

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Télécommunication
Spécialité Systèmes & Télécoms

Présenté par

HATTAB Bey Rania

&

CHETTIH Fadia

Restauration des images des codes à barres 1D , application au décodage de code EAN13 .

Proposé par : Dr. NAMANE Abderrahmane

Année Universitaire 2020-2021

Remerciements

Nous tenons à exprimer nos sincères gratitudee à notre encadreur Mr A. Naaman, pour son précieux soutien, et sa disponibilité tout au long de ce travail. Nous tenant a remercié les responsables de famili shop de nous avoir bien reçu. Nos remerciements vont aussi aux membres de jury qui ont accepté de lire et d'évaluer de modeste travail, et de nous faire part de leur précieuse remarque. Sans oublier tous les professeurs du département Génie Electrique, et particulièrement les professeurs de l'option « Télécommunication ». Nous remercions chaleureusement nos parents pour leurs soutiens durant tout le cursus universitaire et pour tous ceux qui ont contribué de près et de loin à la réalisation de ce travail.

Dédicace :

Je dédie ce modeste travail à :

Mes chers parents

Mon cher mari

Mon frère et mes sœurs

Mes beaux parents

Tous ceux qui m'ont encouragé pour accomplir ce mémoire

Rania Hattab-bey

Je dédie ce modeste travail à :

Mes chers parents

Mes sœurs

Mes copines Chaima et Imene

Tout mes amis

Tous ceux qui m'ont aidé pour accomplir ce mémoire

Fadia Chettih

ملخص:

في مشروع التخرج هذا ، نقتراح حلاً يتمثل في استعادة صور الباركود EAN13 التالفة التي تظهر على المنتجات الاستهلاكية. يحتوي الرمز الشريطي المعني على عيوب تجعل من المستحيل فك شفرته بواسطة ماسحات الليزر الموجودة في مستوى الخروج. نقدم في هذا العمل استعادة أو تصحيح الباركود باستخدام معالجة الصور. لهذا الغرض ، يجب الاحتفاظ بالرمز الشريطي في وضع أفقي يعطي أشرطة عمودية على الصورة التي تم الحصول عليها. تم استخدام العديد من التقنيات ، ولا سيما الترشيح الوسيط والتشكيل الرياضي لاستعادة أو تصحيح الرمز الشريطي من أجل جعله قابلاً للفك. يتم إجراء اختبار الجدوى بواسطة وحدة فك ترميز رمز EAN13 . النتائج التي تم الحصول عليها مرضية .

كلمات المفاتيح: الباركود , ماسحات الليزر , معالجة الصور

Résumé :

Dans ce projet de fin d'études nous proposons une solution qui consiste à la restauration des images codes-barres EAN13 abîmés figurant sur les produits de consommation. Le code à barre en question présente des défauts ce qui le rend impossible d'être décodé par les lecteurs laser se trouvant au niveau des caisses. Nous présentons dans ce travail la restauration ou correction du code à barres en utilisant le traitement d'images. A cet effet, le code à barres doit être maintenu en position horizontale ce qui donne des barres verticales sur l'image acquise. Plusieurs techniques, notamment, le filtrage médian et la morphologie mathématique ont été utilisés pour restaurer ou corriger le code à barre afin de le rendre décodable. Le test de faisabilité est assuré par un décodeur de code EAN13. Les résultats obtenus sont satisfaisants .

Mots clés : code-barres , les lecteurs laser , traitement d'image.

Abstract :

In this graduation project, we are proposing a solution which consists in restoring damaged EAN13 barcode images appearing on consumer products. The bar code in question has defects which make it impossible to be decoded by laser scanners located at the checkout level. We present in this work the restoration or correction of the barcode using image processing. For this purpose, the bar code must be kept in a horizontal position which gives vertical bars on the acquired image. Several techniques, notably, median filtering and mathematical morphology have been used to restore or correct the bar code in order to make it decodable. The feasibility test is ensured by an EAN13 decoder. The results obtained are satisfactory.

Keywords : barcode , laser , image processing .

Listes des acronymes et abréviations :

EAN13 : European Article Numbering

TD : taux de décodage

Table des matières :

Introduction générale.....	1
<u>Chapitre 1 : généralités sue les codes à barres</u>	
1.1.Introduction.....	3
1.2. Historique	3
1.3. Définition de code à barres	4
1.4. Avantages de code à barres.....	4
1.5. Rôle du code à barres.....	4
1.6 .Comment fonctionne le code à barres.....	5
1.7. Choix de la taille du code à barres	5
1.8. Positionnement du code à barres :.....	6
1.9. Choix des couleurs :.....	7
1.10. Lecteurs de code à barres :.....	9
1.10.1. Crayon lecteur :.....	9
1.10.2. Lecteur CCD :	9
1.10.3. Lecteur laser :	10
1.11. Principaux standards :	11
1.11.1. Codes-barres unidimensionnels ou linéaires :	11
1.11.2. Codes-barres linéaires empilés :.....	11
1.11.3. Codes à barres à deux dimensions :.....	11
1.12. Normalisation EAN 13 :.....	14
1.12.1. Calcul du caractère de contrôle :.....	14
1.12.2. Système de codage :.....	15
1.13. Conclusion:.....	18

Chapitre 2 : traitement d'images

2.1. Introduction.....	20
2.2. Définition de l'image :.....	20
2.3. Image numérique (numérisée) :	20
2.4 .Acquisition d'une image :.....	21
2.5 .Caractéristiques d'une image numérique :.....	22
2.5.1. Pixel :.....	22
2.5.2 .Dimension :.....	22
2.5.3 .Texture :.....	22
2.5.4 .Résolution :.....	22
2.5.5 .Bruit :.....	23
2.5.6 .Histogramme :.....	23
2.5.7 .Luminance :.....	24
2.5.8 .Contraste :.....	24
2.6. Types d'images :.....	24
2.6.1. Images binaires (en noir et blanc) :.....	25
2.6.2 .Images à niveaux de gris :.....	25
2.6.3 .Images en couleurs :.....	26
2.7. Passage d'une image couleur vers une image à niveaux de gris :	27
2.8. Prétraitement d'images :.....	28
2.8.1 Réduction du bruit par filtrage :.....	28
2.8.2 Filtres linéaires stationnaires	28
2.8.3 Filtres non linéaires stationnaires :.....	30
2.9. Morphologie mathématique (Filtres morphologiques) :.....	31

2.9.1. Filtres morphologiques :.....	31
2.9.2. Application du filtre morphologique sur des images de barres.....	36
2.10. Quelques exemples concrets du traitement d'images :.....	37
2.11. Domaine d'application :.....	37
2.11.1 Domaine militaire	37
2.11.2 Domaine médical.....	37
2.12. Conclusion :.....	38
<u>Chapitre 3 : application et test</u>	
3.1. Introduction.....	40
3.2 .Environnement du Travail	40
3.3. Logiciel Matlab	41
3.4. Système proposé	41
3.4.1. Présentation de l'application capture _p	41
3.4.2. Etapes à suivre.....	42
3.5. Applications et test	43
3.6. Résultats de l'application de notre système.....	43
3.6.1. Images synthétisées	43
3.6.2. Images réelles.....	48
3.7 . Résultats obtenus.....	55
3.8. Conclusion	58
Conclusion générale	59
Références	60

Liste des figures :

Chapitre 1 : généralités sue les codes à barres

Figure 1.1: Facteur de grossissement 0,80 (80%).....	6
Figure 1.2 : Facteur de grossissement 1,00(100%).....	6
Figure 1.3 : Cas général.....	7
Figure 1.4 : Cas cylindrique.....	7
Figure 1.5 : Crayon lecteur	9
Figure 1.6 : Lecteur CCD.....	10
Figure 1.7 : Lecteur laser.....	10
Figure 1.8 : Un code à barres à la norme EAN 13.....	14
Figure 1.9 : composition de la norme EAN 13.....	16

Chapitre 2 : traitement d'images

Figure 2.1 : Représentation d'une image numérique.....	21
Figure 2.2: Groupe de pixel formant la lettre A.....	22
Figure 2.3: (a) Image sans bruit (b) Image avec bruit.....	23
Figure 2.4 : Image avec son histogramme.....	24
Figure 2.5 : Image binaire.....	25
Figure 2.6 : Image monochrome (256 couleurs).....	26
Figure 2.7 : Image polychrome (65536 couleurs).....	26
Figure 2.8 : Conversion des images en couleur vers des images à niveaux de gris.....	27
Figure 2.9 : Exemple de filtre moyennneur.....	29
Figure 2.10 : Exemple de filtre gaussien.....	29
Figure 2.11: Application du filtre médian.....	30
Figure 2.12: Résultat de l'application du filtre médian.....	31

Figure 2.13: Image formée d'objets.....	31
Figure 2.14: Eléments structurants.....	32
Figure 2.15 : (a) image initial, (b) :image après l'opération érosion , (c) : image après l'opération dilatation	33
Figure 2.16: Analyse d'images par morphologie (a) Image d'entrée binaire. (b) Résultat d'ouverture avec un disque de 9x9. (c) Résultat de fermeture de l'image (b) avec un disque de 9x9.....	34
Figure 2.17 : Prétraitement de l'image acquise par caméra ; (a) Image binarisée. (d) Résultat de fermeture avec un élément structurant disque de taille 7x7.....	34
Figure 2.18: Résultats de l'application d'ouverture morphologique sur l'image (a) pour des éléments structurants ; (b) 3x1 (c) 5x1 (d) 7x1 (e) 9x1 (f) 11x1 (g) 13x1 (h) 31x1.....	36
 <u>Chapitre 3 : application et test</u>	
Figure 3.1 : Ordinateur portable hp EliteBook820	40
Figure 3.2 : Application de la procédure de décodage capture_ p	41
Figure 3.3 : illustration d'organigramme du système proposé	42
Figure 3.4 : Schéma synoptique représentant le traitement d'images sur Matlab.....	43
Figure 3.5 : (a) image avant le traitement, non décodée, (b) image après le traitement (ouverture) , bien décodée 59 fois 6 130646 000745 , (c) image après le traitement (ouverture/fermeture) , bien décodée 59 fois 6 130646 000745. (d) image après le traitement (médian), non décodée.....	44
Figure 3.6 : (a) image avant le traitement, non décodée, (b) image après le traitement (fermeture/ouverture) , bien décodée 75fois 6 130646 000745 , (c) image après le traitement (fermeture), bien décodée 75 fois 6 130646 000745. (d) image après le traitement (médian), bien décodée 76 fois 6 130646 000745.....	45
Figure 3.7 : (a) image avant le traitement, non décodée, (b) image après le traitement (fermeture), bien décodée 75 fois 6 130646 000745, (c) image après le traitement (fermeture/ouverture), bien décodée 75 fois 6 130646 000745. (d) image après le traitement (médian) ,bien décodée 73 fois 6 130646 000745..	46

Figure 3.8 : (a) image avant le traitement, non décodée , (b) image après le traitement (fermeture), bien décodée 4 fois 6 130646 000745 , (c) image après le traitement (ouverture) , bien de décodée 4 fois 6 130646 000745 , (d) image après le traitement (fermeture/ouverture) , bien de décodée 7 fois 6 130646 000745 , (e) image après le traitement (ouverture/fermeture) , bien de décodée 13 fois 6 130646 000745. (f) image après le traitement (médian) , bien de décodée 63 fois 6 130646 000745**47**

Figure 3.9 : (a) image avant le traitement, non décodée, (b) image après le traitement (ouverture), bien décodée 7 fois 6 130766 703694, (c) image après le traitement (ouverture/fermeture), bien décodée 4 fois 6 130766 703694.....**49**

Figure 3.10 : (a) image avant le traitement, non décodée , (b) image après le traitement (ouverture/fermeture), bien décodée 1 fois 6 135116 000645.....**49**

Figure 3.11 : (a) image avant le traitement , non décodée, (b) image après le traitement (fermeture/ouverture) , bien décodée 4 fois 6 131005 000048.....**50**

Figure 3.12 : (a) image avant le traitement, non décodée , (b) image après le traitement (ouverture) bien décodée 3 fois 6 135367 000128, (c) image après le traitement (ouverture/fermeture), bien décodée 2 fois 6 135367 000128 , (d) image après le traitement (fermeture/ouverture), bien décodée 1 seul fois 6 135367 000128.....**50**

Figure 3.13 : (a) image avant le traitement, non décodée , (b) image après le traitement (fermeture), bien décodée 15 fois 4 002810 111972 , (c) image après le traitement (fermeture/ouverture), bien décodée 20 fois 4 002810 111972, (d) image après le traitement (ouverture) , bien décodée 8 fois 4 002810 111972, (e) image après le traitement (ouverture/fermeture) , bien décodée 4 fois 4 002810 111972**51**

Figure 3.14 : (a) image avant le traitement , non décodée , (b) image après le traitement (fermeture), bien décodée 1 seul fois 6 134539 000720, (c) image après le traitement (fermeture/ouverture), bien décodée 1 seul fois 6 134539 000720 , (d) image après le traitement (ouverture) , bien décodée 1 seul fois 6 134539 000720, (e) image après le traitement (ouverture/fermeture) , bien décodée 1 seul fois 6 134539 000720**52**

Figure 3.15: (a) image avant le traitement, non décodée , (b) image après le traitement (ouverture/fermeture) bien décodée 3 fois 6 135116 000898, (c) image après le traitement (fermeture), bien décodée 2 fois 6 135116 000898, (d) image après le traitement (fermeture/ouverture), bien décodée 2 fois 6 135116 000898.....**53**

Figure 3.16: (a) image , bien décodée sans traitement, 6 135263 000147 ,(b) image , bien décodée sans traitement ,6 135636 000118 ,(c) image , bien décodée sans traitement, 6 132004 000022 , (d) images décodées , sans aucun traitement , 6 130004 300111.....**54**

Figure 3.17 : Nombres d'images correctement décodé et non ou mal décodé.....**55**

Figure 3.18 : Première image non décodée, problème de codage ,6 132002 900146.....**56**

Figure 3.19 : deuxième image non décodée, problème de codage , 1 234567 890101**56**

Figure 3.20 : image non décodée, problème d'impression, manque de 2 barres au début du code barres.....**57**

Figure 3.21 : image non décodée, problème d'impression, manque de 2 barres au début du code-barres.....**57**

Listes des tableaux :

Chapitre 1 : généralités sue les codes à barres

Tableau 1.1 : table montre comment choisir les couleurs des codes-barres.....8

Tableau 1.2 : table résume quelques type des codes-barres et leurs applications ainsi que les particularités.....13

Tableau 1.3 : Codage du jeu (les codes des jeux A et B commencent par un espace, alors que ceux de la jeu C commencent par une barre)16

Tableau 1.4 : indique l'utilisation des tables A ou B en fonction du chiffre N°1.....17

Tableau 1.5 : Codage du jeu de caractères en binaire18

Chapitre 2 : traitement d'images

Tableau 2.1: comparaison entre érosion et dilatation35

Tableau 2.2: comparaison entre ouverture et fermeture.....35

Introduction générale

A l'heure actuelle, les systèmes de visions sont de plus en plus répandus ; système de vidéosurveillance, lecture de code à barres sur smart phone (QR code) et les caméras se sont installées partout dans notre quotidien. Elles sont utilisées pour réaliser de la lecture de codes à barres dans les magasins, rues ou aéroports.

Le code à barres est aujourd'hui la solution technique la plus utilisée pour saisir automatiquement une information. Cet état de fait a pour base l'informatisation des entreprises que nous avons connu ces dernières années associées aux normes de qualité du code à barres réalisant la liaison entre le papier et l'informatique.

La reconnaissance des images à partir de code a barre est considérée actuellement comme l'une des tâches les plus importantes pour identifier un article. On remarque que dans la vie quotidienne chacun de nous achète au moins un article dans la journée, des fois nous faisons attention dans les caisses rencontre des problèmes l'ordinateur ne connaît pas le code a barre. La difficulté de la reconnaissance de l'image par ordinateur varie énormément suivant les conditions d'acquisition.

Dans notre travail, nous présentons une solution à la lecture automatique de code à barres abîmé ou dégradé de type EAN13. Cette dégradation peut provenir de rayures dues au déplacement des produits, ou de taches et de poussières. Le système proposé peut lire ces codes à barres moyennant une ou plusieurs corrections appliquées à l'images de code à barres acquise par une webcam ou smartphone. Les produits utilisés proviennent de l'hypermarché FAMILI shop dans lequel nous avons effectué un stage.

Ce mémoire est établi sur trois chapitres. Dans le premier chapitre, nous présentons les généralités sur les codes à barres. Le deuxième chapitre est consacré au traitement d'images dédiés à la correction de l'image de codes à barres. Dans le dernier chapitre, nous présentons la réalisation et les différents tests de simulation et l'application sur des images réelles.

Et enfin, une conclusion récapitule notre travail et présente les connaissances acquises de ce projet fin d'étude ainsi des perspectives relatives à ce projet.

Chapitre 1 :

Généralités

sur

les codes-

barres.

1.1. Introduction :

Les systèmes code à barres deviennent de plus en plus courants dans tous les domaines de la vie. Ils sont faciles à utiliser, permettent d'entrer des données beaucoup plus vite que par des méthodes manuelles et sont très fiables [1]. Leur rapidité et fiabilité permettent d'améliorer de nombreuses opérations telles que fabrication, expéditions, réception et emballage, préparation de commandes, inventaire, gestion de dossiers....

