I.J. Intelligent Systems and Applications, **2012, 7, 50-60**
Published Online June 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2012.07.07

Review and Comparison of Kernel Based Fuzzy Image Segmentation Techniques

Prabhjot Kaur
Dept. of Information Technology,Maharaja Surajmal Institute of Technology, India
Email: thisisprabhjot@gmail.com

Pallavi Gupta
Dept. of Information Technology,Maharaja Surajmal Institute of Technology, India
Email: pallavigupta117@gmail.com

Poonam Sharma
Dept. of Information Technology,Maharaja Surajmal Institute of Technology, India
Email: poonam.paradise@gmail.com

Abstract—This paper presents a detailed study and comparison of some Kernelized Fuzzy C-means Clustering based image segmentation algorithms Four algorithms have been used Fuzzy Clustering, Fuzzy C-Means(FCM) algorithm, Kernel Fuzzy C-Means(KFCM), Intuitionistic Kernelized Fuzzy C-Means(KIFCM), Kernelized Type-II Fuzzy C-Means(KT2FCM).The four algorithms are studied and analyzed both quantitatively and qualitatively. These algorithms are implemented on synthetic images in case of without noise along with Gaussian and salt and pepper noise for better review and comparison. Based on outputs best algorithm is suggested.

Index Terms —Fuzzy Clustering, Fuzzy C-Means(FCM) algorithm, Kernel Fuzzy C-Means(KFCM),Intuitionistic Kernelized Fuzzy C-Means(KIFCM) ,Kernelized Type-II Fuzzy C-Means(KT2FCM),kernel width.

I. Introduction

Image segmentation plays an important role in image analysis and computer vision. The goal of image segmentation is partitioning of an image into a set of disjoint regions with uniform and homogeneous attributes such as intensity, color, tone etc. The image segmentation approaches can be divided into four categories: thresholding, clustering, edge detection, and region extraction. In image processing, two terms are usually seen very close to each other: clustering and segmentation. When we analyze the color information of the image, e.g. trying to separate regions or ranges of color components having same characteristics, the process is called color clustering and mapping the clusters onto the spatial domain by physically separated regions in the image is called segmentation. In images, the boundaries between objects are blurred and distorted due to the imaging acquisition process. Furthermore, object definitions are not always crisp and knowledge about the objects in a scene may be vague. Fuzzy set theory and Fuzzy logic are ideally suited to deal with such uncertainties. Fuzzy sets were introduced in 1265 by LoftiZadeh[1] with a view to reconcile mathematical modeling and human knowledge in the engineering sciences. Medical images generally have limited spatial resolution, poor contrast, noise, and non-uniform intensity variation. The Fuzzy C- means (FCM) [5], algorithm, proposed by Bezdek, is the most widely used algorithm in image segmentation because it has robust characteristics for ambiguity and it can retain much more information than hard segmentation methods. FCM has been successfully applied to feature analysis, clustering, and classifier designs in fields such as astronomy, geology, medical imaging, target recognition, and image segmentation. An image can be represented in various feature spaces and the FCM algorithm classifies the image by grouping similar data points in the feature space into clusters. In case the image is noisy or distorted then FCM technique wrongly classify noisy pixels because of its abnormal feature data which is the major limitation of FCM. Various approaches are proposed by researchers to compensate this drawback of FCM.

Tolias and Panas developed a rule based system that post-processed the membership function of any clustering result by imposing spatial continuity constraints and by using the inherent correlation of an eight-connected neighborhood to smooth the noise effect[6]. Acton and Mukherjee[7] incorporated multiscale information to enforce spatial constraints. Dave[1] proposed the idea of a noise cluster to deal with noisy data using the technique, known as Noise Clustering. He separated the noise into one cluster but this technique is not suitable for image segmentation, as we cannot separate pixels from the image. Another similar technique, PCM, proposed by Krishnapuran and Keller[2] interpreted clustering as a possibilistic partition. However, it caused clustering being stuck in one or two clusters. Ahmed et al.[10] modified the objective function of FCM by introducing a term that allowed the labeling of a pixel to be influenced by the labels in its immediate neighborhood. Zhang Yang, Fu-lai Chuang et al. (2002)[11] developed a robust clustering technique by deriving a novel objective function for FCM. Jiayin Kang et al.[12] proposed another technique which modified FCM objective function by incorporating the spatial neighborhood information into the standard FCM algorithm. Y. Yang et al.[13] proposed a novel penelized FCM (PFCM) for image segmentation where penalty term acts as a regulizer in the algorithm which is inspired by neighborhood maximization (NEM) algorithm[14] and is modified in order to satisfy the criterion of FCM algorithm. S. Shen, W. Sandham et al. [15] presented an algorithm called Improved FCM which changed the distance function used in FCM, i.e. the distance between pixel intensity and cluster intensities, and applied neural network optimization technique to adjust parameters in the modified distance function. Keh-Shih Chuang, Hong-Long Tzeng et al. [16] presented a FCM algorithm that incorporated spatial information into the membership function of clustering. The membership weighting of each cluster is altered after the cluster distribution in the neighborhood is considered. Rhee and Hwang [17] proposed Type 2 fuzzy clustering. Type 2 fuzzy set is the fuzziness in a fuzzy set. In this algorithm, the membership value of each pattern in the image is extended as type 2 fuzzy memberships by assigning membership grades (triangular membership function) to Type 1 fuzzy membership. The membership values for the Type 2 membership matrix are obtained as:

$$a_{ik} = u_{ik} - \frac{1-u_{ik}}{2} \tag{1}$$

Where a_{ik} and u_{ik} are the Type 2 and Type 1 fuzzy membership respectively. The cluster centers are updated according to the conventional FCM by taking into account the new Type 2 fuzzy membership.

While discussing the uncertainty, another uncertainty arised, which is the hesitation in defining the membership function of the pixels of an image. Since the membership degrees are imprecise and it varies on person's choice, hence there is some kind of hesitation present which arised from the lack of precise knowledge in defining the membership function. This idea lead to another higher order fuzzy set called intuitionistic fuzzy set which was introduced by Atanassov's[4] in 1213. It took into account the membership degree as well as non-membership degree. Few works on clustering is reported in the literature on intuitionistic fuzzy sets. Zhang and Chen (2002) suggested a clustering approach where an intuitionistic fuzzy similarity matrix is transformed to interval valued fuzzy matrix. From the similarity degrees between two intuitionistic fuzzy sets, intuitionistic fuzzy similarity matrix is created which is transformed to intuitionistic fuzzy equivalence matrix. Then, a clustering technique that uses intuitionistic fuzzy equivalence matrix is suggested based on the λ-cutting matrix of the intuitionistic fuzzy equivalence matrix. Recently, T. Chaira [2] proposed a novel intuitionistic fuzzy c-means algorithm using intuitionistic fuzzy set theory. This algorithm incorporated another uncertainty factor which is the hesitation degree that arised while defining the membership function.

This paper provides a comparative study of the original Fuzzy c- means clustering algorithm and its three kernalized variations like kernel based fuzzy c-means (KFCM),kernel based intuitionistic fuzzy c-means (KIFCM),kernel based Type-2 FCM (KT2FCM) which are the kernalized extension of their non kernalized counterparts FCM,IFCM and T2FCM respectively. The algorithms are kernalized by adopting a kernel induced metric in the data space to replace the original Euclidean norm metric. By replacing the inner product with an appropriate 'kernel' function, one can implicitly perform a non-linear mapping to a high dimensional feature space in which the data is more clearly separable.

