



Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique

MÉMOIRE DE MASTER

Sciences et Technologies
Electronique
Système Embarqués

Réf. :

Présenté et soutenu par :
Krim Smail

Le : Lundi 09 juillet 2019

Reconnaissance Automatique des plaques d'immatriculation

(R.A.P.I)

(Implémentation sur Raspberry Pi)

Jury :

Mr.	OUAFI ABDELKRIM	PR	Université de Biskra	Président
Mm.	TOUMI ABIDA	MCA	Université de Biskra	Examineur
Mm.	MDOUAKH SAADIA	MAA	Université de Biskra	Encadreur

Année universitaire : 2018 - 2019

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : électronique
Option : Systèmes Embarqués

Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme:

MASTER 2

Thème

**Reconnaissance Automatique des plaques
d'immatriculation
(R.A.P.I)
(Implémentation sur Raspberry Pi)**

**Présenté par :
KRIM SMAIL**

**Avis favorable de l'encadreur :
MEDOUAKH SAADIA**

**Avis favorable du Président du Jury
OUAFI ABDELKRIM
Cachet et signature**



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : électronique

Option : Systèmes Embarqués

Thème

Reconnaissance Automatique des plaques d'immatriculation (R.A.P.I) (Implémentation sur Raspberry Pi)

Proposé par : MEDOUAKH Saadia

Dirigé par : MEDOUAKH Saadia

RESUMES (Français et Arabe)

Résumé :

De nos jours il existe de nombreux systèmes de reconnaissance des plaques d'immatriculation en monde, ces systèmes ont deux axes majeurs, qui sont la détection de la plaque d'immatriculation et la reconnaissance de ses caractères. Cependant ces systèmes sont contrôlés par diverses conditions liées à la couleur des plaques connue d'avance, aux arrières plans fixes ... , et varient en terme d'extraction des caractéristiques. Par conséquent ce présent travail propose de concevoir et implémenter un système de reconnaissance des plaques d'immatriculation algérienne basé sur la détection de contours et les moteurs de reconnaissance de caractères. Le système implémenté, sur une « Raspberry Pi 3 », passe par trois étapes majeures ; prétraitements et amélioration, localisation de la plaque, et enfin la reconnaissance des caractères. Les résultats des expérimentations obtenus montrent que la méthode proposée achève des taux de reconnaissance acceptable malgré le non-respect des normes de construction des plaques d'immatriculation ce qui rend difficile de les identifier automatiquement.

Mots clés : Reconnaissance, Localisation, Segmentation, kNN, Plaques d'immatriculation, Contours, Raspberry Pi, OpenCV.

ملخص :

يوجد حاليا العديد من أنظمة التعرف على لوحات الترخيم في العالم ، وهذه الأنظمة تركز على محورين أساسيين، هما اكتشاف موضع لوحة الترخيم والتعرف بعد ذلك على أرقامها. للقيام بهذه العملية يتم عموما استعمال عدة خصائص من طرف هذه الأنظمة كالاعتماد على لون اللوحات المعروف مسبقا، الخلفيات الثابتة التي تثبت عليها اللوحات ... إلخ ، وتختلف من حيث الطرق المعتمدة لاستخراج الأرقام والتعرف عليها. لذلك، نقترح من خلال هذا العمل تصميم وتنفيذ نظام للتعرف على لوحات الترخيم الجزائرية معتمدين فيه على مبدأ استكشاف ودراسة حواف الأشياء وكذا محركات التعرف على الأحرف. النظام المنجز، والذي يشتغل على جهاز "Raspberry Pi 3"، يمر بثلاث خطوات رئيسية ؛ المعالجة والتحسين، توطين اللوحة، وأخيرا التعرف على الأرقام التي تحويها. تظهر نتائج التجارب التي تم الحصول عليها أن الطريقة المقترحة تحقق معدلات تعرف مقبولة على الرغم من عدم الامتثال لمعايير إنشاء لوحات الترخيم ما يصعب عملية التعرف عليها أوتوماتيكيا.

الكلمات المفتاحية : التعرف، تحديد الموقع، التجزئة ، kNN، لوحات الترخيم، الحواف، راسبيري باي، أوبن سي في

Remerciement

Tous d'abord je tiens à remercier **Dieu** tout puissant pour m'accorder la force et la patience de terminé ce modeste travail.

Ainsi que MDOUAKH SAADIA pour son aide et ses conseils tous au long de ce travail.

Je tien à remercier aussi les membres de jury pour accepter d'évaluer ce modeste travail, ainsi que tous les enseignants du département de génie électrique pour leur soutien durant toute l'année.

Sans oublier toute ma famille et mes collègues pour leur soutien inconditionnel.

Dédicaces

*A mes chers parents, pour tous leurs sacrifices, leur amour, leur
tendresse, leur soutien et leurs prières tout au long de mes études,
A mes chers frères et sœurs pour leurs encouragements permanents,
et leur soutien moral,
A toute ma famille pour leur soutien tout au long de mon parcours,
A tous mes amis et mes collègues de travail pour leur préoccupation,
et ceux qui ont partagé avec moi tous les moments d'émotion lors de
la réalisation de ce travail.
Que ce travail soit l'accomplissement de vos vœux tant allégués, et le
fruit de votre soutien infaillible,
Merci d'être toujours là pour moi.*

Table des matières

Introduction générale	1
Chapitre 01 : Généralités sur le traitement d'images et les techniques RAPI	
1. Introduction	4
2. Traitement d'images	4
3. Image	5
4. Représentation d'image	5
4.1. Image matricielle	5
4.2. Image vectorielle	5
5. Echantillonnage et quantification	6
6. Acquisition d'une image	7
7. Caractéristiques d'une image numérique	7
7.1. Dimension	7
7.2. Résolution	7
7.3. Bruit	7
7.4. Histogramme	8
7.5. Luminance	9
7.6. Contraste	9
7.7. Images à niveaux de gris	9
7.8. Image en couleurs	9
8. Principales étapes de traitement d'images	10
8.1. Prétraitements	10
8.2. Amélioration d'image	10
8.3. Analyse d'image	11
9. Types de données manipulées	11
10. Opérateurs de traitement d'image	12
10.1. Opérateurs locaux	13
10.2. Filtres linéaires	14
10.2.1. Lissage	15
10.2.2. Détection de contours	15
10.3. Opérateurs morphologiques	16
10.4. Construction d'une application de traitement d'images	17
11. Quelques exemples concrets de traitement d'image	18
12. Les techniques de reconnaissance des plaques d'immatriculation	19
12.1. Techniques basant sur les propriétés de la plaque	20
12.2. Techniques basant sur les contours de la plaque	21
12.3. Techniques basant sur l'intelligence artificielle	21
12.4. Techniques basant sur la signature de la plaque	22
13. Domaines et exemples d'utilisation des systèmes RAPI	22
13.1. Le péage urbain	23
13.1.1. Le péage urbain (ex Londres)	23
13.1.2. Le péage cordon de Stockholm	23
13.2. Surveillance du réseau	24
13.3. Gestion des véhicules prioritaires	24
13.4. Télépéage	25
14. Conclusion	26
Chapitre 02 : Raspberry Pi	
1. Introduction	28
2. Ordinateurs à carte unique (SBCs)	28
3. Généralités et historiques	28
4. Composants de base.....	32

5.	Exemples de projets réalisés avec Raspberry Pi	35
6.	Conclusion	41
Chapitre 03 : Conception du système RAPI		
1.	Introduction	43
2.	Caractéristiques des plaques algériennes	43
3.	Le système RAPI à implémenter	44
4.	Présentation de la démarche d'un RAPI	44
4.1.	Prétraitement et amélioration	45
4.2.	Localisation de la plaque	47
4.3.	Reconnaissance de la plaque	49
4.3.1.	Segmentation des caractères	49
4.3.2.	Reconnaissance de caractères	49
5.	Conclusion	51
Chapitre 04 : Implémentation du système RAPI		
1.	Introduction	53
2.	Aspect matériel	53
3.	Aspect logiciel	53
3.1.	Python	54
3.2.	OpenCV	55
3.3.	Algorithmes implémentés	56
3.4.	Description des fonctions Python et OpenCV utilisés pour l'implémentation	59
4.	Résultats et discussions	61
4.1.	Démarche de la technique RAPI	61
4.2.	Présentation Analyse des résultats	62
4.2.1	Résultats de localisation.....	62
4.2.2	Résultats de reconnaissance.....	66
4.3	Analyse des résultats et Discussions.....	74
4.3.1	Résultats de localisation.....	74
4.3.2	Résultats de reconnaissance.....	74
5.	Conclusion	76
Conclusion générale		77
Annexe		79
Bibliographie		84

Liste des figures

Chapitre 01 : Généralités sur le traitement d'images et les techniques RAPI

Figure 1.1 : Echantillonnage et Quantification d'un signal 2D continu selon différentes résolutions spatiales et colorimétriques. [DIA04]	6
Figure 1.2 : Exemples de bruits. [BER15]	8
Figure 1.3 : Exemple de la segmentation de régions (méthode de croissance de région). [BER15]	14
Figure 1.4 : Exemple de la segmentation de contours (avec un filtre de Sobel). [BER15]	14
Figure 1.5 : Exemple des opérateurs morphologiques. [BER15]	17
Figure 1.6 : Pips Technology : système de reconnaissance de plaque d'immatriculation. [LUD13] ...	23
Figure 1.7 : Portails du péage de Stockholm. [LUD13]	24
Figure 1.8 : Caméras 3G embarquées (WCCTV). [LUD13]	24
Figure 1.9 : Péage de la SR91, Californie. [LUD13]	25
Figure 1.10 : Portail de télépéage (Singapour). [LUD13]	25

Chapitre 02 : Raspberry Pi

Figure 2.1 : Les modèles de Raspberry Pi.	30
Figure 2.2 : Projet DoorJam. [PAU17]	36
Figure 2.3 : Projet du scanner 3D. [PAU17]	36
Figure 2.4 : Projet du miroir magique. [PAU17]	37
Figure 2.5 : Projet MintyPi. [PAU17]	38
Figure 2.6 : Projet de verrou intelligent. [PAU17]	39
Figure 2.7 : Projet Picrowave. [PAU17]	39
Figure 2.8 : Projet Pi-hole. [PAU17]	40
Figure 2.9 : Projet du drone semi-autonome. [PAU17]	40

Chapitre 03 : Conception du système RAPI

Figure 3.1 : Exemple de plaque d'immatriculation Algérienne	43
Figure 3.2 : Architecture du système RAPI à implémenter.	44
Figure 3.3 : Effet d'un filtre bilatéral, lissage sans perdre les contours. [HEB17]	46
Figure 3.4 : Le seuillage permet de convertir une image de niveaux de gris vers noir et blanc. [MAZ18]	47
Figure 3.5 : Détecteur de contours de Canny. [BER15]	48

Chapitre 04 : Implémentation du système RAPI

Figure 4.1 : Détection de la plaque d'immatriculation	62
Figure 4.2 : Ségmentation des chiffres de la plaque.....	62

Annexe :

Figure A.1 : Différence entre N-NN classifieur et 5-NN classifieur[BEZ18].....	79
Figure A.2 : Classification d'une nouvelle observation : les 03 plus proches voisins appartiennent à la classe 2[GUIW19].....	80
Figure A.3 : Classification d'une nouvelle observation : 02 des 03 plus proches voisins appartiennent à la classe 2[GUIW19].....	80

Liste des tableaux

Chapitre 02 : Raspberry Pi

Tableau 2.1 : Une comparaison entre un système à carte unique et un ordinateur.....	28
Tableau 2.2 : Les modèles de Raspberry Pi et leurs principales caractéristiques.....	34
Tableau 4.1 : Les statistiques de l'évaluation de la technique de localisation de la plaque	63
Tableau 4.2. : Présentation des cas de problemes de localisation de la plaque.....	63
Tableau 4.3 : Résultats obtenus depuis le module de reconnaissance.....	68

Introduction générale

Dans les dernières années, le nombre de voitures a augmenté, par le fait du besoin croissant de se déplacer. Chaque voiture est caractérisée par son numéro de plaque d'immatriculation. Il est nécessaire d'identifier les plaques d'immatriculation des véhicules pour plusieurs raisons tel que la sécurité.

L'identification des véhicules doit se faire de manière très efficace en utilisant des ressources limitées. L'un des meilleurs moyens est la Reconnaissance Automatique de la Plaque d'Immatriculation (RAPI).

Le numéro de plaque d'immatriculation est un code numérique utilisé pour identifier de manière unique les véhicules. La reconnaissance des plaques d'immatriculation utilise la technologie de traitement des images et de reconnaissance des caractères pour identifier les véhicules en lisant automatiquement les numéros de plaque d'immatriculation.

Le processus de reconnaissance automatique des plaques d'immatriculation (RAPI) peut avoir de nombreuses applications. Il est peut-être utilisé dans les parcs de stationnement, les postes de frontières et de péage, la surveillance des autoroutes, ... etc. L'objectif du modèle (RAPI) est l'identification automatique des voitures à partir de captures d'images prises sur le devant ou à l'arrière des véhicules. Un système de reconnaissance automatique des plaques d'immatriculation robuste et efficace devient de plus en plus une nécessité pour notre vie quotidienne.

Tout système RAPI va passer par quatre étapes [CHH09] : La première consiste à extraire à partir d'une image du véhicule, obtenu d'une caméra haute résolution, la plaque de la voiture, en se basant essentiellement sur une méthode de détection de contours. Cette plaque subit par la suite une segmentation en caractères afin d'être analysée par un système de reconnaissance optique de caractère (OCR) (**optical character recognition**). Afin d'arriver à un meilleur résultat le plus rapidement possible, il faut utiliser un ensemble de prétraitements comme les techniques de la morphologie mathématique.

Un système de reconnaissance automatique des plaques d'immatriculation est une technologie qui trouve son essence dans ces 20 dernières années dans le développement des techniques de traitement d'image ainsi que dans les OCR [CHH09].

De ce fait, une problématique peut être posée de la façon suivante :

- Quels sont les systèmes RAPI existants, et quel sont les avantages et les inconvénients de chaque système ? et quel est le système le plus approprié à notre cas d'étude ?

- Après avoir choisir un système à implanter, quelles sont les étapes à suivre pour qu'on puisse concevoir un système RAPI efficace et performant ?
- Quels sont les outils matériels et logiciels à exploiter pour faire implémenter un tel système ?

Pour arriver à répondre aux questions composant notre problématique on va organiser notre travail, en quatre chapitres de la façon suivante :

Un premier chapitre va nous mettre sur les rails du domaine d'analyse des images et plus précisément, par la suite, des techniques de reconnaissance des plaques d'immatriculation.

Le deuxième chapitre sera sacré à une présentation générale sur la plateforme Raspberry Pi, ainsi que les modèles publiés et les composants de base de chaque modèle. Par la suite, nous allons présenter quelques projets réalisés en utilisant des Raspberry Pi.

Dans le troisième chapitre, nous allons faire une conception pour le système de reconnaissance automatique des plaques d'immatriculation dont le but est de décomposer le système voulu en un ensemble de modules à détailler par la suite.

Le dernier chapitre va présenter une mise en œuvre du système de reconnaissance en utilisant la bibliothèque Open CV et le langage de programmation python. En fin du chapitre on va présenter les résultats obtenus avec une analyse et discussion des différents résultats.

Chapitre 01

Généralités sur le traitement d'images et les techniques

RAPI

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.

On désigne 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 allons aborder les notions de base nécessaires à la compréhension des techniques de traitement d'images. Ensuite, nous allons donner un aperçu sur les différentes techniques de traitement et d'analyse connues dans ce domaine. Enfin, on va citer quelques techniques de reconnaissance automatique des plaques d'immatriculation en tant que l'un des axes de recherche les plus connues en traitement d'image.

2. Traitement d'images

Le traitement d'images (*image processing*) 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 (Ex.: la création d'effets spéciaux au cinéma, l'amélioration de la qualité des photos en corrigeant les conditions de prise de vue, la détection de lettres ou de visages, l'identification de zones cancéreuses en imagerie médicale, la compression JPEG, ... etc). [MAZ18]

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 exemples 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.

3. Image

Une image est obtenue par transformation d'une scène réelle par un capteur, donc c'est 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. Pour un ordinateur, une image est un ensemble de pixels. Un pixel, c'est un élément d'image (*picture element*). [DIA04]

4. Représentation d'image

Les images numériques, destinées à être visualisées sur les écrans d'ordinateur, se divisent en 2 grandes classes : Les images matricielles et les images vectorielles. [BER15]

4.1. Image matricielle

Une image matricielle (image bitmap), est formée d'un assemblage de points ou de pixels. On parle sur des points lorsque ces images sont imprimées ou destinées à l'impression (photographies, publicités, cartes ... etc.) et on parle sur des pixels pour les images stockées sous forme « binaire » ou « numérique ».

L'image est composée d'une matrice (tableau) de points à plusieurs dimensions, chaque dimension représentant une dimension spatiale (hauteur, largeur, profondeur), temporelle (durée) ou autre (par exemple, un niveau de résolution). [MAZ18]

4.2. Image vectorielle

Une image vectorielle en informatique (ou image en mode trait), 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....

Par nature, un dessin vectoriel est dessiné à nouveau à chaque visualisation, ce qui engendre des calculs sur la machine. L'intérêt est de pouvoir redimensionner l'image à volonté sans aucun effet d'escalier. L'inconvénient est que pour atteindre une qualité

photoréaliste, il faut pouvoir disposer d'une puissance de calcul importante et de beaucoup de mémoire. L'avantage de ce type d'image est la possibilité de l'agrandir indéfiniment sans perdre la qualité initiale, ainsi qu'un faible encombrement. [MAZ18]

5. Echantillonnage et quantification

Le codage ou la représentation informatique d'une image implique sa numérisation. L'échantillonnage et la quantification ne sont pas limités à la numérisation d'une image. Ce sont des techniques que l'on utilise également sur des images déjà numérisées, afin de modifier la résolution (on rééchantillonne l'image) ou le nombre de couleurs utilisées (quantification). Donc, cette opération se fait dans deux espaces (Figure 1.1) [DIA04] :

- l'espace spatial où l'image (2D) est numérisée suivant l'axe des abscisses et des ordonnées : on parle d'échantillonnage. Les échantillons dans cet espace sont nommés pixels et leur nombre va constituer la définition de l'image.

- l'espace des couleurs où les différentes valeurs de luminosité que peut prendre un pixel sont numérisés pour représenter sa couleur et son intensité ; on parle de quantification. La précision dans cet espace dépend du nombre de bits sur lesquels on code la luminosité et est appelée profondeur de l'image.

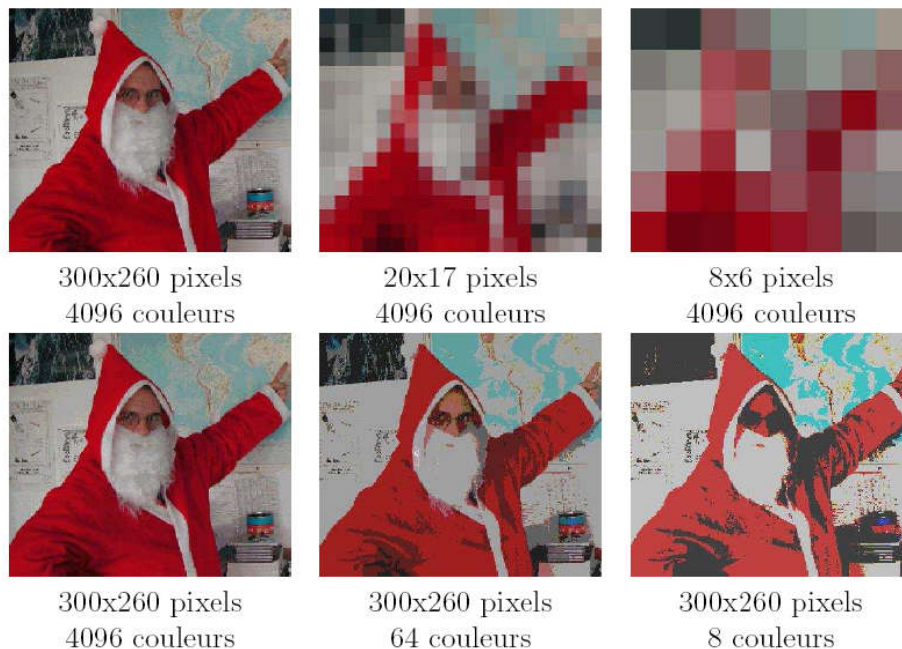


Figure 1.1 : Echantillonnage et Quantification d'un signal 2D continu selon différentes résolutions spatiales et colorimétriques. [DIA04]

6. 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 et les scanners.

