FUZZY C-MEAN TECHNIQUE FOR IMAGE SEGMENTATION

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree
of
MASTER OF TECHNOLOGY
in
ELECTRICAL ENGINEERING
(With Specialization in Instrumentation and Signal Processing)

By

RAJEEV CHANDEL

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)
JUNE, 2013

CANDIDATE'S DECLARATION

I hereby certify that the desertion entitled "**FUZZY C-MEAN TECHNIQUE FOR IMAGE SEGMENTATION**" being submitted by me towards partial fulfilment of the requirements for the award of Master of Technology with specialisation in "**Instrumentation and Signal Processing**" at Department of Electrical Engineering, IIT, Roorkee is a bonafide review work carried out by me under the supervision of Dr R P Maheshwari and Dr. Manoj Tripathy in Department of Electrical Engineering, IIT, Roorkee.

Date: 14 Jun 13

Place: IIT, Roorkee

Rajeev Chandel

M.Tech (I & SP)

CERTIFICATE

This is to certify that the above statement made by the candidate is true to the best of my knowledge and belief.

(Dr. M. Tripathy)
Assistant Professor
Dept of Electrical Engineering
Indian Institute of Technology

(Dr R P Maheshwari)
Professor
Dept of Electrical Engineering
Indian Institute of Technology

ACKNOWLEDGEMENT

I express my regards and sense of gratitude to my guides Dr Rudra Prakash Maheshwari and Dr Manoj Tripathy, Department of Electrical Engineering, Indian Institute of Technology, Roorkee for giving inspiration, expert guidance, moral support and encouragement throughout this work. Their vast experience, sharp and inclusive intelligence, dynamism, timely help and painstaking efforts have unerringly steered the work on smooth and steady course.

My sincere thanks to the Head, Electrical Engineering Department, I.I.T Roorkee for providing necessary research facilities to carry out this work and valuable suggestions and motivations provided by other faculty members of the department are duly acknowledged.

I extend special thanks to all my batch mates for always supporting and standing with me. I also extend my thanks to my parents and wife for supporting and encouraging me to carry out and complete the dissertation work successfully.

This acknowledgment would be incomplete if I do not thank my son for constant inspiration and enduring the journey along.

Rajeev Chandel

CONTENTS

ABSTRACT

In this thesis, Fuzzy C-means technique is presented as basis for image segmentation process. The various aspects of working of Fuzzy C-means algorithms are highlighted and the sequential development of these algorithms is given. The advantages of kernel functions and their use in the process of image segmentation are specified. Fuzzy C-Means (FCM) is a prevalent soft-clustering technique. This clustering technique is widely used in the task of image segmentation because of its ease of execution and rapid convergence. By using kernel properties, the Kernel Fuzzy C-Means algorithm attempts to map the data with nonlinear relationships to appropriate higher dimensional spaces. In the new space, the data can be more easily separated or clustered. Kernel combination, or selection, is fundamental for effective kernel clustering. By incorporating multiple kernels and automatically adjusting the kernel weights, Multiple Kernel Fuzzy C-Mean (MKFC) method is more immune to ineffective kernels and their extraneous features. This makes the choice of kernels less crucial and the method more effective for image segmentation. Effective kernels and associated features tend to contribute more to the clustering and, therefore, improve results of image segmentation. The MKFCM algorithm provides us a new platform to blend different types of image information in image-segmentation problems. In this report, the technique for weight optimization is presented whereby obviating the need for centre calculations for the clusters and improving the performance of Multiple Kernel Fuzzy C-Mean algorithm. Simulations on the segmentation of synthetic, medical image and other images demonstrate the flexibility and advantages of MKFCM based approaches for image segmentation. The values of various parameters involved in the MKFCM algorithm are studied and guidelines for value selection are suggested. The fuzzy c-mean technique for image segmentation is a robust, easy to realize and effective methodology. Apart from these advantages it offers a great benefit by providing a platform for information fusion.

ABBREVIATIONS

FCM	Fuzzy C-Mean
KFCM	Kernel Fuzzy C-Mean
MKFCM	Multiple Kernel Fuzzy C-Mean
A_j	Set of input data points
N	Total number of input data points
$\mathbb{R}^d$	Feature space or Data space
d	Dimension of input data points
CF_{FCM}	Cost function (weighted sum of distance between data and cluster center)
Mem	Fuzzy partition matrix
Cen	Cluster prototype matrix
F	Fuzzification parameter
$Dist$	Distance measure between input data and cluster center
ε	Threshold sensitivity
H	High dimensional feature space
M	Transform function
K	Kernel function
k_{com}	Composite kernel function
LMKFCM	Linear combined multiple kernel fuzzy c-mean
DKFCM	Direct kernel fuzzy C-mean
G	Emphasis factor
O	Prototype vector

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Chapter 1

INTRODUCTION

1.1 Brief Introduction

Image processing is a form of signal processing where the input signal is an image and the output is an image itself or its parameters [1]. In modern times, the field of image processing has revolutionised with the advent of computers. The digital image processing has acquired bigger roles right from household photography, biomedical imaging, and remote sensing to robotics vision. Humans perceive mostly through their sense of vision and use it extensively to convey information. At least about 75 - 80 percent information gathered by humans is through ocular senses [2]. Therefore, there has been a tremendous increase in the arena of image processing and its analysis. The ambit of digital image processing is much larger than that of human vision. The human vision is restricted by the electromagnetic spectrum (visual band), but the modern imaging machines cover entire electromagnetic spectrum. Digital image processing is processing of digital images by digital means mostly by computers [1]. It allows the use of much more complex methods and offers both, implementation of methods which otherwise would be impossible by analog means and better performance at simple tasks.

Image segmentation is a very vital and central task in digital image processing. The process of subdividing the image into its constituent areas or segments is called as segmentation [1, 3]. It simplifies the portrayal of the given image into meaningful information by segregating the image into different areas. These areas are different than each other based upon certain characteristic like colour, intensity, or texture and the region within the area is harmonized. The objective of segmentation is to simplify and/or alter the image representation into something that is more significant and easier to investigate. The process of segmentation is typically used to locate objects, areas and boundaries or edges within an image. There are numerous methods of achieving image segmentation like region growing, thresholding, histogram-based methods, edge detection, watershed transform, model based object. This is used to detect drought conditions by estimating the crop cover of an area under cultivation.

e. **Target Detection**

The input image is segmented and various objects are identified, the objects are then analyzed to decide the suitable target.

f. **Miscellaneous**

It can be utilized in many other fields as detection of brake lights, analyzing aerial imagery, automating and controlling of various metallurgy processes (to control grains orientation of the metal).

Since segmentation is the most preliminary step of digital image processing hence segmentation accuracy is of paramount importance for success or failure of the process for which image analysis is being used.

1.3 Clustering For Segmentation

In the digital image processing image is represented as an array of data representing certain characteristic or combination of more than one feature. The data clustering was envisaged as a method to segment the image and clustering is also the process of dividing data into groups of similar objects for better analysis akin to the goal of segmentation process [5]. In both cases the data within a group is more homogenous than the data in other groups.

In data clustering many approaches have been developed. Soft clustering methods are preferred for image segmentation as there is uncertainty present in an image in terms of the boundaries, vagueness of class definitions, and imprecise gray levels of pixels [6]. Among fuzzy clustering technologies, Fuzzy C-Means (FCM) algorithm is most extensively used for image segmentation. The FCM algorithm designates each pixel to fuzzy clusters and permit pixels of the image to belong to many groups with varying degrees of association to each of these groups. The Fuzzy C-Means algorithm and its variants are the popular choice for image segmentation due to their ease of execution and ability to converge within little time. Many variations like inclusion of pixel neighborhood spatial information in FCM have led to improvement in performance. Apart from these improvements, kernelization of FCM has made considerable improvement in performance. This algorithm which does clustering after doing the mapping with the help of a kernel function are called as Kernel Based Fuzzy C-Mean (KFCM) algorithm. Kernelization helps to map data from data/feature space to a much higher dimensional space called as kernel/Hilbert space. This mapping is achieved by the help of a function called as kernel. The kernels and their properties are given in greater details in chapter three. In this higher dimensional space the data is easily segregated or clustered. The choice of kernel remains an issue. To overcome this drawback we consider multiple kernels instead of one kernel. Multiple kernels give us the flexibility in kernel selection and we can employ various techniques to automatically adjust weights of the different kernels. In practical world the data is collected from various sources, using multiple kernels provide us good tool for information fusion. This improves the performance of the Multiple Kernel Fuzzy C-Mean (MKFCM) method used for image segmentation.

1.4 **Objective Of Dissertation**

As brought out in the above paragraph, the Fuzzy C-Mean technique gives us an easy to implement and effective tool for image segmentation. This technique also provides a platform for image information fusion acquired from several homogenous and heterogeneous sources. These advantages make this technique very useful for image segmentation in various medical, military and other applications. The dissertation is built around Fuzzy C Means technique for image segmentation.

Firstly, it is proposed to consider various algorithms available to achieve image segmentation using Fuzzy C-Mean Technique. The main advantages and the disadvantages of these algorithms are highlighted.

Secondly, it is proposed to suggest some alteration in the best available algorithm in this domain for enhanced performance.

Thirdly, as the performance is dependent on certain parameters, it is proposed to find the optimal solutions for these parameter values for a better performance of this algorithm for image segmentation.

1.5 Organization Of Dissertation

The clustering and its approach to image segmentation along with Fuzzy C Means algorithm is described in chapter two. The kernel concept, properties of kernels and advantages of kerrnelisation are dealt with in chapter three. After understanding kernels, the Kernel Based Fuzzy C Means algorithm and the Multiple Kernel Fuzzy C Means Algorithm are presented. In chapter four, implementation of Multiple Kernel Fuzzy C Means algorithm for image segmentation is explained and the proposed weight optimization technique is discussed. In next chapter the optimal parameter values are presented and in chapter six, the future possibilities which emanates from the dissertation work are discussed before brief conclusions are made.

