

Sommaire

| | |
|--|----------|
| Remerciements..... | i |
| Résumé | ii |
| Sommaire | iii |
| Listes des figures et tableaux | vi |
| Introduction générale..... | 1 |
| <i>Chapitre1 : Etat de l'art</i> | |
| 1.1 Introduction :..... | 3 |
| 1.2 Lecture automatique de plaque d'immatriculation :..... | 3 |
| 1.3 La technique de la reconnaissance optique de caractères :..... | 4 |
| 1.4 Les algorithmes des systèmes LAPI :..... | 4 |
| 1.5 Difficultés des systèmes LAPI :..... | 5 |
| 1.6 Domaines d'applications des systèmes LAPI :..... | 5 |
| 1.7 Quelques systèmes LAPI existants :..... | 5 |
| 1.7.1 Autovu :..... | 6 |
| 1.7.1.1 Avantages d'Autovu :..... | 6 |
| 1.7.2 LAPI-ENGINE :..... | 6 |
| 1.7.2.1 Avantages de LAPI-ENGINE :..... | 6 |
| 1.7.3 SeeTec :..... | 7 |
| 1.7.3.1 Avantages de SeeTec :..... | 7 |
| 1.7.4 Asia Vision Technology Limited (AVT) :..... | 7 |
| 1.7.4.1 Avantages d'Asia Vision Technology Limited :..... | 7 |
| 1.7.5 Technologie AGL (Application de Gestion Locale) :..... | 7 |
| 1.7.5.1 Avantages d'AGL :..... | 8 |
| 1.7.6 Systèmes LAPI-Pryncar :..... | 8 |
| 1.7.6.1 Avantages de LAPI-Pryncar :..... | 8 |
| 1.7.7 Système LAPIA :..... | 8 |
| 1.7.7.1 Avantages du système LAPIA :..... | 9 |
| 1.8 Conclusion :..... | 9 |
| <i>Chapitre2 : Traitement d'images</i> | |
| 2.1 Introduction :..... | 10 |
| 2.2 Caractéristiques des plaques d'immatriculation Algérienne :..... | 10 |
| 2.3 Définitions du traitement d'images :..... | 11 |

| | |
|--|----|
| 2.4 Image : | 12 |
| 2.5 Représentions d'image : | 13 |
| 2.5.1 Image matricielle : | 13 |
| 2.5.2 Image vectorielle : | 13 |
| 2.5.3 Comparaison entre image matricielle et image vectorielle : | 14 |
| 2.6 Principe d'acquisition d'image : | 14 |
| 2.7 Caractéristiques d'une image : | 15 |
| 2.7.1 Taille : | 15 |
| 2.7.2 Résolution : | 15 |
| 2.7.3 Bruit : | 15 |
| 2.7.4 Contraste : | 15 |
| 2.7.5 Histogramme : | 16 |
| 2.7.6 Luminance : | 16 |
| 2.7.7 Image en niveaux de gris : | 16 |
| 2.7.8 Image en couleur : | 16 |
| 2.8 Les principales étapes de traitement d'image : | 17 |
| 2.8.1 Prétraitement des images : | 17 |
| 2.8.2 Amélioration d'images : | 17 |
| 2.8.3 Analyse d'images : | 18 |
| 2.9 Type de données manipulées : | 18 |
| 2.10 Les opérateurs de traitement d'images: | 19 |
| 2.10.1 Les opérateurs locaux : | 20 |
| 2.10.2 Filtres linéaires : | 21 |
| 2.10.3 Lissage : | 21 |
| 2.10.4 Détections de contours : | 22 |
| 2.10.5 Opérateurs morphologiques : | 22 |
| 2.10.6 Construction d'une application traitement d'images : | 23 |
| 2.11 Quelques exemples concrets traitements d'images : | 23 |
| 2.12 Conclusion : | 24 |
| Chapitre3 : Outils de développement du système APR : | |
| 3.1 Introduction : | 25 |
| 3.2 Les outils de développement : | 25 |
| 3.2.1 Le langage java : | 25 |

| | |
|---|----|
| 3.2.2 Javafx : | 25 |
| 3.2.2.1 Fonctionnement de Javafx : | 26 |
| 3.2.2.2 Avantages de Javafx : | 27 |
| 3.2.3 OpenCV : | 27 |
| 3.2.4 Scene builder : | 29 |
| 3.3 Les techniques et outils de reconnaissance optique des caractères : | 29 |
| 3.3.1 Test4j : | 29 |
| 3.3.1.1 Caractéristiques de Test4j : | 29 |
| 3.4 Conclusion : | 30 |
| Chapitre4 : Implémentation du système APR : | |
| 4.1 Introduction : | 31 |
| 4.2 Architecture du système APR : | 31 |
| 4.2.1 Architecture MVC : | 31 |
| 4.2.2 Diagramme de packages du système APR : | 32 |
| 4.3 Les interfaces de notre application : | 34 |
| 4.3.1 L'interface principale de l'application : | 34 |
| 4.4 Fonctionnement du système APR : | 35 |
| 4.4.1 La phase d'acquisition et de prétraitement de l'image : | 35 |
| 4.4.1.1 Interface pour chargement d'image : | 35 |
| 4.4.1.2 Interface pour localisation de la plaque manuellement : | 36 |
| 4.4.1.3 Conversion de l'image en niveaux de gris : | 37 |
| 4.4.1.4 Suppression ou diminution du bruit de l'image : | 37 |
| 4.4.2 Phase pour la détection des contours : | 38 |
| 4.4.2.1 La segmentation : | 38 |
| 4.4.2.1.1 Détection des bords verticaux : | 38 |
| 4.4.2.1.2 Détection des bords horizontaux : | 39 |
| 4.4.2.1.3 Jointure les bords verticaux et horizontaux : | 39 |
| 4.4.2.1.4 Opération du seuil (Threshold operation) : | 40 |
| 4.4.2.1.5 Opération morphologique : | 41 |
| 4.4.3 Phase de lecture OCR : | 42 |
| 4.5 Conclusion : | 43 |
| Conclusion générale | 44 |
| Références bibliographiques | 45 |

Liste des tableaux :

Tableau1 : les caractéristiques des images matricielles et des images vectorielles.....14

Liste des figures :

Figure 1 : la chaine de transformation pour obtenir les caractères ASCII.....3

Figure2 : exemple de plaque d'immatriculation Algérienne.....10

Figure3 : Différents types d'images.....13

Figure4 : Graphe de la hiérarchie utilisée en javafx.....26

Figure5 : Schéma des différents modules de la bibliothèque OpenCV.....27

Figure 6 : Interactions entre le modèle, la vue et le contrôleur.....31

Figure7 : Diagramme de package du système APR.....33

Figure8 : interface de traitement de l'application APR.....34

Figure9 : Interface pour chargement de l'image.....35

Figure10 : Lecture de la plaque en utilisant la localisation manuelle.....36

Figure 11 : Fenêtre de Conversion de l'image en niveaux de gris.....37

Figure 12 : Fenêtre correction du bruit de l'image.....37

Figure13 : Image obtenu après détection des bords verticaux.....39

Figure14 : Image après détection des bords horizontaux.....39

Figure15 : Image après jointure des bords verticaux et horizontaux.....40

Figure16 : Image binaire.....41

Figure17 : Image après délimitation.....41

Figure18 : Fenêtre de lecture OCR.....43

Introduction Générale

Introduction générale

Contexte :

Le numéro d'immatriculation est une information unique pour chaque véhicule, il peut être considéré comme étant son identité, il constitue également un moyen très efficace pour identifier chaque véhicule dans un territoire.

L'identification des plaques d'immatriculation utilisé surtout dans un contexte de sécurité. Les informations extraites de la plaque peuvent être utilisées pour diverses finalités, comme le contrôle d'accès et de flux, la surveillance des passages aux frontières et aux péages, la recherche de véhicules suspects ou encore la lutte contre la criminalité, etc. Ceci rend leur lecture cruciale et inévitable dans tous ces domaines.

Dans notre projet, nous nous intéressons à la reconnaissance et la lecture automatique de plaques d'immatriculation à partir de captures d'images prises sur le devant ou à l'arrière des véhicules.

Problématique et objectif :

La Lapi (Lecture automatique des plaques d'immatriculation) est la partie logicielle basée sur un ensemble d'algorithmes de reconnaissance des images et des caractères qui sont assez complexes et délicats à mettre en œuvre.

Ceci étant dit dans ce domaine la vraie problématique réside dans la partie de reconnaissance optique des caractères sur une petite partie d'image extraite des séquences enregistrées, souvent dans des conditions de grande vitesse et de faible luminosité. De plus, le fait de ne disposer que de très peu d'images hautes définition par seconde sur la plupart des caméras vidéo entraîne un manque de netteté lors de la capture de ces images. Pour cela il faut procéder à la reconnaissance des caractères qui s'apparente plus particulièrement à de l'analyse, de la manipulation, de l'amélioration d'images, de la détection de contour, pour permettre la reconnaissance optique des caractères (OCR). [1]

Ce que nous visons à travers notre travail est d'améliorer la précision de la reconnaissance des caractères du matricule en exploitant les avantages que peut offrir le traitement d'images. Pour cela nous avons développé une application qui consiste à réaliser de Prétraitements de l'image, la détection et la reconnaissance automatique des caractères.

Plan du mémoire :

Nous avons scindé notre mémoire en quatre chapitres :

Dans le premier chapitre, nous commençons par définir la lecture automatique de plaques d'immatriculation. Après ceci, nous allons présenter les algorithmes qui doivent être réalisés pour que le logiciel puisse identifier une plaque d'immatriculation et les difficultés qui en résultent. Nous terminons le chapitre par quelques exemples de systèmes existants.

Dans le deuxième chapitre, nous présentons une étude sur le traitement d'images et les notions de base nécessaires à la compréhension de ces techniques.

Le troisième chapitre, nous présentons un certain nombre d'outils que nous utilisons pour réaliser notre travail.

Le quatrième et dernier chapitre, nous présentons l'implémentation détaillée de l'application en expliquant les interfaces graphiques de notre système.

Enfin, nous terminons ce mémoire par une conclusion générale et des perspectives.

Chapitre I

Etat de l'Art

1.1 Introduction :

Ce chapitre donne un aperçu sur la lecture automatique des plaques d'immatriculation (LAPI) ainsi que la technique de reconnaissance optique de caractères (OCR). Ensuite, nous allons présenter les domaines d'applications des systèmes LAPI. Enfin, nous exposons quelques exemples de systèmes existants sur le marché en donnant leurs avantages.

1.2 Lecture automatique de plaque d'immatriculation :

La lecture automatique de plaques minéralogiques ou lecture automatisée de plaques d'immatriculation est une méthode de surveillance de masse qui utilise une série de techniques de manipulation d'images et de vision par ordinateur pour extraire les caractères alphanumériques contenu sur la plaque d'après l'image de ce dernier sous forme de code ASCII. [2]

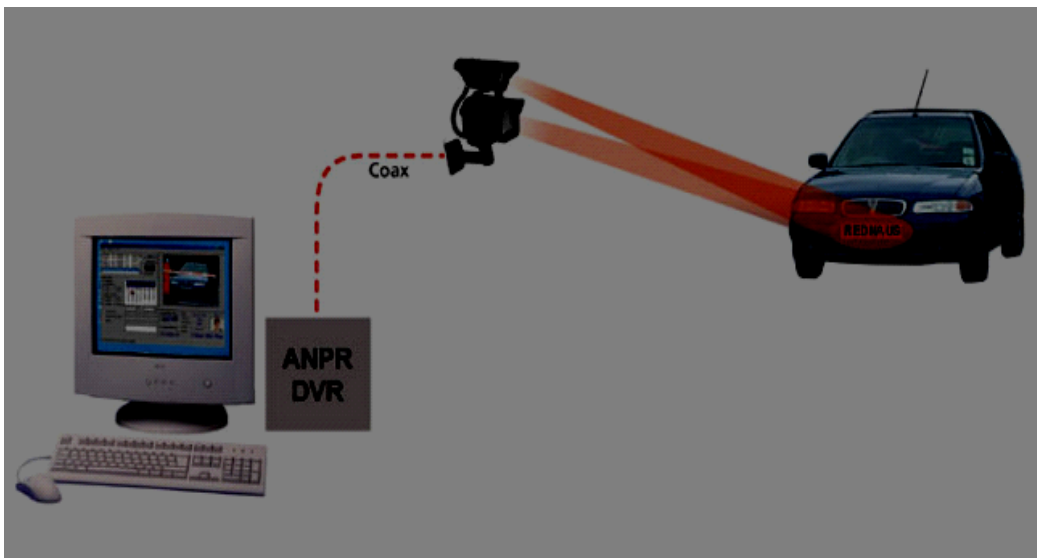


Figure 1: la chaine de transformation pour obtenir les caractères ASCII.

Un système (LAPI) met en dialogue plusieurs périphériques :

- Caméra,
- Radar de vitesse,
- Lasers de détection,
- GPS,
- Pc de surveillance.

