



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران 2 محمد بن أحمد
Université D'Oran 2 Mohamed Ben Ahmed

معهد الصيانة والأمن الصناعي
Institut de Maintenance et de Sécurité Industrielle

Département de Maintenance en Instrumentation

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industriel

Spécialité : Ingénierie de la Maintenance en Instrumentation

Thème

Détection et Reconnaissance d'objet avec MATLAB

Présenté et soutenu publiquement par :

SIBACHIR Abdenour & OUSALEM Imane

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
Djilali Nekrouf	MAA	IMSI-Univ. D'Oran2	Président
Adda Neggaz Samir	MAA	IMSI-Univ. D'Oran2	Encadreur
MEKKI Ibrahim El Khalil	MCA	IMSI-Univ. D'Oran2	Examineur

2023 / 2024



Dédicace

Je dédie le fruit de ce travail tout d'abord à la lumière de ma vie, mon père et mon supporteur « **Boumediene** », grâce à lui je n'ai jamais rien manqué dans la vie, l'homme qui a veillé à mon éducation, j'espère que ce modeste travail soit l'exaucement de ses vœux.

A celle qui a toujours été ma source d'amour inconditionnel, de force et de sagesse ma grand-mère « **OUKACI Zahra** ». Tes sacrifices et ta bienveillance ont été les fondations sur lesquelles je me suis construite. Je t'aime infiniment.

À ma grand père, « **Belkacem** », qui est comme un deuxième père pour moi, pour son amour et ses sages conseils

A mes piliers et mes complices de vie mes petites sœurs que j'aime « **Samia, Sabrina, Yasmine** ».

A ma source d'inspiration et de bonheur mon frère adoré « **Moussa Abdelhak** ».

A mes tentes « **Chahrazed, Messaouda, Warda, Nacira** » Cette dédicace vous est adressée, témoignant de la gratitude et de l'affection que je porte à chacune d'entre vous. Vos conseils, votre soutien et votre amour inconditionnel ont enrichi ma vie d'une manière que je ne saurais exprimer en mots. Puissiez-vous toujours être entourées de bonheur et de succès.

A ma mère « **AIT MOUHEB Nadia** ».

A mes petits cousins que j'aime trop.

A mon cher ami avant d'être binôme « **SIBACHIR Abdenour** » pour son entente et sa patience. À ma meilleure amie, complice de mes joies et de mes peines, à celle qui a illuminé ma vie de son amitié sincère « **MOUAZ Rania** ». Puisses-tu toujours trouver dans ces lignes une parcelle de notre complicité et de notre amour fraternel.

A mon âme sœur « **MOKKADEM Ferial** », qu'aucune dédicace ne pourrait véritablement exprimer la profondeur de ma gratitude pour tout ce que tu as fait pour moi lors de nos moments inoubliables partagés ensemble.

A mes amies « **Kenza, Amel, Wissal, Bouchra, Amira, manel** » en souvenir de notre sincère et profonde amitié et les moments agréables que nous avons passés ensemble.

A tous mes amis qui sont nombreux plus particulièrement « **Walid, Habibou, Othman, Malik** » Avec qui j'ai partagé les moments de stress, de tristesse et de la joie.

A mon amie d'enfance que j'aime énormément « **Najwa** ».

A tous les amis et collègues de l'**IMSI** et tous ceux qui m'estime et pense à moi.

OUSSALEM Imane

Dédicace

Je dédie ce mémoire à ma famille, source inépuisable de soutien et d'inspiration.

À mon père et à ma mère, pour leur amour inconditionnel, leur patience et leur encouragement constant.

À mes frères « **Ahmed** » et « **Hakou** », et à mes sœurs « **Iness** » et « **Amina** », pour leur soutien indéfectible et leur affection.

À ma grande mère, « **KHALLADI Fatna** », qui est comme une deuxième maman pour moi, pour son amour et ses sages conseils.

À ma binôme « **OUSALEM Imane** », pour sa collaboration précieuse et son amitié.

À mon frère et ami « **TOUATI Abderrahmane** » et « **ZIANE Walid** », pour leur présence et leur soutien tout au long de ce parcours.

À mes amis « **Otman** » et « **Habibou** », pour leurs encouragements et leur amitié sincère.

Merci à vous tous d'avoir cru en moi et de m'avoir accompagné dans cette aventure.

SIBACHIR Abdenour

Remerciements

On premier lieu nous tenons à remercier dieu le tout puissant de nous avoir donné la force, et la patience pour terminer nos études et pour élaborer ce modeste travail.

Au terme de ce travail, nous tenons à exprimer notre sincère gratitude à notre encadrant **Adda Neggaz Samir** pour son aide, ses précieux conseils, son soutien et sa disponibilité tout au long de cette aventure.

Merci à nos parents d'avoir su nous écouter et nous motiver tout au long de notre démarche.

Que les membres de jury trouvent, ici, l'expression de nos sincères remerciements pour l'honneur qu'ils nous font en prenant le temps de lire et d'évaluer ce travail.

Nous souhaitons adresser nos remerciements les plus sincères au corps professoral et administratif de l'Institut de Maintenance et De Sécurité Industrielle d'Oran pour leurs efforts durant tous nos années d'études.

Nous voudrions également remercier et exprimer nos gratitude envers nos ami(e)s et nos collègues de l'étude pour les moments de joie, de partage et d'entraide. Votre amitié a été une véritable bouffée d'air frais durant cette période intense.

Pour finir, on souhaite remercier toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

Résumé

Résumé

Ce mémoire se concentre sur l'implémentation en temps réel de techniques de détection et de segmentation d'objets dans des vidéos en utilisant MATLAB. Les algorithmes de détection d'objets tels que Sobel, Prewitt, Canny et YOLOv3 sont combinés avec des méthodes de segmentation pour une analyse précise et efficace des flux vidéo en temps réel. L'étude comprend une évaluation approfondie de la performance des techniques de détection et de segmentation dans des scénarios de surveillance vidéo en temps réel.

Les mots-clés : MATLAB, détection en temps réel, détection d'objets, segmentation, Sobel, Prewitt, Canny, YOLOv3, traitement d'images, détection de contours, surveillance vidéo, implémentation, performance, évaluation.

Abstract

This thesis focuses on real-time implementation of object detection and segmentation techniques in videos using MATLAB. Object detection algorithms such as Sobel, Prewitt, Canny, and YOLOv3 are combined with segmentation methods for precise and efficient analysis of video streams in real-time. The study includes a comprehensive evaluation of the performance of detection and segmentation techniques in real-time video surveillance scenarios.

Keywords: MATLAB, real-time detection, object detection, segmentation, Sobel, Prewitt, Canny, YOLOv3, image processing, edge detection, video surveillance, implementation, performance, evaluation.

ملخص

يتمركز هذا الأطروحة على تنفيذ في الوقت الحقيقي لتقنيات كشف وتقسيم الكائنات في مقاطع الفيديو باستخدام برنامج MATLAB. يتم دمج خوارزميات كشف الكائنات مثل Sobel و Prewitt و Canny و YOLOv3 مع طرق التقسيم لتحليل دقيق وفعال لتدفقات الفيديو في الوقت الحقيقي. تشمل الدراسة تقييما شاملا لأداء تقنيات الكشف والتقسيم في سيناريوهات مراقبة الفيديو في الوقت الحقيقي.

الكلمات الرئيسية : MATLAB ، الكشف في الوقت الحقيقي، كشف الكائنات، التقسيم، Sobel ، Prewitt ، Canny ، YOLOv3، معالجة الصور، كشف الحواف، مراقبة الفيديو، التنفيذ، الأداء، التقييم.

Table de matières

Dédicace.....	
Dédicace.....	
Remerciements.....	
Résumé.....	
Abstract	
Les abréviations	
Introduction générale.....	1
Chapitre I : Généralités sur l'images	3
1 Introduction	3
2 L'image numérique	3
2.1 Définition	3
3 Les différents types d'image	3
3.1 L'imagerie vidéo	4
3.1.1 Définition d'une vidéo	4
3.1.2 Résolution vidéo.....	4
3.1.3 Résolution temporelle d'une vidéo.....	5
3.1.4 Définition d'un objet vidéo	5
3.1.5 Séquence d'image	6
3.1.6 Nature de la scène	6
4 Les formats d'image.....	7
4.1 Image numérique matricielle (Bitmap).....	7
4.2 Image numérique Vectorielle	8
5 Les caractéristiques de l'image.....	8
5.1 Dimension	9
5.2 Pixel	9
5.3 Résolution et définition	10
5.4 Contraste.....	10
5.5 Contour.....	11
5.6 Bruit	11
5.7 Histogramme	11
5.8 Codage.....	12
5.8.1 Images binaires.....	12

5.8.2	Images en niveaux de gris	13
5.8.3	Images en couleurs	13
5.8.4	Images en couleurs indexées	14
5.8.5	Le système HSV (Hue, Saturation, Value) ou TSL (Teinte, Saturation, Luminosité) ...	14
6	Conclusion.....	15
Chapitre II : Traitement d'image.....		17
1	Introduction	17
2	Traitement d'images.....	17
3	Prétraitement d'image	18
3.1	Filtrage	19
3.1.1	Filtres linéaires	19
3.1.2	Filtre moyenné (lissage)	19
3.1.3	Filtre gaussien	20
3.1.4	Filtre rehausseur de contours.....	20
3.1.5	Filtres non linéaires	21
3.2	Binarisation	22
3.3	Les opérateurs de la morphologie mathématique	23
3.3.1	Erosion	23
3.3.2	Dilatation.....	23
3.3.3	Utilisation de l'érosion et de la dilatation.....	24
3.4	Segmentation	25
3.4.1	Définition.....	25
3.4.2	Type de segmentation.....	26
3.4.3	Technique de segmentation	28
4	Conclusion.....	36
Chapitre III : Détection et suivi d'objet.....		37
1	Introduction	37
2	Suivi d'objet.....	37
2.1	Définition	37
2.2	Domain d'application	38
2.3	Les défis du suivi.....	39
2.3.1	Occultations.....	39
2.3.2	Changement d'illumination	40
2.3.3	Changement d'échelle	41

2.3.4	Changement du modèle d'apparence.....	41
2.3.5	Mouvements de caméra.....	42
2.3.6	Objet en mouvement rapide	42
2.4	Méthodes de suivi.....	43
2.4.1	Suivi par points.....	43
2.4.2	Suivi de silhouettes.....	44
2.4.3	Suivi de noyaux.....	45
2.5	Caractéristiques visuelles pour le suivi d'objets	45
2.5.1	Couleur.....	46
2.5.2	Texture	46
2.5.3	Gradient.....	46
3	Détection d'objets.....	47
3.1	Définition	47
3.2	Approches-en une seule étape.....	48
3.2.1	You Only Look Once (YOLO).....	48
3.3	Approche en deux étapes (basée sur la région).....	51
3.3.1	Region Convolutional Neural Network (R-CNN)	51
3.3.2	Rapid R-CNN (Fast R-CNN).....	52
3.3.3	Faster R-CNN.....	53
3.3.4	Mask R-CNN.....	53
4	Conclusion.....	54
Chapitre IV : Implémentation et discussion des résultats		55
1	Introduction	55
3	Présentation de la boîte à outils GUIDE.....	56
4	Comment créer un guide sur MATLAB ?.....	56
5	Ipcam.....	58
6	Les organigrammes d'implémentations.....	59
6.1	Algorithme principale.....	59
6.2	Organigramme de Détection de Contours	61
6.3	Organigramme de Segmentation de Couleurs	66
6.4	Organigramme d'Ajustement de la Luminosité	67
6.5	Organigramme d'érosion et délitation.....	68
6.6	Organigramme de cette application	71
7	Resultats et discussions	72

7.1	Detection des contours	72
7.1.1	Sobel.....	72
7.1.2	prewit.....	72
7.1.3	Canny	72
7.2	Traitement morphologique	73
7.2.1	Érosion	73
7.2.2	Dilatation.....	73
7.3	HSV.....	74
7.4	Segmentation	75
7.5	Surprissions de l'arrière-plan en temps réelle	76
7.6	YOLOv3 en temps réelle.....	76
8	Conclusion.....	77
	Conclusion Générale	78
	Annexe A	
	Annexe B	
	Annexe C	
	Références.....	

Liste des figures

Chapitre I

Figure I. 1: Exemple d'une séquence d'images	6
Figure I. 2: Images matricielle et vectorielle	8
Figure I.3: Distribution des pixels par lignes et colonnes.....	9
Figure I.4: Explication de résolution d'une image	10
Figure I. 5 : Contraste de luminosité	11
Figure I.6 : Exemple bruit de dans une image	11
Figure I.7: Exemple de l'histogramme d'une image	12
Figure I.8: Codage binaire (0,1).....	12
Figure I.9: 256 nuances de gris	13
Figure I.10: Palette de 256 couleurs utilisées.....	14
Figure I.11: Image codée en couleurs indexées	14

Chapitre II

Figure II.1: Chaîne de traitement d'images	18
Figure II.2: Application de filtre moyenné (lissage) (A) image originale (B) après lissage	19
Figure II.3: Exemple d'utilisation du filtre rehausseur de contours	20
Figure II.4: Binarisation d'une image : (A) image originale, (B) image binarisée.	22
Figure II.5: Erosion d'une image (A) image originale (B) image érodée	23
Figure II.6: Dilatation d'une image (A) image originale (B) image dilatée	24
Figure II.7: Comparaison entre érosion et dilatation	25
Figure II 8: Segmentation sémantique. (A) image originale, (B) image segmentée en deux classes, personne et arrière-plan.	26
Figure II.9: Segmentations par instance.	27
Figure II.10: Segmentation panoramiques.	27
Figure II.11: Seuillage simple d'un histogramme	28
Figure II.12: Seuillage multiple d'un histogramme.	29
Figure II.13: Résultat de la segmentation via la méthode de croissance de régions, (a) : image originale, (b) : image binaire, (c) : image segmentée	31
Figure II.14: Résultat de la segmentation via la méthode division-fusion (split and merge) (A) : image initiale, (B) : application de l'algorithme de division, (C) : image segmentée après l'étape de fusion.	32
Figure II.15: (A) Image originale, (B) Image Segmentée par Contour	32
Figure II.16: Quelques modèles de contours.	33
Figure II.17: Image originale, utilisation du Öltre de sobel, prewitt	34
Figure II.18: Image Originale Boat et Détecteur de Canny	36

Chapitre III

Figure III.1: Illustration du suivi par la détection basée sur la classification des SVM. (A) montre la carte de score de visage / non-visage classification ; (B) affiche la zone de recherche et de la région de contexte pour ; (C) trace l'hyperplan de classification	38
Figure III.2: Quelques applications de suivi d'objets	39
Figure III.3: Exemple d'occultation : (A) : occultation partielle, (B) : occultation totale.	40
Figure III.4: Exemple de changement des conditions d'éclairage lors d'un suivi de visage sur la séquence vidéo David.....	40
Figure III. 5: Exemple de changement d'échelle	41
Figure III. 6: Exemples de Changement du modèle d'apparence dû aux rotations.....	41
Figure III. 7: Exemple de mouvement de camera.....	42
Figure III.8: Exemple de mouvement rapide.	42
Figure III. 9: Représentation d'objet par points	43
Figure III. 10: Représentation d'objet par silhouette	44
Figure III. 11: Représentation d'objet le noyau	45
Figure III.12: Exemples de détection d'objet.....	47
Figure III.13: YOLO détection d'objet	48
Figure III. 14: La structure de YOLOV3	50
Figure III.15: R-CNN détection d'objet.....	52
Figure III. 16 : Architecture de Rapid RCNN.....	52
Figure III. 17 : Architecture de Faster RCNN	53
Figure III. 18 : Architecture de Mask RCNN	54
Figure III. 19 : Exemple de détection par Mask RCNN.....	54

Chapitre IV

FigureIV.1: Interface de MATLAB	57
FigureIV.2: Menu de MATLAB.....	57
FigureIV.3: Organigramme principale.....	59
FigureIV.4: Organigramme de détection de contour.....	61
FigureIV.5: Organigramme de Segmentation de Couleurs	66
FigureIV.6: Organigramme d'Ajustement de la Luminosité	67
FigureIV.7: Organigramme d'Érosion.....	68
FigureIV.8: Organigramme de délitation	69
FigureIV.9: Interface graphique de l'application.....	70
FigureIV.10: Organigramme de de cette application	71
FigureIV 11: (A) image d'origine ,(B) niveaux de gris ,(C) Sobel.....	72
FigureIV.12: (A) image d'origine , (B) niveaux de gris ,(C) prewitt.....	72
FigureIV 13: (A)image d'origine , (B) niveaux de gris ,(C) Canny.....	72
FigureIV.14: (A) image d'origine, (B) érosion.....	73
FigureIV.15: (A) image d'origine, (B) dilatation.....	73

FigureIV.16: (A) image d'origine, (B) composante H(hue), (C) composante S(saturation), (D) composante V(value).....	74
FigureIV.17: (A) image d'origine , (B) image segmentation pour les poivrons rouge(couleur blanc)	75
FigureIV.18: (A) image d'origine, (B) image segmentation pour les poivrons rouge (couleur rouge)	75
FigureIV.19: (A) image d'origine, (B) l'arrière-plan supprimé	76
FigureIV.20: Utilisation de YOLOv3 avec ipcam	76

Liste des tableaux

Tableau D1: les avantages et les inconvénients de YOLO	49
Tbleau D2: Les différences entre versions de YOLO (You Only Look Once)	49

Les abréviations

IRM	Imagerie par résonance magnétique
SD	Standard Definition
HD	High Definition
FHD	Full High Definition
BMP	Bitmap
PCX	Picture Exchange
GIF	Graphics Interchange Format
JPEG	Joint Photographic Expert Group
TIFF	Tagged Image File Format
WMF	Windows Metafile
EMF	Enhanced Metafile
CGM	Computer Graphics Metafile
EPS	Encapsulated PostScript
Dpi	Dots Per Inch
PPP	Point Par Pouce
RVB	Rouge, Vert, Bleu
CMJN	Cyan, Magenta, Jaune, Noir
HSV	Hue, Saturation, Value
TSL	Teinte, Saturation, Luminosité
RGB	Red, Green, Blue
R-CNN	Region Cnovolutional Network
RPN	Reverse Polish Notation
SVM	Support Vector Machine
YOLO	You Only Look Once
ROI	Return on Investment
SSD	Single Shot MultiBox Detector
DSOD	Deeply Supervised Object Detector
DSSD	Deeply Supervised Single Shot Detector
HOG	Histogram of Oriented Gradients
SIFT	Scale-Invariant Feature Transform
LBP	Local Binary Patterns

Introduction générale

Introduction générale

Les images jouent un rôle central dans notre société contemporaine, influençant divers aspects de notre vie quotidienne, des médias aux sciences. Dans ce contexte, la détection d'objets dans les images est une tâche cruciale et souvent complexe, ayant des implications majeures dans des domaines tels que la surveillance vidéo, la robotique, la réalité augmentée, la sécurité, et bien d'autres. Ce mémoire se concentre sur l'étude approfondie des techniques de détection d'objets dans les images, explorant à la fois les fondements théoriques et les applications pratiques de ces méthodes. Nous aborderons les différents types d'images, les concepts fondamentaux tels que la résolution et la couleur, ainsi que les techniques de traitement d'images telles que la segmentation, la détection de contours et le filtrage. Ensuite, nous plongerons dans la théorie de la détection d'objets, en examinant les différents algorithmes et techniques utilisés pour localiser et identifier des objets dans les images, y compris les approches traditionnelles et celles basées sur l'apprentissage en profondeur. Nous mettrons également en lumière les nombreuses applications pratiques de la détection d'objets dans divers domaines, et nous décrirons comment cette technologie est utilisée dans des secteurs tels que la sécurité, la surveillance vidéo, la conduite autonome, la médecine, et bien d'autres. Enfin, nous passerons à l'implémentation pratique des techniques de détection d'objets étudiées précédemment, en décrivant les environnements de développement utilisés, les jeux de données nécessaires, ainsi que les méthodes d'évaluation des performances des algorithmes implémentés. La conclusion résumera les principaux points abordés dans ce mémoire et mettra en évidence les conclusions tirées de l'analyse des techniques de détection d'objets, tout en discutant des défis et des perspectives futures de ce domaine en constante évolution, en soulignant les opportunités de recherche et de développement à venir.

Le travail réalisé se compose de quatre chapitres :

- Dans le premier chapitre "Généralités sur l'images", peut aborder les bases de la formation des images, les formats d'image courants, et peut-être également introduire des concepts tels que la résolution et la couleur.
- Le deuxième chapitre "Traitement d'image", vous pouvez explorer les techniques de traitement d'image telles que le filtrage, la segmentation et la transformation. Vous pourriez également discuter de méthodes spécifiques telles que la réduction du bruit et l'amélioration du contraste.
- Dans le troisième chapitre "Détection et suivi d'objet", pourrait se concentrer sur les

Introduction générale

méthodes et les algorithmes utilisés pour détecter et suivre les objets dans les images ou les vidéos. Cela inclurait probablement des discussions sur les descripteurs d'objet, les classificateurs, et peut-être des approches basées sur l'apprentissage automatique comme YOLO.

- Le quatrième chapitre comporte la partie conception et les résultats obtenus.
- Enfin, nous terminerons ce mémoire par une conclusion générale et des perspectives.

Chapitre I : Généralités sur l'image

Chapitre I : Généralités sur l'images

1 Introduction

Les images sont omniprésentes dans notre quotidien et jouent un rôle crucial dans diverses disciplines, allant des arts visuels à la communication, en passant par les sciences et la technologie. Une image peut capturer un moment, raconter une histoire, transmettre une émotion, ou représenter des données de manière visuelle. Dans ce chapitre, nous explorerons les fondements de ce qu'est une image, ses différentes formes, et son importance dans notre monde moderne.

2 L'image numérique

2.1 Définition

Au sens étymologique, le mot « image », découlant du latin *imago*, désigne la représentation Visuelle d'un objet par différents moyens ou supports (dessin, photographie, peinture, Sculpture, . . .) [1].

Une image peut être définie comme une fonction à deux dimensions $f(x, y)$, où x et y sont des coordonnées spatiales, et l'amplitude de f en n'importe quel point de coordonnée (x, y) représente l'intensité ou le niveau de gris en ce point. Dans le cas où les valeurs de x , y et de f sont des valeurs discrètes, l'image est appelée image numérique.

Une image numérique est composée d'un nombre fini d'éléments, chaque élément se caractérise par sa valeur et sa position. Ces éléments sont appelés Picture éléments, image éléments, cellules ou pixels. Le terme pixel est le terme le plus utilisé pour décrire les éléments d'une image numérique. Le domaine du traitement d'images se rapporte au traitement des images numériques par moyen d'ordinateur [2].

3 Les différents types d'image

- Imagerie vidéo (vision industrielle, vidéo surveillance)
- Imagerie thermique (thermographie)
- Imagerie d'éco (radar, sonar, échographe, doppler)
- Imagerie à rayons X (radiologie, scanner)
- Imagerie radioactive (tomographie)
- Imagerie **IRM** (résonance magnétique)
- Imagerie satellitaire

- Imagerie multi spectrale et imagerie couleur
- Imagerie polarimétrie [3].

3.1 L'imagerie vidéo

3.1.1 Définition d'une vidéo

La vidéo est une succession d'images animées défilant à une certaine cadence afin de créer une illusion de mouvement pour l'œil humain. Elle peut être analogique (signal continu d'intensité de luminance) ou numérique (suite de trames ou images) [4].

3.1.2 Résolution vidéo

La résolution vidéo est la quantité de détails contenus dans une vidéo, mesurée par le nombre de pixels contenus dans chaque image. Elle est généralement écrite comme une combinaison de la hauteur et de la largeur, comme 1080p ou 4K [5] [6]. Plus le nombre de pixels est élevé, plus la résolution est élevée et plus la vidéo est nette et claire [5] [6] [7].

3.1.2.1 Résolutions courantes

- **Résolution SD (Standard Definition)** : 720×480 pixels, utilisée pour les vidéos de qualité standard [7].
- **Résolution HD (High Definition)** : 1280×720 pixels (720p) ou 1920×1080 pixels (1080p) , utilisée pour les vidéos de haute qualité [7] .
- **Résolution Full HD (FHD)** : 1920×1080 pixels, utilisée pour les vidéos de haute qualité avec un rapport d'aspect 16 : 9 [7].
- **Résolution 2K** : 2560×1440 pixels, utilisée pour les vidéos de haute qualité avec un rapport d'aspect 16 : 9 [6].
- **Résolution 4K** : 3840×2160 pixels, utilisée pour les vidéos de haute qualité avec un rapport d'aspect 16 :9 [6].
- **Résolution 8K** : 7680×4320 pixels, utilisée pour les vidéos de haute qualité avec un rapport d'aspect 16 : 9 [6].

3.1.2.2 Balayage progressif et balayage entrelacé

- **Balayage progressif** : méthode de chargement des pixels dans chaque nouvelle image, où tous les pixels apparaissent sur l'écran en même temps [5].

- **Balayage entrelacé** : méthode de chargement des pixels dans chaque nouvelle image, où les pixels sont chargés en lignes en alternance, ce qui peut provoquer un effet de scintillement ou des lignes floues lors de mouvements rapides [5].

3.1.2.3 Sélection de la résolution vidéo

- **Choix de la résolution vidéo** : dépend du type de mouvement, de la qualité souhaitée, et du format vidéo [6] [5].
- **Résolution vidéo pour le montage vidéo** : dépend du type de montage, de la qualité souhaitée, et du format vidéo [5] [6].

La résolution vidéo est la quantité de détails contenus dans une vidéo, mesurée par le nombre de pixels contenus dans chaque image. Les résolutions courantes incluent la SD, HD, Full HD, 2K, 4K et 8K. Le choix de la résolution vidéo dépend du type de mouvement, de la qualité souhaitée, et du format vidéo.

3.1.3 Résolution temporelle d'une vidéo

La résolution temporelle d'une vidéo est définie par le nombre d'images défilant par seconde. Afin d'éviter les désagréments dus aux papillotements, et prenant en considération les spécificités de l'œil humain qui garde une image environ 10 micro secondes au niveau des rétines.

Les images doivent défiler à une certaine cadence : Environ 24 *images/s* pour un film de cinéma. 25 *images/s* pour la télévision européenne [4].

3.1.4 Définition d'un objet vidéo

Les objets physiques sont les objets du monde réel qui apparaissent dans les scènes observées par les caméras. Les objets physiques sont divisés en deux types : les objets de contexte et les objets mobiles [8] .

a. Les objets de contexte

Sont des objets physiques qui sont habituellement statiques (p. ex. les murs). Dans le cas où ils ne sont pas statiques, leurs mouvements peuvent être prédits par les informations contextuelles p.ex. Les chaises, les portes sont des objets de contexte [9].

b. Les objets mobiles

Sont des objets physiques qui peuvent être perçus dans les scènes par leurs mouvements. Il est cependant difficile de prédire leurs mouvements par exemple. Les personnes, les véhicules. Les objets dans les vidéos sont des objets mobiles. Ils sont habituellement détectés et suivis dans plusieurs frames (Image composants la vidéo) [9].

3.1.5 Séquence d'image

Une séquence d'images est une succession d'images bidimensionnelles qui montre l'évolution temporelle d'une scène. La cadence est de 25 images par seconde, ce qui correspond au seuil à partir duquel l'œil humain perçoit la séquence comme un stimulus continu, grâce à la persistance rétinienne. Par la suite, nous appellerons « trame » ou « plan » chaque image bidimensionnelle correspondant à un instant donné de la séquence [4].

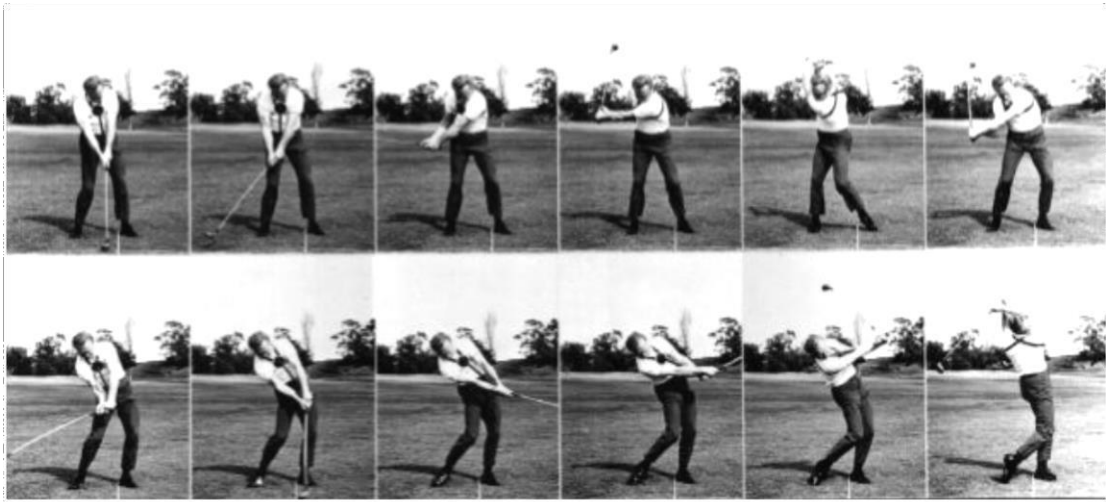


Figure I. 1: Exemple d'une séquence d'images.

3.1.6 Nature de la scène

Selon l'objet présenté et la position de la caméra, le contenu de la séquence vidéo peut changer fortement. Dans les méthodes de détection de mouvement, les objets sont caractérisés par le fait que ce sont les seules parties de la vidéo qui ont un mouvement important. Il s'agit d'abord de distinguer les séquences à caméra fixe et les séquences à caméra mobile [10].

a) Séquences à caméra fixe (Scène statique)

La grande majorité des systèmes telle que les systèmes de vidéosurveillance utilisent des caméras fixes par rapport à la scène filmée. L'image issue des caméras est alors

relativement stable, dans le sens où les changements temporels visibles dans l'image correspondent généralement à des objets d'intérêt [10].

b) Séquences à caméra mobile (Scène dynamique)

Dans le cas des systèmes tel que le cas d'un système embarqué dans un véhicule La mobilité de ce dernier fait perdre à la scène sa stabilité, même lorsque la caméra est fixée. Nous ne pouvons donc pas faire l'hypothèse d'une scène statique et les variations présentes sont très difficiles à modéliser. Les variations temporelles se produisant dans une séquence vidéo. Le mouvement dominant de la scène appliquée à tous les pixels est considéré comme étant le mouvement de la caméra et les pixels ne suivant pas ce mouvement global forment les objets en mouvement [10].

c) Mono et multi-caméra

Des méthodes de suivi existent dans le cas d'une seule caméra ou de plusieurs caméras. La différence essentielle entre ces deux méthodes provient du fait que les images prises de plusieurs caméras doivent se projeter dans le même espace, ce qui implique la calibration des caméras. Cela n'est pas nécessaire dans le cas du suivi mono caméra. La plupart des méthodes de suivi du corps en mouvement sont basées sur des séquences mono caméras. En ce qui concerne l'avantage à utiliser plusieurs caméras est de réduire les ambiguïtés causées par les problèmes d'occultations [10].

4 Les formats d'image

Il existe deux types d'images en informatique. Les images vectorielles et les images matricielle (Bitmap).

4.1 Image numérique matricielle (Bitmap)

Une image matricielle (ou image en mode point, en anglais **bitmap** ou **raster**) qui constituée d'un ensemble de points (les pixels), est celle que nous utilisons généralement pour restituer des photos numériques. Elle repose sur une grille de plusieurs pixels formant une image avec une **définition** bien précise. Lorsqu'on les agrandit trop, on perd de la qualité (pixellisation) [3]. (Voir figure 1).

Format d'images matricielles : BMP, PCX, GIF, JPEG, TIFF.

Avantage et inconvénient

Les images Bitmap offrent des avantages tels que leur facilité d'utilisation et leur haute résolution, mais elles ont également des inconvénients, tels que leur taille de fichier volumineuse et leur capacité limitée à être agrandies sans perte de qualité [11].

4.2 Image numérique Vectorielle

Les images vectorielles sont composées de formes géométriques qui vont pouvoir être décrites d'un point de vue mathématique. Par exemple une droite sera définie par 2 points, un cercle par un centre et un rayon (voir figure 1). Le processeur est chargé de "traduire" ces formes en informations interprétables par la carte graphique (images Word, Publisher, CorelDraw format WMF, EMF, CGM, EPS etc.) [12].

Avantages et inconvénients

Comme avantage, les fichiers qui composent l'image vectorielle sont petits, les redimensionnements sont faciles sans perte de qualité. Par contre comme inconvénient, ce type d'image ne permet de représenter que des formes simples. Elle n'est pas donc utilisable pour la photographie notamment pour obtenir des photos réalistes [13].

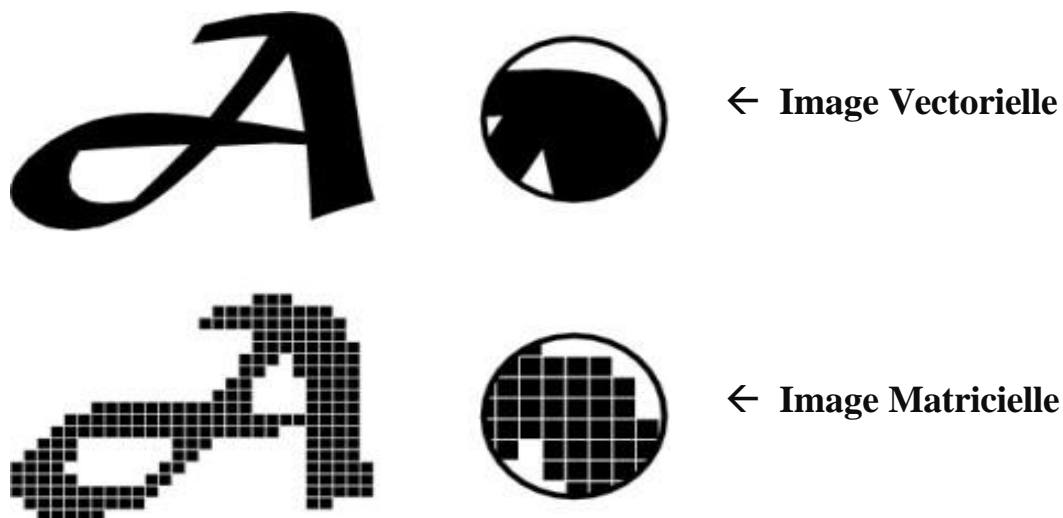


Figure I. 2: Images matricielle et vectorielle

5 Les caractéristiques de l'image

L'image est un ensemble structuré d'informations, elle est caractérisée par un ensemble de paramètres décrit ci-dessous.

