

Fully Unsupervised Texture Segmentation Using Mean-Shift Clustering With Modified Gray Level Co-Occurrence Matrix

Fatna Zegaoua¹, Abderrazak Guessoum²

¹Electronics Department Faculty Of Technology Blida-1 University Blida, 09000, Algeria.

²Computer Science Department LATSI Laboratory, Faculty Of Technology Blida-1 University Blida, 09000, Algeria.

Abstract: Digital image processing applies a number of computer algorithms to treat and deal with digital images. The outcomes of this process can be either images or a set of representative properties to be used in subsequent applications. In view of the diversity of images and the applications goals, it has been proved in a large literature that using hybrid techniques is more advantageous than relying on a single method. The purpose of this paper is to present an unsupervised cooperative segmentation framework that can shorten the computation cost and segment diverse image types without any prior information. The proposed framework generates, at first, a set of feature vectors using a new Modified Gray Level Co-occurrence Matrix (M-GLCM). Then, it applies the Mean-Shift clustering to estimate the number of textures in the original image and initiate afterward the k-Means algorithm in order to assign each pixel to its corresponding class according to a distance measure. The new proposed M-GLCM helps in overcoming the time consuming and the large memory size needed in the initial GLCM. The Mean-shift by its own is known to be fast and induces good estimated clusters. The core aspect of our approach is its adaptability to the image's type where the number of calculated attributes and the window size depends on the global uniformity characteristic of the image; more the image is textured, more the window size is larger and the features space is denser. The results of the proposed work are found to be promising, especially when a step of multi-level thresholding is applied beforehand.

Keywords: Texture segmentation, Clustering, Classification, Mean-Shift, GLCM.

I. INTRODUCTION

Digital Image Processing has become an integral part of various applications across multiple domains, including medical imaging, remote sensing, and computer vision [1][2]. The fundamental technique that aims to obtain multiple and distinct regions within an image, according to a particular characteristic, whether a gray/color tone, a texture or other criterion, is known as "Image Segmentation". This method encompasses various approaches such as region growing, edge detection and machine learning [3][4][5].

As for Texture Analysis, even if it is an extensively studied research topic and the texture aspect is a widespread phenomenon that is easy to recognize, it still lacks an universal or unique definition. The most widely accepted definition describes texture as the spatial variation relationship of the pixels brightness intensity [2]. Identifying particular textures within an image is mainly accomplished by representing texture as a variation in two-dimensional gray levels. The relative brightness of pixels pairs is interpreted as degree of contrast, regularity, coarseness and directionality.

All the methods in this field are classified into four groups: structural, statistical, transform and model-based methods [6][7][8]. While structural methods consider texture as a combination of initial patterns, model-based texture analysis creates a model of the image and defines its parameters to get the basic properties of the texture. As for the transform methods, the image is converted into another space through a particular function, such as Fourier, Gabor, and wavelet transforms. Finally, as regards the

statistical approaches, a series of statistical calculations on the lightness intensity distribution functions of pixels is carried out.

Under these groups, existing techniques for modeling texture, including cooccurrence statistics [9][10], filter banks [11][12], random fields [13][14] and wavelet transforms [15], have gained popularity due to their ability to represent texture information efficiently. However, these methods often require careful parameter tuning, particularly when it comes to the analysis of real images that are rich in detail and encompass a diverse range of uniform and/or textured regions. Recent advancements in texture analysis have focused on developing and selecting discriminative descriptors that can capture its essential characteristics while minimizing computational costs and improving applications performance.

Consequently, from the diversity of image types and the specific requirements of each application arise the quest to overcome the inherent shortcomings of using one image segmentation method. As an efficient alternative, combining diverse techniques in a parallel or sequential structure is increasingly becoming the core of new research frameworks, thereby producing more robust and accurate results [16].

In Many statistical methods, texture features are extracted by analyzing the spatial distribution of pixel gray values. One example is the Gray Level Cooccurrence Matrix "GLCM", proposed by Haralick and al. back in 1973 [17]. Furthermore, clustering is one of the segmentation techniques that aims to organize data, usually represented as a multidimensional measurement vectors, into clusters based on a resemblance criterion [18][19]. Thus, under this category, we found the k-Means algorithm [20] which calls for a known number of classes before assigning the pixels to their clusters. Similarly, the Mean-shift algorithm is an unsupervised clustering algorithm that attempts to assign iteratively the data points to the clusters by shifting them towards highest density centers [21] without a prior knowledge of the classes number.

Endeavoring to develop an approach that can be simple, efficient, fast and don't need a prior information, we opted for an hybrid unsupervised framework that combines a new modified GLCM with a mode-seeking algorithm based on the Mean-Shift clustering algorithm, applying it on different types of images ranging from moderately-textured to highly textured images. The advantage of the proposed work lies in its ability to adjust the features space according to the image context, more the image is textured, more the number of features is higher, thereby optimizing the computation time and the allocated memory size.

II. RELATED WORK

Clustering has been a prominent implemented technique for texture segmentation domain. Besides, there are already a large number of texture segmentation algorithms in the literature that integrates a features extraction technique with a suitable classification algorithm. In our work, all the selected techniques have been previously exploited in multiple papers, albeit in different manners.

With regards to the GLCM, it was used to analyze Electroencephalography (EEG) images [22]. The proposed process generates an EEG signal frequency representation that can be employed in texture analysis, using the GLCM Matrix in combination with the Principal Components Analysis (PCA) to identify sleep disorder breathing. In [23], the GLCM was applied to the vibration signal recorded in images. A grouping of PCA and Sequential Features Extraction (SFE) methods has been used to obtain the most pertinent features. The proposed approach has been tested using a multiclass-Naive Bayes classifier.

Likewise, the GLCM is also applied in conjunction with k-means classification using particular sets of features [24][25]. Yet, in the biomedical field, GLCM has been used, in another perspective, to distinguish benign and malignant breast lesions [26].

In [27], a method based on Active Contours Without Edges (ACWE) is applied fusing the GLCM and Gabor features together to create the final feature space.

When comparing the performance of GLCM methods to the semi-variogram and the Fourier spectra in various classification contexts, the results showed that the GLCM method shows better results for simple situations where the textures are visually easily separable [28].

As for the Mean-Shift technique, a modified discrete wavelet frames (DWF) decomposition was used to extract important features from an image before a mean-shift algorithm is applied together with a fuzzy c-means (FCM) clustering to segment the image into different texture regions [29].

In [30], an automatic brain tumor detection technique in noise corrupted images is presented. Once the noise is removed from the image, it is segmented using mean shift clustering. The segmented parts are sent to the GLCM for feature extraction. The features are used by multi class SVM to detect the tumor in the images. The step followed extracts the tumor with increased precision in noisy images.

The article in [31], explains a proposed advance texture extraction using multi-patch images pixel method with sliding windows that minimize loss of information in each pixel patch. The authors use GLCM texture feature with a Mean-shift filter. The proposed method is then trained using Deep Neural Networks (DNN) and compared to other classification techniques for benchmarking.

