans ce type de filtrage, chaque pixel de la nouvelle image est calculé en prenant en compte la totalité des pixels de l'image de départ[25].

A.2.3 Filtrage local

Dans cette manière de procéder, chaque pixel de la nouvelle image est calculé en considérant seulement les pixels voisins ou de proximité au pixel correspondant dans l'image d'origine. Il est d'usage d'utiliser des voisinages carrés et symétriques autour du pixel considéré[25] sous forme de tableaux à 2 dimension de taille impaire.

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

Voisinage 3x3 du pixel x,y

$$\begin{pmatrix} P_{x-2,y-2} & P_{x-1,y-2} & P_{x,y-2} & P_{x+1,y-2} & P_{x+2,y-2} \\ P_{x-2,y-1} & P_{x-1,y-1} & P_{x,y-1} & P_{x+1,y-1} & P_{x+2,y-1} \\ P_{x-2,y} & P_{x-1,y} & P_{x,y} & P_{x+1,y} & P_{x+2,y} \\ P_{x-2,y+1} & P_{x-1,y+1} & P_{x,y+1} & P_{x+1,y+1} & P_{x+2,y+1} \\ P_{x-2,y+2} & P_{x-1,y+2} & P_{x,y+2} & P_{x+1,y+2} & P_{x+2,y+2} \end{pmatrix}$$

Voisinage 5x5 du pixel x,y

A.2.4 Filtre passe-bas

Permet un adoucissement des détails, ainsi qu'une réduction du bruit en laissant passer les basses fréquences et en atténuant les hautes fréquences, c'est-à-dire les fréquences supérieures à la fréquence de coupure

A.2.5 Filtre passe-haut

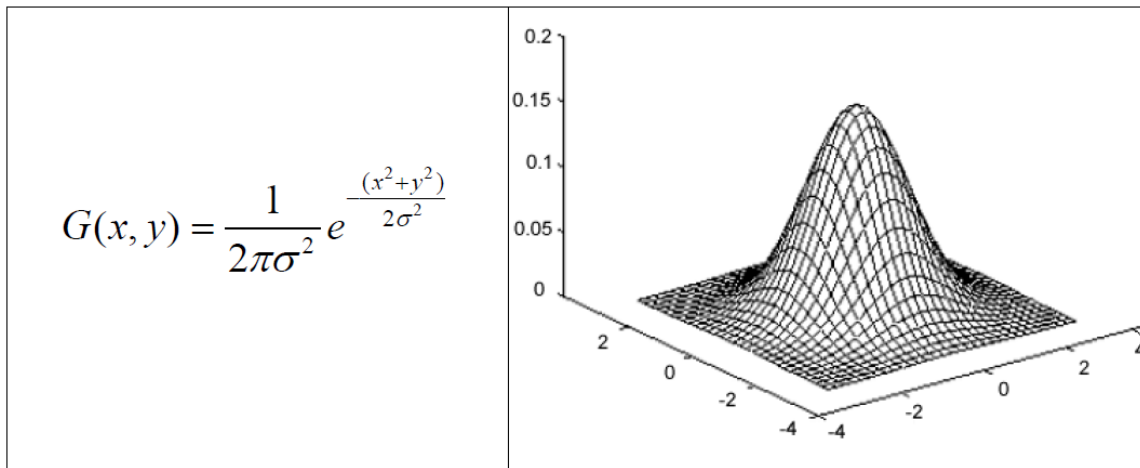
Favorise les hautes fréquences spatiales, comme les détails, et de ce fait, il améliore le contraste². Un filtre "passe haut" est caractérisé par un noyau comportant des

2. Propriété qui quantifie la différence entre les parties claires et foncées d'une image

valeurs négatives autour du pixel central.[28]

A.2.6 Filtre Gaussien

Le filtre Gaussien est un filtre isotrope spécial avec des propriétés mathématiques bien précises. Dans le traitement d'images on traite des données à deux dimensions (X et Y), on introduit donc une fonction gaussienne à deux dimensions $G(x, y)$:



A.3 Convolution

Opérateur bilinéaire³ et produit commutatif, généralement noté * qui à deux fonctions f et g sur un même domaine infini, fait correspondre une autre fonction $f * g$ sur ce domaine, qui en tout point de celui-ci est égale à l'intégrale sur l'entièreté du domaine (ou la somme si celui-ci est discret) d'une des deux fonctions autour de ce point, pondérée par l'autre fonction autour de l'origine.[13] Il est défini dans un domaine continu par :

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(x - t) \cdot g(t) dt \quad (\text{A.5})$$

3. Une fonction est dite bilinéaire si elle est linéaire par rapport à ses deux variables.

A.3.1 Convolution spatiale

Dans le domaine continu, une convolution spatiale entre une image I et un filtre H est définie par :

$$R(x,y)=(I*H)(x,y)=(H*I)(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}I(x-x',y-y').H(x',y')dx'dy'$$

(A.6)

Quant au domaine discret, la convolution spatiale se calcule par :

$$R(x,y)=\sum_{i=-n}^{+n}\sum_{j=-m}^{+m}I(x+i,y+j).H(i,j)$$

(A.7)

Concrètement, nous allons utiliser le produit de convolution pour multiplier des matrices entre elles, ainsi dans l'exemple qui suit, nous mettons en jeu deux matrices très différentes : la matrice image, très grande (par exemple 512 x 512, ce qui représente 262144 pixels) et une matrice plus petite qu'on appelle le noyau parce que c'est le "cœur" de tous les changements qui vont affecter l'image. Le noyau va donc agir sur chacun des pixels, c'est à dire sur chacun des éléments de la matrice "image".

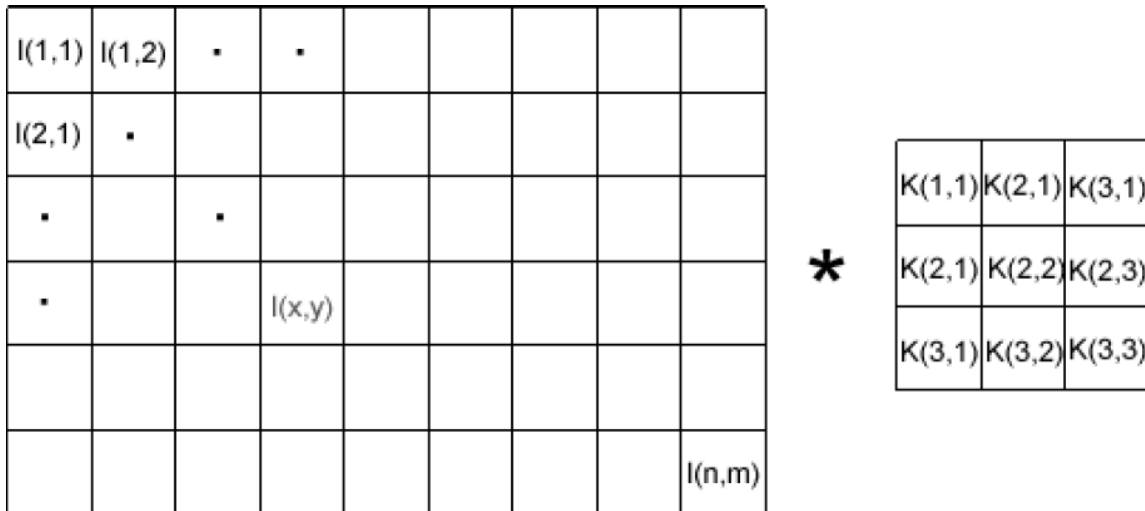


FIGURE A.6 – Produit de convolution

Dans la figure ci-dessus, l'image est représentée par la matrice $[i]$ composée de $n \times m$ éléments. Le noyau est quant à lui composé de la matrice carrée $[k]$ de 3×3 éléments.

Pour calculer la valeur d'un pixel $I(x,y)$ de la matrice image, on multiplie sa valeur par le pixel central du noyau $K(2,2)$ et on additionne ensuite la valeur des produits des pixels adjacents. Il reste ensuite à diviser le résultat par le nombre d'éléments du noyau. La dernière opération n'appartient pas au produit de convolution proprement dit, mais elle nécessaire pour maintenir la dynamique de l'image⁴ ainsi que sa linéarité.

