



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Mohamed Khider – BISKRA  
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie  
**Département d'informatique**

N° d'ordre : **SIOD 30 / M2 / 2022**

## Mémoire

Présenté pour obtenir le diplôme de master académique en

Parcours : **Système d'Information d'Optimisation et de Décision (SIOD)**

---

# A COMPARATIVE ANALYSIS OF SPECIAL FUZZY CLUSTERING ALGORITHM

---

Par : **GHECHAM ELKHENSA**

Soutenu le 28 Juin 2022, devant le jury composant de :

Bougetitich Amina	MAA	Président
Benchabane Moufida	MAA	Examineur
Chighoub Fouzia	MAA	Encadreur

# Thanks

First of all, we thank God for giving us courage and health accomplish this work.

This work would not have been possible without the help and encouragement Many people I am very thankful for.

First of all, I would like to thank my supervisor Mss Chigoub Fouzia, professor at Mohamed Khider University of Biskra, for its interest in our work, For the constructive and enriching discussions you brought to us, she was Present in times of doubt. Her advice was especially valuable to me during Our Scientific Lives, which has provided stimulating guidance for over six months.

We would also like to especially thank all our esteemed professors and all merit He returns to them as we are today, without forgetting to mention all friends and colleagues for the support they gave us.

I would also like to thank the members of my jury for agreeing to review this note.

We reserve our last thanks to all members of my small family, especially to the honorable husband for his full support during this path. Father, mother, my mother in law, brothers and sisters for Dedication, sacrifices and unwavering support.

# Dedication

I dedicate this modest work:

To my dear parents for the sacrifices and the encouragement that they have never ceased to confer on me. God keeps them forever happy.

Dear Husband , and my Kids Djoumana and Djoud , brothers and sisters .

To all my husband family, my father in law Tamami Ahmed lamine may God welcomes in his vast paradise.

To All my teachers.

To all my friends specially Ben Abdalla Fatima Ezzohra ,Maouche Nadia and to all who have contributed directly or indirectly far.

## General Introduction

The image is one of the most effective means of communicating with others, everyone can analyze the image in their own way, to get an impression and extract precise information.

Over the past few years, the field of image processing has grown enormously.

developed and a large number of works have been carried out in different fields applications such as image segmentation, remote sensing, etc.

In an image processing system, image segmentation is the most important operation.

more important because it determines the quality of the interpretation of an image. A good result of segmentation does not necessarily allow a good interpretation, but we cannot get a good interpretation from a bad segmentation result.

Image segmentation is defined as a division of the image into regions, which are net sets (as opposed to fuzzy sets), there is a unique choice for each site to belong or not to each region. But a pixel in "the center" of a homogeneous region, certainly belongs much more to this region than clustered pixels at the end of a growth of regions for example, and a pixel close to an outline is certainly not in the same region as a pixel placed on the other side of this contour. In order to model this knowledge, we propose to construct regions which are no longer net sets, but fuzzy sets, with for each site of the image a degree of belonging to each region. Clustering, or segmentation, is an operation that aims to separate different homogeneous areas of an image, in order to organize the objects into groups (clusters) whose members have various properties in common (intensity, color, texture, etc.). One can group the segmentation methods into two categories: unsupervised segmentation, which aims to automatically separate the image into natural clusters, i.e. without any prior knowledge of the classes; and supervised segmentation, which operates from the knowledge of each of the classes defined by a probabilistic approach.

Segmentation is considered an open problem, not yet solved. He is classified among the problems due to its complexity. Our thesis is to know how to introduce spatial information about the FCM (Fuzzy C-Means) algorithm in a new image calculation for a robust and efficient classification on a classified image, to obtain a better fuzzy partition and to become familiar with this after the comparison process the set of algorithms.

Our thesis has been structured an introduction, four chapters and a conclusion:

The first chapter was devoted to explain the Approaches Segmentation Image. In the second Chapter we presented the main definitions of the variant FCM algorithms.

In the third Chapter we reviewed the design of our work, in which we will talk about the components of our system and the relationships between them.

In The last chapter, all that is concrete in the application is to explain the interface with some results recorded after the comparison of the algorithms.

We end our work with a conclusion and bibliographical references.

## Chapitre I. Approaches Segmentation Image

### I.1.Introduction

Image segmentation is an essential step in any analysis process picture. It is a low-level processing that precedes the stage of measurement, understanding and decision. Its objective is to partition the image into connected and homogeneous regions.

In the sense of a homogeneity criterion that is difficult to define, especially in the case of textured regions.

We will discuss in this chapter the main methods of segmentation, accompanied by a descriptions and some illustrations.

### I.2.Definition of an image

The image can be described in the form of a function  $I(x,y)$  of analog brilliance continuous, defined in a bounded domain, where  $x$  and  $y$  are the coordinates of a point of the image and  $I(x,y)$  represents the observed information whether it is a function of light intensity (grayscale) or color. Under this aspect, the image is unusable by the machine, it must then be digitized.

### I.3.Digital picture

A digital image can be seen as a dot matrix in which each element is called pixel (picture element).

#### I.3.1 Characteristics of a digital image

The image is a structured set of information characterized by several parameters which we describe below.

#### I.3.2 pixels

The pixel is a unit of area that defines the basis of a digital image.

#### I.3.3 The dimension

As the representation of the image is a matrix whose elements are values

numerical, so multiplying the number of rows by the number of columns will give us the image size exp (1024 x 1024).

### I.3.4 The resolution

It determines the number of pixels per unit length of the structure to be digitized.

The more the number of pixels decreases, the more the resolution decreases, and the more the quality of the image digital degrades.

### I.3.5 Contrast

It is the marked opposition between two regions of an image, more precisely between the dark regions and light regions of this image. The contrast is defined according to the luminance of two image areas. If  $L_1$  and  $L_2$  are the degrees of brightness respectively of two neighboring areas  $A_1$  and  $A_2$  of an image, the contrast  $C$  is defined by:

$$c = \frac{L_1 - L_2}{L_1 + L_2} \quad (\text{I-1})$$

### I.3.6 Luminance

It is the degree of luminosity of the points of the image. It is also defined as being the quotient of the luminous intensity of a surface by the apparent area of this surface, for a distant observer, the word luminance is substituted for the word brilliance, which corresponds to the glow of an object.

Good luminance is characterized by:

- Bright (bright) images
- Good contrast: avoid images where the contrast range tends towards white or black, these images result in loss of detail in dark or bright areas.
- The absence of parasites.

The mean or luminance (brightness) of a grayscale digital image is defined as the average of the pixels in the image.

$$lum(I) = \frac{1}{m*n} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} \mathbf{1}(x, \lambda) \quad (\text{I-2})$$

With:  $m$ : number of columns.

$n$ : number of lines.

$I(x, y)$ : the gray level value in the point  $(x, y)$ .

### **I.3.7 Noise**

A noise (parasitic) in an image is considered as a phenomenon of abrupt variation of the intensity of a pixel compared to its neighbors, it comes from the lighting of the optical and electronic devices of the sensor.

### **I.3.8 The histogram**

The histogram is a tool that represents a simple summary of the content of an image.

There are three types of histograms:

- Unimodal histogram: formed of a single peak representing either the object or the background.
- Bimodal histogram: formed of two separate peaks highlighting the existence of a object of a fund.
- Multimodal histogram: made up of several separate modes (peaks separated by several valleys) highlighting several objects. [1]

## **I.4. Image types**

### **I.4.1 1.5.1. Binary image (black and white)**

A binary image is a rectangular matrix whose elements are 0 or 1, where the level '0' represents absolute black and level '1' represents absolute white. It's the type image obtained by scanning texts.

### **I.4.2 Grayscale (monochrome) image**

The gray level is the value of the light intensity at a point. The color of pixel can take values ranging from black to white through a finite number of levels intermediaries.

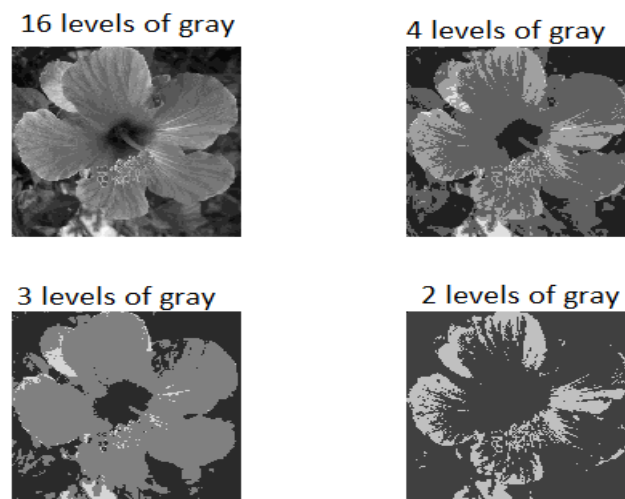
In general, grayscale images contain 256 shades of gray. By convention the value 0 represents absolute black and value 255 represents absolute white (See Figure I.1). In fact each integer representing a 1-byte coded gray level.





**Figure I.1:** Grayscale quantization [1]

The number of shades of gray or colors affects the quality of the image (See Figure I.2)



**Figure I.2:** Grayscale number variation [1]

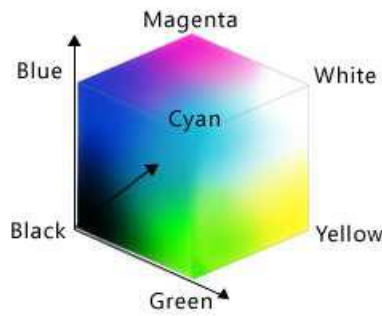
### I.4.3 Color image (full color)

The color space is based on additive color synthesis, that is, the mixture of three components (eg (r, g, b) red green blue, (c, m, y) cyan magenta yellow) gives a color (See Figure I.3).

Most color images are based on three primary colors (red, green and blue) and typically use 8 for each color component, so each pixel requires  $3 * 8 = 24$  bits to encode the three components, and each color component can take one of the values of the interval  $[0 ..255]$ . [2]

You can convert an RGB image to grayscale using several methods, the simplest is to do:

$$\text{gray} = (\text{blue} + \text{green} + \text{red}) / 3 \quad (\text{I-3})$$



**Figure I.3:** Red Green Blue (RGB) color space [2]

### I.5. Image segmentation definition

Segmentation is a low-level processing that consists of cutting an image into connected regions, each region will consist of pixels of the same class. But he can there may also be several regions in the image corresponding to the same class.

There is no single method of image segmentation, the choice of a technique is linked to several factors, such as:

- The nature of the image (optical, infrared, etc...)
- The acquisition conditions (lighting, presence of noise, etc...)
- The primitives to extract (outline, line segments, texture, etc...).

Because of this diversity, it is difficult to define, in an absolute way, a good segmentation. [3]

More precisely, we can define the segmentation as being a partition of an image  $I$  into  $n$  sets  $R_i$  called regions such that:

- 1)  $\bigcup_{i=1}^n C = I$
- 2)  $\forall i, j \in \{1, \dots, n\}^2$  et  $i \neq j \Rightarrow C_i \cap C_j = \emptyset$
- 3)  $\forall i \in \{1, \dots, n\}^2$   $C_i$  is related
- 4)  $\forall i, j \in \{1, \dots, n\}^2$   $P(C_i) = true$
- 5)  $\forall i, j \in \{1, \dots, n\}^2$   $C_i$  is adjacent to  $C_j$  and  $i \neq j \Rightarrow P(C_i \cup C_j) = false$

Where the predicate  $P$  is used to test homogeneity.

The first condition means that the image  $I$  is partitioned into  $n$  classes. And the second explains that all classes are pairwise disjoint.

The third step, the fourth and the fifth impose on each pixel of a class of satisfy the same property in the sense of the predicate  $P$ . The predicate  $P$  is no longer true for the joining of two adjacent classes.

Therefore, the segmentation allowing data analysis by grouping the pixels forming the image to be classified so that the pixels belonging to a class are more similar

between them than those of different classes, we assign to each entity of the treated scene a label indicating its membership in a particular class. The entity used characterizes usually a pixel or a set of pixels, while the tag is a theme chosen by the user.

There is no single method of segmenting an image, the choice of a technique is related:

- the nature of the image:

- uneven lighting, reflections,
- presence of noise, textured areas,
- fuzzy contours, partly obscured,

- operations located downstream of the segmentation:

- location, measurement, 3D calculation,
- pattern recognition, interpretation,
- diagnosis, quality control,

- the primitives to extract:

- contours, line segments, angles, ...
- regions, forms,
- textures,

- operating constraints:

- algorithmic complexity, real-time operation,

- size of memory available in the machine.[4]

## **I.6.Applications Domaine**

Some of the practical applications of image segmentation are:

### ➤ **Content-based image retrieval**

Image retrieval by content (in English: content-based image retrieval or CBIR) is a technique for searching images from their visual characteristics, that is to say induced from their pixels. Images are classically described as reflecting their texture, color, shape. A typical case of use is searching by example where you want to find visually similar images to an example given in the query.

The technique of image search by content is opposed to the search for images by keywords or tags, which was historically proposed by search engines such as Google Image thanks to image banks where the images are found using the accompanying text rather than the content of the image itself (but Google Image now offers filters based on the content (pixels) of the images).

### ➤ **Machine vision**

is the technology and methods used to provide imaging-based automatic inspection and analysis for such applications as automatic inspection, process control, and robot guidance, usually in industry. Machine vision refers to many technologies, software and hardware products, integrated systems, actions, methods and expertise. Machine vision as a system engineering discipline can be considered distinct from computer vision, a form of computer science. It attempts to integrate existing technologies in new ways and apply them to solve real world problems. The term is the prevalent one for these functions in industrial automation environments but is also used for these functions in other environment vehicle guidance.

The overall machine vision process includes planning the details of the requirements and project, and then creating a solution. During run-time, the process starts with imaging, followed by automated analysis of the image and extraction of the required information.

### ➤ **Medical imaging**

including volume rendered images from computed tomography and magnetic resonance imaging.

- Locate tumors and other pathologies.
- Measure tissue volumes.
- Diagnosis, study of anatomical structure.
- Surgery planning.
- Virtual surgery simulation.
- Intra-surgery navigation.

➤ **Object detection**

is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.

- Pedestrian detection.
- Face detection.
- Brake light detection.
- Locate objects in satellite images (roads, forests, crops, etc.).

➤ **Recognition Tasks**

-**Face recognition:** is a technology capable of matching a human face from a digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image.

-**Finger-print recognition:** Fingerprints can be captured as graphical ridge and valley patterns. Because of their uniqueness and permanence, fingerprints emerged as the most widely used biometric identifier in the 2000s. Automated fingerprint verification systems were developed to meet the needs of law enforcement and their use became more widespread in civilian applications. Despite being deployed more widely, reliable automated fingerprint verification remained a challenge and was extensively researched in the context of pattern recognition and image processing. The uniqueness of a fingerprint can be established by the overall pattern of ridges and valleys, or the logical ridge discontinuities known as minutiae. In

the 2000s minutiae features were considered the most discriminating and reliable feature of a fingerprint. Therefore, the recognition of minutiae features became the most common basis for automated fingerprint verification. The most widely used minutiae features used for automated fingerprint verification were the ridge ending and the ridge bifurcation

**-Iris recognition:** is an automated method of biometric identification that uses mathematical pattern-recognition techniques on video images of one or both of the irises of an individual's eyes, whose complex patterns are unique, stable, and can be seen from some distance. The discriminating powers of all biometric technologies depend on the amount of entropy they are able to encode and use in matching. Iris recognition is exceptional in this regard, enabling the avoidance of "collisions" (False Matches) even in cross-comparisons across massive populations. Its major limitation is that image acquisition from distances greater than a meter or two, or without cooperation, can be very difficult.

#### ➤ **Traffic control systems**

Traffic congestion has become a serious issue due to the growing number of vehicles in the world. Traffic light control system is widely used to control the flow of road junction. Currently, most of the traffic light system used pre-time and count down timers to control traffic flow. Due to the fixed-time setting, often the system unable to handle unexpected heavy traffic flows and cause traffic jam. Thus, there is a need of adaptive traffic signals which are able to do real time monitoring to control traffic light signal based on traffic density. Using image segmentation proposed an adaptive traffic light control system that uses image processing and image matching technique in controlling the traffic in an effective manner by taking images of each lane at a junction. The density of traffic in the images at each junction are compared. Results show that more time are allocated for the vehicles on the densest road to pass compared to other less dense road. Edge operation detector is used to detect the density of traffic at each lane.

#### ➤ **Video surveillance**

is the use of video cameras to transmit a signal to a specific place, on a limited set of monitors. It differs from broadcast television in that the signal is not openly transmitted, though it may employ point-to-point (P2P), point-to-multipoint (P2MP), or mesh wired or wireless links. Even though almost all video cameras fit this definition, the term is most often applied to those used for surveillance in areas that require additional security or ongoing monitoring.

➤ **Video object co-segmentation and action localization**

is a special case of image segmentation, which is defined as jointly segmenting semantically similar objects in multiple images or video frames.[5]

### **I.7. Image Segmentation Approaches**

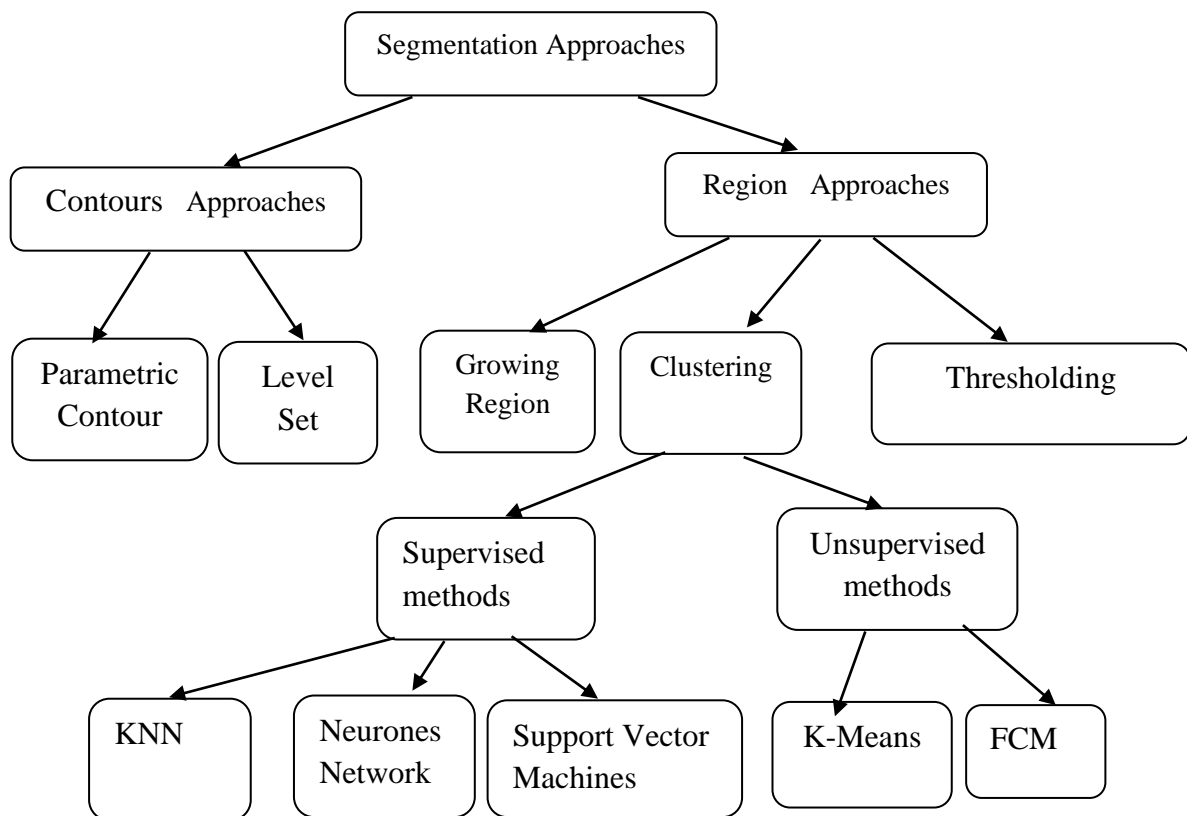
Essentially, the purpose of image analysis is to extract the characteristic information contained in an image. This information can take the shape, color, outline, texture, ... Therefore, it is necessary to proceed first to the segmentation of the lesion. This segmentation is generally approached according to two complementary models: the region, border approach and classification approach (See Figure I.4)

The frontier approach brings together the techniques associated with a variation in intensity: detection of outlines. However, the contours obtained are rarely connected, so it must face to the contour closure problem.

The region approach considers groupings of pixels having common properties, each pixel is assigned to a unique region after a partition of the image.

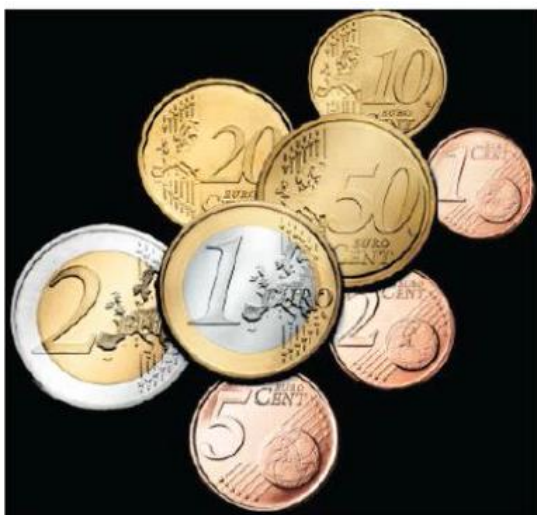
However, there is no specific algorithm to determine regions and boundaries in a picture. Moreover, the researchers have compromised that the ideal segmentation does not exist not, the good segmentation technique will therefore be the one that will make it possible to arrive at a good interpretation [6].

A range of techniques is proposed in the literature where each has advantages and disadvantages. Among the many image segmentation approaches that exist, we present some techniques most quoted in the literature applied to the image segmentation, namely the methods based on pixels, the region approach as well as those contours based.

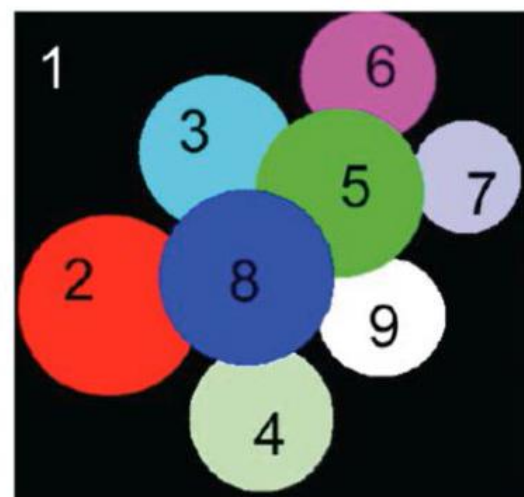


**Figure I.4:** Image Segmentation Approaches [6]

### I.7.1 Region based approaches



**Original image**



**image after the segmentation**

**Figure I.5:** Segmentation by region-based approaches [7]

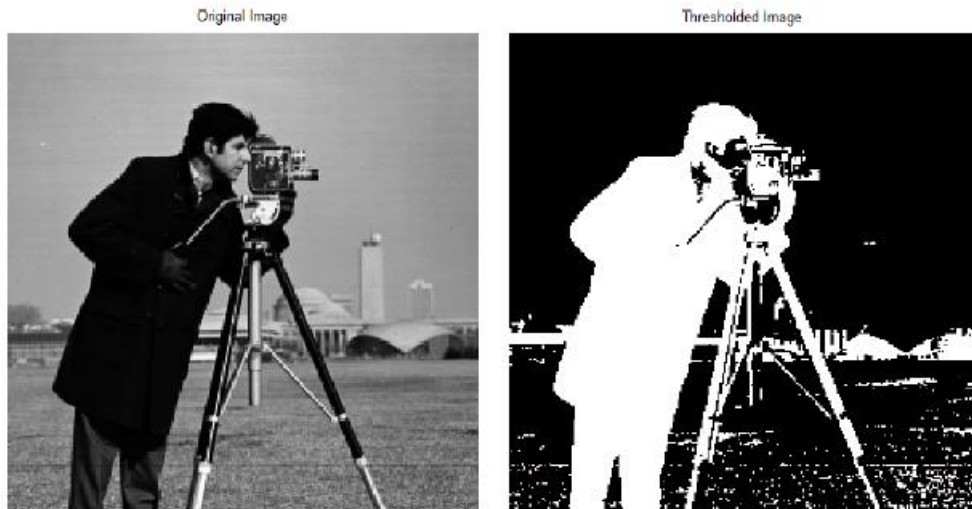
Image segmentation by the region approach consists of dividing the image into regions. Adjacent pixels are grouped into distinct regions according to a given homogeneity or similarity criterion. This criterion can be, for example, the level of gray, color, texture, etc.. (See Figure I.5)



A grouping process is repeated until all pixels in the image are included in regions. This approach aims, therefore, to segment the image based on the intrinsic properties of the regions. [7]

In these there are several methods like region growing, region splitting, region merging, classification which we present below

### I.7.1.1 Thresholding



**Figure I.6:** Segmentation by thresholding [8]

Thresholding is a method that is part of the pixel-based approach, although others references classify this method in the region approach to segmentation, because it highlights regions of an image (See Figure I.6: Segmentation by thresholding.

A threshold is a numerical value corresponding to an image parameter (example: the gray level), and the gray level histogram is used to deduce the threshold.

#### I.7.1.1.1 Thresholding Techniques

Threshold technique is one of the important techniques in image segmentation. This technique can be expressed as:

$$T = T[x, y, p(x, y), f(x, y)] \quad (I-4)$$

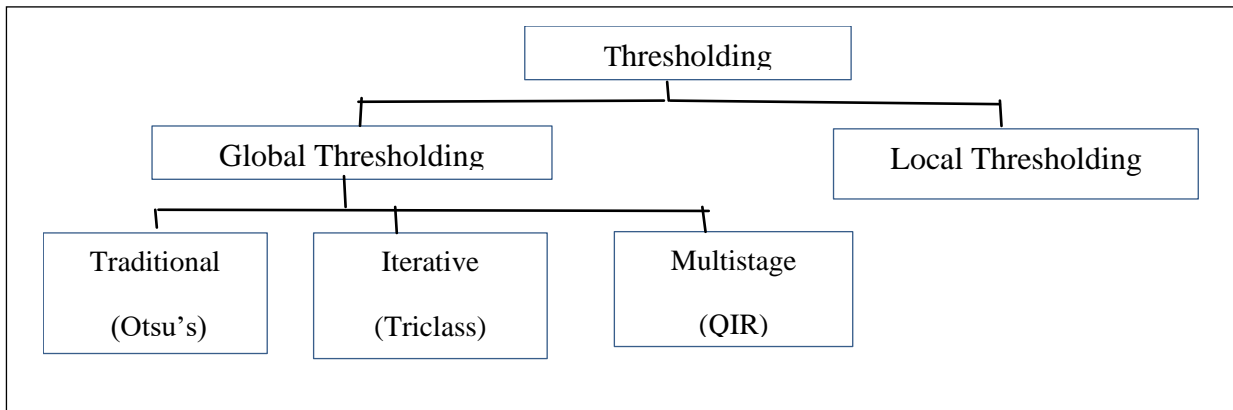
Where  $T$  is the threshold value.  $x, y$  are the coordinates of the threshold value point.

