e-ISSN: 0974-4614
p-ISSN: 0972-0448

Texture Segmentation Algorithm Using Multivariate New Symmetric Mixture Model and Dct Coefficients

Prasanthi.Mandha[1], Dr.Vamsidhar Enireddy[2]

[1]Research Scholar, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation,Guntur(D), Andhra Pradesh, Email: dhavalaprasanthi3@gmail.com
[2]Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur(D), Andhra Pradesh, Email: enireddy.vamsidhar@gmail.com

Received: 14.08.2024 Revised: 17.09.2024 Accepted: 20.10.2024

ABSTRACT

Among the most challenging aspects of processing digital images is image classification, which handles the processing and analysis of the images. In higher level image processing applications like medical imaging, robotics, automation, and other key areas where the object study is thought to be of highest importance, the outputs obtained from the segmentation findings are treated as starting parameters. The segmentation process uses pre-requisite criteria to find objects of interest. The feature vector is typically taken to be this criterion. Separating the diverse set of images is the new goal of texture segmentation into regions of homogeneous pixels that have significance, based on this criterion. Texture is regarded as the most important of these feature vectors since it explains the spatial relationship between individual pixels in an image.

Keywords: robotics, automation, Texture, spatial

1. INTRODUCTION

In the literature, numerous segmentation techniques based on parametric, non-parametric, and pseudo-parametric models are highlighted [1]. These models have been regarded as the foundational models for texture-based segmentation by [2], Classification trees, vector quantization, Markovian models, neural networks, decision trees, and other topics are also the subject of research [3]. Nonetheless, it is clear from the research of [4] others that parametric models are more important and appropriate for modelling image pixels. Consequently, many statistical mixture models are prioritized for efficient picture pixel segmentation based on this fundamental premise [5]. The majority of the research, however, operate under the presumption that any image they study will be symmetrical and that the pixels that make up the image areas are uniform and display a typical Gaussian phenomenon. It is also known as meso-kurtic phenomenon. In actuality, however, natural photographs may display patterns that are partially symmetric or non-symmetric; that is, the pixels within the image regions might exhibit non-Gaussian, mixed-Gaussian, or Gaussian characteristics.Consequently, in order to comprehend these kinds of image fluctuations, the primary observation of the statistical models that can support meso-, platy-, and lepto-kurtic distributions is that the statistical models should be created in a way that better identifies the pixel pattern. Consequently, taking into account the Multivariate New Symmetric Mixture Model is required. Therefore, a statistical model based on the multivariate new symmetric mixture model and dct coefficients (MNSMM) is proposed in this chapter as an attempt to effectively segment the textured image [6].

The k-Means approach, a traditional partition clustering algorithm, is utilized for successfulclustering after we took into account the texture picture's DCT attributes for image segmentation [7]. The benchmark texture database used for the experimentation is Brodatz textures [8]. Quality metrics and segmentation metrics are both used in the review process [9]. This chapter of the thesis uses segmentation measures (PRI, GCE, and VOI) and quality metrics (F-measure, recall, precision, sensitivity, and accuracy) to evaluate the texture pictures [10].In addition, the misclassification rate is calculated. Dendogram-based clustering is another way to improve the experiments. The results obtained are contrasted with those of the Multivariate New Symmetric Mixture Model that is currently in use [11].

2. Feature Vector Extraction Based On Dct Coefficients:

Most of the segmentation work is based on attributes. These traits, which are also known as attributes, help in identifying the target more elegantly and precisely.

Improving the object's focus identification. Various methods for extracting features from textures can be found in the literature. for example, DCT, PCA, LDA, ICA, etc., as discussed in papers by [12]. The segmentation technique relies on a number of shared characteristics shared by most pixels inside texture regions [13]. This is because there is a uniform spatial relationship between neighbouring pixels inside the image regions [14]. Among the dimensionality reduction methods used for segmentation, DCT accounts for features inferred from spatial correlations [15].

Hence, it is presumed that using DCT as a feature vector yields better suitable segmentation results [16]. DCT works by dividing the image into smaller signals in each region, with the goal of storing the visual data in the coefficients. This process takes into account each section of the image [17]. The DCT is capable of shifting from the space-time to the time-frequency domainsfor a texture image [18]. The main reason to use DCT is that it can merge the DCT coefficients into one region by taking on a decomposing nature, depending on the texture qualities that are chosen [19]. Another strong argument in Favor of using DCT is that it lowers the lowest square reconstruction error when applied in conjunction with other dimensionality reduction approaches. Citation [20]. One issue with DCT is that, because to its packing and rebuilding nature, it causes lossy compression [21]. This section of the thesis so focuses on the method of DCT-based texture picture segmentation [22].

It is common practice to use a 2-dimensional array f(x,y) to represent images. Hence, we split the whole picture into non-overlapping subblocks of size M × M for effective texture feature mining. Each pixel's DCT coefficients are determined using the following formula: f(x,y).

$$C(v,u) = \alpha(v)\alpha(u)\sum_{y=0}^{M-1}\sum_{x=0}^{M-1} f(y,x)\beta(y,x,v,u)$$

For v, u = 1,2,3,........,M

with $\alpha(v)$ = $\begin{cases} \sqrt{\dfrac{1}{M}} & \text{for } v = 1 \\ \sqrt{\dfrac{2}{M}} & \text{for } v = 2,3,\ldots\text{M} \end{cases}$

and $\beta(y,x,v,u) = \cos\left(\dfrac{(2y+1)v\pi}{2M}\right)\cos\left(\dfrac{(2x+1)u\pi}{2M}\right)$

The DCT coefficients that were extracted from each image block are laid out in a zigzag manner, as seen in figure 2.1. The important information is projected using the coefficients of the upper triangle block matrix. According to [23] a small number of DCT coefficients, around 16, is sufficient to extract the relevant information about the picture areas' patterns. Once the DCT coefficients have been extracted from each image, the feature vectors can be transformed using the following procedure: $\vec{X}_t = [\vec{X}_1, \vec{X}_2, \vec{X}_3, ..., \vec{X}_N]^T$ symbolizing M multiplied by 16 coefficients, with M representing the overall count of blocks. This means that the proposed segmentation model according to MGGMM acquires these texture attributes [24].

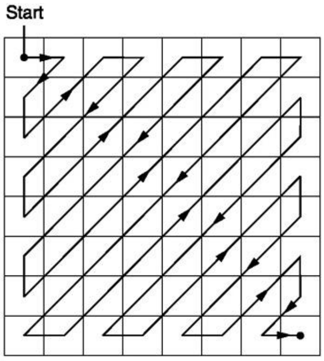


Figure 2.1: Arrangement of DCT Coefficients inpattern of zig-zag manner.

2.2 Multivariate New Symmetric Mixture Model

In this part of the chapter, the Multivariate New Symmetric Mixture Model (MNSMM) methodology is presented. Improved texture segmentation is one goal of the offered methodology. Section 2.2 outlines the methodology that was used for attribute extraction, and the parameters of the proposed model are developed appropriately. Every single image is actually a composite of numerous smaller images, each of which showcases a different texture. It becomes more of a chore to ensure that the growth is proportional to the feature vectors as the number of images grows. With the assumption that the feature vectors in each of these picture regions reflect a multivariate Multivariate New Symmetric Mixture Model, this chapter of the thesis approximates each unique image to follow an M-component mixture distribution. Hence, it is presumed that an M-component multivariate Multivariate New Symmetric Mixture Model applies to each feature vector of subblock within the image [25]. Therefore,the cumulative probability density function for each image region under evaluation will look like this:

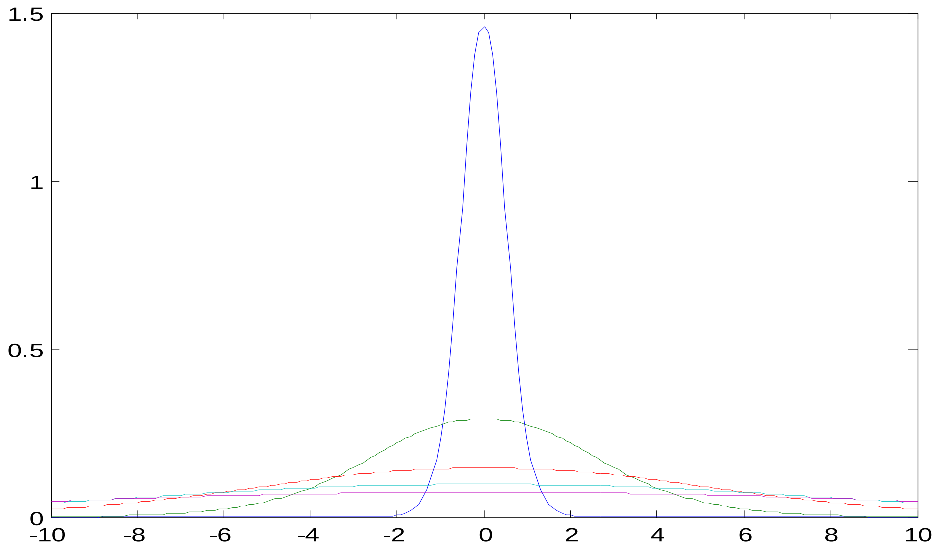
$$p(\vec{x}_r / \theta) = \sum_{i=1}^{M} w_i g_i \left(\vec{x}_{r,} \theta\right)$$

where, $\vec{x}_r = (x_{rij})$, j = 1,2,3,..D, represents the feature vector, where i ranges from 1 to D.The texture areas are represented by M, and r ranges from 1 to.The pixels are represented by T. $\Theta = (\mu, \sigma, \beta)$ is a set of values representing parametric set θ

w$_i$is the combined weight, such that $\sum_{i=1}^{M} w_i = 1$

and the probability that the texture feature vector of each pixel in the picture is

$g_i \left(\vec{x}_{r,} \theta\right)$ (Allili, M. S. et al., 2008) provides the multivariate new symmetric distribution for the D-dimension [26].