The organization of the paper is as follows: Section II reviews Fuzzy C-Means (FCM), Kernelized Fuzzy C-Means (KFCM), Kernelized Type-2 FCM (KT2FCM) and KernelizedIntuitionistic Fuzzy C-means (KIFCM) in detail. Section III evaluates the performance of the algorithms presenting a comparison between them both qualitatively and quantitatively followed by conclusion in section IV.

II. Background Information

This section discusses the Fuzzy C-Means (FCM), Kernelized Fuzzy C-Means (KFCM), Kernelized Type-2 Fuzzy C-means(KT2FCM), and Kernelized Intuitionistic Fuzzy C means (KIFCM) algorithms in detail. In this paper, the data-set is denoted by 'X', where X={x_1, x_2, x_3 …x_n} specifying an image with 'n' pixels in M-dimensional space to be partitioned into 'c' clusters. Centroids of clusters are denoted by v_i and d_{ik} is the distance between x_k and v_i.

A. Kernel based approach

The present work proposes a way of increasing the accuracy of the intuitionistic fuzzy c-means by exploiting a kernel function in calculating the distance of data point from the cluster centers i.e. mapping the data points from the input space to a high dimensional space in which the distance is measured using a Radial basis kernel function.

The kernel function can be applied to any algorithm that solely depends on the dot product between two vectors. Wherever a dot product is used, it is replaced by a kernel function. When done, linear algorithms are transformed into non-linear algorithms. Those non-linear algorithms are equivalent to their linear originals operating in the range space of a feature space φ. However, because kernels are used, the φ function does not need to be ever explicitly computed. This is highly desirable, as sometimes our higher-dimensional feature space could even be infinite-dimensional and thus infeasible to compute.

A kernel function is a generalization of the distance metric that measures the distance between two data points as the data points are mapped into a high dimensional space in which they are more clearly separable. By employing a mapping function $\Phi(x)$, which defines a non-linear transformation: $x \to \Phi(x)$, the non-linearly separable data structure existing in the original data space can possibly be mapped into a linearly separable case in the higher dimensional feature space.

Given an unlabeled data set X={x$_1$, x$_2$, …., x$_n$} in the p-dimensional space R^P, let Φ be a non-linear mapping function from this input space to a high dimensional feature space H:

$$\Phi: R^p \to H, \quad x \to \Phi(x)$$

The key notion in kernel based learning is that mapping function Φ need not be explicitly specified. The dot product in the high dimensional feature space can be calculated through the kernel function $K(x_i, x_j)$ in the input space R^P.

$$\mathrm{K}(\mathrm{x_i}, \mathrm{x_j}) = \Phi(\mathrm{x_i}).\,\Phi(\mathrm{x_j})$$

Consider the following example. For p=2 and the mapping function Φ,

$$\Phi: R^2 \to H = R^3(x_{i1}, x_{i2}) \to (x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2})$$

Then the dot product in the feature space H is calculated as

$$\begin{aligned}
&\Phi(x_i).\,\Phi(x_j) \\
&= (x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}).\,(x_{j1}^2, x_{j2}^2, \sqrt{2}x_{j1}x_{j2}) \\
&= ((x_{i1}^2, x_{i2}^2).\,(x_{j1}^2, x_{j2}^2))^2 \\
&= (x_i.\,x_j)^2 \\
&= K(x_i, x_j)
\end{aligned}$$

where K-function is the square of the dot product in the input space. We saw from this example that use of the kernel function makes it possible to calculate the value of dot product in the feature space H without explicitly calculating the mapping function Φ. Some examples of kernel function are:

Example 1: Polynomial Kernel:

$$K(x_i, x_j) = (x_i.\,x_j + c)^d, \text{ where } c \geq 0,\ d \in N$$

Example 2: Gaussian Kernel:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \text{ where } \sigma > 0$$

Example 3: Radial basis Kernel:

$$K(x_i, x_j) = \exp\left(-\frac{\sum\left|x_i^a - x_j^a\right|^b}{\sigma^2}\right), \text{ where } \sigma, a, b > 0$$

RBF function with a=1, b=2 reduces to Gaussian function.

Example 4: Hyper Tangent Kernel:

$$K(x_i, x_j) = 1 - tanh\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right), \text{ where } \sigma > 0$$

B. The Fuzzy C Means Algorithm (FCM)

Fuzzy c-means (FCM)[1] is a method of clustering which allows one piece of data to belong to two or more clusters. This method (developed by Dunn in 1273 and improved by Bezdek in 1211) is frequently used in pattern recognition. FCM is the most popular fuzzy clustering algorithm. It assumes that number of clusters 'c' is known in priori and minimizes the objective function (J$_{\text{FCM}}$) as:

$$\begin{aligned}
J_{\text{FCM}} &= \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m \|x_k - v_i\|^2 \quad ,1 \leq m < \infty J_{FCM} \\
&= \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m d_{ik}^2
\end{aligned} \tag{2}$$

$$0 \leq u_{ik} \leq 1 \text{for all } i\text{ , }k\text{ ;} \tag{2a}$$

$$\textstyle\sum_{i=1}^{c} u_{ik} = 1 \text{ for all k;} \tag{2b}$$

$$\textstyle\sum_{k=1}^{n} u_{ik} > 0\ \ \forall i \tag{2c}$$

Where$d_{ik} = \|x_k - v_i\|$, and u$_{ik}$ is the membership of pixel 'x$_k$' in cluster 'i', which satisfies the following relationship:

$$\textstyle\sum_{i=1}^{c} u_{ik} = 1;\ \ i = 1,2, \ldots\ldots n \tag{3}$$

Here 'm' is a constant, known as the fuzzifier (or fuzziness index), which controls the fuzziness of the resulting partitionand can be any real number greater than 1 but generally we take it as 2 . Any norm ||·||expressing the similarity between any measured data and the centre can be used for calculating d$_{ik}$. Minimization of J$_{\text{FCM}}$ is performed by a fixed point iteration scheme known as the alternating optimization technique. This algorithm works by assigning membership to each data point corresponding to each cluster centre on the basis of distance between the cluster centre and the data point. More the data is near to the cluster centre more is its membership towards the particular cluster centre. Clearly, summation of membership of each data point should be equal to one.The conditions for local extreme for (1) and (2) are derived using Lagrangian multipliers:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{\|x_i - v_k\|}{\|x_j - v_k\|}\right)^{\frac{2}{m-1}}} \tag{4}$$

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}} \quad \forall\ k, i \tag{5}$$

where$1 \le i \le c;\ \ 1 \le k \le n$

and

$$v_i = \frac{\sum_{k=1}^{n}(u_{ik}^{m} x_k)}{\sum_{k=1}^{n}(u_{ik}^{m})}\ \ \forall\ i \tag{6}$$

The FCM algorithm iteratively optimizes J_{FCM}(U,V) with the continuous update of U and V, until $|U^{(l+1)} - U^{(l)}| <= \varepsilon$, where 'l' is the number of iterations. FCM works fine for the images which are not corrupted with noise but if the image is noisy or distorted then it wrongly classifies noisy pixels because of its abnormal feature data which is pixel intensity in the case of images, and results in an incorrect membership and improper segmentation.