5.1 Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image [14]. La taille d'une image numérique peut se définir par :

1. Sa définition en pixels.
2. Ses dimensions en pouces.
3. Sa résolution en dpi ou ppp.

Ces 3 informations sont liées. Si vous en connaissez 2, vous pouvez calculer la troisième. Les formules :

$$\text{Résolution} = \text{définition} / \text{dimension}$$

$$\text{Définition} = \text{résolution} \times \text{dimension}$$

$$\text{Dimension} = \text{résolution} / \text{définition}$$

5.2 Pixel

Nous avons abordé la notion de pixel à plusieurs reprises, car il s'agit du composant élémentaire de l'image, ce terme est une abréviation de PICTURE Élément. Il représente le plus petit élément constitutif d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions (matrice) constituant l'image, et chaque pixel à sa propre couleur (valeur) [15].

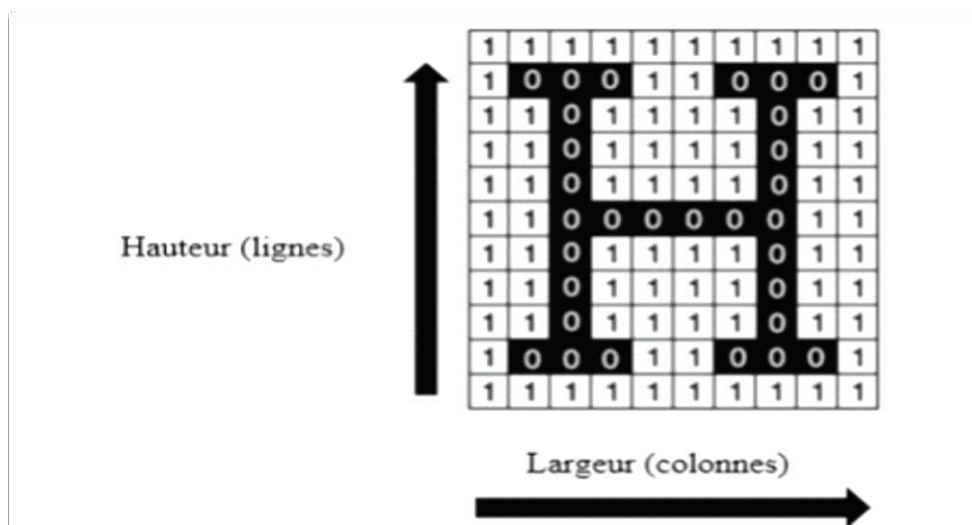


Figure I.3: Distribution des pixels par lignes et colonnes

5.3 Résolution et définition

Les termes résolution et définition sont liés à la notion de pixel, ils définissent la netteté et la qualité d'une image.

La définition est le nombre de pixels que comporte une image numérique en largeur et en hauteur. Une image dont la définition est 1500×1200 correspond à une image de 1500 *pixels* en largeur et 1200 *pixels* en hauteur.

La résolution d'une image est le nombre de pixels contenu dans l'image par unité de longueur, elle s'exprime le plus souvent en PPP (Point Par Pouce) ou dpi (dots per Inch), et parfois en points par cm. Une résolution de 400 dpi signifie donc 400 colonnes et 400 rangées de pixels sur un pouce carré ce qui donne donc 160.000 pixels sur un pouce carré. Sachant qu'un pouce est égal à 2,54 cm [14] .

Notons toutefois que la résolution peut être calculée via la formule suivant :

$$\text{Résolution} = \frac{\text{définition}}{\text{dimension}} \quad (1)$$

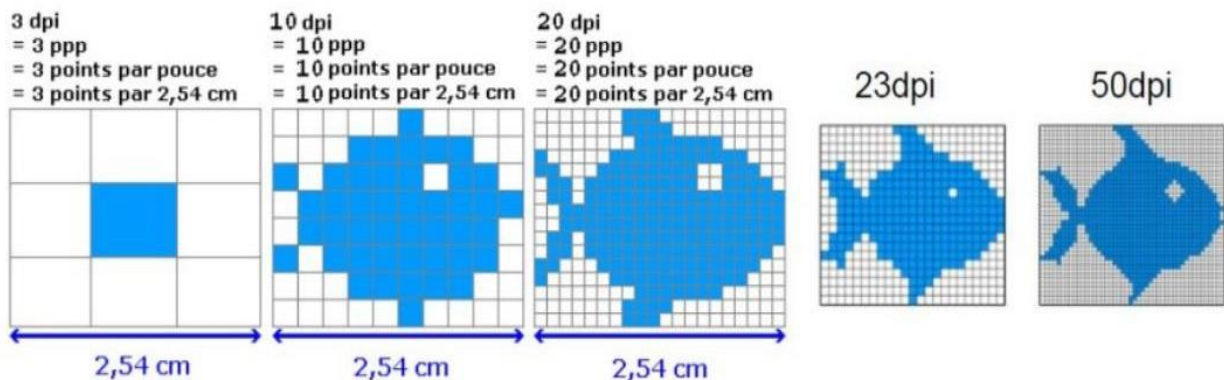


Figure I.4: Explication de résolution d'une image

5.4 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 L_1 et L_2 sont les degrés de luminosité respectivement de deux zones voisines A_1 et A_2 d'une image, le contraste c est défini par le rapport : $c = \frac{L_1 - L_2}{L_1 + L_2}$ [16].

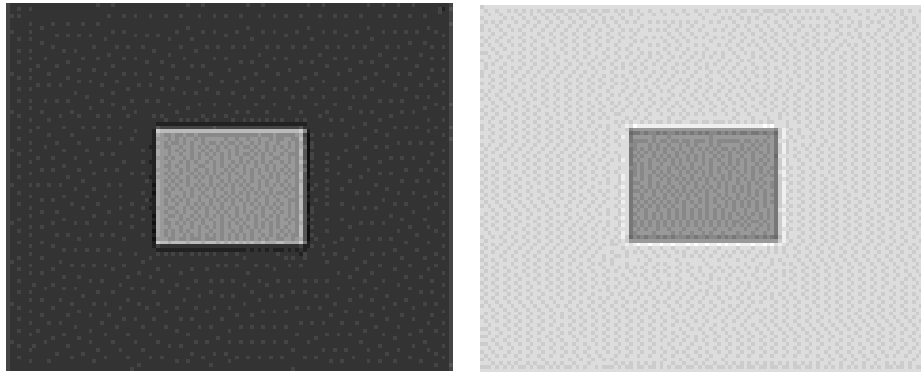


Figure I. 5 : Contraste de luminosité

5.5 Contour

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent différence significative [16].

5.6 Bruit

En traitement d'image, toute brusque fluctuation d'un pixel par rapport à ses voisins étant considéré comme un bruit provoquant ainsi la dégradation de la qualité d'image que ce soit de luminance ou de contraste [4].



Figure I.6 : Exemple bruit de dans une image

5.7 Histogramme

L'histogramme est un vecteur dont la dimension est égale à 256 (dans le cas d'une image à niveau de gris). Chaque élément de ce vecteur (noté $h(i)$) représente le nombre de pixels de l'image possédant le niveau de gris i (sa fréquence d'apparition), on peut donc assimiler l'histogramme à la densité de probabilité des intensités lumineuses. Puisque le nombre des pixels est généralement assez grand, on peut alors normaliser l'effectif de chaque niveau de gris en divisant chaque terme du tableau par la surface du plan exprimée en nombre

total des pixels de l'image. De ce fait, chaque case $h(i)$ du vecteur représente la probabilité d'avoir l'intensité i . Il peut être utilisé pour améliorer la qualité d'image (rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci [17].

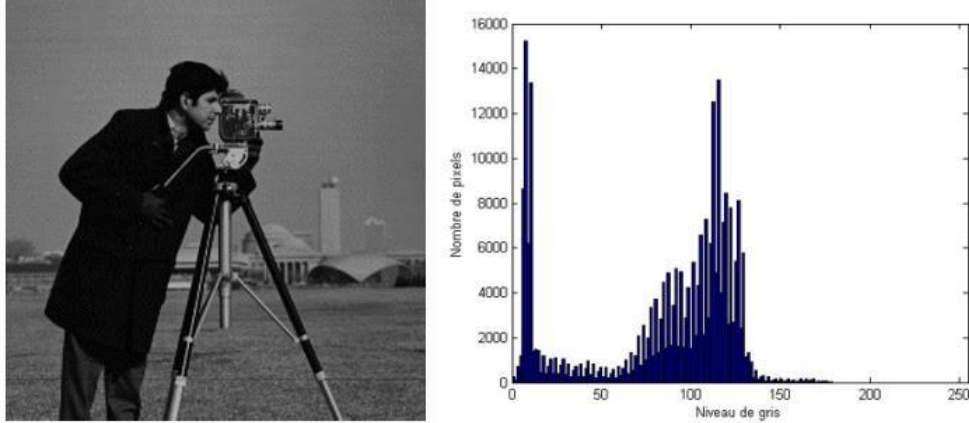


Figure I.7: Exemple de l'histogramme d'une image

5.8 Codage

Une image matricielle est représentée sous forme d'une matrice (d'où son nom d'ailleurs), où ses éléments représentent les couleurs des pixels (voir section 2.2.1). Il existe différents codages possibles, qui dépendent du nombre de couleurs, cette dernière qui détermine à son tour le de nombre de bits utilisés pour stocker cette information. Nous allons voir dans ce qui suit, les différents types de codage existants pour cette catégorie d'image.

5.8.1 Images binaires

Une image binaire est une image dont chaque pixel est codé sur un bit, donc on ne peut avoir pour valeur que 0 qui représente la couleur noir et 1 qui représente la couleur blanche, il s'agit d'une image en noir et blanc (voir Figure I.8) [2].

1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	0	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1
1	0	1	1	1	1	1	1	0	1
1	0	1	0	1	1	0	1	0	1
1	0	1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0	1	1
1	1	1	0	0	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1

Figure I.8: Codage binaire (0,1)

5.8.2 Images en niveaux de gris

A chaque pixel codé en n bits est affecté un nombre binaire variant de « 0 » (pour le noir) à « $2^n - 1$ » (pour le blanc), avec n le nombre de bits pour chaque pixel. Il y aura alors « 2^n » niveaux de gris. [15]

- Si le codage se fait en 8 bits par pixel, on aura $2^n = 2^8$, c'est-à-dire 256 niveaux de gris allant du blanc au noir ;
- Si le codage se fait en 16 bits par pixel, on obtient $2^n = 2^{16}$, valant 65536 niveaux de gris, allant du blanc au noir.

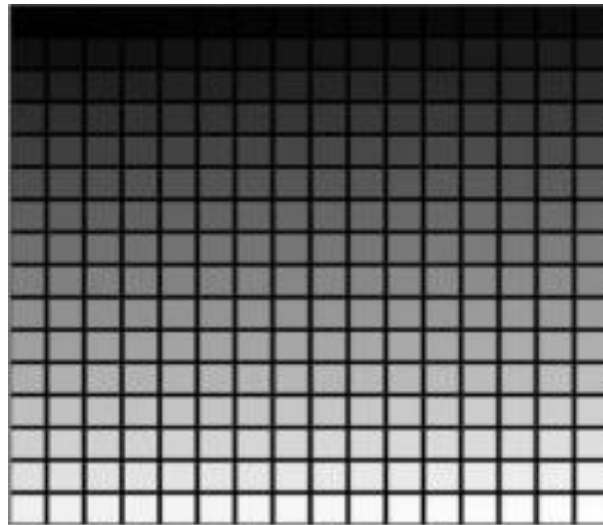


Figure I.9: 256 nuances de gris

5.8.3 Images en couleurs

Il existe plusieurs façons de coder les couleurs parmi, nous citons le système RVB, qui est composée de trois paramètres d'intensité, le rouge, le vert et le bleu. Chaque paramètre est codé souvent sur 8 bits dont la variation d'intensité est comprise entre 0 et 255. La couleur finale est obtenue par synthèse additive de ces trois paramètres, chacun d'eux étant une couleur de base [2].

Ainsi Le système de couleurs CMJN est largement utilisé dans l'industrie de l'impression. C'est un système soustractif qui utilise les couleurs Cyan, Magenta, Jaune et Noir pour créer une gamme de couleurs. Dans ce système, les couleurs primaires sont combinées pour créer des couleurs secondaires, qui à leur tour sont combinées pour créer des couleurs tertiaires. Bien que ce système soit populaire dans l'industrie de l'impression, il est important de comprendre ses avantages et ses limites pour en tirer le meilleur parti [18].

5.8.4 Images en couleurs indexées

Ce type de codage, permet d'obtenir jusqu'à 256 couleurs fixes, définies à l'avance dans une palette. Il n'utilise qu'une seule couche [13].

- Codage en 8 bits par pixel : $2^8 = 256$ possibilités

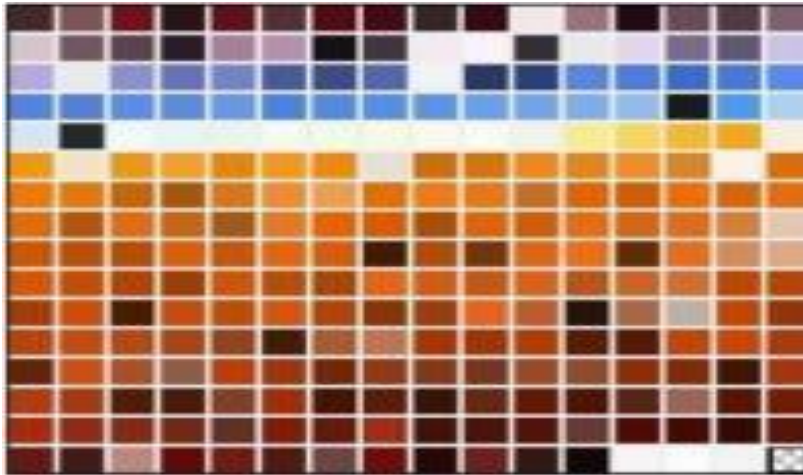


Figure I.10: Palette de 256 couleurs utilisées



Figure I.11: Image codée en couleurs indexées

5.8.5 Le système HSV (Hue, Saturation, Value) ou TSL (Teinte, Saturation, Luminosité)

Le système de codage des couleurs HSV (Teinte, Saturation, Luminosité) ou TSL (Teinte, Saturation, Luminosité) offre une manière intuitive de représenter les couleurs en séparant leurs aspects fondamentaux. Voici une description détaillée du codage dans ce système :

1. Teinte (Hue) : La teinte représente la couleur dominante de la manière suivante :

- 0° correspond au rouge.

- 120° correspond au vert.
- 240° correspond au bleu.
- 360° correspond à nouveau au rouge, marquant ainsi un retour au début du spectre des couleurs.

2. Saturation : La saturation indique l'intensité de la teinte et peut varier de 0° à 100° :

- 0° correspond à une déclinaison de gris allant du blanc au noir selon la luminosité.
- 100° représente l'intensité maximale de la teinte, indiquant une couleur pure et saturée.

3. Luminosité (Value ou Lightness) : La luminosité exprime la quantité de couleur noire ou Blanche mélangée à la teinte, et est également exprimée en pourcentage :

- 0% représente le noir pur.
- 100% représente le blanc pur.
- Les valeurs intermédiaires indiquent la quantité de lumière présente dans la couleur, avec 50% généralement considéré comme une luminosité neutre.

Ces trois composantes (Teinte, Saturation, Luminosité) forment un système de coordonnées Cylindriques dans l'espace des couleurs, offrant une représentation complète et intuitive des couleurs.

Concernant le codage, les valeurs de teinte sont généralement exprimées en degrés (0° à 360°), tandis que la saturation et la luminosité sont souvent exprimées en pourcentage (0% à 100%). Les conversions entre les espaces de couleurs, tels que RGB (Rouge, Vert, Bleu) et HSV, peuvent être effectuées à l'aide de calculs mathématiques, mais des outils automatisés sont également disponibles en ligne.

6 Conclusion

Dans ce chapitre, nous avons établi une base fondamentale pour comprendre l'image. Nous avons commencé par définir ce qu'est une image et exploré divers types, tels que l'imagerie vidéo. Les formats d'image ont été abordés, soulignant l'importance du dimensionnement, de la luminance et de la résolution pour la qualité visuelle.

Les concepts de Trans phase et de contour, montrant leur rôle dans l'amélioration et la détection des bords des images, ont été également traités. Les discussions sur les niveaux de gris et l'imagerie en couleur, notamment les modèles comme RGB, ont approfondi notre compréhension de la représentation visuelle. De plus, le chapitre a traité du bruit, qui peut

altérer la qualité des images, et de l'histogramme, un outil clé pour analyser la distribution des intensités dans une image.

En somme, ce chapitre fournit une vue d'ensemble complète des éléments essentiels de l'imagerie numérique, offrant une base solide pour toute analyse ou application future dans ce domaine.

Chapitre II : Traitement d'image

Chapitre II : Traitement d'image

1 Introduction

Dans notre ère numérique, le traitement d'images est essentiel pour comprendre, analyser et modifier visuellement les données. Il englobe une gamme de techniques, notamment le prétraitement pour améliorer la qualité de l'image, la segmentation pour diviser l'image en régions significatives, et la détection de contours pour identifier les limites des objets. Le filtrage est également utilisé pour modifier les caractéristiques de l'image, tandis que des techniques telles que la binarisation, la dilatation et l'érosion permettent d'effectuer des opérations spécifiques sur les pixels.

En combinant ces techniques, le traitement d'images offre un moyen puissant d'analyser, interpréter et utiliser les données visuelles de manière efficace

2 Traitement d'images

Le traitement d'images est l'ensemble des opérations effectuées sur l'image, afin d'en améliorer la lisibilité et d'en faciliter son interprétation. C'est, par exemple, le cas des opérations de rehaussement de contraste, élimination du bruit et correction d'un flou. C'est aussi l'ensemble d'opérations effectuées pour extraire des "informations" de l'image comme la segmentation et l'extraction de contour. L'analyse d'une image cherche à passer d'une grande quantité d'informations de bas niveau (i.e. les pixels) à une description symbolique de haut niveau pouvant aider à obtenir une interprétation structurelle de cette image. La distinction entre traitement de bas ou de haut niveau est liée au contenu sémantique des entités traitées et extraites de l'image. Les traitements de bas niveau opèrent en général sur les grandeurs calculées à partir des valeurs attachées à chaque point de l'image sans faire nécessairement la liaison avec la réalité qu'elles représentent. Par exemple, la détection des contours est un traitement de bas niveau qui est effectué « sans comprendre » l'image. Le contour détecté peut très bien ne pas correspondre à un bord d'objet dans la réalité et ne présente aucun intérêt. Ainsi les traitements de bas niveau opèrent plutôt sur des données de nature numérique. À l'opposé, les traitements de haut niveau s'appliquent à des entités de nature symbolique associées à une représentation de la réalité extraite de l'image. Ils sont relatifs à l'interprétation et à la compréhension de l'image et sont exprimés avec des mots du vocabulaire de l'application. Par exemple, des zones d'une image aérienne peuvent être caractérisées par leur forme (rectangulaire, linéique, etc.), être étiquetées avec les termes : bâtiment, route, bosquet, ombre, etc. Voilà (Figure II.1) de traitement d'image [19] :

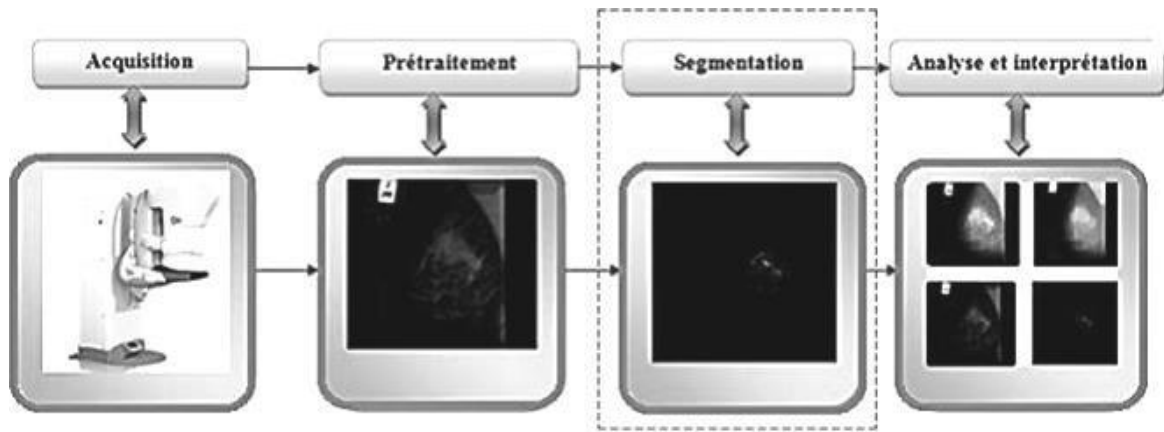


Figure II.1: Chaîne de traitement d'images

Le traitement d'images comprend un certain nombre d'opérations qui parfois doivent être appliquées dans un certain ordre, parmi nous citons [19]:

- Amélioration : augmenter la qualité de la perception visuelle qu'on a d'une image.
- Restauration : compenser les dégradations (bruit, flou, ...).
- Compression : stocker et transférer efficacement les données.
- Segmentation : limiter les "objets".
- Reconstruction 3D : obtenir un volume à partir de plans (images 2D).
- Représentation (modélisation) :
 - Bas niveau : texture, couleur, forme, etc.
 - Haut niveau : caractéristiques (features), apprentissages statistiques, et graphes.
- Analyse : convertir en connaissances.
- Reconnaissance / compréhension et interprétation : identifier le contenu.

3 Prétraitement d'image

L'image brute est une structure simple dans son organisation mais complexe dans son contenu, ceci est dû à plusieurs facteurs, d'une part la grande quantité d'informations qui y réside, d'autre part le nombre important de facteurs indépendants de l'utilisateur qui tendent à y introduire des distorsions indésirables. Le rôle du prétraitement dans sa définition la plus générale est de remédier aux dégradations ayant affecté l'image et/ou de rendre cette image mieux adaptée à une application particulière [20].

3.1 Filtrage

Selon le critère à considérer on peut avoir plusieurs taxinomies, en prenant l'aspect fréquentiel on parle de filtre passe haut (laisse passer les hautes fréquences) et de filtre passe bas (laisse passer les basses fréquences). En considérant les tensions d'entrée et de sortie du signal fréquentiel, on pourra scinder les filtres en deux grandes catégories, les filtres linéaires et les filtres non-linéaires.

3.1.1 Filtres linéaires

Les premières et les plus simples méthodes de filtrage sont basées sur le filtrage linéaire, chacun de ses opérateurs est caractérisé par sa réponse impulsionnelle $h(x, y)$. Le filtrage linéaire est un produit de convolution c'est-à-dire une combinaison linéaire du voisinage du pixel concerné. Nous présentons dans ce qui suit, les principaux types de filtres linéaires [21] :

3.1.2 Filtre moyenné (lissage)

L'intensité du pixel considéré est remplacée par la moyenne des pixels de son voisinage, la taille de la zone (fenêtre) qui entoure le pixel est un paramètre important, plus cette dimension est grande, plus sa sensibilité au bruit diminue, et le lissage devient important (le flou s'accroît). Le filtre moyen est un filtre passe-bas, il laisse passer les basses fréquences (les faibles changements d'intensité de l'image) et atténue les hautes fréquences (variations rapides) [21] .

1	1	1
1	1	1
1	1	1

 $\times 1/9$



(A)



(B)

Figure II.2: Application de filtre moyenné (lissage) (A) image originale (B) après lissage

3.1.3 Filtre gaussien

Il s'agit aussi d'un filtre passe bas, dont la réponse impulsionnelle est une fonction Gaussienne exprimée comme suit :

$$G_0(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

L'avantage du filtre gaussien est de faciliter le contrôle du degré de filtrage grâce au paramètre σ . Ce type de filtrage priorise grandement les pixels rapprochés du pixel central, et priorise proportionnellement moins les pixels éloignés du pixel central. Cependant, comme le filtre moyenné, le filtre gaussien a le désavantage de dégrader les contours [21].

1	2	1
2	4	2
1	2	1

x1/16

3.1.4 Filtre rehausseur de contours

Les filtres rehausseurs de contours sont des filtres passe haut, ils mettent donc en évidence les changements rapides de l'intensité de l'image (changement qui arrive en général au niveau des contours) et laisse les zones uniformes inchangées (généralement les zones entre les contours ayant de basse fréquence) [21].

-1	-1	-1
-1	9	-1
-1	-1	-1

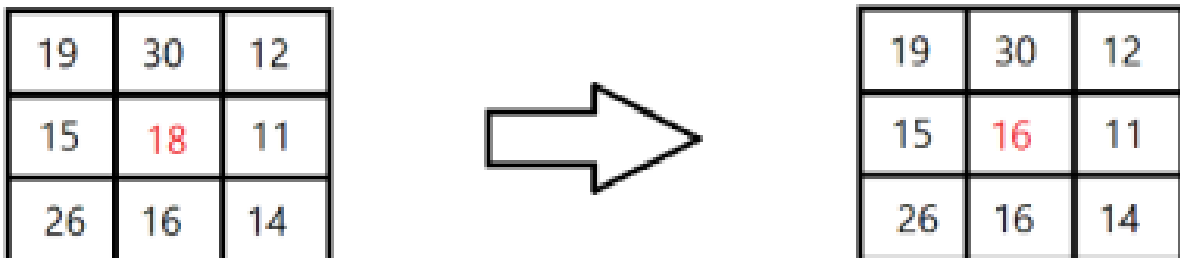


Figure II.3: Exemple d'utilisation du filtre rehausseur de contours

3.1.5 Filtres non linéaires

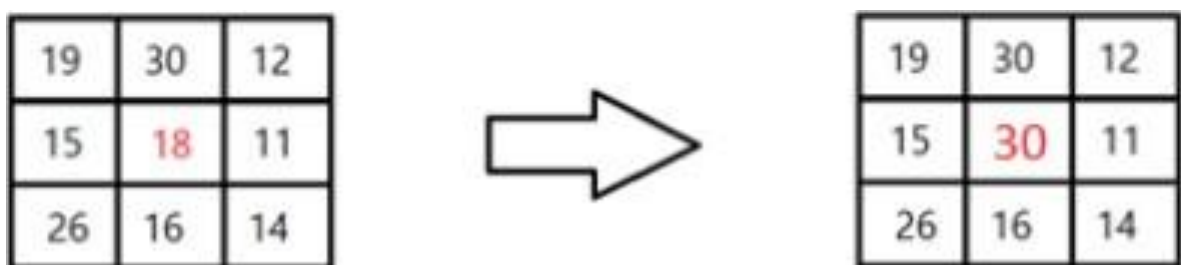
Un filtre non linéaire est un opérateur qui remplace la valeur de chaque pixel par une combinaison non linéaire des valeurs des pixels de ses voisins. Le but des filtres non linéaires est de fournir une solution en employant une (ou plusieurs) des stratégies suivantes :

- Statistiques d'ordre : Opérateurs basés sur une classification des valeurs de pixels voisins (minimum, maximum, médiane (filtre médian, etc.)).
- Moyennes robustes : calcul des moyennes locales pondérées en excluant les valeurs marginales.
- Anisotropie : opérateurs où le support de calcul du filtre effectivement calculé varie localement en fonction du contenu de l'image. Il existe plusieurs filtres non linéaires qui utilisent une ou plusieurs de ces propriétés [22].



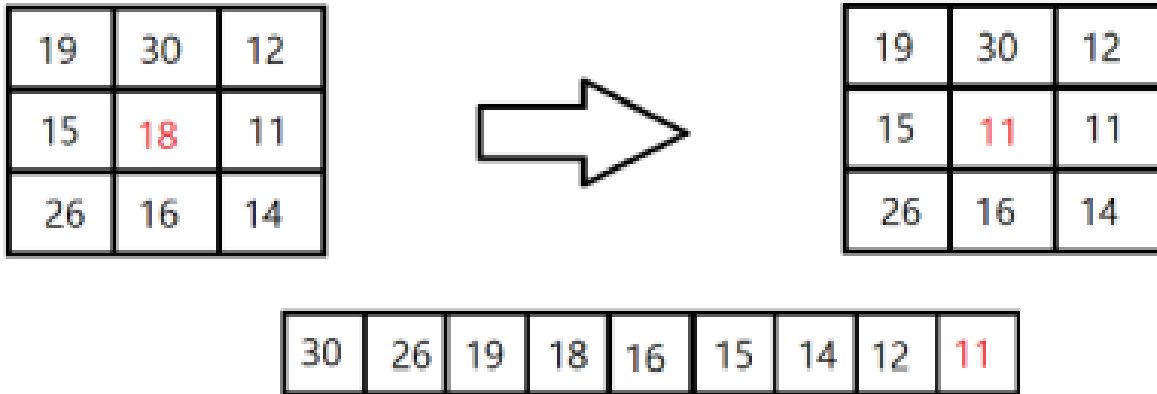
30	26	19	18	16	15	14	11	12
----	----	----	----	----	----	----	----	----

(A) : Filtre médiane



30	26	19	18	16	15	14	11	12
----	----	----	----	----	----	----	----	----

(B) : Filtre maximum



(C) : Filtre minimum

3.2 Binarisation

La binarisation (le seuillage) est la technique de classification la plus simple, où les pixels de l'image sont partagés par un seul seuil S en deux classes ; ceux qui appartiennent au fond et ceux qui appartiennent à la scène (l'objet). L'image est alors séparée en deux classes de façon à ce que l'information comprise entre 0 et S est retenue et l'autre non, ou vice-versa [23].

Soit l'image $(M \times N)$, supposons que $f(x, y)$ représente le niveau de gris du pixel aux coordonnées (x, y) , $0 \leq x < M$, $0 \leq y < N$, et S est le seuil choisi, les pixels de l'objet (segment recherché) sont ceux ayant le niveau de gris inférieur à S et les autres ayant le niveau de gris

$$g(x) = \begin{cases} 0 & \text{si } f(x, y) < s \\ 255 & \text{si } f(x, y) \geq s \end{cases} \quad (\text{II-1})$$



(A)

(B)

Figure II.4: Binarisation d'une image : (A) image originale, (B) image binarisée.

3.3 Les opérateurs de la morphologie mathématique

3.3.1 Erosion

L'érosion constitue un des opérateurs de base de la morphologie mathématique. Elle est une transformation en tout ou rien relative à l'inclusion, notée par $X \ominus B$.

En effet, l'élément structurant B , repéré par son centre, est déplacé pour occuper successivement toutes les positions dans X (représentant une forme ou un objet de l'image initiale). Pour chaque position, on vérifie si B est complètement inclus dans X . Mathématiquement, l'érodé de X par l'élément structurant S est l'ensemble des pixels x pour lesquels S , translaté par x , est inclus dans X [24].

$$X \ominus B = \{X/B^* \subset X\} \quad (\text{II-2})$$

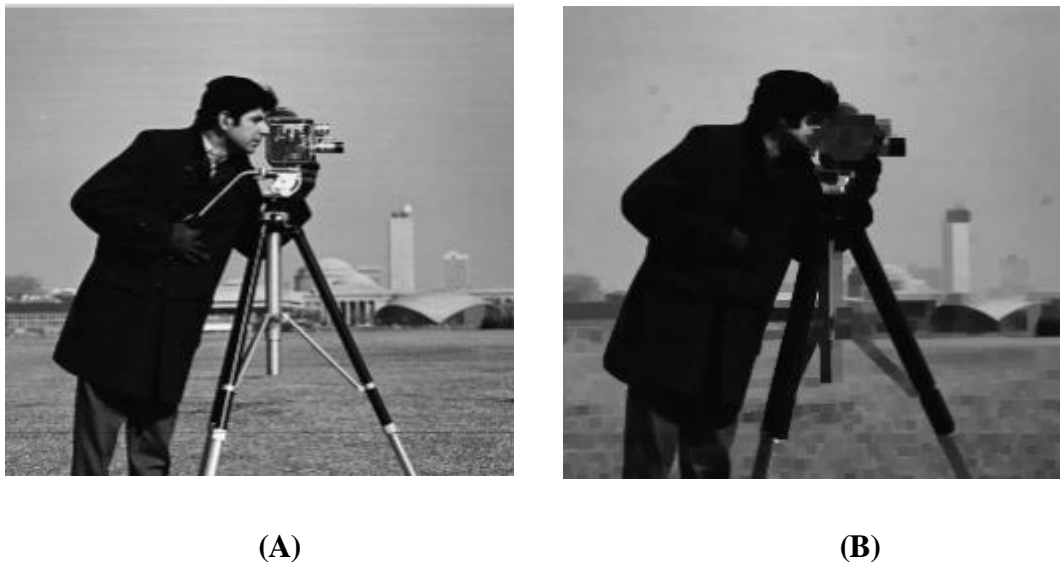


Figure II.5: Erosion d'une image (A) image originale (B) image érodée

Propriétés de l'érosion : [24]

- La taille des formes (objets) réduire.
- Une forme (objet) avec des concavités ou des trous peut être divisés.
- Les petites formes (objets) et les détails disparaissent.

On remarque sur l'image (10), que la taille du visage, la caméra est réduite. Les détails tels que les boutons du manteau, les doigts ont disparu.