Frequency curve of Multivariate New Symmetric Mixture Model distribution

Multivariate New Symmetric Model for $\vartheta_i(\bar{x}_r, \theta)$ is given by

$$\vartheta_i(\bar{x}_r, \theta) = \prod_{j=1}^{D} \left| \frac{2 + \left[\frac{(x_j - \mu_j)}{\sigma_j}\right]^2}{3\sigma_j \sqrt{2\pi}} \right| e^{-\frac{1}{2}\left[\frac{(x_j - \mu_j)}{\sigma_j}\right]^2}$$

Where μ_j ,σ_j are the parameters

The mean value of the MNSMM is given by

$$\mu_{i_j}{}^{(l+1)} = \frac{\sum_{r=1}^{T} t_i \left(\left(\bar{x}_r, \theta^{(l)}\right)\right)\left(x_{i_j}\right) - \sum_{r=1}^{T} t_i\left(\bar{x}_r, \theta^{(l)}\right)\left[\frac{\left[x_{i_j} - \mu_{i_j}{}^{(l)}\right] 2\sigma_{i_j}{}^2}{\left[2\sigma_{i_j}{}^2 + \left[x_{i_j} - \mu_{i_j}{}^{(l)}\right]^2\right]}\right]}{\sum_{r=1}^{T} t_i \left(\bar{x}_r, \theta^{(l)}\right)}$$

The MNSMM is symmetric about the mean, therefore the odd center moments cancel each other out to zero. i.e., $E\left|x_j - \mu_j\right|^t = 0$, t = 1,3,5,…. The sum of the absolute center moments and the even center moments, as calculated using the formula.

$$E\left|x_j - \mu_j\right|^t = \left[\frac{\sigma_j^2 \Gamma\left(\frac{1}{\theta_j}\right)}{\Gamma\left(\frac{3}{\theta_j}\right)}\right]^{t/2} \frac{\Gamma\left(\frac{t+1}{\theta_j}\right)}{\Gamma\left(\frac{1}{\theta_j}\right)}$$

The variance is given by

$$\sigma_{i_j}{}^{2(l+1)} = \frac{\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r, \theta^{(l)}\right)\right)\left[\left(x_{i_j} - \mu_{i_j}{}^{(l+1)}\right)^2\left[1 - \frac{2\sigma_{i_j}{}^{(l)}}{\left[2\sigma_{i_j}{}^{(l)} + \left(x_{i_j} - \mu_{i_j}{}^{(l+1)}\right)^2\right]}\right]\right]}{\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r, \theta^{(l)}\right)\right)}$$

var (x) = $E(x - \overline{x})^2 = E(x - \mu)^2 = \sigma^2$

The investigated distribution's parametric set is defined by, which includes the shape parameter, mixing weights, and a matrix of covariance and mean vector. Choose between a grand co-variance matrix for each Gaussian component or a single co-variance matrix shared by all image textures to specify the unique case-specific implementations of the MNSMM model [27]. Throughout this thesis, we have solely focused on the covariance matrix, which can take on a fully diagonal or partial diagonal form. This depends on what Douglas A. Reynolds and colleagues found in their 1990 study,this chapter examines the half covariance diagonal matrix, which, according to preliminary experimental results, produces higher segmentation performance [28]. As a result, we may express the covariance as

$$\sum_i = \begin{bmatrix} \sigma_{i1}^2 & . & . & . \\ . & \sigma_{i2}^2 & . & . \\ . & . & . & . \\ . & . & . & \sigma_{iD}^2 \end{bmatrix}$$

Each feature is considered independent the feature set's PDF is given by, as a consequence of the diagonal covariance matrix

$$\vartheta_i(\bar{x}_r, \theta) = \prod_{j=1}^{D}\left[\frac{2 + \left[\frac{(x_j - \mu_j)}{\sigma_j}\right]^2}{3\sigma_j\sqrt{2\pi}}\right] e^{-\frac{1}{2}\left[\frac{(x_j - \mu_j)}{\sigma_j}\right]^2}$$

$$= \prod_{j=1}^{D} f_{ij}(x_{rij})$$

Estimation Of Model Parameters By Expectation- Maximization Algorithm

The research paper focuses on the model parameters that should be considered in order to provide the best possible likelihood estimations. The DCT coefficients are shown for every single pixel that is removed from the image regions. as $(\vec{x}_1, \vec{x}_2, \ldots\ldots, \vec{x}_r)$ hence, it is possible to compute the combined PDF using [29].

The mixture model is

$$p\left(\frac{\bar{x}_r}{\theta}\right) = \sum_{l=1}^{m} w_i \, \vartheta_i(\bar{x}_r, \theta)$$

Where $\vartheta_i(\bar{x}_r, \theta)$ is known as equation----1

The likelihood estimate is given by

$$L(\theta) = \prod_{r=1}^{T}\left[\sum_{l=1}^{m} w_i \, \vartheta_i(\bar{x}_r, \theta)\right]$$

$$L(\theta) = \prod_{r=1}^{T}\left[\sum_{l=1}^{m}\left[\prod_{j=1}^{D}\left[\frac{2+\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2}{3\sigma_{i_j}\sqrt{2\pi}}\right]e^{-\frac{1}{2}\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2}\right]\right]$$

The log likely hood is given by

$$log\big(L(\theta)\big) = log\left(\prod_{r=1}^{T}\left[\sum_{l=1}^{m}\left[\prod_{j=1}^{D}\left[\frac{2+\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2}{3\sigma_{i_j}\sqrt{2\pi}}\right]e^{-\frac{1}{2}\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2}\right]\right]\right)$$

The modified parameter estimations w_i, μ_{ij} and σ_{ij} for i=1,2,3,M; j=1,2,..,D, are obtained by maximizing the logarithmic likelihood or anticipated value likelihood function. We use the method described by Shaoquan YU (2012) to calculate the shape parameter, β_{ij}.

To estimate the values of the parameters, w_i, μ_{ij} and σ_{ij} ,EM algorithm comprising of two steps i.e., Uses the two-step EM method, which stands for expectation (E) and maximization (M). Preliminary estimate extraction from input image data containing texture regions is required at the outset of the EM method. The EM approach is designed to assess the maximum likelihood estimate of the unknown parametric set iteratively.'θ'.

Method of Excepectation

In this section of the research process, relative to the original parameter vector, there is an expectation (E) step. $\theta^{(l)}$ is computed. One way to determine the pixel's expected value from the log probability function is to apply [30].

$$Q(\theta,\theta^{(0)}) = E_{\theta^{(l)}}\left\{\log L(\theta / \vec{x}_r)\right\}$$

Given the parameter estimates, say $\theta^{(l)} = (\mu_{ij}^{(l)}, \sigma_{ij}^{(l)})$ for i = 1,2,..M; j = 1,2,3,..D, the estimate the PDF is given like

$$p(\vec{x}_t / \theta) = \sum_{i=1}^{M} w_i . g_i\left(\vec{x}_r, \theta\right)$$

--- (2.21)

If we assume that each given pixel (x_r) corresponds to the M^{th} region, then the conditional probability is

$$t_i(\vec{x}_r / \theta^{(l)}) = p((i/x), \theta^{(l)})$$
$$= \frac{w_i g_i(\vec{x}_r, \theta^{(l)})}{p_i(\vec{x}_r, \theta^{(l)})}$$
$$= \frac{w_i g_i(\vec{x}_r, \theta^{(l)})}{\sum_{i=1}^{M} w_i g_i\left(\vec{x}_r, \theta\right)}$$

--- (2.22)

The expected value of likelihood, L(θ) is

$$Q(\theta,\theta^{(0)}) = E_{\theta^{(l)}}\left\{\log L(\theta / \vec{x}_r)\right\}$$ **- (2.23)**

Following Jeff A. Bilmes's (1997) heuristic reasoning, one can get
The log likely hood is given by

$$log\big(L(\theta)\big) = log\left(\prod_{r=1}^{T}\left[\sum_{l=1}^{m} w_i \vartheta_i(\bar{x}_r, \theta)\right]\right)$$

$$= \sum_{r=1}^{T}\left[\sum_{l=1}^{m} w_i\, \vartheta_i(\bar{x}_r, \theta)\right]$$

$$log\big(L(\theta)\big) = log\left(\sum_{I=1}^{m} w_i\left[\prod_{j=1}^{D}\left[\frac{2+\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2}{3\sigma_{i_j}\sqrt{2\pi}}\right] e^{-\frac{1}{2}\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2}\right]\right)$$

M-Step

This step aims at maximizing $Q(\theta, \theta^{(l)})$ to the point that the phrases that containw$_i$and $\theta^{(l)}$, since they are uncorrelated. The probability function should be maximized in such a way that the mixing weights w$_i$ are updated.

$$\sum_{i=1}^{M} w_i = 1$$

-- (2.25)

Estimation of the first-order Lagrangian type function is possible by means of

$$L=\sum_{r=1}^{T}\sum_{i=1}^{M}\log(w_i)\,t_i(\vec{x}_r, \theta^{(l)}) + \gamma\left(\sum_{i=1}^{M} w_i^{(l)} - 1\right)$$