Algorithmic steps for Fuzzy c-means clustering

Let $X = \{x_1, x_2, x_3 ..., x_n\}$ be the set of data points and $V = \{v_1, v_2, v_3 ..., v_c\}$ be the set of centers.

1) Randomly select 'c' cluster centers.

2) Calculate the fuzzy membership 'μ_{ij}' using:eq(4)

3) Compute the fuzzy centers 'v_j' using:eq(6)

4) Repeat step 2) and 3) until the minimum 'J' value is achieved or ||U(k+1) - U(k)|| < ε.

where'k' is the iteration step.
'ε.' is the termination criterion between [0, 1].

C. Kernelized Fuzzy c-means (KFCM)

Fuzzy C-means (FCM) [3] clustering algorithm is the soft extension of the traditional hard C-means. It considers each cluster as a fuzzy set, while a membership function measures the possibility that each training vector belongs to a cluster. As a result, each training vector may be assigned to multiple clusters. Thus it can overcome in some degree the drawback of dependence on initial partitioning cluster values in hard C-means. However, just like the C-means algorithm, FCM is effective only in clustering those crisp, spherical, and non-overlapping data. When dealing with non-spherical shape and much overlapped data, such as the Ring dataset FCM cannot always work well .Therefore we use the kernel method [3][5] to construct the nonlinear version of FCM, and construct a kernel-based fuzzy C-means clustering algorithm (KFCM).

The basic ideas of KFCM is to first map the input data into a feature space with higher dimension via a nonlinear transform and then perform FCM in that feature space. Thus the original complex and nonlinearly separable data structure in input space may become simple and linearly separable in the feature space after the nonlinear transform (see Fig.1). So we desire to be able to get better performance. Another merit of KFCM is, Unlike the FCM which needs the desired number of clusters in advance, it can adaptively determine the number of clusters in the data under some criteria. The experimental results show that KFCM has best performance in the test for the Ring dataset.

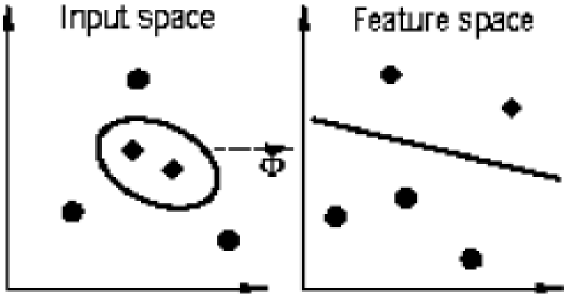


Fig.1 The idea of kernel method

In the FCM algorithm we use the following objective function in eq(2) for minimization. From the equation (2), we can see that classical FCM algorithm is based on the input space sum-of-squares clustering criterion. If the separation boundaries between clusters are nonlinear then FCM will unsatisfactorily work. To solve this problem we adopt the strategy of nonlinearly mapping the input space into a higher dimensional feature space and then performing linear FCM within the feature space. Assume we define the nonlinear map as ,$\varphi: x \rightarrow \varphi(x) \epsilon F$ where $x \epsilon X$. X denotes the input space, and F denotes the feature space with higher dimension. Note that the cluster centre in the feature space can now be denoted by the following form

$$v_i = \sum_{k=1}^{n} \beta_{ik}\, \varphi(x_k) \text{for all i;}$$

where β ik is the coefficients which will be calculated later. So similar to Equation (1) we select the following objective equation to be optimized

$$J_m = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m}\, ||\varphi(x_k) - \sum_{l=1}^{n} \beta_{il}\, \varphi(x_l)||^2 \tag{7}$$

We rewrite the norm in equation (7) as the following:

$$\left\|\varphi(x_k) - \sum_{l=1}^{n} \beta_{il}\, \varphi(x_l)\right\|^2$$

$$= \varphi(x_k)^T \varphi(x_k) - 2\sum_{l=1}^{n} \beta_{il}\, \varphi(x_k)^T \varphi(x_l) + \sum_{l=1}^{n} \sum_{j=1}^{n} \beta_{il} \varphi(x_k)^T \beta_{ij}\, \varphi(x_j) \tag{8}$$

It's interesting to see that equation (2) takes as the form of a series of dot products in feature space. And these dot products can easily be computed through Mercer kernel representations in the input space

$$K(x,y) =. \varphi(x)^T\varphi(y)$$

Thus the equation (8) can be formulated as follows

$$\|\varphi(x_k) - \sum_{l=1}^{n} \beta_{il}\,\varphi(x_l)\|^2 = K_{kk} - 2\beta_i^T K_k + \beta_i^T K \beta_i \tag{9a}$$

where $\beta_i = (\beta_{i1}, \beta_{i2}, \ldots., \beta_{in})^T$, $i = 1,2,\ldots,c$; $K_k = (K_{k1}, K_{k2}, \ldots, K_{kn})^T$

and $K = (K_1, K_{2,\ldots\ldots,}K_n)$ for k =1,2,..., n

We can calculate K_{ij} in the following way

K_{ij}=$K(x_i, x_j)$ for all i,j = 1,2,…,n.

where K is also named the kernel function. While any function satisfying the Mercer condition can be used as the kernel function, the best known is polynomial kernels, radial basis functions and Neural Network type kernels. Substituting for the norm in equation (8) with equation (9a), we get

$$J_m = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m (K_{kk} - 2\beta_i^T K_k + \beta_i^T K \beta_i) \tag{9b}$$

Minimizing J_m with m > 1 under the constraint of equation (1b), we have the expressions of u_{ik} and β_I as follows

$$u_{ik} = \frac{(1/(K_{kk} - 2\beta_i^T K_k + \beta_i^T K\beta_i)^{1/(m-1)}}{\sum_{j=1}^{c}(K_{kk} - 2\beta_i^T K_k + \beta_i^T K\beta_i)^{1/(m-1)}} \text{ for all i,k;} \tag{10a}$$

$$\beta_i = \frac{\sum_{k=1}^{n} u_{ik}^m K^{-1} K_k}{\sum_{k=1}^{n} u_{ik}^m} \quad \text{for all I} \tag{10b}$$

D. *Kernelized Type2 fuzzy c-means clustering algorithm*

Rhee and Hwang [17] extended the type-1 membership values (i.e. membership values of FCM) to type-2 by assigning a membership function to each membership value of type-1. Their idea is based on the fact that higher membership values should contribute more than smaller memberships values, when updating the cluster centers. Type-2 memberships can be obtained as per following equation:

$$a_{ik} = u_{ik} - \frac{1-u_{ik}}{2} \tag{11}$$

Where a_{ik} and u_{ik} are the Type-2 and Type-1 fuzzy membership respectively. From (11), the type-2 membership function area can be considered as the uncertainty of the type-1 membership contribution when the center is updated. Substituting (11) for the memberships in the center update equation of the conventional FCM method gives the following equation for updating centers.

$$v_i = \frac{\sum_{k=1}^{n}(a_{ik})^m x_k}{\sum_{k=1}^{n}(a_{ik})^m} \tag{12}$$

During the cluster center updates, the contribution of a pattern that has low memberships to a given cluster is relatively smaller when using type-2 memberships and the memberships may represent better typicality. Cluster centers that are estimated by type-2 memberships tend to have more desirable locations than cluster centers obtained by type-1 FCM method in the presence of noise. T2FCM algorithm is identical to the type-1 FCM algorithm except equation (12). At each iteration, the cluster center and membership matrix are updated and the algorithm stops when the updated membership and the previous membership i.e. $\max_{ik}\left|a_{ik}^{new} - a_{ik}^{prev}\right| < \varepsilon, \varepsilon$ is a user defined value. Although T2FCM has proven effective for spherical data, it fails when the data structure of input patterns is non-spherical and complex.

