

# Sommaire

<b>Remerciements</b> .....	<b>2</b>
<b>Sommaire</b> .....	<b>5</b>
<b>Liste des figures</b> .....	<b>8</b>
<b>INTRODUCTION GENERALE</b> .....	<b>9</b>
<b><i>Introduction générale</i></b> .....	<b>10</b>
<b>1.Etat de l'art</b> .....	<b>14</b>
1.1 Introduction .....	15
1.2 Lecture automatique de plaques d'immatriculation.....	15
1.2.1 Caractéristiques des plaques algériennes.....	16
1.2.2 La technique de la reconnaissance optique de caractères .....	17
1.3 Algorithmes du système LAPI .....	17
1.4 Difficultés des systèmes LAPI .....	18
1.5 Applications des systèmes LAPI.....	18
1.6 Les différents systèmes existants .....	19
1.6.1 AutoVu.....	19
1.6.2 Avantages d'AutoVu .....	19
1.6.3 LAPI ENGINE .....	19
1.6.4 SeeTec .....	19
1.6.5 Asia Vision Technology Limited (AVT) .....	20
1.6.6 AGL (Application de Gestion LAPI) .....	20
1.6.7 Système LAPI-Pryncar.....	21
1.7 Conclusion .....	21
<b>2.Traitement d'images</b> .....	<b>23</b>
2.1 Introduction .....	24
2.2 Définition du traitement d'images .....	24
2.3 Image .....	25
2.4 Représentation d'image .....	25

2.4.1	Image matricielle .....	26
2.4.2	Image vectorielle .....	26
2.5	Acquisition d'une image .....	26
2.6	Caractéristiques d'une image numérique .....	26
2.6.1	Dimension.....	27
2.6.2	Résolution.....	27
2.6.3	Bruit.....	27
2.6.4	Histogramme .....	27
2.6.5	Luminance .....	27
2.6.6	Contraste .....	28
2.6.7	Images à niveaux de gris .....	28
2.6.8	Image en couleurs .....	28
2.7	Principales étapes de traitement d'images .....	29
2.7.1	Prétraitements .....	29
2.7.2	Amélioration d'image.....	29
2.7.3	Analyse d'image .....	30
2.8	Types de données manipulée.....	30
2.9	Opérateurs de traitement d'image .....	31
2.9.1	Opérateurs locaux .....	32
2.9.2	Filtres linéaires .....	33
2.9.3	Opérateurs morphologiques .....	35
2.9.4	Construction d'une application de traitement d'images.....	35
2.10	Reconnaissance d'objets.....	35
2.11	Quelque exemples concrets de traitement d'image .....	36
2.12	Conclusion.....	38
<b>3.</b>	<b>Outils de développement du système LAPIA .....</b>	<b>39</b>
3.1	Introduction .....	40
3.2	Outils de développement .....	40
3.2.1	Maven.....	40
3.2.2	Github.....	41
3.2.3	Swing .....	41

3.2.4	OpenCV.....	42
3.3	Techniques et outils de Reconnaissance Optique des Caractères (OCR).....	43
3.3.1	Tess4j.....	43
3.4	Conclusion .....	44
<b>4.</b>	<b>IMPLEMENTATION DU SYSTEME LAPIA .....</b>	<b>45</b>
4.1	Introduction .....	46
4.2	Architecture de notre système LAPIA .....	46
4.3	Les interfaces de l'application .....	47
4.3.1	Le formulaire principal .....	47
4.3.2	Phase de traitement d'image .....	48
4.3.3	Phase pour la détection des contours.....	51
4.3.4	Phase de lecture OCR .....	55
4.4	Conclusion .....	58
	<b>CONCLUSION GENERALE .....</b>	<b>59</b>
	<b>Conclusion générale .....</b>	<b>60</b>
	<b>Références Bibliographiques .....</b>	<b>61</b>
	<b>Webographie .....</b>	<b>62</b>
	<b>Résumé .....</b>	<b>63</b>
	<b>Abstract .....</b>	<b>63</b>
	<b>ملخص .....</b>	<b>63</b>

# Liste des figures

Figure 1 : la chaine de transformation pour obtenir les caractères ASCII.....	15
Figure 2 : Exemple de plaque d'immatriculation Algérienne .....	16
Figure 3 : Différents types d'images. ....	25
Figure 4 : architecture du système LAPI. ....	46
Figure 5 : Formulaire principale .....	47
Figure 6 : Phase de traitement d'image.....	48
Figure 7 : Fenêtre chargement d'image.....	49
Figure 8 : Lecture direct de la plaque .....	49
Figure 9 : a-Image originale b-Image avec diminution du bruit .....	50
Figure 10 : Fenêtre Conversion de l'image en niveau de gris.....	51
Figure 11 : Etapes de la détection.....	51
Figure 12 : Fenêtre transformation en binaire. ....	52
Figure 13 : Fenêtre transformation de Canny.....	53
Figure 14 : fenêtre de dilatation .....	54
Figure 15 : Fenêtre d'érosion.....	54
Figure 16 : Fenêtre de délimitation. ....	55
Figure 17 : Phase de lecture OCR.....	55
Figure 18 : Fenêtre de transformation en binaire pour l'OCR.....	56
Figure 19: Fenêtre de dilatation et érosion OCR.....	56
Figure 20 : Fenêtre de lecture OCR.....	57

# **INTRODUCTION GENERALE**

## *Introduction générale*



### **Contexte**

Le numéro d'immatriculation représente un moyen efficace pour identifier les véhicules. Il s'agit d'une information unique pour chaque voiture.

Fréquemment, il est nécessaire d'identifier les plaques d'immatriculation des véhicules pour la sécurité. Les informations extraites peuvent être utilisées pour plusieurs intérêts, 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 leurs lectures cruciales et inévitables 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 vraie problématique pour identifier des plaques d'immatriculation réside dans le fait de pouvoir faire de la reconnaissance optique de caractères sur une petite partie d'image extraite de 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 un prétraitement de ces images à la détection des contours, pour permettre une reconnaissance optique assez fiable des caractères.

Ce que nous visons à travers notre travail est de faciliter la tâche, d'identification des caractères du matricule en exploitant les avantages que peut offrir le traitement d'image. [1] Pour cela, nous avons développé une application qui consiste à réaliser des prétraitements, la détection et la lecture automatique des caractères.

## **Plan de mémoire**

Pour cela, nous avons décomposé notre mémoire comme suit :

Le premier chapitre, nous commencerons 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 finirons ce chapitre par quelques exemples de systèmes existants.

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

Le troisième chapitre, nous avons présenté un certain nombre d'outils que nous avons utilisés pour réaliser notre travail.

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

Nous terminerons ce mémoire par une conclusion générale et des perspectives.