--- (2.26)

where, γ is The Lagrangian function must be differentiated in order to maximize it with respect to wi, and a Lagrangian multiplier is necessary for this. with relation to wi and then set it equal to zero.

$$= \frac{\partial}{\partial w_i}\left[L\right] = 0$$

$$= \frac{\partial}{\partial w_i}\sum_{r=1}^{T}\sum_{i=1}^{M}\log(w_i)\,t_i(\vec{x}_r, \theta^{(l)}) + \frac{\partial}{\partial w_i}\gamma\left(\sum_{i=1}^{M} w_i^{(l)} - 1\right) = 0$$

--- (2.27)

$$\frac{\partial}{\partial \mu_{i_j}}\Big(log\big(L(\theta)\big)\Big) = \frac{\partial}{\partial \mu_{i_j}}\left[\sum_{r=1}^{T}\left[\sum_{i=1}^{m} log(w_i) t_i\left(\bar{x}_r, \theta^{(l)}\right)\right] + \sum_{r=1}^{T}\left[\sum_{I=1}^{m}\left[\sum_{j=1}^{D} log\left(\left[\frac{2+\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2}{3\sigma_{i_j}\sqrt{2\pi}}\right] e^{-\frac{1}{2}\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2}\right) t_i\Big(\left(\bar{x}_r, \theta^{(l)}\right)\Big)\right]\right]\right] = 0$$

$$\begin{aligned}
&\frac{\partial}{\partial \mu_{i_j}}\Big(log\big(L(\theta)\big)\Big) \\
&\qquad = \frac{\partial}{\partial \mu_{i_j}}\left[\sum_{r=1}^{T} log\left[2+\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2\right] t_i\Big(\left(\bar{x}_r, \theta^{(l)}\right)\Big)\right. \\
&\qquad\left. + \frac{\partial}{\partial \mu_{i_j}}\sum_{r=1}^{T}\left[-\frac{1}{2}\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2 t_i\Big(\left(\bar{x}_r, \theta^{(l)}\right)\Big)\right] - \frac{\partial}{\partial \mu_{i_j}}\sum_{r=1}^{T} 3\sigma_{i_j}\sqrt{2\pi}\, t_i\Big(\left(\bar{x}_r, \theta^{(l)}\right)\Big)\right] = 0
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial \mu_{i_j}}\Big(log\big(L(\theta)\big)\Big) &= \left[\sum_{r=1}^{T}\frac{\partial}{\partial \mu_{i_j}}\left[log\left[2+\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2\right]\right] t_i\Big(\left(\bar{x}_r, \theta^{(l)}\right)\Big) - \sum_{r=1}^{T}\frac{\partial}{\partial \mu_{i_j}}\left[\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2 t_i\Big(\left(\bar{x}_r, \theta^{(l)}\right)\Big)\right]\right] \\
&= 0
\end{aligned}$$

$$\frac{\partial}{\partial \mu_{i_j}}\Big(log\big(L(\theta)\big)\Big) = \sum_{r=1}^{T}\left[\frac{-2\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]\frac{1}{\sigma_{i_j}} t_i\Big(\left(\bar{x}_r, \theta^{(l)}\right)\Big)}{2+\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]^2}\right] - \sum_{r=1}^{T}\left[\left[-\frac{2}{\sigma_{i_j}}\right]\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right] t_i\Big(\left(\bar{x}_r, \theta^{(l)}\right)\Big)\right] = 0$$

$$\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)\left[\left[\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right]\left[\frac{-2}{\left[2+\left|\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right|^2\right]}+1\right]\right]=0$$

$$\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)\left[x_{i_j}-\mu_{i_j}\right]\left[\frac{-2\sigma_{i_j}{}^2}{\left[2\sigma_{i_j}{}^2+\left(x_{i_j}-\mu_{i_j}\right)^2\right]}\right]+\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\left(x_{i_j}-\mu_{i_j}\right)\right)=0$$

$$\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\left(x_{i_j}-\mu_{i_j}\right)\right)=\sum_{r=1}^{T}\frac{t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\left[x_{i_j}-\mu_{i_j}\right]2\sigma_{i_j}{}^2\right)}{\left[2\sigma_{i_j}{}^2+\left[x_{i_j}-\mu_{i_j}\right]^2\right]}$$

$$\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\left(x_{i_j}\right)\right)-\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\left(\mu_{i_j}\right)\right)=\sum_{r=1}^{T}\frac{t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\left[x_{i_j}-\mu_{i_j}\right]2\sigma_{i_j}{}^2\right)}{\left[2\sigma_{i_j}{}^2+\left[x_{i_j}-\mu_{i_j}\right]^2\right]}$$

$$\mu_{i_j}\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)=-\sum_{r=1}^{T}\frac{t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\left[x_{i_j}-\mu_{i_j}\right]2\sigma_{i_j}{}^2\right)}{\left[2\sigma_{i_j}{}^2+\left[x_{i_j}-\mu_{i_j}\right]^2\right]}+\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\left(x_{i_j}\right)\right)$$

$$\mu_{i_j}{}^{(l+1)}=\frac{\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)\left(x_{i_j}\right)-\sum_{r=1}^{T} t_i\left(\bar{x}_r,\theta^{(l)}\right)\left[\frac{\left[x_{i_j}-\mu_{i_j}{}^{(l)}\right]2\sigma_{i_j}{}^2}{\left[2\sigma_{i_j}{}^2+\left[x_{i_j}-\mu_{i_j}{}^{(l)}\right]^2\right]}\right]}{\sum_{r=1}^{T} t_i\left(\bar{x}_r,\theta^{(l)}\right)}$$

For updating $\sigma_{i_j}{}^2$

$$\frac{\partial}{\partial\sigma_{i_j}{}^2}Q\left(\theta,\theta^{(l)}\right)=0$$

$$\frac{\partial}{\partial\sigma_{i_j}{}^2}E\left(log\left(\theta^{(l)}\right)\right)=0$$

$$\frac{\partial}{\partial\sigma_{i_j}}\sum_{r=1}^{T}\sum_{i=1}^{m}\sum_{j=1}^{D} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)\left[log\left[2+\left|\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right|^2\right]-\frac{1}{2}\left|\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right|^2-log\left(3\sqrt{2\pi}\right)-\frac{1}{2}log\left(\sigma_{i_j}{}^2\right)\right.$$
$$\left.+\,log(w_i)\right]=0$$

$$\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)\left[log\left[2+\left|\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right|^2\right]-\frac{1}{2}\left|\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right|^2-\frac{1}{2}log\left(\sigma_{i_j}{}^2\right)+log(w_i)\right]=0$$

$$\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)\left[\frac{-\left(x_{i_j}-\mu_{i_j}\right)^2}{\sigma_{i_j}{}^4}\left[\frac{1}{2}-\frac{1}{\left[2+\left|\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right|^2\right]}\right]-\frac{1}{2\sigma_{i_j}{}^2}\right]=0$$

$$\sum_{r=1}^{T} t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)\left[\frac{-\left(x_{i_j}-\mu_{i_j}\right)^2}{\sigma_{i_j}{}^4}\left[\frac{1}{2}-\frac{1}{\left[2+\left|\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right|^2\right]}\right]\right]-\frac{1}{2}\sum_{r=1}^{T}\sigma_{i_j}{}^2\,t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)=0$$

$$\sum_{r=1}^{T}\frac{{\sigma_{i_j}}^{2}}{2}t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)=\sum_{r=1}^{T}t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)\left[\frac{-\left(x_{i_j}-\mu_{i_j}\right)^{2}}{{\sigma_{i_j}}^{4}}\left[\frac{1}{2}-\frac{1}{\left[2+\left|\frac{\left(x_{i_j}-\mu_{i_j}\right)}{\sigma_{i_j}}\right|^{2}\right]}\right]\right]$$

$${\sigma_{i_j}}^{2(l+1)}=\frac{\sum_{r=1}^{T}t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)\left[\left(x_{i_j}-{\mu_{i_j}}^{(l+1)}\right)^{2}\left[1-\frac{2{\sigma_{i_j}}^{(l)}}{\left[2{\sigma_{i_j}}^{(l)}+\left(x_{i_j}-{\mu_{i_j}}^{(l+1)}\right)^{2}\right]}\right]\right]}{\sum_{r=1}^{T}t_i\left(\left(\bar{x}_r,\theta^{(l)}\right)\right)}$$

$$\alpha_i=W_I^{(l+1)}=\frac{1}{T}\sum_{t=1}^{T}\left[\frac{{\alpha_i}^{(l)}\vartheta_i(\bar{x}_r,\theta)}{\sum_{i=1}^{m}{\alpha_i}^{(l)}\vartheta_i(\bar{x}_r,\theta)}\right]$$

Initialization Of Model Parameters

It is necessary to update the parameters using the EM algorithm in order to determine the initial estimations of the parametric set for the specified MNSMM. According to McLachlan et al. (2000), the number of groups and the model's starting parameter estimation are the two most important constraints on the execution of the EM algorithm. It is necessary to compute the revised equations for W_i , the mixing parameter and μ_{ij} and σ_{ij}. Initial knowledge regarding picture parameters is rather uncertain due to the unsupervised nature of the procedure. This is accomplished by applying the methodology of likelihood estimation to the problem of estimating these values, This section covers the k-Means algorithm. In this section, we calculate the images of M-components using a clustering method and the k-Means methodology.