3.3.2 Dilatation :

La dilatation constitue aussi un opérateur de base de la morphologie mathématique. Elle est une transformation en tout ou rien basée sur l'intersection, notée par $X \oplus B$. L'élément structurant B , repéré par son centre, est déplacé pour occuper successivement toutes les

positions dans X . Pour chaque position, on vérifie si B intersecté (a au moins un pixel en commun) X . Mathématiquement, le dilaté de X par B est l'ensemble des pixels x pour lesquels B , translaté par x , intersecté avec X [24].

$$X \oplus B = \left\{ \frac{x}{B} \times \cap X \neq \emptyset \right\} \quad (\text{II-3})$$

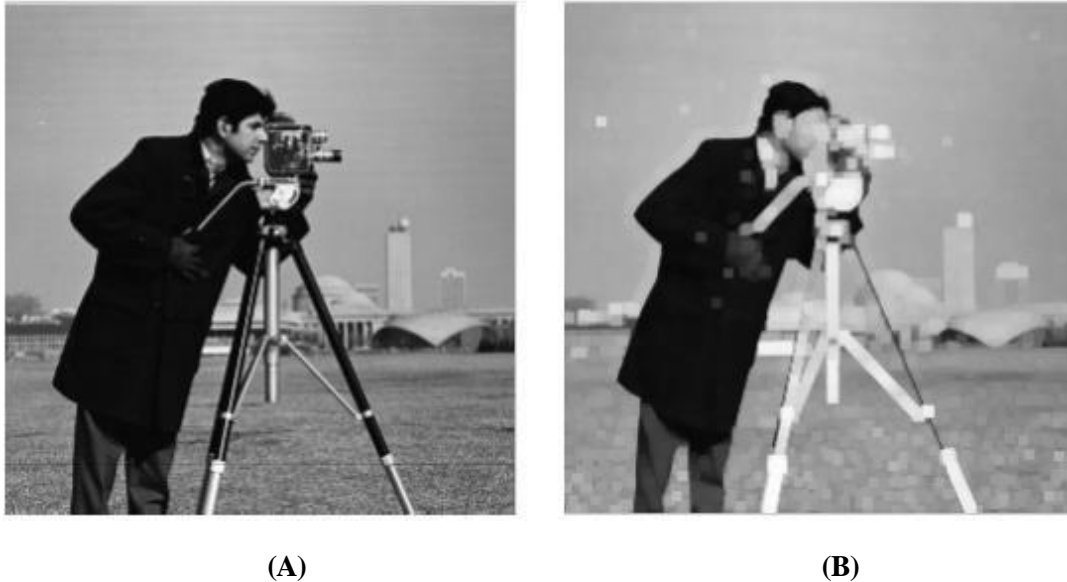


Figure II.6: Dilatation d'une image (A) image originale (B) image dilatée

• **Propriétés de la dilatation :** [24]

- La taille des formes (objets) augmente.
- Les trous et les concavités peuvent être bouchés.
- Les formes (objets) voisines peuvent se connecter.
- Les petits détails disparaissent.

On remarque sur l'image (11), que la taille du visage, de la caméra a augmentée, les trous (fenêtres) qui apparaissent dans l'immeuble an arrière-plan ont été bouchés. La caméra ainsi que son pied sont connectées.

3.3.3 Utilisation de l'érosion et de la dilatation :

La dilatation et l'érosion sont des transformations adjointes et complémentaires. En effet, l'érosion est utilisée pour : agrandir des trous, augmenter les concavités dans les formes, et déconnecter les objets en contact. La dilatation quant à elle, est utilisée pour : boucher les trous, réduire les concavités dans les formes et connecter deux formes voisines [24].

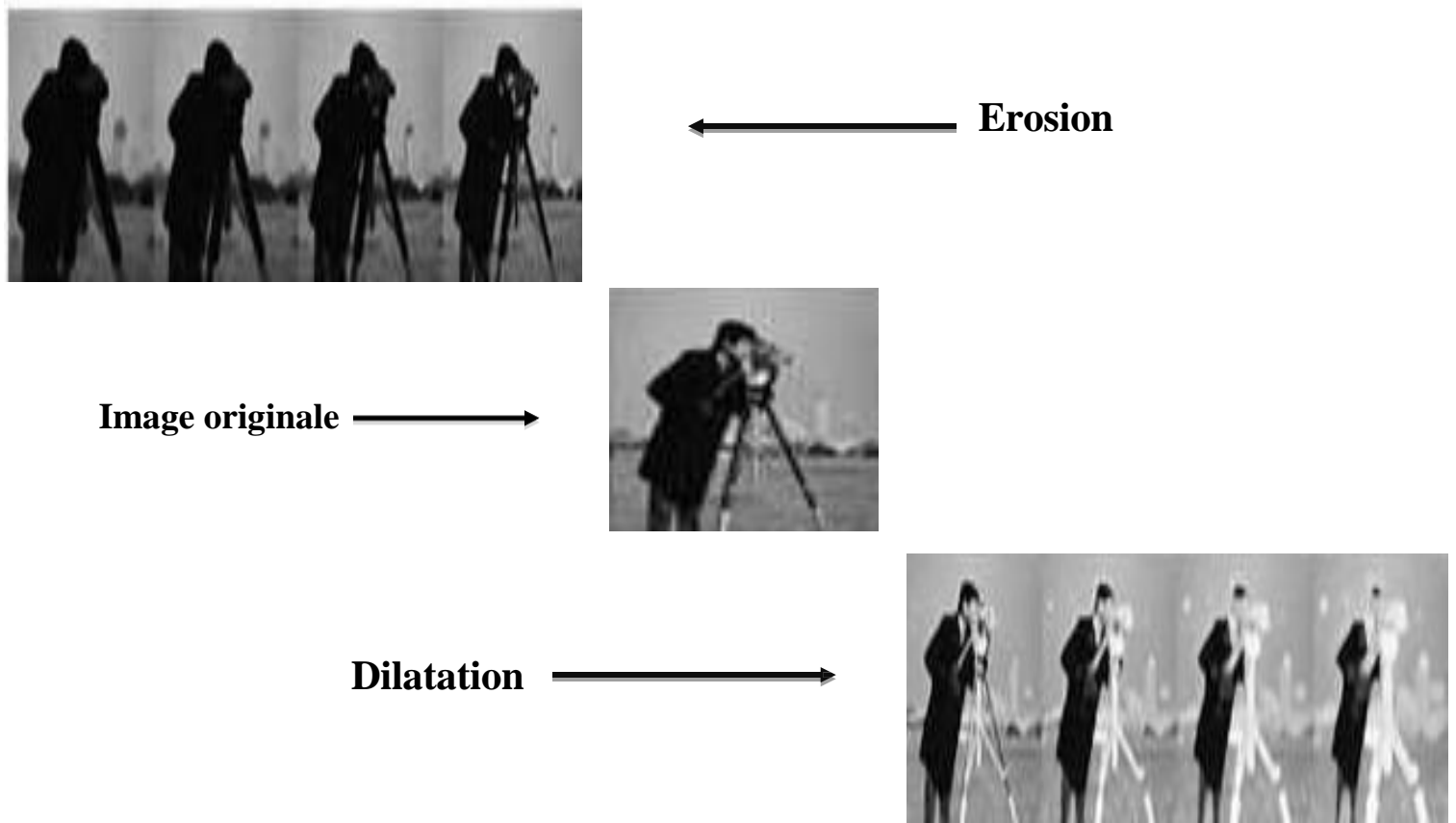


Figure II.7: Comparaison entre érosion et dilatation

3.4 Segmentation

3.4.1 Définition

La segmentation peut être définie comme étant une partition de l'image I en un nombre K de sous-ensembles \mathcal{R}_i (appelés régions) homogènes selon un ou plusieurs critères (niveau de gris, propriétés de textures, etc.) telles que chaque pixel doit être affecté à une classe et les régions doivent être disjointes. Ces conditions peuvent être exprimées mathématiquement comme suit

$$\begin{aligned}
 \bigcup_{i=1}^K \mathcal{R}_i &= I \\
 \mathcal{R}_i \cap \mathcal{R}_j &= \emptyset \quad \forall i, j \neq i
 \end{aligned}
 \tag{II-4}$$

Dans le contexte médical, les régions \mathcal{R}_i correspondent aux différentes structures anatomiques constituant les différentes régions d'intérêt. La détermination automatique du nombre de régions K est un vrai problème. Souvent, la valeur K est connue : c'est une information a priori sur les structures anatomiques en investigation.

La segmentation est une tâche particulièrement importante en traitement d'images, actuellement plusieurs jeux de données existent, ces banques d'images permettent de simuler, et comparer différentes méthodes de segmentation. Elle concerne divers domaines tels que la télédétection, médical (IRM, Radio, scanner,), l'astronomie, la biologie (image cytologique,), militaire (autodirecteur de missile, simulateur d'aviation), multimédia (TV haute définition, amélioration de qualité, compression, tatouage, remplacement des panneaux publicitaire dans les transmissions sportives, suivis d'objet, ...), industriel (contrôle qualité, robotique), etc [25].

3.4.2 Type de segmentation

En considérant l'objectif visé, trois principaux types de segmentation d'images existent à savoir la segmentation sémantique, la segmentation par instance et, la segmentation panoramique.

3.4.2.1 Segmentation Sémantique :

La segmentation sémantique associe une étiquette à chaque pixel d'une image en l'affectant à une catégorie, ainsi il est question de trouvé des catégories distinctes. La séparation d'images en deux classes est un exemple simple de segmentation sémantique. La Figure II.8 (a), illustre une image présentant une personne à la plage, cette image a été segmentée en deux classes distinctes : la personne et l'arrière-plan La Figure II.8(b) [26].

De nombreuses applications en compréhension de scène dépendent d'une classification dense au niveau pixel, notamment en détection d'objets [27].

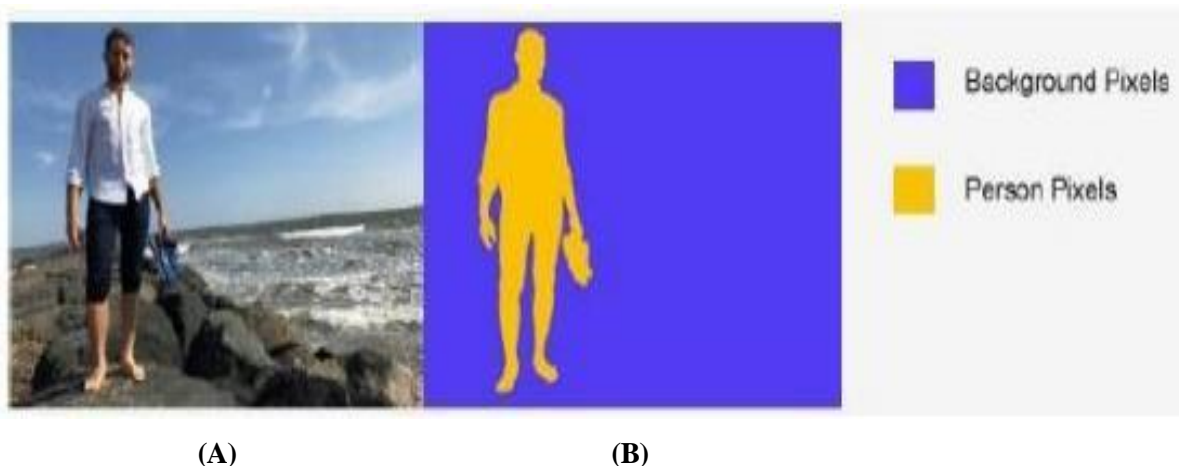


Figure II 8: Segmentation sémantique. (A) image originale, (B) image segmentée en deux classes, personne et arrière-plan.

3.4.2.2 Segmentations par instance :

Il s'agit d'une annotation combinée de la détection de cible et de la segmentation sémantique. La détection de la cible vient en premier, puis chaque pixel est étiqueté (segmentation sémantique). Une segmentation par instance est illustrée dans la (Figure II.9), ou la personne est prise comme objet cible. Par contre dans une segmentation sémantique, il n'y a pas une distinction entre les différentes instances dans la même catégorie.) [28].



Figure II.9: Segmentations par instance.

3.4.2.3 Segmentation panoramique

Une autre combinaison de segmentation sémantique et de segmentation d'instance, mais plus complexe. La segmentation panoramique signifie que toutes les cibles doivent être détectées et que différentes instances de la même catégorie doivent être distinguées

La segmentation panoramique consiste à détecter et à segmenter tous les objets de l'image, y compris l'arrière-plan, et à distinguer les différentes instances (différentes couleurs sont utilisées) [28] (Figure II.10).



Figure II.10: Segmentation panoramiques.

3.4.3 Technique de segmentation

La segmentation d'images peut être considérée comme un traitement de bas niveau, elle n'est généralement qu'une première étape essentielle dans le processus d'interprétation d'une scène. De nombreux travaux ont été réalisés sur ce sujet, dans des domaines aussi variés, c'est toujours un sujet d'actualité et un problème qui reste ouvert. Pour cette subdivision en régions distinctes homogènes, il existe deux grandes approches, l'approche région et l'approche frontière. Ces deux approches sont duales car une région définit une ligne sur son contour et une ligne fermée définit une région intérieure [29].

3.4.3.1 Méthodes du seuillage (threshold Based Methods) :

Le seuillage est la technique de segmentation la plus répandue pour extraire les objets du fond de l'image. Elle est l'opérateur qui associe à chaque point de l'image une classe de luminance. Le seuillage est réalisé après comparaison des luminances à un ou plusieurs seuils. Cette technique tente de déterminer les classes de pixels par l'analyse des histogrammes monodimensionnels, le plus souvent les seuils sont déterminés à l'aide de ces analyses localement ou globalement à l'image ou au volume. Dans le cas d'une image composée d'une ou plusieurs régions de même type, et d'un fond visuellement différent, l'histogramme est bimodal et il est possible de séparer les régions du fond par un simple seuil qui se situe sur la vallée entre les deux modes (figure II.11). Si l'image est constituée de régions ayant des caractéristiques différentes, son histogramme est multimodal (figure II.12), tels que chacun des modes correspond à une région. Le seuillage est alors défini comme le problème de trouver le minimum entre deux modes adjacents c'est-à-dire les vallées.

Les pixels ayant des niveaux de gris entre deux vallées sont alors assignés à la même région [30].

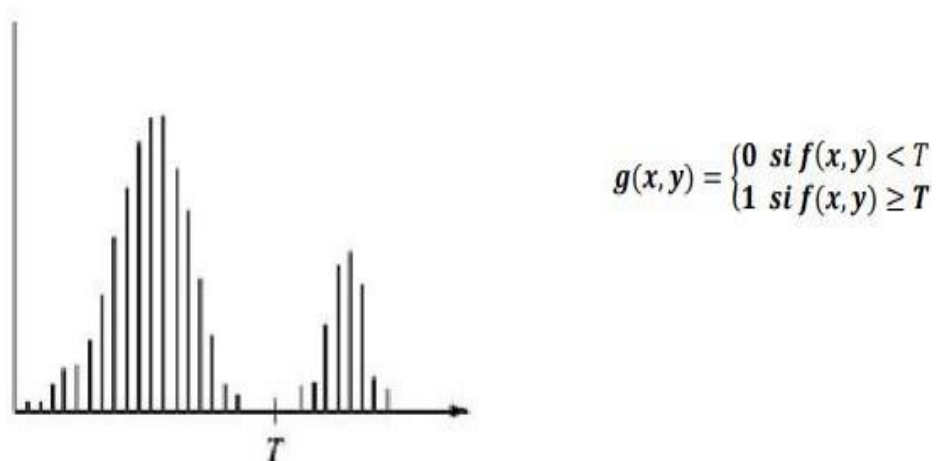


Figure II.11: Seuillage simple d'un histogramme

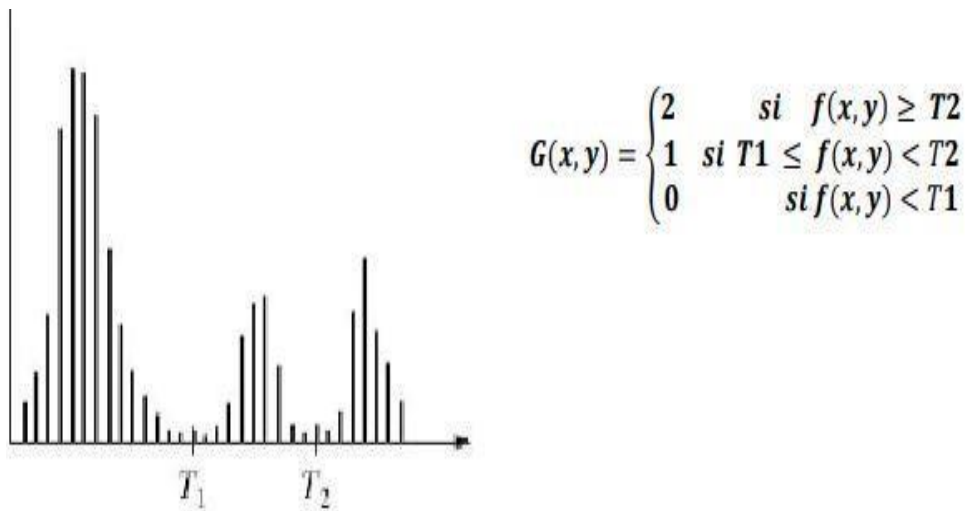


Figure II.12: Seuillage multiple d'un histogramme.

Il existe deux principaux types de seuillage [30] :

- **Seuillage global** : il est question d'un seuil pour toute l'image, il consiste à comparer le niveau de gris de chaque pixel x_i de l'image avec un seuil global fixe T .
- **Seuillage local ou adaptatif** : un seuil pour une portion de l'image. Les méthodes de seuillage local prennent en considération la valeur des pixels voisins pour le calcul des seuils.

3.4.3.1.1 Méthode Otsu

La méthode Otsu est une méthode d'optimisation de la segmentation d'images en utilisant la théorie de la statistique bayésienne. Elle a été développée par Nobuyuki Otsu en 1979 et est largement utilisée dans le domaine de l'analyse d'images.

La méthode Otsu consiste à trouver la valeur optimale de seuil pour la segmentation d'une image en utilisant la théorie de la statistique bayésienne. Elle calcule la variance de la classe de l'objet et de la classe de fond, puis utilise ces valeurs pour déterminer la valeur de seuil optimale. Cette méthode est particulièrement utile pour les images où les objets sont de tailles différentes et où les contrastes sont faibles [31] [32].

La méthode Otsu a été utilisée dans de nombreux domaines, tels que l'analyse d'images médicales, la reconnaissance d'objets, la détection de mouvements, et la segmentation d'images.

Voici quelques exemples de la méthode Otsu en action :

- **Segmentation d'images médicales** : La méthode Otsu a été utilisée pour segmenter des images médicales, telles que des images de mammographies, pour détecter les tumeurs [31].
- **Reconnaissance d'objets** : La méthode Otsu a été utilisée pour reconnaître des objets dans des images, telles que des visages, pour détecter les mouvements et les expressions faciales [31].
- **Détection de mouvements** : La méthode Otsu a été utilisée pour détecter des mouvements dans des images, telles que des vidéos, pour suivre les objets en mouvement [31].

En résumé, la méthode Otsu est une méthode d'optimisation de la segmentation d'images qui utilise la théorie de la statistique bayésienne pour trouver la valeur de seuil optimale. Elle est largement utilisée dans de nombreux domaines, tels que l'analyse d'images médicales, la reconnaissance d'objets, la détection de mouvements, et la segmentation d'images [31] [32].

3.4.3.2 Méthodes par croissance de régions (Région Growing Methods) :

Cette approche consiste à explorer l'image à partir de regroupements initiaux de pixels connexes appelés germes, et à faire croître ceux-ci sous une double contrainte d'adjacence et d'homogénéité. Les germes sont sélectionnés de manière déterministe ou aléatoire par prétraitement de l'image. De manière itérative, de nouveaux pixels sont ajoutés aux germes jusqu'à ce que la région en cours de construction ne respecte plus le ou les prédicats mis en place. La plupart de ces algorithmes parcourent l'image selon un ordre qui peut influencer le résultat de la segmentation. Le parcours le plus fréquent consiste à partir d'un germe en haut à gauche de l'image pour se diriger vers le bas à droite de l'image. L'avantage de cette approche est de préserver la forme de chaque région de l'image. Cependant, une mauvaise sélection (choix) des germes peut entraîner des phénomènes de sous-segmentation ou de sur-segmentation. Plusieurs auteurs tentent d'améliorer les résultats de ce type d'approche en se focalisant sur les germes choisis pour limiter les phénomènes de sous-segmentation ou de sur-segmentation. Pour cela, ils ont opté pour les méthodes hybrides comme l'approche contour-région, pour l'utilisation des champs de Markov et l'algorithme de la ligne de partage des eaux issu de la morphologie mathématique [33].

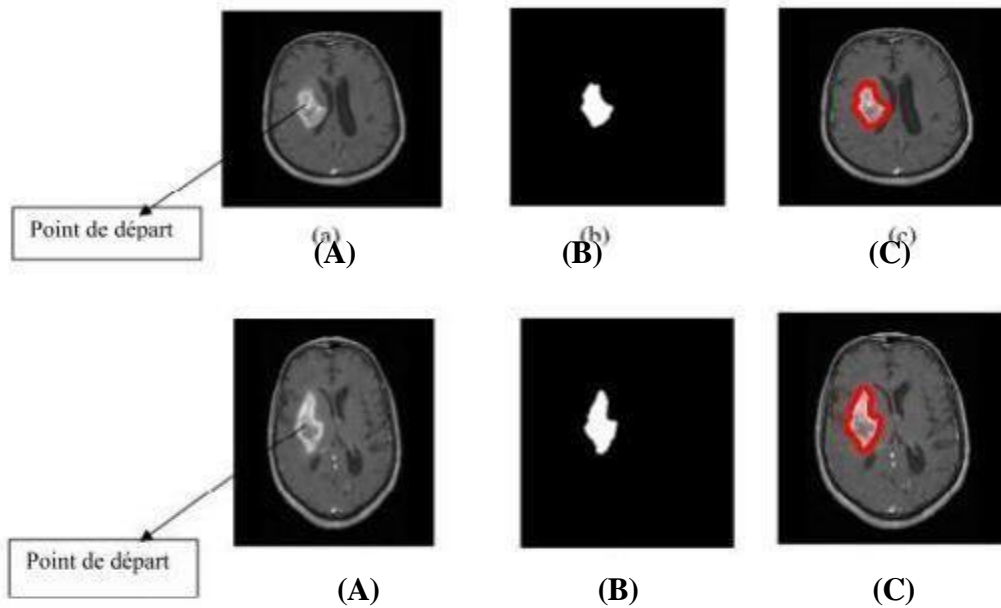


Figure II.13: Résultat de la segmentation via la méthode de croissance de régions, (a) : image originale, (b) : image binaire, (c) : image segmentée

3.4.3.3 Méthodes division-fusion de région (split-merge Methods) :

Le processus de segmentation via cette méthode est décomposé en deux étapes :

- **Division** : analyse de chaque région. Si celle-ci ne vérifie pas le critère d'homogénéité, alors on divise cette région en blocs (le plus généralement en 4 quadrants) et l'on réitère le processus sur chaque sous-région prise individuellement.
- **Fusion** : si l'union de deux régions voisines (R_i, R_j) vérifie le critère d'homogénéité alors fusion des régions. Soit $R (i = 1, \dots, n)$ un ensemble de régions qui correspondent

La partition d'une image, et soit P un prédicat mesurant l'homogénéité de ces régions.

Si $P(R) = \text{faux}$ alors R non homogène (subdivision des régions ou split).

Si $P(R_i \cup R_j) = \text{vrai}$ (sachant que $i \neq j$ et $R_i \cup R_j =$ ensemble connexe) alors R_i et R_j sont homogènes et doivent être fusionnées (fusion des régions ou merge).

La (figure II.14) illustre les résultats obtenus suite à l'application de la méthode de division-fusion [34].

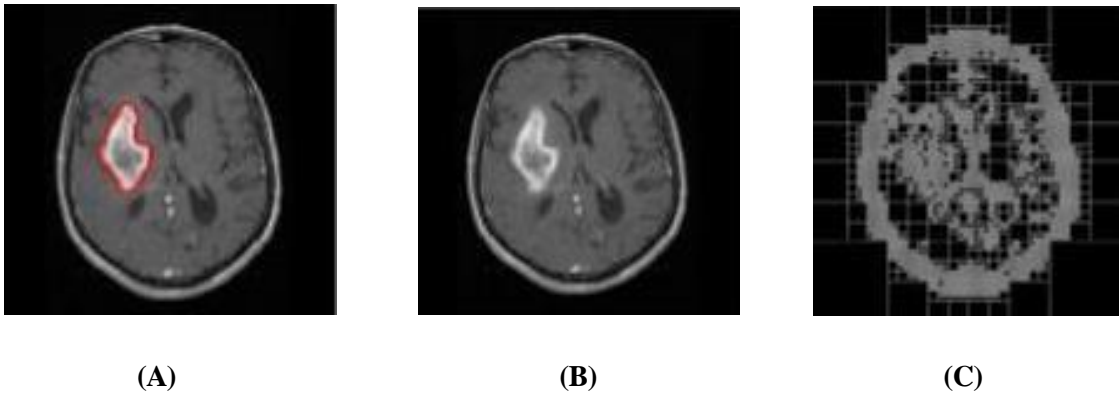


Figure II.14: Résultat de la segmentation via la méthode division-fusion (split and merge)
 (A) : image initiale, (B) : application de l'algorithme de division, (C) : image segmentée après l'étape de fusion.

3.4.3.4 Méthodes basées sur les contours (Edge détection based Methods) :

La détection de contour et une technique de réduction d'information dans les images. Les contours constituent en effet des indices riches, au même titre que les points d'intérêts, pour toute interprétation ultérieure de l'image. Les contours sont caractérisés par des discontinuités de la fonction d'intensité dans les images, le principe de leur détection repose donc sur l'étude des dérivées de la fonction d'intensité dans l'image, les extrema locaux du gradient de la fonction d'intensité et, les passages par zéro du lapalucien. La difficulté réside dans la présence de bruit dans les images [35].

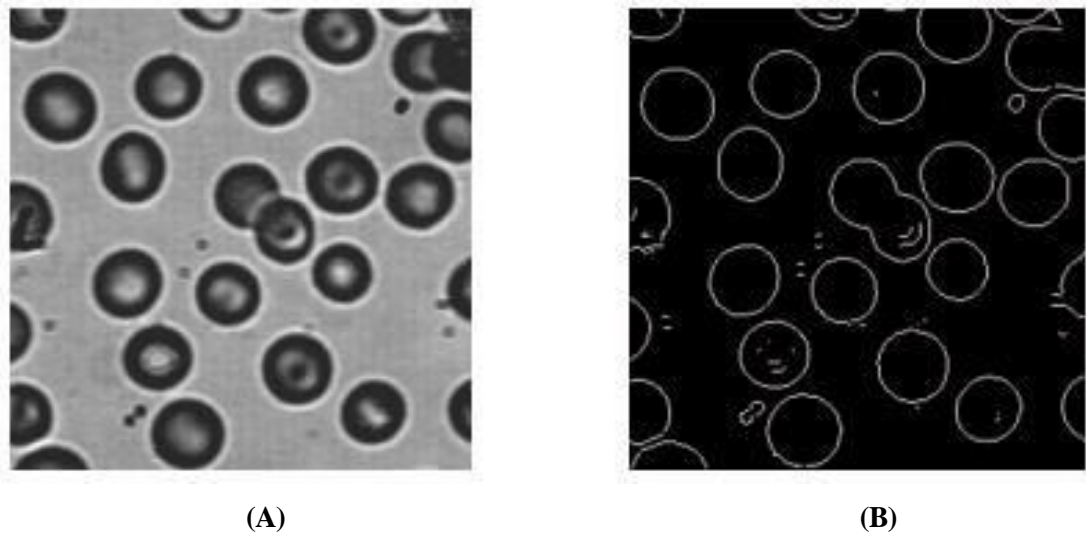


Figure II.15:(A) Image originale, (B) Image Segmentée par Contour

3.4.3.4.1 Quelques méthodes utilisées en détections des contours

Plusieurs algorithmes issus de l'apprentissage automatique, supervisé ou non ont été utilisés afin de réaliser la tâche de détections des contours, nous allons voir dans ce qui suit lequel.

3.4.3.4.2 Approches Contours :

L'approche contour consiste à identifier les changements entre les régions. En général, un élément de contours est un point de l'image appartenant à la frontière de deux ou plusieurs objets ayant des niveaux de gris différents. Les variations d'intensité de lumière et de couleurs sont très bien perçues par le système visuel humain. la figure suivant montre quelques modèles de contours [36].

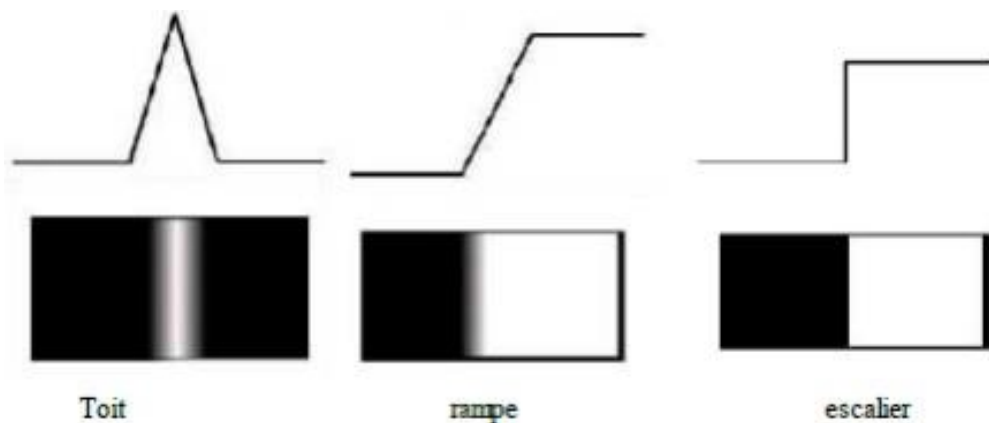


Figure II.16: Quelques modèles de contours.

- Marche d'escalier : le contour est net (contour idéal).
- Rampe : le contour est plus flou.
- Toit : il s'agit d'une ligne sur un fond uniforme.

3.4.3.4.3 Opérateurs de Sobel et Prewitt :

Les opérateurs de "Sobel" et de "Prewitt" permettent d'estimer la norme du gradient bidimensionnel d'une image en niveau de gris. Ces opérateurs consistent en une paire de masques de convolution 3×3 . Pour ces opérateurs les dérivées directionnelles horizontale et verticale s'expriment sous la forme :

$$\frac{\Delta I}{\Delta I} = h_i * I(i, j) \text{ et } \frac{\Delta I}{\Delta J} = h_j * I(i, j)$$

$$\text{Avec : } h_i = \begin{bmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -c \end{bmatrix} \quad \text{Et } h_j = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & -c \\ 1 & 0 & -1 \end{bmatrix}$$

Les matrices h_i et h_j le sont appelées masques. Les masques de Prewitt sont définis par $c = 1$ et les masques de Sobel par $c = 2$.

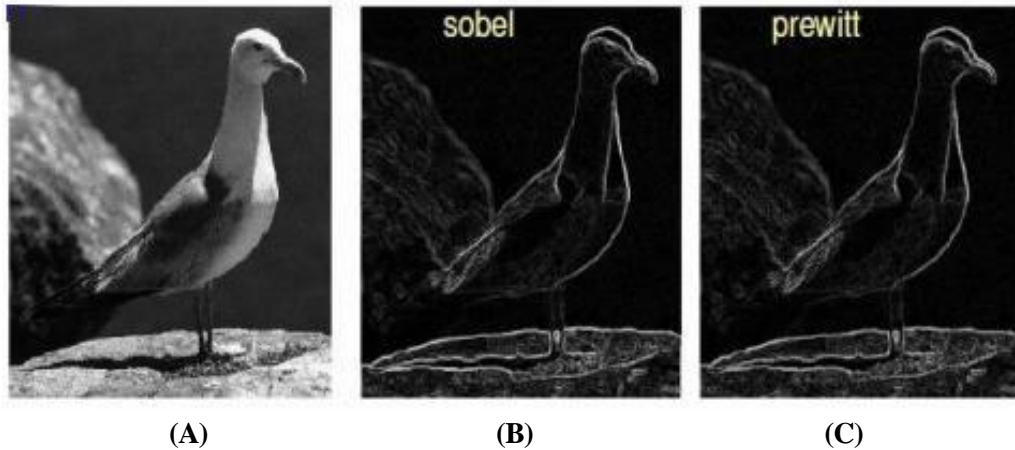


Figure II.17: Image originale, utilisation du filtre de sobel, prewitt

3.4.3.4.4 Opérateur de Roberts (1965)

Le détecteur de Roberts permet de calculer le gradient bidimensionnel d'une image de manière simple et rapide. Ce principe ne diffère pas beaucoup de celui des opérateurs de "Prewitt" et "Sobel" [36].

Les masques de convolution de Robert sont :

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ et } \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

La figure suivante nous montre les contours détectés par ces filtres :

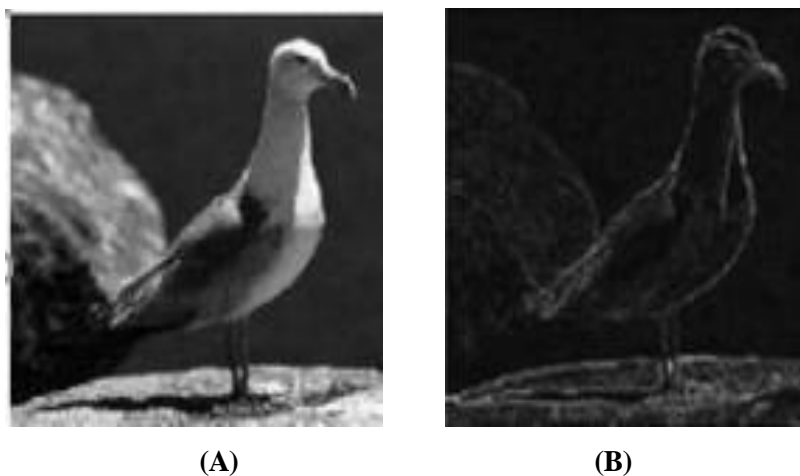


Figure II.18: Détection de contour par les différents filtres.