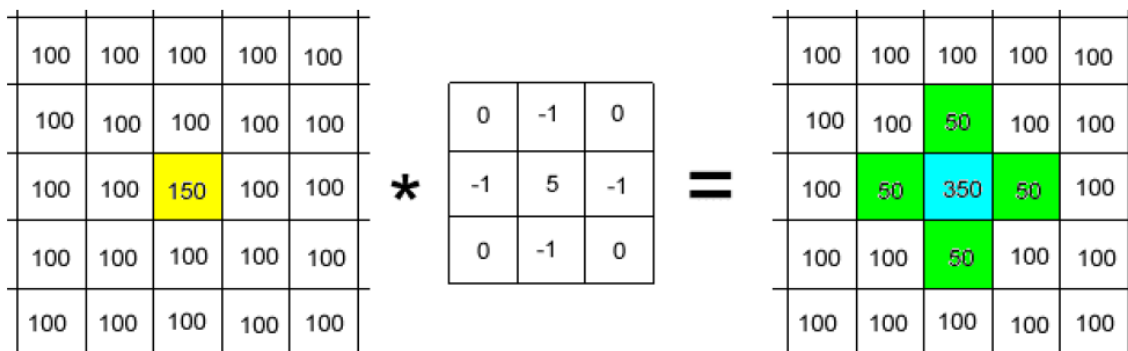


FIGURE A.7 – Exemple illustratif du calcul du produit de convolution

4. Différence entre le niveau de pixel le plus élevé et le plus faible

Si on applique un produit de convolution de cet exemple au pixel central on obtient, d'après les règles mathématiques

$$(150 \times 5) + (-1 \times 100) + (-1 \times 100) + (-1 \times 100) + (-1 \times 100)$$

ce qui est égal à 350.

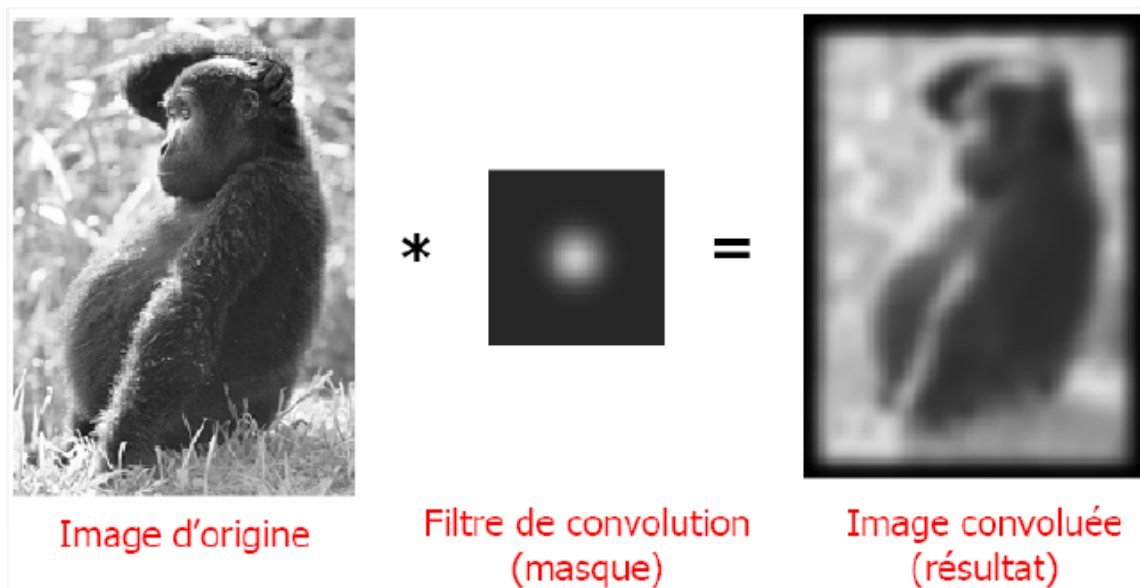


FIGURE A.8 – Exemple de filtre de convolution

A.3.2 Gestion des bords

Dans le cas où le pixel considéré est proche du bord de l'image, il se trouve que certains points du voisinage sont en dehors de l'image d'origine, il est donc convenable de choisir une stratégie pour gérer les pixels extérieurs, on va illustrer dans ce qui suit les méthodes les plus couramment utilisées :

1. Mise à zéro : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est considérée comme nulle, ce qui veut dire que $\text{Image}[-1][y]$ est égal à 0.
2. Continuité : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est celle du pixel le plus proche qui est dans l'image d'origine. C'est à dire : $\text{Image}[-1][y] = \text{Image}[0][y]$.