III. THEORETICAL BACKGROUND

We present in the following subsections the adopted means and methods to segment texture images, namely, the Gray Level Cooccurrence Matrix and its texture features, the Mean-shift mode seeking technique and the segmentation evaluation criteria.

A. Window Size Selection

An important element in texture segmentation is the window size. Extensive experiments have been carried in the last decade to find the optimum size. In [32], Marceau et al. demonstrated that, for classifications incorporating texture, 90% of the variability in the accuracy is accounted for by the size of the texture window, compared to only 10% for the particular texture algorithm.

Also, it has been confirmed that a minimum size window was needed for accurate cognitive (i.e., visual) classification, and that the window size needed increased as spatial resolution increased [33]. Accuracy tended to increase until a threshold was reached, after which no further advantage was gained from larger windows.

No unique conclusion has been reached about the optimum window size in texture calculation. It has been suggested that small window sizes are a better choice than larger windows, due to the contaminating effect of edge pixels in classification [34].

As recent researches have shown that using multiple windows with different size is more powerful and can improve significantly the segmentation results [35], we have opted for a different window sizes depending on the concluded image type, starting from a small window for uniform image to a larger windows as texture aspect increases within the image.

B. Feature Selection and Extraction

The ability to analyze, segment and interpret images using computational algorithms is essential for extracting meaningful information from raw data. Several feature-based models have been proposed to partition an image into homogeneous and separate regions of different textures by working on features derived by different strategies. Yet, it has been confirmed that having numerous features doesn't yield necessarily to robust results, it may instead, cause significant computational load without enhancing applications performance.

Assessing features discrimination ability and choosing the most informative subsets to enhance models accuracy is an essential step. This process is known as "Feature selection". It mainly helps in reducing time computational cost by avoiding unnecessary or irrelevant features that can affect negatively the approaches performance [36].

In feature selection, some researches were carried to evaluate the effectiveness of different combinations and help in outlining the best ones. In [37], the Information Gain has been introduced to preliminarily eliminate irrelevant features in high dimensional datasets and reduce the time cost of the designed algorithm. Combining Correlation, Energy, and Homogeneity features has been confirmed to emerge as best choice for real-time applications which need high accuracy and low complexity [38].

When it comes to features extraction [6], texture attributes are computed using first and second order statistics in statistical approaches. However, in structural approaches, extraction methods look on how to generate texture patterns. For other techniques, the textured image is transformed into the frequency domain in which the extraction of texture features can be done by analyzing the power spectrum. After all, new texture descriptors are continuously developed.

B.1. First-order Features

First-order statistics [39] are Gray-level histogram-based features derived directly from the original image by using pixels values, whether calculated on the global image or within a precise window. They assess the approximate probability density of occurrence of each intensity level.

Considering a grayscale image of dimension $N \times M$ as a function with two variables on the Gray scale domain D , the function f , which takes its values in the discrete set Z of NG elements, is written as follows:

$$p: f(i, j) \in D \rightarrow Z \quad (1)$$

where $f(i, j)$ are the intensities of the image pixels and $Z = \{0, 1, \dots, NG - 1\}$, NG is the total number of image intensities ranging from 0 to 255.

The histogram function is then, given by the following equation:

$$H(k) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \delta(f(i, j), k) \quad (2)$$

where $\delta(n, m)$ is the Kronecher delta function:

$$\delta(n, m) = \begin{cases} 1 & \text{if } n = m \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

When dividing the values of $h(k)$ by the total number of pixels in the image, we obtain the occurrence probability density of the intensity levels $p(k)$:

$$p(k) = \frac{H(k)}{N * M}, \quad k = \{0, 1, \dots, NG - 1\} \quad (4)$$

Various parameters can be derived from the histogram to quantitatively characterize the first-order statistical attributes:

$$1. \text{ The Mean: } \mu = \sum_{k=0}^{NG-1} k * p(k) \quad (5)$$

$$2. \text{ The Variance: } \sigma^2 = \sum_{k=0}^{NG-1} (k - \mu)^2 * p(k) \quad (6)$$

$$3. \text{ The energy: } Enr = \sum_{k=0}^{NG-1} p(k)^2 \quad (7)$$

4. The Entropy: $Ent = -\sum_{k=0}^{NG-1} p(k) * \log_2(p(k))$ (8)

B.2. Second-order Features

Textures cannot completely characterized by first-order features. So, one of the texture outlining characters is the gray values spatial distribution [39][40]. The calculation of texture second-order gray level probability distribution involves assessing the gray levels of pixel in pairs at a time.

In the following paragraphs, we present a long-familiar statistical tool for extracting second-order texture information proposed by Haralick and al. back in 1973 [41][42].

C. Standard Gray Level Cooccurrence Matrix "GLCM"

Similarly to the autocorrelation function, the gray-level cooccurrence matrix is extensively used in texture analysis contributing to the perception of texture. It is a statistical method that captures the spatial relationship between pixel intensities in an image. A large number of researches [43][44] corroborates that the GLCM is an easy method to implement and a very powerful tool for texture segmentation. However, the computational demands of GLCM, particularly in terms of processing time and memory consumption, become significant bottlenecks for high-resolution imagery and real-time applications, especially in scenarios where large volumes of data are involved, causing impractical implementations for pixel-by-pixel image processing.

In Equation 9, $M_{d,\theta}(i,j)$ translates the appearance frequency of the gray levels i and j separated by distance d according to the orientation θ . The adjacency can be defined to take place in each of the chosen directions.

$$M_{d,\theta}(i,j) = \text{card}\{p(x,y) = i, p'(x,y) = j\} \text{ with } |p,p'| = d, \arg(p,p') = \theta \quad (9)$$

The GLCM matrix could be constructed by taking into account all the gray scale, which gave us an 256×256 matrix. Yet, in practice, this size corresponds to the maximum value of gray levels in the image increased by one (to take the zero value into account). The following example demonstrates the calculation manner.