Chapter 2

CLUSTERING

2.1 Data Analysis

Clustering is one of the processes involved in data analysis. Data is subjective information that makes a representation about the system [6]. The data could be about certain parameters of the process, values of turnover of the business, the degree of liking for a particular player or pixel intensity of the image. Analysis of data is always done for acquiring certain information about the system. Type of information required determines the approach for data analysis. However, it is also dependent on the type of data. The data analysis is further subdivided into four steps [7]. These are

a. This step involves an integrity evaluation or dependability check and straightforward occurrence analysis. This is performed to recognize outliers.

b. In this step the data is grouped based on the pattern identification. These identified groups are then structured.

c. At this step the data are scrutinized with the help of some mathematical model.

d. The data are evaluated in this step. Based on these evaluations, the conclusions are drawn and then based on these conclusions the judgments are made.

In simple terms, word clustering is referred to as a group of comparable or alike things. Clustering is the process of dividing data into groups of objects with similar characteristics. Each group is referred to as cluster. A cluster is a congregation of objects that are similar to one another (homogenous) and dissimilar to objects of other groups. The idea of similarity is taken as per data given. During the process of clustering, selection of large number of clusters includes greater information but it increase computational complexity. Clustering data with few clusters drop certain minute details but it attains simplification. Clustering can be referred to as a process of discovering a data structure and it divides a data set into a number of subsets with correlated data called as clusters. Clustering finds application in several fields, such as finance, mathematics, genetics, business, engineering systems, medicine, and image processing. Clustering complexity of data mining applications increases since they need to take into account large datasets, their numerous attributes, and different types of attributes. These additional requirements of clustering increase computational requirements of data analysis. These requirements of clustering led to the emergence of powerful and widely applicable data clustering algorithms.

2.2 Purpose

In general data clustering has found very wide applicability in various fields but it has been used mainly for the following four important purposes.

a. **To Find Main Organization:** To gain insight into data, generate assumptions, detect anomalies, and identify significant features.

b. **For Conclusion Making:** To arrive at certain decision based upon underlying structure.

c. **To Identify Likely Categorization:** To identify the amount of similarity among data instances

d. **Compression:** As a method for organizing the data and storing it as clusters.

2.3 Clustering Techniques

There are various ways for clustering data [8]. Some of the methods are

a. **Hard Clustering** The data point is associated to one and only one data group.

enables us to discover and learn the clear knowledge based depiction of the information which is inbuilt in the data. The technique of fuzzy clustering is a subset of fuzzy data analysis. It comprises of two distinct types

a. The analysis of fuzzy data
b. The analysis of crisp data with fuzzy techniques

In case of image processing, for greyscale images, the pixel intensities can be taken as the degree of association to colour white or black. Similar thing can be done to the coloured images as well. Therefore fuzzy clustering technique can be used for image processing and will be treated as part of fuzzy data analysis [8].

Human beings can differentiate between the two faces or their image in just a glance. Technically, this is a very complex process and even modern day high speed computers cannot achieve such a performance. The difference in performance is not due to the lack of visual input to the humans or the computers in terms of optical sensors or the ability to transmit acquired data. This difference in performance is attributed to the capability to extract useful information from data and its analysis. To identify facial features may look natural to a human brain but it is a herculean task for the computers even with the help of the state of the art algorithms [6]. The automation is forcing and keeping the demand for automatic analysis very high. The automatic analysis may be for controlling certain industrial process, making autopilot for car driving, visual control of quality or identifying images. These processes when performed by humans, seems very easy but humans cannot imitate their own decision making and action taking process on the computers. This is due to the fact that all these process happen unconsciously. But when it comes to finding dependencies in multi-dimensional data, then humans are dependent on computers. This led to development of many data analysis techniques.

In recent times, the application of fuzzy sets in the process of image manipulation has gained popularity. This increased interest is based on the fact that many of the basic concepts in image scrutiny are vague. The very idea of an edge or a corner or a boundary in an image is somewhat vague.

The relations between various areas within an image are not very crisp. Uncertainty in an image can be in the form of geometrical ambiguity or intensity ambiguity or a blend of both. Geometrical/spatial ambiguity is vagueness in the figure, outline and/or geometry of an area within the image. Intensity ambiguity is the vagueness in concluding if a pixel is white or black. Gray tone images have ambiguity because of the possible more than one levels of brightness in the pixels of an image. Since there is uncertainty in an image and these uncertainties are not always defined very clearly, vagueness can happen within each step of image analysis involving these vague features. Impact on all higher level analysis will be obvious due to any decision made at a lower level. Therefore it is always beneficial and wise to avoid hard decisions for all vague features. It will be natural and convenient to allow the image analysis regarding vague features to be fuzzy subsets of the image. Each pixel belonging to these subsets are branded by the probability, possibility or a degree of belonging.

2.5 **Fuzzy C-Means Method**

At first Duda and Hart gave an algorithm for doing crisp clustering on the crisp data set. This algorithm was called as Hard C-Means. Dunn gave the fuzzy edition of the Hard C-Means algorithm. The algorithm given by Dunn was again modified by Bezdek. Bezdek gave the concept of fuzzifier. This final algorithm was named as Fuzzy C-Mean algorithm [8]. Fuzzy C- Means (FCM) algorithm is very widely used and is quite liked method for fuzzy clustering. It is more flexible than many crisp or hard clustering methods and it is easy to implement. This algorithm uses the Euclidian distance as the measure of similarity and hence recognised the spherical groups of data in a given 'p' dimensional space. Every cluster is represented by cluster centre which is also called as prototype.

The FCM groups the data set in C fuzzy clusters the set of data points $A_j \in \mathbb{R}^{\dim};\ j = 1,2 \ldots \ldots, N$.

Where

$\mathbb{R}$ is the feature or data space

'dim' is the input dimension of data

The FCM optimises the cost function

$$CF_{FCM}(Memb, Cen) = \sum_{i=1}^{K}\sum_{j=1}^{N}(Memb_{ij})^{f}\, Dist_{ij} \tag{2.1}$$

also

$$Memb_{ij} \in [0,1] \text{ and } \sum_{i=1}^{K} Memb_{ij} = 1\ \forall\, j \tag{2.2}$$

The various notations are

$Memb = \{Memb_{ij}\}C \times N$ is the fuzzy partition matrix

$Memb_{ij}$ is the fuzzy membership coefficient of the j^{th} object in the i^{th} cluster

$Cen = \{Cen_1, \ldots \ldots . Cen_K\}$ is the cluster centre matrix

$f \in [1, \infty)$ is the fuzzification parameter

$Dist_{ij} = Dist(A_j, Cen_i)$ is the measure of distance between A_j and Cen_i

2.5.1 Steps for Fuzzy C-Means Algorithm

Various steps to achive clustering through the Fuzzy C-Mean (FCM) algorithm are

a. Choose suitable values for f, C, t_{max} and a small positive number (ε) .

b. Initialize the centre matrix Cen.

c. Let step variable be $t = 0$.

d. Find (at $t = 0$) or update (at $t > 0$) the membership matrix $Memb$ by

$$\mathrm{Mem}_{ij}^{t+1} = 1\bigg/\left(\sum_{l=1}^{K}\left(\mathrm{Dist}_{lj}/\mathrm{Dist}_{ij}\right)^{1/(1-f)}\right) \quad (2.3)$$

$$for\ \ i = 1, \ldots.. K\ and\ j = 1, \ldots.. N$$

e. Revise the center matrix Cen by

$$Cen_i^{t+1} = \left(\sum_{j=1}^{N}\left(Mem_{ij}^{(t+1)}\right) A_j\right)\bigg/\left(\sum_{j=1}^{N}\left(Mem_{ij}^{(t+1)}\right)^f\right) \quad (2.4)$$

for $i = 1, \ldots \ldots K$

f. Replicate steps (d) & (e) until $\| Mem^{t+1} - Mem^{t} \| < \varepsilon$ or $t = t_{max}$

The above steps can be put in form of a flow chart as shown in the figure (2.1).

