Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 20(6): 10063–10089.
DOI: 10.3934/mbe.2023442
Received: 16 November 2022
Revised: 27 February 2023
Accepted: 07 March 2023
Published: 29 March 2023

Research article

Restored texture segmentation using Markov random fields

Sanjaykumar Kinge[1], B. Sheela Rani[2,*] and Mukul Sutaone[3]

[1] Department of ECE, Sathyabama Institute of Science and Technology, Chennai, and Assistant Professor, School of Electronics and Communication, MIT-World Peace University, Pune 411038, India
[2] Sathyabama Institute of Science and Technology, Chennai 600119, (Tamil Nadu), India
[3] College of Engineering Pune 411 005, (Maharashtra), India

* **Correspondence:** Email: directorresearch@sathyabama.ac.in.

Abstract: Texture segmentation plays a crucial role in the domain of image analysis and its recognition. Noise is inextricably linked to images, just like it is with every signal received by sensing, which has an impact on how well the segmentation process performs in general. Recent literature reveals that the research community has started recognizing the domain of noisy texture segmentation for its work towards solutions for the automated quality inspection of objects, decision support for biomedical images, facial expressions identification, retrieving image data from a huge dataset and many others. Motivated by the latest work on noisy textures, during our work being presented here, Brodatz and Prague texture images are contaminated with Gaussian and salt-n-pepper noise. A three-phase approach is developed for the segmentation of textures contaminated by noise. In the first phase, these contaminated images are restored using techniques with excellent performance as per the recent literature. In the remaining two phases, segmentation of the restored textures is carried out by a novel technique developed using Markov Random Fields (MRF) and objective customization of the Median Filter based on segmentation performance metrics. When the proposed approach is evaluated on Brodatz textures, an improvement of up to 16% segmentation accuracy for salt-n-pepper noise with 70% noise density and 15.1% accuracy for Gaussian noise (with a variance of 50) has been made in comparison with the benchmark approaches. On Prague textures, accuracy is improved by 4.08% for Gaussian noise (with variance 10) and by 2.47% for salt-n-pepper noise with 20% noise density. The approach in the present study can be applied to a diversified class of image analysis applications spanning a wide spectrum such as satellite images, medical images, industrial inspection, geo-informatics, etc.

Keywords: cellular automata; Markov random fields; noisy textures; texture databases; customized median filter

1. Introduction

Image segmentation is a necessary and important process in analyzing and recognizing images. Some of its applications are retrieving images, analyzing medical images, machine learning, industrial inspections and pattern analysis. Various well-established segmentation techniques include clustering-based approaches, thresholding, level set method, edge-based and parametric methods, etc. Contemporary image segmentation techniques that deliver better performance than the most recent advanced techniques are suggested in [1–3]. A computationally efficient Convolutional Neural Network (CNN) model is developed in [1] using the attention mechanism and adding dilated convolution to enhance segmentation performance for small regions in medical images. A fast and accurate satellite cloud image segmentation technique is suggested in [2] using CNN to extract small regions by fusing together feature information, assigning weights to features based on their importance, and optimizing it using the channel attention technique. Segmentation of natural and medical images is reported in [3] using the contemporary clustering technique. These researchers developed a novel structure for storing prior information about the image at the region and pixel levels. This structure helps the clustering process to efficiently identify pixels belonging to the same class and reduces the similarity of pixels belonging to different classes.

Texture segmentation has been regarded as a well-known research area since it uses image texture as a fundamental feature. Researchers have found that it is not easy to come to an agreement on a unique acceptable texture definition because of the multi-faceted properties of texture. Texture, according to Haralick [4], is a reoccurrence of pixel-grey levels in a neighborhood. They performed the classification of texture into various categories such as fine, coarse, irregular or regular. Researchers around the globe are working on various texture research domains, namely, texture-based classification and segmentation, shape extraction using texture, synthesizing texture, etc. The application of texture analysis includes retrieving images from large databases, diagnoses of medical images, natural scene synthesis, climate predictions, 3D imaging, shape extraction, industrial inspections, etc.

Researchers broadly divided texture analysis approaches into four categories, namely, signal processing method, model-based approach, structural and statistical approach [5–7]. The model-based approach tries to capture the process used for texture generation [8]. Some examples of model-based approaches include Fractal models [9,10], auto-regression models and Markov random field (MRF) models [11,12]. The MRF-based model is found to be very useful for texture-based image segmentation and classification. In the statistical approach, the texture is described using statistical measures estimated from pixels in a local neighborhood. The traditional examples of statistical approaches include local binary patterns (LBP) and gray level co-occurrence approaches. Both of these approaches use local statistics for texture feature extraction and they are widely used for texture classification [13–16].

Signal processing approach for texture feature extraction is also called the transform-based approach and it performs feature extraction based on frequency content in textures. Some remarkable examples of this approach are pseudo-Wigner distribution [17–20], Law's filters [21] and Gabor filters [22–25]. The structural approach is based on the view that textures are made up of texture primitives. In this approach, it is common to identify placement rules that describe texture [4]. The primitives are extracted by edge detection with Laplacian of Gaussian [26], adaptive region extraction or mathematical morphology [27].

The relevant literature on texture segmentation in the past three decades has been reviewed for the present study. Texture segmentation using Gabor filters with conventional classifiers [20,22–24,28,29] has been discussed here, in particular. Markov Random Fields (MRF) based texture segmentation for medical and SAR images have been proposed in [29–32]. In the past few years, deep learning approaches using convolutional neural networks (CNN) have been used for texture analysis because of their excellent performance for applications in image processing and computer vision. However, most of the literature is available for texture synthesis [33–35] and texture classification [36–44] and only a few approaches are published on texture segmentation. An approach using CNN for texture segmentation is reported in [38], which uses the CNN network for feature extraction and support vector machine for classification. An approach in [45] performs texture segmentation using CNN based on a fully convolutional network. It utilizes information in both deep network layers and shallow layers to perform segmentation. U-net-based texture segmentation is performed in [46], and its performance comparison is made with four classic approaches. Texture feature extraction is performed using both CNN and curvelet transform in [47]. They achieved texture segmentation based on information in deep and shallow layers of the network. The technique in [48] used a Siamese CNN for feature extraction and performed texture segmentation using the region merging approach. All of these researchers have focused on augmentation in the segmentation accuracy of noise-free textures.

The Prague texture benchmark is created to assess the quantitative performance of texture segmentation approaches and to perform their ranking [49]. Therefore, this benchmark is used for experimentation in this study [49]. Around 11 plus modern texture, segmentation techniques are evaluated on the Prague benchmark. None of these approaches is evaluated on textures tainted with noise. As per the literature survey performed for this study, many researchers performed experimentation on Brodatz textures. Therefore, the segmentation approach proposed in this study is evaluated on Brodatz textures. In many of the practical applications, Gaussian and salt-n-pepper noise degrade texture images during image acquisition and transmission [50–52]. According to a recent study [53–55], researchers have begun working on textures contaminated with noise for various applications. These applications include retrieving and analyzing texture images contaminated with noise. They used a few databases, namely, Brodatz, Outex, Curet and STex, in experimentation, by and large. As per the most recent literature, some additional restored texture-based applications are thyroid cancer diagnosis [56] and automated quality evaluation of vegetables [57]. As per the literature reviewed for this study, researchers concentrated on augmenting the accuracy of texture segmentation techniques and the segmentation problem of textures tainted with noise is so far not addressed to the expected extent. The approach proposed for the segmentation of restored textures in this investigation is motivated by recent literature in [56,57] and the need for the segmentation of noisy textures.

This study aims to perform the segmentation of noisy color textures and achieve high segmentation accuracy. According to this literature review, the majority of researchers experimented on non-contaminated textures. However, in actual applications, noise gets captured during the measurement process and it has a negative impact on results of segmentation techniques. Therefore, it is crucial to assess how well texture segmentation methods perform on texture images that have been contaminated by various noises, including Gaussian, salt-n-pepper, Poisson, and Speckle noise. As per [58] 25 plus texture databases are available for experimentation and noisy texture segmentation is not so far explored to the expected extent on all such databases.

A three-phase segmentation approach using MRF and Customized Median Filter (CMF) is developed as part of this study and is utilized for the segmentation of noisy benchmark textures in the Brodatz and Prague datasets. In the first phase of this approach, texture images are contaminated by Gaussian and salt-n-pepper noise in this study. Color Block Matching 3D (C-BM3D) algorithm is used to perform restoration of texture benchmark images contaminated by Gaussian noise. This algorithm is used for restoration because of its excellent performance among 15 recent approaches [59–63]. A bio-inspired Cellular Automata (CA) algorithm delivers excellent restoration performance among recent 14 approaches for salt-n-pepper noise [61,63,64] and it preserves edge information unlike to Median filter, Median filter's variants and many other algorithms [64]. Therefore, it is used for the restoration of textures contaminated by salt-n-pepper noise. The performance of denoising algorithms, namely, C-BM3D and CA, is excellent in terms of the restoration metrics viz. SSIM (structural similarity index) and PSNR (peak signal-to-noise ratio). The remaining two phases perform segmentation of restored texture images obtained in the first phase.

In a novel segmentation approach presented here, the closeness of the segmented results to ground truth is quantitatively determined using segmentation performance parameters, namely, SSIM, MSE and PSNR and the window size of the Customized Median Filter (CMF). This approach produces optimal segmentation results better than most recent approaches, namely, fuzzy color aura matrices (FCAM) [65], Karabag [46], and Kiechle [31], even after degrading the texture benchmark images with Gaussian and salt-n-pepper noise. As per the literature survey and to the best of our knowledge, Median filter is yet not customized based on segmentation performance parameters to get optimal segmentation results. The proposed approach also delivers excellent performance for scale and rotational variance, even though that is not an immediate objective of this investigation. The proposed approach achieves higher accuracy due to the excellent restoration performance of denoising algorithms along with color-based discrimination used in the proposed texture segmentation technique. It is concluded from visual inspection of color benchmark textures in Brodatz and Prague benchmark that most of these benchmark images contain texture classes that have distinct colors and the variation of color in a texture segment is very small. Therefore, the classes in these benchmark images can be separated precisely based on color features. The approach developed herein has applications in the analysis of human skin and hair [66], medical image analysis, an inspection of industrial objects [67] and texture-based retrieval and analysis [53], etc.

The further part of this paper is organized as follows: Restoration methods used in the first phase of the proposed technique and the remaining two phases of the proposed segmentation technique using MRF and Customized Median filter are depicted in Section 2. The results along with experimentation, are discussed in Section 3, followed by a conclusion in Section 4.