3. Miroir : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est celle du pixel symétrique par rapport au bord de l'image. C'est à dire : $\text{Image}[-1][y] = \text{Image}[1][y]$.
4. Sphérique : Si un pixel du voisinage est en dehors de l'image d'origine, sa valeur est celle du pixel correspondant si l'image était projetée sur une sphère. C'est à dire : $\text{Image}[-1][y] = \text{Image}[\text{Largeur}-1][y]$. [25]

Le produit de convolution constitue ainsi une base d'application pour la majorité des filtres qu'on va pouvoir utiliser.

A.4 Gradient

En considérant une image dans un repère orthogonal (Oxy), le gradient de l'image en tout point ou pixel de coordonnées (x,y) est désigné par

$$\vec{\text{grad}} f = \vec{\nabla} f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

Module du Gradient

Le module du gradient permet de quantifier l'importance du contour mis en évidence, c'est-à-dire l'amplitude du saut d'intensité relevé dans l'image[23] :

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

A.4.1 Direction du gradient

La direction du gradient permet de déterminer l'arête présente dans l'image. En effet, la direction du gradient est orthogonale à celle du contour :

$$\alpha_0 = \arctan \frac{\partial f / \partial y}{\partial f / \partial x}$$

Le principe de la détection de contours par l'utilisation du gradient consiste premièrement à calculer le gradient de l'image dans deux directions orthogonales

et ensuite calculer le module du gradient. Il s'agira ensuite de procéder à une sélection des contours les plus marqués, à savoir les points de plus fort contraste par un seuillage adéquat.

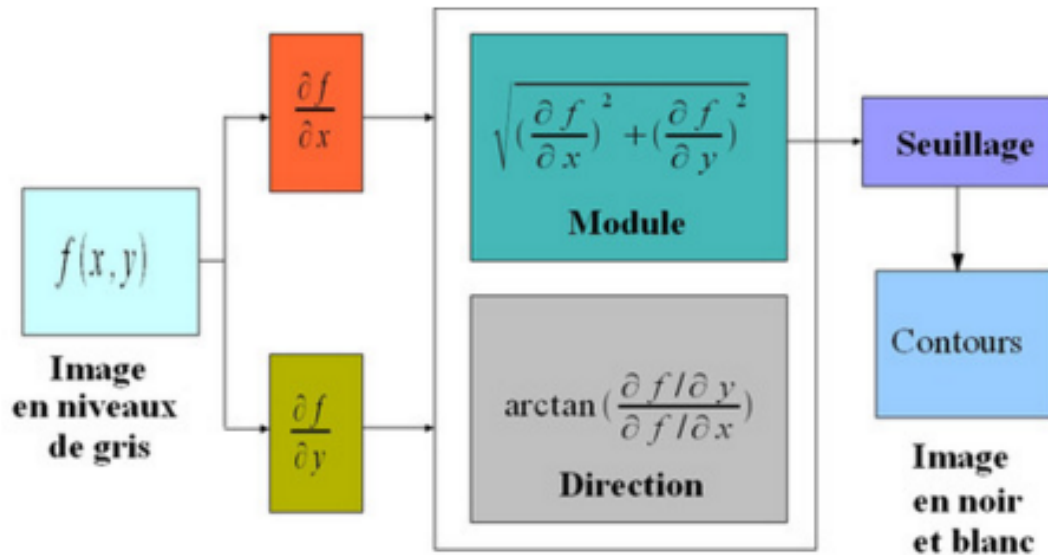


FIGURE A.9 – Schéma récapitulatif de la détection de contours par gradient

A.4.2 Quelques opérateurs Gradient

Ces opérateurs sont à considérer comme des filtres qui vont être corrélés⁵ à l'image. Nous allons passer en revue deux masques (filtres) pour illustrer la détection des contours en utilisant des opérateurs de gradient.

