

الجمهورية الجزائرية الديمقراطية الشعبية

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

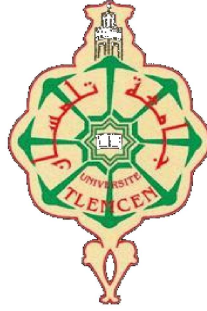
وزارة التعليم العالي والبحث العلمي

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

جامعة أبي بكر بلقايد - تلمسان

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE



## **MEMOIRE**

Présenté pour l'obtention du **diplôme** de **MASTER**

**En** : Automatique

**Spécialité** : Automatique et informatique industrielle

**Par** : REMMANI Yassine

**Sujet**

# **VISION PAR INTELLIGENCE ARTIFICIELLE APPLIQUÉE À LA ROBOTIQUE**

Soutenu publiquement, le 19/09/2024, devant le jury composé de :

Mme BORSALI Salima

MCB

Université de Tlemcen

Présidente

Mme HANDOUZI Wahida

MCB

Université de Tlemcen

Examinatrice

Mme BELDJILALI Nawel

MAA

Université de Tlemcen

Encadrante

Année universitaire : 2023 /2024

# Dédicace

Je dédie cet humble travail à :

Ma tendre mère pour son amour et son aide

À mon cher père, en reconnaissance de tout ce qu'il a fait pour moi.

À mes sœurs, qui m'ont toujours soutenu et encouragé dans ce travail.

Et à toute ma famille.

À tous mes amis(es) et camarades de la faculté de Technologie, ainsi qu'à tous mes enseignants de l'Université Abou Bekr Belkaid de Tlemcen.

# Remerciements

Mes remerciements les plus chaleureux et sincères s'adressent à :

Mme Beldjilali.N pour son important travail d'évaluation et ses conseils,

Mme Borsali.S qui m'a honoré en acceptant de présider le jury,

Mme Hendouzi.W pour avoir bien accepté de juger ce travail,

A tout le staff des enseignants de la faculté de Technologie d'avoir développer en nous la volonté de persévérer dans la quête du savoir.

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Segmentation d'images</b>	<b>2</b>
1.1 Introduction . . . . .	2
1.2 Notions d'images . . . . .	2
1.2.1 Image binaire . . . . .	2
1.2.2 Image en niveau de gris . . . . .	3
1.2.3 Image couleur . . . . .	3
1.3 Caractéristiques d'une image numérique [28] . . . . .	4
1.3.1 Le pixel . . . . .	4
1.3.2 La dimension . . . . .	4
1.3.3 La résolution . . . . .	4
1.3.4 Le bruit . . . . .	4
1.3.5 Le contour . . . . .	4
1.3.6 La région . . . . .	4
1.3.7 La luminance . . . . .	5
1.3.8 Le contraste . . . . .	5
1.3.9 L'histogramme d'une image . . . . .	5
1.4 Définition de la segmentation . . . . .	6
1.5 Approches de la segmentation d'images . . . . .	6
1.5.1 Approches contour . . . . .	6
1.5.1.1 Méthodes dérivatives . . . . .	7
1.5.1.2 Opérateurs dérivatifs du premier ordre . . . . .	7
1.5.1.3 Opérateurs dérivatifs du second ordre . . . . .	8
1.5.2 Approches région . . . . .	8
1.5.2.1 Segmentation par division de région . . . . .	8
1.5.2.2 Segmentation par fusion . . . . .	8
1.5.2.3 Segmentation par division et fusion de région (split and merge) . . . . .	9
1.5.3 Segmentation par coopération régions-contours . . . . .	9
1.5.3.1 Coopération séquentielle . . . . .	9
1.5.3.2 Coopération mutuelle . . . . .	10
1.5.3.3 Coopération des résultats . . . . .	10
1.6 Intérêt de la segmentation d'images dans le domaine de l'automatique et l'informatique industrielle . . . . .	11
1.7 Évaluation d'une méthode de segmentation . . . . .	11
1.8 Conclusion . . . . .	11
<b>2 Les réseaux de neurones</b>	<b>12</b>
2.1 Introduction . . . . .	12
2.2 Les avantages des réseaux de neurones . . . . .	12

2.3	Les composants fondamentaux et la modélisation d'un neurone artificiel . . . . .	13
2.3.1	Le neurone biologique . . . . .	13
2.3.2	Le neurone artificiel . . . . .	13
2.4	Les réseaux de neurones . . . . .	15
2.5	Les différentes structures des réseaux de neurones . . . . .	15
2.5.1	Réseaux de neurones multicouches . . . . .	15
2.5.2	Réseau à connexions locales . . . . .	15
2.5.3	Réseaux à connexions récurrentes . . . . .	16
2.5.4	Réseaux à connexions complètes . . . . .	16
2.6	Les réseaux de neurones convolutifs . . . . .	17
2.6.1	Couches convolutives . . . . .	17
2.6.2	l'opération de mise en commun (Pooling) . . . . .	18
2.7	L'apprentissage des réseaux de neurones . . . . .	18
2.7.1	Apprentissage supervisé . . . . .	19
2.7.2	Apprentissage non-supervisé . . . . .	19
2.7.3	Apprentissage profond ( Deep Learning) . . . . .	19
2.8	Algorithme d'apprentissage . . . . .	19
2.9	Application des réseaux de neurones au traitement d'images . . . . .	20
2.10	Conclusion . . . . .	20
<b>3</b>	<b>La logique floue</b>	<b>21</b>
3.1	Introduction . . . . .	21
3.2	Notions fondamentales sur les ensembles flous . . . . .	21
3.2.1	Définition d'un ensemble flou . . . . .	21
3.2.2	Caractéristique d'un ensemble flou[18] . . . . .	22
3.2.2.1	Support . . . . .	22
3.2.2.2	Noyau . . . . .	23
3.2.2.3	Hauteur . . . . .	23
3.2.2.4	Cardinalité . . . . .	23
3.2.2.5	Sous ensemble convexe . . . . .	23
3.3	Les éléments de base de la logique floue . . . . .	24
3.3.1	Les variables linguistiques . . . . .	24
3.3.1.1	Fonction d'appartenance[19] . . . . .	24
3.3.1.2	Exemple de fonction d'appartenance . . . . .	24
3.3.2	Les déductions floues ( inférences) . . . . .	27
3.3.2.1	Les inférences à une seule règle . . . . .	27
3.3.2.2	Les inférences à plusieurs règles . . . . .	27
3.3.3	Les opérations sur les ensembles flous[45] . . . . .	27
3.4	Segmentation floue . . . . .	28
3.5	Conclusion . . . . .	29
<b>4</b>	<b>Interprétation et résultats</b>	<b>30</b>
4.1	Introduction . . . . .	30
4.2	Segmentation par la logique floue . . . . .	30
4.2.1	Algorithme de segmentation d'image par la logique floue . . . . .	30
4.2.1.1	Histogramme . . . . .	31
4.2.2	Résultats . . . . .	31
4.2.2.1	Rapidité et Efficacité de la Segmentation par Logique Floue . . . . .	31
4.2.2.2	Prétraitement de l'image . . . . .	31
4.2.2.3	Application des Fonctions d'Appartenance Floues . . . . .	32

4.2.2.4	Segmentation des Régions . . . . .	32
4.2.3	Interprétation . . . . .	33
4.2.3.1	Analyse des Résultats . . . . .	33
4.2.3.2	Limitations . . . . .	33
4.3	Segmentation par CNN . . . . .	34
4.3.1	Structure du projet . . . . .	34
4.3.2	Comprendre l'ensemble de données . . . . .	35
4.3.3	Implementation . . . . .	35
4.3.3.1	DATA.PY . . . . .	35
4.3.3.2	MODEL.PY . . . . .	35
4.3.3.3	TRAIN.PY . . . . .	36
4.3.3.4	PREDICT.PY . . . . .	37
4.3.4	Résultats . . . . .	37
4.3.4.1	Temps d'Entraînement et de Prédiction des CNN . . . . .	37
4.3.4.2	Graphiques d'Évaluation . . . . .	37
4.3.5	Images Prédites par le Modèle . . . . .	38
4.4	Conclusion . . . . .	39
<b>Conclusion</b>		<b>40</b>
<b>Annexes</b>		<b>42</b>

# Table des figures

1.1	Exemple d'une image binaire. . . . .	3
1.2	Exemple d'une image en niveaux de gris. . . . .	3
1.3	Exemple d'une image en couleur. . . . .	4
1.4	Image Lenna en niveau de gris et son histogramme. . . . .	5
1.5	Image Lenna en couleur et les histogrammes R, V et B . . . . .	5
1.6	Différents techniques de segmentation . . . . .	6
1.7	Quelques modèles de contours[26] . . . . .	7
1.8	Détection de contours par des filtres différents . . . . .	7
1.9	Détection de contours par le <i>Laplacien</i> . . . . .	8
1.10	Décompositions successives des blocs . . . . .	8
1.11	Exemple de segmentation par fusion . . . . .	9
1.12	Exemple de division-fusion de région . . . . .	9
1.13	Principe de la coopération séquentielle . . . . .	10
1.14	Principe de la coopération mutuelle . . . . .	10
1.15	Principe de la coopération des résultats . . . . .	10
2.1	Un neurone avec son arborisation dendritique . . . . .	13
2.2	Structure d'un neurone artificiel . . . . .	14
2.3	Fonctions d'activation les plus utilisées . . . . .	14
2.4	Réseau de neurones multicouches. . . . .	15
2.5	Réseau à connexions locales . . . . .	16
2.6	Réseau à connexions récurrentes . . . . .	16
2.7	Réseau à connexions complètes . . . . .	16
2.8	Exemple de CNN appelé AlexNet[29] . . . . .	17
2.9	Exemple d'opération de regroupement avec le maximum de Pooling (à droite) et la moyenne Pooling (à gauche) . . . . .	18
3.1	Sous ensemble flou A représentant la population jeune, l'âge est représenté par la variable x . . . . .	22
3.2	Support, Noyau et Hauteur d'un sous ensemble flou . . . . .	23
3.3	Fonction d'appartenance trapézoïdale . . . . .	25
3.4	Fonction d'appartenance triangulaire . . . . .	25
3.5	Fonction d'appartenance exponentielle . . . . .	26
3.6	Fonction d'appartenance gaussienne . . . . .	26
3.7	Opérations floues . . . . .	28
4.1	Image et son histogramme correspondant. . . . .	31
4.2	Application d'un filtre gaussien pour lisser l'image. . . . .	32
4.3	Image floue. . . . .	32
4.4	Régions détectées. . . . .	33
4.5	Exemple avec modification des seuils. . . . .	33
4.6	Architecture originale d'Unet. . . . .	34

4.7	Métriques de Performance du Modèle au Cours de l'Entraînement et de la Validation . . . . .	38
4.8	Image d'entrée, Vérité terrain, Masque prédit. Une partie de l'exemple des résultats des tests. . . . .	39



# Liste des tableaux

2.1	Modélisation des éléments du neurone biologique . . . . .	14
4.1	Détails du Train, Val et des données d'essai . . . . .	35
4.2	Les valeurs d'époque finale. . . . .	36

## Liste des abréviations

<b>MLP</b> :	<i>Multilayer Perceptron</i>
<b>FDP</b> :	<i>Filtres de Dérivée Première</i>
<b>I(x, y)</b> :	<i>Intensité au pixel (x, y)</i>
$\vec{\nabla}I$ :	<i>Gradient de l'image</i>
$\partial I(x, y)/\partial x$ :	<i>Dérivée Partielle selon x</i>
$\partial I(x, y)/\partial y$ :	<i>Dérivée Partielle selon y</i>
$\hat{I}_x$ :	<i>Vecteur Unitaire suivant x</i>
$\hat{I}_y$ :	<i>Vecteur Unitaire suivant y</i>
$E_t$ :	<i>Coefficient de Pondération</i>
$\nabla_x$ :	<i>Gradient suivant x</i>
$\nabla_y$ :	<i>Gradient suivant y</i>
$G(x, y)$ :	<i>Module du Gradient</i>
$G_x(x, y)$ :	<i>Composant Horizontal du Gradient</i>
$G_y(x, y)$ :	<i>Composant Vertical du Gradient</i>
$\arg G(x, y)$ :	<i>Direction du Gradient</i>
<b>Arct</b> :	<i>Fonction Arc Tangente</i>
$\nabla^2 I$ :	<i>Laplacien de l'image (Dérivée Seconde)</i>
$\partial^2 I/\partial x^2$ :	<i>Dérivée Seconde selon x</i>
$\partial^2 I/\partial y^2$ :	<i>Dérivée Seconde selon y</i>
$I(x, y)$ :	<i>Intensité au pixel (x, y)</i>
<b>Convolution</b> :	<i>Opération mathématique sur les matrices d'images</i>
<b>Laplacien</b> :	<i>Opérateur de dérivée seconde dans toutes les directions</i>
<b>CNN</b> :	<i>Convolutional Neural Networks</i>
<b>Convolution</b> :	<i>Opération mathématique appliquée aux données d'entrée</i>
<b>Pooling</b> :	<i>Opération de réduction de la dimension des données</i>
<b>2D</b> :	<i>Deux Dimensions</i>
<b>3D</b> :	<i>Trois Dimensions</i>
<b>h</b> :	<i>Hauteur de l'image</i>
<b>l</b> :	<i>Largeur de l'image</i>
<b>c</b> :	<i>Nombre de canaux de l'image</i>
<b>m</b> :	<i>Hauteur du filtre</i>
<b>n</b> :	<i>Largeur du filtre</i>
<b>f</b> :	<i>Nombre de filtres</i>
<b>X</b> :	<i>Image d'entrée dans l'opération de convolution</i>
<b>A</b> :	<i>Noyau de convolution (ou filtre)</i>
$\beta$ :	<i>Biais de l'opération de convolution</i>
<b>Y</b> :	<i>Sortie de l'opération de convolution</i>
<b>R</b> :	<i>Ensemble des nombres réels (<math>\mathbb{R}</math>)</i>
<b>Pooling</b> :	<i>Opération de mise en commun (ou regroupement)</i>
<b>Neurones</b> :	<i>Unités de calcul d'un réseau neuronal artificiel</i>
<b>FA</b> :	<i>Fonction d'appartenance</i>

# Introduction Générale

Dans le domaine de la vision par ordinateur, la segmentation d'image constitue un processus fondamental. Elle permet de partitionner une image en régions significatives, facilitant ainsi l'extraction et l'analyse des informations visuelles. Cette technique trouve des applications dans divers secteurs, notamment la robotique, la médecine, l'automobile, l'agriculture, et bien d'autres. Cependant, malgré les avancées technologiques, la segmentation d'images reste un défi, surtout lorsque les frontières entre les objets ne sont pas clairement définies ou lorsque les images sont affectées par du bruit.

Les approches traditionnelles de segmentation d'image reposent souvent sur des méthodes de seuils ou de détection de contours, qui peuvent se révéler insuffisantes dans des scénarios complexes. Par conséquent, l'intégration de techniques plus avancées, telles que la logique floue et l'intelligence artificielle (IA), a émergé comme une voie prometteuse pour surmonter ces limitations. La logique floue, en particulier, offre un moyen de modéliser l'incertitude et l'imprécision caractéristiques courantes des données d'images. Combinée avec les modèles d'apprentissage profond, l'IA permet de capturer des motifs complexes et d'améliorer considérablement les performances de segmentation.

La principale problématique abordée dans ce projet est de concevoir un système capable de segmenter efficacement des images, même en présence de bruit, de variations d'éclairage, ou de contours flous. Pour cela, nous proposons d'explorer l'utilisation de la logique floue ainsi que d'intelligence artificielle à travers les réseaux de neurones afin d'améliorer la précision de segmentation.

Ce travail s'articule autour de quatre chapitres mettant chacun l'éclairage sur les techniques utilisées dans l'élaboration de notre algorithme. Le premier chapitre est une introduction à la segmentation d'image et ses différentes approches classiques.

Nous avons ensuite présenté dans les deux chapitres suivants, les principes des RN et de la LF, leurs fonctionnements et leurs contributions au domaine du traitement d'images.

Finalement, dans le chapitre quatre, les algorithmes de segmentation proposés ont été développés et testés sur des images médicales.

Nous concluons finalement ce travail par une conclusion générale et proposons des perspectives qui peuvent suivre ce travail de recherche.

# Chapitre 1

## Segmentation d'images

### 1.1 Introduction

L'un des moyens les plus importants qu'utilise l'homme pour communiquer avec son environnement se résume dans les sons et les images. L'images est une représentation spatiale d'une scène généralement tridimensionnelle qui transforme le support photographique es une collection d'informations élémentaires. Cette dernière, liée à chaque point de l'image, sera transcrite en niveaux de gris ou en couleur donnant ainsi l'aspect numérique à l'image acquise par un capteur donné.