2. Materials and methods

2.1. Restoration methods in the first phase

Cellular Automata (CA) algorithm is used for restoring texture images contaminated with salt-n-pepper noise. It is described in the first part of this section. C-BM3D approach is used for restoring texture images contaminated with Gaussian noise. The second part of this section describes the C-

BM3D algorithm.

2.1.1. Restoration of textures tainted with salt-n-pepper noise

CA algorithm is used for different tasks in analyzing, processing and recognizing images [64,68]. Herein, texture images contaminated by salt-n-pepper noise are restored using this algorithm. An exhaustive description of this algorithm is presented in [64]. The entire input image is treated as a 2D lattice. This algorithm decomposes the entire image into a small array of pixels for filtering noise. Let's establish a few notations for the description of the CA algorithm. P_{max} and P_{min} are used to indicate the pixel's maximum and minimum values, respectively. In this study, $P_{max} = 255$ and $P_{min} = 0$ because pixel value is represented using 8 bits. A 3 by 3 window's center pixel is denoted by the symbol $P_{i,j}$. It further denotes neighborhood by N. There are eight neighbor pixels to the center pixel in a 3 by 3 window. Hence, $N = 8$ in the present study. However, theoretically, there is no higher limit on neighborhood size in CA algorithm. A 3 by 3 window's pixel values are listed in an array denoted by P in ascending order, excluding the center pixel. The study uses I_R, I_G and I_B to designate the three components of the color image.

A set of three local rules are used to update the state of the corrupted pixel. These rules are used to determine the new state of the pixel under processing based on the state of its neighboring pixels. Let us first discuss the need for the CA algorithm using Eqs (1) and (2) in general, is described in the next part of this section and later on, it discusses this algorithm analytically for this study. In general, the CA algorithm needs a set of pixels $P(\vec{r}, t)$ specified by Eq (1). The pixels in this set are associated with each pixel in a 2D image array at the time instants $t = 0, 1, 2, 3, \dots;$ in the CA algorithm.

$$P(\vec{r}, t) = \{P_1(\vec{r}, t), P_2(\vec{r}, t), \dots, P_m(\vec{r}, t)\} \tag{1}$$

where each pixel in the input image is denoted by $\vec{r}$ and the total number of neighbor pixels of the pixel under processing is denoted by m.

$$P_j(\vec{r}, t+1) = R_j\left(P(\vec{r}, t), P(\vec{r} + \vec{\delta}_1, t), P(\vec{r} + \vec{\delta}_2, t), \dots, P(\vec{r} + \vec{\delta}_q, t)\right) \tag{2}$$

The new state of the pixel $P_j(\vec{r}, t+1)$ is determined using Eq (2). In this equation, $R_j$is j^{th} rule in a set of rules $R = \{R_1, R_2, \dots, R_n\}$. In this study, total rules are three. Hence, $n = 3$. Herein, $\vec{r} + \vec{\delta}_q$ indicates one of the pixels in the neighborhood of the center pixel in a 3 by 3 window. Herein, the maximum value of $q = 8$. The rule R_j is applied on pixels residing in the neighborhood of the center pixel under processing to determine the new state of the pixel $P_j(\vec{r}, t+1)$. The set of rules is the same for all pixels in an image matrix. Equation (2) indicates that the new state at time $t + 1$ is determined based on states at time t. However, sometimes the next state of the pixel at instant $t + 1$ is determined using previous states at time instants, namely, $t - 1, t - 2, t - 3, \dots, t - k$. The previous states are used in this study to determine the new state.

The pixel being processed in this approach is the central pixel of a 3 × 3 window. The CA algorithm assumes that neighbors of pixels on the boundary have zero values. When salt-n-pepper noise contaminates the image, it assigns either a maximum value of 255 or a minimum value of zero to the corrupted pixel. Corrupted pixels with the values of 255 and 0 regarded as being in the maximum and minimum states, respectively.

The CA algorithm presented here uses three simple local rules to denoise a noisy pixel. These rules are applied on a 3 by 3-pixel window in the noisy image. The value of the center pixel of this window and its 8-pixel neighbors are used to determine the next state of the pixel.

The center pixel is under processing in a 3 by 3 window located at the indices (i_0, j_0) in a 2D image matrix that can take a total of 256 values as shown by Eq (3) assuming that one pixel in an image is represented by 8-bits.

$$P^t_{(i_0,j_0)} \in \{0, \ldots\ldots, 255\} \tag{3}$$

Equation (4) represents Moore (M) neighborhood for the pixel located at indices (i_0, j_0) in an image array.

$$N(i_0, j_0)^M = \{(i,j): |i - i_0| \le r,\ |j - j_0| \le r\} \tag{4}$$

As discussed previously in this subsection 8-neighborhood is used in the CA algorithm presented here. Therefore, $N = 8$ in this equation. The range of the neighborhood is denoted by r in this equation and its value is one. It means the size of the neighborhood is 3 by 3. The three rules used in the CA algorithm presented here are described in Eqs (5)–(7) as follows. One of these equations is applied to eight pixels in the neighborhood of the pixel under processing to determine its new state. The set of indices of all the pixels in the neighborhood of the pixel under processing is denoted by $(-r \le i, j \le r)$ in all these three equations and $N = 8$ (8-neighborhood).

$$Pnew^{t+1}_{(i,j)} = mean_{-r \le i,j \le r}\left(P^t_{(i,j)}\right),$$

$$if\ \forall\ P^t_{(i\pm r, j\pm\ r)} \in N, \quad \exists P^t_{(i\pm r, j\pm\ r)} = P_{min}\ \ or\ \ P^t_{(i\pm r, j\pm\ r)} = P_{max} \tag{5}$$

Equation (5) stated above represents rule-1 in the pseudo-code of the CA algorithm described in the last part of this subsection and $r = 1$ in this equation. This equation states that the new state of the pixel under processing is the mean of all the values of pixels in the neighborhood of the center pixel in a 3 × 3 window excluding the pixel with values of P_{min} and P_{max}.

$$Pnew^{t+1}_{(i,j)} = \ P_{max}\quad if\ \forall\ P^t_{(i\pm r, j\pm\ r)} \in N,\ P^t_{(i\pm r, j\pm\ r)} = P_{min}\ \ or\ \ P^t_{(i\pm r, j\pm\ r)} = P_{max} \tag{6}$$

Equation (6) stated above represents rule-2 in the pseudo code of the CA algorithm described in the last part of this subsection and $r = 1$ in this equation. This equation states that the new state of the pixel under processing is P_{max} if all the values of pixels in the neighborhood of the center pixel in the 3 × 3 window is either P_{min} or P_{max}.

$$Pnew^{t+1}_{(i,j)} = mean_{-r \le i,j \le r}\left(P^t_{i,j}\right),$$

$$if\ \forall\ P^t_{(i\pm r, j\pm\ r)} \in N,\ P^t_{(i\pm r, j\pm\ r)} \ne P_{min}\ \ or\ \ P^t_{(i\pm r, j\pm\ r)} \ne P_{max} \tag{7}$$

Equation (7) stated above represents rule-3 in the pseudo code of the CA algorithm described in the last part of this subsection and $r = 1$ in this equation. This equation states that the new state of the pixel under processing is the mean of all the values of pixels in the neighborhood of the center pixel in a 3 × 3 window. It means all pixels in the neighborhood have a value greater than zero and lesser than 255.

$$T_{iter} = \left(\frac{d}{10}\right) + 1 \tag{8}$$

Equation (8) states the formula for the total iterations of the CA algorithm T_{iter} required to be applied on the entire noisy input image. In this equation, d indicates the percentage of salt-pepper noise density.

Steps in the CA Algorithm:

1) Read the color RGB input image $I(x, y)$.
2) Separate three color components of the input image $I(x, y)$, namely, I_R, I_G and I_B.
3) Let us denote the center pixel in a 3 x 3 window by $P_{i,j}$ and let P denote all the pixels in neighborhood of $P_{i,j}$
4) If the center pixel $P_{i,j}$ satisfies the condition $0 < P_{i,j} < 255$, it is considered a non-corrupted pixel, and its value is unchanged.
5) Visit pixel $P_{i,j}$ at the center of the 3 × 3 window in one of the three image components.
6) If $P_{i,j}$ is a noisy pixel and its value is zero or 255 for salt pepper noise, then
   Rule 1: If $P_{min} = 0$ and $P_{max} = 255$, then $P_{i,j}$ = the mean of pixels in P, excluding the pixels with values of P_{min} and P_{max} as per Eq (5).
   Rule 2: $P_{i,j} = 255$ if all the elements in P are either zero or 255 as per Eq (6).
   Rule 3: $P_{i,j}$ = the mean of all the elements in P, if $P_{min} > 0$ and $P_{max} > 255$ as per Eq (7).
7) Repeat Steps 5 and 6 for every pixel in each of the three components of the entire noisy image $I(x, y)$ for total iterations T_{iter} as per Eq (8).
8) Combine the three denoised channels, namely, I_R, I_G and I_B to get a restored RGB image.

The CA algorithm applies Step 5 to Step 6 on each of the three-color channels of the input image $I(x, y)$ and these three channels of the denoised image are concatenated to get the final noise-free image.

2.1.2. Restoration of textures tainted with Gaussian noise

This section discusses the Color Block Matching 3D (C-BM3D) algorithm. This algorithm is applied to noisy texture image that has been contaminated by Gaussian noise. The mathematical and theoretical components of this method are fully explained in [59,60,62]. This restoration algorithm consists of two phases, both of which are briefly discussed here. Let us define some notations for the analytical description of this algorithm. Let z_{rgb} denote noisy color image and noisy YUV color space image is denoted by z_{YUV}. The variance of red, green and blue components of the noisy image is denoted by σ_R^2, σ_G^2 and σ_B^2 respectively. True noise-free image is denoted by y_{rgb}. We denote 3D chunks in YUV color space components of the noisy image by z_Y, z_U and $z_{V,}$ respectively in both phases of this algorithm. The noise model of the image is represented as $z_{rgb} = y_{rgb} + \eta_{rgb}$, where η_{rgb} is Gaussian noise. The 3D chunks obtained from the initial denoised image in the second phase of this algorithm are denoted by $\hat{y}_Y^{initial}$, $\hat{y}_U^{initial}$ and $\hat{y}_V^{initial}$.

The input-tainted RGB image z_{rgb} is first transformed into a YUV color space image z_{YUV}. A YUV color space input image is split up into 2D chunks in the first stage. Every 2D chunk is treated as a reference block R, as shown in Figure 1. The sets of 3D chunks are created in an image by grouping 2D chunks that resemble a reference chunk based on ℓ^2 distance [59,60]. Figure 2 shows the grouping of 2D chunks to get 3D chunks on an artificial image. This figure simplifies how the grouping of 2D chunks is performed to get 3D chunks.