1.3 La technique de la reconnaissance optique de caractères

La lecture automatique de plaque d'immatriculation utilise la reconnaissance optique de caractères(OCR) sur des images prises par des caméras.

La Reconnaissance Optique de Caractères ou en anglais Optical Character Recognition est une technique qui, à partir d'un procédé optique, permet à un système informatique de lire et de stocker de façon automatique du texte dactylographié, imprimé ou manuscrit sans qu'on ait à retaper ce dernier. Un logiciel OCR permet par exemple à partir d'un texte scanné, d'extraire la partie textuelle des images, et de l'éditer dans un logiciel de traitement de texte. [3]

Pour cela, le système OCR compare les couleurs noires et blanches d'un document pour déterminer chaque code alphanumérique. Le système reconnaît ensuite chaque caractère, et le convertit en texte ASCII (Code américain normalisé pour l'échange d'information).

Certaines plaques d'immatriculation varient dans la taille des caractères et leur position. Les systèmes de lecture automatique de plaques minéralogiques doivent savoir traiter ces différences pour être vraiment efficaces. Les systèmes les plus évolués savent gérer les variations entre pays, bien que beaucoup de programmes soient spécifiques à un pays.

1.4 Les algorithmes du système LAPI

Le système est composé de cinq algorithmes primaires qui doivent être réalisés pour que le logiciel puisse identifier une plaque d'immatriculation à savoir :

- **Localisation de la plaque** : responsable de trouver et d'isoler la plaque sur l'image.
- **Orientation et dimensionnement de la plaque** : compensation de l'orientation de travers de la plaque et ajustement des dimensions à la taille nécessaire.
- **Normalisation** : ajustement de l'intensité et du contraste de l'image.
- **Segmentation des caractères** : localisation des caractères sur la plaque
- **Reconnaissance optique de caractère**

La complexité de chacun de ces algorithmes détermine la précision du système. Pendant la troisième phase (normalisation), certains systèmes utilisent des techniques de **détection de contour** pour augmenter le contraste entre les lettres et la couleur de fond de la plaque. Un filtre digital peut aussi être utilisé pour réduire le « bruit » visuel de l'image.

1.5 Difficultés des systèmes LAPI

Le logiciel doit être capable de gérer un grand nombre de difficultés possibles. Parmi ces difficultés :

- **Une mauvaise résolution de l'image**, souvent parce que la plaque est trop loin, mais parfois à cause de l'utilisation d'une caméra de mauvaise qualité.
- **Des images floues**, souvent à cause du mouvement, très fréquentes sur les installations mobiles.
- **Un mauvais éclairage et un faible contraste à cause d'une surexposition, d'un reflet, ou d'ombres**
- **Un objet obscurcissant une partie de la plaque**, souvent une barre de remorquage, ou de la poussière sur la plaque
- **Une police de caractère trop originale**, chose fréquente sur les plaques fantaisie (certains pays interdisent de telles plaques, ce qui élimine le problème).
- **La surchauffe des caméras.**

1.6 Domaine d'application des systèmes LAPI

La lecture automatique de plaques minéralogiques peut être utilisée pour :

- Les passages de frontière.
- Les stations-service (enregistrement quand un client part sans payer).
- Le contrôle d'accès des parkings ou routes privées : ouverture automatique, ou enregistrement de l'entrée.
- Un outil de marketing pour enregistrer les modes de consommation.
- Les systèmes de gestion de la circulation, qui calculent la vitesse de circulation en mesurant le temps entre les passages devant deux points de lecture.
- La comparaison des plaques d'immatriculations au Fichier des véhicules volés.

1.7 Quelques systèmes LAPI existants

Le domaine des systèmes LAPI connaît une forte croissance, et actuellement une multitude de ces systèmes existe sur le marché et dans les laboratoires dont nous allons voir quelques-uns.

1.7.1 Autovu

Autovu est le système de reconnaissance automatique de plaques d'immatriculation (RAPI) sur IP qui étend la portée des systèmes de sécurité, il automatise la lecture et l'identification des plaques d'immatriculation afin d'offrir une meilleure visibilité de la situation et facilite les investigations. [4]

1.7.1.1 Avantages d'Autovu

- Localiser plus facilement les véhicules recherchés et appliquer les limites de stationnement.
- Il peut être utilisé pour des installations fixes et mobiles, le système AutoVu est parfaitement adapté à toute une série d'applications et d'entités, notamment les forces de l'ordre, les municipalités et les sociétés commerciales.
- Fournit des lectures de plaques d'immatriculation à haute précision.
- Assure des performances sans interruption, en matière de protection des données et de confidentialité.
- Unifié avec la vidéo surveillance ainsi que le contrôle d'accès de la plate-forme.

1.7.2 LAPI-ENGINE

Le produit LAPI-ENGINE est un OCR de lecture de plaques d'immatriculation permettant d'identifier et de tracer les véhicules. LAPI-ENGINE est un produit autonome, dédié Principalement à la traçabilité des véhicules, et pouvant s'adapter à un large éventail d'applications. [5]

1.7.2.1 Avantages de LAPI-ENGINE

- Interface simple à configurer et disponible sur Windows, Linux et Android.
- Lecture et reconnaissance de plaques d'immatriculation des pays de l'Union Européenne et d'Afrique.
- Système automatique 24h/24h.
- Gestions des formats vidéo AVI - JPEG 2000 - MPEG 4 - H264 - MPEG 2.
- Prise de contrôle à distance.
- Affichage en temps réel des évènements détectés.

1.7.3 SeeTec :

SeeTec est un module d'extension de SeeTec Cayuga qui permet la reconnaissance automatique de plaques d'immatriculation de véhicules à l'arrêt ou en mouvement. [6]

1.7.3.1 Avantages de SeeTec :

- Il peut lire des formats de plaques internationaux, même en caractères arabes et cyrilliques, sur jusqu'à huit voies de circulation par serveur.
- La reconnaissance des plaques d'immatriculation s'effectue en continu ou est pilotée par un déclencheur.
- Le module peut être utilisé avec chaque caméra IP prise en charge par SeeTec et intégrée au système, et cela Même dans des conditions de luminosité difficiles.

1.7.4 Asia Vision Technology Limited (AVT):

En tant que leader du marché de la technologie de l'intelligence artificielle (IA), Asia Vision Technology Limited (AVT) s'engage à fournir une gamme complète de solutions intelligentes pour le trafic intelligent, le stationnement intelligent et les ports intelligents dans le monde. Après plus de 20 ans d'innovation dans la technologie de reconnaissance optique de caractères, le système de reconnaissance de numéro de plaque et de numéro de conteneur lancé par Asia Vision est largement utilisé dans divers domaines avec une précision, une stabilité et une efficacité élevées. [7]

1.7.4.1 Avantages d'Asia vision Technologie Limited :

- Le système détecte, vérifie et reconnaît automatique la plaque d'immatriculation, à la fois alphanumérique et non alphanumérique des véhicules.
- Le système a réussi à reconnaître les plaques d'immatriculations de 20 pays dans plusieurs langues, y compris l'anglais, le chinois, le coréen, le japonais, le thaïlandais, l'espagnol et le portugais.

1.7.5 Technologie AGL (Application de Gestion Locale)

AGL (Application de Gestion LAPI) est une suite logicielle basée essentiellement sur la technologie LAPI. Le but d'AGL est de permettre l'automatisation de processus pour les forces de l'ordre françaises, notamment par l'acquisition de données au travers de capteurs dédiés. Ces

données peuvent ensuite être interprétées de différentes façons, allant de la notification simple d'information (en croisant avec des données internes), à une aide à la verbalisation électronique.[8]

1.7.5.1 Avantages d'AGL :

- Elle est composée de deux versions : la version server et la version cliente.
- La version server permet de rassembler, traiter, et extraire les données. Elle est composée de l'ensemble des capteurs (optiques, RFID, laser, GPS) qui collectent des données de voirie de manière automatique, semi-automatique ou manuelle.
- La version Capture(cliente), qui permet d'acquérir des données, et de les utiliser sur le terrain. Elle est relié d'une part aux services nationaux français, aux prestataires tiers, aux services locaux des collectivités et d'autre part aux systèmes AGL Capture.

1.7.6 Système LAPI-Pryncar :

De jour comme de nuit, le système LAPI enregistre les plaques d'immatriculation dans son champ de vision. L'immatriculation du véhicule peut ainsi être lue et comparée au fichier des véhicules enregistrés dans le cadre d'un contrôle d'accès ou d'une recherche de véhicule. [9]

1.7.6.1 Avantages de LPAI-Pryncar :

- Contrôle d'accès : Parkings privés ou publics, surveillance de sites, péages, stations-service, etc.
- Gestion de trafic, sécurité routière : Villes embouteillées, régulation du trafic, etc.
- Capteur LAPI Fixe autonome.
- Appli Pryncar pour smartphone.

1.7.7 Système LAPIA :

Le système LAPIA a été conçu spécialement pour la reconnaissance des plaques d'immatriculation algérienne. [23]

Il a été développé en 2017 par des étudiants algériens en master2.

Il est capable de prendre en charge tout type de plaque Algérienne.

1.7.7.1 Avantages du Système LAPIA :

- Une précision de plus de 80%.
- Capacité de prendre en charge tout type de plaques Algérienne.
- Possibilité d'être utilisé par les autorités compétant.

Nous avons trouvé beaucoup de systèmes, nous avons cité quelques-uns d'entre Eux et ses différents systèmes proposés sont basés sur des propriétés différentes. Des techniques utilisent des règles simples basées sur des méthodes existantes pour la détection et la reconnaissance des plaques d'immatriculation.

1.8 Conclusion :

Ce chapitre nous a permis de voir la notion de Lecture Automatique de Plaques Minéralogique et ces différents algorithmes qui doivent être réalisés pour que le logiciel puisse identifier une plaque d'immatriculation ainsi que les difficultés qui en résultent.

Le chapitre qui en suit détaillera les techniques de traitement d'images utilisées dans les systèmes LAPI.

Rapport-Gratuit.com

Chapitre II

Traitement d'images

2.1 Introduction

De nos jours nous assistons de plus en plus à un développement très important de la technologie. Le domaine du traitement d'image en fait partie.

Nous désignons par traitement d'images numériques l'ensemble des techniques permettant de modifier une image afin d'améliorer sa qualité ou d'en extraire des informations.

Dans ce chapitre, nous abordons les différentes notions de bases ainsi que les techniques nécessaires à la compréhension du traitement d'images.

Comme nous nous intéressons aux informations extraites de plaques d'immatriculation algériennes, nous commençons par donner quelques explications de base concernant les plaques d'immatriculation de véhicules en Algérie.

2.2 Caractéristiques des plaques d'immatriculation Algérienne

La plaque est composée de dix (10) chiffres (voir la figure2) dont chacun a une spécification bien déterminée :



Figure2 : exemple de plaque d'immatriculation Algérienne.

Nous commençons par la droite :

- 13 correspond au code de la wilaya.
- 12 correspond à l'année de fabrication du véhicule.

- 1 correspond au type du véhicule, ici 1 car c'est un véhicule touristique, 2 pour les camions, etc.
- 08453 signifie que ce véhicule est le 8453ème qui est entré dans la wilaya 13 (Tlemcen) en 2012.
- Les plaques sont de couleur blanche (à l'avant), jaune (à l'arrière) et la couleur des caractères est noire.
- Elles ont une forme rectangulaire, le style des caractères varie d'une plaque à l'autre et la distance entre les caractères n'est pas uniformisée elles varient selon les constructeurs des plaques.
- Récemment les gens aiment ajouter quelques illustrations sur la plaque d'immatriculation de leur véhicule, comme par exemple le drapeau Algérien, le nom de l'entreprise, le numéro de téléphone, etc.

2.3 Définition du traitement d'images

Le traitement d'images est une discipline de l'informatique et des mathématiques appliquées qui étudie les images numériques et leurs transformations, dans le but d'améliorer leur qualité ou d'en extraire de l'information.

Il s'agit d'un sous-ensemble du traitement du signal dédié aux images et aux données dérivées comme la vidéo par opposition au sous-ensemble consacrées aux sons et autres signaux. [10]

La compréhension du traitement d'images commence de ce qu'est une image. Le mode et les conditions d'acquisition et de numérisation des images traitées conditionnent largement les opérations qu'il faudra réaliser pour extraire de l'information. En effet de nombreux paramètres entrent en compte, les principaux étant :

- **La résolution d'acquisition et le mode de codage utilisé lors de la numérisation**, qui déterminent le degré de précision des éventuelles mesures de dimensions.
- **Les réglages optiques utilisés**, (dont la mise au point) qui détermine par exemple la netteté de l'image.

- **Les conditions d'éclairage**, qui déterminent une partie de la variabilité des images traitées ;
- **Le bruit de la chaîne de transmission d'image.**

Quelques exemples types d'informations qu'il est possible d'obtenir d'une image numérique:

- La luminance moyenne ;
- Le contraste moyen ;
- La couleur prédominante ;
- Le taux d'acuité moyen (précis ou flou) ;
- Le taux d'uniformité des couleurs ;
- La présence ou l'absence de certains objets.