L'analyse d'images se résume en l'étude des signaux provenant des capteurs. Ces signaux sont numérisés en vue d'un traitement par ordinateur, afin d'extraire les informations pertinentes en regard de l'application concernée, traités puis interprétés.

La segmentation recherche des primitives dans l'image ou tend à interpréter son contenu sous une représentation particulière en manipulant des données de nature numérique sans comprendre l'image et sans faire se liaison avec la réalité qu'elle représente.

### 1.2 Notions d'images

L'image est un signal bidimensionnel à support borné. Après son échantillonnage et sa quantification, l'image digitale, en niveaux de gris, sera représentée par une matrice  $A$  à valeurs numériques appelée image digitale. Chaque élément de la matrice  $A$  est appelé pixel ou pel (picture element) et il est proportionnel au niveau de gris de l'image en ce point. Dans une image, on veut toujours extraire des sous-ensembles de points représentant des parties significatives de la forme des objets, appelées indices visuels. Parmi ces indices, on a le contour et la région.

Il existe 3 types d'images : binaire, en niveau de gris et couleur[11].

#### 1.2.1 Image binaire

C'est une image à deux niveaux de gris : le niveau zéro qui représente le noiret le niveau 1 pour le blanc. Un seul bit est utilisé pour coder ce type d'images.s[28].



FIGURE 1.1 – Exemple d'une image binaire.

### 1.2.2 Image en niveau de gris

Dans ce cas, on prend la valeur lumineuse en chaque point de l'image et le pel sera codé non pas par un bit mais par un octet pour représenter les 256 niveaux de lumières.[28].



FIGURE 1.2 – Exemple d'une image en niveaux de gris.

### 1.2.3 Image couleur

Chaque pixel est une combinaison des trois couleurs de base RVB ( rouge, vert, bleu) et codé sur trois octets comme le montre la figure 1.3.



FIGURE 1.3 – Exemple d’une image en couleur.

## 1.3 Caractéristiques d’une image numérique [28]

une image se caractérise par les paramètres suivants :

### 1.3.1 Le pixel

C’est le plus petit élément de l’image, généralement appelé PEL ” picture element ”, le pixel peut être codé sur un bit : image binaire, sur un octet : image en niveau de gris ou sur trois octets : image en couleur.

### 1.3.2 La dimension

Elle est donnée par le nombre de lignes et de colonnes de la matrice représentant l’image numérique.

### 1.3.3 La résolution

La résolution est exprimée en nombre de pixels par unité de mesure, plus le nombre est élevé, meilleure est la résolution.

### 1.3.4 Le bruit

Dans une image, le bruit se traduit par une transition brusque de l’intensité d’un pixel comparé à ses pixels voisins..

### 1.3.5 Le contour

Il se définit comme la frontière entre les éléments constituant l’image. Autrement dit, c’est une différence significative de niveau de gris.

### 1.3.6 La région

C’est un groupement de pixels similaires selon un attribut donné.

### 1.3.7 La luminance

En vérité le pixel de l'image contient deux informations : luminance et chrominance ( lumière et couleur). La luminance est le point lumineux de l'image. Il faut noter qu'un point lumineux est plus perceptible par l'oeil humain qu'un point en couleur : on voit bien les étoiles lumineuses en pleine nuit !

### 1.3.8 Le contraste

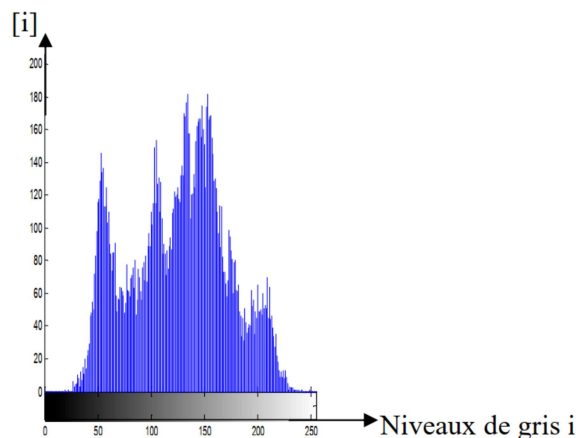
C'est la différence de couleur entre les zones constituant l'image. Une image bien contrastée permet d'obtenir de meilleurs résultats en guise de segmentation d'images.

### 1.3.9 L'histogramme d'une image

Par définition, l'histogramme est une courbe représentative de la fréquence d'apparition des différents niveaux de gris constituant l'image comme le montre le figure 1.4.



a. Image Lenna



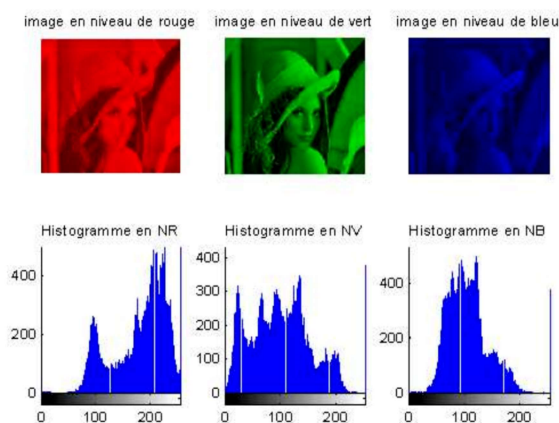
b. Histogramme en niveau de gris

FIGURE 1.4 – Image Lenna en niveau de gris et son histogramme.

Pour les images en couleur trois histogrammes sont nécessaires, un pour chaque couleur de base (RVB) figure 1.5.



a. Image Lenna



b. Histogramme en niveaux de rouge, vert et bleu

FIGURE 1.5 – Image Lenna en couleur et les histogrammes R, V et B

## 1.4 Définition de la segmentation

La segmentation d'image consiste à subdiviser une image en plusieurs segments ou régions distinctes. Chaque segment regroupe des pixels ayant des caractéristiques similaires ( niveau de gris, texture, spectre,...) communs qui les distinguent des pixels des régions voisines.[10] Autrement dit : Soient  $I$  l'image initiale et les  $R_i$  est l'ensemble des régions formant l'image  $I$ . [3] La segmentation doit répondre aux conditions suivantes :

$$\begin{aligned} \bigcup_{i=1}^n R_i &= I \\ R_i &\neq \emptyset \quad \forall i = 1, \dots, n \\ R_i \cap R_j &= \emptyset \quad \forall i, j \text{ avec } i \neq j \end{aligned} \tag{1.1}$$

## 1.5 Approches de la segmentation d'images

La segmentation permet une division de l'image afin d'extraire des informations structurales. Il existe de nombreuses façons de permettre cette division principalement des méthodes par contour et des méthodes par région (voir figure 1.1).

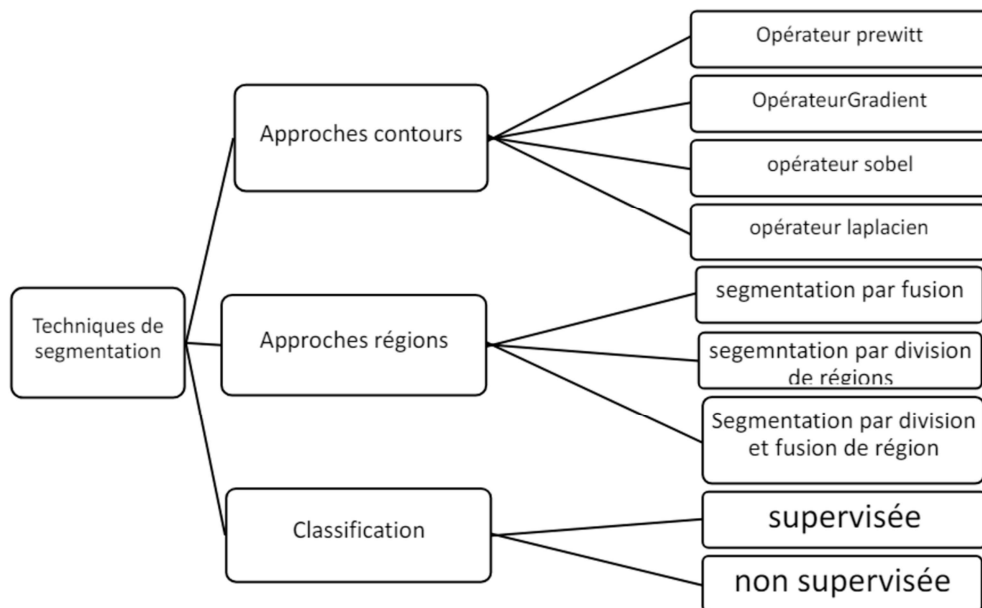


FIGURE 1.6 – Différents techniques de segmentation

### 1.5.1 Approches contour

Pour le traitement d'images, il est très important de déterminer les limites des objets dans la scène. L'approche frontière regroupe des techniques fondées sur les discontinuités entre les propriétés des ensembles connexes de points, qu'on appelle aussi technique de détection de contours. L'objectif est d'identifier les transitions de niveaux de gris dans l'image figure 1.7.



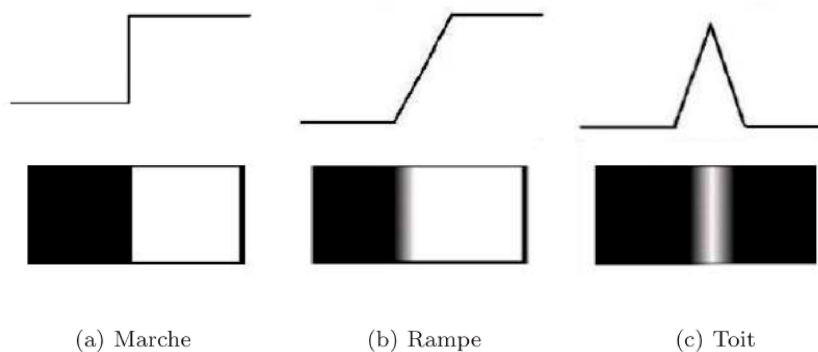


FIGURE 1.7 – Quelques modèles de contours[26]

Nous allons donner de façon très générale les différentes méthodes de détection de contours.

### 1.5.1.1 Méthodes dérivatives

Les méthodes dérivatives sont des techniques qui détectent et localisent immédiatement les discontinuités dans une image et ceci par différentiation numérique. Les techniques dérivatives reposent sur des opérateurs dérivatifs du premier et du second ordres.[12] : La méthode dite dérivée est la méthode la plus directe pour détecter et localiser les changements de niveaux de gris.

### 1.5.1.2 Opérateurs dérivatifs du premier ordre

Dans ce cas, les contours sont assimilés aux points de plus fort gradient dans l'image. Certains opérateurs utilisent des masques de convolution pour calculer le gradient comme c'est le cas des opérateurs de Prewitt, Sobel et Kirsch. D'autres opérateurs intègrent un filtre et une dérivation comme par exemple l'opérateur MDIF et NAFDIF. Ces deux opérateurs donnent des contours hachés dans le cas des transitions lentes ( supérieur à 5 pixels) et cela est du au fait que ces deux opérateurs travaillent sur des voisinages de taille 5\*5. Ces dérivées peuvent être établies selon les formules suivantes[25] :

$$\vec{\nabla}I = \frac{\partial I(x, y)}{\partial x} \vec{I}_x + \frac{\partial I(x, y)}{\partial y} \vec{I}_y \quad (1.2)$$

Où  $\hat{I}_x$  et  $\hat{I}_y$  sont des vecteurs unitaires suivant  $x$  et  $y$  respectivement. La figure 1.8 illustre quelques exemples d'opérateurs du premier ordre.

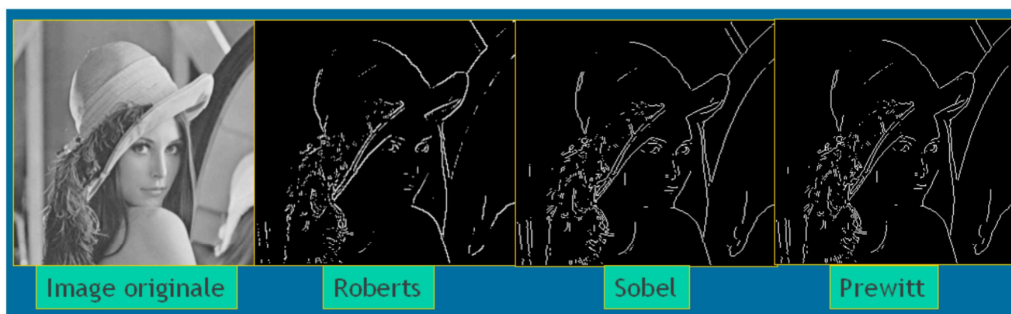


FIGURE 1.8 – Détection de contours par des filtres différents

### 1.5.1.3 Opérateurs dérivatifs du second ordre

L'opérateur *Laplacien* détecte les contours par les passages par zéro de la seconde dérivée de la fonction image. Le *Laplacien* a le désavantage d'être sensible au bruit beaucoup plus que les opérateurs du premier ordre. De façon classique, on peut dire que les opérateurs utilisant la dérivée seconde donnent des contours moins "lisses" que ceux fournis par les opérateurs employant le gradient. Dans la figure 1.9 suivante sera représentée la détection de contours par le *Laplacien*[1] :

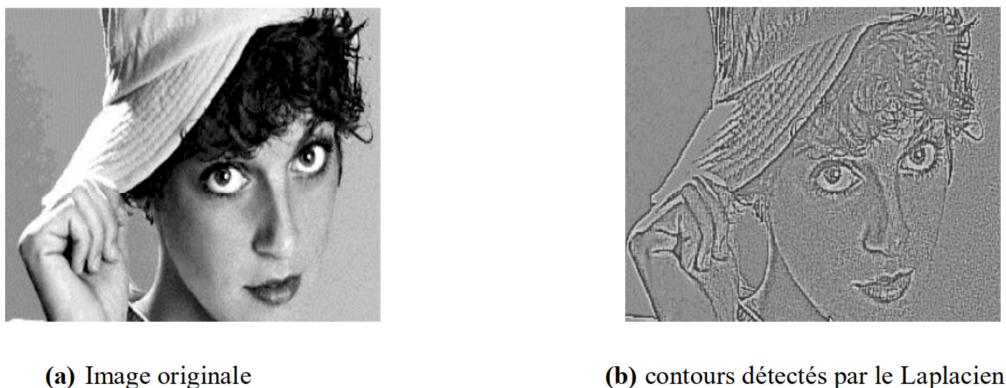


FIGURE 1.9 – Détection de contours par le *Laplacien*

## 1.5.2 Approches région

Dans l'approche région, on regroupe les pixels ayant les mêmes attributs. Les méthodes de segmentation par région aboutissent directement à une partition de l'image étant donné que chaque pixel sera affecté à une région unique. Les trois approches les plus utilisées sont :

- approche par division de régions.
- approche par fusion de régions.
- approche par division et fusion (split and merge).

### 1.5.2.1 Segmentation par division de région

Cette méthode divise l'image en régions homogènes selon un critère donné. Si le critère est valide, l'image est considérée comme segmentée. [17]. Un exemple de segmentation par division de région est donné sur la figure 1.10 qui suit.



FIGURE 1.10 – Décompositions successives des blocs

### 1.5.2.2 Segmentation par fusion

Les régions sont construites à partir de *germes* de la taille d'un ou quelques pixels auxquels on ajoute progressivement les pixels qui leurs sont connexes. Le processus de groupement de

pixels s'arrête lorsque le critère d'homogénéité n'est plus possible comme le montre la figure 1.11.

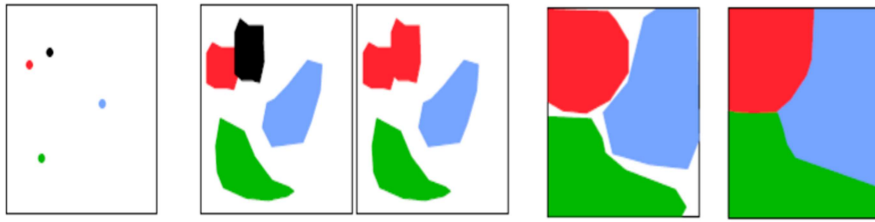


FIGURE 1.11 – Exemple de segmentation par fusion

### 1.5.2.3 Segmentation par division et fusion de région (split and merge)

Elle se base sur la combinaison des deux premières méthodes. Tout d'abord, divisez l'image en régions uniformes. Les régions voisines répondant au critère d'homogénéité sont ensuite fusionnées.