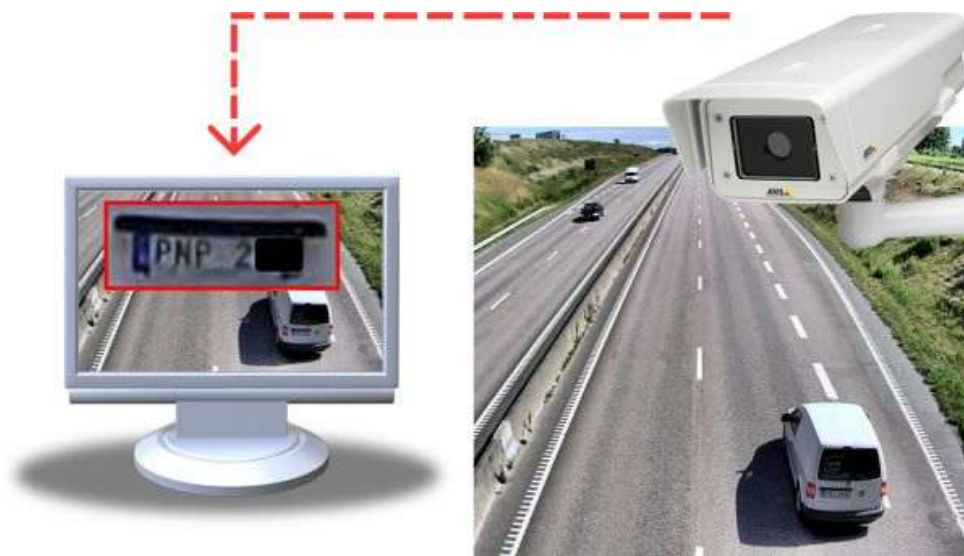
# 1. Etat de l'art

## 1.1 Introduction

Nous commencerons ce chapitre par une définition de la Lecture Automatique de Plaques d'Immatriculation (LAPI) et la technique de la reconnaissance de caractères. Après ceci, nous allons présenter les applications LAPI. Nous finirons ce chapitre par exposer quelques exemples de systèmes LAPI.

## 1.2 Lecture automatique de plaques d'immatriculation

La lecture automatique de plaques minéralogiques ou Lecture Automatisée de Plaques d'Immatriculation (LAPI) est une méthode d'identification qui utilise des techniques de traitement d'images et de vision par ordinateur pour extraire le numéro d'identification d'après l'image de la plaque sous format de caractères en code ASCII .  
[2]



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

Un système (LAPI) permet un dialogue avec un grand nombre de périphériques:

- ❖ Caméra.
- ❖ Radar de vitesse.
- ❖ Lasers de détection.

Comme nous nous intéressons aux plaques d'immatriculation algérienne, nous commençons par donner quelques explications de base concernant les plaques d'immatriculation de véhicules en Algérie :

### 1.2.1 Caractéristiques des plaques algériennes

La plaque est composée de dix chiffres (Fig.2) :



**Figure 2** : 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 de l'automobile. Ici c'est 1 car c'est une voiture touristique, 2 pour les camions, etc.
  - **08453** signifie que ce véhicule est le 8453<sup>ème</sup> qui est entré dans la wilaya 13 (Tlemcen) en 2012.
- 
- La plaque a une forme de rectangulaire, Le style des caractères varie d'une plaque à l'autre et la distance entre les caractères n'est pas standardisé, Elle dépend du constructeur de la plaque.
  - Les plaques sont blanches (à l'avant), jaunes (à l'arrière) et la couleur des caractères est noire.

- Récemment les gens aiment ajouter quelques illustrations sur la plaque d'immatriculation de leur voiture, comme par exemple le drapeau Algérien, un numéro de téléphone, le nom de la ville ou du pays, le nom d'une entreprise, etc.

Basé sur les caractéristiques ci-dessus, nous proposons, dans la section suivante, la technique utilisée dans ce système pour la reconnaissance optique de caractères.

### 1.2.2 La technique de la reconnaissance optique de caractères

La lecture automatique de plaques minéralogiques utilise la reconnaissance optique de caractères (OCR) sur des images prises par des caméras. 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. [2]

### 1.3 Algorithmes du système LAPI

Cinq étapes doivent être réalisées pour que le logiciel puisse identifier une plaque d'immatriculation :

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

La complexité de chacune des étapes du programme 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 numérique peut aussi être utilisé pour réduire le « bruit » visuel de l'image.

## 1.4 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).

## 1.5 Applications des systèmes LAPI

La lecture automatique de plaques minéralogiques peut également être utilisée pour:

- Les passages de frontière.
- Les stations-services (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.
- Comparer les plaques d'immatriculations au Fichier des véhicules volés.

Pour certaines applications, le système peut être associé à d'autres algorithmes de reconnaissance de type facial ou de véhicule.

## 1.6 Les différents systèmes existants

De nos jours, il existe de nombreux systèmes de la reconnaissance de plaques d'immatriculation, tels que :

### 1.6.1 AutoVu

Est le système de reconnaissance de plaques d'immatriculation (RAPI) sur IP de Security Center, la plate-forme de sécurité unifiée de Genetec. [3]

### 1.6.2 Avantages d'AutoVu

- Obtenez des lectures de plaques d'immatriculation précises.
- Assurez des performances interrompues.
- Fournisseur de solutions de RAPI complètes.
- Unification avec la vidéosurveillance.
- Combinez la RAPI fixe.

### 1.6.3 LAPI ENGINE

Le produit LAPI ENGINE représente le cœur technologique permettant la Lecture Automatique de Plaques d'Immatriculation (LAPI - ANPR). Principalement dédiée à la traçabilité de véhicules, LAPI ENGINE est un produit autonome pouvant s'adapter à un large éventail d'applications. [4]

#### 1.6.3.1 Avantages

- Possibilité de crypter les données (images et n° plaques) avec un système de gestion de clé unique.
- Lecture de tous types de plaques réfléchissantes aux infrarouges ou non.
- Maintenance et intervention sur tout le territoire Français (Télé-maintenance possible).
- Moteur LAPI fonctionnant sur Windows, Linux.

### 1.6.4 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. [5]

### **1.6.4.1 Avantages**

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

### **1.6.5 Asia Vision Technology Limited (AVT)**

Est le premier fournisseur mondial de solutions technologiques et de solutions de gestion intelligente des véhicules et des conteneurs. AVT a été un pionnier dans le développement et la fourniture de la technologie de reconnaissance optique de caractères (OCR). [6]

#### **1.6.5.1 Avantages d'Asia Vision**

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

### **1.6.6 AGL (Application de Gestion LAPI)**

Est basée essentiellement sur la technologie LAPI, le but d'AGL est de permettre l'automatisation de processus pour les forces de l'ordre, 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. [7]



### 1.6.6.1 Avantages d'AGL

Elle 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.

Elle permet d'acquérir des données, et de les utiliser sur le terrain. Elle est reliée d'une part aux services nationaux, aux prestataires tiers, aux services locaux des collectivités et d'autre part aux systèmes AGL Capture.

### 1.6.7 Système LAPI-Pryncar

Est enregistrée 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. [8]

#### 1.6.7.1 Avantages de LAPI-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.

Nous avons trouvée beaucoup systèmes, nous avons citée quelques-un d'entre elles 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.