2.4 Image

Une image est avant tout un signal 2D (x, y) Souvent, cette image représente une réalité 3D (x, y, z) , il existe deux points de vue sur la notions de l'image : [11]

- D'un point de vue humain : Une image contient plusieurs informations sémantiques il faut interpréter le contenu au-delà de la valeur des nombres.
- D'un point de vue mathématique : Une image est une matrice de nombres représentant un signal, plusieurs outils permettent de manipuler ce signal.

Il existe trois principaux types d'images :

- ✓ Images en niveaux de gris, dont la valeur appartient à l'ensemble suivant $\text{Image}(x, y) \in [0..255]$.
- ✓ Images binaires (uniquement en noire et blanc), dont la valeur ne peut accéder 0 ou 1 ce qui nous donne la formule suivante : $\text{Image}(x, y) \in \{0, 1\}$.
- ✓ Images couleurs qui fait un mélange des trois couleurs, $\text{ImageRed}(x,y)$ $\text{ImageGreen}(x,y)$ $\text{ImageBlue}(x, y)$.



Images en niveaux de gris

Images binaires

Images couleurs

Figure3 : Différents types d'images.

2.5 Représentation d'image

D'un point de vue informatique, l'image est avant tout un fichier numérique qui va reconstituer un rendu visuel au travers des calculs. A la base, il existe deux technologies bien distinctes pour produire ce rendu. [12]

- **Images matricielles**
- **Images vectorielles**

2.5.1 Image matricielle

Encore appelée image Bitmap, l'image matricielle est certainement celle la plus répandue et la plus couramment utilisée. Elle est tout simplement constituée d'une grille de milliers de Pixels représentant les points de couleur successifs de l'image.

Nous parlons de points lorsque ces images sont imprimées ou destinées à l'impression (photographies, publicités, cartes etc.) et nous parlons de pixels pour les images stockées sous forme « binaire » ou numérique.

Chaque Pixel est alors un tout petit carré ayant une place bien définie avec sa propre couleur et l'image devient alors un quadrillage de Pixels qui, placés les uns à côté des autres, restituent le rendu visuel.

2.5.2 Image vectorielle

L'image Vectorielle utilise également la technique du Pixel, mais cette fois, leur position et leur couleur ne sont pas figées puisqu'elles sont calculées dynamiquement par le logiciel.

Une image vectorielle est une image numérique composée d'objets géométriques individuels (segments de droite, polygones, arcs de cercle, etc.) définis chacun par divers attributs de forme, de position, de couleur, etc.

Par exemple pour afficher une ligne, le logiciel détermine le point de départ, le point d'arrivée puis la trajectoire à suivre. Ensuite, il calcule et positionne l'ensemble des pixels nécessaires pour afficher cette ligne. Il en va de même pour des formes et des couleurs plus complexes.

2.5.3 Comparaison entre image matricielle et image vectorielle

Afin de mieux comprendre les possibilités qu'offrent les deux modes de représentation d'images, jetons un coup d'œil sur le tableau suivant :

| Représentations caractéristiques | Image matricielle | Image vectorielle |
|---|-----------------------------|-------------------------------|
| Rendu de paysages | Adoptée | Non adoptée |
| Rendu de diagramme | Adoptés | Vouées |
| Taille | Volumineux | non volumineux |
| Réalisme | Possible | Impossible |
| images supporté | dessins, carte, schémas,... | cercle, segment, schémas, ... |
| Formats exprimé en | jpeg .jpg .png .gif | .svg .ai |

Tableau1 : les caractéristiques des images matricielles et des images vectorielles.

2.6 Principe d'acquisition d'image

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images.

Principe :

Des sources lumineuses éclairent une scène composée d'objets ; Chaque objet absorbe et renvoie cette énergie lumineuse ; enfin le capteur d'images transforme l'énergie lumineuse renvoyée dans sa direction en un signal électrique. [13]

Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de numérisation.

Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principale :

- ❖ **Acquisition à partir d'une caméra.**
- ❖ **Acquisition à partir d'un scanner.**

2.7 Caractéristiques d'une image

L'image est un ensemble structure d'informations caractérisée par les paramètres suivants : [10]

2.7.1 Taille

On appelle taille le nombre de pixels constituant l'image, c'est-à-dire sa « dimension informatique » (le nombre de colonnes de l'image que multiplie son nombre de lignes).

Une image numérique est composée d'une grille de pixels. Ces pixels sont autant de petits carrés porteurs d'une information de couleur élémentaire.

Une image possédant 640 pixels en largeur et 480 en hauteur aura une dimension de 640 pixels par 480, notée 640x480.

2.7.2 Résolution

La résolution, terme souvent confondu avec la "Dimension", détermine par contre le nombre de points ou pixels par unité de surface, exprimé en points par pouce (PPP, en anglais DPI pour Dots Per Inch) ; un pouce représentant 2.54 cm.

La résolution permet ainsi d'établir le rapport entre le nombre de pixels d'une image et la taille réelle de sa représentation sur un support physique (impression d'une photo). Une résolution de 300 dpi signifie donc 300 colonnes et 300 rangées de pixels sur un pouce carré ce qui donne donc 90000 pixels sur un pouce carré.

2.7.3 Bruit

Le bruit dans une image est l'ensemble des pixels qui parasitent la photo, c'est un phénomène qui consiste en la dégradation progressive de la qualité de l'image au fur et à mesure que l'on augmente la sensibilité du capteur, il est dû aux limites en faible lumière du capteur et de l'électronique de l'appareil photo et par conséquent entraîne une perte de la netteté dans les détails de l'image.

2.7.4 contraste

Le contraste est une propriété intrinsèque d'une image qui quantifie la différence de luminosité entre les parties claires et sombres d'une image, le contraste d'une photo est lié à la

notion de luminosité, instinctivement une image paraît contrastée lorsqu'elle mêle des zones très lumineuses et des zones très sombres.

2.7.5 Histogramme

L'histogramme de niveaux des gris d'une image est une fonction qui permet de représenter la distribution des couleurs de l'image, en indiquant pour chaque valeur entre le noir (0) et le blanc (255), combien il y a de pixels de cette valeur dans l'image; en abscisse (axe x) : le niveau de gris (de 0 à 255); en ordonnée (axe y) : le nombre de pixels.

Les pixels sombres apparaissent à gauche de l'histogramme, les pixels clairs à droite de l'histogramme et les pixels gris au centre de l'histogramme.

2.7.6 Luminance

La luminance encore appelée brillance est le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface pour un observateur lointain.

Une surface très lumineuse présente une forte luminance, tandis qu'une surface parfaitement noire aurait une luminance nulle.

2.7.7 Image en niveaux de gris

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

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

2.7.8 Image en couleur

Même s'il est parfois utile de pouvoir représenter des images en noir et blanc, les applications multimédias utilisent le plus souvent des images en couleurs. La représentation des couleurs s'effectue de la même manière que les images monochromes avec cependant quelques particularités. En effet, il faut tout d'abord choisir un modèle de représentation. Nous pouvons représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes

émettant de la lumière (écrans d'ordinateurs) sont basés sur le principe de la synthèse additive: les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B).

2.8 Les principales étapes de traitement d'image

En réalité, il n'existe pas de méthode de traitement d'images générale pour tous les domaines d'application possible. Il faut appliquer à chaque domaine des algorithmes spécifiques. Ces derniers utilisent le plus souvent des techniques classiques à savoir : la segmentation, la classification, la reconnaissance des frontières, la reconnaissance des caractères, etc.

De manière schématique toute méthode de traitement d'images comprend ces 3 étapes majeures : [14]

- **Prétraitements des images.**
- **Amélioration des images.**
- **Analyse des images.**

2.8.1 Prétraitement des images

Cette phase a lieu juste après l'acquisition des images et a pour objectif d'améliorer la qualité de l'image en vue de son analyse. Son but est d'obtenir une image qui soit la plus proche possible de l'image idéale qui aurait dû être obtenue si le système d'acquisition était parfait. Ainsi, ils peuvent par exemple corriger : [23]

- Les défauts radiométriques du capteur : non linéarité des détecteurs, diffraction de l'optique, etc.
- Les défauts géométriques de l'image dus au mode d'échantillonnage spatial, à l'obliquité de la direction de visée, au déplacement de la cible, etc.
- Le filtrage ou réduction de fréquences parasites, par exemple dus à des vibrations du capteur.
- Les dégradations de l'image dues à la présence de matière entre le capteur et le milieu observé.

2.8.2 Amélioration d'images

L'amélioration d'images consiste à modifier les caractéristiques visuelles de l'image (contraste, histogramme, etc.) pour faciliter son interprétation par l'œil humain. Pour cela elle élimine ou réduit le bruit de l'image. Les principales techniques sont :

- L'amélioration de contraste.
- Modification de l'histogramme.

- Le filtrage linéaire (lissage, mise en évidence des frontières avec l'opérateur "Image - Image lissée", etc.) et transformée de Fourier pour faire apparaître / disparaître certaines fréquences dans l'image.
- Filtrage non linéaire (filtres médians, etc.) pour éliminer le bruit sans trop affecter les frontières, etc.

2.8.3 Analyse d'images

Le but de l'analyse d'images est d'extraire et de mesurer certaines caractéristiques de l'image traitée en vue de son interprétation. Ces caractéristiques peuvent être des données statistiques sur des comptes numériques (moyenne, histogramme, etc.), ou sur des données dérivées (ex. dimensions, ou orientation d'objets présents dans l'image). En général, le type d'information recherché dépend du niveau de connaissance requis pour interpréter l'image. Les applications dans le domaine du guidage et de la télédétection nécessitent souvent des connaissances différentes et de plus haut niveau (ex., cartes 3-D) que dans les domaines du médical, de la géologie, du contrôle de qualité, etc. Ainsi, un robot en déplacement ne nécessite pas le même type d'information qu'un système utilisé pour détecter la présence de matériaux défectueux. Pour ce dernier, il faut prendre uniquement la décision "non défectueux ou défectueux". Cette décision peut être prise à partir d'un "raisonnement" plus ou moins complexe (exemple : détection de la présence de raies d'absorption caractéristiques d'une impureté). Pour le robot, il faut simuler le processus décisionnel d'un individu en déplacement, ce qui nécessite au préalable de reconstituer une carte du lieu de déplacement en temps réel, avec les obstacles à éviter.

2.9 Type de données manipulées :

Le traiteur d'image dispose principalement d'images numériques, donc échantillonnées. Il dispose également de données intermédiaires de diverses natures : cartes de régions, listes de points connexes, tableaux de valeurs mesurées, etc.

En ce qui concerne les images proprement dites, la représentation la plus utilisée est celle d'un tableau à 2 dimensions composé d'un ensemble de lignes et de colonnes.

Chaque cellule du tableau, appelée pixel, contient une valeur quantifiée. Cette valeur est une sémantique dépendant du type de signal qu'elle code (intensité lumineuse du point, distance à un point de référence, ou numéro de la région d'appartenance par exemple). Dans le cas des images 3D d'IRM, la représentation n'est plus un tableau à 2 dimensions mais un tableau à 3 dimensions. [15]

2.10 Les opérateurs de traitement d'image :

Le Par analogie avec les opérateurs mathématiques, on appelle opérateurs de traitement d'images des traitements plus ou moins complexes prenant en entrée une image ou un ensemble d'informations relatif à une image, et produisant une image ou un ensemble d'informations relatif aux données initiales.

On classe généralement les opérateurs en différentes familles, en fonction des Informations qu'ils acceptent en entrée et qu'ils fournissent en sortie, et en fonction des transformations qu'ils font subir aux données. Ainsi, par exemple, on distingue (cette liste est loin d'être exhaustive) : [15]

➤ **Opérateurs image → image :**

- **Opérateurs de modifications pixel à pixel** (aussi appelés opérateurs point à point) : changement de la dynamique de l'image, opérateurs binaires pixel à pixel (et, ou, xor, etc.) ;
- **Opérateurs locaux** (traitent les pixels en fonction de leur voisinage) : opérateurs de flou, opérateurs morphologiques (érosion, dilatation, squelette), opérateurs de détection de contours ;
- **Opérateurs dans l'espace fréquentiel** : opérateurs de réduction du bruit, filtres passe-bande (souvent utilisés en première approche pour améliorer l'image, on les appelle alors des opérateurs de prétraitement) ;
- **Opérateurs globaux** : calcul des distances ;
- **Opérateurs de croissance de régions** : ligne de partage des eaux.

➤ **Opérateurs image → ensemble d'informations :**

- Opérateurs de segmentation en frontières, en régions ;
- Opérateurs de classification de pixels ;
- Opérateurs de calcul de paramètres.

➤ **Opérateurs ensemble d'informations → image :**

- Constructeurs d'image à partir d'une carte de régions ou d'une liste de frontières.

Les parties suivantes s'attachent à détailler les différents opérateurs et leurs Applications habituelles, puis à présenter la manière dont ils sont combinés pour construire une application de traitement d'images.