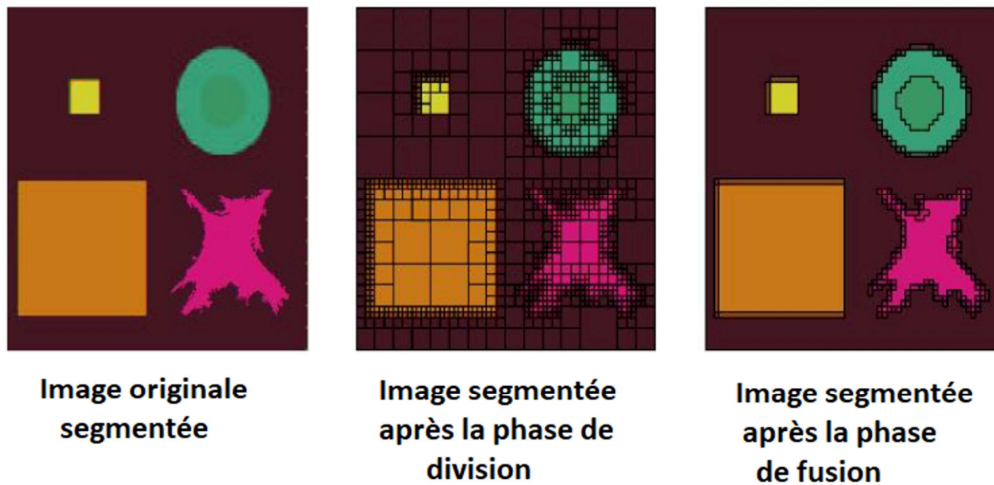


FIGURE 1.12 – Exemple de division-fusion de région

## 1.5.3 Segmentation par coopération régions-contours

Cette méthode est comme une mixture entre les deux concepts régions-contours. Son but est de remédier aux inconvénients des deux approches afin d'améliorer le résultat final. Il existe trois formes de coopération région-contour :

### 1.5.3.1 Coopération séquentielle

Les contours mal détectés par l'approche contour sont reconsidérés par l'approche région comme le montre l'organigramme suivant : [40].

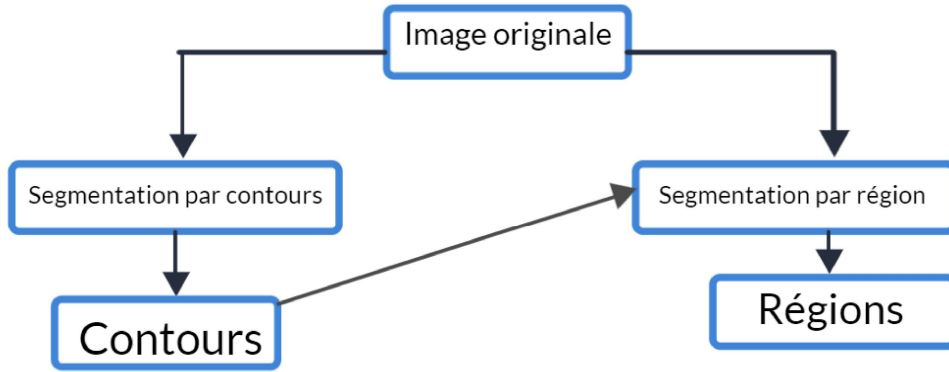


FIGURE 1.13 – Principe de la coopération séquentielle

### 1.5.3.2 Coopération mutuelle

Les deux techniques de segmentation sont exécutées en même temps et s'échangent des informations durant le processus de segmentation. Elle permet donc de prendre des décisions plus sûres et plus fiables[40].

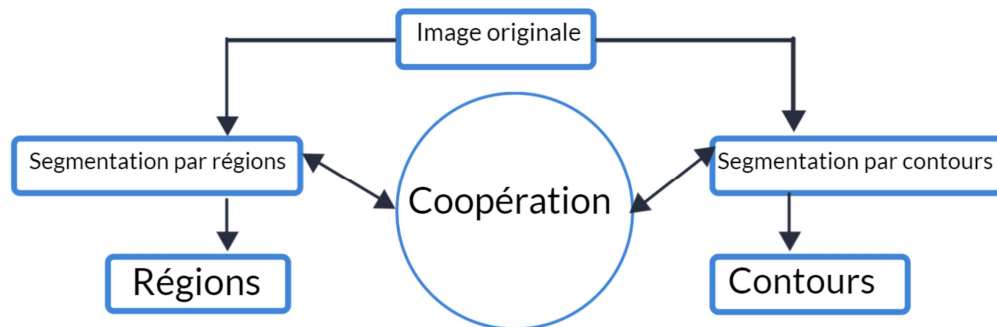


FIGURE 1.14 – Principe de la coopération mutuelle

### 1.5.3.3 Coopération des résultats

Ici, les deux types de segmentation sont réalisés en même temps et indépendamment. L'échange d'information se fait au niveau des résultats. Cette méthode donne de meilleurs résultats comparée aux précédentes. [40].

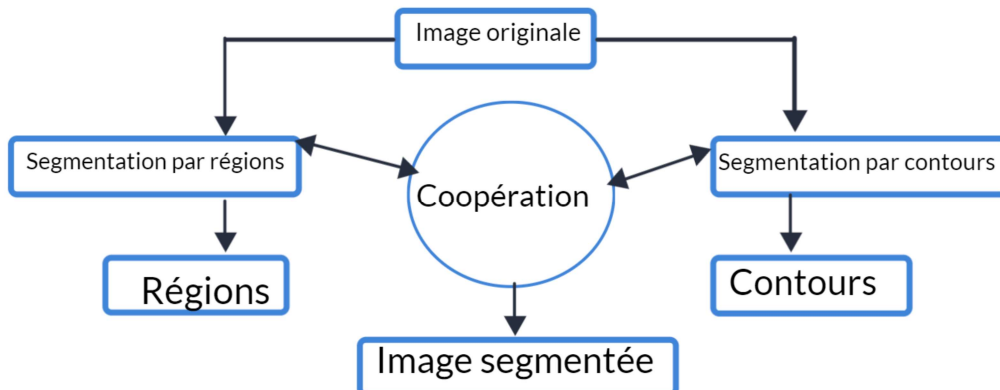


FIGURE 1.15 – Principe de la coopération des résultats

## 1.6 Intérêt de la segmentation d'images dans le domaine de l'automatique et l'informatique industrielle

Dans le domaine de l'informatique industrielle et de l'automatisation, la segmentation d'images joue un rôle crucial dans divers aspects, notamment la surveillance des processus de fabrication, le contrôle qualité, la reconnaissance de défauts, la navigation des robots, etc. Voici quelques contextes spécifiques où la segmentation d'images est utilisée :

La segmentation d'images est utilisée dans le contrôle qualité pour détecter et segmenter des défauts ou des anomalies sur des pièces ou des produits manufacturés.

Les robots industriels utilisent la vision par ordinateur et la segmentation d'images pour naviguer dans leur environnement, éviter les obstacles et interagir avec les objets.

Dans les lignes de production automatisées, la segmentation d'images peut être utilisée pour suivre et analyser le flux des matériaux, surveiller les étapes du processus de fabrication et détecter les erreurs ou les problèmes de fonctionnement.

La segmentation d'images est utilisée pour identifier et suivre les produits à travers différentes étapes de la chaîne logistique. Cela peut inclure la lecture de codes-barres, la reconnaissance de formes ou de caractéristiques spécifiques des produits.

La segmentation d'images est utilisée pour surveiller les zones de travail dangereuses et détecter les comportements non sécuritaires des travailleurs ou des machines.

## 1.7 Évaluation d'une méthode de segmentation

En vision artificielle, il est difficile de donner une appréciation sur le résultat obtenu et juger sa qualité. Les algorithmes de traitement d'images ne travaillent pas sur les mêmes types d'images, les sources d'acquisition sont différentes en fonction de l'environnement et le domaine d'application varie d'un problème à l'autre. L'évaluation d'un résultat de bas niveau, comme la segmentation, n'est pas aussi évidente que l'évaluation d'un résultat de haut niveau. Dans un système de vision de haut niveau, le but est défini d'une façon claire symboliquement ou numériquement vu la connaissance à priori du domaine de l'application. Cependant, dans le cas de la segmentation, une définition claire de l'objectif ne peut être donnée. La raison pour laquelle, l'évaluation de la segmentation reste fondée sur l'appréciation humaine.

## 1.8 Conclusion

Dans ce chapitre, nous avons abordé les principes fondamentaux de la segmentation d'images, une étape cruciale dans le traitement et l'analyse des images. Nous avons exploré les principales techniques de segmentation, telles que les méthodes basées sur les contours et celles basées sur les régions et même la classification, en soulignant leurs caractéristiques et leurs applications spécifiques.

Nous avons également discuté des approches combinées qui intègrent à la fois les informations de contours et de régions pour améliorer la précision de la segmentation. En résumé, nous avons essayé de fournir un aperçu des différentes stratégies de segmentation et de leur importance dans le traitement d'images ainsi que dans le domaine de l'automatique et de l'informatique industrielle.

# Chapitre 2

## Les réseaux de neurones

### 2.1 Introduction

Les réseaux neuronaux artificiels s'inspirent du fonctionnement du système nerveux humain. Leur création découle de la volonté de comprendre et de reproduire les capacités du cerveau, dans le but de concevoir des systèmes capables de remplacer l'humain dans des tâches complexes, telles que l'apprentissage, la mémorisation, l'intelligence, ou la généralisation, entre autres. Un réseau neuronal artificiel se compose de plusieurs unités appelées neurones formels, qui produisent une sortie basée sur un ensemble d'entrées pondérées. Ces neurones peuvent être interconnectés de diverses façons pour former une technique de traitement des données adaptée à une architecture neuronale spécifique, permettant de résoudre des problèmes dans des domaines variés tels que le contrôle des processus, la modélisation, la classification, la prédiction, le diagnostic ou encore la reconnaissance de formes.

Dans ce chapitre, nous présentons le concept des réseaux neuronaux artificiels en détaillant leurs différents composants, les fonctions d'activation, ainsi que leur organisation et leurs connexions, qui permettent de créer diverses structures de réseaux formels. Ces réseaux doivent être équipés d'une méthode permettant de collecter des informations depuis l'environnement extérieur, ou d'une règle d'apprentissage leur permettant de s'adapter aux conditions environnantes, en ajustant et mettant à jour leurs poids, lesquels représentent la force des connexions.

### 2.2 Les avantages des réseaux de neurones

Les réseaux de neurones formels présentent plusieurs avantages, à savoir :

- Les vitesses de calcul,
- le parallélisme et la distribution de l'information à travers toute la structure ( mémoire rapide),
- La grande capacité de résister vis à vis du mauvais fonctionnement d'un petit nombre d'éléments ( robustesse),
- L'adaptation de leur comportement en fonction de l'environnement : ils s'auto-ajustent pour produire des réponses consistantes.

## 2.3 Les composants fondamentaux et la modélisation d'un neurone artificiel

### 2.3.1 Le neurone biologique

Les neurones biologiques sont des cellules vivantes spécialisées dans le traitement de l'information du signal électrique. Ils sont interconnectés par des structures appelées axones, qui jouent un rôle crucial dans le comportement logique global du réseau neuronal. Ces axones transmettent les signaux électriques de la sortie d'un neurone vers l'entrée (synapses) d'un autre. Les neurones additionnent les signaux reçus en entrée et, en fonction du résultat de cette sommation, produisent un courant en sortie.[7].

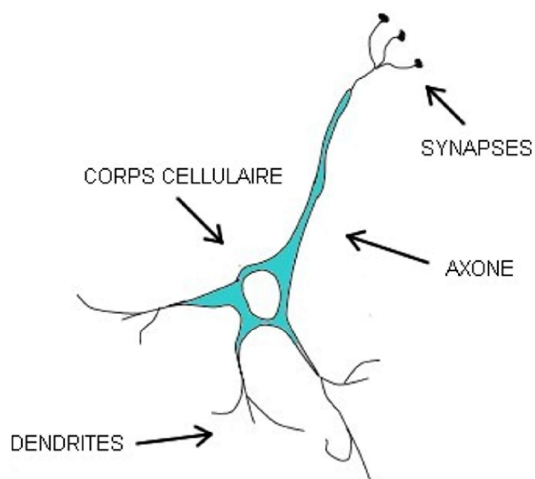


FIGURE 2.1 – Un neurone avec son arborisation dendritique

La structure d'un neurone est constituée de trois parties :

- . Le soma : ou corps cellulaire, situé au centre du neurone, responsable de l'activité nerveuse.
- . L'axone : relié au soma et électriquement actif, il transmet les impulsions vers les synapses.
- . Les dendrites : électriquement inactives, elles reçoivent les impulsions provenant d'autres neurones.

### 2.3.2 Le neurone artificiel

Le neurone artificiel, constitue l'élément fondamental du fonctionnement d'un réseau de neurones artificiels. Il reçoit un ensemble variable d'entrées provenant des neurones de la couche précédente, désignés comme les neurones d'amont. Chaque entrée est pondérée par un coefficient "w" qui représente la force de la connexion. À partir de ces entrées pondérées, la cellule artificielle génère une seule sortie qui se divise ensuite pour alimenter un ensemble variable de neurones situés dans la couche suivante. Chaque connexion est également associée à un poids, déterminant ainsi l'impact relatif de chaque entrée sur la sortie de la cellule artificielle. Le modèle mathématique d'un neurone artificiel est illustré dans la Figure (2.2)[30].

Les réseaux neuronaux formels sont nés d'une tentative de modélisation mathématique du cerveau humain. Les premiers travaux, réalisés en 1943 par McCulloch et Pitts, introduisent un modèle relativement simple de neurones et explorent les potentialités de ce modèle[7].

La modélisation d'un neurone biologique en neurone artificiel consiste à créer un système de réseau neuronal artificiel qui reproduit les éléments clés du neurone biologique. Chaque composant du neurone biologique est modélisé par un élément correspondant dans le neurone artificiel.

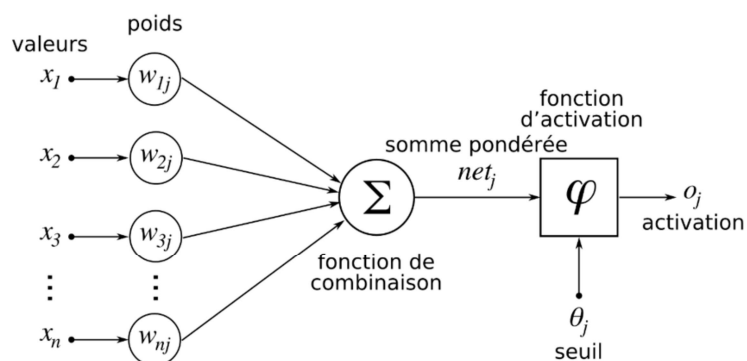


FIGURE 2.2 – Structure d'un neurone artificiel

Le tableau suivant résume les correspondances entre les éléments du neurone biologique et du neurone artificiel :

Neurone biologique	Neurone artificiel
Synapses	Poids de connection
Axones	Signal de sortie
Dendrite	Signal d'entrée
Somma	Fonction d'activation

TABLE 2.1 – Modélisation des éléments du neurone biologique

La fonction d'activation détermine l'état interne du neurone en fonction de la somme de ses entrées. Les fonctions d'activation les plus couramment utilisées sont les suivantes [5] :

- tout ou rien ;
- fonction signe ;
- fonction seuil positif ou négatif ;
- fonction affine ;
- fonction de saturation ;
- sigmoïde ;
- fonction arctangente ;
- fonction radiale de base de type gaussien.

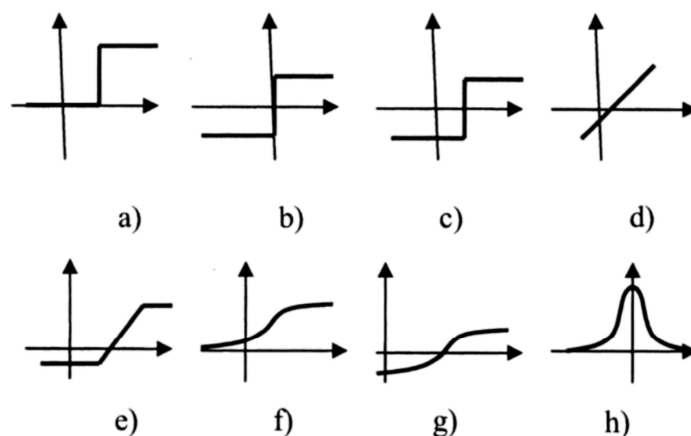


FIGURE 2.3 – Fonctions d'activation les plus utilisées



## 2.4 Les réseaux de neurones

Les capacités d'un neurone seul étant très limitées, il est souvent nécessaire de relier plusieurs neurones entre eux afin de pouvoir implémenter des fonctions plus complexes. Cet ensemble de neurones interconnectés constitue ce que l'on appelle un réseau de neurones. Ce réseau est composé de neurones formels organisés en couches parallèles, qui transmettent l'information à travers différents nœuds. Les interactions entre ces nœuds sont déterminées par les poids des interconnexions. Un algorithme d'apprentissage permet d'ajuster ces poids afin d'atteindre les performances désirées. Les réseaux neuronaux artificiels sont des systèmes fortement interconnectés de processeurs simples opérant en parallèle. Chaque processeur génère une sortie unique basée sur les informations qu'il reçoit. Toute structure hiérarchique de réseaux constitue également un réseau [7].