$p(x, y), f(x, y)$  are points the gray level image pixels .

Threshold image  $g(x, y)$  can be define:

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases} \quad (I-5)$$

Thresholding techniques are classified as bellow:



**Figure I.7:** Thresholding Techniques [8]

Thresholding is classified into two Global Thresholding and Local Thresholding, Global thresholding is divided into Traditional, Iterative, Multistage, (See Figure I.7)

### A. Global Thresholding

Global (single) thresholding method is used when there the intensity distribution between the objects of foreground and background are very distinct. When the differences between foreground and background objects are very distinct, a single value of threshold can simply be used to differentiate both objects apart. Thus, in this type of thresholding, the value of threshold  $T$  depends solely on the property of the pixel and the grey level value of the image. Some most common used global thresholding methods are Otsu method, entropy based thresholding, etc. Otsu's algorithm is a popular global thresholding technique. Moreover, there are many popular thresholding techniques such as Kittler and Illingworth, Kapur , Tsai , Huang , Yen and et al [8].

#### a.1 Traditional Thresholding

(Otsu's Method) In image processing, segmentation is often the first step to pre-process images to extract objects of interest for further analysis. Segmentation techniques can be generally categorized into two frameworks, edge-based and region-based approaches. As a segmentation technique, Otsu's method is widely used in pattern recognition, document binarization, and computer vision. In many cases Otsu's method is used as a pre-processing technique to segment an image for further processing such as feature analysis and quantification. Otsu's method searches for a threshold that minimizes the intra-class variances of the segmented image and can achieve good results when the histogram of the original image has two distinct peaks, one belongs to the background, and the other belongs to the

foreground or the signal. The Otsu's threshold is found by searching across the whole range of the pixel values of the image until the intra-class variances reach their minimum. As it is defined, the threshold determined by Otsu's method is more profoundly determined by the class that has the larger variance, be it the background or the foreground. As such, Otsu's method may create suboptimal results when the histogram of the image has more than two peaks or if one of the classes has a large variance .

### **a.2 Iterative Thresholding**

(A New Iterative Triclass Thresholding Technique) A new iterative method that is based on Otsu's method but differs from the standard application of the method in an important way. At the first iteration, we apply Otsu's method on an image to obtain the Otsu's threshold and the means of two classes separated by the threshold as the standard application does. Then, instead of classifying the image into two classes separated by the Otsu's threshold, our method separates the image into three classes based on the two class means derived. The three classes are defined as the foreground with pixel values are greater than the larger mean, the background with pixel values are less than the smaller mean, and more importantly, a third class we call the "to-be-determined" (TBD) region with pixel values fall between the two class means. Then at the next iteration, the method keeps the previous foreground and background regions unchanged and re-applies Otsu's method on the TBD region only to, again, separate it into three classes in the similar manner. When the iteration stops after meeting a preset criterion, the last TBD region is then separated into two classes, foreground and background, instead of three regions. The final foreground is the logical union of all the previously determined foreground regions and the final background is determined similarly. The new method is almost parameter free except for the stopping rule for the iterative process and has minimal added computational load.

### **a.3 Multistage Thresholding**

(Quadratic Ratio Technique for Handwritten Character) The QIR technique was found superior in thresholding handwriting images where the following tight requirements need to be met: 1. All the details of the handwriting are to be retained 2. The papers used may contain strong colored or patterned background 3. The handwriting may be written by a wide variety of writing media such as a fountain pen, ballpoint pen, or pencil. QIR is a global two stage thresholding technique. The first stage of the algorithm divides an image into three sub

images: foreground, background, and a fuzzy sub image where it is hard to determine whether a pixel actually belongs to the foreground or the background.

Two important parameters that separate the sub images are A, which separates the foreground and the fuzzy sub image, and C, which separate the fuzzy and the background sub image.

If a pixel's intensity is less than or equal to A, the pixel belongs to the foreground. If a pixel's intensity is greater than or equal to C, the pixel belongs to the background. If a pixel has an intensity value between A and C, it belongs to the fuzzy sub image and more information is needed from the image to decide whether it actually belongs to the foreground or the background.

### B. Local Thresholding

A threshold  $T(x,y)$  is a value such that:

$$b(x,y) = \begin{cases} 0 & \text{if } I(x,y) < T(x,y) \\ 1 & \text{otherwise} \end{cases} \quad (\text{I-6})$$

Where  $b(x,y)$  is the binarized image and  $I(x,y) \in [0,1]$  be the intensity of a pixel at location  $(x,y)$  of the image I. In local adaptive technique, a threshold is calculated for each pixel, based on some local statistics such as range, variance, or surface-fitting parameters of the neighborhood pixels. It can be approached in different ways such as background subtraction, water flow model, means and standard derivation of pixel values, and local image contrast. Some drawbacks of the local thresholding techniques are region size dependent, individual image characteristics, and time consuming. Therefore, some researchers use a hybrid approach that applies both global and local thresholding methods and some use morphological operators. Niblack, and Sauvola and Pietaksinen use the local variance technique while Bernsen uses midrange value within the local block.[9]

#### I.7.1.2 Growing region

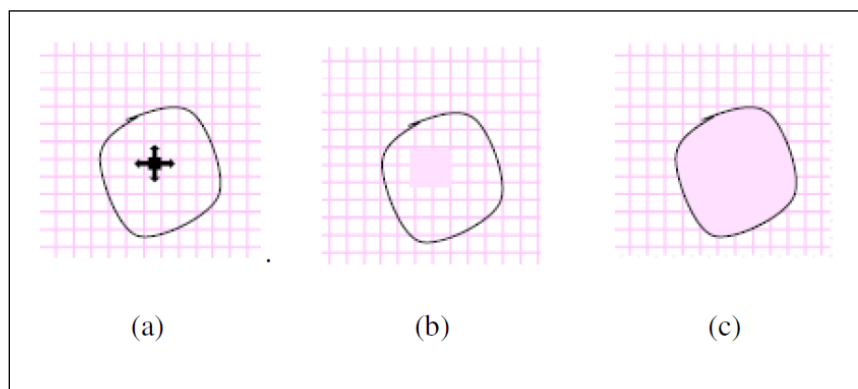
The concept of this method, is to start with a point that meets a detection criterion and to grow the point in all directions to extend the region.

the process starts from an arbitrary pixel. The pixel is labeled as a region that then grows based on a similarity measure. In this approach, a boundary pixel is joined to the current region provided it has the highest grey level among the neighbors of the region. This induces a directional growing such that the pixels of high grey level will be absorbed first. When all the high grey level pixels in the region are absorbed, the process continues by absorbing the boundary pixels with monotonically lower and lower grey levels.

When several pixels with the same grey level jointly become the candidates to join the region, the first-come first-served strategy is used to select one of them. This makes the region more compact, particularly in situations where the grey levels of the background or the region pixels are very homogeneous.

In order to monitor the pixels joining the region, a grey-level mapping is generated. The mapping is very similar to the mapping of data points from a high-dimensional feature space onto a sequence which is used in the mode separating (MODESP) procedure for cluster analysis proposed by Kittler [10].

The stages of the Algorithm can be outlined in the following way (See Figure I.8):



**Figure I.8:** The process of a region growing algorithm.[10]

(a) start of process, the black dot shows the initial point and the arrows the direction of growth.

(b) the growth of pixels after a few iterations.

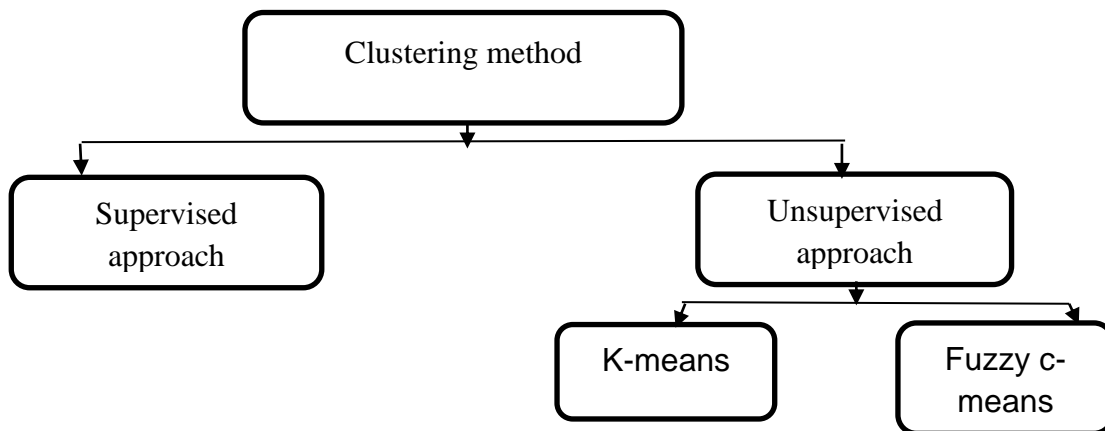
(c) the result of the segmentation.

However, a drawback to this method is that it is recursive: risk of overflows.

### I.7.1.3 Clustering

This method consists of grouping and classifying the pixels of an image into classes by depending on their properties. Each point in the image is associated with a vector of attributes. The classification is then performed on these attribute vectors in order to arrive at a number restricted to homogeneous regions within the image.

So classification is defined as a procedure in which similar pixels in an image are identified and grouped into the same class. It exists two major trends: Unsupervised classification and Supervised classification (See Figure I.9).



**Figure I.9:** Clustering Method [11]

There are two major trends: Unsupervised classification and Classification supervised [11]

#### I.7.1.3.1 Unsupervised classification

In automatic classification methods, classes are obtained by means of algorithms without using a learning base. Several supervised methods have been used to segment meteorological images: K-means, Mean shift, PCM...etc.

We are mainly interested in our work in the FCM algorithm (Fuzzy CMeans) which is based on fuzzy set theory which is an extension of the algorithm K-means.[12]

#### ➤ K-means methods

The k-means method is a classical classification tool that allows to distribute a set of data in homogeneous classes. Most images verify locally homogeneity properties, in particular in terms of light intensity. The algorithm of k-means therefore provides a solution to image segmentation.

The algorithm returns a partition of the data, in which the objects inside of each cluster are as close as possible to each other and as far as possible objects from other clusters. Each cluster in the partition is defined by its objects and its centroid.

### ➤ Fussy c-means methods

Fuzzy classification means that each pixel is not associated with a unique class, but to a vector of probabilities of belonging to the different classes. It allows to pass from the spectral presentation of the pixels to their belonging to the classes presented in the image.[13]

### A. K-Means

This method consists of gathering the pixels into K groups (clusters). K being a parameter prefix that determines the number of regions. And after determining the parameters (color for example) of any region, each point is assigned to the nearest center, to recalculate then the parameter of each region, until they are stable.

### The K-Means algorithm

K-means is one of the best-known methods for unsupervised classification created by Hartigan and Wong in 1979 [14]. It requires as unique parameter a number of classes. It corresponds to the “dynamic clouds” method created in France by Diday in 1971 [15]. This algorithm requires a predefined number of classes.

The k-means algorithm consists of randomly drawing centers of K classes, then iteratively until convergence:

– search for the class closest to each point;

– exchange of points between two classes. to minimize the sum of intra-class distances

$W(C)$  (eq.I-7)

$$w(c) = \frac{1}{2} \sum_{k=1}^k \sum_{c(i)=k} \sum_{c(i')=k} \|x_i - x_{i'}\|^2 \quad (I-7)$$

with  $C(i)$  the class of the point  $x_i$ ,  $\|x_i - x_{i'}\|^2$  the square Euclidean distance between two points  $x_i$  and  $x_{i'}$ :

$$\|x_i - x_{i'}\|_1^2 = \sum_{j=1}^L (x_{ij} - x_{i'j})^2 \quad (I-8)$$

The sum of the intra-class distances can be written [16]:

$$w(c) = \sum_{k=1}^K N_k \sum_{c(i)=k} \|x_i - \bar{x}_k\|^2 \quad (I-9)$$

Where  $x_k = (\bar{x}_{1k}, \dots, \bar{x}_{Lk})$  is the mean vector associated with the  $k^e$  And class, and

$$N_k = \sum_{i=1}^P I(C(i) = k) \quad (I-10)$$

$I(a)$  is the indicator of  $a$  which is 1 if  $a$  holds and 0 otherwise. So the criterion is minimized by assigning the  $P$  observations to the  $K$  classes such that inside each class the difference of the observations to the average of the class, defined by the points of the class, or minimum.

An iterative, top-down algorithm for solving

$$C^* = \min_c \sum_{k=1}^K N_k \sum_{c(i)=k} \|x_i - \bar{x}_k\|^2 \quad (I-11)$$

can be obtained by noting that for each set of observations  $S$ :

$$\bar{x}_s = \operatorname{argmin}_m \sum_{i \in S} \|x_i - m\|^2 \quad (I-12)$$

Thus, we get  $C^*$  by solving the optimization problem:

$$\min_{c\{m_k\}_1^k} \sum_{k=1}^K N_k \sum_{c(i)=k} \|x_i - \bar{x}_k\|^2 \quad (I-13)$$

$c^*$  can be minimized by the alternating optimization procedure described in the Algorithm.

---

### Algorithm Classification kmeans

---

1. For a given assignment  $C$  of the classes, the total variance of the class (eq.I-13), is minimized with respect to  $(m_1, \dots, m_K)$  which modifies the means of the currently defined classes (eq.I-12)
  2. Given the current set of means  $(m_1, \dots, m_K)$ , equation (eq.I-13) is minimized by assignment of each observation to the mean of the closest (current) class, i.e.:
-



$$C(i) = \underset{1 \leq k \leq K}{\operatorname{argmin}} \|x_i - m_k\|^2$$

3. Steps 1 and 2 are iterated until the assignments no longer change

(I-14)

**Note 1.** It is important to run the algorithm many times with centers of <random classes, and to choose the solution for which the C\* function is minimal.

**Note 2.** The metric associated with k-means is the Euclidean metric. It is therefore important to use a space where the data are Euclidean. Factorial spaces of CFA or PCA therefore naturally lend themselves to it since their metric is the Euclidean metric. In addition, these spaces

are small in size, which improves computation times and reduces the Hughes phenomenon.

**K-means Advantages:**

- Simplicity
- Convergence (local min)

**K-means Disadvantages:**

- Expensive in memory space
- Choose a K
- Sensitive to initialization
- Sensitive to artifacts
- Limited to “spherical” clusters [16]

**B. FCM (fuzzy c-means)**

The property of fuzzy sets which is exploited in image processing, and more precisely the classification where the classes, also called regions, are represented by fuzzy sets. This is very useful when the regions cannot be defined in a way clean and precise. Their handling

while keeping the fuzzy character makes it possible to process data imprecise, uncertain and/or redundant in a more flexible way.

Fuzzy classification allows regions to overlap. A non-fuzzy segmentation can be obtained by assigning each pixel to the class for which its degree of membership is maximal.

The fuzzy c-means algorithm generalizes the k-means algorithm [17], allowing fuzzy classification based on set theory or the notion fuzzy set has been introduced in the definition of fuzzy classes. This algorithm has been developed essentially by Bezdek who was interested in it [18] and developed it, based on the original ideas of Ruspini (fuzzy clustering) and Dunn [19]: We present in the following the classification algorithm (FCM) very popular, based on fuzzy logic, known for its efficiency and robustness.

It is based on the minimization of the following objective function:

$$J_m = \sum_{i=1}^N \sum_{j=1}^c U_{ij}^m \|x_j - c_j\|^2, 1 \leq m < \infty \quad (\text{I-15})$$

### **Fuzzy classification algorithm**

The fuzzy C-means algorithm [20] performs iterative optimization by approximating the minima of an error function. There is a whole family of error functions associated with this algorithm which are distinguished by values different takes by an adjustable parameter,  $m$ , called the "fuzzy index" and which determines the degree of blurring of the resulting score. FCMs are a special case algorithms based on the minimization of a criterion or an objective function. In this case, the  $x_j$  are no longer assigned to a single class, but to several through degrees of membership  $u_{ij}$  of vector  $x_j$  to class  $i$ . The goal of the algorithms is then not only to calculate the class B centers but also the set of degrees of membership of the vectors to classes.

### **Operation of the FCM algorithm [21]**

The fuzzy c-means algorithm is an algorithm that is based on the same principle as the K-Means but which uses the logic of fuzzy sets (use of probabilities).

**FCM Algorithm :**

Step1: set the parameters  $c$ ,  $\varepsilon$  and  $m$ :

$c$ : class number.

$\varepsilon$ : Threshold representing the convergence error (for example  $\varepsilon=0.001$ )

$m$ : degree of blur,  $m$  Ballon (1.5, 3) (generally we take  $m=2$ )

Step 2: initialize the membership degree matrix  $U$  by random value in the interval  $[0, 1]$ .

Step 3 : - calculate the centers of the classes by the relation:

$$C_j = \frac{\sum_{i=1}^N u_{ij}^m * x_i}{\sum_{i=1}^N u_{ij}^m} \quad (I-16)$$

-Calculate the objective function by the relation (eq.I-15)

Step4 : - update the membership degree matrix by the relation :

$$U_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{-2}{m-1}}} \quad (I-17)$$

-calculate  $\bar{J}_m$  by the relation (15)

Step 5: repeat steps 3 to 4 until satisfaction of the stopping criterion which is written:

$$|\bar{J}_m - J_m| < \varepsilon \quad (I-18)$$

The following algorithm shows the steps of FCM

1. initialize centers

2. fix the parameter  $m$  (fuzzy coefficient).

3. Calculation of the initial fuzzy partition  $U$  (the membership matrix)

4. Repeat: - calculation of the new centers

- calculation of the new fuzzy partition As long as the stopping criterion is not verified.

**Interest, limitations of FCM [22]****Advantages**

Retains the complexity of mobile centers, while being less sensitive to noise thanks to the weighting by the degrees of membership.

## **Restrictions**

Most of the classification methods used, including the FCM, suffer from three flaws [23]

1. Less sensitivity to noise
2. The number of clusters must be provided in advance (Supervised algorithms),
3. Each region is characterized by a center, and the membership degrees are calculated by bringing in the Euclidean distance, hence a necessarily hyper-spherical shape.
4. For small values of  $\epsilon$  we get the best result, but at the expense of more number of iterations.
5. Huge computation time.
6. Sensitivity to the initial partition (velocity, local minima).

### **I.7.1.3.2 Supervised classification**

#### **A. Knn**

The k-nearest neighbor (k-nearest neighbor, KNN) method is a supervised method. It has been used in statistical estimation and pattern recognition as a nonparametric technique, meaning it makes no assumptions about the distribution of the data.[24]

The KNN algorithm is one of the simplest of all machine learning algorithms. It is a type of learning based on lazy learning. In other words, there is no explicit or very minimal training phase. This means that the training phase is quite fast.[24]

The KNN method assumes that the data is in a feature space. This means that the data points are in metric space. The data can be scalars or even multidimensional vectors.[24, 25]

The k-nearest neighbor method is used for classification and regression. In both cases, the input consists of the k closest training data in the feature space. [24, 25]

To find the class of a new case, this algorithm is based on the following principle: it seeks the k nearest neighbors of this new case, then it chooses among the candidates found the closest and most frequent result.[ 26, 27]

To assign a new individual to a class, the algorithm searches for the  $k$  nearest neighbors among the individuals already classified. Thus, the individual is assigned to the class that contains the most individuals among the candidates found. [26, 27]

This method mainly uses two parameters: a function of similarity to compare individuals in the feature space and the number  $k$  which decides how many neighbors influence the classification.[26]

To test the similarity between two vectors, the distance is used. She permits to measure the degree of difference between two vectors. There are several types of distance among which we find: [28]

- The Euclidean distance:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (\text{I-19})$$

Where:  $x$ ,  $y$  are vectors.

- The Minkowsky distance:

$$d(x, y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}} \quad (\text{I-20})$$

Where:  $x$ ,  $y$  are vectors.

$p$  : parameter

- Distance from Manhattan:

$$\sum_{i=1}^n |x_i - y_i| \quad (\text{I-21})$$

Where:  $x$ ,  $y$  are vectors.

### The $k$ nearest neighbors algorithm

There :

$X$ : training set.

$Y$  : class labels of  $X$

$x$ : unknown individual

For  $i = 1$  a  $m$  do

Calculate the distance  $d(X_i, x)$

End for

Construct the set  $I$  containing indices for  $k$  smallest distance  $d(X_i, x)$

Return Majority label for  $\{Y_i, ou i \in I\}$ [29]

### **Advantages of the k-nearest neighbors method**

The k-nearest-neighbor method has advantages such as:

1. The KNN algorithm is robust against noisy data.[27]
2. The k-nearest neighbor method is effective if the data is large and incomplete.[29]
3. This method is one of the simplest of all machine learning algorithms.[24]

### **Disadvantages of the k-nearest neighbors method**

The k-nearest-neighbor method has drawbacks such as:

1. The need to determine the value of the number of nearest neighbors (the parameter  $k$ ).[25]
2. The prediction time is very long since we have to calculate the distance of all the examples.[27]
3. This method is memory-intensive because it uses large storage capacity for corpora processing.[26]

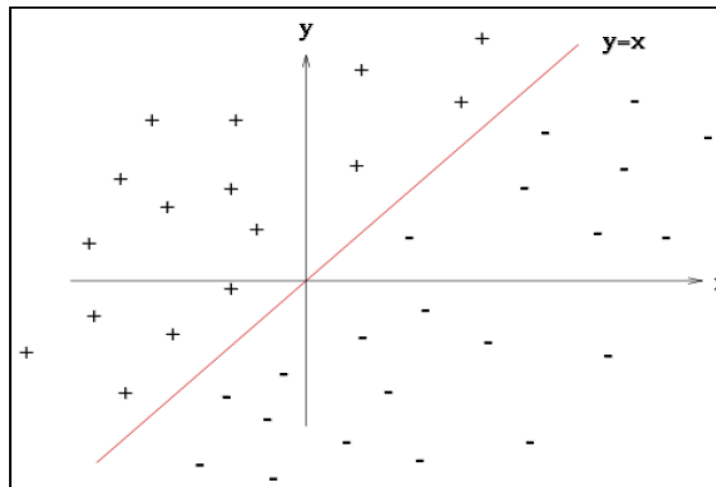
### **B. Support vector machines (SVM)**

Introduced by Vapnik in 1990, support vector machines are supervised learning techniques for solving classification and regression problems. They are based on two main notions: the notion of maximum margin and the notion of kernel function.[25,26]

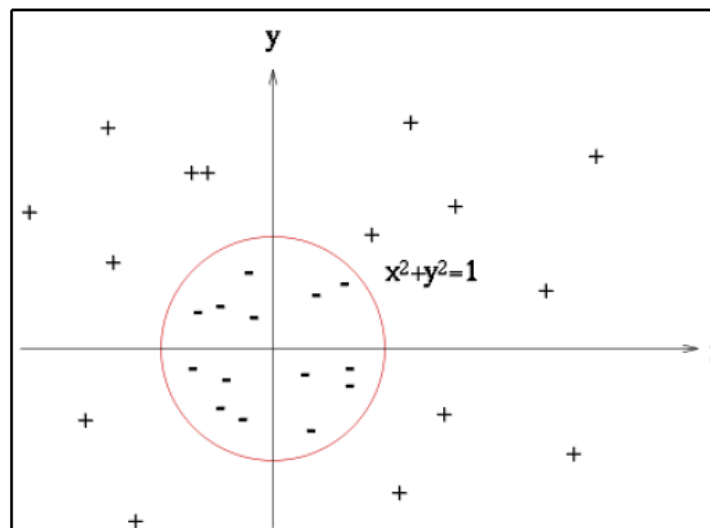
Vector support machines can be used to solve various problems in bioinformatics, information retrieval and computer vision, etc.[30]

## The principle of SVMs

The goal of SVMs is to find a separator between two classes that is maximally far from any point in the training data. If we manage to find a linear separator, i.e. there is a separating hyperplane, then the problem is said to be linearly separable, otherwise it is not linearly separable and there does not exist a separating hyperplane. [26, 30]



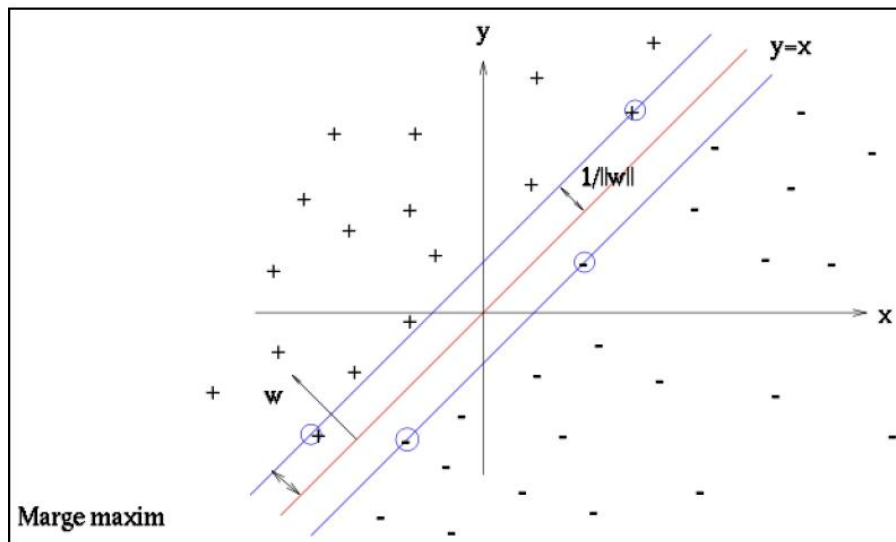
**Figure I.10** :Two-class classification problem with a linear separator [30]



**Figure I.11** :Two-class classification problem with a nonlinear separatrix [30]

For two classes and linearly separable data (See Figure I.10), there are many possible linear separators. SVMs choose only the optimal one, i.e. search for a decision surface that is as far away as possible from any data point. This distance from the decision surface to the nearest data point determines the classifier's maximum margin. Indeed, to obtain an optimal

hyperplane, it is necessary to maximize the margin between the data and the hyperplane (See Figure I.12). [26, 30]



**Figure I.12:** Optimal hyperplane with maximum margin [30]

To solve the problem of separating nonlinearity (See Figure I.11), the idea of SVMs is to increase the dimension of data space. In this case, then it is likely that there is a linear separator. Indeed, the chance of finding a separating hyperplane increases proportionally with the data space dimension. [26, 30]

This space resizing is based on using the Kernel function. There are several types of kernel function such as Gaussian, polynomial and sigmoid.[26]

### Advantages of SVMs

SVMs have several advantages, including:

1. They have a solid theoretical basis.[31]
2. SVMs are efficient in high-dimensional spaces.[26]
3. Different kernel functions can be specified. [26]

### Disadvantages of SVMs

Despite their performance, SVMs also have weaknesses, including the following:

1. They use complex mathematical functions for classification. [26]

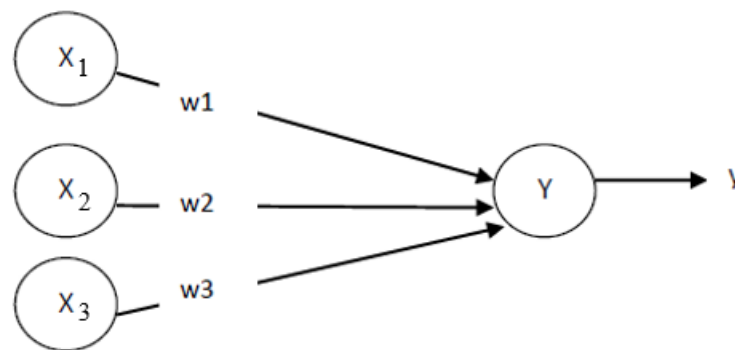


2. Supported vector machines require a huge amount of time during testing. [31]

### C. Neurons network

Neural networks were developed as a generic mathematical model to model biological neurons. They have a number of information processing elements called neurons.[32]

Each neuron has its own internal state interpreted by the activation function. It sends its activation to other neurons in the form of signals. The connection between neurons is achieved through oriented and weighted links.[32]



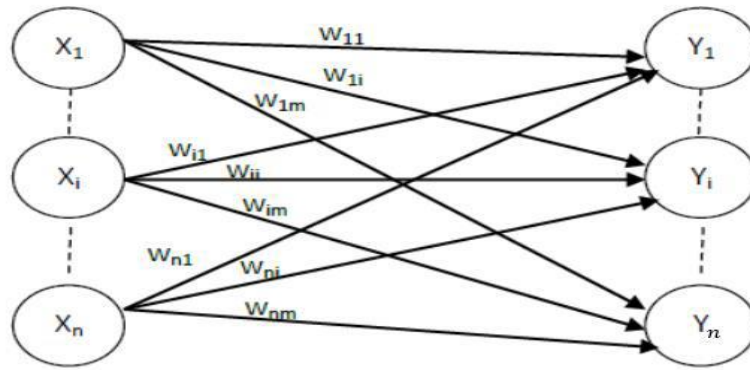
**Figure I.13:** Artificial neuron with a single output [32]

Neuron  $Y$  receives the inputs of  $X_1, X_2$  et  $X_3$  which have as output values  $x_1, x_2$  et  $x_3$ . The connecting link weights of  $X_1, X_2$  et  $X_3$  are  $w_1, w_2$  et  $w_3$ . The input value of neuron  $Y$  is:  $\mathcal{Y} = w_1x_1 + w_2x_2 + w_3x_3$ . The output signal  $y$  is determined by the activation function  $f(\mathcal{Y})$ .