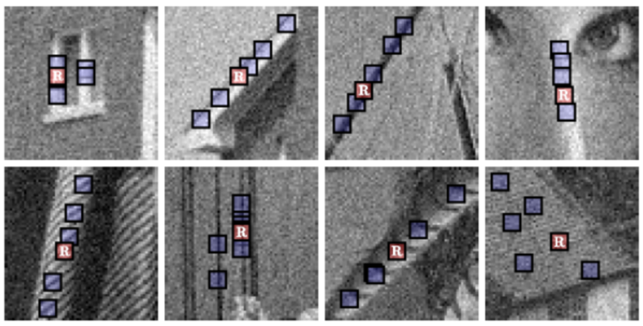

Figure 1. Chunks grouping process in a noisy image.

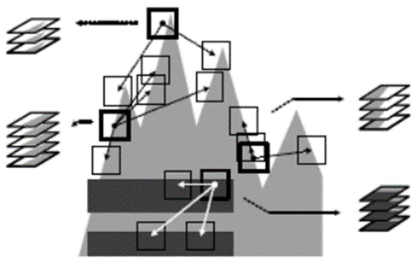

Figure 2. Similar chunks grouping in artificial image.

The remainder part of the present section discusses the analytical details of this algorithm in brief. More subtle mathematical details are presented in [59,60]. The formula for weighted average in the first phase of this algorithm is given in Eq (9) as follows.

$$\omega_{x_R}^{ht} = \begin{cases} \frac{1}{\sigma^2 N_{har}^{x_R}}, & if \quad N_{har}^{x_R} \geq 1 \\ 1, & otherwise \end{cases} \tag{9}$$

where $\omega_{x_R}^{ht}$ denotes the weight used in the first phase of this algorithm, and the hard thresholding operation in the first phase is denoted by ht. The left-top coordinate of the reference chunk is denoted by x_R. The Gaussian noise variance is denoted by σ^2. Dimension of reference chunk with the left-top coordinate x_R is $N_{har}^{x_R}$.

The formula for weighted average in the second phase of this algorithm is given in Eq (10) as follows.

$$\omega_{x_R}^{wie} = \sigma^{-2} \left\| W_{S_{x_R}^{wie}} \right\|_2^{-2} \tag{10}$$

where weight used in the second phase is denoted by $\omega_{x_R}^{wie}$. The second phase uses the Wiener filter, and wie stands for the Wiener filter in the above equation. The symbol used to denote ℓ^2- norms is $\|\cdot\|_2$. Gaussian noise variance is denoted in this equation by σ^{-2}. The matched chunks coordinates are stored in the set $S_{x_R}^{wie}$.

Transforms are applied on these 3D sets, namely, z_Y, z_U and z_V. Transforms used in this algorithm's phases are discrete cosine transforms or Wavelet transforms. Hard-thresholding operation is performed on the transform coefficients in the first phase to filter out noise. Inverse 3D transformation is applied on 3D chunks, followed by a weighted averaging of overlaying 2D chunks to obtain an initial denoised image using the formula in Eq (9).

Six 3D chunks are created in the second phase of this algorithm. A 2D chunk matching procedure is applied on the original texture image contaminated with noise to yield three 3D chunks, namely, z_Y, z_U and z_V out of the total six 3D chunks. A 2D chunk matching procedure is applied to the three denoised image components obtained in the first phase, namely, $\hat{y}_Y^{initial}$, $\hat{y}_U^{initial}$ and $\hat{y}_V^{initial}$ to yield the remaining three 3D chunks. In the second phase, noise is filtered by replacing the hard-thresholding operation with the Wiener filter. The remaining restoration procedure is the same as the first phase of this algorithm. Steps in both phases of this algorithm are described as follows.

Steps in the First Phase of the C-BM3D Algorithm:

1) Transform the color RGB input image z_{rgb} to YUV color space z_{YUV} image.
2) Divide each of the three components of the YUV image into 2D chunks.
3) Each of the total 2D chunks is treated as a reference chunk, and the set of 2D chunks similar to the reference chunk is put together to get a 3D set of similar 2D chunks, namely $z_{Y,}$ z_U and z_V.
4) Apply transforms such as Wavelet (db2, db4, db6) or DCT on each of the 3D sets obtained in step-3.
5) Apply hard-thresholding operation on transform coefficients to filter noise.
6) Compute inverse 3D transform of 3D chunks obtained in Step 5.
7) Apply the weighted averaging formula in Eq (9) on overlapping 2D chunks to get the initial restored estimate of the three YUV image components, namely, $\hat{y}_Y^{initial}$, $\hat{y}_U^{initial}$ and $\hat{y}_V^{initial}$.

Steps in the Second Phase of the C-BM3D Algorithm:

1) Divide each of the three components of the noisy YUV image z_{YUV} into 2D chunks.
2) Each of the total 2D chunks is treated as a reference chunk, and the set of 2D chunks similar to the reference chunk are put together to get three 3D sets of similar 2D chunks, namely $z_{Y,}$ z_U and z_V.
3) Divide each of the three components of the denoised image $\hat{y}^{initial}$ obtained in the first phase into 2D chunks.
4) Each of the total 2D chunks in three components is treated as a reference chunk, and the set of 2D chunks similar to the reference chunk are put together to get another set of three 3D sets of similar 2D chunks, namely, $\hat{y}_Y^{initial}$, $\hat{y}_U^{initial}$ and $\hat{y}_V^{initial}$.

5) Apply transforms such as Wavelet (db2, db4, db6) or DCT on each of the six 3D sets obtained in Steps 2 and 4.
6) Apply Wiener filter on transform coefficients $\omega_{x_R}^{wie}$ in Eq (10) to filter noise.
7) Compute inverse 3D transform of six 3D chunks obtained in Step 6.
8) Apply the weighted averaging formula in Eq (10) on overlapping 2D chunks to get the final restored estimate of the noisy YUV image z_{YUV}.
9) Convert YUV denoised image $\hat{y}$ into an RGB image.

2.2. MRF based segmentation phase

The segmentation problem formulation is discussed in the first part of this section. The label energy component in the segmentation model is described in the second subsection, followed by the feature energy component description in the third subsection.

2.2.1. Segmentation problem formulation in MRF framework

Let us define some of the notations used in formulating the segmentation model. Let $S = \{s_1, s_2, \dots, s_m\}$, where S is the 2D image array and m is the total number of pixels in S. The feature vector for each pixel s is denoted by y_s^k. Herein, k denotes k^{th} feature vector in the feature space. The total number of feature vectors is denoted by n. We denote the feature random field by Y and $Y = \{y_s^k | s \in S\}$. We denote a label assigned to a pixel s after segmentation by x_s and the total number of classes in the image by L. The set containing unique labels in the entire image to be segmented is $\Lambda = \{1, 2, \dots, L\}$. The set of entire labels on S forms a random field $x = \{x_s, s \in S\}$. Optimal labelling obtained after segmentation is $\hat{x}$ and it is obtained by maximizing posterior probability $P(x|Y)$.

$$\hat{x} = \arg\max_{x \in \Omega} P(Y|x)\, P(x) \tag{11}$$

In above Eq (11), Ω is a set containing entire possible labelling configurations. In MRF-based segmentation, the posterior probability $P(x|Y)$ of a pixel s belonging to a class in a 2D image array is maximized. However, maximizing posterior probability in terms of conditional probability $P(Y|x)$ is mathematically not tractable. Therefore, the task of maximizing posterior probability is transformed into minimizing an energy function [69] using an optimization algorithm, namely, simulated annealing in this study. The Gibbs distribution and Bayesian framework are used to formulate the model-based segmentation problem as an energy function [28,30,69]. Label energy and feature energy are the two energy terms that make up the total energy for this model. Here, the total energy is denoted by $U(x, Y)$. The sum of label energy $U_l(x)$ and feature energy $U_f(Y)$ gives total energy $U(x, Y)$.

2.2.2 Label energy component

Hammersley Clifford's hypothesis states that if x is MRF, it is also Gibb's random field (GRF) [69], and it follows Gibbs distribution indicated in Eq (12) as follows.

$$P(x) = \frac{1}{Z}\ exp^{\left(-\ U_l(x)\right)} = \frac{1}{Z}\ \left(exp^{\left(-\sum_{c\ \in\ \varsigma} V_c(x_C)\right)}\right) \tag{12}$$

where $P(x)$ is prior labelling probability, and Z is the normalization constant. $U_l(x)$ is label energy and V_C stands for clique potential of a clique C in 4-neighborhood. x_C is the label configuration for the clique C. The entire set of cliques in the 4-neighborhood is denoted by ς.

$$V_C = \ \delta(x_s, x_t) = \begin{cases} +1 & if\ \ x_s \neq \ x_t \\ \ -1 & otherwise \end{cases} \tag{13}$$

Equation (13) simplifies Eq (12) for the prior probability component to the following equation.

$$P(x) = \ \frac{1}{Z}\exp\left(-\ \textstyle\sum_{\{s,t\}\in C} \delta\left(x_s\ ,x_t\right)\right) \tag{14}$$

Equation (14) can be transformed to label energy U_l as follows.

$$U_l(x) = \ \beta \textstyle\sum_{\{s,t\}\in C} \delta\left(x_s\ ,x_t\right) \tag{15}$$

where β is constant, and it specifies weight for the contribution of prior components. Its value usually is $\beta \geq 2.0$.