C.1. Illustration

We assume that our image I is given as follows :

$$I = \begin{bmatrix} 0 & 0 & 2 & 0 \\ 2 & 2 & 7 & 4 \\ 7 & 4 & 4 & 0 \end{bmatrix}$$

Since image A has 8 values of gray-levels, its corresponding cooccurrence matrix will be an 8×8 matrix. The calculated GLCM depends on the chosen distance and orientation. The conventional orientations for the four offsets at distance of 1 from the pixel of interest are presented in "Fig. 1":

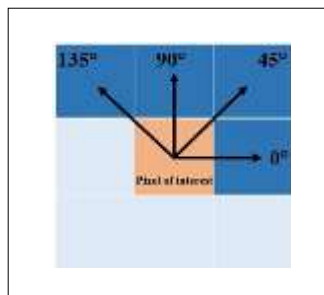


Fig. 1 Offset between the pixel of interest and its neighbors

For $d = 1$ and direction $\theta = 135^\circ$, the resulting GLCM matrix $M_{1,135}(i,j)$ is as follows:

$$GLCM_{1,135^\circ}(i,j) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

C.2. GLCM Features

Since image I has 8 values of gray-levels, its corresponding cooccurrence matrix will be an 8×8 matrix. The calculated GLCM depends on the chosen distance and orientation. The conventional orientations for the four offsets at distance of 1 from the pixel of interest are presented in "Fig. 1".

$$1. \text{ The Energy: } E = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} M_{d,\theta}(i,j)^2 \quad (10)$$

$$2. \text{ The Contrast: } C = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} |(i-j)|^m * M_{d,\theta}(i,j)^n \quad (11)$$

$$3. \text{ The Homogeneity: } H = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} \frac{1}{1+(i-j)^2} M_{d,\theta}(i,j) \quad (12)$$

$$4. \text{ The Entropy: } Ent = \sum_{i=0}^{NG-1} \sum_{j=0}^{NG-1} M_{d,\theta}(i,j) * \log(M_{d,\theta}(i,j)) \quad (13)$$

D. Clustering Algorithms

D.1. Mean-Shift Algorithm

As a robust clustering algorithm utilized in unsupervised learning [21][46][47], Mean-Shift operates without any assumptions, wherefore it is classified as a non-parametric technique. It is especially useful for datasets containing clusters without linear boundaries and shaped arbitrarily. Processing Mean-shift is appropriate in over segmentation, multiple segmentation, tracking and clustering applications [8]. One use of this density-based clustering algorithm is the determination of the features space modes. Its key concept involves moving each data point, within a specific radius, towards the highest density of the points distribution. The algorithm iteratively executes shifts of the data points until they converge to its density local maxima.

Assuming that $\{X_i\}_{i=1,\dots,n}$ are the data-points or the feature vectors with dimension d , the density calculated at each point is given as follows:

$$f_K(x) = \frac{1}{n \cdot h^d} \cdot \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (14)$$

where K is the kernel function, h a bandwidth parameter which represent the radius of a circle centered on x . In [46], the authors detailed how to obtain the so-called "Mean-Shift Vector".

$$m_{h,G}(x) = \left[\frac{x_i \cdot g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x \right] \quad (15)$$

As a result, if we consider $\{y_i\}_{i=1,2,\dots}$; as the successive locations of the Kernel G , we get:

$$y_{i+1} = \left[\frac{x_i \cdot g\left(\left\|\frac{y_i-x_i}{h}\right\|^2\right)}{g\left(\left\|\frac{y_i-x_i}{h}\right\|^2\right)} \right] \quad (16)$$

so the modes are the point-data resulted from the equation $(y_{i+1} - y_i = 0)$.

The pseudo code below explains how the Mean-Shift algorithm works:

- Step 1: Start from random points and calculate the means of data points inside windows centered at each centroid;
- Step 2: Shift the center of each window to the calculated corresponding mean;
- Step 3: Repeat Step 2 untill converging or meeting a specific error limit;

- Step 4: Merge resulting centroids separated by less than the density bandwidth.

D.2. K-Means Algorithm

One of the most popular unsupervised classification algorithms in data science and machine learning is the K-Means technique [48][49], which divides an unlabeled dataset into K distinct and non-overlapping clusters according to a specific similarity criterion.

Given a set of d-dimensional objects $(x_1, x_2, \dots, x_n)$, the K-Means clustering partition this data cloud into K sets $S = \{S_1, S_2, \dots, S_k\}$ with $K \leq n$, in which each object belongs to the cluster with the nearest mean. The K-Means aims to minimize the within-cluster sum of squares by finding a local minimum of the following cost function:

$$J = \sum_{j=1}^k \sum_{i=1}^n \|X_i^{(j)} - C_j\| \quad (17)$$

where C_j is the centroid (mean) of the points in the set S_i .

Equation (18) defines how the Euclidean distance between two vectors X and Y, having N components each, is computed:

$$\|d(X, Y)\| = \sqrt{\sum_{i=0}^{N-1} (X_i - Y_i)^2} \quad (18)$$

The K-Means computational efficiency makes it suitable for large datasets and big data applications. The pseudo code explaining how the k-Means algorithm works is as follows:

- Step 1: Initialize K centroids Randomly;
- Step 2: Assign each point to its nearest centroid based on a distance measurement, usually the quadratic or Euclidean distance;
- Step 3: Update the centroids by calculating the new clusters means taking into account the new categorization done in step 2;
- Step 4: Repeat Steps 2 and 3 until the centroids have no remarkable change or an error limit is reached.

E. Performance Evaluation Criteria

Among the existing evaluation criteria discussed in [50], and since we don't work with any ground truth images, we have selected the Levine and Nazif Uniformity Criterion [51] and the Zeboudj Contrast [52] to check and rate the performance of the proposed work. These criteria are considered as best measures tested to assess the segmentation quality of uniform and textured images.

Lets I be the image to be segmented, $s(i,j)$ its pixels and f a function which takes its values in subset Z (f can be the pixels gray levels), with A is the total number of pixels in the image and A_i the number of pixels in region R_i , μ_i and σ_i denotes respectively the mean and the standard deviation of f on R_i .

If R is the segmentation to be evaluated, so R can be seen as a partition of I into different regions R_i , $i = 1, \dots, N$; verifying the following conditions:

- $R_i \cap R_j = \emptyset$
- $\bigcup_{i=1}^N R_i = I$

E.1. Levine and Nazif uniformity criterion

It is a simple intra-region uniformity criterion based on the sum of the regions variances. It has small values.

$$C_{intra} = \sum_i \sum_{s \in R_i} \left[f(s) - \frac{1}{A_i} \cdot \sum_{s \in R_i} f(s) \right]^2 = \sum_i \frac{\sigma_i^2}{C} \quad (19)$$

with C a normalization coefficient equal to the maximum of the variances.

$$\sigma_{max}^2 = \frac{(f_{max} - f_{min})^2}{2} \quad (20)$$

E.2. Zeboudj contrast

This coefficient takes into account the region's internal and external contrast, calculated on pixels neighboring $W(s)$, according to the following equation:

$$C(s, t) = \frac{|f(s) - f(t)|}{L - 1} \quad (21)$$

where $C(s, t)$ is the contrast between two pixels s and t , with L is the maximum gray level.

- The internal contrast of region R_i will be calculated by the following formula:

$$I_i = \frac{1}{A_i} \cdot \sum_{s \in R_i} \max\{C(s, t), t \in W(s) \cap R_i\} \quad (22)$$

- The external contrast of region R_i will be calculated as follows:

$$E_i = \frac{1}{l_i} \cdot \sum_{s \in F_i} \max\{C(s, t), t \notin W(s) \cap R_i\} \quad (23)$$

with F_i is the edge of R_i and l_i the length of F_i .