Neural networks are characterized by architecture (organization of neurons), learning (method of determining connection weights), and by their activation function.[32]

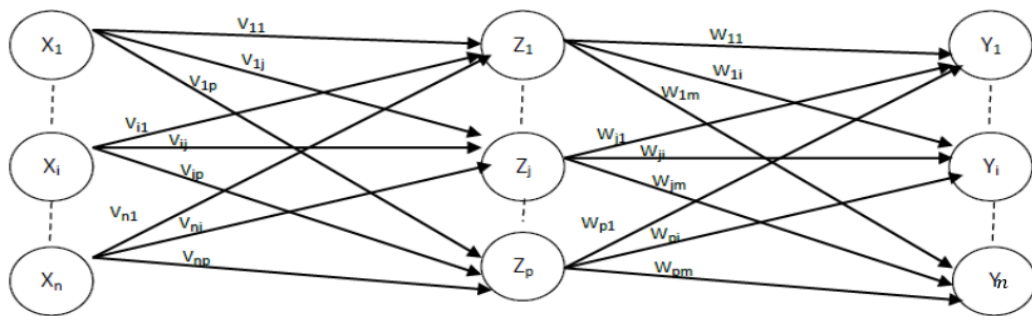
#### Architecture

Neural networks are often classified into two architectures: single-level and multi-level neural networks. The number of levels is calculated without considering the units (See Figure I.14).[32]



**Figure I.14:** The single-level neural network [32]

Neurons in the input layer should only pass and distribute inputs and not perform any computation. Thus, the only true layer of neurons is the one on the right. Each of the  $X_1, X_2, \dots, X_n$  inputs is connected to each neuron in the output layer through the link weight. As each value of the outputs  $Y_1, Y_2, \dots, Y_n$  is calculated from the same set of input values, each output is modified according to the link weights. [32, 33]



**Figure I.15:** The multi-level neural network [33]

Figure I.15 shows the multi-level neural network the latter differs from the single-level network by having one or more hidden layers. In this structure, input nodes pass information to units in the first hidden layer, then outputs from the first hidden layer are passed to the next layer, and so on. [32, 33]

The multi-tier network can also be viewed as a cascade of single-tier network groups. The level of complexity translates into the number of single-layer networks that are combined into that type of network. The designer of a neural network should consider how many hidden layers are required, depending on the complexity of the desired computation. [32, 33]

## Learning

The method of parameterizing the weights (learning) is an important feature to distinguish between different types of neural networks. Two modes of learning exist: supervised learning, and unsupervised learning.[32]

### Supervised learning

In this type of learning, inputs and outputs are provided beforehand. Then the network processes the inputs and compares its results to the desired outputs. The weights are then adjusted using the errors propagated through the system. This process happens repeatedly as long as the weights are continuously improved. The set of data that enables training is called the training set. [32, 33]

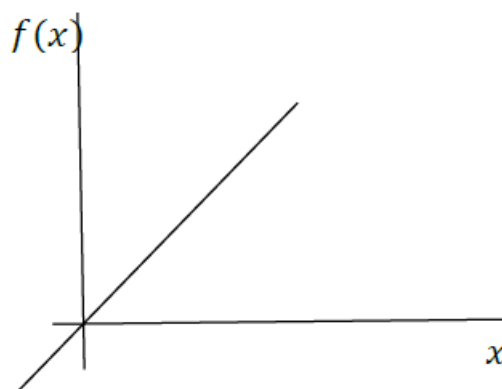
### Unsupervised learning

In unsupervised learning, the network is provided with inputs but not with desired outputs. The system itself must then decide which features it will use to group the input data. This is often called self-organization or adaptation. [32, 33]

### Activation function

The basic function of an artificial neuron is to sum all the input data to produce an output function. This function is the identity function (See Figure I.16).[32]

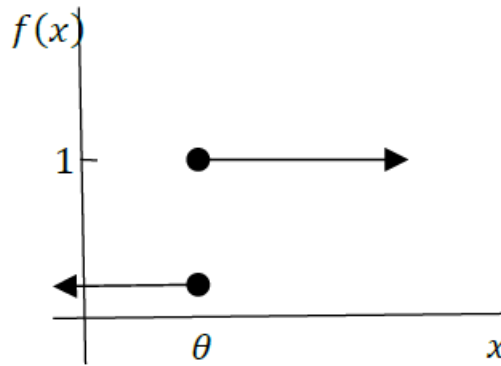
$$f(x)=x \text{ for all } x \quad (\text{I-22})$$



**Figure I.16:** Identity function [32]

Another activation function is the function of a threshold. The output value is 1 when the weighted sum of the input values is greater than or equal to  $\theta$ , otherwise the value is 0 (See Figure I.17). [32]

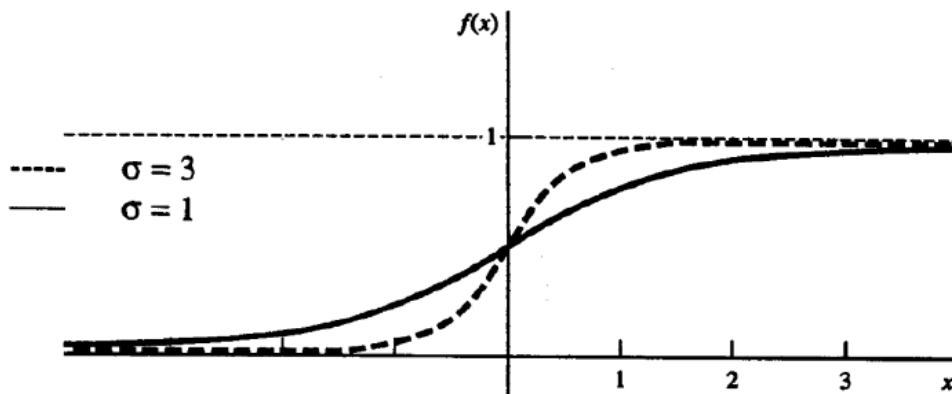
$$f(x) = \begin{cases} 1 & \text{si } x \geq \theta \\ 0 & \text{si } x < \theta \end{cases} \quad (\text{I-23})$$



**Figure I.17:** Function of a threshold  $\theta$  [32]

The sigmoid function from 0 to 1 is often used as an activation function for neural networks in which the desired output values are either binary or in an interval between 0 and 1 (See Figure I.18). [32]

$$f(x) = \frac{1}{1 + \exp(-\sigma x)} \quad (\text{I-24})$$



**Figure I.18:** Function sigmoïde [32]

### Advantages of neural networks

Neural networks represent advantages, such as:

1. Neural networks are flexible and generic. They can solve different types of problems whose result can be: classification, data analysis, etc.[34]

2. They deal with unstructured problems about which no information is available in advance.[34]

3. Neural networks perform well because even in very complex domains they perform better than statistical or decision trees.[35]

4. Neural networks operate on incomplete or noisy data. This information gap can be filled by adding other neurons to the hidden layer.[34]

### **Disadvantages of neural networks**

Neural networks also have drawbacks, such as:

1. Slow learning.[26]

2. The difficulty of choosing initial values of the connection weights as well as the adaptation of the learning step.[35]

3. Learning at the expense of generalization.[35]

4. If there is an error in the output results, the user has no information about the internal operation.[26]

### **I.7.2 Contours based approaches**

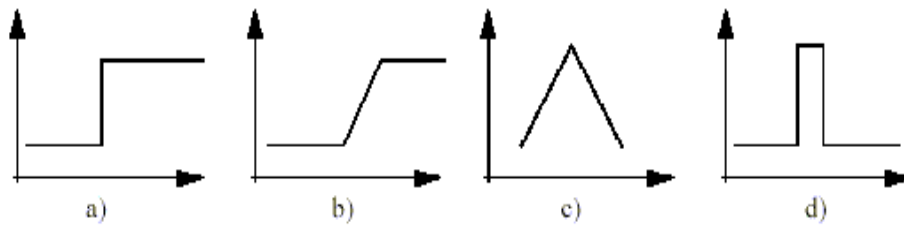
Edge detection is often the first problem encountered when processing a picture. The difficulty increases with the amount of noise present.

#### **Definition of contour**

a contour can be roughly defined as a border between two regions where the intensity of the pixels changes abruptly.

Generally, the use of such a contour operator is combined with thresholding and as being the latter is generally imperfect, we obtain, on the one hand, contours which do not limit closed regions. So we have to resort to closure algorithms outlines. On the other hand,

the areas of strong variations do not necessarily correspond to an object outline. Then a post-processing is necessary to analyze the different contours obtained (See Figure I.19)



**Figure I.19:** 3 Different types of outlines [36]

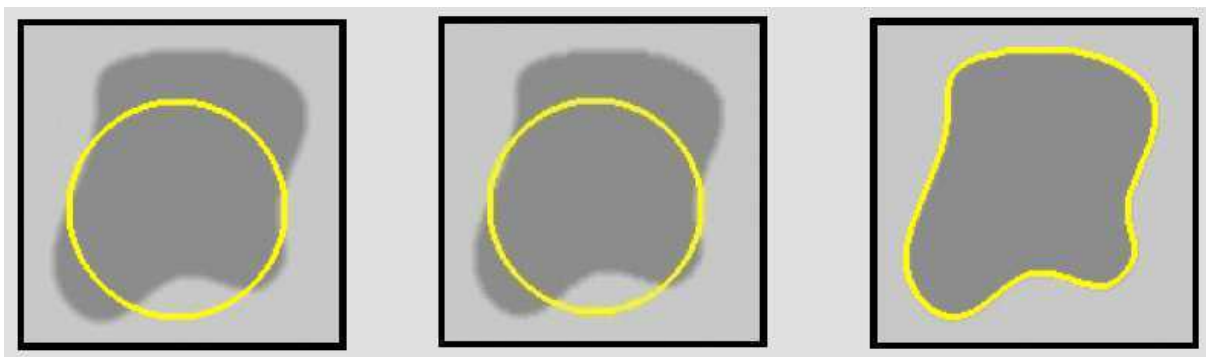
### I.7.2.1 Parametric contour

The idea of this method is to move the points to bring them closer to the strong radiant areas while maintaining characteristics such as the curvature of the contour or the distribution of points on the outline or other constraints related to the layout of the points.

At the start of the algorithm, the contour is arranged uniformly around the object to be trimmed then it will retract to fit the shape as well as possible.

At each iteration, the algorithm will try to find a better positioning for the outline to minimize drifts with respect to the constraints used.

The algorithm will stop when it is no longer possible to improve the positioning or simply when the maximum number of iterations has been reached. We use the concepts of internal and external energies to respectively characterize the shape of the contour and its positioning on the image taking into account the gradient lines (See Figure I.20).[36]



**Figure I.20:** Active contour principle [36]

### I.7.2.2 Level set

The Zero Level Sets method is a numerical simulation method used for the evolution of curves and surfaces in discrete domains [37].

The basic idea of the level sets method is to consider a curve (or interface) moving like the zero level of a higher dimensional function. For a curve in 2D, this interface ( $\Psi$ ) is the intersection of a hypersurface (of dimension 3) with a map.

The points defining this interface will move towards the normal at a speed  $F$  according to the following equation [38]:

$$\Psi_{t+1} + F |\nabla \Psi_t| = 0 \quad (\text{I-25})$$

This speed  $F$  is made up of three terms: a constant term (similar to the force of inflation used in deformable models), a term depending on the local curvature at each point and a term depending on the image.

The numerical scheme of the interface displacement equation is described by the equation:

$$\Psi_{n+1} = \Psi_n - dt * k_1(x, y) * (U_n - \epsilon k) * |\nabla \Psi_t|, \epsilon \in [0, 1] \quad (\text{I-26})$$

### I.7.2.3 advantages and disadvantages of contour approaches

Edge detectors give good results when the processed image is not too noisy, otherwise they require the use of smoothing filters which can affect the borders between the homogeneous regions and thus relocate the contour points.

In addition, the thresholds are difficult to set without a priori information which is not always available. Optimal filters present a good compromise between good detection and a good location.

The contour point map obtained at the end of the segmentation process should often be transformed into a set of regions richer in information and more easily exploitable by the high level. A contour closing step will therefore be generally necessary before starting the interpretation phase.[36]

## **I.8. Conclusion**

Segmentation is an essential step in image processing and because there is no single method of segmentation, it is difficult to define, in an absolute way, a “good” segmentation. Segmentation is not an end in itself, its quality depends on the results obtained by downstream processing that uses the extracted primitives.

This chapter briefly presents a state of the art of the main segmentation methods.

Each has its advantages and disadvantages: the major problem of segmentation contour-based is contour closure, on the other hand, segmentation based on the regions can be computationally "greedy". In the absence of specific contexts, Since the quality of the result is mainly Subjective and the finality of use of the segmentation obtained influences the perception of quality of this segmentation method.

As a result, choosing one technique over another is quite difficult.



## Chapitre II. Segmentation by fuzzy c-means method

### II.1. Introduction

Classification consists of grouping data into the most homogeneous categories possible, like most scientific activities, the development of the various techniques of classification benefited greatly from the event and the improvement of tools classification methods include the fuzzy c-mean (FCM).

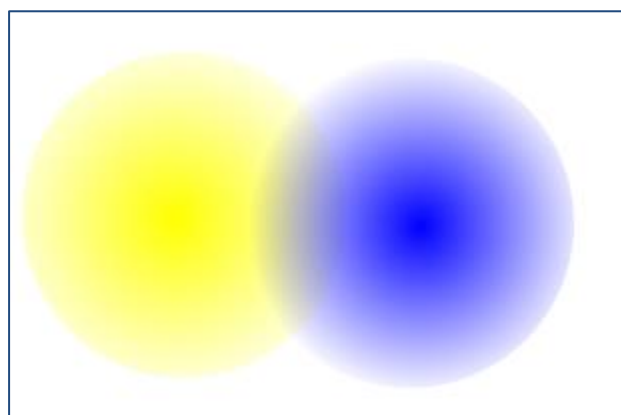
In this chapter we present the essential points of fuzzy logic, the FCM classification and some of its variants as well as its advantages and disadvantages.

### II.2. fuzzy logic

We hear more and more often about fuzzy logic as a method offering outstanding performance, allowing to manage complex systems of intuitive way.

Lukasiewicz proposed in 1920 a logic having the following three truth values: "true", "false" and "doubt". These values were represented by the set  $\{0, 1, 0.5\}$ , and were then extended to the interval  $[0,1]$ .

Fuzzy logic was developed and formalized by Lofti Zadeh in 1965. This theory is based on fuzzy set theory and logic. In particular, it generalizes the Modus Ponens of classical logic (See Figure II.1).



**Figure II.1:** Fuzzy logic

### II.3. The fuzzy subset theory

fuzzy subset theory is a mathematical theory of the domain abstract algebra developed by Lotfi Zadeh in 1965 to represent mathematically the inaccuracy relating to certain classes of objects. This theory is based on the notion of degree of belonging. These degrees are values that express membership uncertain from a pixel to a region. The degree of membership is in the interval  $[0,1]$  and the classes obtained are not necessarily disjoint. In this case, the data is no longer assigned to a single class, but to several through degrees of membership of the vector to class  $i$ .

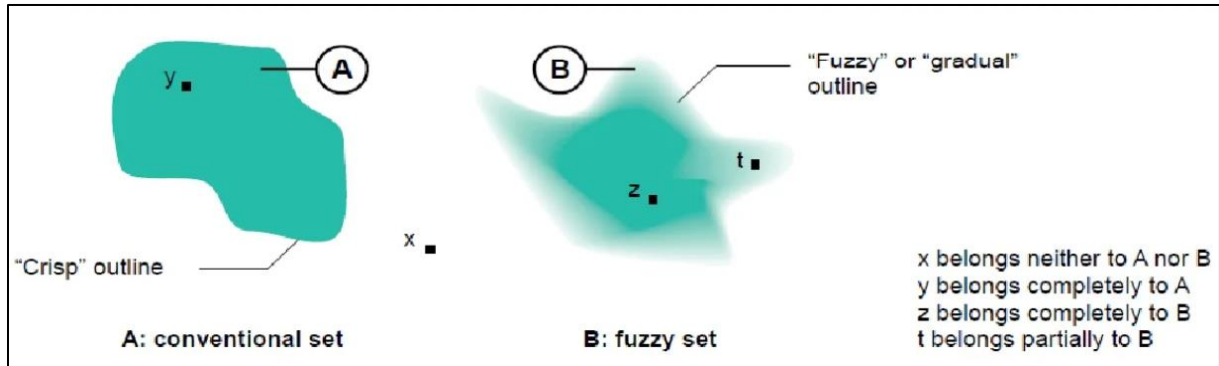
#### Concept of set and subset

A vocabulary problem arises for the word fuzzy which means (vague, confused). A fuzzy set theory makes it possible to define all the classical operations (conjunction, disjunction, etc.). It encompasses classical set theory. One of the concepts fundamental in mathematics is the notion of set, created by the mathematician George Cantor. He defined sets as collections of objects, called elements, although specified and all different. In set theory, an element belongs or does not belong not to a set. Thus, we can define a set by a characteristic function for all the elements  $x$  of the universe of discourse  $U$ . The universe of discourse is the referential set which contains all the elements that are related to the given context. Function characteristic of the set  $E$ ,  $\mu_E: U \rightarrow \{0,1\}$  is defined as:

$$E(x) = 1 \text{ if } x \text{ belongs to } E$$

$$E(x) = 0 \text{ if } x \text{ does not belong to } E$$

Zadeh extended the notion of a classical set to the fuzzy set which defines it as being “a collection such that the membership of any element in this collection can take all values between 0 and 1”. Fuzzy set theory is based on the notion partial membership: each element belongs partially or gradually to the fuzzy sets that have been defined. The contours of each fuzzy set are not "sharp", but “fuzzy” or “gradual” (See Figure II.2) [39]



**Figure II.2:** Comparison of a classical set and a fuzzy set [39]

#### II.4. Clustering definition

Clustering is a process for classifying objects or patterns in such a way that samples of the same cluster are more similar to one another than samples belonging to different clusters. There are two main clustering strategies: the hard clustering scheme and the fuzzy clustering scheme. The conventional hard clustering methods classify each point of the data set just to one cluster.

As a consequence, the results are often very crisp, i.e., in image clustering each pixel of the image belongs just to one cluster. However, in many real situations, issues such as limited spatial resolution, poor contrast, overlapping intensities, noise and intensity inhomogeneities reduce the effectiveness of hard (crisp) clustering methods. Fuzzy set theory has introduced.

the idea of partial membership, described by a membership function. Fuzzy clustering, as a soft segmentation method, has been widely studied and successfully applied in image clustering and segmentation is the most popular method used in image segmentation because it has robust characteristics for ambiguity and can retain much more information than hard segmentation methods [40].

#### II.5. Fcm algorithm

fuzzy c-means (FCM) algorithm is the most popular method used in image segmentation because it has robust characteristics for ambiguity and can retain much more information than hard segmentation methods [40].

Although the conventional FCM algorithm works well on most noise-free images, it is very sensitive to noise and other imaging artifacts, since it does not consider any information about spatial context.

To compensate this drawback of FCM, a pre-processing image smoothing step has been proposed. However, by using smoothing filters important image details can be lost, especially boundaries or edges. Moreover, there is no way to control the trade-off between smoothing and clustering.

Thus, many researchers have incorporated local spatial information into the original FCM algorithm to improve the performance of image segmentation.

The fuzzy c-means (FCM) clustering algorithm was first introduced by Dunn [41] and later extended by Bezdek [42].

The algorithm is an iterative clustering method that produces an optimal  $c$  partition by minimizing the weighted within group sum of squared error objective function  $J_m$  [42]:

$$J_m = \sum_{i=1}^N \sum_{j=1}^c u_{ji}^m d^2(x_i, v_j) \quad (\text{II-1})$$

where  $X = \{x_1, x_2, \dots, x_N\} \subseteq IR^m$  is the data set in the  $m$ -dimensional vector space,  $N$  is the number of data items,  $c$  is the number of clusters with ( $2 \leq c < N$ ),  $u_{ji}$  is the degree of membership of  $x_i$  in the  $j^{\text{th}}$  cluster,  $m$  is the weighting exponent on each fuzzy membership,  $v_j$  is the prototype of the center of cluster  $j$ ,  $d^2(x_i, v_j)$  is a distance measure between object  $x_i$  and cluster center  $v_j$ . A solution of the object function  $J_m$  can be obtained through an iterative process, which is carried as follows:

- 1) set values for  $c$ ,  $m$ , and  $\epsilon$ ,
- 2) initialize the fuzzy partition matrix  $U^{(0)}$ ,
- 3) set the loop counter  $b = 0$ ,
- 4) calculate the  $c$  cluster centers  $v_j^{(b)}$  with  $U^{(b)}$ :

$$v_j^{(b)} = \frac{\sum_{i=1}^N (u_{ji}^{(b)})^m x_i}{\sum_{i=1}^N (u_{ji}^{(b)})^m} \quad (\text{II-2})$$

- 5) calculate the membership matrix  $U^{(b+1)}$ :

$$u_{ji}^{(b+1)} = \frac{1}{\sum_{k=1}^c \left( \frac{d_{ji}}{d_{ki}} \right)^{\frac{2}{m-1}}} \quad (\text{II-3})$$

- 6) if  $\max \{ U^{(b)} - U^{(b+1)} \} < \epsilon$  then stop, other wise, set  $(b = b + 1)$  and go to step 4.

## II.6. Fcm algorithm shortcomings

Image segmentation based FCM suffer from initialize cluster centroids and number of clusters. Image segmentation by clustering based approach still has major weaknesses such as in the capability to obtain a fully automatic segmentation approach without prior knowledge and location centroid. Meanwhile, the optimal clusters number in each image is supposed to be known and provided by the experts or operators, making this approach a semiautomatic segmentation method which consumes time because of the experts' availability. In the image segmentation that is cluster based, determining the number of clusters in a given dataset is considered as a main challenge. In fact, many researchers have worked during the last several years on the development of a clustering approach which could determine the appropriate number of clusters without experts, operators and any prior knowledge. A fuzzy clustering approach based on metaheuristic optimization search algorithms is considered a suitable choice to solve the problems of fuzzy clustering based. [40]

## II.7. Fcm variants

many researchers have incorporated local spatial information into the original FCM algorithm to improve the performance of image segmentation:

### II.7.1 Fuzzy Clustering with Constraints (FCMS) and its Variants

Ahmed and al. [43] proposed a modification of the standard FCM by introducing a term that allows the labelling of a pixel to be influenced by labels in its immediate neighborhood [44].

The neighborhood effect acts as a regularize and biases the solution toward piecewise-homogeneous labelling. The modified objective function of FCM S is defined as follows:

$$J_m = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m ||x_k - v_i||^2 + \frac{\alpha}{N_R} \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m \sum_{r \in N_k} ||x_r - v_i||^2 \quad (\text{II-4})$$

where  $x_k$  is the gray level value of the  $k^{\text{th}}$  pixel,  $N$  is the total number of pixels,  $v_i$  is the prototype value of the  $i^{\text{th}}$  cluster,  $u_{ik}$  represents the fuzzy membership of the  $k^{\text{th}}$  pixel.

with respect to cluster  $i$ ,  $N_R$  is its cardinality,  $x_r$  represents the neighbor of  $x_k$ .

The parameter  $m$  is a weighting exponent on each fuzzy membership that determines the amount of fuzziness of the resulting classification. The parameter  $\alpha$  is used to control the effect of the neighbors term. By definition, each sample point  $x_k$  satisfies the constraint that

$$\sum_{i=1}^c U_{ik} = 1.$$

Two necessary but not sufficient conditions for  $J_m$  to be at its local extreme will be obtained as follows:

$$\mathbf{u}_{ik} = \frac{\left(\|x_k - v_i\|^2 + \frac{\alpha}{N_R} \sum_{r \in N_k} \|x_r - v_i\|^2\right)^{-\frac{1}{(m-1)}}}{\sum_{j=1}^c \left(\|x_k - v_j\|^2 + \frac{\alpha}{N_R} \sum_{r \in N_k} \|x_r - v_j\|^2\right)^{-\frac{1}{(m-1)}}} \quad (\text{II-5})$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^N \mathbf{u}_{ik}^m \left(x_k + \frac{\alpha}{N_R} \sum_{r \in N_k} x_r\right)}{(1+\alpha) \sum_{k=1}^N \mathbf{u}_{ik}^m} \quad (\text{II-6})$$

The second term  $\sum_{j \in N_k} \frac{x_j}{N_R}$  in the numerator of (eq.II.6) is in fact a neighbor average gray level value around  $x_k$  within a window. The image composed by all the neighbor average values around the image pixels forms a so-called mean-filtered image.

A shortcoming of Eqs. (II.5) and (II.6) is that computing the neighbor term will take much time in each iteration step.

In order to reduce the computation, Chen and Zhang [46] proposed FCM S1, a variant of FCM S, where the neighborhood term is presented much more simplified. The objective function is written as follows:

$$J_m = \sum_{i=1}^c \sum_{k=1}^N \mathbf{u}_{ik}^m \|x_k - v_i\|^2 + \alpha \sum_{i=1}^c \sum_{k=1}^N \mathbf{u}_{ik}^m \|\bar{x}_k - v_i\|^2 \quad (\text{II-7})$$

Where  $\bar{x}_k$  is the average of neighboring pixels lying within a window around  $x_k$ .