On trouve les codes à barres dans les supermarchés, les bibliothèques, les banques de sang, les usines, les ponts de chargements. Ils sont utilisés avec des appareils de lecture de code à barres en mode manuel ou automatique [3]. Le terme de codes à barres est utilisé pour décrire un large éventail de techniques de cryptage de données lisibles par machines optiques.

1.2. L'historique :

C'est en 1948 que le président d'une chaîne de magasin alimentaire émit le besoin d'une solution de saisie automatique des données relatives aux produits passant aux caisses de ses magasins [2]. La soustraction des sorties aux entrées des produits lui aurait procuré un état des stocks permanent et économe. L'idée n'était pas nouvelle, le système des cartes perforées développées en 1890 pour réaliser le recensement des Etats-Unis avait déjà permit une telle automatisation.

Norman Joseph Woodland professeur au « Philadelphia's Drexel Institut of Technologie » en sera informé et se passionnera pour le projet auquel il a consacré à plusieurs mois avant de présenter une solution dénommée « code à barre linéaire », évolution détournée et fusion du système de codage des bandes son de film et du code morse.

Le premier brevet d'application est déposé en 1949. Inexploitable en l'état, il faut attendre 1969 pour que la technologie mûrisse. La société Computer Identique Corporation installe les premiers lecteurs de codes à barres « fait main » utilisant un rayon laser d'un milliwatt. Quatre ans plus tard, « l'Universal Product Code » développé par IBM aidé de Woodland salarié de l'entreprise et inventeur du code à barres, standardise la codification. L'avènement de l'outil informatique permettant de stocker, traduire et manipuler cette codification permet alors aux fournisseurs des différents secteurs de la distribution de rentabiliser rapidement les investissements réalisés dans l'identification de leurs produits. De 1970 à 1980 tous les standards de codification sont développés, multipliés pour répondre aux exigences des différents secteurs d'activités [2]. De 1988 à 1995 les codes dit « empilés » et « matriciels » voient le jour pour répondre à un volume d'information à encoder toujours plus important ; les uns reprenant la même technologie de lecture, les autres nécessitant une caméra ! Dans un même temps, les standards Européens et mondiaux sont arrêtés en 1993 se limitant au premier type code inventé par Woodland, le code à barres linéaire.

1.3. Définition de code à barres :

Un code à barres, souvent appelé " code barres ", représente la codification graphique d'une information. Variant selon les algorithmes de codage, cette codification est optimisée selon les besoins pour encoder du texte, des chiffres, des caractères de ponctuation ou encore une combinaison de ses derniers [1].

La représentation obtenue est optimisée pour une lecture optique. Les barres doivent contraster avec les espaces, ce qui explique que celles-ci soient souvent noires sur fond blanc. Il faut savoir que les espaces peuvent également codifier l'information.

Le code à barres est un numéro d'identification, une clé d'accès à un enregistrement préalable dans une base de données, utile aux fabricants comme aux commerçants. Le code à barres n'est donc pas une étiquette comme les autres [2]. Il ne contient ni le secteur d'activité, ni la taille de l'entreprise.

1.4. Avantages du code à barres :

- Fiabilité et rapidité de lecture (avec la possibilité d'une erreur de lecture sur 20 000 à 2 000 000 de codes lus [2], selon la codification)
- Garantie de lecture du code grâce à la redondance en hauteur même si une partie est illisible (défauts d'impression, des déchirements légers ou encore des frottements)
- Facilité et le faible coût d'impression du code à barres
- Des dimensions variables, des lectures par différents types de lecteurs, une information utile quelque soit l'orientation du code...
- Tous types de caractères peuvent être codés

1.5. Rôle du code à barres :

Le code à barres permet de limiter les temps de saisie nécessaires au suivi d'un produit dans un processus de fabrication ou d'un document devant circuler au sein d'un service, d'une entreprise ou d'un couple fournisseur et client [2].

La lecture automatique autorise l'interconnexion des systèmes d'information et leur mise à jour en temps réel.

- Dans la grande distribution, le code à barres permet entre autres :
 - d'entrer rapidement un produit en stock,
 - d'en connaître l'origine,
 - d'en faciliter le réapprovisionnement,

- d'automatiser la sortie des stocks et d'obtenir un inventaire simplifié permanent,
- d'optimiser les temps de traitement aux caisses ...
- Dans une usine de fabrication, le code à barres permet également :
 - de connaître l'origine des matières premières,
 - d'avoir en continu des informations actualisées sur l'état de la production,
 - d'aiguiller automatiquement un produit en fonction de ses caractéristiques, de ses défauts,
 - d'identifier sa production,
 - d'identifier les unités logistiques transmises à ses clients.

1.6. Comment fonctionne le code à barres :

Lors du passage d'une source lumineuse sur les barres sombres et claires d'un code à barres, l'intensité lumineuse varie. Celle-ci, captée par un récepteur photosensible, est amplifiée, filtrée et digitalisée pour être convertie en information numérique qu'un décodeur retranscrit en caractères ASCII directement exploitable par un système informatique par le biais d'une liaison hertzienne ou filaire [2].

- Dans les magasins :

Lors du passage en caisse, vous constatez que la lecture des codes à barres s'opère de façon rapide et fiable.

Un dispositif de lecture scanner, douchette ou pistolet laser est relié à un fichier informatique, lui-même relié aux caisses [1]. La lecture du code à barres déclenche l'accès à ce fichier, qui aura été préalablement renseigné par le commerçant : le code de l'article, son nom, son prix et la quantité disponible en stock.

Ainsi, en une fraction de seconde, et sans frappe au clavier, le code à barres est décodé, le nom de l'article et son prix sont imprimés sur le ticket de caisse. Le fonctionnement décrit ici s'adapte à toutes les variétés de magasins, quelle que soit leur taille.

Hypermarchés, supermarchés, grands magasins, commerces de proximité ou librairies, surfaces spécialisées de bricolage, de textile, etc. Tous les commerçants ont à leur disposition les équipements adaptés à leurs besoins.

1.7. Choix de la taille du code à barres :

La taille d'un code à barres varie dans certaines limites selon les conditions d'impression : technique d'impression, qualité du support, caractéristiques de l'imprimante...

C'est donc l'imprimeur qui indiquera la taille minimum que le responsable du graphisme de l'emballage doit respecter ce qui est représenté dans la Figure 1.1 et la

Figure 1.2. Les marges claires à droite et à gauche du code à barres doivent impérativement être respectées [1]. Elles sont indispensables pour la reconnaissance du code par le lecteur.

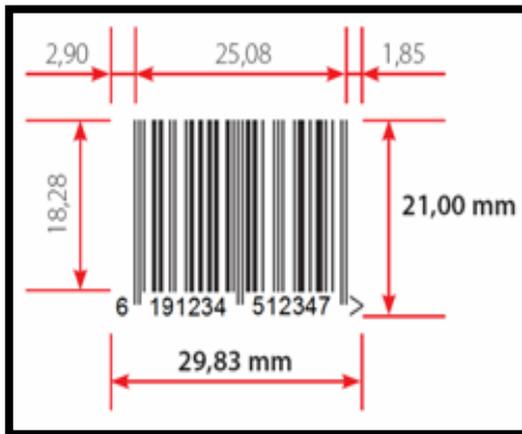


Figure 1.1 :

Facteur de grossissement 0,80 (80%)

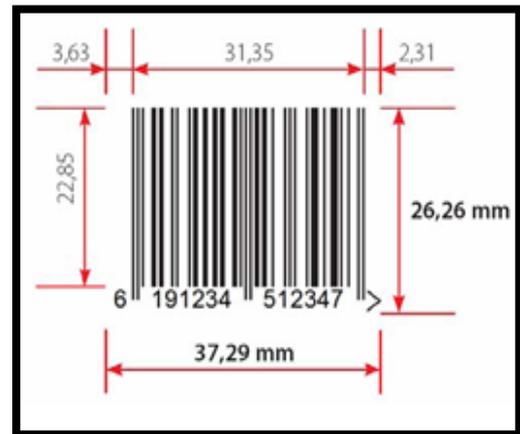


Figure 1.2 :

Facteur de grossissement 1,00(100%)

1.8. Positionnement du code à barres :

Il s'agit d'évaluer l'emplacement le plus approprié pour le code à barres dès la conception du packaging.

D'une façon générale, il est conseillé de mettre le code à barres dans la partie basse de la face arrière du produit (Figure 1.3) [1]. Dans tous les cas, il faut penser à faciliter le travail de la caissière qui aura à présenter le code à barres devant la fenêtre de lecture des caisses.

Pour les produits cylindriques (Figure 1.4), il faut tenir compte du rayon de courbure pour déterminer l'orientation du code à barres [1]. Ainsi en EAN 13 par exemple, sur un produit qui mesure moins de 5 cm. de diamètre, les barres du symbole, quel que soit le facteur de grossissement, devront obligatoirement être perpendiculaires à l'axe du cylindre.

Il faut éviter que le code à barres soit placé sous une soudure, une pliure de film d'emballage souple ou masqué par le produit lui-même quand il est mis sous blister.

Lorsque le produit est de taille insuffisante pour recevoir un code GTIN 13, il est possible d'obtenir auprès de votre organisation locale un code plus petit à 8 chiffres.

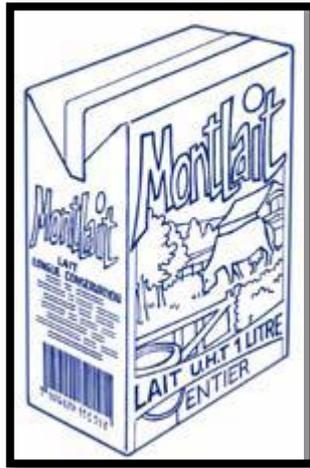


Figure 1.3 : cas général



Figure 1.4 : cas cylindrique

1.9. Choix des couleurs :

Un contraste de couleur suffisant entre les barres foncées et les espaces clairs doit être assuré (des barres noires sur un fond blanc constituent une combinaison optimale pour une bonne lecture) [1]. Mais il faut savoir que le lecteur optique n'apprécie pas les contrastes tout à fait comme l'œil humain.

D'une manière générale :

- Les codes à barres exigent des couleurs foncées (par exemple, noir, bleu, marron foncé, ou vert foncé).
- Les symboles de code à barres exigent des fonds clairs pour les espaces et les marges (le blanc exemple).

- Le rouge et ses dérivés (rose, orange, jaune) sont des couleurs claires utilisables pour le fond, non pour les barres.

- Les matières métalliques brillantes et les encres réfléchissantes (dorées, argentées) sont généralement perçues comme des teintes foncées.

- Dans la plupart des cas, le fond du symbole n'est pas imprimé. Ce sont les barres qui sont imprimées. Si le fond lui-même venait à être imprimé sous les barres, il devrait l'être comme une couleur à part entière.

- Si on emploie des couches multiples d'encre pour augmenter l'opacité du fond, chaque couche doit être une couleur à part entière.

En utilisant des barres noires sur fond blanc, on choisit la combinaison optimale. Cependant d'autres combinaisons sont utilisables.

Ce qui est montré dans les tableaux ci-dessous :

Ce qu'il est possible de faire :

noir sur blanc	
noir sur jaune	
noir sur orange	
noir sur rouge	
bleu sur blanc	
bleu sur jaune	
bleu sur orange	
vert sur blanc	
vert sur jaune	
vert sur rouge	
marron foncé sur blanc	

Ce qu'il ne faut pas faire :

jaune sur blanc	
orange sur blanc	
rouge sur blanc	
noir sur vert	
noir sur bleu	
marron clair sur blanc	
noir sur marron foncé	
noir sur violet	
violet sur noir	
rouge sur marron clair	
noir sur gris foncé	

Tableau 1.1 : table montre comment choisir les couleurs des codes-barres

1.10. Lecteurs de code à barres :

Différents types de matériels de lecture peuvent être utilisés aux caisses des magasins. La connaissance succincte de leur mode de fonctionnement permet d'éviter certaines erreurs en matière de positionnement des symboles sur les produits.

1.10.1. Crayon lecteur :

C'est le plus simple des appareils de lecture. Le crayon effectue une lecture par passage manuel sur le code à barres. Réalisée au contact, la qualité de la lecture varie selon l'inclinaison du crayon et dépend énormément de la constance avec laquelle le code est parcouru [2].

Son petit prix en fait un modèle adapté aux faibles besoins de lecture, éventuellement un outil de contrôle de l'information codée.

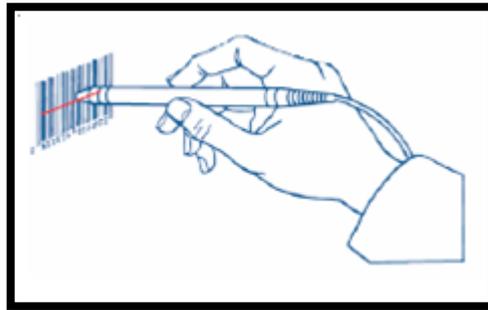


Figure 1.5 : crayon lecteur

1.10.2. Lecteur CCD :

La douchette CCD autorise une lecture automatique du code à barres, nul besoin de parcourir le code et chacune de ses barres. La distance de lecture est variable (jusqu'à 20 cm) et dépend des réglages apportés au lecteur [1].

Plus la distance est importante et plus l'éclairage du code est faible. Les modèles CCD ne comportent pas d'élément mécanique et présentent généralement une grande robustesse.

Plus économique que les modèles lasers, les lecteurs CCD sont capables de lire tous les codes à barres linéaires. Certains modèles sont capables de lire les codes empilés.

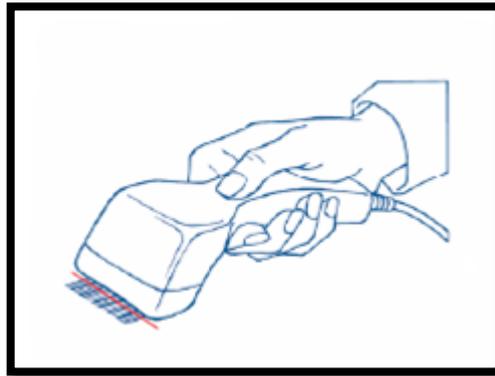


Figure 1.6 : lecteur CCD

1.10.3. Lecteur laser :

Construit autour d'un jeu de miroirs, le lecteur laser utilise un seul rayon lumineux généré par une diode laser [1]. La source lumineuse est dense et précise, et autorise une lecture rapprochée ou distante de plusieurs mètres ainsi qu'une lecture au vol, sur des objets ou documents en mouvement. Tout comme les lecteurs CCD, le lecteur laser réalise une lecture automatique du code. Nul besoin de parcourir le code dans sa longueur, un miroir motorisé le fait en réfléchissant le rayon laser de part et d'autre du code donnant ainsi l'illusion optique d'un trait continu. Certains lecteurs réalisent également ce balayage sur la hauteur du code (Raster en Anglais ou multi trames), d'autres encore démultiplient le balayage dans le but d'obtenir une grille sous laquelle le code peut être placé sous un degré d'orientation quelconque (omnidirectionnel).

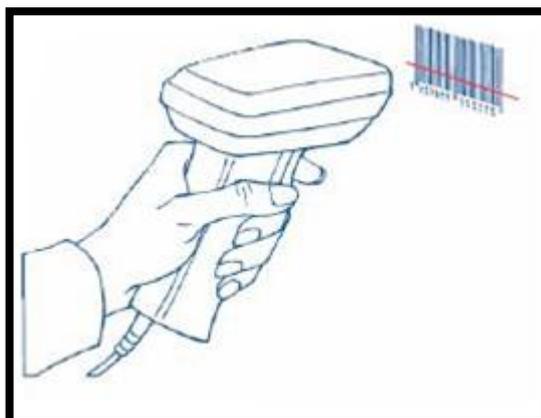


Figure 1.7 : Le lecteur laser

1.11. Principaux standards :

Les codes-barres utilisent divers protocoles de codification, qui diffèrent en fonction des contraintes d'utilisation ou de normalisation.

Voici, les trois types, de codes barres les plus utilisés :

- Les codes-barres unidimensionnels ou linéaires.
- Les codes barres linéaires empilés.
- Les codes à barres à deux dimensions.

1.11.1. Codes-barres unidimensionnels ou linéaires :

- EAN 8, EAN 13, UPC
- Code 11
- Code 39
- Code 93
- Code 128
- ITF ou “ Interleaved 2 of 5”
- CIP Pharmacode France
- Laetus Pharmacode

1.11.2. Codes-barres linéaires empilés :

Ainsi nommés car ils sont constitués de plusieurs codes-barres linéaires empilés les uns sur les autres. Ils sont lus verticalement avec un lecteur à balayage automatique.

- Code 16K
- Code 49
- PDF 147
- le Postnet

1.11.3. Codes à barres à deux dimensions :

Ainsi nommés car ils sont codés et lus horizontalement et verticalement. Ils permettent donc de coder un plus grand nombre de données sur une même surface [4].

