A ROBUST CLUSTERING ALGORITHM FOR SAR IMAGE SEGMENTATION USING HYBRID SPATIAL INFORMATION

A DESSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

MASTER OF TECHNOLOGY

IN

SIGNAL PROCESSING AND DIGITAL DESIGN

Submitted by:

Pratibha Singh Jaiswal

2K18/SPD/05

Under the supervision of

PROF. M. S. CHOUDHRY

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

SEPTEMBER 2020

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

Candidate's Declaration

I, Pratibha Singh Jaiswal (2K18/SPD/05) student of M. Tech. (Signal Processing and Digital Design), hereby declare that the project Dissertation titled "A Robust Clustering Algorithm for SAR Image Segmentation using Hybrid Spatial Information" which is submitted by me to the Department of Electronics and Communication Engineering, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

Date: 4th September 2020

PRATIBHA SINGH JAISWAL

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

Certificate

I hereby certify that the Project Dissertation titled "A Robust Clustering Algorithm for SAR Image Segmentation using Hybrid Spatial Information" which is submitted by Ms. Pratibha Singh Jaiswal (2K18/SPD/05) of Electronics and Communication Engineering Department, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date: September 2020

PROF. M. S. CHOUDHRY

(SUPERVISOR)

Department of E&C

DTU Delhi

Acknowledgement

I would like to thank Defence Research and Development Organisation for giving me this opportunity to foster my academic interest.

I am exceedingly grateful to my supervisor Prof. M. S. Choudhry for his invaluable guidance, constructive criticism and incessant encouragement in spite of his heavy work commitments and busy schedule. It has been an enriching experience to learn and work under him.

The in-time facilities provided by the department are also equally acknowledgeable. I would like to convey my sincere thanks to the teaching and nonteaching staff of the Department of Electronics and Communication Engineering for their invaluable help and support.

And last but not the least, I take immense delight in extending my acknowledgement to my husband who have helped and encouraged me throughout this research work. I also acknowledge the patience of my loving son while performing this research work.

Date: 4th September 2020

PRATIBHA SINGH JAISWAL
2K18/SPD/05
M.Tech. (SPDD)

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Synthetic aperture RADAR (SAR) is one of the primary sensor used to perform the task of remote sensing for various applications. All weather and day and night operation of SAR makes them advantageous to their optical counterparts. SAR has been used for many applications like DEM generation, deforestation, activity monitoring, military surveillance, volcano eruption monitoring etc. High-resolution images of large areas are obtained by synthetic aperture radar imaging techniques. Spatial orientation, dielectric constant and roughness of the imaging area affects the intensities of pixels in a SAR image.

Being an active sensor, SAR depends on its own transmitted energy. It generates the images of area based on the return scattered by it back to the radar's antenna. Raw SAR signal data is processed to generate spatial image. The SAR imaging can be performed from satellite or airborne platform with a side looking antenna. SAR images are inherently contaminated by speckle noise, which is multiplicative in nature. Speckle is caused by interference of coherent wave fronts .Being an active imaging system, SAR images suffer from the inherent multiplicative noise known as speckle, which originates from the interference of the coherent wave fronts. The presence of speckle makes image processing tasks challenging for SAR images [1].

Segmentation of image is one of the most critical pre-processing step done before classification and identification of different regions and objects present in the image. Segmentation of image is to partition it into *regions* (also called *segments*, *classes* or *subsets*) which are similar with respect to one or more characteristics or features e.g. grey tone or texture. The performance of conventional intensity based image segmentation techniques deteriorates for speckle contaminated SAR images. Various speckle reduction techniques like spatial filtering and multi-look processing can be used, but it will reduce the resolution of image. Requirement of high resolution in remote sensing application make performing segmentation in presence of speckle inevitable.

Given the challenge of segmenting SAR images in presence of speckle, thirteen image segmentation techniques are explored in this study. For speckle corrupted SAR images, intensity is not a suitable image feature for segmentation purposes, thus seventeen

different texture based image features have also been explored in the study. As a results, a total of 221 feature-segmentation technique combinations have been analysed in the study.

Following SAR image features have been analysed:

a. Intensity
b. Roughness from G^0_A distribution [2]
c. Scale from G^0_A distribution [3]
d. Roughness and Scale based feature map [4]
e. Roughness and Scale
f. Shannon entropy
g. Renyi entropy [5]
h. Tsallis entropy
i. Wavelet energy [6]
j. Local binary pattern (LBP)
k. Grey-Level Co-occurrence Matrices (GLCM)
l. Central statistical moments
m. Grey-Level Difference Method (GLDM) probability density functions
n. Local pdf
o. Grey-Level Run Length Matrix (GLRLM)
p. Histogram of gradients
q. Variogram

For the above features, following image segmentation techniques have been analysed:

a. K-means clustering
b. Fuzzy C-Means (FCM) clustering
c. Spatial FCM [7]
d. Fuzzy Local Information C-Means (FLICM) clustering [8]
e. Pixel Intensity and Location information FCM (ILKFCM) clustering [6]
f. Gaussian Mixture Model (GMM)
g. Self-Organising Map (SOM)
h. Modified ILKFCM
i. Kernel method Weighted FLICM (KWFLICM) [9]
j. Fuzzy Local Information and L_p- norm distance-based clustering (FLILp) [10]

k. Modified FLICM
l. Modified KWFLICM
m. Modified Spatial FCM

For evaluating the performance of these segmentation techniques using different textural features experimentation is performed on 72 synthetic and 3 real SAR images. Synthetic SAR images are generated using different foreground and background parameter combinations. These varying combinations maps to different "Difficulty of Segmentation" metric for different images. Performance evaluation for synthetic SAR images are based on overall accuracy, normalized accuracy and Kappa coefficient. For real SAR images performance is evaluated in terms of normalized accuracy, Kappa coefficient, F-Measure and cross-region fitting measure.

Chapter 2
Image Features

2.1 Intensity

Pixel intensity value is the primary image information. For conventional image segmentation, intensity is the most popular and fundamental feature [11]. In a SAR image, the intensity value for each pixel is a single value as it is a grey-level image.

2.2 Roughness from G^0_A distribution

Segmentation techniques generally use the most evident image information, i.e., intensity of pixels. However, in the presence of noise, the intensity based segmentation may not be the best method. To tackle it, texture based methodology for automatic segmentation of SAR images can be used.

Roughness from G^0_A distribution is one of the texture feature that can be used for segmentation purposes. It encompasses adopting a suitable data model (the G^0_A distribution) and parameter estimation (roughness and scale).

The performance of SAR image segmentation technique is highly dependent on choice of an appropriate probability distribution to model image data [1]. In this study, G^0_I and G^0_A distributions [1] are used to model SAR image data in intensity and amplitude, respectively. These distributions are advantageous since they can accurately model varying degree of inhomogeneity of a SAR image [1]. In addition, other distributions such as K and Weibull do not present this feature.

For SAR image segmentation, G^0_I and G^0_A distributions are used to estimate the roughness parameter maps of image. A nonlinear system of log cumulants of orders 1 and 2 is solved to estimate parameters for both distributions [3].

2.2.1 Statistical Models for SAR Data

The probability density functions of general models for SAR intensity G^0_I and amplitude G^0_A with return data (Z) are defined as follows:

- G^0_I distribution, i.e.,

$$f_{G_I^0}(z,\alpha,\gamma,L) = \frac{L^L\Gamma(L-\alpha)}{\gamma^\alpha\Gamma(-\alpha)\Gamma(L)}\, z^{L-1}(\gamma+Lz)^{\alpha-L} \tag{2.1}$$

Where,

$\Gamma(L)$ is the Gamma function;

$L \geq 1$ is the number of looks;

$-\alpha,\ \gamma, z > 0$;

and the r[th]-order moment is given by

$$E_{G_I^0}[Z^r] = \left(\frac{\gamma}{L}\right)^r \frac{\Gamma(-\alpha-r)\Gamma(L+r)}{\Gamma(-\alpha)\Gamma(L)}, \quad \alpha < -r \tag{2.2}$$

- G^0_A distribution, i.e.,

$$f_{G_A^0}(z,\alpha,\gamma,L) = \frac{2L^L\Gamma(L-\alpha)}{\gamma^\alpha\Gamma(-\alpha)\Gamma(L)} z^{2L-1}(\gamma+Lz^2)^{\alpha-L} \tag{2.3}$$

Where,

$-\alpha,\ \gamma, z > 0$;

and the r[th]-order moment is given by

$$E_{G_A^0}[Z^r] = \left(\frac{\gamma}{L}\right)^{\frac{r}{2}} \frac{\Gamma\left(-\alpha-\frac{r}{2}\right)\Gamma\left(L+\frac{r}{2}\right)}{\Gamma(-\alpha)\Gamma(L)}, \quad \alpha < -\frac{r}{2} \tag{2.4}$$

For both distributions, α and γ correspond to the roughness and scale parameters, respectively. The α parameter can be interpreted in terms of roughness [1], [3].

2.2.2 Parameter Estimation

As reasoned in [2], the parameter estimation for both G^0_I and G^0_A distributions is performed by method of log cumulants (MoLC). Alternative classical approaches are maximum-likelihood (ML) and method of moments (MoM) [12]:

- The method of logarithmic cumulants (MoLC) parameter estimation approach is specifically developed for positive-valued PDFs. Strategy involved in this method is somewhat similar to classical method of moments. MoLC uses the Mellin integral transform which, is a tool used in studying the distributions of products of nonnegative RVs. MoLC can provide analytical solutions for RVs which come from families with complicated PDF expressions. The Mellin transform and MoLC are well-suited for SAR image because they have speckle present in it which constitute multiplicative model. MoLC generates strongly consistent estimates and

is computationally faster than ML, and it becomes particularly useful when the classical approaches turns out to be unfeasible.

MoLC is based on Mellin transform of a PDF, and it is defined by [12]:

$$\phi_z(s) = \mathcal{M}[p_z(u)] = \int_0^\infty u^{s-1}\, p_z(u) du \tag{2.5}$$

where s is a complex number with unity norm, and $p_Z(u)$ is the PDF of the random variable Z.

The Mellin transform of a function f only exists if f is defined over $\mathbb{R}^+$, and the log cumulant of order v is defined as:

$$\widetilde{k_v} = \left.\frac{d^v \psi_z(s)}{ds^v}\right|_{s=1}, \quad v \in N \tag{2.6}$$

With,

$$\psi_z(s) = \ln\big(\phi_z(s)\big) \tag{2.7}$$

Another important relation is that the log moment of order v can be obtained by:

$$\widetilde{m_v} = \left.\frac{d^v \phi_z(s)}{ds^v}\right|_{s=1}, \quad v \in N \tag{2.8}$$

Here, log cumulants and log moments of orders 1 and 2 are used, which are related by [12]:

$$\widetilde{k_1} = \widetilde{m_1} \tag{2.9}$$

$$\widetilde{k_2} = \widetilde{m_2} - \widetilde{m_1^2} \tag{2.10}$$

The log cumulants of orders 1 and 2 for G^0_I are given by [13]:

$$\widetilde{k_1} = \log\left(\frac{\gamma}{L}\right) + \Psi^0(L) - \Psi^0(-\alpha) \tag{2.11}$$

$$\widetilde{k_2} = \Psi^1(L) + \Psi^1(-\alpha) \tag{2.12}$$

and for G^0_A, the log cumulants of orders 1 and 2 are defined by [14]:

$$2\widetilde{k_1} = \log\left(\frac{\gamma}{L}\right) + \Psi^0(L) - \Psi^0(-\alpha) \tag{2.13}$$

$$4\widetilde{k_2} = \Psi^1(L) + \Psi^1(-\alpha) \tag{2.14}$$

where, in both distributions, $\Psi^0(\cdot)$ is the digamma function, and $\Psi^k(\cdot)$ is the digamma of k[th] order.

The above systems of nonlinear equations either for image intensity or amplitude are solved to estimate α and γ. The sample log moments of orders *v* are given as

$$\widehat{\widetilde{m}_v} = \frac{1}{n}\sum_{i=1}^{n} \log z_i^v \tag{2.15}$$

with $z_i,\ i \in \{1,2,\dots,n\}$, being a sample of a random variable $Z \sim G^0_I$ or $Z \sim G^0_A$. These sample moments are used as estimates of moments in eq 2.9 and 2.10.

2.3 Scale from G^0_A distribution

As described in section 2.2, scale parameter γ as obtained using the MoLC is also explored as independent feature for image segmentation. The amount of backscatter amplitude can be represented by scale parameter and hence scale can also be used as distinct texture representation of the image.

2.4 Roughness and Scale based feature map

Together roughness and scale parameters (as defined in section 2.2) uniquely define SAR images and hence are essential for efficient discriminants of different regions in SAR image. Roughness and scale are used to define a function [4], which is used for segmentation purpose. Let *I* be the image which can be segmented in distinct regions $\{\Omega_1, \Omega_2, \dots \Omega_n\}$. Samples of distinct image regions follow distinct distributions [15]. Thus, when G^0_I or G^0_A distribution is used to describe the image regions, roughness and scale parameters for the segmented image regions will be different. Then, cumulative distribution function (CDF) of these regions can be used to differentiate them. Thus, we can use following function for segmentation purpose:

$$F(z, \alpha, \gamma, L) = \Upsilon_{2L,-2\hat{\alpha}}\left(-\frac{\hat{\alpha}}{\hat{\gamma}} z_m^2\right) \tag{2.16}$$

Here, $\Upsilon_{2L,-2\hat{\alpha}}$ is the Snedecor's F-distribution with numerator degrees of freedom $2L$ and denominator degrees of freedom $-2\hat{\alpha}$. z_m is the optimum value of z that maximises the F-distribution.

Using the above function, energy map ξ is obtained for the entire image. As roughness parameter increases the kurtosis of the energy function also increases, but it is independent of scale parameter. However, variations of the distribution function results when scale parameter increases and it consequently leads to decrement in the energy. Such observations suggests over the boundary regions the amplitude of energy variations should reach the highest values. However, for homogeneous regions these fluctuations should be limited to low values. This discriminatory property of energy variations serves as a reliable feature for segmenting images consisting of homogeneous, heterogeneous and extreme heterogeneous regions. Here, total energy variation is used to evaluate energy oscillations over region boundaries:

$$C = \sum\sum|\nabla\xi| \tag{2.17}$$

Where $|\nabla\xi|$ is the magnitude of the spatial gradient for energy map. The energy function provides the maximal discrimination when

$$z_m = arg\max_z\{C\} \tag{2.18}$$

2.5 Roughness and Scale

As described in section 2.2, roughness and scale parameter as obtained using the MoLC can be used as two dimensional feature vector (for each image pixel) for segmentation od images.

2.6 Shannon entropy

Entropy gives the degree of uncertainty associated with a random variable [16]. Let X be a discrete random variable probability mass function $p(x)$.

For a discrete random variable X, the Shannon entropy $H(X)$ is defined as:

$$H(X) = -\sum_{x\in\chi} p(x)\log p(x) \tag{2.19}$$

Entropy of a region in an image is directly related to its textural uniformity. For texturally non-uniform images, entropy is large. Shannon entropy assumes implicit trade-off between contribution from the main mass and the tails of the distribution. In some applications, explicit control of this trade-off is desirable. Entropy measures that depend on powers of probabilities provides such control. Renyi and Tsallis entropy described in following sections are generalised entropies which provides desired control over shape of the probability distribution.

2.7 Renyi entropy

For a discrete random variable X, the Renyi entropy $I_\alpha(X)$ is defined by [17]:

$$I_\alpha(X) = \frac{1}{1-\alpha}\log\left(\sum_{i=1}^{n} p_i^\alpha\right) \tag{2.20}$$

Here, α is a real parameter and it is used to control the sensitivity of entropy with shape of probability distributions. Renyi entropy is more sensitive for high probable values if α has large positive value and for negative values of α sensitivity shifts towards the tail of the distribution. In the limit as $\alpha \to 1$, Renyi entropy approaches the Shannon entropy.

2.8 Tsallis entropy

For a discrete random variable X, the Tsallis entropy $I_\alpha(X)$ is defined as [17]:

$$S_\alpha(X) = \frac{1}{\alpha-1}\left(1-\sum_{i=1}^{n} p_i^\alpha\right) \tag{2.21}$$

Here, α is a real parameter and it is used to control the sensitivity of entropy with shape of probability distributions. The sensitivity variation with α is same as described in section 2.7. In the limit as $\alpha \to 1$, Tsallis entropy approaches the Shannon entropy.

2.9 Wavelet energy

The wavelet transform is a tool which can perform multiresolution analysis and it can represent the local features of signals with varying details in both time and frequency domain. The wavelet transform has found a wide application in field of image processing applications. Wavelet transform generated features are used for image compression,

denoising, segmentation etc. An image can be decomposed using wavelet transform by carrying out 1-D filtering along directions x and y directions. With first level of decomposition, 4-channel information i.e. LL, HL, LH, and HH are obtained. LL contains low frequency coarser information, HL, LH and HH gives horizontal, vertical, and oblique details of the image, respectively. The dominant spatial- frequency information of the original image is represented by the energy of each channel. Hence, wavelet energy can be regarded as the texture representation of the image. Also, the feature extraction with multi-level wavelet decomposition provides greater details as compared to one level decomposition [6]. The energy of each channel is calculated as:

$$e = \frac{1}{PQ}\sum_{p=1}^{P}\sum_{q=1}^{Q}|x(p,q)| \tag{2.22}$$

Where, $x(p,q)$ is the coefficient value at $(p,q)$and PQ is the size of each channel coefficient matrix.

2.10 Local binary pattern (LBP)

Local binary pattern (LBP) is an effective texture descriptor which compare neighbouring pixels with central pixel and generate binary number as descriptor. For each pixel x_c, binary pattern is computed by thresholding its neighbourhood. It is based on the difference between the feature value of the centre pixel q_c and its circular neighbourhood with specified radius R centred at x_c. Thus, the Local Binary Pattern codes are computed as [18]:

$$LBP_{N,R}(x_c) = \sum_{n=0}^{N-1} s(j)2^n \tag{2.23}$$

where $j = x_p - x_c$ denotes the intensity level difference between the central pixel (x_c) and of the neighbouring pixels (x_p) within the circular neighbourhood of radius R and N neighbouring pixels (Figure 2.1) [19]. Here, $s(j)$ is defined as [20]:

$$s(j) = \begin{cases} 1 & j \geq 0 \\ 0 & otherwise \end{cases} \tag{2.24}$$

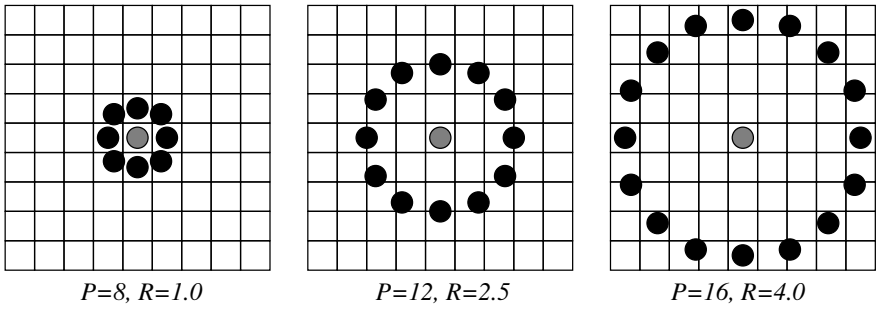


Figure 2.1: Circular Neighborhood of Radius R and P Neighboring Pixels

The LBP code has an added advantage that they are invariant against any monotonic transformation of image brightness. To generate texture descriptors of the image, histograms of these labels are used.

Figure 2.2 [21] depicts the process for computation of LBP using grey-level values for a sample image region.

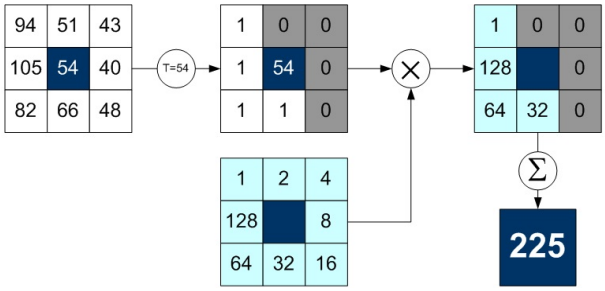


Figure 2.2: Computation of Local Binary Pattern

2.11 Grey-Level Co-occurrence Matrices (GLCM)

GLCM is a statistical textural feature descriptor. Texture is one of the important feature of an image which can be used as an input to various image processing applications [22].

The perception of texture depends on spatial arrangement of grey- level values in a local region. A GLCM is formed by analysing the co-occurrence of pixels with given grey-levels.

The values of the co-occurrence matrix elements denotes the number of times with which two neighbouring pixels separated by distance d and in a given angular direction appear on the image, where one of them has grey level i and other j.

A GLCM is defined as $M(i, j, d, \theta)$, a matrix with a size of $L \times L$ describing how often a pixel with grey value i occurs adjacent to a pixel with the value j. The two pixels are

separated by a distance of d pixels in the direction of θ. L is the number of grey-level present in the image. To avoid computation complexity, number of grey levels in image are often reduced to 4 or 8. Figure 2.3 shows the procedure of constructing GLCM and computing its Homogeneity and Contrast features on a 2D example image.

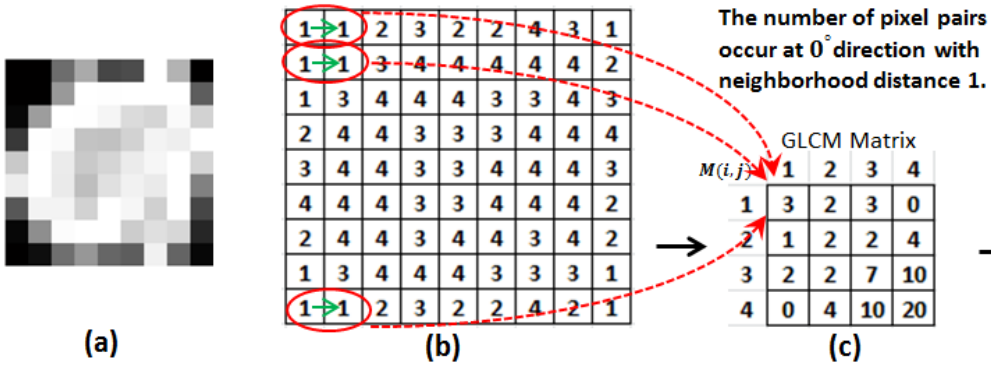


Figure 2.3: Computation of GLCM Matrix

The calculation of co-occurrences matrix of whole image is computationally heavy. To reduce the computational load, its calculations can be performed in small windows scanning over the complete image. In this study, 5x5 size window around centre pixel is used to calculate GLCM.