## 1.7 Conclusion

Nous avons vu dans ce chapitre la notion de Lecture Automatique de Plaques Minéralogique et ses différents algorithmes qui doivent être réalisés pour que le logiciel puisse identifier une plaque d'immatriculation et les difficultés qui en résultent. Comme nous le verrons quelques exemples de système existants.

Le chapitre suivant détaillera les techniques de traitement d'images utilisées dans les systèmes LAPI.

## 2.Traitement d'images

## 2.1 Introduction

Le traitement d'images est un domaine très vaste qui a connu, et qui connaît encore, un développement important depuis quelques dizaines d'années.

Nous avons désignée par traitement d'images numériques l'ensemble des techniques permettant de modifier une image numérique afin d'améliorer ou d'en extraire des informations.

Dans ce chapitre, nous abordons les notions de base nécessaires à la compréhension des techniques de traitement d'images. Ensuite, nous allons donner un aperçu sur les différentes techniques connues dans ce domaine.

## 2.2 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. [9]

La compréhension du traitement d'images commence par la compréhension 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 déterminent par exemple la netteté de l'image.
- **Les conditions d'éclairage**, qui déterminent une partie de la variabilité des images traitées.

Quelques exemple de 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.

### 2.3 Image

Une image est avant tout un signal 2D  $(x, y)$ , qui représente souvent une réalité 3D  $(x, y, z)$ . D'un point de vue mathématique, une image est une matrice de nombres représentant un signal, plusieurs outils permettent de manipuler ce signal. [10]

Il existe trois principaux types d'images :

1. Les images binaires (uniquement en noir et blanc) et dont la valeur soit 0 soit 1.
2. Les images en niveaux de gris, dont la valeur appartient à l'ensemble  $\{0,1,\dots,255\}$ .
3. Les images couleurs.

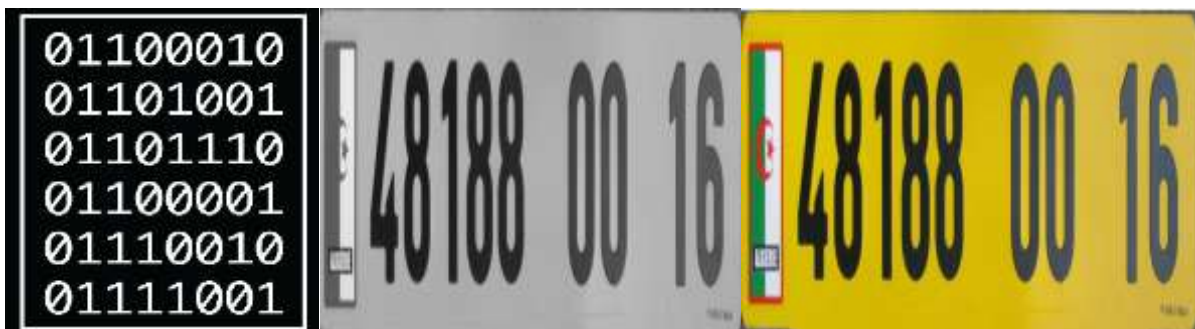


Figure 3 : Différents types d'images.

### 2.4 Représentation d'image

Les images numériques, destinées à être visualisées sur les écrans d'ordinateur, se divisent en 2 grandes classes : [11]

- Les images matricielles
- Les images vectorielles.

### 2.4.1 Image matricielle

Encore appelée image bitmap, Une image matricielle est formée d'un assemblage de points ou de pixels.

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

### 2.4.2 Image vectorielle

Une image vectorielle en informatique, 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. (définis de manière mathématique).

Par exemple, une image vectorielle d'un cercle est définie par des attributs de types : position du centre, rayon, etc....

## 2.5 Acquisition d'une image

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

- Les caméras numériques.
- Les scanners.

## 2.6 Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants : [12]

### 2.6.1 Dimension

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

### 2.6.2 Résolution

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

### 2.6.3 Bruit

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

### 2.6.4 Histogramme

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

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant.

### 2.6.5 Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour

un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes).
- Un bon contraste: il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir, ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites.

### **2.6.6 Contraste**

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

### **2.6.7 Images à niveaux de gris**

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, 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.

### **2.6.8 Image en couleurs**

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

## 2.7 Principales étapes de traitement d'images

Il n'existe pas de méthode de traitement d'images générale à tous les domaines d'application possibles. Il faut en général employer des algorithmes spécifiques. Ces derniers sont souvent des combinaisons de techniques classiques (segmentation, classification, reconnaissance de frontières, etc.). De manière schématique, toute méthode de traitement d'images comprend 3 étapes majeures : [12]

- Prétraitement des images.
- Amélioration des images.
- Analyse des images.

### 2.7.1 Prétraitements

Ils préparent l'image pour son analyse ultérieure. Il s'agit souvent d'obtenir l'image théorique que l'on aurait dû acquérir en l'absence de toute dégradation. Ainsi, ils peuvent par exemple corriger :

- 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.

### 2.7.2 Amélioration d'image

Elle a pour but d'améliorer la visualisation des images. Pour cela, elle élimine / réduit le bruit de l'image et/ou met en évidence certains éléments (frontières, etc.) de l'image. Elle est souvent appliquée sans connaissance à priori des éléments de l'image. Les principales techniques sont :

- L'amélioration de contraste.

- 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.

### 2.7.3 Analyse d'image

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 (e.g. 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.

## 2.8 Types 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.

## 2.9 Opérateurs de traitement d'image

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.

Nous avons classé 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) : [10]

### ➤ **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, or, 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.

### 2.9.1 Opérateurs locaux

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

- Les filtres linéaires.
- Les filtres 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-Zeno (détecteur de contour généralisé au cas couleur), le filtre Retinex, ainsi que les opérateurs homomorphiques (ceux qui travaillent sur le logarithme de l'image).

Nous avons 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, nous pouvons 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 de contour : nous nous trouvons 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 (chaîne 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 nous retrouvons souvent une partie de classification des pixels en classes. Nous essayons 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.

## 2.9.2 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. [9]

### 2.9.2.1 Lissage

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.



### 2.9.2.2 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. [10]

- **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.

### 2.9.3 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. [9]

### 2.9.4 Construction d'une application de traitement d'images

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, nous allons 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.

De nombreuses briques sont disponibles permettant de créer des applications complexes et évoluées. [9]

## 2.10 Reconnaissance d'objets

La reconnaissance d'objets est une branche de la **vision artificielle** et un des piliers de la **vision industrielle**. Elle consiste à identifier des formes pré-décrites dans une **image numérique**, et par extension dans un flux **vidéo numérique**.

Il ne faut pas confondre reconnaissance d'objets (en anglais : « object recognition » ou « shape recognition ») et **reconnaissance de formes** (« pattern recognition » en anglais). La première s'attache à reconnaître des formes géométriques dans une image, alors que la seconde cherche à identifier des motifs dans des données statistiques. La confusion vient du fait qu'on utilise souvent la reconnaissance de formes comme technique appliquée à la reconnaissance d'objets.