Kernel based extension (KT2FCM)

(KT2FCM) adopts a kernel induced metric which is different from the Euclidean norm in the original Type-2 fuzzy c-means. KT2FCM minimizes the objective function:

$$J_{KT2FCM} = 2\sum_{i=1}^{c}\sum_{k=1}^{n} a_{ik}^m \|\Phi(x_k) - \Phi(v_i)\|^2 \tag{13a}$$

Here aik is the Type-2 membership. Where $\|\Phi(x_k) - \Phi(v_i)\|^2$ is the square of distance between $\Phi(x_k)$and $\Phi(v_i)$.

The distance in the feature space is calculated through the kernel in the input space as follows:

$$\begin{aligned}\|\Phi(x_k) - \Phi(v_i)\|^2 &= (\Phi(x_k) - \Phi(v_i)).(\Phi(x_k) - \Phi(v_i)) \\ &= \Phi(x_k).\Phi(x_k) - 2\Phi(x_k)\Phi(v_i) + \Phi(v_i).\Phi(v_i) \\ &= K(x_k, x_k) - 2K(x_k, v_i) + K(v_i, v_i)\end{aligned} \tag{13b}$$

We can use any kernel given in the section *A* examples. As Hyper Tangent kernel has been used in this technique so:

$$K(x,y) = 1 - \tanh\left(-\frac{\|x-y\|^2}{\sigma^2}\right),$$

Where is σ is defined as kernel width and it is a positive number, then

$$K(x,x) = 1$$

Thus (13a) can be written as

$$J_{KT2FCM} = 2\sum_{i=1}^{c}\sum_{k=1}^{n} a_{ik}^m (1 - K(x_k, v_i))$$

$$J_{KT2FCM} = 2\sum_{i=1}^{c}\sum_{k=1}^{n} a_{ik}^m (\tanh(-\frac{\|x_k - v_i\|^2}{\sigma^2})) \tag{14}$$

Given a set of points X, we minimize J_{KT2FCM} in order to determine a_{ik}& .v_i. We adopt an alternating optimization approach to minimize J_{KT2FCM} and need the following theorem:

Theorem 1: The necessary conditions for minimizing J_{KT2FCM} under the constraint of U, we get:

$$a_{ik} = \frac{\left(\frac{1}{\tanh\left(-\frac{\|x_k-v_i\|^2}{\sigma^2}\right)}\right)^{\frac{1}{m-1}}}{\sum_{i=1}^{c}\left(\frac{1}{\tanh\left(-\frac{\|x_k-v_i\|^2}{\sigma^2}\right)}\right)^{\frac{1}{m-1}}} \quad (15)$$

$$v_i = \frac{\sum_{k=1}^{n} a_{ik}^{m} K(x_k,v_i)\left(1+\tanh\left(-\frac{\|x_k-v_i\|^2}{\sigma^2}\right)\right)x_k}{\sum_{k=1}^{n} a_{ik}^{m} K(x_k,v_i)\left(1+\tanh\left(-\frac{\|x_k-v_i\|^2}{\sigma^2}\right)\right)} \quad (16)$$

The optimal values of the kernel width can be obtained through (14),

E. Kernelized Intuitionistic fuzzy c-means clustering algorithm

Intuitionistic fuzzy c-means clustering algorithm is based upon intuitionistic fuzzy set theory. Fuzzy set generates only membership function $\mu(x), x \in X$, whereas Intutitionistic fuzzy set (IFS) given by Atanassov considers both membership $\mu(x)$ and non-membership $v(x)$. An intuitionistic fuzzy set A in X, is written as:

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

Where $\mu_A(x) \to [0,1], v_A(x) \to [0,1]$ are the membership and non-membership degrees of an element in the set A with the condition:

$$0 \le \mu(x) + v_A(x) \le 1.$$

When $v_A(x) = 1 - \mu_A(x)$ for every x in the set A, then the set A becomes a fuzzy set. For all intuitionistic fuzzy sets, Atanassov also indicated a hesitation degree, $\pi_A(x)$, which arises due to lack of knowledge in defining the membership degree of each element x in the set A and is given by:

$$\pi_A(x) = 1 - \mu_A(x) - v_A(x); \qquad 0 \le \pi_A(x) \le 1$$

Due to hesitation degree, the membership values lie in the interval

$$[\mu_A(x), \mu_A(x) + \pi_A(x)]$$

Intuitionistic fuzzy c-means (T. Chaira, 2011)[2] objective function contains two terms: (i) modified objective function of conventional FCM using Intuitionistic fuzzy set and (ii) intuitionistic fuzzy entropy (IFE). IFCM minimizes the objective function as:

$$J_{IFCM} = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{*m} d_{ik}^2 + \sum_{i=1}^{c} \pi_i^* e^{1-\pi_i^*} \quad (17)$$

$u_{ik}^* = u_{ik} + \pi_{ik}$, where u_{ik}^* denotes the intuitionistic fuzzy membership and u_{ik} denotes the conventional fuzzy membership of the kth data in ithclass.

$\pi_{ik}$is hesitation degree, which is defined as:

$$\pi_{ik} = 1 - u_{ik} - (1 - u_{ik}{}^{\alpha})^{1/\alpha}, \alpha > 0,$$

and is calculated from Yager's intuitionistic fuzzy complement as under:

N(x) = $(1 - x^{\alpha})^{1/\alpha}$, $\alpha > 0$, thus with the help of Yager's intuitionistic fuzzy compliment , Intuitionistic Fuzzy Set becomes:

$$A_A^{IFS} = \{x, \mu_A(x), (1 - \mu_A(x)^{\alpha})^{1/\alpha} | x \epsilon X\} \quad (18)$$

And hesitation degree is given by:

$$\pi_i^* = \frac{1}{N}\sum_{k=1}^{n} \pi_{ik}, \text{k} \in [1, N]$$

Second term in the objective function is called intuitionistic fuzzy entropy (IFE). Initially the idea of fuzzy entropy was given by Zadeh in 1262. It is the measure of fuzziness in a fuzzy set. Similarly in the case of IFS, intuitionistic fuzzy entropy gives the amount of vagueness or ambiguity in a set. For intuitionistic fuzzy cases, if $\mu_A(x_i), v_A(x_i), \pi_A(x_i)$ are the membership, non-membership, and hesitation degrees of the elements of the set X=$\{x_1, x_2, \ldots\ldots x_n\}$, then intuitionistic fuzzy entropy, IFE that denotes the degree of intuitionism in fuzzy set, may be given as:

$$IFE(A) = \sum_{i=1}^{n} \pi_A(x_i) e^{[1-\pi_A(x_i)]} \quad (19)$$

Where $\pi_A(x_i) = 1 - \mu_A(x_i) - v_A(x_i)$.

IFE is introduced in the objective function to maximize the good points in the class. The goal is to minimize the entropy of the histogram of an image.