Unlike (II.4),  $\bar{x}_k$  can be computed in advance, reducing the whole computation time since

$$\sum_{r \in N_k} \frac{\|x_r - v_i\|^2}{N_R} \quad \text{in (eq.II-5) is replaced by } \|\bar{x}_k - v_i\|^2$$

An iterative algorithm for the minimization of (II.7) with respect to  $u_{ik}$  and  $v_i$  can be similarly to FCM\_S derived as follows:

$$\mathbf{u}_{ik} = \frac{\left(\|x_k - v_i\|^2 + \alpha \|\bar{x}_k - v_i\|^2\right)^{-\frac{1}{(m-1)}}}{\sum_{j=1}^c \left(\|x_k - v_j\|^2 + \alpha \|\bar{x}_k - v_j\|^2\right)^{-\frac{1}{(m-1)}}} \quad (\text{II-8})$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^N \mathbf{u}_{ik}^m (x_k + \alpha \bar{x}_k)}{(1+\alpha) \sum_{k=1}^N \mathbf{u}_{ik}^m} \quad (\text{II-9})$$

The essence of FCM\_S1 is to make both the original image and the corresponding local neighbor average or mean-filtered image have the same segmentation result with aiming to guarantee the gray homogeneity. FCM\_S1 not only considerably reduces the execution time for clustering an image but also improves the robustness to Gaussian noise [45]

However, FCM\_S1 is unsuitable for images corrupted by impulse noise [45]. In order to overcome this problem, Chen and Zhang [45] designed the FCM\_S2, a variant of FCM S1 in which the median-filtered image is replaced by the median-filtered one. To enhance the robustness to impulse noises

The iterative algorithm that solves the objective functions of FCM\_S, FCM\_S1 and FCM\_S2 is more or less the same used in the classical FCM algorithm. Just the corresponding functions should be replaced and the mean and median-filtered images (only for FCM\_S1 and FCM\_S2) should be calculated before the first iteration.

---

### FCM\_S, FCM\_S1 and FCM\_S2 Algorithms

---

Step 1: The number  $c$  of these prototypes or clusters ranges from 2 to  $c_{\max}$ , fix a certain value  $c$  and then select initial class prototypes and set  $\varepsilon > 0$  to a very small value.

Step 2: For FCM\_S1 and FCM\_S2 only, compute the mean or median filtered image.

Step 3: Update the partition matrix using (Eq.II-5) (FCM\_S) or (Eq.II.8) (FCM\_S1 and FCM\_S2).

Step 4: Update the prototypes using (Eq.II-6) (FCM\_S) or (Eq.II-9) (FCM\_S1 and FCM\_S2).

Repeat Steps 3 and 4 until the following termination criterion is satisfied:

$$|V_{\text{new}} - V_{\text{old}}| < \varepsilon$$

where  $V = [v_1, v_2, \dots, v_c]$  are the vectors of cluster prototypes.

---

### II.7.2 C. Enhanced Fuzzy C-Means Clustering (EnFCM)

Szilagy *et al.* [46] proposed the EnFCM algorithm to speed up the clustering process for gray level images. In order to accelerate FCM S, a linearly-weighted sum image  $\xi$  is unadvanced formed from the original image and its local neighbor average image in terms of:

$$\xi_k = \frac{1}{1+\alpha} (x_k + \frac{\alpha}{N_R} \sum_{j \in N_k} x_j) \quad (\text{II-10})$$

where  $\xi_k$  denotes the gray level value of the  $k$ -th pixel of the image  $\xi$  and  $N_k$  stands for the set of neighbors ( $x_j$ ) falling into a window around  $x_k$ . The parameter  $\alpha$  plays the same role as before, that is to control the effect of the neighbors term. Then, the clustering method [47] is performed on the gray level histogram of the newly generated image  $\xi$ .

As a consequence, the objective function in this case is defined as:

$$J_s = \sum_{i=1}^c \sum_{l=1}^q \gamma_l u_{il}^m (\xi_l - v_i)^2 \quad (\text{II-11})$$

where  $v_i$  represents the prototype of the  $i$ -th cluster,  $u_{il}$  represents the fuzzy membership of gray level value  $l$  with respect to cluster  $i$ ,  $q$  denotes the number of gray levels of image  $\xi$ , which is generally much smaller than  $N$ , and  $\gamma_l$  is the number of pixels having gray level value equal to  $l$ , where  $l=1, \dots, q$ . Thus,  $\sum_{l=1}^q \gamma_l = N$  and under the constraint that  $\sum_{i=1}^c u_{il} = 1$  for any  $l$ , the  $J_s$  (II.11) is minimized, using the following equations for calculation of the membership partition matrix and the cluster centers:

$$u_{il} = \frac{(\xi_l - v_i)^{-\frac{2}{m-1}}}{\sum_{j=1}^c (\xi_l - v_j)^{-\frac{2}{m-1}}} \quad (\text{II-12})$$

$$v_i = \frac{\sum_{l=1}^q \gamma_l u_{il}^m \xi_l}{\sum_{l=1}^q \gamma_l u_{il}^m} \quad (\text{II-13})$$

The iterative process of the EnFCM algorithm is similar to FCM, but it is applied to the new image  $\xi$  (eq. II-10) by using (eq. II-12) and (eq. II-13).

Due to fact that the gray level value of the pixels is generally encoded with 8 bit resolution (256 gray levels), the number  $q$  of gray levels is generally much smaller than the size  $N$  of the image. Thus, the execution time is significantly reduced.

EnFCM provides comparable segmenting results to FCM S, but the segmenting quality depends on the chosen window size, the parameter  $\alpha$  and the filtering method. If the parameter  $\alpha$  is chosen large enough, then the method is resistant to noise, but, on the other hand, when  $\alpha$  is chosen small enough the segmented image maintain its sharpness and details. However, when there is no prior knowledge about the image noise, the selection of parameter  $\alpha$  is not an easy task and has to be made by experience or by using the trial-and-error method.



### II.7.3 Fast Generalized Fuzzy C-Means Clustering (FGFCM)

Cai *et al.* [47] proposed the fast generalized fuzzy c-means (FGFCM) algorithm to improve the clustering results, as well as to facilitate the choice of the neighboring control parameter.

In order to improve the clustering results FGFCM exploits a local similarity measure that combine both spatial and gray level image information, in terms of:

$$S_{ij} = \begin{cases} e^{-\frac{\max(|p_i-p_j|, |q_i-q_j|)}{\lambda_s} - \frac{\|x_i-x_j\|^2}{\lambda_g \sigma_i^2}} & i \neq j \\ 0 & i = j \end{cases} \quad (\text{II-14})$$

where the  $i$ -th pixel is the center of the local window and the  $j$ -th pixel represents the set of the neighbors falling into the window around the  $i$ -th pixel.  $(p_i, q_i)$  are the coordinates of pixel  $i$  and  $x_i$  is its gray level value.  $\lambda_s$  and  $\lambda_g$  are two scale factors playing a role similar to factor  $\alpha$  in EnFCM, and  $\sigma_i$  is defined as:

$$\sigma_i = \sqrt{\frac{\sum_{j \in N_i} \|x_i - x_j\|^2}{N_R}} \quad (\text{II-15})$$

FGFCM incorporates local and gray level information (II.14) into its objective function generating in advance a new image  $\xi$  as follows:

$$\xi_i = \frac{\sum_{j \in N_i} S_{ij} x_j}{\sum_{j \in N_i} S_{ij}} \quad (\text{II-16})$$

where  $\xi_i$  denotes the gray level value of the  $i$ -th pixel of the image  $\xi$ ,  $x_j$  represents the gray level value of the neighbors of  $x_i$  (window center),  $N_i$  is the set of neighbors falling in the local window and  $S_{ij}$  is the local similarity measure between the  $i$ -th and the  $j$ -th pixel.

Furthermore, Cai *et al.* [47] proposed two variants of the FGFCM algorithm, the FGFCM S1 and the FGFCM S2, which incorporate modified similarity measures in their objective functions. FGFCM S1 uses  $S_{ij} = 1$  for all  $i$  and  $j$ , and naturally  $\xi_i$  is equal to the mean of the neighbors within a specified window, including the  $i$ -th pixel. On the other hand, FGFCM S2, uses as similarity measure the  $S_{ij} = \text{median}\{x_j\}$ , that is,  $\xi_i$  is the median of the neighbors within a specified window, including the  $i$ -th pixel.

---

**The FGFCM algorithm can be summarized as follows:**

---

- 1) set values for  $c$ ,  $m$ , and  $\zeta$ ,
  - 2) compute the new image  $\zeta$  in terms of (eq .II-16),
  - 3) initialize the fuzzy partition matrix  $U^{(0)}$ , and set the loop counter  $b = 0$ ,
  - 4) update the cluster centers  $v_j^{(b)}$  using (eq.II-13):
  - 5) update the membership matrix  $U^{(b+1)}$  using (eq.II-12):
  - 6) if  $\max \{U^{(b)} - U^{(b+1)}\} < \zeta$  then stop, otherwise, set  $b = b + 1$  and go to step 4.
- 

The significant reduction of execution time attributes to taking into account a range of distribution of gray levels of the given image, as in EnFCM algorithm.

FGFCM and its variants, provide good enough segmenting results, but the segmenting quality depends on the chosen window size and the parameters  $\lambda_s$  and  $\lambda_g$ . Parameter  $\lambda_s$  could be fixed to 3 [47]. Thus, parameter  $\lambda_g$  controls the algorithm. If  $\lambda_g$  is chosen large enough, then the method is resistant to noise, but on the other hand, when  $\lambda_g$  is chosen small enough the segmented image maintain its sharpness and details. However, because there is no prior knowledge about the image noise, the selection of parameter  $\lambda_g$  is not an easy task, and has to be made by experience or by using the trial and-error method.

#### **II.7.4 FUZZY LOCAL INFORMATION C-MEANS (FLICM) CLUSTERING ALGORITHM**

Krindis and Chatzis [48] have developed Fuzzy Logic Information C-Means Clustering (FLICM) algorithm to overcome the drawbacks in algorithms available in the Previous Algorithms

- 1) Although the introduction of local spatial information enhances their insensitiveness to noise to some extent, they still lack enough robustness to noise and outliers, especially in absence of prior knowledge of the noise.
- 2) There is a crucial parameter  $\sigma$  (or  $\lambda$ ) in their objective functions, used to balance between robustness to noise and effectiveness of preserving the details of the image. Generally, its selection has to be made by experience or trial and error experiments.
- 3) They are all applied on a static image, which has to be computed in advance. Details of the original image could be lost depending on the method used to generate the new image.

In order to overcome the above mentioned disadvantages a new factor in FCM objective function is needed. The new factor should have some special characteristics:

$$G_{ki} = \sum_{\substack{j \in N_i \\ i \neq j}} \frac{1}{d_{ij}^{+1}} (\mathbf{1} - \mathbf{u}_{kj})^m \|\mathbf{x}_j - \mathbf{v}_k\|^2 \quad (\text{II-17})$$

the new factor  $G_{ki}$  has following characteristics :

- to incorporate local spatial and local gray level information in a fuzzy way in order to preserve robustness and noise insensitiveness.
- to control the influence of the neighborhood pixels depending on their distance from the central pixel.
- to use the original image avoiding preprocessing steps that could cause detail missing.
- to be free of any parameter selection.

where  $i$ th pixel is the center of the local window e.g,  $3 \times 3$ ,  $k$  is reference cluster and  $j$ th pixel belongs to the set of neighbors that fall into a window around  $i$ th pixel ( $N_i$ ) and  $d_{ij}$  is Euclidean distance between pixels  $i$  and  $j$  [3]. The factor  $G_{ki}$  is free from any control parameter for balancing image detail and noise. The stability between noise and image details is automatically attained by the fuzziness of every pixel of the image. The factor  $G_{ki}$  formulates the influence of the pixels within the local window, to change flexibly according to their distance from the middle pixel by using Euclidean distance that is  $d_{ij}$ . It is important to note that the factor  $G_{ki}$  reflects the damping extent of the neighbors with the spatial distances from the central pixel [49]. As compared to other parameters ( $\alpha$  or  $\lambda$ ) presented in [49], [46] the factor  $G_{ki}$  can vary adaptively with diverse spatial locations or distances from the middle pixel [48]. This factor is used in the objective function of the enhanced FCM algorithm as:

$$J_m = \sum_{i=1}^N \sum_{k=1}^c [\mathbf{u}_{ki}^m \|\mathbf{x}_i - \mathbf{v}_k\|^2 + G_{ki}] \quad (\text{II-18})$$

The two necessary conditions for  $J_m$  to be at its local minimal extreme, with respect to  $\mathbf{u}_{ki}$  and  $\mathbf{v}_k$  is obtained as follows:

$$\mathbf{u}_{ki} = \frac{1}{\sum_{j=1}^c \left( \frac{\|\mathbf{x}_i - \mathbf{v}_k\|^2 + G_{ki}}{\|\mathbf{x}_i - \mathbf{v}_j\|^2 + G_{ji}} \right)^{\frac{1}{m-1}}} \quad (\text{II-19})$$

$$\mathbf{v}_k = \frac{\sum_{i=1}^N \mathbf{u}_{ki}^m \mathbf{x}_i}{\sum_{i=1}^N \mathbf{u}_{ki}^m} \quad (\text{II-20})$$

**The FLICM algorithm is given as follows:**

- 
1. Set the number  $c$  of the cluster prototypes, fuzzification parameter  $m$  and the stopping condition  $\xi$ .
  2. Initialize randomly the fuzzy partition matrix.
  3. Set the loop counter  $b = 0$ .
  4. Calculate the cluster prototypes using (eq.II-18).
  5. Compute membership values using (eq.II-17).
- If  $\max\{U^{(b)} - U^{(b+1)}\} < \xi$  then stop, otherwise, set  $b = b + 1$  and go to step 4.
- 

When the algorithm has converged, a defuzzification process takes place in order to convert the fuzzy partition matrix  $U$  to a crisp partition. The maximum membership procedure is the most important method that has been developed to defuzzify the partition matrix  $U$ . This procedure assigns the pixel  $i$  to the class  $C$  with the highest membership:

$$C_i = \arg_k \{ \max\{u_{ki}\} \}, k = 1, 2, \dots, c. \quad (\text{II.21})$$

It is used to convert the fuzzy image achieved by the proposed algorithm to the crisp segmented image.

The measure used in the FLICM objective function (II.18) is still the Euclidean metric as in FCM, which is computationally simple.

### II.7.5 Spatial FCM

Chuang et al [50] proposed a spatial FCM algorithm, in which spatial information can be integrated into the fuzzy membership function. The spatial function is defined for each pixel by the sum of the membership functions of its neighbors which are defined by a square window centered on the given pixel, according to the following formula

$$h_{ij} = \sum_{k \in NB(x_i)} u_{ik} \quad (\text{II.22})$$

Where  $NB$  (represents a square window centered on the pixel in the spatial domain. A 5x5 window was used in this work. Like the membership function, the spatial function represents the probability that the pixel belongs to its group. The function spatial of one pixel for one the cluster is large if the majority of its neighborhood belongs to the same class. The spatial function is integrated with the membership function as follows

$$\mathbf{u}'_{ij} = \frac{\mathbf{u}_{ij}^p h_{ij}^q}{\sum_{k=1}^c \mathbf{u}_{kj}^p h_{kj}^q} \quad (\text{II.23})$$

where  $p$  and  $q$  are parameters to control the importance of the two functions ( $u_{ij}$  and  $h_{ij}$ ). In a homogeneous region, the spatial functions reinforce the original membership and the clustering result remains unchanged. However, for a noisy pixel, this formula reduces the weighting of a noisy cluster by the labels of its neighboring pixels. As a result, misclassified pixels of noisy regions can easily be corrected.

starting from the idea that the FCM algorithm is very sensitive to noise, Pham [51] proposed the following objective function

$$J_{RFCM} = J_{FCM} + \sum_{j \in \Omega} \sum_{k=1}^c \mathbf{u}_{jk}^q \sum_{l \in N_j} \sum_{m \in M_k} \mathbf{u}_{lm}^q \quad (\text{II.24})$$

$$J_{FCM} = \frac{\beta}{2} = \sum_{j \in \Omega} \sum_{k=1}^c \mathbf{u}_{jk}^q \|\mathbf{y}_j - \mathbf{v}_k\|^2 \quad (\text{II.25})$$

Where  $\beta$  is a parameter that weights the influence of the second term of the objective function.

$N_j$ : represents all the neighboring pixels of pixel  $y_j$

$M_k = \{1 \dots k\} \setminus \{k\}$  represents the set of classes except class  $k$ .

In this algorithm, the membership of a pixel to a class is conditioned by the membership of its neighbors to other classes. In other words, if the membership of neighboring pixels to other classes is low then this increases the membership of pixel  $y_j$  to class  $k$ . This method therefore depends on the values of the other classes and not only on the considered class. As there are no constraints on the centers of the classes, the equation which calculates the  $v_k$  remains identical to that of FCM. On the other hand, the fuzzy partition function  $U$  is different, it is obtained by Lagrangian optimization

$$\mathbf{u}_{jk} = \frac{(\|\mathbf{y}_j - \mathbf{v}_k\|^2 \beta \sum_{l \in N_j} \sum_{m \in M_k} \mathbf{u}_{lm}^q)^{-1/(q-1)}}{\sum_{i=1}^c (\|\mathbf{y}_j - \mathbf{v}_i\|^2 \beta \sum_{l \in N_j} \sum_{m \in M_k} \mathbf{u}_{lm}^q)^{-1/(q-1)}} \quad (\text{II.26})$$

### II.7.6 robust fuzzy c-means with adaptive spatial & intensity constraint and membership linking (FCM\_SICM)[52]

FCM is proven to be a sensitive to noise and therefore it is hard to segment noisy images. The idea is to improved FCM that incorporates local spatial & intensity information and membership linking for noise image segmentation.

The objective function  $j_{FCM\_SICM}^{(a)}$  of this proposed method is firstly given as:

$$j_{FCM\_SICM}^{(a)} = \sum_{i=1}^k \frac{\sum_{j=1}^N \alpha (u_{ij}^{(a)})^m \|y_j - c_i^{(a)}\|^2 + \sum_{j=1}^N \beta (u_{ij}^{(a)})^m \|\bar{y}_j - c_i^{(a)}\|^2}{\ln^2(\sum_{e=1}^N u_{ie}^{a-1} + 1)} \quad (II.27)$$

It is necessary to consider how to select proper  $\alpha$  and  $\beta$ . When noise is heavier, it is necessary to select a smaller  $\alpha$  and a larger  $\beta$ . By doing this, the segmentation result can be better influenced by the bilateral filtered image while original image is preserved to a certain

extent. In practice, different  $\alpha$  and  $\beta$  are needed to be applied for segmenting different images. Also,  $\alpha$  and  $C$  are fixed for all pixels i.e. tradeoffs between pixels in original image and bilateral filtered image are the same. Therefore, the proposed method is not flexible by applying  $\alpha$  and  $\beta$ .

For a noisy image, the more seriously is pixel  $y_j$  corrupted by noise, the larger is the  $|y_j - \bar{y}_j|$  i.e. the intensity difference between the original pixel and the bilateral filtered pixel is larger.

That is, the segmentation result is supposed to be better influenced by the bilateral filtered image (a larger  $\beta$  is needed) and less influenced by the original image (a smaller  $\alpha$  is needed).

Here,  $|y_j - \bar{y}_j|$  and  $\frac{1}{|y_j - \bar{y}_j|}$  are used to measure the extent of influence of bilateral filtered image and the original image on segmentation result respectively:

$$\beta = |y_j - \bar{y}_j| \quad (II.28)$$

$$\alpha = \frac{1}{|y_j - \bar{y}_j|} \quad (II.29)$$

It has been proved in practice that replacing  $\beta$  and  $\alpha$  with Eqs. is not enough for the proposed method to output a satisfied segmentation result. It is discovered that only by increasing  $|y_j - \bar{y}_j|$  can a better quality segmentation result be obtained. After several experiments, it is discovered that timing  $|y_j - \bar{y}_j|$  by 20 is suitable enough. Meanwhile, for the prevention of Inf and zero, a small number  $\epsilon$  is added to Eqs. . Consequently, Eqs. are modified as:

$$\beta = 20 |y_j - \bar{y}_j| + \epsilon \quad (II.30)$$

$$\alpha = \frac{1}{20 |y_j - \bar{y}_j| + \epsilon} \quad (II.31)$$

By applying Eqs. into (Eq. 0-27), the objective function  $J_{\text{FCM\_SICM}}^{(a)}$  is modified as (Eq. 0-33). Considering that  $|y_j - \bar{y}_j|$  is related with only pixels in original image and bilateral filtered image and does not change during iterations,  $|y_j - \bar{y}_j|$  can be computed only once before iterations.

$$\mathbf{Let} \quad \Delta I = 20 |I - \bar{I}| + \text{eps} \quad (\text{II.32})$$

where  $I$  is the original image,  $\bar{I}$  is the bilateral filtered image.  $\Delta I$  is the difference between  $I$  and  $\bar{I}$ . The objective function  $J_{\text{FCM\_SICM}}^{(a)}$  is finally given as:

$$J_{\text{FCM\_SICM}}^{(a)} = \sum_{i=1}^k \frac{\sum_{j=1}^N \frac{1}{20|y_j - \bar{y}_j| + \text{eps}} (u_{ij}^{(a)})^m \|y_j - c_i^{(a)}\|^2 + \sum_{j=1}^N (20|y_j - \bar{y}_j| + \text{eps}) (u_{ij}^{(a)})^m \|\bar{y}_j - c_i^{(a)}\|^2}{\ln^2(\sum_{e=1}^N u_{ie}^{(a-1)} + 1)} \quad (\text{II.33})$$

$$\sum_{i=1}^k \mathbf{u}_{ij}^{(a)} = \mathbf{1}$$

where  $\Delta y_j$  is the  $j^{\text{th}}$  value of  $\Delta I$ .

$u_{ij}^{(a)}$  and  $c_i^{(a)}$  are defined by taking the partial derivative of Eq. (II.33) with respect to membership degree  $u_{ij}^{(a)}$ , and cluster center  $c_i^{(a)}$ , respectively

$$\Rightarrow \mathbf{u}_{ij}^{(a)} = \frac{1}{\sum_{r=1}^k \left[ \frac{\frac{1}{\Delta y_j} \|y_j - c_i^{(a)}\|^2 + \Delta y_j \|\bar{y}_j - c_i^{(a)}\|^2}{\ln^2(\sum_{e=1}^N u_{ie}^{(a-1)} + 1)} \right]^{\frac{1}{m-1}}} \quad (\text{II.34})$$

$$\Rightarrow \mathbf{c}_i^{(a)} = \frac{\sum_{j=1}^N \left( \frac{1}{\Delta y_j} y_j + \Delta y_j \bar{y}_j \right) (u_{ij}^{(a)})^m}{\sum_{j=1}^N \left( \frac{1 + (\Delta y_j)^2}{\Delta y_j} \right) (u_{ij}^{(a)})^m} \quad (\text{II.35})$$

By updating  $u_{ij}^{(a)}$  and  $c_i^{(a)}$  using Eqs. respectively, the objective function can be minimized. Then, the proposed method outputs optimal membership degrees  $u_{ij}^{(a)}$ .

Finally, a labeling stage is required by attributing each pixel to the cluster whose membership degree is the highest.

**the FCM\_SICM algorithm is given as follows:**

Input: Set number of clusters  $K$ , weighting exponent on membership

degree  $m$ , minimum error  $\varepsilon$ , small number  $eps = 0.000001$ ,

geometric spread  $\sigma_d$  and photometric spread  $\sigma_k$ .

1. Normalize the original image into  $[0, 1]$ .
2. Filter the original image using fast bilateral filter proposed in [52] with  $\sigma_d$  and  $\sigma_r$  specified in manual.
3. Find the difference image  $\Delta I$  between original image and bilateral filtered image.
4. Set center vectors  $\mathbf{c}_i^{(1)}$  to 0.
5. Randomly initialize all membership degrees  $\mathbf{u}_{ij}^{(1)}$
6. Set  $\mathbf{u}_{ij}^{(0)} = \mathbf{u}_{ij}^{(1)}$  to compute membership linking in the first iteration.
7. Set loop counter  $a = 1$ .
8. Repeat
  - a. Update center vectors  $\mathbf{c}_i^{(a)}$  using Eq. (II.35).
  - b. Update membership degrees  $\mathbf{u}_{ij}^{(a)}$  using equation Eq. (II.34).
  - c. Update objective function  $J_{FCM\_SICM}^{(a)}$  using Eq. (II.33).
  - d.  $a = a + 1$ .
9. Until  $\|J_{FCM\_SICM}^{(a)} - J_{FCM\_SICM}^{(a-1)}\| < \varepsilon, a > 1$
10. Return Membership degrees  $\mathbf{u}_{ij}^{(a)}$
11. Apply each pixel into the cluster whose membership degree is the highest.

## II.8. Conclusion

In this chapter, we presented a method for unsupervised classification images.

The fuzzy c-means (FCM) algorithm is close to the kmeans algorithm but has the advantage of lower initialization sensitivity. Where one introduces the notion of fuzzy set in the definition of classes. In the next chapter we will apply the FCM algorithm and some its variants to segmented images and we will evaluate the results of this technique.



## Chapitre III. Design

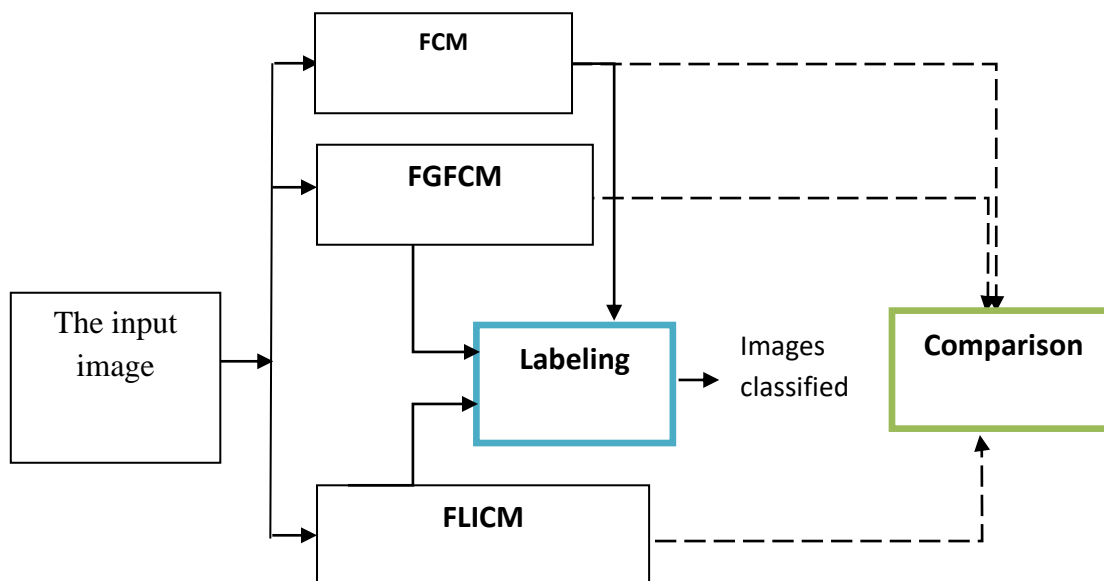
### III.1. Introduction

Design is a process to formalize the initial phases of software development to meet all needs. The launch is always an informal statement (desired result) and requires deep research of information from experienced and functional users (future users of the program) with good and logical analysis.