Following standard GLCM features are computed and used for segmentation purpose:

a. Angular Second Moment (Energy): It provides the measure of uniformity of texture of an image. For constant or periodic grey level distribution, energy reaches its highest value. For homogeneous images very few significant intensity variations occurs and therefore GLCM of this image will have fewer entries of larger magnitude. This will result in significantly higher value of energy feature. For matrix with large number of entries with smaller values, energy feature will have relatively lower values.

$$ASM = \sum_{i=1}^{L}\sum_{i=1}^{L}[p(i,j)]^2 \tag{2.25}$$

b. Contrast: It provides the measure of the amount of local variations present in an image.

$$Contrast = \sum_{n=0}^{L-1} n^2 \left\{ \sum_{\substack{i=1 \\ |i-j|=n}}^{L} \sum_{j=1}^{L} p(i,j) \right\} \tag{2.26}$$

c. Correlation: It quantifies the grey-level relation between pixel and its neighbours. The correlation value is 1 for completely positive ,-1 for completely negative correlated and 0 for uncorrelated image.

$$Corr = \frac{\sum_{i=1}^{L}\sum_{i=1}^{L} ijp(i,j) - \mu_i(i)\mu_j(j)}{\sigma_i(i)\sigma_j(j)} \tag{2.27}$$

d. Sum of squares (Variance): This feature is significant when elements of GLCM vary from the average value of $p(i,j)$

$$SumSquare = \sum_{i=1}^{L}\sum_{j=1}^{L}(i-\mu)^2 p(i,j) \tag{2.28}$$

e. Inverse Difference Moment (Homogeneity): This parameter is the measure of homogeneity and it attains largest value for diagonally concentrated GLCM. Homogeneity have inverse relationship with GLCM contrast

$$Homogeneity = \sum_{i=1}^{L}\sum_{j=1}^{L}\frac{p(i,j)}{1+|i-j|} \tag{2.29}$$

f. Entropy: It is the measure of the disorder present in an image. When all elements in GLCM matrix are equal, its attains its highest value. For texturally non-uniform images entropy value is very small because of presence of many GLCM elements with very small values. Therefore, entropy have inverse relationship with GLCM energy.

$$Entropy = -\sum_{i=1}^{L}\sum_{j=1}^{L} p(i,j)\log_2[p(i,j)] \tag{2.30}$$

g. Sum average

$$SA = \sum_{i=2}^{2L}\left[ip_{x+y}(i)\right] \tag{2.31}$$

h. Sum entropy

$$SE = -\sum_{i=2}^{2L} p_{x+y}(i)\log_2\left[p_{x+y}(i)\right] \tag{2.32}$$

i. Sum variance

$$SV = \sum_{i=2}^{2L} (i - SE)^2 p_{x+y}(i) \tag{2.33}$$

j. Difference Variance

$$DV = variance\ of\ p_{x-y} \tag{2.34}$$

k. Difference Entropy

$$DE = \sum_{i=0}^{L-1} p_{x-y}(i) \log_2 \left[p_{x-y}(i)\right] \tag{2.35}$$

l. Informational measure of correlation 1:

$$IMC1 = \frac{HXY - HXY1}{\max\{HX, HY\}} \tag{2.36}$$

m. Informational measure of correlation 2:

$$IMC2 = \sqrt{1 - e^{-2(HXY2 - HXY)}} \tag{2.37}$$

where:

$p(i,j)$: the probability distribution matrix of co-occurrence matrix $M(i,j,d,\theta)$,

L: the number of discrete intensity levels in the image,

μ: the mean of $p(i,j)$,

$p_x(i) = \sum_{j=1}^{L} p(i,j)$ is the marginal row probabilities,

$p_y(j) = \sum_{i=1}^{L} p(i,j)$ is the marginal column probabilities,

μ_x: the mean of p_x,

μ_y: the mean of p_y,

σ_x: the standard deviation of p_x,

σ_y: the standard deviation of p_y,

$p_{x+y}(k) = \sum_{i=1}^{L} \sum_{j=1}^{L} p(i,j)\,, i + j = k, k = 2,3, \dots, 2L$

$p_{x-y}(k) = \sum_{i=1}^{L} \sum_{j=1}^{L} p(i,j)\,, |i - j| = k, k = 2,3, \dots, 2L$

$HX = -\sum_{i=1}^{L} p_x(i) \log_2 [p_x(i)]$ is the entropy of p_x

$HY = -\sum_{i=1}^{L} p_y(i) \log_2[p_y(i)]$ is the entropy of p_y

$HXY = -\sum_{i=1}^{L} \sum_{j=1}^{L} p(i,j) \log_2[p(i,j)]$ is the entropy of $p(i,j)$

$HXY1 = -\sum_{i=1}^{L} \sum_{j=1}^{L} p(i,j) \log_2[p_x(i)p_y(j)]$

$HXY1 = -\sum_{i=1}^{L} \sum_{j=1}^{L} p_x(i)p_y(j) \log_2[p_x(i)p_y(j)]$

2.12 Central statistical moments

Another method of obtaining texture features is direct computation of moments in local regions from the grey-level image [23]. For purpose of computing central-moments 2-D image is reorganised as a 1-D vector. Then nth order central moment of image function $f(i)$ (where i is discrete) is given as:

$$\mu = m_1 = \frac{1}{N} \sum_i f(i) \tag{2.38}$$

And for $n > 1$:

$$m_n = \frac{1}{N} \sum_i [f(i) - \mu]^n \tag{2.39}$$

Where, N is the total number of pixels.

2.13 Grey-Level Difference Method (GLDM) probability density functions

The basis of GLDM is the occurrence of two pixels separated by a specific displacement and with a given absolute difference in grey level [24]. For any given displacement vector $\delta = (\Delta x, \Delta y)$ let

$$S_\delta(x,y) = |I(x,y) - I(x + \Delta x, y + \Delta y)| \tag{2.40}$$

Where, $I(x,y)$ represents the given image.

Then, estimated probability-density function $D(i|\delta)$ is defined by

$$D(i|\delta) = \text{Prob}[S_\delta(x,y) = i] \tag{2.41}$$

In this work, the vector δ is considered to take four possible forms given as: $(0, d)$, $(-d, d)$, $(d, 0)$, and $(-d, -d)$, where d is the inter-sample spacing. The four forms represent four directions of $[0°, 45°, 90°, 135°]$.

2.14 Local pdf

Based on principle that different regions in image follow different probability distribution function (pdf), another feature is computed by finding the pdf of the grey-level image in local regions. The pdf are found using histograms of image grey-levels.

2.15 Grey-Level Run Length Matrix (GLRLM)

GLRLM is a texture descriptor which depends on the fact that for coarse textures many neighbouring pixels have the same grey level. For fine textures, a low number of neighbouring pixels have the same grey level. In GLRLM, the run length or primitive is maximum number of pixels with same grey level in a given direction [18]. A primitive for a given grey level is characterised by its length l and its direction. The elements of the GLRLM $e(g, l)$ are number of runs for pixels with grey level g and run-length l. For e.g., in horizontal direction, GLRLM is computed by searching number of times all possible grey levels are present in horizontal direction. For a given image, different GLRLM can be computed for different directions. In practice, four different directions horizontal, vertical and two diagonals are used to compute different GLRLM. For each matrix, a 2D run-length histogram ($H_{g,l}$) is computed and is used as a base for creating feature vectors.

If $P(g, l)$ denotes the probability of a specific run-length, several image texture features can be calculated from GLRLM [25]:

a. Short-Run Emphasis (SRE):

   It indicates the dominance of short runs which are present in fine textures. It is defined as:

$$SRE = \sum_{g=0}^{G-1} \sum_{l=1}^{L} \frac{P(g, l)}{l^2} \tag{2.42}$$

b. Long-Run Emphasis (SRE):

   It indicates the dominance of long runs which are present in coarse textures. It is defined as:

$$LRE = \sum_{g=0}^{G-1} \sum_{l=1}^{L} P(g,l) l^2 \tag{2.43}$$

c. Grey-Level Nonuniformity (GLNU):

It is dominant when many grey-level outliers rule over the histogram. GLNU is defined as:

$$GLNU = \sum_{g=0}^{G-1} \left[\sum_{l=1}^{L} P(g,l) \right]^2 \tag{2.44}$$

d. Run-Length Nonuniformity (RLNU):

It is dominant when few grey-level outliers rule over the histogram. RLNU is defined as:

$$RLNU = \sum_{l=1}^{L} \left[\sum_{g=0}^{G-1} P(g,l) \right]^2 \tag{2.45}$$

e. Run Percentage (RP):

It represents the homogeneity of the histogram and obtains maximum value when all runs have unit length. This feature is defined as:

$$RP = \sum_{g=0}^{G-1} \sum_{l=1}^{L} \frac{1}{P(g,l) l} \tag{2.46}$$

f. Low Grey-Level Run Emphasis (LGRE)

This metric attains significant value when the texture is dominated by several runs of low grey-value. It can be defined as:

$$LGRE = \sum_{g=0}^{G-1} \sum_{l=1}^{L} \frac{P(g,l)}{(g+1)^2} \tag{2.47}$$

g. High Grey-Level Run Emphasis (HGRE)

This metric gives significant value when the texture is dominated by several runs of high grey-value. It can be expressed as:

$$HGRE = \sum_{g=0}^{G-1} \sum_{l=1}^{L} P(g,l)(g+1)^2 \tag{2.48}$$

Here, G represents the number of greyscales in which image is quantized and L is the length of longest run.

2.16 Histogram of gradient magnitudes

In this feature descriptor, magnitude of gradients is used to compute histogram in local neighbourhood. In this texture descriptor the orientation information of gradients is ignored[26]. In general, gradient gives indication of changes occurring in the image regions, hence it provides the information of edges present in the image. The effect of ignoring orientation is that this feature descriptor is rotation invariant and is also computationally efficient.

2.17 Variogram

In an image, the pixels may be spatially correlated depending on the region in which they are present. Variogram is an effective measure of spatial correlation present in the data. For an image having N pairs of pixel values and grey level levels $I(x, y)$ positioned at row x and column y, the variogram for pixels separated by a distance h is defined as [18]:

$$2\gamma(h) = \int_x \int_y [I(x,y) - I(x',y')]^2 dydx \tag{2.49}$$

Where h is the Euclidean distance between the pixel value at row x and column y and the pixel value at row x' and y'. In practice, the variogram is approximated as:

$$2\gamma(h) = \frac{1}{N}\sum_{i=1}^{N}[I(x,y) - I(x',y')]^2 \tag{2.50}$$

For many applications, instead of variogram the semi-variogram is computed:

$$\gamma(h) = \frac{1}{2N}\sum_{i=1}^{N}[I(x,y) - I(x',y')]^2 \tag{2.51}$$

While computing variograms, different spatial directions can be chosen. For example, the E-W direction gives:

$$\gamma(h) = \frac{1}{2N}\sum_{i=1}^{N}[I(x,y) - I(x+h,y)]^2 \tag{2.52}$$

The calculation of variogram starts with one pixel offset distance i.e. $h = 1$ and then incrementing the semi-variogram is computed by starting at $h = 1$ (a one-pixel offset), then incrementing the value of h till maximum. The variogram plot is the plot of $\gamma(h)$ as a function of distance h. From this plot, three important attributes which characterise variogram are computed:

a. Nugget: It is the y-intercept of the plot. Though the variogram at zero distance should be zero, but it is non-zero because of the variability at distances smaller than the sample spacing,
b. Sill: It is the maximum variance reached by the variogram.
c. Range: It is the distance at which the sill is reached. For the distances larger than range, pixels are uncorrelated.

Variogram is beneficial because it obtains texture heterogeneity at different scales. Also, this method for textural feature extraction is computationally simple and easy to understand [25].

Chapter 3
Segmentation Techniques

3.1 K-means clustering

Clustering is a technique to group data into different objects based on similarity metric. One of the widely used clustering algorithm is K-means clustering algorithm. It is an unsupervised statistical clustering algorithm based upon the index of similarity or dissimilarity between pairs of data components. It classifies the input data points based on their inherent distance from each other [27]. K-means algorithm is, nondeterministic, iterative and unsupervised technique. The aim of K-means technique is to minimize the sum of squared distances between the cluster centre and all points [28].

Initially, algorithm takes number of clusters as input and arbitrarily locates the cluster centres in the multidimensional feature space. Then distance between each data point and cluster centre is calculated, allocation is done to the cluster for which distance is minimum. After classification of all the pixels, revised mean vectors for each cluster are computed. This process is repeated until there is no significant change in the location of cluster mean vectors between successive iterations of the algorithm [29]. A flow chart demonstrating the steps of algorithm is given in Figure 3.1.

K-means algorithm is advantageous because of its simplicity of implementation and convergence speed, but it has three basic disadvantages [27]:

a. Number of clusters K must be known.
b. Results may vary depending on initial conditions.
c. Noise and outliers affects the performance of algorithm and pull the cluster prototypes away from optimum location.

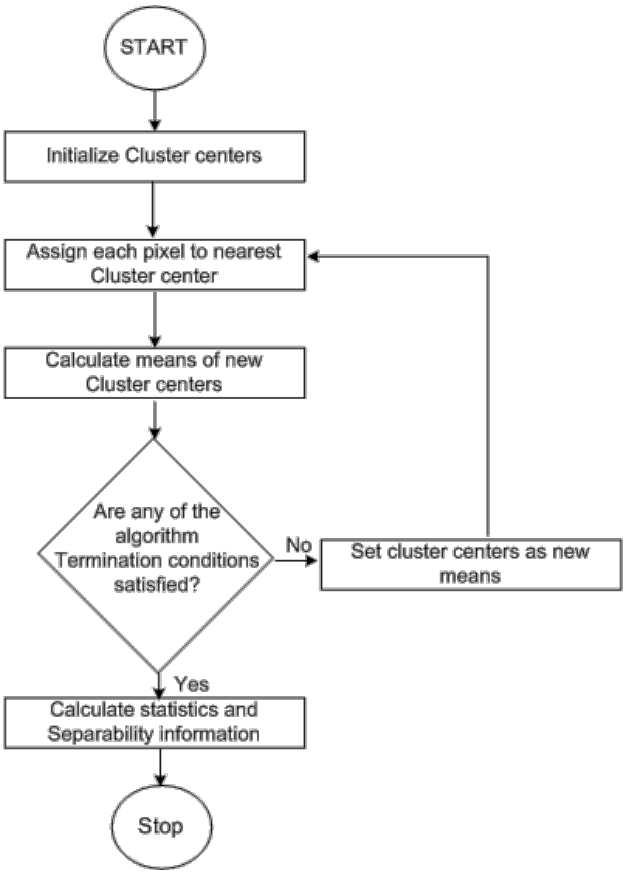


Figure 3.1: K-Means Clustering Flowchart

3.2 Fuzzy C-Means (FCM) clustering

K-means clustering detailed in previous section is a hard clustering technique in which a data point can belong to one cluster only. In contrast, Fuzzy C-Means Clustering (FCM) is a soft clustering technique in which it is possible that a data sample belongs to multiple clusters at the same time. FCM has found wide range of applications namely from data analysis to segmentation of images [30]. In this technique membership value is computed which indicates the similarity of a data point with a cluster. The range of membership values is 0 to 1, more the similarity higher the membership value. At the end of algorithm, a defuzzification is applied to make hard assignments of data to the clusters. FCM is an iterative algorithm which repetitively update the cluster centres and membership values. The update equations for cluster centres and membership values are obtained by solving the cost function.

Let $X = \{x_1, x_2, x_3, \dots x_N\}$ denotes N data samples; it has to be separated into c-clusters by minimizing the following cost function

$$J_m = \sum_{j=1}^{N} \sum_{i=1}^{c} u_{ij}^{m} \left\| x_j - v_i \right\|^2 \tag{3.1}$$

Where, u_{ij} represents the membership of x_j with the i^{th}cluster, v_i is the i^{th} cluster center, $\|.\|$ is Euclidean norm and m is a fuzzification constant which controls the fuzziness of the resulting partition. Higher value of m will increase the fuzziness in the partitioned data. Membership value u_{ij} follows:

$$0 \leq u_{ij} \leq 1 \text{ for all } i, j \tag{3.2}$$

$$\sum_{i=1}^{c} u_{ij} = 1 \text{ for all } j \tag{3.3}$$

$$0 \leq \sum_{j=1}^{n} u_{ij} < n \text{ for all } i \tag{3.4}$$

For minimizing the cost function the derivative of the cost function is taken and equated to zero by using Lagrange constrained optimization method. After solving, following equations are obtained:

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^{m} x_j}{\sum_{j=1}^{N} u_{ij}^{m}} \tag{3.5}$$

$$u_{ij} = \sum_{k=1}^{c} \left(\frac{\left\| x_j - v_i \right\|}{\left\| x_j - v_k \right\|} \right)^{-2} \tag{3.6}$$

Thus, the FCM algorithm is given as follows:

a. Step 1: Input number c of the cluster, fuzzification parameter m and the stopping criterion ε.
b. Step 2: Randomly initialize the cluster centres and fuzzy partition matrix.
c. Step 3: Set the loop counter $n = 0$.
d. Step 4: Calculate the updated cluster centres using equation (3.5).
e. Step 5: Compute updated membership values using equation (3.6).
f. Step 6. If stopping criterion i.e. $\max\{U^n - U^{n+1}\} < \varepsilon$ is reached then stop, otherwise, set $n = n + 1$ and go to step 4.

Conventional FCM do not consider the spatial relationship among the neighbouring pixels and hence its performance deteriorated in presence of noise and outliers. To overcome this limitation many variations of FCM have been proposed which incorporate spatial relationship among pixels in different ways.

3.3 Self-Organising Map (SOM)

A SOM is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional) [31]. Self-organizing maps are based on competitive learning and they preserve the topological properties of the input space using a neighbourhood function. First, weight vectors are initialized randomly. Then a randomly selected sample vector is used to find its best match weight vector. The weight that is best matching is chosen is updated so that it is like that randomly selected sample vector. The neighbours of best matching units are also updated so that they also become more like selected sample vector. With the progresses of algorithm, the size of neighbouring window around winning neuron is decreased. This makes the map to grow in different shapes.

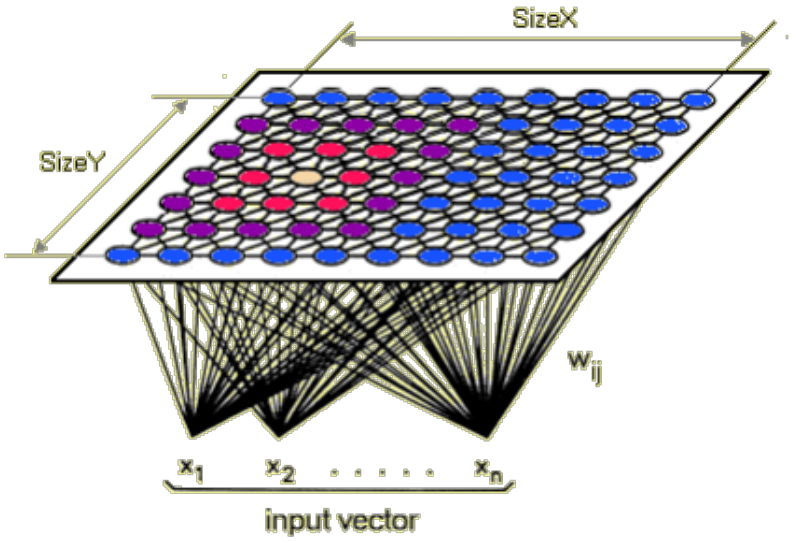


Figure 3.2: Dimensionality Reduction in SOM

The Algorithm:

a. Input number of iterations N and learning rate. Randomly initialize each node's weights.

b. Randomly select a vector from the training data set.

c. From all the nodes, winning neuron i.e. *Best Matching Unit* (BMU) is identified which best represent the input vector.

d. Calculate the neighbourhood of the BMU.

e. The winning weight is rewarded by updating it so that it become more like the sample vector. The neighbours also updated to become more like the sample vector. The weights of nodes which are closer to winning node, gets altered more as compared to the nodes which are the farther.

f. Repeat step b to e for N iterations.

In SOM, the data centres are fixed and predefined, hence it can be considered as constrained K-means clustering.

3.4 Spatial FCM

In an image, spatial information is of great importance. In general, in an image, there is high correlation between its pixels and the probability of them belonging to a particular cluster is significantly high. This spatial information should be incorporated in image analysis techniques to improve performance. A traditional FCM algorithm doesn't use spatial information in the image. The spatial fuzzy c-means (sFCM) modifies the membership function by incorporating this spatial information [7].

The modification is done by defining a spatial function h_{ij}:

$$h_{ij} = \sum_{k \in N(x_j)} u_{ik} \tag{3.7}$$

Where $N(x_j)$ is a square window centered around pixel x_j. The spatial function h_{ij} can be interpreted as the probability of pixel x_j belonging to i^{th} cluster. For pixel x_j, the value of h_{ij} is large if majority of its neighbourhood pixels belongs to the i^{th} cluster. The membership function is updated by incorporating the spatial function into it:

$$u'_{ij} = \frac{u_{ij}^{p} h_{ij}^{q}}{\sum_{k=1}^{c} u_{kj}^{p} h_{kj}^{q}} \tag{3.8}$$

Where u_{ij} is the membership of pixel x_j for the i^{th} cluster. Parameters p & q are used to control the relative importance of membership and spatial function respectively. For a homogenous region, original membership is retained in spatial function, and hence the

results of clustering are same as that obtained by conventional FCM. But for a noisy region, the effect of noisy pixel is reduces by weighing the labels of its neighbouring pixels. This will improve the classification performance for noisy regions. The spatial FCM with parameter p and q is denoted $sFCM_{p,q}$.

Spatial FCM consists of two main steps for each iteration. In first step, membership values are calculated in similar way as done in conventional FCM. In second step, while considering the local neighbourhood the spatial function is computed and membership values are updated. In next iteration of the algorithm the new updated values of membership functions are used. These iterations continues till the stopping criterion is met. For, final assignments of pixels a defuzzification is applied in which pixels are assigned to those clusters for which they have maximum membership value.

The advantages of the technique are:

a. For the case of noisy data, its performance is significantly better than conventional FCM.
b. It will reduce the noisy spots and spurious blobs.

3.5 Fuzzy Local Information C-Means (FLICM) clustering

FLICM is another widely used algorithm which incorporates spatial information in FCM framework. In this algorithm, the objective function of conventional FCM is modified by introducing a local spatial and grey level information based fuzzy factor G_{ki} [8]:

$$J_m = \sum_{i=1}^{N}\sum_{k=1}^{c}[u_{ki}^{m}\|x_i - v_k\|^2 + G_{ki}] \tag{3.9}$$

Where, c represents the number of clusters with $2 \leq c < N$, $\{x_1, x_2, \ldots, x_N\}$ is the data set with N data items, u_{ki} is the membership value of data point x_i for k^{th} cluster, m is the fuzzification factor, v_k is the k^{th} cluster centre.

In FLICM, the fuzzy factor G_{ki} is introduced in cost function and this factor is defined as:

$$G_{ki} = \sum_{\substack{j\in N_i \\ i\neq j}} \frac{1}{d_{ij}+1}(1-u_{kj})^m \left\|x_j - v_k\right\|^2 \tag{3.10}$$

Where, N_i is the local window centred on the i^{th} pixel, k is the reference cluster and the j^{th} pixel belongs to the neighbourhood window centred around i^{th} pixel (N_i). Spatial Euclidean distance between pixels i and j is denoted by $d_{i,j}$.

The factor G_{ki} possesses following characteristics:

a. Introduces robustness and noise insensitiveness by incorporating neighbourhood grey-level information in a fuzzy way.
b. Based on the distance from the central pixel, the effect of the neighbourhood pixels is dampened.
c. It uses original image and avoid pre-processing steps that may lead to missing details.
d. The balance between the image noise and the image details is achieved automatically as no parameter selection is required.

In the algorithm, membership function and cluster centres are updated as:

$$u_{ki} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|x_i - v_k\|^2 + G_{ki}}{\|x_i - v_j\|^2 + G_{ji}} \right)^{\frac{1}{m-1}}} \tag{3.11}$$

$$v_k = \frac{\sum_{i=1}^{N} u_{ki}^m x_i}{\sum_{i=1}^{N} u_{ki}^m} \tag{3.12}$$

The algorithm for FLICM is same as that of FCM as described in Section 3.2 except for using equation (3.11) and equation (3.12) for updating membership values and cluster prototypes respectively. After convergence is reached, fuzzy partition matrix U is changed to crisp partition by a defuzzification process. The pixel i will be assigned to the cluster C for which it have highest membership value:

$$C_i = \arg_k \{\max\{u_{ki}\}\}, \qquad k = 1,2, \dots, c \tag{3.13}$$

3.6 Kernel method Weighted FLICM (KWFLICM)

KWFLICM algorithm is an improvement over FLICM algorithm. It introduces a trade-off weighted fuzzy factor and kernel method in the original FLICM [9].

In KWFLICM, the objective function is modified as:

$$J_m = \sum_{i=1}^{N}\sum_{k=1}^{c} u_{ki}^{m}\left(1 - K(x_i, v_k)\right) + G'_{ki} \tag{3.14}$$

The reformulated fuzzy factor is given by:

$$G'_{ki} = \sum_{i=1}^{N}\sum_{k=1}^{c} u_{ki}^{m} \sum_{\substack{j\in N_i \\ i\neq j}} w_{ij}\left(1 - u_{kj}\right)^{m}\left(1 - K(x_i, v_k)\right) \tag{3.15}$$

Where, N_i is the local neighbours window around pixel x_i , w_{ij} is the trade-off weighted fuzzy factor for pixel j^{th} in the neighbourhood around pixel x_i , $1 - K(x_i, v_k)$ is enhanced kernel method based non-Euclidean distance measure and the penalty term $(1 - u_{ki})^m$ is responsible for accelerating the iterative convergence to some extent. $\{v_k\}_{k=1}^{c}$ is the cluster prototype centres and the array $\{u_{ki}\}$ represents a membership matrix of pixel x_i for cluster $v_{k.}$

The updating formulas for membership values and cluster centres are based on minimizing the cost function $J_{m.}$, the update expressions for u_{ki} and v_k are given as follows:

$$u_{ki} = \frac{1}{\sum_{l=1}^{c}\left(\dfrac{\left(1 - K(p_i, v_k)\right) + \sum_{\substack{j\in N_i \\ i\neq j}} w_{ij}\left(1 - u_{kj}\right)^{m}\left(1 - K(x_i, v_k)\right)}{\left(1 - K(p_i, v_l)\right) + \sum_{\substack{j\in N_i \\ i\neq j}} w_{ij}\left(1 - u_{kj}\right)^{m}\left(1 - K(x_i, v_l)\right)}\right)^{\frac{1}{m-1}}} \tag{3.16}$$

$$v_k = \frac{\sum_{i=1}^{N}(u_{ki}^{m} K(p_i, v_k) x_i)}{\sum_{i=1}^{N}\left(u_{ki}^{m} K(p_i, v_k)\right)} \tag{3.17}$$

3.6.1 Trade-Off Weighted Fuzzy Factor

The trade-off weighted fuzzy factor G'_{ki} defined in KWFICM depends on both local as spatial as well as grey-level constraint. The spatial constraint reflects the extent of dampening of neighbouring pixels. This dampening extent for pixel x_i with coordinate (p_i, q_i) is defined as:

$$w_{sc} = \frac{1}{d_{ij} + 1} \tag{3.18}$$

Where, N_i is the local window around i^{th} pixel and the j^{th} pixel represents pixels falling in the neighbouring window around the i^{th} pixel, d_{ij} represents the spatial Euclidean distance between the j^{th} pixel and i^{th} the central pixel. This definition of the spatial component will effective adjust the influence of the neighbouring pixels according to their distance from the central pixel, which in fact will lead to enhanced local information.