- Code one

- Code Aztec
- Data Matrix
- QR Code

Le tableau suivant résume les applications ainsi que les particularités de quelques codes à barres [4] :

Les codes-barres	Particularités	Organismes de régulation Applications
<p>Interleaved 2of5 ITF ITF14</p> 	<ul style="list-style-type: none"> • Numérique (0-9) • Longueur variable <ul style="list-style-type: none"> • Check digit optionnel • Haute densité 	<p>IATA (International Air Transport Association) Étiquettes bagages ITF-14 or SCC-14 (Shipping Container Code) UCC HIBC Health Industry Barcode USPS special services OPC (Optical Product Code Council) Deutsche Post Identcode and Leitcode CIP (Pharma France)</p>
<p>EAN-8 JAN-8</p>  <p>1234 5670</p>	<ul style="list-style-type: none"> • Numérique (0-9) • Longueur fixe • Check digit 	<p>EAN (European Article Numbering System)</p>
<p>EAN-13 JAN-13</p>  <p>0 000123 456784</p>	<ul style="list-style-type: none"> • Numérique (0-9) • Longueur fixe • Check digit 	<p>EAN (European Article Numbering System) Bookland ISBN (International Standard Book Numbers) ISSN (International Standard Serial Numbering) Spanish Pharmacode JAN (Japan Article Number) Swissmedic Swiss Pharmacode</p>

<p>Code 39 USD-3 Code 3 of 9</p> 	<ul style="list-style-type: none"> • Numérique (0-9) • Majuscules (A-Z) • 7 car. spéciaux (space+.\$/%) <p>ETENDU:</p> <ul style="list-style-type: none"> • 128 caractères ASCII • Longueur variable • Check digit optionnel 	<p>CIP (Club Inter Pharmaceutique France) Pharmacode France ODETTE - European Auto Industry LOGMARS (Logistics Applications of Automated Marking and Reading Symbols) PZN (Pharma Zentral Nummer) Pharmacode Allemagne HIBC - Health Industry Barcode GSA - General Services Administration AIAG - Automotive Industry Action Group DoD MIL-STD-1189 LOGMARS DoD MIL-STD-129P</p>
<p>Code 128</p> 	<ul style="list-style-type: none"> • 128 caractères ASCII • Caractères de contrôle • Numérique (00-99) • Longueur variable • Check digits • Haute densité 	<p>HIBC (Health Industry Barcode) USPS Confirmation de livraison American Association of Blood Banks (ISBT 128) GALIA Auto Industry SISAC - (Serials Industry Systems Advisory Committee)</p>
<p>PDF417</p> 	<ul style="list-style-type: none"> • Modes multiples pour texte et données. • Longueur variable <ul style="list-style-type: none"> • Capacité: <ul style="list-style-type: none"> • 2710 digits • 1850 characters • 1108 bytes 	<p>AAMVA (American Assoc of Motor Vehicle Administrators) drivers' license cards</p>
<p>DataMatrix</p> 	<ul style="list-style-type: none"> • Modes multiples pour texte et données. • Longueur variable <ul style="list-style-type: none"> • Capacité: <ul style="list-style-type: none"> • 3116 digits • 2335 characters • 1556 bytes • Très haute densité 	<p>Swiss Post WebStamp Deutsche Post Stampit Industrie pharmaceutique pour l'emballage de doses unitaires AIAG (Automotive Industry) NASA EIA (Electronic Industry Alliance)</p>

Tableau 1.2 : table résume quelques types des codes-barres et leurs applications ainsi que les particularités

1.12. Normalisation EAN 13 :

L'EAN ou European Article Numbering [5] est une norme garantissant que le code-barres d'un article sera reconnu dans tous les pays de l'Union Européenne. L'EAN assure aussi une compatibilité avec les codes U.P.C. utilisés en Amérique du Nord. Le mot codé est constitué de 13 caractères répartis entre trois zones [5]. En lisant de la gauche vers la droite, on trouve :



Figure 1.8 : Un code à barres à la norme EAN 13

- **le préfixe** : qui représente le code du pays ayant délivré le numéro du participant (2 ou 3 chiffres) [4].

-**le numéro du participant** : qui est délivré par l'organisation EAN du pays concerné (les 4 ou 5 chiffres suivants) [6].

-**le numéro d'article du producteur de l'objet** : étiqueté sur 5 chiffres.

-**le chiffre de contrôle ou "Check Digit"** : qui est calculé selon les 12 premiers chiffres qui composent le code.

Donc si on prend le code-barres de la Figure 1.8 on trouve que :

- Préfixe 76,
- Numéro de Participant 12345,
- Numéro d'Article 67890,
- Check Digit 0.

1.12.1. Calcul du caractère de contrôle :

Voici un exemple de calcul du caractère de contrôle d'un code EAN 13.
Soit le code à barres 4 007630 00011X (où X désigne le chiffre de contrôle), on peut calculer ce caractère de contrôle en se basant sur la table suivante :

Code EAN	4 007630 00011X
Chiffres	4 0 0 7 6 3 0 0 0 0 1 1
Multiplieur par	1 3 1 3 1 3 1 3 1 3 1 3
Résultat	4 0 0 21 6 9 0 0 0 0 1 3
Somme	4 + 0 + 0 + 21 + 6 + 9 + 0 + 0 + 0 + 0 + 1 + 3 = 44
Division par 10	44 / 10 = 4 reste 4
Caractère de contrôle	10 - reste = 6

Le code EAN final est donc dans ce cas 4 007630 000116.

1.12.2. Système de codage :

- Le premier chiffre n'est pas codé mais il détermine les tables à utiliser pour le codage des 6 chiffres suivants.
- La largeur d'une barre fine (0,33 mm en standard) est appelée le module.
- Il existe des barres de 1, 2, 3 ou 4 modules ; de même pour les espaces.
- Chaque chiffre est codé sur 7 modules comprenant chacun 2 barres et deux espaces.

Symbolisons le module de barre noir par le "1" binaire et le module de barre blanche (espace) par le "0" binaire.

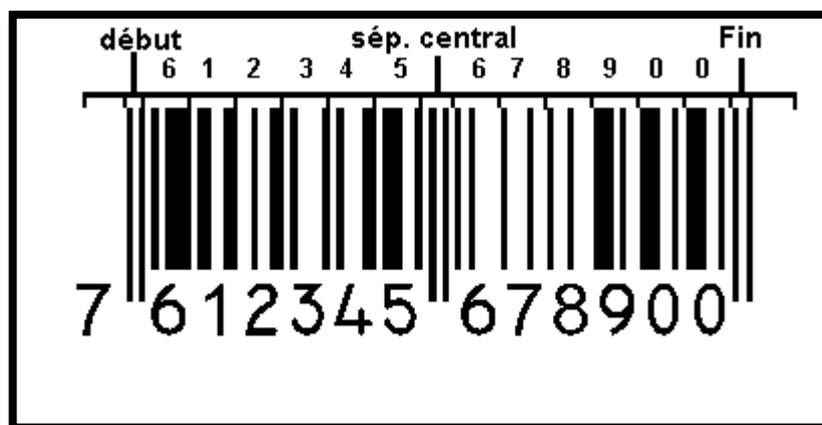


Figure 1.9 : composition de la norme EAN 13

- La marque de début de même que la marque de fin est : "101"
- La marque de séparation placée au milieu après le 7ème chiffre est : "01010"
- La longueur totale est toujours de $12 * 7 + 2 * 3 + 5 = 95$ modules ; il y a toujours 30 barres au total.
- Les barres des séparateurs sont plus longues de 5 modules que les autres barres.
- Il faut des marges blanches de 11 modules à gauche et de 7 modules à droite au moins.

- on notera que les séparateurs ont des barres plus hautes que les barres qu'ils encadrent

Trois jeux différents sont utilisés selon la position du chiffre dans le code. Les jeux A et B sont utilisés uniquement à gauche du milieu du code barres [6], sans ordre particulier, tandis que le jeu C est utilisé uniquement à droite de ce même milieu. Ce que montre la table suivante :

Caractère codé	Jeu A	Jeu B	Jeu C
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

Tableau 1.3 : Codage du jeu (les codes des jeux A et B commencent par un espace, alors que ceux de la jeu C commencent par une barre) .

Observations :

- Tous les caractères de la partie gauche du code EAN commencent toujours par un 0 (espace) alors que ceux de la partie droite commencent toujours par un 1 (barre).
- Le codage des Set A et C est similaire, les 0 du Set A devenant des 1 en Set C.
- Le codage du Set B est basé sur celui du Set A.
Pour obtenir le Set B à partir du Set A faire ce qui suit :
1) Changer tous les 1 en 0 et tous les 0 en 1 du Set A.
2) Lire le résultat de droite à gauche. On obtient le Set B.

Le premier chiffre n'est pas codé, chacun des 6 chiffres suivants est converti soit par la table A soit par la table B, et cela en fonction du premier chiffre [5]. Les 6 derniers chiffres sont convertis par la table C.

Une nouvelle table (table de parité) indique l'utilisation des tables A ou B en fonction du chiffre N°1 :

Selon chiffre 1	chiffre 2	chiffre 3	chiffre 4	chiffre 5	chiffre 6	chiffre 7
0	A	A	A	A	A	A
1	A	A	B	A	B	B
2	A	A	B	B	A	B
3	A	A	B	B	B	A
4	A	B	A	A	B	B
5	A	B	B	A	A	B
6	A	B	B	B	A	A
7	A	B	A	B	A	B
8	A	B	A	B	B	A
9	A	B	B	A	B	A

Tableau 1.4 : indique l'utilisation des tables A ou B en fonction du chiffre N°1

Observations :

- Le second caractère du Préfixe est toujours codé en Set A.
- Tout EAN-13 (ne commençant pas par zéro) comprend toujours 3 caractères en Set A et 3 caractères en Set B.

Alors si on reprend l'exemple de code à barres précédent 76 12345 67890 0

- Préfixe 76,
- Numéro de Participant 12345,
- Numéro d'Article 67890,
- Check Digit 0.

Le premier caractère du préfixe est 7.

En consultant la table de parité pour la valeur 7, on trouve que le second caractère du Préfixe et le Numéro de Participant utilisent les jeux :

A B A B A B

Ce qui signifie que :

- le second caractère du Préfixe sera codé en utilisant le Set A,
- le premier caractère du Numéro de Participant sera codé en utilisant le Set B, etc.

Nous pouvons maintenant composer notre code pas à pas :

Nom du caractère	caractère	Jeu	Valeur binaire
Caractère Début	-	-	101
2ème car du Préfixe	6	A	0101111
1er car. N° Participant	1	B	0110011
2ème car. N° Participant	2	A	0010011
3ème car. N° Participant	3	B	0100001
4ème car. N° Participant	4	A	0100011
5ème car. N° Participant	5	B	0111001
Séparateur Central	-	-	01010
1er car. N° Article	6	C	1010000
2ème car. N° Article	7	C	1000100
3ème car. N° Article	8	C	1001000
4ème car. N° Article	9	C	1110100
5ème car. N° Article	0	C	1110010
Check Digit	0	C	1110010
Caractère Fin	-	-	101

Tableau 1.5 : Codage du jeu de caractères en binaire.

Le Code EAN est ainsi composé en mettant simplement bout à bout les chaînes de "0" et de «1 ". ("1" représente une barre et "0" un espace).

Soit :

1010101111011001100100110100001010001101110010101010100001000100100100011101001110010111001010

1

Et graphiquement :



Et donc si nous convertissons cette chaîne de 1 et 0 selon leur représentation graphique, nous obtenons le code EAN de la Figure 1.8.

1.13. Conclusion :

De nos jours, les codes barres sont des acteurs discrets de la vie courante. Ils trouvent leurs applications dans des domaines très variés, ils permettent d'assurer la diversité de l'offre des produits. Depuis l'utilisation systématique des codes à barres optimise la gestion des stocks, rationalise les flux d'informations dans la chaîne d'approvisionnement, simplifie les inventaires et garantit la traçabilité des produits. Dans ce chapitre, on a présenté les différentes propriétés ainsi les méthodes de lecture des codes à barres pour pouvoir les traiter, sans doute, les codes à barres rendent la gestion, dans des domaines divers, plus simple, plus rapide et plus efficace.

Chapitre 2 :

Traitement

d'images.

2.1. Introduction :

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

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.

De ce fait, le traitement d'images est l'ensemble des méthodes et techniques opérant sur celles-ci, dans le but de rendre cette opération possible, plus simple, plus efficace et plus agréable, d'améliorer l'aspect visuel de l'image et d'en extraire des informations jugées pertinentes.

Dans ce chapitre, nous abordons les notions de base nécessaires à la compréhension des techniques de traitement d'images. Ensuite, nous allons nous intéresser sur les opérateurs locaux (filtrage, morphologie mathématique), que nous avons utilisés dans notre projet de fin d'études [8].

2.2. Définition de l'image :

Une image est plutôt difficile à décrire d'une façon générale. Une image est une représentation d'une scène. En traitement d'images, la majorité du temps, on considère qu'il s'agit d'une fonction mathématique de $R \times R$ dans R où le couplet d'entrée est considéré comme une position spatiale, le singleton de sortie comme l'intensité (couleur ou niveaux de gris) du phénomène physique. Il arrive cependant que l'image soit dite "3D" donc la fonction est de $R \times R \times R$ dans R . Les images couleurs peuvent être représentées soit par trois images représentant les trois couleurs fondamentales, soit par une image de $R \times R$ dans $R \times R \times R$ [8].

2.3. Image numérique (numérisée) :

L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs.

La numérisation d'une image est la conversion de celle-ci de son état analogique en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques $f(x,y)$, comme la montre la figure 2.1 où :

x,y : coordonnées cartésiennes d'un point de l'image.

$f(x, y)$: niveau d'intensité. La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur [9].

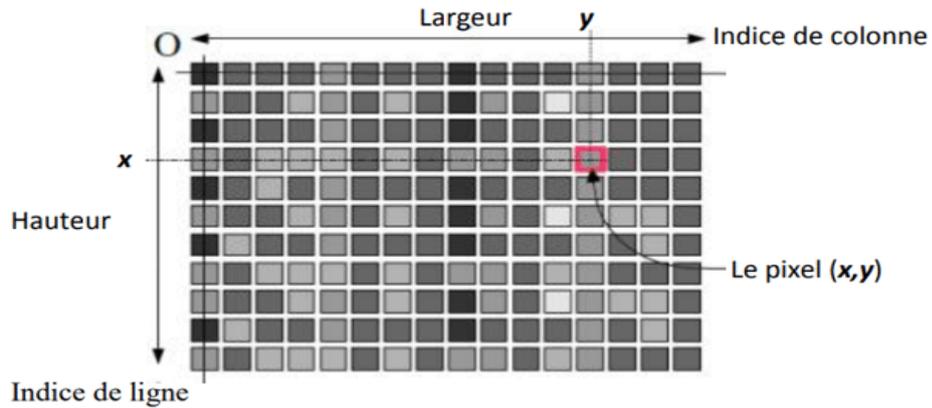


Figure 2.1: Représentation d'une image numérique.

2.4. 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 :

- Caméras numériques,
- Scanners.

A ce niveau, notons que le principe utilisé par le scanner est de plus en plus adapté aux domaines professionnels utilisant le traitement de l'image comme la télédétection, les arts graphiques, la médecine, etc. Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur C.C.D. (Charge Coupled Device). La carte d'acquisition reçoit les images de la camera, de la TV ou du scanner afin de les convertir en informations binaires qui seront stockées dans un fichier [8] .

L'une des caractéristiques intéressantes de ces capteurs est la taille du plus petit élément (pixel), mais aussi l'intercorrélacion de deux éléments voisins : plus cette intercorrélacion est faible, meilleure est l'image [8].

2.5. Caractéristiques d'une image numérique :

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

2.5.1. Pixel :

Contraction de l'expression anglaise " Picture Elements ": éléments d'image, le pixel est le plus petit point de l'image, c'est une valeur numérique représentative des intensités lumineuses.

Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peut manipuler les matériels et logiciels sur l'image. La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure ci-dessous [9].

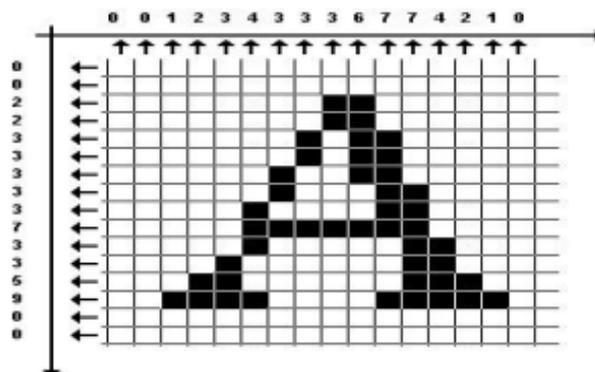


Figure 2.2 : Groupe de pixel formant la lettre A.

2.5.2. Dimension :

La dimension représente 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 [9].

2.5.3. Texture :

Une texture est une région dans une image numérique qui a des caractéristiques homogènes. Ces caractéristiques sont par exemple un motif basique qui se répète. La texture est composée de Texel, l'équivalent des pixels [9].

2.5.4. Résolution :

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot

résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur; plus grand est ce nombre, meilleure est la résolution [8].

2.5.5. 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 [9].



Figure 2.3 : (a) Image sans bruit , (b) Image avec bruit.

2.5.6. 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 [9].

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 (voir figure 2.4) [8].