7. Caractéristiques d'une image numérique

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

7.1. Dimension

La Dimension 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.

7.2. Résolution

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

7.3. 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.

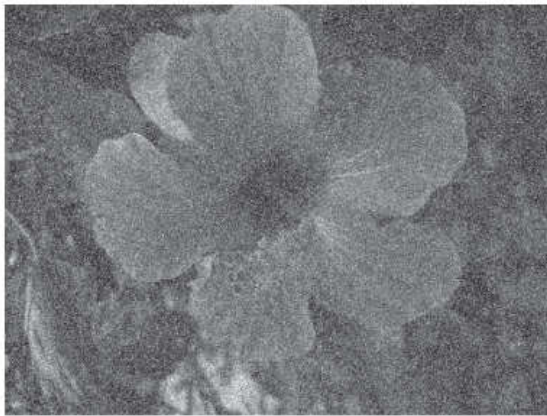
Visuellement, on distingue en général deux types de bruit d'image qui s'accumulent [BER15] :

- le bruit de chrominance, qui est la composante colorée des pixels bruités : il est visible sous la forme de tâches de couleurs aléatoires,

– le bruit de luminance, qui est la composante lumineuse des pixels bruités : il est visible sous la forme de tâches plus foncées ou plus claires donnant un aspect granuleux à l'image.



(a) Image originale

(b) Bruit blanc gaussien $\sigma = 0.2$ ($\simeq 50$ niveaux de gris)

(c) Bruit poivre et sel



(d) Bruit multiplicatif

Figure 1.2 : Exemples de bruits. [BER15]

7.4. 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. Donc, c'est la fonction discrète "**h**" telle que "**h(i) = n**", où **n** est le nombre de pixels de l'image ayant l'intensité **i**, que l'on appelle « effectif ». [MAZ18]

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.

7.5. Luminance

La luminance 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.

7.6. Contraste

Le contraste 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.

7.7. 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, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par un bit, mais par un octet. Pour cela, il faut que le matériel utilisé pour afficher l'image soit capable de produire les différents niveaux de gris correspondant.

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

7.8. 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.

On peut représenter les couleurs à l'aide de leurs composantes primaires. Les systèmes émettant de la lumière (écrans d'ordinateurs, ...) sont basés sur le principe de la synthèse additive : les couleurs sont composées d'un mélange de rouge, vert et bleu (modèle R.V.B).

8. 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 trois étapes majeures : [NOR17]

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

8.1. Prétraitements

Les prétraitements 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.

8.2. Amélioration d'image

L'amélioration d'image 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.

8.3. 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.

9. 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.

10. 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.

On a 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) : [WITR18]

- **Opérateurs image $\rightarrow$ 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 $\rightarrow$ ensemble d'informations**
- **Opérateurs de segmentation en frontières, en régions.**
- **Opérateurs de classification de pixels.**
- **Opérateurs de calcul de paramètres.**
- ***Opérateurs ensemble d'informations $\rightarrow$ 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.

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

On a 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 (figure 1.3).
- La segmentation de contour (figure 1.4.) : 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 (chaîne de Markov et champs de Markov) a permis de travailler en réduisant la connaissance a priori nécessaire pour obtenir un traitement satisfaisant.

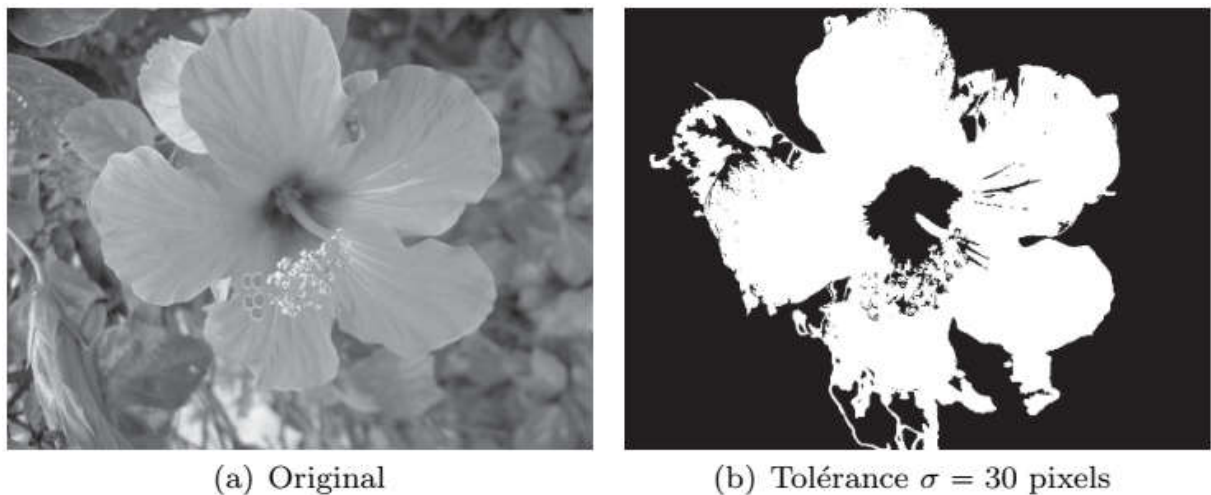


Figure 1.3 : Exemple de la segmentation de régions (méthode de croissance de région). [BER15]

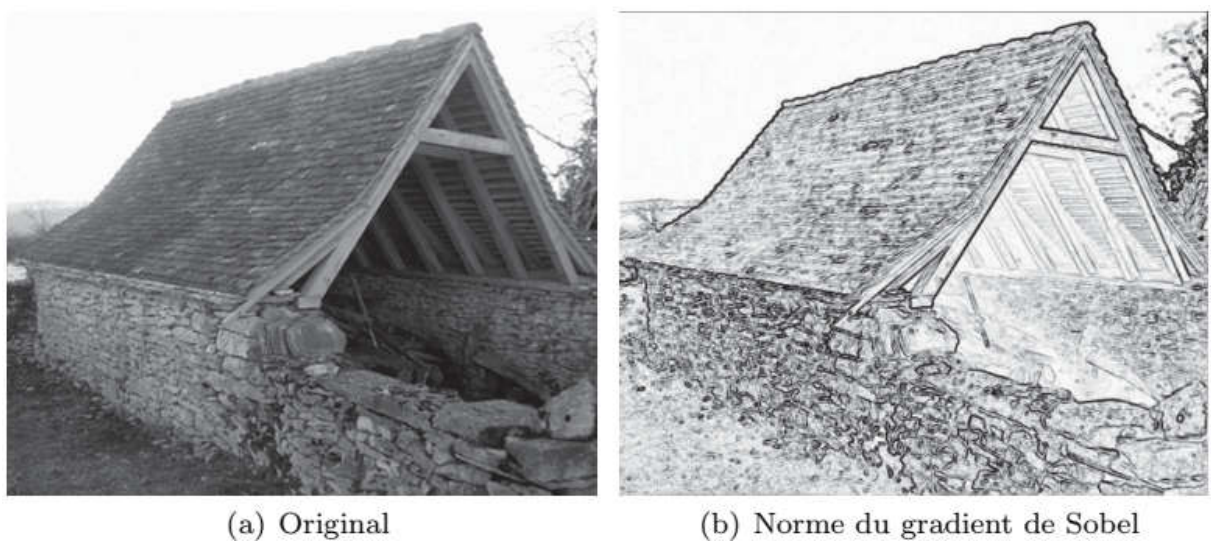


Figure 1.4 : Exemple de la segmentation de contours (avec un filtre de Sobel). [BER15]

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. [ETC08]

10.2.1. 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.

10.2.2. 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. [WITR18]

- **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.

10.3. 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 multispectrales. Parmi les opérateurs morphologiques on peut citer : Gradient morphologique, dilatation, érosion, ouverture et fermeture (voir la figure 1.5.).

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. [ETC08]

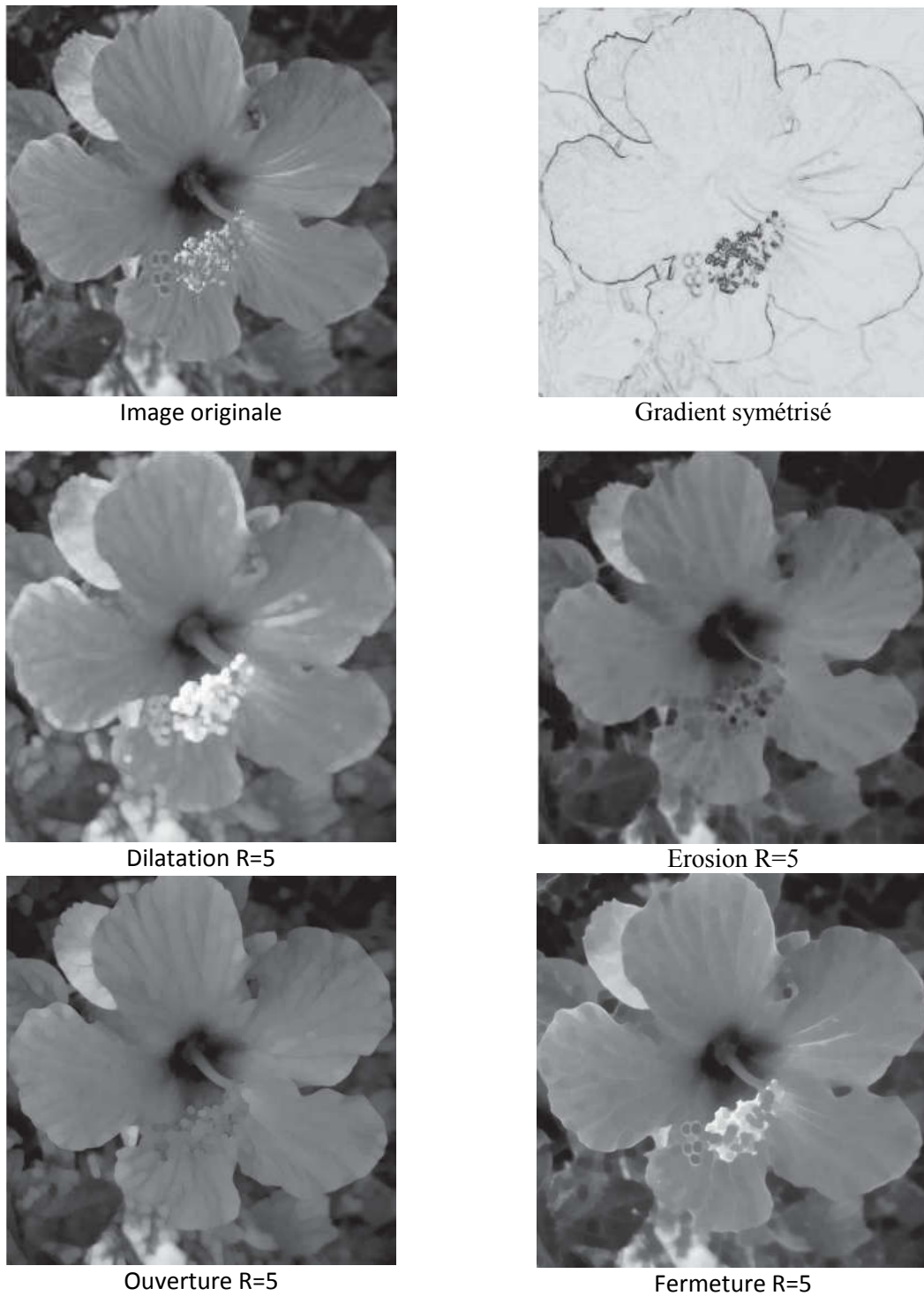


Figure 1.5 : Exemple des opérateurs morphologiques. [BER15]

10.4. 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.

Donc, 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. De nombreuses briques sont disponibles permettant de créer des applications complexes et évoluées.

11. Quelques exemples concrets de traitement d'image

On peut citer ici quelques exemples des cas utilisant le traitement d'images :

- **Contrôle de présence/absence** : Sur des chaînes de production, on a 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, par celles 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... On peut 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 a pour but l'analyse de 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. [WITR18]

Dans la partie qui suit on va essayer de détailler l'une des techniques utilisant le traitement d'images comme un outil d'analyse et d'aide à la décision ; La reconnaissance automatique des plaques d'immatriculation (RAPI).

12. Les techniques de reconnaissance des plaques d'immatriculation

Dans leur version la plus simple, les systèmes de RAPI utilisent, en cascade, deux techniques de traitements d'images [LUD13] : la détection d'objet pour repérer les plaques d'immatriculation potentielles suivie de la reconnaissance optique des caractères afin d'identifier les caractères alphanumériques de la plaque.

Pour obtenir des résultats satisfaisants, ce sont des séries de techniques de traitement d'image qui sont appliquées afin de détecter, normaliser et agrandir l'image de la plaque d'immatriculation, et enfin de pouvoir réaliser la reconnaissance optique de caractères (OCR) pour extraire les caractères alphanumériques de la plaque.

La fiabilité et la précision d'un système RAPI dépendent de la complexité de chacune de ces étapes. Dans quelques cas, la RAPI est couplée à une phase de détection/reconnaissance de véhicule afin de renforcer les performances de la détection de plaque. Afin d'accélérer et d'améliorer les performances de la phase de détection des plaques,

une technique courante est d'utiliser un a priori sur les zones potentielles de détection et de réduire la recherche à celles-ci. Ces zones sont déterminées par l'analyse des localisations des détections précédentes. Cette technique est appliquée dans le cadre des caméras ayant une prise de vue large (plusieurs voies). L'application de cette technique permet d'accélérer les traitements et donc d'augmenter le nombre de plaque lisible à la seconde. Cela est utile dans le cas où les dispositifs doivent travailler dans un environnement proche du temps réel tout en ayant une puissance de calcul modérée. Toutefois, elles s'exposent également au risque de ne pas détecter les véhicules qui ne passent pas dans les zones de pré-détection. C'est pourquoi elles ne sont pas appliquées dans le cas des dispositifs qui nécessitent d'identifier toutes les plaques à des fins de sécurité. Dans ces cas-là, soit le traitement est fait hors ligne, en temps différé, ou bien en temps réel mais avec un système plus couteux car nécessitant une puissance de calcul élevée.

La littérature nous dévoile un ensemble de méthodes qui ont été proposées pour la détection et la reconnaissance des plaques ; généralement on peut classer ces méthodes en quatre catégories [CHH09] :

12.1. Techniques basant sur les propriétés de la plaque

Pour *Anagnostopoulos* [JFC04], il utilise la couleur de la plaque comme une propriété déterminante, l'image passe par un filtre de couleur et la sortie est comparée à la forme de la plaque.

Wang et al [BAI04], utilisent un filtre spécial en plus du filtre de couleur. Ces filtres sont appliqués à une image au niveau de gris et la sortie contiendra les formes des caractères. Toutes les régions sont testées pour en extraire celles susceptibles d'être la plaque.

Pour *Beatriz et Broumandnia* [BEA04] et [BRO05], ils appliquent une méthode qui travaille avec des seuils afin de segmenter l'image en noir et blanc pour ensuite en extraire la plaque.

Comeli et al [CHO03], utilisaient un test se basant sur la taille des caractères et la distance qui existe entre eux. Ils ont eu jusqu'à 91.07% de réussite.

Les algorithmes basés sur les couleurs sont moins efficaces face à un changement d'éclairage car les couleurs apparaissent différemment sous différents éclairages et en plus les plaques ont plusieurs couleurs et plusieurs formes.

Avec ce type de méthodes on n'est pas sûr que les régions détectées vont correspondre à la plaque.

12.2. Techniques basant sur les contours de la plaque

Pour détecter la plaque, certains algorithmes utilisent les contours des caractères et de la plaque comme des points de référence pour l'extraction.

L'intensité des pixels dans les caractères et dans les contours des plaques est complètement différente que celle des voisins. En effet dans la majorité des cas ils sont d'une couleur différente que celle du reste de l'image. Une arête peut être détectée dans une image en utilisant le gradient comme par exemple avec le seuil OTSU.

La seconde partie de la détection localise les contours de la plaque à partir d'une image en noir et blanc.

Pour *Dai* et al. et *David* et al [DAI01] et [DVA05]. ils ont utilisé des opérations morphologiques sur l'image transformée en contours.

Pour *Dlagnekov* et *Elliman* [DLA02] et [ELL90] , on applique une transformée de Hough sur l'image et les lignes qui traversent la plaque seront déterminées ainsi que les objets de formes rectangulaires.

Anagnostopoulos [JFC86] a développé une méthode qui décrit les irrégularités dans la plaque. Cette méthode ressemble à la méthode précédente parce qu'elle dépend de l'intensité des pixels.

Les algorithmes présentés sont défaillants lorsque les bords de la plaque ne présentent pas une grande variation par rapport au reste de l'image en plus ces algorithmes utilisent un seuil qui doit être déterminé automatiquement ce qui est difficile sous différentes conditions d'éclairage.

12.3. Techniques basant sur l'intelligence artificielle

Nijhuis et al [JSE82]. ont utilisé les caractéristiques de la plaque allemande qui se compose d'un arrière-plan jaune pour créer une fonction pour chaque pixel à travers un histogramme. Les autres propriétés floues sont dégagées à partir du niveau de gris de chaque pixel et de ses voisins.

Zimic et al [JXI05]. ont appliqué le même concept de la logique floue. Ils ont divisé l'image en un ensemble de rectangles de la taille de la plaque.

Les auteurs de *Matas* [MAT05] ont réussi à localiser la plaque en utilisant la logique floue avec un pourcentage de réussite de 97.9%.

Néanmoins cette méthode dépend de l'éclairage parce qu'elle est tributaire de la couleur de la plaque.

Park et al [NIJ95]. ont utilisé un réseau de neurones à deux entrées comme un filtre horizontal et vertical pour détecter les plaques coréennes. L'intersection entre les deux filtres localise la région de la plaque.

Chacon et Zimmermann [OTS79] ont utilisé un réseau de neurones pour déterminer les régions susceptibles d'être la plaque qu'on va analyser par la suite par une transformé de Fourier pour détecter la bonne région, l'algorithme sera répété jusqu'à la détermination de la plaque. Ils ont réussi à extraire 85% des plaques ainsi la réussite de ce type d'algorithme dépend de leur capacité à s'adapter aux différents éclairages car l'entrée des réseaux est la valeur de niveau de gris pixel.

12.4. Techniques basant sur la signature de la plaque

Dlagnekova [PAM02] implémenté l'algorithme d'Adaboost comme on l'a utilisé pour la détection des visage (*Viola and Jones*).

Matlas et Zemmerman [PAR96] proposent un algorithme qui choisit une région du texte à partir d'un ensemble puis il exploite le fait qu'une plaque contient des caractères et des symboles qui sont clairement visibles même lorsque plusieurs propriétés de la plaque sont cachées. Cependant le problème est que certaines autres régions outre que plaque peuvent contenir du texte.

Elliman et Lancaster [SHY04] ont fait évoluer l'utilisation du spectre de fourrier pour distinguer entre les différentes régions de l'image qui présentent des informations. *Broumandnia et al* ont parcouru l'image verticalement et ont distingué le nombre d'arêtes pour chaque parcourt pour localiser la ligne horizontale à laquelle est lié la plaque.