Through what we have examined in the previous two chapters, we have been able to understand and define the basics of starting our project, and work to establish and develop our program clearly and unambiguously, with the introduction of the general and specific structure of our program, from there we will explain and develop this whole structure.

### III.2. General design

In this section, we examine the general structure of the system, which consists of inputs of a noisy image and the algorithms used in the classification, then we move on to the process of labeling as well as the transition comparison between the results of the algorithms. . (Figure III-1).



**Figure III.1:** Overall System Architecture

Initially we choose the image we want to classify, then we apply one of the following three algorithms (standard FCM, FGFCM, FLICM). After the convergence of the classification process with the chosen algorithm, labeling step will take place to get the classified image. The comparison step is a semi-automatic step, in which we use some fuzzy evaluation functions on the centers and the membership degree matrix obtained by each algorithm; through these functions the best partition obtained will be chosen.

### III.2.1 Input picture

The input image of our application is a digital noisy image chosen by the user in JPEG, PNG, JPG format, ... this image can be in color or at gray level.

### III.2.2 Classification methods

The methods used to classify the image in our work they have fuzzy classification methods based on the algorithms (standard FCM, FGFCM, FLICM), the two algorithms integrate the spatial information under the classical classification process standard FCM. The classification process of each algorithm is an iterative process based on two essential steps: updating the membership degree matrix and updating the class centers.

### III.2.3 labeling

When we wait for the maximum number of iterations of the algorithms used, we must go through a labeling step, then we will value the different solutions of the last generation according to their fitness and the best solution found must label, under this step we assign each pixel to its corresponding class according to its maximum membership degree.

### III.2.4 Comparison methods

In this step, we compare the algorithms in terms of the best fuzzy partition obtained.

There are several functions (indexes) for evaluating fuzzy partitions, the indices used in this section are Partition entropy ( $Pe V_{PE}$ ) [53], Partition coefficient ( $Pc V_{PC}$ ) [53], and *Xie Beni* ( $V_{XB}$ ) [54], ( $F$ ) [55] proposed by Liu and Yang, ( $\hat{F}$ ) [56] proposed by Borsotti et Al, ( $Q$ ) [56] refined from  $F(I)$  by Borsotti et Al, ( $H_j$ ) [57] the *layout entropy* and the time of execution.

**Partition Entropy (Pe  $V_{PE}$ )**, it works at the expense of a total sum of product between the membership function and its logarithm, according to the following formula

$$V_{PE} = -\frac{1}{N} \sum_{i=1}^C \sum_{j=1}^n u_{ij} \log u_{ij} \quad (\text{III-1})$$

The best partition is obtained when  $V_{PE}$  is minimal.

**Partition Coefficient ( $V_{PC}$ )**, used to evaluate the cluster quality. it works at the sum of the total square of the membership function, as the following formula:

$$V_{PC} = \frac{1}{N} \sum_{i=1}^C \sum_{j=1}^n u_{ij}^2 \quad (\text{III-2})$$

This benchmark is used to measure the fuzziness of a clustering result and the  $V_{PC}$  value can take on any value ranges from 0 to 1. The best partition is obtained when  $V_{PC}$  is maximum.

**Xie Beni ( $V_{XB}$ )**, the  $V_{XB}$  should decrease monotonically when the cluster number is close to the number of pixels in the image and furthermore, a better clustering result should produce smaller  $V_{XB}$  value. the  $V_{XB}$  index uses two pieces of information, the intra-variation and the inter-variation (the separation), the intra-variation measures the variation within each class, while the inter-variation measures the separation between the different centers, the  $V_{XB}$  index is defined by the following formula:

$$V_{XB} = \frac{\sum_{i=1}^C \sum_{j=2}^n u_{ij}^m \|x_j - v_i\|^2}{N \times \min_{i \neq k} \|x_j - v_i\|^2} \quad (\text{III-3})$$

The best partition is obtained when the minimum intra-variation and maximum inter-variation, and then  $V_{XB}$  is minimal.

### **$F(I)$ proposed by Liu and Yang**

The squared color error of region  $j$  is defined as:

$$e_j^2 = \sum_{x \in \{r, g, b\}} \sum_{p \in R_j} (C_x(p) - \widehat{C}_x(R_j))^2 \quad (\text{III-4})$$

$R_j$ : denote the set of pixels in region  $j$

$C_x(p)$ : to denote the value of component  $x$  for pixel  $p$

$x$ :  $x$  might be the red, green, or blue intensity value

$\widehat{C}_x(R_j)$ : define the average value of component  $x$  in region  $j$

$$F(I) = \sqrt{N} \sum_{j=1}^N \frac{e_j^2}{\sqrt{s_j}} \quad (\text{III-5})$$

Observe that for any segmentation  $I$  in which the color error is zero for all segments (there is no variance in color of the each region), the value of  $F(I) = 0$  and hence a

segmentation in which each pixel is in its own region will minimize the value of  $F$ . Suppose we have a complex image in which all cannot be zero, except for a segmentation in which each pixel is its own region. Still  $F$  has two strong biases: segmentations with lots of regions are heavily penalized by  $\sqrt{N}$ , and segmentations that have regions with large areas are heavily penalized unless the large region is very uniform in color, since the total error (not average error), is used and only divided by the square root of the area of the region (versus being divided by  $S_j$ , which would give the average squared error).

### $\tilde{F}$ proposed by Borsotti and Al, and $Q$

refined from  $F(I)$  by Borsotti et al.

for Liu and Yang's method the presence of many regions in the segmented image is penalized only by the global measure  $\sqrt{N}$ .

When the number of regions in the image is large enough that the total color is close to zero, the  $F$  rating measure will be favorable even though the image may be over-segmented.

So Borsotti et al proposed the following evaluation function,  $F'$ , to improve Liu and Yang's method:

$$\tilde{F}(I) = \frac{1}{1000 * S_I} \sqrt{\sum_{a=1}^{Max Area} [N(a)]^{1+\frac{1}{a}} \sum_{j=1}^N \frac{e_j^2}{\sqrt{S_j}}} \quad (III-6)$$

note that if the segmentation has many regions consisting of a single pixel, the multiplicative factor preceding the summation will be  $0 (N^2)/S_I$  which is a much larger penalty than the  $\sqrt{N}$  that occurs in  $F$ .

$F'$  will correctly evaluate segmentations with many regions as very poor (unless all color errors are 0) while  $F$  will misclassify them as good segmentations.

Both  $F'$  and  $F$  reach their minimum value of zero on an image in which each region is its own pixel. This is not a big deal as one should never consider allowing the number of regions in the segmentation to be as large as the surface of the image. However, a more serious problem is that  $F$  and  $F'$  strongly penalize segmentations with a large number of regions and only when the squared error in all regions becomes very small (which usually will not occur on natural scenes until the number of regions approaches the image area) a

segmentation with more than a few regions being rated as the best. Thus, Borsotti et al. further improved over F and F', and proposed the evaluation function Q:

$$Q(I) = \frac{1}{1000.S_I} \sqrt{N} \sum_{j=1}^N \left[ \frac{e_j^2}{1+\log S_j} + \left( \frac{N(S_j)}{S_j} \right)^2 \right] \quad (\text{III-7})$$

that all of the three evaluation functions have favored segmentation approach by giving relatively small values.

*the layout entropy  $H_l(I)$* , To encode the segmentation representation itself, each pixel must be placed in one of the regions. Whereas the expected region entropy generally decreases with the number of regions, we expect the number of bits for specifying a region for each pixel, a measure we call the layout entropy, to increase with the number of regions. Hence the two factors can be used to counteract the effects of over-segmenting or under-segmenting when evaluating the effectiveness of a given segmentation. We define the layout entropy as:

$$H_l(I) = - \sum_{j=1}^N \left( \frac{S_j}{S_I} \right) \log \left( \frac{S_j}{S_I} \right) \quad (\text{III-8})$$

$S_I$  be the area (as measured by the number of pixels)

$S_j = |R_j|$  to denote the area of region j

$p_j = S_j / S_I$  as the probability that each pixel in the image belongs to region j under a probabilistic assumption that each pixel is independently selected to be in region j with probability  $p_j$ .

*the layout entropy* favored segmentation approach by giving relatively small values.

### III.3. The detailed design

In the detailed design, we get a description of the data structures of the algorithms and their connections.

Our application accepts as an input a digital image, After the reading, the user can classify his image by the classification methods (Standard FCM, FGFCM, FLICM,) then the resulting image will save in a specified folder.

In this system we used the image as a data structure and the latter is a matrix of pixels identified by their coordinates (height \* width , num-Bands).

- The algorithm (Standard FCM).

- The algorithm (FGFCM), this algorithm as we learned in the second chapter integrates spatial information by calculating a new image.

- The algorithm (FLICM), this algorithm, as we learned in the second chapter, the FLICM algorithm integrates spatial information in the process of fuzzy classification by the combination of the degrees of global membership and local using , In order to overcome the above mentioned disadvantages a new factor in FCM objective  $G_{ki}$  (See Eq.II-17).

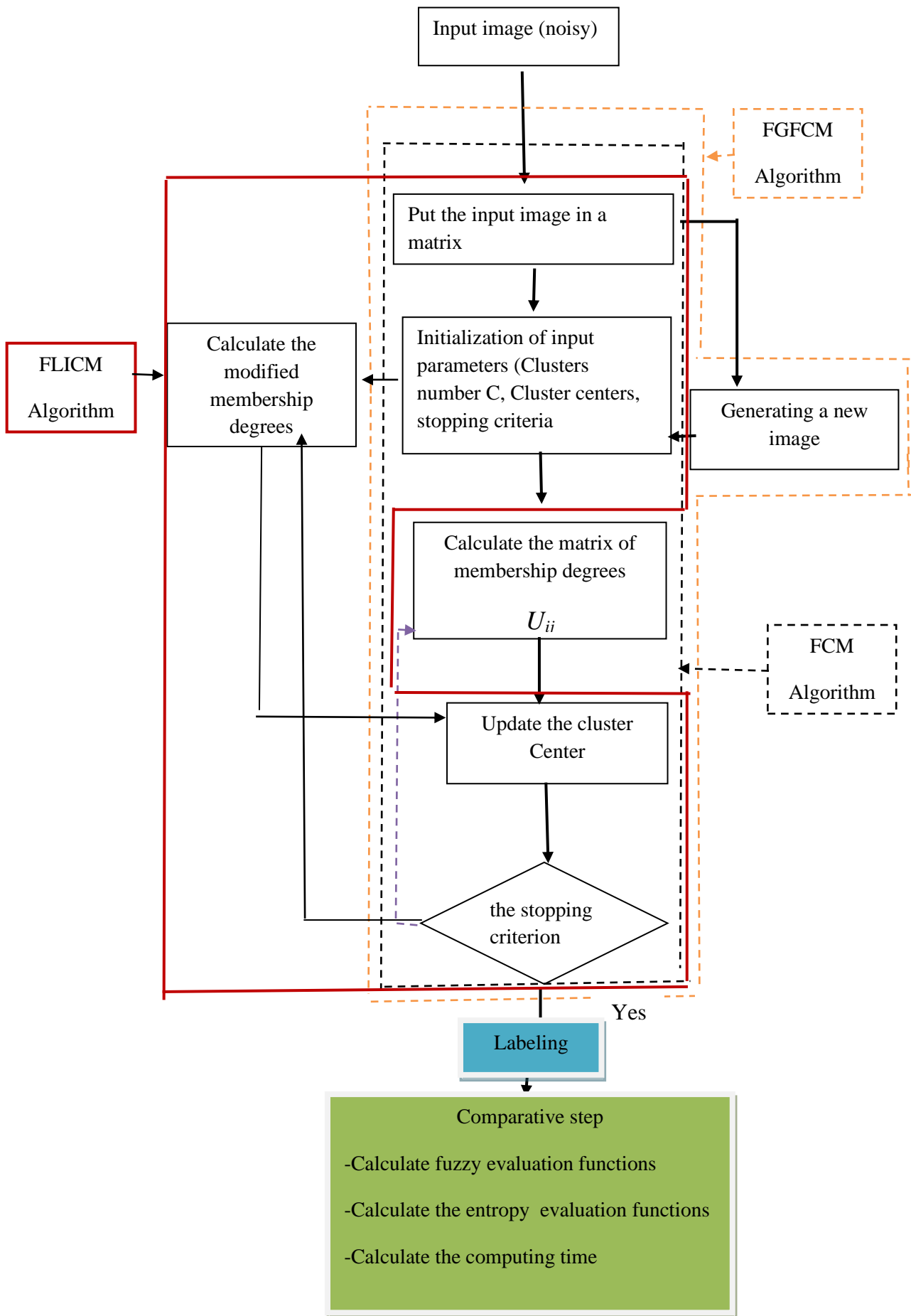


Figure III.2: Detailed system architecture

### III.3.1 Saving the input image

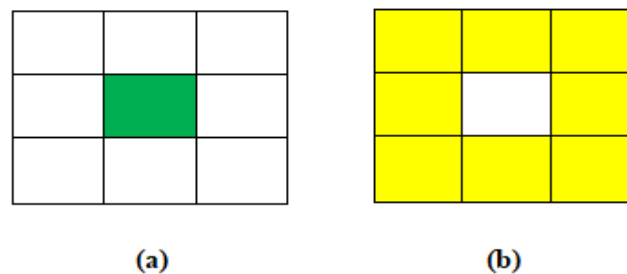
This is an important step after selecting the image to classify; we convert it into a digital image to be used in the work of each algorithm, each number being a pixel and each pixel a color. If the image is colored, it varies from 0 to 255 between red, green and blue. But if it is in the gray level, it goes from 0 to 255 between black and white.

Keep the image in a 2D matrix  $mat [nb\_ligne][ numBands]$  such that  $num\_ligne$  is the total number of pixels in the input image ( $nb\_ligne=height * width$ ) and  $numBands$  corresponds to the number of channels on the input image input, it is equal to three for color images and one for grayscale images.

### III.3.2 Generation of a new image

In the algorithm (FGFCM), we adopted the strategy of generating a new image to exploit the spatial information in the classification process to improve the robustness of the standard FCM against the effect of noise.

The term of similarity  $S_{ij}$  expresses the degree of contribution of each point of the neighbors  $j$  on the definition of the intensity (color) of the central point  $i$ , it corresponds to the sum of the distances of the degrees of belonging to the various classes between the point  $i$  and its neighbors  $j$ ; the similarity measure defined in (eq. II-14)



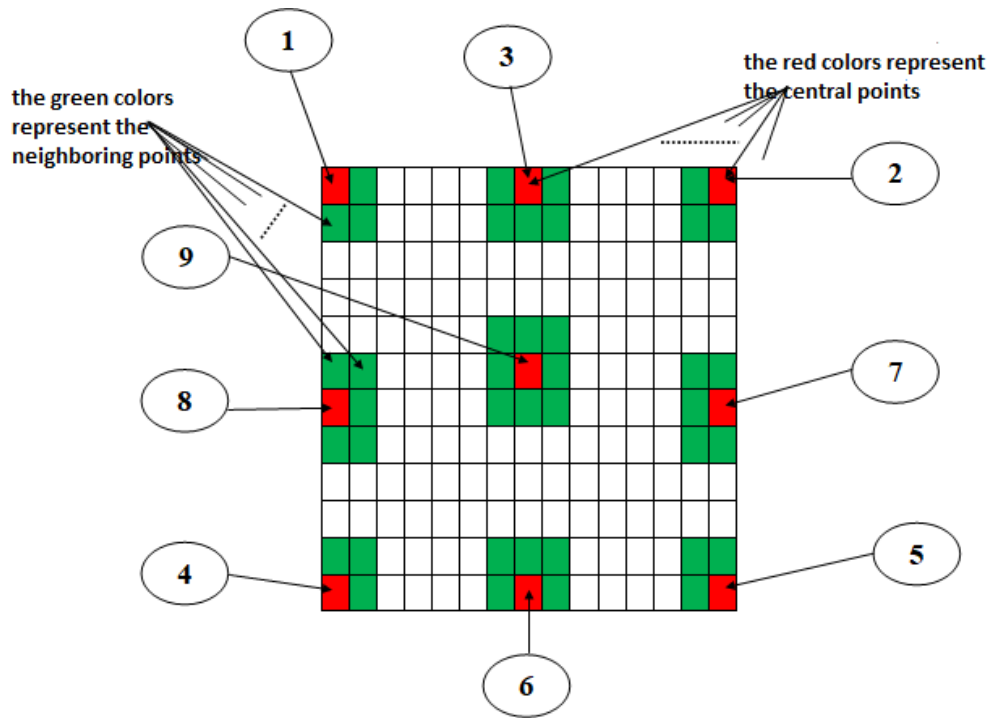
**Figure III.3:** (a) the central point, (b) the neighbors of the central point

After calculating the similarity between each central point and its neighbors we can now generate the new image using the formula in (eq. II-16)

To calculate the levels of influence of the neighbors for each central point we used the term of similarity defined in the previous section, and this part requires the study of all the



possible cases of position of the neighbors, for our work we define the neighbors of a central pixel in a window of size 3\*3



**Figure III.4:**The possible cases of the positions of central pixel (red) and its neighbors (green) in a window of size 3\*3.

For the data structure used to store the new image generated for FGFCM algorithm and the FLICM algorithm, we used a 2D matrix, `CalculeNewImageR [nb_ligne][ nb_col]`, `CalculeNouvImgEn [nb_ligne][ nb_col]`, such that `nb_ligne` is the number total pixels of the input image (`nb_ligne=height * width`) and `numBands` corresponds to the number of bands on the input image, it is equal to three for color images and one for grayscale images .

New Image Generation Algorithm

The entrees

- Image I, nb\_column, nb\_line

The exits

- the new NI image

for each pixel i of image I do

//case 1

If (i=1) then

Calculate the similarity  $S_{ij}$  between the central points i and their neighbor j by (eq. II-14)

Calculate the new pixel  $y_i$  by (eq. II-12)

Else

//case 2

If (i=nb\_column) then

Calculate the similarity  $S_{ij}$  between the central points i and their neighbor j by (eq. II-14)

Calculate the new pixel  $y_i$  by (eq. II-12)

Else

//case 3

If (i<nb\_column) then

Calculate the similarity  $S_{ij}$  between the central points i and their neighbor j by the (eq. II-14)

Calculate the new pixel  $y_i$  by (eq. II-12)

Else

//case 4

If (i=nb\_line\*nb\_column) then

Calculate the similarity  $S_{ij}$  between the central points i and their neighbor j by the (eq. II-14)

Calculate the new pixel  $y_i$  by (eq. II-12)

Else

//case 5

If (i=nb\_line\*nb\_column) then

Calculate the similarity  $S_{ij}$  between the central points i and their neighbor j by the equation (eq. II-14)

```
Calculate the new pixel  $y_i$  by (eq. II-12)

Else

//case 6

If  $(i > (nb\_line-1)*nb\_col \ \&\& \ i < nb\_line*nb\_col-1)$  then

Calculate the similarity  $S_{ij}$  between the central points  $i$  and their neighbor  $j$  by the (eq. II-14)

Calculate the new pixel  $y_i$  by (eq. II-12)

Else

//case 7

If  $(i \% nb\_column = nb\_column-1)$  then

Calculate the similarity  $S_{ij}$  between the central points  $i$  and their neighbor  $j$  by the (eq. II.14)

Calculate the new pixel  $y_i$  by (eq. II-12)

Else

//case 8

If  $(i \% nb\_column = 0)$  then

Calculate the similarity  $S_{ij}$  between the central points  $i$  and their neighbor  $j$  by the (eq. II-14)

Calculate the new pixel  $y_i$  by (eq. II-12)

//case 9

Else

Calculate the similarity  $S_{ij}$  between the central points  $i$  and their neighbor  $j$  by the (eq. II-4)

Calculate the new pixel  $y_i$  by (eq. II-12)
```

### III.3.3 Parameter and input initialization

The fuzzy classification requires a step of specifying the number of classes  $C$  as well as the values of the different centers and the threshold of the stopping criterion.

The step of specifying the number of classes leaves it to us for user intervention.

To initialize the cluster centers, we start with the first random estimate of the cluster centers, this is the average location of each cluster.

The structure used to keep the centers calculated is a 2D matrix `clusterCenters` [`numClusters`] [`numBands`] whose rows are the numbers of the classes and the columns are the number of the bands.

The input parameters initialization step is a mandatory step for the four algorithms (Standard FCM, FGFCM, and FLICM).

### III.3.4 Calcul des centres et des degrés d'appartenance

The minimization of the objective function  $(U, C)$  (eq.II-1) can be brought about by an iterative process in which the updating of the membership degree  $U_{ij}$  and the class centers are done for each iteration of the classification.

In this step we seek to calculate membership degrees of the three algorithms (Standard FCM, FGFCM and FLICM) using the equation (eq.II-3) the data structure that we used to store the membership degree matrix is a 2D matrix `memberships`[the number of pixels][`numClusters`] such that the number of pixels is equal to the size of the input image (`width*height`) and `numClusters` is the number of classes.

The structure used to keep the centers calculated by the equation (eq.II-2) is a 2D matrix `clusterCenters` [`numClusters`] [`numBands`] whose rows are the numbers of the classes and the columns are the number of the channels.

### III.3.5 The stopping criterion

The stopping criterion used for the four algorithms, either after a specified number of iterations or when the centers of the classes no longer evolve between two successive iterations (i.e. the change of centers between two successive iterations does not exceed a certain threshold).

### III.3.6 Labeling

The objective of this stage is the construction of the final image from the matrix of membership degrees obtained after the convergence of the classification algorithm, and to carry out this stage each color is assigned (the pixels characterized by this color) to its own class (the class with the highest degree of membership)

### III.4. Interface module

Our application accepts a noisy image as input. In order for the user to be able to classify his image by the different FCM methods (Standard FCM, FGFCM FCM, FLICM), To obtain a classified image, click on one of the buttons "Standard FCM" "FGFCM" and "FLICM", wait to finish and display the classified image. if it is a color image, a pixel is encoded by three components (r, g, b) (each broadly between 0 and 255), representing respectively the "doses" of red, green and blue which characterize the color of the pixel if it is a gray level image, it is coded by a component comprised in the broad sense between 0 and 255 representing the luminosity of the pixel. Processing always applies to color images.

#### Algorithm Function\_Objective ();

Variable index,w,c, width , height, numcluster;

```

begin
  if (w < width * height) then
    if (c < numcluster) then
      index = membership degree [w][c] * || Distance between point w and
      class center k ||2;
    end if
  end if
  return index;
end

```

#### Algorithm Function calculateCenters();

Variable w,c, width , height, numbands: integer

bottom ,top :real;

```

begin
  if (w < width * height * numbands) then

```

```

    if (c < width * height * numbands) then
        top = membership degree [w][c] * le pixel j ;
        bottom = membership degree [w][c];
    end if

```

End if

Return (top /bottom);

End

### Algorithm Function CalculateMemberShip ();

Variable w,c, width , height, numbands, numcluster: integer

bottom ,top, memberships:real;

begin

```

    if(w < numcluster) then
        if (c < width * height * numbands) then
            top = Distance between pixel j and center of their class i;
            bottom = Distance between pixel j and the centers of other class k;
            memberships =(|| top|| / ||Botom||)(1/(m-2))
        end if
    End if
    Return(1/ memberships);

```

End

### Function getImageInMatriceNouv (var mat[][] : real);

Var sumterms : de réel ; h , i , b , w,ck : integer;

begin

```

    mat [height*width][ numbands];
    i ←0;
    for (h← 0 to height) do
        for(w ← 0 to width) do
            Index ← (h*width+w)*numBands;
            for(b ← 0 à numbands) do mat[i][b] ← inputData2[index+b];
        end for
        i ← i+1;
    end for

```

```

        |
        | end for
    |
    | end for
    |
    | Return mat;
    |
    | End

```

**Algorithm Function Indice\_Xie\_Beni ( ) ;**

Var w, b, c, c1, c2, index : integer;

DistancePixelToCluster, minDist, distance, cs : real;

**begin**

**for**(w 1 to (width\*height)) do

Index w\*numBands;

**for**(b =1 to numBands) do

aPixel [b]= InputData [index + b];

**end for ;**

**for**(c=1 to numClusters) do

DistancePixelToCluster = calcSquaredDistance (a  
Pixel,clusterCenters[c]);

Cs+=Pow ( membership[w][c]m fuzziness)\*

DistancePixelToCluster\* DistancePixelToCluster;

**end for ;**

**end for;**

cs =cs/ (height\*width) ;

minDist = Float.MAX\_VALUE;

**for**(c1 =1 to (numClusters-1)) do

**for**(c2= c1+1 to numClusters)do

Distance calcSquaredDistance(clusterCenters [c1],clusterCenters[c2]);

MinDist minimum (minDist, distance);

```

    |   |   End for ;
    |   |   End for ;
    |   |   cs= cs/(minDist*minDist);
    |   |   Return (cs);
    |   |
    |   | End

```

**Algorithm Function getPartitionEntropy2D ();**

Var width , height, numcluster, numBands: integer

**begin**

Pe : reall;

Pe =0;

**for**(h = 1 to (height\*width)) do

**for**(c = 1 to numClusters) do

        Pe = Pe + memberships[h][c] \* log (memberships[h][c]);

**End for** ;

**End for** ;

pe = -pe / (height \* width \* numBands);

Return (pe);

**End**

**Algorithm Function getPartitionCoefficient2D ();**

Var width , height, numcluster, numBands: integer

**begin**

Pc : réel ;

Pc =0 ;



```
for (h =1 to (height*width))do
    |
    for (c 1 to numClusters)do
        |
        Pc = Pc + memberships[h][c] * memberships[h][c];
    End for ;
End for ;

pc = pc / (height * width * numBands);

Return (pc);

End
```

### III.5. Conclusion

In this chapter, we have presented the different steps of our classification algorithm (Standard FCM, FGFCM, FLICM), as well as the general and detailed design, explaining each step of the detailed design of each algorithm. The next chapter is devoted to the implementation of our application.

## Chapitre IV. Implementation

### IV.1. Introduction

In this last chapter and after the theoretical overview of the previous chapters, we will present the practical side of the application. Our goal is the realization of a robust and reliable system with the noise and fuzzy contours which make the classification of the images, we start with the description of the base used, the choice of the working environment passing by the language and software of coding as well as results on different images with two tables for the results of the comparative indices between the three algorithms and for the execution time

### IV.2. The work environment

For our work to achieve the objective we were aiming for, we took the initiative to exploit and implement our algorithm on: Windows 10. This choice is reflected in the efficiency of this environment with regard to the structure of event interaction that it has to communicate with active applications, as well as the resources of the machines that it offers to the different applications, finally, its memory allocation system which is one of the best presents in this field.

### IV.3. The coding language

Our application was coded in its entirety by the JAVA language through the NetBeans platform, this choice is based on the fact that it is:

Popular: especially in business, the Java language is an investment for those who learn it.

Rich: there are many libraries in all fields; these have the considerable advantage of being standardized.

International: Java programs support international Unicode standards for character encoding for all languages.

Clean: it preserves a fairly strong typing of the data present; the arithmetic operations are very well specified.

### IV.4. Coding software

Integrated development environments (IDEs) are software comprising a set of tools necessary for software development in one (or more) programming language(s) (Java, PHP, Ruby, C/C++, etc.).