Filtre de Sobel et Prewitt

Les opérateurs de Sobel et de Prewitt permettent d'estimer localement la norme du gradient spatial bidimensionnel d'une image en niveau de gris. Ils amplifient les régions de fortes variations locales d'intensité correspondant aux contours [24]

Chaque pixel des masques est normalisé par :

$$\frac{1}{C + 2}$$

5. Corrélation :Relation existant entre deux notions dont l'une ne peut être pensée sans l'autre, entre deux faits liés par une dépendance nécessaire[20]

Le calcul de gradient est mené par l'intermédiaire de deux masques, le premier effectuant un gradient horizontal, le second un gradient vertical. Le deuxième masque se déduit du premier par une rotation de $\pi/2$. Les masques sont donnés ci-dessous pour les contours horizontaux puis verticaux. L'application séparée de chacun des masques donne une estimation des composantes horizontales et verticales du gradient, notées respectivement ∇_x et ∇_y par un simple filtrage linéaire avec un masque 3x3.[24]

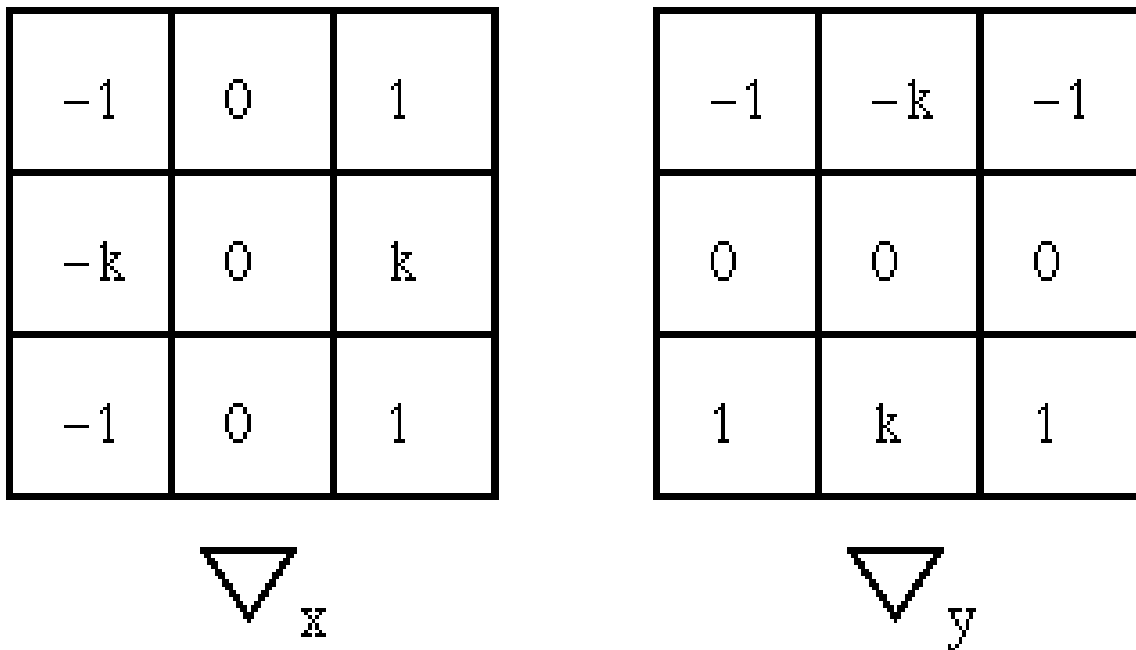


FIGURE A.10 – Masques de convolution des operateurs de Prewitt ($k=1$) et de Sobel ($k=2$)

Par la suite, il est possible de calculer la norme (module) et la direction du gradient en chaque point à partir des composantes ∇_x et ∇_y . La norme du gradient ainsi estimée correspond à l'intensité attribuée au pixel courant. C'est donc l'image de la norme du gradient que l'on visualise généralement. Ces masques ont comme atout de produire deux effets. En sus de calculer le gradient dans une direction, ces masques effectuent un lissage dans la direction orthogonale. Ce lissage rend ces masques un peu moins sensibles au bruit. On considère comme origine de ces masques le pixel central. La réponse impulsionnelle⁶ $h(m,n)$ de ces