Tout d'abord objet d'algorithmes dirigés par l'homme, jusqu'en 1995 (tentatives de reproduire par un algorithme un raisonnement humain d'identification, comme dans « un vélo possède deux roues, un cadre... »), la reconnaissance d'objets a fait l'objet de progrès importants par la suite au travers de la mise en œuvre de techniques d'apprentissage, comme les **séparateurs à vaste marge**. Ces techniques visent à faire exploiter des bases d'exemples positifs et négatifs (contre-exemples) par un algorithme de recherche de critères discriminants, c'est-à-dire de critères permettant de séparer au mieux les exemples des contre-exemples. [9]

## 2.11 Quelques exemples concrets de traitement d'image

- **Contrôle de présence/absence** sur des chaînes de production, nous avons vérifié 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.
- **Contrôle du niveau de maturation des fruits sur une chaîne de conditionnement.** Il s'agit de reconnaître à la couleur et à la texture du fruit son degré de maturité et donc la catégorie sous laquelle il sera emballé puis vendu.
- **Construction et correction de cartes géographiques d'après des images satellites ou des images aériennes.** On recalcule 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...



- **Surveillance et évaluation de la production agricole.** Il est possible de déterminer le degré de maturation des cultures, la quantité d'eau nécessaire pour l'irrigation, le rendement moyen... Nous pouvons ainsi établir des prévisions à large échelle de la récolte à venir.
- **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 process industriels...
- **Segmentation et suivi de cellules vivantes en microscopie.** Cela permet d'analyser le comportement d'une population de cellules et ainsi de détecter certaines anomalies.

- **Analyse et authentification de tableaux.** L'étude des niveaux des couleurs des pigments et des vernis permet une analyse approfondie des œuvres. Il est ainsi possible de voir les restaurations successives et d'identifier les faux. [9]

## 2.12 Conclusion

Nous avons introduit dans ce chapitre la notion de base qui représente l'image et ses caractéristiques puis la compréhension de différentes techniques de traitement d'images et leurs opérateurs

Le chapitre suivant représente un certains nombre d'outils que nous avons utilisés pour réaliser notre travail.

Rapport-Gratuit.com

# 3. Outils de développement du système LAPIA

### 3.1 Introduction

Ce chapitre présente des outils de développement et différentes méthodes et outils disponibles pour la reconnaissance de forme, que nous avons utilisée dans notre application de détection de plaques minéralogiques.

### 3.2 Outils de développement

Parmi les différents outils de développement, nous avons choisi ces outils que nous avons utilisés pour réaliser notre travail.

#### 3.2.1 Maven

Maven est un outil de construction de projets (build) open source développé par la fondation Apache, initialement pour les besoins du projet Jakarta Turbine. Il permet de faciliter et d'automatiser certaines tâches de la gestion d'un projet Java. [13]

Il permet notamment :

- D'automatiser certaines tâches : compilation, tests unitaires et déploiement des applications qui composent le projet
- De gérer des dépendances vis-à-vis des bibliothèques nécessaires au projet
- De générer des documentations concernant le projet

Au premier abord, il est facile de croire que Maven fait double emploi avec Ant. Ant et Maven sont tous les deux développés par le groupe Jakarta, ce qui prouve bien que leur utilité n'est pas aussi identique que cela. Ant, dont le but est d'automatiser certaines tâches répétitives, est plus ancien que Maven. Maven propose non seulement les fonctionnalités d'Ant mais en propose de nombreuses autres.

Pour gérer les dépendances du projet vis-à-vis de bibliothèques, Maven utilise un ou plusieurs dépôts qui peuvent être locaux ou distants.

Maven est extensible grâce à un mécanisme de plugins qui permettent d'ajouter des fonctionnalités. [13]

### 3.2.1.1 Les concepts

Maven repose sur l'utilisation de plusieurs concepts :

- Les artefacts : composants identifiés de manière unique.
- Le principe de convention over configuration : utilisation de conventions par défaut pour standardiser les projets.
- Le cycle de vie et les phases : les étapes de construction d'un projet sont standardisées.
- Les dépôts (local et distant).

### 3.2.2 Github

**GitHub** est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. GitHub propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de logiciels libres. Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet. [14]

Le nom Github est composé du mot « **git** » faisant référence à un système de contrôle de version open-source et le mot « **hub** » faisant référence au réseau social bâti autour du système Git.

### 3.2.3 Swing

**Swing** est une bibliothèque graphique pour le langage de programmation Java, faisant partie du package Java Foundation Classes (JFC), inclus dans J2SE. Swing constitue l'une des principales évolutions apportées par Java 2 par rapport aux versions antérieures. [15]

Swing offre la possibilité de créer des interfaces graphiques identiques quel que soit le *système d'exploitation* sous-jacent, au prix de performances moindres qu'en utilisant Abstract Window Toolkit (AWT). Il utilise le principe *Modèle-Vue-Contrôleur* (MVC, les composants Swing jouent en fait le rôle de la vue au sens du MVC) et dispose de plusieurs choix d'apparence pour chacun des composants standards.

Avec l'apparition de Java 8 en mars 2014, JavaFX devient la bibliothèque 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).

### 3.2.4 OpenCV

OpenCV (**Open** Source **C**omputer **V**ision) est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de vision par ordinateur, accessibles au travers d'API pour les langages C, C++, et Python. Elle est distribuée sous une licence BSD (libre) pour les plate-formes Windows, GNU/Linux, Android et MacOS. [16]

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, pour java deux adaptation du projet JavaCV ont été faite afin de permettre son utilisation, la première est un artifact du projet OpenCV dans la repository de Maven, la deuxième est un artifact bytedeco qui se trouve aussi dans la repository Maven : ce sont deux interface DLL différente qui se sert du même noyau qui est OpenCV sous c++.

Afin de mieux vous présenter son étendue et ce qu'elle permet de faire, jetons un œil aux principaux modules accessibles :

- **core** : les fonctionnalités de base.

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

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

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

- **features2d** : descripteurs.

Ce module concerne principalement l'extraction de descripteurs selon deux

approches courantes (SURF et StarDetector), que nous aborderons lorsque nous nous intéresserons à la caractérisation d'images.

- **objdetect** : détection d'objets.

Cette bibliothèque permet de faire de la reconnaissance d'objets dans une image au moyen de l'algorithme Adaboost (Viola & Jones, 2001). Nous y reviendrons lorsque nous parlerons d'apprentissage et de reconnaissance de formes.

- **video** : traitement de flux vidéo.

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

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

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

- **calib3d** : calibration, estimation de pose et stéréovision.

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

### 3.3 Techniques et outils de Reconnaissance Optique des Caractères (OCR)

C'est un ensemble de techniques informatiques qui réalisent une reconnaissance de la forme des caractères, après captation d'image par procédé optique.