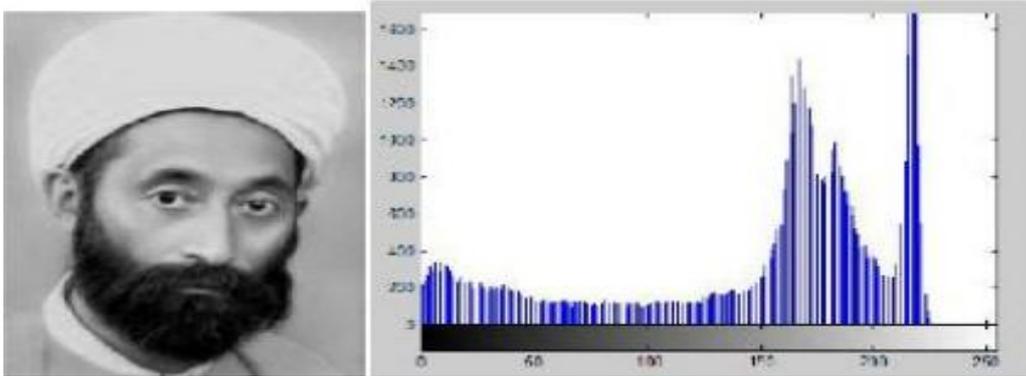


Figure 2.4 : Image avec son histogramme.

2.5.7. Luminance :

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet . Une bonne luminance se caractérise par :

- Des images lumineuses (brillantes).
- Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- L'absence de parasites [9] .

2.5.8. Contraste :

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images. Si L1 et L2 sont les degrés de luminosité respectivement de deux zones voisines A1 et A2 d'une image, le contraste C est défini par le rapport [9] :

$$C = \frac{L1 - L2}{L1 + L2}$$

2.6. Types d'images :

On distingue trois types d'images :

- Binaire : 2 couleurs (arrière-plan et avant-plan).
- Monochrome : variations d'une même teinte.
- Polychrome : " vraies " couleurs [9].

2.6.1. Images binaires (en noir et blanc) :

Les images binaires sont les plus simples. Bichromes (dont les pixels ne peuvent avoir que les valeurs 0 et 1). Le 0 correspond à un pixel noir et 1 à un pixel blanc. Le niveau de gris est donc codé sur un seul bit [9].

Une image binaire peut être stockée dans la mémoire sous forme de bitmap, un ensemble de bits compressés. Une image de 640×480 nécessite 37,5 kB de stockage. En raison de la petite taille des fichiers image, les solutions de télécopie et de gestion de documents utilisent généralement ce format. La plupart des images binaires se compressent bien avec des schémas de compression simples. Le domaine du traitement de l'image morphologique a été largement inspiré par ce point de vue [9].



Figure 2.5 : Image binaire.

2.6.2. Images à niveaux de gris

Le niveau de gris est la valeur de l'intensité lumineuse en un point. La couleur du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter les images à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur peut être comprise par exemple entre 0 et 255. Chaque pixel n'est donc plus représenté par 1 bit, mais par 1 octet (8bits). Pour cela, il faut que le matériel utilisé pour afficher l'image, soit capable de produire les différents niveaux de gris correspondant [9].

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 [9].



Figure 2.6 : Image monochrome (256 couleurs)

2.6.3. Images en couleurs :

Elle est obtenue par la combinaison de trois couleurs dites primaires : rouge, vert et bleu (RVB). Chaque couleur est codée comme une image à niveaux de gris, avec des valeurs allant de 0 à 255. Pour $R=V=B=0$ nous aurons un noir pur, et pour $R=V=B=255$ nous aurons un blanc pur. La représentation des images couleurs se fait donc soit par une image dont la valeur du pixel est une combinaison linéaire des valeurs des trois composantes couleurs, soit par trois images distinctes représentant chacune une composante couleur, on distingue généralement deux types d'images:

- Images 24 bits.
- Images à palettes ou indexée [9].



Figure 2.7: Image polychrome (65536 couleurs)

2.7. Passage d'une image couleur vers une image à niveaux de gris :

on doit exploiter une des deux équations du codage YUV ou YIQ qui sont identiques et qui est :

$$G = 0,299 * R + 0,587 * V + 0,114 * B$$

Avec G est le niveau de gris du pixel,

et R,V et B sont les composantes rouge, vert et bleue du pixel.

Pour afficher une image à niveaux de gris sur un écran couleur, les trois composantes RVB doivent être identiques comme suit :

$$R = V = B = G.$$

L'application de cette conversion est illustrée par les images suivantes :

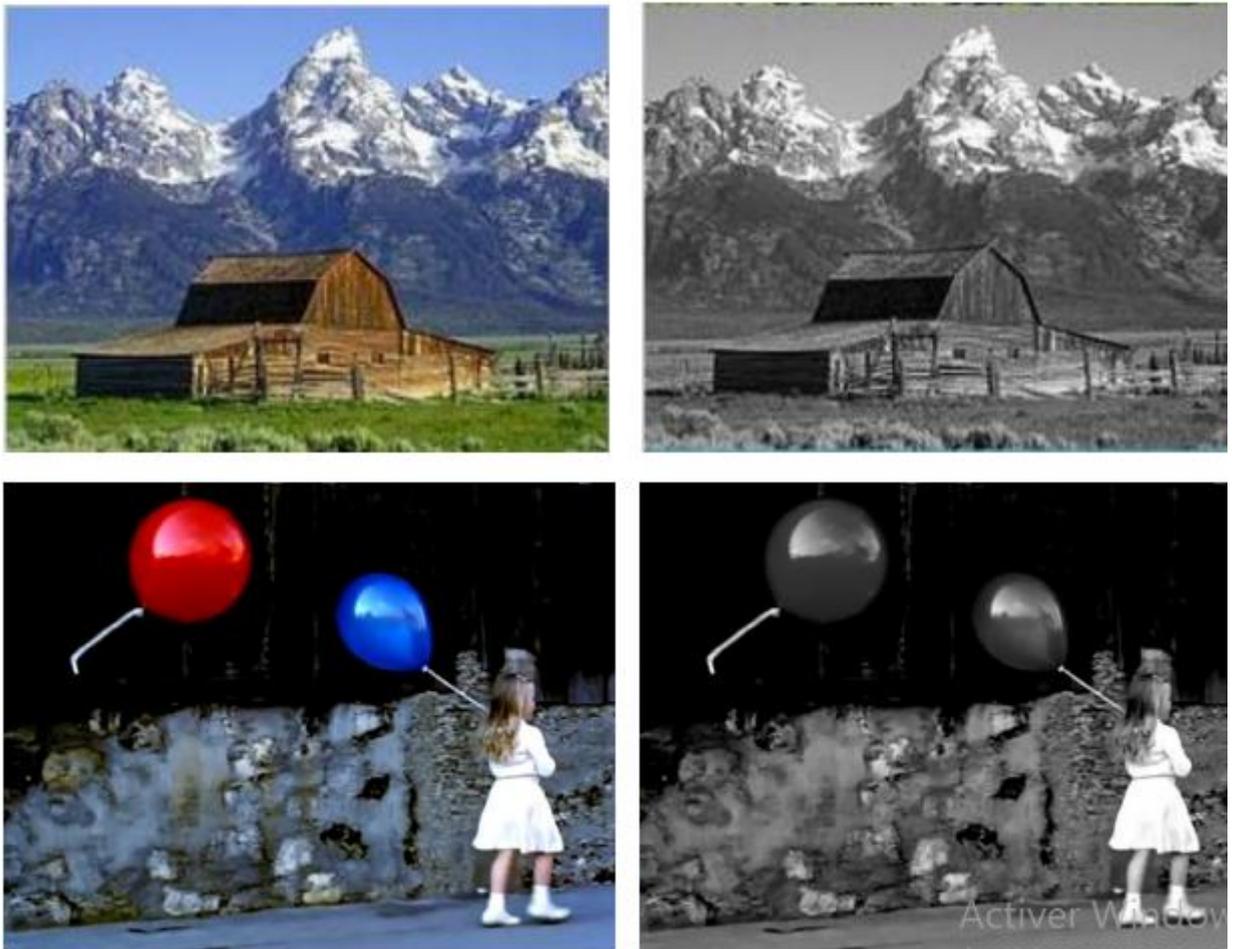


Figure 2.8 : Conversion des images en couleur vers des images à niveaux de gris

2.8. Prétraitement d'images :

Cette phase a lieu juste après l'acquisition des images et a pour objectif d'améliorer la qualité de l'image en vue de sa segmentation. Le temps de traitement (temps CPU) est très important. C'est un facteur décisif, il doit être le plus petit possible. Ceci implique que les opérateurs doivent être locaux, c'est-à-dire qu'ils doivent intervenir sur un nombre limité de pixels et essentiellement sur des pixels dans le voisinage proche du pixel courant .

Les techniques de pré-traitements les plus courantes qu'on va présenter est

- La réduction du bruit par filtrage [8].

2.8.1. Réduction du bruit par filtrage :

Le bruit peut provenir de différentes causes :

- Environnement lors de l'acquisition.
- Qualité du capteur,
- Qualité de l'échantillonnage.

En général, le bruit d'images est considéré comme un champ aléatoire caractérisé par sa densité de probabilité f et sa fonction de répartition F . La nature plus ou moins impulsionnelle du bruit peut être décrite à l'aide de densité de probabilités. Plusieurs filtres sont utilisés pour la réduction de bruit, ils sont divisés en deux catégories [8] :

2.8.2. Filtres linéaires stationnaires :

Les filtres linéaires sont une classe particulière de filtres, dont les propriétés mathématiques les rendent intéressants à utiliser. Dans le cas d'un filtrage linéaire de l'image, les opérations impliquées ne sont que des additions (ou soustractions) et multiplications (ou divisions) par des valeurs constantes. En pratique, cela consiste à réaliser une moyenne pondérée de la valeur des pixels du voisinage. Par exemple, une opération courante consiste à remplacer la valeur de chaque pixel par la valeur moyenne de ses voisins. La nouvelle valeur du pixel $I'(u,v)$ est déterminée par l'équation suivante [16]:

$$I'(u,v) = [I(u-1, v-1) + I(u, v-1) + I(u+1, v-1) + \\ I(u-1, v) + I(u, v) + I(u+1, v) + \\ I(u-1, v+1) + I(u, v+1) + I(u+1, v+1)] / 9$$

- **Filtre moyenneur :**

Les filtres moyenneurs, comme leur nom l'indique, calculent la moyenne, éventuellement pondérée, des pixels situés dans le voisinage de chaque pixel. Cette famille de filtres permet de réduire le bruit dans l'image, ce qui rend les zones homogènes plus lisses. Cependant, les contours sont fortement dégradés, et les structures trop fines peuvent devenir moins visibles [16]. (voir figure 2.9)



Figure 2.9 : Exemple de filtre moyenneur

- **Filtre gaussien :**

Le lissage par un filtre gaussien est justifié puisque l'on ne connaît rien a priori sur les interactions entre pixels de l'image. On fait donc l'hypothèse d'une loi d'interaction moyenne, c'est à dire gaussienne [17]. Une justification mathématique de l'emploi d'un lissage par une gaussienne a été formulée par Canny , qui montre que la dérivée première d'un filtre gaussien fournit une valeur proche de la valeur optimale pour le critère par lequel il se propose de définir un bon détecteur de contours (voir figure 2.10).



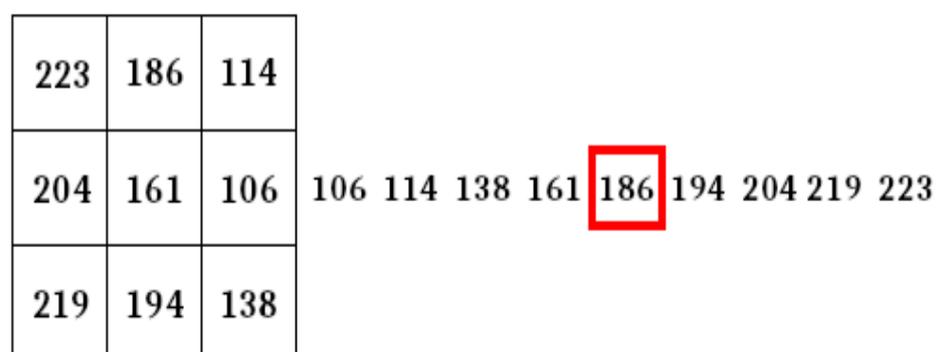
Figure 2.10: Exemple de filtre gaussien

2.8.3. Filtres non linéaires stationnaires :

Ces opérateurs ont été développés pour pallier aux insuffisances des filtres linéaires : principalement la mauvaise conservation des contours. Ils ont le défaut d'infliger des déformations irréversibles à l'image. Nous ne développerons pas ici de théorie des filtres non-linéaires : ils sont chacun basés sur des bases mathématiques ou empiriques différentes [34].

- **Filtre médian :**

Le filtre médian est un filtre numérique non linéaire, souvent utilisé pour la réduction de bruit. La réduction de bruit est une étape de prétraitement classique visant à améliorer les résultats de traitements futurs (détection de bords par exemple). La technique de filtre médian est largement utilisée en traitement d'images numériques car il permet sous certaines conditions de réduire le bruit tout en conservant les contours de l'image . Le niveau de gris du pixel central est remplacé par la valeur médiane de tous les pixels de la fenêtre d'analyse centrée sur le pixel. La taille du kernel dépend de la fréquence du bruit et de la taille des détails significatifs de l'image traitée [18]. Le filtrage médian passe par deux étapes ; la première consiste à mettre les intensités des pixels voisins en ordre croissant puis en deuxième étape prendre la valeur médiane qui sera considéré comme résultat du filtrage comme l'indique la figure 2.11 . Le résultat de l'application de ce filtre est illustré par la figure 2.12 .



Un environnement 3x3. Valeurs par ordre ascendant. Médiane entourée de rouge.

Figure 2.11 : Application du filtre médian

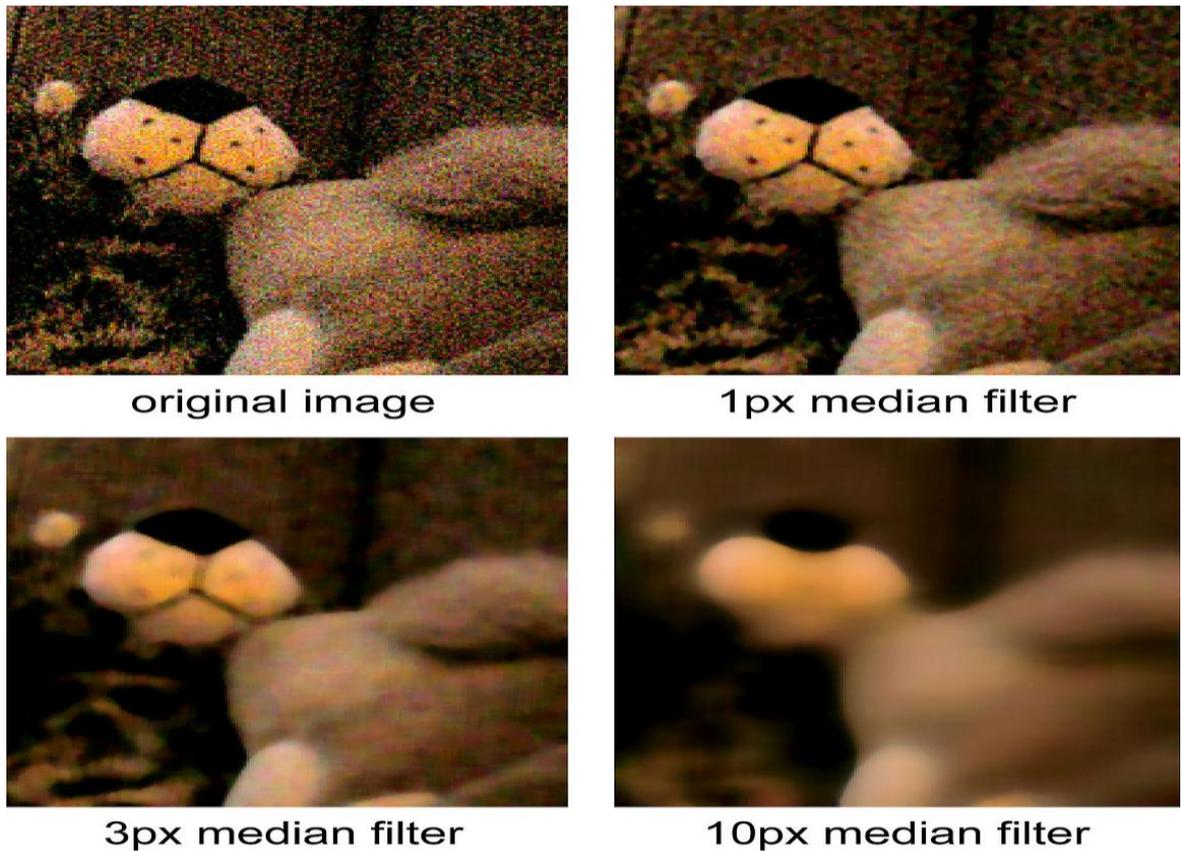


Figure 2.12 : Résultat de l'application du filtre médian

2.9. Morphologie mathématique (Filtres morphologiques) :

2.9.1. Filtre morphologique:

La morphologie mathématique est dédiée à l'analyse des structures spatiales. Elle est dite morphologique parce qu'elle vise à analyser la forme (morphologie) des objets. Une image est vue comme un ensemble de formes (voir figure 2.13).

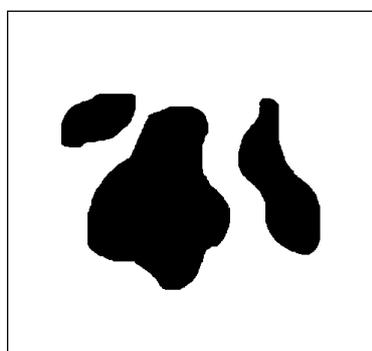


Figure 2.13 : Image formée d'objets

Les filtres morphologiques sont souvent utilisés pour éliminer des pixels isolés considérés comme un bruit dans une image. Pour analyser les images, ces méthodes utilisent des formes connues appelées éléments structurants (en anglais ; structuring elements) comme l'indique la figure 2.14 .