2.2.3 Feature energy component

Input texture image is converted to Luv color space image, and it is used as feature image. This feature data is multivariate, and its probability $P(Y|x_s)$ follows multivariate Gaussian distribution specified in Eq (16) as follows.

$$N(\mu_m, \Sigma_m) = \ \frac{1}{\sqrt{(2\pi)^n|\Sigma_m|}}\ exp\left(-\ \frac{1}{2}\ (Y - \ \mu_m)\ \Sigma_m^{-1}(Y - \ \mu_m)^T\right) \tag{16}$$

where a class in the feature image is denoted by m and the covariance of this class is denoted by $\Sigma_{m.}$ The mean of class m is μ_m. Total feature vectors are three in this study and denoted by n. A pixel in a class m can have a label in the set $\Lambda = \{1, 2, \dots, L\}$. Total classes L in the texture image can be a number with any value between three and 16 depending upon the texture benchmark image. The features of the Luv image are assumed to be independent in this study. Therefore, likelihood probability $P(Y|x)$ takes the form in Eq (17) as follows.

$$P(Y|x) = \ \textstyle\prod_{s\ \in S} P(y_s^k|x_s) = \ \prod_{s\ \in S} \frac{1}{\sqrt{(2\pi)^n|\Sigma_{x_s}|}} \exp\left(-\frac{1}{2}\ \left(y_s^k - \ \mu_{x_s}\right)\ \left(y_s^k - \ \mu_{x_s}\right)^T\right) \tag{17}$$

Equation (17) can be transformed to feature energy of entire image $U_f(Y)$ as follows.

$$U_f(Y) = \ \textstyle\sum_{s\ \in S}\left(\ln\left(\sqrt{(2\pi)^n\left|\Sigma_{x_s}\right|}\right) + \ \frac{1}{2}\ \left(y_s^k - \ \mu_{x_s}\right)\ \Sigma_{x_s}^{-1}\left(y_s^k - \ \mu_{x_s}\right)^T\right) \tag{18}$$

Total energy is sum of Eqs (15) and (18)

$$U(x,Y) = \sum_{s\,\in S}\left(\ln\left(\sqrt{(2\pi)^n\left|\Sigma_{x_s}\right|}\right) + \frac{1}{2}\left(y_s^k - \mu_{x_s}\right)\Sigma_{x_s}^{-1}\left(y_s^k - \mu_{x_s}\right)^T\right) + \beta\sum_{\{s,r\}\in C}\delta(x_s,\ x_t)(19)$$

where Y represents features of the entire image and y_s^k is k^{th} feature for pixel s.

Total energy $U(x,Y)$ is minimized using simulated annealing to get segmentation results using MRF in the second phase. Minimizing energy means maximizing posterior probability $P(x|Y)$. A small portion of each texture segment is chopped to estimate MRF parameters, namely, covariance and mean. The segmentation results obtained using MRF and Customized Median Filter are discussed together for the sake of simplicity in the next subsection.

2.3. The proposed segmentation approach using MRF and customized median filter

The energy in Eq (19) is minimized using simulated annealing to get segmentation results on the restored input image using MRF in the second phase of the proposed approach. The first image on the left most side in the first row in Figure 3 is a restored texture benchmark image in the Prague benchmark database. The second image in the first row is the ground truth of this restored image. The third image in the first row from the left side in the first row of this figure is the result obtained by minimizing energy in the MRF model.

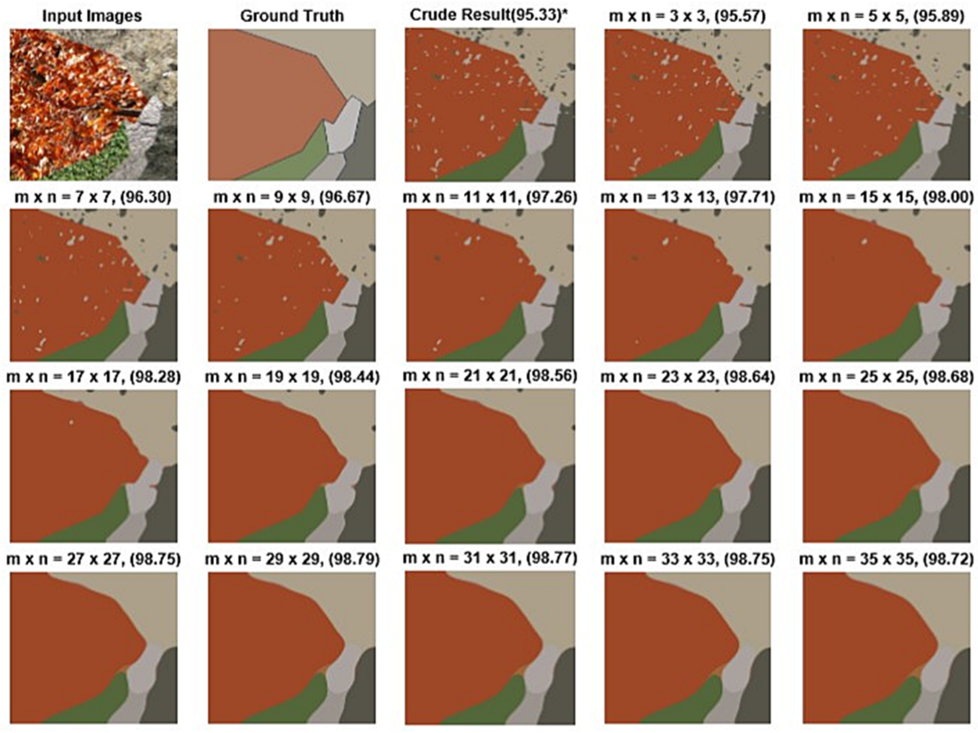

Figure 3. Qualitative results of the second and third phases of the proposed method obtained using MRF and CMF. (*Note: Dimensions of filter window are denoted by m and n. The number in parentheses shows segmentation accuracy.)

When the machine segmented result accuracy is 100%, the segmentation result will be the same as the ground truth. However, the segmented output obtained by minimizing energy using MRF has sharp boundaries, and it contains small dots and tiny islands. The aim of the proposed approach is to obtain the highest accuracy on the restored texture image. The segmented result will be close to the ground truth if we remove dots and tiny islands, ideally without changing the boundaries in the segmented result. The Customized Median Filter (CMF) objectively improves segmented results obtained by minimizing energy in the MRF model to reach close to the ground truth using segmentation performance parameters, namely, SSIM, MSE and PSNR. Each segmentation performance parameter objectively determines the closeness of the segmented result to ground truth. As CMF is applied to the result obtained in the second phase of the proposed technique (result obtained by minimizing energy in MRF model), iteratively with odd dimensions of CMF window results, go on improving. The dimensions of the CMF window are denoted by $m \times n$ in Figure 3.

The segmentation performance parameters are evaluated for every iteration of CMF. The dots and tiny islands go on disappearing with an increase in iterations, and the segmentation result improves over the iterations. The optimal segmented result is achieved for the iteration and CMF window dimensions for which segmentation performance parameters, namely, SSIM, MSE and PSNR, reach close to their ideal values. Each segmentation performance parameter value is estimated using ground truth and the segmented result obtained at every iteration of CMF. When the segmented result reaches objectively close to ground truth, the SSIM parameter value will be close to 1, the MSE value will be close to zero, and PSNR will be very high. This study achieves objective closeness of machine-segmented results to ground truth using CMF and segmentation performance parameters. As per the literature survey and to the best of the authors' knowledge, the Median filter is yet not customized based on segmentation performance parameters to obtain optimal segmentation results.

Table 1. CMF window dimensions estimation using SSIM, MSE and PSNR quantitative parameters.

Window Size (m × n)	SSIM	MSE	PSNR (dB)	Accuracy
3 × 3	0.8881	0.1489	21.12	95.57
5 × 5	0.8994	0.1295	21.64	95.89
7 × 7	0.9142	0.1049	22.34	96.30
9 × 9	0.9290	0.0795	23.33	96.76
11 × 11	0.9417	0.0619	24.56	97.26
13 × 13	0.9547	0.0444	25.88	97.71
15 × 15	0.9611	0.0381	26.86	98.00
17 × 17	0.9663	0.0346	28.13	98.28
19 × 19	0.9696	0.0326	28.71	98.44
21 × 21	0.9714	0.0317	29.11	98.56
23 × 23	0.9723	0.0314	29.43	98.64
25 × 25	0.9725	0.0340	29.43	98.68
27 × 27	0.9729	0.0367	29.26	98.75
29 × 29	0.9733	0.0384	29.03	98.79
31 × 31	0.9726	0.0399	28.80	98.77
33 × 33	0.9717	0.0411	28.48	98.75
35 × 35	0.9706	0.0417	28.21	98.72

As reported in [70] performance of the SSIM metric is better than MSE and PSNR. SSIM considers structural information and boundaries while objectively determining the closeness between the segmentation result of an image and its ground truth. Hence, optimal segmentation of texture image is achieved when SSIM reaches its expected ideal value of one. The largest value of SSIM is 0.9733, and it reaches for the window having dimensions of 29 by 29. The segmentation output for these dimensions of CMF is optimal. A quantitative metric developed by Hoover in [71] is used for accurate measurement, and this metric is known as correct segmentation (CS). Gabor filter and local binary patterns are used as features in literature with Luv color space for color texture segmentation [72]. These feature increase feature space dimensions and hence computations. Here, CMF eliminates these features and reduces the computations needed for color texture segmentation.

The segmentation accuracies are obtained using a Customized Median Filter and MRF approach for restored six class Prague texture images which have been tainted with 20% density of salt-n-pepper noise, as shown in the rightmost column in Table 1. These accuracies are obtained based on three metrics, namely, MSE, SSIM and PSNR and different dimensions of the CMF window, as shown in Table 1. As stated previously, optimal segmentation results are achieved when SSIM reaches its expected highest value close to one. As per Table 1, the highest value of SSIM is 0.9733, and the window dimensions for this value are 29 by 29. The accuracy achieved for this value of SSIM is 98.79%. The window dimensions for the largest accuracy of 98.68% are 25 × 25 for the PSNR metric, and the value of PSNR for this case is 29.43 dB. The largest accuracy of 98.64% is obtained for the smallest value of MSE = 0.0314. Based on the discussion presented here, let us define a rule for the determining optimal window dimensions of CMF.
Rule: When CMF is applied iteratively on the segmented results obtained using the MRF model on the input image, the optimal window dimensions are those for which the segmentation performance parameter value is closest to its ideal value. For example, the ideal value of SSIM is one in this study.As shown in Table 1, the SSIM value close to its ideal value is 0.9733 for CMF window dimensions of 29 by 29. The accuracy obtained for this window dimensions is the largest, which is 98.79%.

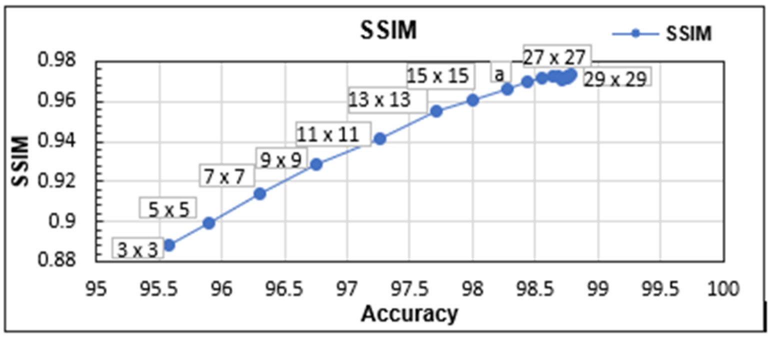

Figure 4. Plot of SSIM versus accuracy for variable window dimensions. (Note [a]: Filter window dimensions are not shown due to limited space.)

