International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Hybrid Cellular Automata - Patch Based Local Principal Component Analysis Techniques for Improving Image De-Noising

Suresh. A[1], Ranjithkumar. S[2], Nithya. N S[3], S. Vinoth Kumar[4], S. Girirajan[5]

Submitted: 15/10/2023 **Revised:** 06/12/2023 **Accepted:** 10/12/2023

Abstract: The primary motivation behind this article lies in addressing the growing demand for image processing applications within critical environments where the presence of noise can significantly degrade the quality of the outputs. Existing methods primarily consider the noisy version of the image for denoising, neglecting crucial properties such as the dimension of image pixels. This oversight can lead to inaccurate estimates of image values. In the case of Patch-based Local Principal Component Analysis (PL-PCA) filters, the choice of similarity weights and the reduction of sample sizes are pivotal factors. Increasing the sample size can complicate image handling. Regrettably, existing filtering techniques focus solely on noise removal without taking into account the size of input samples. This article aims to resolve these challenges by introducing the H-PL-PCA (Hybrid Patch-based Principal Component Analysis) approach for efficient image noise filtering. The proposed method addresses issues related to dimensionality reduction and the selection of neighboring cells, ensuring that the end results are obtained through the comprehensive analysis of various parameters for effective denoising.

Keywords: *Cellular Automata; Image denoising; Dimensionality Reduction; Principal Component Analysis; Speckle Suppression Index SSI; Speckle Suppression and Mean Preservation Index SMPI*

1. Introduction

The foremost source of noise in digital images arises during the image acquisition or digitization process. This noise is often associated with the inherent limitations of image sensors and can be influenced by various factors. Environmental conditions during image capture, such as temperature and ambient light levels, significantly impact the quality of images. Network issues or problems with the transmission medium, such as packet loss or corruption, can also lead to data loss or errors in the transmitted images, contributing to noise in the final received images [1].
Color quantization methods aim to reduce the number of colors in an image, which is especially useful for displaying images on devices with limited color support. Popular algorithms include the nearest-color algorithm, median-cut algorithm, and octree-based approaches. This technique takes advantage of the human eye's ability to perceive differences in brightness over a large area while being less sensitive to rapid, high-frequency variations, allowing for data reduction. Various denoising techniques, such as Cellular Automata (CA) filtering, have been developed to address speckle noise, and their effectiveness can be evaluated based on parameters like noise density as discussed [2].
To distinguish between noise and the actual image content, a thresholding technique is employed. This technique involves using a mask with a 4x4 grid to identify patterns for detecting salt and pepper noise [3-5]. The mask is moved across the image in a systematic manner, and rules guide the identification of noise within the cells of the mask. While traditional mean filtering techniques can effectively remove salt and pepper noise from images, they often introduce a blurring effect. This is because, in mean filtering, each pixel is replaced with a fixed mean that is calculated based on both affected and unaffected pixels. This blurring effect can hamper the achievement of a higher Peak Signal-to-Noise Ratio (PSNR) and better image quality [6-8]. Therefore, there is a need for more advanced denoising techniques to address this challenge and improve the quality of denoised images affected by salt and pepper noise [9, 10].

[1]*Associate Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India. Email: a.suresh@vit.ac.in*

[2]*Assistant Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India. Email: ranjithkumar.s@vit.ac.in*

[3]*Associate Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India. Email: nithya.ns@vit.ac.in*

[4]*Associate Professor, School of Computing, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, India. Email: profsvinoth@gmail.com*

[5]*Assistant Professor, Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur, India. Email: girirajans.cse@gmail.com*

2. Related Works

The introduced denoising algorithm makes use of cellular automata with a Moore neighborhood in its local transition function. This CA-based method is designed for filtering images that are affected by salt and pepper noise. Through a comparison with the traditional median filter using the Hamming distance metric, the algorithm has demonstrated its superiority in terms of denoising effectiveness, especially when the intensity of salt and pepper noise exceeds 40%. This demonstrates the effectiveness of CA-based approaches in handling this type of noise. To address salt and pepper noise in images, a thresholding method is applied to distinguish noise from the actual image content. A 4x4 mask with specific patterns is employed for detecting salt and pepper noise. Specific criteria guide the identification of noise using this mask, with a focus on border cells that should not be considered noise. This process is essential for accurately identifying and removing the noise while preserving the image's essential features and details [11,12]. Cellular Automata's local rule-based processing and adaptability through rule design make them effective in image processing tasks such as denoising and edge detection [16].
The proposed mean filtering method offers an enhancement to the traditional approach by introducing two novel features. First, it

only considers unaffected pixels when calculating the arithmetic mean, effectively ignoring the noise-affected pixels. This selective consideration of clean pixels during mean computation helps preserve the original image details and reduces the impact of salt and pepper noise. Second, the introduction of a tolerance value is a significant advancement. This tolerance value allows for more controlled replacement of noisy pixels, which contributes to improved denoising results. By specifying a threshold, the method can decide whether to replace a pixel with the mean value, depending on the extent of its deviation from the mean. This adaptive approach to pixel replacement plays a vital role in enhancing the denoising effectiveness [13]. While the proposed method with tolerance-based modifications addresses this problem to some extent, it's essential to strike a balance between denoising and image quality, as excessively aggressive noise removal can lead to overly smoothed or blurred images. Achieving a high Peak Signal to Noise Ratio (PSNR) is one aspect of evaluating denoising methods, but it should be complemented by visual inspection to ensure that important image features are preserved [14].