Le résultat est souvent transmis sous forme d'un fichier.

Nous avons testé un ensemble d'OCR, et nous avons opté pour le choix de «**Tess4J**» pour l'importance de son taux de reconnaissance.

#### 3.3.1 Tess4j

Tess4J est un wrapper JNA pour l'API de reconnaissance de caractères Tesseract.

Tess4J est diffusé et distribué sous Licence Apache, v2.0 et est également disponible auprès de Maven Central Repository.

Tess4J est en cours de développement et testé sur Windows et Linux. [17]

### 3.3.1.1 Caractéristiques :

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

- TIFF, JPEG, GIF, PNG et BMP formats d'image.
- Images TIFF multipages.
- PDF format de document.

### 3.4 Conclusion

Nous avons présenté un certain nombre de notions et de définitions concernant les outils de développements que nous avons utilisé pour réaliser notre travail.

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



# **4.IMPLEMENTATION DU SYSTEME LAPIA**

## 4.1 Introduction

Après avoir vu les différents outils de développement de notre système, nous passons à l'implémentation de l'application définies et détaillée dans ce chapitre en expliquant les écrans de notre système.

## 4.2 Architecture de notre système LAPIA

Pour qu'un système soit évolutif, extensible et maintenable, une bonne architecture doit être choisie. La figure ci-dessous montre l'architecture de notre système.

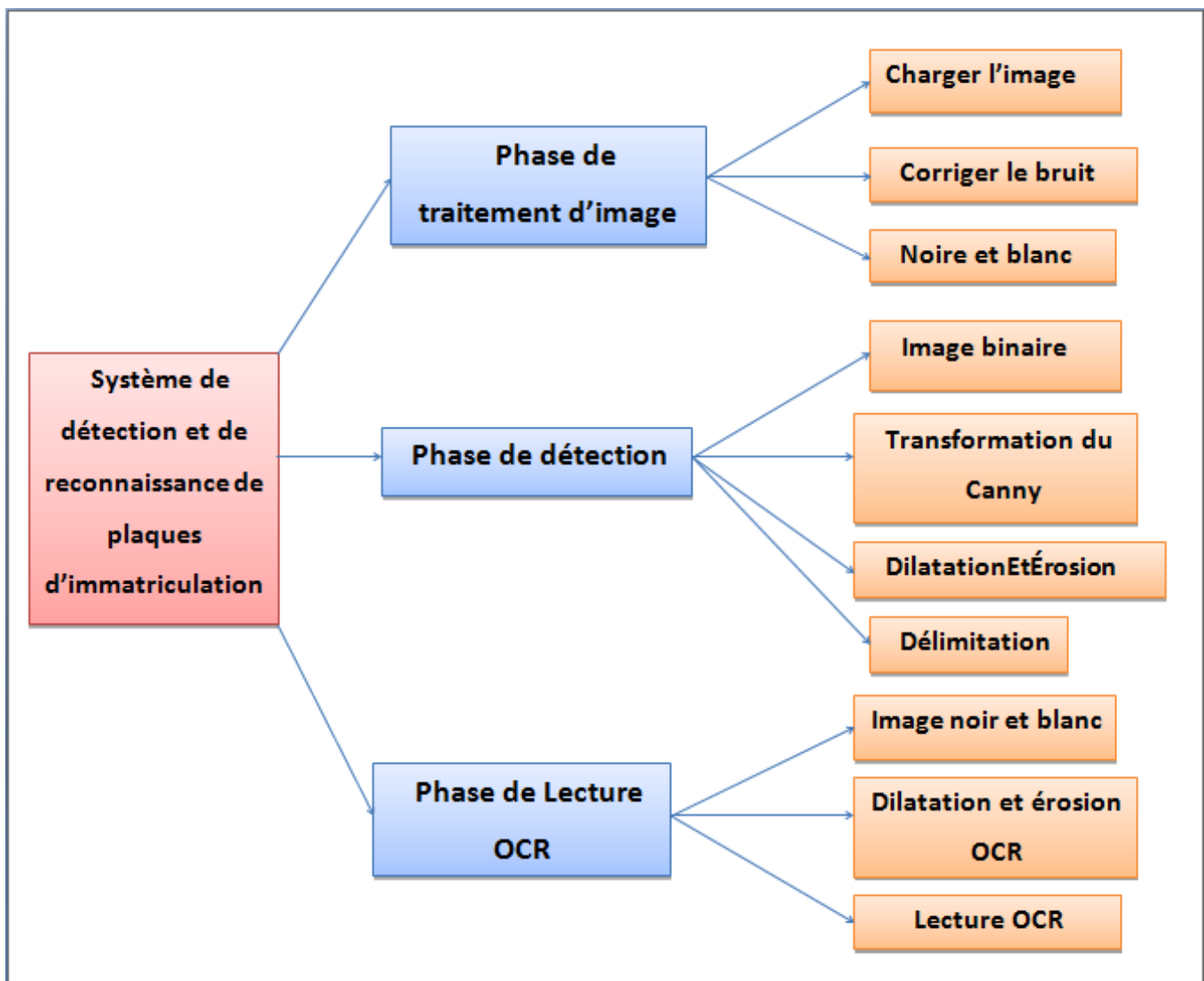


Figure 4: Architecture de notre système LAPIA.

### 4.3 Les interfaces de l'application

Pour mieux comprendre le fonctionnement de l'application, les figures qui suivent décrivent la chronologie des principales fonctionnalités de l'application.

#### 4.3.1 Le formulaire principal

Au lancement de notre application, une fenêtre se compose de trois parties (image, détection et lecture OCR) contenant les étapes en ordre allant de gauche vers la droite.

C'est une succession d'étapes ou chaque phase utilise le résultat de celle qui la précède.