- The contrast of each region R_i is given by the following equation:

$$C(R_i) = \begin{cases} 1 - \frac{I_i}{E_i} & \text{si } 0 < I_i < E_i \\ E_i & \text{si } I_i = 0 \\ 0 & \text{sinon} \end{cases} \quad (24)$$

So, the global contrast is:

$$C_Z = \frac{1}{A} \sum_i A_i \cdot C(R_i) \quad (25)$$

IV. PROPOSED FRAMEWORK

As each segmentation method has its own advantages and limitations, and there is no ideal segmentation procedure, implementing cooperative approaches can be more effective and powerful, enhancing the desired outcomes. Therefore, our proposed cooperative framework entails the following steps:

- Compute the uniformity parameter in order to classify the original and point out the options to be applied (the number of features and the analysis window size).
- Reduce the number of intensities in the original image via multi-level thresholding process.
- For each category, extract texture features using the new Modified Gray-Level Cooccurrence Matrix "M-GLCM".
- Apply the Mean-shift clustering on the features space derived from the "M-GLCM".
- Segment the original image by applying the k-means classification using the Mean-Shift centroids as the initial starting means.
- Evaluate the performance of the proposed framework and suggest improvement guidelines.

The subsequent sections highlight the impact of our framework's robustness keys on the time cost and memory size minimization, namely, the M-GLCM compared to the GLCM Matrix for both cases: original and multi-level thresholded images; and the adaptability of the features space to the image's type.

A. Image Categorization

To reach the set goal and adapt the applied processing to the image context, our first step is to define the image type, precisely, uniform, lowly-textured or highly-textured image, depending on a specific uniformity parameter.

Inspired by the work of C. Rosenberger [53] which aimed to locate textured and uniform regions within the image to adapt the utilized segmentation techniques to each zone, we have chosen to assess the uniformity of the global image to apply the appropriate methods and select the suitable features according to the resulting categorization. We have picked a simple uniformity criterion, which is fast to implement

and provides good results. The uniformity criterion calculates a ratio based on the variance statistics using a sliding window of size $w \times w$ pixels centered on the processed pixel (w is impair). The details of this step is not discussed in this paper.

B. New Modified GLCM

Developing Modified GLCM reduces both computational time and memory usage while maintaining the robustness of the extracted features. Many papers have been already published in that research axis. In [41], a parameter called rate was introduced to represent the scale of the features of M-GLCM, which is no longer calculated in the single direction, but all element pairs that meet the scale of the features conditions are calculated.

To illustrate our idea, we will use the same Matrix A showed previously. The size of its GLMC is 8×8 for any particular distance d and orientation θ . However, the number of the gray levels that actually exists in the matrix A is 4 given by the set of values $V = \{0, 2, 4, 7\}$.

The idea consists of using the set of values V as an indexation vector. Thus, the size of the new modified matrix "M-GLCM" will be 4. The figure (2) below illustrates this transformation:

For the same GLCM presented in the previous subsection (C.1), the size of the new M-GLCM is reduced to the half. The gain in size and so, in the needed processing time depends on the actual number of gray levels the image contains. It ranges from no changes when the original image holds all the gray scale values (256 levels), to just a few number of gray levels when the image is uniform.

Figure 2 demonstrates how to get the new M-GLCM from the previous matrix $GLCM_{1,135^\circ}$:

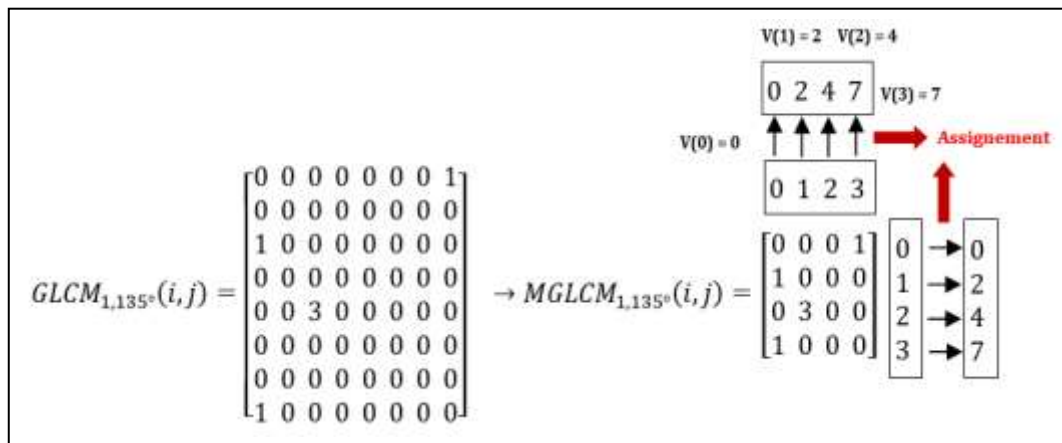


Fig. 2 The new M-GLCM calculation

B.1. The M-GLCM features

The different features extracted from the new M-GLCM could be driven directly from the original ones by using the indexation vector V. The features given previously can be reformulated as follows:

$$1. \text{ Contrast : } \text{Contr} = \sum_{i=0}^{NG'-1} \sum_{j=0}^{NG'-1} |v(i) - v(j)|^m \cdot (MI_{d,\theta}(i, j))^n, \quad m, n > 0 \quad (26)$$

$$2. \text{ Entropy: } \text{Entrp} = - \sum_{i=0}^{NG'-1} \sum_{j=0}^{NG'-1} MI_{d,\theta}(i, j) \cdot \log (MI_{d,\theta}(i, j)) \quad (27)$$

$$3. \text{ Energy: } \text{Energ} = \sum_{i=0}^{NG'-1} \sum_{j=0}^{NG'-1} (MI_{d,\theta}(i, j))^2 \quad (28)$$

$$4. \text{ Homogeneity : } \text{Homog} = \sum_{i=0}^{NG'-1} \sum_{j=0}^{NG'-1} \frac{1}{1+(v(i)-v(j))^2} \cdot MI_{d,\theta}(i, j) \quad (29)$$

C. Selected Features and Window Size

For each image category, a specific set of features is calculated within a precise window size, as shown in the following list:

- 1) Uniform image case: For each pixel, only first-order attributes are computed, mainly the mean and variance within a window of 5 x 5 pixels.
- 2) Low-textured image case: For each pixel, two M-GLCM features (Energy and Contrast) are added to the space used in the uniform case. The window size is taken as 9 x 9 pixels.
- 3) Highly image case: The features space is expanded with all the features presented earlier using 15 x 15 pixels for the window size.

D. Modes Detection using Mean-shift Space-Range presentation

The Mean-shift algorithm is known for its ability to adaptively estimate the number of clusters based on the density of data points, making it suitable for texture segmentation tasks. In our work, we adopted the Space-Range presentation given in [46]. The space of the lattice is known as the spatial domain (pixel's coordinate), while the range domain is represented by the gray level of the monochrome images or the color components in case of colored images.