13. Domaines et exemples d'utilisation des systèmes RAPI

Certains systèmes utilisés pour la LAPI, notamment le capteur vidéo, peuvent être utilisés pour d'autres applications comme [LUD13] : la surveillance et la DAI, le recueil de données (débit, vitesse, taux d'occupation), et gestion dynamique du trafic (régulation d'accès, régulation de carrefours à feux ...), la surveillance de sites et de zones, les passages de frontière, les stations-service (enregistrement quand un client part sans payer), le contrôle des entrées et sorties de parkings, de péages, ouverture automatique, la lutte anticriminalité en comparant les plaques d'immatriculations au Fichier des Véhicules Volés (FVV) ou à la liste des propriétaires de véhicules suspects, la traçabilité des parcours individuels afin d'alimenter

les modèles de trafic (microscopique, mésoscopique, et macroscopique)...

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

Plusieurs solutions de gestion de trafic ont déjà été éprouvées à l'étranger. Cette liste, non exhaustive, permet d'illustrer la variété des applications en lien avec les caméras vidéo.

13.1. Le péage urbain

13.1.1. Le péage urbain (ex Londres)

Le système du péage urbain de Londres fonctionne grâce à l'analyse des plaques d'immatriculation enregistrées par un réseau d'environ 800 caméras de surveillance disposées en entrée et sortie de la zone de péage de la ville. Il compare ensuite les données recueillies à une base de données centralisée pour identifier le propriétaire du véhicule et vérifier dans les délais son paiement effectif. Ce péage a permis notamment une réduction de circulation d'environ 20% dans le centre de Londres.



Figure 1.6. : Pips Technology : système de reconnaissance de plaque d'immatriculation. [LUD13]

13.1.2. Le péage cordon de Stockholm

La direction des routes de Suède a mis en place un système de péage autour de Stockholm.

L'identification des véhicules se fait de deux manières :

- par photographie des plaques d'immatriculation ;
- par lecture d'un badge électronique pour les véhicules équipés.

Les équipements étaient installés sur trois portiques :

- le premier portique indiquait par panneau l'entrée du point de contrôle et le prix à payer ainsi que des caméras pour l'enregistrement des plaques arrières du véhicules;

- le deuxième portique était équipé de récepteurs permettant la lecture des badges à bord du véhicule ;
- le troisième portique équipé de caméras enregistrait les plaques avant du véhicule.



Figure 1.7. : Portails du péage de Stockholm. [LUD13]

13.2. Surveillance du réseau

Les caméras peuvent être employées pour une vidéo-surveillance du réseau routier ; ces systèmes peuvent être installés en poste fixe ou bien mobile (embarqué sur véhicule de patrouille). Ils comportent également des moyens de communication permettant de fournir des images en direct à distance permettant une prise de décision optimisée.



Figure 1.8. : Caméras 3G embarquées (WCCTV). [LUD13]

13.3. Gestion des véhicules prioritaires

Les Etats Unis ont aménagé sur leurs infrastructures routières des voies pour bus et co-voiturage appelée « High Occupancy Vehicle » (HOV) ainsi que des voies appelées « High Occupancy Toll » (HOT) autorisant l'accès sur la voie de véhicules avec un seul passager à bord mais s'acquittant d'un péage modulé en fonction de la fréquentation de cette voie.

En Californie sur la SR 91 (figure 1.5), le dispositif permettant de contrôler l'accès à

cette voie spécialisée est composée d'un portique équipé d'une antenne de lecture RFID et de 3 caméras dont une utilisée en lecture automatique des plaques d'immatriculation. De plus, sur le trajet de 17 km de long, plusieurs dizaines de caméras sont également utilisées pour la détection d'incidents.



Figure 1.9. : Péage de la SR91, Californie. [LUD13]

13.4. Télépéage

Le télépéage permet aux véhicules de franchir les péages à la vitesse normale du trafic, réduisant ainsi la congestion aux zones de péage (ex : Singapour). Techniquement, le télépéage appelé « Electronic Road Pricing » (ERP) fonctionne à l'aide d'un boîtier de lecture dans le véhicule, d'une « cash card » sorte de carte de péage rechargeable, des portails d'accès (cf. photo ci-dessus) et d'une centrale informatique. Si la « cash card » est vide ou absente du véhicule, un système LAPI enregistre le numéro d'immatriculation.



Figure 1.10. : Portail de télépéage (Singapour). [LUD13]

14. Conclusion

Nous avons introduit dans ce chapitre les notions de base qui représentent l'image et ses caractéristiques puis la compréhension de différentes techniques de traitement d'images et leurs opérateurs. Par la suite nous avons fait un tour sur les techniques de reconnaissance automatiques des plaques d'immatriculation avec une présentation de quelques exemples d'utilisation des systèmes RAPI.

Dans le chapitre suivant, nous allons présenter certains outils matériels que nous allons utiliser pour réaliser un système RAPI.

Chapitre 02

Raspberry Pi

1. Introduction

Nous allons penser à intégrer un système RAPI dans un système embarqué, pour cela nous avons choisi le Raspberry Pi (un ordinateur à carte unique) pour l'implémentation. De ce fait, ce chapitre fournit des informations générales sur le RaspberryPi. Nous commençons par des informations générales telles que les modèles publiés et leurs composants de base. Il fournit également des informations sur l'historique de cet ordinateur à carte unique, ainsi qu'un détail de son système sur puce (SoC), notamment CPU, GPU et de la RAM.

2. Ordinateurs à carte unique (SBCs)

Un ordinateur à carte unique (*Single Board Computers*) est un système de calcul fonctionnel complet construits autour d'une carte à circuit imprimée. Un tel système possède un (des) micro-processeur(s), mémoire, entrées/sorties et d'autres composants nécessaires au fonctionnement minimal d'un ordinateur. Tous les composants dans un tel système sont intégrés sur la carte, donc ils ne peuvent pas être mis-à-jours. Les différences entre un SBCs et un ordinateur normal sont cités dans le tableau suivant [PAJ17] :

Tableau 2.1 : Une comparaison entre un système à carte unique et un ordinateur.

Un ordinateur à carte unique	Un ordinateur normal
Non modulaire	Modulaire
Les composants ne peuvent pas être mis-à-jours ou remplacés	Les composants ne peuvent pas être mis-à-jours ou remplacés
Le système est chargé sur puce	Le système n'est pas chargé sur puce
Taille petite	Taille grande
Portable	Non portable ou semi-portable
Consomme peu d'énergie	Consomme beaucoup d'énergie
Moins cher	Plus cher

Parmi les systèmes SBCs existants on peut trouver le Raspberry Pi, qu'on va détailler leurs caractéristiques dans ce qui suit.

3. Généralités et historiques

A la taille d'une carte de crédit, Raspberry Pi, est un nano-ordinateur mono carte à processeur ARM d'une petite taille que fonctionne sous Linux et qui peut être interfacé avec des composants électronique externes. De ce fait, elle intègre des puissances de calcul considérable. Elle est développée au Royaume Unie par la fondation Raspberry Pi créé en 2009. [PAJ17, UPT17, BEN18]

L'idée de la production de Raspberry Pi a commencé en 2006 avec une prise de conscience du fait que la jeune génération manque de connaissances sur le fonctionnement de l'ordinateur. Un groupe d'universitaires et d'ingénieurs de l'Université de Cambridge a donc décidé de développer un très petit ordinateur que tout le monde pourrait se permettre d'acheter pour créer un environnement d'apprentissage dans la programmation : **EbenUpton** a alors décidé de créer un ordinateur à faible prix, car il supposait que les parents craignaient de laisser leurs enfants jouer avec des PC modernes relativement coûteux. C'est décemment l'idée qu'est partie le développement du Raspberry Pi très bon marché. Cet ordinateur devait donner aux jeunes l'opportunité d'apprendre et de pratiquer la programmation, sans que leurs parents ne s'inquiètent des dégâts éventuels sur la machine. [BEN18]

Le projet Raspberry Pi est devenu prometteur avec l'apparition de processeurs mobiles économiques et puissants dotés de nombreuses fonctionnalités avancées permettant un développement possible de Raspberry Pi, qui s'est poursuivi avec la fondation spécialement créée Raspberry Pi avec le premier produit lancé en 2012.

Il est fourni nu, c'est-à-dire la carte mère seule, sans boîtier, alimentation, clavier, souris ni écran, dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération.

Le développement de l'ordinateur Raspberry Pi est un processus continu. Jusqu'au Novembre 2018, la fondation Raspberry Pi a publié onze modèles sur trois générations d'ordinateur [WIRA19].

Les modèles sont des versions mises à niveau de leurs versions précédentes pour rendre l'ordinateur plus efficace et pratique pour les utilisateurs, notamment en raison de sa faible consommation énergétique et de ses ports USB (Universal Serial Bus).

Les différences entre ces modèles sont la taille du GPIO, le nombre de ports USB et la présence ou non d'Ethernet, le support de carte microSD et la taille de la RAM.

La première génération est composée de quatre modèles. Le premier modèle publié c'est le 1B, suivi par le modèle 1A, suivi des modèles 1B + et 1A +, respectivement. Les deux derniers modèles sont des versions mises à niveau de leurs versions précédentes pour rendre l'ordinateur plus efficace et plus pratique pour les utilisateurs.

La deuxième génération comprend un seul modèle appelé Raspberry Pi 2 modèle B, dont les spécifications sont basées sur le modèle Raspberry Pi B + mais avec un processeur plus rapide et plus de mémoire.

La troisième génération comprend trois modèles appelés Pi 3 modèle B, et Pi 3 modèle B+, et Pi 3 modèle A+ dont les spécifications sont basées sur le modèle Raspberry Pi 2B mais avec un processeur plus rapide.

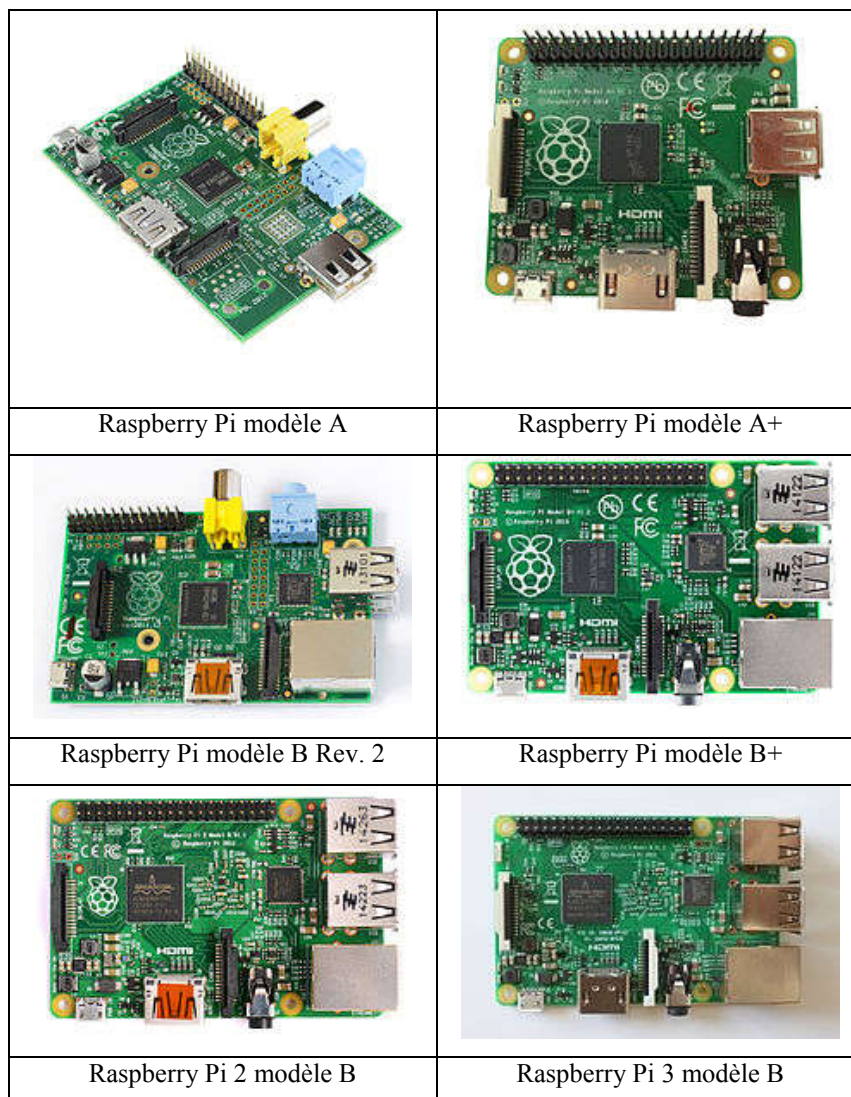


Figure 2.1 : Les modèles de Raspberry Pi.

Raspberry Pi Model 1B, est sorti en début 2012 avec la spécification de 256 Mo de RAM, deux ports USB et un port Ethernet. Plus tard dans la même année, dans une nouvelle version, la quantité de RAM augmentée à 512 Mo, suivie de la sortie de la carte de spécifications inférieures, le modèle 1A a été libéré avec la même quantité de RAM que l'ancien modèle B à 256 Mo, mais avec un port USB et pas de port Ethernet. Ces ordinateurs de première génération utilisent le SoC Broadcom, BCM2835, qui intègre un processeur ARM1176JZF-S mono cœur de 700 MHz, un processeur graphique Vidéo-Core IV et divers périphériques. Bien que le modèle 1B puisse être utilisé dans toutes les applications, le modèle A moins cher est utile dans des applications spécifiques nécessitant une consommation légère et faible, telles que la robotique ou tout service multimédia portable.

En 2014, une version mise à niveau des deux modèles précédents a été lancée, sous les noms 1B + et 1A +. Bien que les deux nouveaux modèles disposent des mêmes CPU, GPU et RAM que leurs versions précédentes, plusieurs mises à niveau et nouvelles fonctionnalités ont été ajoutées, telles qu'un support de mémoire micro SD remplaçant un emplacement pour carte SD, une version mise à niveau de GPIO de 26 à 40 broches, un son à faible bruit et une consommation d'énergie réduite de 0,5 à 1 W grâce à un régulateur de commutation au lieu d'un régulateur linéaire. En outre, le modèle 1A + est environ 2 cm plus court que son prédécesseur, tandis que le modèle 1B + fournit quatre ports USB au lieu de deux dans le modèle 1B.

Au début de 2015, la nouvelle génération d'ordinateur Raspberry Pi, appelée Raspberry Pi 2 modèle B, est sortie. Le modèle représente une mise à niveau importante des capacités de Raspberry Pi et a été accueilli avec enthousiasme par la communauté. Raspberry Pi 2 modèle B utilise un BCM2836 beaucoup plus puissant qui offre un processeur ARM Cortex-A7 quadri-core à 900 MHz, 1 Go de RAM et les mêmes spécifications de processeur graphique que précédemment, ce qui le rend environ six fois plus rapide que ses prédécesseurs sans changement prix. De plus, les supports de la version Internet des objets (IdO) de Windows 10 sont disponibles gratuitement pour le Raspberry Pi 2 modèle B, offrant ainsi un grand potentiel pour une utilisation future.

En février 2016, le Raspberry Pi 3B est le début de la troisième génération de Raspberry Pi. Il remplaçait le Raspberry Pi 2 Model B et comportait quatre modifications fondamentales: le 900MHz de processeur augmenté à 1,2 GHz et avec quelques nouvelles fonctionnalités de connexion sans fil 802.11n LAN sans fil Bluetooth 4.1 Bluetooth Low Energy (BLE).

En mars 2018 le Raspberry Pi 3 modèle B+, est sortie avec un changement du précédent de 1.4GMHz de processeur au lieu de 1.2GHZ, suivi en novembre 2018 de la création du Raspberry Pi 3 modèle A+, avec quelques modifications du model précédent de moins de mémoire 512Mo, et un seul port USB, et moins chère que 3B+.

Il existe un dernier modèle qui en fait n'appartient pas vraiment à une des générations à proprement parlé.

Le 26 novembre 2015, la fondation Raspberry Pi annonce la sortie du Raspberry Pi Zero. Il reprend les spécifications du modèle A/B avec un processeur cadencé à 1 GHz au lieu de 700 MHz, il est cependant plus petit, disposant d'une connectique minimale. Son prix de 5 \$ US est largement revu à la baisse par rapport aux autres Raspberry Pi.

On peut dire que c'est une sorte d'ovni dans l'horizon Raspberry, C'est le plus petit et le moins cher de la famille. Il n'appartient à aucune génération parce qu'il est en fait une version light du Rpi1. Ce modèle est une bonne occasion de tester le Raspberry et ses fonctionnalités à prix réduit. Il saura ravir les plus curieux malgré son cruel manque de puissance, ce n'est pas comme un vrai Rpi car il n'a pas d'Ethernet, mais il est parfaitement adapté pour remplir le but original de la Raspberry Fondation : lutter contre la fracture numérique et apporter le savoir numérique à moindre coût.

Le 28 février 2017, pour le cinquième anniversaire du Raspberry Pi, le Raspberry Pi Zero W est maintenant doté de Wi-Fi et de Bluetooth. Ces nouveautés lui permettent de se connecter à Internet et à d'autres appareils et donc d'en multiplier les usages. Le Raspberry Pi Zero, lui, n'avait pas de connexion Internet et cela pouvait poser des problèmes pour certains projets.

Ce nouveau modèle est commercialisé 10\$ à sa sortie soit deux fois plus cher que le modèle précédent.

Le 15 janvier 2018, Raspberry Pi sort un nouveau modèle Zero, le Raspberry pi Zero WH. La seule différence avec le précédent modèle est la présence de ports GPIO soudés sur la Raspberry Pi.

Bien qu'un nouveau modèle Raspberry Pi offre davantage de fonctionnalités, il est préférable de connaître les applications ou les projets pour lesquels la carte doit être utilisée, afin de sélectionner la solution la plus efficace en termes de consommation et de coût.

Fin 2012, le code de pilote GPU de Raspberry Pi fonctionnant sur ARM a été publié en tant que source ouverte et disponible au téléchargement. Puis, en 2014, la documentation complète au niveau du registre, comprenant une pile de pilotes graphiques du chi p, a été publiée publiquement.

4. Composants de base

Les détails des composants de base d'une Raspberry Pi sont expliqués ci-dessous. Bien que la liste soit basée sur la seconde génération de la carte, les détails des composants de la première génération sont également expliqués s'ils sont différents.

Spécifications matérielles et architectures

Le Raspberry Pi possède un **processeur ARM11** à 700 MHz. Il inclut 1, 2 ou 4 ports USB, un port RJ45 et 256 Mo de mémoire vive pour le modèle d'origine (1 Go sur les

dernières versions). Son circuit graphique **BMC Vidéo Core 4** permet de décoder des flux **Blu-Ray full HD** (1080p 30 images par seconde), d'émuler d'anciennes consoles et d'exécuter des jeux vidéo relativement récents.

Le tableau 2.2 résume les principales caractéristiques des modèles Raspberry Pi:

Tableau 2.2 : Les modèles de Raspberry Pi et leurs principales caractéristiques.