3.4.3.4.5 Algorithme de Canny

Pour détecter les contours d'objets présents dans une image, l'algorithme de Canny utilise les cartes de gradient de celle-ci. Il permet de satisfaire les trois conditions à respecter lors de la conception des filtres de détection :

- Qualité de la détection des vrais points de contour,
- Localisation des points détectés,
- Unicité de la solution (en cas de détection multiple).

La prise en compte de ces trois critères par l'opérateur, fournit une première approximation de la fonction de Gauss $g(i)$, proposée par Canny :

$$G(x, y, \sigma) = \left(\frac{1}{\sqrt{2\pi\sigma}} \right) e^{-(x^2+y^2)/2\sigma^2}$$

La dérivée de l'image $I_f(x, y)$ filtrée est :

$$I_f(x, y) = \nabla(G(x, y) \otimes I_i(x, y))$$

Avec :

$I_i(x, y)$: Image d'entrée

$G(x, y)$: Fonction de Gauss

$I_f(x, y)$: Image de sortie

$$\begin{aligned} I_f(x, y) &= \left(\frac{\partial}{\partial x} + \frac{\partial}{\partial y} \right) (G(x, y) \otimes I_i(x, y)) \\ &= G'_x(x, y) \otimes I_i(x, y) + G'_y(x, y) \otimes I_i(x, y) = f_x + f_y \end{aligned}$$

Avec :

$$G'_x(x, y) = \left(\frac{-x}{\sqrt{2\pi\sigma^3}} \right) e^{-(x^2+y^2)/2\sigma^2}, \quad G'_y(x, y) = \left(\frac{-y}{\sqrt{2\pi\sigma^3}} \right) e^{-(x^2+y^2)/2\sigma^2}$$

Les filtres étant séparables, nous pouvons réaliser séparément les convolutions suivant x et y :

$$G'_x(x, y) = G'_x(x) \otimes G(y) \quad \text{et} \quad G'_y(x, y) = G'_y(y) \otimes G(x)$$

$$\text{D'où} \quad f_x = G'_x(x, y) \otimes I_i(x, y) = G'_x(x) \otimes G(y) \otimes I_i(x, y)$$

$$\text{Et} \quad f_y = G'_y(x, y) \otimes I_i(x, y) = G'_y(y) \otimes G(x) \otimes I_i(x, y)$$

La direction et l'amplitude sont alors données par les relations suivantes :

$$A = \sqrt{f_x^2 + f_y^2} \quad \text{Et} \quad \text{Dir} = \arctan \left(\frac{f_y}{f_x} \right)$$

Afin d'obtenir les cartes de la norme des gradients et de leurs orientations locales, on suit ces étapes :

- Filtrage de l'image par un filtre Gaussien dont la variance 2σ fixe le degré de lissage.
- Estimation de l'amplitude et de l'orientation du gradient en appliquant un filtre de Sobel
- Sélection des maxima locaux.
- Utilisation du seuillage par hystérésis pour détecter et relier les contours [37].

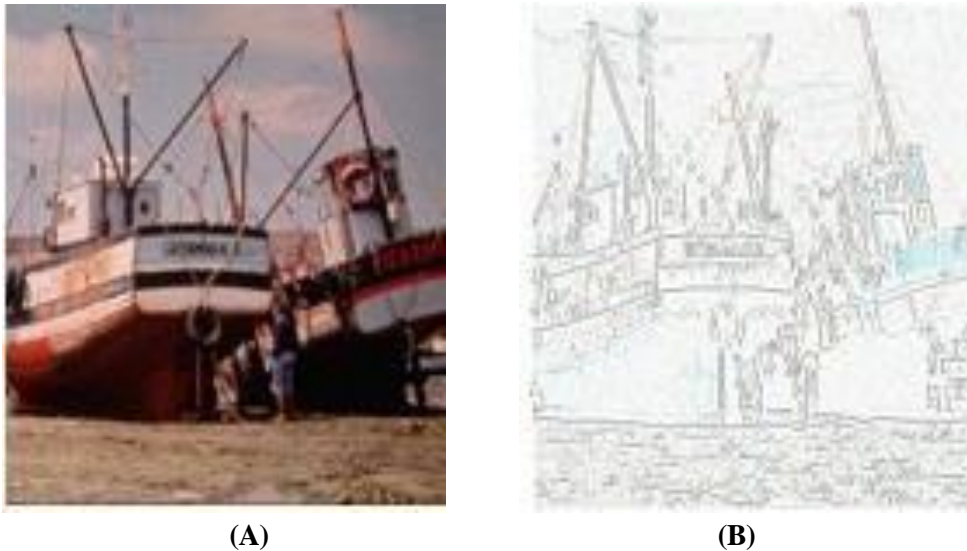


Figure II.18: Image Originale Boat et Détecteur de Canny

4 Conclusion

Dans ce chapitre nous avons exposé quelques notions de base sur la segmentation des images qui représente notre domaine d'application. Nous survoler quelques approches et méthodes utilisées pour aboutir à la segmentation. Il y a des approches qui sont basées seulement sur l'information du pixel (vue locale) au moment de la segmentation, tel que la méthode par croissance de régions, il y a celle qui sont basées sur les contours, et d'autres approches basées sur l'information issue de toute l'image pour faire la segmentation (vue globale) telles que les méthodes de seuillage. Il existe aussi l'approche hybride basée ces deux approches à la fois (vue globale et locale) comme les méthodes division-fusion de région et les méthodes de regroupement. Plusieurs algorithmes sont utilisés pour réaliser la tâche de segmentation.

Chapitre III : Détection et suivi d'objet

Chapitre III : Détection et suivi d'objet

1 Introduction

La détection et le suivi d'objets sont parmi les problèmes les plus étudiés ces dernières années. Ils sont des tâches importantes et difficiles dans de nombreuses applications de vision par ordinateur telles que la robotique, la vidéosurveillance. La détection d'objet consiste à localiser l'objet dans chacune des trames d'une séquence vidéo. Le suivi d'objet est le processus de localisation spatiotemporelle d'un objet en mouvement au cours d'une séquence vidéo. Chaque méthode de suivi d'objet nécessite un mécanisme de détection d'objet, soit dans chaque trame ou lorsque l'objet apparaît d'abord dans la vidéo.

Le suivi d'objets dans une séquence d'images vidéo est un problème qui demande une extraction et un traitement d'informations provenant d'images complexes. Ce problème devient de plus en plus difficile si la contrainte temps réel est exigée. Il existe dans la littérature un grand nombre important de méthodes de suivi. Ce nombre est dû, d'une part au nombre important de problèmes à régler et d'autre part à la diversité des types d'applications concernées par le suivi. Ainsi, chacune des méthodes peut traiter certains aspects et échoue sur d'autres. Ce chapitre introductif est consacré à la description des difficultés que les algorithmes de suivi d'objet sont susceptibles de rencontrer. Ensuite, une brève présentation sur la représentation de la forme d'objet et les primitives de suivi d'objet. Enfin, nous décrivons les principales techniques de détection et suivi d'objet dans une séquence d'images.

2 Suivi d'objet

2.1 Définition

Le suivi d'objet consiste à estimer la trajectoire d'un objet en mouvement dans une séquence vidéo, en identifiant sa position dans chaque image après une détection initiale. Ce processus s'appuie sur des caractéristiques visuelles telles que la couleur, la forme et la vitesse. Les techniques de suivi comprennent deux étapes principales : la détection des candidats potentiels dans chaque image et l'appariement de ces candidats d'une image à l'autre pour maintenir un suivi cohérent.

Les méthodes de suivi varient : certaines utilisent la soustraction de fond pour détecter les objets en mouvement, bien que cela soit inefficace si la caméra bouge. D'autres se basent sur des paramètres de l'objet, reconnaissant l'objet par son modèle pour estimer sa position. Récemment, le suivi d'objets a évolué vers des approches basées sur la détection dynamique,

utilisant des modèles statistiques classés en génératifs, discriminatifs ou hybrides. Ces méthodes de suivi par détection sont très efficaces, bénéficiant des avancées en algorithmes de détection d'objets [38].

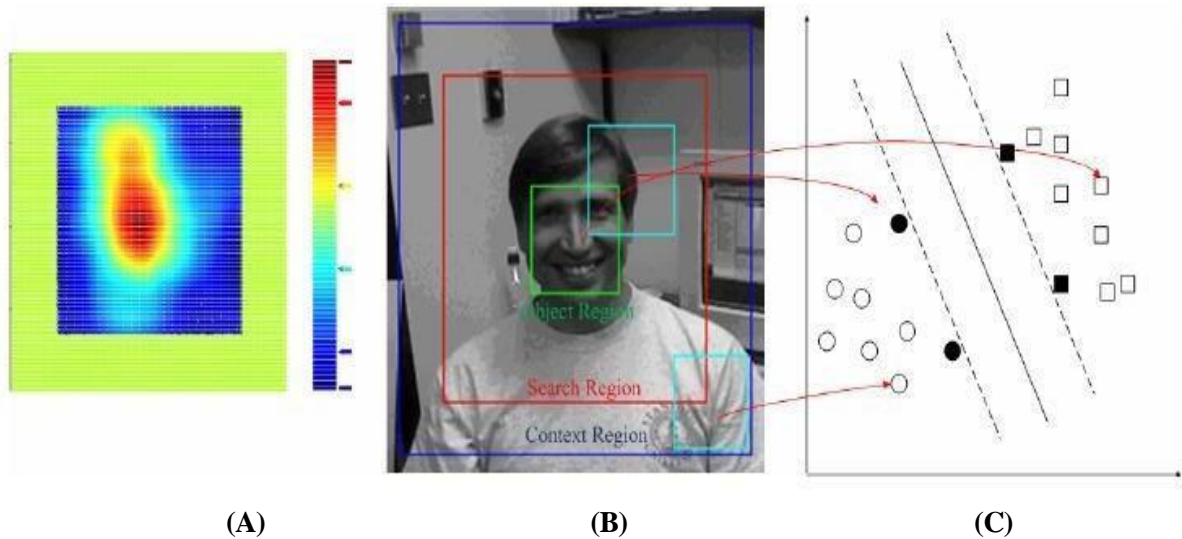


Figure III.1: Illustration du suivi par la détection basée sur la classification des SVM. (A) montre la carte de score de visage / non-visage classification ; (B) affiche la zone de recherche et de la région de contexte pour ; (C) trace l'hyperplan de classification

2.2 Domain d'application

Le suivi d'objets dans des séquences vidéo a suscité un grand intérêt dans la dernière décennie en raison de la variété de ses domaines d'applications, tels que :

- La vidéosurveillance (détection, suivi, reconnaissance du comportement de personnes, d'intrus),
- Le militaire (suivi des cibles ou guidage de missiles),
- La vidéoconférence (suivi des interlocuteurs),
- La gestion et l'analyse du trafic (le suivi d'une voiture ou des bords d'une route depuis une caméra embarquée sur un véhicule),
- La robotique : suivie d'obstacles pendant une phase d'évitement,
- Suivi d'indices visuels dans une tâche asservie par vision,
- Suivi d'un opérateur (corps, visage, main...) pour définir des modes d'interaction évolués entre l'Homme et la Machine..., etc.
- Imagerie médicale [38].

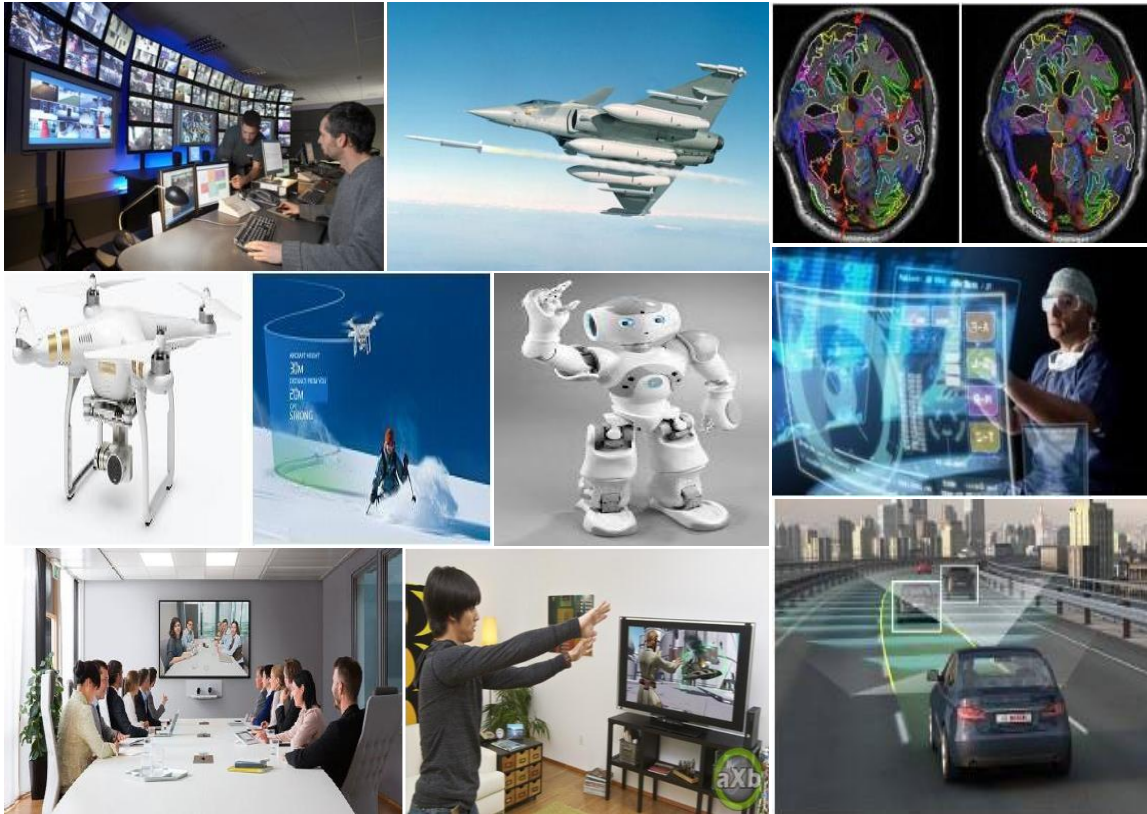


Figure III.2: Quelques applications de suivi d'objets.

2.3 Les défis du suivi

Malgré les progrès significatifs en vision par ordinateur (notamment dans la modélisation des régions d'intérêt, le calcul des caractéristiques visuelles, la modélisation des mouvements et la détection d'objets), il reste très difficile de concevoir un modèle d'apparence robuste et une stratégie de recherche efficace sans connaissance préalable du modèle d'apparence. Cette difficulté est due aux environnements non contraints et à la complexité des apparences et des mouvements des objets suivis, ainsi qu'à divers facteurs liés aux données ou à l'application [39] :

2.3.1 Occultations

Occultations, l'observation d'un objet peut être cachée par un autre objet ou un autre composant de la scène. Deux types d'occultations dans une scène sont possibles :

- Occultation partielle, quand un objet déplace devant un mur, une table et d'autres objets statiques. Ce type d'occultation affecte uniquement une petite partie de l'observation de l'objet d'intérêt (figure III.3 (A)).

- Occultation totale affecte toute la région de l'objet d'intérêt où l'observation de l'objet disparaît complètement certains temps (figure III.3 : (B))



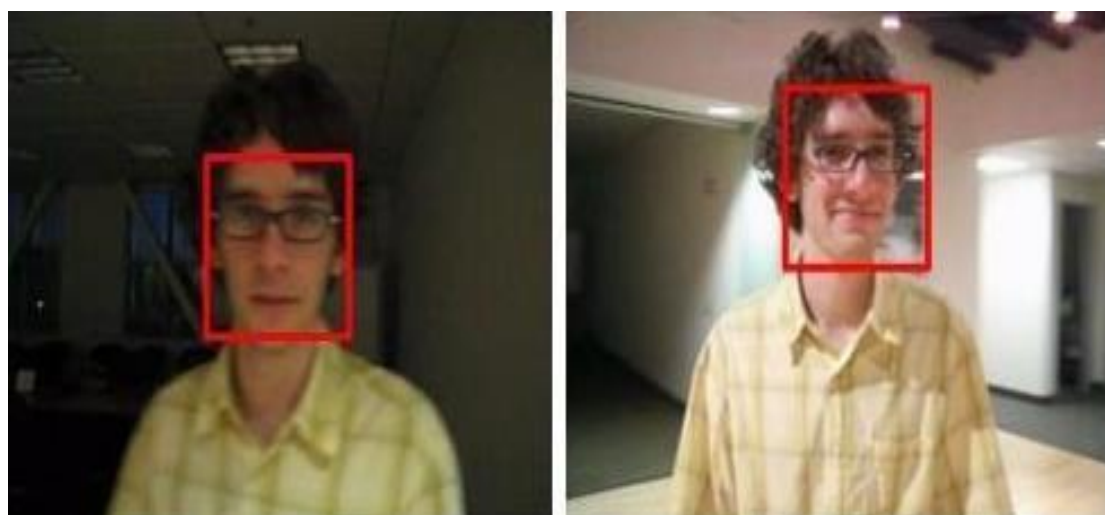
(A)

(B)

Figure III.3: Exemple d'occultation : (A) : occultation partielle, (B) : occultation totale.

2.3.2 Changement d'illumination

Les images enregistrées par une caméra dépendent du contenu physique de la scène aussi bien que des conditions d'éclairage. Où la signature visuelle d'un objet change avec le changement de l'environnement de capture, par exemple : la couleur d'un objet dans une salle est totalement différente de la couleur du même objet à l'extérieur de cette salle.



(A)

(B)

Figure III.4: Exemple de changement des conditions d'éclairage lors d'un suivi de visage sur la séquence vidéo David.

2.3.3 Changement d'échelle

Cette transformation change la taille de l'objet recherché et donc éventuellement du patron associé. La majeure partie des applications ne traitent pas ce cas de figure et considèrent une fenêtre de taille fixe. Le suivi est alors moins précis (figure III.5)



Figure III. 5: Exemple de changement d'échelle

2.3.4 Changement du modèle d'apparence

Les objets non rigides peuvent subir des changements au niveau de leurs apparences. En fait, les objets cibles sont généralement représentés par des formes géométriques avec une orientation initiale. Dans le monde réel, ces objets peuvent subir des variations de rotations importantes soit sur le plan de l'image ou des rotations sur le plan réel tridimensionnel (par exemple, une personne vue de face initialement après une rotation, on aura sa vue de dos) [40].

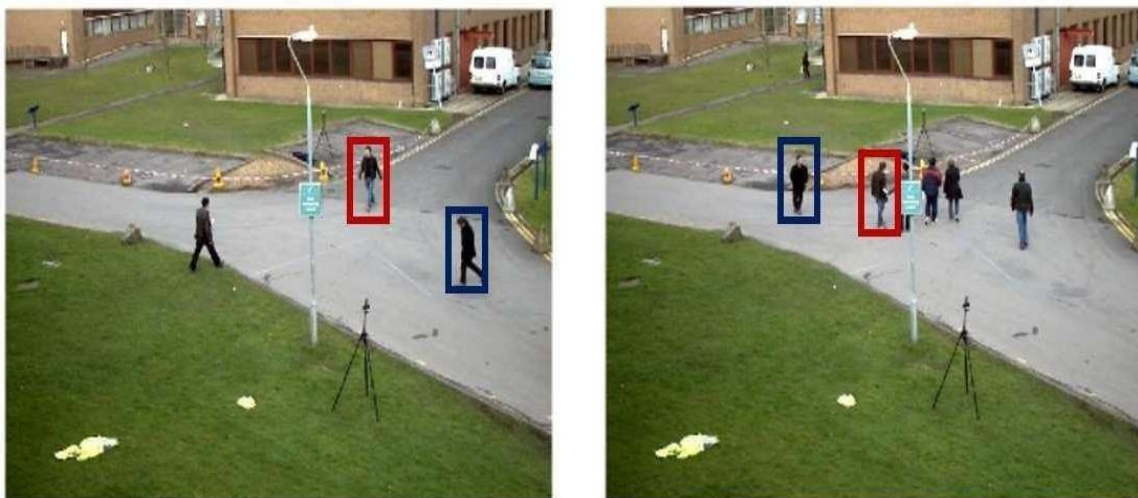


Figure III. 6: Exemples de Changement du modèle d'apparence dû aux rotations.

2.3.5 Mouvements de caméra

Certains systèmes de suivi supposent le statisme de la caméra pour détecter les objets en mouvement [39]. En effet, il est plus facile de modéliser l'arrière-plan lorsque la caméra est fixe. Les régions de l'image où des changements se produisent sont ainsi extraites et considérées comme des cibles potentielles (car ne suivant pas le modèle de l'arrière-plan). Cependant, la contrainte de la caméra stationnaire n'est souvent pas valide. Par exemple, un suivi hors-ligne peut être effectué sur une séquence vidéo enregistrée par une caméra déplacée et orientée par un opérateur humain [41]



Figure III. 7: Exemple de mouvement de camera.

2.3.6 Objet en mouvement rapide

Il résulte d'un mouvement rapide le phénomène de « flou de bougé » gommant les contours et diluant les couleurs et textures. Les applications se basant sur ces traits seront donc pénalisées. De plus, lors d'un mouvement rapide, un suivi imprécis perdra plus facilement l'objet (figure III.8) [42].



Figure III.8: Exemple de mouvement rapide.

2.4 Méthodes de suivi

2.4.1 Suivi par points

Un objet est représenté par un point, ou par un ensemble de points si ce point est son centroïde. Cette méthode est adaptée pour suivre des objets occupant de petites zones dans une image. Dans cette catégorie, les approches peuvent être déterministes ou probabilistes [43].

❖ L'approche déterministe

Dans l'approche déterministe, le suivi s'effectue en minimisant une distance calculée sur certaines caractéristiques de l'objet. Ces caractéristiques sont l'apparence (Similarité de forme et/ou de contenu photométrique et/ou de mouvement). Les modèles d'objet basés sur l'apparence peuvent être des densités (histogrammes de couleur ou de contour), une carte de contours (contour ouvert ou fermé de l'objet) ou une combinaison de ces modèles [43].

❖ L'approche probabiliste

Les méthodes probabilistes permettent de gérer les fluctuations causées par des variations de mouvement et d'apparence des objets au fil du temps et les cibles candidates qui sont souvent corrompues par du bruit, en ajoutant une incertitude au modèle de l'objet et aux modèles des cibles candidates. Ces méthodes ont un faible coût calculatoire, mais elles ont un inconvénient majeur parce qu'elles dépendent exclusivement de la qualité de la détection. En cas de fausse détection, le suivi est détérioré [43].



Figure III. 9: Représentation d'objet par points

2.4.2 Suivi de silhouettes

Un contour délimite les frontières d'un objet, tandis que la région à l'intérieur représente sa silhouette. Les méthodes basées sur le contour et la silhouette modélisent efficacement les objets de forme complexe en utilisant les informations à l'intérieur de la région objet. Leur objectif est de déterminer la région occupée par l'objet à partir d'un modèle généré à partir des images précédentes. Ces méthodes sont utilisées pour extraire la silhouette d'un objet qui se déforme au fil du temps. Ce suivi peut être réalisé à l'aide de modèles d'état ou en minimisant des fonctions d'énergie sur le contour, où le terme d'attache aux données favorise le positionnement du contour estimé le long des zones à fort gradient de l'image [44].

• Les approches explicites

La plupart des méthodes de suivi par contour cherchent à représenter celui-ci par un ensemble de paramètres et à le suivre par une méthode de filtrage appropriée. Ces paramètres jouent le rôle du modèle d'état de l'algorithme de filtrage. Ces méthodes peuvent être mises en œuvre avec un coût calculatoire assez faible, mais elles suivent plus difficilement les changements topologiques, la division ou la fusion des régions [44].

• Les approches implicites

Contrairement aux méthodes précédentes, les méthodes fondées sur une représentation implicite sont robustes aux changements de topologie des objets grâce à la minimisation d'une fonctionnelle d'énergie qui permet de suivre un contour ou une région malgré des changements de topologie. Mais la minimisation de ces fonctions est généralement plus coûteuse et la convergence vers un minimum globale n'est pas nécessairement assuré [44].

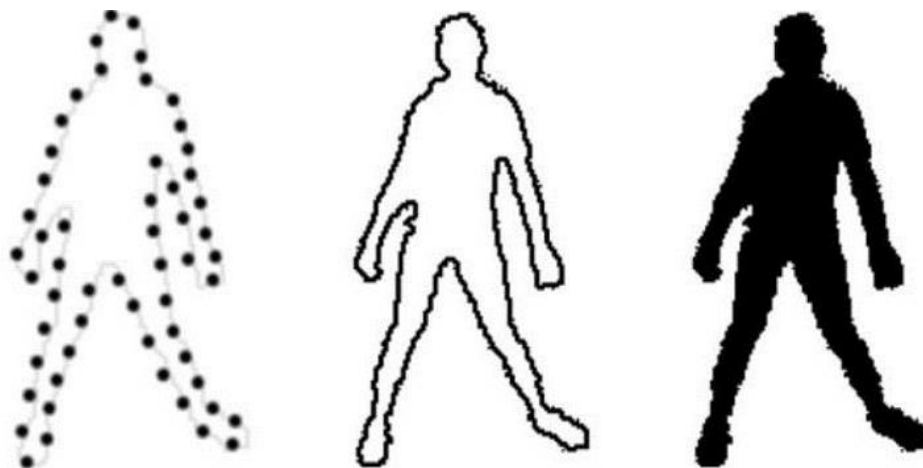


Figure III. 10: Représentation d'objet par silhouette

2.4.3 Suivi de noyaux

Le suivi de noyaux implique de suivre un objet représenté par une forme géométrique simple, comme un rectangle ou une ellipse, avec un mouvement généralement paramétrique (translation, rotation, etc.). L'objet est représenté par un histogramme pondéré calculé sur une région elliptique contenant l'objet. L'algorithme de suivi de noyau le plus utilisé est le mean shift, qui utilise une fonction de densité des histogrammes de couleur de l'objet et une fenêtre candidate. Cette méthode permet d'estimer la position de l'objet en quelques itérations. Cependant, en présence d'objets ayant des distributions de couleurs similaires et situés très près les uns des autres, l'estimation peut devenir ambiguë. Pour éviter cette ambiguïté, MIKRAM et al. ont utilisé une série d'histogrammes à différentes échelles d'une image [45].



Figure III. 11: Représentation d'objet le noyau

2.5 Caractéristiques visuelles pour le suivi d'objets

Les caractéristiques visuelles, généralement extraites d'un modèle d'objet, d'une partie d'objet ou du voisinage d'un point (descripteurs locaux), sont largement utilisées par les trackers. Elles offrent une description de l'apparence de l'objet plus robuste que l'information d'intensité des pixels, qui est sensible aux perturbations telles que le bruit, les variations d'illumination et la réflectance. Les caractéristiques de forme sont moins affectées par les changements de luminosité et les dégradations de l'image, tandis que les caractéristiques de couleur sont plus résistantes aux transformations géométriques [46].

2.5.1 Couleur

La couleur, grâce à son pouvoir discriminant, est l'un des attributs les plus utilisés pour représenter des objets. De nombreux espaces de couleurs existent, chacun avec ses caractéristiques, mais partageant le même principe de base : décomposer les couleurs en composantes primaires. La recherche se concentre souvent sur l'identification de l'espace couleur le plus discriminant. Chaque espace crée une décomposition des couleurs selon ses propres paramètres. Le choix de l'espace colorimétrique est crucial, bien qu'il n'y ait pas d'espace idéal pour la représentation. Par exemple, dans l'espace RVB, chaque couleur est une combinaison de rouge, vert et bleu. Certains chercheurs utilisent l'espace RVB, tandis que d'autres préfèrent des espaces alternatifs comme HSV, qui sépare la teinte, la saturation et l'intensité, offrant une meilleure invariance, ou CIELab, considéré comme perceptuellement uniforme. Cependant, tous les espaces couleur sont sensibles au bruit [39].

2.5.2 Texture

La texture mesure la variation d'intensité d'une surface, quantifiant des propriétés comme la douceur et la régularité. Contrairement à la couleur, la texture nécessite une étape de traitement pour générer des descripteurs. Moins sensibles aux changements d'éclairage que la couleur, les caractéristiques de texture sont essentielles dans les algorithmes de détection et de reconnaissance. Utiliser la texture pour représenter un objet est courant et souvent discriminant. La texture a de nombreuses qualités, dont sa faible sensibilité aux variations d'illumination. Elle peut être décrite en termes de direction, de grain, de contraste, etc. Cette diversité permet de multiples représentations. Les modèles binaires locaux (LBP) sont basés sur les caractéristiques de texture [39].

2.5.3 Gradient

L'information de gradient spatial est largement utilisée pour décrire la forme et le contour des objets dans les images. Elle offre une meilleure résilience aux variations de luminosité que les informations de couleur. Les contours dérivés du gradient sont employés dans diverses méthodes de suivi d'objet, permettant l'identification de points caractéristiques. Des approches telles que CONDENSATION et les techniques de minimisation d'énergie le long des contours sont utilisées pour suivre les objets avec des contraintes de régularisation. Les histogrammes d'orientations de gradient (HOG) servent de base pour construire des descripteurs comme SIFT, qui intègrent à la fois la détection et la description d'objets invariants à l'échelle [47].

3 Détection d'objets

3.1 Définition

La détection d'objets est un domaine de recherche crucial pour diverses applications telles que la classification d'images, l'analyse du comportement humain et la conduite autonome. Elle consiste à localiser et classer les objets dans une image en les encadrant avec des boîtes de délimitation. Les méthodes de détection se divisent en deux grandes catégories : celles qui suivent un processus traditionnel de propositions de régions avant la classification, et celles qui abordent la détection comme un problème de régression ou de classification. Parmi les méthodes populaires, on trouve R-CNN, Fast R-CNN, Faster R-CNN, et Mask R-CNN pour la première catégorie, et MultiBox, AttentionNet, YOLO, SSD, YOLOv2, DSSD, et DSOD pour la seconde. L'objectif principal de ces approches est d'améliorer la précision de détection tout en réduisant la complexité de calcul pour permettre des performances en temps réel sur diverses plateformes. Les modèles de détection peuvent être classés en approches en une seule étape ou en deux étapes, selon leur architecture globale [48].

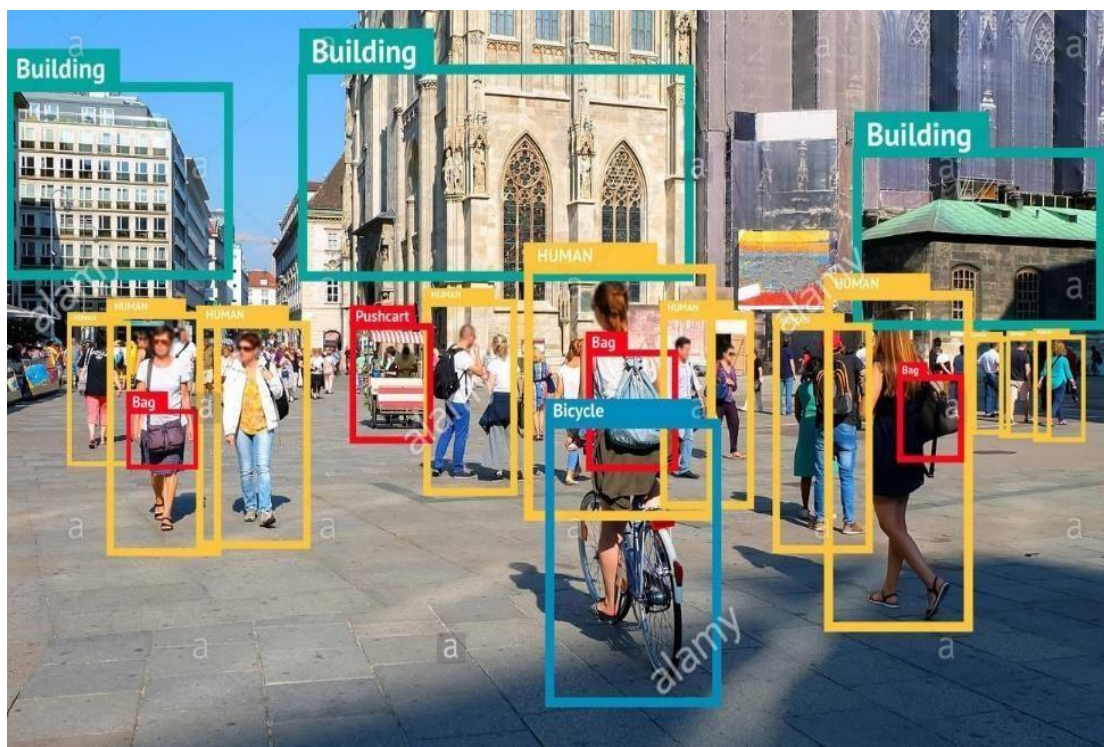


Figure III.12: Exemples de détection d'objet

3.2 Approches-en une seule étape

Les détecteurs à pas unique sont des modèles simples et rapides avec une précision raisonnable par rapport aux réseaux régionaux. Ils sont similaires aux réseaux RPN, mais au lieu de prédire des objets/non-objets, ils prédisent directement les classes d'objets et coordonnent les décalages. Certains des détecteurs à une seule étape sont YOLO.