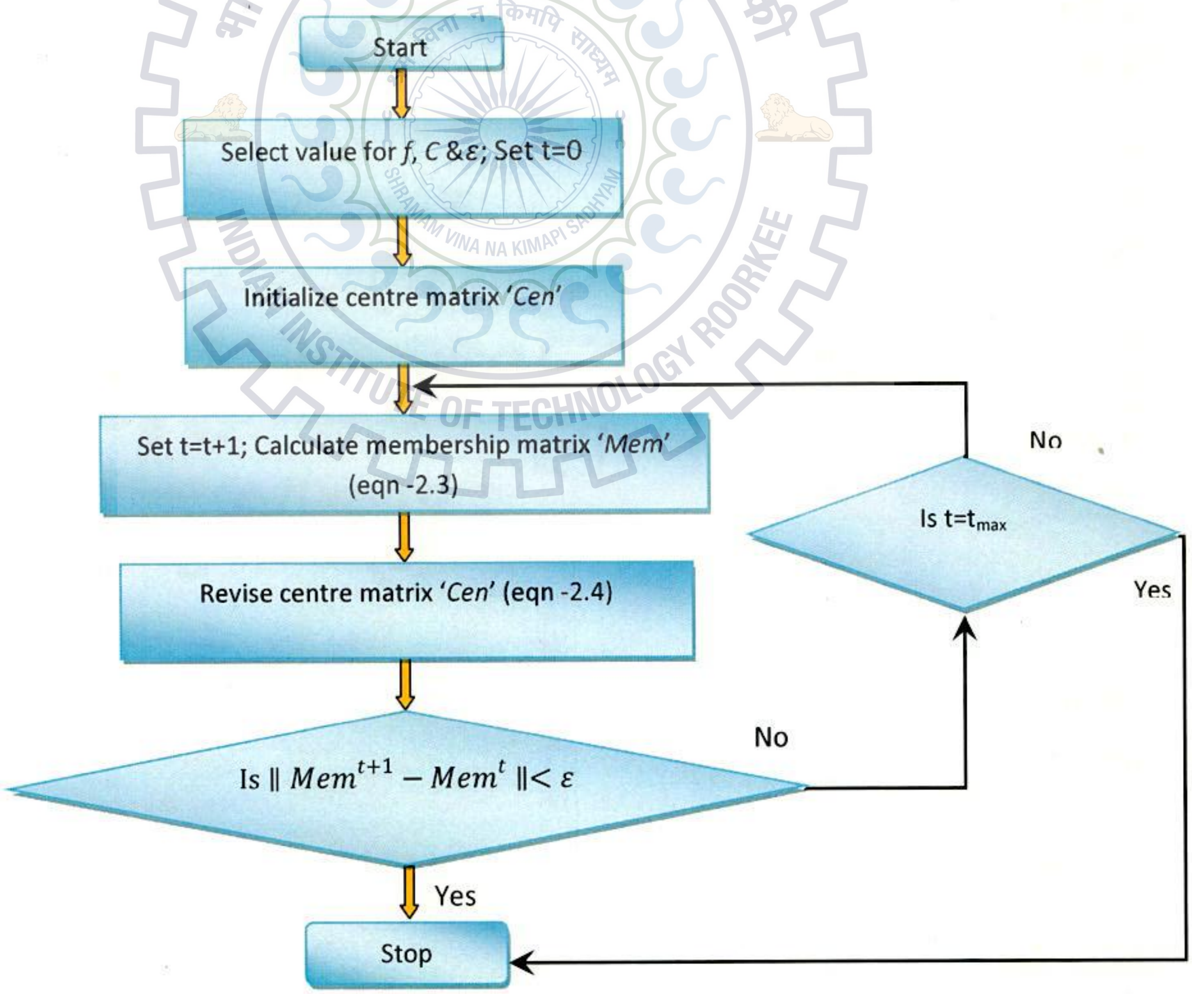

Figure 2.1: Flow Diagram For FCM Algorithm

In Fuzzy C-Means algorithm, the number of groups in which the input data is to be clustered is decided. The stopping criterion is decided. The stopping criteria can be in terms of the maximum number of steps or the minimum improvement achieved in a particular step. The centres of the clusters are chosen, albeit randomly. The fuzzification parameter is also set. Thereafter the membership values are calculated using equation number (2.3). The Euclidian distances are calculated for the data sets to decide the membership value of a particular data instance for a given group. Once the membership matrix is formed, the same is used to revise the centre matrix as per equation (2.4). Now the stopping criteria are checked. If these are not satisfied then the membership matrix and the centre matrix are iteratively updated till any of the stopping criterions is met.

Fuzzy C-Means algorithm is fast and easy to implement. It performs well for the database having data instances which are overlapping. The clusters thus formed are not very well defined and require fuzzy membership value to be awarded to every data instance. A data point can belong to more than one group with varying degree of association value. It can be seen from equation(2.1) that the membership value will remain between 0 and 1. Membership value of 0 will indicate that the data point does not belong to that particular cluster. Association value of 1 indicates that a particular data point belongs to one and only one group and with that group the membership value of that data point is 1. In between values of membership coefficient gives the probability with which a particular data point belongs to that cluster. The distance is the metric for deciding the membership coefficient. Larger the distance, lesser will be the membership coefficient and vice versa.

2.6 Application In Image Segmentation

The Fuzzy C-Mean algorithm has the input data as the image data which is representing certain features of the image. The different features could be intensity of pixels, the spatial information, texture information or could be the entropy information of the image. The number of clusters is decided. The more the number of clusters, better will be segmentation as the algorithm will be able to group the data very well based upon the finer details [10].

However, more the number of the clusters lager will be the computation time. This is the trade off one has to accomplish while deciding the number of the clusters. Very high number of clusters may lead to a situation where we have very less or zero data points associated to it with maximum degree of membership. The algorithm is then applied on the image data and the final value of membership matrix and the cluster centres is taken. The membership matrix gives the association of each data point to the given clusters. To reconstruct the segmented image maximal membership of the data point is considered and the particular data point is assigned to the cluster with which it has the maximum association or the degree of membership. Now this cluster wise data is reproduced as the segmented image.

The distance measure is based on the Euclidian distances in case of fuzzy c means algorithm. The choice of Euclidian distance is made due to the fact that this metric is more resistant to noise. The results are also affected for noise and outliers. To improve the performance several modifications have been done in FCM algorithm [11-13]. More information is added into the algorithm to achieve better results. One such example is the addition of spatial information into the algorithm to make it more resistant to the noise. Spherical data clusters can be easily segregated by this type of metric. However the same cannot be said for the more complex cluster shapes. The performance of FCM deteriorates for more general data sets. To overcome this major drawback kernel based clustering was introduced. The kernel maps the data set to a higher dimension space. The mapping to higher dimensional space increases the chances of separating data linearly and hence enables algorithms to recognise complex data cluster shapes. The kernel based algorithms are more resistant against the noise and outliers [14]. This increases the scope of applicability of the kernel based FCM algorithm to various types of data sets.

2.7 **Simulation Results**

The 128 by 128 pixel synthesized image with two pixel intensities was segmented by using FCM and KFCM algorithm for two groups. Then the white Gaussian noise with zero mean was added to this image and the image was again segmented by these two algorithms. The Gaussian kernel was used in KFCM algorithm to map the input data to the feature or the high dimensional space.

Figure 2.2: Segmented Image with 3% Gaussian Noise

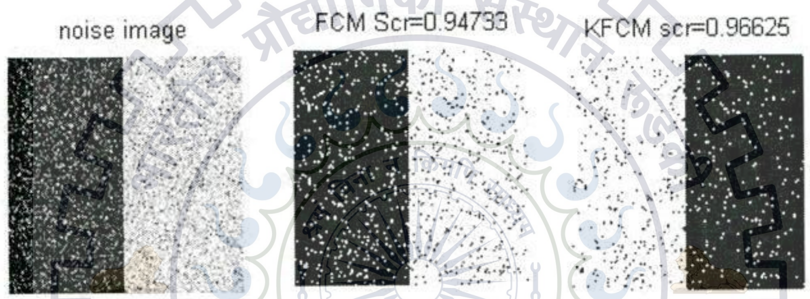

Figure 2.3: Segmented Image with 5% Gaussian Noise

The figure (2.1) and (2.2) shows better segmentation results in case of KFCM. The figure of merit is taken as the segmentation accuracy. The figure of merit (SA) in this case is calculated as

$$SA = \frac{Correctly\ classified\ pixels\ by\ algorithm}{Total\ number\ of\ pixels\ in\ image} \tag{2.5}$$

The performance plot for both these algorithms was plotted by varying Gaussian noise in the steps of 3 percent. The performance of Kernel Fuzzy C-Mean algorithm (KFCM) was found to be better in term of segmentation accuracy than Fuzzy C-Mean algorithm (FCM) under noised conditions.

2.7.1 **Analysis**

The better performance of KFCM in case of images with noise is attributed to the fact that the outliers and the noised data when mapped into the higher dimensional space can be better separated. This happens due to the fact that the non linear relations are better understood in higher dimensional space as they are linearly mapped and hence can be accurately classified by the similarity metric. The similarity metric being the Euclidean distance in this case, separates data which is more spherically clustered.

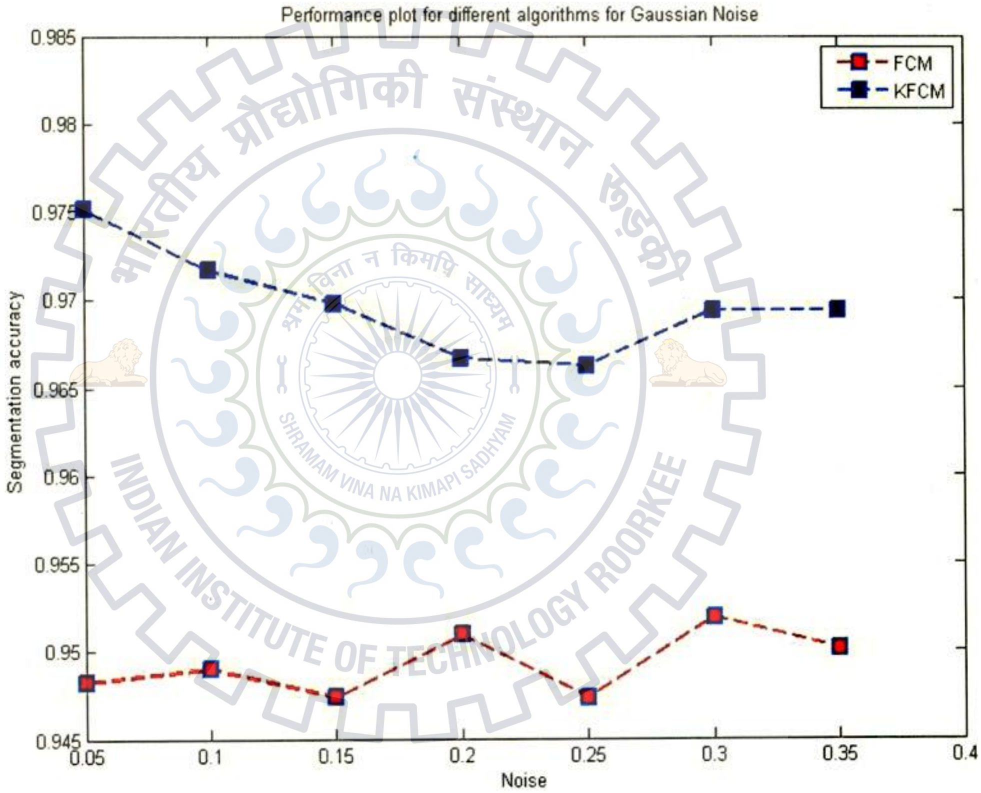

Figure 2.4: Performance Of FCM And KFCM With Varying Noise

Chapter 3

KERNEL BASED AGORITHMS

3.1 Kernel Function

The kernel functions are used to map the data set from the data space or feature space to a much higher dimension space called as kernel space or Hilbert space [15]. To achieve this we have an embedding map

$$\mathrm{M} : A \in \mathfrak{M}^n \mapsto \mathrm{M}(A) \in H \supseteq \mathfrak{M}^n$$

Where

M is the embedding map

A is the data set in a real data space $\mathfrak{M}$ with dimensions as 'n'

H is the higher dimensional space called Hilbert space

The selection for map M is based on the criteria that it should change the non linear relations of the data set to the linear ones. We can have an inner-product kernel function to calculate this complex mapping. This can be done using the so called distance kernel trick:

$$\| \mathrm{M}(A_i) - \mathrm{M}(A_j) \|^2 = \mathrm{M}(A_i).\mathrm{M}(A_i) + \mathrm{M}(A_j).\mathrm{M}(A_j) - 2\mathrm{M}(A_i)\mathrm{M}(A_j)$$

$$= K(A_i, A_i) + K(A_j, A_j) - 2K(A_i, A_j) \qquad (3.1)$$

Where

K is the kernel function

From equation (3.1), it can be seen that the complex or at times infeasible mapping from feature space to higher dimensional space can be calculated with a relative ease using only the inner product. The significant point to note is that we can now calculate the Euclidean distances in the higher dimensional space without unambiguously identifying the mapping M. Kernel function is the one which carry out this estimation.