For each pixel j, the local coefficient of variation C_j is defined as follow:

$$C_j = \frac{var(x)}{(\bar{x})^2} \tag{3.19}$$

where $var(x)$ is the variance and $\bar{x}$ is the mean of intensity in a local window of the image.

Next C_j is projected into kernel space. The defined exponential kernel decays rapidly, hence the large distance between C_j and the mean of these local coefficients of variation will lead to nearly zero weights. Subsequently, the weights are normalized. Finally, C_j is compared with $\bar{C}$ (the mean of C_j in local window) to provide varying compensation to C_j, it will lead enlargement in the discrepancy of neighbourhood damping extent.

Thus, the formulas are given as:

$$\bar{C} = \frac{\sum_{j \in N_i} C_j}{n_i} \tag{3.20}$$

$$\xi_{ij} = \exp\left(-\left(C_j - \bar{C}\right)\right), \qquad j \in N_i \tag{3.21}$$

$$\eta_{ij} = \frac{\xi_{ij}}{\sum_{k \in N_i} \xi_{ik}} \tag{3.22}$$

$$w_{gc} = \begin{cases} 2 + \eta_{ij} & C_j < \bar{C} \\ 2 - \eta_{ij} & C_j \geq \bar{C} \end{cases} \tag{3.23}$$

Where N_i is the local window centred around i^{th} pixel, the j^{th} pixel is the pixel falling in this neighbourhood window. The constant 2 is introduced so that the weight w_{gc} is non-negative. The local coefficient of variation is denoted as C_j, this coefficient explains the local distribution of the j^{th} pixel, the mean value of C_j is denoted by $\bar{C}$ and n_i is its local cardinality.

Hence, the trade-off weighted fuzzy factor can be written as:

$$w_{ij} = w_{sc} \times w_{gc} \tag{3.24}$$

The degree of grey-level homogeneity in the local window is reflected by the value of C_j. For edges and noise corrupted areas it will exhibit high values and for homogeneous regions it will have relatively low value.

3.6.2 Non-Euclidean Distance Based on Kernel Metric

FLICM uses Euclidean metrics to obtain distance metrics of objective function. Although Euclidean distance based computations are convenient but its use may lead to degraged segmentation performance for images corrupted by noise and outliers. In KWFLICM a 'kernel method' is used instead of Euclidean distance. A kernel function in the feature space can given as:

$$K(x, y) = \langle \Phi(x). \Phi(y) \rangle \tag{3.25}$$

where $\Phi(*)$ denotes an implicit nonlinear map and $\langle \Phi(x). \Phi(y) \rangle$ refers to the inner product operation.

There are many kernel functions in literature. In KWFLICM Gaussian Radial basis function (GRBF) kernel is used. Its mathematical formulation is as follow:

$$K(x, y) = \exp\left(- \frac{\left(\sum_{i=1}^{d} |x_i - y_i|^a \right)^b}{\sigma} \right) \tag{3.26}$$

Where d represents the dimensions of vector x, σ is the bandwidth of kernel, and $a \geq 0$; $1 \leq b \leq 2$. $K(x, x) = 1$ for all x.

In KWFLICM the distance variance of all data points is used to select bandwidth. For KWFLICM:

$$K(x_i, v_k) = \exp\left(- \frac{\|x_i - v_k\|^2}{\sigma} \right) \tag{3.27}$$

And to calculate σ these steps are followed:

For a given dataset $\Omega = \{x_1, x_2, \ldots, x_N\}$, the data center is given by:

$$\bar{x} = \frac{\sum_{n=1}^{N} x_i}{N} \tag{3.28}$$

Let the distance from data point x_i to the data centre $\bar{x}$ is denoted by $d_i = \|x_i - x\|$. be the distance from data point x_i to the data centre $\bar{x}$. The mean of the distances d_i is then given by:

$$\bar{d} = \frac{\sum_{n=1}^{N} d_i}{N} \tag{3.29}$$

The bandwidth σ is calculated as :

$$\sigma = \left(\frac{1}{N-1} \sum_{i=1}^{N} \left(d_i - \bar{d} \right)^2 \right)^{1/2} \tag{3.30}$$

The distance variance of the data points represents the degree of aggregation around the clusters. For compact clusters which are well separated around clusters, the value of variance is small. While for fuzzy or undistinguished clusters, the variance attains large value.

Then, the distance metric based on kernel method can be written as:

$$D_{ik}^2 = 1 - K(x_i, v_k) = 1 - \exp\left(-\frac{\|x_i - v_k\|^2}{\sigma} \right) \tag{3.31}$$

3.6.3 Framework of KWFLICM Iteration

The algorithm for KWFLICM is same as that of conventional FCM with added computations of trade-off weighted fuzzy factor w_{ij} using equation (3.24) and the modified distance measurement D_{ik}^2 using equation (3.31). After convergence of algorithm, the defuzzification process takes place to obtain crisp segmentation.

3.7 Pixel Intensity and Location information FCM (ILKFCM) clustering

In this section, a variant of FCM algorithm with pixel intensity and location information (ILKFCM) specifically designed for SAR image segmentation is described [6]. This algorithm uses weighted fuzzy factor into the objective function which incorporates a pixel spatial and intensity information. Here, kernel metric is used to measure feature similarity.

The ILKFCM's framework consists of three main, first is the feature extraction, second is weighted fuzzy factor computation, and the last step is ILKFCM iteration.

3.7.1 Weighted Fuzzy Factor

To overcome the limitations of various FCM and its spatial variants for SAR image segmentation, upgradation of FCM objective function is needed. Hence a weighted fuzzy factor is introduced. This factor has three special features given as follows:

a. It simultaneously considers the neighbour pixel intensity and location information. It will improve the algorithm's robustness to speckle and outliers;
b. It needs few parameter selections.
c. Use of the original image avoids the pre-processing steps which may lead to missing image information.

The fuzzy factor consists the intensity distance factor and the spatial distance factor. For every pixel, the location constraint factor reflects the damping extent of the neighbours with the spatial distance from the central pixel and is defined as:

$$w_{sd} = \frac{1}{d_{ij}^{S} + 1} \tag{3.32}$$

Where, all the symbols have conventional meanings as defined in the previous sections also. The above expression makes the influence of the pixels within the local window flexibly change according to their distance from the central pixel, in this way more local information can be used.

Since he Euclidean distance based metrics is not robust for multiplicative noise present in SAR images, a ratio based distance measure is used in this algorithm. This distance measure is termed as ratio relativity and is defined as:

$$d_{ij}^{I} = \frac{1}{M}\sum_{k=1}^{M}\frac{I_{N_i}(k)}{I_{N_j}(k)}, \quad I_{N_j}(k) \neq 0 \tag{3.33}$$

where I_{N_i} and I_{N_j} denote the intensity vectors of two same- sized square image patches N_i and N_j, i.e., which are centred around pixels i and j. M is the total pixel count of the image patches. For smaller distance, ratio relativity attains value that is closer to 1 and for larger distances , it is farther from. Hence, the ratio relativity is mapped as a rational

distance. In this algorithm, a natural logarithm function is used to map the relativity into the intensity distance factor, which is given as:

$$w_{id} = 1 - \log\left(d_{ij}^{I}\right) \tag{3.34}$$

Here constant 1 is included to ensure that intensity distance factor w_{id} is nonnegative.

Hence overall weighted fuzzy factor is given by:

$$w_{ij} = w_{sd} \cdot w_{id} \tag{3.35}$$

ILKFCM algorithm considers the spatial distance and the pixel intensity distance simultaneously, hence it makes use of more local information, and the robust weighted fuzzy factor. These properties makes it more appropriate for SAR image segmentation application.

3.7.2 Framework of ILKFCM Iteration

The objective function of ILKFCM with added fuzzy factor is defined as:

$$J_m = \sum_{i=1}^{N}\sum_{k=1}^{c}[u_{ki}^{m}\|\Phi(p_i) - \Phi(v_k)\|^2 + G_{ki}] \tag{3.26}$$

Where, all the symbols have conventional meanings defined in previous sections. Here, Symbol $\|\cdot\|$ is the Euclidean norm, and $\Phi(\cdot)$ is an implicit nonlinear map. The inner product between $\Phi(p_i)$ and $\Phi(v_k)$ in the feature space is $\Phi(p_i)^T\Phi(v_k) = K(p_i, v_k)$. Through the kernel substitution, we get:

$$\begin{aligned}\|\Phi(p_i) - \Phi(v_k)\|^2 &= \left(\Phi(p_i) - \Phi(v_k)\right)^T\left(\Phi(p_i) - \Phi(v_k)\right)\\ &= \Phi(p_i)^T\Phi(p_i) - \Phi(v_k)^T\Phi(p_i) - \Phi(p_i)^T\Phi(v_k) + \Phi(v_k)^T\Phi(v_k)\\ &= K(p_i, p_i) + K(v_k, v_k) - 2K(p_i, v_k)\end{aligned} \tag{3.37}$$

The algorithm considers Gaussian radial basis function (GRBF) kernel, its use will modify equation 3.37 in following form :

$$\|\Phi(p_i) - \Phi(v_k)\|^2 = 2\left(1 - \mathrm{K}\left(p_i, v_k\right)\right) \tag{3.38}$$

Kernel distance $\mathrm{K}(p_i, v_k)$ is defined as:

$$\mathrm{K}(p_i, v_k) = \exp\left(-\frac{\|p_i - v_k\|^2}{\sigma}\right) \tag{3.39}$$

Where, σ is the bandwidth of the GRBF kernel. The parameter is set on the basis of the distance variance of all feature vectors [6]. Let $D_i = \|p_i - \bar{p}\|$ be the distance from feature vector p_i to feature average $\bar{p}$. Then, we can get the mean distance of D_i as follows:

$$\bar{D} = \frac{1}{N} \sum_{i=1}^{N} D_i \tag{3.40}$$

and σ can be set as:

$$\sigma = \left(\frac{1}{N-1} \sum_{i=1}^{N} (D_i - \bar{D})^2 \right)^{1/2} \tag{3.41}$$

The weighted fuzzy distance G_{ki} is written as follows:

$$G_{ki} = \sum_{j \in N_i, j \neq i} w_{ij} \left(1 - u_{kj}\right)^m \|\Phi(p_i) - \Phi(v_k)\|^2 \tag{3.42}$$

Similar to the FCM iteration, the two necessary conditions for J_m to be at its local minimal extreme, with respect to u_{ki} and v_k, is obtained as follows:

$$u_{ki} = \frac{1}{\sum_{j=1}^{c} \left(\frac{2\left(1 - K(p_i, v_k)\right) + G_{ki}}{2\left(1 - K\left(p_i, v_j\right)\right) + G_{ji}} \right)^{\frac{1}{m-1}}} \tag{3.43}$$

$$v_k = \frac{\sum_{i=1}^{N} (u_{ki}^m K(p_i, v_k) p_i)}{\sum_{i=1}^{N} \left(u_{ki}^m K(p_i, v_k) \right)} \tag{3.44}$$

The algorithm for ILKFCM is same as of conventional FCM with added calculations of weighted fuzzy factor w_{ij} and the kernel distance of the feature vectors, as depicted in equation (3.35) and equations (3.38) & (3.39), respectively. After convergence of algorithm, a defuzzification process takes place in order to convert fuzzy partition matrix U to the crisp segmented image.

3.8 Gaussian Mixture Model (GMM)

Gaussian Mixture Model (GMM) is another type of soft assignment clustering algorithm which assumes Gaussian distribution for the dataset [32]. In GMM, a data point can

belong to multiple clusters with different Gaussian distributions. In effect, each distribution has some 'responsibility' for generating a particular data point.

Due to applicability of central limit theorem, many real life datasets can be modelled by Gaussian Distribution. Hence it is appropriate to assume that different clusters can be modelled as different Gaussian Distributions. In GMM, it is assumed that dataset can be modelled as a mixture of several Gaussian Distributions. The presence of latent variables makes EM algorithm appropriate choice for parameter estimation.

The EM algorithm generally consists E-step or Expectation step and M-step or Maximisation step:

- The E-Step

  The Gaussian Mixture distribution can be written as a combination of Gaussians with weights equal to π. Where K is the number of Gaussians we want to model.

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k) \tag{3.45}$$

  Here K is the number of distributions considered. Next, posterior distribution of the responsibilities can be calculated as:

$$\gamma(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)}$$

- The M-Step

  In M-step, the parameters for each Gaussian are estimated based on the posteriors calculated in E-step. These two E and M steps are then repeated till convergence is acheived.

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(Z_{nk}) x_n \tag{3.46}$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(Z_{nk})(x_n - \mu_k^{new})\,(x_n - \mu_k^{new})^T \tag{3.47}$$

$$\pi_k^{new} = \frac{N_k}{N} \tag{3.48}$$

$$N_k = \sum_{n=1}^{N} \gamma(Z_{nk}) \tag{3.49}$$

$$\ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n|\mu_k, \Sigma_k) \right\} \tag{3.50}$$

3.9 Fuzzy Local Information and L_p- norm distance-based clustering (FLILp)

FLILp is another variant of fuzzy c-means (FCM) clustering algorithm which uses L_p norm distance in objective function. This algorithm has find wide application in image segmentation [10]. The use of L_p- norm will improve performance over the FLICM algorithm. The incorporation of local and grey-scale information and spatial information enhances the performance of algorithm in presence of noise and outliers. L_p-norm distance is approximated by weighted L_2 distance to avoid mathematical inconvenience.

Considering the case of grey-level images, i.e. $x_i \in R$. Let $p \in (0,1]$. We denote the local window centred at x_i by N_i. Let u_{kj} be the degree of membership of the j^{th} pixel in the k^{th} cluster with centre v_k. The objective function of the FLILp technique is given as:

$$J_m = \sum_{i=1}^{N} \sum_{k=1}^{c} u_{ki}^{m} (\|x_i - v_k\|^p + \alpha G_{ki}) \tag{3.51}$$

Where, N_i is the local window centred around pixel x_i, u_{kj} represents the membership value of j^{th} pixel for the k^{th} cluster with centre v_k, $\alpha > 0$ provides noised robustness by controlling the contribution of the local factor (α=1 in FLICM). Large values of α is used for higher noise-levels. In equation (3.51) $\|\cdot\|$ represents the Euclidean distance. L_p-norm-based local factor G_{ki} is introduced to incorporate the local information, this factor is given as:

$$G_{ki} = \sum_{j \in N_i} K_{ij} \left(1 - u_{kj}\right)^m \|x_i - v_k\|^p \tag{3.52}$$

One choice of K_{ij} is,

$$K_{ij} = \begin{cases} 0, & i = j \\ \dfrac{1}{d_{ij}+1} & i \neq j \end{cases} \tag{3.53}$$

Where d_{ij} represents spatial Euclidean distance between the i^{th} and j^{th} pixel.

The update formula for membership functions is:

$$u_{ki} = \frac{1}{\sum_{j=1}^{c} \left(\frac{D_{ki}}{D_{kj}}\right)^{\frac{1}{m-1}}} \tag{3.54}$$

Where,

$$D_{ki} = \|x_i - v_k\|^p + \alpha G_{ki} \tag{3.55}$$

And cluster centres are updated using:

$$v_k = \frac{\sum_{i=1}^{N} w_{ki}\, x_i}{\sum_{i=1}^{N} w_{ki}} \tag{3.56}$$

Where,

$$w_{ki} = \frac{u_{ki}^m + \alpha(K\, u_{ki}^m)(1-u_{ki})^m}{\left\|x_i - v_k^e\right\|^{2-p}} \tag{3.57}$$

The algorithm for FLILp is same as that of conventional FCM while using equation (3.54) to update the membership functions and equation (3.56) to update the centres of clusters. The final crisp segmentation result is obtained by assigning the pixel i to the cluster C for which it has maximum membership value. It can be given as:

$$C_i = \arg_k\{\max\{u_{ki}\}\}, \qquad k = 1,2,\dots,c \tag{3.58}$$

The advantage of FLILp are that it is comparatively robust to noise and outliers. Also, it is applied to original image and hence is computationally efficient.

3.10 Modified FLICM

In Modified FLICM, an enhanced fuzzy factor is used, in which weighted membership value is used to indicate the implicit Grey level information of the neighbourhood. Since, this factor does not use the Grey level values explicitly it is more robust for the case of SAR images.

The implicit Grey level information is incorporated using two steps procedure. First, for pixel x_j and cluster k, weight w_{kj} is calculated by averaging the membership values of neighbouring pixels (for cluster k) in the local window centred on the current pixel x_j:

$$w_{kj} = \sum_{\substack{l \in N(x_j) \\ l \neq k}} u_{kl} \tag{3.59}$$

Where $N(x_j)$ is a square window centred on pixel x_j.

The weights are calculated for each pixel and cluster combination. For given cluster c and pixel x_j , w_{cj} is large if majority of pixels in the neighbourhood of pixel x_j belong to cluster c.

In second step, these weights w_{kj} are used to update the membership values u_{kj} of the pixels:

$$u'_{kj} = \frac{u_{kj} w_{kj}}{\sum_{l=1}^{c} u_{lj} w_{lj}} \tag{3.60}$$

If majority of non-noisy neighbouring pixels of x_j belongs to cluster c, then the effect of weight w_{kj} will adjust membership value of pixel x_j towards cluster c. This way of weighing gives consistent membership values when centre pixel x_j or few of its neighbours are noisy. Hence with the incorporation of w_{kj} the algorithm becomes more robust for noisy SAR images.

Enhanced fuzzy factor with weighted membership function is defined as

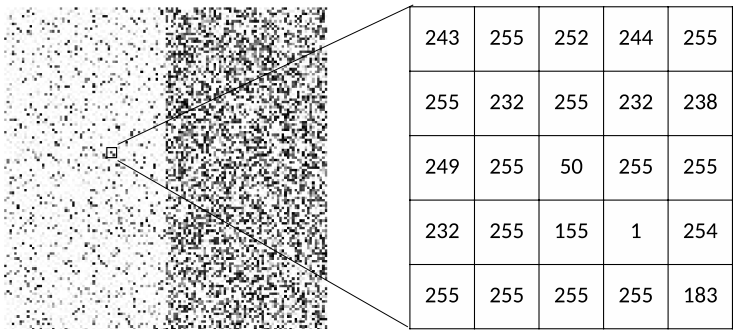
$$G'_{ki} = \sum_{\substack{j \in N_i \\ i \neq j}} \frac{1}{d_{ij} + 1} \left(1 - u'_{kj}\right)^m \left\| x_j - v_k \right\|^2 \tag{3.61}$$

And the cost function with enhanced fuzzy factor is,

$$J_m = \sum_{i=1}^{N} \sum_{k=1}^{c} [u_{ki}^m \| x_i - v_k \|^2 + G'_{ki}] \tag{3.62}$$

At each iteration the algorithm calculates membership functions using equation (3.11) replacing G_{ij} by G'_{ij} and then updates it using equation (3.60). Cluster prototypes are

calculated using equation (3.12) replacing u_{ij} by u'_{ij}. Equation (3.62) is used to calculate the cost function. The iterations continue until the algorithm converges.



243	255	252	244	255
255	232	255	232	238
249	255	50	255	255
232	255	155	1	254
255	255	255	255	183

(a)

0.435	0.566	0.545	0.467	0.448
0.387	0.530	0.542	0.437	0.436
0.385	0.539	0.498	0.461	0.477
0.455	0.491	0.519	0.473	0.492
0.489	0.478	0.486	0.503	0.531

Initial

0.856	0.890	0.892	0.846	0.817
0.855	0.778	0.755	0.702	0.672
0.851	0.746	0.479	0.634	0.630
0.892	0.754	0.561	0.448	0.620
0.914	0.793	0.739	0.705	0.617

Final

(b)

0.517	0.559	0.490	0.550	0.505
0.509	0.531	0.557	0.570	0.569
0.471	0.453	0.535	0.529	0.456
0.397	0.379	0.430	0.458	0.426
0.418	0.395	0.417	0.420	0.424

Initial

0.992	0.992	0.994	0.994	0.973
0.999	0.974	0.984	0.991	0.981
0.999	0.953	0.806	0.884	0.921
0.997	0.987	0.915	0.914	0.983
0.991	0.998	0.971	0.981	0.987

Final

(c)

Figure 3.3: a) Noisy image and Grey level values for 5x5 patch; (b) Initial and final membership values using FLICM; (c) Initial and final membership values using Modified FLICM (after 50 iterations)

The effect of adding smoothing weight factor can be understood as follows. If a neighbourhood pixel 'A' is noisy, then its membership value for a given cluster will be significantly different from centre pixel and other neighbouring non-noisy pixels. If this membership value is directly used to calculate fuzzy factor G_{ki}, it will increase the value of cost function and will also affect the partition matrix and cluster prototypes calculations. But if its weighted membership value is used in calculation of G_{ki}, the effect

of noisy membership value smooth out. This situation is further illustrated in Figure 3.3. A 5×5 patch is taken from a noisy image. In this patch centre pixel as well as neighbouring pixels are noisy. With FLICM, the membership value of the centre pixel and its neighbouring noisy pixel did not converge towards true cluster as shown in Figure 3.3(b). But with Modified FLICM, membership values of all pixels converged towards true cluster as shown in Figure 3.3(c).

3.11 Modified KWFLICM

KWFLICM takes into account both grey level and spatial distance information to incorporate spatial context information. Modified KWFLICM uses neighbourhood information in similar way as specified in KWFLICM (Section 3.6) , but it also enhance this neighbourhood information using membership value of adjacent pixels as done in sFCM. Hence, this technique takes advantage of both KWFLICM and Spatial FCM algorithms. At each iteration the algorithm updates the cluster prototypes and membership functions as is done in KWFLICM algorithm. At end of the KWFLICM iteration the membership functions are modified using membership function of neighbouring pixel as is done in Spatial FCM using equations (3.7) and (3.8). The iterations continue until the algorithm converges.

3.12 Modified ILKFCM

Modified ILKFCM is reduced version of ILKFCM. It is based on the observation that weight w_{id} as defined using equations (3.34) and (3.35) is not normalised i.e. it may fall outside the range $[0,1]$. Experimentally it was found that this weight affect the clustering result severally in some cases. This is due to the fact that a value more than 1 or less than 0 can adversely affect the clustering performance of the algorithm.

The modified ILKFCM algorithm is same as ILKFCM except the fact that weighted fuzzy factor is computed as:

$$w_{ij} = w_{sd} \tag{3.63}$$

3.13 Modified Spatial FCM

In spatial FCM, to exploit the spatial information; a spatial function is used to modify the membership function at each iteration. However, this technique do not incorporates the

damping extent of neighbouring pixels based on their distance from centre pixel. Hence, to further enhance the representation of this spatial information, the elements of the spatial function are weighted according to their Euclidean distance from the current pixel. The weights are defined as:

$$w_{jk} = \begin{cases} 0, & j = k \\ \dfrac{1}{d_{jk} + 1} & j \neq k \end{cases}, where\ k \in N(x_j) \tag{3.64}$$

Where $N(x_j)$ represents a square window centered on pixel x_j in the spatial domain and d_{jk} is the spatial Euclidean distance of the j^{th} pixel and k^{th} pixel.

Thus the spatial function is given by:

$$h_{ij} = \sum_{k \in N(x_j)} w_{jk}\, u_{ik} \tag{3.65}$$

At each iteration, the algorithm updates the cluster prototypes and membership functions as is done in FCM algorithm. At end of the FCM iteration the membership functions are modified using h_{ij} as is done in Spatial FCM using equations (3.8). The iterations continue until the algorithm converge.

Chapter 4
Performance Evaluation of Segmentation

In the project, two different methodologies have been used for performance evaluation of synthetic images and real SAR images.

4.1 Performance Evaluation for Synthetic Images

The Ground Truth data is readily available in case of synthetic images and hence a confusion/error matrix can be computed.

Suppose there are r segments in the image, given by $S_1, S_2, \dots, S_r$. Then the confusion matrix has dimensions of $r \times r$ and is given by [33]:

$$CM = \begin{bmatrix} cm_{11} & cm_{12} & \cdots & cm_{1r} \\ cm_{21} & cm_{22} & \cdots & cm_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ cm_{r1} & cm_{r2} & \cdots & cm_{rr} \end{bmatrix} \tag{4.1}$$

In matrix CM, rows correspond to ground truth and columns to segmented results. Element cm_{ij} is equal to count of pixels belonging to segment S_i and classified as segment S_j. Therefore, the row total is equal to the pixel count of corresponding image segment in ground truth whereas column total is given by the pixel count of corresponding image segment in segmented result.

An confusion is an efficient means of reflecting accuracy/inaccuracy in segmentation, because it provides a clear way of deriving the individual accuracies of each segment as well as errors related to false positives i.e. wrongly assigning a class to an image area(commission errors) and false negatives i.e. wrongly omitting an image area from the class to which it belongs (omission errors) in the segmented image. [34].

The overall accuracy, normalised accuracy and Kappa statistics can be computed from the error matrix for assessing the segmentation performance.