## 2.5 Les différentes structures des réseaux de neurones

### 2.5.1 Réseaux de neurones multicouches

Les neurones au sein des réseaux de neurones sont disposés en couches, sans connexion entre les neurones d'une même couche. Les connexions se font uniquement avec les neurones des couches suivantes. En général, chaque neurone est relié à tous les neurones de la couche suivante et uniquement à ceux-ci. Cette organisation introduit la notion de direction de flux (ou d'activation) de l'information dans le réseau, permettant ainsi de définir les neurones d'entrée et les neurones de sortie. Par extension, l'ensemble des neurones d'entrée est désigné comme couche d'entrée, tandis que l'ensemble des neurones de sortie est appelé couche de sortie. Les couches qui n'interagissent pas directement avec l'environnement extérieur sont qualifiées de couches cachées [8].

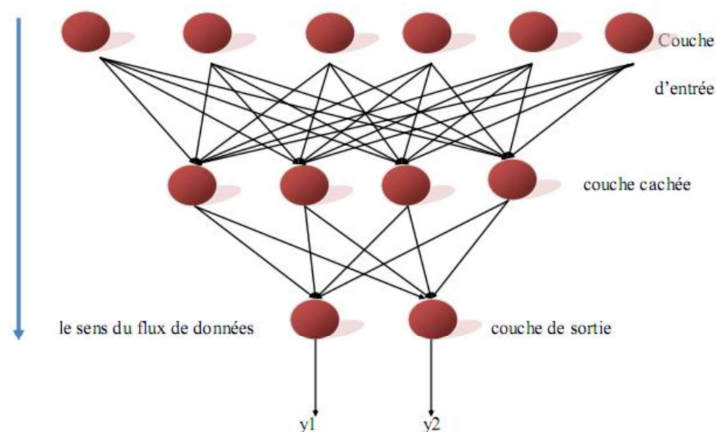


FIGURE 2.4 – Réseau de neurones multicouches.

### 2.5.2 Réseau à connexions locales

Il s'agit d'une structure multicouche, comme la rétine, qui conserve une certaine topologie. Chaque neurone est connecté à un petit nombre local de neurones dans la couche suivante. Le nombre de connexions est donc inférieur à celui des réseaux multicouches classiques. Ce tissu permet des connexions plus spécifiques et restreintes, similaires à la fonction de la rétine[32].

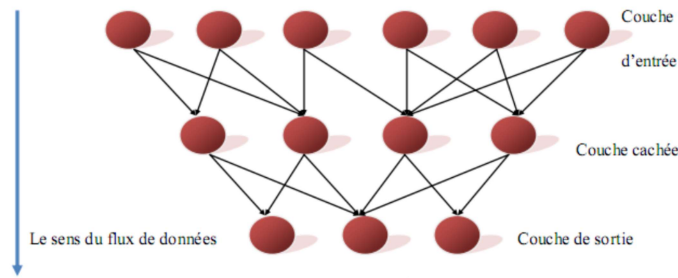


FIGURE 2.5 – Réseau à connexions locales

### 2.5.3 Réseaux à connexions récurrentes

Les connexions circulaires reculent par rapport à la direction de propagation définie dans le réseau multicouches. Ces connexions sont généralement locales.

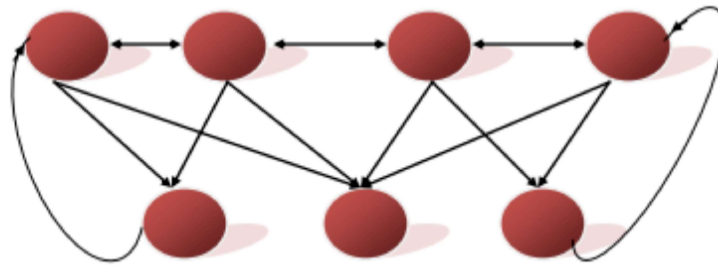


FIGURE 2.6 – Réseau à connexions récurrentes

### 2.5.4 Réseaux à connexions complètes

La structure d'interconnexion la plus générale, illustrée par la figure (2.7), indique que chaque neurone est relié à tous les autres neurones du réseau, y compris à lui-même. Cette interconnexion intégrale favorise une communication et une interaction directe entre tous les neurones, facilitant ainsi la circulation globale des informations au sein du système [32].

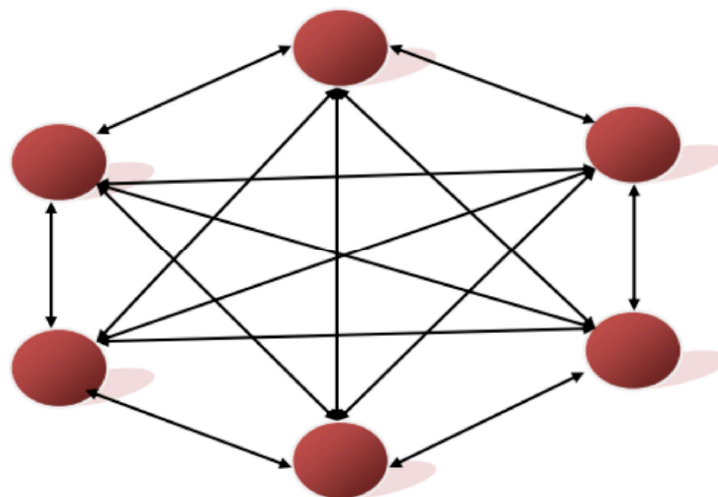


FIGURE 2.7 – Réseau à connexions complètes

## 2.6 Les réseaux de neurones convolutifs

Les réseaux neuronaux convolutifs (CNN) ont été proposés par Lecun et al. [27]. Ce qui distingue les CNN est l'utilisation de l'opération de convolution dans les premières couches intermédiaires du réseau. À l'origine, cette opération était utilisée comme filtre dans les domaines de l'image et du son pour mettre en évidence des motifs ou réduire certains types de bruit. Dans le cas des CNN, le modèle apprend lui-même les filtres associés aux différentes convolutions, ce qui permet de mettre en avant les motifs présents dans les données d'entrée utilisées dans les couches suivantes. Un CNN classique est généralement constitué de quatre types de couches :

- . Les couches convolutives, qui appliquent plusieurs opérations de convolution sur la même entrée,
- . Les couches de pooling,
- . Les couches d'activation,
- . Les couches entièrement connectées.

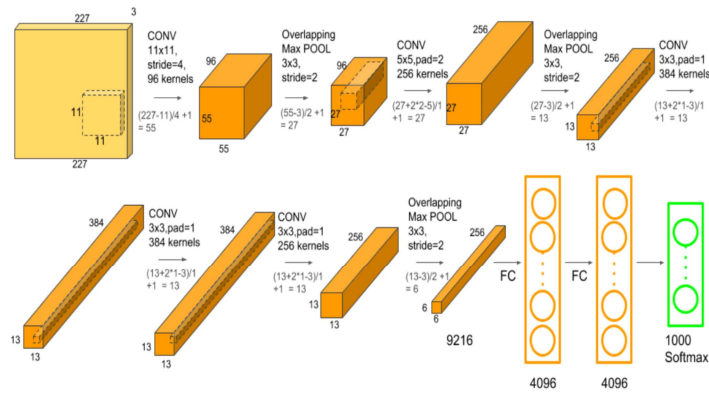


FIGURE 2.8 – Exemple de CNN appelé AlexNet[29]

Dans cette représentation, les neurones sont organisés selon les dimensions de largeur, de hauteur et de profondeur. À la différence des couches entièrement connectées, les couches convolutives préservent la cohérence spatiale des informations. Chaque dimension est spécifiée dans la figure(2.8). En entrée, le réseau neuronal reçoit une image de  $227 \times 227$  pixels comportant 3 canaux de couleur.

### 2.6.1 Couches convolutives

À l'origine, l'opération de convolution était appliquée à des données temporelles (comme le son) ou spatiales (comme les images) en tant que filtre linéaire. Dans cette section, nous examinerons une opération de convolution 2D, utilisée pour des données telles que des images  $X = (x_{i,j,z})$  où  $1 \leq i \leq h$ ,  $1 \leq j \leq l$ ,  $1 \leq z \leq c$ , avec  $h \times l$  représentant les dimensions de l'image et  $c$  le nombre de canaux. Cette opération est définie par un noyau  $A = (a_{i,j,z,k})$  pour  $1 \leq i \leq m$ ,  $1 \leq j \leq n$ ,  $1 \leq z \leq c$ ,  $1 \leq k \leq f$ , où  $m \times n$  est la taille du filtre et  $f$  est le nombre total de filtres, en plus d'un biais  $\beta \in \mathbb{R}^f$ .

La sortie de l'opération de convolution  $Y \in \mathbb{R}^{h \times l \times f}$  est calculée de la manière suivante :

$$y_{i,j,k} = \sum_{i'=0}^m \sum_{j'=0}^n \sum_{z'=0}^c x_{i+i',j+j',z'} \cdot a_{i',j',z',k} + \beta_k$$

Pour simplifier cette formule, nous n'avons pas pris en compte le traitement des bords de l'image d'entrée lors de l'opération de convolution. Les convolutions peuvent également être

appliquées à des données unidimensionnelles (comme le son [4]) ou tridimensionnelles (comme la vidéo ou les scans 3D).

## 2.6.2 l'opération de mise en commun (Pooling)

Les couches convolutives peuvent être suivies d'opérations de pooling, qui réduisent la dimensionnalité des données en consolidant les informations des neurones adjacents. Cette opération se fait à l'aide d'une fenêtre glissante qui parcourt les neurones, appliquant une fonction de pooling, comme le max pooling ou le average pooling. Un exemple de cette opération est montré dans la figure (2.9).

Les couches entièrement connectées et d'activation dans les CNN fonctionnent de manière similaire à celles des réseaux de neurones multicouches (MLP), avec les premières souvent placées à la fin du réseau, juste avant la couche de sortie. Elles établissent des corrélations entre les motifs détectés par les couches convolutives précédentes.

Les couches d'activation, qui sont ajoutées après chaque couche convolutive et chaque couche entièrement connectée, ne contiennent pas de connexions à apprendre, contrairement aux couches neuronales traditionnelles. Bien que le nombre de paramètres à ajuster dans les CNN soit généralement inférieur à celui des MLP, le volume d'opérations est plus élevé, car les filtres de petite taille sont partagés entre les neurones de la couche de sortie. Les CNN sont particulièrement efficaces dans le domaine de l'imagerie, surpassant d'autres techniques d'apprentissage[41].

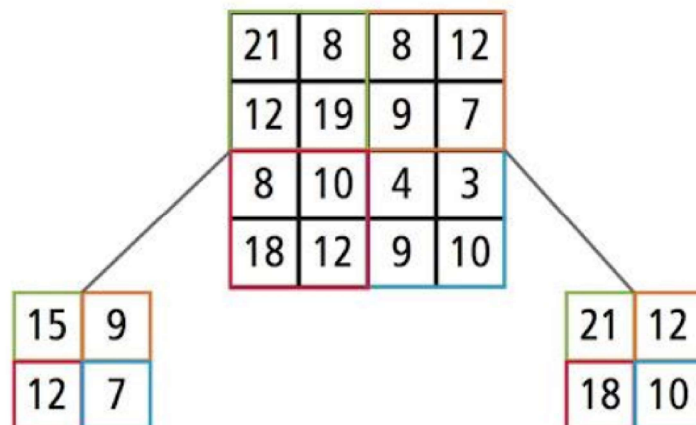


FIGURE 2.9 – Exemple d'opération de regroupement avec le maximum de Pooling (à droite) et la moyenne Pooling (à gauche)

La fenêtre coulissante mesure  $2 \times 2$  et se déplace de 2 par 2 le long de l'axe des y et de 2 par 2 le long de l'axe des x.

## 2.7 L'apprentissage des réseaux de neurones

Pour qu'un réseau puisse générer des résultats, il doit d'abord passer par une phase d'apprentissage. L'apprentissage est réalisé par application d'un ensemble d'entrées, les poids seront alors ajustés par une certaine procédure. Durant cette phase d'apprentissage, les poids du réseau convergent graduellement vers des valeurs de façon à produire le vecteur de sortie désiré. Les algorithmes d'apprentissage sont catégorisés comme supervisés ou non-supervisés, et ceci en fonction de la connaissance a priori ou non du vecteur de sortie[42].

### 2.7.1 Apprentissage supervisé

Dans l'algorithme supervisé, on a besoin de deux vecteurs : le vecteur d'entrée et le vecteur cible ( target vector) et ensemble, ces deux vecteurs sont appelés " la paire d'apprentissage". Le vecteur d'entrée est appliqué, la sortie est calculée et comparée au vecteur cible. La différence ( l'erreur) sera réfléchié à travers le réseau et les poids seront modifiés par un algorithme qui tend à minimiser cette erreur. Cette opération sera répétée jusqu'à ce que l'erreur devienne très faible[42].

### 2.7.2 Apprentissage non-supervisé

L'apprentissage supervisé est biologiquement impossible : le cerveau ne peut pas avoir un mécanisme qui lui permet de comparer les sorties présentes à celles désirées, et si c'est le cas, d'où viennent alors les ensembles de sorties désirées? L'apprentissage non-supervisé est une approche plus réaliste du système biologique humain. Ce type d'apprentissage, développé par Kohonen '(1984) et beaucoup d'autres, n'a pas besoin du vecteur cible pour produire les sorties et donc, pas de comparaison pour déterminer les bonnes réponses. dans ce cas, l'apprentissage consiste uniquement en les vecteurs d'entrées. L'algorithme de cet apprentissage modifie les poids du réseau pour produire en sortie des vecteurs consistants. Autrement dit, le réseau s'auto-ajuste pour déterminer ses propres propriétés[42].

### 2.7.3 Apprentissage profond ( Deep Learning)

Un réseau de neurones profond ( Deep Neural Network) est composé de plusieurs couches de neurones. L'entraînement des réseaux profonds nécessite de grandes quantités de données et de puissance de calcul. Il consiste à ajuster les poids des connexions pour améliorer la précision du modèle. Cela se fait généralement par rétro-propagation, une méthode qui ajuste les poids en fonction de l'erreur de prédiction.

## 2.8 Algorithme d'apprentissage

La majorité des algorithmes d'apprentissage ont été développés à partir des résultats de D.O.Hebb ( 1961) qui utilise un modèle non-supervisé basé sur la modification des synapses reliant les neurones. La règle de Hebb consiste à faire accroître le poids synaptique  $w_{ij}$  chaque fois que les neurones  $i$  et  $j$  sont simultanément activés. Le réseau de neurone artificiel augmente ses poids en fonction des niveaux de l'excitation de la source et des neurones de destination. Symboliquement on a :

$$w_{ij}(n + 1) = w_{ij}(n) + \alpha \cdot out_i \cdot out_j \quad (2.1)$$

où :

- $w_{ij}(n + 1)$  : poids du neurone  $i$  vers le neurone  $j$  après ajustement,
- $w_{ij}(n)$  : poids du neurone  $i$  vers le neurone  $j$  avant ajustement,
- $\alpha$  : coefficient d'apprentissage,
- $out_i$  : sortie du neurone  $i$  vers l'entrée  $j$ ,
- $out_j$  : sortie du neurone  $j$ .

## 2.9 Application des réseaux de neurones au traitement d'images

Les réseaux de neurones formels, ont trouvés une large utilisation dans le domaine du traitement d'images et plus particulièrement, dans la segmentation d'images. Cette nouvelle technologie a pu s'adapter au monde réel, qui ne peut être modélisé que par des équations mathématiques consistantes, grâce à son caractère d'auto-ajustement.

Les CNN :

- sont utilisés pour attribuer des étiquettes à des images en fonction de leur contenu. LeNet, AlexNet, VGG, ResNet, et Inception sont des architectures qui ont été conçues pour améliorer la précision de la classification d'image[24].

- peuvent détecter des objets dans une image et localiser leur position à l'aide de bounding boxes. Des modèles comme R-CNN, Fast R-CNN, Faster R-CNN, et YOLO sont populaires pour cette tâche[35].

- comme U-Net et SegNet sont spécialement conçus pour la segmentation d'image qui consiste à partitionner une image en différentes régions ou objets[36].

- sont également utilisés pour améliorer la qualité des images en termes de réduction du bruit, de super-résolution, et d'augmentation de la clarté[14].

- peuvent être utilisés pour détecter les contours dans les images, ce qui est utile pour la reconnaissance de formes et la segmentation[43].