K-Means Clustering Algorithm

Because the M-components estimation inside the picture regions is completely random, an unsupervised learning approach should be considered. In this section of the thesis, we examine the use of k-Means clustering, which depends on random sampling and can estimate the count of clusters in both low and high dimensions, to evaluate the count of components within the texture, say m. (Mclanchan and Peel, 2000).

At the outset, we separate each image into k blocks by finding the centroid of each. Once convergence occurs, the centroids' computation continues in a loop. According to Patil and Jondhale (2010), the technique has the extra benefit of maximizing the sum of squared errors among the pixels in each cluster. Determining the starting point for the number of clusters is another bottleneck in k-Means. To get around this problem and get the starting count of clusters needed to run the k-Means algorithm, this chapter of the thesis plots the histogram of the entire image. Avoiding over- or under-fitting is possible using this approach to determining the starting point for cluster size. The following is a flow diagram of the k-Means algorithm.

1. Pick M starting clusters at random from the complete dataset. These numbers stand for the initial cluster centers.
2. Find the distance between the cluster and each pixel in the image centroid by computing their Euclidean distance. Then, allocate each data point to the cluster center that is closest to it. In the beginning, points represent the cluster centers.
3. Make sure that the squared error distance between each cluster is lowest by recalculating the new cluster center.
4. Keep repeating steps two and three until the clustering centers stay the same.
5. Stop the process

After cluster centers are assigned to their closed cluster centers, the cluster centroid is updated. The k-Means algorithm's primary strength lies in its inherent knowledge of cluster properties, particularly the requirement for low levels of similarity within clusters. The technique requires the user to input the optimal number of clusters for a given texture image, which is a potential downside(Turi, Rose H., 2001). Crucially, the k-Means algorithm's efficacy is solely dictated by the initial estimations and the quantity of clusters chosen.

Modeling can be accomplished using the following formulathe sample moments of the M groups, which are initial estimations.

w_i= 1/M

σ_{ij} = Std.

Deviation of Mth Class

$$\mu_{ij} = \frac{1}{T}\sum_{r=1}^{T} x_{rij}$$

Substituting these values into the aforementioned equations in MATLAB allows one to get the refined parameter estimations using the EM Algorithm.

Hierarchical Clustering Algorithm

Using hierarchical clustering methodology, this thesis further estimates the M-components of the picture regions. When the segments are very densely packed, hierarchical segmentation works well. By implementing the hierarchical clustering algorithm, this section of the thesis is further expanded. Its operation is similar to that of the k-Means algorithm, with the exception that this one uses a dendogram to determine the starting point for the number of clusters. Data clustering is accomplished using the constructed dendogram. Clustering is performed with respect to a threshold that is originally identified using a rough estimate. In order to determine how many clusters can primarily be addressed using the single, average, and complete linkage approaches, the threshold values that have been acquired can be used. Hierarchical clustering using average linkages is the focus of this section of the thesis.

Presented below is the detailed procedure that comprises a hierarchical clustering approach (Johnson S.C., 1967):

1) The initial process is carried out by assigning each pixel on the image region to a segment. Thereby every N pixels are associated with N segments containing only a unique set.
2) Identify the nearest pair of pixels within a segment and merge them such that each pixel is connected to another pixel.
3) Keep going until every single pixel is assigned to a specific zone.

Methods a) Single-Linkage, b) Complete-Linkage, and c) Average-Linkage segmentation are three distinct approaches to Step 3. The segmentation approach is examined in this thesis by taking the Average -Linkage methodology into account. With Average-Linkage segmenting, the rows and columns of the closeness matrix are erased as old segments are merged into new ones. This is the main advantage of the technique, which is also known as the unweighted pair-group method using arithmetic averages. The process involves finding the average distance between two segments, which is calculated by comparing the pixels in one segment to those in the other.

The closeness matrix, often known as the proximity matrix, is represented by the equation D= [d(i,j)]. Every segment has a numeric value between 0 and (n-1), with L(K) being the level of the Mth segment. A segment with sequence number m is represented by (m), while the proximity of segments (r) and (s) is denoted by d[(r),(s)].

The algorithm is illustrated as below-

1. Start with non intersecting segments at level 0 such that L(0)=0 with sequence number m equal to 0.
2. Compute the average among the pair of pixels from dissimilar segments denoted as [(r), (s)] and the process is repeated for all the pixels in the current segment.
3. Update the sequence number by one i.e., m=m+1. Combine the segments (r) and (s) in to a set to form the next segment m and set the level as L (m) = d[(r),(s)] for this segment.
4. Make a new row and column for the newly generated segment and remove the rows and columns that correspond to segments (r) and (s) from the proximity matrix, D. Newly formed segment (r,s) and prior segment (M) closeness is recomputed using the formula.

$$d_{(r,s)\,M} = \frac{\sum_{i}\sum_{j} d(i,j)}{N_{(r,s)}\,N_M} \qquad \textbf{--- (2.50)}$$

The distance among objects i and j in clusters (r, s) and M, respectively, is represented by d(i, j). , $N_{(r,s)}$ and $N_{(M)}$ being the number of pixels in the clusters (r,s) and M respectively.

Repeating this procedure continues until the distance among every two clusters falls below a certain threshold. Initial parameter estimates are derived using sample moments in a manner similar to the k-Means approach once the final group M value has been identified.

w_i= 1/M

σ_{ij} = Std. Deviation of Mth Class

$$\mu_{ij} = \frac{1}{T}\sum_{r=1}^{T} x_{rij}$$

After plugging in these numbers as starting points, one can use the EM Algorithm to simultaneously solve the equations in a PYTHON environment to get the refined estimated parameters.

A New Symmetric Mixture Movement-Based Texture Segmentation Algorithm

Here we present the algorithm for texture segmentation. The first stage, after determining the final estimations, is to catalog the picture textures' characteristics and assign each characteristic to a certain image segment.

This process is carried out by segmentation algorithm comprising of the following steps.

Step 1: Using the process discussed in section 2.2, the feature vectors are estimated from each of the textured images.

Step 2: Partition the image pixels in to M regions either employing a hierarchical clustering technique or k-Means clustering.

Step 3: Calculate μ_{ij} and σ_{ij}, the mean and variance vector for each region among the multivariate data.

Step 4: Obtain the mixing weight from the formula $w_i = 1/M$, for i=1,2,3,..M values.

Step 5: Using the heuristics presented in section 2.4, the updated parameters w_i, μ_{ij} and σ_{ij} are computed for each image region.

Step 6: The process of assigning the feature vectors to the corresponding segments (j^{th}segment) is carried out using the process of maximum likelihood estimation and the equation for which is given by

$$L_j = \max\left\{\prod_{j=1}^{D} \frac{\exp\left(-\left|\frac{x_{ij}-\mu_{ij}}{A(\rho_{ij},\sigma_{ij})}\right|^{\beta_{ij}}\right)}{2\Gamma\left(1+\frac{1}{\beta_{ij}}\right)A(\beta_{ij},\sigma_{ij})}\right\}$$

Experimentation And Results

The suggested approach is tested on a texture dataset, specifically the Brodatz texture databases, in order to demonstrate its performance. As part of the experiment, we look at every image in the dataset and plot the histogram to find out how many peaks (M) there are in each one. The amount of usable image regions can be deduced from these peaks. The clustering method, k-Means in this instance, is fed these peak values (M) in order to achieve a more thorough segmentation procedure. The picture feature vectors' multivariate data is handled using the k-Means algorithm. which helps to divide the image into M groups. The other advantage with the clustering algorithm is that it helps to formulate the initial parameters w_i, μ_{ij} and σ_{ij} by following the method of moments. Figure 2.4 displays the histograms of five photos. Below are the corresponding photographs: Image 1 through Image 5.

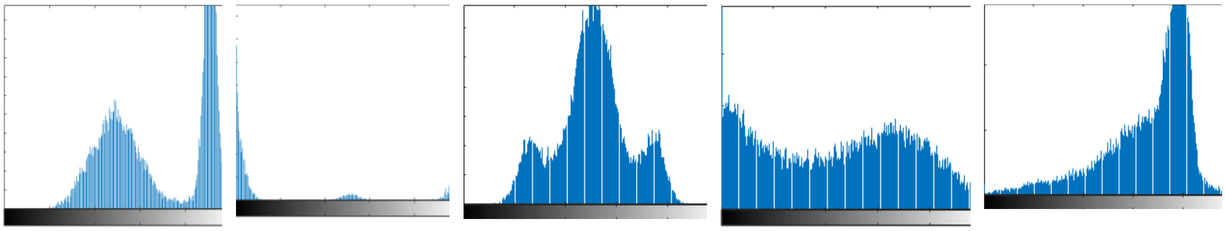


Image 1 **Image 2** **Image 3** **Image 4** **Image 5**

Figure 2.3: Histogram plot of the images**.**

In order to apply the hierarchical clustering technique, a comparable comparison is made. Applying this clustering technique to multivariate image data with various feature vectors allows for the dendogram approach to partition the data into M image areas, each with its own threshold. The starting points for estimating the parameters for each area of the image, w_i, μ_{ij} and σ_{ij} are obtained.

To get the updated model parameter equations, we use the EM approach that was suggested in section 2.4. On top of the updated estimations, the texture segmentation process is executed, with the goal of segmenting each image pixel according to the maximum probability criterion. Consequently, the segmentation algorithm provided in section 2.6 is used to segment the image.