Un élément structurant est un masque binaire muni d'un point d'ancrage (point de fixation) comme l'indique la figure 2.14 .

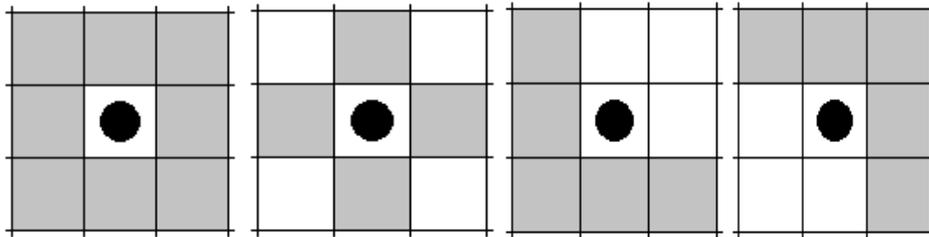


Figure 2.14 : Eléments structurants.

L'analyse d'images par la morphologie mathématique nécessite l'utilisation d'opérations morphologiques, Parmi ces opérations il y a la dilatation, l'érosion, l'ouverture et la fermeture mathématiques.

- **Dilatation :**

La dilatation a pour objectif de dilater les objets ou formes dans une image. Pour une image binaire X et un élément structurant B , nous avons :

$$D^B(x) = X \oplus B = \{x \mid B_x \cap X \neq \Phi\}$$

où $B_x = \{b + x \mid b \in B\}$

B_x : l'ensemble de B traduit de x .

Dans une image binaire cette opération permet d'éliminer les pixels noirs isolés mais ajoute des pixels blancs au contour des objets présents dans l'image. Le résultat de cette opération est l'augmentation de la taille de ces objets.

- **Erosion :**

L'érosion a pour objectif de d'éroder les objets ou formes dans une image. Pour une image binaire X et un élément structurant S , nous avons :

$$E^B(x) = X \oplus B = \{x \mid Bx \subseteq X\}$$

Elle permet d'éliminer les pixels blancs isolés au milieu des parties noires de l'image. Le résultat de cette opération est la diminution de la taille des objets présents dans l'image.

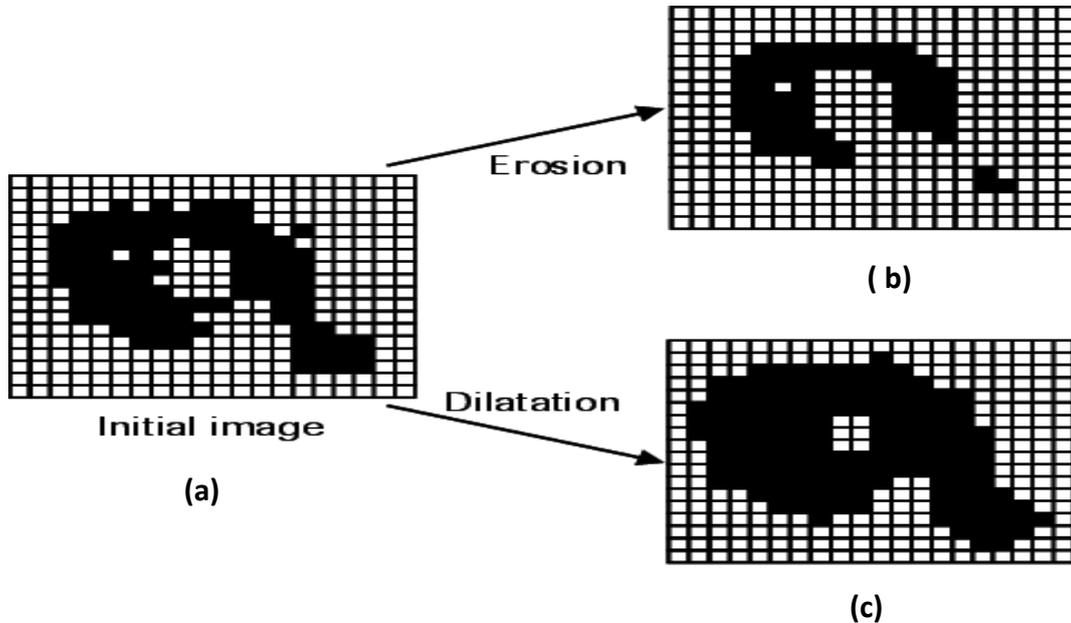


Figure 2.15 : (a) image initial , (b) :image après l'opération érosion , (c) : image après l'opération dilatation .

- **Ouverture :**

L'ouverture est constituée par une opération d'érosion suivie d'une dilatation. Elle permet d'éliminer les taches blanches dans le fond de l'image.

- **Fermeture :**

La fermeture est l'opération inverse de l'ouverture, qui consiste à faire subir à l'image une dilatation suivie d'une érosion. Elle permet d'éliminer les noirs qui se trouvent dans l'objet.

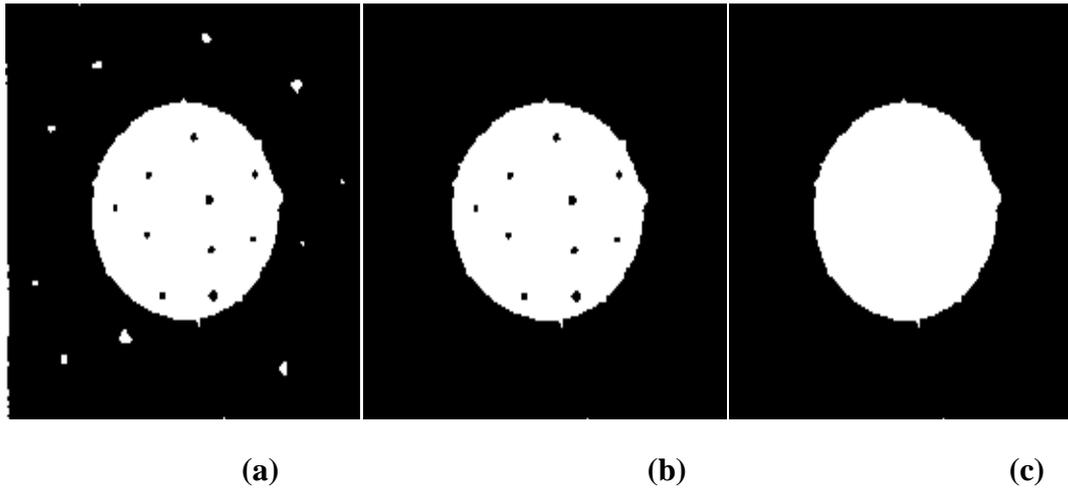


Figure 2.16: Analyse d'images par morphologie (a) Image d'entrée binaire. (b) Résultat d'ouverture avec un disque de 9x9. (c) Résultat de fermeture de l'image (b) avec un disque de 9x9.

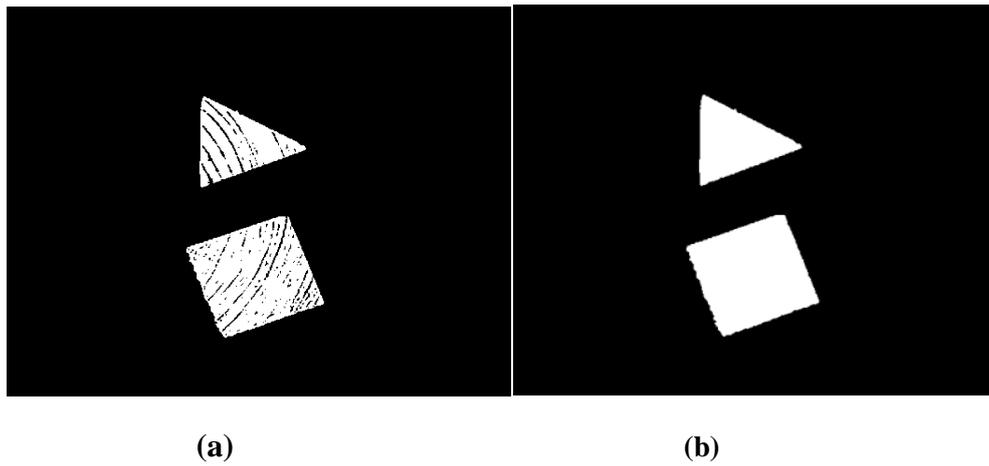


Figure 2.17 : Prétraitement de l'image acquise par caméra ; (a) Image binarisée. (d) Résultat de fermeture avec un élément structurant disque de taille 7x7.

➤ **Comparaison entre les opérations morphologiques :**

Erosion	Dilatation
• Augmente la taille des objets	• Diminue la taille des objets
• Bouche les petits trous	• Élargit les trous
• Soude les objets proches	• Sépare les objets connectés par un petit pont

Tableau 1.1: comparaison entre érosion et dilatation

Ouverture	Fermeture
• Élimine les petits objets	• Bouche les petits trous
• Lisse les contours	• Soude les formes proches

Tableau 2.2: comparaison entre ouverture et fermeture

2.9.2. Application du filtre morphologique sur des images de barres :

La figure 2.18 montre l'application d'ouverture morphologique à une image constituée d'une seule barre.

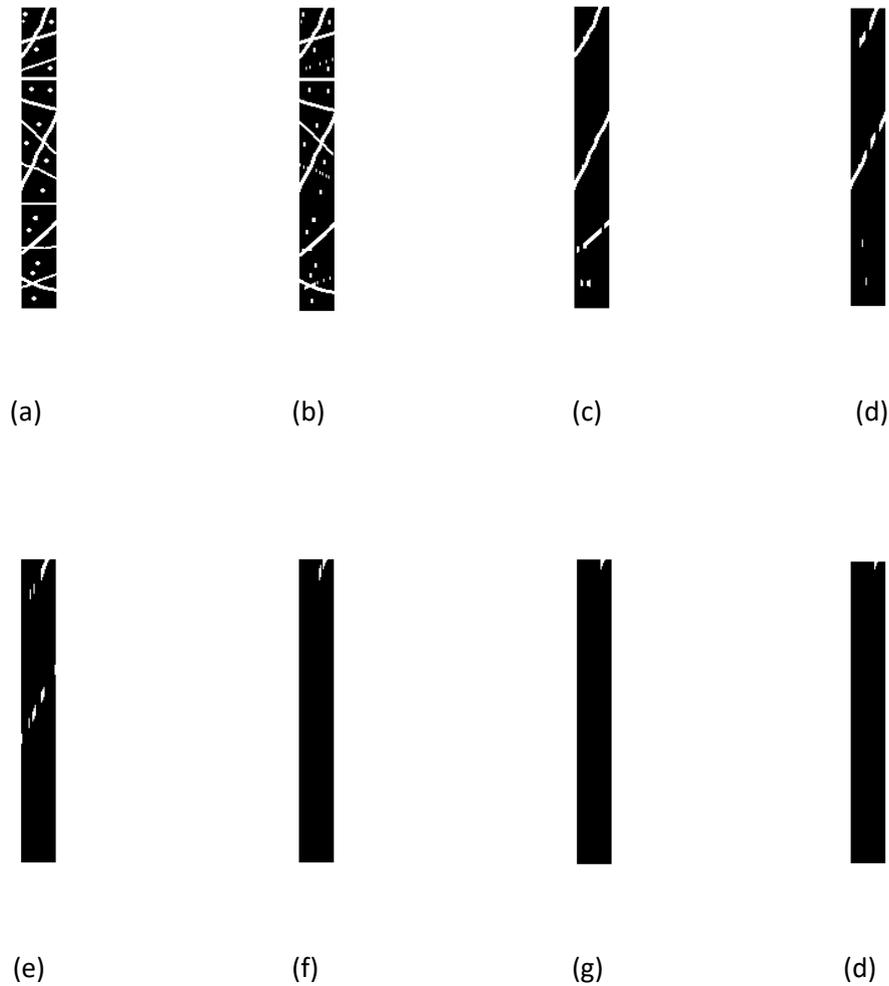


Figure 2.18: Résultats de l'application d'ouverture morphologique sur l'image (a) pour des éléments structurants ; (b) 3x1 (c) 5x1 (d) 7x1 (e) 9x1 (f) 11x1 (g) 13x1 (h) 31x1.

Il est clair de voir d'après la figure 2.18 que la qualité de l'image ayant subi une ouverture dépend essentiellement de la taille de l'élément structurant en question. Plus l'élément est long est plus le résultat est meilleur. Ladite qualité est au détriment du nombre de calcul qui augmente avec la taille du filtre.

2.10. Quelques exemples concrets du traitement d'images :

- Contrôle de présence/absence. Sur des chaînes de production, on vérifie en bout de chaîne avec une caméra vidéo la présence d'une pièce dans un ensemble plus complexe. Pour cela fréquemment 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 ainsi qu'à la texture du fruit son degré de maturité et par conséquent la catégorie sous laquelle il sera emballé puis vendu.
- Construction et correction de cartes géographiques selon des images satellites ou des images aériennes. On recale selon des informations topographiques les images reçues, puis on les met sur la carte en correspondance avec les informations trouvées dans l'image : voies de communication, voies et plans d'eau, parcelles agricoles...etc.
- Surveillance et évaluation de la production agricole. Il est envisageable de déterminer le degré de maturation des cultures, la quantité d'eau indispensable pour l'irrigation, le rendement moyen...etc. 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...etc [11].

2.11. Domaine d'application :

Les exemples et domaines d'applications du traitement d'image sont très nombreux. Les deux principaux domaines, qui ont permis au traitement d'image de se développer sont :

2.11.1. Domaine militaire :

- Missiles en tous genres (autodirecteur (courte portée), de croisière (longue portée),... etc.
- Renseignement (télédétection à partir d'images satellite dont la précision peut aller aujourd'hui jusqu'à quelques centimètres, ou aériennes, la photo-interprétation).
- Simulateurs « réels » (de vol avion, de char,etc.) [12]

2.11.2. Domaine médical :

- Angiographie,
- Radiographie, échographie,
- Scanner, IRM, etc [12].

Et d'autres domaines comme :

- Photographie
- Astronomie
- Contrôle qualité
- Sécurité.....[11].

2.12. Conclusion :

Dans ce chapitre nous avons parlés et définit les méthodes qui nous semble les plus utilisées et courantes dans le processus de traitement et analyse d'images. Nous avons fait un rappel sur des techniques nécessaires pour notre projet, notamment, le filtrage linéaire, non linéaire et la morphologie mathématique. Ainsi avec toutes ces informations utiles nous pouvons aborder l'étape suivante de notre projet qui consiste à mettre en œuvre un système capable de restaurer ou corriger une image de code à barre abîmée ou dégradée afin de le rendre décodable.

Chapitre 3 :

Application

et

test .

3.1. Introduction :

Dans ce chapitre nous allons présenter les tests et les résultats du travail fourni. On va utiliser le logiciel Matlab pour présenter les méthodes de traitement d'images, dont nous avons parlé le chapitre précédent, nous allons étudier un cas réel dans lequel nous traitons des images de code à barre abîmées. Ces images ont été apportées des articles de la grande surface Famili Shop, nous avons passées les images par une application capture_p dont le but de voir si les images sont correctes ou incorrectes, dont le cas où elles sont incorrectes nous faisons un traitement à base de logiciel Matlab.

Nous allons voir les résultats de ce traitement dans la suite du chapitre.

3.2. Environnement du Travail :

Pour mener à bien notre application, nous avons utilisé les moyens suivants :

- Camera de téléphone :
 - Huawei Gr5 : Résolution 6Mpx 3264×1840(16:9)
 - Samsung j4 core : Résolution 4Mpx 2960×1440(18:5:9)
- Un ordinateur portable :
 - hp EliteBook i5 (voir figure 3.1)
 - Windows 10 Professionnel 64bits (10.0 , build 19042)
 - Processeur Intel (R) Core(TM) i5-4200U CPU @ 1.60GHZ (4CPU) ~2.3GHZ
 - Mémoire installée (RAM) 4,00Go
 - Résolution 640 X 480 pixels



Figure 3.1 : Ordinateur portable hp EliteBook 820.

3.3. Logiciel Matlab :

Le logiciel Matlab est un logiciel de manipulation de données numériques et de programmation dont le champ d'application est essentiellement les sciences appliquées. Son objectif, par rapport aux autres langages, est de simplifier au maximum la transcription en langage informatique d'un problème mathématique, en utilisant une écriture la plus proche possible du langage naturel scientifique.

MATLAB est une plateforme de calcul numérique et de programmation utilisée par des millions d'ingénieurs et de scientifiques pour analyser des données, développer des algorithmes et créer des modèles, il exécute les opérations demandées séquentiellement, avec possibilité de boucle, teste et saut.

3.4. Système proposé :

Notre système est basé principalement sur l'application des opérations du traitement d'image et le module de décodage capture_p fourni par notre encadreur.

3.4.1. Présentation de l'application capture_p :

Cette application a été réalisé par notre encadreur Dr A. Namane.

C'est un lecteur de code-barres servant à lire les informations stockées sous forme de code-barres, dont le but de décoder les codes a barre existants dans l'image acquise par camera est de voir s'ils sont correcte ou incorrecte et calculer et comme bien de fois code a barres est correctement décodés. Le schéma synoptique de la figure 3.2 représente l'application du logiciel Capture_p.



Figure 3.2 : Application de la procédure de décodage capture_p.