3.1.1 **Kernel Function Definition**

A kernel function or kernel as it is referred to is the function K which satisfies

$$K(A,B) = \langle \mathrm{M}(A), \mathrm{M}(B) \rangle \qquad \forall\ A,B \in \mathfrak{M}$$

Where

K is the kernel function

M is the embedding map from the real space to higher dimensional space

A & B are the data set in a real data space $\mathfrak{M}$

To amplify the above fact we can consider a simple example. Suppose there are data points in one dimensional plane as shown in the figure (3.1). There can only be vertical lines possible to segregate the points in this case. No such line can do the division. However if we map this data to the two dimension space as $X \rightarrow (X, X^2)$, then we get the data representation which is linearly separable.

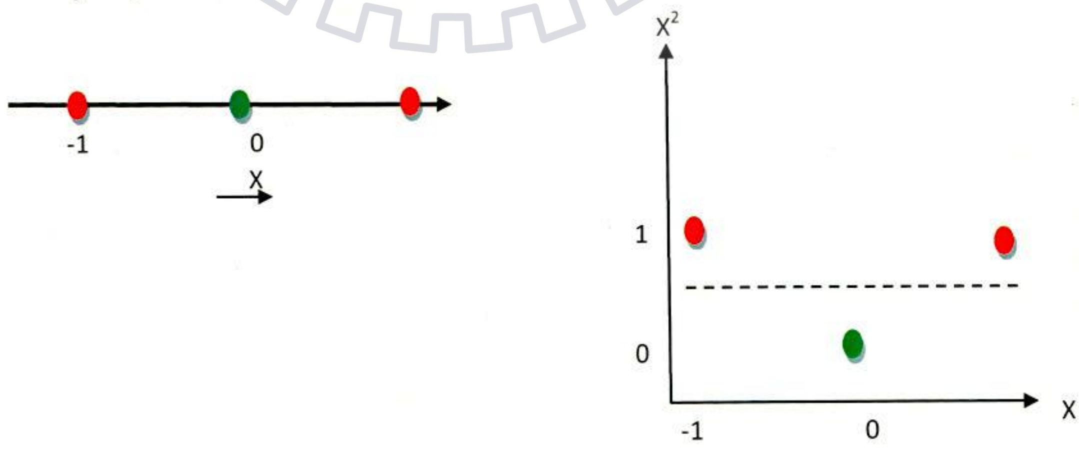

Figure 3.1: Example Of Kernel Mapping

3.2 **Kernel Trick**

Consider a 2D vector space $A \subseteq \mathfrak{M}^2$

And a mapping

$$\mathrm{M} : A = (A_1, A_2) \mapsto \mathrm{M}(A) = \left(A_1^2, A_2^2, \sqrt{2}A_1A_2\right) \in H = \mathfrak{M}^3$$

The suggested linear function in higher dimensional space will be

$$f(A) = V_{11}A_1^2 + V_{22}A_2^2 + V_{12}\sqrt{2}A_1A_2$$

The map carries data from 2-D to 3-D space in the manner such that the quadratic relations in vector space relate to linear relations in Hilbert space. The mapping with kernel trick would become

$$\langle \mathrm{M}(A), \mathrm{M}(B) \rangle = \langle \, (A_1^2, A_2^2, \sqrt{2}A_1A_2), (B_1^2 B_2^2 \, \sqrt{2}B_1B_2) \, \rangle$$

$$= A_1^2B_1^2 + A_2^2B_2^2 + 2A_1A_2B_1B_2$$

$$= (A_1B_1 + A_2B_2)^2 \quad = \langle A, B \rangle^2$$

Therefore, the kernel function becomes

$$K(A, B) = \langle A, B \rangle^2$$

Hence it can be seen that the two points can be mapped from input space to higher dimensional space without overtly calculating their mapping. This saves lot of computational power and renders relative ease of implementation of kernel functions. The kernelisation attains linearly separable hyperplane in Hilbert space and thus makes clustering algorithm more effective in spotting the random shaped data groups [16].

There are various types of kernel functions but the most commonly used kernels are

a. **Gaussian Kernel** This kernel is the most used kernel function. This is due to the fact that value of $k(A_i, A_i) = 1$. This is useful when we calculate the value of the mapping using the kernel trick.

Equation (3.1) reduces to $= 2(1 - k(A_i, A_j))$. The Gaussian kernel is given as

$$k(A,B) = e^{\left(-\|A-B\|^2/r^2\right)}$$

b. Linear kernel

$$k(A,B) = A.B$$

c. Polynomial Kernel

$$k(A,B) = (A.B + d)^p$$

Where

d is the constant

p is the degree of the function

3.3 Kernel Fuzzy Clustering

Let there be a data set such that

$$A = (a_1, a_2, \ldots\ldots\ldots\ldots\ldots a_n)\ \forall\, a_j \in D \subseteq \mathfrak{M}^d$$

Where

D is the vector space

A is the data set with 'n' elements in a vector data space

$\mathfrak{M}$ is a data space with dimensions as 'd'

The kernel based algorithm will project this data from the data space to a higher dimensional space. Let this mapping be $\mathrm{M}: D \mapsto H$. Where H is the kernel space. This mapping will enable data to be mapped to new space as simpler structures. In this case the data groups will be more sphere-shaped. These can then be easily detected by the Euclidian distance based algorithms. The function which will be minimised is given as

$$CF_{KFCM} = \sum_{i=1}^{C}\sum_{j=1}^{n} Mem_{ij}^{f} \parallel \mathrm{M}(a_j) - O_i \parallel^2 \tag{3.2}$$

Where

C   is the number of groups/clusters

O is the initial sample matrix

The membership matrix is iteratively updated

$$Mem_{ij} = \frac{1}{\sum_{h=1}^{C}(\frac{d\mathrm{M}_{\mathrm{ij}}^2}{\mathrm{dM}_{\mathrm{hj}}^2})^{\frac{1}{\mathrm{f}-1}}} \tag{3.3}$$

Where

$$\mathrm{dM}_{ij}^2 = k(a_j, a_j) - \frac{2\sum_{h=1}^{n} Mem_{ih}^{f} k(a_h, a_j)}{\sum_{h=1}^{n} Mem_{ih}^{f}} + \frac{\sum_{h=1}^{n}\sum_{l=1}^{n} Mem_{ih}^{f} Mem_{il}^{f}\, k(a_h a_l)}{(\sum_{h=1}^{n} Mem_{ih}^{f})^2} \tag{3.4}$$

Introduction of kernels removed the drawbacks of the Fuzzy C-Mean clustering. It facilitated recognition of generalised data groups or clusters and made KFCM more effective against noise and outliers [17-20]. These advantages come with increased computational cost. However the reduced computational cost and the ease of implementation are achieved by use of kernel trick. Cover's theorem forms the basis of all kernel based clustering [21].

The limitation of kernel based clustering algorithms is the problem of choosing the best kernel function for the assigned task [22-24]. The performance of entire clustering algorithm depends on kernel function and the value of various parameters associated with this function. To overcome this limitation of kernel based clustering, the combination of multiple kernels is used. This combination of kernels overcomes unrelated, unnecessary, erratic and unproductive features of single kernel [25, 26].

3.4 **Multiple Kernel Fuzzy C Means Algorithm**

The multiple kernel fuzzy c means algorithm for fuzzy clustering attempts to have good clustering of the data instances into significant clusters. It also aims at attaining the feature map which is optimally brought out by the kernel function. This map should be brought out in as far as possible unsupervised way. Multiple Kernel Fuzzy C-Mean algorithm (MKFCM) is a generalization of the Kernel Based Fuzzy C-Means algorithm (KFCM). It attains the best convex combination of homogenous kernels with respect to each cluster by using an optimization criterion. The MKFCM algorithm thus achieves the benefits of kernel capabilities without the effort of kernel selection and its value tuning. The single kernel may not be suitable for a given data set as it may not highlight its features in the best possible manner. Therefore it is better to have more kernels and have an algorithm to decide the best combination of the given kernels[27]. There can be two approaches to have such a combination of kernels.

a. Different kernels may represent different features of the dataset. These kernels are then combined with the help of an algorithm and best combination of these kernels is picked.

b. Various kernels are used for inputs coming from heterogeneous sources. The combining algorithm is used and the best combination of kernels is achieved. This type of combination of kernel can be used to fuse information from various sources [28]. Various types of combinations of information are

I. Early Amalgamation

In this merger, the information from sources is concatenated and fed to the algorithm.

II. Intermediate Amalgamation

In this case the input from the sources is kernel mapped and the combined.

I. Late Amalgamation

In this case the different information is first given to the classifier and the verdicts are then merged by the algorithm.