	Modèle 1B	Modèle 1A	Modèle 1B+	Modèle 1A+	Modèle 2B	Modèle Zero	Modèle 3B	Modèle Zero W	Modèle Zero WH	Modèle 3B+	Modèle 3A+
Date de lancement	Avril 2012	Fevrier 2013	Juillet 2014	Novembre 2014	Fevrier 2015	Novembre 2015	Fevrier 2016	Fevrier 2017	Janvier 2018	Mars 2018	Novembre 2018
SoC	BroadcomBCM2835(CPU.GPU.DSP.SDRAM.port USB)				Broadcom BCM2836	Broadcom BCM2835	Broadcom BCM2837	Broadcom BCM2835		Broadcom BCM2837B0	
CPU	700 MHZ ARM 1176JZF-S core(ARM11)				900 MHZ quadricoeur ARM Cortex-A7(jeu d'instructions ARM v7	01 GHZ ARM 1176JZF-S core(ARM11)	1.2 GHZ quadricoeur ARM Cortex-A53	700 MHZ ARM 1176JZF-S core(ARM11)		1.4 GHZ quadricoeur ARM Cortex-A53	
GPU	BroadcomVideocoreIV, Open GL ES 2.0, MPEG-2 et VC-1 avec licence, 1080p30 h.264/MPEG-4 high-profile décodeur et compresseur										
Memoir SDRAM	512 Mo	256 Mo	512 Mo	256 Mo	01 Go	512 Mo	01 Go	512 Mo		01 Go	512 Mo
Nbr de ports USB 2.0	02	01	04	01	04	01	04	01 (micro USB)		04	01
Sortie Vidéos	HDMI et composite (via prise RCA)		HDMI et composite (via connecteur JACK)			Mini HDMI et composite (via soudure)	HDMI et composite (via connecteur JACK)	Mini HDMI et composite (via soudure)		HDMI et composite (via connecteur JACK)	
Sortie Audio	Stereo jack 3.5 mm (sortie son 5.1 sur la prise HDMI) et composite					Mini HDMI	Stereo jack 3.5 mm (sortie son 5.1 sur la prise HDMI) et composite	Mini HDMI		Stereo jack 3.5 mm (sortie son 5.1 sur la prise HDMI) et composite	
Unité de lecture/écriture	SD/MMC/fente pour carte SDIO(3.3v)		Micro SD								
Carte connectionreseau	10/100 ethernet	NON	10/100 ethernet	NON	10/100 ethernet	NON	10/100 ethernet,wifi 802.11n, Bluetooth 4,1	Wifi 802.11n Bluetooth 4.1		10/100 ethernet(max 300Mbps),wifi 802.11ac, Bluetooth 4,2	Wifi 802.11ac Bluetooth 4.2
Systeme d'exploitation	Debian GNU/Linux,Raspbian OS, Fedora,Arch Linux ARM,RiscOS,FreeBSD,Plan 9,Kali Linux				Idem modèle 1+Snappy Ubuntocore,SolydX RPI , Windows 10 IoT	Debian GNU/Linux,Raspbian OS, Fedora,Arch Linux ARM,RiscOS,FreeBSD,Plan 9,Kali Linux	Idem modèle 1+Snappy Ubuntocore,SolydX RPI , Windows 10 IoT	Debian GNU/Linux,Raspbian OS, Fedora,Arch Linux ARM,RiscOS,FreeBSD,Plan 9,Kali Linux		Idem modèle 1+Snappy Ubuntocore,SolydX RPI , Windows 10 IoT	

Bien qu'un nouveau modèle Raspberry Pi offre davantage de fonctionnalités, il est préférable de connaître les applications ou les projets pour lesquels la carte doit être utilisée, afin de sélectionner la solution la plus efficace en termes de consommation et de coût.

5. Exemples de projets réalisés avec Raspberry Pi

La carte Raspberry Pi, dont le prix tourne autour de 40 euros, inspire un grand nombre de projets, dont certains très innovants. Voici une sélection de 8 d'entre eux parmi les plus inattendus. [PAU17]

Très certainement, l'un des attraits du Raspberry Pi est imputable à ses fortes capacités d'innovation et de créativité. En ce sens, l'ambition d'Ebon Upton et de la Fondation Raspberry Pi d'inciter des élèves à apprendre l'informatique et de permettre à des bidouilleurs de réaliser leurs projets les plus fous sans avoir à casser leur tirelire semble avoir atteint ses objectifs. Voici 8 projets parmi les plus inventifs, les plus surprenants et les plus incroyables réalisés ces dernières années avec le Raspberry Pi. Mieux encore : la plupart des inventeurs livrent tous les détails pour reproduire leurs projets à la maison ou dans un laboratoire.

a- Doorjam, un portier musical personnalisé

L'idée pourrait venir d'un ado discutant avec ses copains dans la cour du lycée, imaginant que chacun puisse entendre sa musique préférée quand il entre dans une pièce. Et voilà que le Raspberry Pi offre justement la technologie pour mettre au point un tel système. C'est ainsi qu'est né le projet Doorjam mis en œuvre par l'agence Redpepper. Doorjam associe une chaîne stéréo portable, un récepteur Raspberry Pi, un récepteur Bluetooth iBeacon et une application personnalisée pour smartphone. Quand le tout a été assemblé, chacun choisit la musique qu'il veut entendre quand il entre dans une pièce via l'API Spotify de l'application. Dès que le système détecte la personne qui a franchi la porte, il lance sa musique favorite. Ça ressemble à l'entrée sur le ring des boxeurs, mais pour les comptables et les analystes de données.



Figure 2.2 : Projet DoorJam. [PAU17]

b- Un scanner 3D à taille humaine

Bien sûr, il est toujours possible d'utiliser l'application Paint 3D de Windows 10 Creators Update pour créer un beau poisson en 3D, et peut-être qu'un jour, nous pourrions utiliser nos smartphones pour numériser un modèle 3D. Mais **Richard Garsthagen**, des Pays-Bas, avait une autre idée en tête. Surtout, il voulait quelque chose de grand capable de scanner... une personne tout entière ! Ce directeur EMEA du développement de l'activité cloud d'Oracle a donc construit son propre scanner 3D. Son énorme machine utilise 98 cartes Raspberry Pi réparties sur une vingtaine de trépieds et une centaine de caméras. On se croirait dans Terminator, mais la machine ne permet pas de voyager dans le temps. Elle peut cependant capturer toutes les données brutes pour créer un modèle 3D d'un être humain.

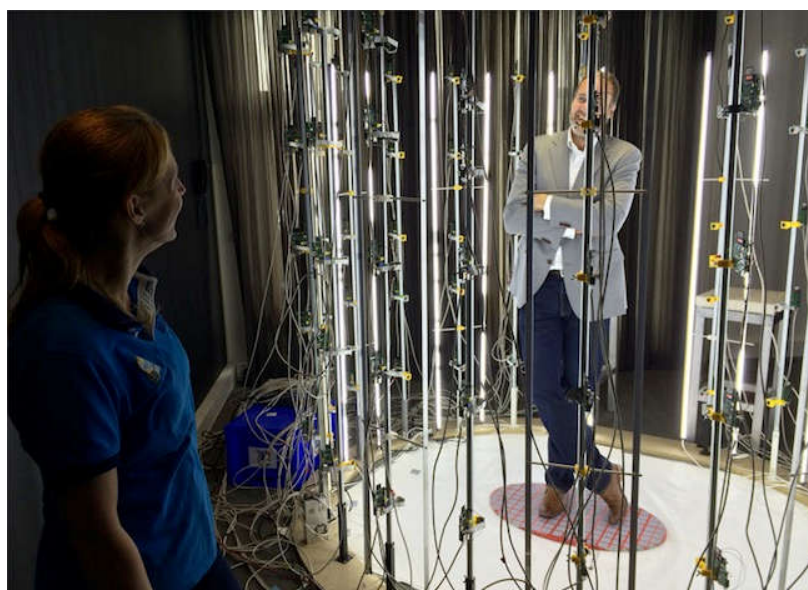


Figure 2.3 : Projet du scanner 3D. [PAU17]

c- Magic mirror, dernières infos avant le départ

L'idée du développeur **Michael Teeuw** est originale : Créer un miroir high-tech pour la maison qui permet de voir la météo, l'heure et les principales informations du jour pendant que l'on se prépare. Pour ce projet, le développeur a relié un miroir unidirectionnel à un écran et fixé une carte Raspberry Pi à l'arrière. Il a réuni le tout dans une boîte. C'est un vrai « killer concept », et son « miroir magique » a inspiré d'autres à créer leurs propres versions. Tous les détails sur le projet Magic Mirror ainsi que des photos de toutes les variantes réalisées par d'autres développeurs partout dans le monde sont disponibles sur le blog de **Michael Teeuw**.



Figure 2.4 : Projet du miroir magique. [PAU17]

d- MintyPi 2.0, une console neorétro avec écran

Le Raspberry Pi a été très utilisé pour reproduire une console de jeu rétro. Mais certains bidouilleurs ont poussé l'idée beaucoup plus loin : fabriquer une console de jeux portable. C'est le cas de cette petite console appelée mintyPi 2.0 que l'on peut voir sur le site Sudomod.com. Pour ce projet, le créateur **Warner Skoch**, connu en ligne sous le pseudonyme **Wermy**, a fait rentrer un Raspberry Pi Zero, un contrôleur Nintendo DS Lite et d'autres composants dans une boîte de pastilles mentholées de la marque Altoids. Il a complété le tout avec des pièces imprimées en 3D pour protéger les câbles. Au final, sa console portable miniature est très léchée et facile à loger dans une poche. **Wermy** a présenté son mintyPi 2.0 en avril, mais il n'a pas encore fourni de guide détaillé de tous les éléments qu'il a utilisés pour construire sa console. Mais, il est possible d'acheter certaines des pièces imprimées en 3D dans la boutique en ligne de Sudomod.



Figure 2.5 : Projet MintyPi. [PAU17]

e- Un verrou intelligent sous Windows IoT

Le Raspberry Pi n'est plus réservé à Linux. En effet, depuis un peu plus de deux ans, Microsoft a livré une version de Windows 10 appelée Windows IoT Core compatible avec le mini-PC. En janvier 2016, Microsoft a démarré un projet basé sur Windows IoT et sur sa technologie de reconnaissance faciale Project Oxford pour créer un verrou intelligent : l'idée est de permettre à une personne d'ouvrir une porte verrouillée en appuyant simplement sur la sonnette. Le système déverrouille la porte quand il reconnaît la personne. En fait, pour réaliser son prototype, Microsoft a utilisé une MinnowBoard Max d'Intel. Mais, il est tout à fait possible de faire la même chose avec une carte Raspberry Pi, comme le montre Microsoft sur Hackster.io. Le système présenté par Microsoft était juste une preuve de concept, donc assez rudimentaire, mais il est facile de l'améliorer sans trop d'efforts. Cela dit, il est difficile de garantir la sécurité du système, car Project Oxford étant basée sur la détection des visages dans les images, il suffirait de montrer une photo de la personne pour tromper le système. Reste que c'est un projet amusant à réaliser avec le Raspberry Pi.



Figure 2.6 : Projet de verrou intelligent. [PAU17]

f- Picrowave : Piloter un micro-ondes à la voix

Personnaliser son banal four à micro-ondes avec une carte Pi ? C'est ce qu'a voulu faire le développeur Nathan Broadbent. Notamment, il a doté son four de commandes vocales, d'un scanner de code barres qui lui permet d'accéder à une base de données en ligne pour connaître les temps de cuisson des aliments, d'une interface Web pour l'accès à distance et l'envoi de tweets automatiques quand le temps est écoulé.



Figure 2.7 : Projet Picrowave. [PAU17]

g- Pi-hole, le trou noir des annonces publicitaires

Il est devenu relativement facile de bloquer les annonces publicitaires sur son navigateur avec un simple plug-in, mais si vous ne supportez vraiment pas les publicités, le projet Pi-hole mérite le détour (<https://pi-hole.net/>). L'idée de Pi-hole est de créer un trou noir dans le

réseau destinée pour piéger les annonces. Pi-hole se substitue au serveur DNS et il avale les publicités acheminées vers toutes les machines du réseau. Il est nécessaire de procéder à quelques ajustements sur le routeur. Les instructions sont disponibles sur le site Web de Pi-Hole. Pi-Hole fonctionne avec le système d'exploitation Raspbian de Raspberry Pi et avec plusieurs distributions Linux standard, dont Ubuntu, Fedora, Debian et CentOS.

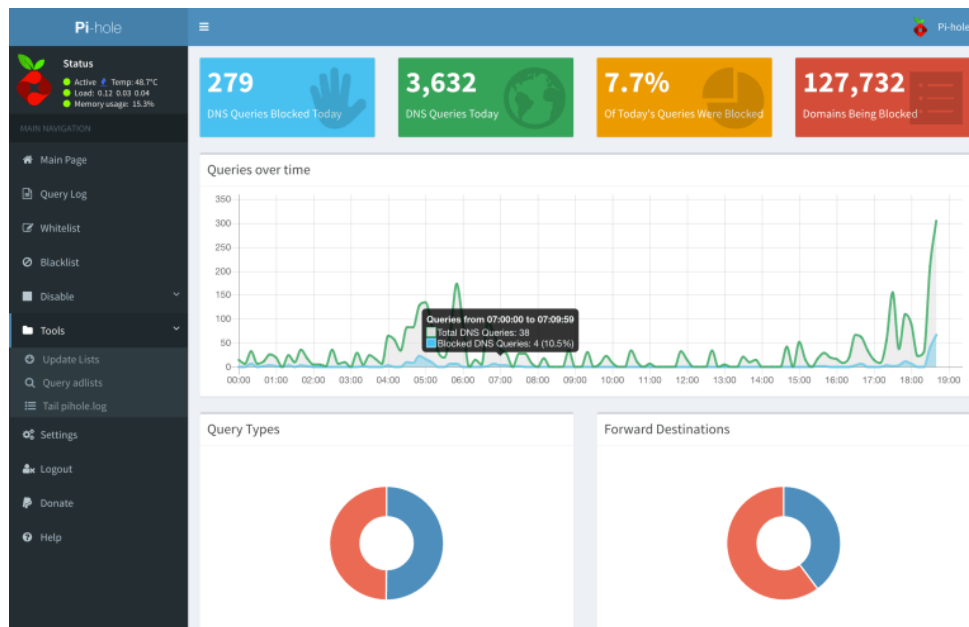


Figure 2.8 : Projet Pi-hole. [PAU17]

h- Un drone semi-autonome fait maison

C'est un projet plus coûteux, qui demande pas mal de compétences mécaniques, mais créer son propre drone avec un Raspberry Pi est assurément très gratifiant. Le Britannique Andy Baker explique comment il a construit un quad-copter semi-autonome préprogrammé avec son plan de vol. C'est un très bon point de départ pour ce genre de projet. Au mois de février, **Andy Baker** s'est lancé dans un nouveau projet de quad-copter.



Figure 2.9 : Projet du drone semi-autonome. [PAU17]

6. Conclusion

Nous avons donné dans ce chapitre quelques détails techniques sur le système choisi pour faire intégrer un système de reconnaissance des plaques d'immatriculation : Un Raspberry Pi. Pour mieux maîtriser l'implémentation d'un tel système on doit passer dans le chapitre suivant à la préparation d'une conception (globale et détaillée) permettant de décomposer le système à implémenter en un ensemble de modules. Les modules seront détaillés par la suite selon la démarche à suivre.

Chapitre 03

Conception du système RAPI

1. Introduction

Avec l'augmentation du nombre de voitures, Le contrôle de ces voitures est trop difficile et coûte très cher. Pour surmonter ces difficultés, nous devons utiliser un système automatisé qui repose sur la reconnaissance des plaques d'immatriculation (RAPI).

2. Caractéristiques des plaques algériennes

La plaque est composée de dix chiffres (Figure 3.1.) :



Figure 3.1. : Exemple de plaque d'immatriculation Algérienne.

La description des trois parties du système d'immatriculation algérien, de la droite vers la gauche, est la suivante :

- **07** : correspond au code de la wilaya (07 = biskra).
- **118** : "18" correspond à l'année de mise en circulation du véhicule, et "1" correspond au type de l'automobile. Ici c'est "1" car c'est une voiture touristique, (ex: "2" pour les camions lourd, "5" pour les semi-remorques, "8" pour les remorques, etc.....)
- **00535** : signifie que ce véhicule est le "535"^{ème} qui est entré dans la wilaya "07" (biskra) en "2018".

La plaque a une forme de rectangulaire. Le style des caractères varie d'une plaque à une autre et la distance entre les caractères n'est pas standardisé, elle dépend au 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.....

Donc en prenant en compte ces caractéristiques, nous proposons, dans la section suivante, la technique utilisée dans ce système pour la reconnaissance optique de caractères.

3. Le système RAPI à implémenter

Nous avons choisi d'implémenter une méthode légère et robuste pour la reconnaissance des plaques standards d'immatriculation algérienne. L'idée de base du système RAPI est qu'une voiture s'approche à l'endroit où la caméra est installée, l'image est capturée par la caméra et transmise au système RAPI. Le numéro reconnu par le système est transmis à la base de données et une décision est prise en conséquence. Considérant que la technique RAPI proposée consiste en trois modules principaux :

- Module de prétraitement qui va permettre de bien préparer l'image à analyser aux étapes suivantes.
- Module de localisation de plaques d'immatriculation en utilisant le détecteur **Canny** et quelques propriétés de la plaque tel que le rapport Largeur/Hauteur de la plaque.
- Module de reconnaissance du numéro de licence utilisant OCR (dans notre cas, on va utiliser un algorithme de reconnaissance de k plus proches voisins) pour reconnaître des caractères [CHE07].

4. Présentation de la démarche d'un RAPI

L'algorithme à utiliser va nous permettre de détecter et reconnaître automatiquement la plaque d'immatriculation d'un véhicule. L'entrée du système est une image prise par la caméra à une distance limitée. La technique RAPI à implémenter passe par les étapes suivantes (Figure 3.2) :

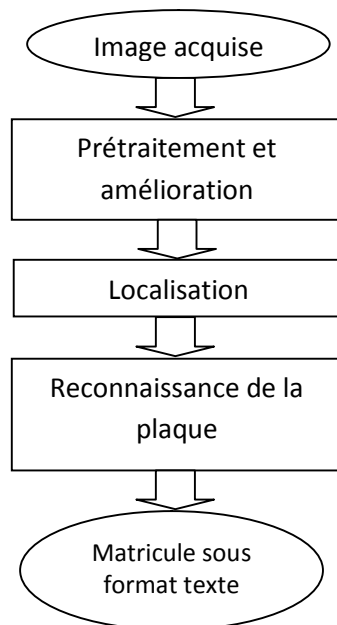


Figure 3.2 : Architecture du système RAPI à implémenter.

4.1. Prétraitement et amélioration

Durant la capture des images ou des trames vidéo à travers un appareil photo numérique (Webcam PiCamera / HR) les images peuvent contenir des bruits. La mauvaise qualité de l'image est la principale cause d'échec de la détection de la plaque d'immatriculation du véhicule. L'algorithme de prétraitement aide à améliorer la qualité de l'image capturée ou de l'image extraite de la plaque. Cela implique :

a- Redimensionner l'image

Cela implique de redimensionner l'image prise à des fins d'optimisation. Le redimensionnement est utile car la qualité de l'image diffère et afin de localiser la plaque d'immatriculation ou de pouvoir reconnaître la plaque d'immatriculation sur l'image, le redimensionnement sera très utile. Nous redimensionnerons alors l'image capturée à 500 px de largeur.

b- Lissage de l'image (Filtre flou)

Pour éliminer les bruits dans les images à analyser par la suite, nous allons utiliser un filtre bilatéral (filtre non linéaire). L'idée à la base du filtre bilatéral [HEB17] est d'ajouter des poids à la convolution qui tiennent compte de la distance entre les pixels dans l'espace des illuminances. Ainsi, deux pixels voisins dans l'espace image mais d'illuminance très différentes (comme sur une discontinuité) ne seront pas moyennés car le poids associé à leur distance dans l'espace des illuminances sera faible. En général, on choisit les poids selon une distribution gaussienne.