2.10.1 Les opérateurs locaux

Il faut alors utiliser des opérateurs de traitement plus complexes scindés bien souvent en deux sous-catégories : [15]

- **Les opérateurs linéaires ;**
- **Les opérateurs non linéaires.**

La première sous-catégorie comprend tous les opérateurs pouvant exprimer leur résultat comme une combinaison linéaire des niveaux de gris d'un voisinage de l'image. Ces filtres possèdent des caractéristiques spectrales, on parle ainsi de filtre passe-bas (l'image devient floue) ou de filtre passe-haut (les contours ressortent).

La deuxième sous-catégorie comprend le domaine de la morphologie mathématique, ainsi que d'autres traitements comme les détecteurs de points caractéristiques, l'opérateur de Di-Zenzo (détecteur de contour généralisé au cas couleur), le filtre Retinex, ainsi que les opérateurs homomorphies (ceux qui travaillent sur le logarithme de l'image), mais aussi tous les opérateurs permettant d'extraire par exemple des informations sur la texture de l'image (matrice de concurrence, indice fractal, longueur de plage...).

On a souvent l'habitude de voir un détecteur de contours s'appliquer après un filtre linéaire passe-bas qui rend l'image floue. La plupart du temps il faut combiner astucieusement filtre non linéaire et filtre linéaire afin de détecter ce que l'on souhaite tout en faisant abstraction du bruit.

Une fois le bruit éliminé et l'image restaurée afin de compenser les déformations introduites par le milieu de transmission et l'optique d'acquisition, on peut passer à l'étape de segmentation qui doit permettre de réaliser une partition de l'image en ensembles connexes homogènes.

Il existe deux grandes catégories de segmentations :

- La segmentation de région ;
- La segmentation orientée contour — on se trouve alors confronté à un problème de représentation du résultat par des primitives simples.

La segmentation orientée contour connaît de nombreux progrès autour de l'utilisation de contours actifs ou des ensembles de niveaux. L'introduction d'aspects probabilistes (chaînes de Markov et champs de Markov) a permis de travailler en réduisant la connaissance a priori nécessaire pour obtenir un traitement satisfaisant.

Dans cette étape on retrouve souvent une partie de classification des pixels en classes. On essaye de regrouper au sein d'un même ensemble, aussi appelé classe, les pixels présentant une même caractéristique : niveau de gris compris dans un certain intervalle ou dérivée seconde supérieure à un certain seuil.

2.10.2 Filtres linéaires

Un filtre linéaire transforme un ensemble de données d'entrée en un ensemble de données de sortie selon une opération mathématique appelée convolution. Lorsqu'il s'agit de données numérisées comme dans le cas du traitement d'image, la relation entre les valeurs des pixels de sortie et celle des pixels d'entrée est décrite par un tableau de nombres, généralement carré, appelé matrice de convolution. Le temps de calcul est souvent réduit lorsqu'on veut séparer un filtre en deux filtres dont la convolution mutuelle permet de le reconstituer. Cette remarque est utilisée en particulier pour créer un filtre à deux dimensions à partir de deux filtres à une seule dimension (vecteurs) dans le sens horizontal et le sens vertical. [15]

2.10.3 Lisage

Ceux-ci sont des filtres passe-bas qui coupent plus ou moins les plus hautes fréquences. Ils sont utilisés pour atténuer les bruits d'origines les plus diverses qui polluent l'information, en particulier dans la détection de contours considérée ci-après.

Techniquement, il s'agit de traductions discrètes de filtres continus qui, comme ceux-ci, ne modifient pas le niveau global du signal. Les termes de la matrice de convolution sont donc généralement des entiers à diviser par leur somme.

- **Filtre uniforme** : il est obtenu par convolution de deux filtres unidimensionnels rectangulaires. Toutes les composantes de la matrice ont la même valeur. L'imperfection de ce filtre réside dans le fait qu'il introduit des déphasages.
- **Filtre pyramidal** : la convolution d'un filtre rectangulaire avec lui-même conduit à un filtre triangulaire grâce auquel les phases ne sont plus modifiées. Le filtre pyramidal est obtenu à partir de filtres triangulaires dans les deux directions.
- **Filtre gaussien** : ce filtre très populaire utilise la loi de probabilité de Gauss (voir Loi normale multidimensionnelle). Des approximations de plus en plus précises peuvent être obtenues, selon le théorème central limite par itération de l'un des filtres précédents.

Le filtre gaussien est utilisé comme constituant du masque flou qui améliore la netteté apparente des photographies numériques. Bien qu'il soit popularisé par la photographie plus ou moins artistique, il est également utilisé dans certaines techniques, comme l'astronomie.

2.10.4 Détection de contours

Ces filtres transforment l'image d'entrée en une image noire sauf aux points où un contour est détecté qui est marqué en blanc. Les valeurs absolues importent peu, il est sans intérêt de changer d'échelle comme pour un lissage.

La détection est basée sur la dérivation selon les deux coordonnées. Si on considère classiquement les signaux comme des sommes de sinusoides, la dérivation apparaît comme un filtre passe-haut qui introduit donc du bruit à l'origine de faux contours. Pour l'amateur il est recommandé, avant d'utiliser un filtre simple, d'atténuer ce bruit par passage dans un filtre flou. Des méthodes plus élaborées ont été systématisées pour les professionnels. [15]

- **Filtre dérivées premières** : Le filtre le plus simple consiste à calculer les différences entre pixels voisins sur les horizontales puis sur les verticales. Chaque extremum correspond à un point d'un contour.
- **Filtre de Prewitt** : le filtre de Prewitt introduit un flou, chacune des deux matrices étant le produit du filtre dérivation dans la direction considérée par un filtre de flou rectangulaire selon l'autre direction.
- **Filtre de Sobel** : la technique précédente est améliorée en remplaçant le filtre rectangulaire par un filtre triangulaire.
- **Filtre de Canny** : c'est un filtre de Sobel précédé par un lissage gaussien et suivi par un seuillage. Ce filtre est conçu pour être optimal, au sens de trois critères.
- **Filtre de Deriche** : variante du filtre de Canny tout aussi efficace.
- **Filtre dérivées secondes** : celles-ci se calculent simplement en différences finies et c'est maintenant un changement de signe qui correspond à un point d'un contour. On les utilise généralement à travers leur somme qui est le laplacien.
- **Filtre de Marr-Hildreth** : le calcul du laplacien est précédé par un lissage gaussien avec deux variances ajustables pour filtrer les hautes fréquences.

2.10.5 Opérateurs morphologiques

La morphologie mathématique offre des opérateurs non linéaires particulièrement utiles pour filtrer, segmenter et quantifier des images. Initialement destinée au traitement des

images binaires, elle a très vite été généralisée aux images à niveaux de gris, puis aux images en couleurs et multi-spectrales.

La nature des opérateurs morphologiques fait qu'ils se prêtent bien au développement de circuits électroniques spécialisés (ou bien à l'utilisation de FPGA) dans les opérateurs morphologiques.

2.10.6 Construction d'une application traitement d'image

Les objectifs des applications peuvent être de différentes natures :

- Détecter la présence d'un objet ou son absence ;
- Calculer les caractéristiques d'un ou de plusieurs éléments de l'image.

Dans tous les cas, l'idée est, en partant d'une image initiale, d'en extraire des informations. Pour cela, on va utiliser les opérateurs à la manière de « briques logicielles », en les combinant et en les enchaînant. Ces techniques sont la base des systèmes de vision industrielle.

2.11 Quelques exemples concrets de traitement d'images

Parmi les exemples de traitement d'images existants on peut citer :

- **Contrôle de présence/absence** Sur des chaînes de production, on vérifie en bout de chaîne avec une caméra vidéo la présence d'une pièce dans un ensemble plus complexe. Pour cela bien souvent il suffit de faire un simple seuillage dans une région spécifique.
- **Construction et correction de cartes géographiques d'après des images satellites ou des images aériennes.** On recale d'après des informations topographiques les images reçues, puis on les met sur la carte en correspondance avec les informations trouvées dans l'image : voies de communication, voies et plans d'eau, parcelles agricoles...
- **Reconnaissance de l'écriture.** La reconnaissance de l'écriture manuscrite progresse de jour en jour. Elle est suffisamment opérationnelle pour que la majorité des adresses, même manuscrites, soient reconnues automatiquement sur le courrier postal.
- **Recherche d'image par le contenu.** L'objectif de cette technique est de rechercher, parmi une base de données d'images, les images similaires à une image exemple, ou ayant certaines caractéristiques, par exemple rechercher toutes les images comportant un vélo.

- **Analyse de la vidéo.** L'objectif de cette technique devenue une discipline depuis les années 2000 (lorsque la puissance des processeurs peu onéreux et en particulier des PC a permis des traitements puissants en temps réel) est d'interpréter les faits observés à l'image afin de signaler ou d'enregistrer des faits marquants. Le plus souvent, la caméra est fixe et observe les mouvements d'une scène. Les applications sont nombreuses : protection des biens (détection d'intrusion, détection d'objet abandonné ou déposé...), identification (biométrie faciale), sécurité des personnes (détection de chutes de personnes, franchissement de rambardes...), animations (planchers animés selon les mouvements des danseurs en boîte de nuit), détection de feux (industriel, forêts, tunnels...), surveillance de tunnels (comptage, mesure de vitesse, détection de fuites/anomalies dans les plafonds), surveillance de tuyaux et autres processus industriels...

2.12 Conclusion

Dans ce chapitre nous avons vu la notion de l'image ainsi que ses différentes caractéristiques, puis les opérateurs de traitement d'image pour finir avec quelques exemples concrets du domaine de traitement d'image.

Le chapitre suivant fera l'objet d'un aperçu sur les différents outils utilisés dans la réalisation de notre travail.

Chapitre III

*Outils de développement du système
APR*

3.1 Introduction

Dans ce chapitre, nous allons présenter les outils de développement ainsi que les techniques et outils de reconnaissance optique des caractères, qui ont été utilisés dans la réalisation de notre système de détection de plaques d'immatriculation.

3.2 Les Outils de développement

Parmi tant d'outils de développement, notre choix c'est porté sur les outils suivants pour la réalisation de notre projet.

3.2.1 Le langage Java

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais **Java**. [16]

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que Unix, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et Framework associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

Comme nous avons utilisé la bibliothèque **javafx** dans notre application voyons alors quelques notions sur ce dernier.

3.2.2 Javafx

Javafx est une technologie créée par Sun Microsystems qui appartient désormais à Oracle, à la suite du rachat de Sun Microsystems par Oracle le 20 avril 2009.

Avec l'apparition de Java 8 en mars 2014, Javafx devient la bibliothèque de création d'interface graphique officielle du langage Java, pour toutes les sortes d'application (applications mobiles, applications sur poste de travail, applications Web), le développement de son prédécesseur **Swing** étant abandonné (sauf pour les corrections de bogues). [17]

Javafx est désormais une pure API Java (le langage de script spécifique qui a été un temps associé à Javafx est maintenant abandonné).

Javafx contient des outils très divers, notamment pour les médias audio et vidéo, le graphisme 2D et 3D, la programmation Web, la programmation multi-fils etc.

Le SDK de Javafx étant désormais intégré au JDK standard Java SE, il n'y a pas besoin de réaliser d'installation spécifique pour Javafx.

3.2.2.1 Fonctionnement de javafx

Des projets libres complètent Javafx en fournissant des composants absents de Javafx. Javafx utilise la métaphore d'un théâtre pour modéliser l'application graphique. Une étape (définie par la classe `javafx.stage.Stage`) représente le conteneur de niveau supérieur (fenêtre). Les contrôles (ou composants) individuels sont contenus dans une scène (définie par la classe `javafx.scene.Scene`).