6. la réponse impulsionnelle est la sortie qui est obtenue lorsque l'entrée est une impulsion, c'est-à-dire une variation soudaine et brève du signal

masques pour la mise en évidence des contours horizontaux est telle que :

$$h(-1,1) = h(1,1) = 1$$

$$h(-1,-1) = h(1,-1) = -1$$

$$h(0,1) = c$$

$$h(0,-1) = -c$$

La sortie obtenue après filtrage est :

$$\begin{aligned} [\text{H}] \sum_{i=-n}^{+n} \sum_{j=-m}^{+m} h(m,n) \cdot f(m+1, n+j) &= \frac{1}{c+2} \cdot (h(-1,1) \cdot f(i-1, j+1) + \\ &h(0,1) \cdot f(i, j+1)) + \frac{1}{c+2} \cdot (h(1,1) \cdot f(i+1, j+1) + h(-1,-1) \cdot f(i-1, j- \\ &1)) + \frac{1}{c+2} \cdot (h(-1,0) \cdot f(i-1, j) + h(-1,1) \cdot f(i-1, j+1)) \end{aligned}$$

$$\begin{aligned} \sum_{i=-n}^{+n} \sum_{j=-m}^{+m} h(m,n) \cdot f(m+1, n+j) &= \frac{1}{c+2} \cdot (f(i-1, j+1) + cf(i, j+1) + f(i+ \\ &1, j+1)) - \frac{1}{c+2} \cdot (f(i-1, j+1) + cf(i-1, j) + f(i-1, j+1)) \end{aligned}$$

L'équation laisse apparaître la double action avec un moyennage horizontal sur trois pixels sur les lignes au dessus et au dessous du pixel central et un calcul de gradient vertical entre les deux lignes.[23] Pour la mise en évidence des contours verticaux, c'est l'autre filtre qui est utilisé. La sortie obtenue après filtrage peut se mettre sous la forme :

$$\begin{aligned} \sum_{i=-n}^{+n} \sum_{j=-m}^{+m} h(m,n) \cdot f(m+1, n+j) &= \frac{1}{c+2} \cdot (f(i+1, j+1) - f(i-1, j+1)) - \\ &\frac{1}{c+2} \cdot (cf(i+1, j) - cf(i-1, j)) + \frac{1}{c+2} \cdot (f(i+1, j-1) - f(i-1, j-1)) \end{aligned}$$

La figure qui va suivre propose une illustration des masques de sobel :