Figure 4 shows the plot of the SSIM metric against segmentation accuracy along with the dimensions of the Customized Median Filter's window. As window dimensions of CMF increase, SSIM value increases. This increases the closeness of the segmented machine result to ground truth. When SSIM reaches the most considerable value, close to one accuracy is highest. This figure indicates that SSIM reaches the largest value, close to one, for CMF window dimensions of 29 by 29.

The plot of MSE versus accuracy and the plot of PSNR versus accuracy is shown in Figures 5 and 6, respectively. These two graphs shown below describe results obtained using CMF and segmentation performance parameters, namely, MSE and PSNR.

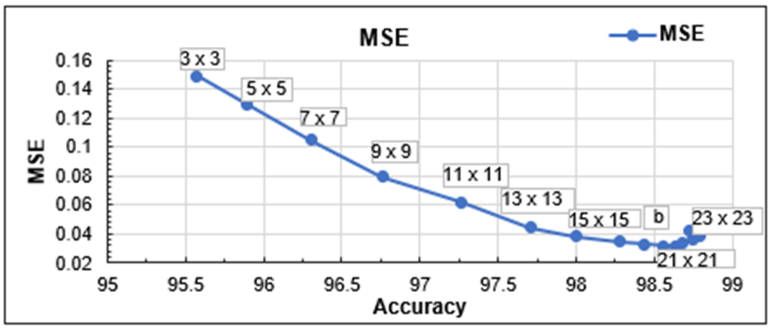

Figure 5. Plot of MSE versus accuracy for variable window dimensions. (Note ^{b}: Filter window dimensions are not shown due to limited space.)

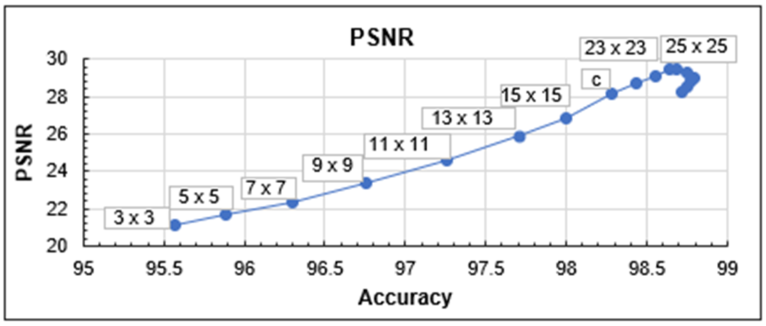

Figure 6. Plot of PSNR versus accuracy for variable window dimensions. (Note ^{c}: Filter window dimensions are not shown due to limited space.)

When the MSE value is obtained from the segmented result, and ground truth reaches the smallest value close to zero, the accuracy is highest. The smallest value of MSE = 0.0314 is reached for CMF window dimensions of 23 by 23. The PSNR value obtained using ground truth as the ideal result, and the segmented machine result is the largest for CMF window dimensions of 25 by 25. The largest PSNR value in Figure 6 is 29.43 dB, and the highest accuracy obtained for this PSNR value is 98.68%. However, optimal segmentation of texture image is achieved when SSIM reaches its expected ideal value of one, as discussed previously in this section.

3. Results and discussion

Texture benchmark images from Brodatz and Prague datasets are used to evaluate the proposed approach’s segmentation performance for the reasons discussed in the introduction section. There are two benchmark images from the Brodatz dataset and 80 benchmark images from the Prague dataset. Prague benchmark images are constructed from snipped pieces of 114 parent textures. Each of these 114 textures has 10 distinct types of textures [49]. In the Prague dataset, the lowest number of classes is 3, while the highest number is 12. The Prague benchmark is developed for evaluating the performance of unsupervised and supervised texture segmentation approaches.

The benchmark images from the Brodatz dataset are deteriorated with zero mean Gaussian noise with several steps of variance values, namely, 10, 20, 30, 40 and 50. These tainted textures are restored using the C-BM3D algorithm. The graphs for restoration performance in terms of SSIM metric and PSNR metric for this algorithm are shown in Figures 7 and 8, respectively, for Brodatz texture images. The SSIM metric values are very near to the expected value of one. The PSNR metric values are also very high, as expected. The PSNR metric value is 28.84 dB for a variance value of 50, as shown in Figure 8. The proposed color texture segmentation approach is applied to noisy Bordatz textures and the results are discussed in the following part of this section.

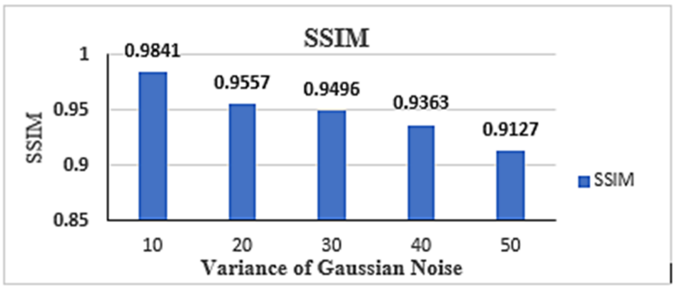

Figure 7. Denoising results in terms of SSIM metric on Brodatz images for Gaussian noise.

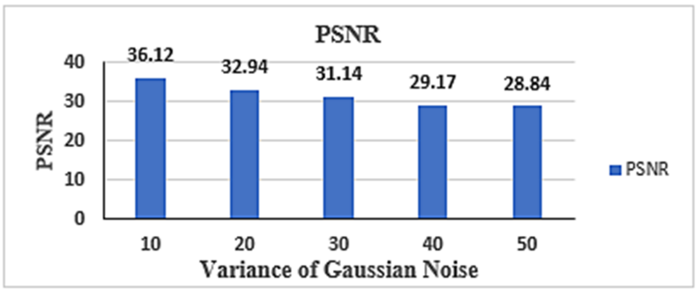

Figure 8. Denoising results in terms of PSNR metric on Brodatz images for Gaussian noise.

Table 2. Segmentation performance on 16-class Brodatz texture image.

Noise type	Noise parameter	Approach	Accuracy (CS)	SSIM
None	None	MRF without noise	98.82	NP[d]
Salt-Pepper	20% noise density	MRF with CA	98.62	0.9544
	70% noise density	MRF with CA	98.54	0.8950
Gaussian	Variance 10	MRF with C-BM3D	98.14	0.9060
	Variance 50	MRF with C-BM3D	97.60	0.8288
No noise	None	Deep learning using U-net [46]	82.50	NP

**Note* [d]: NP stands for Not Provided for this algorithm.

The performance of the proposed segmentation approach on the restored 16-class Brodatz image is shown in Table 2. The segmentation performance metrics used here are the CS metric and the SSIM metric. The CS metric computes the percentage of correctly labeled pixels using ground truth and SSIM metrics evaluate the closeness of ground truth to segmented result.

The accuracy achieved on this benchmark image without degrading it with noise using a deep learning algorithm with U-net is 82.50% [46]. The accuracy achieved on the restored benchmark image contaminated with Gaussian noise having variance 50 is 97.60% and the accuracy achieved on the same benchmark image restored after tainting with 70% salt-n-pepper noise density is 98.54%. These higher values of the accuracies are achieved even though there is a loss of information in the restored image. The SSIM metric values are near to the expected value of one. This indicates that the segmentation performance of the proposed approach in terms of CS and SSIM metrics is robust.

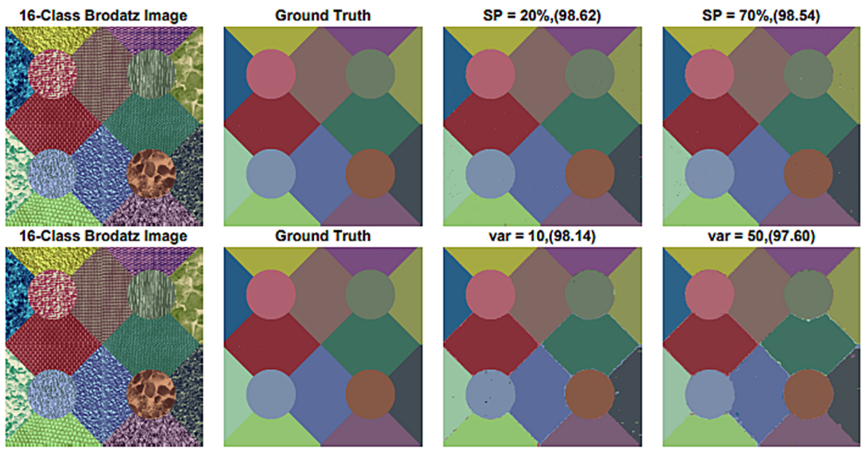

Figure 9. Segmentation performance on the restored 16-Class Brodatz image. (Note: The number in parentheses shows segmentation accuracy.)

The segmentation performance on the restored 16-class Brodatz image is shown in Figure 9. This figure shows that all texture segments are separated nicely even though the information is lost during the restoration process for both Gaussian and salt-n-pepper noise. Qualitative segmentation results are not provided for this image using U-net in [46].

Table 3. Segmentation performance on the restored five-Class Brodatz image.

Noise type	Noise parameter	Approach	Accuracy (CS)	SSIM
None	None	MRF without noise	98.82	NP[e]
Salt-Pepper	20% noise density	MRF with CA	98.70	0.9571
	70% noise density	MRF with CA	98.65	0.8991
Gaussian	Variance 10	MRF with C-BM3D	98.16	0.9314
	Variance 50	MRF with C-BM3D	97.80	0.8748
No noise	None	Deep learning using U-net [46]	97.40	NP

**Note* [e]: NP stands for Not Provided for this algorithm.

The segmentation performance of the approach developed in this study on 5-class Brodatz texture images is shown in Table 3. The segmentation accuracy achieved on this restored image which was tainted with 70% salt-n-pepper noise, is 98.65%, and the accuracy obtained on the same restored image, which was tainted with Gaussian noise, is 97.80%, and it is higher than recent deep learning algorithm using U-net suggested in [46].

PSNR and SSIM are used as metrics to measure the performance of the CA algorithm. Table 4 demonstrates the quantitative denoising results of this algorithm on 80 Prague texture images in terms

Table 4. Quantitative denoising results of the CA algorithm on Prague textures.

Noise type	Noise parameter	Algorithm	SSIM	PSNR
Salt-n-Pepper Noise	20% noise density	CA	0.9859	35.48 dB
Salt-n-Pepper Noise	70% noise density	CA	0.9043	27.19 dB

of mean values of these metrics. This table indicates that even with a salt-n-pepper noise density of 70% the value of the SSIM metric is 0.9043, and it is very near to its expected value of one. The PSNR value for 70% salt-n-pepper noise density is 27.19 dB, and it is very large.