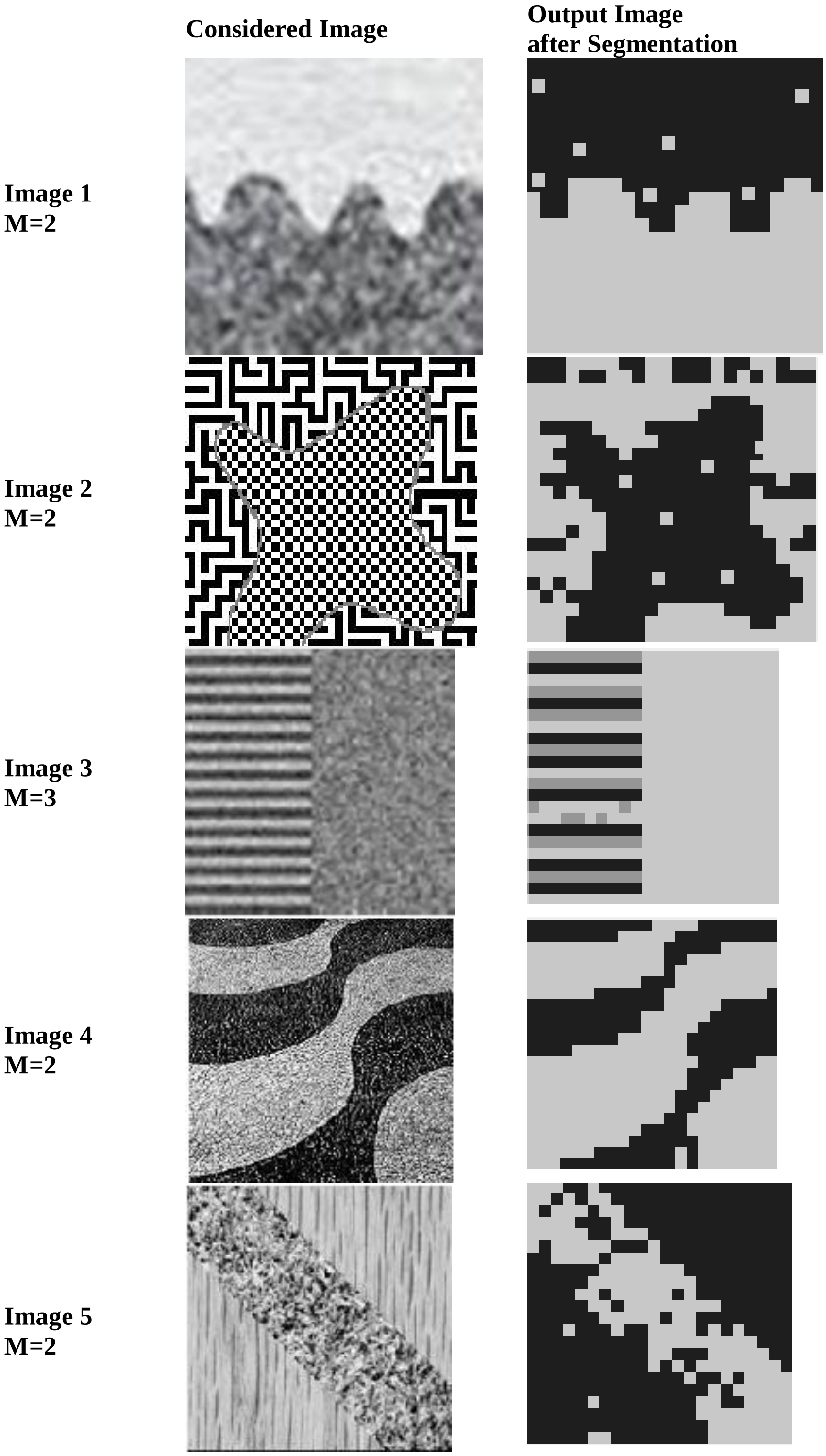


Figure 2.4: As a result of implementing the suggested model with the k-Means clustering technique**,**

Considered Image **Output Image after Segmentation**

Image 1
M=2

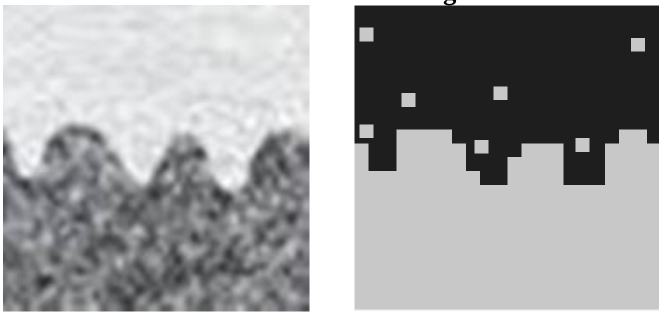


Image 2
M=4

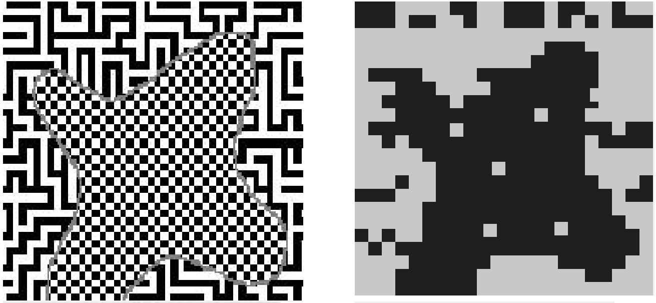


Image 3
M=3

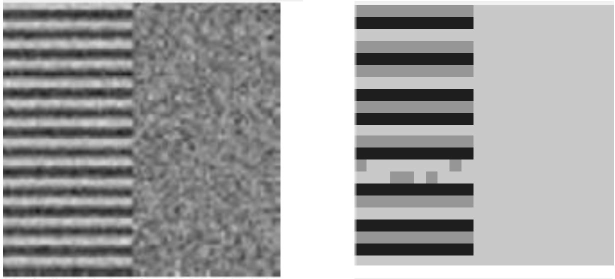


Image 4
M=2

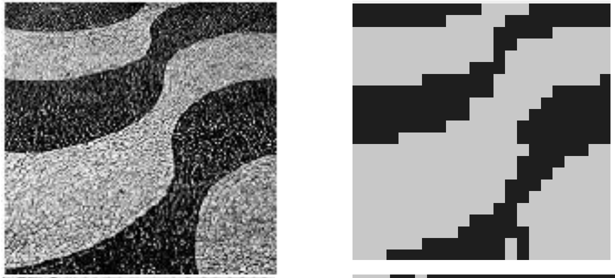
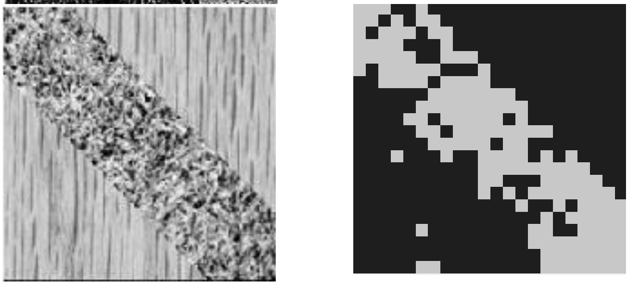
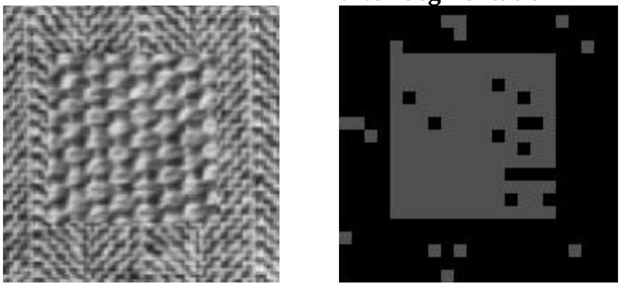


Image 5
M=2



Considered Image **Output Image after Segmentation**

Image 1
M=2



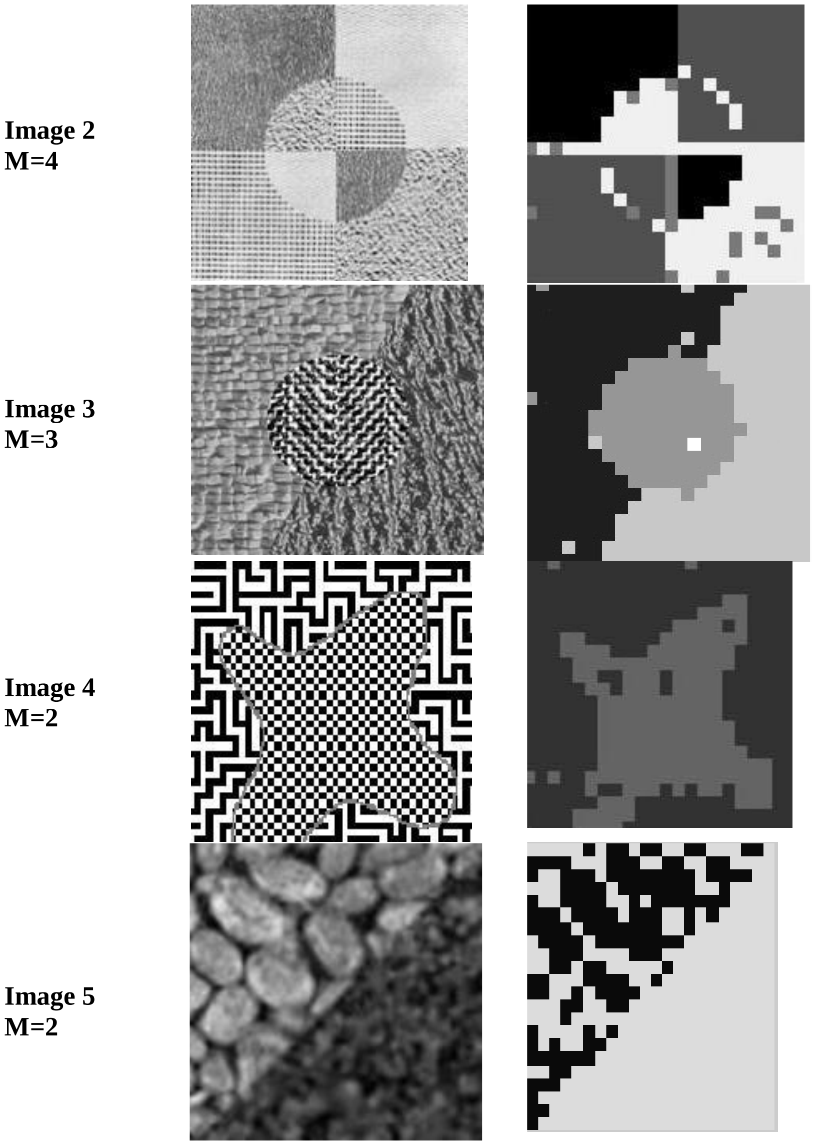


Figure 2.5: Results derived on application of the proposed model using hierarchical clustering algorithm