Among all the existing development environments, our choice fell on NetBeans (Figure IV 1) thanks to its speed in setting up an application which is one of the strong points of the latter.

A toolkit that saves a lot of development time and effort.

A set of Standards to enhance and strengthen consistency and interoperability between applications and operating systems.

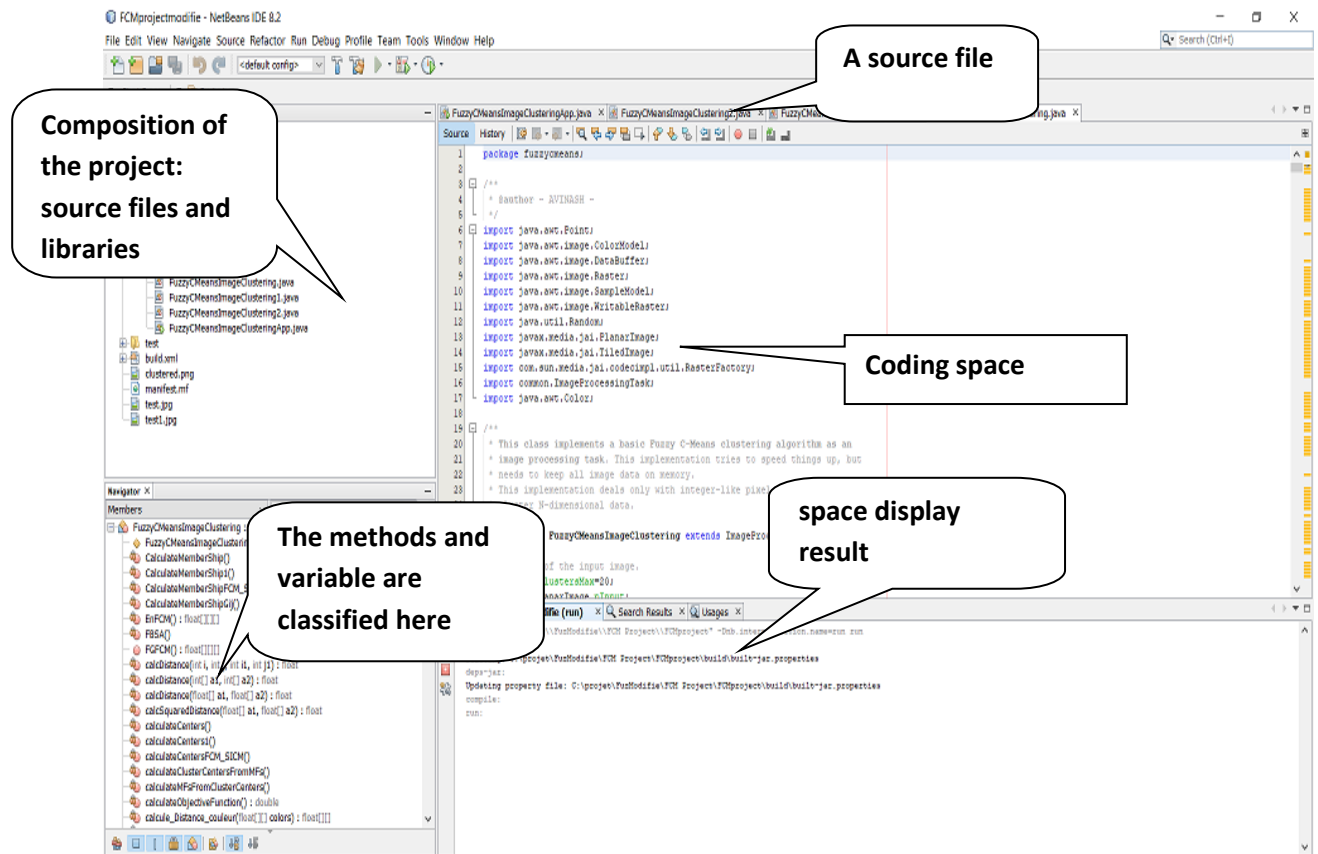


Figure IV.1: The environment interface

#### IV.5. Libraries to use

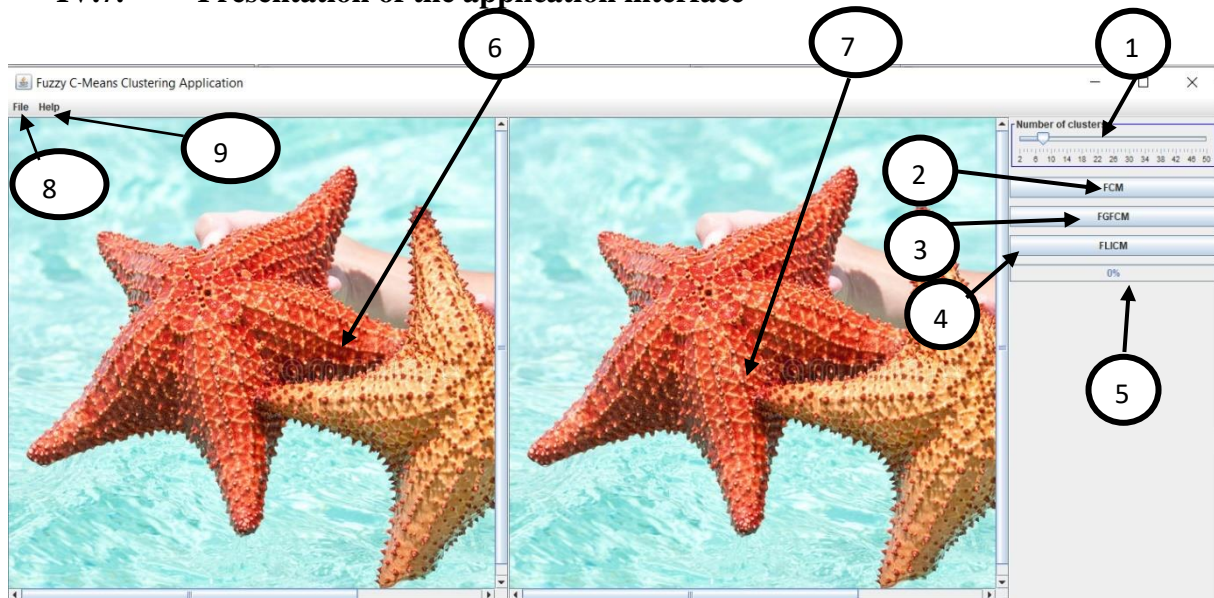
- jackson-core-2.5.3
- jackson-jr-objects-2.5.3
- jai\_core
- java-scanner-access-twain-12.0.9

#### IV.6. Machine Features

The implementation of the project is carried out on a laptop computer which is technically characterized by:

- Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
- Windows 10 pro 64 bit.
- A RAM memory of 8 GB (7.89 GB usable).
- A GeForce GT 630M graphics card version 376.54.

#### IV.7. Presentation of the application interface

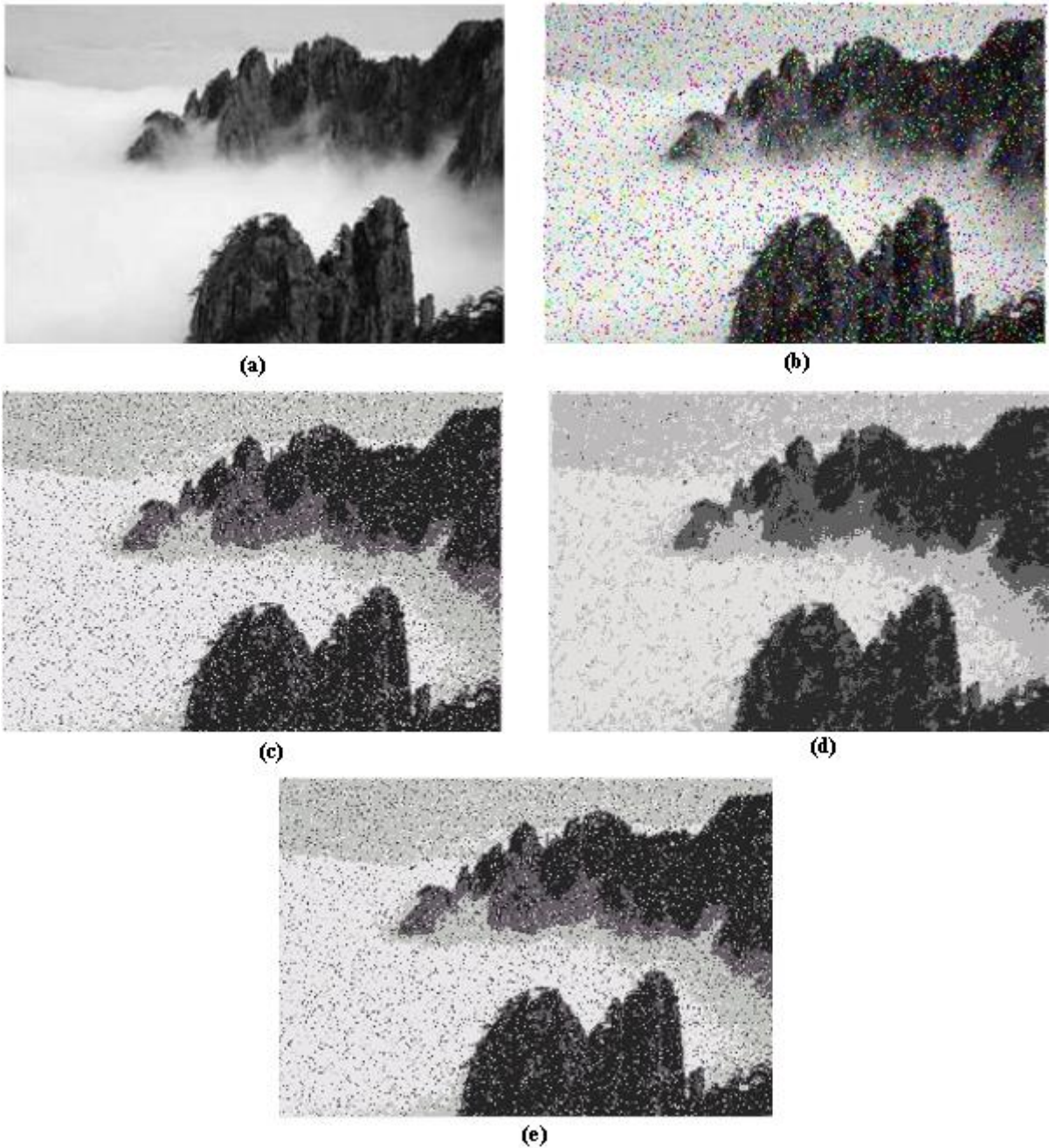


**Figure IV.2:** Home window

- (1) The number of classes, In order to select the number of categories used in the classification process by scroll bar
- (2) To run the FCM algorithm, by means of a button-like compound.
- (3) To run the FGFCM algorithm, by means of a button-like compound.

- (4) To start the FLICM program (Chuang method), by means of a button type compound.
- (5) The loading bar to present the remaining time to run by means of a Progress Bar
- (6) The initial image. Space to display input image which selected by user in a Panel
- (7) The image after result. The space to display the classified image in a Panel
- (8) File: (Open file) to select the input image, and (Exit) to close the program, by means of a Menu Bar type compound
- (9) help: Contains a description of the application

IV.8. Evaluation on segmentation results



**Figure IV.3:** Segmentation results on the mountain corrupted by Salt and Pepper noise. (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.

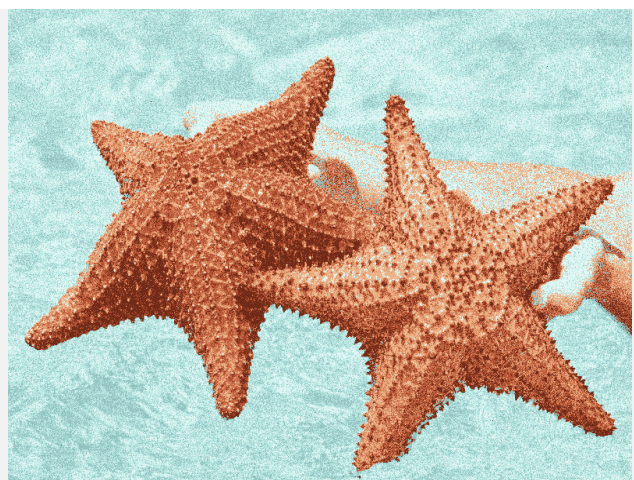




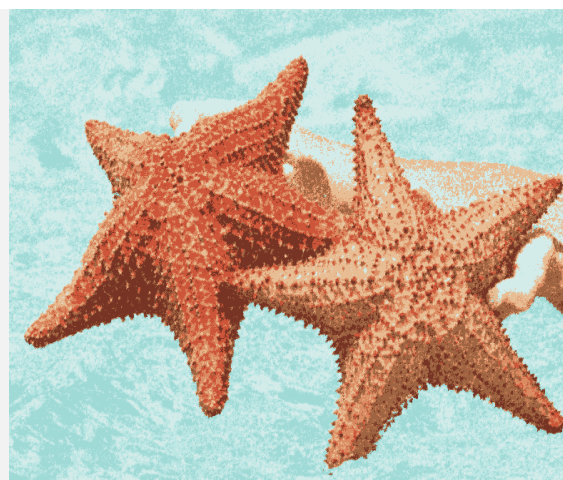
(a)



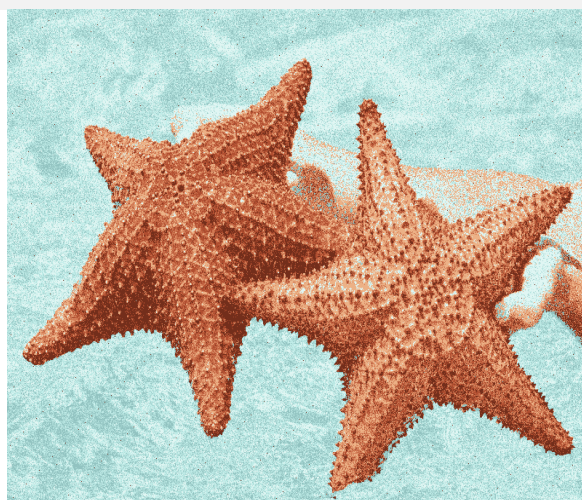
(b)



(c)



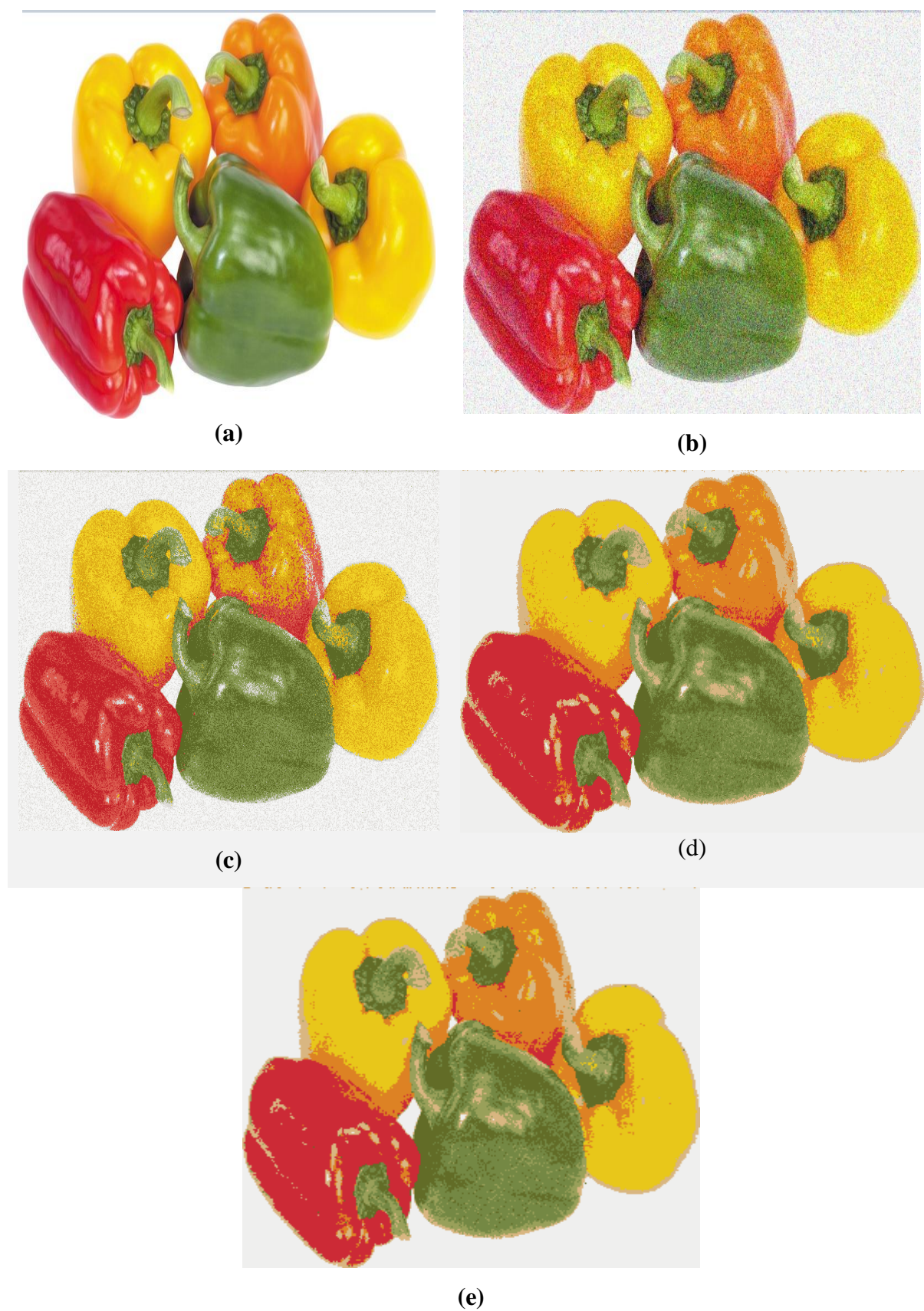
(d)



(e)

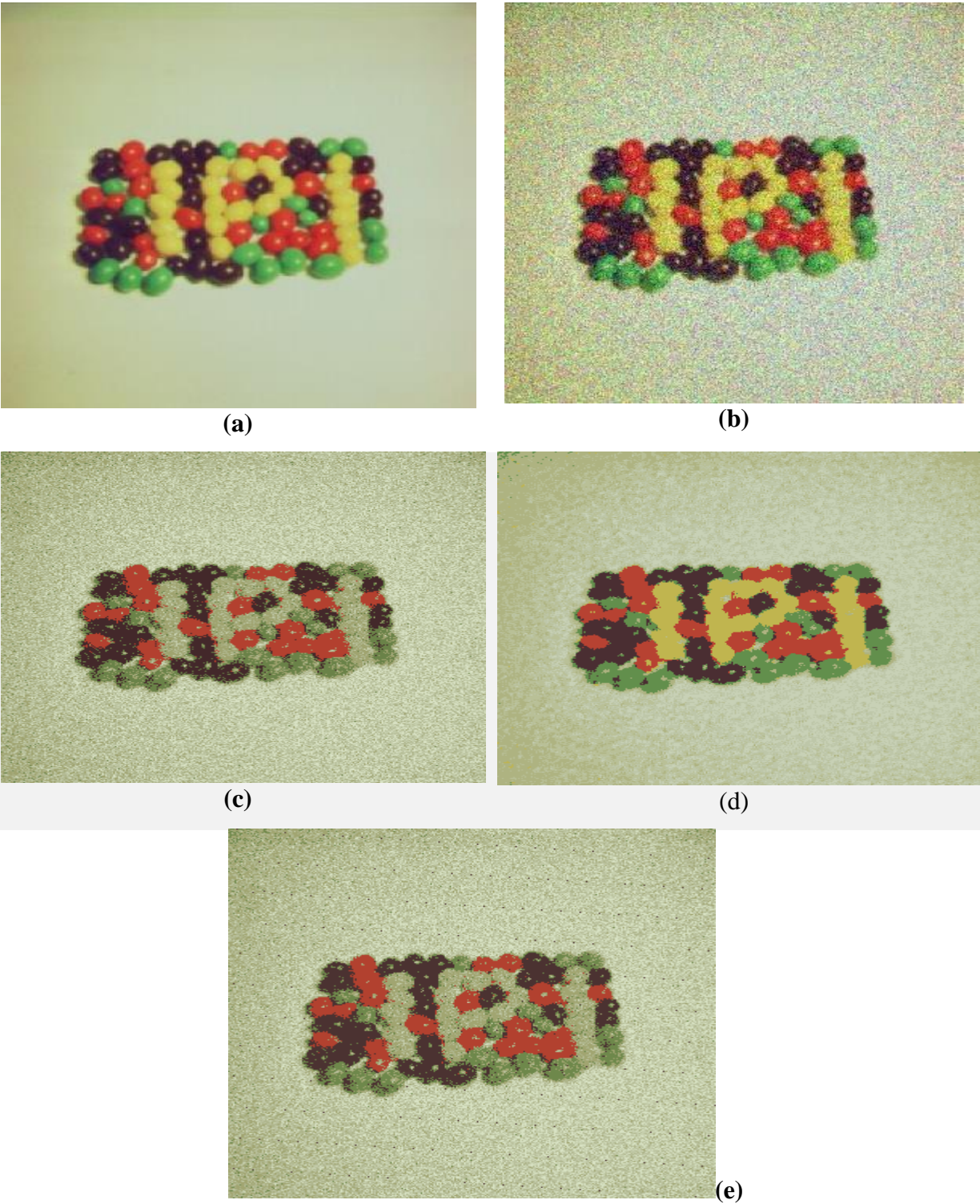
**Figure IV.4:** The image of a Star Fish : : (a) original image, (b) corrupted by Gaussian noise ( $\sigma=30$ ) (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.



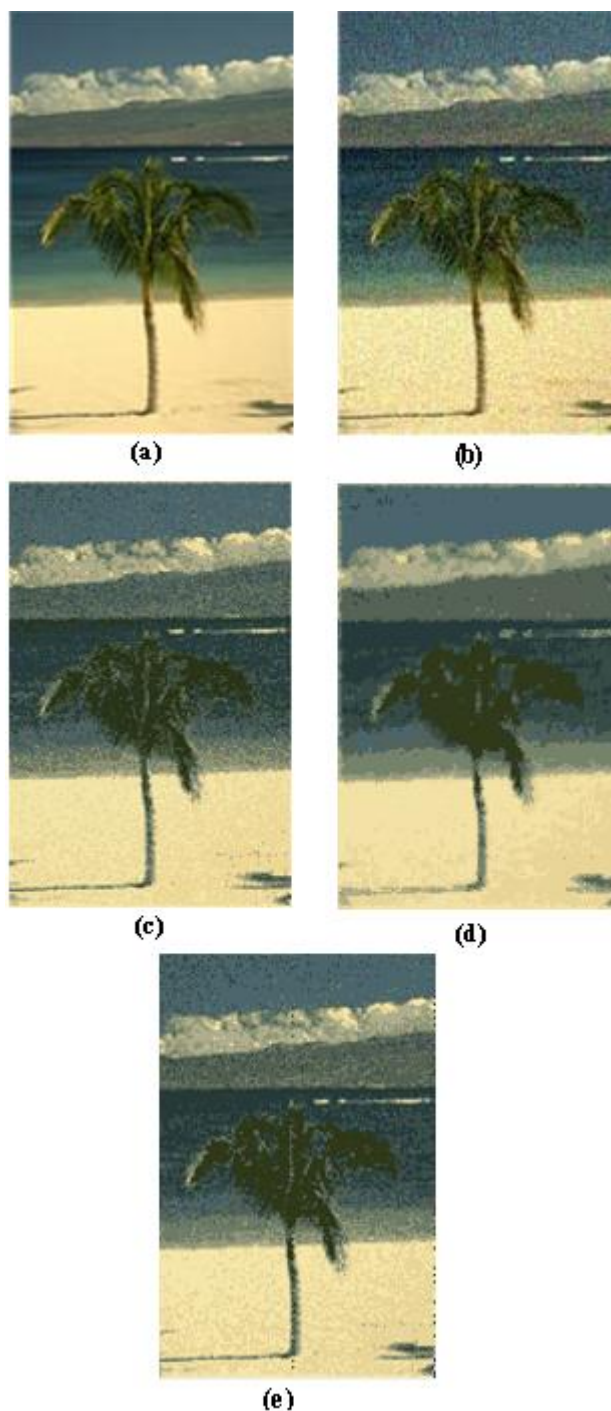


**Figure IV.5:** The image of a Pepper : (a) original image, (b) corrupted by Gaussian noise ( $\sigma=30$ ) (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.





**Figure IV.6 :**The image of Balls: (a) original image, (b) corrupted by Gaussian noise ( $\sigma=15$ ) (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.



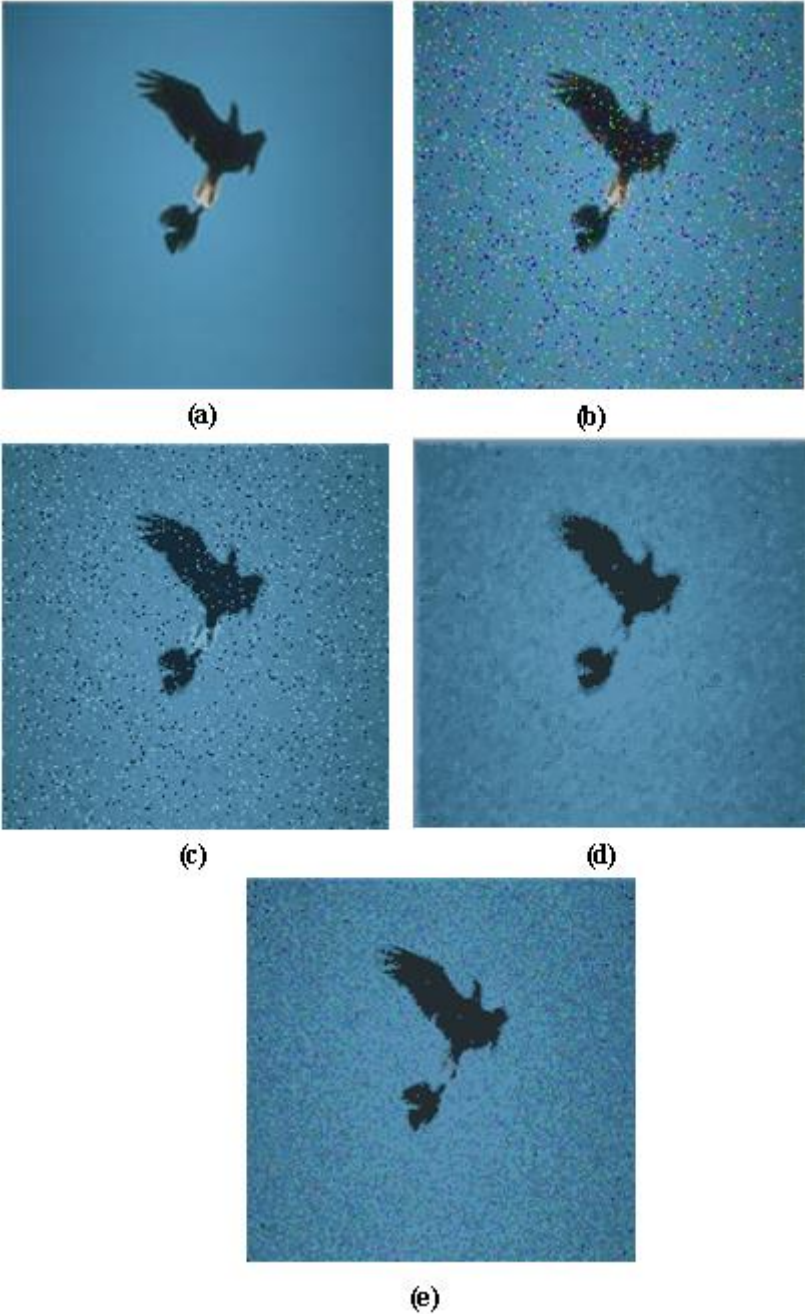
**Figure IV.7:** Segmentation results on the Beach image corrupted by Gaussian noise ( $\sigma=15$ ).