4.1.1 Overall Accuracy

The major diagonal of the error matrix represents the count of correctly segmented pixels. The overall accuracy is given by trace (i.e., the correctly segmented pixels) of the error matrix divided by the pixel count (N) in the confusion matrix. It can be computed as:

$$OA = \frac{1}{N}\sum_{i=1}^{r} c_{ii} \tag{4.2}$$

4.1.2 Normalised Accuracy

The error matrix is "normalised" or standardized using an algorithm called Margfit [34]. An iterative procedure using proportional fitting is utilised to force marginals i.e. rows and columns in the matrix, such that each row/column sum equals some predetermined value; hence the name Margfit-'marginal fitting'. When this value is equal to 1, matrix elements lie in the range [0,1] and can be converted to percentage value by multiplying with 100. Otherwise if this value if set to 100, directly percentage values are obtained.

The normalization process eliminates the effect of any difference existing in size of individual classes; making the value of individual cells directly comparable with each other. During iterations, sum of rows and columns is calculated which is used to normalise the matrix, thus the normalized error matrix is more representative of the values of the off-diagonal cells (omission and commission errors). During the process, cell values along the major matrix diagonal also gets modified. Overall accuracy computed for this normalised matrix in termed as normalized accuracy.

Normalized accuracy incorporates information from major diagonal as well as off diagonal elements, thus when compared with overall accuracy, it is a better accuracy metric.

4.1.3 Kappa Statistics

Kappa analysis is used for comparing two error matrices statistical analysis [34]. The comparison is based on $\widehat{K}$ statistic (i.e. estimated Kappa). $\widehat{K}$ can also be used as metric of agreement. $\widehat{K}$ is based on difference between the actual agreement and chance agreement in the error matrix. Actual agreement is given by major diagonal of the error matrix whereas row and column totals indicate the chance agreement.

$\widehat{K}$ statistic can be computed using following equations:

Let p_{ij}, p_{i+}, and p_{+j}

$$p_{ij} = \frac{c_{ij}}{N} \tag{4.3}$$

$$p_{i+} = \sum_{j=1}^{r} p_{ij} \tag{4.4}$$

$$p_{+j} = \sum_{i=1}^{r} p_{ij} \tag{4.5}$$

Then, the actual agreement is given by:

$$p_a = \sum_{i=1}^{r} p_{ii} \tag{4.6}$$

And the chance agreement is given by:

$$p_c = \sum_{i=1}^{r} p_{i+}\, p_{+j} \tag{4.7}$$

Then the $\widehat{K}$ is given by:

$$\widehat{K} = \frac{p_a - p_c}{1 - p_c} \tag{4.8}$$

Combining above equations, we get:

$$\widehat{K} = \frac{N \sum_{i=1}^{r} c_{ii} - \sum_{i=1}^{r} c_{i+}\, c_{+i}}{N^2 - \sum_{i=1}^{r} c_{i+}\, c_{+i}} \tag{4.9}$$

Where, c_{i+} is row sum and c_{+i} is the column sum:

$$c_{i+} = \sum_{j=1}^{r} c_{ij} \tag{4.10}$$

$$c_{+j} = \sum_{i=1}^{r} c_{ij} \tag{4.11}$$

The range of KHAT is [-1,1]. However, when used for current purpose of indication of accuracy, $\widehat{K}$ is expected to lie in the range [0,1] as positive correlation between the segmentation results and ground truth data exists.

4.2 Performance Evaluation for Real SAR Images

For real SAR test images, cross-region fitting index (CRF) is used to assess the segmentation performance. CRF is used as ground truth data, which may not be available for the real images, is not required for its calculation. The values CRF lie in the range [0, 1], where CRF = 1 indicates the best segmentation result. The CRF depend on Difficulty of Segmentation (DoS). On the basis of the contrast between foreground and background regions, DoS quantifies the difficulty in segmenting the two regions. CRF and DoS indexes depend on the arithmetic-geometric distance between samples from the two regions [35, 36].

- Arithmetic-Geometric Distance

  CRF and DoS are computed based on arithmetic-geometric distance. For speckled data this distance provides the best discrimination ability in performing the contrast quantification [36].

  The arithmetic-geometric distance between $Z_1 \in \Omega_1$ and $Z_2 \in \Omega_2$ is given by [4]:

$$S_{AG}(Z_1, Z_2) = \frac{1}{2}\int_{\mathbb{R}^+} \left[\left(f_{G_I^0}(z, \theta_1, n) + f_{G_I^0}(z, \theta_2, n)\right) \times \log\left(\frac{f_{G_I^0}(z, \theta_1, n) + f_{G_I^0}(z, \theta_2, n)}{2\sqrt{f_{G_I^0}(z, \theta_1, n) \times f_{G_I^0}(z, \theta_2, n)}}\right)\right] \quad (4.12)$$

- Segmentation Distance

  Segmentation distance is an indicator of the statistical distance between the reference region and segmentation result. It can be expressed as [4]:

$$SD = S_{AG}\left(Z_f^r, Z_f^s\right) \quad (4.13)$$

  where $Z_f^r \in \Omega_f^r$ is the reference foreground and $Z_f^s \in \Omega_f^s$ is segmented area/region.

- Difficulty of Segmentation (DoS)

  DoS quantifies the difficulty in segmenting the two regions, the foreground (Ω_f) and background (Ω_b) in present study. Low region contrast implies small $S_{AG}(Z_f,Z_b)$; and thus greater difficulty in segmentation them. It is calculated as [4]:

$$DoS = \frac{1}{S_{AG}(Z_f, Z_b)} \tag{4.14}$$

- Cross Region Fitting (CRF)

  It quantifies the ability of the technique to segment the image areas/regions correctly. CRF can be calculated as [4]:

$$CRF = \frac{1}{1 + \sqrt{DoS \times \left| S_{AG}(Z_f^r, Z_b^s) - S_{AG}(Z_f^s, Z_b^r) \right|}} \tag{4.15}$$

Chapter 5
Experiments and Results

To analyse segmentation performance of different segmentation techniques with different feature combinations, the experiments were conducted on synthetic as well as real SAR images. For all the features extraction methods requiring neighbourhood operations, the size of window is taken to be 5x5 until specified otherwise.

5.1 Synthetic Data Generation

First, for conducting systematic Monte-Carlo experiments, synthetic SAR images are generated. Here, G^0_A distribution is used to model different regions in the synthetic images as it can accurately model homogeneous, moderately heterogeneous, and extremely heterogeneous areas in a real SAR image. The greyscale synthetic SAR images of size 128×128 pixels contain a foreground rectangular patch of size 90×91 pixels, resulting in approximately equal number of foreground and background pixels. Single look ($L = 1$) synthetic images are used to analyse the performance of segmentation. 72 different greyscale images of same size and shape with different foreground (f) and background (b) regions following the $G^0_A\ (\alpha_f, \gamma_f, L)$ and $G^0_A\ (\alpha_b, \gamma_b, L)$ distributions, respectively are used. To generate images various values of (α_f, γ_f) and (α_b, γ_b) were used. For generating the synthetic images, following equation is used [4]:

$$Z = \sqrt{-\frac{\gamma}{\alpha}\Upsilon_{M,N}^{-1}(U)} \tag{5.1}$$

Where, $M = 2L, N = -2\alpha$ and $\Upsilon_{M,N}^{-1}$ is the inverse cumulative distribution function of a Snedecor's $F_{M,N}$ distributed random variable with M and N degrees of freedom, and U is a uniformly distributed random variable over [0, 1]. As result amplitude images are obtained. The intensity image is formed by squaring the amplitude values (Z).

5.2 Results with Synthetic Data

A total of 72 synthetic images were generated. The parameters used for the synthetic images are listed in Table 5.1. The table also lists the Difficulty of Segmentation (DoS) parameter for these images as computed using the relations given in Section Chapter 4.

Table 5.1: Parameters for Synthetic Images

Image Name	Foreground		Background		DoS
	α	γ	α	γ	24.44
Synth_1	-1	1	-1	10	15.55
Synth_2	-1	1	-1	25	2.81
Synth_3	-1	1	-4	1	76.66
Synth_4	-1	1	-4	10	29.51
Synth_5	-1	1	-4	25	1.83
Synth_6	-1	1	-10	1	200.68
Synth_7	-1	1	-10	10	92.86
Synth_8	-1	1	-10	25	24.24
Synth_9	-1	10	-1	1	269.31
Synth_10	-1	10	-1	25	1.57
Synth_11	-1	10	-4	1	108.71
Synth_12	-1	10	-4	10	1734.96
Synth_13	-1	10	-4	25	0.5
Synth_14	-1	10	-10	1	12.33
Synth_15	-1	10	-10	10	93.15
Synth_16	-1	10	-10	25	15.65
Synth_17	-1	25	-1	1	523.63
Synth_18	-1	25	-1	10	1.38
Synth_19	-1	25	-4	1	48.23
Synth_20	-1	25	-4	10	164.49
Synth_21	-1	25	-4	25	0.41
Synth_22	-1	25	-10	1	9.54
Synth_23	-1	25	-10	10	43.08
Synth_24	-1	25	-10	25	2.96
Synth_25	-4	1	-1	1	1.57
Synth_26	-4	1	-1	10	1.4
Synth_27	-4	1	-1	25	1.96
Synth_28	-4	1	-4	10	1.63
Synth_29	-4	1	-4	25	20.21

Synth_30	-4	1	-10	1	4.06
Synth_31	-4	1	-10	10	2.06
Synth_32	-4	1	-10	25	81.31
Synth_33	-4	10	-1	1	100.89
Synth_34	-4	10	-1	10	40.99
Synth_35	-4	10	-1	25	2.01
Synth_36	-4	10	-4	1	179.82
Synth_37	-4	10	-4	25	0.88
Synth_38	-4	10	-10	1	29.84
Synth_39	-4	10	-10	10	8710.55
Synth_40	-4	10	-10	25	30.03
Synth_41	-4	25	-1	1	1968.63
Synth_42	-4	25	-1	10	157.01
Synth_43	-4	25	-1	25	1.64
Synth_44	-4	25	-4	1	192.61
Synth_45	-4	25	-4	10	0.57
Synth_46	-4	25	-10	1	14.07
Synth_47	-4	25	-10	10	164.8
Synth_48	-4	25	-10	25	1.72
Synth_49	-10	1	-1	1	0.51
Synth_50	-10	1	-1	10	0.43
Synth_51	-10	1	-1	25	19.68
Synth_52	-10	1	-4	1	0.91
Synth_53	-10	1	-4	10	0.58
Synth_54	-10	1	-4	25	2.12
Synth_55	-10	1	-10	10	0.92
Synth_56	-10	1	-10	25	145.51
Synth_57	-10	10	-1	1	11.84
Synth_58	-10	10	-1	10	8.82
Synth_59	-10	10	-1	25	3.89
Synth_60	-10	10	-4	1	30.1
Synth_61	-10	10	-4	10	14.26
Synth_62	-10	10	-4	25	2.3
Synth_63	-10	10	-10	1	29.08

Synth_64	-10	10	-10	25	95.84
Synth_65	-10	25	-1	1	102.72
Synth_66	-10	25	-1	10	45.06
Synth_67	-10	25	-1	25	2.02
Synth_68	-10	25	-4	1	7899.06
Synth_69	-10	25	-4	10	186.71
Synth_70	-10	25	-4	25	0.89
Synth_71	-10	25	-10	1	31.03
Synth_72	-10	25	-10	10	24.44

The ground truth for all the synthetic images is as per Figure 5.1. Each image is a 200×200 pixel image. In each image the foreground occupies 140×140 pixels.

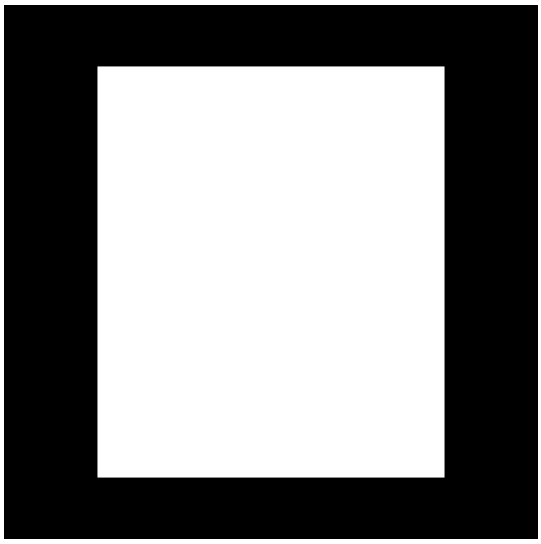


Figure 5.1: Ground Truth for Synthetic Images

For each image, features listed in Section Chapter 2 were computed and for each of these features, segmentation techniques listed in Section Chapter 3 were applied to segment the image.

For each feature-technique combination accuracy of segmentation, normalized accuracy and Kappa coefficient were computed. The segmentation performance of the techniques is given in subsequent sub-sections. The performance is shown in terms of feature-wise plots of DoS (of 72 images) vs segmentation accuracy.

5.2.1 Intensity

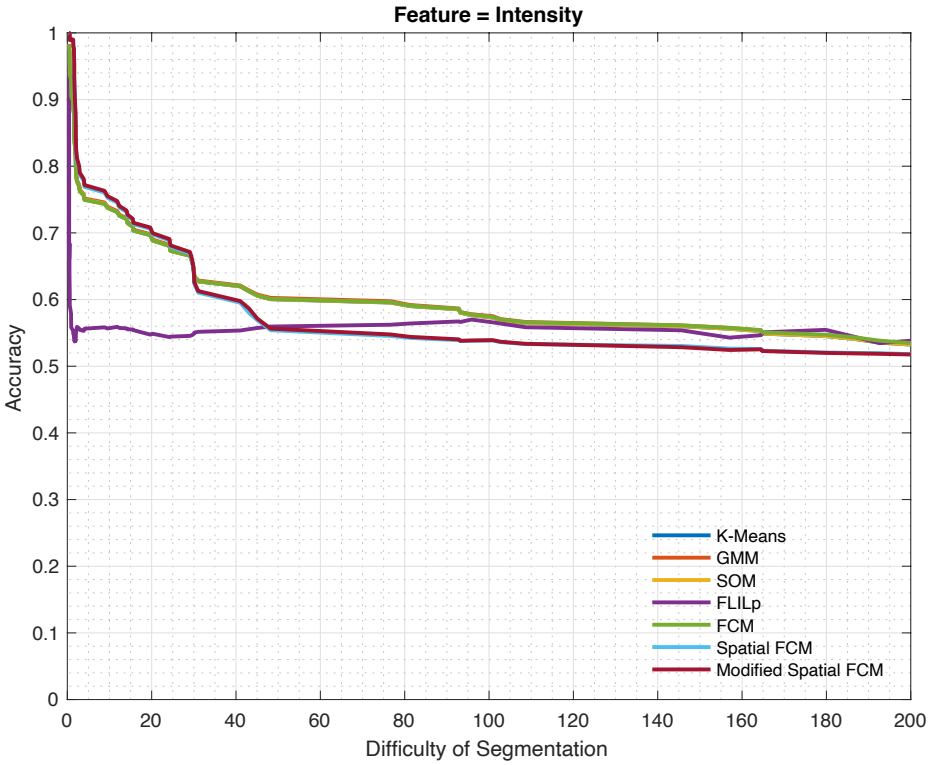


Figure 5.2: DoS vs Segmentation Accuracy for Feature 'Intensity' (1)

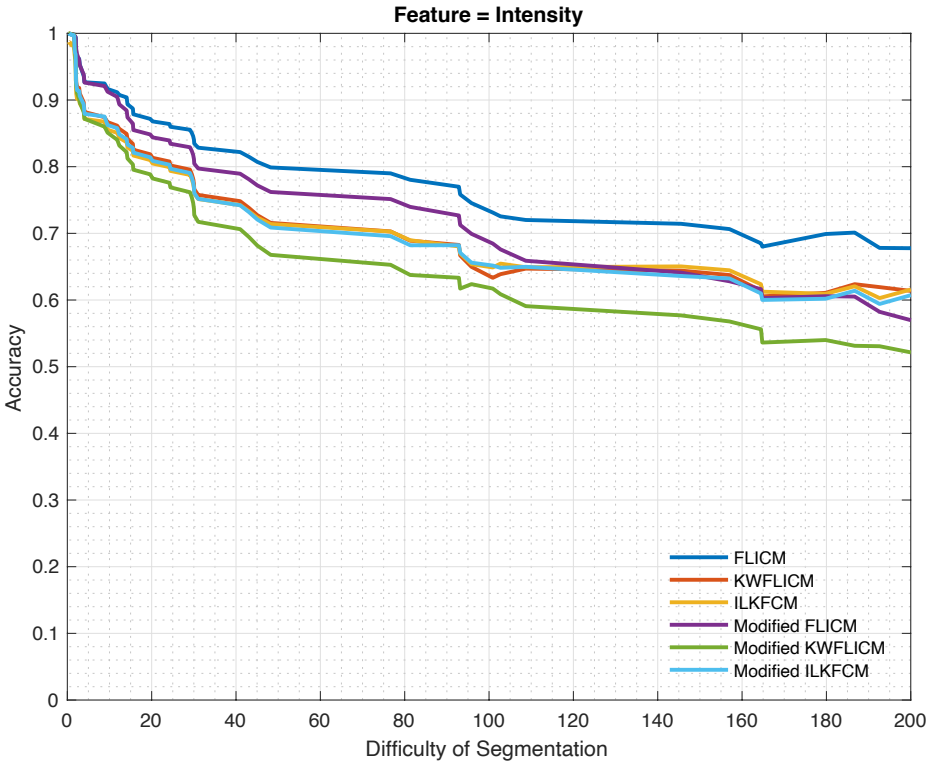


Figure 5.3: DoS vs Segmentation Accuracy for Feature 'Intensity' (2)

5.2.2 Roughness

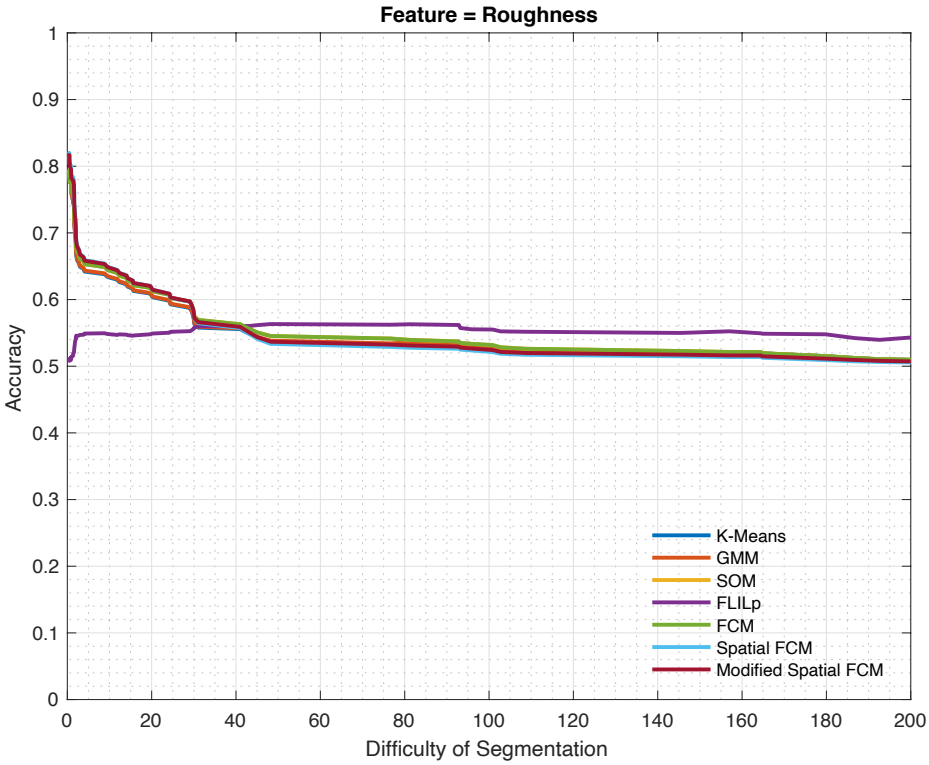


Figure 5.4: DoS vs Segmentation Accuracy for Feature 'Roughness' (1)

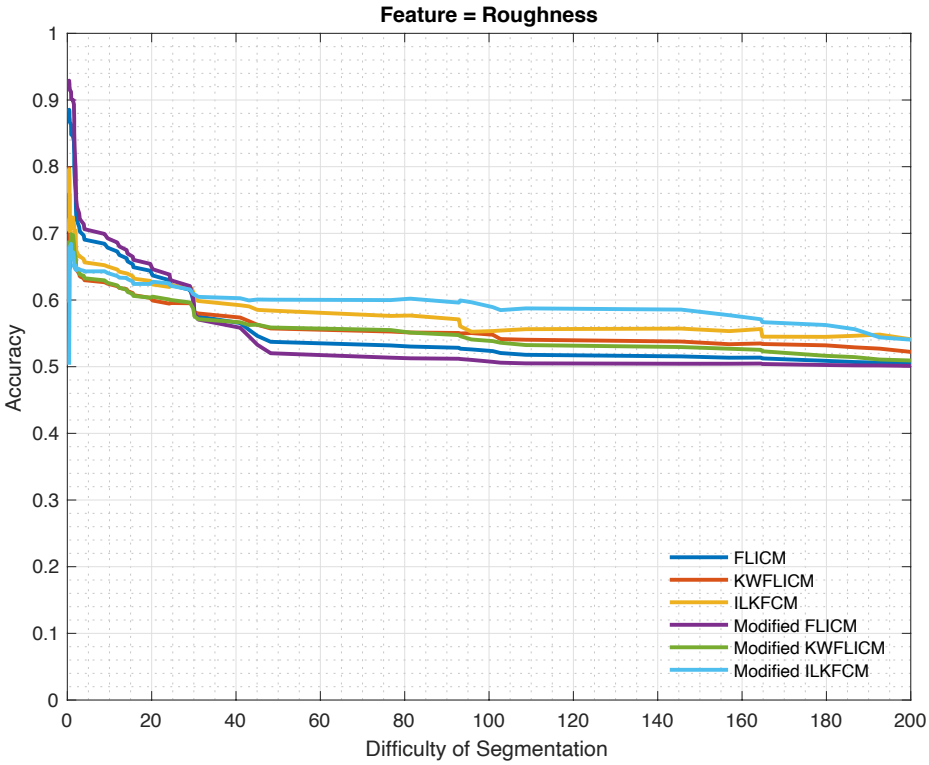


Figure 5.5: DoS vs Segmentation Accuracy for Feature 'Roughness' (2)

5.2.3 Scale

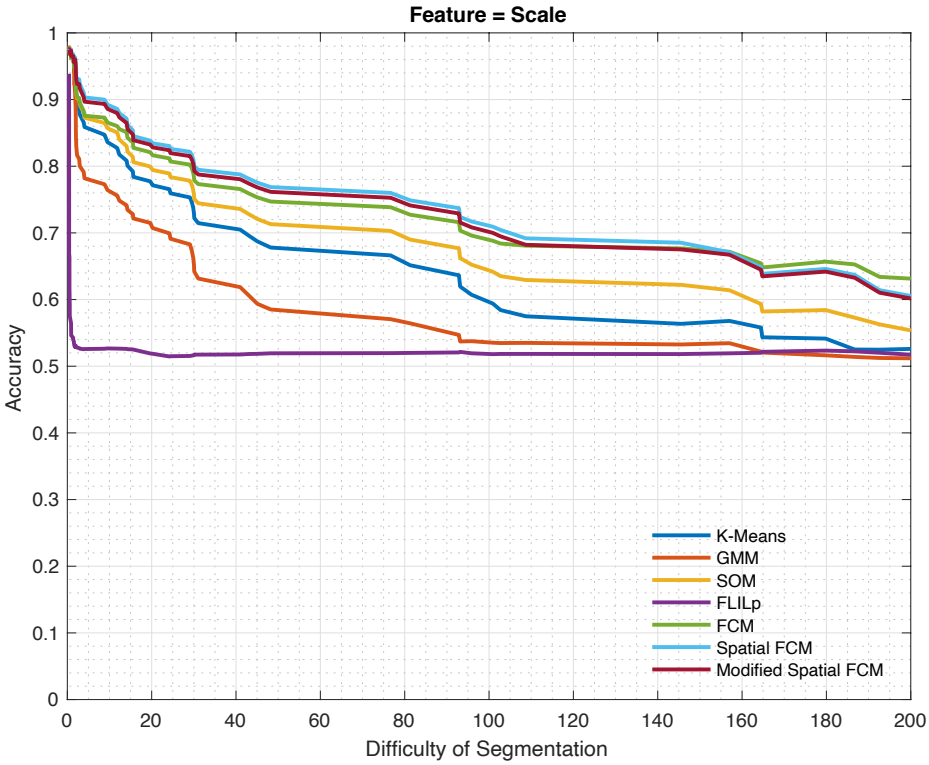


Figure 5.6: DoS vs Segmentation Accuracy for Feature 'Scale' (1)