## 2.10 Conclusion

Ce chapitre était un tour d'horizon sur les concepts de base des réseaux de neurones, leurs principes de fonctionnement et surtout leurs applications dans différents domaines y compris le traitement d'images. Mais étant donné que cette nouvelle technologie est toujours en cours de recherche, afin de simuler de plus en plus l'intelligence humaine, il n'y a aucun algorithme universel ni aucune règle précise pour déterminer le nombre de couches ainsi que le nombre de neurones par couches dans le cadre d'une application donnée. Tout dépend du domaine d'application, du besoin de l'utilisateur, des conditions de l'environnement, du but recherché,... Le problème reste posé et seuls les euristiques sont utilisées pour trouver des solutions intelligentes.

# Chapitre 3

## La logique floue

### 3.1 Introduction

La logique classique, développée par Aristote, est basée sur le principe du tiers exclu : une proposition ne peut être que vraie ou fausse[13]. Cette logique binaire est efficace pour traiter des problèmes bien définis, mais elle s'avère souvent inadaptée à des situations complexes et incertaines qui caractérisent le monde réel.

En 1965, Lotfi Zadeh a introduit la logique floue, une extension de la logique classique qui permet de représenter et de traiter l'information vague et imprécise. Contrairement à la logique classique, la logique floue permet aux propositions d'être partiellement vraies ou fausses, avec des degrés d'appartenance compris entre 0 et 1.

Le raisonnement flou est basé sur l'utilisation de règles floues, qui sont des propositions conditionnelles de la forme "Si A alors B", où A et B sont des ensembles flous. Les règles floues sont exprimées en langage naturel, ce qui les rend intuitives et faciles à comprendre.

Le processus de raisonnement flou implique l'évaluation des degrés d'appartenance des prémisses (la partie "Si" de la règle floue) et la propagation de ces degrés d'appartenance à la conclusion (la partie "alors" de la règle floue). L'agrégation des résultats obtenus à partir de plusieurs règles floues permet de déduire une conclusion globale[44].

Dans ce chapitre, nous donnons un aperçu sur la théorie des sous-ensembles flous en présentant leur concepts de base ainsi que les opérations les plus couramment utilisées. Nous donnons également un aperçu sur l'utilisation de la logique floue en traitement d'images.

### 3.2 Notions fondamentales sur les ensembles flous

#### 3.2.1 Définition d'un ensemble flou

Un sous-ensemble flou  $A$  au sein de l'univers  $X$  est caractérisé par une fonction d'appartenance  $\mu_A$ , qui attribue à chaque élément  $x$  de  $X$  un degré  $\mu_A(x)$  compris entre 0 et 1, représentant le niveau d'appartenance de  $x$  à  $A$ .

**Exemple :** Considérons  $X$  comme l'ensemble des âges possibles d'une personne. On peut alors définir le sous-ensemble flou  $A$  représentant les personnes jeunes. Le degré d'appartenance d'une personne de  $x$  ans à ce sous-ensemble flou  $A$  augmente à mesure que son âge diminue.

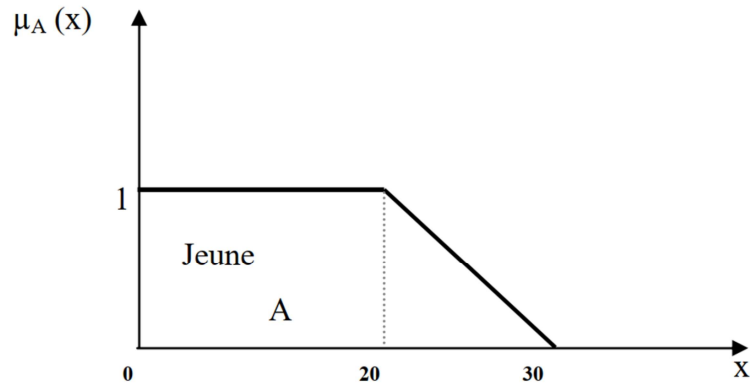


FIGURE 3.1 – Sous ensemble flou A représentant la population jeune, l'âge est représenté par la variable x

Soit un sous-ensemble flou  $A$  au sein de l'univers  $X$ , défini par une fonction d'appartenance

$$\mu_A : X \rightarrow [0, 1]$$

qui associe à chaque élément  $x$  de  $X$  le degré  $\mu_A(x)$  :

$$x \mapsto \mu_A(x)$$

Le sous-ensemble flou  $A$  peut alors s'écrire sous la forme :

$$A = \{(x, \mu_A(x)) \mid x \in X\}$$

La fonction  $\mu_A : X \rightarrow [0, 1]$  est désignée comme fonction d'appartenance, et  $\mu_A(x)$  représente le degré d'appartenance de l'élément  $x$  au sous-ensemble flou  $A$ .

En fonction des valeurs de  $\mu_A(x)$ , nous pouvons distinguer trois cas :

$$\begin{aligned}
 A &= \{(x, \mu_A(x)) \mid x \in X\}, \\
 \mu_A(x) = 1 &\text{ correspond à l'appartenance totale de } x \text{ à } A, \\
 \mu_A(x) = 0 &\text{ indique la non-appartenance totale de } x \text{ à } A, \\
 0 < \mu_A(x) < 1 &\text{ traduit une appartenance partielle de } x \text{ à } A.
 \end{aligned}$$

Dans le cas où  $\mu_A$  ne prend que les valeurs 0 ou 1, le sous-ensemble flou  $A$  devient équivalent à un sous-ensemble classique de  $X$ .

### 3.2.2 Caractéristique d'un ensemble flou[18]

Diverses notions complémentaires, couramment utilisées, permettent de mieux distinguer les sous-ensembles flous des ensembles classiques :

#### 3.2.2.1 Support

Soit  $A$  un sous-ensemble flou de l'univers  $X$ . Le support du sous-ensemble flou  $A$  est défini comme un sous-ensemble classique de  $X$  dont chaque élément a un degré d'appartenance non nul par rapport à  $A$ .

$$\text{Support}(A) = \{x \in X \mid \mu_A(x) \neq 0\}$$



### 3.2.2.2 Noyau

Le noyau d'un sous-ensemble flou  $A$  de  $X$  est un sous-ensemble classique de  $X$  où chaque élément a un degré d'appartenance égal à 1 :

$$\text{Noyau}(A) = \{x \in X \mid \mu_A(x) = 1\}$$

Si  $\text{Noyau}(A) = \emptyset$ , nous disons que le sous-ensemble flou  $A$  est normal ou normalisé; sinon, il est considéré comme sous-normal.

### 3.2.2.3 Hauteur

La hauteur d'un sous-ensemble flou  $A$  de  $X$  correspond à la valeur maximale de la fonction d'appartenance, soit le plus grand degré d'appartenance à  $A$  :

$$h(A) = \max(\mu_A(x))$$

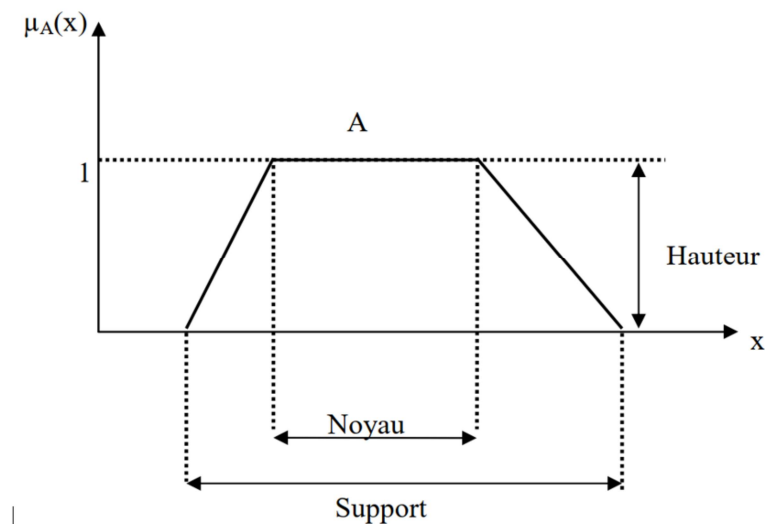


FIGURE 3.2 – Support, Noyau et Hauteur d'un sous ensemble flou

Un sous ensemble flou est normalisé si

$$h(A) = 1.$$

### 3.2.2.4 Cardinalité

La cardinalité d'un sous ensemble flou  $A$  de  $X$  est définie par :

$$|A| = \sum_{x \in X} \mu_A(x) \quad \text{si } X \text{ est fini.}$$

qui indique le degré avec lequel les éléments de  $X$  appartiennent à  $A$ .

### 3.2.2.5 Sous ensemble convexe

Un sous ensemble flou  $A$  de  $X$  est convexe si, pour tout couple d'éléments  $a$  et  $b$  de  $X$  et pour tout nombre

$$\gamma \in [0, 1]$$

, la fonction d'appartenance de A vérifie :

$$\mu_A(\gamma a + (1 - \gamma)b) \geq \min(\mu_A(a), \mu_A(b)) \quad (3.1)$$

Si A et B sont deux sous ensembles convexes flous de X, leur intersection est convexe.

Les sous ensembles flous les plus répandus sont ceux qui ont une fonction d'appartenance sans rupture brusque, c'est-à-dire régulière (sous ensemble convexe).

### 3.3 Les éléments de base de la logique floue

Les éléments de base de la logique floue sont :

- les variables linguistiques,
- Les déductions floues,
- Les opérateurs.

#### 3.3.1 Les variables linguistiques

L'élément de base dans le raisonnement approximatif est la notion de la variable linguistique. Elle sert à décrire une certaine situation ou un phénomène dont la valeur précise, dans le domaine de définition, est inconnue. dans son livre " The concept of Linguistic Variable and its Application to Approximate Reasoning", Zadeh dit :

" *By a linguistic variable we mean a variable whose values are words or sentences in a natural or artificial language. For example, Age is a linguistic variable if its values are linguistic rather than numerical, i.e., young, not young, very young, quite young, old, not very old and not very young, etc., rather than 20, 21, 22, 23 ..*" [15]

Une variable linguistique possède une valeur linguistique qui peut être formulée par le langage naturel, tel que petit, grand, clair, foncé,... Dans ce cas, les expressions sont beaucoup plus utilisées que les valeurs numériques. Dans une image, on peut considéré que *le niveau de gris* d'un pixel est une variable linguistique dont l'ensemble de termes peut être : T(niveau de gris) = blanc, gris claire, gris foncé, noir. Dans une image à 256 niveaux de gris, on considère que le niveau de gris 1 est le blanc, le niveau de gris 256 est le noir et le niveau de gris 128 est le gris. Si on est en dessous du niveau de gris 128, c'est le gris claire et en dessus de 128, c'est le gris foncé.

##### 3.3.1.1 Fonction d'appartenance[19]

Comme son nom l'indique, une fonction d'appartenance permet de définir le degré d'appartenance d'un objet à un ensemble donné. Elle constitue l'un des paramètres de conception de la logique floue à savoir la définition d'un sous-ensemble flou sur un univers de discours.

##### 3.3.1.2 Exemple de fonction d'appartenance

Plusieurs fonctions d'appartenance sont utilisées dans la logique des ensembles flous. Les plus répandues sont les fonctions d'appartenance trapézoïdale, exponentielle, gaussienne et singleton.

- Fonction d'appartenance trapézoïdale  
Elle est donnée par l'équation suivante :

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a - c \text{ ou } x \geq b + d \\ \frac{x}{c} + (1 - \frac{a}{c}) & \text{si } a - c < x < a \\ 1 & \text{si } a \leq x \leq b \\ -\frac{x}{d} + (1 + \frac{b}{d}) & \text{si } b < x < b + d \end{cases}$$

Sa courbe est donnée comme suit :

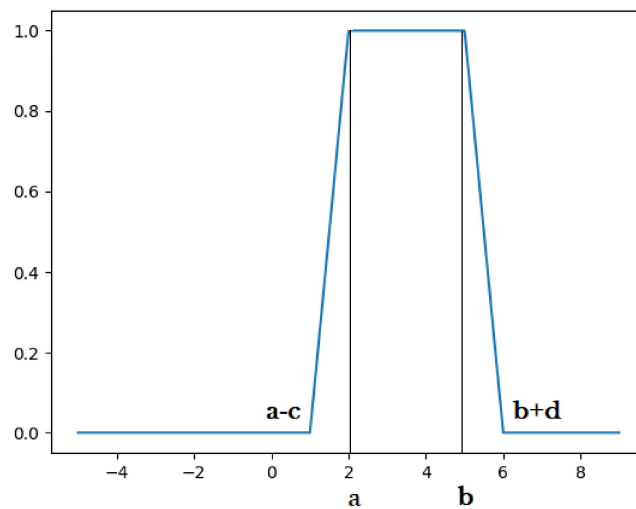


FIGURE 3.3 – Fonction d'appartenance trapézoïdale

Dans le cas où  $a=b$  la fonction est dite triangulaire et sa courbe est représentée par la figure suivante :

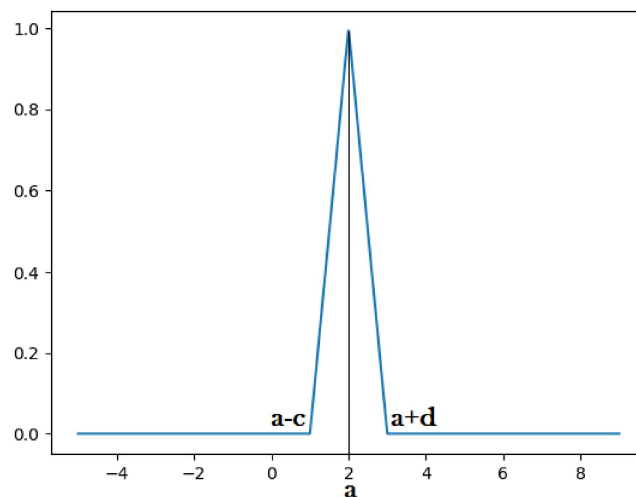


FIGURE 3.4 – Fonction d'appartenance triangulaire

— Fonction d'appartenance exponentielle  
Elle est donnée par la relation suivante :

$$\mu_A(x) = \begin{cases} \exp(x - a) & \text{si } x < a \\ 1 & \text{si } a \leq x \leq b \\ \exp(-x + b) & \text{si } x > b \end{cases}$$

Sa courbe est donnée comme suit :

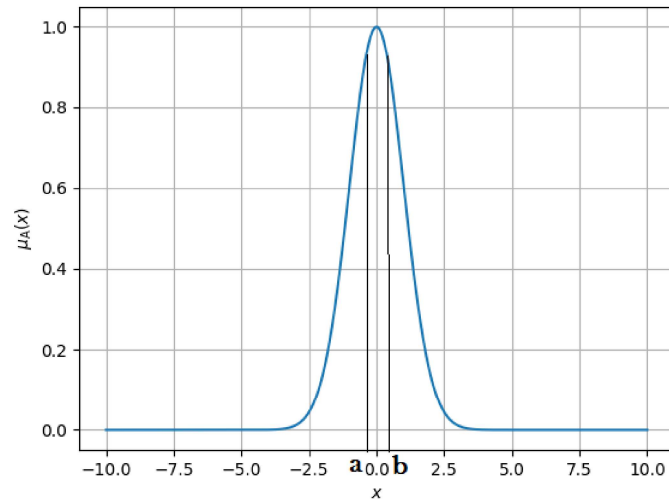


FIGURE 3.5 – Fonction d'appartenance exponentielle

— Fonction d'appartenance gaussienne

Elle est donnée sous la forme :

$$\mu_A(x) = \exp \left[ -\frac{(x - m)^2}{2\sigma^2} \right]$$

Sa courbe est donnée comme suite :

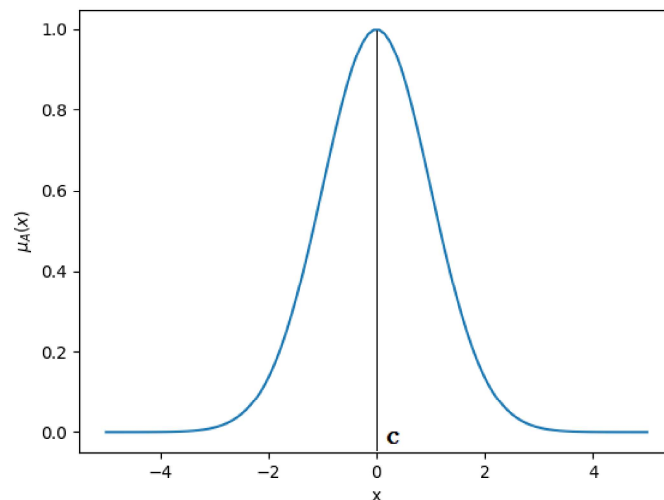


FIGURE 3.6 – Fonction d'appartenance gaussienne

### 3.3.2 Les déductions floues ( inférences)

Les variables linguistiques définies par leurs fonctions d'appartenance, sont souvent liées entre elles par des règles afin de tirer des déductions. On parle alors de déductions floues ou inférences. Il existe deux grandes classes d'inférences : les inférences à une seule règle et les inférences à plusieurs règles[6].