The multivariate kernel, in the joint spatial-range domain, is defined as the product of two radially symmetric kernels as shown as follows:

$$K_{h_s, h_r}(x) = \frac{c}{h_s^2 \cdot h_r^2} \cdot k \left[\left\| \frac{x_s}{h_s} \right\|^2 \right] \cdot k \left[\left\| \frac{x_r}{h_r} \right\|^2 \right] \quad (31)$$

where x_s is then the space part (pixel coordinates) and x_r the range part (features) of the pixel. h_s and h_r are the respective bandwidth for both domains and they have to be given before computation.

At the end of the algorithm execution, all the resulting modes that are closer than h_r in the range domain and h_s in the spatial domain are merged.

E. K-Means Classification

At the final stage, the K-Means algorithm is initialized by the modes derived from the Mean-shift step. It is then executed until it converges. For each resulting texture class, the original image pixels that belongs to it, are grouped and labeled in a distinctive unique color producing the segmented images.

F. Multi-level Thresholding Effect

The aim of a multi-level thresholding phase is to reduce the number of gray levels to be processed without altering the image valuable information. Thresholding methods have, in general, a simple implementation and high calculation speed [54]. In two-level thresholding procedure, the image is divided into two categories (object and background), whereas the multi-level thresholding method computes multiple thresholds by which the image is divided into several classes [55][56]. This technique is applied prior to feature extraction in order to achieve the expected improvements.

As the GLCM matrices are very sensitive to the texture samples size, the number of gray levels is often reduced. Thus, applying a multi-level thresholding on the image and then, computing the proposed M-GLCM matrix will reduce notably the computation time and so, the needed processing memory size.

For that purpose, we have applied the OTSU's multilevel thresholding method [55] carrying out an analysis window of 8 x 8 pixels. The resulting thresholds are processed to get a unique set of values that is then used to reclassify the original pixels according to their nearest threshold using the Euclidean measurement. The details of the OTSU's multi-level thresholding is not given in this article.

A comparison of the M-GLCM performance, when applied on the original image and on the multilevel thresholded image, is shown in the next section.

V. EXPERIMENTAL RESULTS AND DISCUSSION

In the field of texture processing, a large number of datasets have been developed. After the implementation of the techniques presented earlier, we have assessed the developed framework outcomes by conducting our experiments on natural texture image databases, including the Brodatz texture database and the USC-SIPI texture mosaic [56], in which the images were digitized to 256 gray levels, the size of each image was 256 x 256 pixels.

A. GLCM versus M-GLCM

Figure 3 demonstrates the advantage of the new M-GLCM matrix over the traditional GLCM. The calculation time given below reveals the valuable amelioration that could be reached when a pixel-by-pixel processing is required in real applications.

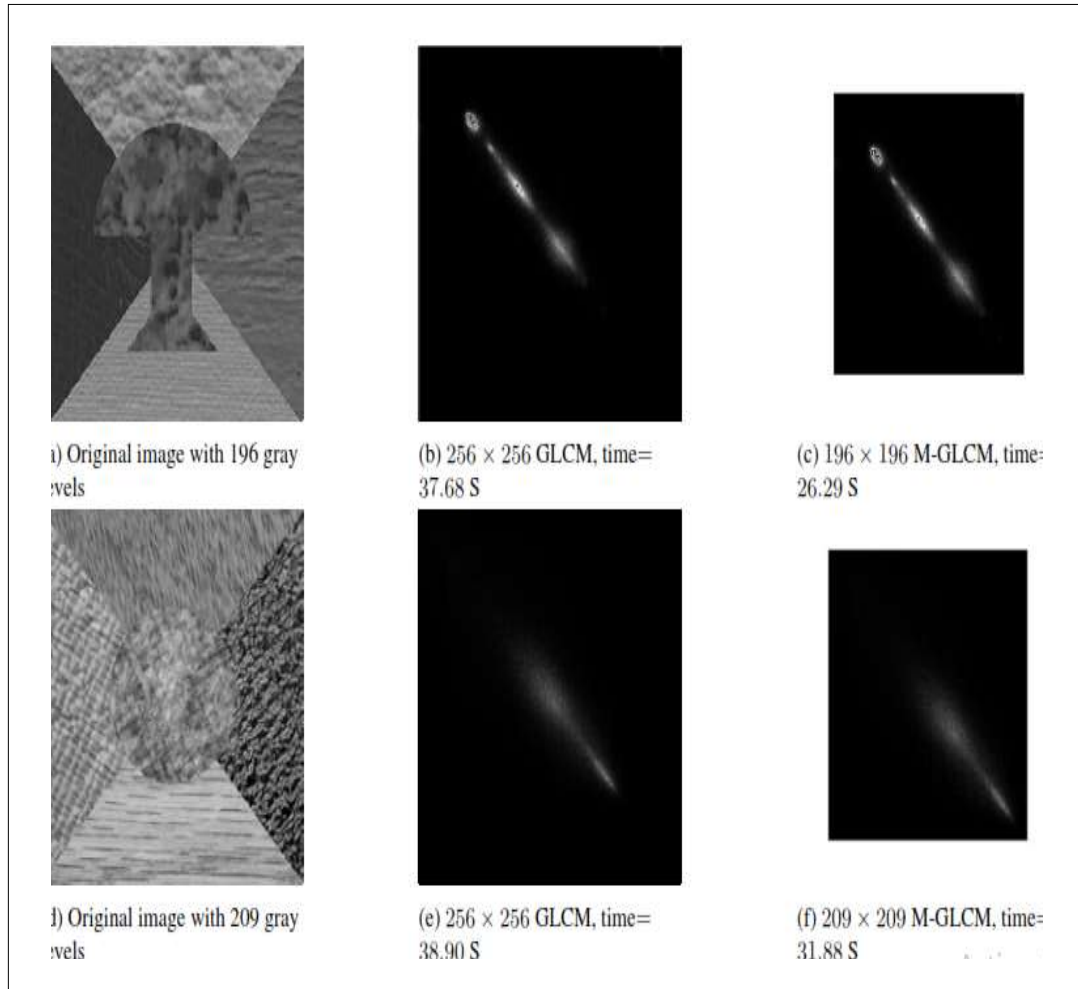


Fig. 3 GLCM versus M-GLCM

B. OTSU's multi-level thresholding Benefit

The gain in processing duration is more accentuated when we applied the Otsu's multi-level thresholding on the original image before the M-GLCM computation as shown in "Fig. 4":