3.2.1 You Only Look Once (YOLO)

3.2.1.1 Définition

YOLO (You Only Look Once) est une nouvelle approche de détection d'objets qui se différencie des méthodes antérieures en utilisant la détection comme un problème de régression. Un seul réseau neuronal prédit directement les boîtes englobantes et les probabilités de classe à partir des images complètes en une seule évaluation. Bien que YOLO soit proche du temps réel, il présente des limitations spatiales et des déséquilibres de classe pendant l'entraînement, ce qui peut affecter sa performance sur les petits objets ou ceux qui apparaissent en groupes [49]

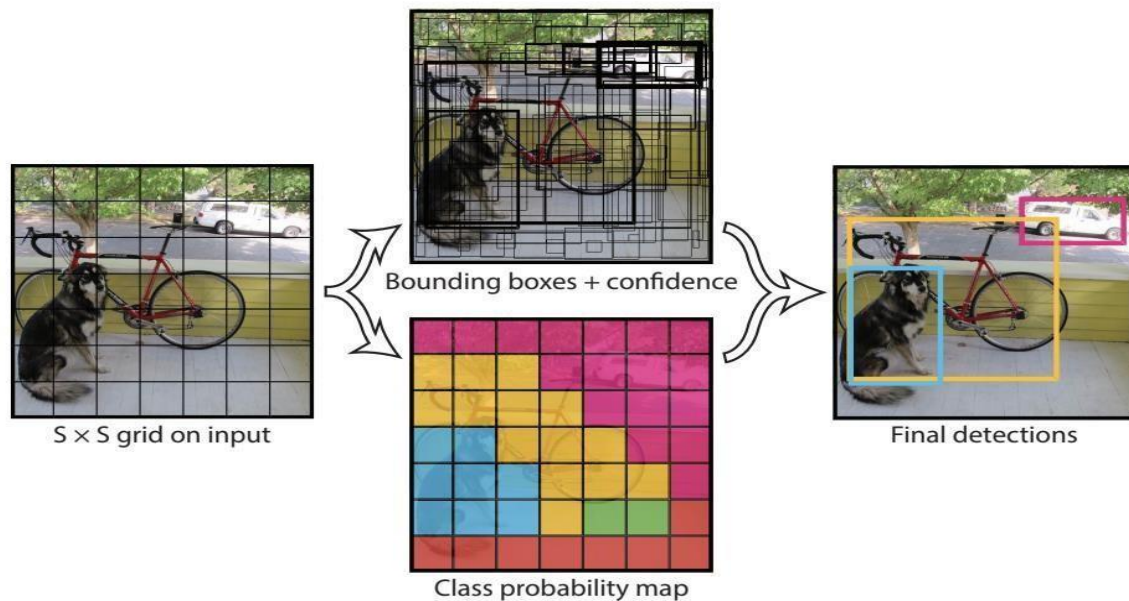


Figure III.13: YOLO détection d'objet

3.2.1.2 Les avantages et les désavantages de YOLO

Le tableau suivant représente les avantages et désavantage de YOLO [49] :

Tableau D1: les avantages et les inconvénients de YOLO

Avantages de YOLO	Inconvénients de YOLO
<ul style="list-style-type: none"> • YOLO se distingue par son faible taux de fausses détections d'arrière-plan et sa rapidité d'exécution. • Il est simple à mettre en œuvre. Une autre force de YOLO réside dans sa capacité à détecter des objets dans des œuvres artistiques, surpassant largement les méthodes de détection traditionnelles telles que les séries RCNN en termes de taux de détection d'objets sur des images non naturelles. 	<ul style="list-style-type: none"> • Le modèle d'apprentissage YOLO est restreint à utiliser la résolution d'entrée de l'image utilisée lors de l'apprentissage, car sa couche de sortie est entièrement connectée. • Chaque grille de détection ne peut prédire qu'un seul objet, ce qui pose problème lorsque plusieurs objets se trouvent dans une même grille, comme dans le cas d'images contenant des troupeaux ou des oiseaux. Cette limitation constitue un défaut de la méthode YOLO.

3.2.1.3 Comparaison entre les versions YOLO [50]

YOLO (You Only Look Once) est une famille d'architectures de réseaux neuronaux convolutifs utilisés pour la détection d'objets en temps réel. Chaque version de YOLO apporte des améliorations et des modifications par rapport à la version précédente. Voici les principales différences entre les différentes versions de YOLO :

Tableau D2: Les différences entre versions de YOLO (You Only Look Once).

	YOLO V1	YOLO V2	YOLO V3	YOLO V4	YOLO V5
Année	2015	2016	2018	2020	2020
Backbone	Darknet	Darknet	Darknet, MobileNetV1	CSPDarknet, CSPResNeXt	CSPDarknet
Couches	24	23	53	110-137	137-184
Détecteurs	1	5	9	11	3
Rendu en temps réel	Non	Oui	Oui	Oui	Oui
Précision	Moins précis que v2	Amélioration de la précision par rapport à v1	Meilleure précision par rapport à v2	Meilleure précision par rapport à v3	Variable en fonction des configurations
Objectifs	Détection d'objets	Détection d'objets, segmentation d'instance	Détection d'objets, segmentation d'instance, estimation de pose	Détection d'objets, segmentation d'instance, estimation de pose	Détection d'objets, segmentation d'instance

3.2.1.4 YOLO V3

3.2.1.4.1 Définition [51]

YOLO v3, abréviation de You Only Look Once version 3, est la troisième itération de l'algorithme de détection d'objets YOLO. Introduit en 2018.

YOLO v3 est un algorithme de détection d'objets rapide et précis, largement utilisé dans divers domaines pour sa capacité à détecter des objets en temps réel. Ses améliorations architecturales et ses performances en font un choix populaire pour de nombreuses applications de vision par ordinateur.

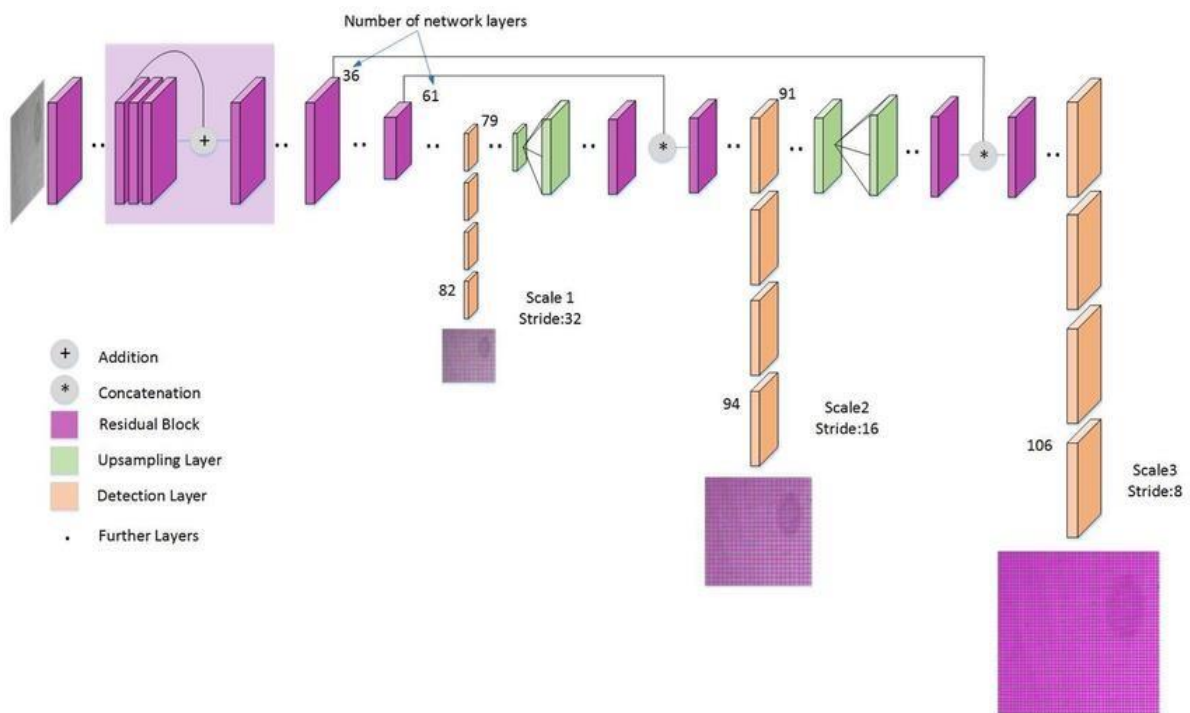


Figure III. 14: La structure de YOLOV3

3.2.1.4.2 Architecture et Performances

YOLOv3 utilise l'architecture Darknet-53, une variante de ResNet, pour la détection d'objets. Cette architecture comprend 53 couches convolutionnelles et a montré des performances de pointe dans la détection d'objets [52].

YOLOv3 fonctionne plus rapidement que d'autres méthodes de détection avec des performances comparables. Par exemple, à une résolution de 320x320, YOLOv3 s'exécute en 22 ms (comme un puissant GPU avec ~45 FPS) avec une précision de 28,2 mAP (mean Average Precision), aussi précis que SSD mais trois fois plus rapide [52].

3.2.1.4.3 Améliorations et Limitations

YOLOv3 a apporté des améliorations incrémentielles par rapport aux versions précédentes, avec des ajustements de conception pour améliorer la précision et la vitesse de détection [52].

Malgré ses performances, YOLOv3 peut rencontrer des difficultés avec les petits objets et la séparation des objets en groupes dans les images, en raison de contraintes liées à la détection d'un seul objet par grille [53].

3.2.1.4.4 Utilisation et Variantes

YOLOv3 est largement utilisé pour la détection d'objets en temps réel, notamment dans des domaines tels que la conduite autonome et la surveillance [53].

Différentes variantes de YOLO, telles que YOLOv4, YOLOv5 et YOLACT, ont été développées pour répondre à des besoins spécifiques en termes de précision et de vitesse de détection.

3.3 Approche en deux étapes (basée sur la région)

Les détecteurs basés sur la région divisent la détection d'objets en deux étapes. La première étape génère un ensemble de régions de l'image qui ont une forte probabilité d'être un objet. La deuxième étape effectue ensuite la détection finale et la classification des objets en prenant ces régions en entrée. Certains des détecteurs basés sur la région sont R-CNN, Fast R CNN, Faster R-CNN, Mask R-CNN.

3.3.1 Region Convolutional Neural Network (R-CNN)

Le modèle R-CNN a été un pionnier dans la structuration des détecteurs d'objets contemporains en introduisant l'approche en deux étapes. Il utilise un modèle de proposition de région pour identifier les zones dans une image susceptibles de contenir des objets, employant l'algorithme de Recherche sélective pour générer ces propositions. La méthode de Recherche sélective analyse l'image à travers des fenêtres coulissantes de différentes dimensions, cherchant à regrouper les pixels adjacents selon la couleur, l'intensité ou la texture pour repérer les objets. Cependant, malgré son influence, R-CNN présente une lenteur significative, prenant environ 47 secondes pour détecter les objets dans une seule image, principalement en raison de son pipeline d'entraînement complexe nécessitant un ajustement minutieux des paramètres [54].

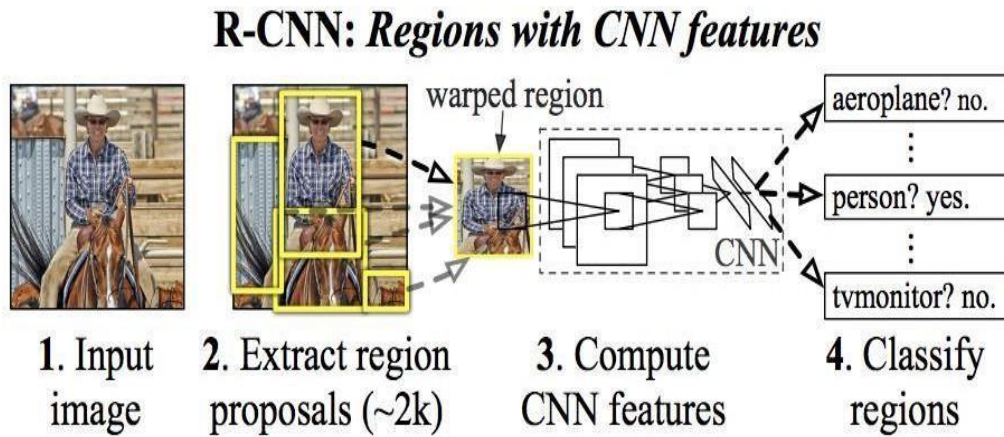


Figure III.15: R-CNN détection d'objet

3.3.2 Rapid R-CNN (Fast R-CNN)

Fast R-CNN est une amélioration de R-CNN qui se compose d'une tête convolutive et de deux têtes SVM (le SVM était le programme qui obtenait le meilleur score en termes de détection et de classification) pour la classification et la régression. Le Fast RCNN vise à exécuter la base convolutive une seule fois sur toute l'image pour générer une carte de caractéristiques. Il recadre les régions à partir de cette carte de caractéristiques générée au lieu de l'image d'entrée entière, réduisant ainsi à la fois le temps et l'espace. La procédure de recadrage est effectuée par un algorithme appelé ROI pooling qui prend les coordonnées des régions obtenues via la recherche sélective et les rogne directement à partir de la carte des caractéristiques de l'image d'origine. De plus, Fast RCNN a introduit un pipeline de formation en une seule étape et une perte multitâche, permettant aux chefs de classification et de régression d'être formés simultanément.

Fast R-CNN est plus rapide et plus efficace en mémoire par rapport au RCNN [55].

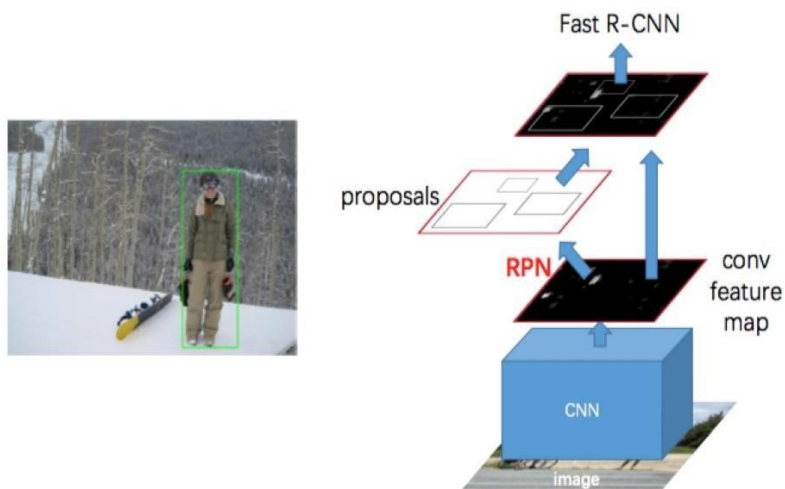


Figure III. 16 : Architecture de Rapid RCNN

3.3.3 Faster R-CNN

Même si Fast RCNN s'est beaucoup amélioré en termes de vitesse et d'empreinte mémoire, il souffre toujours d'une surcharge de l'étape de proposition de région effectuée par un algorithme de recherche sélective, prenant beaucoup de temps pour générer des propositions. Le RCNN plus rapide a introduit le Réseau de proposition de région (RPN) pour remplacer la recherche sélective, qui a la capacité de prédire des régions d'échelles et de rapports d'aspect multiples à travers l'image à l'aide d'un concept appelé ancres. Les ancres sont des cultures rectangulaires d'images couvrant toutes les formes et tailles de régions. Le RCNN plus rapide dispose d'un pipeline d'apprentissage en profondeur de bout en bout et a atteint un temps d'inférence de 198 ms par image.

Invariance d'échelle l'un des problèmes non traités par Faster RCNN. Où le modèle devrait être capable de détecter l'objet de près et aussi de loin [56].

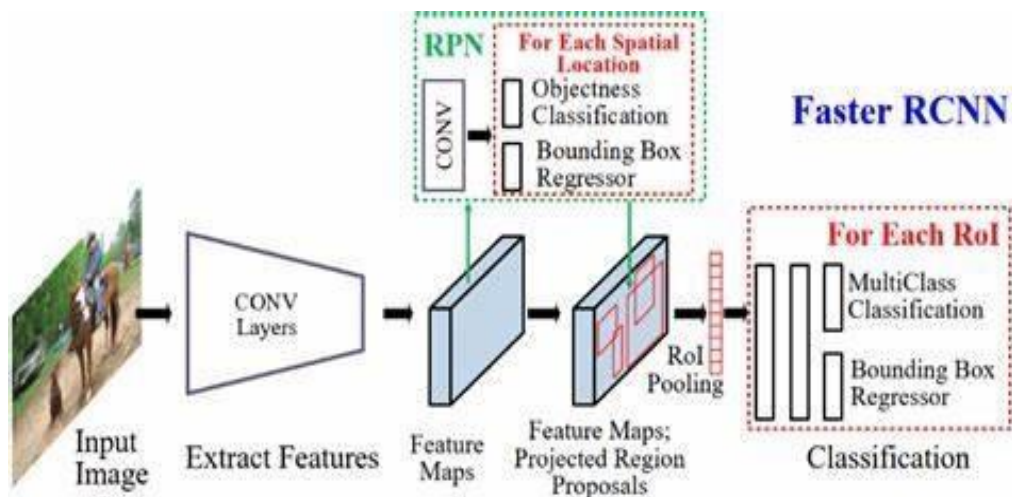


Figure III. 17 : Architecture de Faster RCNN

3.3.4 Mask R-CNN

Le modèle Mask R-CNN est une extension du modèle Faster R-CNN, comme les modèles précédents, permet de détecter des objets et de les classifier. Sa particularité est d'ajouter à cette tâche de détection la segmentation d'instance, c'est à dire que chaque pixel de l'image sera classé. Ainsi, ce double "compétence" représente un avantage par rapport à des modèles de détection, elle vient affiner le résultat proposé. De plus, contrairement à la segmentation sémantique, qui permet d'associer à chaque pixel un label, la segmentation d'instance associe un masque et un label à chaque objet, même si ces objets appartiennent à la même classe. Il est rapide, relativement simple à implémenter et flexible quant aux tâches pouvant lui être incombées [55].

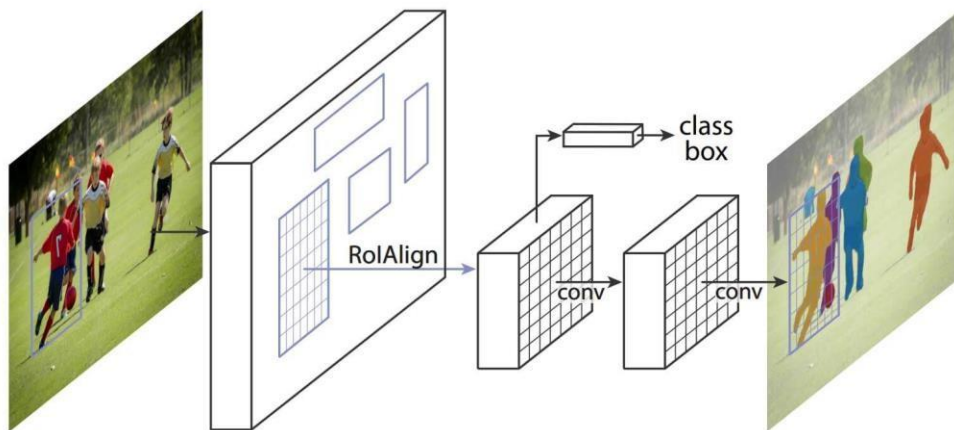


Figure III. 18 : Architecture de Mask RCNN

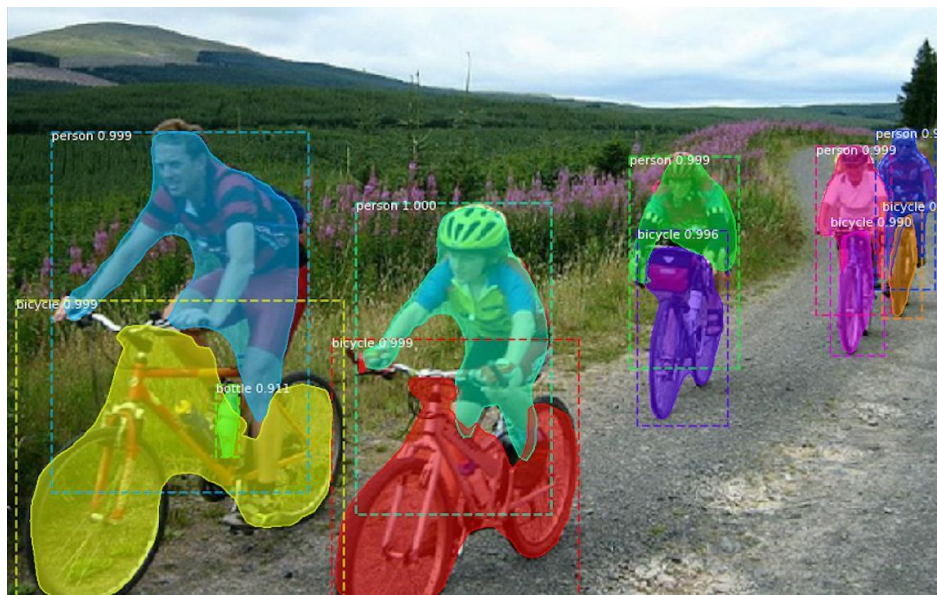


Figure III. 19 : Exemple de détection par Mask RCNN

4 Conclusion

Nous avons conclu notre chapitre sur la détection et le suivi d'objets en vision par ordinateur en soulignant l'importance et les multiples applications de ces techniques, notamment dans la surveillance de sécurité et l'assistance à la conduite autonome. Nous avons couvert à la fois les méthodes traditionnelles et modernes de détection, incluant les réseaux de neurones convolutifs et les algorithmes de deep learning tels que YOLO et SSD. Nous mettons en avant les progrès technologiques réalisés ainsi que les défis persistants, en insistant sur l'importance d'une recherche continue pour améliorer l'efficacité et la précision dans des conditions complexes et variées.

Chapitre IV : Implémentation et discussion des résultats

Chapitre IV : Implémentation et discussion des résultats

1 Introduction

Dans les chapitres précédents, nous avons couvert les bases théoriques du traitement d'images (Chapitre 2) et des techniques de suivi et de détection d'objets (Chapitre 3). Ce chapitre se concentre sur l'implémentation pratique de ces concepts.

Nous présenterons le cadre de développement, les environnements logiciels, et les outils utilisés. Nous détaillerons les étapes spécifiques de l'implémentation, incluant le prétraitement des données visuelles, suivi de la mise en œuvre des algorithmes de suivi et de détection d'objets, en expliquant leur intégration et interaction avec les images prétraitées. Les défis rencontrés et les solutions adoptées seront également discutés.

Nous évaluerons les performances de notre implémentation en termes de précision, rapidité et robustesse, et comparerons les résultats aux attentes théoriques. Ce chapitre vise à démontrer la puissance et l'applicabilité des techniques de traitement d'images et de suivi/détection d'objets.

2 Pourquoi a-t-on choisi MATLAB ?

MATLAB est souvent choisi pour le traitement d'images, la détection et le suivi d'objets en raison de ses nombreux avantages :

- 1. Environnement Intégré :** MATLAB offre un IDE convivial pour l'écriture, le débogage et l'exécution du code.
- 2. Bibliothèques Riches :** Des toolboxes spécialisées pour le traitement d'images et la vision par ordinateur avec des fonctions pré-implémentées.
- 3. Visualisation et Interface Graphique :** Excellentes capacités de visualisation et outils interactifs pour analyser les résultats.
- 4. Support Mathématique Puissant :** Conçu pour les calculs mathématiques avancés, essentiel pour les algorithmes complexes.
- 5. Prototypage Rapide :** Syntaxe simple et bibliothèques riches facilitant le prototypage et la modification rapide des algorithmes.
- 6. Documentation et Communauté :** Documentation complète et vaste communauté d'utilisateurs, avec de nombreuses ressources et exemples.
- 7. Interopérabilité :** Capacité d'interagir avec d'autres langages de programmation et outils, facilitant l'intégration dans des pipelines existants.

8. Applications Diverses : Utilisé dans de nombreux domaines, rendant MATLAB polyvalent pour divers projets.

9. Performance et Optimisation : Outils pour optimiser le code et améliorer les performances, y compris le support pour le calcul parallèle et le traitement sur GPU.

3 Présentation de la boîte à outils GUIDE

GUIDE (Graphical User Interface Development Environment) est un outil intégré de MATLAB destiné à la création d'interfaces utilisateur graphiques (GUI). Il permet aux utilisateurs de concevoir et de déployer facilement des interfaces interactives sans nécessiter de compétences approfondies en programmation.

Les fonctionnalités principales de GUIDE MATLAB incluent :

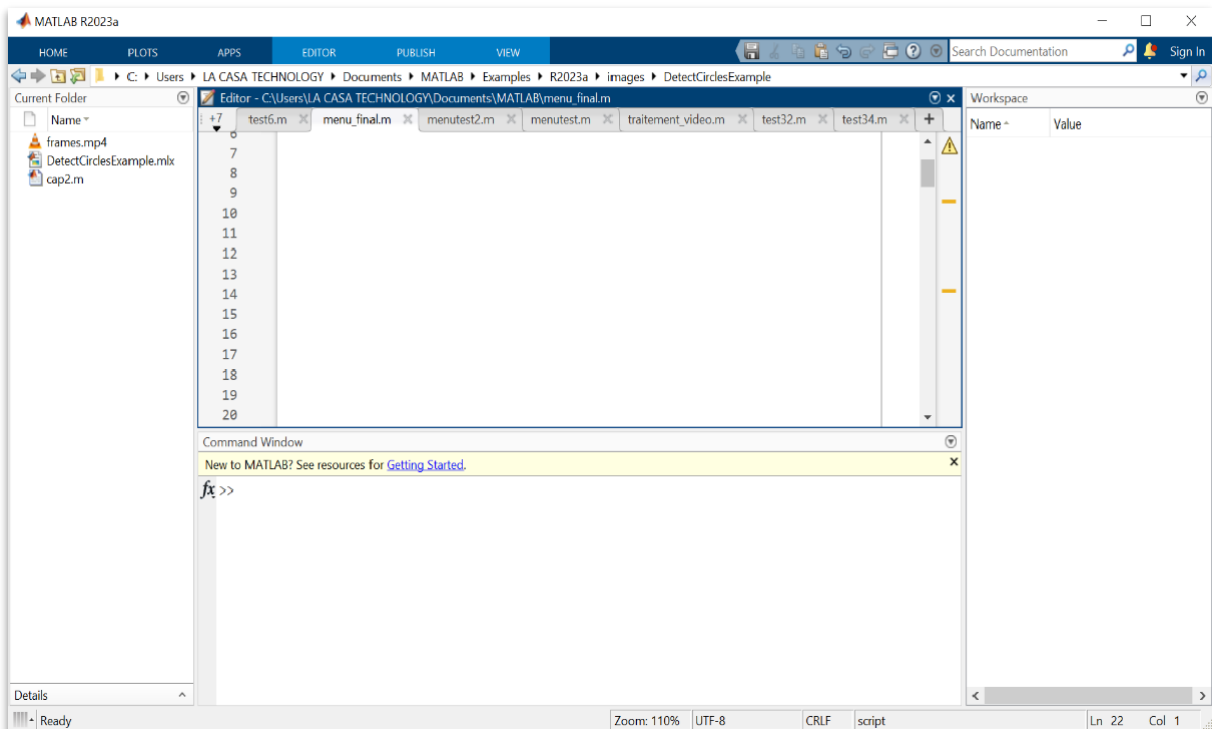
- **Glisser-déposer des composants** : Vous pouvez ajouter des boutons, des cases à cocher, des menus déroulants, des axes graphiques, etc., à votre interface utilisateur en les faisant simplement glisser depuis la palette des composants.
- **Personnalisation** : GUIDE offre de nombreuses options de personnalisation pour chaque composant, notamment la taille, la couleur, la police, etc.
- **Programmation interactive** : Vous pouvez associer des fonctions MATLAB à des événements tels que les clics de souris, les sélections de menu, etc., pour rendre votre interface utilisateur interactive.
- **Génération automatique de code** : GUIDE génère automatiquement le code MATLAB correspondant à votre interface utilisateur, ce qui facilite son utilisation et sa modification ultérieure.

4 Comment créer un guide sur MATLAB ?

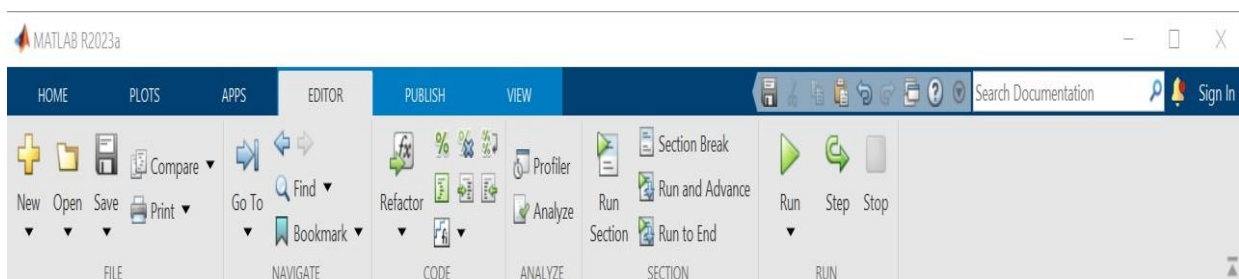
Pour créer un guide sur MATLAB, suivez ces étapes :

1. Ouvrez MATLAB et tapez "guide" dans la fenêtre de commande pour ouvrir l'environnement de développement GUIDE.
2. Une fois dans GUIDE, créez une nouvelle interface utilisateur en sélectionnant "Nouveau" dans le menu "Fichier".
3. Vous pouvez maintenant concevoir votre interface utilisateur en faisant glisser et en déposant les composants souhaités depuis la palette des composants vers la fenêtre de conception.

4. Personnalisez les propriétés des composants selon vos besoins en utilisant l'inspecteur de propriétés.
5. Associez des fonctions MATLAB à des événements en double-cliquant sur les composants et en écrivant le code correspondant dans l'éditeur de code MATLAB.
6. Une fois que votre interface utilisateur est prête, générez automatiquement le code MATLAB en sélectionnant "Générer le code" dans le menu "Fichier".
7. Vous pouvez maintenant exécuter votre interface utilisateur en sélectionnant "Exécuter" dans le menu "Fichier" ou en cliquant sur le bouton "Exécuter" dans l'environnement de développement GUIDE.



FigureIV.1: Interface de MATLAB



FigureIV.2: Menu de MATLAB

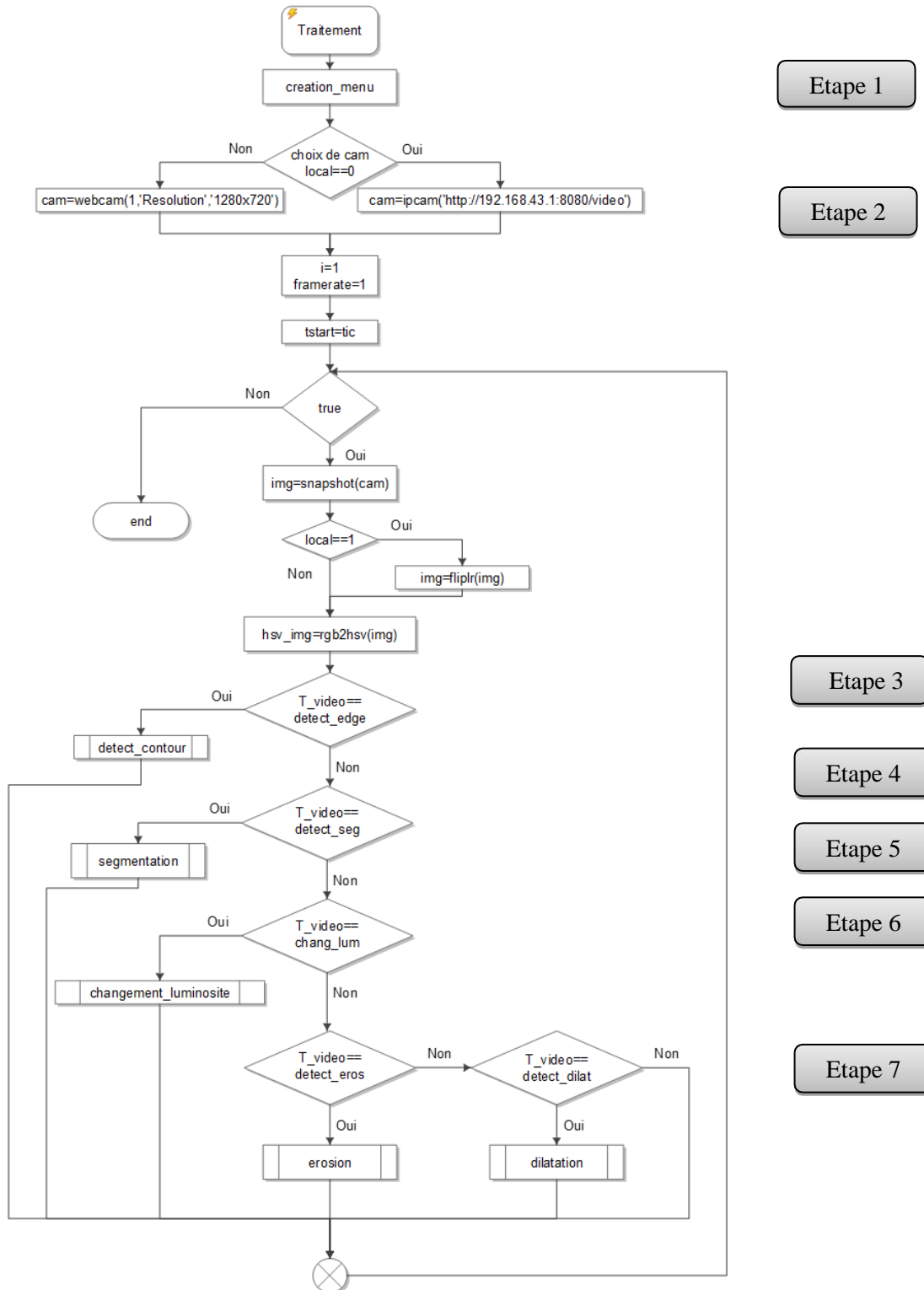
5 Ipcam

L'intégration des caméras IP avec MATLAB peut se faire à l'aide des API (interfaces de programmation d'application) disponibles pour se connecter aux caméras et récupérer les données de celles-ci. Voici quelques étapes générales pour l'intégration :

1. **Identifier la caméra** : Vous devez d'abord déterminer l'adresse IP de la caméra IP à laquelle vous souhaitez vous connecter.
2. **Utiliser l'API** : Vous pouvez utiliser des APIs telles que *ipcam* pour vous connecter à la caméra et récupérer la vidéo à partir de celle-ci. Vous pouvez utiliser les commandes spécifiques de VideoReader pour ajuster les paramètres tels que la résolution et le taux de trame.
3. **Traitement des données** : Une fois que vous avez récupéré les données vidéo de la caméra, vous pouvez les traiter à l'aide des outils de traitement d'image et des bibliothèques d'apprentissage automatique disponibles dans MATLAB, pour des tâches telles que la détection d'objets ou l'analyse visuelle.
4. **Contrôle de la caméra** : Dans certains cas, vous pouvez utiliser les API pour contrôler la caméra, telles que la capture d'images ou le réglage des paramètres.