Modified cluster centers are:

$$v_i^* = \frac{\sum_{k=1}^{n} u_{ik}^* x_k}{\sum_{k=1}^{n} u_{ik}^*} \quad (20)$$

At each iteration, the cluster center and membership matrix are updated and the algorithm stops when the updated membership and the previous membership i.e.

$max_{ik}\left|U_{ik}^{*new} - U_{ik}^{*prev}\right| < \varepsilon, \varepsilon$is a user defined value.

Kernel based extension

The model RBF kernel based Intuitionistic fuzzy c-means (KIFCM) adopts a kernel induced metric which is different from the Euclidean norm in the original intuitionistic fuzzy c-means. KIFCM minimizes the objective function:

$$J_{KIFCM} = 2\sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{*m} \|\Phi(x_k) - \Phi(v_i)\|^2 + \sum_{i=1}^{c} \pi_i^* e^{1-\pi_i^*} \quad (21)$$

where $\|\Phi(x_k)-\Phi(v_i)\|^2$ is the square of distance between $\Phi(x_i)$ and $\Phi(v_k)$ and can be calculated as in eq (13b). We can use any kernel given in the section *A* examples As Radial basis kernel has has been used in this technique so:

$$K(x,y)=\exp\left(-\frac{\sum\left|x_i^a-x_j^a\right|^b}{\sigma^2}\right)$$

Where a and b is greater than 0, and σ is defined as kernel width and it is a positive number, then

$$K(x,x)=1.$$

Hence,

$$\|\Phi(x_k)-\Phi(v_i)\|^2=2(1-K(x_k,v_i))$$

Thus (21) can be written as:

$$J_{KIFCM}=2\sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^{*m}(1-K(x_k,v_i))+\sum_{i=1}^{c}\pi_i^*e^{1-\pi_i^*}$$

$$J_{KIFCM}=2\sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^{*m}\left(1-\exp\left(-\frac{\sum\left|x_i^a-x_j^a\right|^b}{\sigma^2}\right)\right)+\sum_{i=1}^{c}\pi_i^*e^{1-\pi_i^*} \tag{22}$$

Given a set of points X, we minimize J_{KIFCM} in order to determine u_{ik}^* & v_i . We adopt an alternating optimization approach to minimize J_{KIFCM} and need the following theorem:

Theorem 1: The necessary conditions for minimizing J_{KIFCM} under the constraint of U, we get:

$$u_{ik}^*=\frac{\left(\frac{1}{\left(1-K(x_k,v_i)\right)}\right)^{\frac{1}{m-1}}}{\sum_{i=1}^{c}\left(\frac{1}{\left(1-K(x_k,v_i)\right)}\right)^{\frac{1}{m-1}}} \tag{23}$$

$$v_i=\frac{\sum_{k=1}^{n}u_{ik}^{*m}\times K(x_k,v_i)\times x_k}{\sum_{k=1}^{n}u_{ik}^{*m}\times K(x_k,v_i)} \tag{24}$$

III. Simulations & Results

In this section, we describe the experimental results on four synthetic images bacteria and hand. There are total of four algorithms used i.e. FCM, KFCM, KIFCM and KT2FCM. Note that the kernel width σ in kernelized methods has a very important effect on performances of the algorithms. However how to choose an appropriate value for the kernel width in kernelized methods is still an open problem. In this project we adopt the “trial and error” for each image for each algorithm and set the parameter σ variably. In addition we set the parameter N (total number of iterations)=200, m=2, ξ=3e-2 in the rest experiments. Experiments are implemented and simulated using MATLAB Version R2001a and Intel(R) Core(TM) i5 CPU 2.27 GHZ 3GB RAM, 64BIT Operating System.

We test the algorithms’ performance when corrupted by “Gaussian” and “salt and pepper” noises both qualitatively and quantitatively. Qualitative analysis of the algorithms’ is discussed in section III.A and quantitative analysis is discussed in section III.B.

A. *Qualitative Analysis*

Bacteria Image

Test Image Bacteria (123x123) is generated as shown in fig 1(a). Salt and pepper noise and Gaussian noise of intensity 0.02 is added to generate image 1(b) and (c) respectively. By analyzing the 2nd row fig 1(d), 1(e), 1(f) which is the output when FCM algorithm is applied we observe segmentation is good in all the three cases but by looking at the bacteria we observe that FCM is unable to remove Gaussian or salt and pepper noise. KFCM (fig 1(g), 1(h), 1(i)) shows good segmentation, but we observe that it results in increased size in outputs. Looking at the bacteria we see KFCM removes both the type of noises. KIFCM fig 1(j),1(k), 1(l), as with KFCM it results in increased size in outputs .Results are almost same as KFCM but more noise is removed in case of KFCM . With KT2FCM fig 1(m),1(n),1(o), in case of noiseless segmentation is not very clear on boundaries though its clearer when noise is present.KT2FCM even remove some noise. So in case our image is noiseless FCM and KFCM both show equivalent result so any can be used. While when salt and pepper is present KT2FCM should be used and in case Gaussian is present any of the kernel methods can be used.

Hand Image

Test Image Hand (303x243) is generated as shown in fig 2(a). Salt and pepper noise and Gaussian noise of intensity 0.02 is added to generate image 2(b) and 2(c) respectively. By analyzing the 2nd row fig 2(d), 2(e), 2(f) which is the output when FCM algorithm is applied we observe good performance in all the three cases, though best as expected with noiseless. FCM shows better performance with salt n pepper as compared to with Gaussian. FCM is unable to remove Gaussian noise. Moving to result of KFCM (fig 2(g), 2(h), 2(i)) we observe that KFCM shows its best performance with salt n pepper. It almost removes salt n pepper. Performance with noiseless is also good. In case of Gaussian noise it succeeds in marginally reducing the noise but segmentation of some portion of ring not clear. Analysing output of KIFCM fig 2(j), 2(k), 2(l), segmentation is not proper with noiseless and salt n pepper as ring area not properly detected. Segmentation is better with Gaussian but unable to remove Gaussian noise or even salt n pepper. With KT2FCM fig 2(m), 2(n), 2(o) best segmentation as compared with other algorithms is observed. KT2FCM even removes salt and pepper and Gaussian to some extent. So we can say that if noiseless case then, undoubtedly FCM is best and very near to it is KT2FCM. With KIFCM and KFCM segmentation not good with the ring part. In case its salt and pepper then KFCM is the winner. KT2FCM removes some noise but fails in ring segmentation. For Gaussian there is a close fight among all the algorithms. While KFCM removes some noise, segmentation is not that good in the ring part. While segmentation is best with KIFCM, noise not at all removed.