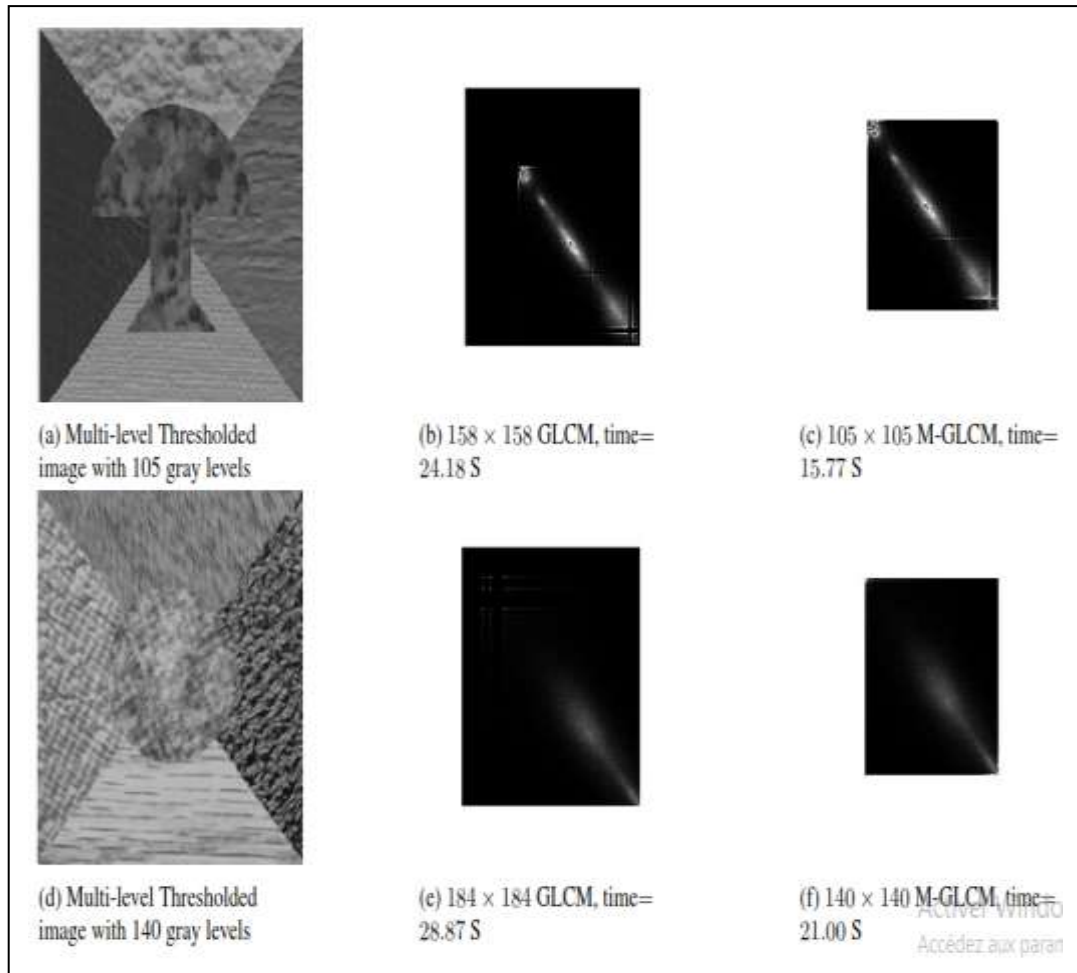


Fig. 4 Upturn of the Multi-level thresholding process

C. Segmentation results

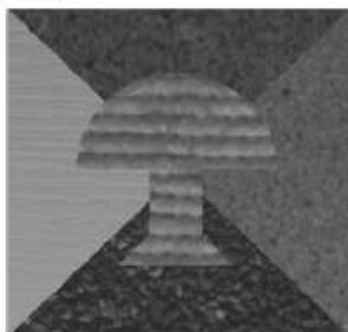
For all our results, the texture features were calculated based on four (04) M-GLCMs matrices, applying the four directions (horizontal, vertical, left and right diagonal) with distance $d = 1$. The average of the four resulting parameters for each pixel is then used. Figure 5 shows some of the proposed framework results:



(a) Original image with 102 gray levels



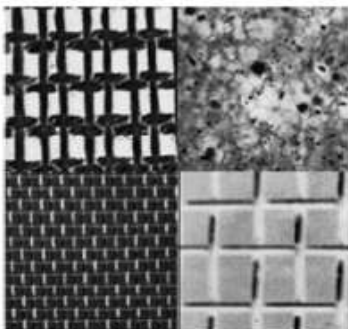
(b) Segmented Image, $hs = 7$, $hr = 37$, $time = 22.62$ S



(c) Original image with 155 gray levels



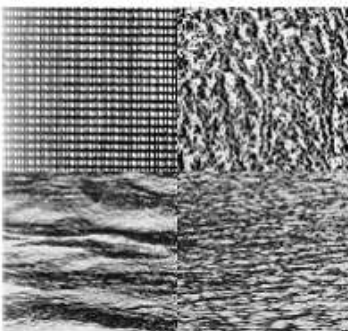
(d) Segmented Image, $hs = 9$, $hr = 23$, $time = 25.31$ S



(e) Original image with 249 gray levels



(f) Segmented Image, $hs = 15$, $hr = 27$, $time = 970.16$ S



(g) Original image with 190 gray levels



(h) Segmented Image, $hs = 15$, $hr = 27$, $time = 355.06.2$ S

Fig. 5 Proposed framework segmentation results

For all the resulting images, we have reported their evaluation criteria values in Table (1) below. The obtained texture segmentation results are judged to be satisfying, leading to low values of Levine and Nazif criterion and high values for Zeboudj criterion.

Concerning the computation time, when the images are classified as uniform or lowly-textured, the processing was notably fast. However, it is considerably high for the images highly-textured, especially when they contains coarse textures.

TABLE 1 Evaluation results

Segmented Image of "Fig. 5"	Zeboudj Values	Levine and Nazife Values
Image (b)	0.81	0.40
Image (d)	0.62	0.04
Image (f)	0.45	0.17
Image (h)	0.41	0.24

In all the carried experiments, the best performance are achieved when the images do not contain many coarse textures. The developed framework is much suitable for uniform and fine textures.

Considering the low number of features used in this paper, extending the space of features would be a good option to increase the system performance. Assessing the outcomes using other evaluation criteria would be recommended.

IV. CONCLUSION

Discriminating different textures in an image is a challenging task. In this paper, we have implemented an hybrid and adaptive segmentation approach to pick the appropriate parameters according to the type of the image to be processed. The proposed research aims to introduce a novel texture segmentation algorithm that combines the strengths of the Modified GLCM with the advantages of the Mean-shift algorithm. This M-GLCM exploits the actual gray levels that exist in the image. This transformation reduces the memory size to be allocated and shortens the calculation time. The advantage of this new matrix is more emphasized when the image is multi-level thresholded beforehand.

For future work, based on the research experiments, we highlight the importance of the texture features selection procedure, which can influence considerably the final framework performance. We recommend also adding other texture features derived from other suitable methods to enhance the system ability to discriminate more coarse textures.