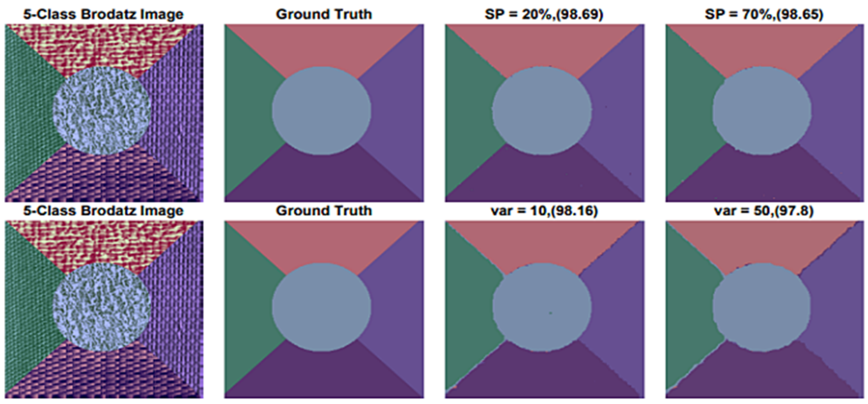

Figure 10. Segmentation performance on 5-Class Brodatz image. (Note: Segmentation accuracy is shown by the number in parentheses.)

Figure 10 demonstrates subjective segmentation outputs for a five class Brodatz image. This figure shows that classes are nicely separated with excellent boundary performance even though the information is lost during the restoration process. This indicates the robust performance of the segmentation approach developed in this study.

PSNR and SSIM are used as metrics to measure the performance of the C-BM3D algorithm. Figures 11 and 12 demonstrate the quantitative denoising results of this algorithm on 80 Prague textures in terms of mean values of PSNR and SSIM, respectively.

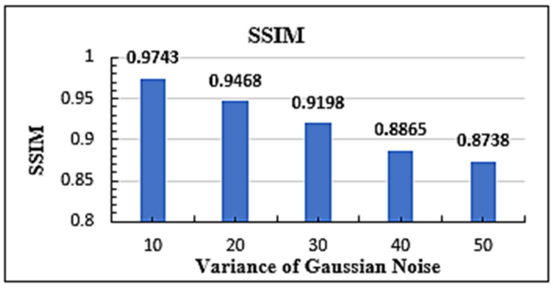

Figure 11. Denoising results in terms of SSIM metric on Prague textures for Gaussian noise.

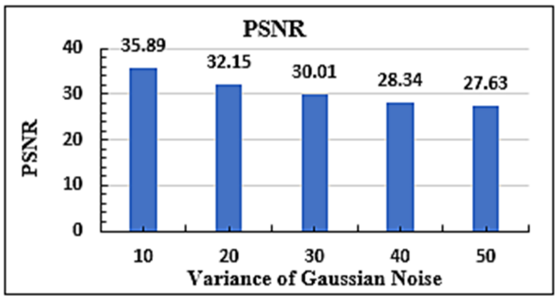

Figure 12. Denoising results in terms of PSNR metric on Prague textures for Gaussian noise.

All 80 images were tainted with Gaussian noise. As shown in these figures, five values of variance are used to degrade Prague texture images. SSIM metric values for all the variance values are very near to their expected value of one. PSNR metric values are also very high for these five variance values. The results of segmentation obtained on 80 Prague texture images in terms of CS and SSIM metrics are shown in Table 5. This table indicates that the segmentation performance of the approach developed here is better than three recent approaches reported in [65], [31] and [45]. The most recent approach in [65] achieves 91.27% accuracy on noise-free textures. The proposed approach achieves 95.70% accuracy on noise-free textures. This approach achieves accuracies of 95.35 and 92.78% on textures contaminated with Gaussian noise having variances of 10 and 50, respectively. These accuracies are higher than the most recent approach in [65] even after degradation by Gaussian noise. The accuracy of the proposed technique is 93.74% on textures contaminated by 20% salt-n-pepper noise and is higher than the most recent technique given in [65]. The SSIM metric quantitatively measures the degree of closeness of ground truth to the segmentation result. The mean value of the SSIM metric is near to its expected value of one. This indicates excellent segmentation performance in terms of the SSIM metric.

Table 5. Segmentation results on the restored texture benchmark images from the Prague texture dataset.

Noise Type	Noise Parameter	Approach	Accuracy (CS)	SSIM
None	None	MRF without noise	95.70	0.9140
Salt-Pepper	20% noise density	MRF with CA	93.74	0.8838
	70% noise density	MRF with CA	89.71	0.8495
Gaussian	Variance 10	MRF with C-BM3D	95.35	0.8788
	Variance 50	MRF with C-BM3D	92.78	0.8682
No noise	None	FCAM [65]	91.27	NP[f]
No noise	None	EWT-FCNT [47]	98.78	NP
No noise	None	Kiechle [31]	77.73	NP
No noise	None	Deep learning using FCNT [45]	79.34	NP

**Note* [f]: NP stands for not Provided for these two algorithms.

A comparison of the subjective segmentation output of the proposed technique with four modern techniques is shown in Figure 13.

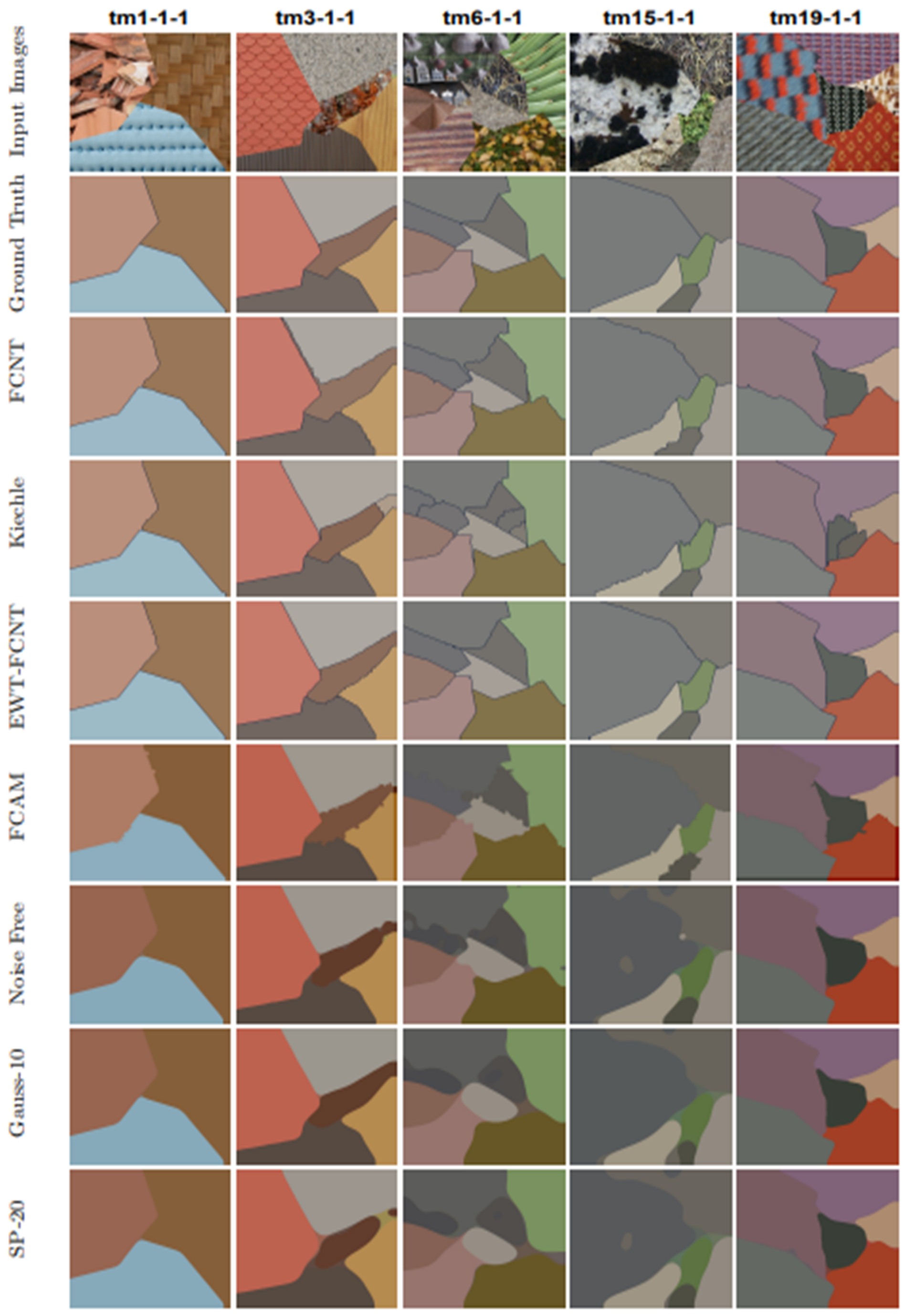

Figure 13. Comparison of segmentation outputs of the proposed technique with contemporary techniques.

The row named FCNT (Fully Convolutional Network) shows the results obtained using FCNT algorithm in [45]. The fourth row with the name Kiechle shows the result obtained using the recent approach in [31]. The fifth row, named EWT-FCNT (Empirical Wavelet Transform-FCNT) are the result of the approach in [47]. The sixth row with the name FCAM shows the results of the approach in [65]. The remaining next three rows show subjective performance using the proposed technique. The row labeled noise-free shows results obtained on noise free Prague textures. The segmentation performance on these six restored images which were contaminated with Gaussian noise having variance ten is shown in the row named Gaussian-10. The row named SP Noise-20 shows the segmentation result obtained on the six restored images which were tainted with 20% salt-n-paper noise density. The technique presented in this study delivers better boundary performance than the techniques in [65], [31] and [45]. The segmentation performance for restored images, namely, tm1-1-1, tm3-1-1 and tm19-1-1 in the second and third columns is better than the recent approaches in [65], [45] and [31] despite information loss during the restoration of noisy images. The segmentation performance for the other three restored images is better for some texture segments in these images and the results for other texture segments are degraded. The reasons for the degradation of results are discussed in the last part of this section.