#### 3.3.2.1 Les inférences à une seule règle

Parfois on a besoin de comparer plusieurs concurrents ( objets ou personnes) dans une certaine situation et en choisir l'optimum. Si  $x_1, x_2, \dots, x_n$  sont des variables linguistiques pour évaluer une certaine situation, le critère  $Y$  pour la prise de décision peut être formulé par une règle, comme par exemple  $Y = [x_1 \text{ ou } (x_2 \text{ et } x_3) \text{ ou } \dots ] \text{ et } x_n$ . La règle de décision dépend du problème en question[6]. A chaque variable est attribuée une fonction d'appartenance  $U_{x_1}, U_{x_2}, U_{x_3}, \dots, U_{x_n}$  dont les valeurs numériques ou les facteurs d'appartenances sont différents pour chaque concurrent. Le concurrent gagnant est celui dont le facteur d'appartenance est le maximum.

#### 3.3.2.2 Les inférences à plusieurs règles

Dans ce cas, une ou plusieurs variables floues nécessitent une prise de décision différente. Les règles peuvent être exprimées sous la forme :

- Opération = SI condition 1, ALORS opération 1, ou
- SI condition 2, ALORS opération 2, ou
- 
- 
- 
- SI condition m, ALORS opération m

Dans ce cas d'inférences, les variables sont liées entre elles par des opérateurs flous de forme **ET** ou **OU**[6]. L'opération de décision doit tenir compte des différentes conditions.

### 3.3.3 Les opérations sur les ensembles flous[45]

les différentes opérations sont représentées sur la figure(3.7). Soient  $A$  et  $B$  deux ensembles flous définis sur l'univers de discours  $U$ , ayant respectivement les fonctions d'appartenances  $\mu_A$  et  $\mu_B$  :

1. **Égalité** :  $A$  et  $B$  sont égaux, si leurs FAs prennent la même valeur pour tout élément de  $U$ .

$$A = B \text{ si } \forall x \in U, \mu_A(x) = \mu_B(x)$$

2. **Inclusion** :  $A$  est inclus dans  $B$ ,

$$A \subseteq B \text{ si } \forall x \in U, \mu_A(x) \leq \mu_B(x)$$

Union :  $A \cup B$  est définie comme le plus petit ensemble contenant à la fois  $A$  et  $B$ , tel que :

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x)$$

3. **Intersection** :  $A \cap B$  est définie comme le plus grand ensemble contenu à la fois dans  $A$  et  $B$ , tel que :

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x)$$

4. **Complément** :  $\bar{A}$  est un ensemble flou dont la FA est définie par :

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

**Illustrations des opérations floues :**

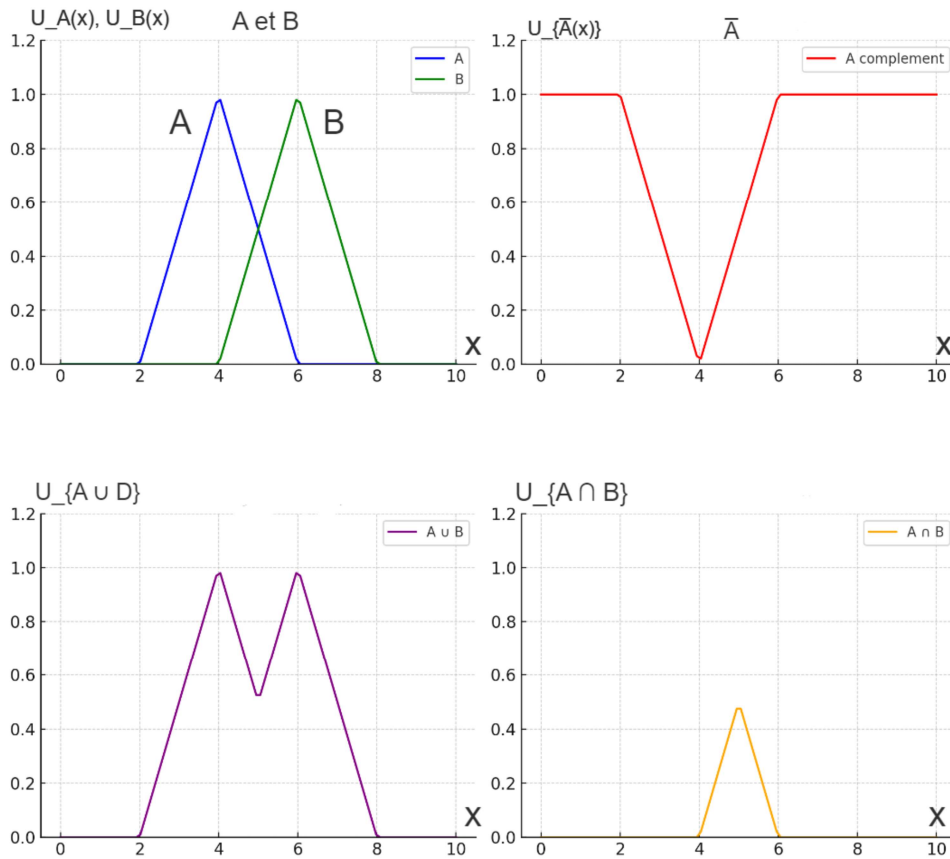


FIGURE 3.7 – Opérations floues

### 3.4 Segmentation floue

La segmentation d'image consiste à diviser une image en régions distinctes, caractérisées par des limites nettes (en opposition à des limites floues). Chaque pixel est attribué à l'une des deux régions, mais un pixel situé au « centre » d'une région uniforme a plus de chances d'appartenir à celle-ci qu'un pixel situé à la périphérie d'une région en croissance, par exemple. Un pixel proche d'une bordure est moins susceptible d'appartenir à la même zone qu'un pixel situé près de l'autre côté de cette bordure. Ainsi, il est souvent recommandé de considérer des régions comme des ensembles flous, où chaque pixel a un degré d'appartenance à chaque région. Les degrés d'appartenance des pixels peuvent être calculés en utilisant un double critère de proximité et de différence de gradient par rapport aux germes des régions, reposant sur la « distance topographique » définie dans [34].

Soit  $\Omega$  un référentiel fini (ensemble de  $N$  pixels). Une segmentation floue de  $\Omega$  est définie par un ensemble de  $M$  régions floues  $R_j$  dont les supports sont inclus dans  $\Omega$  et qui respectent les deux axiomes suivants : Si  $U_j(s)$  représente le degré d'appartenance du pixel  $s$  à la région  $R_j$ , alors :

$$\forall s \in \Omega, \forall j, \mu_j(s) \in [0, 1]$$

$$\forall j, \sum_{s \in \Omega} \mu_j(s) \in ]0, N[$$

Les degrés d'appartenance varient entre 0 et 1, valant 1 pour les pixels du noyau et 0 pour ceux n'appartenant pas à la région floue. Le second axiome stipule qu'une région floue ne doit être ni vide, ni complète (égale à  $\Omega$ ). Les régions peuvent se chevaucher.

Cette définition s'inspire de la notion de partition floue de Ruspini [37], mais sans le troisième axiome qui impose que la somme des degrés d'appartenance de chaque pixel à toutes les régions soit égale à 1. Cet axiome n'est pas pertinent en segmentation, car il peut engendrer de fortes disparités entre pixels voisins, lorsque l'un appartient à une région différente de l'autre, et il augmente (jusqu'à atteindre la valeur 1) les degrés d'appartenance de pixels éloignés de tout noyau, s'ils n'appartiennent qu'à une seule région.

### 3.5 Conclusion

Ce chapitre a fourni une vue d'ensemble approfondie de la logique floue, en introduisant ses concepts fondamentaux et ses applications pratiques. Nous avons abordé les notions essentielles telles que les ensembles flous, leurs caractéristiques, et les fonctions d'appartenance. Ces éléments constituent la base théorique sur laquelle reposent les méthodes floues, permettant une modélisation plus flexible et nuancée des systèmes complexes.

Nous avons également exploré diverses fonctions d'appartenance, telles que trapézoïdale, exponentielle et gaussienne, illustrant leur rôle crucial dans la définition des ensembles flous. L'application de la logique floue à la segmentation d'images a été discutée, démontrant comment ces concepts peuvent être utilisés pour améliorer la précision et la robustesse des systèmes de traitement d'images. En somme, ce chapitre établit les fondations nécessaires pour une compréhension approfondie et une application efficace de la logique floue dans des contextes variés.

# Chapitre 4

## Interprétation et résultats

### 4.1 Introduction

Les détecteurs de contours et de régions sont apparus en réponse au besoin de minimiser la quantité d'informations présente dans l'image à traiter. Dans la vision de bas niveau, la détection de primitives est d'une difficulté particulièrement sous-estimée. Jusqu'aujourd'hui, il n'y a pas de méthode polyvalente pouvant être appliquée à tout type d'images quelles que soient les conditions de prise de vue. Tous les algorithmes de segmentation proposés sont destinés à un certain type d'images, dans des milieux bien déterminés. Les difficultés de traitement sont liées à l'environnement, à l'éclairage, au contraste, aux transitions faibles, à la texture, aux formes géométriques des régions etc... Ces différents facteurs ont fait que le problème de segmentation reste un sujet de recherche qui fascine de plus en plus tout chercheur s'intéressant de près ou de loin à ce bas niveau de vision.

Ce chapitre présente les algorithmes ainsi que les résultats obtenus à partir de la mise en œuvre du modèle U-Net pour la segmentation d'images. Nous détaillerons les différentes étapes de la réalisation du projet, ainsi que l'interprétation des résultats obtenus.

### 4.2 Segmentation par la logique floue

Dans l'algorithme proposé, la segmentation se fera par ajustement des régions en appliquant l'appartenance floue d'un pixel aux différentes régions.

#### 4.2.1 Algorithme de segmentation d'image par la logique floue

L'algorithme principal de la segmentation par la logique floue se présente comme suit :

1. Charger l'image médicale en niveaux de gris,
2. Définir les seuils de logique floue,
3. Transformer l'image en une matrice floue (2D) :
  - Appliquer une fonction d'appartenance gaussienne sur l'image avec `fuzz.membership.gaussmf` (moyenne 0.5, écart type 0.15).
4. Définir les fonctions d'appartenances triangulaires,
5. Remodeler les résultats selon les dimensions de l'image d'origine,
6. Créer les régions en utilisant la logique floue :
  - Initialiser une matrice région avec des zéros de la même taille que `image-fuzzy`.
  - Assigner les valeurs suivantes en fonction des seuils :
    - 0 pour les pixels où `image-fuzzy`  $\leq$  `low-threshold`.
    - 0.5 pour les pixels où `low-threshold`  $<$  `image-fuzzy`  $<$  `high-threshold`.
    - 1 pour les pixels où `image-fuzzy`  $\geq$  `high-threshold`.



Pour ce qui est de l'outil informatique, nous avons utilisé le logiciel **PyCharm**, qu'on peut interfacer avec le langage **Python** pour une éventuelle réalisation. L'image d'entrée X est une image IRM cérébrale de dimension 228x221 en niveaux de gris.

#### 4.2.1.1 Histogramme

La détermination du nombre de régions est effectuée à l'aide de l'histogramme de l'image, qui représente par définition la fréquence des différents niveaux de gris dans cette image.

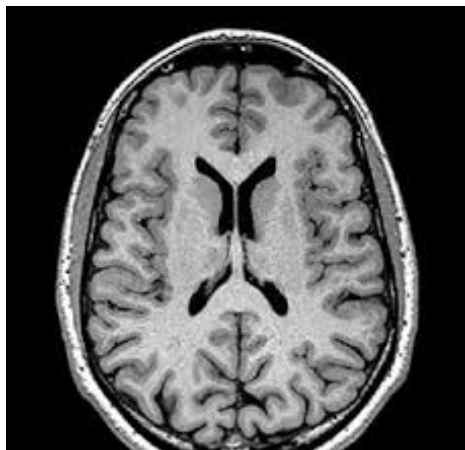
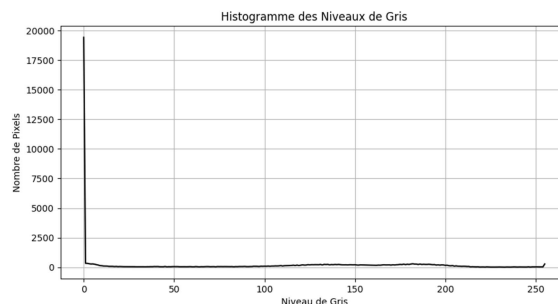


Image IRM cérébrale



l'histogramme d'image

FIGURE 4.1 – Image et son histogramme correspondant.

## 4.2.2 Résultats

### 4.2.2.1 Rapidité et Efficacité de la Segmentation par Logique Floue

La segmentation basée sur la logique floue s'est avérée très efficace dans notre projet, produisant des résultats remarquablement bons tout en nécessitant très peu de temps de traitement. Cette méthode est rapide, car elle ne requiert pas de processus d'entraînement prolongé. En quelques secondes, la segmentation est réalisée. Grâce à cette rapidité, la logique floue se distingue comme une méthode idéale pour des applications nécessitant des réponses immédiates, tout en maintenant une performance solide dans notre cas d'usage.

### 4.2.2.2 Prétraitement de l'image

L'image IRM cérébrale originale a été prétraitée à l'aide d'un filtre gaussien pour lisser les détails et réduire le bruit. La Figure 4.2 montre l'image originale et l'image après filtrage. L'application du filtre a permis de supprimer les artefacts tout en préservant les structures importantes (voir annexe).

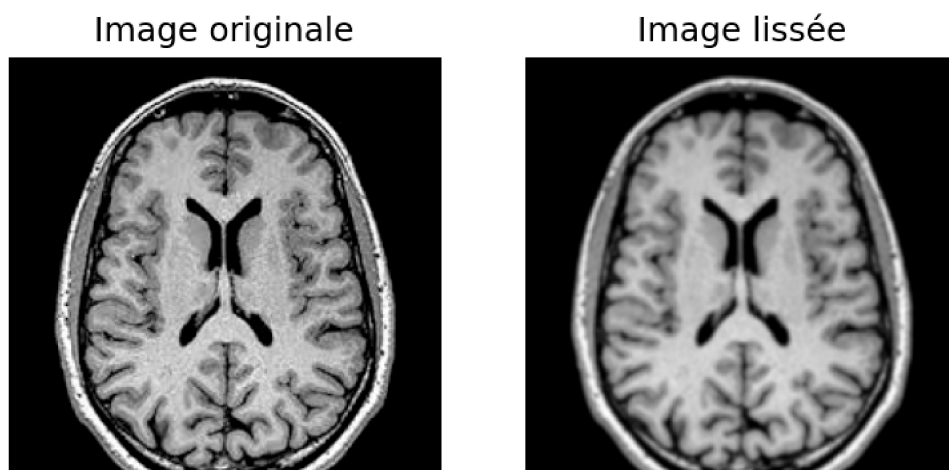


FIGURE 4.2 – Application d’un filtre gaussien pour lisser l’image.

#### 4.2.2.3 Application des Fonctions d’Appartenance Floues

La transformation en une matrice floue a été réalisée en appliquant une fonction d’appartenance gaussienne. La Figure(4.3) illustre l’image après l’application de cette fonction, montrant comment les valeurs de l’image filtrée sont converties en une matrice floue. Les fonctions d’appartenance triangulaires ont ensuite été définies pour segmenter les régions basse et haute.



FIGURE 4.3 – Image floue.

#### 4.2.2.4 Segmentation des Régions

Les régions détectées sont illustrées en Figure (4.4). Grâce à l’application de la segmentation floue en trois niveaux distincts — **basse, moyenne et haute intensité** —, il a été possible de distinguer avec précision les différentes zones de l’image IRM cérébrale.

La région de basse intensité correspond généralement aux tissus moins denses, tels que le liquide céphalorachidien ou les zones sans signal important. La région de moyenne intensité capture les structures cérébrales standard comme la matière grise, tandis que la région de haute intensité met en évidence les zones où la densité est plus élevée, telles que la matière blanche ou des anomalies potentielles, telles que les tumeurs ou les lésions.

Cette approche de segmentation multi-niveaux fournit une vue détaillée et structurée des différentes parties du cerveau, facilitant ainsi l'interprétation médicale et la détection précoce de pathologies. Ce processus de segmentation permet non seulement une meilleure lisibilité de l'image, mais aussi une aide à la décision clinique plus précise.

### Régions détectées



FIGURE 4.4 – Régions détectées.

## 4.2.3 Interprétation

### 4.2.3.1 Analyse des Résultats

La segmentation par logique floue a montré des résultats prometteurs en identifiant les régions d'intérêt dans l'image IRM cérébrale. La qualité de la segmentation est globalement bonne. L'impact des différents paramètres sur les résultats a été examiné, révélant que les seuils jouent un rôle crucial dans la définition des régions comme le montre la figure 4.4.