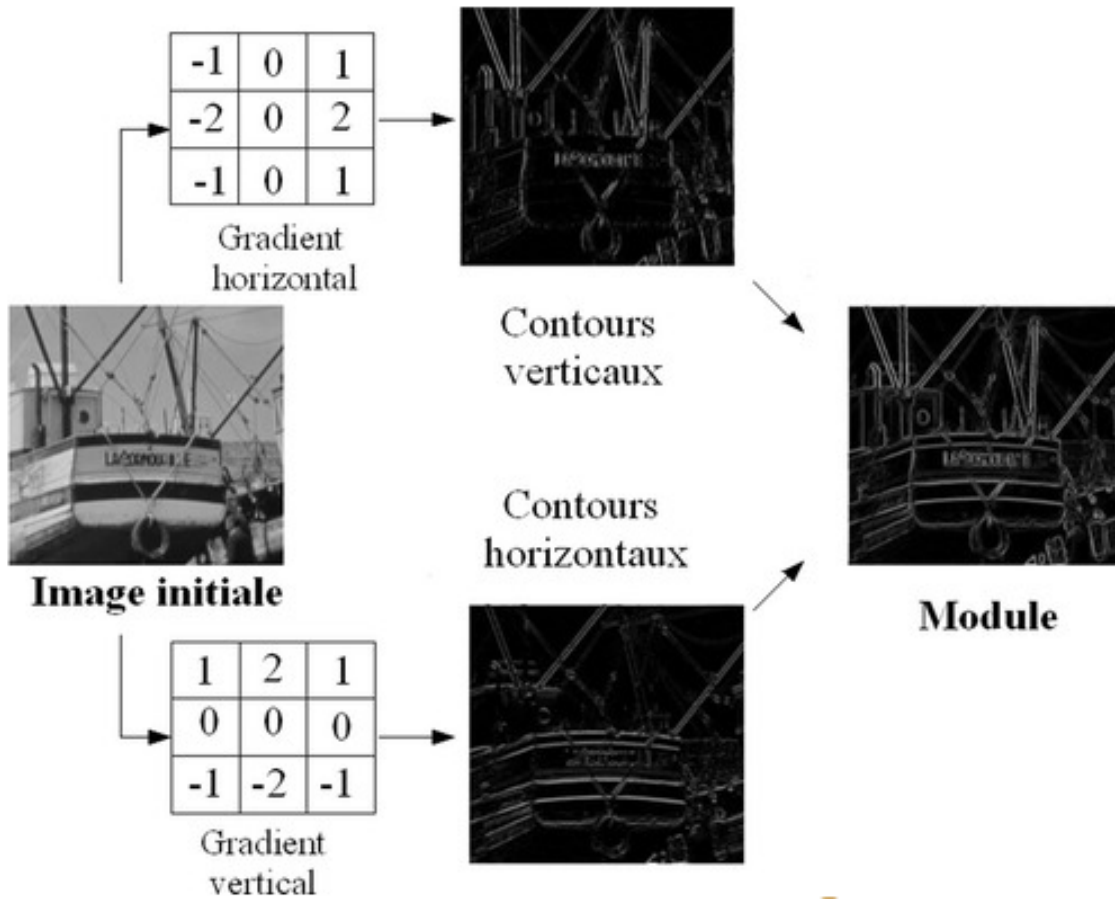


FIGURE A.11 – Exemple d'application de l'opérateur de Sobel

Conclusion Générale

En raison de la croissance continue de l'utilisation de l'audio-visuel, le type d'information disponible pour les utilisateurs est de plus en plus enrichi, en effet, l'information véhiculée dans ces derniers prend de plus en plus d'ampleur et elle est de plus en plus cruciale dans divers systèmes et applications comme les applications de reconnaissance ou encore la reconnaissance de plaques d'immatriculation dans des images comme le cas qu'on a traité dans notre mémoire. L'utilisation de cette information contenue dans des images ou dans des vidéos facilite effectivement diverses opérations et processus, notamment dans le domaine de la sécurité ainsi que la classification automatique des informations contenues dans ces dernières. Dans ce présent mémoire nous avons présenté une approche permettant la détection et lecture automatique de plaque d'immatriculation Algériennes à partir d'une image numérique, pour ce faire, on eu recours a plusieurs notions et concepts notamment en traitement d'images afin d'effectuer les prétraitements nécessaires qui nous ont considérablement facilité les phases suivantes de l'approche, nous avons ensuite fortement côtoyé diverses fonctions de la bibliothèque multimédia OpenCv afin d'imposer des contraintes fondamentales et spécifiques aux plaques d'immatriculation ce qui nous a permis de correctement les détecter et de les localiser. Une fois que nous avons pu localiser les plaques minéralogiques sur l'image nous avons entamé la deuxième phase du traitement dans un souci de pouvoir lire et exploiter les caractères figurant sur la plaque, pour cela, nous avons abordé la notion de Reconnaissance Optique des Caractères (OCR). Ainsi la réalisation de ce système nous a permis d'acquérir et d'asseoir de nouvelles connaissances tant bien théoriques que pratiques, de nous familiariser avec certains concepts du vaste domaine de l'intelligence artificielle et surtout mettre en pratique toutes les connaissances acquises tout au long de notre cursus universitaire. Toutes fois, le système réalisé tout au long de ce mémoire, nécessite certainement une maintenance et une amélioration afin de s'adapter aux différents changements qui peuvent survenir. A cet effet nous avons dressé une liste de perspectives et de fonctionnalités à ajouter ainsi qu'à améliorer dans notre application qui sont les suivantes :

- La détection des plaques d'immatriculation : en effet, dans certaines conditions

défavorables (mauvaise qualité de l'image fournie, l'angle de la prise de l'image) le système détecte mal la plaque, on envisage donc d'ajouter d'autres contraintes et certains prétraitements de l'image permettant un meilleur taux de détection.