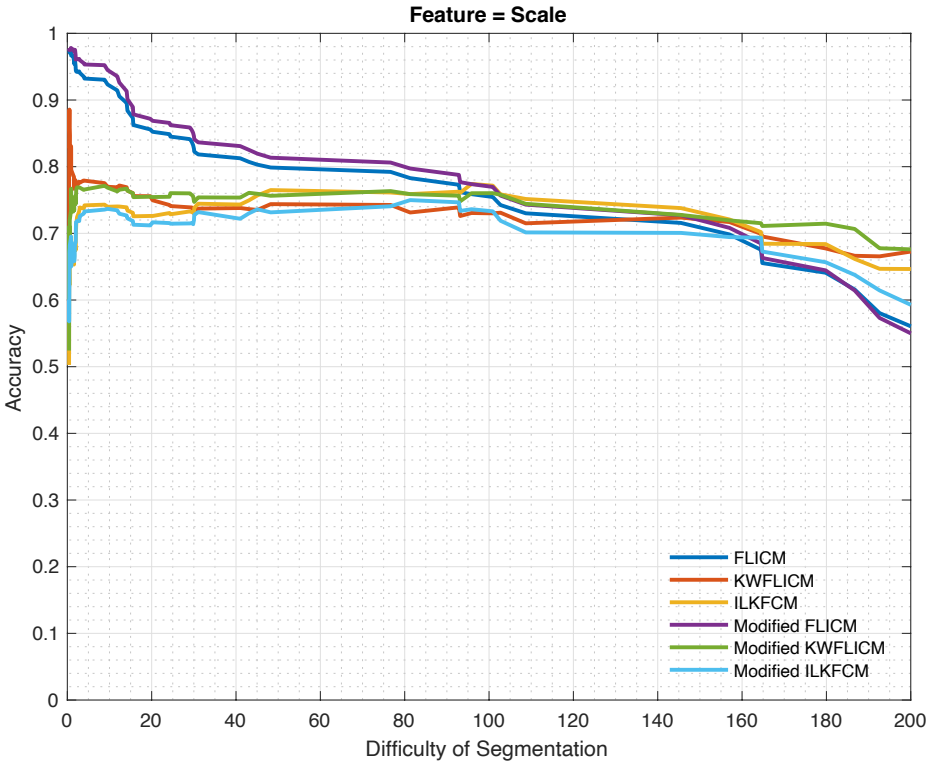


Figure 5.7: DoS vs Segmentation Accuracy for Feature 'Scale' (2)

5.2.4 Roughness and Scale based Feature Map

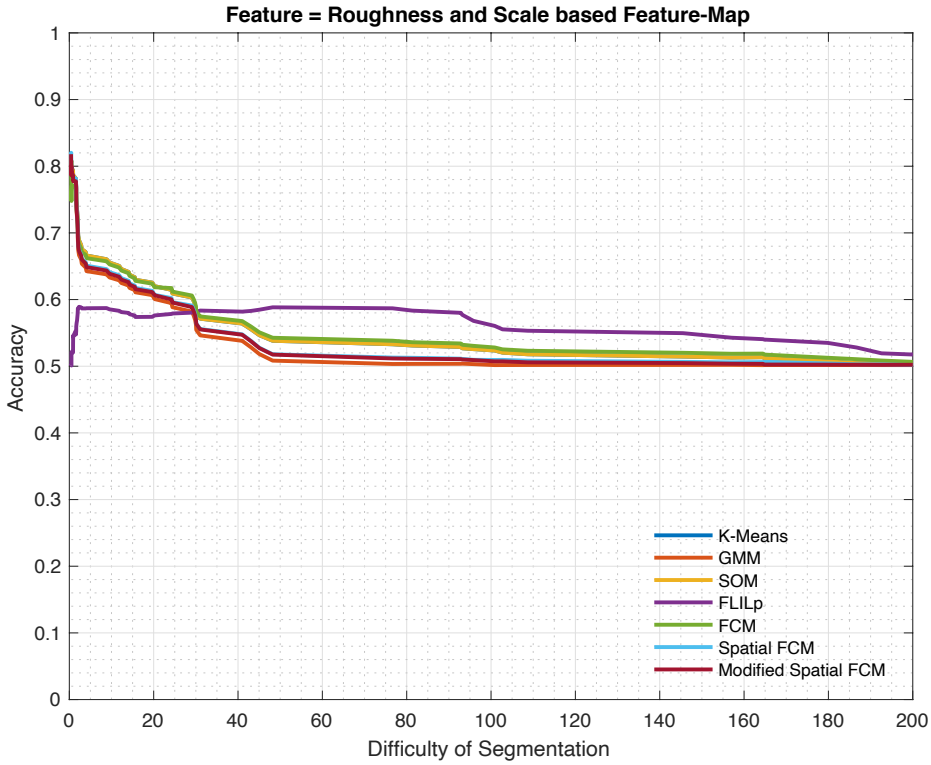


Figure 5.8: DoS vs Segmentation Accuracy for Feature 'Roughness and Scale based Feature-Map' (1)

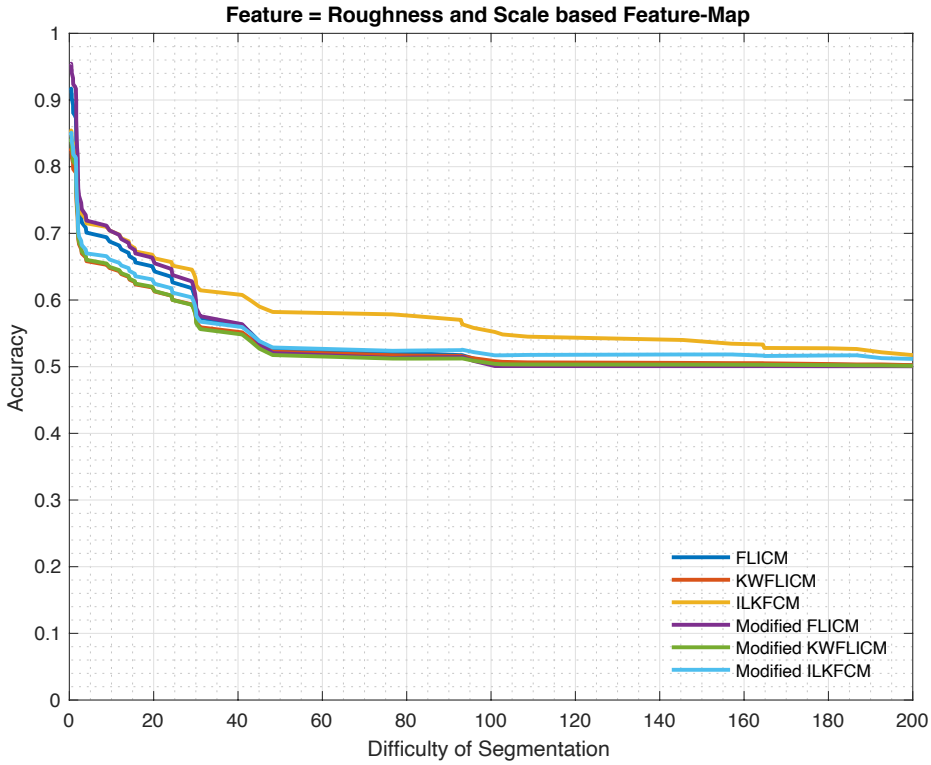


Figure 5.9: DoS vs Segmentation Accuracy for Feature 'Roughness and Scale based Feature-Map' (2)

5.2.5 Roughness and Scale

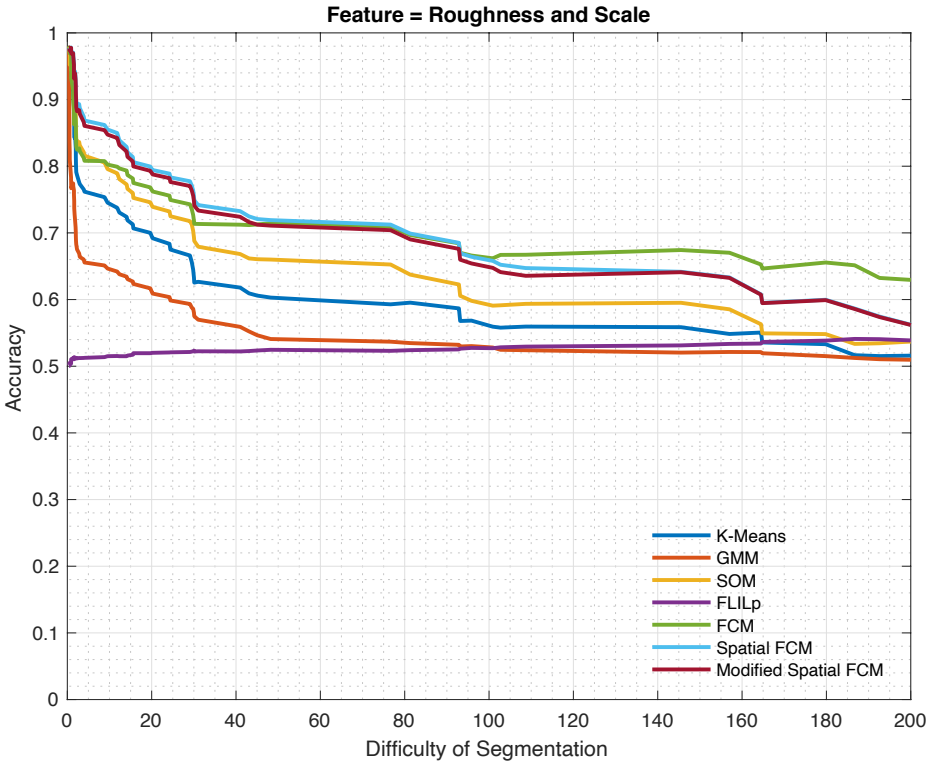


Figure 5.10: DoS vs Segmentation Accuracy for Feature 'Roughness and Scale' (1)

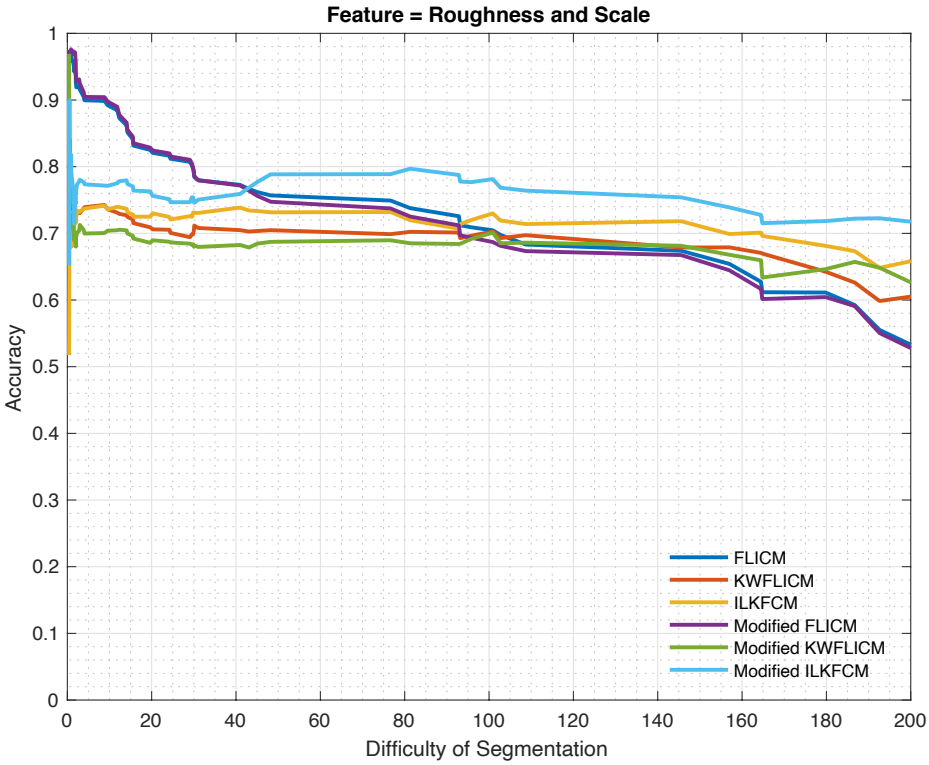


Figure 5.11: DoS vs Segmentation Accuracy for Feature 'Roughness and Scale' (2)

5.2.6 Shannon Entropy

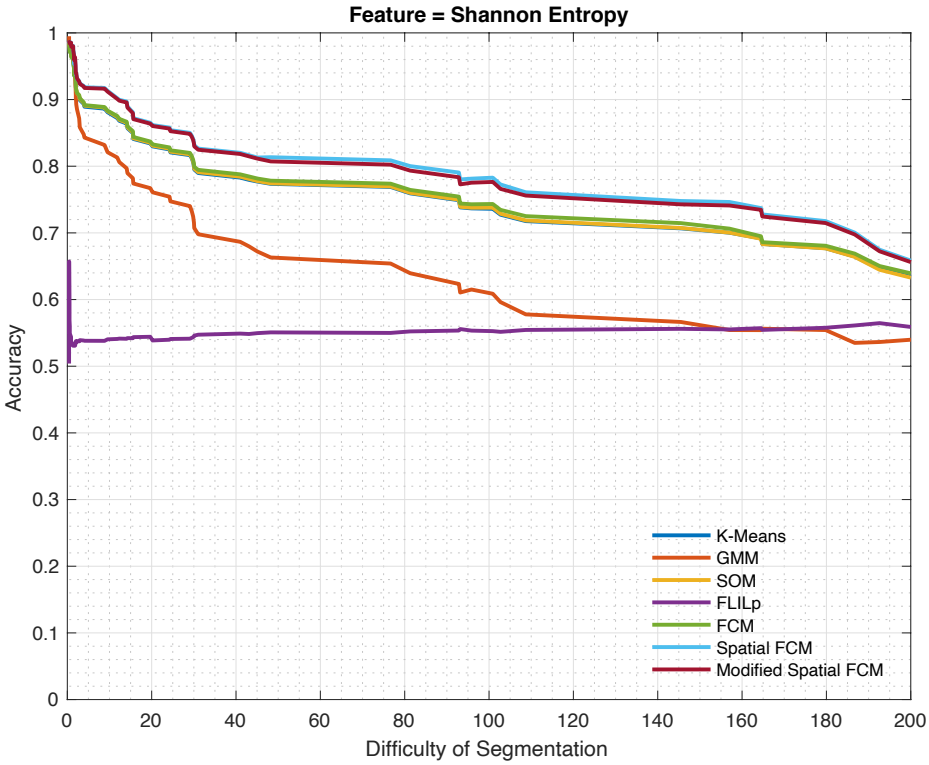


Figure 5.12: DoS vs Segmentation Accuracy for Feature 'Shannon Entropy' (1)

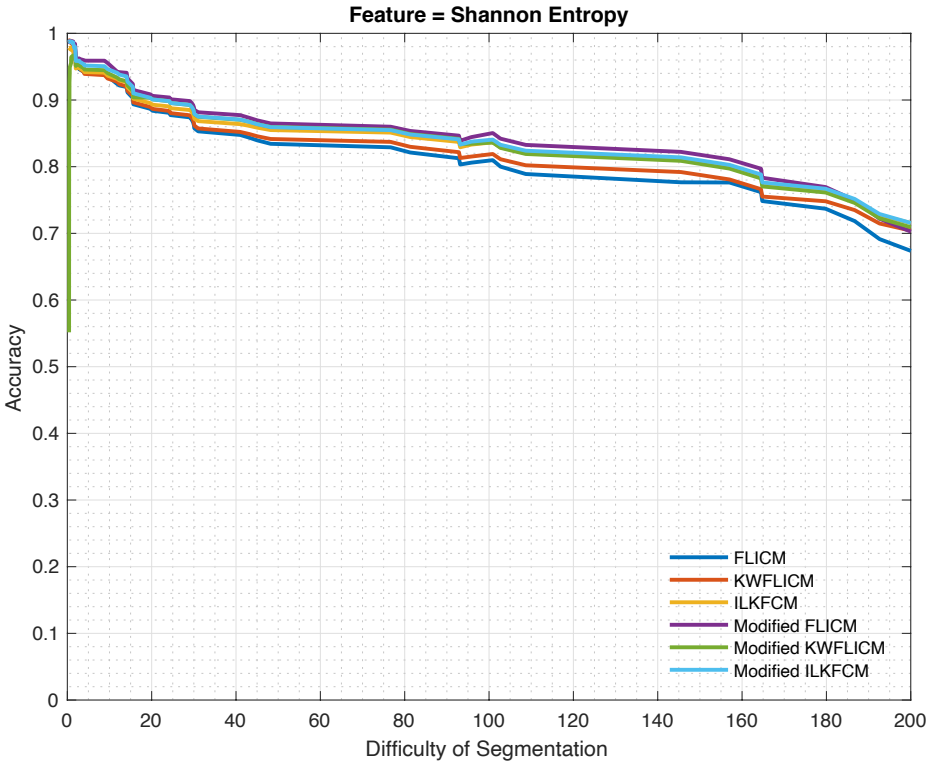


Figure 5.13: DoS vs Segmentation Accuracy for Feature 'Shannon Entropy' (2)

5.2.7 Renyi Entropy

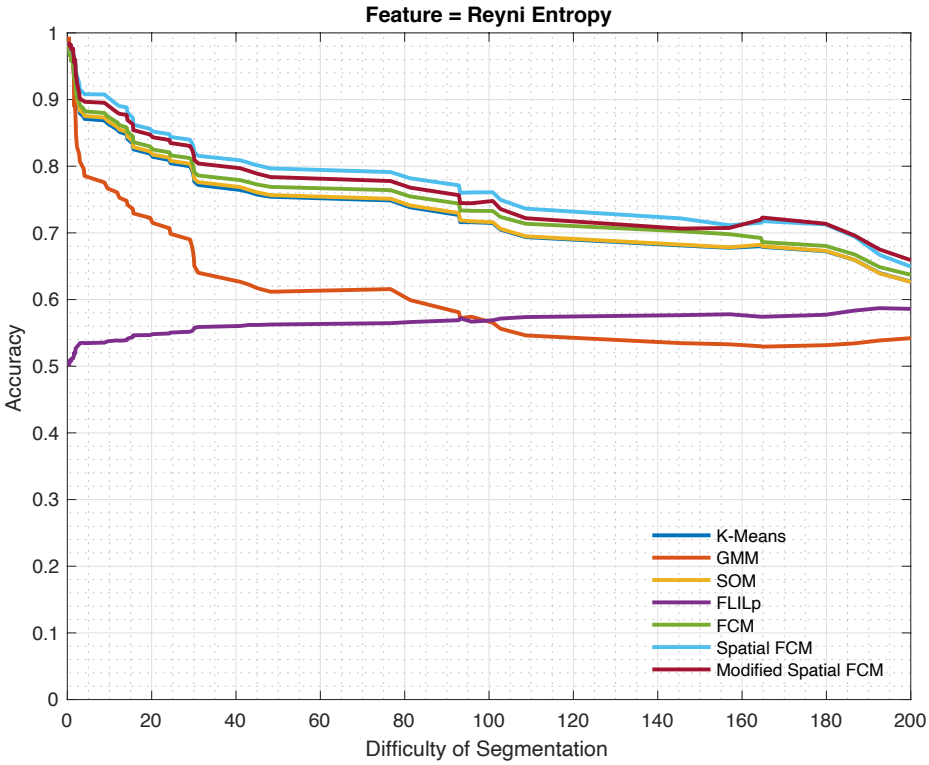


Figure 5.14: DoS vs Segmentation Accuracy for Feature 'Renyi Entropy' (1)

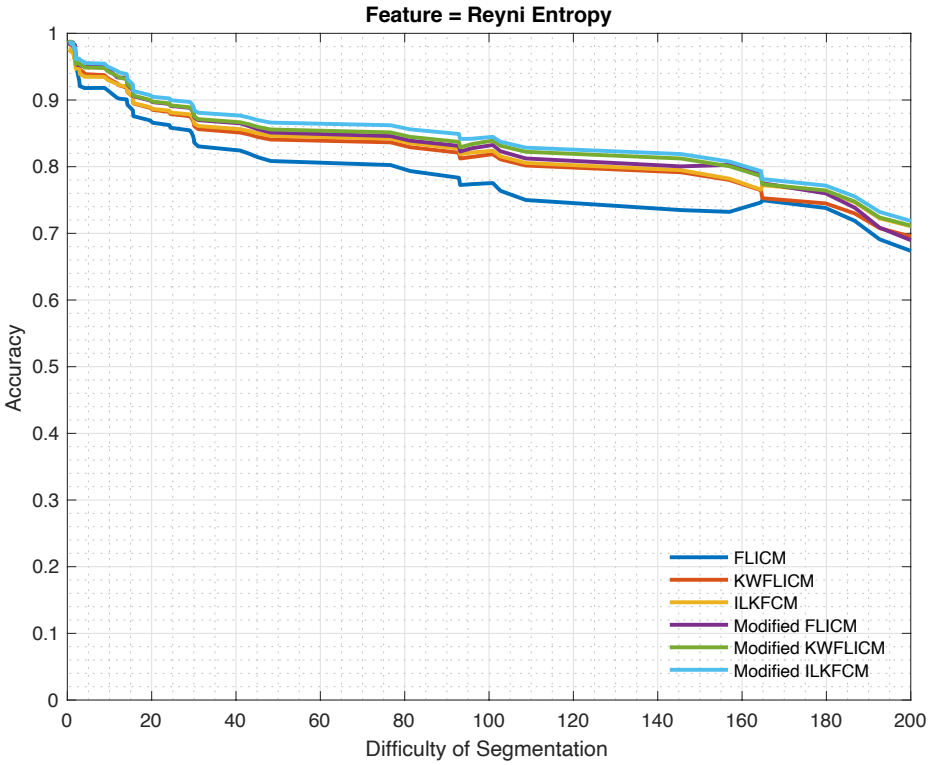


Figure 5.15: DoS vs Segmentation Accuracy for Feature 'Renyi Entropy' (2)

5.2.8 Tsallis Entropy

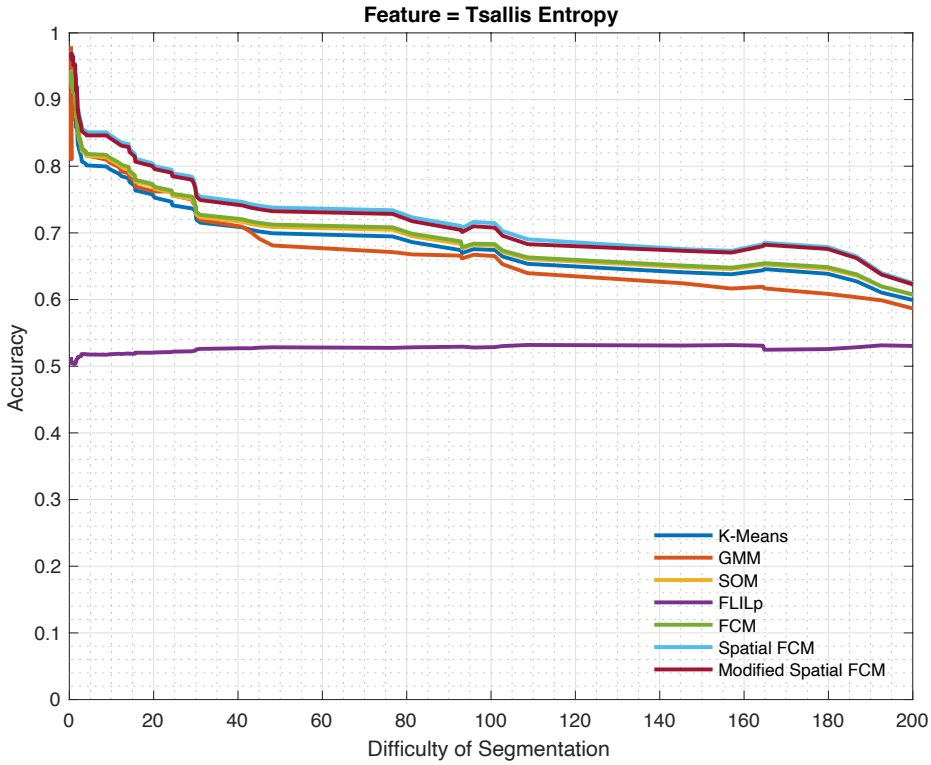


Figure 5.16: DoS vs Segmentation Accuracy for Feature 'Tsallis Entropy' (1)

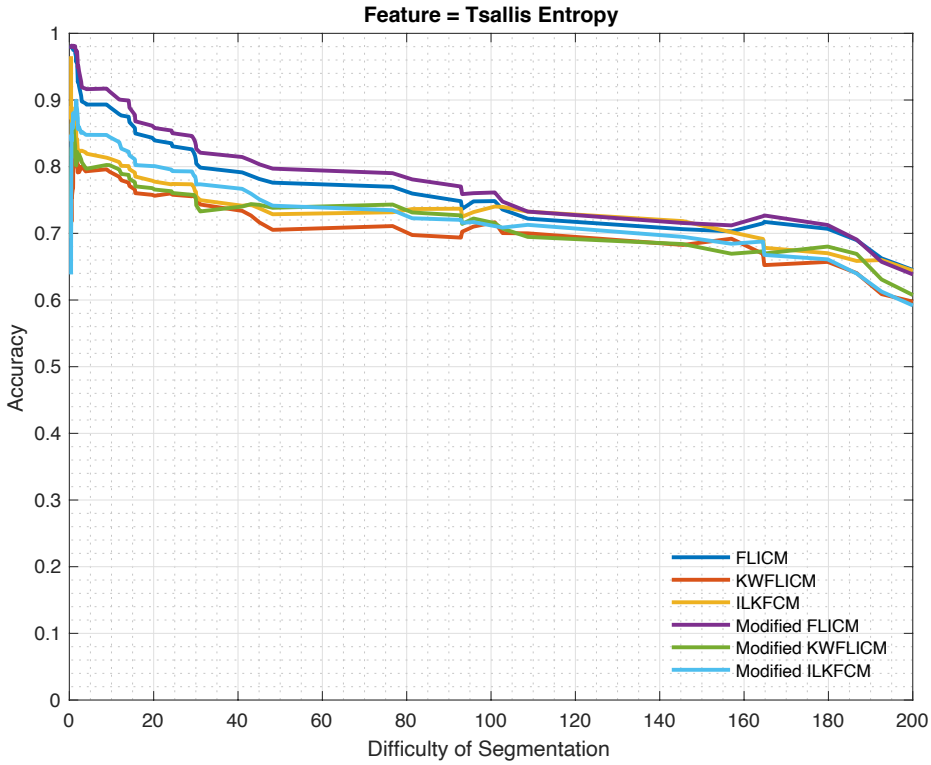


Figure 5.17: DoS vs Segmentation Accuracy for Feature 'Tsallis Entropy' (2)