6 Les organigrammes d'implémentations

6.1 Algorithme principale



FigureIV.3: Organigramme principale

Cet organigramme représente un processus de traitement vidéo avec différentes étapes. Voici une explication détaillée des différentes étapes du processus illustré dans l'organigramme :

Étape 1 : Création de menu

- **creation_menu** : Cette étape consiste probablement à créer une interface utilisateur ou un menu pour l'utilisateur.

Étape 2 : Choix de la caméra

- **Choix de cam** : Ici, l'utilisateur doit choisir entre une webcam locale (local=0) et une caméra IP (local=1).
- **Si local=0** : La webcam locale est utilisée avec une résolution de 1280x720.
- **Si local=1** : La caméra IP est utilisée avec l'URL spécifiée `http://192.168.43.188:8080/video`.

Initialisation des paramètres

- **j=1** : Initialisation du compteur j.
- **framerate=1** : Initialisation de la fréquence d'images à 1.
- **tstart=1** : Démarrage du temps à 1.

Étape 3 : Capture d'image

- Une boucle infinie commence (condition true).
- **img=snapshot(cam)** : Capture d'une image à partir de la caméra.
- **Local=j++** : Incrémentation du compteur local.
- **Mg=flip(img)** : Inversion de l'image (horizontalement).

Conversion de couleur

hsv_img=rgb2hsv(img) : Conversion de l'image de RGB à HSV.

Étape 4 : détection de contour

- **detected_contour** : Si la détection des contours est activée (T_video = detect_edge).
- Si oui, procéder à la segmentation.

Étape 5 : Segmentation

- **Segmentation** : Si la segmentation est activée (T_video = detect_seg).
- Si oui, procéder au changement de luminosité.

Étape 6 : Changement de luminosité

- **Changement luminosité** : Si le changement de luminosité est activé (T_video = chang_lum).
- Si oui, vérifier pour l'érosion ou la dilatation.

Étape 7 : Détection d'érosion et de dilatation

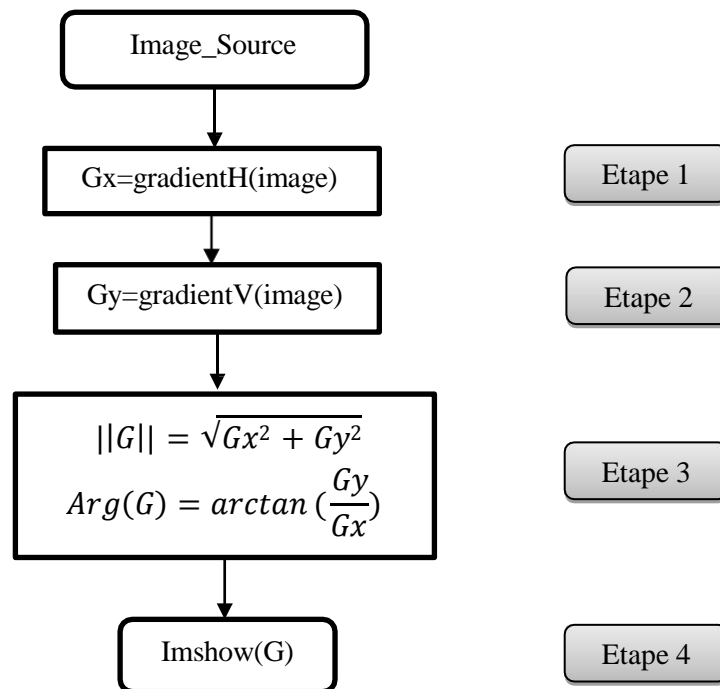
- **T_video = detect_eros** : Si l'érosion est détectée.
- Si oui, appliquer l'érosion (erosion).
- **T_video = detect_dilat** : Si la dilatation est détectée.
- Si oui, appliquer la dilatation (dilation).

Fin de la boucle

- Le processus se termine par un nœud de fin si les conditions sont remplies.

Objectif

L'organigramme montre un processus conditionnel qui traite les images capturées par une caméra, en passant par plusieurs étapes comme la conversion des couleurs, la détection des contours, la segmentation, et des ajustements de luminosité avant d'appliquer des opérations morphologiques comme l'érosion et la dilatation. Les boutons Etape 1 à Etape 6 sur le côté droit de l'organigramme semblent représenter des étapes clés ou des points de contrôle dans le processus global.

6.2 Organigramme de Détection de Contours**FigureIV.4:** Organigramme de détection de contour**➤ Sobel****Étape 1 : $Gx = \text{gradientH}(\text{image_source})$**

Objectif : Calculer le gradient horizontal (Gx) de l'image.

Description

- La convolution avec (G_x) consiste à appliquer un noyau de filtre (par exemple, le noyau Sobel ou Prewitt horizontal) sur l'image source pour détecter les variations d'intensité dans la direction horizontale.
- Le noyau de filtre met en évidence les changements rapides de luminosité qui correspondent aux contours verticaux de l'image.

Détail Technique

- Le noyau Sobel horizontal est généralement :

$$h = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

- Chaque pixel de l'image résultante (G_x) est une somme pondérée des pixels voisins dans l'image source.

Étape 2 : Convolution (G_y)

Objectif : Calculer le gradient vertical (G_y) de l'image.

Description

- La convolution avec (G_y) utilise un noyau de filtre (par exemple, le noyau Sobel ou Prewitt vertical) sur l'image source pour détecter les variations d'intensité dans la direction verticale.
- Le noyau de filtre met en évidence les changements rapides de luminosité qui correspondent aux contours horizontaux de l'image.

Détail Technique

- Le noyau Sobel vertical est généralement :

$$v = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- Chaque pixel de l'image résultante (G_y) est une somme pondérée des pixels voisins dans l'image source.

Étape 3 : Transformation de Coordonnées

Objectif : Calculer la magnitude $(\|G\|)$ et la direction $(Arg(G))$ du gradient pour chaque pixel.

Description

- Magnitude du Gradient $(\|G\|)$: Représente l'intensité du changement de luminosité à chaque pixel, ce qui indique la présence et la force d'un contour.
- Direction du Gradient $(Arg(G))$: Indique l'orientation du contour à chaque pixel.

Détail Technique

- La magnitude du gradient est calculée en utilisant :

$$||G|| = \sqrt{Gx^2 + Gy^2}$$

- La direction du gradient est calculée en utilisant :

$$Arg(G) = \arctan\left(\frac{Gy}{Gx}\right)$$

Étape 4 : Affichage

Objectif : Visualiser les résultats de la détection de contours.

Description

Après le calcul des magnitudes et des directions des gradients, l'image résultante peut être affichée pour montrer les contours détectés.

Les contours apparaissent comme des lignes claires sur un fond sombre, représentant les bords où il y a un changement significatif de luminosité.

Détail Technique

- La magnitude des gradients ($||G||$) est souvent utilisée pour créer une image binaire ou une image en niveaux de gris où les contours sont visibles.
- Les valeurs de ($||G||$) peuvent être normalisées pour être affichées correctement sur une échelle de 0 à 255.

➤ **Prewitt****Étape 1 : Convolution (Gx)**

Objectif : Calculer le gradient horizontal (Gx) de l'image.

Description :

- La convolution avec (Gx) consiste à appliquer un noyau de filtre (par exemple, le noyau Prewitt horizontal) sur l'image source pour détecter les variations d'intensité dans la direction horizontale.
- Le noyau de filtre met en évidence les changements rapides de luminosité qui correspondent aux contours verticaux de l'image.

Détail Technique

- Le noyau Prewitt horizontal est généralement :

$$h = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

- Chaque pixel de l'image résultante (Gx) est une somme pondérée des pixels voisins dans l'image source.

Étape 2 : Convolution (G_y)

Objectif : Calculer le gradient vertical (G_y) de l'image.

Description

- La convolution avec (G_y) utilise un noyau de filtre (par exemple, le noyau Prewitt vertical) sur l'image source pour détecter les variations d'intensité dans la direction verticale.
- Le noyau de filtre met en évidence les changements rapides de luminosité qui correspondent aux contours horizontaux de l'image.

Détail Technique

- Le noyau Prewitt vertical est généralement :

$$v = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- Chaque pixel de l'image résultante (G_y) est une somme pondérée des pixels voisins dans l'image source.

Étape 3 : Calculer le Gradient

Objectif : Calculer la magnitude ($\|G\|$) et la direction ($Arg(G)$) du gradient pour chaque pixel.

Description

- Magnitude du Gradient ($\|G\|$) : Représente l'intensité du changement de luminosité à chaque pixel, ce qui indique la présence et la force d'un contour.
- Direction du Gradient ($Arg(G)$) : Indique l'orientation du contour à chaque pixel.

Détail Technique

La magnitude du gradient est calculée en utilisant :

Étape 4 : Affichage

Objectif : Visualiser les résultats de la détection de contours.

Description

Après le calcul des magnitudes et des directions des gradients, l'image résultante peut être affichée pour montrer les contours détectés.

- Les contours apparaissent comme des lignes claires sur un fond sombre, représentant les bords où il y a un changement significatif de luminosité.

Détail Technique

- La magnitude des gradients ($\|G\|$) est souvent utilisée pour créer une image binaire ou une image en niveaux de gris où les contours sont visibles.
- Les valeurs de ($\|G\|$) peuvent être normalisées pour être affichées correctement sur une échelle de 0 à 255.
- **Canny :**

Étape 1 : Filtrage Gaussien

Avant de détecter les contours, l'image subit un filtrage gaussien visant à atténuer le bruit et à lisser les transitions d'intensité.

Détails Techniques

Un noyau gaussien est appliqué à l'image pour flouter les détails fins tout en préservant les contours.

Étape 2 : Calcul des gradients

Les gradients de l'image sont calculés en utilisant des opérateurs de dérivation, tels que les opérateurs de Sobel ou de Prewitt, dans les directions horizontale et verticale. Cela permet de détecter les changements d'intensité significatifs qui pourraient indiquer la présence de contours.

Détails Techniques

Les opérateurs de Sobel ou de Prewitt sont appliqués pour calculer les gradients dans les directions x et y.

Étape 3 : Suppression des non-maxima

Les pixels qui ne correspondent pas à des valeurs de gradient maximales le long de la direction du gradient sont supprimés. Cela réduit l'épaisseur des contours détectés et les rend plus précis.

Détails Techniques

Seuls les pixels ayant des valeurs de gradient maximales sont conservés, tandis que les autres sont supprimés.

Étape 4 : Seuillage par hystérésis

Un seuillage par hystérésis est appliqué pour identifier les pixels appartenant à des contours forts et à des contours faibles. Les pixels forts sont conservés, tandis que les pixels faibles sont conservés s'ils sont connectés à des pixels forts.

Détails Techniques

Deux seuils sont utilisés : un seuil haut et un seuil bas. Les pixels ayant des valeurs de gradient supérieures au seuil haut sont considérés comme des contours forts, ceux ayant des valeurs entre les deux seuils sont considérés comme des contours faibles.

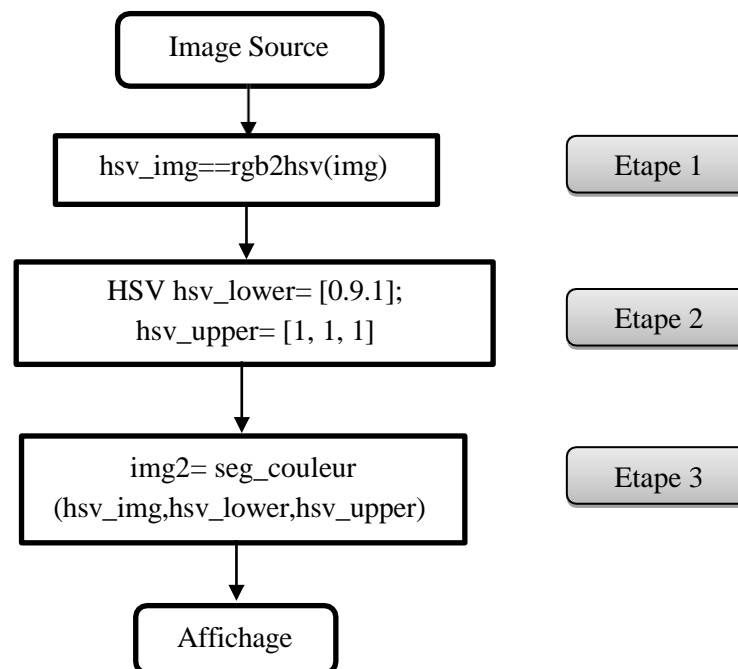
Étape 5 : Détection des contours

Les contours sont finalement détectés en utilisant les pixels conservés après le seuillage par hystérésis. Ces contours représentent les bords significatifs dans l'image.

Détails Techniques

Les contours détectés sont représentés par des pixels blancs sur un fond noir dans l'image binaire résultante.

6.3 Organigramme de Segmentation de Couleurs :



FigureIV.5: Organigramme de Segmentation de Couleurs

Étape 1 : Convertir en HSV

Transformer l'image de l'espace de couleur RGB à l'espace de couleur HSV, ce qui est plus adapté pour la segmentation basée sur les couleurs.

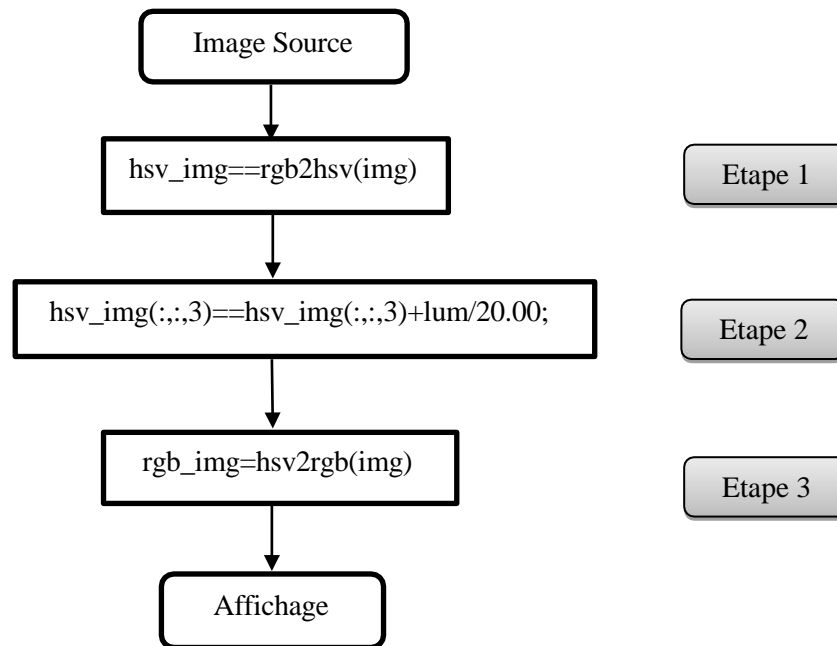
Étape 2 : Définir Plages HSV :

Déterminer les valeurs de teinte (Hue), de saturation (Saturation) et de valeur (Value) qui délimitent la plage de couleurs d'intérêt pour la segmentation.

Étape3 : Appliquer Segmentation :

- Filtrer l'image en utilisant les plages HSV définies pour isoler les pixels correspondant aux couleurs d'intérêt.
- Créer un masque binaire où les pixels correspondant aux plages définies sont blancs (valeur 1) et les autres sont noirs (valeur 0).

6.4 Organigramme d'Ajustement de la Luminosité



FigureIV.6: Organigramme d'Ajustement de la Luminosité

Etape 1 : Convertir en HSV

Une fois l'image capturée, elle est convertie de l'espace de couleur RVB (Rouge, Vert, Bleu) à l'espace de couleur HSV. L'espace de couleur HSV est souvent utilisé pour travailler avec la luminosité et la teinte de manière plus intuitive.

Etape 2 : Ajuster la luminosité

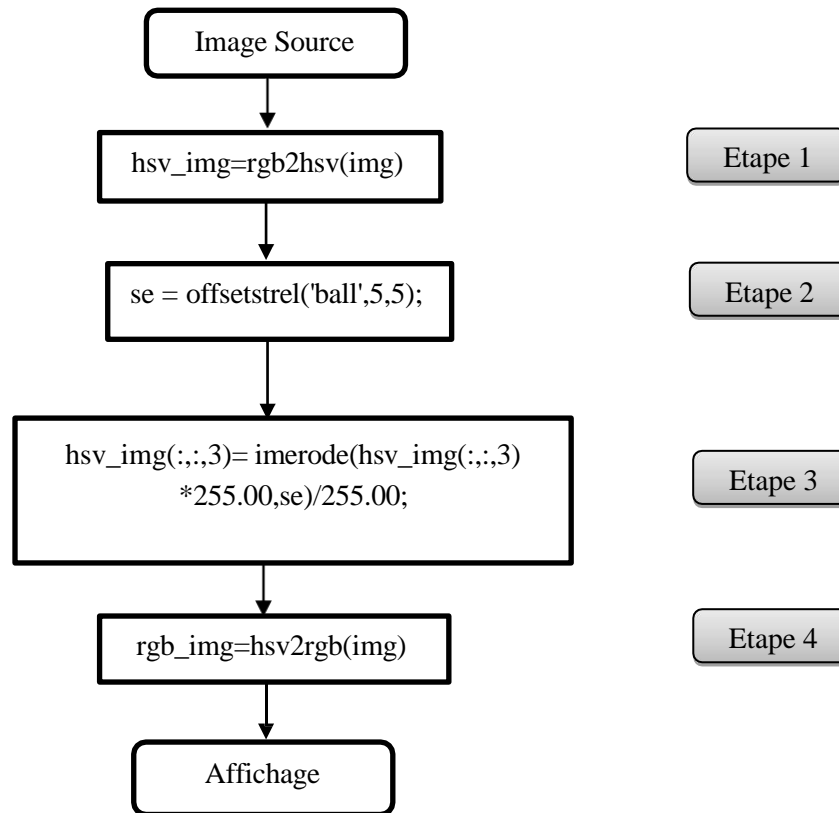
Dans cette étape, la luminosité de l'image est modifiée. Cela se fait en agissant sur le canal V (Valeur) de l'image HSV. On peut ajouter ou soustraire une valeur à ce canal pour augmenter ou diminuer la luminosité de l'image.

Etape3 : Reconvertir en RGB

Une fois que la luminosité a été ajustée dans l'espace de couleur HSV, l'image est reconvertie en RVB pour pouvoir l'afficher correctement sur un écran ou la sauvegarder dans un format image courant. Cette étape implique de convertir l'image de l'espace de couleur HSV à l'espace de couleur RVB.

6.5 Organigramme d'érosion et délitation

➤ Érosion



FigureIV.7: Organigramme d'Érosion

Étape 1 : Convertir en HSV

Une fois l'image capturée, elle est convertie de l'espace de couleur RGB (Rouge, Vert, Bleu) à l'espace de couleur HSV (Teinte, Saturation, Valeur). Cette conversion est effectuée pour faciliter le traitement de l'image en mettant en évidence les variations de teinte et de saturation, qui sont importantes dans de nombreux types de manipulations d'image.

Étape 2 : Définir Structurant

Dans cette étape, l'utilisateur définit l'élément structurant (ES) qui sera utilisé pour l'opération d'érosion. Cela implique de spécifier la forme, la taille et d'autres caractéristiques de l'élément structurant. Par exemple, l'ES peut-être une matrice carrée, une croix, un cercle, ou toute autre forme appropriée pour l'application spécifique.

Étape 3 : Appliquer Érosion

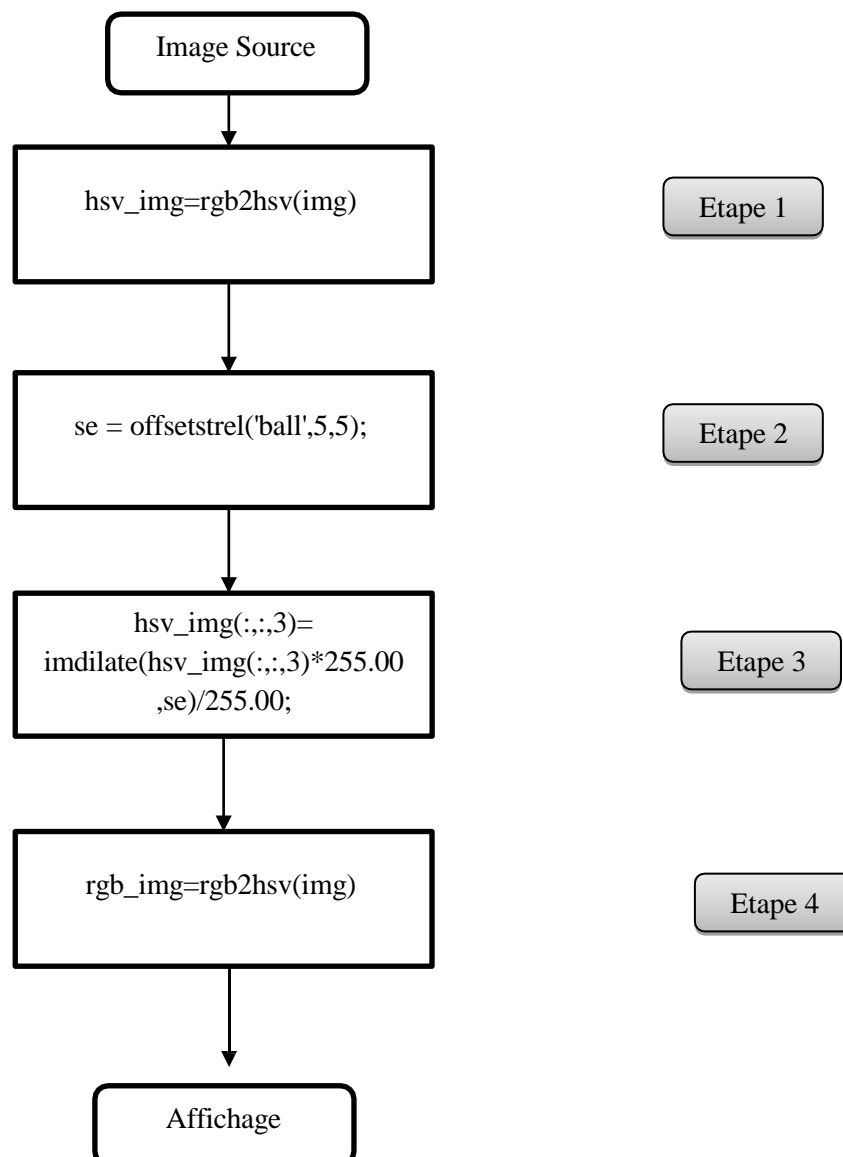
Une fois que l'ES est défini, l'opération d'érosion est appliquée à l'image. Cela implique de parcourir l'image pixel par pixel et d'appliquer l'ES à chaque pixel pour déterminer s'il doit être modifié ou non en fonction des règles d'érosion spécifiées. Typiquement, l'érosion

implique de comparer l'ES avec chaque région de l'image et de modifier les pixels en fonction de la correspondance avec l'ES.

Étape 4 : Reconvertir en RGB

Une fois que l'érosion est effectuée, l'image résultante est souvent convertie à nouveau de l'espace de couleur HSV à l'espace de couleur RGB. Cela est nécessaire pour afficher ou sauvegarder l'image dans un format compatible avec la plupart des visualiseurs ou logiciels de traitement d'images, qui utilisent généralement l'espace de couleur RGB.

➤ Délitiation :



FigureIV.8: Organigramme de délitage

Source d'image : Il s'agit de l'étape initiale où l'image d'entrée ("img") est obtenue.

Étape 1 : Convertir l'image RGB en espace colorimétrique HSV en utilisant la fonction "rgb2hsv(img)". Le résultat est stocké dans "hsv_img".

Étape 2 : Créer un élément structurant pour les opérations morphologiques. Dans ce cas, "se = offsetstrel('ball', 5, 5);" crée un élément structurant en forme de boule avec un rayon de 5.

Étape 3 : Effectuer une dilatation d'image sur le canal de valeur (le troisième canal de l'image HSV). L'opération est réalisée en utilisant la fonction "imdilate(hsv_img(:,:,3)=imdilate(hsv_img(:,:,3) * 255.00, se) / 255.00;

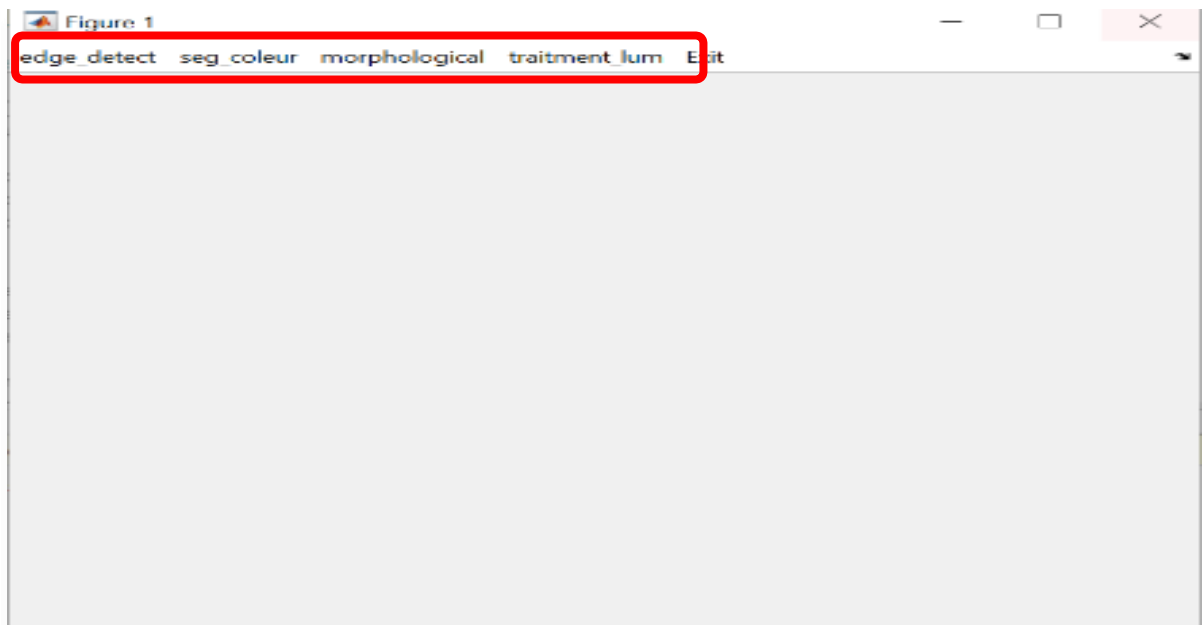
Cette ligne commence par mettre à l'échelle le canal de valeur par 255, effectue la dilatation, puis remet à l'échelle à l'inverse.

Étape 4 : Convertir l'image HSV traitée en espace colorimétrique RGB en utilisant "hsv2rgb(hsv_img)". Cette étape semble avoir une erreur dans le schéma original qui mentionne "rgb2hsv", mais la conversion correcte doit être faite avec "hsv2rgb". La fonction correcte est donc "rgb_img = hsv2rgb(hsv_img)".

Affichage : Enfin, l'image résultante est affichée.

Présentation de l'application

➤ Interface graphique



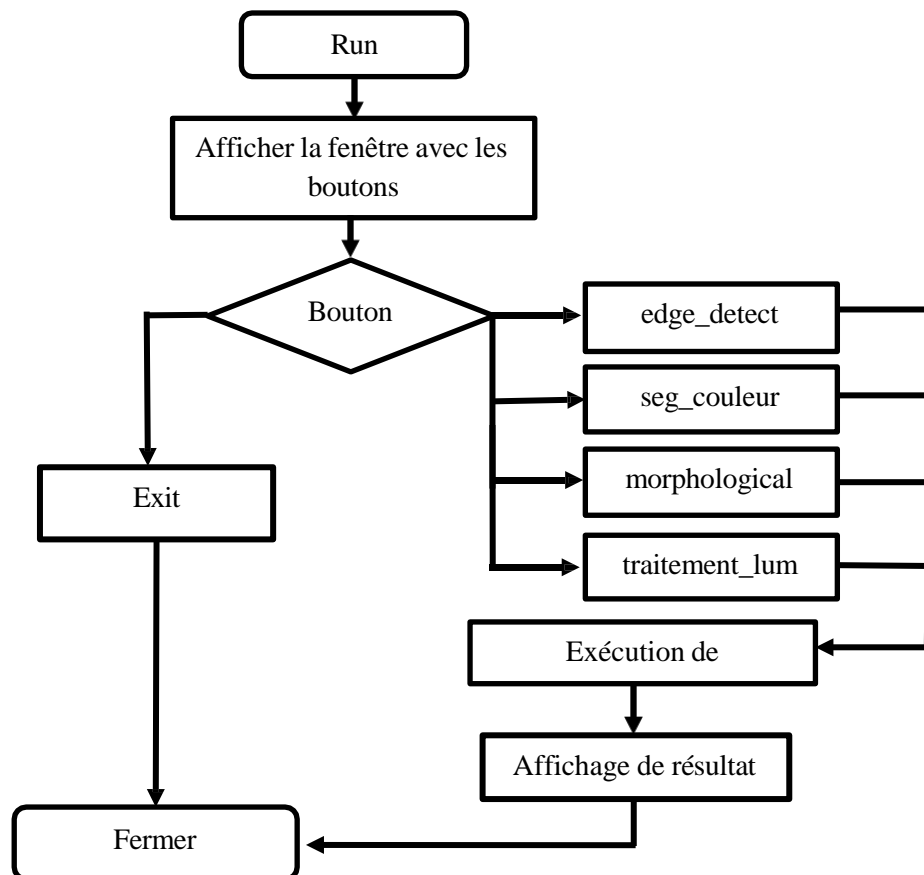
FigureIV.9: Interface graphique de l'application

L'image fournie montre une interface graphique avec plusieurs boutons dans une barre de menus. Voici une description de l'interface et de la fonction probable de chaque bouton :

- **edge_detect** : Probablement un bouton pour lancer une détection de contours sur l'image (sobel,canny,prewitt).
- **seg_couleur** : Probablement un bouton pour effectuer une segmentation basée sur la couleur.
- **Morphological** : Probablement un bouton pour appliquer des opérations morphologiques comme l'érosion ou la dilatation sur l'image.
- **traitement_lum** : Probablement un bouton pour effectuer un traitement de la luminance (ou de la luminosité) de l'image.
- **Exit** : Un bouton pour quitter l'application.

Cette interface semble être conçue pour traiter des images en utilisant diverses techniques de traitement d'image. Chaque bouton déclencherait une fonction spécifique associée à la tâche mentionnée sur le bouton.