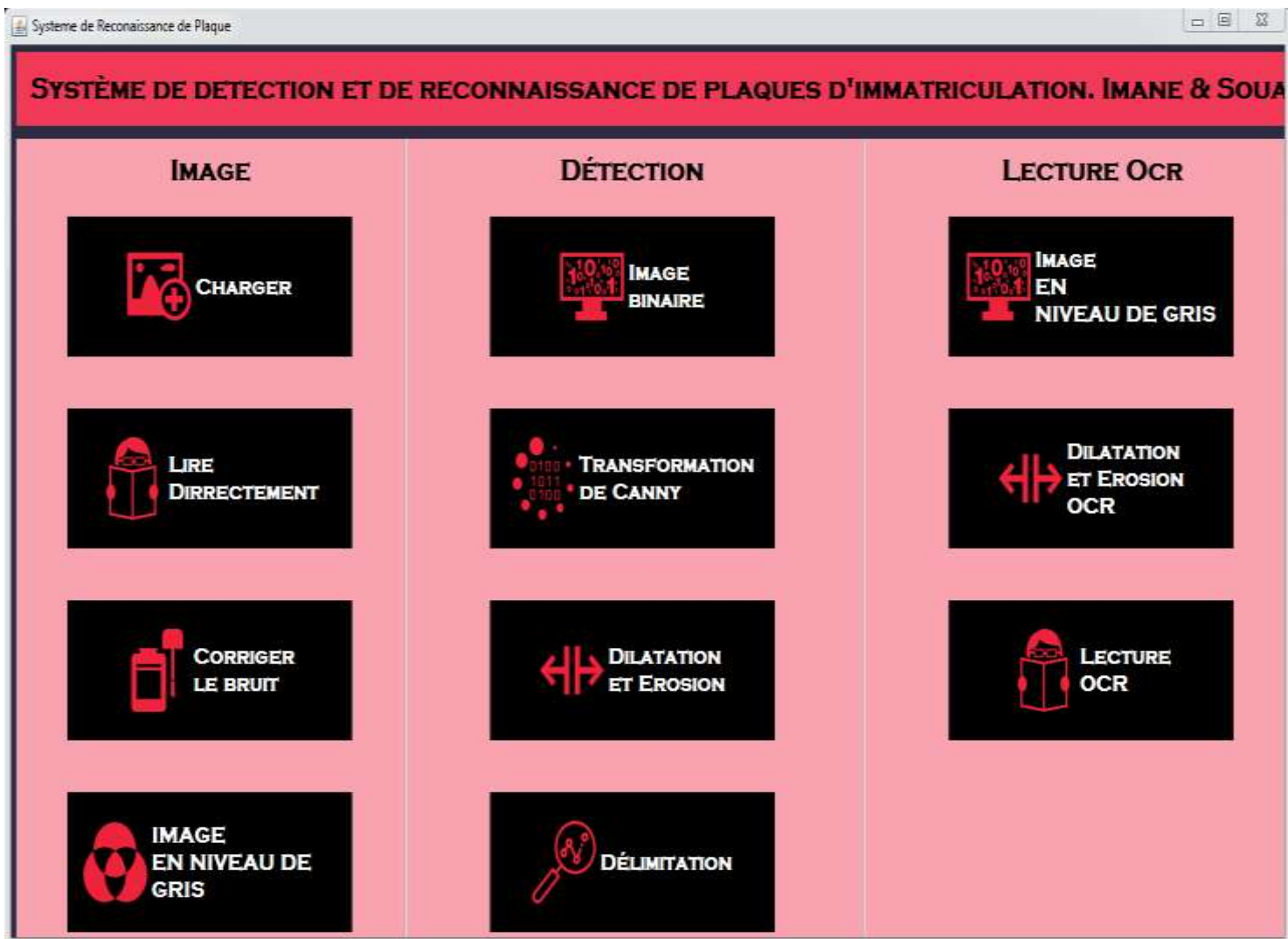


Figure 5 : Formulaire principale.

### 4.3.2 Phase de traitement d'image

On a quatre étapes dans cette phase :

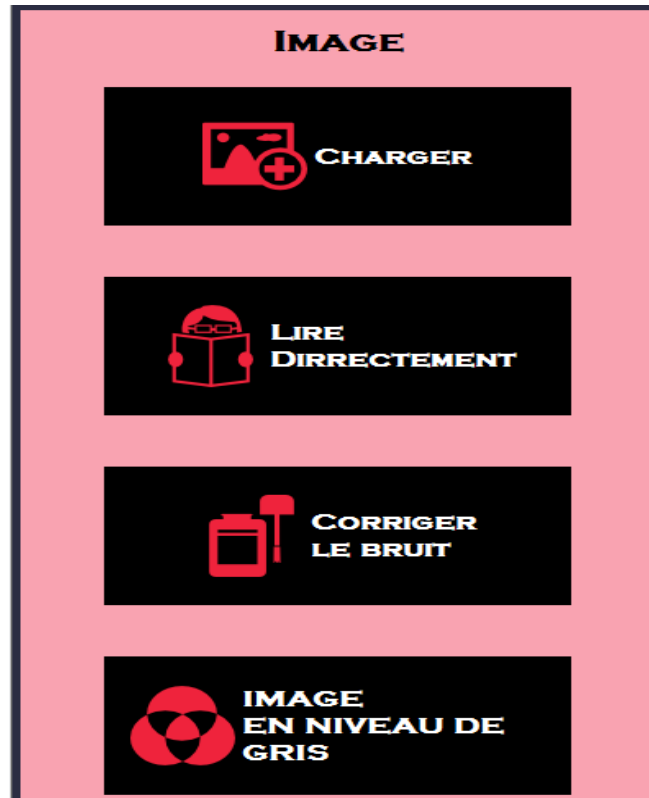


Figure 6 : Phase de traitement d'image.

Grace à cette phase, nous pouvons charger l'image, corriger le bruit et transformer l'image en niveau de gris, quand on clique sur le bouton «Charger», il aura accès à la fenêtre suivant :

### 4.3.2.1 Interface pour chargement de l'image



Figure 7 : Fenêtre chargement d'image.

### 4.3.2.2 Lecture direct

Dans cette étape ont essayé de lire une plaque normale directement mais le OCR donne un résultat erroné parce qu'il y'a encore trop de détails (bruits) qui rend la lecture impossible (voir fig.8).



Figure 8 : Lecture direct de la plaque.

### 4.3.2.3 Suppression ou diminution du bruit

Après chargement d'image, nous avons fait la correction de bruit et pour réaliser cette action, on clique sur le bouton «**Corriger le bruit**» et la fenêtre suivant s'affiche à l'écran.



**Figure 9 :** a-Image originale b-Image avec diminution du bruit.

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 afin de mieux convertir et détecter les couleurs et leurs bornes.

#### **4.3.2.4 Conversion de l'image en niveau de gris**

Après avoir réduit le TSL nous passons à une conversion de couleur d'un niveau RGB à un niveau de gris. Cela nous permet de réduire les nuances et préparer un passage plus sûr vers une image composée que du blanc et du noir.



Figure 10 : Fenêtre Conversion de l'image en niveau de gris.

### 4.3.3 Phase pour la détection des contours

Dans cette phase, nous nous sommes passés par quatre étapes :

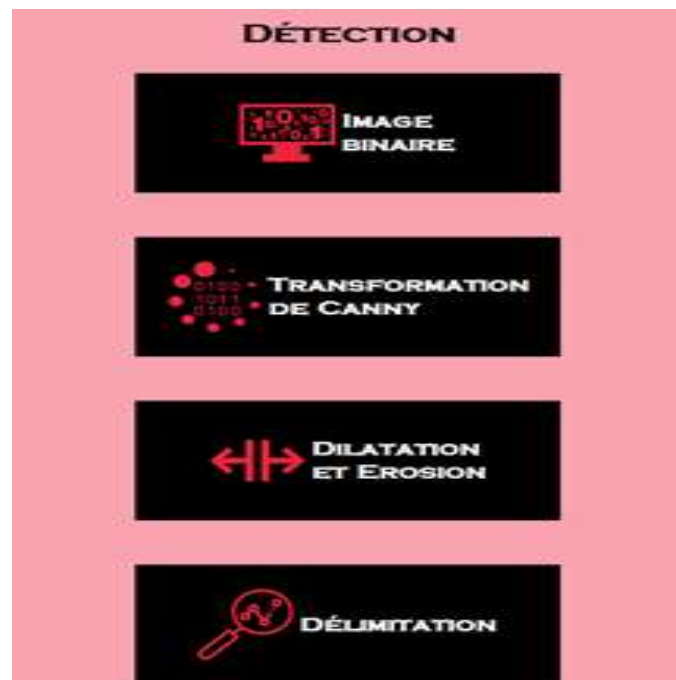


Figure 11 : Etapes de la détection.