5.2.9 Wavelet Energy

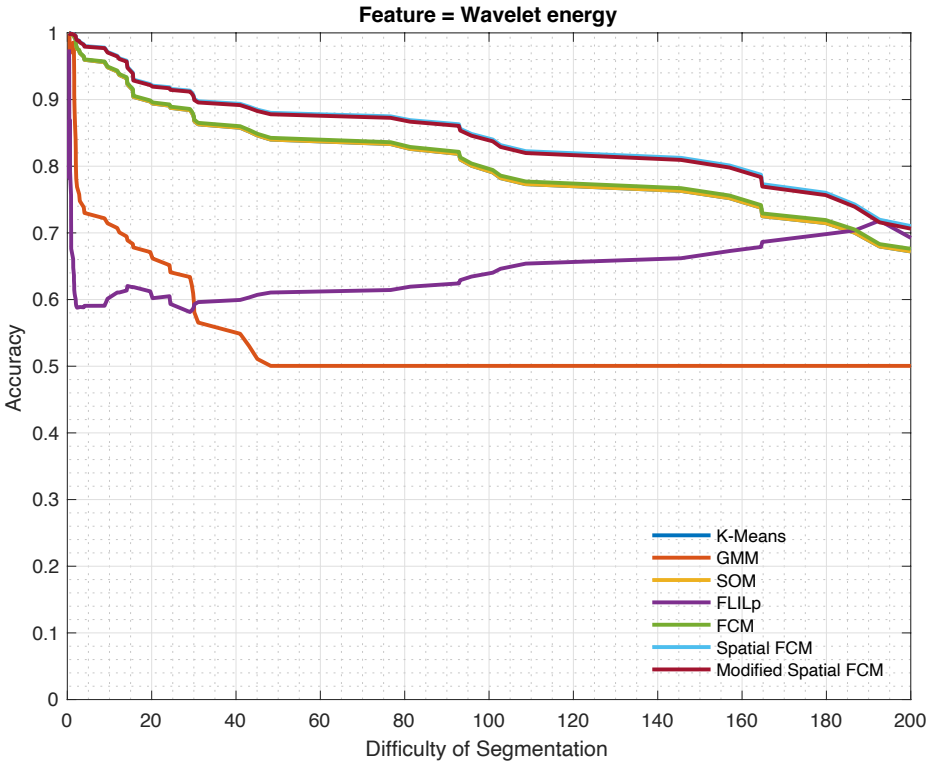


Figure 5.18: DoS vs Segmentation Accuracy for Feature 'Wavelet Energy' (1)

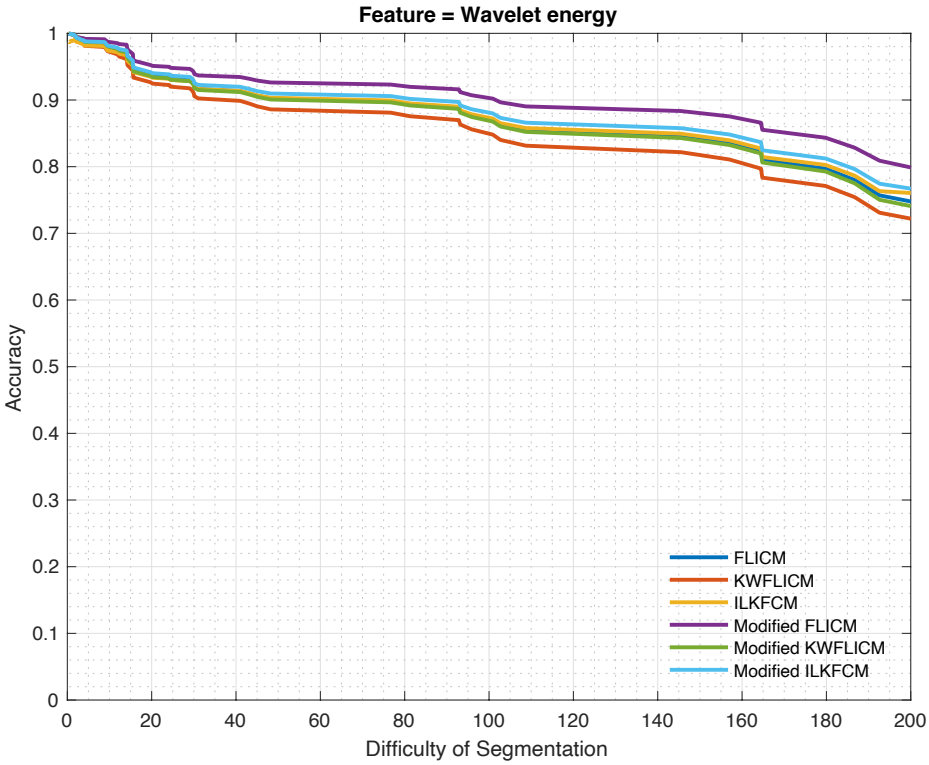


Figure 5.19: DoS vs Segmentation Accuracy for Feature 'Wavelet Energy' (2)

5.2.10 Local Binary Pattern

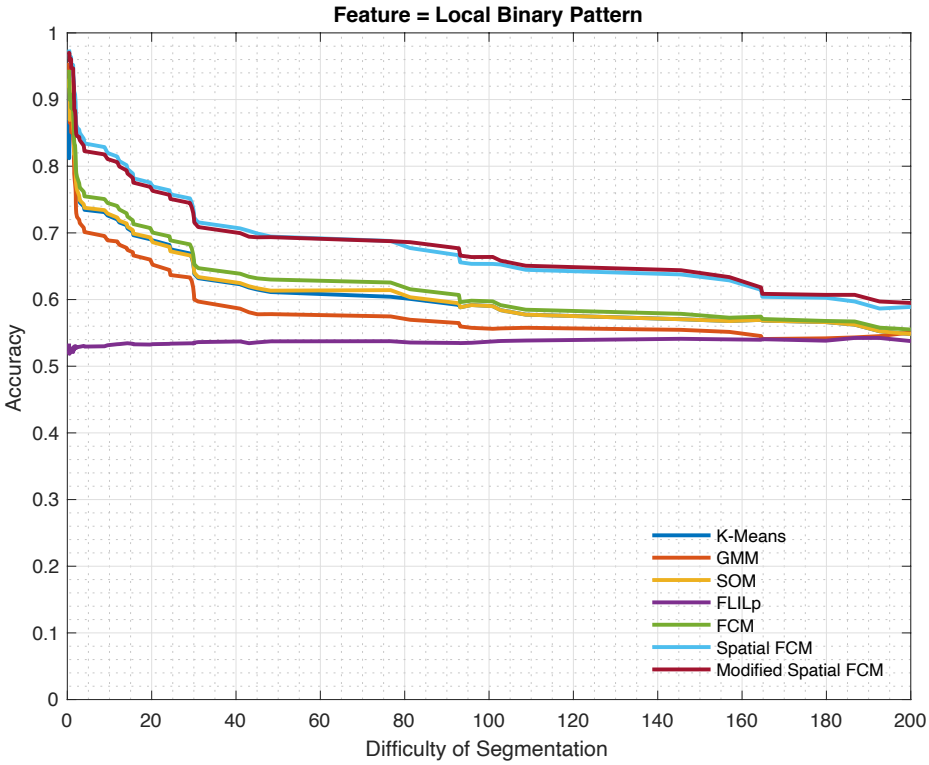


Figure 5.20: DoS vs Segmentation Accuracy for Feature 'Local Binary Pattern' (1)

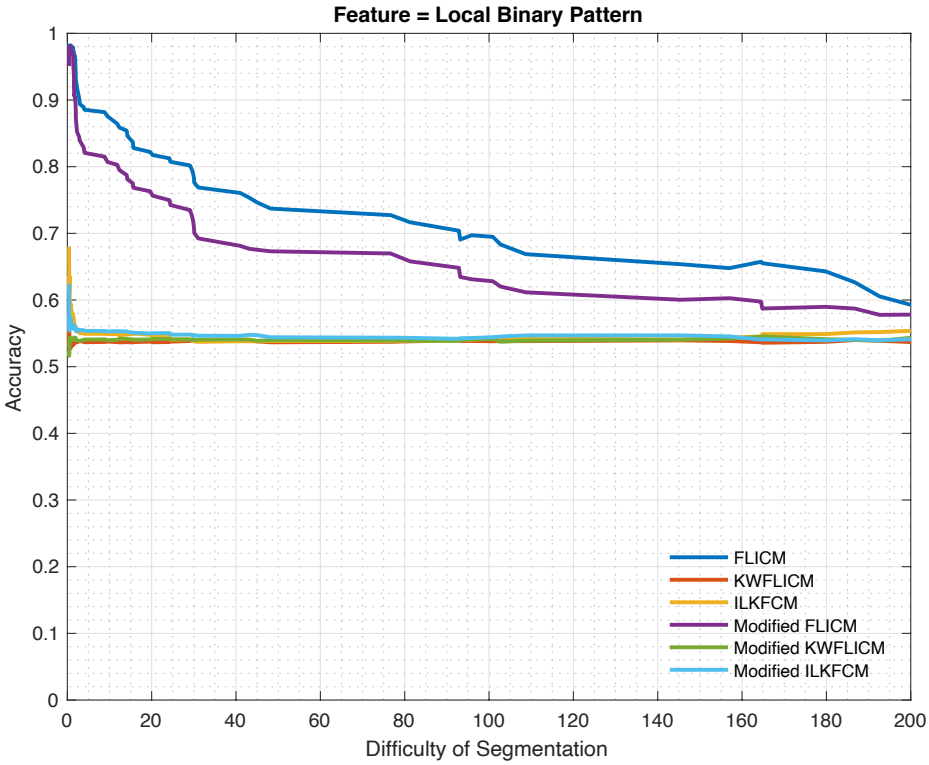


Figure 5.21: DoS vs Segmentation Accuracy for Feature 'Local Binary Pattern' (2)

5.2.11 GLCM

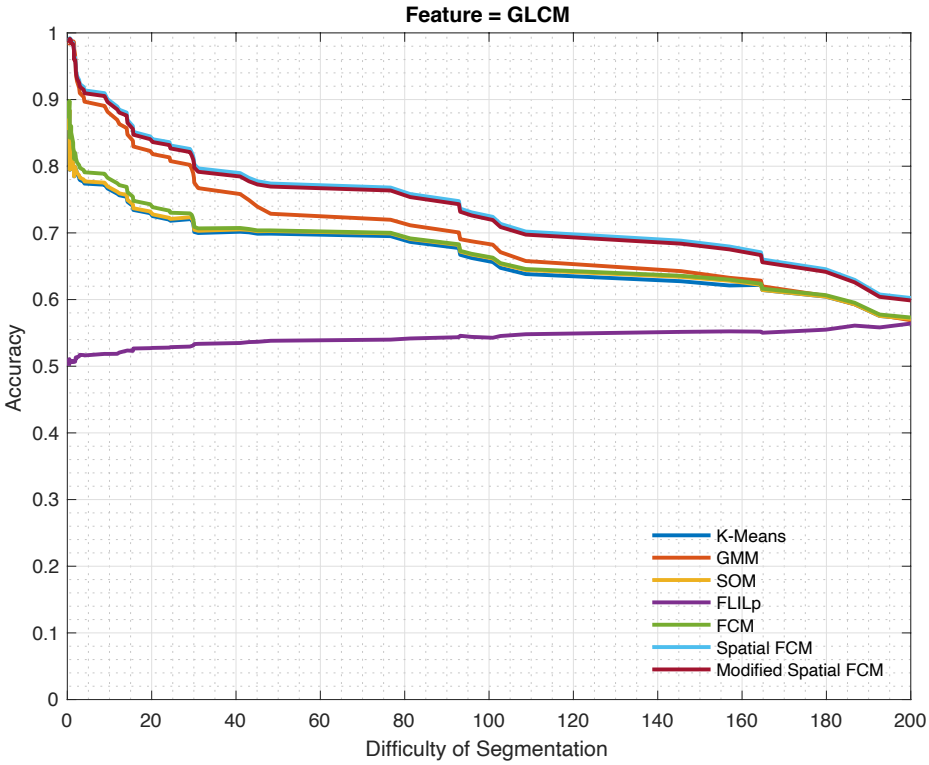


Figure 5.22: DoS vs Segmentation Accuracy for Feature 'GLCM' (1)

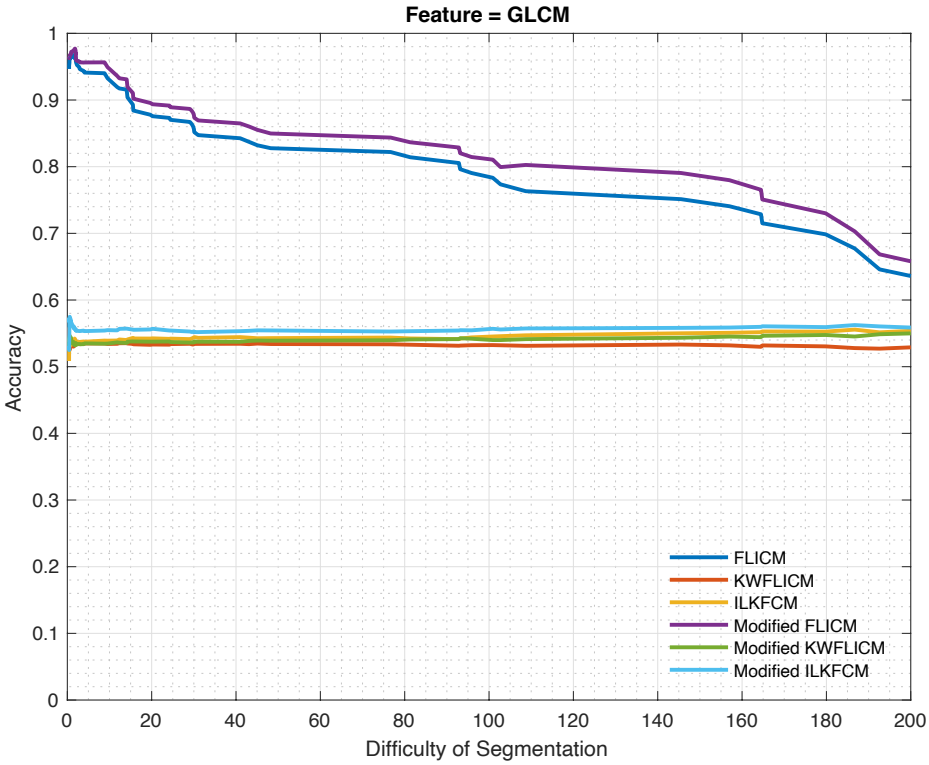


Figure 5.23: DoS vs Segmentation Accuracy for Feature 'GLCM' (2)

5.2.12 Central Statistical Moments

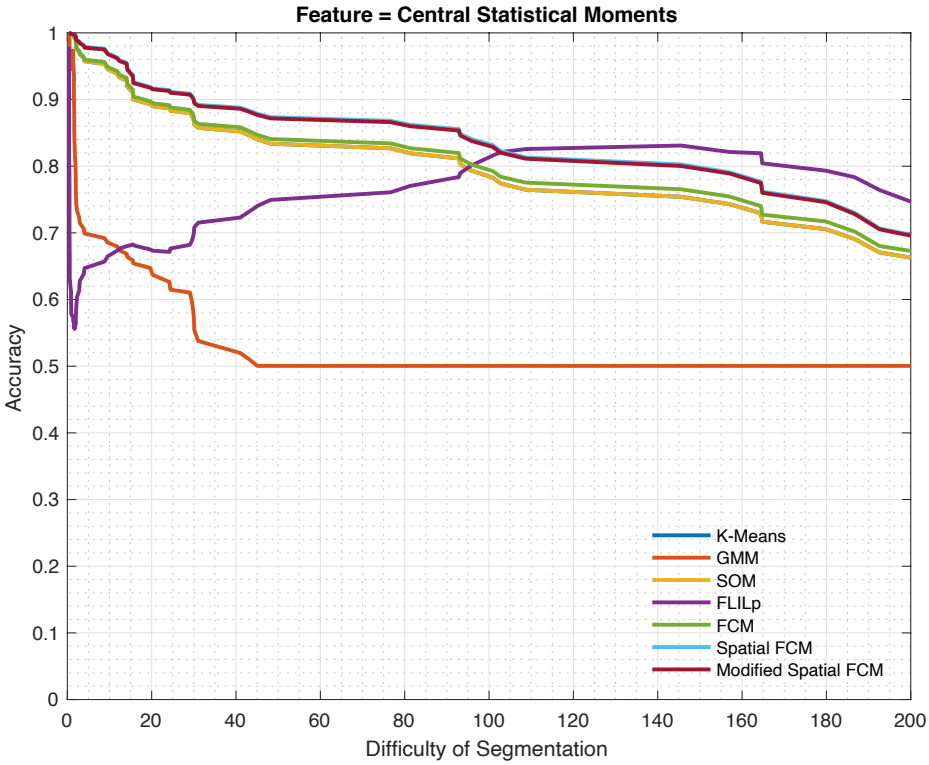


Figure 5.24: DoS vs Segmentation Accuracy for 'Central Statistical Moments' (1)

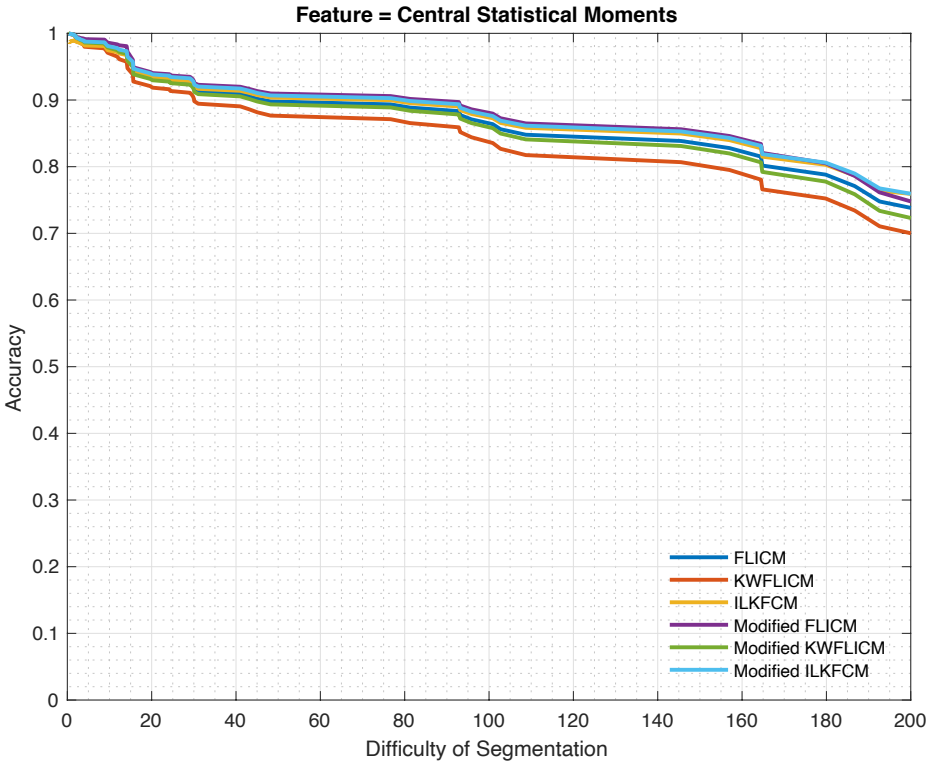


Figure 5.25: DoS vs Segmentation Accuracy for 'Central Statistical Moments' (2)

5.2.13 GLDM

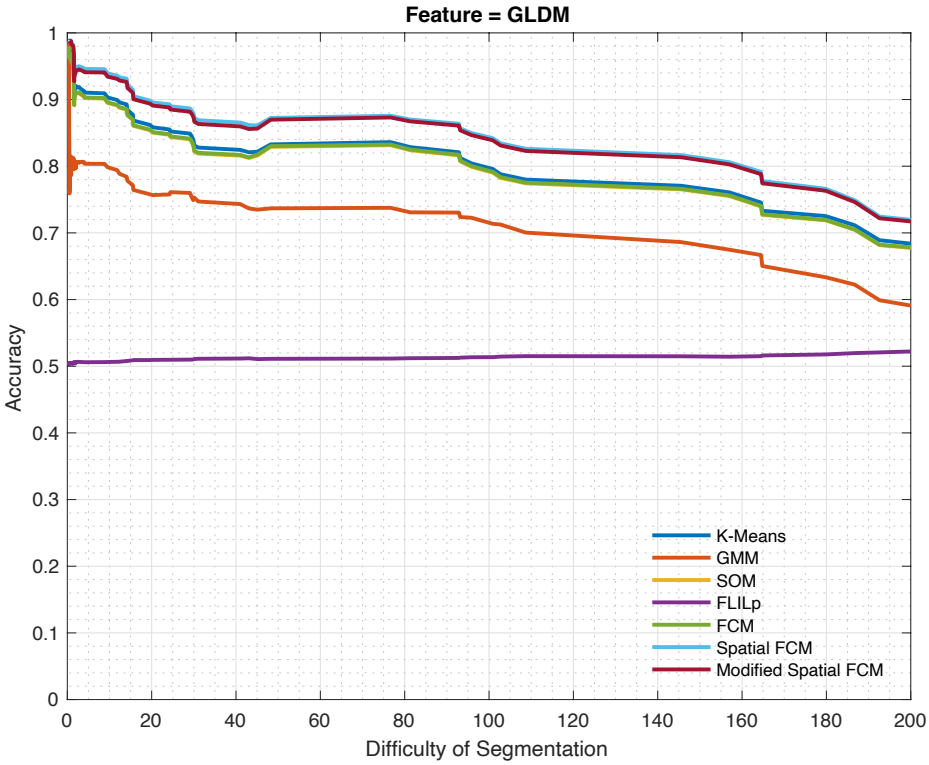


Figure 5.26: DoS vs Segmentation Accuracy for Feature 'GLDM' (1)

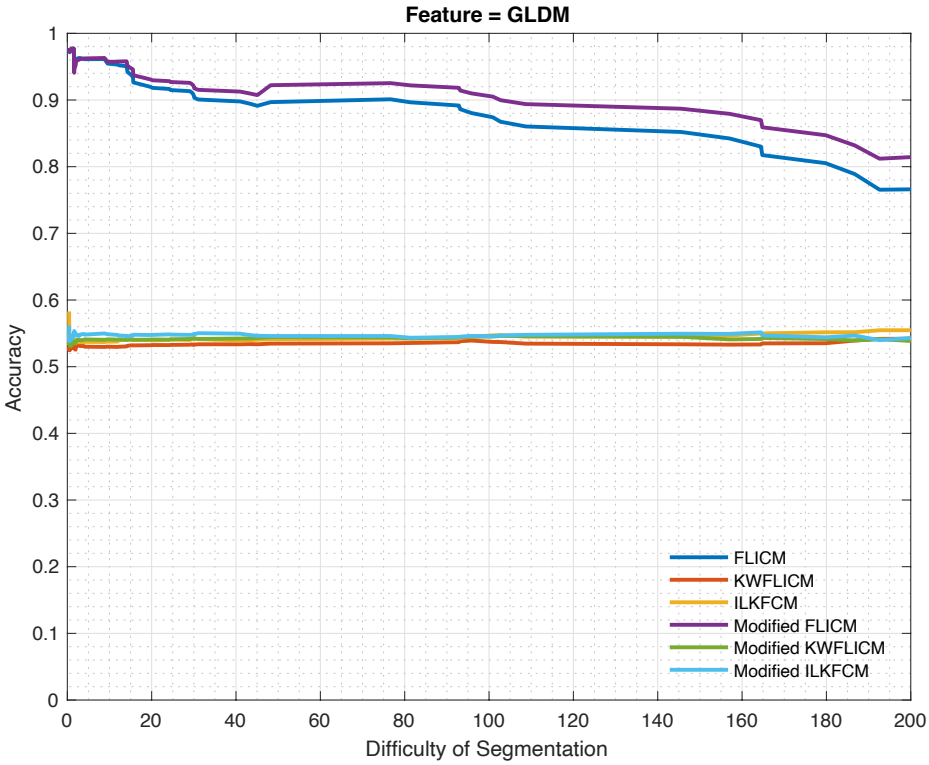


Figure 5.27: DoS vs Segmentation Accuracy for Feature 'GLDM' (2)