(a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.

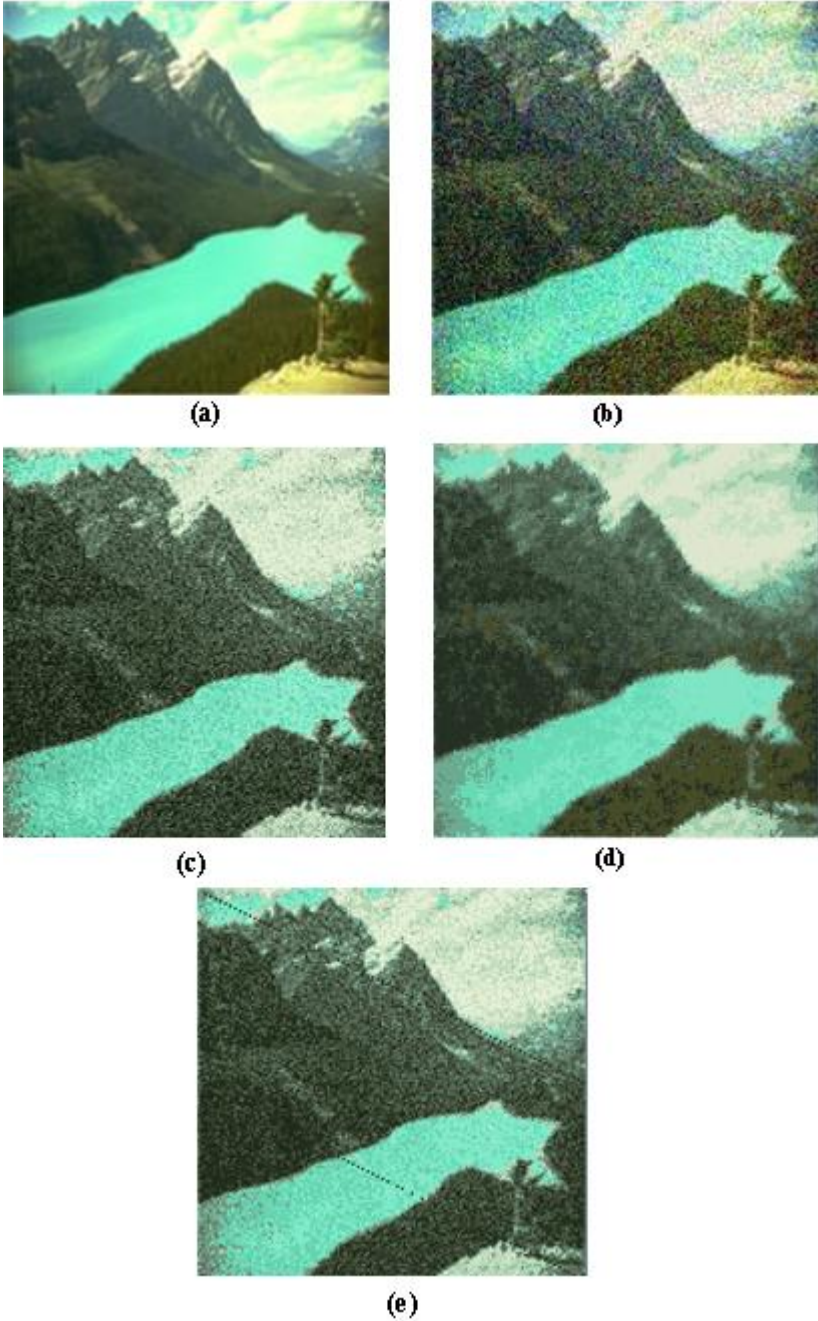


**Figure IV.8:** Segmentation results on the Lena image corrupted by Gaussian noise ( $\sigma=25$ ). (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result.(e) FLICM result.

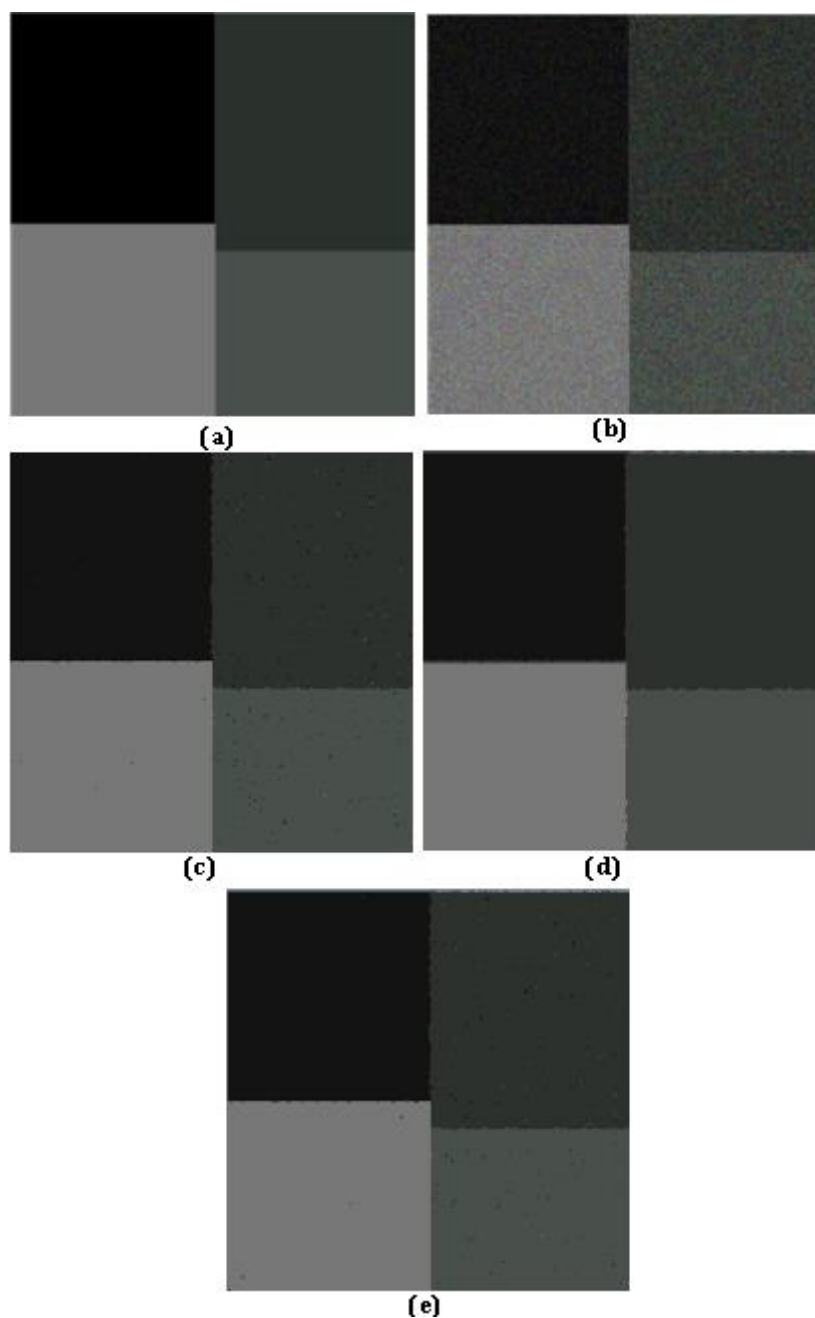




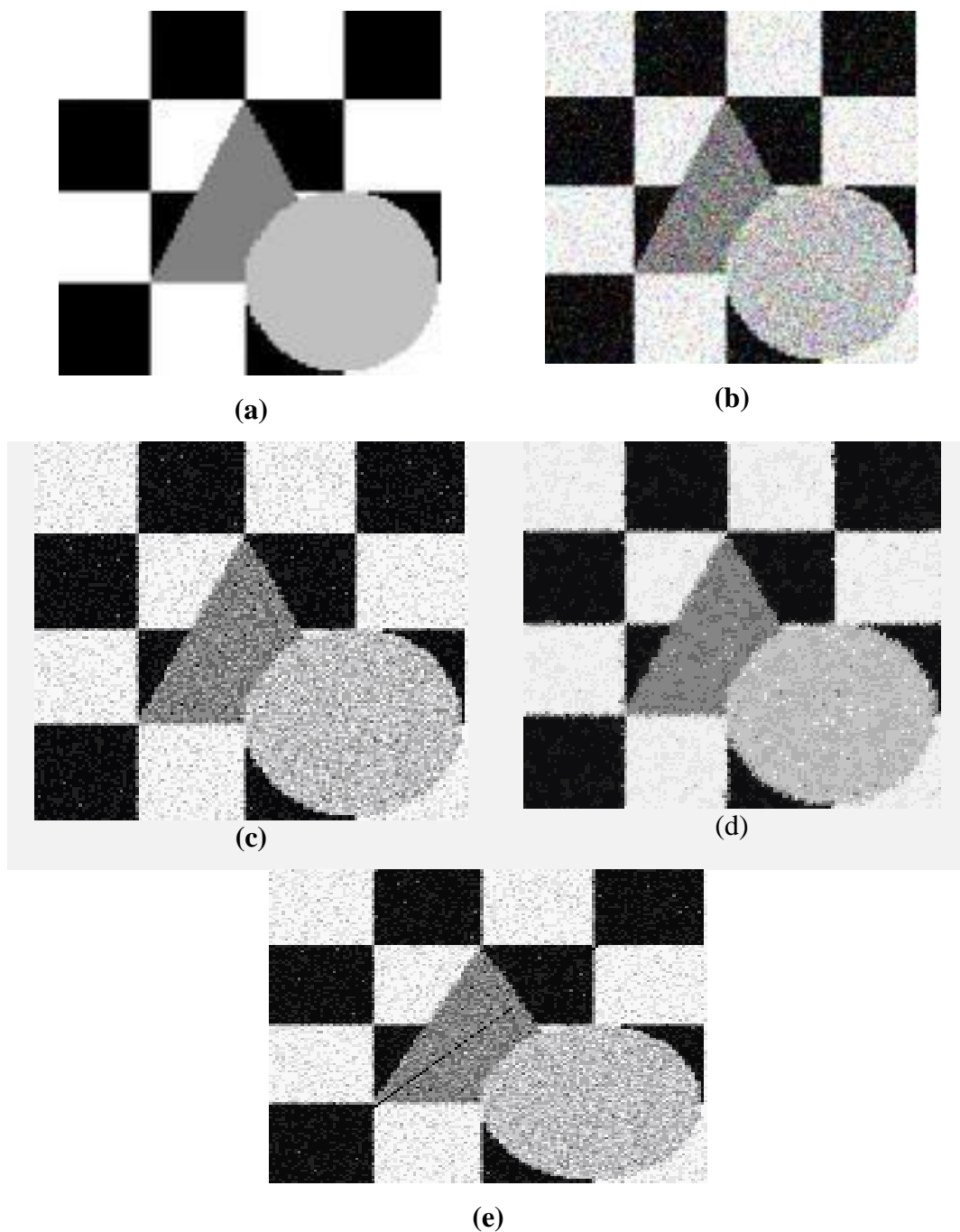
**Figure IV.9:** Segmentation results on the Bird image corrupted by Salt an Pepper noise (30%).(a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.



**Figure IV.10:** Segmentation results on the Lake image corrupted by Gaussian noise ( $\sigma=30$ ). (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.

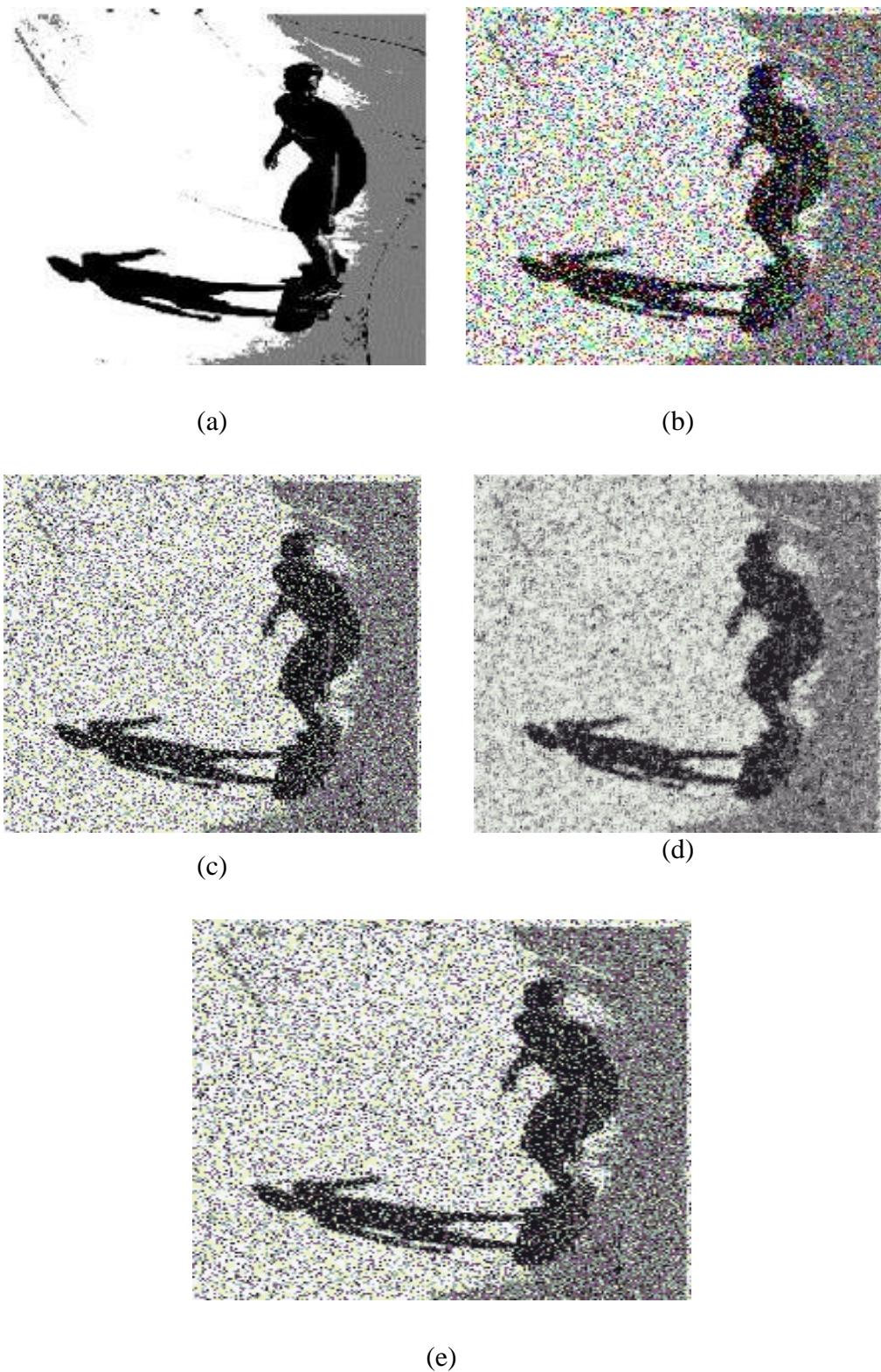


**Figure IV.11:** Segmentation results on the first synthetic image (SI1) corrupted by Gaussian noise ( $\sigma=25$ ). (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.



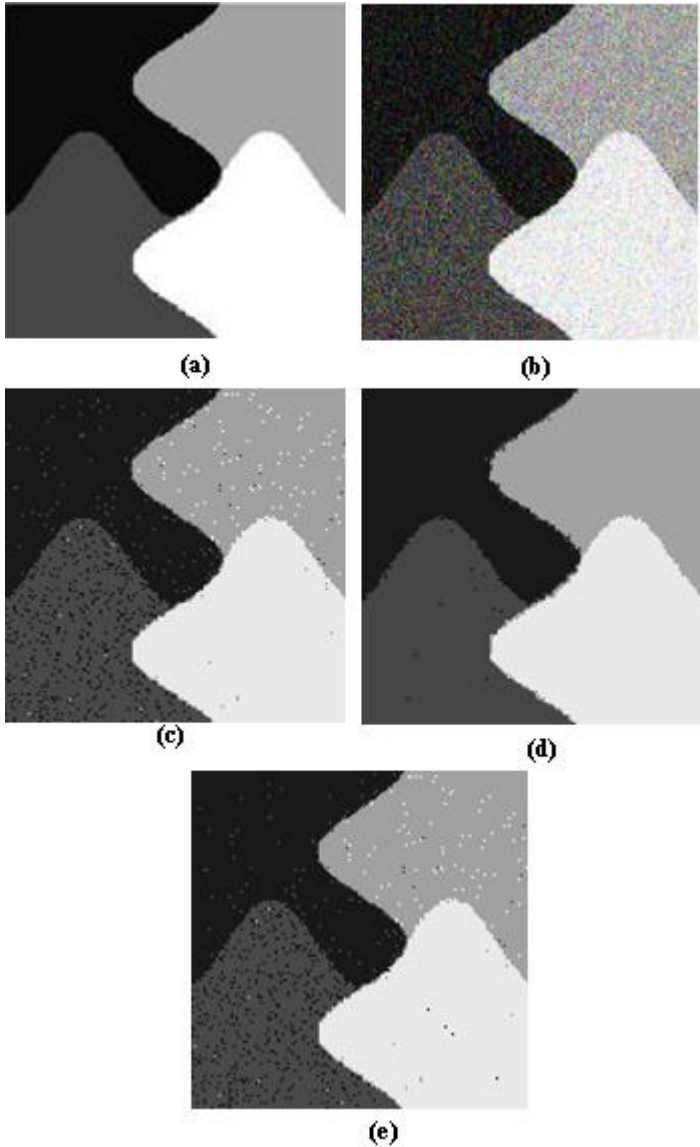
**Figure IV.12:** The image of a Shapes: (a) original image, (b) apply of Gaussian noise( $\sigma=30$ ),and Salt and Paper noise (15%) (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.



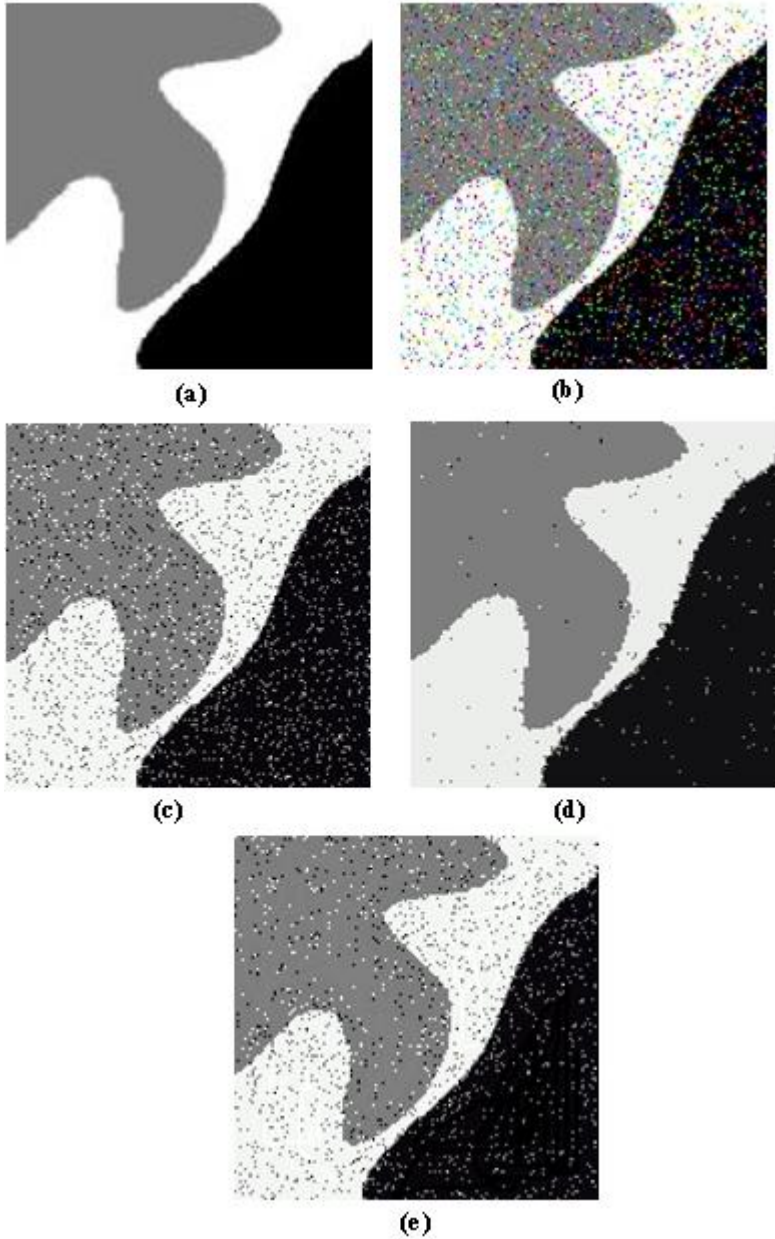


**Figure IV.13:** The image of a Skier: (a) original image, (b) corrupted by Gaussian noise ( $\sigma=15$ ) and Salt and Paper (30%) (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.

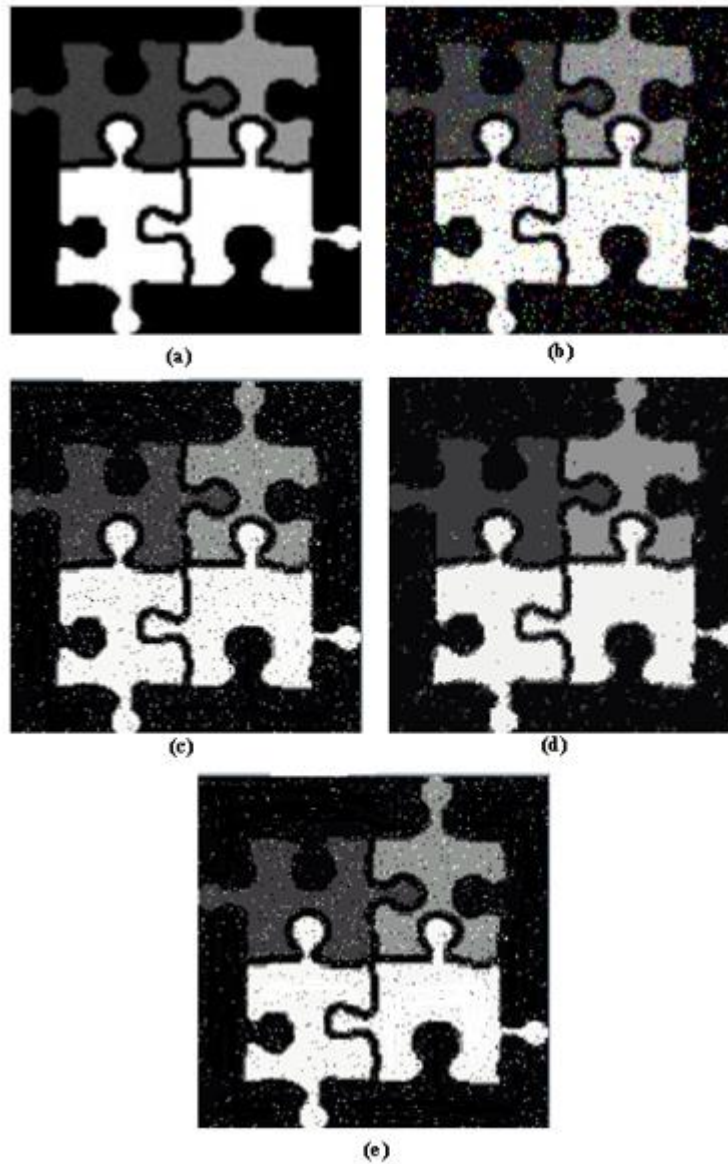




**Figure IV.14:** Segmentation results on the second synthetic image (SI2) corrupted by Gaussian noise ( $\sigma=25$ ). (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.



**Figure IV.15:** Segmentation results on the third synthetic image (SI3) corrupted by Salt and Pepper noise (20%) . (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.



**Figure IV.16:** Segmentation results on the fourth synthetic image (SI4) corrupted by Salt and Pepper noise (15%). (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.

## IV.9. The comparison between the algorithms

## IV.9.1 Comparison by the Validity Indices functions

The images	Algorithm used	Evolution functions		
		$V_{PE}$	$V_{PC}$	$V_{XB}$
The mountain	<i>Standard FCM</i>	0.4970	0.7353	0.1976
	<i>FGFCM</i>	<b>0.4878</b>	<b>0.7371</b>	5.4677
	<i>FLICM</i>	0.5166	0.2751	<b>0.4968</b>
Star Fish	<i>Standard FCM</i>	1.1209	0.4700	<b>3.6012</b>
	<i>FGFCM</i>	<b>1.004</b>	<b>0.5150</b>	20.36
	<i>FLICM</i>	1.1830	0.4409	6.04419
pepper	<i>Standard FCM</i>	0.8058	0.6265	20.2009
	<i>FGFCM</i>	<b>0.6656</b>	<b>0.7101</b>	1.0804
	<i>FLICM</i>	0.7794	0.6614	<b>0.7302</b>
The balls	<i>Standard FCM</i>	1.1920	0.4351	<b>4.6272</b>
	<i>FGFCM</i>	<b>1.0757</b>	<b>0.4794</b>	37.0759
	<i>FLICM</i>	1.2661	0.4023	11.56
Beach	<i>Standard FCM</i>	1.002	0.5412	<b>1.0604</b>
	<i>FGFCM</i>	<b>0.9247</b>	<b>0.5739</b>	9.4610
	<i>FLICM</i>	1.0629	0.5155	3.8368
Lena	<i>Standard FCM</i>	1.3067	0.4290	<b>0.9667</b>
	<i>FGFCM</i>	<b>1.2334</b>	<b>0.4609</b>	2.9760
	<i>FLICM</i>	1.3831	0.4004	4.6915
Bird	<i>Standard FCM</i>	1.1108	0.4589	<b>7.1172</b>
	<i>FGFCM</i>	1.1361	0.4465	181.4667
	<i>FLICM</i>	<b>1.0469</b>	<b>0.4852</b>	7.4629
Lake	<i>Standard FCM</i>	1.2456	0.4486	<b>2.4169</b>
	<i>FGFCM</i>	<b>1.1892</b>	<b>0.4701</b>	21.7377
	<i>FLICM</i>	1.3426	0.4072	11.0605
SI1	<i>Standard FCM</i>	0.2879	0.8679	0.3790
	<i>FGFCM</i>	<b>0.1464</b>	<b>0.9415</b>	<b>0.0383</b>
	<i>FLICM</i>	0.2719	0.8770	0.0499
Shapes	<i>Standard FCM</i>	0.68855	0.6624	<b>0.5610</b>
	<i>FGFCM</i>	<b>0.6381</b>	<b>0.6697</b>	481.8114

	<i>FLICM</i>	0.7340	0.6414	4.4752
<b>skier</b>	<i>Standard FCM</i>	<b>1.0223</b>	<b>0.5225</b>	<b>1.9552</b>
	<i>FGFCM</i>	1.1815	0.4400	225.7265
	<i>FLICM</i>	1.0925	0.4918	32.3468
<b>SI2</b>	<i>Standard FCM</i>	<b>0.4458</b>	0.7727	<b>0.1206</b>
	<i>FGFCM</i>	<b>0.1996</b>	<b>0.9106</b>	0.1640
	<i>FLICM</i>	0.4395	0.7767	0.2019
<b>SI3</b>	<i>Standard FCM</i>	0.1725	0.1941	<b>0.0055</b>
	<i>FGFCM</i>	<b>0.1586</b>	<b>0.9272</b>	0.1004
	<i>FLICM</i>	0.1732	0.9136	0.0080
<b>SI4</b>	<i>Standard FCM</i>	<b>0.1964</b>	<b>0.9060</b>	<b>0.02384</b>
	<i>FGFCM</i>	0.2412	0.8804	1.3213
	<i>FLICM</i>	0.1997	0.9045	0.0524

**Table IV-1:** Table of evaluation functions ( $V_{PE}$ ,  $V_{PC}$ ,  $V_{XB}$ ) of the fuzzy partitions of the three algorithms

The bold values represent the best results obtained for the comparison.