3.4.1 **Classification of Kernel Combining Algorithms**

There are various algorithms used for kernel combining. The algorithm used for kernel combining can be broadly classified in accordance with certain properties [29]. These are

a. Training Method

I. One step:- Function features and parameters of algorithm are estimated in one iteration.

II. Two Step:- in this case, algorithms parameters are found out while other parameters are fixed. In second step features of function are calculated and algorithm parameters are fixed.

b. Learning Procedure

The algorithms for kernel combining employ various procedures for automatically finding the optimal solution. These are

I. Fixed:- Functions do not have additive or multiplicative functions of kernels.

II. Enhancement approach:- A new kernel is added till performance stops to enhance.

III. Optimization:- Kernel function is a combination of various parameters. The variables are learned by optimizing certain parameter.

IV. Heuristic:- Kernel function is a combination of various variables. These variables are calculated from each kernel in disjoint manner.

c. Target Function

Different functions are optimized while selecting kernel combination algorithm. These are classified as

I. Resemblance based:- The similarity between best possible kernel matrix and combined matrix is found out.

II. Error based:- The combination algorithm tries to minimize the error based term. This error term is the measure of the system functioning.

d. **Computational Complexity**

This depends primarily on the training technique being used by the algorithm.

e. **Functional Form**

There are various methods to combine the kernels. The broad classification of amalgamation is

I. Linear :- This is an easy to implement method. In this case the emphasized or un emphasized sum is taken.

II. Nonlinear :- In this case nonlinear mixture of kernels is considered. These can be power, exponential or multiplication or combination of these.

III. Input Reliant :- kernel emphasis factors ae assigned to every input data point. This enables to establish distribution in data and learn optimum kernel kernel amalgamation for each distribution.

3.5 **Mercer Kernel Properties**

The combination of the kernels and formation of new kernels are governed by the Mercer kernel properties [30, 31]. Multiple Kernel Fuzzy C-Mean (MKFCM) algorithms use certain Mercer kernel properties. The essential properties are listed before the discussion about the algorithm itself. The properties are:

Theorem: If k_1 and k_2 be kernels functions in vector space $A \times A\ \forall\ A \subseteq \mathbb{R}^d$ and k_3 be a kernel function over $\mathbb{R}^q \times \mathbb{R}^q$ (where'd' and 'q' are the dimensions of the vector). And let the mapping be $\mathrm{M}: A \longrightarrow \mathbb{R}^q$.

Then

a. $k(\boldsymbol{A}, \boldsymbol{B}) = k_1(\boldsymbol{A}, \boldsymbol{B}) + k_2(\boldsymbol{A}, \boldsymbol{B})$ is also a kernel.
b. $k(\boldsymbol{A}, \boldsymbol{B}) = \alpha k_1(\boldsymbol{A}, \boldsymbol{B})$ is a kernel, when $\alpha > 0$.
c. $k(\boldsymbol{A}, \boldsymbol{B}) = k_1(\boldsymbol{A}, \boldsymbol{B}) k_2(\boldsymbol{A}, \boldsymbol{B})$ is also a kernel.
d. $k(\boldsymbol{A}, \boldsymbol{B}) = k_3(\mathrm{M}(\boldsymbol{A}), \mathrm{M}(\boldsymbol{B}))$ is a kernel.

The goal of MKFCM is minimization of the same cost function as the single fixed KFCM, The composite kernel k_{com} is replaced with the single kernel during the calculation process [32]. k_{com} is defined as a combination of multiple kernels using Mercer properties introduced in above theorem. For instance, two straightforward composite kernels could be $k_{com} = k_1 + \alpha k_2$ and $k_{com} = k_1 k_2$. Given that k_1 and k_2 are kernels in data space. Properties a, b, and c in above theorem conclude that the combination of the kernel will also be a kernel. It may be noted that the composite kernel k_{com} is also a Mercer kernel. In the naming convention of MKFCM, the Multiple Kernel Fuzzy C Means algorithm will be called as MKFCM_K if the first of the samples are taken in the kernel space. The algorithm is called as MKFCM_F, if the prototypes are supposed to be in the data space or the feature space [26]. If the composite kernel has small number of kernels as its constituents then trial and error method and past experience can help in kernel selection for the given task. The various parameters of the kernels can also be estimated and tuned using the same technique. However this is not a very viable solution when the number of constituents is more. In such cases a learning algorithm is used to find out the optimal combination of the kernels and fine tune their parameters.

The Multiple Kernel Fuzzy C-Mean algorithm (MKFCM) also uses the same objective function as shown in equation (3.2) [33, 34]. The only difference is that the mapping function $'M'$ is changed to a composite mapping function M_{com}. The equation thus becomes

$$CF_{MKFCM} = \sum_{i=1}^{C}\sum_{j=1}^{n} Mem_{ij}^{f} \parallel M_{com}(a_j) - O_i \parallel^2 \qquad (3.5)$$

The composite mapping function M_{com} is obtained from the combination of the kernels. Similarly the membership matrix is also acquired as in case of KFCM with the difference that the composite kernel is considered in case of Multiple Kernel Fuzzy C Mean algorithm.

3.6 Linear Kernel Combination

The most commonly used combination of the kernels is the linear combination. The emphasis of various kernels is determined automatically. For this purpose learning procedure is used. The kernel is given as

$$K_{lin} = G_1^w K_1 + G_2^w K_2 + \dots\dots\dots\dots\dots\dots + G_l^w K_l \qquad (3.6)$$

Where

w is coefficient akin to f

G is the emphasis or weighing factor

The emphasis factors are guided by the rule that the sum of all the factors is unity and all the weighing factors are non zero.

The cost function to minimise in this case is also similar to the function given in equation number (3.5). The only change being that the mapping function changes from $\mathrm{M_{com}}$ to $\mathrm{M_{lin}}$. The membership matrix is also updated as per equation (3.3) and equation (3.4). In this case also the change is that linear combination of kernel functions is used to evaluate the membership matrix. If we take the restricting rule of the emphasis factor into consideration and put the Lagrange term in the cost function then the cost function will be given as

$$CF_{MKFCM} = \sum_{i=1}^{C}\sum_{j=1}^{n} Mem_{ij}^{f} \parallel \mathrm{M_{lin}}(a_j) - O_i \parallel^2 + \eta\left(1 - \sum_{i=1}^{l} G_i\right) \qquad (3.7)$$

From above equation we formulate the revising rule for the emphasis factor.

$$\frac{\partial CF_{MKFCM}}{\partial G_i} = 0 \implies G_i = \frac{1}{\sum_{h=1}^{n}\left(\frac{CF_i}{CF_h}\right)^{\frac{1}{w-1}}} \qquad (3.8)$$

The cost function CF_h is give by the equation (3.5), the only change is that mapping function is changed to $\mathrm{M_h}$. Thus the equation becomes

$$CF_h = \sum_{i=1}^{C}\sum_{j=1}^{n} Mem_{ij}^{f} \parallel \mathrm{M_h}(a_j) - O_i \parallel^2 \quad (h = 1,2,\ldots,l) \qquad (3.9)$$

Putting the value of mapping function from equation (3.6), we get

$$\parallel \mathrm{M_h}(a_j) - O_i \parallel^2 = k_h(a_j, a_j) - \frac{2\sum_{l=1}^{n} Mem_{il}^{f} k_h(a_l, a_j)}{\sum_{l=1}^{n} Mem_{il}^{f}} + P_m \qquad (3.10)$$

Where

$$P_m = \frac{\sum_{m=1}^{n}\sum_{l=1}^{n} Mem_{im}^{f} Mem_{il}^{f}\, k_h(a_m, a_l)}{(\sum_{m=1}^{n} Mem_{im}^{f})^2}$$

This variant of Multiple Kernel FCM algorithm is called as Linear Combined Multiple Kernel Fuzzy C-Mean algorithm [26]. The greatest advantage of this variant of algorithm is its ability to optimise the emphasis factors for the kernels. This reduces the redundant and ineffective effects of Kernels. In this type of algorithm, generally we describe the prototypes in Hilbert space. This is due the fact that it is cumbersome to generate automatic updating rules for the emphasis factor in the data space as the linear amalgamation of kernels may not be a Gaussian. It therefore will have more complicated calculation and more time consuming implementation.

There are many other variants of Multiple Kernel Fuzzy C-Mean algorithm. Many of the improved kernel based algorithms were proved to a case of MKFCM by L Chen et.al [26]. If we combine the features of the image and then apply a kernel mapping on it, then this variant is called as the Direct Kernel Fuzzy C-Mean algorithm. If additional term is added to the cost function then it is called as the Additional Kernel Fuzzy C-Mean algorithm. The use of method for feature extraction is also sometime reflected in the name of the algorithm. The space in which the prototypes are selected can also be reflected in the naming convention of the algorithms.

3.7 **Advantages of MKFCM**

The advantages of Multiple Kernel Fuzzy C-Mean algorithm (MKFCM) are the programmed calculation of the emphasis factors of the kernels. The unwanted and surplus features of the kernels are removed or suppressed to a large extent by assigning these kernels a very small emphasis factor which is very close to zero. The kernels which contribute significantly are awarded a large value of emphasis factor which is close to unity. Its inherent nature to automatically find the optimal emphasis factor gives this algorithm its greatest advantage. This algorithm can be utilised to fuse different information acquired from various sources in the Hilbert space. In case of image segmentation, we can acquire different features of the image from various sources and then use them using individual kernel functions for each feature. These features are then fused by using composite kernel function. This benefit increases the utility of the algorithm in fields like medical imaging, satellite imagery, unmanned aerial vehicle imagery and aerial reconnaissance. In these fields, the information from various sources is fused to have better image segmentation results.

Chapter 4

MULTIPLE KERNELS BASED SEGMENTATION

The Multiple Kernel Based Fuzzy C-Mean algorithms were introduced in the field of image segmentation by L Chen et.al [26]. The improved variants of kernel based Fuzzy C-Mean were proved to be a special cases of Multiple Kernel Based Fuzzy C-Mean algorithms (MKFCM). The linear combined Fuzzy C-Mean algorithm is used in image segmentation to automatically decide the emphasis of various kernels in a linear combination of them. This algorithm was discussed in detail in the previous chapter. It used derivative of cost function with respect to the emphasis factor and equated it to zero while using the constraint that the sum of all the weights is unity (equation (3.9)). This equation gave us the weight updating rule. The factor w akin to fuzzification factor f was introduced. This variable also needed an optimum value for the best operation of the algorithm. We propose another weight optimisation technique to eliminate this variable and calculate the weights automatically.