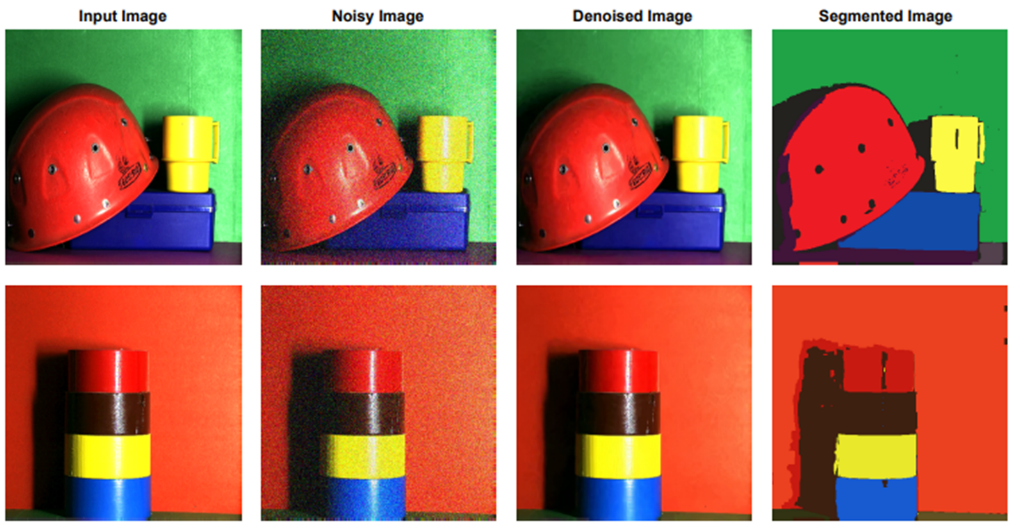

Figure 14. Results of the proposed technique on natural color images.

The technique presented in this paper is evaluated and tested on noisy natural color images from the color image dataset developed at Simon Fraser University to prove the robustness of this technique. Figure 14 shows two natural color images in the first column, and the second column of this figure shows these images contaminated by Gaussian noise having a variance of 20. Restored images are shown in the third column, and segmented denoised images are shown in the fourth column. The upper input image contains three objects: one cap with holes, one mug with a handle, and a blue box. These three objects are appropriately segmented along with their boundaries. The holes on the cap and handle of the mug are adequately detected. However, the shadows of the objects are undesirable and the vertical black line due to the reflection effect on the upper portion of the mug is also undesirable. The bottom input image contains four color objects and these objects are correctly segmented along with their boundaries. However, undesirable effects of shadow and reflection exist in the segmented image. These effects can be rectified by capturing images under proper lighting conditions or by using an advanced algorithm in the pre-processing stage after capturing images.

The segmentation approach developed in this study offers high segmentation accuracy than recent approaches, namely, FCAM in [65], the deep learning approach with U-net in [46], the approach in [31] and FCNT in [45]. It is concluded from visual inspection of color benchmark textures in the Brodatz and Prague datasets that most of these benchmark images contain texture classes with distinct colors and a very small variation of color in a texture segment. Therefore, the classes in these benchmark images can be separated precisely based on color features. When the texture image is transformed to Luv color space, discrimination between texture classes is robust and hence precise boundary detection is possible because Luv space is uniform Euclidean space [73]. Variation in the color shade in a texture class leaves dots and tiny islands in the segmented images when only the Luv image is used as a feature. These dots and tiny islands are removed using a Customized Median Filter. This filter allows homogeneous separation of texture classes. Thus, Luv color features and the Customized Median Filter together deliver high segmentation accuracy. The tainted texture images are restored before applying a segmentation algorithm to them. Information loss in restored images degrades segmentation performance because SSIM and PSNR metrics values reduce due to information loss in the texture images.

If two different colors exist in one texture segment, the segmentation approach developed herein will not be able to treat it as a single class because the proposed algorithm uses color as features. In the texture image named tm15-1-1 in Figure 13, left most class contains two colors, namely, white and black in a single texture class and both white and black color exists on the boundaries of this class. Therefore, boundary detection performance degrades for this image along with accuracy. In texture image named tm6-1-1 in Figure 13 contain two texture segments at the left-top part with very close color shades. Therefore, these two textures tend to merge and degrade accuracy along with boundary detection performance. The existence of more than one color in a single class degrades color distance estimation among classes to discriminate them in Luv color space. These reasons are applicable to all texture images for the degradation of segmentation performance.

4. Conclusions and future directions

When compared to the four latest methods for both noise-free and noise-contaminated textures, the authors were able to achieve segmentation accuracy greater than the three latest techniques, on both Brodatz and Prague benchmark textures, using the proposed three-phase technique. When the proposed approach is evaluated on Brodatz textures, an improvement of up to 16% segmentation accuracy for salt-n-pepper noise with 70% noise density and 15.1% accuracy for Gaussian noise (with a variance of 50) has been made in comparison with the benchmark approaches. On Prague textures, accuracy is improved by 4.08% for Gaussian noise (with variance 10) and by 2.47% for salt-n-pepper noise with 20% noise density. Higher accuracies could be achieved on highly contaminated texture images due to the superior performance of the proposed technique achieved using a Customized Median Filter and excellent performance of restoration algorithms used herein. Another metric used to evaluate segmentation results is SSIM and it is concluded that as this metric degrades, segmentation accuracy exhibits a declining trend. The approach developed as part of this study has applications in retrieving textures contaminated with noise, analyzing satellite and medical images, industrial inspections and geo-informatics, etc. As a future investigation for this technique, texture images can be tainted with Poisson and Speckle noise or the combinations of various other noise types, including salt-n-pepper and Gaussian noise.

Acknowledgments

The presentation of this manuscript as a preprint version is available in the following link: https://www.researchsquare.com/article/rs-1622365/v1

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. H. Liu, G. Huo, Q. Li, X. Guan, M. L. Tseng, Multiscale lightweight 3D segmentation algorithm with attention mechanism: Brain tumor image segmentation, *Expert Syst. Appl.*, **214** (2023), 119166. https://doi.org/10.1016/j.eswa.2022.119166
2. X. Li, S. Chen, J. Wu, J. Li, T. Wang, J. Tang, et al., Satellite cloud image segmentation based on lightweight convolutional neural network, *Plos One*, **18** (2023), e0280408. https://doi.org/10.1371/journal.pone.0280408
3. L. Guo, P. Shi, L. Chen, C. Chen, W. Ding, Pixel and region level information fusion in membership regularized fuzzy clustering for image segmentation, *Inf. Fusion*, **92** (2023), 479–497. https://doi.org/10.1016/j.inffus.2022.12.008
4. R. M. Haralick, K. Shanmugan, I. Dinstein, Textural features for image classification, *IEEE Trans. Syst. Man Cybern.*, **3** (1973), 610–621. https://doi.org/10.1109/TSMC.1973.4309314
5. J. Chaki, N. Dey, *Texture Feature Extraction Techniques for Image Recognition*, Springer, Singapore, (2020).
6. A. Distante, C. Distante, *Handbook of Image Processing and Computer Vision: Volume 3: From Pattern to Object*, Springer International Publishing, Basel, Switzerland, (2020).
7. C. C. Hung, E. Song, Y. Lan, *Image Texture Analysis. Foundations, Models and Algorithms*, Springer International Publishing, Basel, Switzerland, (2019).
8. Y. Chen, E. R. Dougherty, Gray-scale morphological granulometric texture classification, *Opt. Eng*., **33** (1994), 2713–2722. https://doi.org/10.1117/12.173552
9. B. B. Chaudhuri, N. Sarkar, Texture segmentation using fractal dimension, *IEEE Trans. Pattern Anal. Mach. Intell.*, **17** (1995), 72–77. https://doi.org/10.1109/34.368149
10. J. M. Keller, S. Chen, R. M. Crownover, Texture description and segmentation through fractal geometry, *Comput. Vis. Graph. Image Process*., **45** (1989), 150–166. https://doi.org/10.1016/0734-189X(89)90130-8
11. R. Chellappa, S. Chatterjee, Classification of textures using Gaussian Markov random fields, *IEEE Trans. Acoust. Speech Signal Process.*, **33** (1985), 959–963. https://doi.org/10.1109/TASSP.1985.1164641
12. G. R.Cross, A. K. Jain, Markov random field texture models, *IEEE Trans. Pattern Anal. Mach. Intell.*, (1983), 25–39. https://doi.org/10.1109/tpami.1983.4767341.
13. L. Alparone, F. Argenti, G. Benelli, Fast calculation of co-occurrence matrix parameters for image segmentation, *Electron. Lett.*, **26** (1990), 23–24.
14. R. Davarzani, S. Mozaffari, K. Yaghmaie, Scale-and rotation-invariant texture description with improved local binary pattern features, *Signal Process.*, **111** (2015), 274–293. https://doi.org/10.1016/j.sigpro.2014.11.005.
15. C. C. Gotlieb, H. E. Kreyszig, Texture descriptors based on co-occurrence matrices, *Comput. Vis. Graph. Image Process.*, **51** (1990), 70–86. https://doi.org/10.1016/S0734-189X(05)80063-5

16. M. Topi, O. Timo, P. Matti, S. Maricor, Robust texture classification by subsets of local binary patterns, in *Proceedings of the 15th International Conference on Pattern Recognition ICPR-2000*, Barcelona, Spain, **3** (2000), 935–938. https://doi.org/10.1109/ICPR.2000.903698
17. V. Durgamahanthi, R. Rangaswami, C. Gomathy, A. C. J. Victor, Texture analysis using wavelet-based multiresolution autoregressive model: Application to brain cancer histopathology, *J. Med. Imaging Health Inf.*, **7** (2017), 1188–1195.
18. L. D. Jacobson, H. Wechsler, Joint Spatial/Spatial frequency representations, *Signal Process.*, **14** (1988), 37–68. https://doi.org/10.1016/0165-1684(88)90043-6
19. C. S. Lu, P. C. Chung, C. F. Chen, Unsupervised texture segmentation via wavelet transform, *Pattern Recognit.*, **30** (1997), 729–742. https://doi.org/10.1016/S0031-3203(96)00116-1
20. T. Randen, J. H. Husoy, Filtering for texture classification:A comparative study, *IEEE Trans. Pattern Anal. Mach. Intell.*, **21** (1999), 291–310. https://doi.org/10.1109/34.761261
21. K. J. Laws, Textured image segmentation, in *University of Southern California Los Angeles Image Processing INST*, (1980).
22. D. A. Clausi, K-means Iterative Fisher (KIF) unsupervised clustering algorithm applied to image texture segmentation, *Pattern Recognit.*, **35** (2002), 1959–1972. https://doi.org/10.1016/S0031-3203(01)00138-8
23. A. K. Jain, F. Farrokhnia, Unsupervised texture segmentation using Gabor filters, *Pattern Recognit.*, **24** (1991), 1167–1186. https://doi.org/10.1016/0031-3203(91)90143-S
24. S. Kinge, B. S. Rani, Mukul Sutaone, A multi-class fisher linear discriminant approach for the improvement in the accuracy of complex texture discrimination, *Helix*, **9** (2019), 5108–5121. https://doi.org/10.29042/2019-5108-5121
25. N. Senin, R. K. Leach, S. Pini, L. A. Blunt, Texture-based segmentation with Gabor filters, wavelet and pyramid decompositions for extracting individual surface features from areal surface topography maps, *Meas. Sci. Technol.*, **26** (2015), 095405. https://doi.org/10.1088/0957-0233/26/9/095405
26. M. Tuceryan, A. K. Jain, Texture Segmentation using voronoi polygons, *IEEE Trans. Pattern Anal. Mach. Intell.*, **12** (1990), 211–216. https://doi.org/10.1109/34.44407
27. G. Matheron, *Random Sets and Integral Geometry*, Wiley Publications, New York, (1975).
28. H. Deng, D. A. Clausi, Unsupervised image segmentation using a simple MRF model with a new implementation scheme, *Pattern Recognit.*, **37** (2004), 2323–2335. https://doi.org/10.1016/j.patcog.2004.04.015
29. A. K. Qin, D. A. Clausi, Multivariate image segmentation using semantic region growing with adaptive edge penalty, *IEEE Trans. Image Process.*, **19** (2010), 2157–2170. https://doi.org/10.1109/TIP.2010.2045708
30. Z. Kato, H. T. C. Pong, A Markov random field image segmentation model for color textured images, *Image Vision Comput.*, **24** (2006), 1103–1114. https://doi.org/10.1016/j.imavis.2006.03.005
31. M. Kiechle, M. Storath, A. Weinmann, M. Kleinsteuber, Model-based learning of local image features for unsupervised texture segmentation, *IEEE Trans. Image Process.*, **27** (2018), 1994–2007. https://doi.org/10.1109/TIP.2018.2792904
32. M. Pereyra, S. McLaughlin, Fast unsupervised Bayesian image segmentation with adaptive spatial regularisation, *IEEE Trans. Image Process.*, **26** (2017), 2577–2587. https://doi.org/10.1109/TIP.2017.2675165