- Taux de reconnaissance : L'OCR utilisé dans notre mémoire présente encore certaines erreurs de reconnaissance et confond parfois entre les caractères qui ont des gabarits très similaires, la perspective est donc d'opérer des traitements intelligents après le passage de l'OCR pour un meilleur taux de reconnaissance.
- Faire une application avec une base de données répartie, ce qui va permettre une meilleure fluidité des informations.
- Migrer vers des solutions mobiles et plus accessibles.
- Migrer vers une application temps réel, pouvant détecter à partir d'une vidéo.

Bibliographie

- [1] <http://www.xavierdupre.fr/>.
- [2] <http://france.abbyy.com/finereader/qu-est-ce-que-l-ocr/reconnaissance-texte>.
- [3] http://glotin.univ-tln.fr/MCBIR/Segmentation;images_principes.pdf.
- [4] <http://perso.ensta-paristech.fr/~manzaner/Cours/IAD/AMsegmentation.pdf>.
- [5] <http://scikit-learn.org/stable/modules/multiclass.html>.
- [6] <https://diuf.unifr.ch/courses/04-05/improc/Slides/slides/ImgProc11MorphOp.pdf>.
- [7] <https://dpt-info.u-strasbg.fr/~cronse/TIDOC/MM/deof.html>.
- [8] https://en.wikipedia.org/wiki/Optical_character_recognition.
- [9] https://en.wikipedia.org/wiki/Region_growing.
- [10] https://fr.wikipedia.org/wiki/Apprentissage_automatiqueTypes_d.27apprentissage.
- [11] <https://fr.wikipedia.org/wiki/NumPy>.
- [12] <https://fr.wikipedia.org/wiki/OpenCV>.
- [13] https://fr.wikipedia.org/wiki/Produit_de_convolution.
- [14] https://fr.wikipedia.org/wiki/Résolution_spatiale_des_images_matricielles.
- [15] https://www.cs.cf.ac.uk/Dave/Vision_lecture/node35.html.
- [16] <https://www.univ-lemcen.dz/~benmammari/IA2.pdf>.
- [17] http://urfist.enc.sorbonne.fr/anciensite/image_numerique/segmentation.html.
- [18] <http://www-ia.lip6.fr/~tollaris/ARTICLES//Articles/DEA2003/node3.html>.
- [19] <http://www.a2ia.com/fr/la-reconnaissance-optique-de-caractères>.
- [20] <http://www.larousse.fr/dictionnaires/francais/correlation/19435>.
- [21] <http://www.lifl.fr/~pietquin/teaching/FAACours1.pdf>.

- [22] <http://www.mit.edu/9.520/spring09/Classes/multiclass.pdf>.
- [23] http://www.optique-ingenieur.org/fr/cours/OPI_frM04C05/co/Contenu.html.
- [24] http://www.tsi.enst.fr/pages/enseignement/ressources/mti/localisation_et_paisseur/pres_P.S.h.
- [25] <http://xphilipp.developpez.com/articles/filtres>.
- [26] *OPENCV Détection d'objets par le classificateur de Haar 11/11/2015*.
- [27] www.edu.upmc.fr/physique/lp326/dossiers/phys-chap2.pdf.
- [28] xmcvs.free.fr/astroart/Chapitre4.
- [29] Hinde ANOUAL. *Détection et Localisation de texte dans les images de scènes naturelles : Application à la détection des plaques dimmatriculation marocaines*. PhD thesis, 2012.
- [30] Laurent Miclet Antoine Cornuéjois. *Apprentissage artificiel concept et algorithmes*. PhD thesis.
- [31] Eric Nowak. *Reconnaissance de catégories d'objets et d'instances d'objets à l'aide de représentations locales*. PhD thesis, 2008.
- [32] Johann Poignant. *Détection et reconnaissance de texte dans les documents vidéos- et leurs apports à la reconnaissance de personnes*.
- [33] Ray Smith. *An Overview of the Tesseract OCR Engine*. PhD thesis, 2007.