B. Quantitative Analysis

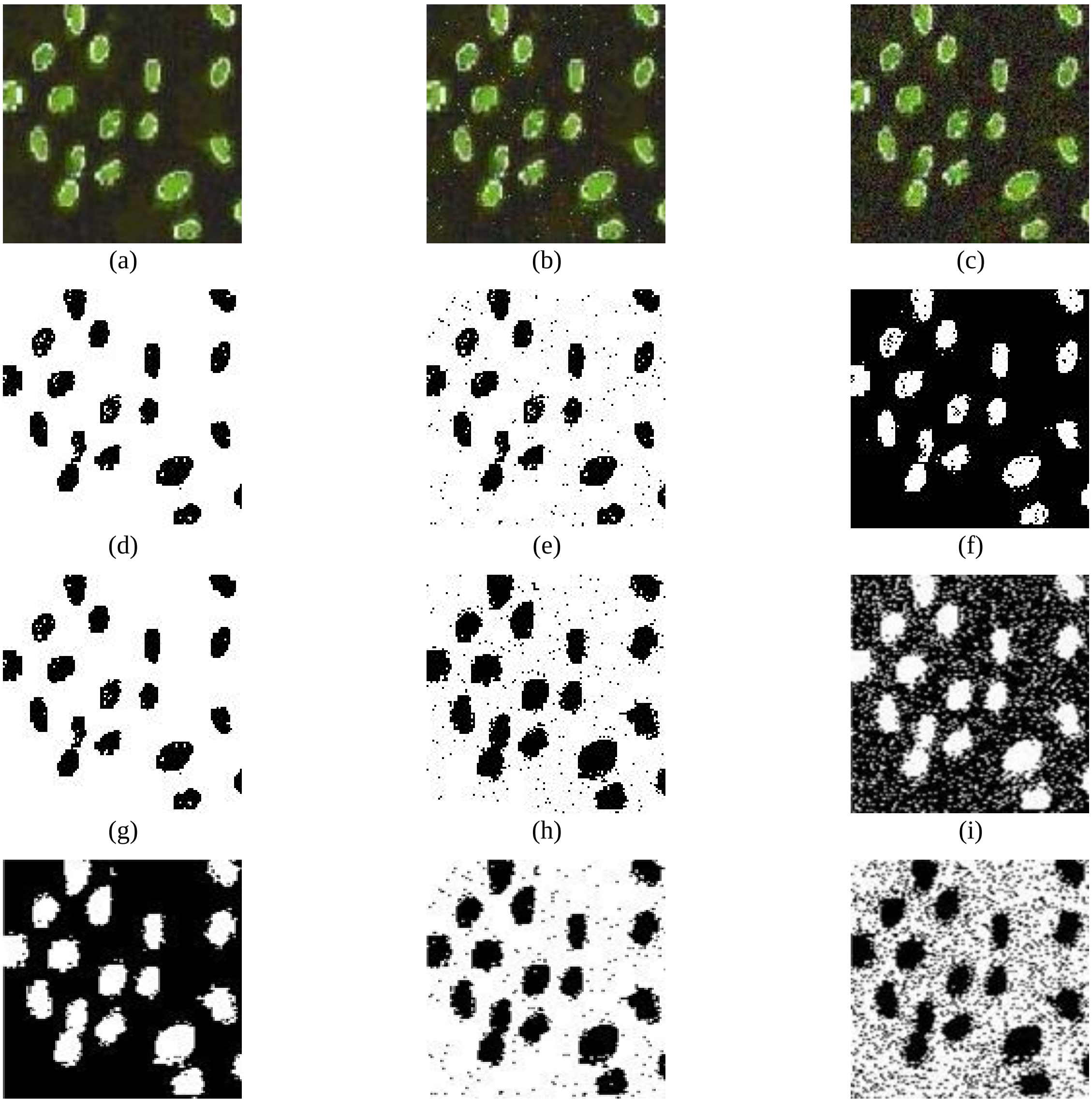
Based on Execution Time

Bacteria Image

Fig 3 shows the bar chart of the execution time of FCM,KFCM,KIFCM and KT2FCM for Bacteria Image in three different case of noiseless, salt and pepper and Gaussian. FCM performs best with noiseless and worst with Gaussian. While it is seen that KFCM performs best with Gaussian and worst with salt and pepper. KIFCM has max execution time with Gaussian and least with noiseless. KT2FCM least execution time as compared to other algorithm. It shows almost same time in all the three cases. Hence KFCM andKT2FCM show best execution time while KIFCM the worst performance.

Hand Image

Fig 4 shows the bar chart of the execution time of FCM,KFCM,KIFCM and KT2FCM for Hand Image in three different case of noiseless, salt and pepper and Gaussian. In FCM we observe least execution time with noiseless and maximum with salt and pepper. With KFCM best performance with salt and pepper and worst with Gaussian. In case its KIFCM we see large execution time as compared to other algorithms. KIFCM shows best in Gaussian and worst as expected with noiseless.KT2FCM shows least execution time as compared to other algorithms. Best with Gaussian and worst in salt and pepper. Based on execution time we can say , KIFCM has worst performance while KT2FCM shows best.

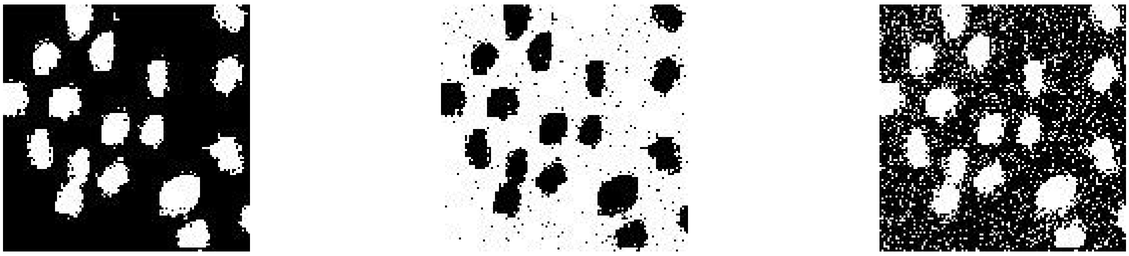


(a) (b) (c)

(d) (e) (f)