3.4.2. Etapes à suivre :

Notre travail est basé sur les différentes étapes représentées par la figure. 3.3 :

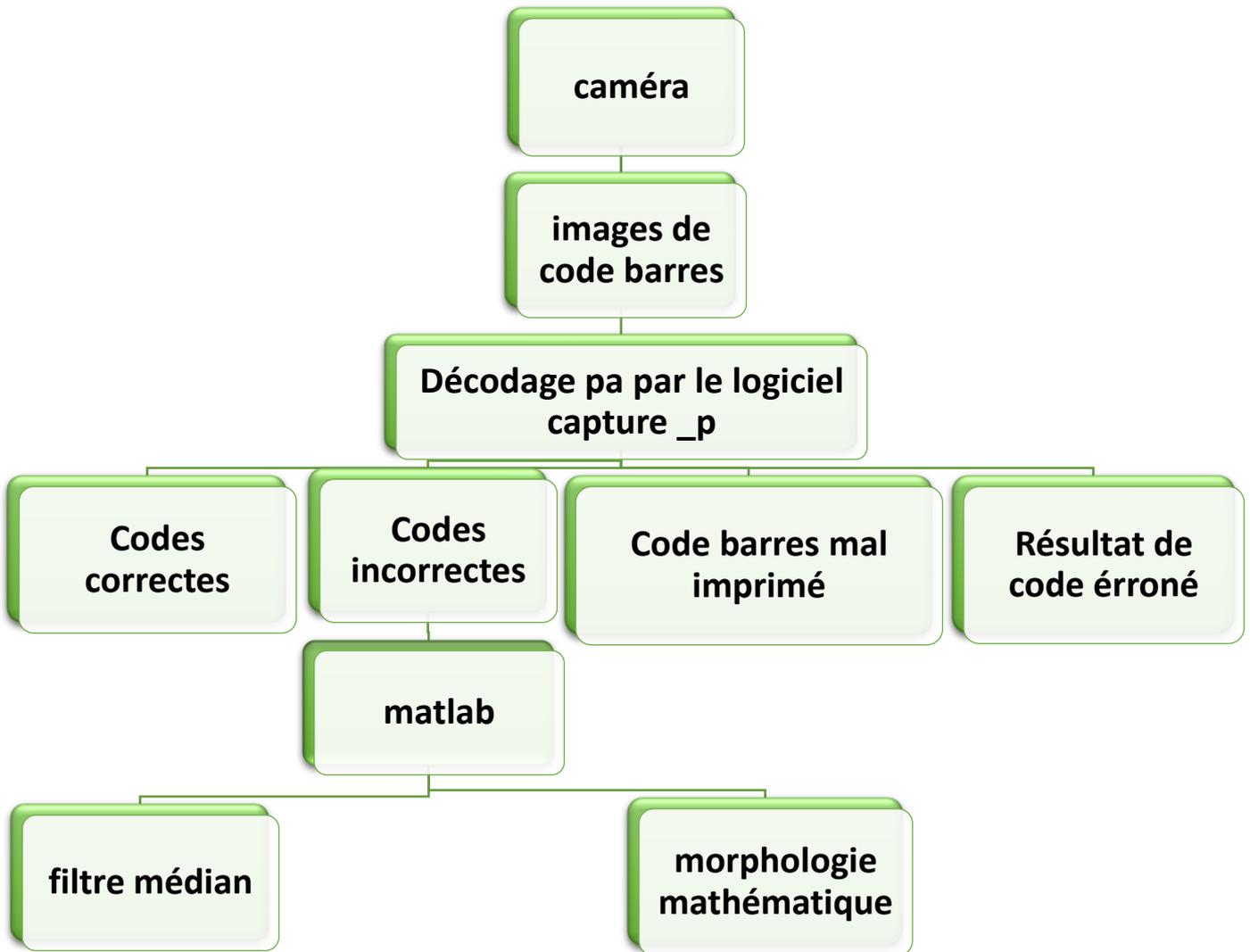


Figure 3.3 : illustration d'organigramme du système proposé.

3.5. Applications et test :

Dans cette partie nous allons présenter notre travail d'application et test sur des images de code à barres abimées acquises par Smartphone durant stage au supermarché Famili Shop de Blida. Tout d'abord nous allons tester ces images par l'application capture_p, après et suivant le résultat obtenu, nous faisons un traitement d'image sur Matlab, puis nous repassons l'image traitée encore une fois par l'application Capture_p pour voir le résultat de décodage final. Nous avons adopté le schéma synoptique de la figure 3.4.

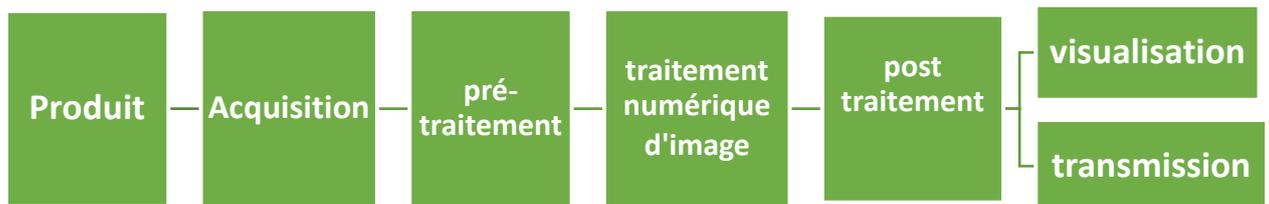


Figure 3.4 : Schéma synoptique représentant le traitement d'images sur Matlab.

- Dans la section suivante, nous allons vous montrer les différents traitements et résultats obtenus sur des images synthétisées (travail de simulation) et images réelles.

3.6. Résultats de l'application de notre système:

Pour le filtrage médian, nous avons choisi une taille du filtre de 30x1 et pour le filtrage à base de morphologie mathématique, nous avons choisi une taille de l'élément structurant de 30x1.

3.6.1. Images synthétisées :

- La figure 3.5 montre l'application de notre système à une image de codes à barres représentant des coupures au niveau des barres. Malgré la forte dégradation qui a conduit au non-décodage de l'image, le traitement par morphologie a donné un résultat très satisfaisant (voir figure 3.5 (b) et (c)).

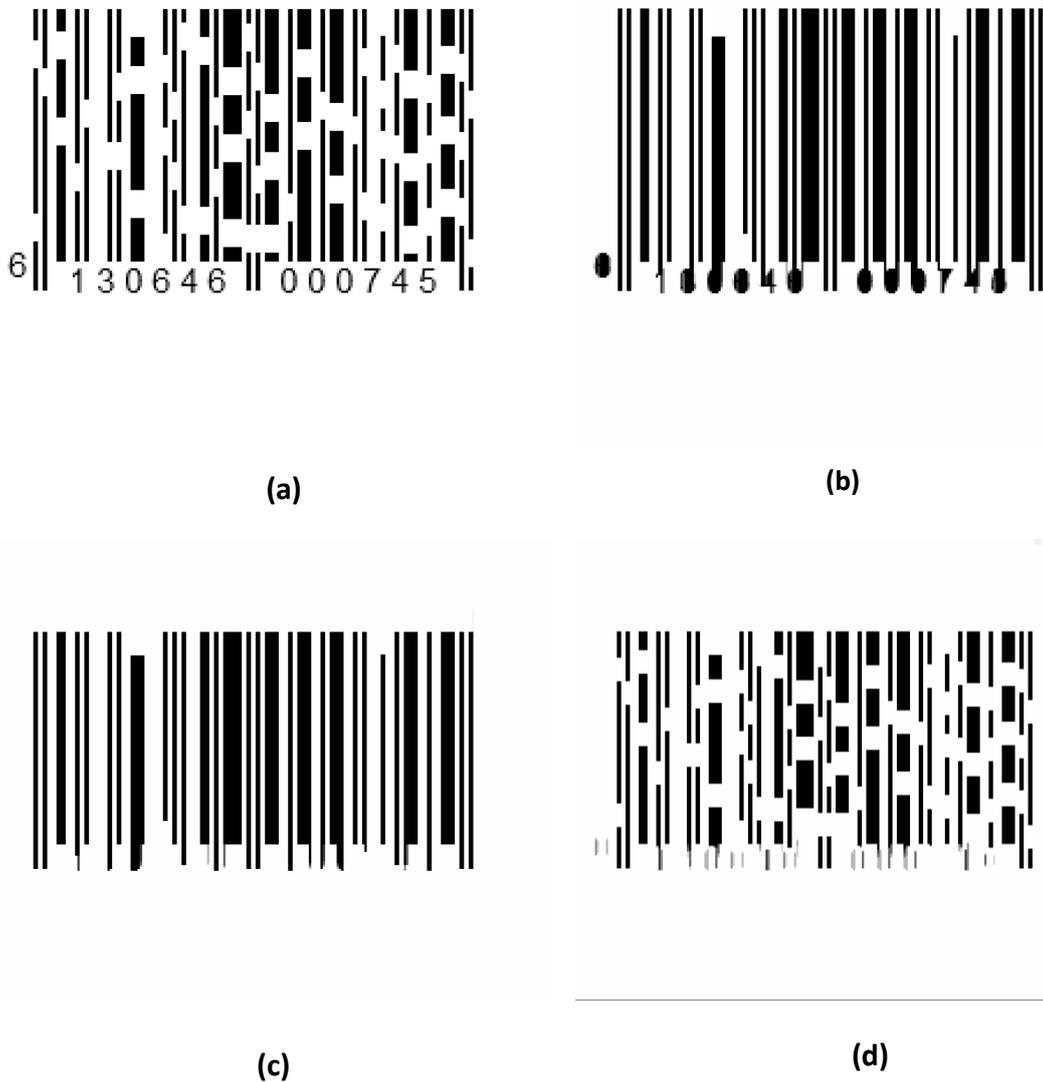


Figure 3.5 : (a) image avant le traitement, non décodée, (b) image après le traitement (ouverture), bien décodée 59 fois 6 130646 000745 , (c) image après le traitement (ouverture/fermeture) , bien décodée 59 fois 6 130646 000745. (d) image après le traitement (médian), non décodée.

- La figure 3.6 montre l'application de notre système à une image de codes à barres représentant des ratures au niveau des barres et espaces. Malgré la forte dégradation qui a conduit au non décodage de l'image, le traitement par morphologie a donné un résultat très satisfaisant (voir figure 3.6 (b) et (c)).

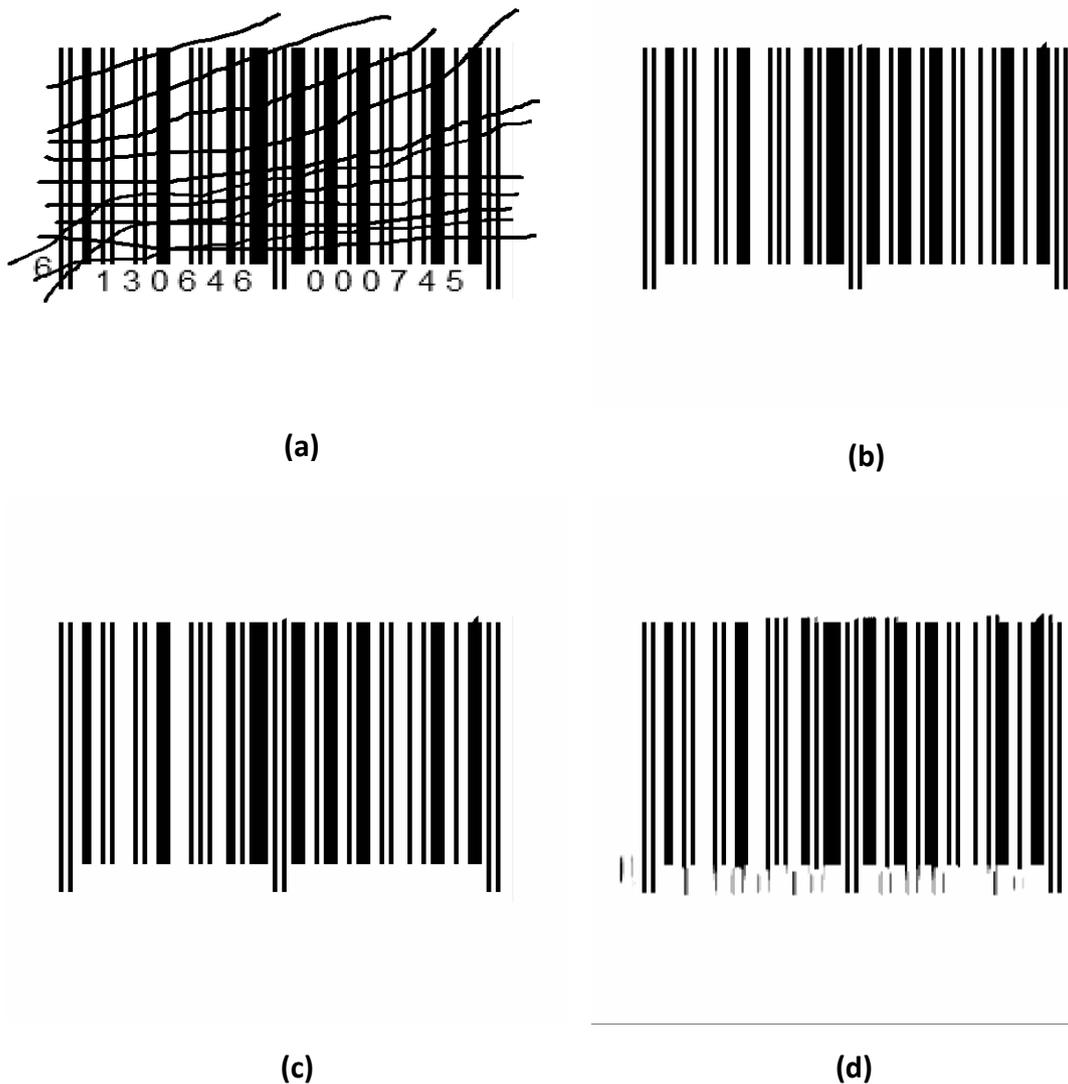


Figure 3.6 : (a) image avant le traitement, non décodée, (b) image après le traitement (fermeture/ouverture) , bien décodée 75fois 6 130646 000745 , (c) image après le traitement (fermeture), bien décodée 75 fois 6 130646 000745. (d) image après le traitement (médian), bien décodée 76 fois 6 130646 000745.

- La figure 3.7 montre l'application de notre système à une image de codes à barres représentant de fortes ratures au niveau des barres et espaces. Malgré la forte dégradation qui a conduit au non-décodage de l'image, le traitement par morphologie a donné un résultat très satisfaisant (voir figure 3.7 (b) et (c)).

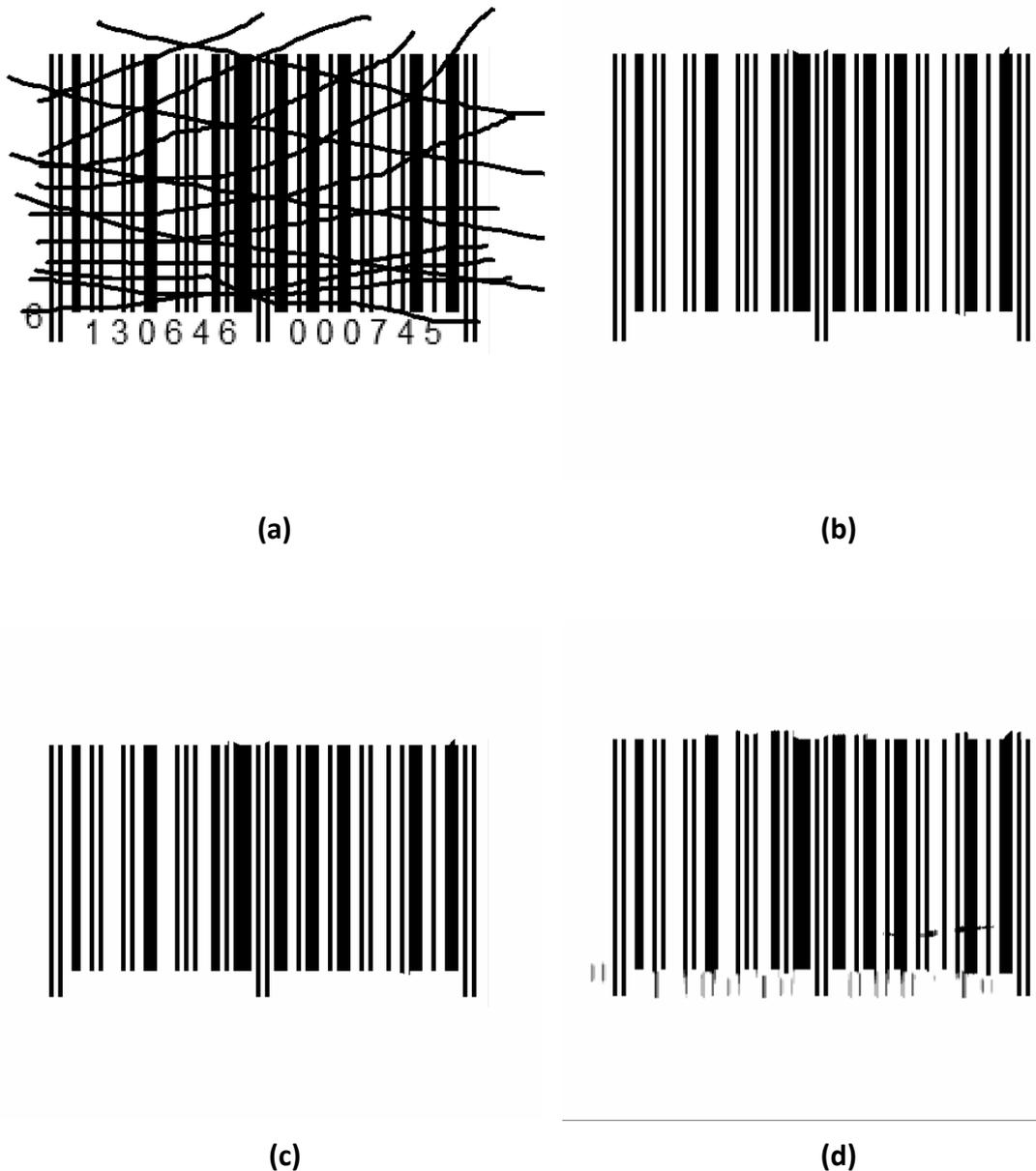
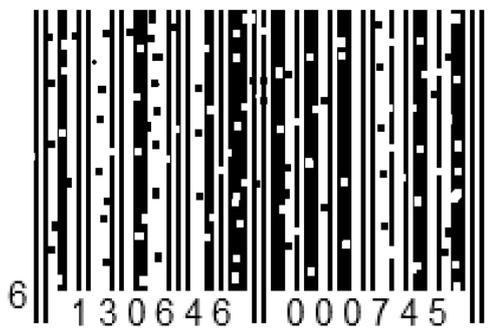
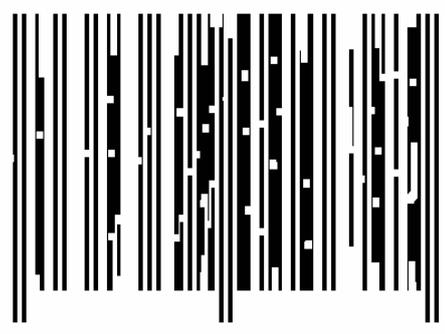


Figure 3.7 : (a) image avant le traitement, non décodée, (b) image après le traitement (fermeture), bien décodée 75 fois 6 130646 000745, (c) image après le traitement (fermeture/ouverture), bien décodée 75 fois 6 130646 000745. (d) image après le traitement (médian), bien décodée 73 fois 6 130646 000745 .