5.2.14 Local PDF

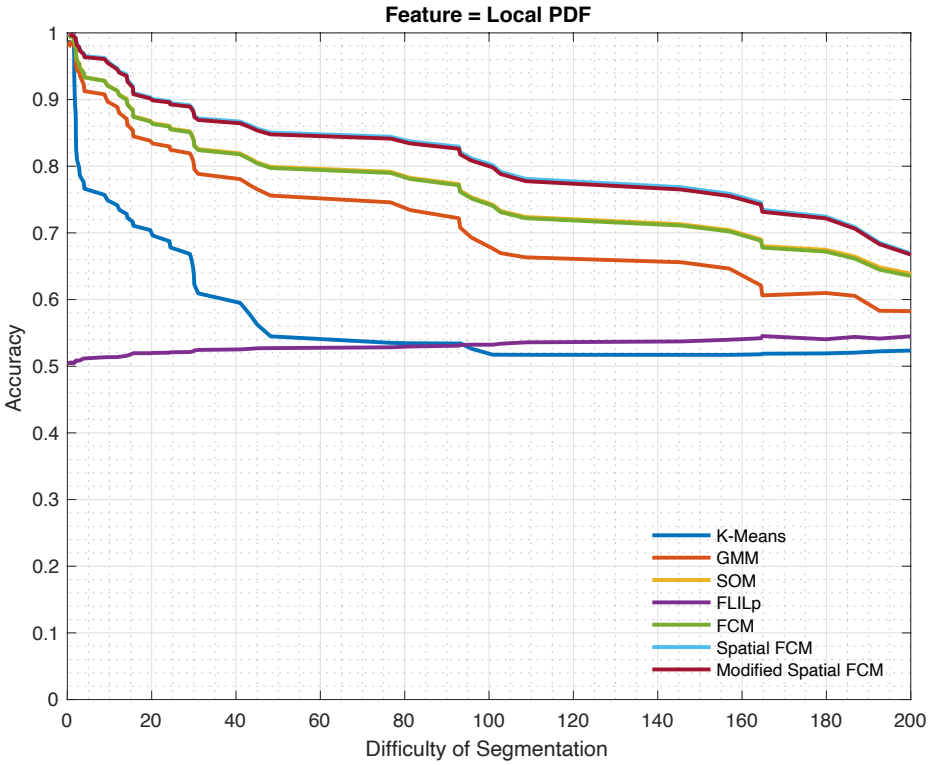


Figure 5.28: DoS vs Segmentation Accuracy for Feature 'Local PDF' (1)

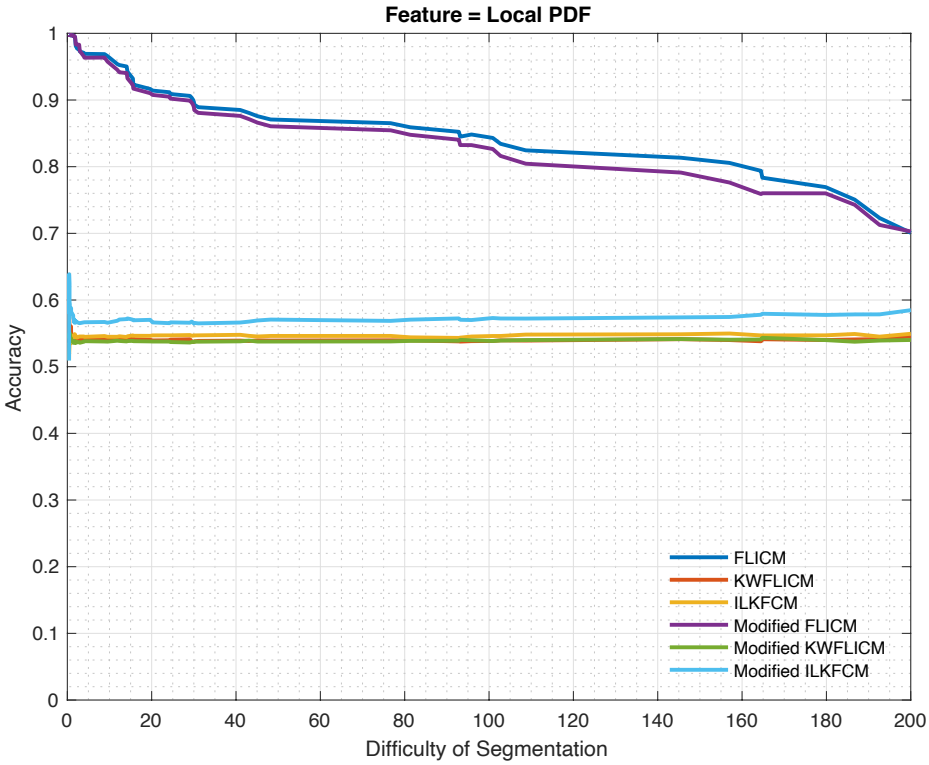


Figure 5.29: DoS vs Segmentation Accuracy for Feature 'Local PDF' (2)

5.2.15 GLRLM

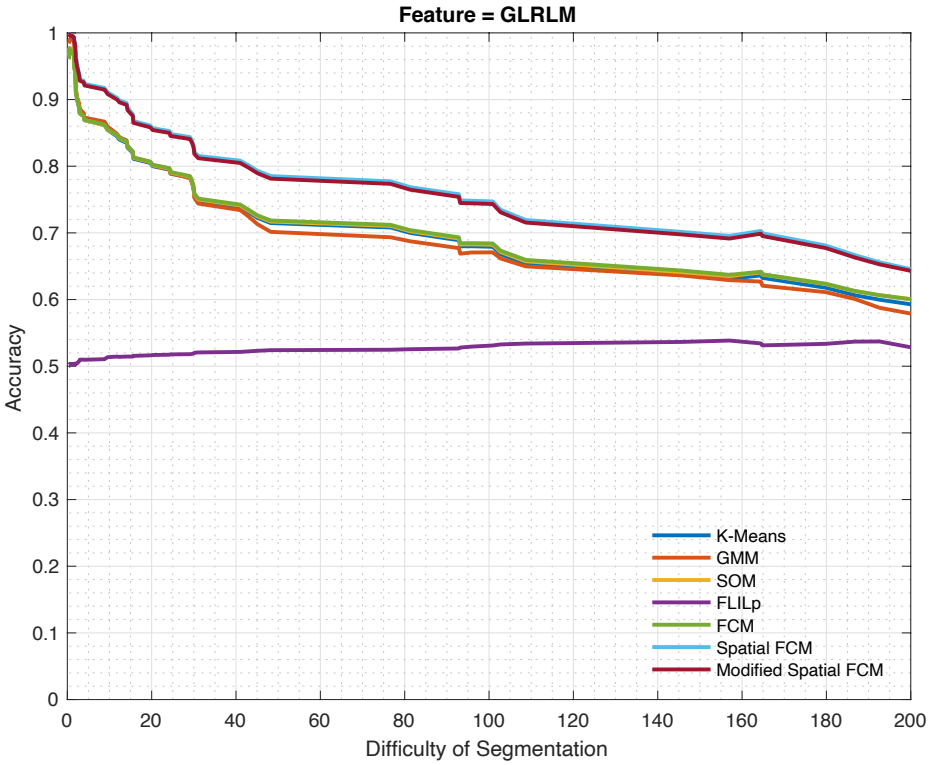


Figure 5.30: DoS vs Segmentation Accuracy for Feature 'GLRLM' (1)

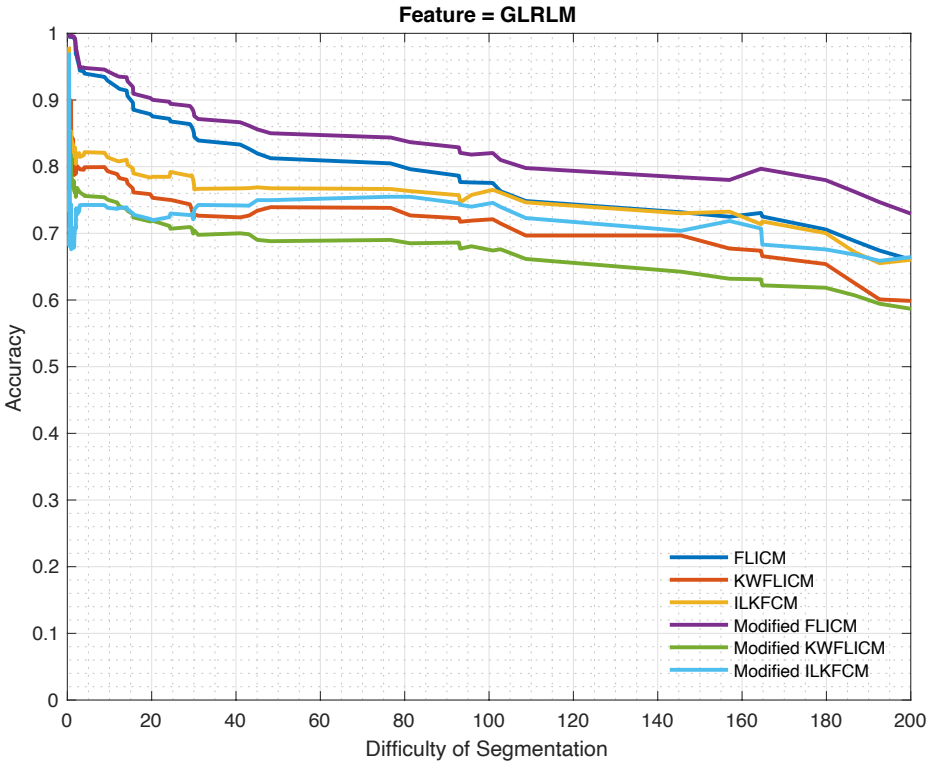


Figure 5.31: DoS vs Segmentation Accuracy for Feature 'GLRLM' (2)

5.2.16 Histogram of Gradients

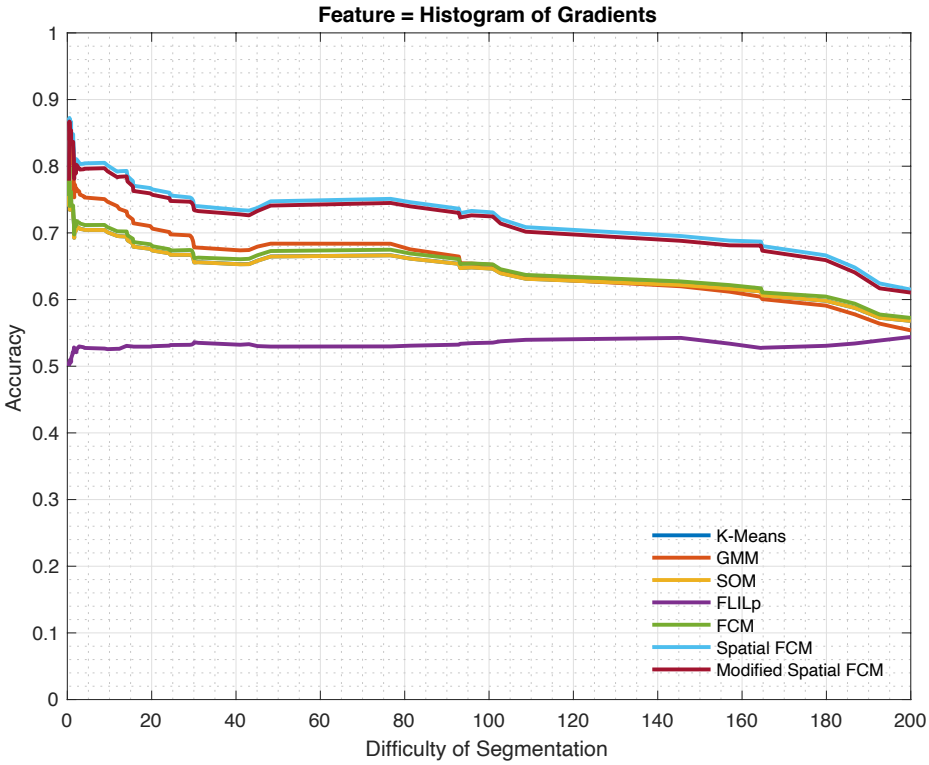


Figure 5.32: DoS vs Segmentation Accuracy for Feature 'Histogram of Gradients' (1)

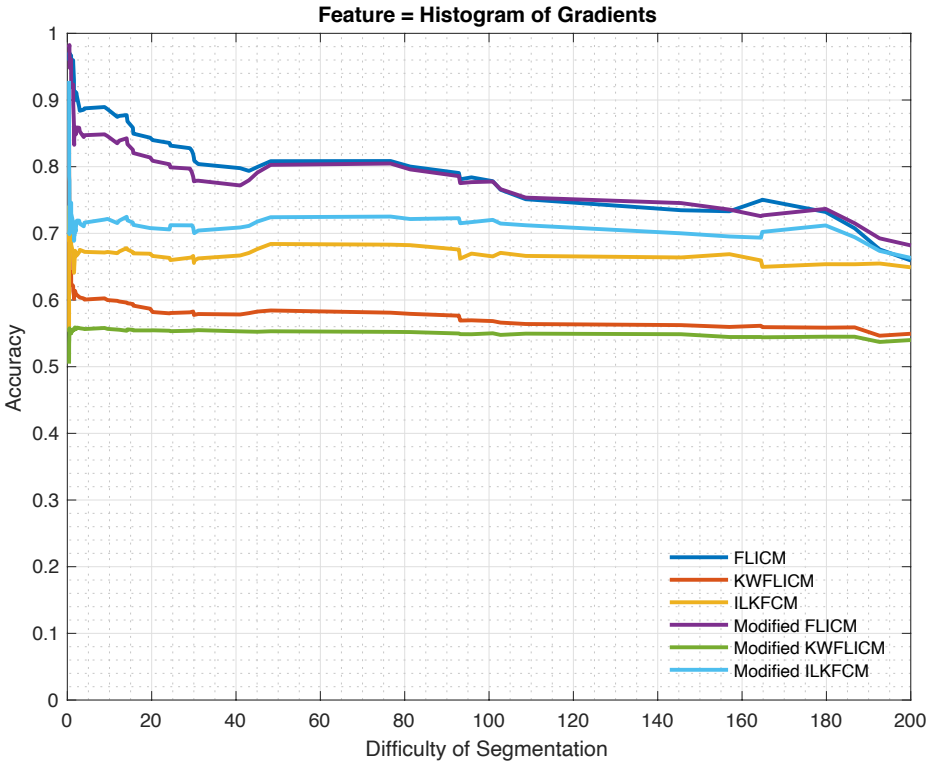


Figure 5.33: DoS vs Segmentation Accuracy for Feature 'Histogram of Gradients' (2)

5.2.17 Variogram

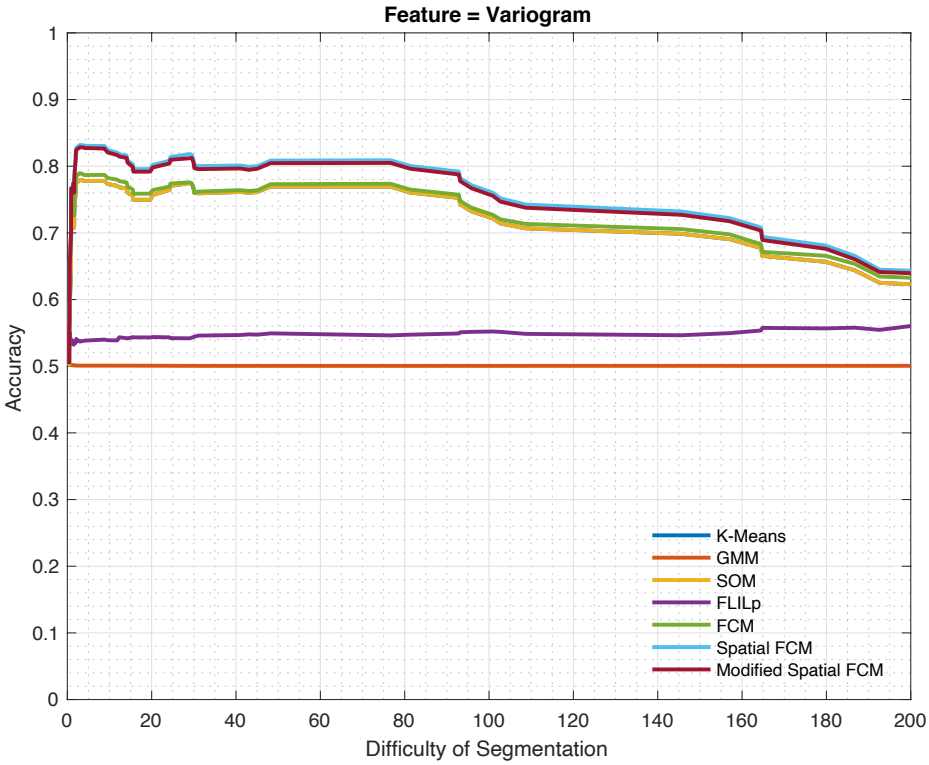


Figure 5.34: DoS vs Segmentation Accuracy for Feature 'Variogram' (1)

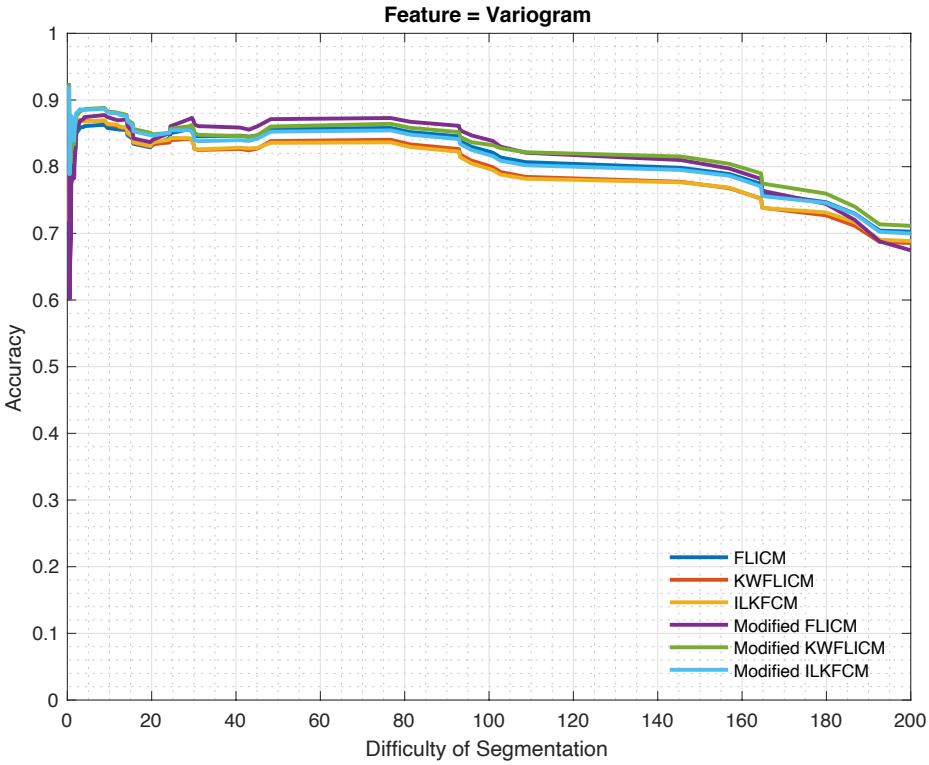


Figure 5.35: DoS vs Segmentation Accuracy for Feature 'Variogram' (2)

5.3 Comparison of Results for Synthetic Images

Based on the results obtained for synthetic images (Figure 5.2 to Figure 5.35), Figure 5.36 shows the feature-technique combinations which have performed better than the others. These combinations are:

a. Wavelet Energy – FLICM (WE-FLICM)
b. Wavelet Energy – ILKFCM (WE-ILKFCM)
c. Wavelet Energy – Modified FLICM (WE-mFLICM)
d. Wavelet Energy – Modified ILKFCM (WE-mILKFCM)
e. Central Statistical Moments – FLICM (CM-FLICM)
f. Central Statistical Moments – ILKFCM (CM-ILKFCM)
g. Central Statistical Moments – Modified FLICM (CM-mFLICM)
h. Central Statistical Moments – Modified ILKFCM (CM- mILKFCM)
i. GLDM – FLICM (GL-FLICM)
j. GLDM – Modified FLICM (GL-mFLICM)

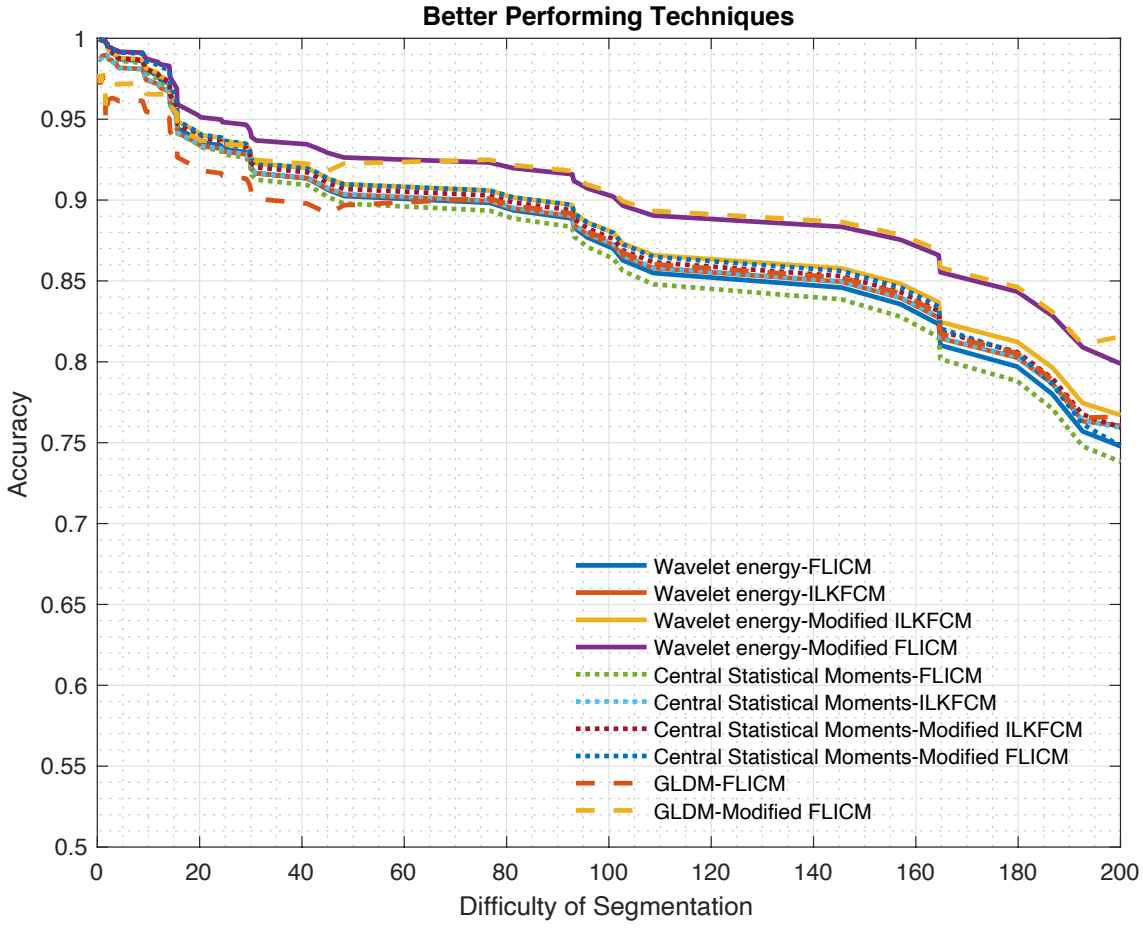


Figure 5.36: Comparison of Results for Synthetic Images

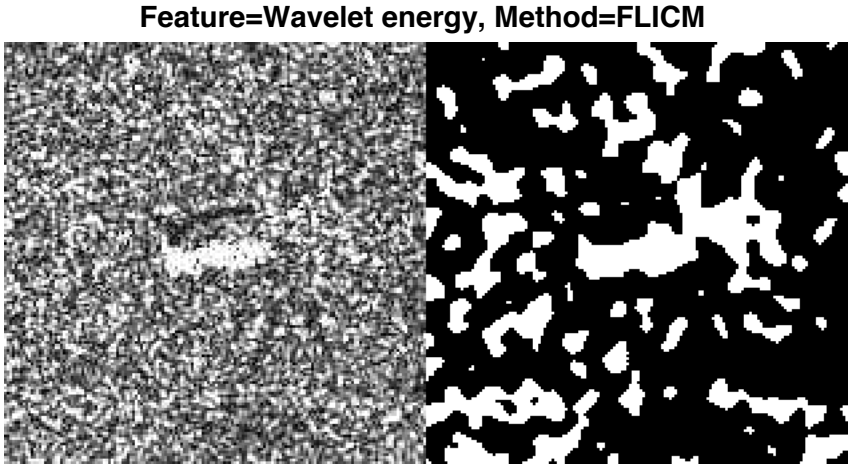
Here it is observed that, Modified FLICM with Wavelet Energy feature of image gives the best overall performance for synthetic SAR images.

Next these techniques are used to segment real SAR images.