6.6 Organigramme de cette application :

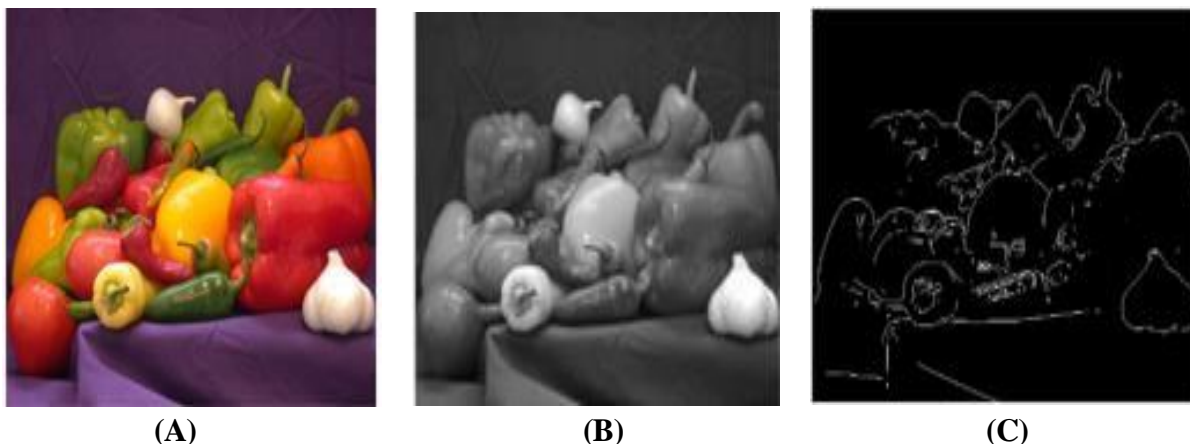


FigureIV.10: Organigramme de de cette application

7 Resultats et discussions

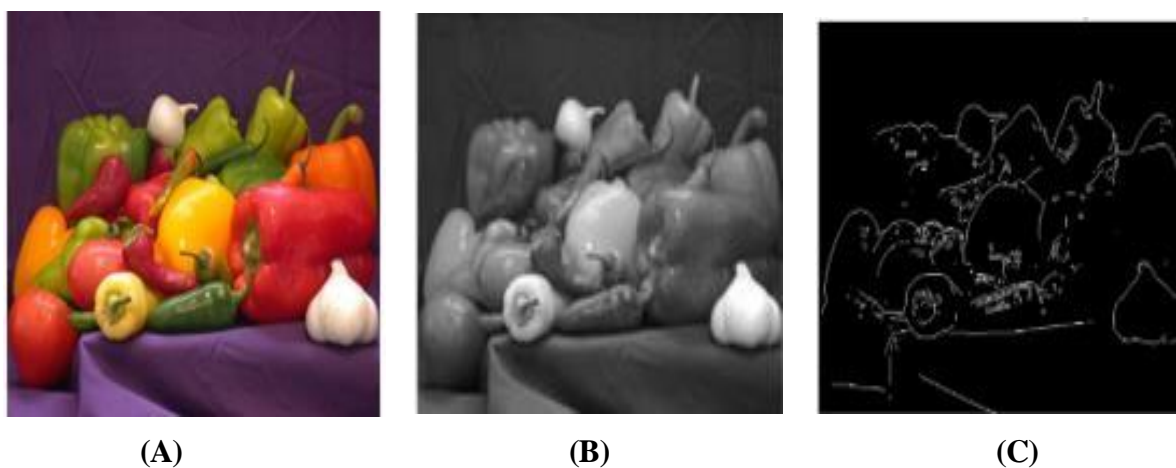
7.1 Detection des contours

7.1.1 Sobel



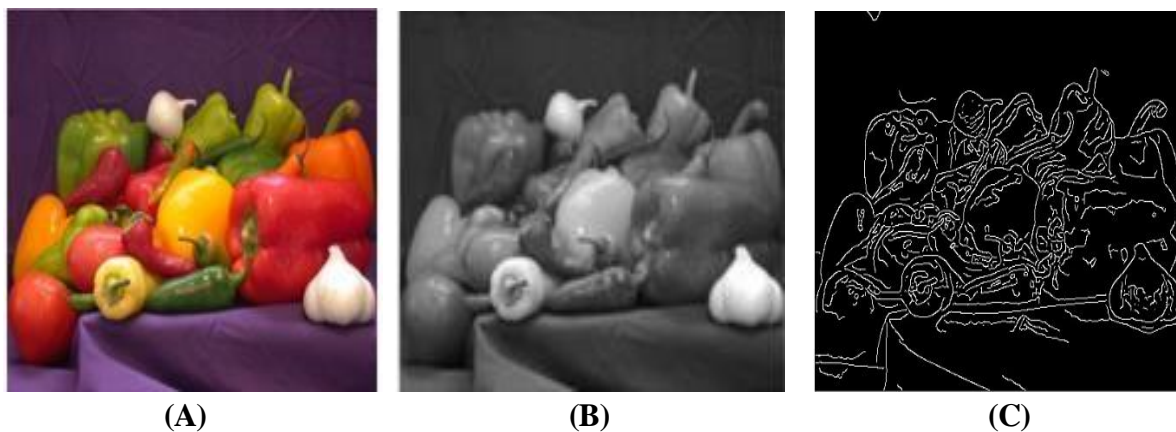
FigureIV 11: (A) image d'origine ,(B) niveaux de gris ,(C) Sobel

7.1.2 prewit



FigureIV.12: (A) image d'origine , (B) niveaux de gris ,(C) prewitt

7.1.3 Canny



FigureIV 13: (A)image d'origine , (B) niveaux de gris ,(C) Canny

La discussion

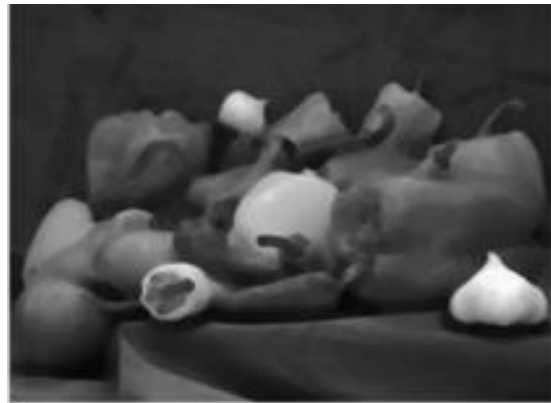
Les algorithmes Sobel, Prewitt et Canny sont utilisés pour détecter les contours dans une image. On remarque que Canny est considéré comme le plus précis. Si les contours détectés par Canny sont trop nets pour vos besoins.

7.2 Traitement morphologique

7.2.1 Érosion



(A)



(B)

FigureIV.14: (A) image d'origine, (B) érosion

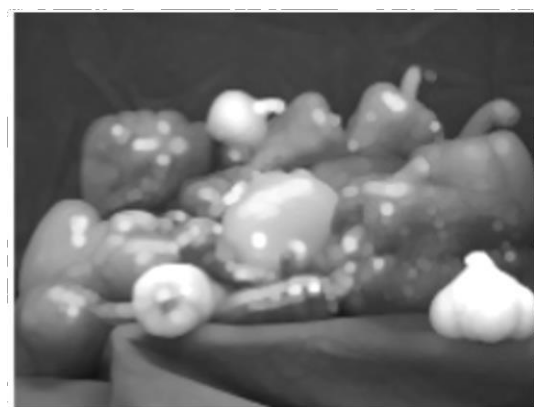
La discussion

L'érosion a efficacement réduit le bruit et affiné les contours des objets dans l'image, bien que cela se soit fait au prix de la perte de certains détails fins. Cela montre l'efficacité de l'érosion pour le nettoyage et l'amélioration de la clarté des objets principaux, mais aussi la nécessité de trouver un équilibre entre la réduction du bruit et la préservation des détails

7.2.2 Dilatation



(A)



(B)

FigureIV.15: (A) image d'origine, (B) dilatation

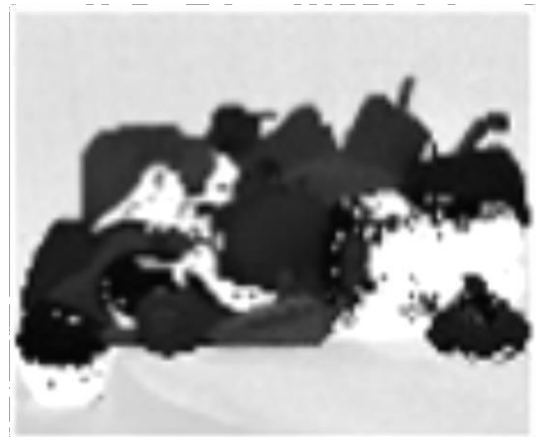
La discussion

La dilatation a efficacement agrandi les objets et comblé les petites lacunes, améliorant ainsi la continuité et la visibilité des formes principales. Cependant, cela s'accompagne d'une Figure IV.16 augmentation de l'épaisseur des contours et d'une possible perte de détails fins. Ces résultats montrent l'utilité de la dilatation pour renforcer et connecter les structures dans une image, tout en nécessitant une attention particulière pour préserver les détails importants.

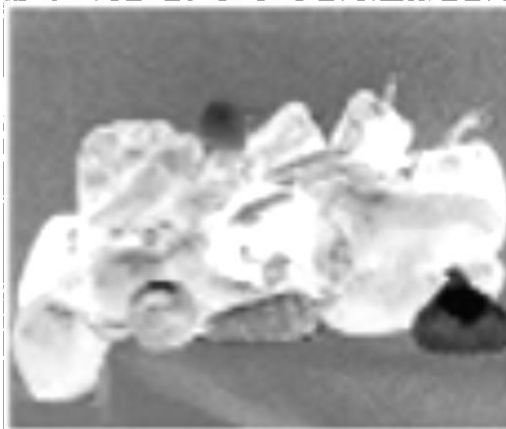
7.3 HSV



(A)



(B)



(C)



(D)

FigureIV.16: (A) image d'origine, (B) composante H(hue), (C) composante S(saturation), (D) composante V(value)

La discussion

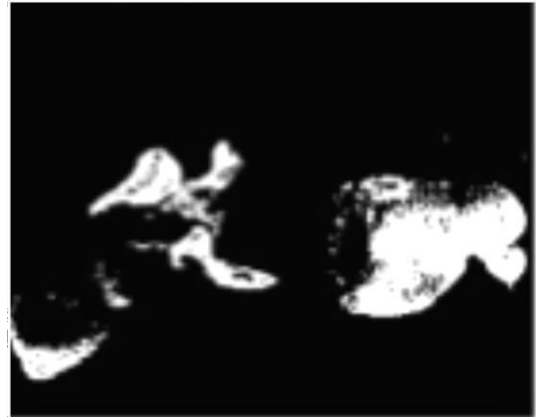
La modification de la composante V (Valeur) dans l'espace de couleur HSV a donné de meilleurs résultats que la modification des composantes S (Saturation) et H (Teinte). En ajustant la Valeur, nous avons pu améliorer le contraste et la visibilité des objets sans altérer leurs couleurs originales. Cette approche est particulièrement efficace pour les applications de

détection et de segmentation d'objets où la préservation des détails et la cohérence des couleurs sont essentielles.

7.4 Segmentation



(A)

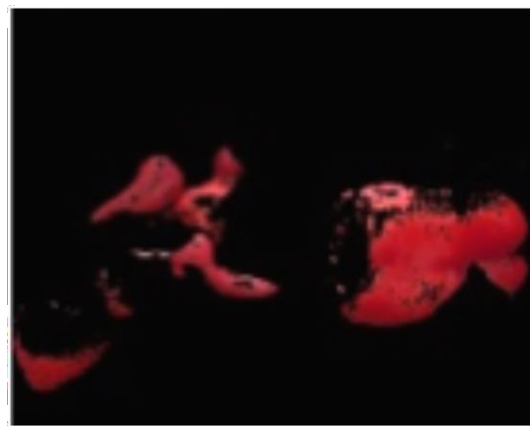


(B)

FigureIV.17: (A) image d'origine , (B) image segmentation pour les poivrons rouge(couleur blanc)



(A)



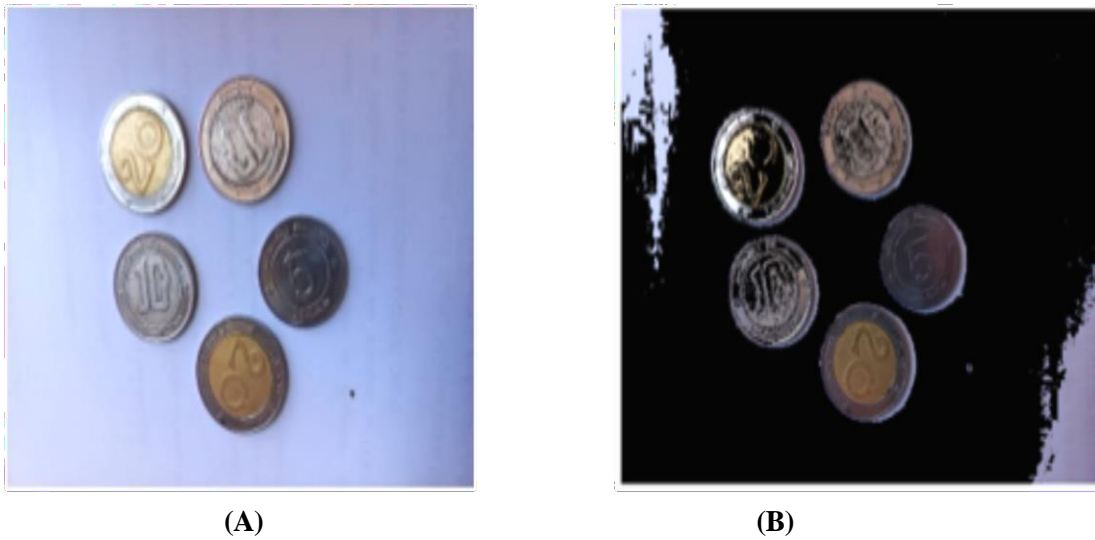
(B)

FigureIV.18: (A) image d'origine, (B) image segmentation pour les poivrons rouge (couleur rouge)

La discussion

Le résultat de la segmentation des poivrons rouges n'a pas été satisfaisant malgré les ajustements apportés aux composantes de couleur. Cela souligne la complexité de la tâche et la nécessité de méthodes plus sophistiquées pour une segmentation efficace. Il serait bénéfique d'explorer des techniques de prétraitement supplémentaires pour réduire le bruit et des approches avancées de segmentation basées sur l'apprentissage pour améliorer la précision et la clarté des résultats.

7.5 Surprissions de l'arrière-plan en temps réelle :

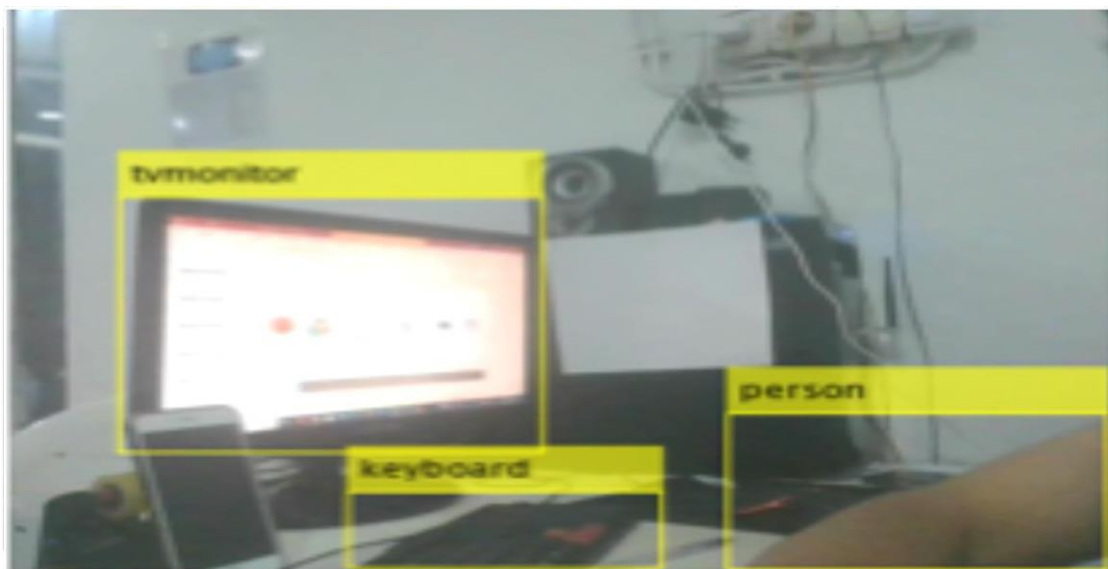


FigureIV.19: (A) image d'origine, (B) l'arrière-plan supprimé

La discussion

Les résultats montrent une suppression efficace de l'arrière-plan, mettant en valeur les pièces de monnaie de manière plus claire et détaillée. Cependant, quelques artefacts noirs restent visibles, ce qui indique que des améliorations mineures pourraient encore être apportées pour perfectionner la technique.

7.6 YOLOv3 en temps réelle



FigureIV.20: Utilisation de YOLOv3 avec ipcam

La discussion

L'image montre l'utilisation de YOLOv3 en temps réel pour la détection d'objets via une caméra IP. Les objets détectés, comme "tvmonitor," "keyboard," et "person," sont encadrés en jaune avec des étiquettes appropriées. Les résultats démontrent l'efficacité de YOLOv3 pour identifier et classer divers objets en temps réel avec une bonne précision, ce qui est prometteur pour des applications de surveillance et d'automatisation.

8 Conclusion

Ce chapitre a offert une vue d'ensemble pratique de l'implémentation des concepts théoriques discutés dans les chapitres précédents. En présentant le cadre de développement et les environnements logiciels utilisés, nous avons mis en lumière les outils essentiels pour le traitement d'images ainsi que la détection et le suivi d'objets. Chaque étape de l'implémentation a été détaillée, depuis le prétraitement des données visuelles jusqu'à l'intégration des algorithmes, en soulignant les défis rencontrés et les solutions adoptées. L'évaluation des performances a révélé que notre approche est efficace en termes de précision, de rapidité et de robustesse, alignant ainsi les résultats pratiques avec les attentes théoriques. De plus, nous avons inclus des discussions approfondies sur les différents résultats obtenus, ce qui a permis de mieux comprendre les implications et les nuances de notre approche. Ce chapitre a démontré la faisabilité et l'efficacité des techniques de traitement d'images et de détection/suivi d'objets dans des applications réelles, consolidant ainsi les bases théoriques et offrant une compréhension approfondie de leur applicabilité pratique.

Conclusion Générale

Conclusion Générale

La détection d'objets dans les images occupe une place centrale dans le paysage technologique moderne, façonnant des domaines aussi variés que la surveillance vidéo, la médecine et la robotique. Ce mémoire a scruté les multiples dimensions de ce domaine, des théories fondamentales aux applications pratiques, mettant en lumière ses défis actuels et ses perspectives futures. L'exploration des différentes techniques de détection d'objets, qu'elles soient traditionnelles ou basées sur l'apprentissage en profondeur, a révélé la complexité et la richesse de cette discipline.

En soulignant les applications concrètes de la détection d'objets et en examinant les obstacles rencontrés lors de leur mise en œuvre, nous avons souligné l'importance cruciale de la recherche continue dans ce domaine dynamique. Bien que des progrès significatifs aient été réalisés, des défis persistent, notamment en ce qui concerne la précision des algorithmes et la gestion des volumes de données. Cependant, ces défis offrent également des opportunités d'innovation, notamment dans les domaines de l'intelligence artificielle et de l'apprentissage automatique.

Cette étude a mis en évidence l'importance vitale de la détection d'objets dans les images et les avancées substantielles qui ont été réalisées. En orientant le regard vers l'avenir, en soulignant les perspectives de recherche et les opportunités de développement, nous aspirons à encourager une exploration continue et approfondie de ce domaine. Notre objectif ultime est de contribuer à l'avancement global de la science et de la technologie, dans le but de servir au mieux les besoins de la société moderne.

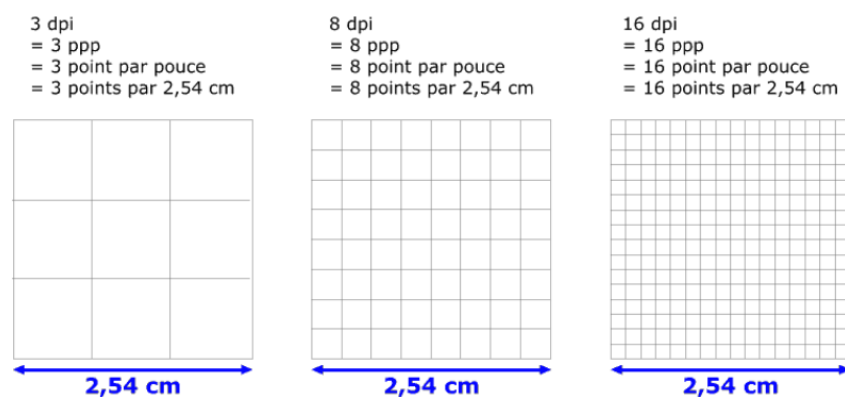
Annexe A

A.1 Résolution

Tout d'abord, selon le Grand dictionnaire terminologique, la résolution est le « nombre de points ou de pixels, par unité de longueur, que peut comporter une image reproduite sur un écran ou une feuille de papier. »

En résumé, la résolution est le nombre de points ou de pixels qui peuvent être insérés sur une longueur d'un pouce, que ce soit sur un écran ou une page imprimée. Les unités de mesure de la résolution sont DPI et PPI (ou PPP). [57]

- 1. DPI (dots per inch)** est une mesure de la résolution d'impression, indiquant le nombre de points d'encre par pouce imprimés sur un support physique comme du papier. [58] [57]
- 2. PPP (pixels par pouce)** est une mesure de la résolution d'affichage, indiquant le nombre de pixels par pouce sur un écran d'ordinateur ou un autre dispositif d'affichage numérique. En résumé :
 - DPI s'applique à l'impression, PPP s'applique à l'affichage numérique
 - DPI indique la densité des points d'encre, PPP indique la densité des pixels
 - Une résolution d'impression élevée en DPI (300 DPI ou plus) permet une meilleure qualité d'impression
 - Une résolution d'affichage élevée en PPP (300 PPI ou plus) permet une meilleure qualité d'image à l'écran
 - Bien que les termes soient parfois utilisés de manière interchangeable, il est important de faire la distinction entre DPI et PPP pour comprendre la résolution d'une image ou d'un document. [57]



A.2 Les codecs

Les codecs vidéo sont des algorithmes logiciels ou matériels qui permettent de compresser et décompresser les données vidéo [59] [60] [61]. Ils sont essentiels pour le stockage, la transmission et la lecture des vidéos numériques.

Quelques caractéristiques des codecs vidéo :

- Ils éliminent les données redondantes à l'intérieur des images et entre les images successives [59].
- Ils permettent de réduire considérablement la taille des fichiers vidéo tout en préservant une qualité acceptable [60].
- Il existe des codecs propriétaires (H.264, HEVC) et des codecs libres (VP9, AV1) [61].
- Les codecs récents comme HEVC permettent de diviser par deux la taille des fichiers par rapport à H.264 pour une même qualité [59].
- Le choix du codec dépend des priorités entre qualité, taille de fichier et compatibilité avec les appareils [60].

Les principales étapes du traitement vidéo avec les codecs sont :

1. Encodage : Compression des données vidéo brutes dans un format codé
2. Transmission/Stockage : Envoi ou sauvegarde du fichier vidéo codé
3. Décodage : Décompression du fichier codé pour afficher la vidéo

En résumé, les codecs vidéo jouent un rôle essentiel en permettant de stocker, transmettre et lire efficacement les contenus vidéo numériques, grâce à leurs algorithmes de compression performants [59] [60] [61].

A.2.1 L'encodage vidéo

L'encodage vidéo, également appelé conversion vidéo, est le processus de transformation d'une vidéo brute en un format numérique compatible avec divers lecteurs web et appareils mobiles. [60] Il permet la lecture de fichiers et de flux vidéo en déterminant la configuration d'un codec dans un conteneur. Les formats d'encodage vidéo les plus populaires sont :

- **MP4** : Le format MP4 (MPEG-4 Part 14), défini par le groupe MPEG, est un conteneur multimédia capable de stocker des vidéos, audios, sous-titres, images et métadonnées. [62] [63]

Ce format est largement utilisé pour partager des vidéos sur Internet et est compatible avec de nombreux appareils, y compris les appareils mobiles comme l'iPhone, l'iPad et l'iPod. [62]

[63] Il est idéal pour les vidéos asynchrones ou les flux enregistrés et est couramment utilisé par les sites de streaming vidéo [62]. Le MP4 est devenu le standard pour le flux de travail des médias numériques, de plus en plus important pour les appareils portables et les nouvelles applications multimédias [64] [62], offrant une haute compatibilité et une qualité audiovisuelle élevée. C'est un choix populaire pour le partage en ligne de vidéos et d'autres contenus multimédias [63].

- **MOV** : Le format MOV (QuickTime Movie), créé par Apple Inc. en 1998, est un conteneur multimédia utilisé principalement sur les appareils iOS et macOS. Il peut stocker des données vidéo, audio, texte et de réalité virtuelle, ce qui le rend idéal pour le stockage et la manipulation de contenus multimédias. Utilisant le codec de compression MPEG-4, le format MOV est compatible avec de nombreux lecteurs multimédias et outils de montage vidéo, ainsi que les systèmes d'exploitation Macintosh et Windows. Cette polyvalence en fait un format populaire pour le montage vidéo professionnel et la transmission de fichiers sur Internet, grâce à sa capacité à gérer séparément les différents éléments qu'il contient. En somme, le MOV est un format de fichier conteneur reconnu pour sa qualité et son adaptabilité sur diverses plateformes [63] [65] [66].

- **MKV** : Le format MKV (Matroska Video) est un conteneur multimédia open source et gratuit capable de stocker de la vidéo, de l'audio, des sous-titres et d'autres données dans un seul fichier. [67] [65] [63] Contrairement à des formats comme AVI ou MP4, il peut contenir plusieurs pistes vidéo, audio et de sous-titres, ce qui le rend très flexible et adapté aux films multilingues avec plusieurs options de sous-titres [67] [65].

Voici quelques caractéristiques clés du format MKV :

- Support de nombreux codecs vidéo et audio (H.264, H.265, MP3, AC3, etc.)
- Inclusion de chapitres, de menus et de métadonnées
- Compatibilité avec le streaming en ligne
- Résistance aux erreurs et récupération des fichiers corrompus
- Sélection de flux vidéo, audio et de sous-titres multiples

Bien que très polyvalent, le format MKV n'est pas toujours pris en charge nativement par tous les lecteurs et appareils, nécessitant parfois des lecteurs tiers comme VLC. Néanmoins, il reste un choix populaire pour le stockage et la lecture de contenus multimédias complexes [65] [63].

- **AVI** : Le format AVI (Audio Video Interleave), développé par Microsoft, est un conteneur multimédia permettant de regrouper des flux audio et vidéo dans un seul fichier portant l'extension ".avi". Il peut contenir un flux vidéo et jusqu'à 99 flux audio.

AVI est souvent utilisé avec les codecs de compression vidéo DivX et XviD, et est compatible avec certains lecteurs DVD, Blu-ray, disques durs multimédia, home cinéma, et baladeurs vidéo. Cependant, pour lire un fichier AVI, le lecteur doit disposer des codecs correspondants aux flux audio et vidéo encapsulés. Ainsi, un ordinateur sans les bons codecs ne pourra pas lire tous les fichiers AVI.

En résumé, AVI est un format conteneur pratique pour combiner vidéo et audio dans un seul fichier, mais sa lecture dépend de la disponibilité des codecs nécessaires [68] [69].

- **WebM** : WebM est un format conteneur multimédia open source conçue principalement pour le web. Basé sur le conteneur Matroska, il utilise les codecs vidéo VP8 et VP9 et les codecs audio Vorbis et Opus. [70]

Caractéristiques clés du format WebM :

- Extension de fichier "webm"
- Vidéo compressée en VP8 ou VP9
- Audio compressé en Vorbis ou Opus
- Conçu pour HTML5
- Alternative open source aux formats MP4/H.264
- Support natif dans Chrome, Firefox, et Opera
- Nécessite un plugin pour Internet Explorer et un codec tiers pour Safari

WebM est couramment utilisé par YouTube pour les vidéos haute définition et est idéal pour le streaming en ligne en raison de ses codecs efficaces et de l'absence de redevances.

En résumé, WebM est un format conteneur open source utilisant VP8/VP9 et Vorbis/Opus, conçu pour la vidéo en ligne et supporté par les principaux navigateurs [70] [71].

- **MPEG-2** : MPEG-2, développé par le groupe d'experts MPEG en 1994, est une norme de compression vidéo et audio succédant à MPEG-1. Elle est essentielle pour la compression et le transport des images et du son dans la télévision numérique.

Caractéristiques principales de MPEG-2 :

- Compression d'image et de son, avec transport via réseaux pour la télévision numérique
- Prend en charge la compression vidéo entrelacée, courante en télévision analogique
- Moins efficace à bas débit, surtout en dessous de 1 Mbit/s en définition standard

- Utilisé pour les DVD et la diffusion de télévision numérique (satellite, câble, réseau, TNT)
- Contient des brevets logiciels gérés par le consortium MPEG LA

MPEG-2 comprend plusieurs parties normatives :

- Part 1 (Systèmes) : formats de transport et de programme
- Part 2 (Vidéo) : compression vidéo
- Part 3 (Audio) : compression audio

En résumé, MPEG-2 est une norme de compression vidéo et audio largement utilisée pour les DVD et la télévision numérique, malgré l'émergence de normes plus récentes comme H.264 et H.265 [61].

Le format H.264 (MPEG-4 AVC) et le format H.265 (HEVC) sont deux normes de compression vidéo très utilisées dans l'industrie du multimédia [72]. Voici un résumé des différences clés entre ces deux formats :

H.264 : H.264 (MPEG-4 AVC)

- Développé conjointement par l'UIT-T et le MPEG
- Norme de compression vidéo très répandue, utilisée dans de nombreux appareils et plateformes
- Résolution maximale : 4096×2304
- HDR : *HLG*
- FPS max : $4K/59,97FPS$
- Qualité/poids : $10 \text{ min } 1080p \text{ à } 20 \text{ Mbs} / 1,5Go$ [72] [73]

H.265 : H.265 (HEVC)

- Développé pour offrir une compression plus efficace que H.264
- Résolution maximale : $8192/4320$
- HDR : *HDR10*
- FPS max : $8K/300FPS$
- Qualité/poids : $10 \text{ min } UHD \text{ à } 20 \text{ Mbs} / 1,5Go$

En résumé, H.265 offre une résolution et un débit d'images plus élevés que H.264, ainsi qu'une meilleure gestion de la HDR et des FPS. Cependant, H.264 est toujours très répandu et compatible avec de nombreux appareils et plateformes [72] [73].

Chaque format est composé d'un codec audio, d'un codec vidéo et d'un conteneur. Les codecs utilisent la compression avec perte ou sans perte pour réduire la taille des fichiers. L'encodage peut se faire en temps réel, comme pour les diffusions en direct, ou en plusieurs passes pour optimiser la qualité et le débit. Le décodage est effectué par les lecteurs logiciels ou matériels pour afficher la vidéo. En résumé, l'encodage est essentiel pour stocker, transmettre et visualiser les données vidéo de manière efficace et compatible sur différentes plateformes. Le choix du format dépend des priorités entre qualité, taille de fichier et compatibilité.

A.2.2 Transmission/Stockage :

Transmission

La transmission vidéo se fait généralement de trois façons [74] [75] :

- Wi-Fi dans le port : pour transmettre les séquences vidéo brutes lors du retour au port
- Données mobiles : pour une transmission proche du temps réel pendant la pêche
- Satellite : pour les zones sans couverture mobile, mais avec un débit plus limité

Les codecs vidéo comme H.264 permettent de compresser les données pour optimiser la transmission. Le choix du mode dépend de la couverture réseau, du débit disponible et de la latence acceptable [74] [75] .

Stockage

Le stockage des données vidéo peut se faire de plusieurs manières :

- Disque dur local : pour un stockage préliminaire sur le navire
- Serveur local : pour un stockage à plus long terme à terre
- Serveur dans le cloud : pour un stockage centralisé et sécurisé

Les organismes de gestion des pêches (ORGP) doivent définir des normes sur la durée de conservation des vidéos après examen. Cela dépend des objectifs du programme et des besoins d'accès aux enregistrements.

Le stockage local sur disque dur est plus coûteux en maintenance et vulnérable aux pannes. Le cloud offre plus de sécurité et de fiabilité mais nécessite une bonne connectivité [74].

En résumé, la transmission vidéo utilise le Wi-Fi, les données mobiles ou le satellite selon le contexte. Le stockage se fait sur disque dur local, serveur ou cloud en fonction des besoins de conservation et de sécurité. Les codecs optimisent la transmission et le stockage.

A.2.3 Décodage

Le processus de décodage vidéo est essentiel pour lire et afficher des fichiers vidéo compressés. Voici comment il fonctionne :

- 1. Entrée du flux vidéo :** Le décodeur reçoit un flux vidéo compressé, généralement sous forme de bits, provenant d'un fichier vidéo ou d'une source en direct.
- 2. Démultiplexage :** Le flux vidéo est démultiplexé pour extraire les données vidéo des autres composants du conteneur multimédia, tels que l'audio, les sous-titres, etc.
- 3. Décompression :** Les données vidéo compressées sont décompressées à l'aide des algorithmes de décodage appropriés. Cela implique d'inverser les processus de compression appliqués lors de l'encodage, tels que la transformation en cosinus discrète (DCT) ou la prédiction de mouvement.
- 4. Reconstruction des images :** Les images vidéo sont reconstruites à partir des données décompressées. Les images sont assemblées en séquences pour recréer le flux vidéo original.
- 5. Affichage des images :** Les images décodées sont ensuite affichées sur l'écran, permettant aux utilisateurs de visualiser la vidéo [76] [77].

Le décodage vidéo peut être effectué par des logiciels décodeurs ou des décodeurs matériels spécialisés. Les décodeurs doivent être compatibles avec le format de compression utilisé dans le fichier vidéo pour garantir une lecture correcte et fluide.

En résumé, le processus de décodage vidéo consiste à extraire, décompresser et reconstruire les données vidéo compressées pour afficher la vidéo de manière compréhensible et fluide pour les utilisateurs [77] [78].

A.3 Le débit de compression vidéo,

Le débit de compression vidéo, également appelé débit binaire, est crucial dans le processus de compression des données vidéo. Il représente la quantité d'informations vidéo traitées par seconde pendant la lecture [5] [6] [79]. Un débit binaire plus élevé correspond généralement à une meilleure qualité d'image, mais aussi à une taille de fichier plus importante [79].

Voici quelques points clés sur le débit de compression vidéo basés sur les sources fournies :

- 1. Réglage du débit binaire :** Certains codecs vidéo permettent de définir le débit binaire, qui varie selon le contenu visuel de chaque image [5].

2. Optimisation du débit : Le débit binaire doit être ajusté en fonction de la finalité de la vidéo, que ce soit pour la production de DVD, la lecture sur disque dur, la lecture sur CD-ROM, la lecture dans un intranet ou la vidéo en continu sur le web [5] [6] [80].

3. Débit binaire variable : Les codecs récents utilisent le débit binaire variable (VBR) pour ajuster automatiquement le débit en fonction du contenu de l'image, offrant une meilleure qualité d'image avec des tailles de fichiers relativement faibles [79].

4. Débit binaire constant : Il est possible d'utiliser un débit binaire constant (CBR) pour contrôler précisément la taille du fichier final, mais le débit binaire variable est généralement préféré pour une meilleure qualité d'image [79].

En résumé, le débit de compression vidéo joue un rôle essentiel dans la qualité et la taille des fichiers vidéo, et son réglage approprié est crucial pour obtenir le bon équilibre entre qualité visuelle et taille de fichier [5] [6] [79] [80].

Annexe B

B.1 Mean shift

Mean Shift est une technique d'analyse non paramétrique utilisée pour localiser les maxima d'une fonction de densité dans l'espace des caractéristiques [81]. Cet algorithme itératif recherche les modes (ou maxima locaux) de la fonction de densité.