### Régions Détectées



FIGURE 4.5 – Exemple avec modification des seuils.

### 4.2.3.2 Limitations

Malgré les résultats positifs, certaines limitations ont été observées. Les variations dans les seuils peuvent conduire à des résultats incohérents, et la méthode peut être sensible aux

variations de l'image initiale. Pour améliorer les résultats, des ajustements dans les paramètres ou l'intégration d'autres techniques de prétraitement pourraient être envisagés.

## 4.3 Segmentation par CNN

U-Net est un réseau neuronal entièrement convolutif qui a été développé par Olaf Ronneberger au département d'informatique de l'Université de Fribourg, en Allemagne. Il a été spécialement développé pour la segmentation d'images biomédicales.

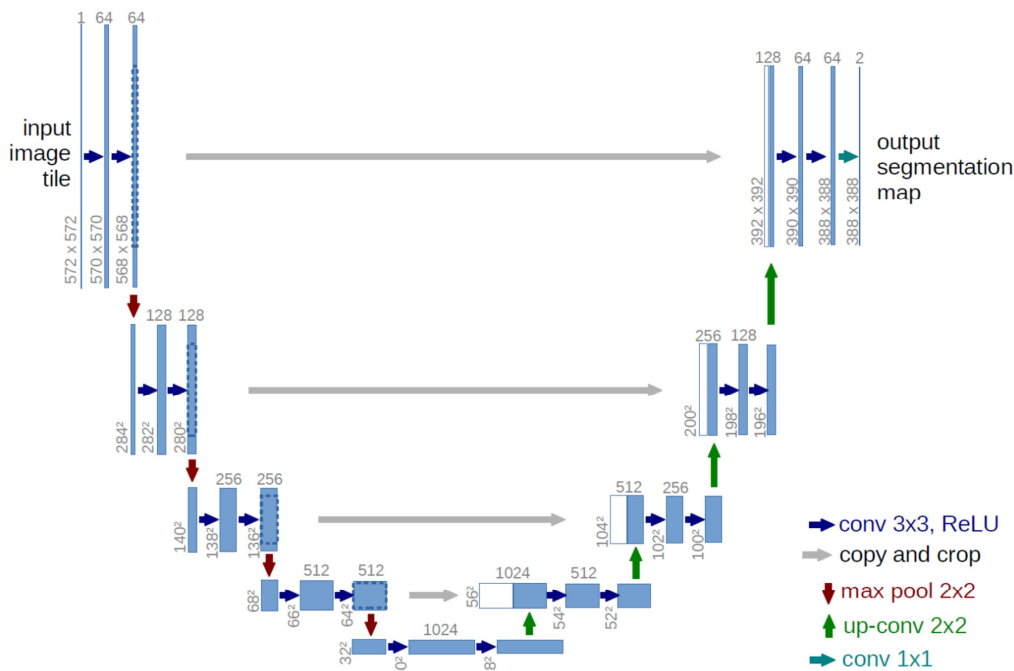


FIGURE 4.6 – Architecture originale d'U-Net.

Le modèle U-Net utilisé dans notre projet se compose d'une entrée de dimensions 256 x 256 x 3 (196,608 neurones), avec un encodeur de quatre blocs de convolution (16, 32, 48, et 64 filtres), chacun suivi de MaxPooling, réduisant progressivement le nombre de neurones dans les couches intermédiaires (128 x 128, 64 x 64, 32 x 32, et 16 x 16). Un pont central utilise deux couches de convolution avec 64 filtres (16 x 16 x 64 neurones). Le décodeur, symétrique à l'encodeur, applique l'UpSampling et la concaténation avec les couches de saut, augmentant les dimensions des neurones à 32 x 32, 64 x 64, 128 x 128, et 256 x 256, avec des filtres de 64, 48, 32, et 16. En tout, le modèle comprend 18 couches de convolution, 4 couches de MaxPooling, 4 couches d'UpSampling, et une couche de sortie avec 256 x 256 x 1 (65,536 neurones).

### 4.3.1 Structure du projet

Le projet comporte quatre dossiers :

1. **CVC-612/** : Il s'agit de l'ensemble de données que nous allons utiliser pour ce projet. Il contient deux sous-dossiers : images et masques. Comme leur nom l'indique, ces sous-dossiers contiennent les images et les masques.

2. **files/** : Ce dossier est utilisé pour stocker le fichier CSV qui contient toutes les informations pendant l'entraînement du modèle. Il stocke également le fichier de poids du modèle.

3. **logs/** : il contient les fichiers log du TensorBoard.

4. **results/** : Il est utilisé pour stocker les résultats après que nous ayons fait des prédictions sur l'ensemble de données de test.

Le projet dispose également de quatre scripts python :

1. **data.py** :Ce fichier contient le code pour le chargement du jeu de données, la lecture des images et des masques. Il est également utilisé pour créer un pipeline `tf.data` pour l'entraînement, la validation et le test d'un ensemble de données.

2. **model.py** :Ce fichier contient le code de l'architecture UNet qui consiste à segmenter les images des polypes.

3. **train.py** :Ce fichier permet au modèle de s'entraîner sur le jeu de données d'entraînement. Il est également utilisé pour enregistrer le modèle qui servira ensuite à faire des prédictions sur l'ensemble de données de test.

4. **predict.py** :Une fois l'entraînement terminé, ce fichier est utilisé pour effectuer des prédictions sur le jeu de données de test.

### 4.3.2 Comprendre l'ensemble de données

L'ensemble de données contient l'image extraite des vidéos de coloscopie. Ces images contiennent différents types de polypes. Les images du dossier 'Images' sont la propriété de Hospital Clinic, Barcelone, Espagne, les images du dossier 'Masks' sont une propriété du Computer Vision Center, Barcelone, Espagne, et les données sont divisées en train, test et val pour former et évaluer le modèle CNN développé. Le tableau 4.1 donne les détails de l'ensemble d'entraînement, de validation et de test.

Données	Pourcentage	Échantillons
Train	80%	490
Test	10%	61
Val	10%	61

TABLE 4.1 – Détails du Train, Val et des données d'essai

### 4.3.3 Implementation

Pour cela, on a utilisé TensorFlow 2.0 avec Python3.10

#### 4.3.3.1 DATA.PY

Ce fichier aide à gérer l'ensemble de données utilisées pour la segmentation, en le divisant en ensembles d'entraînement, de validation et de test. Il lit les images et les masques puis, crée un pipeline `tf.data` pour le traitement. Après l'importation des bibliothèques comme NumPy et TensorFlow, la fonction `load-data` charge les données CVC-ClinicDB et les divise selon un ratio 80:10:10 à l'aide de `train-test-split`. Les images et masques sont triés et redimensionnés en 256x256 pixels, puis normalisés. Les fonctions `read-image` et `read-mask` traitent respectivement les images RGB et les masques en niveaux de gris, tandis que `tf-parse` et `tf-dataset` créent le pipeline pour l'entraînement.

#### 4.3.3.2 MODEL.PY

Le fichier `model.py` contient l'architecture UNet destinée à l'entraînement sur le jeu de données de segmentation des polypes. Après avoir importé les bibliothèques nécessaires, le bloc de convolution (`conv-block`), composé de deux convolutions 3x3 avec normalisation de lot et activation ReLU, constitue le cœur du modèle. La fonction `build-model` construit l'architecture

complète avec une entrée de 256x256 pixels et un nombre de filtres [16, 32, 48, 64]. Une liste skip-x stocke les cartes de connexions de saut de l’encodeur, qui est construit en utilisant la fonction conv-block. Un pont relie l’encodeur au décodeur, dont l’objectif est de générer la carte de segmentation. Le décodeur applique des couches d’upsampling et concatène les sorties avec les connexions de saut. Enfin, une convolution 1x1 avec activation sigmoïde produit le masque binaire final.

### 4.3.3.3 TRAIN.PY

Toutes les fonctions nécessaires sont importées à partir des bibliothèques requises. La fonction iou calcule l’intersection sur l’union (IOU) entre les vérités terrain (y-true) et les prédictions (y-pred), renvoyant une valeur entre 0 et 1. Les hyperparamètres sont utilisés pour entraîner l’architecture UNet, tandis que les pipelines des jeux de données d’entraînement et de validation sont créés avec la fonction tf-dataset, qui prend les chemins d’images et de masques sous forme de liste (voir annexe).

L’architecture UNet est définie et construite à l’aide de la fonction build-model, optimisée à l’aide de l’optimiseur Adam avec un taux d’apprentissage spécifié. Les performances de l’architecture sont évaluées à l’aide de quatre indicateurs : l’accuracy, le recall, la precision et l’IOU.

Les rappels suivants sont utilisés :

**ModelCheckpoint** : Enregistre le fichier de poids du modèle après chaque époque.

**ReduceLROnPlateau** : Réduire le taux d’apprentissage lorsqu’une métrique a cessé de s’améliorer.

**CSVLogger** : Enregistre toutes les données d’entraînement dans un fichier CSV.

**TensorBoard** : Aide à la visualisation des données.

Nous avons entraîné le modèle sur 34 époques et la valeur finale de l’époque dans le tableau ci-dessous.

Métriques	Valeurs
Loss	0.0892
Accuracy	0.9796
Recall	0.8597
Precision	0.9337
IoU	0.4369
Validation loss	0.1712
Validation accuracy	0.9432
Validation Recall	0.5986
Validation precision	0.7197
Validation IoU	0.2863

TABLE 4.2 – Les valeurs d’époque finale.

Les résultats obtenus après l’entraînement du modèle de segmentation d’images montrent une performance globale solide sur les données d’entraînement, avec une précision élevée (97.96 %) et un faible taux de perte (0.0892), ce qui indique que le modèle a bien appris à segmenter les

images. Le rappel de 85.97 % et la précision de 93.37 % témoignent de sa capacité à identifier correctement les objets pertinents tout en minimisant les fausses détections. Cependant, les résultats de validation révèlent une légère diminution de la performance, avec une perte de validation plus élevée (0.1712) et une IoU plus basse (0.2863). Bien que le modèle conserve une bonne exactitude de validation (94.32 %), le rappel et la précision de validation inférieurs (59.86 % et 71.97 %, respectivement) indiquent qu'il pourrait bénéficier de raffinements pour améliorer sa capacité à généraliser sur des données non vues et à mieux capturer les détails complexes des images de validation.

#### 4.3.3.4 PREDICT.PY

Le fichier `predict.py` est utilisé pour évaluer le jeu de données de test et également pour enregistrer le masque prédit.

Les bibliothèques et fichiers requis sont importés. Les fonctions `read-image` et `read-mask` sont utilisées pour lire respectivement l'image et le masque, avec la fonction `read-image` normalisant l'image pour l'introduire dans l'architecture UNet pour la prédiction. La fonction `mask-parse` combine l'image d'entrée, le masque de vérité terrain et le masque prédit pour former une seule image. Un exemple de résultats de tests est donné ci-dessous.

### 4.3.4 Résultats

#### 4.3.4.1 Temps d'Entraînement et de Prédiction des CNN

Dans notre expérience, l'entraînement du modèle CNN a duré environ 8 heures, un délai prolongé justifié par la nécessité pour le modèle de capturer des caractéristiques complexes à partir d'un large volume de données. Ce long temps d'entraînement permet au modèle d'affiner sa compréhension des structures d'image. Même lors de la phase de prédiction, le processus est relativement long, car le modèle génère les résultats sous forme de 61 images de test. Toutefois, malgré le temps de calcul important, cette approche permet d'obtenir des résultats précis, notamment en capturant les détails complexes des images et en améliorant la détection des anomalies, ce qui se traduit par une meilleure performance globale du modèle.

#### 4.3.4.2 Graphiques d'Évaluation

Les figures suivantes illustrent les performances de notre modèle à différentes époques au cours du processus d'entraînement. Le premier graphique représente l'évolution de la précision, où la précision de l'entraînement et la précision de la validation sont montrées pour observer comment le modèle se généralise à des données invisibles. Le deuxième graphique illustre la perte, mettant en évidence la diminution de la perte d'apprentissage et de validation, ce qui indique la minimisation de l'erreur au fil du temps. Enfin, l'intersection sur l'union (IoU), qui est une mesure clé pour les tâches de segmentation d'images, est tracée pour évaluer le chevauchement entre les segments prévus et réels. Ces visualisations permettent une évaluation complète des performances du modèle pendant l'entraînement.

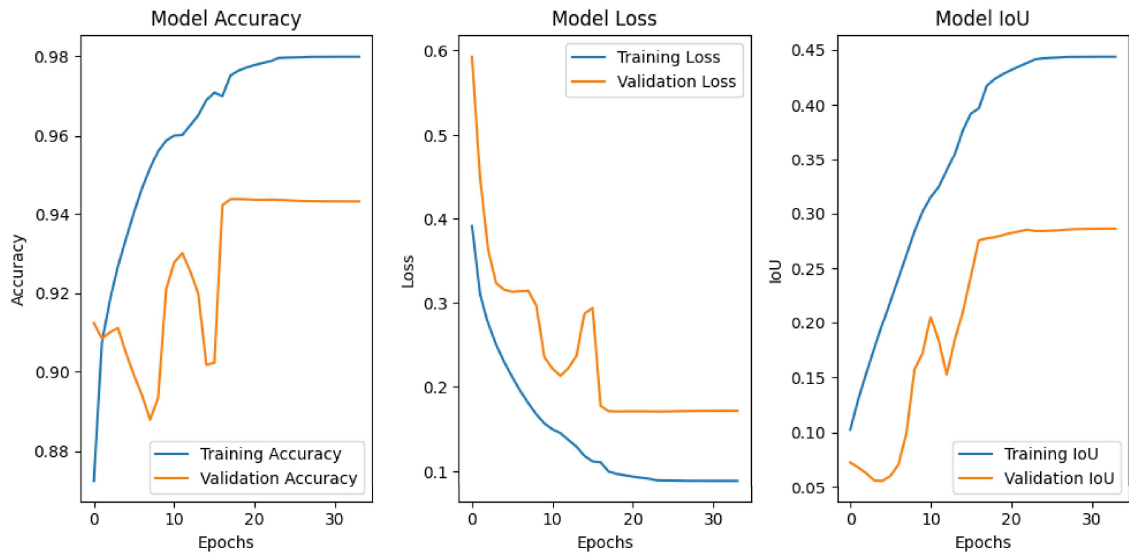
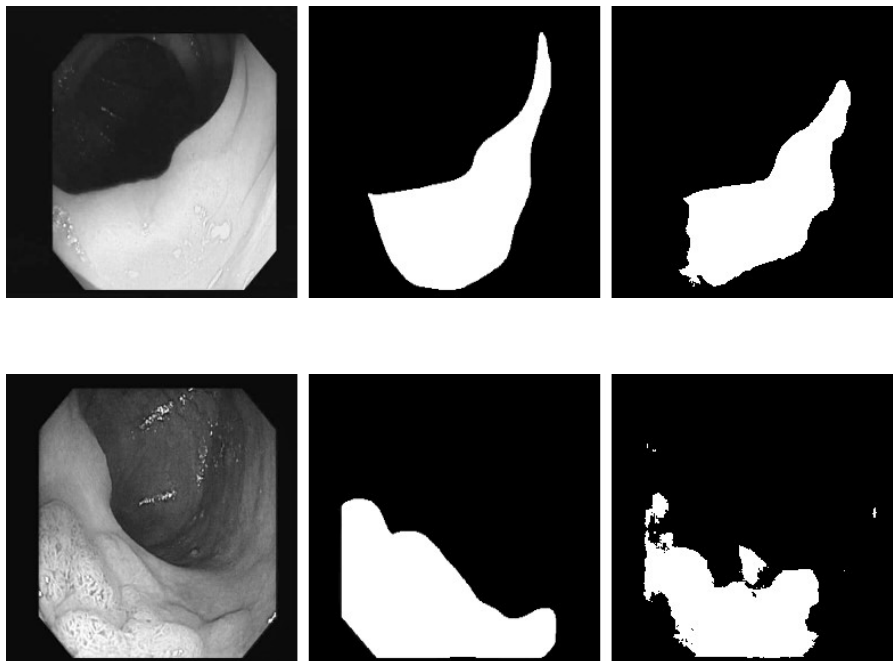


FIGURE 4.7 – Métriques de Performance du Modèle au Cours de l’Entraînement et de la Validation

### 4.3.5 Images Prédites par le Modèle

Les images segmentées par le modèle montrent que le modèle U-Net est capable de détecter et de segmenter les polypes avec une précision élevée. Les zones segmentées correspondent bien aux polypes visibles dans les images originales, ce qui confirme l’efficacité du modèle.

Toutefois, il est essentiel d’examiner les cas où le modèle peut échouer, comme les petites régions ou les polypes proches des bords. Des améliorations peuvent être envisagées, telles que l’augmentation des données, le réglage des hyperparamètres ou l’utilisation de techniques de régularisation supplémentaires.