Performance Evaluation Of The Algorithm

After conducting experiments in the PYTHON environment, the generated model is evaluated for performance. It will be helpful to utilize both quantitative and qualitative measures while evaluating the performance. This section of the thesis utilises a quantitative method to benchmark the performance measurements using approaches like the Probabilistic Rand Index (PRI), the Variation of Information (VOI), and the Global Consistency Error (GCE).

Unni krishnan developed the Rand index as a metric. R et al. (2005) determined the number of pixels in segments where the labeling is consistent between the segment in question and the ground truth. Unnikrishnan R. et al. (2007) suggested a probabilistic rand index (PRI) that is based on this measure.

Using the picture data set, extract a batch of manually segmented images. (ground truth) say {S_1, S_2, …,S_K} against the set original image say X={x_1, x_2, . . . x_i, . . . , x_N}, for each N pixels, where an index denoted by a subscript. Let S_{test} choose the segmentation result to compare with the human-labeled set. Mark the location of the tagx_i by $l_i^{S_{test}}$ in segmentation S_{test} and by $l_i^{S_k}$ in the human segmented image S_k. It is supposed that every label $l_i^{S_k}$ can acquire values in a distinct set of size L_k, and thereby $l_i^{S_{test}}$ acquires one of L_{test} values. Our preferred approach is to use an unknown underlying distribution to simulate label connections for each pair of pixels. It is reasonable to presume that all segmenters contribute data regarding the segmentation S_k of the image in the form of binary numbers $\mathrm{II}(l_i^{S_k} = l_j^{S_k})$ for every set of pixels (x$_i$,x$_j$). Every perceptually correct segmentation is a random variable with an expected value, since they all follow a Bernoulli distribution over this number, p_{ij}. Therefore, the set {p$_{ij}$} for all disordered pairs (*i,j*) defines a probabilistic model of correct segmentations for the image *X*. The probabilistic rand (PRI) index is given by

$$PRI(S_{test}, \{S_k\}) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i<j}} \left[\mathrm{II}(l_i^{S_k} = l_j^{S_k}) p_{ij} + \mathrm{II}(l_i^{S_k} \neq l_j^{S_k})(1-p_{ij}) \right] \quad \text{--- (2.55)}$$

Let c_{ij} denote the event of a pair of pixels *i andj* having the same label in the test image S_{test}: $c_{ij} = \mathrm{II}(l_i^{S_k} = l_j^{S_k})$ Then the PR index can be written as:

$$PRI(S_{test}, \{S_k\}) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i<j}} \left[c_{ij} p_{ij} + (1-c_{ij})(1-p_{ij}) \right] \quad \text{--- (2.56)}$$

The PRI ranges from 0 *to* 1, where 0 signifies S_{test} and {*S$_1$, S$_2$.,S$_K$*} have dissimilarities and 1 indicate that all segmentations are identical.

As $c_{ij} \in \{0,1\}$ the above equation can be rewritten as

$$PRI(S_{test}, \{S_k\}) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i<j}} \left[p_{ij}^{c_{ij}} (1-p_{ij})^{(1-c_{ij})} \right] \quad \text{--- (2.57)}$$

The relationship between a pixel and its cluster is the basis of the variation of information (VOI) measure, which was proposed by Meila. M. (2007). It uses entropy and metrics based on shared information to measure distance between two sets of clusterings distributed across the data. It shows how much valuable information is lost or gained when the clusters are swapped. Think about the set S, which has been divided into groups, *D* say {*R$_1$ .R$_2$*, R$_3$,…R$_M$ } with the condition that $R_k \bigcap R_l = 0$ and $\bigcup_{k=1}^{M} R_k = S$. Pretend that there are an equal amount of data points in cluster D R_k be n and n$_k$ respectively, where $n = \sum_{k=1}^{M} n_k$. Let the number of pixels among the intersection of clusters R_k of S and $R_k^{'}$ of $S^{'}$ be denoted by $n_{kk^{'}}$, where $n_{kk^{'}} = | R_k \bigcap R_k^{'} |$.

d_{VOI} (Variation of information): $d_{VOI}(S, S^{'}) = H(S) + H(S^{'}) - 2I(S, S^{'})$ --- **(2.58)**

where, H and I symbolize the entropies and the shared information between the two clustering's.

$$H(S) = -\sum_{k=1}^{M} \frac{n_k}{n} \log \frac{n_k}{n}, \quad \text{--- (2.59)}$$

$$I(S, S^{'}) = \sum_{k=1}^{M} \sum_{k^{'}=1}^{N} \frac{n_{k,k^{'}}}{n} \log \frac{n_{k,k^{'}}}{n} \frac{n_k}{n} \frac{n_{k^{'}}^{'}}{n} \quad \text{--- (2.60)}$$

To find out how much one segmentation map is like a modification of segmentation, Martin. D. et al. (2001) came up with a measure called global consistency error (GCE). In a perfect match, each area in one segmentation would be identical to, or slightly different from, an area in the other segmentation. Think about the parts of S1 and S2 that include the pixel. The segments consist of sets of data for each pixel. The local error will be 0 unless the two segments are appropriate subsets of each other. If there is no relationship between subsets, then inconsistency could arise from overlapping regions, and the local error would be less than zero. Perhaps the local alteration error can be defined as

$$E(S_1, S_2, p_i) = \frac{| R(S_1, p_i) \setminus R(S_2, p_i) |}{R(S_1, p_i)} \quad \text{--- (2.61)}$$

where \ denote set difference, and |x| is the cardinality of set x.

Taking into consideration the image's pixel count, the Global Consistency Error (GCE) can be described as

$$GCE(S_1, S_2) = \frac{1}{n}\min\left\{\sum_{\mathrm{i}} \mathrm{E}(S_1, S_2, \mathrm{p_i}), \sum_{\mathrm{i}} \mathrm{E}(S_1, S_2, \mathrm{p_i})\right\} \quad \text{--- (2.62)}$$

Using k-means and hierarchical clustering in the Multivariate New Symmetric Mixture model and VOI as segmentation performance measures, we computed and analyzed the four texture pictures, and Gaussian mixture model.

Table 2.1: Evaluation of Segmentation Process of Textured Images

Evaluation of Segmentation Process				
Description	**Model**	**PRI**	**GCE**	**VOI**
Optimal Values		**1**	**0**	**As high as possible**
Textured Image 1	GMM-K	0.68	0.15	0.81
	GMM-H	0.85	0.14	0.71
	MNSMM -K	0.90	0.10	0.55
	MNSMM -H	0.92	0.05	1.00
Textured Image 2	GMM-K	0.47	0.50	0.78
	GMM-H	0.50	0.40	1.76
	MNSMM -K	0.51	0.40	1.79
	MNSMM -H	0.54	0.15	2.31
Textured Image 3	GMM-K	0.34	0.31	1.25
	GMM-H	0.46	0.29	1.23
	MNSMM -K	0.56	0.27	1.32
	MNSMM -H	0.58	0.28	1.51
Textured Image 4	GMM-K	0.75	0.19	0.54
	GMM-H	0.83	0.15	0.71
	MNSMM -K	0.84	0.14	0.70
	MNSMM -H	0.92	0.13	1.54
Textured Image 5	GMM-K	0.581	0.158	1.121
	GMM-H	0.684	0.151	1.139
	MNSMM -K	0.811	0.168	0.776
	MNSMM -H	0.948	0.142	1.346