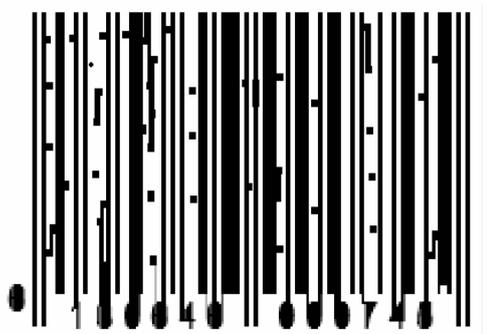
- La figure 3.8 montre l'application de notre système à une image de codes à barres représentant des tâches au niveau des barres et espaces. Malgré la forte dégradation qui a conduit au non-décodage de l'image, le traitement par morphologie a donné un résultat très satisfaisant (voir figure 3.8 (b) et (c)).



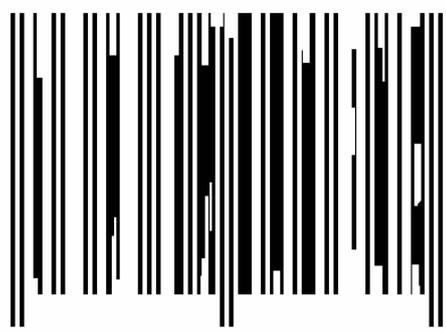
(a)



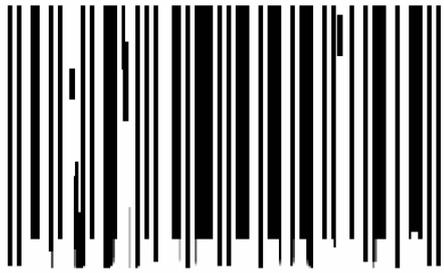
(b)



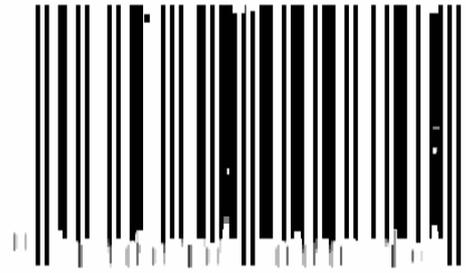
(c)



(d)



(e)

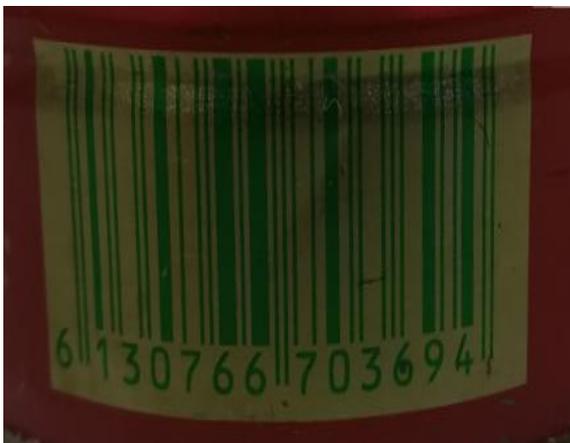


(f)

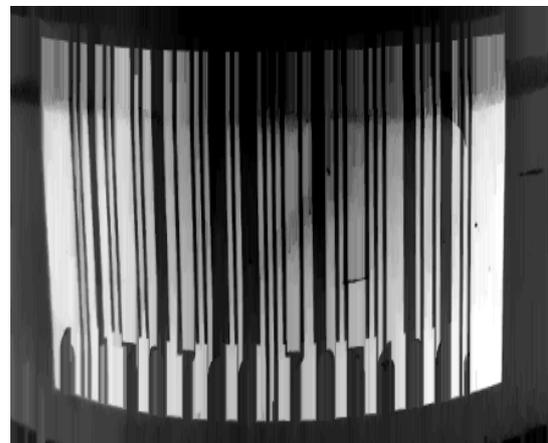
Figure 3.8 : (a) image avant le traitement, non décodée, (b) image après le traitement (fermeture), bien décodée 4 fois 6 130646 000745, (c) image après le traitement (ouverture) , bien de décodée 4 fois 6 130646 000745 , (d) image après le traitement (fermeture/ouverture) , bien de décodée 7 fois 6 130646 000745 , (e) image après le traitement (ouverture/fermeture) , bien de décodée 13 fois 6 130646 000745. (f) image après le traitement (médian) , bien de décodée 63 fois 6 130646 000745 .

- D'après les figures ci-dessus, on peut conclure que le filtrage morphologique donne de meilleurs résultats que le filtrage médian. Les résultats que nous allons voir dans la section suivante sont basés sur l'utilisation du filtrage morphologique et son application aux images réelles. Les figures 3.9 ,3.10, 3.11, 3.12 , 3.13 , 3.14 et 3.15 montrent les résultats de l'application du filtrage morphologique.

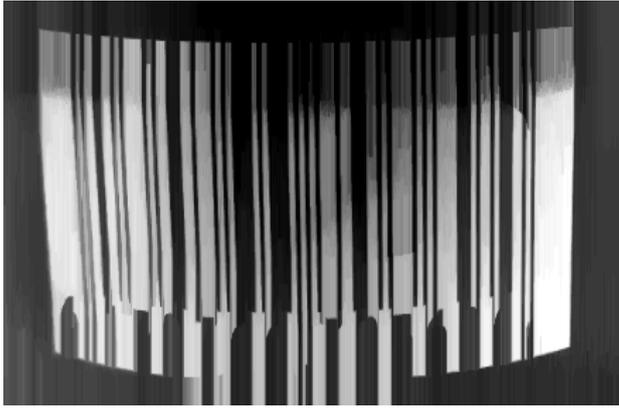
3.6.2 Images réelles :



(a)



(b)

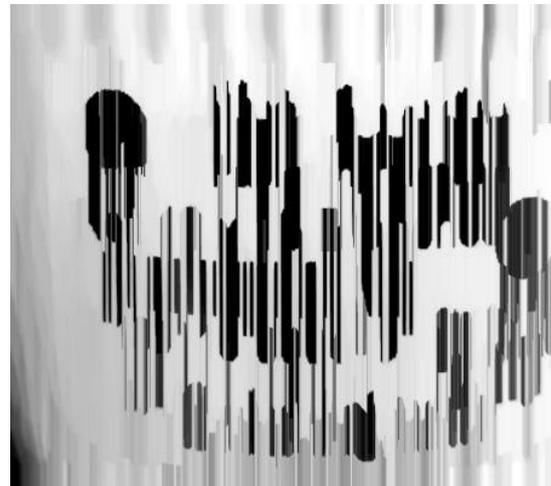


(c)

Figure 3.9 : (a) image avant le traitement, non décodée, (b) image après le traitement (ouverture), bien décodée 7 fois 6 130766 703694, (c) image après le traitement (ouverture/fermeture), bien décodée 4 fois 6 130766 703694.



(a)

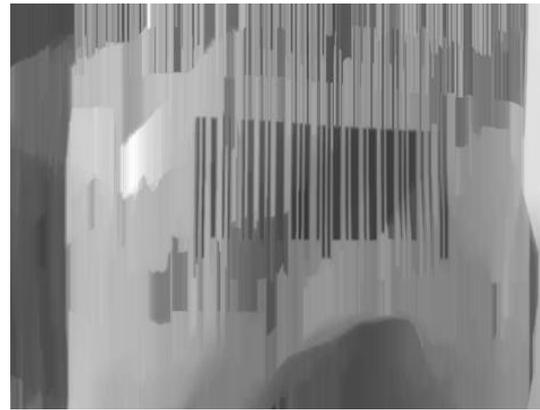


(b)

Figure 3.10 : (a) image avant le traitement, non décodée, (b) image après le traitement (ouverture/fermeture), bien décodée 1 fois 6 135116 000645 .



(a)



(b)

Figure 3.11 : (a) image avant le traitement, non décodée, (b) image après le traitement (fermeture/ouverture), bien décodée 4 fois 6 131005 000048.



(a)



(b)



(c)

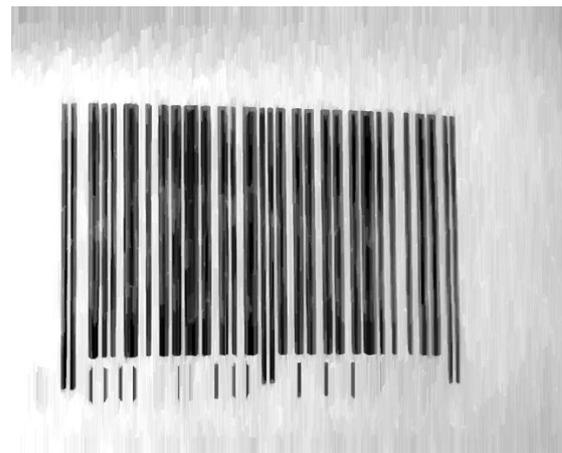


(d)

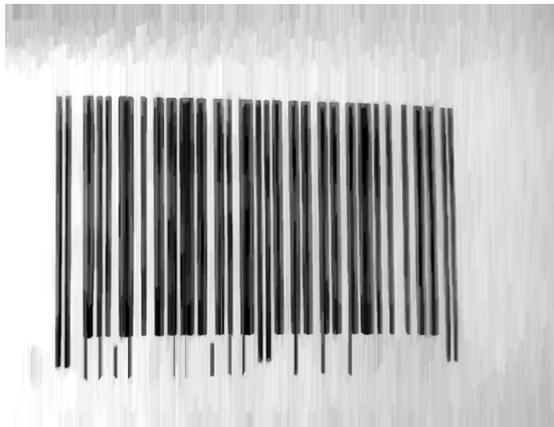
Figure 3.12 : (a) image avant le traitement, non décodée, (b) image après le traitement (ouverture) bien décodée 3 fois 6 135367 000128, (c) image après le traitement (ouverture/fermeture), bien décodée 2 fois 6 135367 000128 , (d) image après le traitement (fermeture/ouverture), bien décodée 1 seul fois 6 135367 000128 .



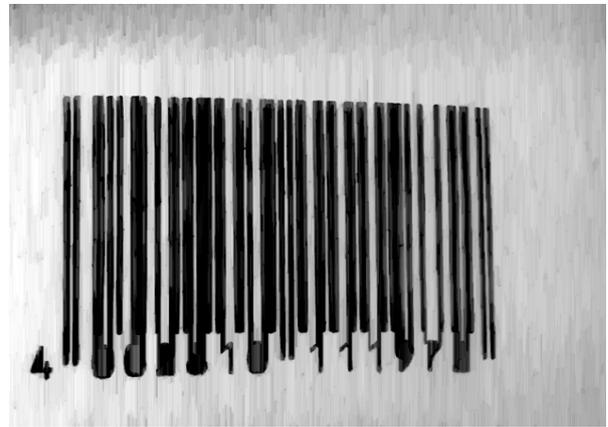
(a)



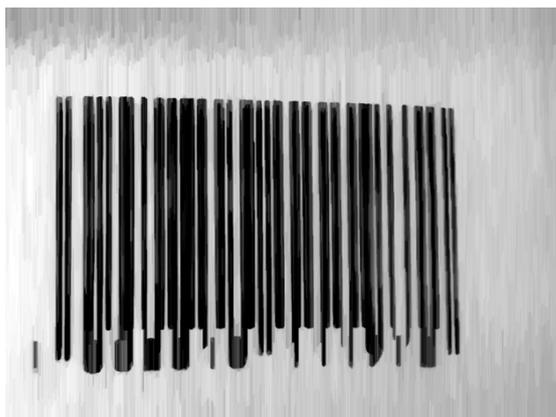
(b)



(c)

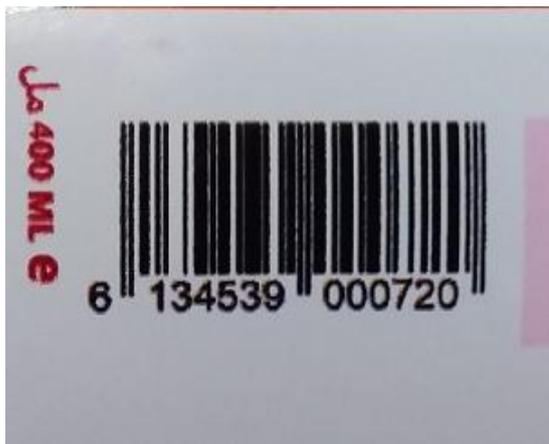


(d)



(e)

Figure 3.13 : (a) image avant le traitement, non décodée, (b) image après le traitement (fermeture), bien décodée 15 fois 4 002810 111972 , (c) image après le traitement (fermeture/ouverture), bien décodée 20 fois 4 002810 111972, (d) image après le traitement (ouverture) , bien décodée 8 fois 4 002810 111972, (e) image après le traitement (ouverture/fermeture) , bien décodée 4 fois 4 002810 111972 .



(a)



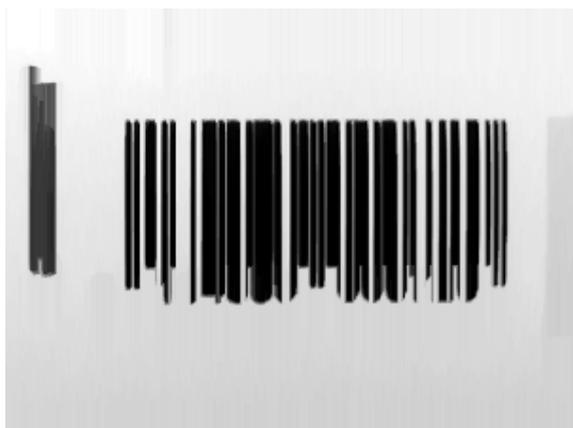
(b)



(c)



(d)



(e)

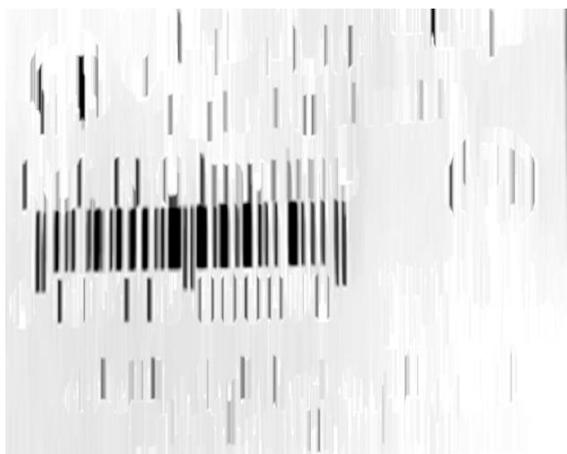
Figure 3.14 : (a) image avant le traitement, non décodée, (b) image après le traitement (fermeture), bien décodée 1 seul fois 6 134539 000720, (c) image après le traitement (fermeture/ouverture), bien décodée 1 seul fois 6 134539 000720 , (d) image après le traitement (ouverture) , bien décodée 1 seul fois 6 134539 000720, (e) image après le traitement (ouverture/fermeture) , bien décodée 1 seul fois 6 134539 000720 .