La figure 3.3. présente un exemple sur l'effet d'un filtre bilatérale où les contours restent clairs :

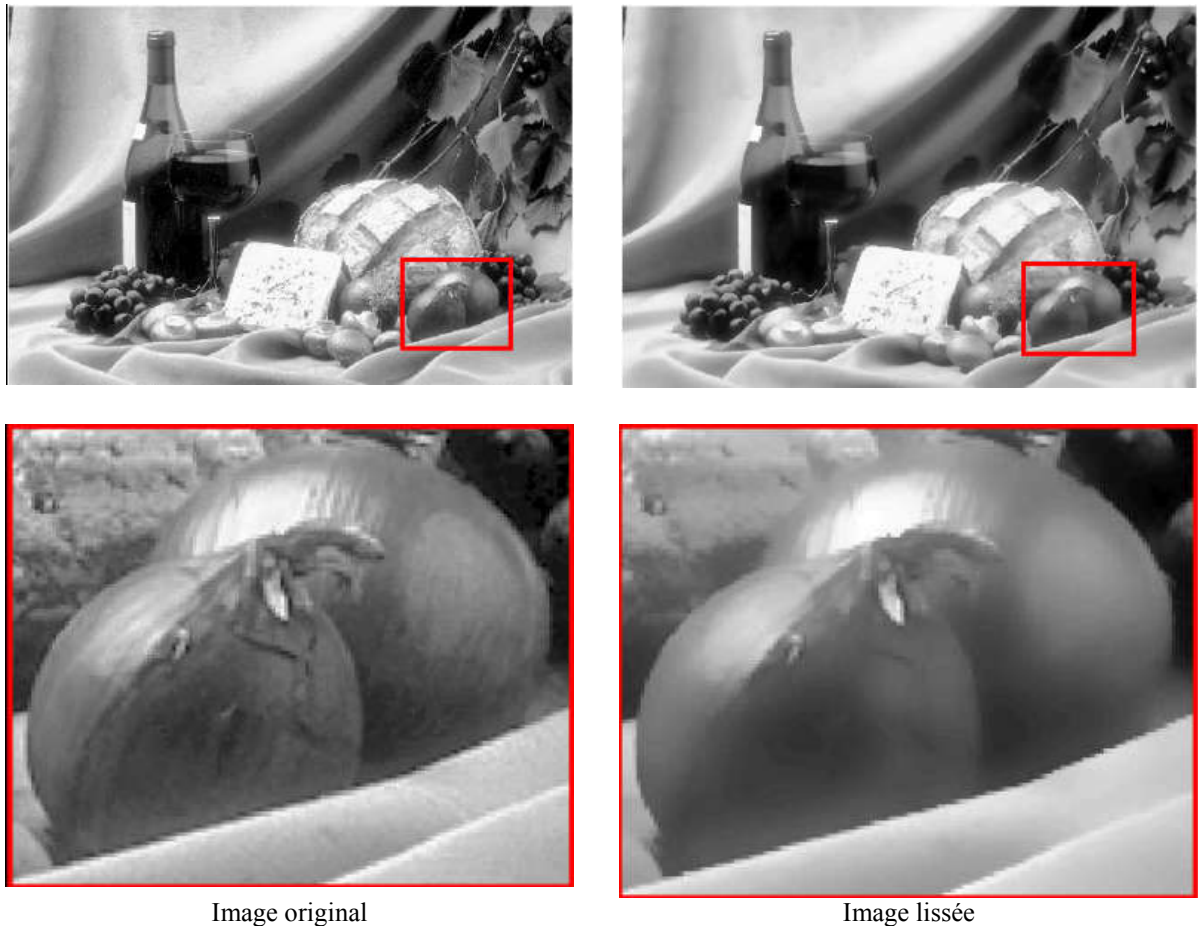


Figure 3.3. : Effet d'un filtre bilatéral, lissage sans perdre les contours. [HEB17]

c- Conversion en niveaux de gris

L'image couleur contient des couleurs 24 bits, il est donc difficile de manipuler l'image capturée. Pour résoudre le problème, nous appliquons la conversion en niveaux de gris à l'image ; nous réduisons les couleurs à 8 bits, ce qui est plus facile à gérer et nous réduisons également la valeur de pixel de 0 à 255, 0 représentant le noir et 255 à une couleur plus claire ou blanche. En appliquant cette méthode, les valeurs rouge, vert et bleu seront représentées par la valeur grise.

Cela nous aide à identifier les contours importants et d'autres caractéristiques de l'image qui peuvent être très utiles ces derniers temps pour retrouver la plaque d'immatriculation sur l'image, extraire et reconnaître de caractères de l'image, tout ce qui est mentionné est très difficile à réaliser avec l'image couleur.

d- Seuillage

Le seuillage est le moyen rapide de convertir l'image en niveaux de gris en une image binaire. Ici, les pixels de l'image en niveaux de gris vont être séparés ou arrangés. L'image en niveaux de gris contient des pixels de 0 à 255. Ainsi, 0-127 correspondra à 0 et 128-255 à 1,

ce qui explique pourquoi nous disons que c'est le moyen le plus rapide de convertir l'image en niveaux de gris en image binaire (voir figure 3.4.).



Figure 3.4. : Le seuillage permet de convertir une image de niveaux de gris vers noir et blanc. [MAZ18]

4.2. Localisation de la plaque

Cette technique est utilisée pour identifier la région de la plaque d'immatriculation à partir de l'image donnée. L'objectif principal de ce type de systèmes est de localiser la région de la plaque d'immatriculation à partir de l'image des véhicules capturés.

La détection et l'isolement de la plaque : c'est la phase la plus importante et la plus difficile, elle détermine la rapidité et la robustesse du système. Plusieurs études et recherches sont consacrées à cette phase depuis plusieurs années ainsi différentes méthodes ont été proposées. On peut citer dans ce cas les méthodes basées sur la détection de contours, les méthodes basées sur les opérations morphologiques et d'autres sur les caractéristiques de l'image comme la couleur [DIA04].

La détection de contours est le processus d'identification de l'occurrence de changement entre les couleurs. Un contour est détecté en vérifiant le pixel avec son voisin et s'il est déterminé qu'ils diffèrent, nous avons un contour, mais si la comparaison est identique, nous n'avons pas de contour. La méthode de détection des contours serait appliquée aux images binaires de cet algorithme. La détection des contours est simplement un ensemble de méthodes mathématiques visant uniquement à déterminer les occurrences de changement des intensités survenant dans l'image. Encore une fois peut dire est un ensemble de pixels connectés qui se trouvent à la frontière entre deux régions. La détection de modifications nettes sur les images lumineuses permet de capturer les événements importants survenus. La détection des contours est très utile car elle peut aider à filtrer les informations qui peuvent être considérées comme moins pertinentes. La première dérivée peut être utilisée pour

détecter la présence d'un bord en un point d'une image, tandis que la deuxième dérivée peut être utilisée pour déterminer si le pixel est sur l'obscurité ou sur une image [WITR18].

Le filtre de **Canny** fonctionne dans un processus en plusieurs étapes. Tout d'abord, l'image est lissée en appliquant la convolution gaussienne. Le premier opérateur de dérivée (un peu comme Robert Cross) est appliqué à l'image lissée afin de mettre en évidence des régions de l'image avec des premières dérivées spatiales élevées. L'algorithme suit ensuite le haut de ces crêtes et met à zéro tous les pixels qui ne sont pas réellement au sommet de la crête afin de donner une ligne fine dans la sortie. Ce processus est appelé suppression non maximale [ETC08].

Le filtre de **Canny** a plusieurs avantages [BER15] :

- **détection** : bonne détection des points de contours (notamment pour les contours faibles),
- **localisation** : localisation fiable des points de contours,
- **unicité de la détection** : pour un contour, on a un seul point de contour.

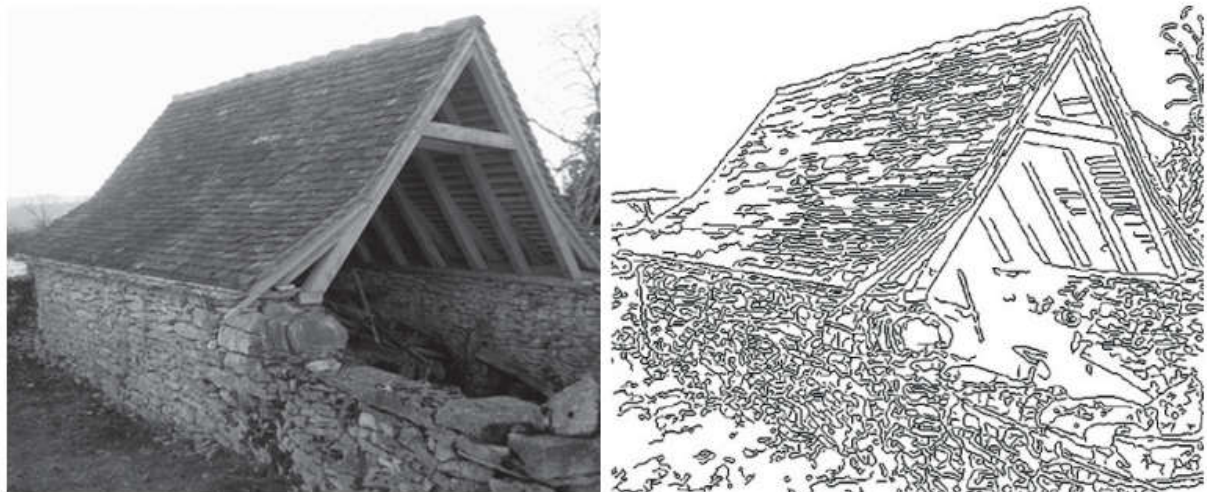


Figure 3.5. : Détecteur de contours de Canny. [BER15]

Afin de détecter les régions de l'image candidate à la plaque, nous appliquons tout d'abord l'algorithme de **Canny** pour détecter les objets à frontière fermée. Un certain nombre d'algorithmes d'évaluation candidats sont appliqués sur des images de contour obtenues à partir de l'algorithme de détection de contour afin de séparer les objets de plaque. Cependant, cet algorithme rencontre des difficultés pour traiter des images de mauvaise qualité en raison

de rayures, d'hélices de plug-in. Dans ces cas, l'algorithme de contour produit des lignes de limites fermées incomplètes qui ne détecte pas correctement les images de la plaque.

Une étape d'évaluation des contours fermés candidats repose sur une vérification du rapport entre la hauteur et la largeur du candidat. Nous vérifions et sélectionnons uniquement les candidats dont le rapport hauteur / largeur est satisfait par la contrainte suivante :

$$\text{Seuil min} \leq \text{Largeur} / \text{Hauteur} \leq \text{Seuil max}$$

Nous avons choisi Seuil min = 3,5 et Seuil max = 6. Les candidats qui satisfont aux contraintes sont sélectionnés et passés à l'étape suivante, à savoir le module de reconnaissance des numéros de la plaque.

4.3. Reconnaissance de la plaque

Les systèmes de reconnaissance optique des caractères ont été développés il y'a de ça une vingtaine d'années et leurs performances dépend des conditions environnantes et doivent être capables de gérer un grand nombre de difficultés possibles. Parmi lesquelles :

- Une mauvaise résolution de l'image à cause d'une plaque trop éloignée ou d'une caméra de mauvaise qualité,
- Des images floues souvent à cause du mouvement, un mauvais éclairage et un faible contraste, un reflet ou une ombre, un objet obscurcissant une partie de la plaque (barre de remorquage ou de la poussière).

Le module de reconnaissance sera divisé en deux parties, la première partie implique la segmentation des caractères et la seconde partie comprend la reconnaissance des caractères.

4.3.1. Segmentation des caractères

Les composants connectés numérisent une image et regroupent ses pixels en composants basés sur la connectivité des pixels. Tous les pixels d'une composante connectée partagent des valeurs d'intensité de pixel similaires et sont en quelque sorte connectés les uns aux autres. Cette méthode nécessite que l'image d'entrée soit convertie en binaire pour traitement. La limite de chaque objet est vérifiée pour voir s'il occupe la surface attendue. Cette méthode utilise la propriété que toutes les plaques d'immatriculation sont rectangulaires et que leurs caractères partagent le même arrière-plan.

4.3.2. Reconnaissance des caractères

Après la segmentation des caractères, nous devons reconnaître tous les caractères en utilisant un moteur OCR pour obtenir le code ASCII. Nous décrivons d'abord le fond des moteurs OCR et leur fonctionnement.

OCR est l'acronyme de Reconnaissance Optique de Caractères. Cette technologie permet de reconnaître automatiquement les caractères grâce à un mécanisme optique. Dans le cas des êtres humains, nos yeux sont un mécanisme optique. L'image vue par les yeux est une entrée pour le cerveau. La capacité à comprendre ces entrées varie chez chaque personne en fonction de plusieurs facteurs. L'OCR est une technologie qui fonctionne comme une capacité de lecture humaine, bien que l'OCR ne puisse pas rivaliser avec les capacités de lecture l'être humain. Parmi les techniques existantes nous pouvons citer : Tesseract Engine et l'algorithme de k-plus proches voisins (KNN en anglais k nearest neighbour).

a- Moteur Tesseract

Tesseract est un moteur de reconnaissance optique de caractères pour différents systèmes d'exploitation. C'est un logiciel libre, publié sous la licence Apache, version 2.0, et le développement est sponsorisé par Google depuis 2006. Tesseract est considéré comme l'un des moteurs de OCR open source les plus précis actuellement disponibles.

b- Algorithme KNN

L'algorithme KNN figure parmi les plus simples algorithmes d'apprentissage artificiel. Dans un contexte de classification d'une nouvelle observation x , l'idée fondatrice simple est de faire voter les plus proches voisins de cette observation. La classe de x est déterminée en fonction de la classe majoritaire parmi les k plus proches voisins de l'observation x . La méthode KNN est donc une méthode à base de voisinage, non-paramétrique ; Ceci signifiant que l'algorithme permet de faire une classification sans faire d'hypothèse sur la fonction $y=f(x_1, x_2, \dots, x_p)$ qui relie la variable dépendante aux variables indépendantes. La méthode du plus proche voisin est une méthode non paramétrique où une nouvelle observation est classée dans la classe d'appartenance de l'observation de l'échantillon d'apprentissage qui lui est la plus proche, au regard des covariables utilisées. La détermination de leur similarité est basée sur des mesures de distance. [DUP10]

On va essayer d'utiliser une des deux techniques pour faire la reconnaissance des caractères. Donc, les caractères segmentés vont être envoyés à un algorithme de reconnaissance, qui renvoie l'ASCII de chaque numéro de la plaque.

5. Conclusion

Nous avons présenté dans ce chapitre une conception détaillée d'un système de reconnaissance automatique des plaques d'immatriculation. Cette conception va nous permet de faciliter la démarche d'implémentation logiciel du système. L'implémentation sera le sujet du chapitre suivant.

Chapitre 04

Implémentation du système

RAPI

1. Introduction

Après avoir définir et décrire les différents modules constituant un système de reconnaissance automatique des plaques d'immatriculation, on va passer à décrire les différents composants du système à réaliser. Nous allons, donc, décrire brièvement les différentes étapes de la réalisation du projet. On va acheminer notre travail par l'environnement matériel et logiciel qui assure la réalisation de notre plateforme électronique et de ses applications.

2. Aspect matériel

Pour faire implémenter la technique RAPI selon la conception faite dans le chapitre précédent, nous avons choisi d'utiliser un Raspberry Pi 3 modèle B+. Par rapport à celles de son prédécesseur, la vitesse et la puissance de Raspberry Pi 3 B+ ont gagné globalement 15 %. Elle est caractérisée par :

- Accélération du processeur quadricœur : environ 15 % de capacité de traitement supplémentaire,
- Prise en charge du décodage H.264 MPEG-4 (1080p30), du codage H.264 (1080p30) et de la gestion graphique Open GL ES 1.1 et 2.0,
- 1 Go de mémoire LPDDR2 SDRAM,
- Communication sans fil double bande 802.11ac (2,4 GHz / 5,0 GHz),
- Gestion de l'alimentation améliorée,
- Gestion thermique améliorée.

Pour la capture des vidéos et des images nous allons utiliser une caméra Longruner Camera pour Raspberry avec vision de nuit. Cette caméra dotée d'un capteur Omnivision 5647 possède une résolution de 5 Megapixels 2592 x 1944 pixels et une résolution vidéo maximale de 1080p @30fps, 720p @60fps et 640*480 @90fps.

3. Aspect logiciel

Pour notre Raspberry, nous avons installé sur une carte mémoire SD le système d'exploitation Raspbian (C'est un système d'exploitation libre et gratuit basé sur Debian optimisé) pour faire fonctionner le Raspberry Pi. Étant donné les ressources limitées de cet ordinateur, Raspbian utilise des logiciels réputés pour être légers : l'environnement de bureau par défaut est PIXEL, acronyme de « Pi Improved Xwindows Environment Lightweight », « Environnement Xwindows amélioré et léger pour PI », le navigateur web par défaut est Chromium.

Pour la réalisation de ce projet, nous avons utilisé un ensemble d'outils de développements tel que : Le langage de programmation Python 3.5, la bibliothèque de vision par ordinateur OpenCV 4.

3.1. Python

Le langage de programmation Python [FUC19] a été créé en 1989 par **Guido van Rossum**, aux **Pays-Bas**. Le nom Python vient d'un hommage à la série télévisée *Monty Python's Flying Circus* dont **G. van Rossum** est fan. La première version publique de ce langage a été publiée en 1991.

La dernière version de Python est la version 3. Plus précisément, la version 3.7 a été publiée en juin 2018. La version 2 de Python est désormais obsolète et cessera d'être maintenue après le 1er janvier 2020. Dans la mesure du possible évitez de l'utiliser.

La Python Software Foundation est l'association qui organise le développement de Python et anime la communauté de développeurs et d'utilisateurs.

Ce langage de programmation présente de nombreuses caractéristiques intéressantes [FUC19, SWI10]:

- Il est multiplateforme (portable). C'est-à-dire qu'il fonctionne sur de nombreux systèmes d'exploitation : Windows, Mac OS X, Linux, Android, iOS, depuis les mini-ordinateurs Raspberry Pi jusqu'aux supercalculateurs.

- Il est gratuit. Vous pouvez l'installer sur autant d'ordinateurs que vous voulez (même sur votre téléphone !).

- C'est un langage de haut niveau. Il demande relativement peu de connaissance sur le fonctionnement d'un ordinateur pour être utilisé.

- C'est un langage interprété. Un script Python n'a pas besoin d'être compilé pour être exécuté, contrairement à des langages comme le C ou le C++. Il convient bien à des scripts d'une dizaine de lignes qu'à des projets complexes de plusieurs dizaines de milliers de lignes.

- Il est relativement simple à prendre en main.

- il est très utilisé en bioinformatique et plus généralement en analyse de données.

- Python est orienté-objet. Il supporte l'héritage multiple et la surcharge des opérateurs. Dans son modèle objets, et en reprenant la terminologie de C++, toutes les méthodes sont virtuelles. Il est possible de concevoir en Python des entités qui miment celles

du monde réel (une cellule, une protéine, un atome, etc.) avec un certain nombre de règles de fonctionnement et d'interactions.

— Python intègre, comme Java ou les versions récentes de C++, un système d'exceptions, qui permettent de simplifier considérablement la gestion des erreurs.

— Python est dynamique (l'interpréteur peut évaluer des chaînes de caractères représentant des expressions ou des instructions Python), orthogonal (un petit nombre de concepts suffit à engendrer des constructions très riches), réflexif (il supporte la métaprogrammation, par exemple la capacité pour un objet de se rajouter ou de s'enlever des attributs ou des méthodes, ou même de changer de classe en cours d'exécution) et introspectif (un grand nombre d'outils de développement, comme le debugger ou le profiler, sont implantés en Python lui-même).

— Enfin, Comme Scheme ou SmallTalk, Python est dynamiquement typé. Tout objet manipulable par le programmeur possède un type bien défini à l'exécution, qui n'a pas besoin d'être déclaré à l'avance.

Toutes ces caractéristiques font que Python est désormais enseigné dans de nombreuses formations, depuis l'enseignement secondaire jusqu'à l'enseignement supérieur.

3.2. OpenCV

OpenCV (Open Source Computer Vision Library) [EBO] est une bibliothèque de logiciels et de logiciels d'apprentissage automatique. Il a été conçu pour divers objectifs tels que l'apprentissage automatique, la vision par ordinateur, l'algorithme, les opérations mathématiques, la capture vidéo, le traitement d'images, ... etc. , android, ios). En outre, il a été enveloppé dans divers langages de programmation renommés. Dans le cadre du contrat de licence, les entreprises peuvent accéder au code et le modifier.