Image 1

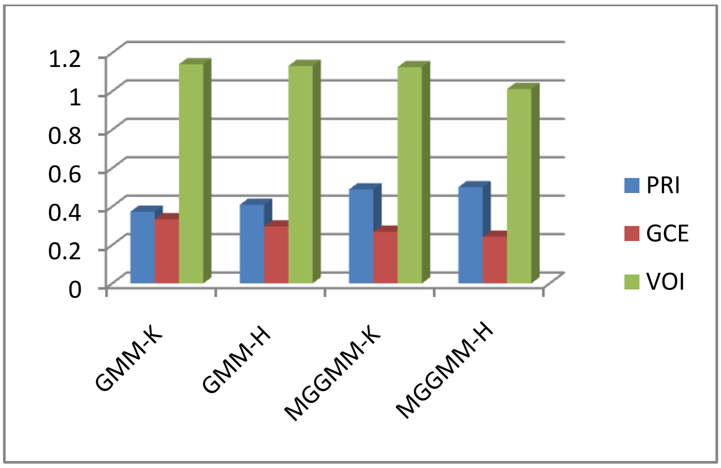


Image 2

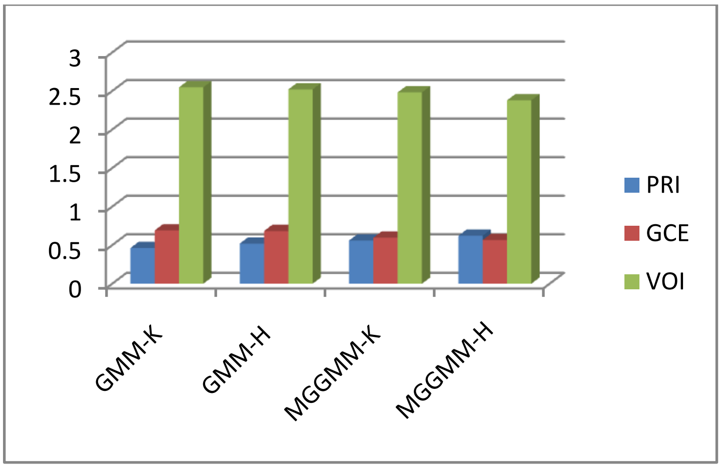


Image 3

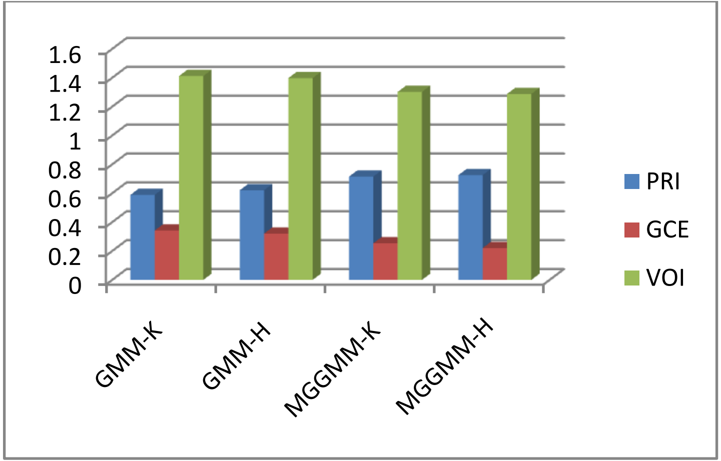


Image 4

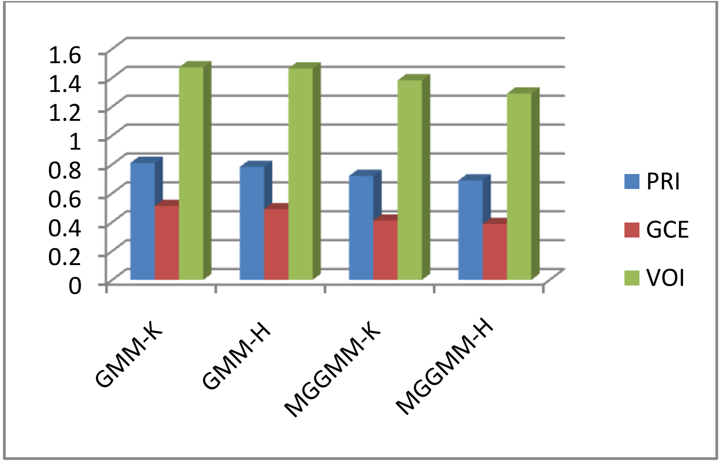


Image 5

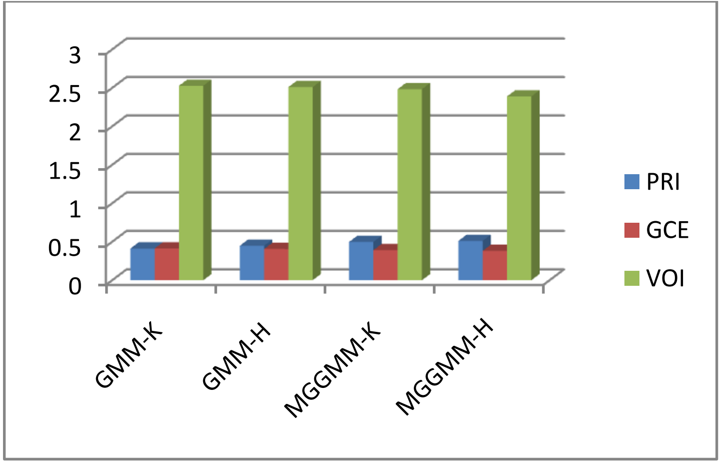


Figure 2.6: Graph Plot of Segmentation Metricsfor Different Textured Images

The graph in figure 2.6 and the data reported in table 2.1 are the outputs of the MNSMM-based model that was created. When compared to models already published in the literature, the results shown here demonstrate a substantial improvement.

Comparative Study

Performance evaluation measures are used to ensure that the results obtained from the suggested segmentation method are accurate. The findings are evaluated in relation to the misclassification rate, and the model is contrasted with an existing GMM-based model. The rate of misclassification is the amount of pixels that are incorrectly assigned to certain segments. According to Huang, Q., et al. (1995), a lower misclassification rate indicates better accuracy. Table 2.2 displays the outcomes obtained from various clustering methods, such as hierarchical clustering and k-Means, when tested against the misclassification rate. The suggested mixed model and the existing model based on GMM are compared in terms of the percentage of misclassification among the picture pixels used as a sample.

Table 2.2: Classifier Accuracy of the Model

Model	**Classifier Accuracy**
GMM with K	24%
GMM with H	22%
MNSMM with K	19%
MNSMM with H	17%

In comparison to GMM, the misclassification rate using MNSMM is lower, as seen in table 2.2. Using the confusion matrix for segmented image regions, as suggested by Martin et al. (2001), the segmentation quality metrics are calculated for the sample images.

The four qualities that form the basis of the confusion matrix include True Positive (TP), which counts the number of positively labelled pixels, and False Positive (FP), which counts the number of pixels that have been mistakenly tagged as positive. Finally, False Negative (FN) indicates a set of pixels that are mistakenly labeled as negative, whereas True Negative (TN) indicates pixels that correspond to negatives and are accurately labeled as negative. You can see the confusion matrix down below.

Table 2.3: Confusion Matrix

Label	**Actual Positive**	**Actual Negative**
Predicted Positive	TP	FP
Predicted Negative	FN	TN

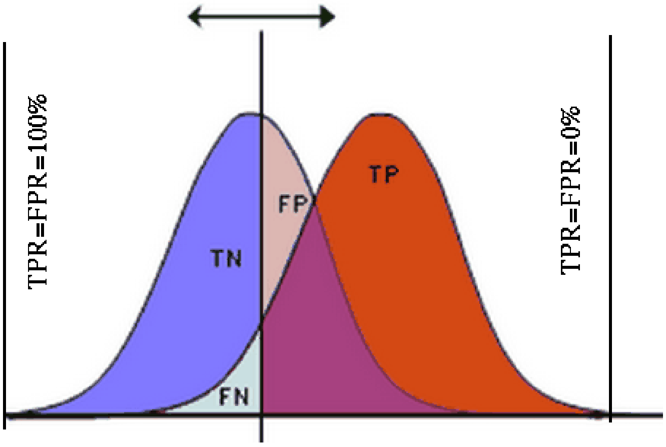


Figure 2.7: Visual interpretation of the Confusion matrix

Using the equations provided by Davis et al. (2006), the values of Accuracy, Sensitivity (TPR), 1-Specificity (FPR), Precision, Recall, and F-measure are computed from the confusion matrix.

TPR/Sensitivity = $\frac{TP}{TP+FN}$

FPR= $\frac{FP}{FP+TN}$

Equivalent to (1-Specifictry)

Recall= $\frac{TP}{TP+FN}$

Precision= $\frac{TP}{TP+FP}$

F-Measure= $\frac{2*\mathrm{Pr}ecison*\mathrm{Re}call}{\mathrm{Pr}ecison+\mathrm{Re}call}$

Accuracy= $\frac{TP+TN}{TP+FP+TN+FN}$

Using the above formulae, the metrics are calculated and shown in the table BELOW

Table 2.4: Comparative study of MNSMM versus GMM w.r.t. k- Means and Hierarchical based clustering algorithm

Quality Metrics							
Description	**Model**	**accuracy**	**sensitivity**	**specificity**	**precision**	**recall**	**F_measure**
Optimal Values		**1**	**1**	**1**	**1**	**1**	**1**
Textured Image 1	GMM-K	0.09	0.14	0.04	0.14	0.14	0.18
	GMM-H	0.08	0.16	0.05	0.13	0.16	0.25
	MNSMM - K	0.05	0.10	0.08	0.08	0.10	0.59
	MNSMM - H	0.58	0.59	0.10	0.99	0.59	0.74
Textured Image 2	GMM-K	0.48	0.58	0.14	0.48	0.58	0.48
	GMM-H	0.45	0.61	0.17	0.56	0.61	0.58
	MNSMM - K	0.44	0.60	0.15	0.56	0.60	0.58
	MNSMM - H	0.64	0.68	0.22	0.90	0.68	0.78
Textured Image 3	GMM-K	0.29	0.81	0.18			
	GMM-H	0.32	0.83	0.20	0.19	0.93	0.31
	MNSMM - K	0.33	0.83	0.21	0.19	0.93	0.31
	MNSMM - H	0.70	0.90	0.35	0.72	0.95	0.80
Textured Image 4	GMM-K	0.87	0.78	0.58	0.21	0.39	
	GMM-H	0.91	0.81	0.61	0.24	0.71	0.815
	MNSMM - K	0.91	0.81	0.68	0.35	0.78	0.87
	MNSMM - H	0.94	0.98	0.74	0.38	0.81	0.90
Textured Image 5	GMM-K	0.188	0.185	0.057	0.348	0.189	0.321
	GMM-H	0.196	0.288	0.067	0.369	0.288	0.323
	MNSMM - K	0.105	0.280	0.040	0.402	0.298	0.348
	MNSMM - H	0.027	0.309	0.012	0.493	0.309	0.364

In comparison to the Multivariate New Symmetric Mixture Model, the suggested classifier clearly outperforms it, as shown in table 2.4.