Une application peut avoir plusieurs scènes, mais une seule des scènes peut être affichée sur la scène à un moment donné. Le contenu d'une scène est représenté dans un graphe de scène hiérarchique de noeuds (défini par `javafx.scene.Node`). (Figure4)

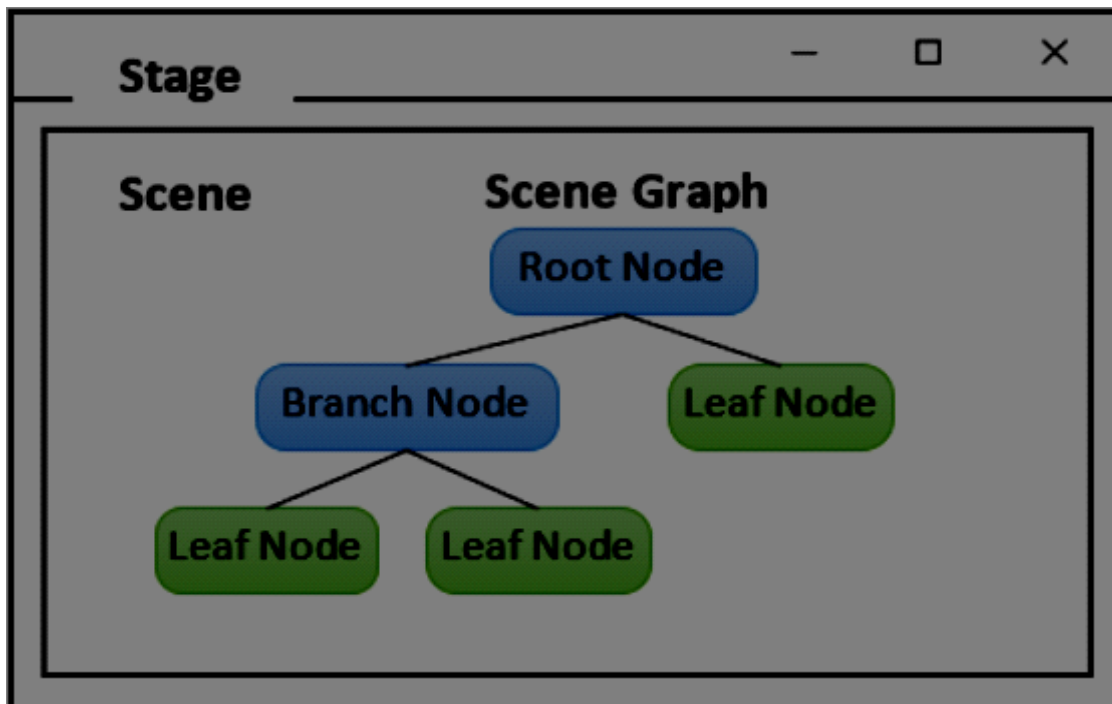


Figure4 : Graphe de la hiérarchie utilisée en javafx

3.2.2.2 Avantages de javafx

L'avantage de javafx nous a permis de séparer le design de la logique. Nous pouvons faire le design en créant des fichiers FXML.

Le FXML est un format de fichier propre à Javafx et servant, entre autres, à définir des interfaces graphiques. Ce format utilise tout simplement la syntaxe XML.

3.2.3 OpenCV

OpenCV (**O**pen **S**ource **C**omputer **V**ision) est la librairie de vision par ordinateur open source proposant plus de 500 algorithmes optimisés pour la vision par ordinateur. Elle est distribuée avec une licence BSD (libre), ce qui signifie que vous pouvez faire une publicité de votre application sans révéler votre code source. [18]

OpenCV possède des interfaces C++, C, Python et Java, et il prend en charge Windows, Linux, MacOS, iOS et Android. OpenCV a été conçu pour l'efficacité de calcul et avec un fort accent sur les applications en temps réel.

Afin de mieux vous présenter son étendue et ce qu'elle permet de faire, jetons un Coup d'œil aux principaux modules accessibles (**voir figure5**) :

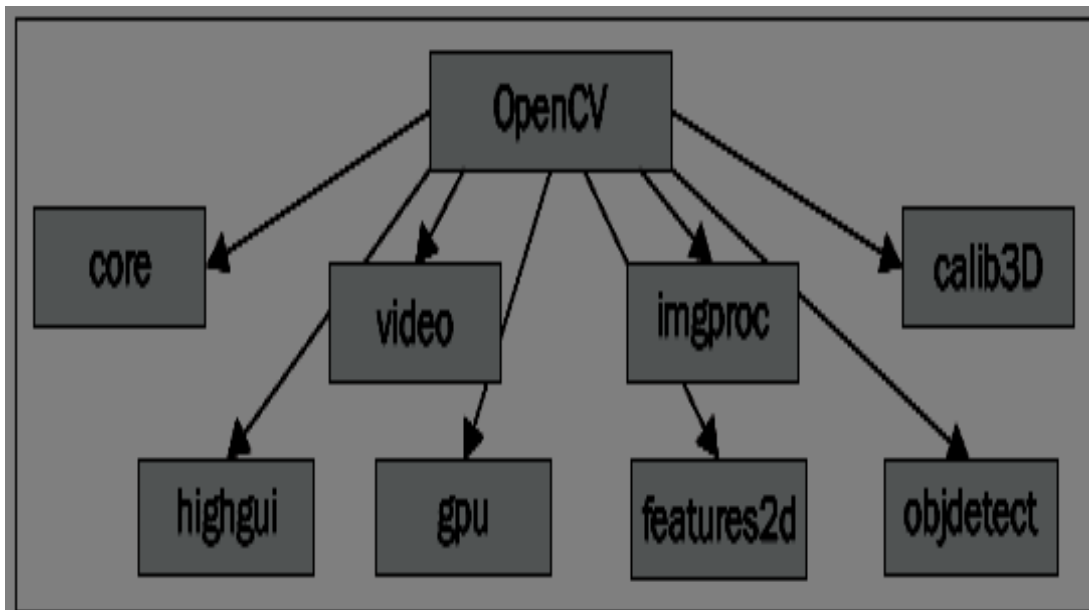


Figure5 : Schéma des différents modules de la bibliothèque OpenCV [18]

- **Core** : fonctionnalités de base.

C'est un module compact définissant les structures de données de base, y compris les matrices multidimensionnelles et les fonctions de base utilisées par tous les autres modules.

- **Imgproc** : traitement d'image.

C'est le module le plus intéressant dans le vif de notre sujet, qui comprend des fonctions linéaires et non linéaires à savoir : filtrage d'image, transformations d'images géométriques (redimensionnement, déformation affine et perspective), détections de contours, conversion de l'espace colorimétrique, histogrammes, etc.

- **Vidéo** : analyse et traitement de flux vidéo.

Les fonctions essentielles de ce module sont : l'estimation du mouvement, soustraction d'arrière-plan et algorithmes de suivi d'objets dans un flux vidéo.

- **Calib3D** : calibration, estimation de pose et stéréovision.

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

Algorithmes de géométrie de base à vues multiples, simples et calibration de caméra stéréo, estimation de pose d'objet, stéréo algorithmes de correspondance et éléments de reconstruction 3D.

- **Features2D** : Descripteurs.

Ce module concerne principalement l'extraction descripteurs pour caractériser une image.

- **Objdetect** : Détection d'objets.

Cette bibliothèque permet de faire de la reconnaissance d'objets dans une image au moyen de l'algorithme **Adaboost** et d'instances des classes prédéfinies pour exemple, les visages, les yeux, les tasses, les gens, les voitures, et ainsi de suite.

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

OpenCV intègre sa propre bibliothèque facile à utiliser pour ouvrir, enregistrer et afficher des images et des flux vidéo ainsi que de simples capacités pour réaliser des interfaces utilisateur, mais largement suffisant pour tester notre programme.

- **Gpu** : Algorithmes accélérés par GPU à partir de différents modules OpenCV.

3.2.4 Scene builder

Scene Builder est un outil de présentation visuelle qui permet aux utilisateurs de concevoir rapidement des interfaces utilisateur d'application Javafx, sans codage.

Les utilisateurs peuvent glisser et déposer des composants de l'interface utilisateur dans une zone de travail, modifier leurs propriétés, appliquer des feuilles de style et le code FXML de la mise en page qu'ils créent est automatiquement généré en arrière-plan. Le résultat est un fichier FXML qui peut ensuite être combiné avec un projet Java en liant l'interface utilisateur à la logique de l'application. [19]

3.3 Les techniques et outils de reconnaissance optique des caractères

C'est un ensemble de techniques informatiques qui réalisent une reconnaissance de la forme des caractères alphanumériques, sur des images prises par caméra, par procédé optique pour transmettre un résultat le plus souvent sous forme de fichier.

Dans notre travail nous avons opté pour le choix de « **Tess4J** » pour l'importance de son taux de reconnaissance.

3.3.1 Tess4j

Tess4j est Un wrapper Java JNA pour l'API OCR Tesseract.

Tess4J est publié et distribué sous la licence Apache, **v2.0** et est également disponible auprès de Maven Central Repository.

La bibliothèque est en cours de développement mais testé sur Windows et l'Unix. [20]

Elle est également utilisée pour des projets sous Android mais seulement a 1%.

3.3.1.1 Caractéristiques de Tess4j

La bibliothèque fournit un support de reconnaissance optique de caractères (OCR) pour :

- ❖ Formats d'image TIFF, JPEG, GIF, PNG et BMP.
- ❖ Images TIFF multipages.
- ❖ Format de document PDF.

3.4 Conclusion

Dans ce chapitre Nous avons présenté les notions de base ainsi que quelques caractéristiques concernant les outils de développements et technologiques utilisés pour réaliser notre application. Ce chapitre nous a également permis de nous familiariser avec ces outils.

Le prochain chapitre détaillera les étapes de l'implémentation de notre Application.

Chapitre IV

Implémentation du système APR

4.1 Introduction

Dans le chapitre précédent nous avons parlé des différents outils de développement utilisés dans la réalisation de notre application. Dans ce chapitre, nous expliquerons le fonctionnement de notre système à travers son interface.

4.2 Architecture du système APR

Pour qu'un système soit évolutif, extensible et maintenable, une bonne architecture doit être choisie tout au long de sa réalisation.

Notre système est basé sur l'architecture MVC (modèle/vue/contrôleur).

4.2.1 L'architecture MVC

L'architecture MVC est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans le système. [21]

Les différentes interactions entre le modèle, la vue et le contrôleur sont résumées par le schéma de la figure suivante :

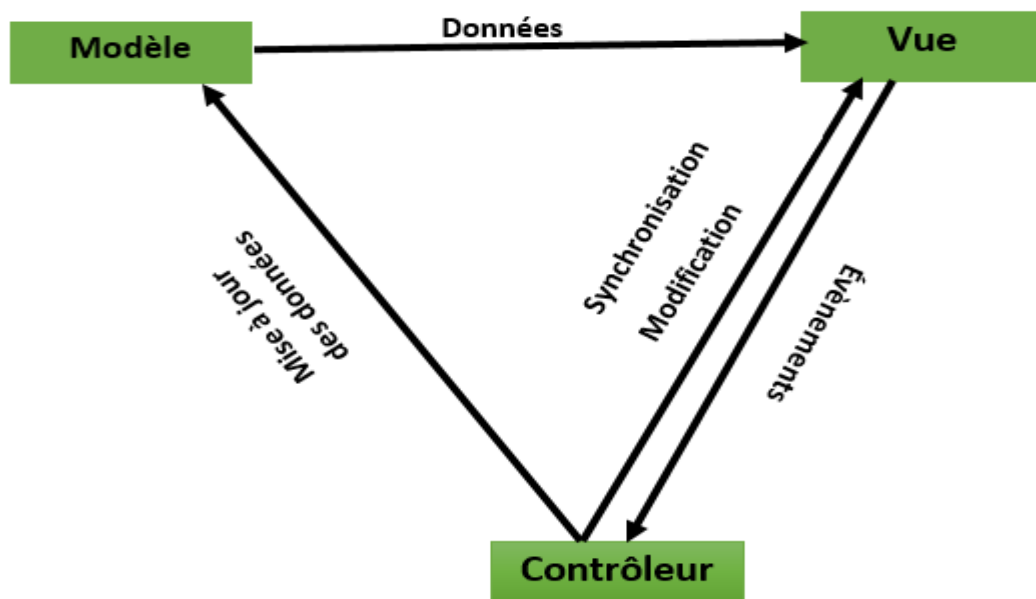


Figure 6 : Interactions entre le modèle, la vue et le contrôleur.

- **Modèle** : il contient les données et par conséquent assure leur intégrité, il est responsable

de la mise à jour des données (insertion, suppression, changement de valeur).

- **Vue** : fait l'interface avec l'utilisateur, elle permet d'afficher les données, elle permet également de recevoir toutes les actions de l'utilisateur (entrées, sortie, clic...) ses différents événements sont par la suite envoyés au contrôleur.
- **Contrôleur** : son rôle est d'assurer la synchronisation du modèle et de la vue, de ce fait il reçoit tous les événements de l'utilisateur et assure le déclenchement des actions à effectuer.

Cette architecture peut être vue à travers le diagramme de packages suivant :

4.2.2 Diagramme de packages du système APR

Dans le cas de notre système, l'architecture MVC citée dans la section précédente est représentée par le diagramme de packages du système (voir figure7).