#### IV.9.2 Comparison by Entropy-based Objective Evaluation functions

The images	Algorithm used	objective quantitative evaluation functions			
		$F$	$\hat{F}$	$Q$	$H_l$
<b>The mountain</b>	<i>Standard FCM</i>	4.5306	1.8338	19.0543	7.5769
	<i>FGFCM</i>	<b>4.3551</b>	<b>1.7630</b>	<b>18.0832</b>	<b>7.0214</b>
	<i>FLICM</i>	4.4358	1.7955	18.8958	7.5677
<b>Star Fish</b>	<i>Standard FCM</i>	1.2592	0.7663	9.9426	9.9207
	<i>FGFCM</i>	1.3400	0.8154	9.9912	<b>8.2807</b>
	<i>FLICM</i>	<b>1.0354</b>	<b>0.6301</b>	<b>8.2629</b>	9.9129
<b>pepper</b>	<i>Standard FCM</i>	<b>5.2231</b>	<b>1.8849</b>	<b>12.8765</b>	10.4403
	<i>FGFCM</i>	5.7528	2.0761	13.8857	<b>8.7528</b>
	<i>FLICM</i>	5.8491	2.1109	13.9579	9.1776
<b>The balls</b>	<i>Standard FCM</i>	3.8329	1.1285	9.1630	11.1746
	<i>FGFCM</i>	3.7472	1.1033	9.3655	<b>10.8954</b>
	<i>FLICM</i>	<b>3.2541</b>	<b>0.9581</b>	<b>7.6170</b>	11.0980

<b>Beach</b>	<i>Standard FCM</i>	2.9709	3.9451	14.1792	14.5741
	<i>FGFCM</i>	2.6859	3.5666	12.9635	<b>13.9261</b>
	<i>FLICM</i>	<b>2.1366</b>	<b>2.8372</b>	<b>10.1972</b>	13.9598
<b>Lena</b>	<i>Standard FCM</i>	3.1631	4.1238	13.6809	13.8367
	<i>FGFCM</i>	<b>2.6834</b>	<b>3.4983</b>	<b>11.8590</b>	<b>13.1614</b>
	<i>FLICM</i>	2.8003	3.6508	12.2333	13.6656
<b>Bird</b>	<i>Standard FCM</i>	4645882.1136	0.4892	2.0687	9.2350
	<i>FGFCM</i>	<b>4560861.8391</b>	<b>0.4803</b>	<b>1.9440</b>	<b>8.0651</b>
	<i>FLICM</i>	5102422.0943	0.5373	2.0802	9.0944
<b>Lake</b>	<i>Standard FCM</i>	<b>4.3694</b>	<b>5.9874</b>	<b>19.1346</b>	16.9079
	<i>FGFCM</i>	4.5715	6.2643	20.3740	17.5682
	<i>FLICM</i>	4.9937	6.8429	22.1682	<b>16.8080</b>
<b>SI1</b>	<i>Standard FCM</i>	6239418.661	0.2712	2.9830	6.3241
	<i>FGFCM</i>	5292261.82	0.2300	2.4557	<b>6.1469</b>
	<i>FLICM</i>	<b>3498454.53</b>	<b>0.1520</b>	<b>1.6547</b>	6.1643
<b>Shapes</b>	<i>Standard FCM</i>	4.5427	6.8960	26.9486	12.1362
	<i>FGFCM</i>	5.4264	8.2375	33.3878	<b>9.9914</b>
	<i>FLICM</i>	<b>4.4699</b>	<b>6.7855</b>	<b>26.5436</b>	12.2062
<b>skier</b>	<i>Standard FCM</i>	8.4326	5.5410	32.8269	11.8988
	<i>FGFCM</i>	<b>3.8014</b>	<b>2.4979</b>	<b>13.6047</b>	<b>9.7949</b>
	<i>FLICM</i>	6.4575	4.2430	22.2699	11.2397
<b>SI2</b>	<i>Standard FCM</i>	3.2198	3.4061	25.1684	7.6301
	<i>FGFCM</i>	3.1240	3.3048	24.8449	<b>7.5666</b>
	<i>FLICM</i>	<b>2.9309</b>	<b>3.1005</b>	<b>22.9231</b>	7.5900
<b>SI3</b>	<i>Standard FCM</i>	<b>1.3952</b>	<b>1.0885</b>	<b>10.8924</b>	4.8639
	<i>FGFCM</i>	1.50005	1.1702	11.7729	4.7919
	<i>FLICM</i>	2.1444	1.6730	16.2737	<b>4.8630</b>
<b>SI4</b>	<i>Standard FCM</i>	3.7434	2.4729	18.9914	6.3641
	<i>FGFCM</i>	<b>3.7024</b>	<b>2.4458</b>	<b>18.9022</b>	<b>5.7939</b>
	<i>FLICM</i>	3.7559	2.4812	19.0605	6.3765

The bold values represent the best results obtained for the comparison.

**Table IV-2:** The Table of evaluation functions ( $F$ ,  $\tilde{F}$ ,  $Q$ ,  $H_i$ ) of the fuzzy partitions of the three algorithms

## IV.9.3 Comparison by the Execution time

The images	Algorithm used	Execution time (seconds)
<b>The mountain</b>	<i>FCM Standard</i>	<b>21</b>
	<i>FGFCM</i>	36
	<i>FLICM</i>	157
<b>Star Fish</b>	<i>FCM Standard</i>	<b>487</b>
	<i>FGFCM</i>	494
	<i>FLICM</i>	11063
<b>pepper</b>	<i>Standard FCM</i>	57
	<i>FGFCM</i>	<b>46</b>
	<i>FLICM</i>	989
<b>The balls</b>	<i>Standard FCM</i>	<b>66</b>
	<i>FGFCM</i>	142
	<i>FLICM</i>	615
<b>Beach</b>	<i>Standard FCM</i>	55
	<i>FGFCM</i>	<b>40</b>
	<i>FLICM</i>	44
<b>Lena</b>	<i>Standard FCM</i>	30
	<i>FGFCM</i>	<b>21</b>
	<i>FLICM</i>	28
<b>Bird</b>	<i>Standard FCM</i>	<b>19</b>
	<i>FGFCM</i>	32
	<i>FLICM</i>	30
<b>Lake</b>	<i>Standard FCM</i>	57
	<i>FGFCM</i>	<b>43</b>
	<i>FLICM</i>	49
<b>SI1</b>	<i>Standard FCM</i>	<b>32</b>
	<i>FGFCM</i>	48
	<i>FLICM</i>	53
<b>Shapes</b>	<i>FCM Standard</i>	<b>14</b>
	<i>FGFCM</i>	21
	<i>FLICM</i>	30
	<i>FCM Standard</i>	<b>33</b>

<b>skier</b>	<i>FGFCM</i>	60
	<i>FLICM</i>	98
<b>SI2</b>	<i>FCM Standard</i>	<b>34</b>
	<i>FGFCM</i>	45
	<i>FLICM</i>	51
<b>SI3</b>	<i>FCM Standard</i>	<b>31</b>
	<i>FGFCM</i>	40
	<i>FLICM</i>	52
<b>SI4</b>	<i>FCM Standard</i>	<b>43</b>
	<i>FGFCM</i>	52
	<i>FLICM</i>	60

**Table IV-3:** The convergence time in seconds (s) for the three algorithms

The bold values represent the best results obtained for the comparison.

By to the results obtained for the evaluation functions and the convergence time of three algorithms (standard FCM, FGFCM and FCLIM), we noticed that the resulting image of FGFCM Algorithm is better than the result of FLICM Algorithm and FCM , and FLICM is better than FCM in terms of image quality, also the FGFCM method shows a good robustness to noise especially for the evaluation functions  $V_{PE}$  and  $V_{PC}$ , and for Convergence time it takes extra time to compute the new image if we Compared with FCM , but it converges faster than FLICM algorithm.

#### IV.10. Conclusion

In this last chapter, we talked about the environment used, as well as the Java programming language and the use of the latter only, the names of the libraries used in our application with some explanations on the interface. Then we present the results obtained for the three implemented algorithms, and we also made a comparative study on the results obtained in terms of evaluation functions and convergence time.



## General Conclusion

In this work, we approached the problem of the classification of the images with unsupervised method, among the multitude of unsupervised method which can be used, we supposed for the algorithm of fuzzy classification FCM spatial (FGFCM), this algorithm largely increases the robustness of classic FCM thanks to the integration of spatial information in the fuzzy classification process.

The integration of spatial information is provided by the FGFCM algorithm through the generation of a new image from the input image, and it also defined a new similarity term between each centered point in a window and one of its neighbors, this term expresses the degree of influence of each neighbor on the new value (intensity or color) of central pixel.

The experimental results clearly show in terms of the evaluation functions used the superiority of this algorithm compared to the standard FCM and Flicm, but for the convergence time FGFCM took an extra time compared to the FLICM and even the FCM sometimes.

Among the perspectives of this work, the proposal of a new method to ensure an automatic initialization of the input parameters of the FGFCM algorithm (the number of classes, the centers), the exploitation of non-local information can influence largely on the results of classification, the last perspective of this work is the use of other functions to evaluate the results obtained.

## References

- [1] Tamdrari, Y and Bakiri, D (2011) Grayscale image segmentation by modeling based on ordered gray levels. Master's thesis, mouloud university Mammeri Tizi Ouzou.
- [2] Achat, T and Benhamaouche F (2015) Image segmentation by fuzzy classifiers: Application to Magnetic Resonance Imaging (MRI). Master's thesis, University Abderrahmane MIRA, Bejaia.
- [3] Ameer, S and Ameer, Z (2005) Review of image segmentation approaches textures: example of meteorological images", 3rd International Conference: Science of Electronic, Technologies of Information and Telecommunication, Tunisia.
- [4] J. C. Bezdek , "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, 1981 .
- [5] [https://en.wikipedia.org/wiki/Image\\_segmentation#Applications](https://en.wikipedia.org/wiki/Image_segmentation#Applications) (25/03/2022-19 :19)
- [6] S. Bres, J. Jolion, F. Lebourgeois. Traitement et analyse des images numerique. Paris, Lavoisier, Oct 2003, 411 pages.
- [7] T. Bousch, The fish has no bones, Ann. Inst. H. Poincar´e Probab. Statist. 36(4), 489–508, 2000.
- [8] Priyanka G. Kumbhar, 2 Prof. Sushilkumar N. Holambe, " A Review of Image Thresholding Techniques", International Journal of Advanced Research in Computer Science and Software Engineering , Volume 5, Issue 6, June 2015.
- [9] T.Romen Singh<sup>1</sup>, Sudipta Roy<sup>2</sup>, O.Imocha Singh<sup>3</sup>, Tejmani Sinam<sup>4</sup>, Kh.Manglem Singh, "A New Local Adaptive Thresholding Technique in Binarization," IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 2, November 2011.
- [10] J. Kittler, "A locally sensitive method for cluster analysis," Pattern Recognin , vol. 8, pp. 23–33, 1976.
- [11] Aissou,N and Benkaci, A (2016) Cooperative method for image segmentation Application to medical imaging. Master's thesis Abderrahmane Mira-Bejaia University.
- [12] chemchedine, M Toumi, L et Moussaoui. A (2009) Initialisation médiane de l’algorithme FCM : "Application à la segmentation des IRM cérébrales". Conference Paper, Département de l’informatique Université de Sétif.
- [13] Bitam, A.M,(2013) Analyse et segmentation d’images multispectrale : Application aux images MSG. Thèse doctorat, Universite Mouloud Mammeri Tizi Ouzou.
- [14] J. A. Hartigan and M. A. Wong, A k-means clustering algorithm, Applied Statistics 28 (1979), 100–108.
- [15] E. Diday, A new method in automatic classification and pattern recognition the method dynamic clouds, Revue de statitique appliqué 19 (1979), no. 2, 19–33.
- [16] T. Hastie, R. Tibshirani, and J. Friedman, The elements of statistical learning : Data mining, inference, and prediction, Springer Series in Statistics, Springer, 2003.

## References

- [17] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Plenum, 1981.
- [18] Bezdek, J, Keller, J, Pal, N et Krisnapuram, R (1995) Fuzzy Models And Algorithms For Pattern Recognition And Image Processing. Kluwer Academic Publishers.
- [19] Dunn, J (1973): A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Journal of Cybernetics, 3:32–57
- [20] Bezdek, J(1981). Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.
- [21] AKROUR, N CHABI, L (2009) Une plateforme évolutive pour le développement d'approches biomimétiques en segmentation d'images. Mémoire de fin d'études, Ecole nationale Supérieure d'Informatique (ESI) Oued-Smar, Alger.
- [22] Crucianu, M(2009), Reconnaissance des formes et méthodes neuronales, Chapitre 5 : Classification automatique, CNAM Paris.
- [23] Lambert, P et Grecu, H (2003) A quick and coarse color image segmentation , ICIP 2003, 14-17
- [24] Thirumuruganathan, s., a detailed introduction to k-nearest neighbor (knn) algorithm , <<https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/>>, may 2010.
- [25] Wikipedia, k-nearest neighbors algorithm, < [https://en.wikipedia.org/wiki/k-nearest\\_neighbors\\_algorithm](https://en.wikipedia.org/wiki/k-nearest_neighbors_algorithm) >, 2016
- [26] Hilali, h., application of textual classification for the extraction of maximal association rules. Master's thesis in computer science, University of Quebec at Trois-Rivières, Trois-Rivières, 2009.
- [27]. Sabrina, t., machine learning and dimensionality reduction, <<http://www-ia.lip6.fr/~tollaris/articles/these/node7.html>>, 2008.
- [28] Descôteaux, s., maximum association rules for the interpretation of classification results. master's thesis in computer science, university of quebec at Trois-Rivières, Trois-Rivières, 2014: p. 174.
- [29] Tay, b., j.k. hyun, and s.o.s. oh, computational and mathematical methods in medicine ,<[https://www.researchgate.net/figure/260397165\\_fig7\\_pseudocode-for-knn-classification](https://www.researchgate.net/figure/260397165_fig7_pseudocode-for-knn-classification)>, jan 2014.
- [30] Wikipedia, machine à vecteurs de support ,<[https://fr.wikipedia.org/wiki/machine\\_%c3%a0\\_vecteurs\\_de\\_support](https://fr.wikipedia.org/wiki/machine_%c3%a0_vecteurs_de_support) >, 2017.
- [31] Learn, s., support vector machines ,<<http://scikit-learn.org/stable/modules/svm.html> >.
- [32] Gurney, k., an introduction to neural networks. 1997.

## References

- [33] Fausett, I.v., fundamentals of neural networks: architectures, algorithms and applications. 1993.
- [34] Wikiversite, réseaux de neurones : avantages et possibilités ,  
<[https://fr.wikiversity.org/wiki/r%C3%A9seaux\\_de\\_neurones/avantages\\_et\\_possibilit%C3%A9s](https://fr.wikiversity.org/wiki/r%C3%A9seaux_de_neurones/avantages_et_possibilit%C3%A9s)>,18 mai 2016.
- [35] Amira, I., B. Khaoukha, and I. Adila, identification et commande des systèmes non linéaires. 2011.
- [36] J-P. Cocquerez, S. Phlipp. Analyse d'images : filtrage et segmentation. Masson, 1995, 457 pages.
- [37] J.M. Rendon Mancha, " Régions Actives Morphologiques : Application à la Vision par Ordinateur ", Thèse de doctorat, Université René Descartes - Paris V, (2002).
- [38] J-P. Cocquerez, S. Phlipp. Analyse d'images : filtrage et segmentation. Masson, 1995, 457 pages.
- [39] CHEVRIE, F et GULY, F(1998), Cahier technique n° 191 « La logique floue », Collection Technique, CT 191 édition.
- [40] Pham, "An adaptive fuzzy C-means algorithm for image segmentation in the presence of intensity inhomogeneities," Pattern Recognition Letters, vol. 20, pp. 57–68, 1999.
- [41] Dunn, J (1973): A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Journal of Cybernetics, 3:32–57
- [42] Bezdek, J(1981). Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.
- [43] M.N. Ahmed, S.M. Yamany, N. Mohamed, A.A. Farag, T. Moriarty, A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data, IEEE Trans. Med. Imaging 21 (2002) 193–199.
- [44] M.N. Ahmed, S.M. Yamany, N.A. Mohamed, A.A. Farag, T. Moriarty, Bias field estimation and adaptive segmentation of MRI data using modified fuzzy c-means algorithm, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, vol. 1, 1999, pp. 250–255.
- [45] S.C. Chen, D.Q. Zhang, Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure, IEEE Trans. Systems Man Cybernet. B 34 (4) (2004) 1907–1916.
- [46] L. Cai, Z. Beny, S.M. Szilagy, H.S. Adam, MR brain image segmentation using an enhanced fuzzy c-means algorithm, in: 25th Annual International Conference of IEEE EMBS, 2003, pp. 17–21.
- [47] W. Cai et al. / Pattern Recognition 40 (2007) 825 – 838
- [48] S.Kirindis and V. Chatzis, " A Robust Fuzzy Local Information CMeans

## References

Clustering Algorithm,” IEEE Trans. Image Process. Vol.19, No. 5 , pp. 1328-1337,May 2010

[49] M. Ahmed, S. Yamany, N. Mohamed, A. Farag, and T. Moriarty, “A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data,” IEEE Trans.Med. Imag., vol. 21, pp. 193–199,2002.

[50] K.S. Chuang, H.L. Tzeng, S.W. Chen, J. Wu, T.J. Chen, “Fuzzy c-means clustering with spatial information for image segmentation”, Computerized medical imaging and graphics, 30(1) , 2006, pp. 9–15.

[51] D. L. Pham, "Fuzzy clustering with spatial constraints", in Proceedings International Conference on Image Processing : Vol. II, 2002, pp. 65–68.

[52] C. Tomasi, R. Manduchi, Bilateral filtering for grayscale and color image, in: 6th IEEE International Conference on Computer Vision, 1998, pp. 839–846

[53] S.K. Adhikari, J. K. Sing, D. K. Basu, M. Nasipuri, “Conditional spatial fuzzy C-means clustering algorithm for segmentation of MRI images ”, Applied Soft Computing, (34), pp.758-769, 2015.

[54] : X. Xie, G. Beni, “A validity measure for fuzzy clustering”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(8), 1991, pp. 841–847.

[55] Jianqing Liu and Yee-Hong Yang, “Multi-resolution color Image segmentation,” IEEE Transaction on Pattern Analysis and Machine Intelligence, 16(7), pp. 689–700, 1994.

[56] M. Borsotti, P. Campadelli, and R. Schettini, “Quantitative evaluation of color image segmentation results, ” Pattern Recognition Letters, 19, pp. 741–747, 1998.

[57] J. Rissanen, “A universal prior for integers and estimation by minimum description length,” Annals of Statistics, 11(2), pp. 416–431.[57]

# Index

<b>General Introduction .....</b>	<b>1</b>
<b>Chapter I. Approaches Segmentation Image .....</b>	<b>3</b>
I.1. Introduction.....	3
I.2. Definition of an image .....	3
I.3. Digital picture .....	3
I.3.1 Characteristics of a digital image.....	3
I.3.2 pixels.....	3
I.3.3 The dimension.....	3
I.3.4 The resolution .....	4
I.3.5 Contrast.....	4
I.3.6 Luminance .....	4
I.3.7 Noise .....	5
I.3.8 The histogram .....	5
I.4. Image types.....	5
I.4.1 1.5.1. Binary image (black and white).....	5
I.4.2 Grayscale (monochrome) image.....	5
I.4.3 Color image (full color) .....	6
I.5. Image segmentation definition.....	7
I.6. Applications Domaine .....	9
I.7. Image Segmentation Approaches .....	12
I.7.1 Region based approaches.....	13
I.7.2 Contours based approaches.....	34
I.8. Conclusion .....	37
<b>Chapitre II. Segmentation by fuzzy c-means method.....</b>	<b>38</b>
II.1. Introduction.....	38
II.2. fuzzy logic .....	38
II.3. The fuzzy subset theory .....	39
II.4. Clustering definition .....	40
II.5. Fcm algorithm.....	40
II.6. Fcm algorithm shortcomings .....	42
II.7. Fcm variants.....	42
II.7.1 Fuzzy Clustering with Constraints (FCMS) and its Variants .....	42
II.7.2 C. Enhanced Fuzzy C-Means Clustering (EnFCM).....	44
II.7.3 Fast Generalized Fuzzy C-Means Clustering (FGFCM) .....	46
II.7.4 FUZZY LOCAL INFORMATION C-MEANS (FLICM)CLUSTERING ALGORITHM .....	47

The FLICM algorithm is given as follows:.....	49
II.7.5    Spatial FCM.....	49
II.7.6    robust fuzzy c-means with adaptive spatial & intensity constraint and membership linking (FCM_SICM)[52].....	50
II.8.    Conclusion .....	53
<b>Chapitre III. Design.....</b>	<b>54</b>
III.1.    Introduction.....	54
III.2.    General design .....	54
III.2.1    Input picture.....	55
III.2.2    Classification methods.....	55
III.2.3    labeling.....	55
III.2.4    Comparison methods .....	55
III.3.    The detailed design .....	58
III.3.1    Saving the input image .....	61
III.3.2    Generation of a new image .....	61
III.3.3    Parameter and input initialization.....	65
III.3.4    Calcul des centres et des degrés d'appartenance .....	65
III.3.5    The stopping criterion.....	65
III.3.6    Labeling .....	66
III.4.    Interface module .....	66
III.5.    Conclusion .....	70
<b>Chapitre IV. Implementation .....</b>	<b>71</b>
IV.1.    Introduction.....	71
IV.2.    The work environment.....	71
IV.3.    The coding language.....	71
IV.4.    Coding software.....	72
IV.5.    Libraries to use.....	73
IV.6.    Machine Features.....	73
IV.7.    Presentation of the application interface.....	73
IV.8.    Evaluation on segmentation results .....	75
IV.9.    The comparison between the algorithms .....	89
IV.9.1    Comparison by the Validity Indices functions.....	89
IV.9.2    Comparison by Entropy-based Objective Evaluation functions .....	90
IV.9.3    Comparison by the Execution time.....	92
IV.10.    Conclusion .....	93
<b>General Conclusion .....</b>	<b>94</b>
<b>References .....</b>	<b>95</b>

É

\*



## Figure Index

Figure I.1: Grayscale quantization.....	6
Figure I.2: Grayscale number variation .....	6
Figure I.3: Red Green Blue (RGB) color space.....	7
Figure I.4: Image Segmentation Approaches.....	13
Figure I.5: Segmentation by region-based approaches .....	13
Figure I.6: Segmentation by thresholding.....	14
Figure I.7: Thresholding Techniques .....	15
Figure I.8: The process of a region growing algorithm. ....	18
Figure I.9: Clustering Method.....	19
Figure I.10 :Two-class classification problem with a linear separator .....	28
Figure I.11 :Two-class classification problem with a nonlinear separatrix .....	28
Figure I.12: Optimal hyperplane with maximum margin .....	29
Figure I.13: Artificial neuron with a single output .....	30
Figure I.14: The single-level neural network.....	31
Figure I.15: The multi-level neural network.....	31
Figure I.16: Identity function.....	32
Figure I.17: Function of a threshold $\theta$ .....	33
Figure I.18: Function sigmoide.....	33
Figure I.19: 3 Different types of outlines.....	35
Figure I.20: Active contour principle.....	35
Figure II.1: Fuzzy logic.....	38
Figure II.2: Comparison of a classical set and a fuzzy set.....	40
Figure III.1: Overall System Architecture .....	54
Figure III.2: Detailed system architecture .....	60
Figure III.3: (a) the central point, (b) the neighbors of the central point.....	61
Figure III.4:The possible cases of the positions of central pixel (red) and its neighbors (green) in a window of size 3*3.....	62
Figure IV.1: The environment interface .....	72
Figure IV.2: Home window .....	73
Figure IV.3: Segmentation results on the mountain corrupted by Salt an Pepper noise. (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result.(e) FLICM result. ....	75
Figure IV.4: The image of a Star Fish : : (a) original image, (b) corrupted by Gaussian noise ( $\sigma=30$ ) (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.....	76

Figure IV.5: The image of a Pepper : (a) original image, (b) corrupted by Gaussian noise ( $\sigma=30$ ) (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.....	77
Figure IV.6 :The image of Balls: (a) original image, (b) corrupted by Gaussian noise ( $\sigma=15$ ) (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result. ....	78
Figure IV.7: Segmentation results on the Beach image corrupted by Gaussian noise ( $\sigma=15$ ). ....	79
Figure IV.8: Segmentation results on the Lena image corrupted by Gaussian noise ( $\sigma=25$ ). (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result.(e) FLICM result. 80	
Figure IV.9: Segmentation results on the Bird image corrupted by Salt an Pepper noise (30%).(a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.....	81
Figure IV.10: Segmentation results on the Lake image corrupted by Gaussian noise ( $\sigma=30$ ). (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result. ....	82
Figure IV.11: Segmentation results on the first synthetic image (SI1) corrupted by Gaussian noise ( $\sigma=25$ ). (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result. ....	83
Figure IV.12: The image of a Shapes: (a) original image, (b) apply of Gaussian noise( $\sigma=30$ ),and Salt and Paper noise (15%) (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result. ....	84
Figure IV.13: The image of a Skier: (a) original image, (b) corrupted by Gaussian noise ( $\sigma=15$ ) and Salt and Paper (30%) (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result.....	85
Figure IV.14: Segmentation results on the second synthetic image (SI2) corrupted by Gaussian noise ( $\sigma=25$ ). (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result. ....	86
Figure IV.15: Segmentation results on the third synthetic image (SI3) corrupted by Salt an Pepper noise (20%) . (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result. ....	87
Figure IV.16: Segmentation results on the forth synthetic image (SI4) corrupted by Salt an Pepper noise (15%) . (a) Original image. (b) Noisy image. (c) FCM result. (d) FGFCM result. (e) FLICM result. ....	88