La bibliothèque contient plus de 2500 algorithmes optimisés, qui ont une excellente précision en termes de performances et de vitesse. Ces algorithmes peuvent être utilisés pour détecter et reconnaître des visages, identifier des objets, classer des actions humaines dans des vidéos, suivre des mouvements de caméra, suivre des objets en mouvement, extraire des modèles 3D d'objets, produire des nuages de points 3D à partir de caméras stéréo image d'une scène entière, trouver des images similaires à partir d'une base de données d'images, supprimer les yeux rouges des images prises, suivre les mouvements oculaires, reconnaître les décors et établir des marqueurs pour la superposer à la réalité augmentée. , les développeurs et les chercheurs, le nombre est plus de 47 000 et le nombre estimé de téléchargements dépasse

7 millions. La bibliothèque est largement constituée de sociétés professionnelles, de groupes de recherche et d'autres groupes.

De nombreuses entreprises bien établies telles que Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda et Toyota qui emploient la bibliothèque, ainsi que de nombreuses startups telles que Applied Minds, VideoSurf et Zeitera, utilisent largement OpenCV. Les utilisations déployées d'OpenCV s'étendent de l'assemblage d'images StreetView, à la détection d'intrusions dans la vidéo de surveillance en Israël, à la surveillance des équipements miniers en Chine, à la détection des accidents de noyade en Europe et à l'art interactif. L'Espagne et New York vérifient les pistes en Turquie et inspectent les étiquettes des produits dans les usines du monde entier pour détecter rapidement les visages au Japon. Il possède des interfaces C++, C, Python, Java et MATLAB et supporte Windows, Linux, Android et Mac OS. OpenCV s'appuie principalement sur les applications de vision en temps réel et tire parti des instructions MMX et SSE lorsqu'elles sont disponibles. Des interfaces complètes CUDA et OpenCL sont en cours de développement. Il existe plus de 500 algorithmes et environ 10 fois plus de fonctions qui composent ou supportent ces algorithmes. OpenCV est écrit nativement en C++ et possède une interface basée sur des modèles qui fonctionne de manière transparente avec les conteneurs STL.

3.3. Algorithmes implémentés

Nous allons présenter ci-dessous, sous format algorithmique, les modules du système RAPI que nous avons fait leur conception.

Le premier algorithme (Algorithme 1) permet de préparer chaque trame reçue depuis la caméra à l'étape de localisation de la plaque d'immatriculation. Le prétraitement de l'image passe par les étapes de redimensionnement, conversion en niveau de gris et l'élimination des bruits en utilisant un filtre bilatéral.

Algorithme 1 : Prétraitement et amélioration de la trame lue

Algorithme Prétraitement(trame_lue)

Entrées :

trame_lue : image représentant une trame vidéo

Résultats :

Image_filtrée : image qui a subis un ensemble de pré-traitements.

Début

Redimensionner(trame_lue , largeur=500)

Convertir l'image redimensionnée en niveaux de gris

```
Appliquer un filtre bilatéral sur l'image en niveau de gris (pour éliminer
les bruits)
```

```
Renvoyer Image_filtrée
```

Fin

L'algorithme 2 sert à localiser une plaque d'immatriculation s'il existe dans l'image prétraitée. Cela sera effectué en suivant les étapes suivantes : Conversion de l'image en noir et blanc, l'extraction de la liste des contours fermés, le tri des contours par ordre de surface et on va choisir enfin celui qui a quatre coins et qui va vérifier un facteur largeur/hauteur entre 3.5 et 6.

Algorithme 2 : Localisation de la plaque d'immatriculation

Algorithme Localizer(image_entrée)

Entrées :

image_entrée : image pré-traitée

Résultats :

Détectée : Booléen indiquant si une plaque est détectée,

PlaqueImg : Image extraite de l'image d'entrée représentant la plaque,

PlLargeur, PlHauteur : Les dimensions de la plaque.

Début

```
Appliquer un seuillage adaptatif sur l'image d'entrée pour avoir une image
noire et blanc
```

```
Appliquer un filtre de Canny sur l'image noir et blanc
```

```
Extraire, de la liste des contours, seulement les contours fermés.
```

```
Trier la liste des contours extraites selon leurs surfaces du plus grand
vers le plus petit
```

```
Détectée ← faux
```

```
Pour chaque contour de la liste faire
```

```
    Si le contour possède quatre (04) coins Alors
```

```
        Extraire les coordonnées du rectangle englobant le contour
```

```
        Calculer la largeur et la hauteur du contour : PlLargeur, PlHauteur
```

```
        Rapport ← PlLargeur / PlHauteur
```

```
        Si Rapport >=3.5 et Rapport<=6 Alors
```

```
            Détecté = vrai
```

```
            Sortir de la boucle
```

```
        FinSi
```

```
    FinSi
```

```
FinPour
```

Si DéTECTÉE **Alors**

Extraire l'image de la plaque détectée soit : PlaqueImg

FinSi

Renvoyer (DéTECTÉE, PlaqueImg, PlLargeur, PlHauteur)

Fin

L'algorithme suivant (Algorithme 3) permet de segmenter l'image de la plaque détectée en se basant sur les contours et d'identifier les chiffres du matricule. Pour la liste des contours détectés on va choisir celui qui ont un rapport Largeur/Hauteur entre 0.25 et 1 et un rapport Hauteur/Hauteur plaque entre 0.5 et 0.8. La zone candidate sera envoyée à un algorithme de reconnaissance de k plus proches voisins (kNN) qui va nous renvoyer un chiffre correspondant. Les chiffres correspondant aux zones segmentées vont construire le matricule de la voiture.

Algorithme 3 : Segmentation de la plaque d'immatriculation et reconnaissance des chiffres

Algorithme SegNumEtConvert(PlaqueImg, PlLargeur, PlHauteur)

Entrées :

PlaqueImg : image de la plaque détectée

PlLargeur, PlHauteur : Largeur et hauteur de la plaque

Résultats :

Matricule : Chaîne de caractère représentant les chiffres définissant le matricule reconnu.

Début

Matricule ← ""

Conversion de l'image de la plaque en niveaux de gris

Appliquer sur l'image en niveau de gris un filtre bilatéral

Appliquer sur l'image filtrée un seuillage adaptatif pour avoir une image noire et blanc

Agrandir la taille de l'image de la plaque

Appliquer une autre fois un seuillage normal pour confirmer l'élimination de tout bruit

Appliquer un filtre de Canny pour extraire les contours

Trouver la liste des contours fermés

Trier la liste des contours trouvés du gauche vers la droite

Pour tous les contours triés **faire**

 Calculer pour le rectangle englobant le contour : Hauteur et Largeur

 Rapport ← Largeur / Hauteur

```

RapportChiffrePlaque ← Hauteur / PlHauteur

Si Rapport < 1 et Rapport>0.25 et RapportChiffrePlaque>0.5 et
RapportChiffrePlaque<0.8 Alors

    Extraire depuis l'image binaire imgROI englobée par le
    rectangle calculé

    Redimensionner imgROI à une taille de 20x30

    Construire depuis imgROI un vecteur uni-dimensionnel

    Trouver le chiffre approprié en utilisant l'algorithme de "k
    plus proches voisins"

    Ajouter le chiffre reconnu à (Matricule) la liste des
    chiffres de la plaque

FinSi

FinPour

Renvoyer Matricule

Fin

```

Le dernier algorithme représente la suite des étapes à suivre pour accomplir le processus de la reconnaissance automatique des plaques d'immatriculation.

Algorithme 4 : Programme principale de RAPI

Algorithme RAPI ()

Début

```

Lancer l'apprentissage pour l'algorithme de kNN (k plus proches voisins)

Pour toutes les images capturées faire

    Image_Filtrée=Pré_traitement (Trame courrente)

    (DéTECTÉE,Image_Plaque,PLargeur,PHauteur) = Localizer(Image_Filtrée)

    Si DéTECTÉE Alors

        Matricule=SegNumEtConvert(Image_Plaque, PLargeur, PHauteur)

        Afficher(Matricule)

    FinSi

FinPour

```

Fin

3.4. Description des fonctions Python et OpenCV utilisés pour l'implémentation

Nous avons utilisé plusieurs bibliothèques python pour arriver à réaliser notre système RAPI. Parmi les plus importantes on peut citer les suivantes [COP19] :

a. NumPy

NumPy est une bibliothèque numérique apportant le support efficace de larges tableaux multidimensionnels, et de routines mathématiques de haut niveau (fonctions spéciales, algèbre linéaire, statistiques, etc.).

b. Matplotlib

Matplotlib est une bibliothèque graphique de visualisation 2D (avec un support pour la 3D, l'animation et l'interactivité), permettant des sorties de haute qualité « prêtes à publier ». C'est à la fois une bibliothèque de haut niveau, fournissant des fonctions de visualisation « clés en main » (échelle logarithmique, histogramme, courbes de niveau, etc.), et de bas niveau, permettant de modifier tous les éléments graphiques de la figure (titre, axes, couleurs et styles des lignes, ... etc.).

c. ImUtils

ImUtils est une série de fonctions pratiques permettant de simplifier les fonctions de traitement d'images de base telles que la translation, la rotation, le redimensionnement, la squelettisation et l'affichage facile des images Matplotlib avec OpenCV sous Python 2.7 et Python 3.

D'un autre côté, nous avons utilisé aussi la bibliothèque de la vision par ordinateur OpenCV. Parmi les fonctions les plus importantes on peut citer les suivantes [EBO] :

a. bilateralFilter

Il existe un compromis entre la perte de structure et la suppression du bruit, car la méthode la plus populaire pour éliminer le bruit est le flou gaussien qui ne connaît pas la structure de l'image, par conséquent, il supprime également les bords. La plupart du temps, les bords contiennent des informations précieuses sur la scène et nous ne voulons pas les perdre. Le filtre bilatéral connaît la structure de la scène et a tendance à agir comme un filtre de flou classique lorsqu'il se trouve sur une zone sans arêtes. Cependant, quand il voit un bord, il change de comportement de sorte que le flou ne fonctionne pas sur tous les bords, mais fonctionne le long des bords, ce qui signifie qu'ils sont des filtres préservant les bords.

b. arcLength

Le périmètre de chaque objet peut être trouvé en utilisant la fonction « arcLength », selon la syntaxe : `Perimetre = arcLength(contours[i], true)`.

c. findContours

La fonction `findContours` permet d'obtenir, comme son nom l'indique, tous les contours présents dans une image binaire. Pour chaque contour obtenu, nous calculons la périphérie de la forme pour faire ensuite une approximation des formes rectangulaires.

d. `approxPolyDP`

La fonction `findContours` trouve beaucoup trop de points pour déterminer les segments de droite minimaux qui composent la figure. Nous devons donc utiliser une fonction `approxPolyDP` qui nous permettra de réduire le nombre de point utilisé pour définir le contour de la courbe. L'idée de cette méthode est d'approcher une courbe (un contour) par une série de courtes droites. La fonction prend en paramètre : le vecteur 2D des points trouvés par `findContours`, une valeur qui représente une précision pour effectuer l'approximation, epsilon qui représente une valeur de 1% à 3% du périmètre donne de bons résultats, et elle va nous renvoyer un vecteur de point qui recevra les points approximés.

e. `KNearest_create, train`

D'abord `KNearest_create` va nous créer un classifieur kNN. Par la suite la fonction `train` va recevoir un jeu de données, que nous allons utiliser, constitué d'un ensemble de d'images 30x20 pixels en noir et blanc annotées du chiffre correspondant (entre 0 et 9) et les lettres (de A à Z). L'objectif de ce jeu de données était de permettre à un ordinateur d'apprendre à reconnaître des nombres et chiffres automatiquement.

f. `findNearest`

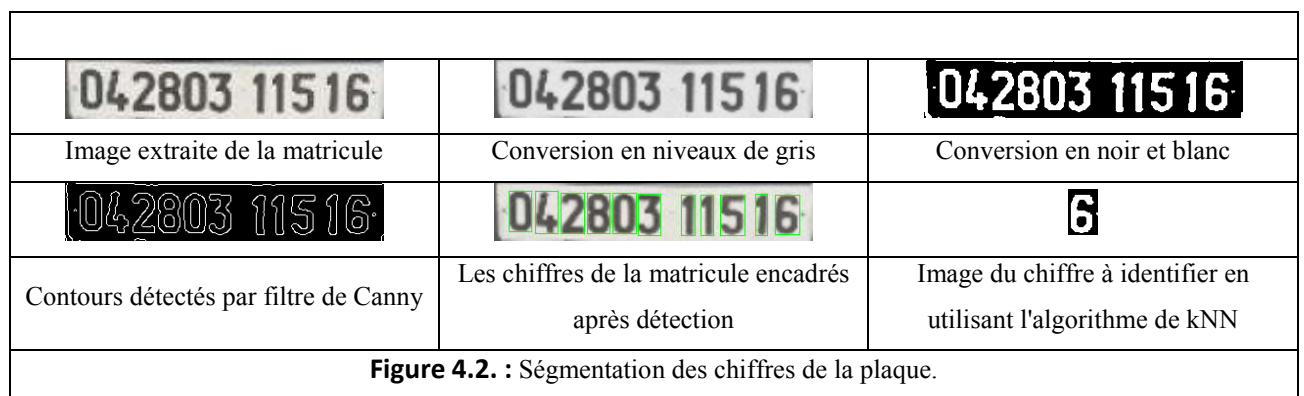
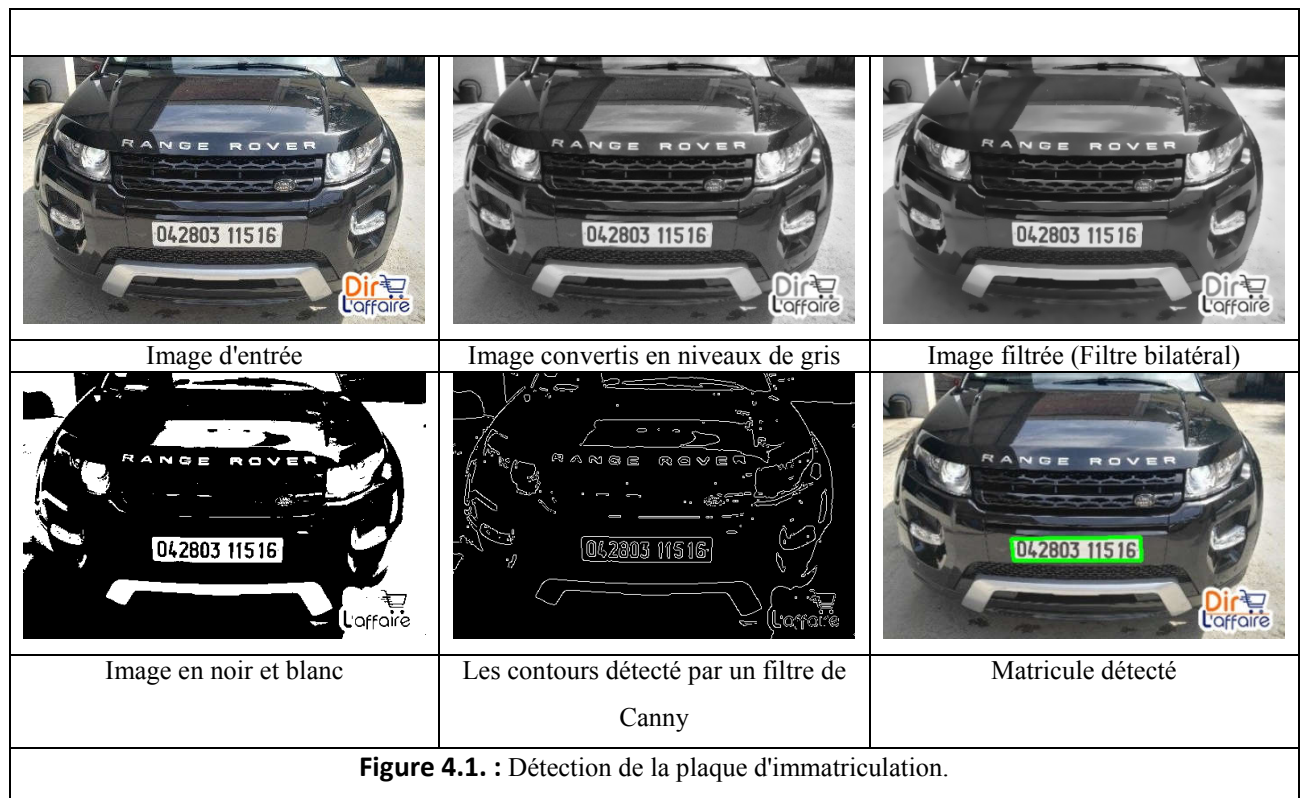
Cette fonction sert à trouver le chiffre le plus proche de celui que nous avons donné en appliquant l'algorithme kNN.

4. Résultats et discussions

Nous allons commencer cette partie par une présentation générale de la démarche suivie pour localiser les plaques et identifier par la suite les chiffres dans chacune des plaques. Nous allons passer par la suite à une analyse des résultats obtenues en utilisant un ensemble d'images de voitures dans différents cas pour pouvoir évaluer les performances de la méthode implémentée. Enfin, on va passer à une discussion des résultats obtenus.

4.1. Démarche de la technique RAPI

Nous allons présenter ici les résultats intermédiaires de l'exécution de notre programme en suivant les étapes de chaque module :



4.2. Présentation des résultats obtenus

Nous avons séparé les résultats obtenus en deux parties ; Les résultats de module de localisation de la plaque d'immatriculation, et les résultats du module de reconnaissance des chiffres.