33. L. Gatys, A. S. Ecker, M. Bethge, Texture synthesis using convolutional neural networks, *Adv. Neural Inf. Process. Syst.*, (2015), 262–270.
34. X. Snelgrove, High-resolution multi-scale neural texture synthesis, in *SIGGRAPH Asia Technical Briefs*, (2017)*,* 1–4. https://doi.org/10.1145/3145749.3149449
35. Y. Zhou, Z. Zhu, X. Bai, D. Lischinski, D. Cohen-Or, H. Huang, Non-stationary texture synthesis by adversarial expansion, preprint, arXiv:1805.04487.
36. V. Andrearczyk, P. F. Whelan, Using filter banks in convolutional neural networks for texture classification, *Pattern Recognit. Lett.*, **84** (2016), 63–69. https://doi.org/10.1016/j.patrec.2016.08.016
37. X. Bu, Y. Wu, Z. Gao, Y. Jia, Deep convolutional network with locality and sparsity constraints for texture classification, *Pattern Recognit.*, **91** (2019), 34–46. https://doi.org/10.1016/j.patcog.2019.02.003
38. M. Cimpoi, S. Maji, I. Kokkinos, A. Vedaldi, Deep filter banks for texture recognition, description, and segmentation, *Int. J. Comput. Vis.*, **118** (2016), 65–94. https://doi.org/10.1007/s11263-015-0872-3
39. U. Dixit, A. Mishra, A. Shukla, R. Tiwari, Texture classification using convolutional neural network optimized with whale optimization algorithm, *SN Appl. Sci.*, **1** (2019). https://doi.org/10.1007/s42452-019-0678-y
40. T. Y. Lin, S. Maji, Visualizing and understanding deep texture representations, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, LasVegas, USA, (2016), 2791–2799.
41. L. Liu, J. Chen, G. Zhao, P. Fieguth, X. Chen, M. Pietikäinen, Texture classification in extreme scale variations using GANet, *IEEE Trans. Image Process.*, **28** (2019), 3910–3922. https://doi.org/10.1109/TIP.2019.2903300
42. A. Shahriari, Parametric learning of texture filters by stacked fisher autoencoders, in *Proceedings of the 2016 International Conference on Digital Image Computing: Technique and Applications (DICTA)*, Gold Coast, Australia, (2016), 1–8.
43. Y. Song, F. Zhang, Q. Li, H. Huang, L. J. O'Donnell, W. Cai, Locally-transferred fisher vectors for texture classification, in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, (2017), 4912–4920.
44. H. Zhang, J. Xue, K. Dana, Deep ten: Texture encoding network, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, (2017), 708–717.
45. V. Andrearczyk, P. F. Whelan, Texture segmentation with fully convolutional networks, preprint, arXiv:1703.05230.
46. C. Karabag, J. Verhoeven, N. Miller, C. Reyes-Aldasoro, Texture segmentation: an objective comparison between five traditional algorithms and a deep-learning U-Net architecture, *Appl. Sci.*, **9** (2019), 3900. https://doi.org/10.3390/app9183900
47. Y. Huang, F. Zhou, J. Gilles, Empirical curvelet based fully convolutional network for supervised texture image segmentation, *Neurocomputing*, **349** (2019), 31–43. https://doi.org/10.1016/j.neucom.2019.04.021
48. R. Yamada, H. Ide, N. Yudistira, T. Kurita, Texture segmentation using Siamese network and hierarchical region merging, in *Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR)*, Beijing, China, (2018), 2735–2740. https://doi.org/10.1109/ICPR.2018.8545348

49. S. Mikes, M. Haindl, Texture segmentation benchmark, *IEEE Trans. Pattern Anal. Mach. Intell.*, **41** (2021), 1–16. https://doi.org/10.1109/TPAMI.2021.3075916
50. A. Awad, Denoising images corrupted with impulse, Gaussian, or a mixture of impulse and Gaussian noise, *Eng. Sci. Technol. Int. J.*, **22** (2019), 746–753. https://doi.org/10.1016/j.jestch.2019.01.012
51. R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, Third Edition Pearson Education, 2009.
52. S. Walid, B. Xi, A neighborhood regression approach for removing multiple types of noises, *EURASIP J. Image Video Process.*, **2018** (2018). https://doi.org/10.1186/s13640-018-0259-9
53. M. Alkhatib, A. Hafiane, Robust adaptive median binary pattern for noisy texture classification and retrieval, *IEEE Trans. Image Process.*, **28** (2019), 5407–5418. https://doi.org/10.1109/TIP.2019.2916742
54. S. Dash, M. R. Senapati, Noise robust Law's filters based on fuzzy filters for texture classification, *Egypt. Inf. J.*, **21** (2020), 37–49. https://doi.org/10.1016/j.eij.2019.10.003
55. C. Vacar, J. Giovannelli, Unsupervised joint deconvolution and segmentation method for textured images:a Bayesian approach and an advanced sampling algorithm, *EURASIP J. Image Video Process.*, **2019** (2019), 1–17. https://doi.org/10.1186/s13634-018-0597-x
56. H. A. Nugroho, E. L. Frannita, I. Ardiyanto, L. Choridah, Computer aided diagnosis for thyroid cancer system based on internal and external characteristics, *J. King Saud Univ. Comput. Inf. Sci.*, **33** (2021), 329–339. https://doi.org/10.1016/j.jksuci.2019.01.007
57. A. Bhargava, A. Bansal, Fruits and vegetables quality evaluation using computer vision: A review, *J. King Saud Univ. Comput. Inf. Sci.*, **33** (2021), 243–257. https://doi.org/10.1016/j.jksuci.2018.06.002
58. S. Hossain, S. Serikawa, Texture databases—A comprehensive survey, *Pattern Recognit. Lett.*, **34** (2013), 2007–2022. https://doi.org/10.1016/j.patrec.2013.02.009
59. K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space, in *Proceeding of IEEE International Conference on Image Processing*, (2007), 313–316. https://doi.org/10.1109/ICIP.2007.4378954
60. K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3D transform-domain collaborative filtering, *IEEE Trans. Image process.*, **16** (2007), 2080–2095. https://doi.org/10.1109/TIP.2007.901238
61. L. Fan, Z. Fan, H. Fan, C. Zhang, Brief review of image denoising techniques, *Visual Comput. Ind., Biomed., Art*, **2** (2019), 1–12. https://doi.org/10.1186/s42492-019-0016-7
62. S. Kinge, B. S. Rani, Mukul sutaone, quantitative restoration of noisy colour texture segmentation benchmark images using state-of-the-art algorithm, in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, (2020), 37–42. https://doi.org/10.1109/ICICCS48265.2020.9120927
63. R. R. Nair, E. David, S. Rajagopal, A robust anisotropic diffusion filter with low arithmetic complexity for images, *EURASIP J. Image Video Process.*, **2019** (2019), 1–14. https://doi.org/10.1186/s13640-019-0444-5
64. D. Tourtounis, N. Mitianoudis, G. C. Sirakoulis, Salt-n-pepper noise filtering using cellular automata, preprint, arXiv:1708.05019.
65. Z. Haliche, K. Hammouche, O. Losson, L. Macaire, Fuzzy color aura matrices for texture image segmentation, *J. Imaging*, **8** (2022), 1–20. https://doi.org/10.3390/jimaging8090244

66. C. Bontozoglou, P. Xiao, Applications of capacitive imaging in human skin texture and hair analysis, *Appl. Sci.*, **10** (2020), 256. https://doi.org/10.3390/app10010256
67. Y. Liu, K. Xu, J. Xu, An improved MB-LBP defect recognition approach for the surface of steel plates, *Appl. Sci.*, **9** (2020), 4222. https://doi.org/10.3390/app9204222
68. Z. Jeelani, F. Qadir, Cellular automata-based approach for salt-and-pepper noise filtration, *J. King Saud Univ. Comput. Inf. Sci.*, **2018** (2018). https://doi.org/10.1016/j.jksuci.2018.12.006
69. S. Z. Li, *Markov Random Field Modeling in Image Analysis*, 2nd edition, Springer-Verlag, New York, 2009.
70. Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: From error visibility to structural similarity, *IEEE Trans. Image Process.*, **13** (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861
71. A. Hoover, G. J. Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, et al., An experimental comparison of range image segmentation algorithms, *IEEE Trans. Pattern Anal. Mach. Intell.*, **18** (1996), 673–689. https://doi.org/10.1109/34.506791
72. Z. Hu, Z. Wu, Q. Zhang, Q. Fan, J. Xu, A spatially-constrained color-texture model for hierarchical VHR image segmentation, *IEEE Geosci. Remote Sens. Lett.*, **10** (2013), 120–124. https://doi.org/10.1109/LGRS.2012.2194693
73. S. Sangwine, R. Horne, *The Colour Image Processing Handbook*, Chapman and Hall, 1998.