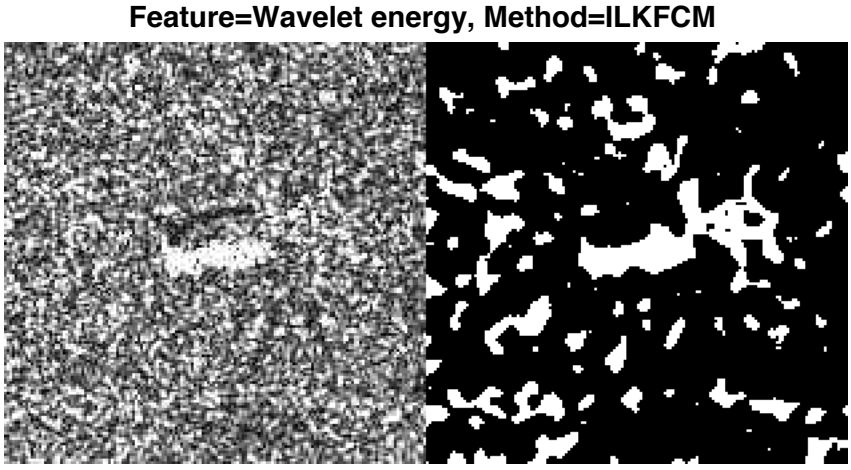
5.4 Results with Real SAR Images

Those feature-technique combinations which performed better on synthetic SAR images were also analysed with real SAR data. For testing on real SAR data, intensity images from the MSTAR public target dataset [37] are used. These images have military vehicles in a vegetation background. Out of the various images available in the dataset, three images, one image each of 2S1 Gvozdika self-propelled howitzer, BRDM-2 amphibious armoured patrol car and T-72 tank, are used as the test images.

Figure 5.37 shows the segmented images using the ten feature-techniques for 2S1.

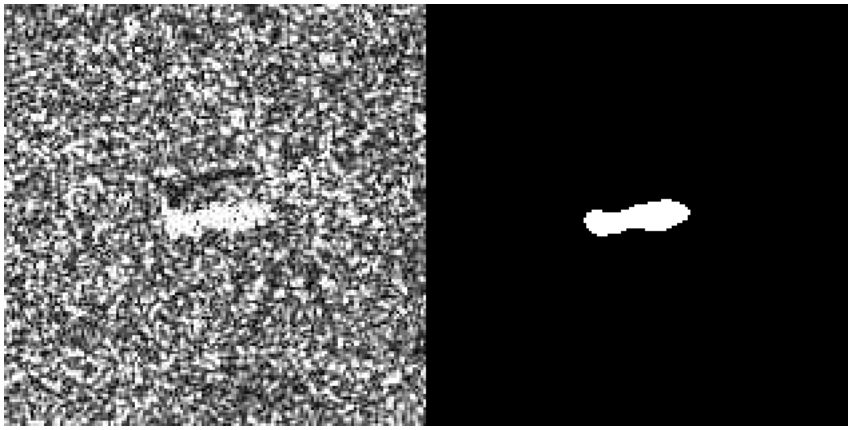


(a)



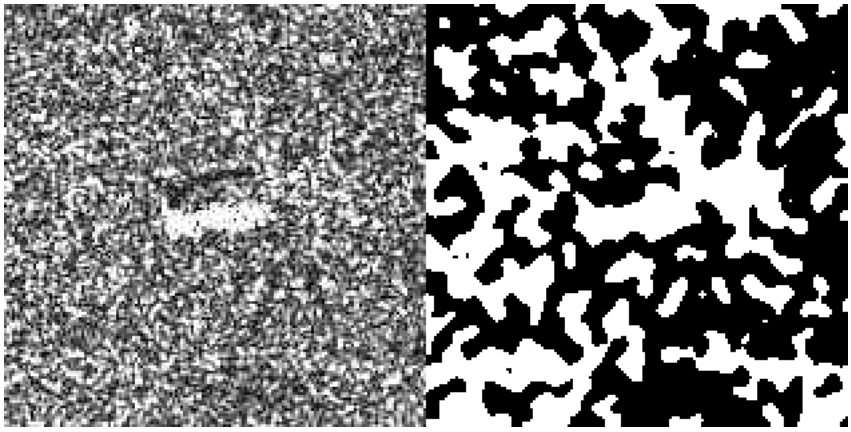
(b)

Feature=Wavelet energy, Method=Modified FLICM



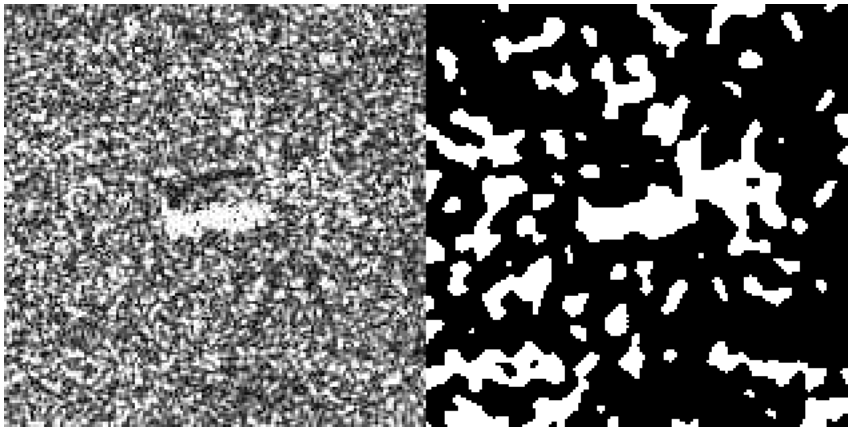
(c)

Feature=Wavelet energy, Method=Modified ILKFCM



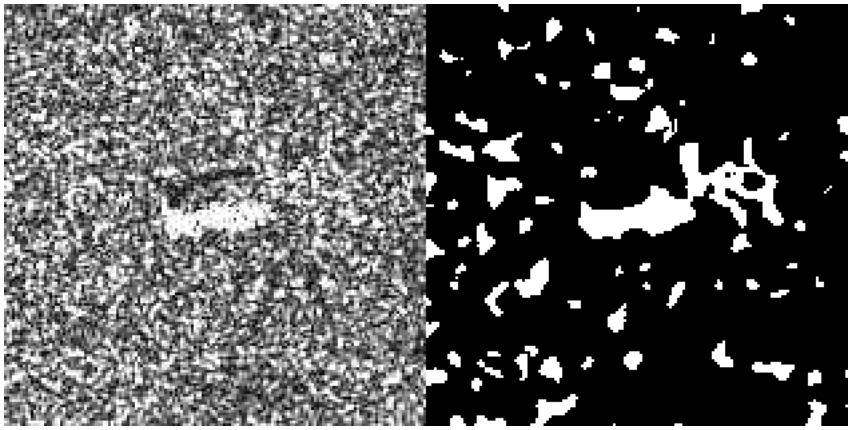
(d)

Feature=Central Statistical Moments, Method=FLICM



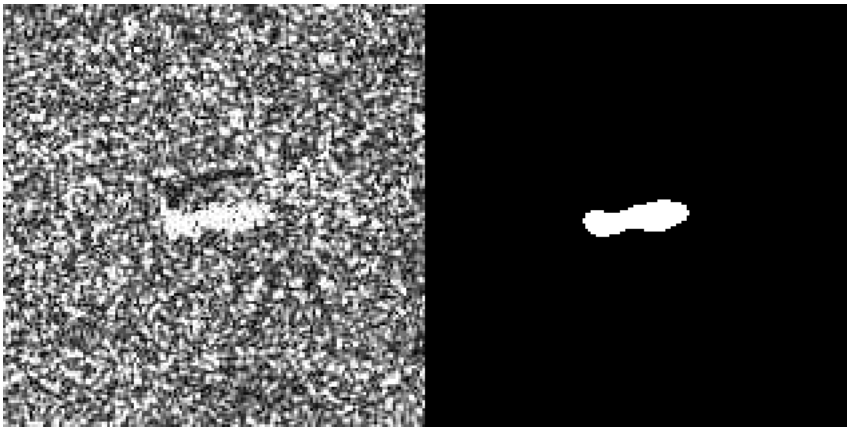
(e)

Feature=Central Statistical Moments, Method=ILKFCM



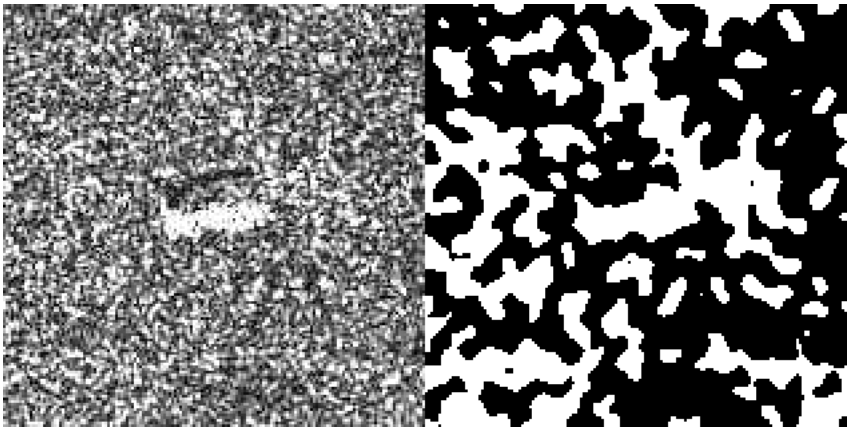
(f)

Feature=Central Statistical Moments, Method=Modified FLICM

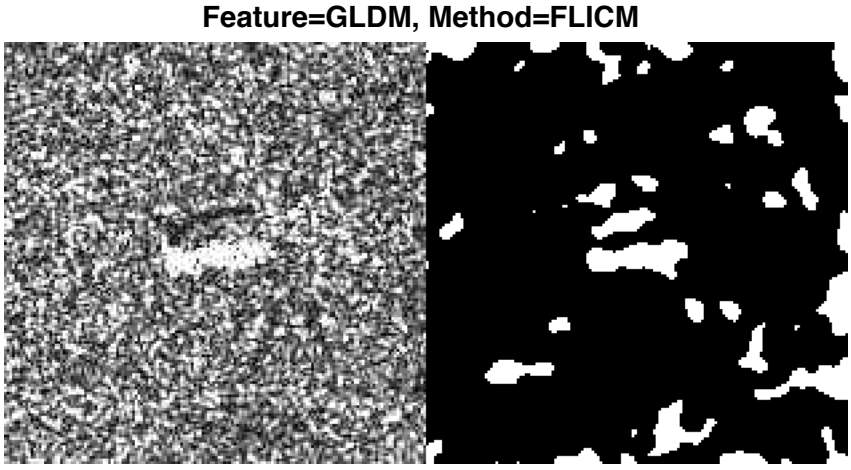


(g)

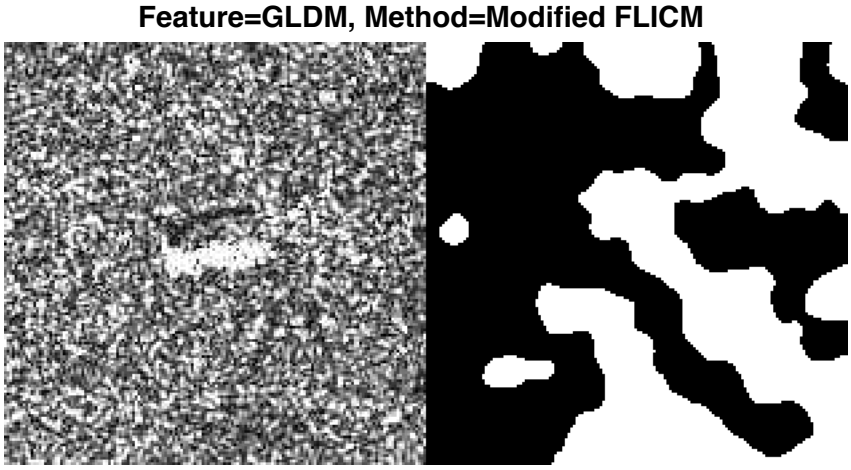
Feature=Central Statistical Moments, Method=Modified ILKFCM



(h)



(i)



(j)

Figure 5.37: Segmented Images for SAR Image of 2S1 [(a)-{j)]

Error! Reference source not found. Table 5.2, Table 5.3 and Table 5.4 list the performance of the ten feature-techniques for the three images:

Table 5.2: Segmentation Performance for SAR Image of 2S1 (DoS=2.141)

Technique	Accuracy	Normalised Accuracy	Kappa	CRF	F-Measure
WE-FLICM	0.755	0.969	0.071	0.645	0.093
WE-ILKFCM	0.841	0.942	0.113	0.686	0.134
WE-mFLICM	**0.997**	**0.986**	**0.878**	**0.974**	**0.88**

WE-mILKFCM	0.590	0.955	0.034	0.601	0.058
CM-FLICM	0.756	0.969	0.071	0.646	0.094
CM-ILKFCM	0.894	0.947	0.168	0.721	0.187
CM-mFLICM	**0.997**	**0.986**	**0.874**	**0.819**	**0.876**
CM-mILKFCM	0.633	0.958	0.041	0.611	0.064
GL-FLICM	0.896	0.896	0.160	0.499	0.179
GL-mFLICM	0.584	0.563	0.007	0.482	0.032

Table 5.3: Segmentation Performance for SAR Image of BRDM2 (DoS=2.345)

Technique	Accuracy	Normalised Accuracy	Kappa	CRF	F-Measure
WE-FLICM	0.996	0.984	0.881	0.794	0.883
WE-ILKFCM	0.885	0.924	0.200	0.605	0.224
WE-mFLICM	**0.996**	**0.985**	**0.892**	**0.941**	**0.894**
WE-mILKFCM	0.665	0.931	0.062	0.629	0.094
CM-FLICM	0.996	0.984	0.881	0.794	0.883
CM-ILKFCM	0.922	0.938	0.277	0.624	0.299
CM-mFLICM	**0.996**	**0.985**	**0.893**	**0.720**	**0.895**
CM-mILKFCM	0.627	0.956	0.054	0.515	0.086
GL-FLICM	0.581	0.952	0.045	0.507	0.077
GL-mFLICM	0.542	0.061	0.035	0.498	0.014

Table 5.4: Segmentation Performance for SAR Image of T72 (DoS=2.228)

Technique	Accuracy	Normalised Accuracy	Kappa	CRF	F-Measure
WE-FLICM	0.996	0.988	0.920	0.665	0.922
WE-ILKFCM	0.989	0.988	0.797	0.861	0.802
WE-mFLICM	**0.997**	**0.990**	**0.934**	**0.966**	**0.935**
WE-mILKFCM	0.648	0.965	0.074	0.538	0.113
CM-FLICM	0.996	0.988	0.919	0.962	0.921
CM-ILKFCM	0.993	0.981	0.809	0.888	0.813
CM-mFLICM	**0.997**	**0.990**	**0.932**	**0.966**	**0.933**
CM-mILKFCM	0.813	0.977	0.160	0.525	0.194
GL-FLICM	0.773	0.974	0.130	0.513	0.165
GL-mFLICM	0.781	0.975	0.135	0.512	0.171

Thorough experimentation and analysis of segmentation performance with various features and techniques suggests that Wavelet energy and Central statistical moments are the best features to perform segmentation of SAR images. Technique-wise FLICM, mFLICM, ILKFCM, mILKFCM perform best on synthetic SAR images with Wavelet energy based mFLICM outperforming all other feature-techniques. For real SAR data same consistency is observed in terms of results .i.e. Wavelet energy based mFLICM, hereafter called Weighted Membership Fuzzy Local Information C-Means (WMFLICM) gave best segmentation results in terms of considered performance metrics. The performance of Central moment based mFLICM is also found to be comparable to wavelet energy based mFLICM, suggesting it to be the second best technique for segmentation of SAR images.

Chapter 6
Conclusion and Future Work

A research on image segmentation techniques has been carried out under this project. Current work focused on image segmentation techniques for SAR images specifically. Seventeen different textural features were extracted for SAR images and fed to thirteen different segmentation techniques. Out of these 13 techniques, 9 techniques already exists in literature and have found wide application in variety of domains. To enhance spatial context information used in existing techniques, 4 modifications are suggested in the these techniques. The suggested modifications are to improve utilisation of spatial context information and to provide two level robustness for noisy data.

After performing extensive experiments on SAR images of different background and foreground characteristics, it is observed that segmentation techniques perform best with Central statistical moments and Wavelet energy as features. Out of all the techniques considered FLICM, mFLICM, ILKFCM, mILKFCM perform best in overall sense for synthetic SAR data. From the experimentation on synthetic SAR images, 10 best performing feature-technique combinations were identified and were further used for segmentation of real SAR images. For real SAR data, it is observed that WMFLICM performs best in overall sense. Central statistical moments based mFLICM also segmented real SAR images accurately with performance metrics marginally lower than WMFLICM.

In current work, performance evaluation was carried out on images with single foreground region. In future, extensive experiments can be carried out on images with multiple foreground regions.

As mFLICM emerged as best performing technique for SAR images, improvement in its performance can further be analysed with more image features. Computational efficiency improvement aspect can also be explored for the better performing techniques.

References

[1] A. C. Frery, H. J. Muller, C. Yanasse and S. Sant'Anna, “A Model for Extremely Heterogeneous Clutter,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 35, no. 3, pp. 648-659, May 1997.

[2] F. A. Á. Rodrigues, J. F. S. R. Neto, R. C. P. Marques, F. N. S. d. Medeiros and J. S. Nobre, “SAR Image Segmentation Using the Roughness Information,” *EEE Geoscience and Remote Sensing Letters,* vol. 13, no. 2, pp. 132-136, Feb 2016.

[3] M. E. Mejail, J. C. Jacobo-Berlles, A. C. Frery and O. H. Bustos, “Classification of SAR Images using a General and Tractable Multiplicative Model,” *International Journal of Remote Sensing,* vol. 24, no. 18, p. 3565–3582, 2003.

[4] R. C. P. Marques, F. N. Medeiros and . J. S. Nobre, “SAR Image Segmentation Based on Level Set Approach and {\cal G}_A^0 Model,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 34, no. 10, pp. 2046-2057, Oct 2012.

[5] R. H. Nobre, F. A. A. Rodrigues, R. C. P. Marques, J. S. Nobre, J. F. S. R. Neto and F. N. S. Medeiros, “SAR Image Segmentation With Rényi's Entropy,” *IEEE Signal Processing Letters,* vol. 23, no. 11, pp. 1551-1555, Nov 2016.

[6] D. Xiang, T. Tang, C. Hu, Y. Li and Y. Su, “A Kernel Clustering Algorithm With Fuzzy Factor: Application to SAR Image Segmentation,” *IEEE Geoscience and Remote Sensing Letters,* vol. 11, no. 7, pp. 1290-1294, 2014.

[7] K. S. Chuang, H. L. Tzeng, S. Chen, J. Wu and T. J. Chen, “Fuzzy c-means clustering with spatial information for image segmentation,” *Computerized Medical Imaging and Graphics,* vol. 30, no. 1, pp. 9-15, 2006.

[8] S. Krinidis and V. Chatzis, “A Robust Fuzzy Local Information C-Means Clustering Algorithm,” *IEEE Transactions on Image Processing,* vol. 19, no. 5, pp. 1328 -1337, 2010.

[9] M. Gong, Y. Liang, J. Shi, W. Ma and J. Ma, “Fuzzy C-Means Clustering With Local Information and Kernel Metric for Image Segmentation,” *EEE Transactions on Image Processing,* vol. 22, no. 2, pp. 573-584, 2013.

[10] F. Li and J. Qin, “Robust fuzzy local information and Lp-norm distance-based image segmentation method,” *IET Image Processing,* vol. 11, no. 4, pp. 217-226, 2017.

[11] D. W. Sun, “Classification-based segmentation,” in *Computer Vision Technology for Food Quality Evaluation*, London, Elsevier, 2008, p. 50.

[12] V. A. Krylov, G. Moser, S. B. Serpico and J. Zerubia, “On the Method of Logarithmic Cumulants for Parametric Probability Density Function Estimation,” *IEEE Transactions on Image Processing,* vol. 22, no. 10, pp. 3791-3806, Oct 2013.

[13] S. Cui and M. Datcu, “Coarse to Fine Patches-based Multitemporal Analysis of Very High Resolutions Satellite Images,” in *6th International Workshop on the Analysis of Multi-temporal Remote Sensing Images (Multi-Temp)*, Trento, Italy, 2011.

[14] S. Cui, G. Schwarz and M. Datcu, “A Comparative Study of Statistical Models for Multilook SAR Images,” *IEEE Geoscience and Remote Sensing Letters,* vol. 11, no. 10, pp. 1752-1756, Oct 2014.

[15] A. C. Frery, A. D. C. Nascimento and R. J. Cintra, “Contrast in speckled imagery with stochastic distances,” in *2010 IEEE International Conference on Image Processing*, Hong Kong, 2010.

[16] T. M. Cover and J. A. Thomas, “Entropy, Relative Entropy, and Mutual Information,” in *Elements of information theory*, New Jersey, John Wiley & Sons, Inc., 2006, pp. 13-16.

[17] T. Maszczyk and D. Wlodzislaw, “Comparison of Shannon, Renyi and Tsallis Entropy Used in Decision Trees,” in *International Conference on Artificial Intelligence and Soft Computing*, Berlin, 2008.

[18] A. Humeau-Heurtier, “Texture Feature Extraction Methods: A Survey,” *IEEE Access,* vol. 7, pp. 8975-9000, 2019.

[19] T. Mäenpää and M. Pietikäinen, “Texture analysis with local binary patterns,” in *Handbook of Pattern Recognition and Computer Vision, 3rd Edition*, Singapore, World Scientific Publishing, 2005, pp. 197-216.

[20] T. Ojala, M. Pietikainen and T. Maenpaa, “Multiresolution grey-scale and rotation invariant texture classification with local binary patterns,” *EEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 7, pp. 971-987, 2002.

[21] E. Prakasa, “Texture Feature Extraction by Using Local Binary Pattern,” *NKOM Journal of Informatics, Control Systems, and Computers,* vol. 9, no. 2, pp. 45-48, 2015.

[22] R. M. Haralick, K. Shanmugam and I. Dinstein, “Textural Features for Image Classification,” *IEEE Transactions on Systems, Man, and Cybernetics,* Vols. SMC-3, no. 6, pp. 610 - 621, 1973.

[23] M. Tuceryan, “Moment-based texture segmentation,” *Pattern Recognition Letters,* vol. 15, no. 7, pp. 659-668, 1994.

[24] J. K. Kim and H. W. Park, “Statistical textural features for detection of microcalcifications in digitized mammograms,” *IEEE Transactions on Medical Imaging,* vol. 18, no. 3, pp. 231-238, 1999.

[25] J. Chaki and N. Dey, “Statistical Texture Features,” in *Texture Feature Extraction Techniques for Image Recognition*, Singapore, Springer, 2020, pp. 7-24.

[26] M. Sharma and H. Ghosh, “Histogram of gradient magnitudes: A rotation invariant texture-descriptor,” in *IEEE International Conference on Image Processing*, Quebec City, 2015.

[27] D. Naik and P. Shah, “A Review on Image Segmentation Clustering Algorithms,” *International Journal of Computer Science and Information Technologies,* vol. 5, no. 3, p. 2014, 3289 - 3293.

[28] A. P. Dhawan, “Image Segmentation,” in *Medical Image Analysis*, New Jersey, John Wiley & Sons, 2011, pp. 229-264.

[29] O. Dikshit, “Remote Sensing: Information Extraction,” NPTEL, [Online]. Available: https://nptel.ac.in/courses/Webcourse-contents/IIT-KANPUR/ModernSurveyingTech/lectureD_28/D_28_4.htm. [Accessed 17 May 2019].

[30] A. Jain, D. Singh and D. A. Kumar, “A Comparative Review of Various Image Segmentation Techniques for MRI Images,” Delhi Technological University, Delhi, 2014.

[31] A. Ralhan, “Self Organizing Maps,” Towards Data Science, [Online]. Available: https://towardsdatascience.com/self-organizing-maps-ff5853a118d4. [Accessed 11 May 2019].

[32] D. Foley, “Gaussian Mixture Modelling,” Towards Data Science, [Online]. Available: https://towardsdatascience.com/gaussian-mixture-modelling-gmm-833c88587c7f. [Accessed 11 May 2019].

[33] R. Nishii and S. Tanaka, “Accuracy and inaccuracy assessments in land-cover classification,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 37, no. 1, pp. 491-498, 1999.

[34] R. G. Congalton and K. Green, Assessing the Accuracy of Remotely Sensed Data, Principles and Practices, Third Ed., New York: CRC Press, 2019.

[35] A. D. C. Nascimento, R. . J. Cintra and A. C. Frery, “Hypothesis Testing in Speckled Data With Stochastic Distances,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 48, no. 1, pp. 373-385, Jan 2010.

[36] A. C. Frery, A. D. C. Nascimento and R. J. Cintra, “Contrast in Speckled Imagery with Stochastic Distances,” in *IEEE International Conference on Image Processing*, Hong Kong, China, Sept 2010.

[37] “MSTAR Dataset,” [Online]. Available: https://www.sdms.afrl.af.mil/index.php?collection=mstar. [Accessed 02 November 2019].