4.1 Weight Optimisation

Consider a linear ensemble of emphasis factors given as

$$K_{lin} = G_1K_1 + G_2K_2 + \dots\dots\dots\dots\dots + G_lK_l \tag{4.1}$$

The given ensemble is different from the previous case in the sense that the variable w is not considered. The constraint is that the sum of the emphasis factors or weights is unity.

Now to find out optimal weights, let us consider the value of distance between data point (i) and Cluster centre (c)

$$Dist_{ic}^2 = (\mathrm{M}(A_i) - Cen_c)^T\,(\mathrm{M}(A_i) - Cen_c)$$

$$= \sum_{k=1}^{l} G_k^2 k_k(A_i, A_i) - 2\sum_{j=1}^{n}\sum_{k=1}^{l} Mem_{jc} G_k^2 k_k(A_i, A_j) + P_h \tag{4.2}$$

Where

$$P_h = \sum_{j=1}^{n}\sum_{h=1}^{n}\sum_{k=1}^{l} Mem_{jc}Mem_{hc}G_k^2 k_k(A_j, A_h)$$

As the value of the membership coefficient is fixed and values of kernel functions can be calculated, the above equation reduces to

$$Dist_{ic}^2 = \sum_{k=1}^{l} \gamma_{ick} G_k^2 \quad (4.3)$$

Where

$$\gamma_{ick} = k_k(A_i, A_i) - 2\sum_{j=1}^{n} Mem_{jc}k_k(A_i, A_j) + \sum_{j=1}^{n}\sum_{h}^{n} Mem_{jc}Mem_{hc}k_k(A_j, A_h) \quad (4.4)$$

If we substitute this value in the cost function equation and then put the constraints

a. Sum of all the emphasis factors is unity.
b. All the emphasis factors are non zero.
c. Sum of all the membership values of a data point is unity
d. All the membership values are greater than unity

We get the equation as

$$CF_G = \sum_{k=1}^{l} \delta_k G_k^2$$

Where

$$\delta_k = \sum_{i=1}^{n}\sum_{c=1}^{C} Mem_{ic}^f \gamma_{ic} \quad (4.5)$$

Solving above equation as constrained optimisation problem, we get

$$J(G,\lambda) = \sum_{k=1}^{l} \delta_k G_k^2 - 2\lambda\left(\sum_{k=1}^{l} G_k - 1\right)$$

To solve we differentiate with respect to emphasis factor and equating it to zero we get

$$G_k = \frac{\lambda}{\delta_k}$$

Applying the constraint that sum of all the emphasis factors is unity and solving the above equation we get

$$\lambda = {}^{1}\Big/\left(\frac{1}{\delta_1} + \frac{1}{\delta_2} + \ldots\ldots\ldots + \frac{1}{\delta_l}\right)$$

And

$$G_k = \frac{\frac{1}{\delta_k}}{\left(\frac{1}{\delta_1} + \frac{1}{\delta_2} + \ldots\ldots\ldots + \frac{1}{\delta_l}\right)} \tag{4.6}$$

4.2 **Steps For MKFCM**

The various steps for implementing the algorithm are as shown below.

a. Decide the number of clusters, the fuzzification parameter and the kernel functions. Decide the stopping criteria.

b. Initialize the membership matrix.

c. Calculate the value for γ_{ick} (from equation 4.4).

d. Find out values for δ_k (from equation (4.5)).

e. Calculate values of emphasis factor using equation (4.6).

f. Calculate the distance from equation (4.3).

g. Update membership matrix by using equation (3.3).

h. Check if the improvement in membership matrix is less than the stopping criteria. If yes then stop or repeat step ‘c’ to ‘h’ again.

The multiple kernel based FCM algorithm using this weight optimisation technique is implemented. The Gaussian kernel is considered for the mapping into the Hilbert space. The algorithm is evaluated against three set of data. These are

a. **Synthetic Data**

The artificial data is used to check the functioning of the algorithm. The algorithm is tested against a ten feature data and grouped it into 8 groups. The data set had total of one hundred sixty data instances.

b. **Synthetic Images**

The algorithm was tested in the image processing domain with the help of simple images. The images used were generated with three intensity pixels. The advantage it offers is that the ground truth is available since the image is generated locally. The testing can be comprehended easily as the images are simple and thus fault rectification if any becomes easy.

c. **Medical images**

The medical images were treated with the algorithm and the segmentation results were analyzed. Medical resonance image was used to analyzed the segmentation results. The MRI and its ground truth were acquired from brain image repository [35].

4.3 Simulation Results

The given weight optimisation technique was tested for the three intensity, 64 by 64 pixel image. The Gaussian kernel was used as the kernel for mapping various features of the image to the higher dimensional space. The measure of performance was chosen to be the segmentation accuracy. The segmentation accuracy was defined as given in equation (2.5). Three different features of the image were considered. These were the pixel intensity of the image, the spatial information and the standard variance of the intensities of the image.

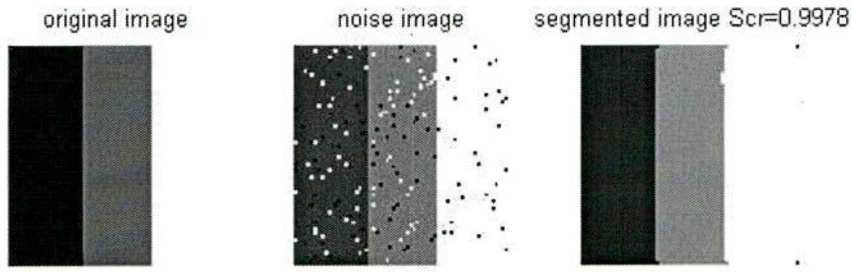

Figure 4.1 : Segmentation Results With 5% Salt And Pepper Noise

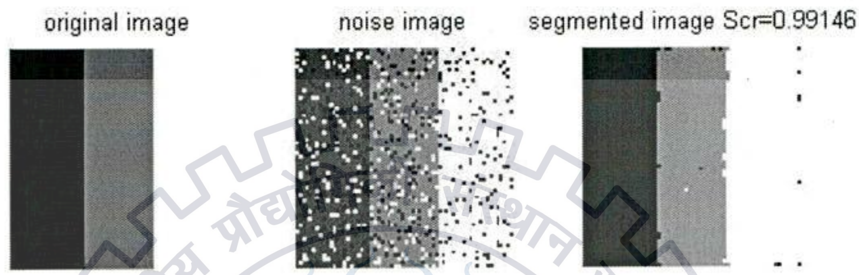

Figure 4.2 : Segmentation Results With 20% Salt And Pepper Noise

It can be seen from the above results that the performance of MKFCM is very good under the noise conditions. In case of the Gaussian noise the performance of the algorithm was checked. The results were as follows

Figure 4.3 : Segmentation Results With 10% Gaussian Noise

4.3.1 **Analysis**

The performance of the MKFCM is improved because of the combination of the different features of the image help the algorithm to classify the image pixels correctly. The selections of features which are given as input are of critical importance. In case of salt and pepper noise the spatial information of image extracted as filtered intensity of image was very significant. The significance of the given features is automatically decided by the algorithm and is reflected as the magnitude of the emphasis factor. The magnitude of the emphasis factor for 10% Gaussian noise and 10 % Gaussian noise are reflected in the table (4.1).

Features	Pixel Intensity	Spatial Information (Mean)	Standard variance of the intensities
For 10% Gaussian Noise	0.11896	0.65451	0.22654
For 10% Salt & Pepper Noise	0.84533	0.11161	0.04307

Table 4.1 : Variation Of Weights with Mean Filter

Features	Pixel Intensity	Spatial Information (Median)	Standard variance of the intensities
For 10% Gaussian Noise	0.08446	0.09863	0.81691
For 10% Salt & Pepper Noise	0.00786	0.98909	0.00305

Table 4.2 : Variation Of Weights With Median Filter

It can be seen that the filtered intensity of the pixel was considered as the most important feature among the entire ensemble. When the mean filtered value of intensities is considered as spatial information the spatial information gets the maximum weight assigned to it (Table 4.1). Similarly when median filtered value of pixel intensities is considered as spatial information then this feature acquires significance in case salt and pepper noise is present (Table 4.2).

4.3.2 **Medical Image**

To check the performance of the algorithm for medical images a magnetic resonance image was considered. The image and the ground truth images were acquired from brainweb MRI repository [35]. The image considered is a MR phantom of slice thickness of 1 mm. The image was segmented into three clusters to obtain segments for white matter, Cerebrospinal Fluid and gray matter.