4.2.1. Résultats de localisation

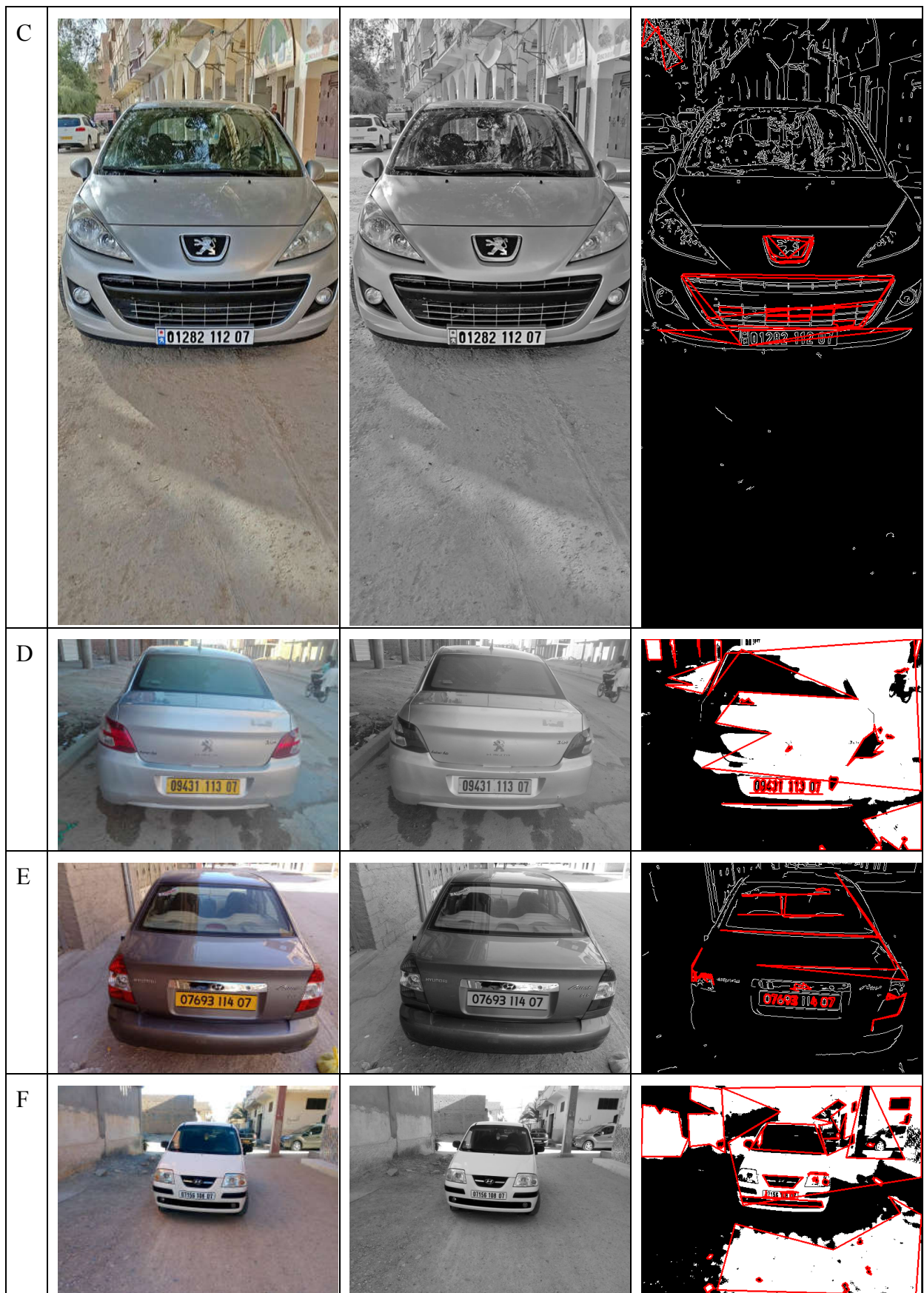
Pour évaluer le module de localisation implémenté en se basant sur la détection et l'analyse des contours, nous avons utilisé une base d'images collectée depuis le web. Nous avons utilisé 80 images de voitures de couleurs et de positions différentes. Le tableau 4.1. représente des statistiques sur les résultats obtenus à travers le système implémenté :

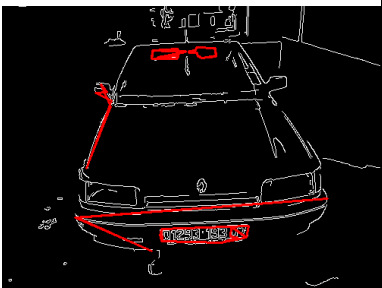
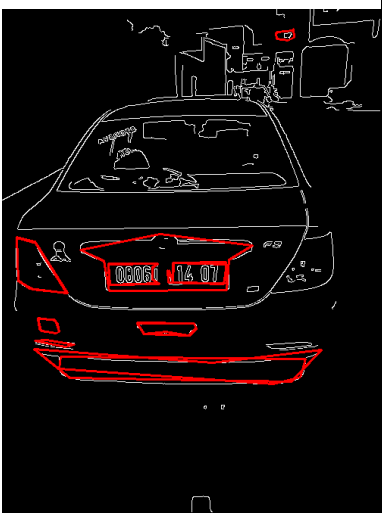
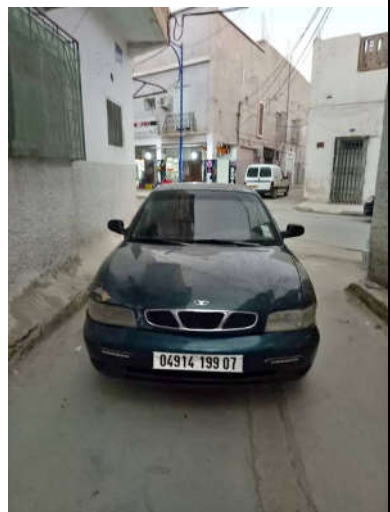
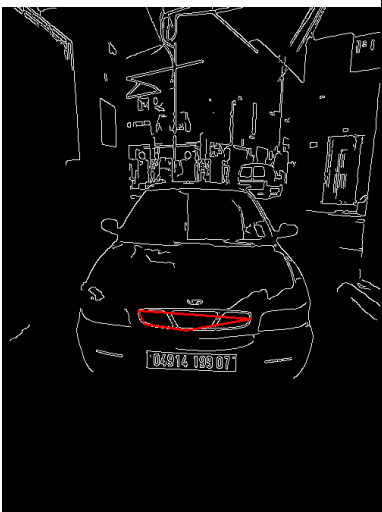
Tableau 4.1. : Les statistiques de l'évaluation de la technique de localisation de la plaque.

	Nombre de cas	Taux
Détections signalées	52	65.00% (de 80 images)
Détections correctes	33 (1-2-8-11-13-16-17-18-19- 20-21-25-26-28-29-30-31- 33-36-40-41-49-54-55-56- 57-62-66-67-69-71-77-79)	63.46% (de 52 images) 41.25% (de 80 images)
Fausses détections	19 (4-14-22-34-37-38-39-42-43- 44-46-47-53-58-60-64-68- 73-76)	36.53% (de 52 images)

Nous allons présenter, dans le tableau 4.2., une liste des images représentant les cas des fausses détections et les cas des plaques non détectées. Pour chaque image nous allons donner sa conversion en niveaux de gris et la liste des contours traités correspondants :

Tableau 4.2. : Présentation des cas des problèmes de localisation de la plaque.			
	Image	Image en niveaux de gris	Contours traités
A			
B			



G			
H			
I			
J			
K			



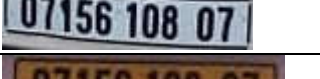
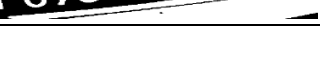
4.2.2. Résultats de reconnaissance

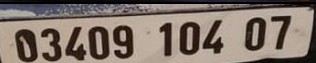
Nous avons essayé d'utiliser le moteur OCR *Teserract* mais il n'a pas donné des résultats pour la reconnaissance. Nous avons, donc, décider de passer à l'utilisation du moteur à base de l'algorithme des k-plus proches voisins. Cet algorithme nécessite de passer premièrement par une phase d'apprentissage. Pour ce faire, nous avons utilisé un ensemble de données (data set) téléchargé à partir de la page web : <https://github.com/muchlisinadi/ALPR-Indonesian-plate>. La base originale permet de faire un apprentissage sur les lettres de A à Z et des chiffres de 0 à 9. Nous avons extrait depuis cette base la partie permettant de reconnaître les chiffres seulement où chacune des dix classes (représentant un chiffre par son code ASCII) comporte cinq observations différentes pour le même chiffre.

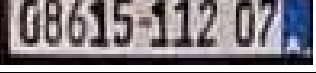
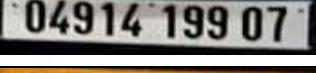
Pour qu'on puisse évaluer le module de la reconnaissance nous avons essayer d'extraire

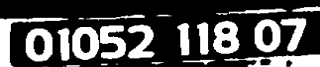
manuellement les zones d'image contenant les plaques. Les images extraites sont passés directement au module de la reconnaissance des chiffres. Le module va segmenter l'image de la plaque pour extraire les zones contenant des chiffres, et va passer par la suite chaque segment au moteur de reconnaissance à base de l'algorithme de k-plus proches voisins. Les résultats obtenus sont présents dans le tableau 4.3.

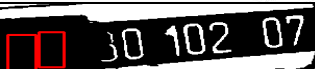
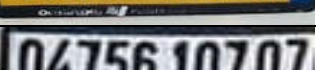
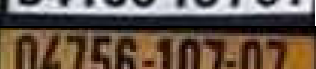
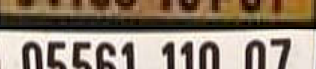
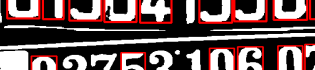
Tableau 4.3. : Résultats obtenus depuis le module de reconnaissance.

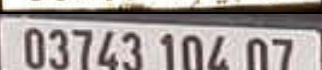
	Image du matricule	Segmentation des chiffres	Chaine détecté					
			K=1	C.C.	K=3	C.C.	K=5	C.C.
1			0045819987	08	9845849987	05	0045049907	09
2			45018807	06	45048807	05	45043307	05
3			5553822110077	09	3553822130077	08	3553822330077	07
4			8128211287	08	8128241207	08	0428244207	07
5			0943111387	09	0943111387	09	0943111307	10
6			8715818887	06	8715810887	07	8735830887	05
7			8715618887	07	8715838807	06	8715838807	06
8			042607	05	042407	05	042407	05
9			0789711407	09	0769731407	09	0789713407	08
10			0789711487	08	0789731487	07	0769711407	10
11			0148919907	10	0148015007	07	0148919907	10
12			Non détecté	00	Non détecté	00	Non détecté	00
13			Non détecté	00	Non détecté	00	Non détecté	00

14			0769311407	10	0769371407	09	0769372407	08
15			Non détecté	00	Non détecté	00	Non détecté	00
16			8348910487	07	8348930487	06	8340930407	08
17			8841318987	06	8841318987	06	0641310907	10
18			Non détecté	00	Non détecté	00	Non détecté	00
19			8614574947	05	0644574447	04	8644544947	04
20			0811518887	05	0841548987	04	0834548807	03
21			Non détecté	00	Non détecté	00	Non détecté	00
22			8843010807	07	8643040807	07	0843040307	07
23			0653310807	10	0653310807	10	0653310807	10
24			0653310807	10	0653310807	10	0653310807	10
25			Non détecté	00	Non détecté	00	Non détecté	00
26			Non détecté	00	Non détecté	00	Non détecté	00
27			847071898	06	847074898	05	047074090	08
28			Non détecté	00	Non détecté	00	Non détecté	00

29			Non détecté	00	Non détecté	00	Non détecté	00
30			Non détecté	00	Non détecté	00	Non détecté	00
31			6345810107	08	6346810107	08	6346810107	08
32			8340810107	09	0340810107	10	0340810107	10
33			0523210207	10	0523210207	10	0523210207	10
34			68415412070	07	68415412670	06	63415412678	06
35			0496810071067	09	0496810011007	10	0496800073067	08
36			10187	04	10107	05	10107	05
37			0494448907	07	0404443307	05	0404443307	05
38			8491419987	08	0491419987	09	0493439907	08
39			Non détecté	00	Non détecté	00	Non détecté	00
40			0128313307	08	0128343307	07	0423343307	06
41			022431607	09	022431607	09	022431607	09
42			0224311687	09	0224311607	10	0224311607	10
43			8888811487	05	8888814487	04	8806034407	07

44			8808811407	07	8808034407	06	0806034407	08
45			83382184871	07	83382184871	07	03382104071	10
46			833821848750	07	033821848760	08	033822048760	08
47			0843010807	08	8843040807	06	0843040807	07
48			1046511287	09	3048531207	07	1048531207	08
49			Non détecté	00	Non détecté	00	Non détecté	00
50			Non détecté	00	Non détecté	00	Non détecté	00
51			Non détecté	00	Non détecté	00	Non détecté	00
52			8871831807	06	8874634807	05	0074634807	07
53			8248838886	03	8248830885	05	0240830805	08
54			0882711107	09	0682711107	10	0882711107	09
55			0032711107	07	0682711107	10	0632111107	08
56			4328189877	06	4320139877	07	4320188077	07
57			8432018807	07	2422043907	06	0432048807	07
58			00088838007	07	00038333007	07	70038333007	06

59			880027210807	10	800027210887	09	880027210807	10
60			0027210807	10	8027218807	08	0027210807	10
61			0547110107	10	8547330307	06	0547330307	07
62			8444818887	06	0444810007	10	0444818087	08
63			Non détecté	00	Non détecté	00	Non détecté	00
64			8395710707	09	8895710707	08	0896710707	09
65			0785440607	08	0785440607	08	0785440807	07
66			8611611187	06	8613531187	05	0633531107	05
67			8511511187	08	8511577787	05	0511517107	09
68			4475614747	07	4476640707	07	4475640707	08
69			0475610707	10	0475640707	09	0475640707	09
70			8558111807	06	8558411007	07	0558434007	06
71			8558111987	06	0558333087	05	0558333007	06
72			015648807	07	045648807	06	045648807	06
73			050607	06	050807	05	050607	06

74			0437818887	06	0437030807	08	0437010807	09
75			0110518107	09	8110518407	07	0440548407	05
76			Non détecté	00	Non détecté	00	Non détecté	00
77			0841511187	08	0844543487	04	0844543407	05
78			Non détecté	00	Non détecté	00	Non détecté	00
79			4374240407	07	0374340407	09	8374340407	08
80			0484918147	09	8484978787	06	0484918787	08

4.3. Analyse des résultats et discussions

Nous allons discuter les résultats de localisation et de reconnaissance séparément :

4.3.1. Résultats de localisation

Pour les cas des plaques non détectées, nous avons constaté que cela est due à plusieurs facteurs tel que :

- a- Un des chiffres de la plaque est collé juste à la frontière de la plaque : Dans ce cas, et si le niveau de gris de la couleur juste à côté du chiffre est proche de celui de la voiture le contour de la plaque ne sera pas fermé, et par la suite ne sera pas détectée (Les cas A et B).
- b- Le niveau de gris de la plaque est proche de celui de la voiture : Dans ce cas, le contour de la plaque ne sera pas détecté ou le contour est détecté mais il ne sera pas fermé. La technique implémentée prend en compte, seulement, les contours fermés pour l'évaluation de la zone selon leurs dimensions. (Les cas C, D et E)
- c- La distance de la voiture par rapport à la caméra : Si la voiture est un peu loin de la caméra, le contour de la plaque (malgré s'il est détecté dans quelques cas) sera négligé par rapport aux autres contours les plus grands. La technique, et pour des raisons de performance de calcul, ne prend pas en compte tous les contours fermés : on garde dans notre cas seulement les 30 premiers plus grands contours fermés. (les cas F, G et H).
- d- Les dessin ou motifs ajoutés sur la plaque affectent la bonne détection parce qu'ils changent la forme rectangulaire de la plaque. La technique ne prend en compte que les contours à quatre côtés (espérant d'être des contours rectangulaires) pour l'évaluation. (les cas I et J).
- e- L'éclairage faible de la scène où la voiture est capturée peut entrainer des fausses détections de contours, surtout dans les cas des voitures à couleurs sombres. (les cas K et L).
- f- Parmi les cas de fausse détection, est le cas où le premier grand contour fermé répond aux critères (quatre côtés et facteur largeur/hauteur) sera considéré comme étant une zone de plaque. (le cas M)

4.3.2. Résultats de reconnaissance

Nous avons rencontré plusieurs problèmes concernant la partie reconnaissance. Ces problèmes sont classés en deux catégories :

a- **Problèmes de détection des segments représentant les chiffres** : Pour ce cas nous avons constaté qu'il existe plusieurs causes de ce problème tel que :

- La forme inclinée de la plaque qui peut conduire à détecter une fausse mesure de la hauteur de la plaque (les cas 6, 8, 13 et 15 par exemple). Le même cas se reproduit s'il y'a des zones supplémentaires détectées avec la plaque (les cas 34, 45, 46 et 59 par exemple). La solution consiste à détecter au premier lieu s'il y-a une capture inclinée de la plaque (en utilisant le transformé de Hough par exemple) et de faire corriger cette inclinaison pour avoir une partie plaque rectangulaire.
- Les seuils utilisés concernant les rapports « hauteur chiffre/largeur chiffre » et « hauteur plaque / hauteur chiffre » peuvent entraîner soit des fausses détections soit des chiffres non détectés (par exemple la modification du 2^{ème} seuil pour les cas 25, 26, 29 et 51 résulte d'une détection de tous les segments des chiffres). La solution est de faire dérouler le programme plusieurs fois en modifiant à chaque fois les seuils et les valeurs des paramètres utilisés pour avoir les meilleurs résultats.

b- **Problèmes de reconnaissance du chiffre correcte** : Cela est due à plusieurs raisons tel que :

- La forme inclinée du chiffre qui peut conduire à une fausse classification du chiffres (le cas 80 par exemple où le 1 à gauche est reconnu comme étant 7 et le zéro du gauche est reconnu comme étant 8). La solution consiste à essayer de corriger l'inclinaison des chiffres pour avoir un format correct de chaque chiffre (La correction de l'inclinaison de la plaque peut corriger ce problème).
- La taille petite des segments ou la qualité faible des images capturés (flou) peut entraîner des fausses identifications. Une des solutions est de capturer des images (vidéos) à haute résolution ce qui va entraîner des coûts supplémentaires de calcul, ou de faire des captures très proches à la voiture pour avoir des images de plaques plus grandes.

5. Conclusion

Nous avons présenté dans ce chapitre la démarche suivie pour implémenter un système de reconnaissance automatique des plaques d'immatriculation dans leurs côtés ; matériel et logicielle. Nous avons présenté séparément, par la suite, les résultats obtenus pour les deux modules ; localisation et reconnaissance. Enfin, nous avons fait une analyse des résultats obtenues en présentant les problèmes rencontrés et les solutions proposés pour améliorer le travail.

Conclusion générale

Le présent travail représente une étude sur les systèmes de reconnaissance automatique des plaques d'immatriculation suivi d'une conception et une implémentation de tel système sur un ordinateur à puce « Raspberry Pi » en tant qu'un système embarqué, dans lequel nous nous sommes intéressés à extraire la zone de la plaque en utilisant les contours et à identifier par la suite le matricule correspondant en utilisant l'algorithme des k-plus proche voisins. Premièrement nous avons exposé des généralités sur le domaine de traitement des images dont on a utilisé plusieurs techniques et concepts de bases dans ce domaine : Conversion en niveaux de gris, seuillage, détection de contours, ... etc. Ensuite nous avons introduit une étude technique sur « Raspberry Pi », dont on a présenté des généralités suivies d'un historique sur cet ordinateur à puce. Nous avons donné aussi une idée sur les composants électroniques du « Raspberry Pi » et aussi quelques projets réalisés qui sont réalisés avec. Pour tester et évaluer notre système de reconnaissance, nous avons utilisé une base de données des plaques d'immatriculation algérienne téléchargés depuis internet.

Parmi les systèmes RAPI existants nous avons choisi celui utilisant l'approximation des contours et les caractéristiques des plaques pour classifier et extraire le contour le plus approprié à une plaque, du fait que ces techniques nous offrent des temps de calculs un peu rapide par rapport à d'autres. Pour les systèmes de reconnaissances nous avons utilisé un algorithme de k-plus proches voisins du fait qu'il est facile à implémenter et à utiliser.

Du côté implémentation matériel nous avons utilisé un « Raspberry Pi 3 – B+ » la dernière version disponible sur le marché, du fait que cet ordinateur nous offre la possibilité de choisir les outils informatiques voulus pour la programmation, dont on a implémenté le système en utilisant le langage de programmation Python avec la bibliothèque de vision par ordinateur OpenCV ce qui a facilité le procédé d'implémentation.

Les résultats de reconnaissance obtenus montrent l'impact de plusieurs facteurs pour avoir une bonne localisation de la plaque et une bonne reconnaissance des chiffres de la plaque lui-même, dont on peut citer les suivants : Le bon éclairage au moment de la capture, les normes qui doivent être respectés pour la construction de la plaque, le choix des bons facteurs et des bons seuils pour les algorithmes implémentés, faire des captures de proche pour les véhicules.

Les résultats obtenus peuvent être jugé comme étant acceptables, pour un système complexe traitant une multitude de données pour localiser des zones composées sur des images (La plaque est une zone rectangulaire qui contient plusieurs segments à savoir qu'ils

soient des chiffres ou non). Malgré les efforts déployés pour améliorer les résultats obtenus on arrive qu'à détecter que 41 % des plaques des voitures utilisés comme base de teste, et quelques problèmes pour la reconnaissance des chiffres extraits de chaque plaque.

Comme perspective, et pour arriver à améliorer ce travail nous pouvons citer qu'il est possible de faire plusieurs autres expériences pour arriver à fixer les bons facteurs et les bons seuils. De même nous pouvons utiliser une base d'apprentissage plus large pour le moteur des k-plus proches voisins parce que la base utilisée comporte cinq observations pour chacune des dix classes des chiffres pour améliorer le taux de reconnaissance. De plus, nous pouvons tester d'autres techniques de localisation tel que celles qui basent sur les propriétés chromatiques des plaques et nous pouvons aussi tester d'autres techniques de reconnaissance des chiffres tel que les réseaux de neurones.

Annexe

L'algorithme des k-plus proches voisins (K Nearest Neighbors - kNN)

1. Découverte de l'algorithme K Nearest Neighbors

L'algorithme K-NN (K-nearest neighbors) est une méthode d'apprentissage supervisé[BEZ18]. Il peut être utilisé aussi bien pour la régression que pour la classification. Son fonctionnement peut être assimilé à l'analogie suivante "*dis moi qui sont tes voisins, je te dirais qui tu es...*".