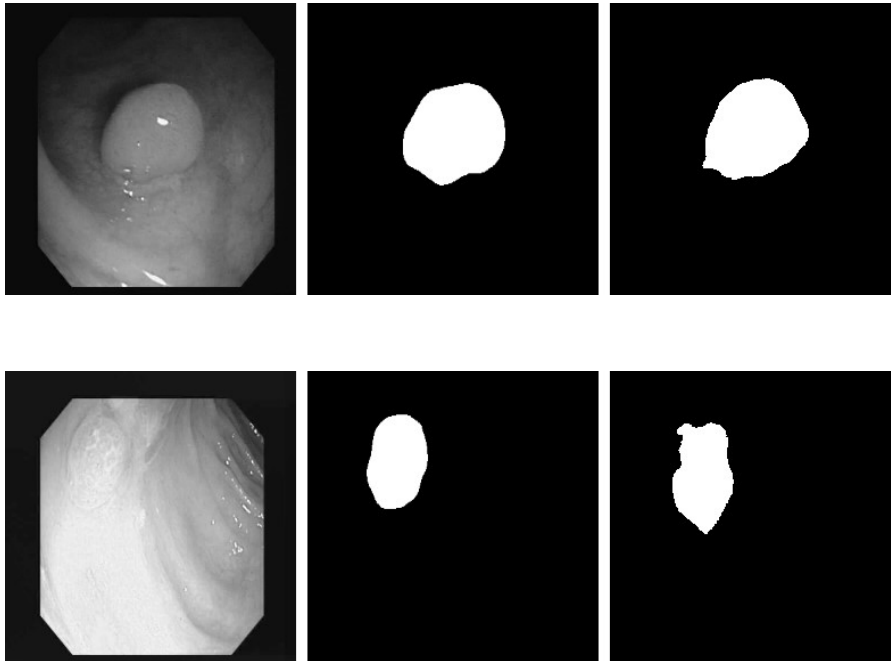


FIGURE 4.8 – Image d’entrée, Vérité terrain, Masque prédit. Une partie de l’exemple des résultats des tests.

## 4.4 Conclusion

Cette partie s’est attachée à décrire les algorithmes de segmentation par intelligence artificielle à travers la logique floue et les réseaux de neurones formels. Grâce à leur caractère d’auto-ajustement, les CNN ont pu détecter les régions, contenues dans l’image à traiter. L’idée de donner un aspect flou à l’appartenance des pixels aux régions s’est avérée très fructueuse. Car en effet, un pixel peut appartenir à différentes régions en même temps mais de différentes manières. Cette différence est justement quantifiée par les fonctions d’appartenances de la logique floue. Les résultats obtenus montrent que l’on peut utiliser ces deux technologies, dans le cas de la vision de bas niveau, en vue de se rapprocher le plus possible du caractère imprécis du système de vision humain.

# Conclusion Générale

Le travail présenté se situe dans le cadre de la vision robotique par intelligence artificielle et plus précisément le bas niveau de vision à savoir la segmentation d'images par l'approche région. Il a semblé particulièrement intéressant de simuler le fonctionnement du cerveau humain sur les robots à travers la logique floue et les réseaux de neurones. L'aspect flou de la vision humaine par l'appartenance des pixels aux différentes régions se fait grâce aux inférences floues. Cette appartenance diffère d'une région à l'autre en fonction du niveau de gris du pixel à traiter. Les réseaux de neurones, quand à eux, se rapproche du raisonnement humain grâce à leur caractère d'auto-ajustement. Les résultats obtenus par ces deux technologies sont très encourageants, surtout dans un domaine où la précision est vital, qui est le domaine médical. Cependant, les algorithmes développés au cours de ce modeste travail n'ont pas la prétention d'être fonctionnels pour tout types d'images. L'avenir de la vision robotique est vraisemblablement dans la coopération entre ces deux technologies. Nous pensons que cette idée pourra être étendue à une plus grande diversité de types d'images et domaines d'application. Ainsi, le problème de bas niveau de vision devient un problème d'enchaînement entre ces deux techniques. La vision robotique reste plus que jamais un domaine d'actualité. Il nous donne régulièrement une bonne leçon d'humilité en nous montrant que la technologie actuelle, alliée au traitement d'image, n'est pas encore sur le point d'inquiéter la capacité de l'oeil humain. Mais pour combien de temps encore ?

# Annexes

**Le filtre gaussien** : est un filtre linéaire utilisé pour lisser une image et réduire le bruit, en appliquant une fonction de distribution gaussienne pour pondérer les pixels voisins d'un pixel donné. Il est défini par la fonction gaussienne en deux dimensions, où chaque pixel est remplacé par une moyenne pondérée de ses voisins selon une distribution gaussienne centrée sur le pixel à traiter. Ce filtre est isotrope, ce qui signifie qu'il applique le même lissage dans toutes les directions de l'image. Le paramètre clé du filtre est l'écart type  $\sigma$ , qui contrôle la largeur du lissage : plus  $\sigma$  est grand, plus l'image est lissée[20].

**L'IoU (Intersection over Union)** : L'IoU est défini comme le ratio entre l'intersection et l'union des zones prédite et réelle, selon la formule suivante :

$$\text{IoU} = \frac{\text{Aire de l'intersection}}{\text{Aire de l'union}} \quad (4.1)$$

- Intersection : La zone commune entre la prédiction du modèle et le masque réel.

- Union : La somme des zones de la prédiction et du masque réel, en tenant compte des parties non chevauchantes.

L'IoU varie entre 0 et 1:

-Un IoU de 1 signifie une correspondance parfaite entre la prédiction et le masque réel (intersection complète).

-Un IoU de 0 signifie qu'il n'y a aucun chevauchement entre la prédiction et la vérité terrain.

**Loss (Perte)** : La perte est une mesure de l'erreur globale du modèle sur les données d'entraînement. Une perte plus faible indique que le modèle fait moins d'erreurs dans ses prédictions.

**Accuracy (Exactitude)** : L'exactitude mesure la proportion de prédictions correctes par rapport au nombre total de prédictions.

**Recall (Rappel)** : Le rappel, ou sensibilité, mesure la proportion des véritables positifs correctement identifiés par le modèle.

**Precision (Précision)** : La précision mesure la proportion des prédictions positives correctes par rapport à l'ensemble des prédictions positives faites par le modèle.

**Validation Loss (Perte de validation)** : La perte de validation est la mesure de l'erreur du modèle sur les données de validation (données non utilisées pour l'entraînement).

**Validation Accuracy (Exactitude de validation)** : L'exactitude de validation mesure la proportion de prédictions correctes sur les données de validation.

**Validation Recall (Rappel de validation)** : Le rappel de validation mesure la proportion des véritables positifs correctement identifiés sur les données de validation.

**Validation Precision (Précision de validation)** : La précision de validation mesure la proportion des prédictions positives correctes sur les données de validation.

**Validation IoU (IoU de validation)** : L'IoU de validation mesure le degré de chevauchement entre les régions prédite et réelle sur les données de validation.

# Bibliographie

- [1] Nawal Akrouf and Lilia Chaibi. Bioseg : une plateforme évolutive pour le développement d'approches biomimétiques en segmentation d'images, 2009.
- [2] MIHAI BUDESCU ALEXANDRINA-ELENA PANDELEA and GABRIELA COVATARIU. Image processing using artificial neural networks, 2015.
- [3] A.MEZIANE. 89étude et comparaison des méthodes de segmentation d'image cérébrales. *Thèse de magister, Université Abou-Bakr Belkaid-Tlemcen*, 2011.
- [4] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jing Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. *arXiv preprint*, 2015.
- [5] Pierre Borne, Mohamed Benrejeb, and Haggège Joseph. *Les réseaux de neurones : présentation et applications*, volume 15. 2007.
- [6] H. Bühler. *Réglage Par Logique Floue*. Presses Polytechniques et Universitaires Romandes, 1994.
- [7] touzet C. *Les réseaux de neurones artificiels introduction au connexionnisme*. 1992.
- [8] Azeddine Chaiba. *Commande de la machine asynchrone à double alimentation par des techniques de l'intelligence artificielle*. PhD thesis, Université de Batna, 2010.
- [9] Djazia Chami. Une plate-forme orientée agent pour le data mining, 2010.
- [10] J.-P. Cocquerez and Masson S. Philipp. Analyse d'images : filtrage et segmentation. *Masson*, page 2, 1995.
- [11] J.P. Coquerez and S. Philipp. *Analyse d'images, filtrage et segmentation*. Masson, Paris, 1995.
- [12] Fekrache Dalila. Segmentation d'images par colonies de fourmis artificielles hybrides, 2012.
- [13] Jean Larguier des Bancel. La logique d'aristote et le principe du tiers exclu. *Revue de théologie et de philosophie*, 14, 1926.
- [14] Chao Dong, Chen Change Loy, Kai He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2), 2016.
- [15] D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer, 1996.
- [16] Fatima EDDAOUDI. Aide à la détection par les techniques de segmentation et de codage de textures des images. *faculté des sciences université Mohammed v – agdal*, 2012.
- [17] Hayet MEDROUK et Souad TOUCHERIFT. Segmentation par classification d'une partie de l'image, mémoire de fin d'étude. *ummto*, 2010.
- [18] Claude Gacogne. Introduction à la logique floue, s.d.
- [19] T. Gadi and R. Benslimane. Fuzzy hierarchical segmentation. *Traitement du Signal*, 17(1), 2000.

- [20] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 3rd edition, 2008.
- [21] IBM. Image segmentation.
- [22] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:4–37, 2000.
- [23] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. Data clustering : A review. *ACM Computing Surveys*, 31(3) :264–323, 1999.
- [24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012.
- [25] Mohamed Lakhdari. Segmentation d’images par contour actif en appliquant les algorithmes génétiques, 2008.
- [26] Jérémy Lecoœur and Christian Barillot. Segmentation d’images cérébrales : Etat de l’art. Technical Report 6306, IRISA, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France, 2008.
- [27] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998.
- [28] Kahina Lemikchi, Fatiha Ousmaal, and Aldjia Rahali. Segmentation markovienne des images multispectrales msg. Master’s thesis, Université Mouloud Mammeri de Tizi Ouzou, Faculté de Génie Électrique et Informatique, Département d’Électronique, 2007. Mémoire d’ingénieur, Proposé et dirigé par Sadia Alkama.
- [29] T.-W. Lin. Compressed quadtree representations for storing similar images. *Image and Vision Computing*, 15:833–843, 1997.
- [30] parizeau M. *Réseaux de neurones*. Université Laval, Québec, Canada, 2004.
- [31] M.Meliani. Segmentation d’image par coopération régions-contours. *Thèse de magister, Ecole nationale Supérieure en Informatique, Oued-SmarAlger, Ecole Doctorale STIC*, 2012.
- [32] F Mouzaoui. Commande d’un système photovoltaïque connecté à un réseau de distribution par les techniques de l’intelligence artificielle. Master’s thesis, Université Dr. Yahia Fares de Médéa, 2018.
- [33] Amir NAKIB. *Conception de méta-heuristiques d’optimisation pour la segmentation d’images : application à des images médicales*. PhD thesis, Université de Paris 12, VAL DE MERNE, 2007.
- [34] S. Philipp-Foliguet, M. B. Vieira, and A. de A. Araujo. Segmentation into fuzzy regions using topographic distance. In *14th SIBGRAPI*, pages 282–288, Florianopolis, Brazil, 2001.
- [35] Joseph Redmon, Santosh Divvala, and Ross Girshick. You only look once : Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.
- [36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net : Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2015.
- [37] E. H. Ruspini. A new approach to clustering. *Information and Control*, 15(1) :22–32, 1969.
- [38] André Salem. Segments répétés et analyse statistique des données textuelles. *Histoire et mesure*, pages 5–28, 1986.
- [39] Joseph Salmon. *Agrégation d’estimateurs et méthodes à patch pour le débruitage d’images numériques*. PhD thesis, Diderot-Paris VII, Paris, 2010.

- [40] Iman Sebari and Dong-Chen He. *Les approches de segmentation par coopération région-contour*. PhD thesis, Université de Sherbrooke, Québec, Canada, 2010.
- [41] Christian Szegedy and Liu. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, Boston, MA, USA, 2015.
- [42] P. D. Wasserman. *Neural Computing : Theory and Practice*. Van Nostrand Reinhold, 1991.
- [43] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015.
- [44] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3) :338–353, 1965.
- [45] Ourda ZIANE. Sur les nombres flous et ses opérations, 2018.

## Le résumé

La vision robotique est une technologie qui permet aux robots de percevoir et d'interpréter des informations visuelles à l'aide de caméras et de logiciels spécialisés. La segmentation d'image est cruciale pour la vision robotique. Son principe est d'extraire les primitives, constituant des informations relatives au contenu de l'image, nécessaires pour un traitement de plus haut niveau. Les primitives ou les indices visuelles sont des informations comme les contours des objets ou les régions homogènes. Ces informations sont obtenues par des calculs directs sur les pixels. Mais jusqu'à présent il n'y a pas de méthode polyvalente pouvant être appliquée à tout type d'images quelle que soient les conditions de prise de vue. Tous les algorithmes de segmentation proposés sont destinés à un certain type d'images, dans des conditions bien déterminées. Certains de ces algorithmes opèrent par détermination des points frontières ce qui donne une segmentation par frontière. D'autres algorithmes opèrent sur le groupement des pixels de la même région, se basant sur la notion de similarité, ce qui donne une segmentation par région. Au cours de ce travail, une méthode de segmentation par région est proposée utilisant l'intelligence artificielle (IA) se basant sur les CNN et la LF. Ces deux technologies simulent le fonctionnement du cerveau et donc le jugement des résultats se rapprochera le plus possible du raisonnement humain.

**Les mots clés :** Vision robotique, Segmentation d'image, Intelligence artificielle (IA), Réseaux de neurones convolutifs (CNN), Traitement d'image, Machine learning ———

## Abstract

Robotic vision is a technology that allows robots to perceive and interpret visual information using specialized cameras and software. Image segmentation is a crucial step in robotic vision. Segmentation is a low-level processing that consists of extracting the primitives, constituting information relating to the content of the image, necessary for a higher level of processing. Primitives or visual cues are information such as the contours of objects or homogeneous regions. This information is obtained by direct calculations on the pixels. But so far, there is no versatile method that can be applied to all types of images, regardless of the shooting conditions. All the segmentation algorithms offered are intended for a certain type of image, under well-defined conditions. Some of these algorithms operate by determining boundary points, which results in a boundary segmentation. Other algorithms operate on the grouping of pixels in the same region, based on the notion of similarity, which gives segmentation by region. During this work, a segmentation method by region is proposed using artificial intelligence (AI) based on neural networks and then on fuzzy logic. Both technologies simulate how the brain works and therefore the judgment of the results will be as close as possible to human reasoning.

**Keywords :** Robotic vision, Image segmentation, Artificial intelligence (AI), Convolutional neural networks (CNN), Image processing, Machine learning



## ملخص

الرؤية الروبوتية هي تقنية تسمح للروبوتات بإدراك وتفسير المعلومات المرئية باستخدام كاميرات وبرامج متخصصة. تجزئة الصورة هي خطوة حاسمة في الرؤية الروبوتية. التجزئة هي عملية معالجة منخفضة المستوى تتكون من استخراج الأوليات ، وتشكل المعلومات المتعلقة بمحتوى الصورة ، وهي ضرورية للمعالجة ذات المستوى الأعلى. الأوليات أو الإشارات البصرية هي معلومات مثل ملامح الكائنات أو المناطق المتجانسة. يتم الحصول على هذه المعلومات عن طريق الحسابات المباشرة على وحدات البكسل. ولكن حتى الآن ، لا توجد طريقة متعددة الاستخدامات يمكن تطبيقها على جميع أنواع الصور ، بغض النظر عن ظروف التصوير. جميع خوارزميات التجزئة المقدمة مخصصة لنوع معين من الصور ، في ظل ظروف محددة جيدا. تعمل بعض هذه الخوارزميات عن طريق تحديد نقاط الحدود ، مما يؤدي إلى تجزئة الحدود. تعمل الخوارزميات الأخرى على تجميع وحدات البكسل في نفس المنطقة ، بناء على مفهوم التشابه ، والذي يعطي التجزئة حسب المنطقة. خلال هذا العمل ، تم اقتراح طريقة تجزئة حسب المنطقة باستخدام الذكاء الاصطناعي استنادا إلى الشبكات العصبية ثم على المنطق الضبابي. تحاكي كلتا التقنيتين كيفية عمل الدماغ ، وبالتالي فإن الحكم على النتائج سيكون أقرب ما يمكن إلى التفكير البشري.

الكلمات المفتاحية : الرؤية الروبوتية، تقسيم الصور، الذكاء الاصطناعي ، الشبكات العصبية الالتفافية ، معالجة الصور، التعلم

الآلي