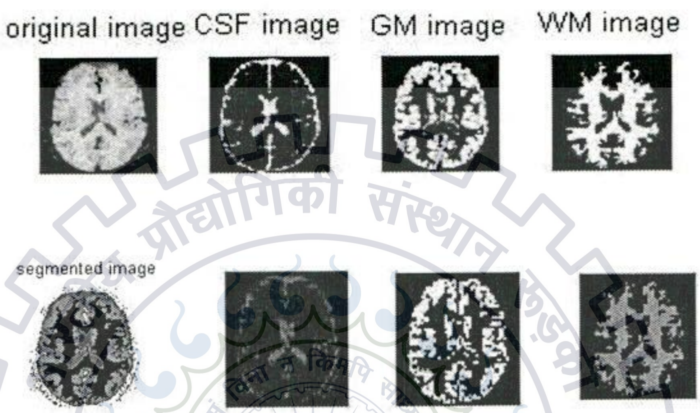

Figure 4.4 : Segmentation Results For Subject 1

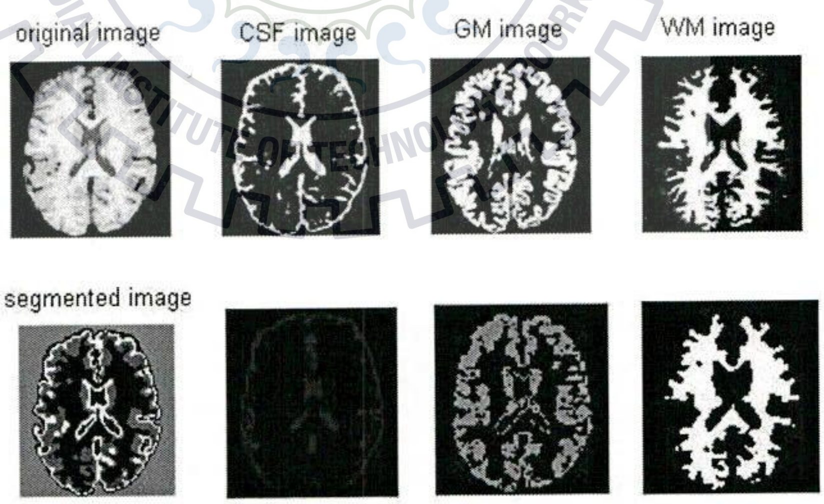

Figure 4.5 : Segmentation results for Subject 2

The segmentation accuracy was taken as the figure of the merit. The segmentation accuracy for this case was given as

$$SA = \frac{Img_{Seg} \cap GT}{Img_{Seg} \cup GT}$$

Where

SA is the segmentation accuracy

Img_{Seg} is the group of pixels found to belong to a class after segmentation

GT is the ground truth for that class of image

The segmentation accuracy achieved were

	Cerebrospinal Fluid SA_{CSF}	Gray Matter SA_{GM}	White Matter SA_{WM}
Subject 1	0.8274	0.7446	0.8618
Subject 2	0.7918	0.8326	0.7264

Table 4.3 : Segmentation accuracy

These results were taken for 10 readings and the mean value of these were considered after ignoring the maximum and the minimum values achieved. The values of segmentation accuracy obtained by the LKFCM algorithm proposed by L Chen et.al [26] were given as 0.78, 0.87 and 0.70 for CSF,GM and White Matter respectively. It can be seen that the results were comparable to what were obtained for the case of LKFCM.

4.3.3 **Analysis**

The results obtained are comparable to the one calculated by the LKFCM algorithm proposed by L Chen et.al [26]. The marginal increase may be due to the fact that the proposed method eliminates the fuzzifier like variable w, whose optimum value and its impact on the entire emphasis factor ensemble may affect the results.

Chapter 5

PARAMETER OPTIMISATION

5.1 Fuzzification Parameter

The Fuzzy C-Mean based technique was introduced by Bezdek by introducing the fuzzification parameter f. This parameter gave the fuzzy nature to the algorithm. The value of this parameter is of critical importance for the performance of the algorithm. An attempt was done to find the optimum values of fuzzifier parameter for all sets of data which were discussed in the previous chapter.

a. For Synthetic Data Set

The artificial data sets were obtained from UCI machine learning repository [36]. The data set used is representing ten features and is divided in to eight groups. The data set has total of 160 data instances.

I. Performance Of Algorithm

The performance plot for the FCM and the MKFCM was plotted. The Normalized Mutual Information was used as the measure of the merit.

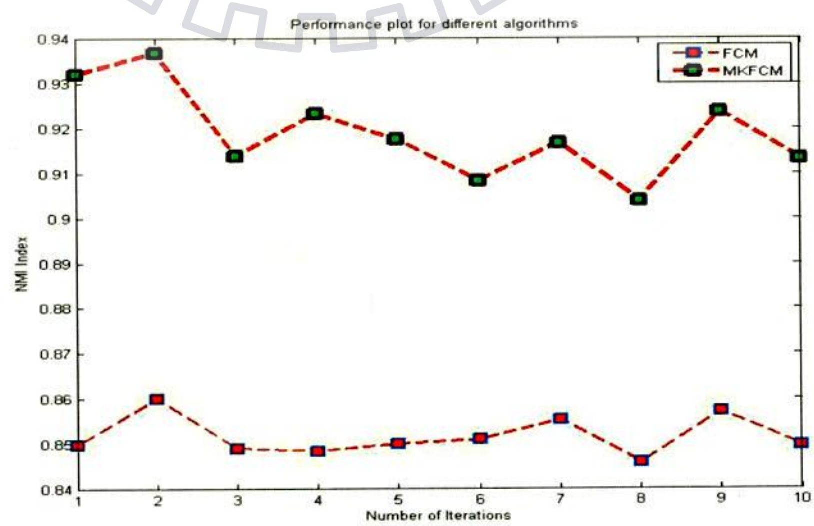

Figure 5.1 : Performance Of MKFCM

5.1.2 Analysis

The performance of the algorithm enhances by addition of multiple kernels. These kernels are combined effectively to produce a better clustering. The unwanted features of a kernel are suppressed by the automatic selection of the kernel weights. The performance of MKFCM is considerably better than that of FCM.

5.2 Performance for the synthetic images

The performance of the algorithm was measured for the three pixel intensity synthesized images of size 64 by 64 pixels. Gaussian kernels were selected to map the data from input vector space to the higher space. The pixel intensities, the spatial pixel information and the standard variance of pixel intensities are considered as the input features. The value of fuzzification parameter was varied and the graph was plotted.

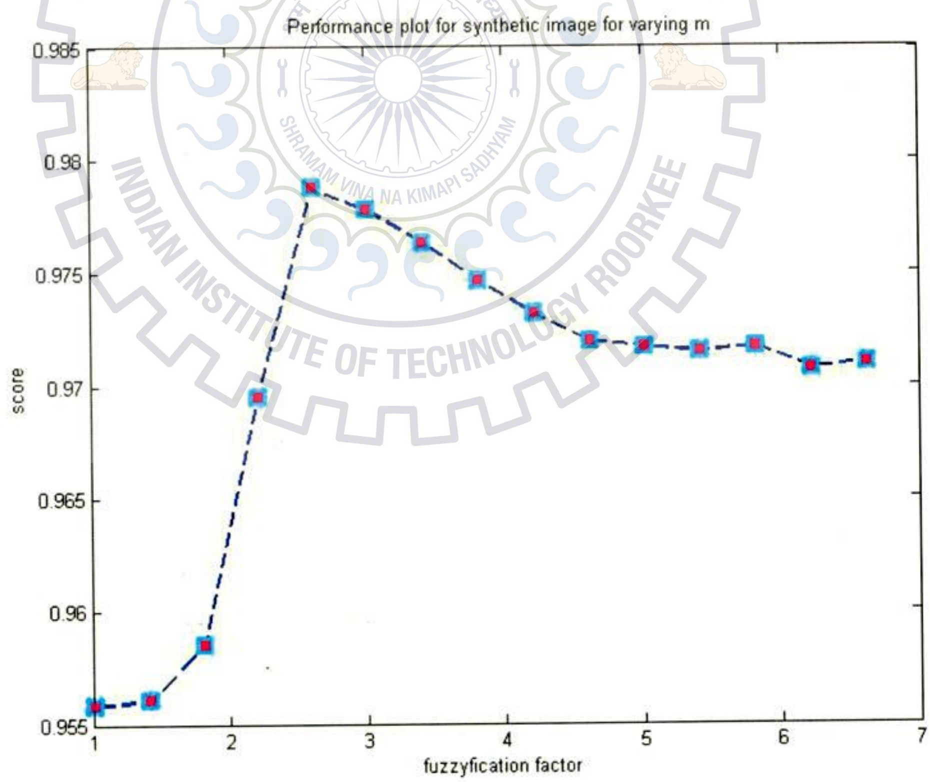

Figure 5.2: Performance Plot Of MKFCM For Synthetic Image

5.2.1 **Analysis**

The performance of algorithm varies with the variation of the fuzzifier factor f. The performance in this case increases with the value of f and then starts to decrease thereafter. The other factor which we need to consider while selecting the fuzzifier is that as we increase the value of f, the total number of iterations taken for convergences by the algorithm increases. This in turn adds to computational time. Therefore it is suggested that we should restrict to the smaller possible value of the fuzzifier. The graph for number of iterations taken by the algorithm to converge and the value of fuzzifier is plotted. It can be seen that the number of iteration increases with the increasing value of f.

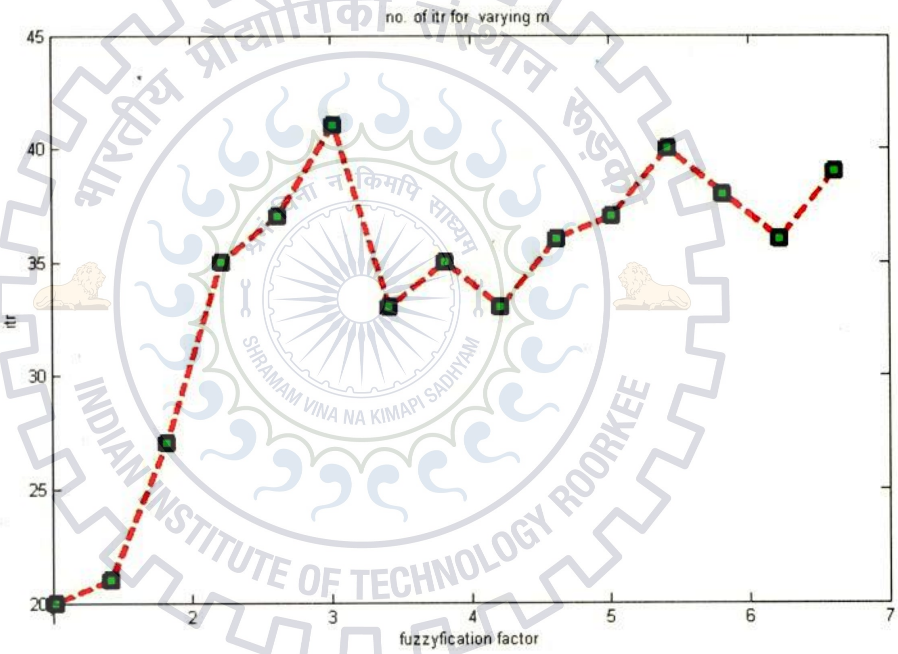

Figure 5.3 : Convergence For Variation in f

The third factor which we need to consider while deciding the value of fuzzification factor is that for the crisp clustering (f = 1) the membership values are either one or zero. This shows that a particular pixel belongs to a particular cluster. As the value of f increases these association values start to become smaller for a particular group and when the value for f would reach infinity then all the values of association would become reciprocal of the total number of the centres. The reduced values of membership function make the decision difficult. Therefore it is suggested that the smallest possible value of fuzzifier f be decided based on the optimum performance of the algorithm for given input data set.