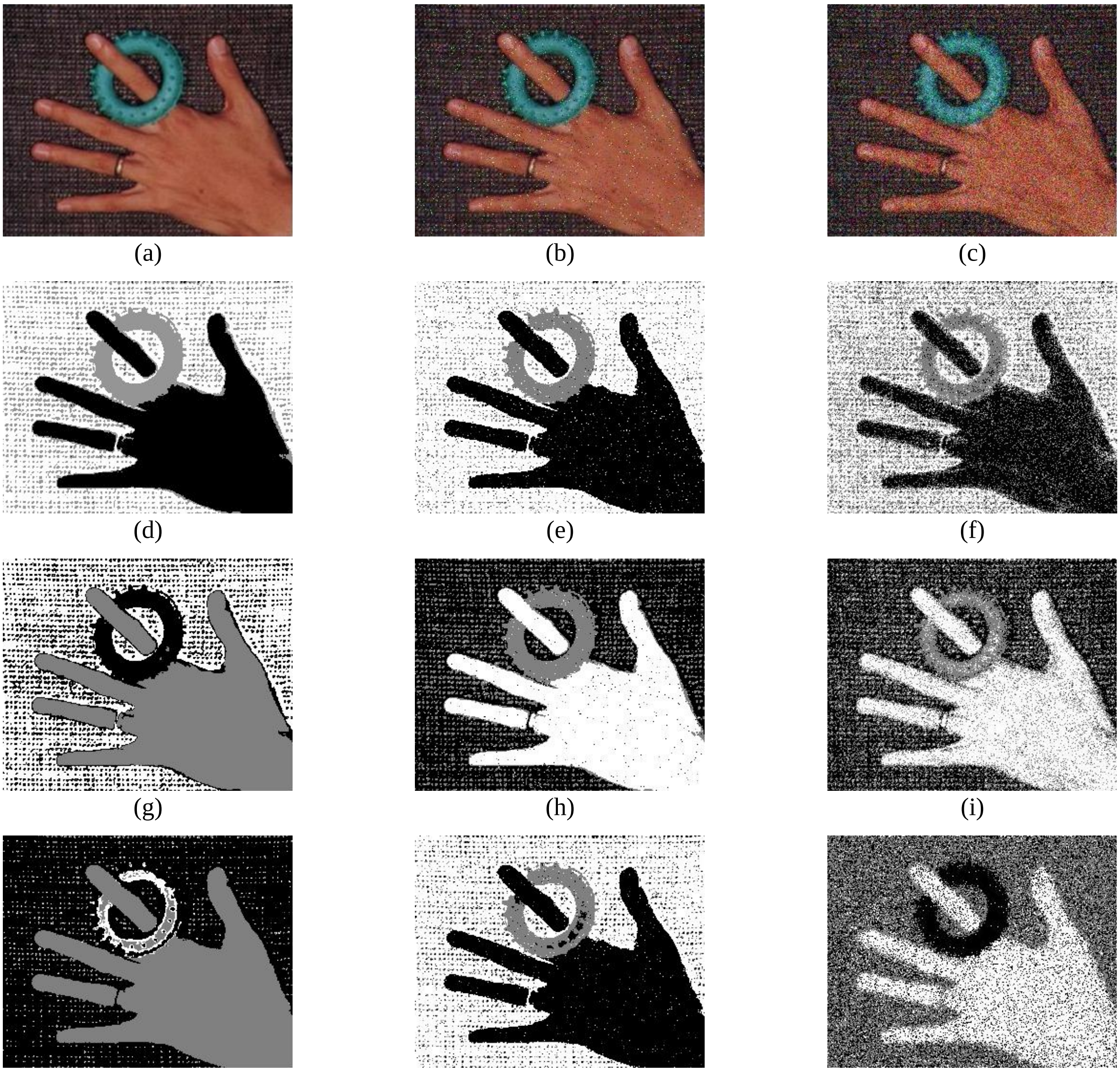
(g) (h) (i)

(j) (k) (l)



(m) (n) (o)

Fig.1: Ouputs on bacteria image Column 1 shows output of algorithms in noiseless case, column 2 shows output in salt and pepper noise and cloumn 3 shows output with gaussian noise.
Fig 1(a),1(b),1(c) are the original images in the 3 cases of noiseless,salt and pepper, and gaussian respectively.
Fig 1(d), 1(e), 1(f) represents output of FCM in the 3 cases.
Fig 1(g),1(h),1(i) represents output of KFCM in the 3 cases.
Fig 1(j),1(k),1(l) represents output of KIFCM in the 3 cases.
Fig 1(m),1(n),(o) represents output of K2FCM in the 3 cases.

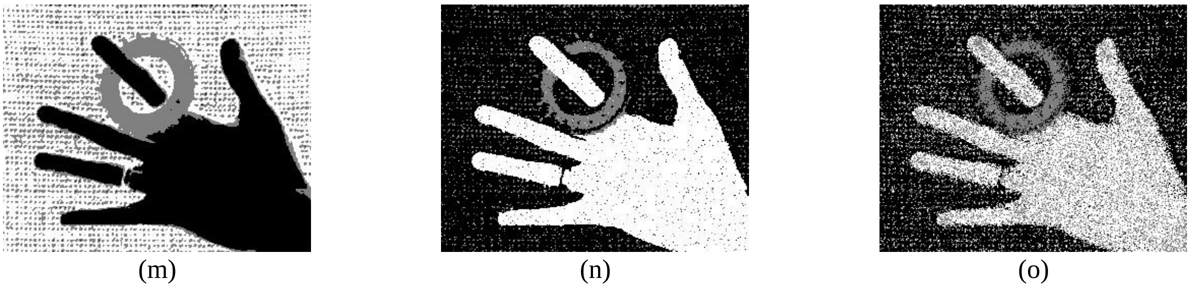


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)



(m) (n) (o)

Fig. 2: Ouputs on hand image Column 1 shows output of algorithms in noiseless case, column 2 shows output in salt and pepper noise and cloumn 3 shows output with gaussian noise.

Fig 2(a),2(b),2(c) are the original images in the 3 cases of noiseless,salt and pepper, and gaussian respectively.

Fig 2(d), 2(e), 2(f) represents output of FCM in the 3 cases.

Fig 2(g),2(h),2(i) represents output of KFCM in the 3 cases.

Fig 2(j),2(k),2(l) represents output of KIFCM in the 3 cases.

Fig 2(m),2(n),2(o) represents output of K2FCM in the 3 cases.