Pour effectuer une prédiction, l'algorithme K-NN ne va pas calculer un modèle prédictif à partir d'un *Training Set* comme c'est le cas pour la régression logistique ou la régression linéaire. En effet, K-NN **n'a pas besoin de construire un modèle prédictif**. Ainsi, pour K-NN il n'existe pas de phase d'apprentissage proprement dite. C'est pour cela qu'on le catégorise parfois dans le **Lazy Learning**. Pour pouvoir effectuer une prédiction, K-NN **se base sur le jeu de données** pour produire un résultat.

2. Méthode de prédiction de K-NN

Pour effectuer une prédiction [BEZ18], l'algorithme K-NN va **se baser sur le jeu de données en entier**. En effet, pour une observation, qui ne fait pas parti du jeu de données, qu'on souhaite prédire, l'algorithme va chercher les **K instances du jeu de données les plus proches de notre observation**. Ensuite pour ces K voisins, l'algorithme se basera sur leurs variables de sortie (*output variable*) y pour calculer la valeur de la variable y de l'observation qu'on souhaite prédire.

Par ailleurs :

- Si K-NN est utilisé pour la régression, c'est la moyenne (ou la médiane) des variables y des K plus proches observations qui servira pour la prédiction.
- Si K-NN est utilisé pour la classification, c'est le mode des variables y des K plus proches observations qui servira pour la prédiction.

3. Ecriture algorithmique

On peut schématiser le fonctionnement de K-NN en l'écrivant en pseudo-code suivant [BEZ18]:

Début Algorithme

Données en entrée :

- un ensemble de données D .
- une fonction de définition distance d .
- Un nombre entier K

Pour une nouvelle observation X dont on veut prédire sa variable de sortie y Faire :

1. Calculer toutes les distances de cette observation X avec les autres observations du jeu de données D
2. Retenir les K observations du jeu de données D les proches de X en utilisation la fonction de calcul de distance d
3. Prendre les valeurs de y des K observations retenues :
 1. Si on effectue une régression, calculer la moyenne (ou la médiane) de y retenues
 2. Si on effectue une classification, calculer le mode de y retenues,
4. Retourner la valeur calculée dans l'étape 3 comme étant la valeur qui a été prédite par K-NN pour l'observation X .

Fin Algorithme

4. Calcul de similarité dans l'algorithme K-NN

Comme on vient de le voir dans l'écriture algorithmique, K-NN a besoin d'une fonction de calcul de distance entre deux observations. Plus deux points sont proches l'un de l'autre, plus ils sont similaires et vice versa [BEZ18].

Il existe plusieurs fonctions de calcul de distance, notamment, la distance euclidienne, la distance de Manhattan, la distance de Minkowski, celle de Jaccard, la distance de Hamming...etc. On choisit la fonction de distance en fonction des types de données qu'on manipule. Ainsi pour les données quantitatives (exemple : poids, salaires, taille, montant de panier électronique etc...) et **du même type**, la distance euclidienne est un bon candidat. Quant à la distance de Manhattan, elle est une bonne mesure à utiliser quand les données (*input variables*) ne sont pas du même type (exemple : âge, sexe, longueur, poids etc...).

Il est inutile de coder, soi-même ces distances, généralement, les librairies de Machine Learning comme **Scikit Learn**, effectue ces calculs en interne. Il suffit juste d'indiquer la mesure de distance qu'on souhaite utiliser.

Voici les définitions mathématiques des distances qu'on vient d'évoquer.

La distance euclidienne:

- Distance qui calcule la racine carrée de la somme des différences carrées entre les coordonnées de deux points :

$$D_e(x,y) = \sqrt{\sum_{j=1}^m (x_j - y_j)^2} \quad \text{.....Eq 01}$$

Distance Manhattan :

- La distance de Manhattan: calcule la somme des valeurs absolues des différences entre les coordonnées de deux points :

$$D_m(x,y) = \sum_{i=1}^k |x_i - y_i| \quad \text{.....Eq 02}$$

Distance Hamming :

- la distance entre deux points données est la différence maximale entre leurs coordonnées sur une dimension.

$$D_h(x,y) = \sum_{i=1}^k |x_i - y_i| \quad \text{.....Eq 03}$$

Avec

$$- \quad x = y \quad \Rightarrow \quad D = 0$$

$$- \quad x \neq y \quad \Rightarrow \quad D = 1$$

Notez bien qu'il existe d'autres distances selon le cas d'utilisation de l'algorithme, mais la **distance euclidienne** reste la plus utilisée.

5. Comment choisir la valeur K ?

Le choix de la valeur K à utiliser pour effectuer une prédiction avec K-NN, varie en fonction du jeu de données. En règle générale, moins on utilisera de voisins (un nombre K petit) plus on sera sujette au sous apprentissage (underfitting). Par ailleurs, plus on utilise de voisins (un nombre K grand) plus, sera fiable dans notre prédiction. Toutefois, si on utilise K nombre de

voisins avec $K=N$ et N étant le nombre d'observations, on risque d'avoir du overfitting et par conséquent un modèle qui se généralise mal sur des observations qu'il n'a pas encore vu [BEZ18].

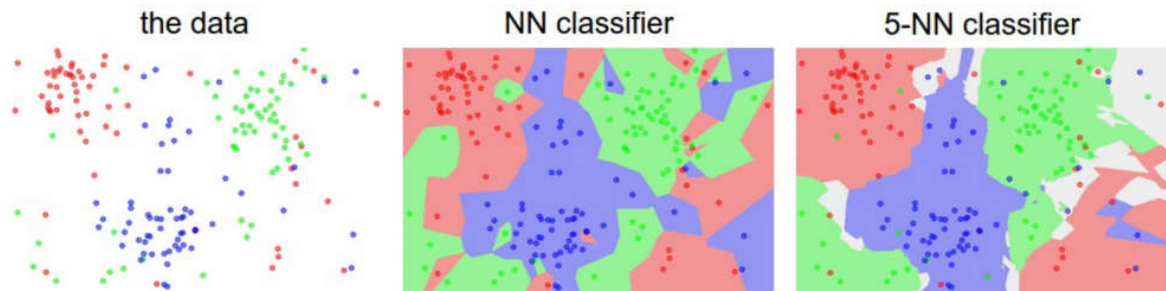


Figure A.1 : Différence entre N-NN classifieur et 5-NN classifieur. [BEZ18]

Sur la figure 1 ci-dessus, l'image à gauche représente des points dans un plan 2D avec trois types d'étiquetages possibles (rouge, vert, bleu). Pour le 5-NN classifieur, les limites entre chaque région sont assez lisses et régulières. Quant au N-NN Classifieur, on remarque que les limites sont "chaotiques" et irrégulières. Cette dernière provient du fait que l'algorithme tente de faire rentrer tous les points bleus dans les régions bleues, les rouges avec les rouges etc... c'est un cas d'overfitting.

Pour cet exemple, on préférera le 5-NN classifieur sur le NN-Classifieur. Car le 5-NN classifieur se généralise mieux que son opposant.

6. Exemple d'application

Les figures 1 et 2 illustrent comment la méthode des k plus proches voisins résout une tâche de classification supervisée binaire ($y_i \in \{1,2\}$) dans $\mathbb{R}^2$, selon la distance euclidienne et pour $K=3$. La figure 1 montre comment (en 3 étapes, de gauche à droite) une nouvelle observation est attribuée à la classe 2, son voisinage étant composé de 3 observations appartenant à la classe 2. La figure 2 montre comment une nouvelle observation est attribuée à la classe 2, son voisinage étant composé de 2 observations appartenant à la classe 2 et d'une observation appartenant à la classe 1. [GUIW19]

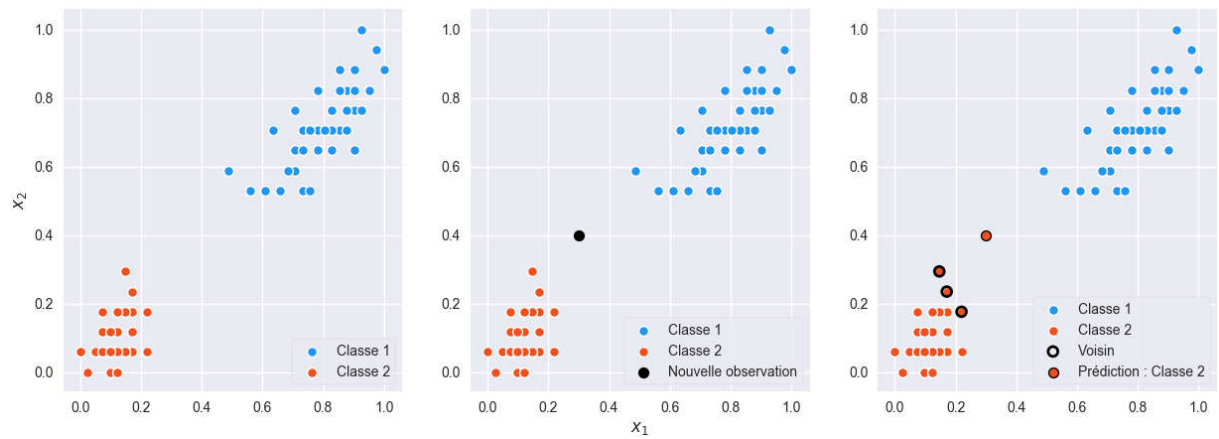
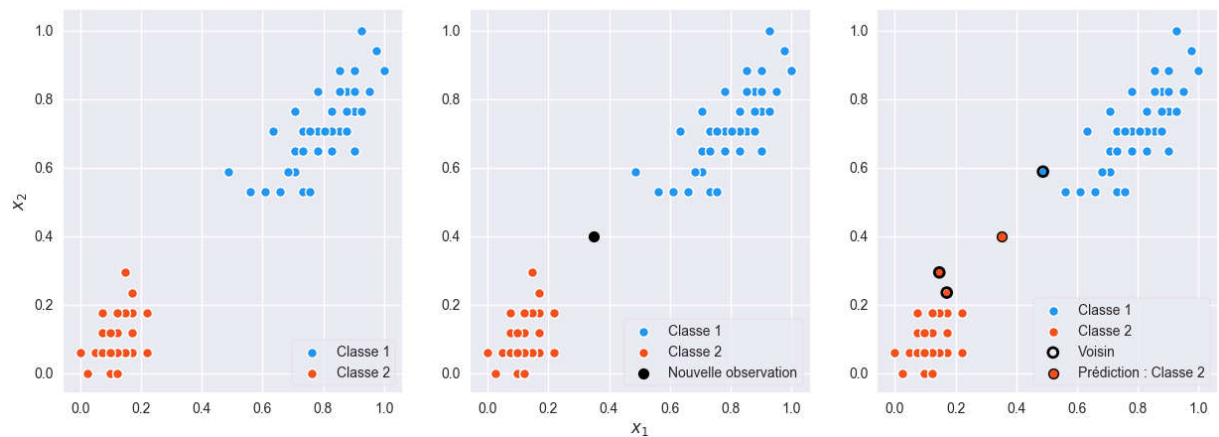


Figure A.2 : Classification d'une nouvelle observation : les 3 plus proches voisins appartiennent à la classe 2.

[GUIW19]



FigureA.3 : Classification d'une nouvelle observation : 2 des 3 plus proches voisins appartiennent à la classe 2.

[GUIW19]

7. Limitations de K-NN

K-NN est un algorithme assez simple à appréhender [BEZ18]. Principalement, grâce au fait qu'il n'a pas besoin de modèle pour pouvoir effectuer une prédiction. Le contre coût est qu'il doit **garder en mémoire l'ensemble des observations** pour pouvoir effectuer sa prédiction. Ainsi il faut faire attention à **la taille du jeu d'entraînement**.

Également, le choix de la méthode de calcul de la distance ainsi que le nombre de voisins K peut ne pas être évident. Il faut essayer plusieurs combinaisons et faire du tuning de l'algorithme pour avoir un résultat satisfaisant.

Bibliographie

- [BAI04] BAI, H LIU C."A hybrid license plate extraction method based on edge statistics and morphology in :proceedings of the 17th international conference on pattern recognition" vol 2, pp, 831 ,23-26Aug 2004
- [BEA04] Beatriz, A"Experiments in image segmentation for automatic US license plate recognition." Dissertation, Virginia Polytechnic Institute and state university 2004
- [BEN18] Ben khalfa Ismail, " Réalisation d'un système de contrôle autour de Raspberry Pi pour ladomotique", Mémoire de Master en électronique (Electronique des systèmes embarqués), Université Mohamed Khider Biskra, Juin 2018.
- [BER15] Maïtine Bergounioux, " Introduction au traitement mathématique des images – méthodes déterministes", Edition Springer, 2015.
- [BEZ18] Younes Benzaki, "Introduction à l'algorithme K Nearest Neighbors (K-NN)", Article sur le blog Mr. Mint, Octobre 2018, consulté le : 15-06-2019.
U.R.L. : <https://mrmint.fr/introduction-k-nearest-neighbors>
- [BRO05] Broumandnia A, Fathy M : "Application of pattern recognition for Farsi license plate recognition " In Vision and image processing (GVIP) December 2005.
- [CHE07]. Mohamed cheriet, Nawwaf kharma, Cheng lin liu, Ching y suen, " Character Recognition System" 2007
- [CHH09] Ameer CHHAYDER et Imene BELHADJ MOHAMED, "Système deReconnaissance Automatique des PlaquesMinéralogiques", Communication, 5th International Conference: Sciences of Electronic, Technologies of Informationand Telecommunications, Tunisie, Mars 2009.
- [CHO03] Chacon I , Zimmerman A"License plate location based on a dynamic PCNN schema " In : proceedings of the international join conference on neural networks, Vol 2 pp 1195-1200, 20-24 July 2003.
- [COP19] Yannick Copin, "Analyse scientifique avec Python", Cours en ligne, Février 2019.
URL : <https://buildmedia.readthedocs.org/media/pdf/informatique-python/m2/informatique-python.pdf>
- [DAI01] Dai, Y., Ma, H., Liu, J., Li, L., "A high performance license plate recognition system based on the web technique". In: Proceedings of IEEE Intelligent Transportation Systems, pp. 325–329 (2001)
- [DIA04] Lingrand Diane, " COURS DE TRAITEMENT D'IMAGES", Rapport de recherche, Laboratoire I3S, Univeristé de Nice, Janvier 2004.
- [DLA02] Dlagnekov, L., "Video-based car surveillance: License plate makes, and model recognition". Dissertation, U.C. San Diego. AdaBoost and a detector cascade. In: NIPS, vol. 14 (2002).

- [DUP10] Eve Mathieu-Dupas, "Algorithme des k plus proches voisins pondérés et application en diagnostic", Article au 42èmes Journées de Statistique, Laboratoire INRIA 2010, Marseille, France.
- [DVA05] David, L., Andres, M., Vicente, P., Juan, V.: "Car license plates extraction and recognition based on connected components analysis and HMM decoding". In: IbPRIA, vol. 1, pp. 571–578 (2005)
- [EBO] Contributeurs de Stack Overflow, "Apprenez OpenCV", eBook gratuit.
URL : <https://riptutorial.com/Download/opencv-fr.pdf>
- [ELL90] Elliman, G., Lanxaster, T.: "A review of segmentation and contextual analysis techniques for text recognition". In: Pattern Recognition, vol. 23, No. 3337–346 (1990).
- [ETC08] J P Gastellu-Etchegorry, "Acquisition et traitement d'image numerique", Avril 2008
- [FUC19] Patrick Fuchs et Pierre Poulain, "Cours de Python", Cours de Université Paris Diderot-Paris 7 – Paris – France, Mai 2019.
URL : <https://python.sdv.univ-paris-diderot.fr/cours-python.pdf>
- [GUIW19] Adrien Guille, "Méthode des k plus proches voisins", Cours en ligne, *Université Lyon 2*, consulté le : 15-06-019.
U.R.L. : http://mediamining.univ-lyon2.fr/people/guille/m4101/k_plus_proches_voisins.html
- [HEB17] Patrick Hébert et Denis Laurendeau, "Traitement des images (Partie 1: pré-traitement)", Cours de traitement d'images, Université Laval, Mai 2017.
URL : <http://wcours.gel.ulaval.ca/2017/a/GIF4100/default/5notes/A2017TraitementImagesPartie1PageWeb.pdf>
- [JFC86] J.F Canny "A computational approach to edge detection", IEEE transaction on pattern analysis and machine intelligent, November 1986
- [JSE82] J.Serra, "Image Analysis and Mathematical Morphology", Academic Press, London, 1982.
- [JXI05] JXifan, S., Weizhong, Z., Yonghang, S., "Automatic license plate recognition system based on color image processing". ICCSA4, 1159–1168 (2005).
- [LUD13] Simon Ludovic, "Systèmes et équipements de lecture automatique de plaques d'immatriculation des véhicules Principes et aperçu des applications", Rapport d'études, SETRA, Ministère de l'Ecologie – du Développement durable et de l'Energie – France, Mai 2013.
- [MAT05] Matas, J., Zimmermann, K., "Unconstrained licence plate Detection", In: Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems. pp. 572–577, isbn 07803-9216-7, Wien Austria (2005)
- [MAZ18] Vincent Mazet, "Traitements et outils de base", Cours de la matière Outils fondamentaux en traitements d'images, Université de Strasbourg, 2018-2019.
- [NIJ95] Nijhuis, J., Brugge, M., Helmholt, K., Pluim, J., Spaanenburg, L., Venema, R., Westenberg M., "Car license plate recognition with neural networks and fuzzy logic", In: Proceedings of IEEE Inter-national Conference on Neural Networks, vol. 5, pp. 2232–2236 (1995)

- [NOR17] NOR Imane et SIDHOUM Souad, "Développement d'une application de détection et de reconnaissance de plaques d'immatriculation (LAPIA)", Mémoire Master en Informatique (SIC), Université Abou Bakr Belkaid– Tlemcen, Juillet 2017.
- [OTS79] Otsu, N.: A threshold selection method from Gray level histograms. In: IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-9, pp. 62–66 (1979)
- [PAJ17] Ashwin Pajankar, "Raspberry Pi, Image Processing, Programming", Edition APress, 2017.
- [PAM02] P. A. Mlsna and J. J. Rodriguez, "Gradient and Laplacian type edge detection", In handbook of Image and Video Processing, Academic Press (2002)).
- [PAR96] Parker, J., Federl, P., "An approach to license plate recognition". In: Computer Science Technical reports, University of Calgary, Alberta Canada, 591–11, October (1996).
- [PAU1] Ian Paul, "8 projets Raspberry Pi vraiment innovants", Article sur web, TOUTE L'ACTUALITÉ / HARDWARE, publié le 28 Juin 2017.
URL : <https://www.lemondeinformatique.fr/actualites/lire-8-projets-raspberry-pi-vraiment-innovants-68673.html>
- [SHY04] Shyang-Lih, C., Li-Shien, C., Yun-Chung, C., Sei-Wan, C. Automatic license plate recognition. In: IEEE Transactions on Intelligent Transportation Systems, vol.5, No.1, pp.42–53(2004)
- [SUN17]. D Sundarajan, "Digital Image Processing", A signal processing and algorithmic approach, 2017
- [SWI10] Gérard Swinnen, "Apprendre à programmer avec Python 3", Edition Eyrolles (Paris), 2010.
URL : <http://inforef.be/swi/python.htm>
- [UPT17] Eben Upton, "Raspberry Pi, Le guide de l'utilisateur", Edition Dunod, 2017.
- [WIRA19] Article disponible sur: https://fr.wikipedia.org/wiki/Raspberry_Pi. La dernière modification de cette page a été faite le 08 AVRIL 2019.
- [WITR18] Article disponible sur: https://fr.wikipedia.org/wiki/Traitement_d%27images, La dernière modification de cette page a été faite le 12 décembre 2018.