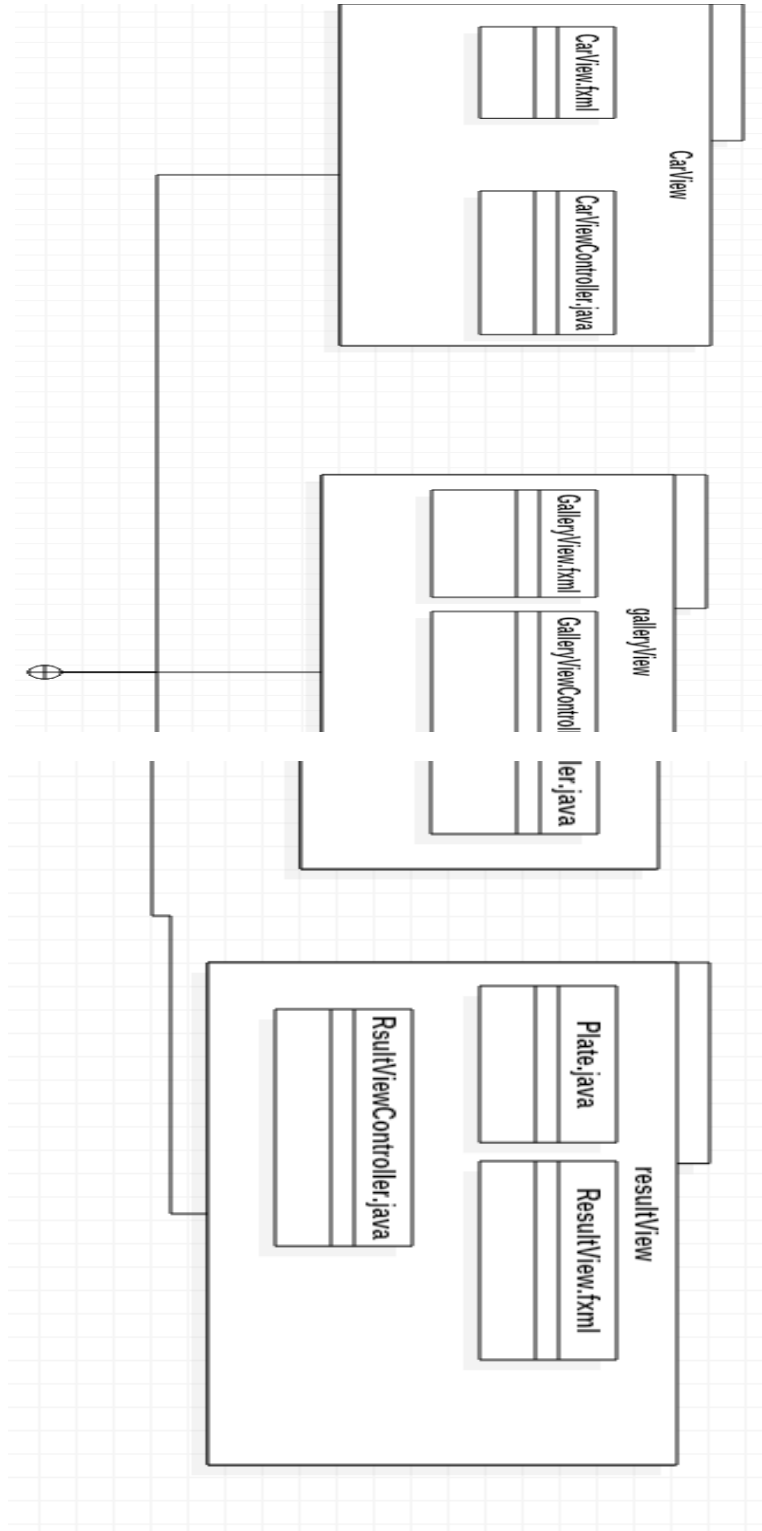


Figure7 : Diagramme de package du système APR.

4.3 Les interfaces de notre application

Enfin de mieux comprendre le fonctionnement de l'application, les figures qui suivent décrivent la chronologie des principales fonctionnalités de notre système ainsi que leur utilité.

4.3.1 L'interface principale de l'application

Au lancement de notre application, une fenêtre composée des trois parties suivantes sera affichée :

- GalleryView : qui sert à charger les images.
- CarView : qui sert à choisir le mode de localisation de la plaque (manuelle ou automatique).
- ResultView : qui sert à afficher les caractères contenant sur la plaque ainsi que le pourcentage de confidentialité.

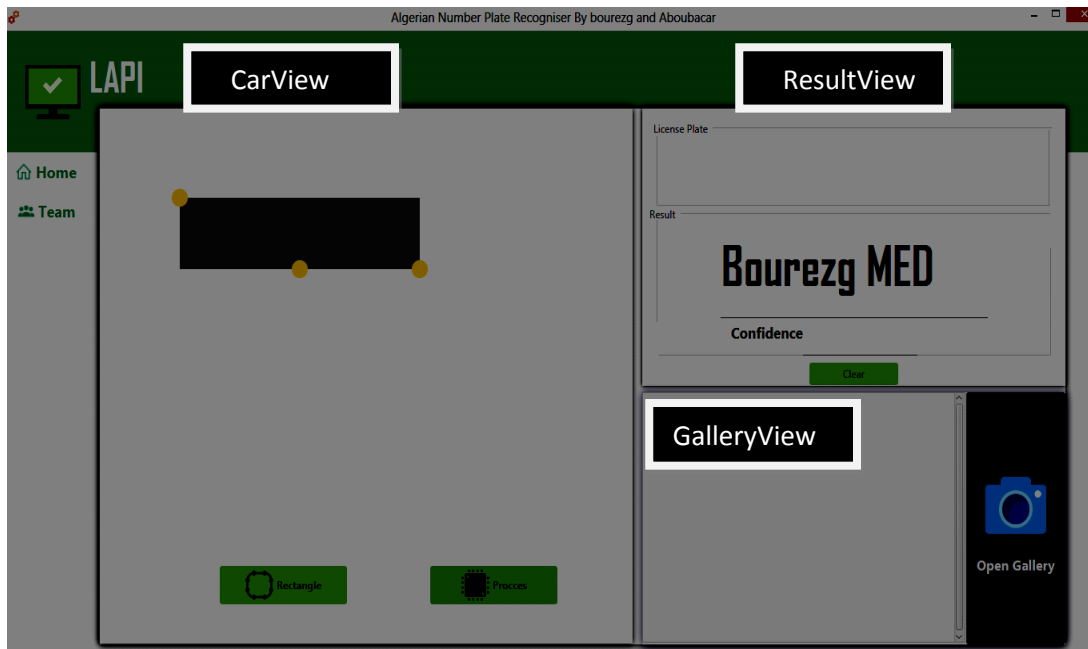


Figure8 : interface de traitement de l'application APR

4.4 Fonctionnement du système APR

L'objectif de notre système APR est la lecture des caractères affichés d'une plaque d'immatriculation contenu dans une image.

Afin d'obtenir le résultat escompté, nous avons réalisé une application qui passe par 3 phases, à savoir :

- L'acquisition de l'image contenant la plaque et prétraitement de l'image.
- Détection des contours.
- Reconnaissance des caractères de la plaque (OCR).

Nous allons voir plus de précisions et de détails, concernant chaque phase ainsi que les opérations, qui nous ont permis la réalisation de cette application.

4.4.1 La phase d'acquisition et de prétraitement de l'image

Grâce à cette phase, nous pouvons charger l'image, corriger le bruit et transformer l'image en niveaux de gris. Quand on clique sur l'icône « **Open Gallery** », on aura accès à la fenêtre suivante :

4.4.1.1 Interface pour chargement de l'image

Après choix de l'image dans la base photo on aura la fenêtre suivante :

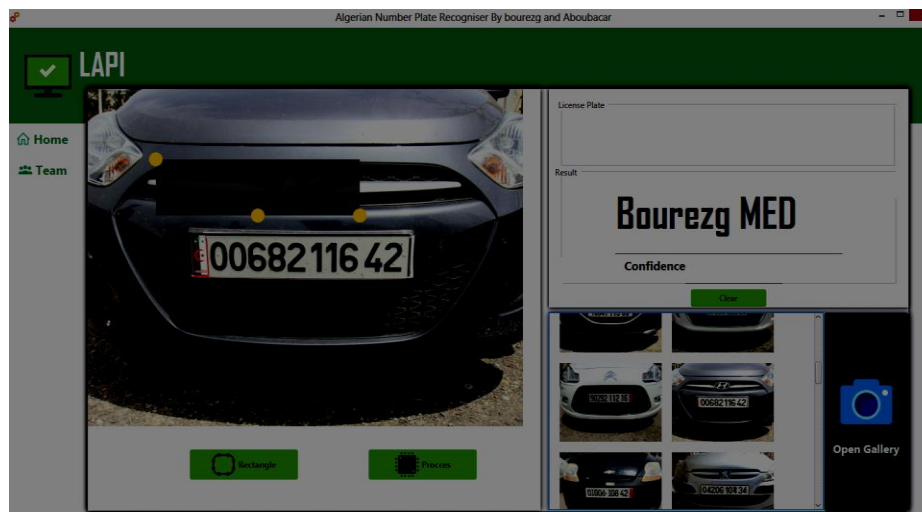


Figure9 : Interface pour chargement de l'image

4.4.1.2 Interface pour localisation de la plaque d'immatriculation manuellement

Pour localiser la plaque d'immatriculation manuellement, nous avons créé une forme rectangulaire qui est liée à 3 cercles en haut à gauche et en bas à droite peut redimensionner le rectangle et le cercle en bas peut être utilisé pour faire glisser le rectangle.

Une fois le rectangle positionné on clique sur l'icône «rectangle» on aura accès à la fenêtre suivante:

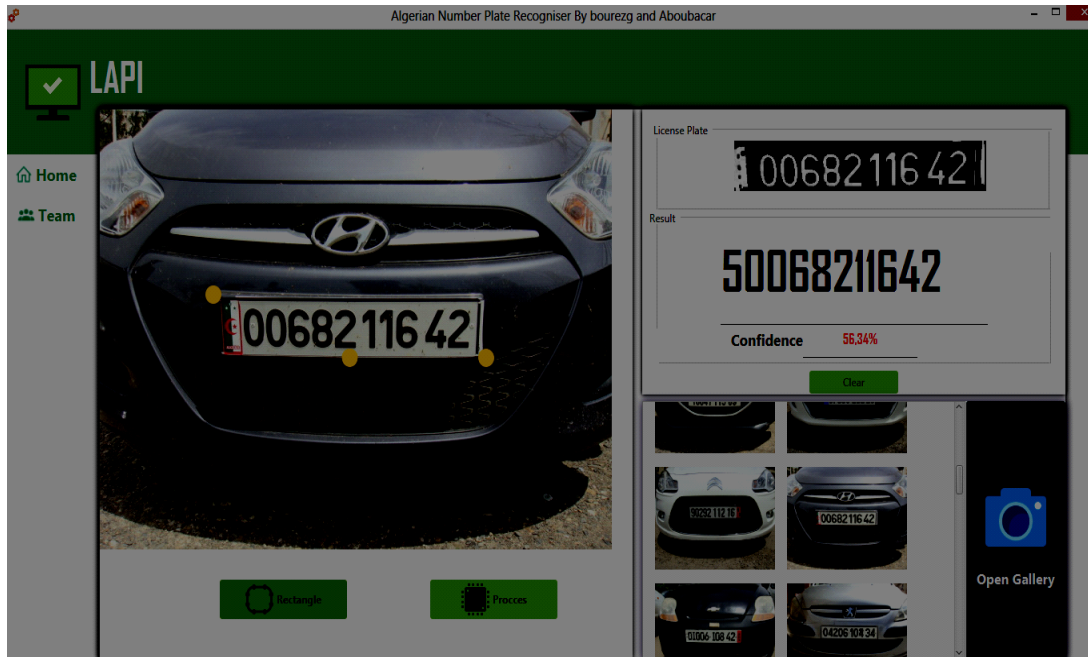


Figure10 : Lecture de la plaque en utilisant la localisation manuelle.

Dans ce résultat tout dépend de la manière dont le rectangle a été positionné. Un mauvais positionnement du rectangle peut donner un faux résultat.

Nous avons songé à améliorer ce résultat grâce à une localisation automatique de la plaque d'immatriculation, que nous allons voir dans la suite.

4.4.1.3 Conversion de l'image en niveaux de gris

Après avoir chargé l'image nous passons à un prétraitement de l'image qui consiste à convertir l'image couleur en image avec des niveaux de gris. Cela nous permettra de réduire les nuances et préparer un passage plus sûr vers une image composée que du blanc et du noir.



Figure 11 : Fenêtre de Conversion de l'image en niveaux de gris.

4.4.1.4 Suppression ou diminution du bruit de l'image

Une fois l'image convertie en niveaux de gris nous passons à la correction du bruit par ajout du flou gaussien pour réduire les détails.

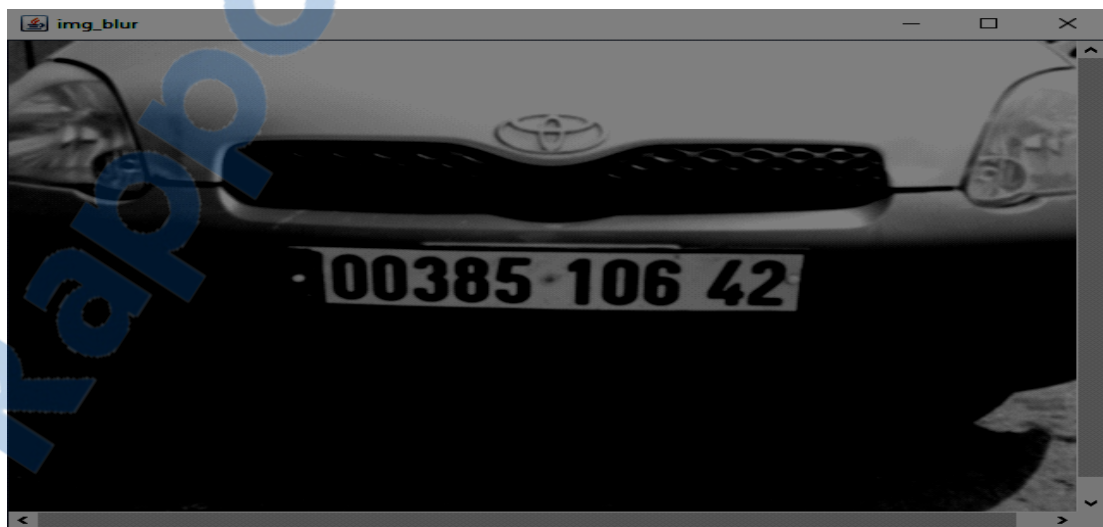


Figure 12 : Fenêtre correction du bruit de l'image.