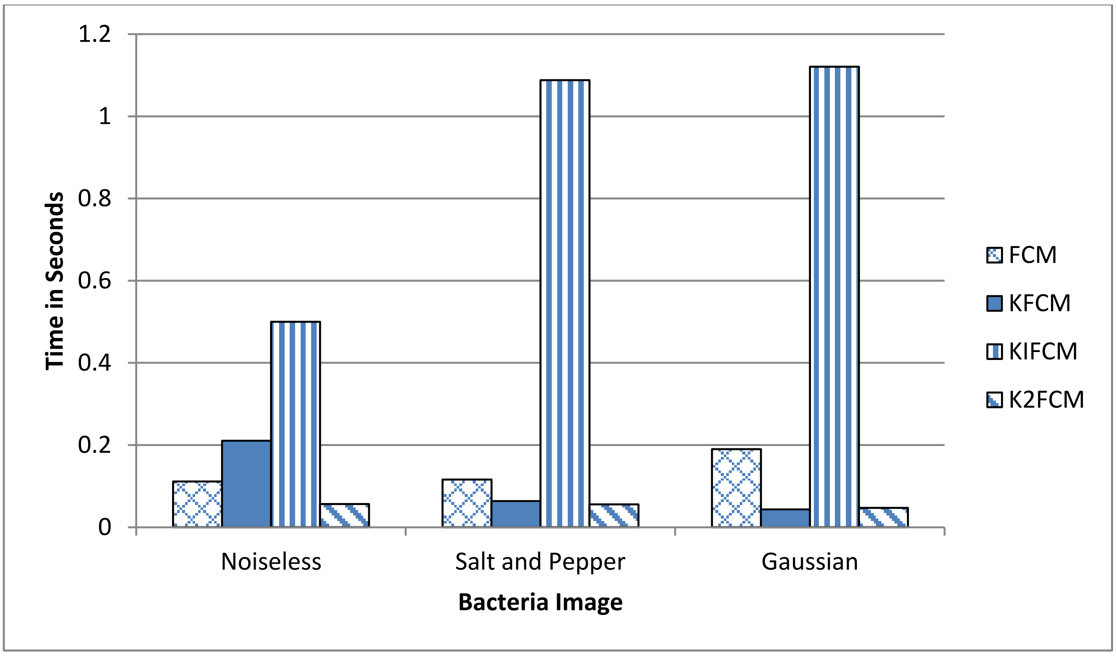


Fig. 3: Bar Chart of Bacteria Image

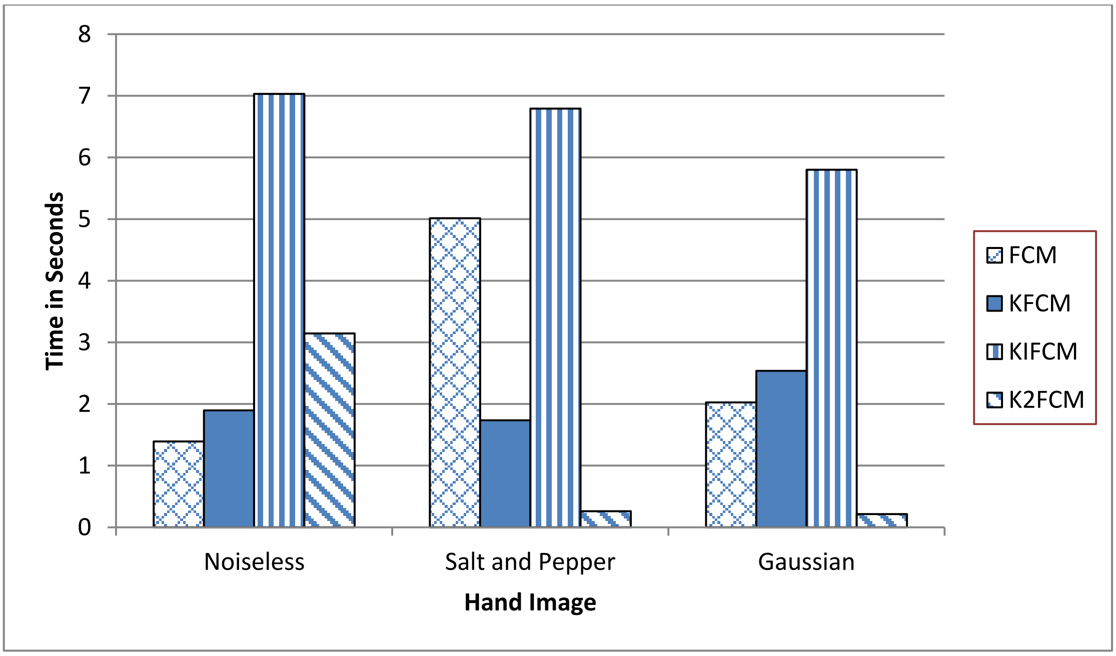


Fig. 4: Bar Chart of Hand Image

IV. Conclusion

In this paper we have discussed the performance of the four algorithms FCM, KFCM, KIFCM, KT2FCM for the synthetic digital images in noiseless case and when corrupted with salt and pepper and Gaussian noise. We compare these algorithms both qualitatively i.e. segmentation evaluation and quantitatively i.e. based on execution time and validity function. Based on segmentation evaluation we can say that FCM dominates when it comes to noiseless case closely followed by KFCM. In case noise is present kernel methods show better performance if compared with FCM. Among the kernel methods KFCM seems to be the best followed by close fight between KT2FCM and KIFCM depending on the input image. Analyzing based on execution time and comparing the results of all the four images, we can say FCM executes in smallest time. Comparing the kernel methods we can conclude that KIFCM takes the maximum time while KFCM and KT2FCM the least. Also as expected time to execute is lesser with two clusters than with three clusters.The results reported in the project show that kernel method is an effective approach to constructing a robust image clustering algorithm. This method can also be used to improve the performance of other FCM like algorithms based on adding some penalty terms to the original FCM objective function.

References

[1] L.A. Zadeh (1265), ―Fuzzy Sets, Inform. and Control, 1,331-353.

[2] A novel intuitionistic fuzzy C means clustering algorithm and its application tomedical images, Tamalika Chaira

[3] Applied Soft Computing 11 (2011) 1711–1717, Fuzzy Clustering Using Kernel Method, Daoqiang Zhang Songcan Chen Proceedings of 2012 international conference on control and automation ,Xiamen,china,june 2002

[4] K.T. Atanassov's, Intuitionistic fuzzy sets, VIII TKR's Session, Sofia, 213 (Deposed in Central Science – Technology Library of Bulgaria Academy of Science –1627/14).

[5] J.C. Bezdek (1211), ―Pattern Recognition with Fuzzy Objective Function Algorithm, Plenum, NY.

[6] Y.A. Tolias and S.M. Panas(1221), On applying spatial constraints in fuzzy image clustering using a fuzzy rulebased system., IEEE Signal Processing Letters 5, 245.247 (1221).

[7] S. T. Acton and D. P. Mukherjee(2000), ―Scale space classification using area morphology, IEEE Trans. Image Process., vol. 2, no. 4, pp. 623–635, Apr. 2000.

[8] R.N. Dave and R. Krishnapuram (1227), ―Robust Clustering Methods: A Unified View, IEEE Transactions on Fuzzy Systems, May, Vol 5, No. 2.

[9] R. Krishnapuram and J. Keller (1223), "A Possibilistic Approach to Clustering", IEEE Trans. on Fuzzy Systems, vol .1. No. 2, pp.21-1 10.

[10] M. N. Ahmed, S. M. Yamany, N. Mohamed, A. A. Farag, and T. Moriarty(2002), ―A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data, IEEE Trans. Med. Imag., vol. 21, no. 3, pp. 123–122, Mar. 2002.

[11] Yang Zhang, Chung Fu-Lai, et al,(2002), Robust fuzzy clustering-based image segmentation, Applied Soft Computing, vol(2), no. 1, Jan (2002), pp. 10-14.

[12] Kang Jiayin, Min Lequan et al. (2002), ―Novel modified fuzzy c-means algorithm with applications, Digital Signal Processing. March (2002),vol 12, no. 2, pp. 302-312.

[13] Yang Y., Zheng Ch., and Lin P. Fuzzy c-means clustering algorithm with a novel penalty term for image segmentation. Opto-electronic review, Vol.13, Issue 4, 2005, pp. 302-315.

[14] C. Ambroise and G. Govaert,(1221), Convergence of an EM-type algorithm for spatial clustering., Pattern Recognition Letters 12, 212.227(1221).

[15] Shen Shan, Sandham W., and Sterr A.(2005), MRI fuzzy segmentation of brain tissue using neighbourhood attraction with neural network optimization, IEEE transactions on information technology in biomedicine,Vol. 2, issue 3, September 2005, pp. 452-467.

[16] Chuang Keh-Shih, Tzeng Hong-Long, Chen Sharon et al.(2006), Fuzzy C-means clustering with spatial information for image segmentation, Computeried Medical Imaging and graphics,30(2006) 2-15.

[17] F.C.H. Rhee, C. Hwang, A Type-2 fuzzy c means clustering algorithm, in: Proc. in Joint 2th IFSA World Congress and 20th NAFIPS International Conference 4,2001, pp. 1226–1222.

PRABHJOT KAUR has completed her B. Tech. in 1999 and M. Tech. in 2003. Presently she is working as a Reader in Maharaja Surajmal Institute of Technology and pursuing her P.hD. from Sharda University, Greater Noida, New Delhi. Her area of interest includes Soft computing, Image processing, Medical Image segmentation.

PALLAVI GUPTA (1989-), female, pursuing her Bachelor of Information Technology from Maharaja Surajmal Institute of Technology, GGSIPU, New Delhi, India, her research directions include image processing and human computer interaction.

POONAM SHARMA (1990-),female, pursuing her Bachelor of Information Technology from Maharaja Surajmal Institute of Technology, GGSIPU, New Delhi,, India. Currently, she is doing her major project and this paper is the implementation of her project. Her area of research interest is image processing.