Chapter 6

CONCLUSION

6.1 Future Prospective

In future, the efforts can be done in following areas based on the work done in this thesis.

a. The work can be extended for choice of basic kernels for the given image types. This would facilitate LKFCM algorithm to have no undesirable kernels. This will further increase the performance of the MKFCM algorithm.

b. The efforts can be made to consider non linear ensembles for automatically updating the kernels weights.

c. Alternative weight optimization techniques can be contemplated to reduce the computational complexity of the algorithm.

6.2 Conclusion

Fuzzy C-Mean technique for clustering has been effectively used since it was proposed by Bezdek. This is an easy to execute and effective method of clustering. Its use for complex data clustering like image processing was not a very popular option till recently. However with use of kernel functions the more general clusters were also effectively clustered. Usage of kernel for the mapping presented another problem of selecting the optimum kernel. This problem of selecting optimum kernels and rendering their futile features ineffective was overcome by combining multiple kernels. This combination of kernels is used to map various features of input data before clustering. These algorithms are called as Multiple Kernel Fuzzy C-Mean algorithms. The MKFCM gives us a great tool to not only cluster the data but also a great platform to combine information. This information can come from same sensor or from different sensors. The use of MKFCM in image segmentation gives us a great tool to segment images with noise and the images which are acquired from multiple sources.

In this work the Fuzzy C-Mean technique for image segmentation has been brought out in detail. The various algorithms which are available were presented with their advantages and disadvantages. For MKFCM, the value of various parameters involved in the algorithm play a critical role in its performance. The weight optimisation method suggested in chapter four reduces one such variable and marginally improves performance. The fuzziness factor f which is the bases of the algorithm has different optimum value for different input data. However it is always a better option to select a lower possible value of this factor. With increase in computational power, we can effectively implement "Fuzzy C-Mean Technique For Image Segmentation". It can be used in various applications like satellite imagery, aerial reconnaissance, medical imaging and Biometrics.

BIBLIOGRAPHY

1. R. C. Gonzalez, R. E. Woods, and S. L. Eddins, "Digital Image Processing Using MATLAB". Upper Saddle River, NJ: Prentice-Hall, 2004.

2. Nikhil R. Pal and Sankar K. Pal, "A Review On Image Segmentation Techniques", Pattern Recognition, Vol. 26, No. 9, pp. 1277-1294, 1993.

3. P.G. Peter Ho , "Image Segmentation," Intech Publishing, Croatia, 2011

4. S Sapna Varshney, Navin Rajpa and Ravindar Purwar, "Comparative Study of Image Segmentation Techniques and Object Matching using Segmentation", International Conference on Methods and Models in ComputerScience,2009, 636-642 .

5. Y. Xia, D. G. Feng, T. J. Wang, R. C. Zhao, and Y. N. Zhang, "Image Segmentation By Clustering Of Spatial Patterns," Pattern Recognition Letter, vol. 28, no. 12, pp. 1548–1555, Sep. 1, 2007.

6. James C. Bezdek, James Keller, Raghu Krisnapuram and Nikhil R. Pal," Fuzzy Models And Algorithms For Pattern Recognition And Image Processing", Springer-USA 2005.

7. F. Hoppner, F. Klauonn, R. Kruse, and T.Runale, "Fuzzy Cluster Analysis- Methods for classification, Data Analysis and Image Recognition," Willey Publication, N.Y, 2000.

8. Sadaaki Miyamoto, Hidetomo Ichihashi and Katsuhiro Honda," Algorithms for Fuzzy Clustering", Springer-Verlag Berlin Heidelberg, 2008.

9. L. Zadeh, "Fuzzy sets," Information Control, vol. 8, pp. 338-353, 1965.

10. H. Zhou, G. Schaefer, and C. Shi, "Fuzzy C-Means Techniques for Medical Image Segmentation," Springer-Verlag Berlin Heidelberg , 257-271, 2009.

11. Weiling Cai, Songcan Chen and Daoqiang Zhang "Fast and Robust Fuzzy C-Means Clustering Algorithms Incorporating Local Information for Image Segmentation" Pattern Recognition, vol 40, no 3, pp825-838, Mar2007.

12. M. Ameer Ali, G. Karmakar and L.S. Dooley, "Fuzzy image segmentation of Generic Shaped Clusters", IEEE Int. Conf. on Image Processing, pp.1202-1205, 2005.

13. Masulli F and Schenone A, "A Fuzzy Clustering Based Segmentation System As Support To Diagnosis In Medical Imaging", Artif Intell Med, vol. 16, 29–47, 1999.

14. Daniel Graves and Witold Pedrycz, "Kernel-Based Fuzzy Clustering And Fuzzy Clustering: A Comparative Experimental Study" Fuzzy Sets and Systems 161 (2010) 522–543.

15. J. Shawe Taylor and N. Cristianini, Kernel Methods for Pattern Analysis. Cambridge, U.K.: Cambridge University Press, 2004.

16. Jacek Kawa and Ewa Pietka, Member, IEEE, "Kernelized Fuzzy C-Means Method in Fast Segmentation of Demyelination Plaques in Multiple Sclerosis", Proceedings of the 29th Annual International Conference of the IEEE, August 23-26, 2007 472-481.

17. S.R. Kannan, S.Ramathilagam, A.Sathya and R.Pandiyarajan, "Effective Fuzzy C-Means Based Kernel Function In Segmenting Medical Images", Computers in Biology and Medicine 40 (2010) pp572–579.

18. Huaxiang Zhang and Jing Lu, "Semi-supervised fuzzy clustering: A kernel-based approach", Knowledge-Based Systems 22 (2009) 477–481.

19. E.A. Zanaty, "Determining The Number Of Clusters For Kernelized Fuzzy C-Means Algorithms For Automatic Medical Image Segmentation," Egyptian Informatics Journal, vol.13, 39-58, 2012.

20. R. J. He, S. Datta, B. R. Sajja, and P. A. Narayana, "Generalized Fuzzy Clustering For Segmentation Of Multi – Spectral Magnetic Resonance Images," Comput. Med. Imaging Graph., vol. 32, no.5, 353-366, 2008.

21. D. Q. Zhang and S. C. Chen, "A Novel Kernelized Fuzzy C-Means Algorithm With Application In Medical Image Segmentation," Artif. Intell. Med.,vol. 32, no. 1,Sep. 2004, pp. 37–50.

22. G. Padmavathi, M Muthukumar and S Thakur, "Non Linear Image Segmentation Using Fuzzy C Means Clustering Method With Thresholding for Underwater Images", International Journal of Computer Sciences Issues,Vol 7,Issue 3, No 9, May 2010, pp 35-40.

23. D. Q. Jhang and S.C. Chen, "Clustering Incomplete Data Using Kernel Based Fuzzy C- Means Algorithm", Neural Process. Lett. Vol18, no. 3, 155-162, 2003.

24. D W Kim, K Y Lee, D Lee and K H Lee, "Evaluation Of The Performance Of Clustering Algorithms In Kernel Induced Feature Space", Pattern recognition, vol.38, no.4, 607-611, Apr 2005.

25. B. Zhao, J. Kwok, and C. Zhang, "Multiple Kernel Clustering," in Proc.9th SIAM Int. Conf. Data Mining, Sparks, NV, 2009, pp. 638–649.

26. Long Chen, C. L. Philip Chen, and Mingzhu Lu, "A Multiple-Kernel Fuzzy C-Means Algorithm for Image Segmentation", IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, pp-120-140.

27. Mehmet Gonen and Ethem Alpaydin, "Multiple Kernels Learning Algorithms", Journal of Machine Learning Research 12 (2011) 2211-2268.

28. C. Y. Yeh, C. W. Huang, and S. J. Lee, "Multi-Kernel Support Vector Clustering For Multi-Class Classification," Int. J. Innovative Computer Appl., vol. 6, no. 5, pp. 2245–2262, May 2010.

29. M. G. Genton, "Classes Of Kernels For Machine Learning: A Statistics Perspective" , J. March. Learning Res., vol. 2, 299-312, 2001.

30. M Girlami, "Mercer Kernel Based Clustering in Feature Space", IEEE Transaction on neural network, Vol 13, No. 3, pp 780-784, May 2002.

31. H.Shen, J. Yang, S. Wang and X. Liu. "Attribute Weighted Mercer Kernel Based Fuzzy Clustering Algorithm For General Non Spherical Datasets", Soft Comput., vol.10, no. 11, 1061-1073, 2006.

32. Hsin-Chien Huang, Yung-Yu Chuang, Member, IEEE, and Chu-Song Chen, Member, IEEE, "Multiple Kernel Fuzzy Clustering," IEEE Transactions on fuzzy systems, vol. 20, no. 1, February 2012, pp120-134.

33. S. Sonnenburg, G. Ratsch, C. Schafer, andB. ScholKopf, "Large Scale Multiple Kernel Learning," J. March. Learning Res., vol. 7, 1531-1565, 2006.

34. C. Y. Yeh, C. W. Huang, and S. J. Lee, "Multi- Kernel Supports Vector Clustering For Multi – Class Classification", Int. J. Innovative Comput. Appl., vol. 6, no. 5, 2245-2262, 2010.

35. A. Frank and A. Asuncion, "UCI Machine Learning Repository, 2010. (Web Citation).

36. C. A. Cocosco, V. Kollokian, R. K. S. Kwan, and A. C. Evans, "Brain Web: Online Interface To A 3D MRI Simulated Brain Database," Neuroimage, vol. 5, no. 4, 1997 (Web Citation)