La diminution du bruit ou sa suppression est le fait de traiter le côté TSL (Teinte, Saturation, Lumière) de l'image, ce qui va permettre une meilleure détection des couleurs. C'est un facteur primordial pour les prochaines étapes car si nous n'appliquons pas une méthode d'élimination du bruit, nous pouvons obtenir beaucoup d'arêtes verticales et horizontales, ce qui mènera à l'échec de la détection.

4.4.2 Phase pour la détection des contours

Dans cette étape, nous devons détecter toutes les plaques dans un cadre de caméra actuel.

La principale tâche utilisée dans la détection des contours est la **segmentation** dont nous allons voir en détails les différentes opérations.

4.4.2.1 La segmentation

La segmentation est le processus de division d'une image en plusieurs segments. Ce processus consiste à simplifier l'image pour l'analyse et faciliter l'extraction des caractéristiques. [23]

Dans cette première étape (segmentation), nous appliquerons différents filtres, opérations morphologiques, pour récupérer des parties de l'image qui pourraient avoir une plaque.

4.4.2.1.1 Détection des bords verticaux

Une caractéristique importante de la segmentation des plaques est le nombre élevé de bords verticaux dans une plaque d'immatriculation, en supposant que l'image a été prise frontalement et la plaque n'est pas tournée et sans distorsion en perspective.

Cette fonctionnalité peut être exploitée lors de la première étape de segmentation pour éliminer les régions qui n'ont pas de bords verticaux. Pour faire cela nous allons appliquer un filtre « sobel », ainsi on obtiendra le résultat suivant comme le montre la **figure suivante**.

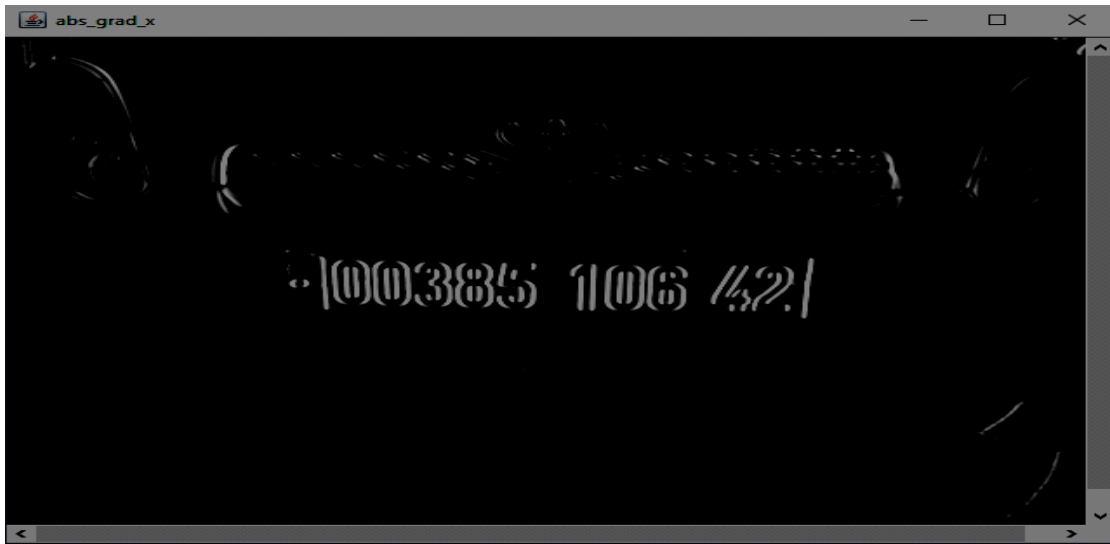


Figure13 : Image obtenu après détection des bords verticaux

4.4.2.1.2 Détection des bords horizontaux

Non seulement les plaques contiennent une forte densité de bords verticaux ce qui est de même pour les bords horizontaux.

La même opération qui a servie à la détection des bords verticaux sera appliquée à l'image pour la détection des bords horizontaux comme quoi après traitement nous aurons la **figure suivante** :

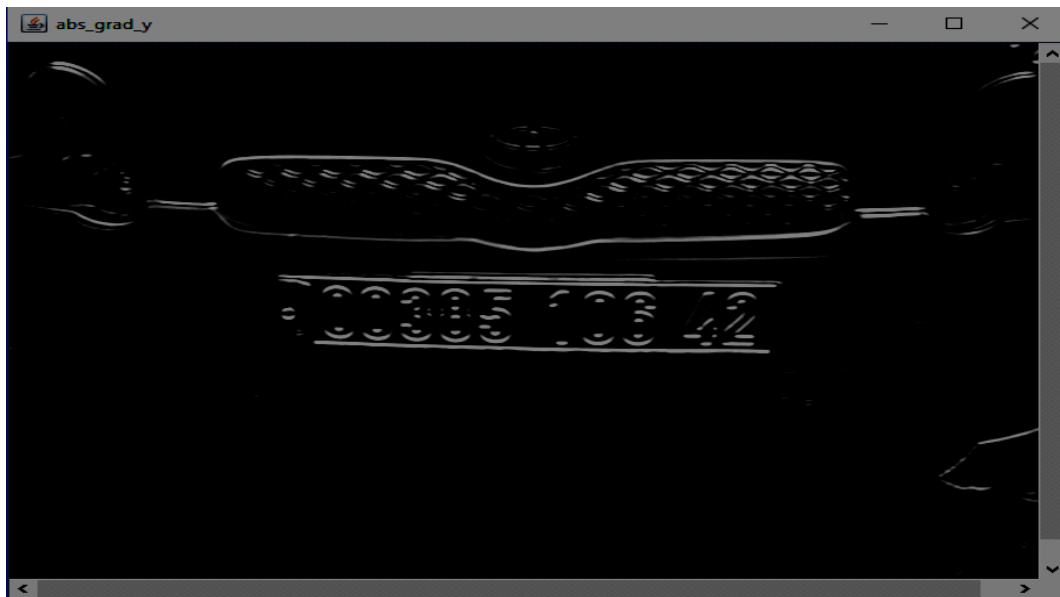


Figure14 : Image après détection des bords horizontaux

4.4.2.1.3 Jointure des bords verticaux et horizontaux

Nous avons déjà vu que les plaques contiennent une forte densité de bords verticaux et horizontaux, dans cette partie nous avons reliés les zones qui contiennent des lignes verticales et horizontales pour dire que la plaque de la voiture est sans nulle doute contenu dans cette zone ce qui va faciliter la détection des contours vu que la recherche se fait dans une zone limitée.

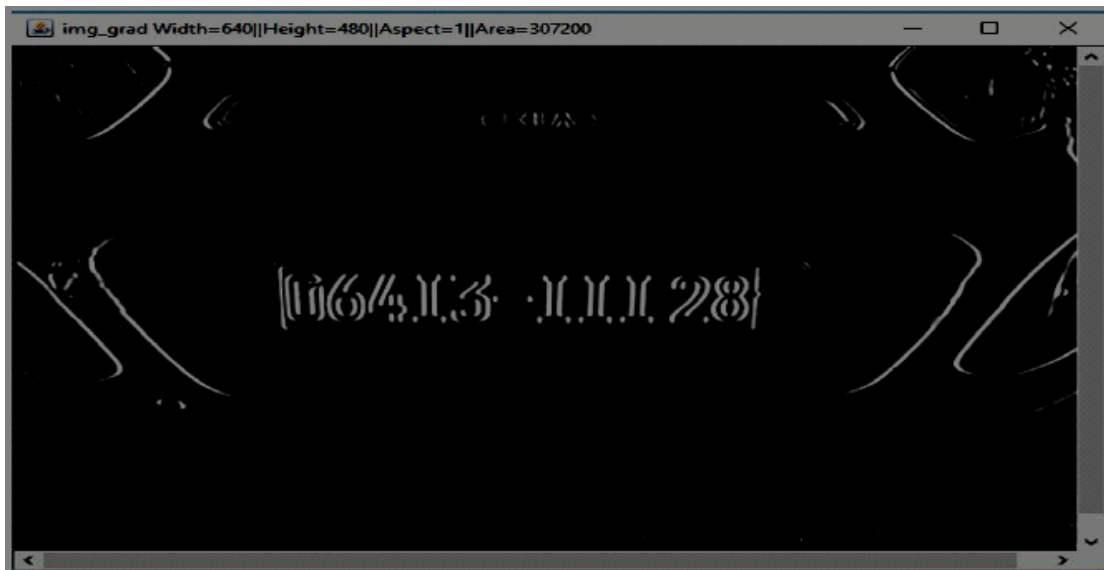


Figure15 : Image après jointure des bords verticaux et horizontaux

4.4.2.1.4 opération du seuil (Threshold Oporation)

Après détection des bords horizontaux et verticaux, nous appliquerons un filtre de seuil pour obtenir une image binaire avec un seuil. Cela nous permettra d'effectuer certain traitement nécessaire à la phase de détection de contour qui sera effectué par OpenCV.

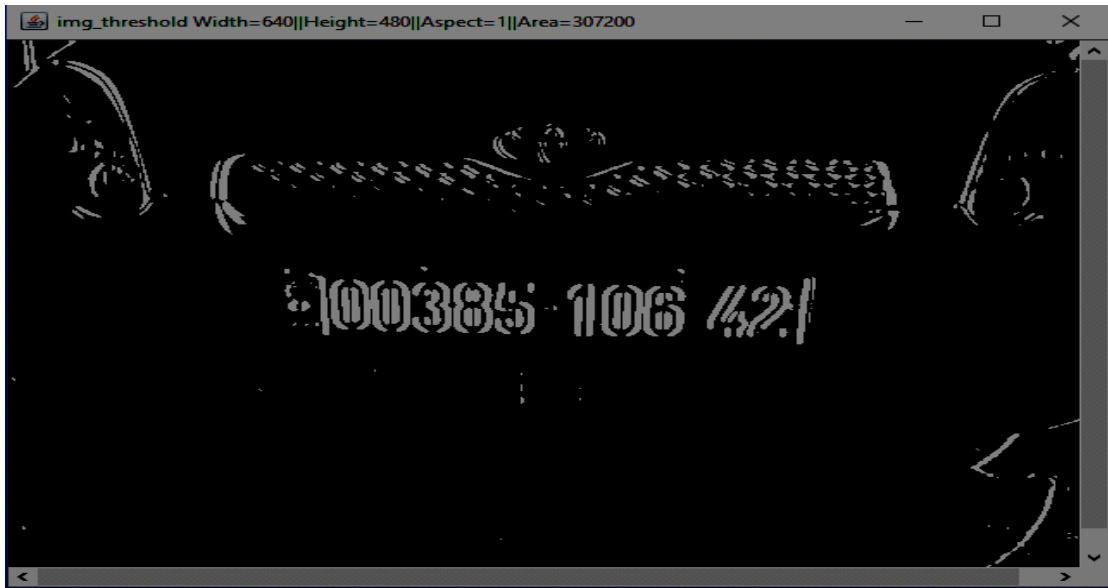


Figure16 : Image binaire.

4.4.2.1.5 Opération morphologique

En appliquant une opération morphologique proche, on peut supprimer des espaces vides entre chaque bord vertical ligne et connectez toutes les régions qui ont un nombre élevé d'arêtes. Dans cette étape, nous avons des régions pouvant contenir des plaques.