CONCLUSION

In the realm of digital image processing, image classification presents significant challenges, especially in high-level applications like medical imaging, robotics, and automation, where precise object detection is critical. The segmentation process serves as a foundational step, using feature vectors to identify and isolate regions of interest based on predefined criteria. Among these features, texture plays a pivotal role, as it captures the spatial relationships between pixels and provides essential insights into the structural patterns within an image. The primary goal of texture segmentation is to divide images into homogeneous regions that carry meaningful information, facilitating more accurate classification and analysis. This approach enhances the model's ability to interpret diverse images, making it an indispensable tool for real-world applications in fields requiring high precision.

REFERENCES

1. AmadasunM.,KingR.,(1989),"Textural features corresponding to textural properties," IEEE Transactions on Systems, Man and Cybernetics, Vol. 19(5), pp.1264-1274.
2. JyothirmayiT.,“Studies on Image Segmentation Integrating Generalized Laplace Mixture Model and Hierarchichal Clustering”, “International Journal of Computer Applications”, Vol 128, No 12 pp 7-13, 2015.
3. Haralick, R.M., (1979),–Statistical and Structural Approaches to Texture, ‖ Proceedings of the IEEE Computer Society, Vol.67, pp. 786-804.
4. Lu C., Chung P. and ChenC. (1997), –Unsupervised texture segmentation via wavelet Transform‖, IEEE Trans. on Pattern Recognition,Vol.30(5),pp.729-742.
5. S. Krishnamachariand R. Chellappa (1997),–Multiresolution Gauss-Markov Random field models for Texture segmentation ‖,IEEE Trans. Image Processing,Vol.2,pp.171-179.
6. Kostas Haris, Serafim N. Efstratiadis, Nicos Maglaveras, and Aggelos K.Katsaggelos(1998), –Hybrid Image Segmentation Using Water sheds and Fast Region Merging ‖, IEEE Transactions on Image Processing,Vol.7(12).
7. Y.Zhang, M.Brady, and S.Smith,(2001),–Segmentation of Brain MRI Images througha Hidden Markov Random Field model and E Malgorithm‖, IEEE Trans.Med.Imaging.,Vol.20(1),pp.45-57.
8. Vaijinath V. Bhosle, Vrushsen P.Pawar (2013), "Texture Segmentation: Different Methods", International Journal of Soft Computing and Engineering,Vol.3(5),pp.69-74.
9. Pal S.K and Pal N.R. (1993), –A review on Image Segmentation Techniques ‖,IEEE Trans. on Pattern Recognition,Volume26(9),pp.1277-1294.
10. Haim Permuteret al. (2006), –A study of Gaussian mixture models of color and texture features for image classification and segmentation ‖,The Journal of the Pattern Recognition Society, Vol.39,pp.695-706.
11. M.N.Thanh,Q.M.J.Wu(2011),–Gaussian mixture model based spatial neighbourhood relationships for pixel a beling problem‖, IEEE Trans. Syst. Man Cybern.,Vol.99,pp.1-10.
12. PaulD.,McNicholas(2011),"OnModel-BasedClustering,Classification,andDiscriminantAnalysis",Journal of Iranian Statistical Society, Journal of IranianStatisticalSociety
13. G.V.S.RajKumar,K.SrinivasaRaoandP.SrinivasaRao(2011),–ImagesegmentationandRetrievalsbasedon finite doubly truncated bivariate Gaussain mixture modelandK-means‖, International Journal of Computer Applications, Volume25(5), pp.5-13.
14. Yu-Len Huang (2005), –A fast method for textural analysisofDCT-based image‖, Journal of Information Science and Engineering, Volume21,pp.181-194.
15. SrinivasY.and SrinivasaRao K.(2007), –Unsupervised image segmentation using finite doublytruncated Gaussian mixture model and Hierarchical clustering ‖,Journal ofCurrentScience,Vol.93(4),pp.507-514
16. P.Brodatz,(1966)—Texture:aphotographicalbumforartistsanddesigners,‖,Dover,NewYork,USA(http.://sipi .usc.edu/database/database.php?volume=textures).
17. M.S.Alliliand Nizar Bougila(2008), –Finite generalized Gaussian mixture modeling and applications to image andvideoforegroundsegmentation‖,JournalofElectronicImaging,Volume17(13),pp.05-13
18. C. Lu, P. Chung and C. Chen(1997), –Unsupervised TextureSegmentationviaWaveletTransform‖,IEEETrans.onPatternRecognition,Volume30(5),pp.729-742
19. T. P. Weldon,W.E. Higgins(1996),–Design ofmultiple Gabor filters for texture segmentation‖, Proc. of Intl. Conf.onAcoustics,SpeechandSignalProcessing,pp.2243-2246.
20. RaoK.R.andYipP.(1990),–DiscreteCosineTransform– Algorithms, Advantages, Applications‖, Academic press,NewYork, USA
21. Gonzales R.C., Woods R.E.,(1992), –Digital Image Processing‖, Addison–Wesley
22. Mclanchlan G. and Peel D.(2000), — The EM Algorithmfor Parameter Estimations‖, John Wileyand Sons,NewYork, USA

23. ShaoquanYU, Anyi Zhang, Hongwei LI (2012), ―A Review on estimating the Shape Parameters of Generalized Gaussian Distribution ‖,Journal of Information Systems,Volume8(21),pp.9055-9064.
24. Jeff A.Bilmes(1977), ―A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation forGaussianMixtureandHiddenMarkovModels‖,Intl.ComputerScienceInstitute,Berkeley.
25. Unnikrishnan R.,Pantofaru C.,and Hernbert M.(2007),―Toward objective evaluation of image segmentation algorithms,‖ IEEE Trans.Pattern Annl.&Mach.Intell,Vol.29(6),pp.929-944.
26. M.Meila(2007),―Comparing clusterings-an information based distance‖, Journal of Multivariate Analysis, Vol. 98,pp.873–895
27. D. Martin, C. Fowlkes, D. Tal and J. Malik (2001), "ADatabaseof HumanSegmentedNatural ImagesanditsApplication to Evaluating Segmentation Algorithms and Measuring Ecological Statistics", Proc.8 thInt'lConf.Computer Vision,Vol.2,pp.416-423
28. Powers,DavidM.W.(2011),―Evaluation:From Precision, Recall and F-Measure to ROC, Informedness, Markednes sand Correlation‖,JournalofMachineLearningTechnologies,Vol.2(1),pp.37-63.
29. Poornima, R., Nagavarapu, S., Navva, S., Katkoori, A. K., Mohsen, K. S., & Saikumar, K. (2024, July). Multi-Modal Meta Multi-Task Learning for Social Media Rumor Detection. In 2024 2nd World Conference on Communication & Computing (WCONF) (pp. 1-6). IEEE.
30. Swarnalatha, T., Supraja, B., Akula, A., Alubady, R., Saikumar, K., &Prasadareddy, P. (2024, July). Simplified Framework for Diagnosis Brain Disease Using Functional Connectivity. In 2024 2nd World Conference on Communication & Computing (WCONF) (pp. 01-06). IEEE.
31. Reddy, D. S., Charan, K. S., Kayam, S. K., & Rajalakshmi, P. (2024, March). Robust Obstacle Detection and Collision Warning for Autonomous Vehicles Using Autoware Universe. In 2024 16th International Conference on Computer and Automation Engineering (ICCAE) (pp. 378-384). IEEE.
32. Kolli, S., Hanuman, A. S., Ratnam, J. V., Naidu, J. J., Shankar, D., & Saikumar, K. (2024). Generous Information Safety System for Investors in Online Trading Secretly using KP-ABE Machine Learning Method. International Journal of Intelligent Systems and Applications in Engineering, 12(7s), 285-297.
33. Kumar, G. N. K., Reddy, M. S., Malleswari, D. N., Rao, K. M., & Saikumar, K. (2024). A Real-Time Hadoop Bigdata Maintenance Model using A Software-Defined and U-Net Deep Learning Mode. International Journal of Intelligent Systems and Applications in Engineering, 12(7s), 364-376.
34. Saikumar, K., & Rajesh, V. (2024). A machine intelligence technique for predicting cardiovascular disease (CVD) using Radiology Dataset. International Journal of System Assurance Engineering and Management, 15(1), 135-151.
35. Kolli, S., Hanuman, A. S., Ratnam, J. V., Naidu, J. J., Shankar, D., & Saikumar, K. (2024). Generous Information Safety System for Investors in Online Trading Secretly using KP-ABE Machine Learning Method. International Journal of Intelligent Systems and Applications in Engineering, 12(7s), 285-297.