Dans cette phase nous avons appliqué les opérations suivantes :

- Transformation en binaire
- Transformation de Canny
- Dilatation et Erosion
- Délimitation.

#### 4.3.3.1 Transformation de l'image en binaire

Quand on clique sur le bouton « **Image binaire** », la fenêtre suivante s'affiche à l'écran. (Fig.12).

Une fois l'image en gris obtenue, et afin de valider une sorte de détection de contour, nous commençons d'abord par générer une image binaire (noir et blanc). Cela nous permettra d'effectuer certain traitement nécessaire à la phase de détection de contour qui sera effectué par OpenCV.



Figure 12 : Fenêtre transformation en binaire.

#### 4.3.3.2 Transformation de Canny

La transformation de Canny est une transformation matricielle nécessaire à toutes fonctions de détection de contour, implémentée par OpenCV, de ce fait dans cette étape nous appliquons cette transformation afin de permettre à OpenCV de bien détecter les contours dans notre image finale. En effet Canny a la capacité de repérer et répertorier



toutes les lignes verticales et horizontales dans l'image, ce qui est nécessaire pour la détection de contour de forme rectangulaire.



**Figure 13:** Fenêtre transformation de Canny.

#### 4.3.3.3 Dilatation et érosion

Cette étape est justifiée vu que Canny nous fait perdre des détails. La dilatation et l'érosion ont pour objectif à réparer l'image en dilatant les lignes détectées par Canny, ce qui va réparer et faire mieux paraître nos rectangles dans l'image (Fig.14). Le traitement est achevée avec une érosion en motif rectangulaire, ce qui va réparer les coins des rectangles et supprimer Les point apparus dans l'image du a quelques restes de pixels blancs des étapes précédentes (Fig.15).



Figure 14 : fenêtre de dilatation.



Figure 15 : Fenêtre d'érosion.

#### 4.3.3.4 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 bien clairs grâce à la dilatation et bien détectables grâce à Canny. Nous ignorons les petits rectangles détectés, car une plaque a 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.



Figure 16 : Fenêtre de délimitation.

#### 4.3.4 Phase de lecture OCR

Trois étapes sont nécessaires dans cette phase :

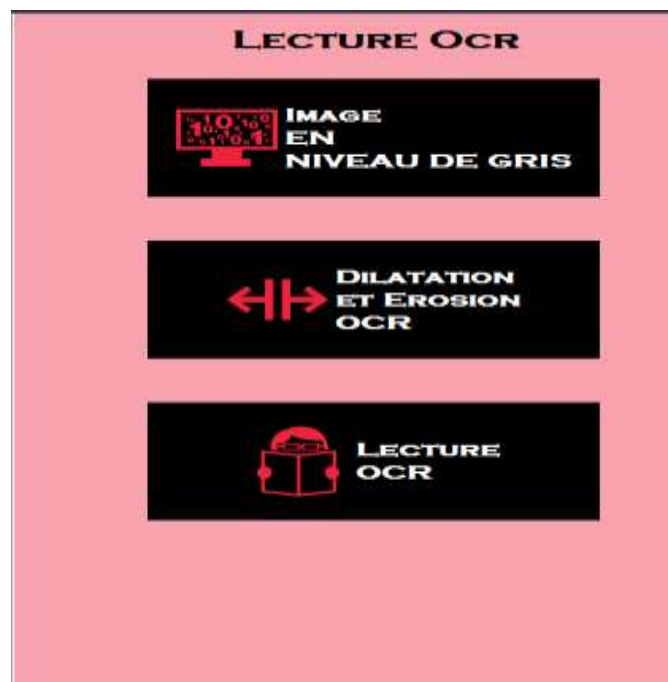


Figure 17 : Phase de lecture OCR.

Cette phase rassemble des techniques qui réalisent une reconnaissance de la forme des caractères :

- Image noir et blanc.
- Dilatation et érosion.
- Lecture OCR.

Quand on clique sur le bouton «**Image noir et blanc**», on aura accès à la fenêtre suivante :

#### 4.3.4.1 Conversion de l'image en niveau de gris OCR

Ayant notre délimitation, nous allons retravailler l'image originale. L'objectif est d'avoir une image d'arrière-plan blanc avec une écriture noire de bonne qualité. Pour cela, nous commençons par une conversion en gris avec une petite érosion d'un motif rond qui va améliorer les détails des chiffres circulaires.



**Figure 18** : Fenêtre Conversion de l'image en niveau de gris pour l'OCR.

#### 4.3.4.2 Dilatation et érosion pour OCR

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 idéal pour un meilleur résultat de lecture afin de maximiser nos chances d'avoir le bon matricule.



Figure 19 : Fenêtre de dilatation et érosion OCR.

#### 4.3.4.3 Lecture OCR

Finalement une lecture est effectuée et le résultat est affiché à droite. Les OCR ne sont pas si précis. Ce qui explique la mauvaise lecture du 4 en 1. Cependant, sans ces étapes, avoir un tel résultat serait impossible. Nous avons montré cela, dans l'étape de la lecture directe où l'OCR échoué.