Les étapes principales de l'algorithme Mean Shift sont les suivantes [81] :

1. Définir une fonction noyau $K(x_i - x)$ pour déterminer le poids des points voisins dans le calcul de la moyenne.
 2. Calculer la moyenne pondérée $m(x)$ de la densité autour de l'estimation actuelle x :
- $$m(x) = \frac{\sum K(x_i - x) \times x_i}{\sum K(x_i - x)}$$
3. Déplacer l'estimation actuelle x vers la nouvelle moyenne $m(x)$.
 4. Répéter les étapes 2 et 3 jusqu'à ce que l'estimation x converge.

Les points finaux convergés représentent les modes ou maxima locaux de la fonction de densité sous-jacente [82] [81]. Cela rend Mean Shift particulièrement utile pour le clustering de données sans nécessiter la spécification préalable du nombre de clusters [83].

Mean Shift est largement utilisé en vision par ordinateur pour des tâches telles que la segmentation d'images basée sur les niveaux de couleur ou de texture, la détection d'objets, l'amélioration d'images et le suivi d'objets en mouvement. Il est également appliqué en analyse de données pour segmenter des cartes, détecter des clusters d'activité humaine, segmenter des clients en marketing et identifier des communautés sur les réseaux sociaux [83].

En résumé, Mean Shift est une méthode itérative et non paramétrique de recherche des modes qui déplace les points de données vers la moyenne de leurs voisins, convergeant vers les maxima locaux de la fonction de densité sous-jacente [81] [82] [83]. Cette approche puissante et flexible est applicable dans de nombreux domaines.

B.2 Darknet 53

1. Architecture : Darknet-53 est un réseau convolutionnel à 53 couches, spécialement conçu pour YOLOv3. Il se compose de couches convolutionnelles suivies de normalisation par batch et d'activation Leaky ReLU [82] [84] [85].

2. Convolutions : Les couches convolutionnelles utilisent des filtres de taille 3x3 avec des poids initialisés aléatoirement. Ces convolutions sont suivies de normalisation par batch pour minimiser l'effet de saturation des poids [82] [84].

3. Connexions Résiduelles : Darknet-53 intègre des connexions résiduelles pour améliorer la propagation des informations dans le réseau. Cela aide à réduire les erreurs d'apprentissage et à augmenter la précision du modèle [82] [84].

4. Taille d'Entrée de l'Image : Les images d'entrée sont de 256x256 pixels, ce qui permet de réduire les coûts de calcul tout en améliorant la précision du modèle [84].

5. Pré-entraînement : Darknet-53 est pré-entraîné sur le jeu de données ImageNet, qui comprend plus d'un million d'images. Cela permet de développer des représentations de caractéristiques riches pour une grande variété d'images [84].

6. Utilisation : Darknet-53 est fréquemment utilisé comme base pour les tâches de détection d'objets et dans les workflows YOLO. Il est adapté à des applications telles que la détection d'objets en temps réel, la reconnaissance d'images et la surveillance [84].

En résumé, l'architecture Darknet-53 est un réseau convolutionnel à 53 couches conçu pour YOLOv3, intégrant des connexions résiduelles, des normalisations par batch et des activations Leaky ReLU. Pré-entraîné sur ImageNet, il est largement utilisé pour des applications de détection d'objets en temps réel et de reconnaissance d'images [82] [84] [85].

B.3 Le mean Average Precision (mAP)

Le mean Average Precision (mAP) est une métrique couramment utilisée pour évaluer la performance des modèles de détection d'objets. Elle combine la précision et le rappel pour fournir une mesure globale de l'efficacité d'un modèle de détection. Voici une explication détaillée :

1. Précision et Rappel :

- **Précision :** La précision mesure la proportion de détections correctes parmi toutes les détections effectuées par le modèle. Elle est calculée comme suit :

$$\text{Précision} = \frac{\text{vrais positifs (VP)}}{\text{vrais positifs (VP)} + \text{faux positifs (FP)}}$$

- **Rappel :** Le rappel mesure la proportion de vraies instances détectées parmi toutes les instances réelles dans les données. Il est calculé comme suit :

$$\text{Précision} = \frac{\text{vrais positifs (VP)}}{\text{vrais positifs (VP)} + \text{faux négatifs (FN)}}$$

2. Courbe Précision-Rappel :

La courbe précision-rappel est tracée en variant le seuil de confiance pour les détections. Elle montre comment la précision et le rappel changent en fonction du seuil.

3. Average Precision (AP) :

Pour une classe donnée, l'AP est calculée en prenant la moyenne des précisions à différents niveaux de rappel. Il est souvent approximé par l'aire sous la courbe précision-rappel (AUC-PR).

Formule simplifiée pour l'AP :

$$AP = \int_0^1 p(r) dr$$

Où $p(r)$ est la précision en fonction du rappel (r)

4. mean Average Precision (mAP) :

Le mAP est la moyenne des APs calculées pour chaque classe. Si un modèle détecte plusieurs classes, l'AP est calculée pour chaque classe individuellement, et le mAP est la moyenne de ces APs.

Formule pour le mAP :

$$mAP = \frac{1}{N} \sum_{i=0}^N AP_i$$

Où N est le nombre de classes et AP_i est l'AP pour la i -ème classe.

5. Utilisation en Pratique :

Le mAP est utilisé pour comparer la performance des modèles de détection d'objets. Un mAP élevé indique que le modèle a une bonne capacité à détecter et localiser les objets avec précision.

Dans les compétitions et benchmarks, le mAP est une métrique standard pour évaluer et comparer différents modèles de détection.

En résumé, le mean Average Precision (mAP) est une mesure complète de la performance des modèles de détection d'objets, intégrant à la fois la précision et le rappel pour évaluer l'efficacité globale de la détection à travers différentes classes d'objets.

Annexe C

C.1 Gradient H :

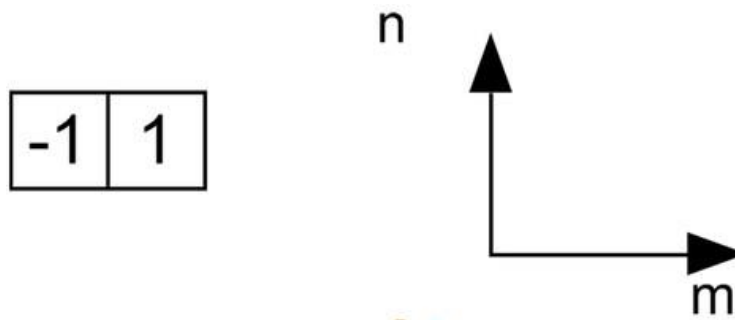
Ces opérateurs sont à considérer comme des filtres qui seront corrélés à l'image. Les réponses impulsionnelles de ces filtres peuvent se présenter sous forme de fonctions analytiques souvent d'une seule variable ou bien sous forme de masques bidimensionnels. Dans les deux cas, le filtrage se déroule en deux étapes : un filtrage suivant les lignes de l'image puis suivant les colonnes dans le cas d'une réponse impulsionnelle monodimensionnelle du filtre, ou une corrélation bidimensionnelle de l'image avec deux masques modélisant deux contours dans des directions orthogonales dans l'autre cas. [86]

Opérateurs de Gradient par Masques

Introduction

Pour chaque opérateur, deux masques sont utilisés afin de déterminer le gradient de l'image dans deux directions orthogonales.

Approximation de Base



Le masque le plus intuitif à mettre en œuvre est un masque à deux éléments :

Tableau 1

-1	1
----	---

L'origine du masque est le point -1. Dans le repère ci-dessus, la réponse impulsionnelle $h(m, n)$ du masque est définie par :

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

La corrélation de ce masque avec une image de luminance $f(i, j)$ s'écrit :

$$g_h(i, j) = f(i + 1, j) - f(i, j)$$

Annexe D

Il s'agit bien d'un calcul de gradient suivant l'axe horizontal. En faisant subir une rotation de 90° au premier masque, il apparaît le filtre suivant dont l'origine est le point 1 :

Tableau 2

1
-1

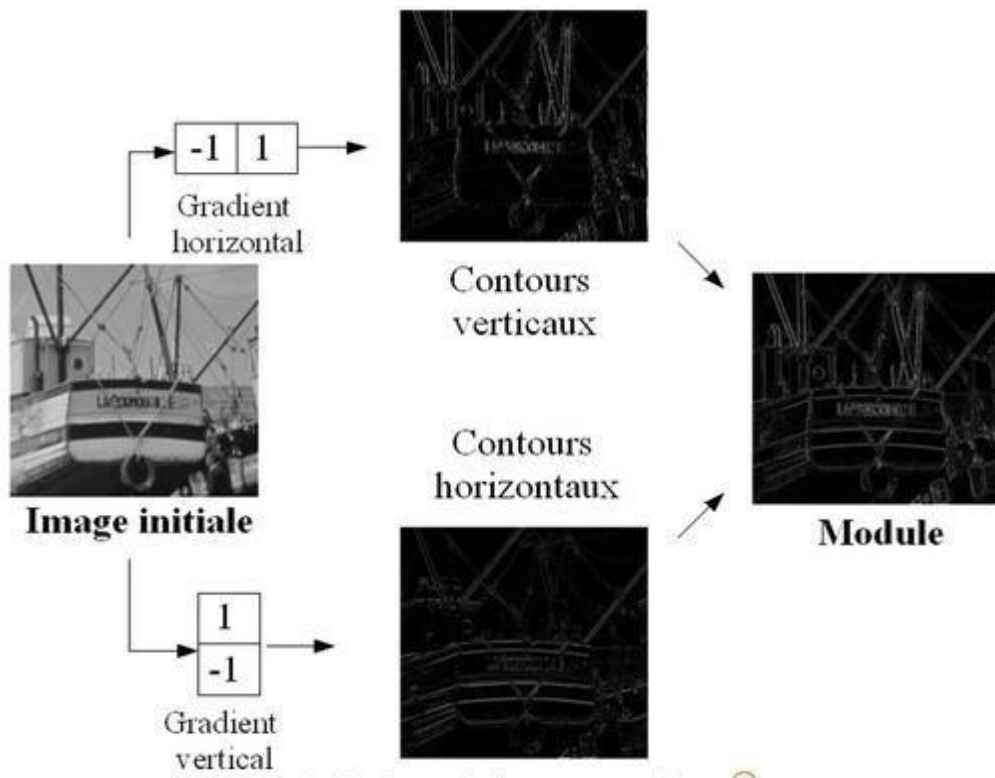
La réponse impulsionnelle est telle que :

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

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

La corrélation de ce masque avec l'image $f(i,j)$ permet bien d'implanter un gradient dans la direction verticale :

$$g_v(i,j) = f(i,j) - f(i,j-1)$$



C.2 Programmes sous MATLAB :

```
% Déclaration des variables globales utilisées dans différentes fonctions
global exitFrame;
global T_video;
global methode_edge;
global hsv_lower;
global hsv_upper;
global lum_;
global contrast_;

% Initialisation de certaines variables globales
T_video=traitement_video.detect_edge;
exitFrame=false;

% Création de la figure principale avec une barre d'outils et un menu désactivé
f=figure('ToolBar','none','MenuBar','none');

% Création des menus pour les différentes fonctionnalités
exitFrame=false;
medge = uimenu(f,'Text','&edge_detect');

% Sous-menus pour les différents opérateurs de détection de contours
medge_sobel = uimenu(medge,'Text','&Sobel');
medge_sobel.Accelerator = 'S';
medge_sobel.MenuSelectedFcn = @Pedge_sobel;
medge_Prewitt = uimenu(medge,'Text','&Prewitt');
medge_Prewitt.Accelerator = 'p';
medge_Prewitt.MenuSelectedFcn = @Pedge_prewitt;
medge_canny = uimenu(medge,'Text','&Canny');
medge_canny.Accelerator = 'c';
medge_canny.MenuSelectedFcn = @Pedge_canny;

% Menu pour la segmentation par couleur
```

Annexe D

```
mseg=uimenu(f,"Text",&seg_coleur');
```

```
% Sous-menus pour les différentes couleurs
```

```
mseg_Rouge = uimenu(mseg,'Text",&Rouge');
```

```
mseg_Rouge.MenuSelectedFcn = @Pseg_Rouge;
```

```
mseg_Orange = uimenu(mseg,'Text",&Orange');
```

```
mseg_Orange.MenuSelectedFcn = @Pseg_Orange;
```

```
mseg_Jaune = uimenu(mseg,'Text",&Jaune');
```

```
mseg_Jaune.MenuSelectedFcn = @Pseg_Jaune;
```

```
mseg_Vert = uimenu(mseg,'Text",&Vert');
```

```
mseg_Vert.MenuSelectedFcn = @Pseg_Vert;
```

```
mseg_Blue = uimenu(mseg,'Text",&Blue');
```

```
mseg_Blue.MenuSelectedFcn = @Pseg_Blue;
```

```
mseg_Violet = uimenu(mseg,'Text",&Violet');
```

```
mseg_Violet.MenuSelectedFcn = @Pseg_Violet;
```

```
% Menu pour les opérations morphologiques
```

```
mmorpho=uimenu(f,"Text",&morphological');
```

```
mmorpho_eros = uimenu(mmorpho,'Text",&Érosion');
```

```
mmorpho_eros.MenuSelectedFcn = @Pmorpho_eros;
```

```
mmorpho_Dilat = uimenu(mmorpho,'Text",&Dilatation');
```

```
mmorpho_Dilat.MenuSelectedFcn = @Pmorpho_dilat;
```

```
% Menu pour le traitement de la luminosité
```

```
mtrait_lum=uimenu(f,"Text",&traitement_lum');
```

```
% Sous-menu pour le contraste
```

```
mtrait_contrast = uimenu(mtrait_lum,'Text",&Contrast');
```

```
mcontrast_plus = uimenu(mtrait_contrast,'Text",&Augmenter');
```

```
mcontrast_plus.MenuSelectedFcn = @Pcontrast_plus;
```

```
mcontrast_plus.Accelerator = '1';
```

```
mcontrast_moins = uimenu(mtrait_contrast,'Text",&Diminuer');
```

```
mcontrast_moins.MenuSelectedFcn = @Pcontrast_moins;
```

```
mcontrast_moins.Accelerator = '2';
```

Annexe D

```
% Sous-menu pour la luminosité
mtrait_luminosite = uimenu(mtrait_lum,'Text','&luminosite');
mluminosite_plus = uimenu(mtrait_luminosite,'Text','&Augmenter');
mluminosite_plus.Accelerator = '=';
mluminosite_plus.MenuSelectedFcn = @Pluminosite_plus;
mluminosite_moins = uimenu(mtrait_luminosite,'Text','&Diminuer');
mluminosite_moins.Accelerator = '-';
mluminosite_moins.MenuSelectedFcn = @Pluminosite_moins;
```

```
% Menu pour quitter l'application
mExit=uimenu(f,"Text",&Exit);
mExit.Callback=@PExit;
```

```
% Bouton de sortie
Bexit=uicontrol(f,'Style','pushbutton');
Bexit.Position=[20 20 50 50];
Bexit.String='Exit';
Bexit.Callback=@PExit;
```

```
% Boucle principale de traitement vidéo
while true %debut de la boucle
    state = read(kb);
    if state.ESC
        break
    end
```

```
% Capture de l'image
    if exitFrame,
        break
    end
    if (local~=2),img=snapshot(cam);end;
    if (local==1), img=fliplr(img); end
```

```
% Conversion de l'image en HSV
```

Annexe D

```
hsv_img=rgb2hsv(img);

% Traitement de détection de contours
if (T_video==traitement_video.detect_edge)
edgesC=Edge_detect(hsv_img,methode_edge);
imshow(edgesC);drawnow;% axis off;
end

% Segmentation par couleur
% methode segmentation detection d'une couleur rouge ou
% vert ou blue
if (T_video==traitement_video.detect_seg)
img2=seg_couleur(hsv_img,hsv_lower,hsv_upper);
imshow(img2);drawnow;
end

% changement de luminosite
if (T_video==traitement_video.chang_lum)
hsv_img(:,:,3)=luminositec(hsv_img(:,:,3),lum_);
imglum=hsv2rgb(hsv_img);
imshow(imglum);drawnow;% axis off;
end

% Opération morphologique d'érosion
if (T_video==traitement_video.chang_contrast)
hsv_img(:,:,3)=contrastc(hsv_img(:,:,3),contrast_);
imglum=hsv2rgb(hsv_img);
imshow(imglum);drawnow;% axis off;
end

% Opération morphologique de dilatation
if (T_video==traitement_video.detect_eros)
se = offsetstrel('ball',5,5);
hsv_img(:,:,3)= imerode(hsv_img(:,:,3)*255.00,se)/255.00;
```

Annexe D

```
imgEros=hsv2rgb(hsv_img);
imshow(imgEros);drawnow;axis off;
imgEros=ecretage(imgEros);
end

% Opération morphologique de erosion
if (T_video==traitement_video.detect_dilat)
se = offsetstrel('ball',5,5);
hsv_img(:,:,3)=imdilate(hsv_img(:,:,3)*255.00,se)/255.00;
imgdil=hsv2rgb(hsv_img);
imshow(imgdil);drawnow;
imgdil=ecretage(imgdil);
writeVideo(vidwriter,rescale(imgdil));
end
end

% Définition des fonctions callback pour les différents menus
function PExit(src,event)
global exitFrame;
exitFrame=true;
end

function Pedge_sobel(src,event)
global T_video;
global methode_edge;
T_video=traitement_video.detect_edge;
methode_edge='Sobel';
end

function Pedge_prewitt(src,event)
global T_video;
global methode_edge;
T_video=traitement_video.detect_edge;
methode_edge='Prewitt';
```

Annexe D

end

```
function Pedge_canny(src,event)
global T_video;
global methode_edge;
T_video=traitement_video.detect_edge;
methode_edge='Canny';
end
```

```
function Pseg_Rouge(src,event)
global T_video;
global hsv_lower;
global hsv_upper;
hsv_lower=[0, 0.1, 0.1];hsv_upper=[0.01, 1, 1];
T_video=traitement_video.detect_seg;
end
```

```
function Pseg_Orange(src,event)
global T_video;
global hsv_lower;
global hsv_upper;
hsv_lower=[0.0733, 0.1, 0.1];hsv_upper=[0.0933, 1, 1];
T_video=traitement_video.detect_seg;
end
```

```
function Pseg_Jaune(src,event)
global T_video;
global hsv_lower;
global hsv_upper;
hsv_lower=[0.1657, 0.1, 0.1];hsv_upper=[0.1677, 1, 1];
T_video=traitement_video.detect_seg;
end
```

```
function Pseg_Vert(src,event)
```

Annexe D

```
global T_video;
global hsv_lower;
global hsv_upper;
hsv_lower=[0.3233, 0.1, 0.1];hsv_upper=[0.3433, 1, 1];
T_video=traitement_video.detect_seg;
end
```

```
function Pseg_Blue(src,event)
global T_video;
global hsv_lower;
global hsv_upper;
hsv_lower=[0.6567, 0.1, 0.1];hsv_upper=[0.6767, 1, 1];
T_video=traitement_video.detect_seg;
end
```

```
function Pseg_Violet(src,event)
global T_video;
global hsv_lower;
global hsv_upper;
hsv_lower=[0.8017, 0.1, 0.1];hsv_upper=[0.7817, 1, 1];
T_video=traitement_video.detect_seg;
end
```

```
function Pmorpho_eros(src,event)
global T_video;
T_video=traitement_video.detect_eros;
end
```

```
function Pmorpho_dilat(src,event)
global T_video;
T_video=traitement_video.detect_dilat;
end
```

```
function Pluminosite_plus(src,event)
global T_video;
```

Annexe D

```
global lum_;  
lum_=lum_+1;  
T_video=traitement_video.chang_lum;  
end
```

```
function Pluminosite_moins(src,event)  
global T_video;  
global lum_;  
lum_=lum_-1;  
T_video=traitement_video.chang_lum;  
end
```

```
function Pcontrast_plus(src,event)  
global T_video;  
global contrast_;  
T_video=traitement_video.chang_contrast;  
contrast_=contrast_*1.25;  
end
```

```
function Pcontrast_moins(src,event)  
global T_video;  
global contrast_;  
contrast_=contrast_/1.25;  
T_video=traitement_video.chang_contrast;  
end
```


Références

- [1] N. Azouza, «Segmentation par classification floue : application à l'imagerie par résonance magnétique. Mémoire de Master,» , Université Akli Moand Oulhadje, Bouira., 2018.
- [2] I. S. D. M Amine, «Compression Multimodale D'Images Numériques: Application Aux Images Médicales,» Doctoral dissertation, Université Mouloud Mammeri., 2011.
- [3] B. A. & L. N. E. islem, « Comparaison entre les différents filtres d'images,» Mémoire de fin d'étude, Université Abou Bakr Belkaid, Tlemcen, 2012.
- [4] F. ., Medjahed, «Détection et Suivi d'Objets en Mouvement Dans Une Séquence d'Images,» Diss.Usto, 2012.
- [5] «adobe,» 2024. [En ligne]. Available: <https://helpx.adobe.com/fr/premiere-elements/using/compression-data-rate-basics.html>.
- [6] «filmora,» [En ligne]. Available: <https://filmora.wondershare.fr/video-editing-tips/what-is-video-bitrate.html>. [Accès le 2024].
- [7] «gouvenelstudio,» 2023. [En ligne]. Available: <https://gouvenelstudio.com/homecinema/resolution.htm>.
- [8] S. B. e. A. Hamel, «Segmentation d'image par Coopération région-contours,» Université KasdiMerbah, Ouargla, 2016.
- [9] B. a. B. N. Manel, «Détection et suivi d'un objet suspect dans le contexte de vidéosurveillance,» Université Kasdi Merbah, Ouargla, 2021.
- [10] A. M. S. e. A. O. Abidi, «Suivi d'objets en mouvement dans une séquence des images. Diss,» Université Ahmed Draia, Adrar, 2017.
- [11] j. Smith, «étude des images numériques matricielles et vectorielles : avantages, inconvénients et techniques de traitement,» thèse de doctorat.
- [12] Z. Yahiaoui, «Etude et implémentation de l'algorithme c moyenne floue et ses variantes (Doctoral dissertation),» (Doctoral dissertation) Faculté des Mathématiques et de l'Informatique-Université Mohamed Boudiaf, Msila, 2013.
- [13] A. ., S. E. G. Badaoui, «Etude des suites chaotiques et leurs applications en cryptage d'images,» (Doctoral dissertation)., 2021.
- [14] h. Guesmia, «Algorithme de segmentation par chan-veuse pour les différentes modalités des images,» (Doctoral dissertation, Université Mohamed Boudiaf, Msila, 2019.
- [15] F. Hadji, « Conception et réalisation d'un système de cryptage pour les images médicales,» Doctoral dissertation, Université Mohamed Boudiaf Msila Faculté des mathématique et de l'informatique . Filière: Informatique), 2018.

Références

- [16] M. F. e. B. Abdelfattah, «Débruitage de l'image par filtrage stochastique,» Mémoire fin d'étude, Institut de maintenance et de sécurité industrielle, Oran, 2016.
- [17] C. M. Tahar, «Amélioration des images par un modèle de réseau de neurones (Comparaison avec les filtres de base),» Mémoire de fin d'études pour l'obtention du diplôme de Master en Informatique, 2011.
- [18] S. M. Tripathi, « "Évaluation de la qualité des images en couleur reproduites en utilisant le système de couleurs CMJN",» Université de Pune, 2016.
- [19] H. & B. M. Taibi, «Un modèle de deeplearning pour la détection des lésions de COVID-19,» Doctoral dissertation, Université Ibn Khaldoun-Tiaret-, 2022.
- [20] I. Refice, «Critère d'évaluation pour les méthodes de segmentation d'images,» Doctoral dissertation Université Mohamed Boudiaf, M'Sila, 2012.
- [21] A. & D. A. Dicko, «Mise en place d'un système d'indexation et de recherche d'image par le contenu,» Doctoral dissertation.
- [22] S. O. U. K. H. A. L. & N. N. A. I. D. J. I. Fairouz, «Implémentation des filtres non linéaires 2D sur FPGA avec HDL Coder,» Doctoral dissertation, faculté des sciences et de la technologie univbba., 2022.
- [23] H. & S.Toufik, «Segmentation d'images IRM: application sur des IRM de patients atteints d'Alzheimer,» Doctoral dissertation, Université Mouloud Mammeri, 2015.
- [24] B. & B. D. .. Sebaa, «Traitement d'image et morphologie mathématique».
- [25] M. G. EL-Hachemi, «Performances dans la Segmentation d'images médicales,» École nationale Supérieure en Informatique, 2018.
- [26] A. Kifadji, «Segmentation des images de drones par apprentissage profond ou deep learning. ,,» Master'sthesis Université Badji Mokhtar, Annaba, 2020.
- [27] Audebert, N., Boulch, A., Le Saux, B., & Lefèvre, S, «Segmentation sémantique profonde par régression sur cartes de distances signées. In Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP).,» 2018, June.
- [28] «ichi.pro.,» 2022. [En ligne]. Available: <https://ichi.pro>.
- [29] K. & S. A. Benahmed, «La segmentation des images médicales IRM par ensembles de niveau (Level sets),» Doctoral dissertation, Université Ibn Khaldoun-Tiaret-, 2020.
- [30] S. & D. W. Faid, «Modèle de segmentation d'image à champ aléatoire de markov (MRF),» (Doctoral dissertation, Univ M'sila)., 2020.
- [31] «lawinsider,» 2013. [En ligne]. Available: <https://www.lawinsider.com/dictionary/ots-method>. [Accès le 2024].
- [32] «otsgroup,» 2024. [En ligne]. Available: <https://otsgroup.es/en/ots-method/>.

Références

- [33] N. Rabia, « Revue des méthodes de segmentation d'images texturées: Cas des images couleur (Doctoral dissertation, Université Mouloud Mammeri).,» 2011.
- [34] M. & K. F. Mohammedi, «Segmentation d'image IRM cérébrale en utilisant l'intensité et l'information contextuelle par le champ aléatoire de Markov,» Doctoral dissertation, univmsila, 2021.
- [35] H. & S. Y. Saadi, «Deep-Learning applique a la segmentation des images médicales,» Doctoral dissertation, université akli mohand oulhadj-bouira, 2021.
- [36] S. & H. A. Benfriha, «Segmentation d'image par Coopération région-contours.,» 2016.
- [37] B. N. B. & S. B. Chabha, «Segmentation d'image IRM cérébrale par contours actifs (Snake classique et snake GVF) (Doctoral dissertation, Université Mouloud Mammeri).,» 2009.
- [38] S. Medouakh, «Détection et suivi d'objets.,» Diss. Université Mohamed Khider, Biskra, 2019.
- [39] A. Djenaidia, «Suivi d'objets robuste en utilisant un histogramme conjoint de couleur et de texture,» Université Mohamed Khider, Biskra, 2018.
- [40] D. Riahi, «Suivi multi-objets par la détection: Application à la vidéo surveillance,» Diss. Ecole Polytechnique, Montreal (Canada), 2016.
- [41] W. Bouachir., «Suivi d'objets par caractéristiques locales encodant la structure. Suivi d'objets par caractéristiques locales encodant la structure,» Thèse de doctorat, université de Montréal, Montréal, 2014.
- [42] Trichet. R, «suivi d'objet pour la télévision interactive,» thèse de doctorat, L'école télécom, paris, 2008.
- [43] H. a. H. O. Nouria, «Etude et réalisation d'un système de détection et suivi d'un objet en mouvement,» Diss. Université Mouloud Mammeri, Tizi Ouzou, 2017.
- [44] M. Mounia, «Suivi d'objets dans une séquence d'images par modèle d'apparence : conception et évaluation,» Diss. Bordeaux 1, 2008.
- [45] A. a. A. L. .. Yasmine, «Détection et suivi automatiques d'objets en mouvement,» Diss. Université Mouloud Mammeri, Tizi Ouzou, 2015.
- [46] I. Leang, «Fusion en ligne d'algorithmes de suivi visuel d'objet,» Thèse de doctorat, université Pierre et Marie Curie , Paris VI, Décembre 2016.
- [47] M. Brulin, «Analyse sémantique d'un trafic routier dans un contexte de vidéo-surveillance,» Diss. Bordeaux 1, 2012.
- [48] W. a. A. H. BEY, «Collaboration entre drones pour une meilleure gestion du territoire,» Diss. Université Ibn Khaldoun, Tiaret, 2021.
- [49] [. i.-e.-g.-a.-l.-d.-d.-o.-d.-l.-p.-e.-. renouvelables-226754316739090. [En ligne].
- [50] W. Benkamouch, «Détection d'objets et deep Learning dans un trafic routier,» Université 8 mai

1945, Guelma, 2023.

- [51] linkedin, «linkedin,» 2024. [En ligne]. Available: <https://fr.linkedin.com/pulse/comprendre-yolo-yannick-lecroart>.
- [52] B. Cayla. [En ligne]. Available: <https://datacorner.fr/yolo-class-reduc/>.
- [53] blent.ai, 2024. [En ligne]. Available: <https://blent.ai/blog/a/detection-images-yolo-tensorflow>.
- [54] R. e. a. Girshick, «Rich feature hierarchies for accurate object detection and semantic segmentation,» Proceedings of the IEEE conference on computer vision and pattern recognition, 2014.
- [55] R. Girshick, «Fast r-cnn,» Proceedings of the IEEE international conference on computer vision, 2015.
- [56] S. e. a. Ren, «Faster r-cnn: Towards real-time object detection with region proposal networks,» Advances in neural information processing systems 28, 2015.
- [57] uni-d.ca, 2022. [En ligne]. Available: <https://uni-d.ca/resolution-dpi-ppi-ppp-dimages/>.
- [58] «aplicit,» [En ligne]. Available: <https://www.aplicit.com/dpi-ppp-pixels-comment-sen-sortir/>.
- [59] infomir, «infomir,» 2001. [En ligne]. Available: <https://www.infomir.eu/fra/blog/articles/43-video-codecs/>. [Accès le 2024].
- [60] dacast, «dacast,» [En ligne]. Available: <https://www.dacast.com/fr/le-blog-des-experts-video/quest-ce-quun-codec-video/>. [Accès le 2024].
- [61] wikipedia, «wikipedia,» 2024. [En ligne]. Available: <https://fr.wikipedia.org/wiki/Codec>.
- [62] aiseesoft, «aiseesoft,» 2024. [En ligne]. Available: <https://www.aiseesoft.fr/resource/mp4.html>.
- [63] movavi, «movavi,» 2024. [En ligne]. Available: <https://www.movavi.com/fr/learning-portal/what-is-mp4.html>.
- [64] wikipedia, «wikipedia,» 15 janvier 2024. [En ligne]. Available: <https://fr.wikipedia.org/wiki/MPEG-4>.
- [65] recoverit.wondershare, «wondershare,» 2024. [En ligne]. Available: <https://recoverit.wondershare.fr/video-recovery/what-is-mov-file.html>.
- [66] apowersoft, «apowersoft,» 12 janvier 2016. [En ligne]. Available: <https://www.apowersoft.fr/cest-quoi-un-fichier-mov.html>.
- [67] «cloudinary,» 2024. [En ligne]. Available: <https://cloudinary.com/guides/video-formats/mkv-format-what-is-mkv-how-it-works-and-how-it-compares-to-mp4>.
- [68] homecine, «homecine,» 2004. [En ligne]. Available: <https://www.homecine-compare.com/definition-de-avi.htm>. [Accès le 2024].

Références

- [69] «futura,» 2001. [En ligne]. Available: <https://www.futura-sciences.com/tech/definitions/informatique-avi-8261/>. [Accès le 2024].
- [70] ionos, «ionos,» [En ligne]. Available: <https://www.ionos.fr/digitalguide/sites-internet/creation-de-sites-internet/webm/>. [Accès le 2024].
- [71] lcd-compare, «lcd-compare,» 2004. [En ligne]. Available: <https://www.lcd-compare.com/definition-de-webm.htm>. [Accès le 2024].
- [72] apprendrelavideo, «apprendrelavideo,» [En ligne]. Available: <https://apprendrelavideo.fr/codecs-video-lequel-choisir/>. [Accès le 2024].
- [73] h264distribution, «h264distribution,» 2024. [En ligne]. Available: <https://www.h264distribution.com/>.
- [74] «pewtrusts,» 2024. [En ligne]. Available: https://www.pewtrusts.org/-/media/assets/2021/02/em-toolkit/fr_toolkit_5_final.pdf.
- [75] «visualsfrance,» [En ligne]. Available: <https://www.visualsfrance.com/137-transmission-streaming>.
- [76] «ceco,» 2024. [En ligne]. Available: https://kost-ceco.ch/cms/kad_mpeg-2_fr.html.
- [77] «microsoft,» 2024. [En ligne]. Available: <https://learn.microsoft.com/fr-fr/windows/win32/medfound/mpeg-2-video-decoder>.
- [78] «malavida,» 2024. [En ligne]. Available: <https://mpeg-2-video-decoder.fr.malavida.com/windows/>.
- [79] «magix,» 2003. [En ligne]. Available: <https://www.magix.com/fr/montage-video/compresser-une-video/>. [Accès le 2024].
- [80] «digiforma,» [En ligne]. Available: <https://help.digiforma.com/fr/articles/3379886-comment-compresser-des-videos-quelle-est-la-taille-maximale-pour-une-video-e-learning>.
- [81] «wikipedia,» octobre 2023. [En ligne]. Available: https://en.wikipedia.org/wiki/Mean_shift. [Accès le 2023].
- [82] «researchgate,» 2008. [En ligne]. Available: https://www.researchgate.net/figure/Determination-de-quatre-modes-principaux-par-Mean-Shift_fig3_274148087. [Accès le 2024].
- [83] «datascientest,» 2024. [En ligne]. Available: <https://datascientest.com/meanshift-tout-savoir#>.
- [84] «mathworks,» [En ligne]. Available: <https://www.mathworks.com/help/deeplearning/ref/darknet53.html>.
- [85] «paperswithcode,» [En ligne]. Available: <https://paperswithcode.com/method/darknet-53>.
- [86] [En ligne]. Available: http://www.optique-ingenieur.org/fr/cours/OPI_fr_M04_C05/co/Contenu_02.html. [Accès le 10 06 2024].