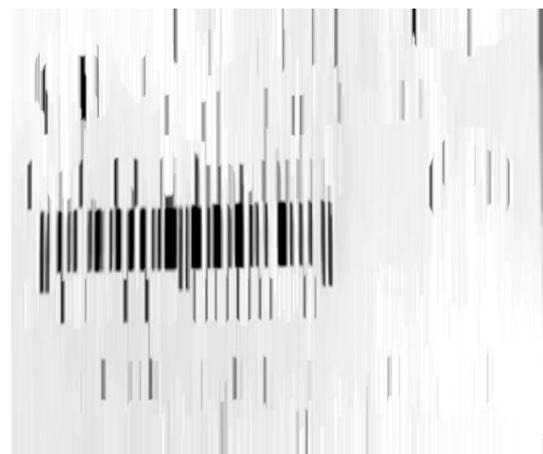
(a)



(b)



(c)



(d)

Figure 3.15 : (a) image avant le traitement, non décodée, (b) image après le traitement (ouverture/fermeture) bien décodée 3 fois 6 135116 000898, (c) image après le traitement (fermeture), bien décodée 2 fois 6 135116 000898, (d) image après le traitement (fermeture/ouverture), bien décodée 2 fois 6 135116 000898.

- La figure 3.16 , Quelques images qui n'ont pas subi de traitement et sont correctement décodé.



(a)



(b)



(c)



(d)

Figure 3.16 : (a) image, bien décodée sans traitement, 6 135263 000147, (b) image , bien décodée sans traitement ,6 135636 000118 ,(c) image , bien décodée sans traitement, 6 132004 000022, (d) images décodées , sans aucun traitement , 6 130004 300111 .

3.7 Résultats obtenus :

Une base de 66 images de codes à barres a été acquise au niveau des caisses de Famili Shop. La figure 3.17, montre le nombre d'images de code à barres décodé et non décodé en appliquant notre système de restauration.

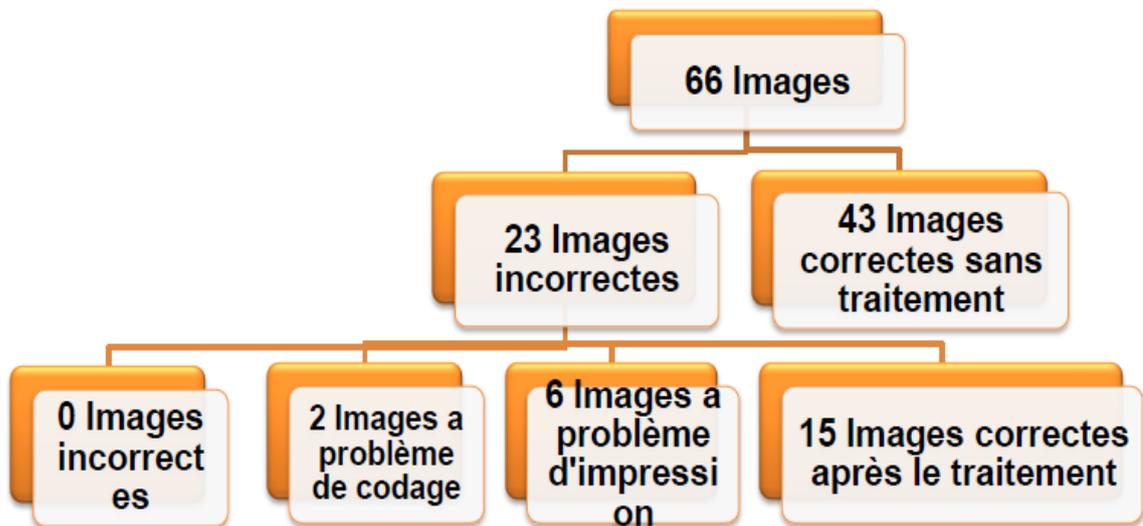


Figure 3.17 : Nombres d'images correctement décodés et non ou mal décodés.

- Ces images ont servies de test en utilisant l'application Capture_p, ce qui a donné 43 d'images de code barres correctement décodés et 23 autres non ou mal décodé. Ce résultat montre un taux de décodage de TD:

$$TD = 100 \times \frac{\text{Nombre correctement décodé}}{\text{Nombre total}} = 100 \times \frac{43}{66} = 65\%$$

- Les 23 codes barres mal ou non décodés ont subit un traitement moyennant un filtrage morphologique qui apparemment donne de meilleur résultat que le filtrage médian. Les résultats obtenus sont satisfaisants montrant un nombre de 15 images correctement décodées parmi les 23 images, d'où l'obtention d'un nouveau résultat du taux de décodage :

$$TD = 100 \times \frac{\text{Nombre correctement décodé}}{\text{Nombre total}} = 100 \times \frac{43 + 15}{66} = 88\%$$

- D'après la figure 3.17 on remarque que 02 images représentent des images mal codés, cela veut dire que le logiciel Capture_p affiche deux codes qui n'ont rien avoir avec les codes affichés en chiffre sous les codes barre (voir figure 3.18 et figure 3.19).

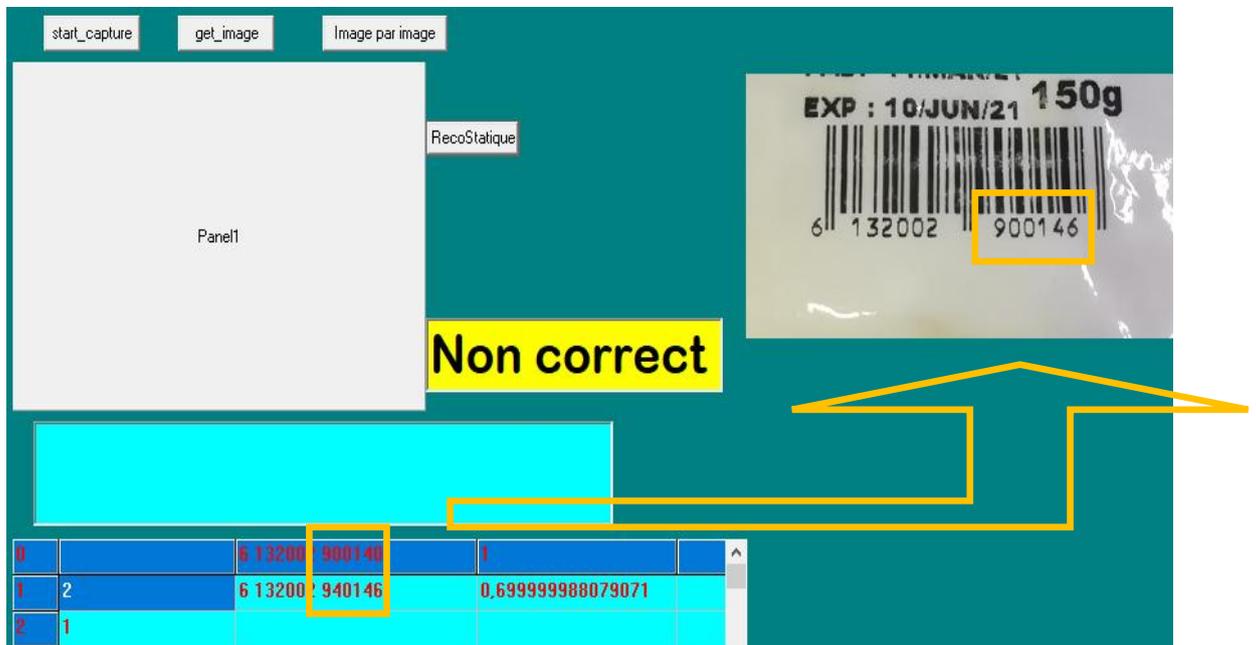


Figure 3.18 : Première image non décodée, problème de codage ,6 132002 900146 .

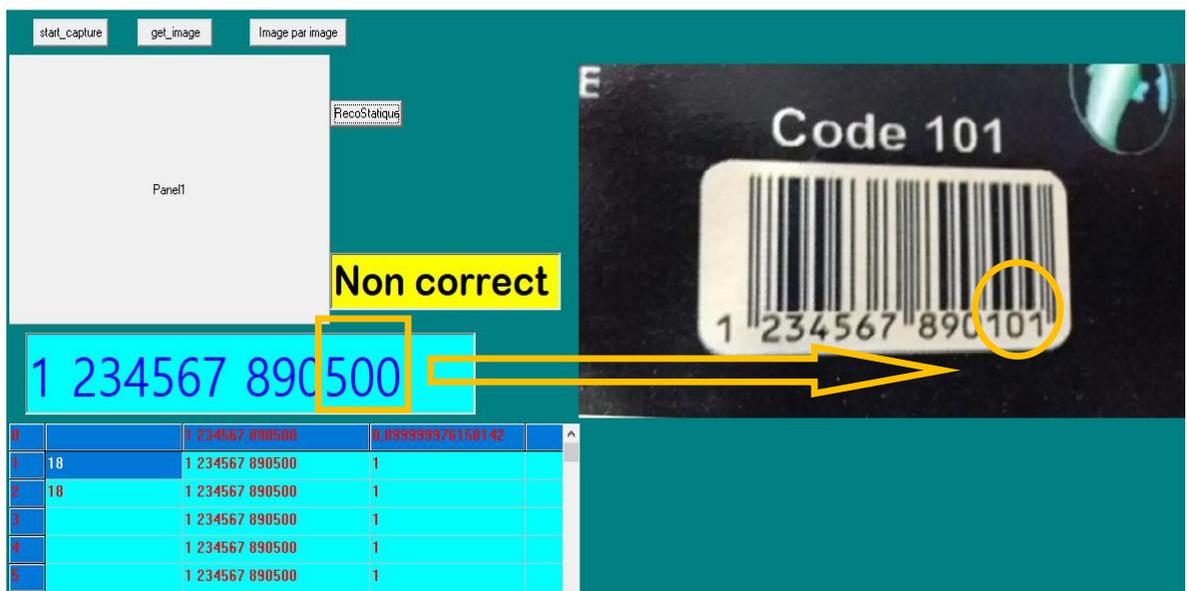


Figure 3.19 : deuxième image non décodée, problème de codage , 1 234567 890101

- Les 06 autres images n'ont pas été décodées, ces codes à barres présentent un problème d'impression comme le montre les figures 3.20, 3.21, avec des barres qui manquent au début et à la fin du code barres.



Figure 3.20 : image non décodée, problème d'impression, manque de 2 barres au début du code barres.



Figure 3.21 : image non décodée, problème d'impression, manque de 2 barres au début du code barres.

- Nous pouvons conclure que le système proposé est prometteur, puisqu'il arrive à décoder correctement des codes à barres abîmés que les lecteurs laser ne peuvent pas.

3.8 Conclusion :

Dans ce chapitre, nous avons montré comment réparer un code à barres à l'aide de techniques de traitement d'images avec le logiciel Matlab.

Pour nous assurer que le code à barres a été bien réparé ou corrigé, nous avons utilisé une application (capture _p) créé par notre encadreur. Au cours de notre travail nous avons trouvés des images de codes à barres qui sont non décodable par le lecteur laser mais par contre sont décodable par notre système.

D'après les résultats, nous pouvons conclure que notre système (logiciel MATLAB + application capture _p) donne de bonnes performances mieux que le lecteur laser.

Conclusion générale

En guise de conclusion générale, nous allons tenter d'établir une synthèse globale sur le travail qui a été réalisé dans le cadre de ce projet de fin d'étude.

L'objectif de ce travail est de proposer une solution pour restaurer des images code-barres EAN13 abimés figurant sur les produits de consommations afin de les rendre décodable aux niveaux des caisses. Pour cela, nous avons proposé dans ce travail une technique de réparation des codes à barres. Ce travail s'articule essentiellement autour de deux parties. La première partie concerne l'acquisition et traitement de l'image, quant à la deuxième partie, elle concerne bien l'utilisation d'un logiciel de décodage pour lire l'image restaurée.

Pour bien mener notre projet, nous avons divisé notre travail en trois chapitres, le premier chapitre est consacré aux généralités de code à barres. Dans le deuxième chapitre, nous avons abordé les notions de base nécessaires à la compréhension des techniques de traitement d'images. Nous avons proposé les méthodes et les opérations que nous avons utilisé dans ce travail. Le troisième chapitre est consacré dans un premier stade à l'applications et test sur des images synthétisées (générées par logiciel) pour tester la précision et fiabilité du système proposé. Une fois le travail de simulation a été mené à bien, nous somme passées aux tests sur des images réelles.

Les codes à barres sont très utiles dans notre vie quotidienne facilitent l'activité dans différents domaines, d'après notre étude et notre application de traitement sur les images de code à barres abimées nous conclurons que nous pouvons régler et proposer une solution, et les résultats que nous avons obtenus sont satisfaisants.

Nous proposons comme travail de perspective d'utiliser un système embarqué sur une carte Raspberry muni d'une caméra qui va transmettre le code correct via une sortie USB.

Liste de références bibliographiques :

- [1] : Ce qu'il faut savoir sur le code à barre , GS1 TUNISIA.
- [2] : Pradier consulting , Alain Pradier.
- [3] : <http://www.cipam.com/identification-automatique/generalites-sur-le-code-a-barres.html>.
- [4] : <http://www.gomaro.ch>.
- [5] : <http://www.eanint.org>.
- [6] : Les codes barres, Aurélien Jarno,1998-2008.
- [7] : Guerfi, Souhila. Authentification d'individus par reconnaissance de caractéristiques biométriques liées aux visages 2D/3D. Diss. Université d'Evry-Val d'Essonne, 2008.
- [8] : Boukhlof, Djemaa. Résolution de problèmes par écosystèmes: Application au traitement d'images. Diss. Université Mohamed Khider-Biskra, 2005.
- [9] : https://fr.wikipedia.org/wiki/Traitement_d%27images
- [10] : BENFRIHA Saraa, HAMEL Asma. « Segmentation d'image par Coopération région-contours », Mémoire Pour l'obtention du diplôme de MASTER PROFESSIONNEL en Informatique Option Réseaux convergence et sécurité université KasdiMerbah-OuarglaUniversité ,2016 .
- [11] : http://www.traitement-signal.com/traitement_d_images.php .
- [12] : Patrick , Bonnin , developpez.com/cours/vision/Bases-du-Traitement-Image/Chap1, Publié le 17 mars 2016 - Mis à jour le 28 mai 2016 .
- [13] : DERDOUR, Khedidja. Reconnaissance de formes du chiffre arabe imprimé: Application au code à barre d'un produit. Diss. Université de Batna 2, 2009.
- [14] : Katona, M. and Nyul, L. G. , Efficient 1D and 2D , barcode detection using mathematical morphology.
- [15] : In 11th International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing, pages 464–475, (2013).
- [16] : Site web PLASTIC- filtrage linéaire .

[17] : HADJ ABDELKAFER khir eddine, BOUBOUAHINE belhadj , réduction du bruit des images par filtrage spatial ,Aéronautique ,option installation , université Blida ,2007-2008.

[18] : https://fr.wikipedia.org/wiki/Filtre_médian

[19] : [French Wikipedia](#).Traitement d'image , on 27-Jul-2004.

[20] : Compression des images fixes par fractales basée sur la triangulation Delaunay et la quantification vectorielle ,PFE USTHB 1999.

[21] : M.ANDRE Eyrolles ,Introduction aux techniques de traitement d'images,1987.

[22] : R.C.GONZALES & P.WINTZ ,Digital Image Processing -Addison Wesley - ,1977.

[23] : Encyclopédie ENCARTA,MICROSOFT 1997.

[24] : M.HADALLAH ,Codage des images fixes par une méthode hybride basée sur la QV et les approximations fractales, PFE USTHB1997.

[25] : M.KUNT ,Traitement numérique des images, 1993.

[26] : Compression d'images animées à très faible débit par la géométrie des fractales,PFE USTHB1996

[27] : Fou du multimédia ,A.d'HARDANCOURT Sybex,1995.

[28] : [Daniel Peraya](#) ,Les types d'images

[29] : S.DEKKAR PFE INI ,Compression des images fixes par la méthode des fractales ,1996.

[30] : G.BARTHOLOMEW ,Opérateur de traitement d'image,1920.

[31] : D.MARR ET HILDRETH ,Theory of edge detection,1980.

[32] : https://fr.wikipedia.org/wiki/Traitement_d%27images.

[33] : Christelle, Gendrin . Chemical imaging and chemometrics for the analysis of pharmaceutical solid dosage forms , November 2008 .

[34] : Mira Ouassila. Saada Sara, Présenté pour l'obtention du diplôme de MASTER, UNIVERSITE DE M'SILA -MOHAMMED BOUDIAF,2019/2020.

[35] : Christian RONSE , ICube, CNRS, Département d'Informatique de l'Unistra,22/10/2013.

[36] : <https://fr.mathworks.com/products/matlab.html>.

- [37] : Pierre Bonnet, Outils de simulation, USTL - Master ASE M1, 2009-2010.
- [38] : E.N.S.B.A.N.A. Initiation au traitement d'images avec MATLAB ,2ème année - TD d'informatique scientifique - 2002/2003.
- [39] : B. BOULEBTATECHE.cours : IMAGERIE ET VISION INDUSTRIELLE, chapt 7 : Traitement d'Images Binaires Par Morphologie Mathématique, 2020 .
- [40] : Hattab bey Zineb , Nebih Nesma Rim, Les interactions verbales en classes de langues ; Cas des élèves de 4AM , Université LOUNICI ALI -Blida 2- Septembre 2018.
- [41] : LAMRI , LAOUAMER, APPROCHE EXPLORATOIRE SUR LA CLASSIFICATION APPLIQUÉE AUX IMAGES, L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES, avril 2006.