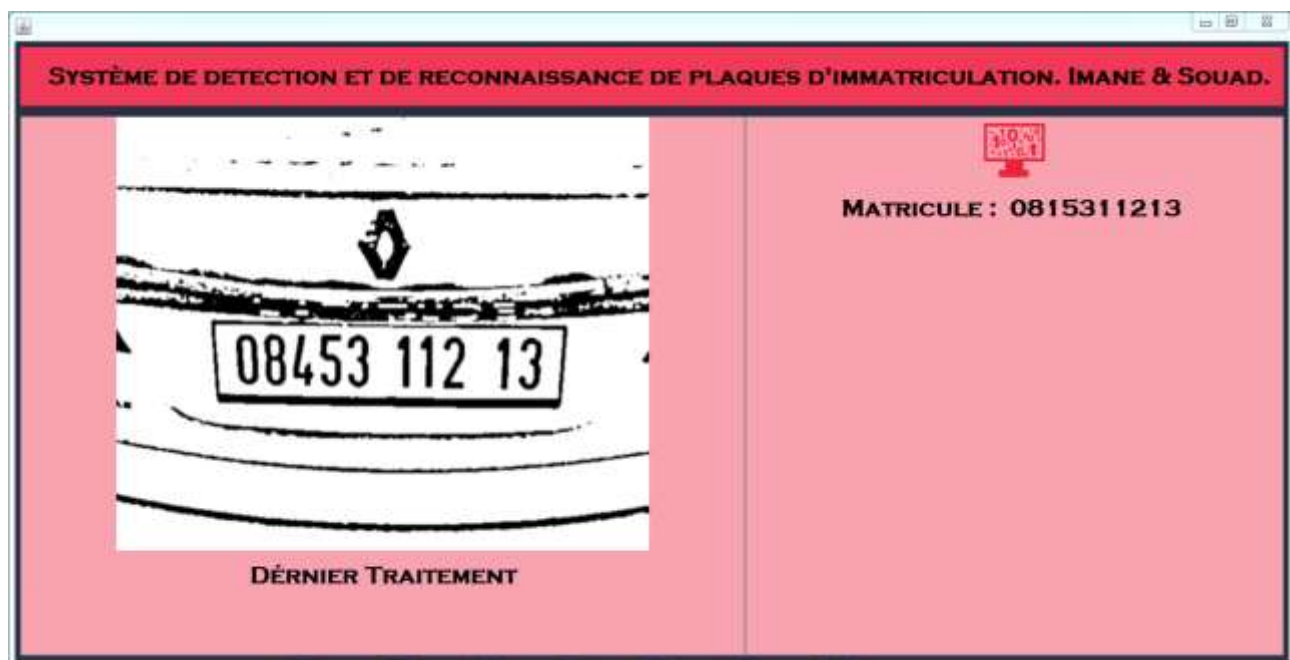


Figure 20 : Fenêtre de lecture OCR.



#### **4.4 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'objectifs escompté.

Nous avons réalisé notre application de lecture automatique de plaques d'immatriculations en utilisant le langage JAVA, et plusieurs librairies sous l'environnement de développement NetBeans.

# **CONCLUSION GENERALE**

## Conclusion générale

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

Après avoir présenté un état de l'art sur les systèmes existants, nous avons présenté notre système LAPIA : Lecture Automatique de Plaques d'Immatriculation Algériennes où nous avons réussi à détecter avec une grande précision les chiffres formant le matricule de du véhicule.

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.

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 souhaiterons améliorer les résultats obtenus en analysant les cas de défaillances et en spécifiant mieux l'OCR.



## Références Bibliographiques

- [1]** A. Bertrand, " LAPI ou VPI, quelles différences ? ", Association nationale de la vidéo protection, Vidéo Engineering, 16/08/2013-Lyon.
- [3]** A.NOBEL, "Reconnaissance automatique de plaques d'immatriculation (AutoVu) ", Genetec, 2013-Canada.
- [9]** H. Maître (ss la direction de), " Le traitement des images ", Hermes Lavoisier IC2 2003.
- [10]** C.F.CREATIS, " Traitement et analyse d'image-ANIMAG ", Institut national des sciences, Université Claude Bernard, 2013-Lyon.
- [11]** Cours en ligne disponible sur <http://www.imedias.pro/cours-en-ligne>.
- [12]** J.P.Gastellu-Etchegorry, "d'acquisition et traitement d'image numérique", IUT-Département de mesures physique, Université Paul Sabatier, Avril 2008.
- [13]** J.M.DOUDOUX, "Maven", Développons en Java, Copyright(c) 1999-2016.
- [14]** M.G Gauthier, "GitHub, u'est-ce que c'est ? ", OpenClassrooms, 12 décembre 2016-Europe.
- [15]** M.DALMAU, " Ecriture d'application Java avec swing ", IUT de Bayonne, 2013-Lyon.
- [16]** A.Calmettes, "Introduction à la vision par ordinateur", OpenClassrooms, 5 décembre 2013-Europe.

## Webographie

- [2]** " Lecture automatique de plaques minéralogiques ",  
[https://fr.wikipedia.org/wiki/Lecture\\_automatique\\_de\\_plaques\\_minéralogiques](https://fr.wikipedia.org/wiki/Lecture_automatique_de_plaques_minéralogiques), Dernière visite : 15/06/17.
- [4]** "LAPIEngine", <http://www.alphanumeric-vision.com/fr/logiciel-lapi/>, Dernière visite : 31/05/17.
- [5]** "Reconnaissance de plaques d'immatriculation avec LAPI de SeeTec", <https://www.seetec-video.com/fr/produits/extensions-seetec-cayuga/reconnaissance-des-plaques-dimmatriculation/>, Dernière visite : 31/05/17.
- [6]** "Asia Vision Technology Limited (AVT) ", <https://asiavisionweb.com/> Dernière visite : 03/06/17.
- [7]** "Nos logiciels - AFS2R", <http://afs2r.fr/nos-produits/>, Dernière visite : 31/05/17.
- [8]** "TEB - Système LAPI : Lecture Automatique de Plaque d'Immatriculation",<http://www.teb-online.com/fr/analyse-images/lapi-lecture-automatique-plaque-immatriculation.php>,  
Dernière visite : 02/06/17.
- [17]** " Tess4J - JNA wrapper for Tesseract - SourceForge",  
<http://tess4j.sourceforge.net/>, Dernière visite : 19/05/17.

## Résumé

Le but de ce projet est d'effectuer la détection et la lecture automatique de plaques d'immatriculation de véhicules automobiles 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.

Dans ce système, plusieurs prétraitements de filtrage de bruits, de transformations d'images et de détection ont été réalisés. Ensuite, des opérations de segmentation, de détection de contours et de reconnaissance de forme de caractères ont permis de répondre aux besoins exigés dans ce projet.

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

## Abstract

The aim of this project is to carry out the detection and the automatic reading of license plates of Algerian motor vehicles. For this purpose we have carried out a state of the art and a preliminary study in order to gather the information necessary to identify the different treatments to be implemented in this system. An interactive portable application has been developed and the results have been very satisfactory.

In this system, several pre-treatments of noise filtering, image transformations and detection have been carried out. Subsequent segmentation, contour detection and character recognition operations were carried out to meet the needs of the project.

**Keywords:** Automatic number plate recognition (ANPR), Canny, OCR, Segmentation.

## ملخص

الغرض من هذا المشروع هو إجراء الكشف والقراءة التلقائية للوحات المعدنية لسيارات الجزائر . لهذا قمنا بدراسة أولية لجمع المعلومات اللازمة لتحديد العلاجات المختلفة لتنفيذ هذا النظام . تم تطوير تطبيقات الهاتف المتحرك التفاعلي وكانت النتائج مرضية للغاية.

في هذا النظام، قدمنا عدة عمليات منها تصفية الصور واكتشاف التغيرات، ثم عمليات التجزئة، والكشف عن الحافة والتعرف على نمط الأحرف مما سمح لنا التكفل بجميع الاحتياجات المطلوبة من قبل موضوع هذا المشروع.

**الكلمات المفتاحية :** Automatic number plate recognition (ANPR), Canny, OCR, Segmentation