REFERENCES

- [1]. Rafael C. Gonzalez, Richard E. Woods. Digital image processing. Fourth Edition, Global Edition. Pearson Education Limited, 2018.
- [2]. Mihran Tuceryan, Anil K. Jain. Texture analysis (Chapter 2.1). The Handbook of Pattern Recognition and Computer Vision (2nd Edition), by C. H. Chen, L. F.

- Pau, P. S. P. Wang (eds.), pp. 207-248, World Scientific Publishing Co., 1998.
<https://doi.org/10.1142/97898143431380010>.
- [3]. Angulakshmi M., Lakshmi Priya G.G. Automated brain tumour segmentation - a review. *International Journal of Imaging Systems and Technology*. Vol. 27, pp 66–77, 2017.
- [4]. Rundo L. et al. Combining split-and-merge and multi-seed region growing algorithms for uterine fibroid segmentation in MRgFUS treatments. *Medical, & Biological Engineering & Computing*. Vol. 54, No. 7, pp. 1071–1084, 2016.
<https://doi.org/10.1007/s11517-015-1404-6>, 2-s2.0-84946151274.
- [5]. Liang H. Weller, D. S. Edge-based texture granularity detection. *IEEE International Conference on Image Processing (ICIP)*, pp. 3563-3567, 2016.
<https://doi.org/10.1109/ICIP.2016.7533023>.
- [6]. R. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67:786–804, 1979. <https://doi.org/10.1109/PROC.1979.11328>.
- [7]. L. Armi and S. Fekri-Ershad. Texture image analysis and texture classification methods - a review. *International Online Journal of Image Processing and Pattern Recognition*, Vol. 2, No.1, pp. 1-29, 2019.
- [8]. R. N. Khadse, A. K. Khursange, S. V. Khekare, R. R. Rathod, Prof. Ajay D. Nanure. Texture analysis method – a survey. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, Volume 3, Issue 1, December 2023.
- [9]. A. Eleyan, and H. Demirel. Co-occurrence matrix and its statistical features as a new approach for face recognition. *Turkish Journal of Electrical Engineering and Computer Sciences*: Vol. 19, No. 1, Article 8. <https://doi.org/10.3906/elk-0906-27>.
- [10]. B. Julesz. Visual pattern discrimination. *IRE Transactions on Information Theory*, IT-8:84–92, 1962. <https://doi.org/10.1109/TIT.1962.1057698>.
- [11]. T. Randen and J. H. Husoy. Filtering for texture classification: a comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291-310, April 1999. <https://doi.org/10.1109/34.761261>.
- [12]. Mircea Cimpoi, Subhransu Maji, Andrea Vedaldi. Deep filter banks for texture recognition and segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. <https://doi.org/10.1109/CVPR.2015.7299007>.
- [13]. J. Zhang, P. Fieguth, and D. Wang. Random field models. In *Handbook of Image and Video Processing*, pages 301–312. Academic Press, San Diego, CA, 2000.
- [14]. G. R. Cross and A. K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 1, pp. 25-39, Jan. 1983. <https://doi.org/10.1109/TPAMI.1983.4767341>.
- [15]. Yang P. Yang G. Feature extraction using dual-tree complex wavelet transform and gray level co-occurrence matrix. *Neurocomputing*. Vol. 197, pp 212-220, 2016.
- [16]. Alamgir N. Myeongsu K. Yung-Keun K. Cheol-Hong K. and Jong-Myon K. Hybrid technique for medical image segmentation. *Journal of Biomedicine and Biotechnology*. Corporation Issue 1, Vol. 2012.
<https://doi.org/10.1155/2012/830252>.
- [17]. Robert M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transaction on Systems, Man and Cybernetics*, Vol. SMC-3, No. 6, p.p 610–621, Nov. 1973. Printed in the U.S.A.

- [18]. O. Mahmoud, Abu Abbas. Comparisons between data clustering algorithms. International Arab Journal of Information Technology, Vol. 5, No. 3, July 2008, p.p: 320-325.
- [19]. Akula Ratna Babu, Miriyala Markandeyulu, Bussa V R R Nagarjuna. Pattern clustering with similarity measures. International Journal of Computer Technology \& Applications, Vol. 3, No. 1, p.p 365-369, ISSN:2229-6093.
- [20]. S. M. Aqil Burney and H. Tariq. K-Means cluster analysis for image segmentation. International Journal of Computer Applications (0975 – 8887), vol. 96, N° 4, pp: 1-8, Jun. 2014. <https://doi.org/10.5120/16779-6360>.
- [21]. Claude Cariou, Steven Le Moan, Kacem Chehdi. A Novel meanshift algorithm for data clustering. IEEE Access, 2022, 10, pp.14575-14585. <https://doi.org/10.1109/ACCESS.2022.3147951>.
- [22]. Mustafa, M., Taib, M.N., Murat, Z.H. and Hamid, N.H.A. GLCM texture classification for EEG spectrogram image. IEEE EMBS Conference on Biomedical Engineering \& Sciences, Kuala Lumpur, 30 November-2 December 2010, 373-376.
- [23]. Mireille Pouyap, Laurent Bitjoka, Etienne Mfoumou and Denis Toko. Improved bearing fault diagnosis by feature extraction based on GLCM, fusion of selection methods, and multiclass-naïve bayes classification. Journal of Signal and Information Processing, Vol. 12, No. 4, pp. 7185, 2021.
- [24]. O.M. O'Byrne, B. Ghosh, V. Pakrashi and F. Schoefs. Texture analysis based detection and classification of surface features on ageing infrastructure elements. BCRI 2012 Bridge \& Concrete Research in Ireland, 2012, Cork, Ireland.
- [25]. L.D.O. Bastos, P. Liatsis, and A. Conci. Automatic texture segmentation based on K-Means clustering and efficient calculation of co-occurrence features. In 15th International Conference on Systems, Signals and Image Processing, IWSSIP, June 2008. Bratislava, Slovakia: Inst. of Elec. and Elec. Eng. Computer Society.
- [26]. B. S. Garra, et al. Improving the distinction between benign and malignant breast lesions: The value of sonographic texture analysis. Ultrason. Imag., Oct. 1993, Vol. 15, N° 4, pp: 267-285.
- [27]. Q. Wu, Y. Gan, B. Lin, Q. Zhang and H. Chang. An active contour model based on fused texture features for image segmentation. Neurocomputing 151, 2015, pp: 1133-1141.
- [28]. P. Maillard. Comparing texture analysis methods through classification. Photogrammetric Eng. Remote Sens., vol. 69, no. 4, pp: 357-367, Apr. 2003.
- [29]. M F.A Fauzi and P. H Lewis. A Fully unsupervised texture segmentation algorithm. British Machine Vision Conference 2003, Norwich, United Kingdom. Sep 2003. pp. 519-528.
- [30]. R. B Vallabhaneni and V. Rajesh. Brain tumour detection using meanshift clustering and GLCM features with edge adaptive total variation denoising technique. Alexandria Engineering Journal, Vol. 57, Issue 4, Dec. 2018, pp: 2387-2392. <https://doi.org/10.1016/j.aej.2017.09.011>.
- [31]. T. Haryanto et al. Multipatch-GLCM for texture feature extraction on classification of the colon histopathology images using deep neural network with GPU acceleration. Journal of Computer Science, 2020, Vol. 16(3), pp: 280.294. <https://doi.org/10.3844/jcssp.2020.280.294>
- [32]. D. J. Marceau, P. J. Howarth, J. M. Dubois and D. J. Gratton. Evaluation of The grey-level co-occurrence matrix method for land-cover classification using spot imagery. in IEEE Transactions on Geoscience and Remote Sensing, vol. 28, no. 4, pp. 513-519, July 1990. <https://doi.org/10.1109/TGRS.1990.572937>.