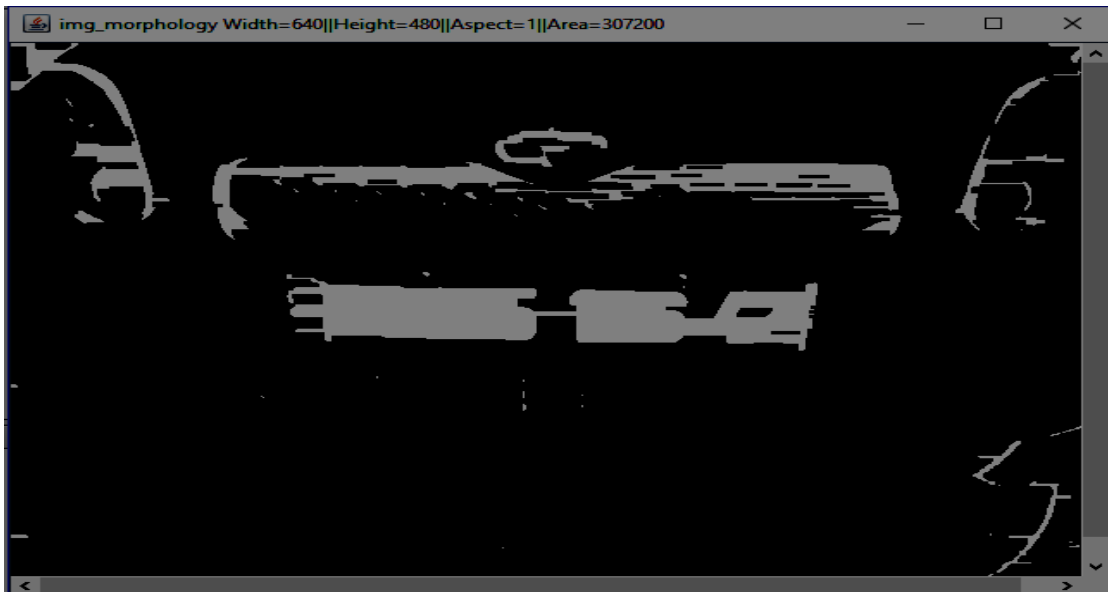


Figure17 : Image après délimitation.

Tout le traitement précédent a été réalisé en préparation du terrain pour cette étape qui va servir à OpenCv (CvFindContour) à scanner l'image et à détecter les contours qui sont plus clairs et plus détectable. Nous ignorons les petits rectangles détectés, car une plaque à d'importantes dimensions par rapport à la proportion de l'image.

A la fin nous ne gardons que les rectangles de tailles importantes afin de les fournir au lecteur OCR qui va se servir pour la lecture.

4.4.3 Phase de lecture OCR

Cette phase rassemble des techniques qui visent à récupérer les caractères de la plaque d'immatriculation avec la reconnaissance optique des caractères.

Maintenant nous avons un contour valide qui représente une plaque d'immatriculation, nous pouvons mettre en œuvre le processus OCR pour reconnaître les numéros.

A ce stade on ne fait que transformer l'image en gris en une image binaire qui satisfait notre besoin d'une écriture noire sur un arrière-plan blanc car notre OCR Tess4J est dédié à la lecture de documents administratifs à arrière-plan blanc ce qui est la situation idéale pour un meilleur résultat de lecture afin de maximiser nos chances d'avoir le bon matricule.

En effet la bibliothèque OCR que nous avons utilisé c'est à dire TEST4j utilise un algorithme d'apprentissage automatique du réseau neuronal artificiel pour reconnaître les caractères. [22]

Enfin pour mettre le processus de lecture OCR sur l'image de la plaque, il suffit de cliquer sur l'icône « **process** », cette fenêtre sera affichée qui contient à droite les numéros de la plaque ainsi que le pourcentage de confidentialité.

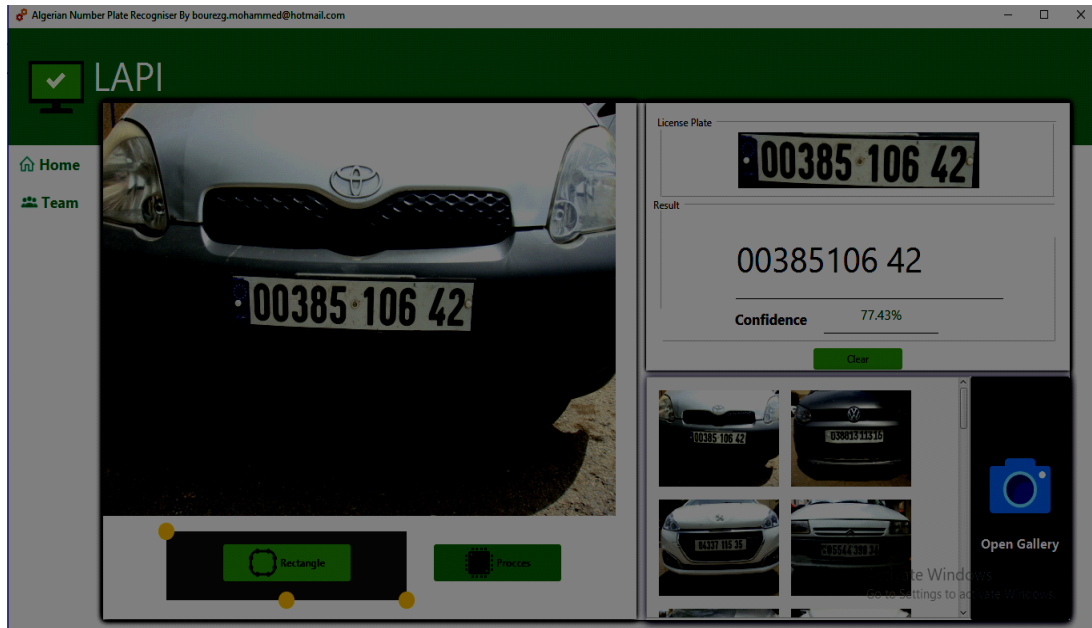


Figure18 : Fenêtre de lecture OCR.

Les OCR ne sont pas si précis on remarque une augmentation du chiffre 1 dans la lecture ; due au fait que nous avons utilisé que les caractères numériques et non alphanumériques car les plaques algériennes ne contiennent que des caractères numériques.

Cependant, sans ces étapes, avoir un tel résultat serait impossible. Nous avons montré cela, dans l'étape de la lecture manuelle où la confidentialité était faible.

4.5 Conclusion

Nous avons présenté dans ce chapitre notre système de reconnaissance de plaques Minéralogiques algériennes où nous avons détaillé les différentes étapes nécessaires afin d'obtenir l'objectif escompté.

Ce système a été réalisé en grande partie grâce aux différentes phases de traitement que nous avons utilisé.

Conclusion Générale

Conclusion générale

L'objectif de notre travail est de développer un système de détection et de reconnaissance de plaques d'immatriculation ayant la capacité d'extraire les numéros d'immatriculation algérienne.

Dans ce projet de fin d'études, nous avons présenté un état de l'art sur les systèmes LAPI existants, les techniques de traitements d'images ainsi que les outils technologiques qui nous ont permis de réaliser un système de détection de plaques d'immatriculation algériennes nommé APR (Algerian Plate Recognition).

Grâce à la réalisation de ce système, nous avons atteint l'objectif de notre travail qui consiste à réaliser un système de détection et de reconnaissance de plaques minéralogiques ayant la capacité d'extraire les numéros d'immatriculation algérienne.

En plus, d'avoir atteint l'objectif final demandé, ce projet nous a été très bénéfique car il nous a permis de maîtriser plusieurs techniques et de manipuler des outils très complexes dans le domaine de traitement d'image. Enfin, ce projet était une bonne occasion pour réaliser un travail très concret, avec des objectifs clairs et bien définis et de se familiariser avec un environnement de développement professionnel.

Comme perspectives, nous proposons une amélioration des résultats obtenus en utilisant des OCR open source.

Bibliographie

Bibliographie

- [4] A.NOBEL, "Reconnaissance automatique de plaques d'immatriculation (AutoVu) ", Genetec, 2013-Canada.
- [10] H. Maître (ss la direction de), " Le traitement des images ", Hermes Lavoisier IC2 2003.
- [11] C.F.CREATIS, " Traitement et analyse d'image-ANIMAG ", Institut national des sciences, Université Claude Bernard, 2013-Lyon.
- [12] cours en ligne disponible sur <http://www.informatique-enseignant.com/image-bitmap-ou-vectorielle/>
- [13] Hervé MATHIEU "la chaine de l'acquisition d'images", http://www.inrialpes.fr/sed/people/mathieu/biblio.html#cours_acquis, INRIA Rhône- Alpes.
- [14] J.P.Gastellu-Etchegorry, "d'acquisition et traitement d'image numérique", IUT Département de mesures physique, Université Paul Sabatier, Avril 2008.
- [15] "généralités sur le traitement d'image", https://fr.wikipedia.org/wiki/Traitement_d'images
- [17] Herbert Schildt, "introducing javafx8 programing", oracle press, new York 2015.
- [18] Jayneil Dalal, Sohil Patel, "opencv starter", packt publishing 2013.
- [21] cours en ligne disponible sur <https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/MasterInfo/Cours/Swing/mvc.html>
- [22] Daniel Lelis, Shervin Emami, David Milan, Jason Saragih, "Mastering Opencv3 " second Edition, packt publishing 2017.
- [23] Nor Imane, Sidhoum Souad, "development d'une application de detection et de reconnaissance de plaques d'immatriculation" Université de tlemcen, Tlemcen 2017.

Webographie

- [1] A. Bertrand, "Lapi ou VPI quelles différences",
<http://cameravideosurveillancesecurite.blogspot.com/p/lapi-ou-vpi-quelles-differences.html>,
dernière visite le 05/06/2018
- [2] " Lecture automatique de plaques minéralogiques ",
https://fr.wikipedia.org/wiki/Lecture_automatique_de_plaques_minéralogiques, dernière visite :
25/5/2018.
- [3] "définition OCR ", <https://www.futurasciences.com/tech/definitions/informatique-ocr-3953/>,
dernière visite le 7/06/2018
- [5] "LAPI-Engine", <http://www.alphanumeric-vision.com/fr/logiciel-lapi/>, dernière visite :
25/05/2018.
- [6] "Reconnaissance de plaques d'immatriculation avec LAPI de SeeTec",
<https://www.seetecvideo.com/fr/produits/extensions-seetec-cayuga/reconnaissance-des-plaquesdimmatriculation/>, dernière visite : 25/05/2018.
- [7] "Asia Vision Technology Limited (AVT) ", <https://asiavisionweb.com/> , derrière visite :
25/05/2018.
- [8] "Nos logiciels - AFS2R", <http://afs2r.fr/nos-produits/>, dernière visite : 29/05/2018.
- [9] "TEB - Système LAPI : Lecture Automatique de Plaque d'Immatriculation",
<http://www.tebonline.com/fr/analyse-images/lapi-lecture-automatique-plaque-immatriculation.php>, 07/06/2018.
- [16] "Qu'est-ce que c'est que Java et pourquoi j'en ai besoin",
https://www.java.com/fr/download/faq/whatis_java.xml , dernière visite :21/05/2018.
- [19] " java scene builder",
<http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-info-2157684.html>, derrière visite : 21/05/2018.
- [20] " Tess4J - JNA wrapper for Tesseract - SourceForge", <http://tess4j.sourceforge.net/>,
dernière visite : 21/05/2018.

Résumé :

Ce projet consiste à réaliser un système de détection et de reconnaissance des plaques d'immatriculation de véhicules algériens. Pour cela nous avons réalisé un état de l'art et une étude préalable afin de recueillir les informations nécessaires pour permettre de cerner les différents traitements à implémenter dans ce système. Une application portable interactive a été développée et les résultats ont été très satisfaisants.

Notre système dénommé APR a été développé en passant par trois phases. Tout d'abord un prétraitement a été pour filtrer et réduire le bruit de l'image originale. Ensuite une phase de détection de contour sur l'image obtenue en utilisant la segmentation, et enfin nous appliquons une lecture OCR pour reconnaître les formes de caractères sur la plaque d'immatriculation.

Mots clés : Lecture Automatique de Plaques d'Immatriculation (LAPI), OCR, Segmentation.

Abstract :

This project involves the creation of a system for the detection and recognition of Algerian vehicle license plates. For that we realized a state of the art and a preliminary study in order to gather the necessary information to make it possible to define the different treatments to be implemented in this system. An interactive portable application has been developed and the results have been very satisfactory.

Our system called APR has been developed through three phases. Firstly a pretreatment was to filter and reduce the noise of the original image. Then a contour detection phase on the image obtained using segmentation, and finally we apply an OCR reading to recognize the character forms on the license plate.

Keywords : Automatic License Plate Reading (LAPI), OCR, Segmentation.

ملخص:

يتضمن هذا المشروع إنشاء نظام للكشف عن لوحات ترخيص المركبات الجزائرية والتعرف عليها. من أجل ذلك قمنا بدراسة أولية من أجل جمع المعلومات اللازمة لجعل من الممكن تحديد التقنيات اللازمة التي سيتم تطبيقها في هذا النظام. تم تطوير تطبيق محمول تفاعلي وكانت النتائج مرضية للغاية.

لقد تم تطوير نظام تحت تسمية APR من خلال ثلاث مراحل. أولاً، كانت المعالجة المسبقة التي هي تصفية وتقليل ضوضاء الصورة الأصلية. ثم مرحلة الكشف عن الأشكال الهندسية في الصورة التي تم الحصول عليها باستخدام التقسيم وأخيراً تطبيق قراءة OCR للتعرف على أحرف لوحة الترخيم. الكلمات المفتاحية: القراءة التلقائية لوحة الترخيم (LAPI)، OCR، تقسيم.