- [33]. M.E. Hodgson. What size window for image classification? A cognitive perspective. *Photogrammetric Engineering & Remote Sensing*, 1998, Vol. 64(8), pp: 797-807.
- [34]. L.V. Dutra and N.D.A. Mascarenhas. Some experiments with spatial feature extraction methods in multispectral classification. *International Journal of Remote Sensing*, Vol. 5(2), pp: 303-313.
- [35]. D. Puig, M.A. García). Pixel-based texture classification by integration of multiple texture feature evaluation windows. In: Perales, F.J., Campilho, A.J.C., de la Blanca, N.P., Sanfeliu, A. (eds). *Pattern Recognition and Image Analysis. IbPRIA 2003. Lecture Notes in Computer Science*, Vol. 2652. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-44871-6_92.
- [36]. P. Dhal and C. Azad. A comprehensive survey on feature selection in the various fields of machine learning. *Applied Intelligence*, 2022, Vol. 52(4), pp: 4543–4581. <https://doi.org/10.1007/s10489-021-02550-9>.
- [37]. Q. Kanglin, Xu Jiucheng, H. Qincheng, Q. Kangjian and S. Yuanhao. Feature selection using information gain and decision information in neighborhood decision system. *Applied Soft Computing*, Vol. 136, 2023, ISSN 1568-4946. <https://doi.org/10.1016/j.asoc.2023.110100>.
- [38]. T.K. Florentina, I. Sembiring, A. Setiawan, I. Setyawan and R.R Huizen. GLCM-based feature combination for extraction model optimization in object detection using machine learning. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, Vol. 9, No. 4, December 2023, pp. 1196-1205, ISSN: 2338-3070. <https://doi.org/10.26555/jiteki.v9i4.27842>.
- [39]. N. Aggarwal and R. K. Agrawal. First and second order statistics features for classification of magnetic resonance brain images. *Journal of Signal and Information Processing*, 2012, 3, 146-153. <http://dx.doi.org/10.4236/jsip.2012.32019>
- [40]. A. Materka and M. Strzelecki. Texture analysis methods – a review. Technical University of Lodz, Institute of Electronics, COST B11 report, Brussels 1998
- [41]. R. W. Connors and C. A. Harlow. A theoretical comparison of texture algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. PAMI-2, No. 3, pp. 204–222, May 1980.
- [42]. Fritz Albregtsen. Statistical texture measures computed from gray level co-occurrence matrices. Image Processing Laboratory. Department of Informatics, University of Oslo. November 5, 2008. Available at the site:
- [43]. <https://www.uio.no/studier/emner/matnat/ifi/INF4300/h08/undervisningsmateriale/glcm.pdf>
- [44]. A. Gelzinisa, A. Verikasa, M. Bacauskienea, Increasing the discrimination power of the co-occurrence matrix-based features. *Pattern Recognition*, vol. 40, pp. 2367–2372, 2004.
- [45]. M. O'Byrne, B. Ghosh, V. Pakrashi and F. Schoefs. Texture analysis based detection and classification of surface features on ageing infrastructure elements. BCRI2012 Bridge & Concrete Research in Ireland, 2012, Cork, Ireland. fhal-01009012f.
- [46]. S. Sharma, T. Choudhury and Y. Singh. Advanced feature extraction for mammogram mass classification: a multi-scale multi-orientation framework. *International Journal on Smart Sensing And Intelligent Systems*. Issue 1, Vol. 18, 2025. <https://doi.org/10.2478/ijssis-2025-0022>.
- [47]. Dorin Comaniciu, Member, IEEE, and Peter Meer, Senior Member, IEEE. MeanShift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 24, N°5, May 2002.

- [48]. S. Balasubramanian Subbiah and C. Seldev. Image Classification through integrated K- Means Algorithm. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 2, March 2012, ISSN (Online): 1694-0814.
- [49]. D. Nameirakpam, M. Khumanthem and J.C. Yambem. Image Segmentation using K-means Clustering Algorithm and Subtractive Clustering Algorithm. Procedia Computer Science 54, 2015, pp:764-771.
<https://doi.org/10.1016/j.procs.2015.06.090>.
- [50]. J.M.H. du Buf, M. Kardan, M. Spann. Texture feature performance for image segmentation. Pattern Recognition, Vol.23, Issues 3-4, 1990, pp 291-309. ISSN 0031-3203. [https://doi.org/10.1016/0031-3203\(90\)90017-F](https://doi.org/10.1016/0031-3203(90)90017-F).
- [51]. Levine, M. and Nazif, D. Dynamic measurement of computer generated images segmentation. PAMI, (March 1985), 155-164.
<https://doi.org/10.1109/TPAMI.1985.4767640>
- [52]. R. Zeboudj. Filtrage, seuillage automatique, contraste et contours : du pré-traitement à l'analyse d'image. PhD, Université de Saint-Etienne, 1988.
- [53]. C. Rosenberger. Mise en œuvre d'un système adaptatif de segmentation d'images. Doctorate thesis, Rennes 1 university, 1999, N°2292.
- [54]. Juan Pablo Balarini, Sergio Nesmachnow. A C++ Implementation of Otsu's Image Segmentation Method. Published in Image Processing On Line, on 2016-07-10, ISSN 2105-1232. Available at the site: <http://dx.doi.org/10.5201/ipol.2016.158>.
- [55]. Elsayed Abd Elaziz. M, Oliva. D and Ewees. A. Multi-level Thresholding-based Gray Scale Image Segmentation Using Multi-objective Multi-verse Optimizer. Expert Systems with Applications. Vol.125, 2019.
<https://doi.org/10.1016/j.eswa.2019.01.047>.
- [56]. Deng-Yuan Huang et al. Automatic Multilevel Thresholding based on Two-stage Otsu's Method with Cluster Determination by Valley estimation. International Journal of Innovative Computing, Information and control, Vol. 7, N°10, October 2011 ISSN 1349-4198, pp 5631-5644. Available at the site: <http://www.ijicic.org/ijicic-10-05033.pdf>