3. Proposed Framework

3.1 Cellular Automata for Image De-noising

One of the key strengths of cellular automata in image processing is their local rule-based processing. Each cell's behavior is determined by its immediate neighborhood, consisting of neighboring cells' states. This localized approach is advantageous in denoising tasks as it allows cells to consider their local context when deciding on state changes.

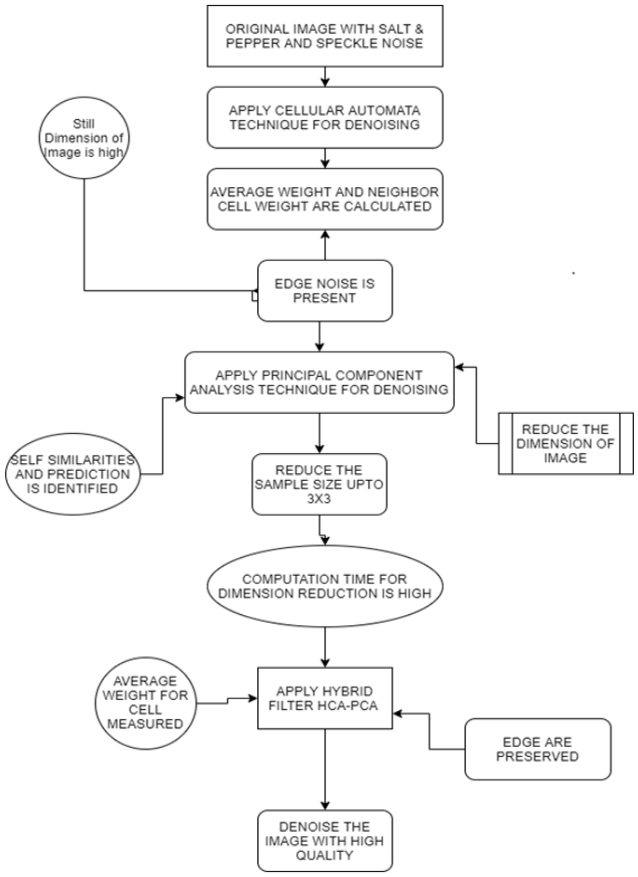

Fig. 1. Proposed Framework

The presented algorithm delineates a denoising procedure employing Cellular Automata (CA) to enhance the quality of noisy images by examining each pixel's local surroundings. The process is structured as follows:

Step 1: **Image Input**: Commence by reading the input image (x, y).

Step 2: **Color Channel Handling**: If the input image is in RGB format, convert it to grayscale. Alternatively, you can opt to process each color channel independently.

Step 3: **Local Neighborhood**: Define a 2-D window of size 3 × 3 to traverse the image I (x, y).

Step 4: **Central Pixel Identification**: Let Ci,j denote the central pixel within a 2D Moore's neighborhood, an integral element of the Cellular Automata (CA).

Step 5: **Neighbor Pixel Vector**: Create a vector B, sized 8 × 1, to store pixel values extracted from the window while excluding the central pixel. Arrange these values in ascending order.

Step 6: **Min-Max Pixel Values**: Identify the minimum pixel value, Bmin, and the maximum pixel value, Bmax, from vector B.

Step 7: **Uncorrupted Pixels**: If Ci,j is an uncorrupted pixel ($0 < Ci,j < 255$), it remains unchanged.

Step 8: **Handling Noisy Pixels**: When Ci,j is a noisy pixel ($Ci,j = 0$ or $Ci,j = 255$), various actions are taken based on the following cases:

Case 1: If Bmin = 0 and Bmax = 255, set Ci,j to the mean of the values in vector B, excluding 0 and 255.

Case 2: If all elements in vector B are 0 or 255, Ci and j remain unchanged.

Case 3: If Bmin > 0 and Bmax < 255, set Ci,j to the mean of the values in vector B.

Step 9: **Iterative Denoising**: Reiterate steps 6 to 8 for each pixel in the input image I(x, y) for a specified number of iterations. The number of iterations is calculated using "n/10 + 1," where "n" represents the level of noise in the image. The aim of the iterations is to progressively reduce noise in the image.

This algorithm provides a fundamental illustration of how CA can be employed for image denoising. It relies on evaluating the local pixel values to decide whether a pixel requires correction. The approach offers distinct rules for addressing noisy pixels, contingent on the values within their vicinity. It also adapts the number of iterations to the prevailing noise level. In practical applications, fine-tuning the specific values used for noise thresholds and the number of iterations may be necessary to optimize denoising outcomes for different images and noise levels.

Within step 8 of the algorithm, three distinct cases govern the handling of noisy central pixels in a 2D cellular automaton:

1. When the central pixel falls within a window encompassing values from 0 to 255, its value is updated to the mean of the surrounding values, excluding 0 and 255.
2. If all neighboring values are either 0 or 255, the central pixel remains unaltered.
3. In situations where the surrounding values fall within the range (0, 255), the central pixel is adjusted to the mean of the neighboring values.

The algorithm's innovation lies in the exclusion of minimum and maximum values from the mean calculation. This approach results in smoother preservation of edges and mitigates abrupt changes in the denoised image, particularly for noise densities spanning from 10% to 90%.

The computational complexity analysis offers an estimation of the computational resources necessary for the denoising process. It emphasizes that the algorithm's efficiency can improve as the number of iterations advances, a common attribute of iterative denoising techniques. The exact computational performance will depend on specific image characteristics, the noise level, and other influencing factors. However, this upper-bound complexity estimate serves as a valuable guideline for assessing the algorithm's resource demand.

3.2 Hazard in Cellular Automata

One crucial aspect in designing an effective CA filter is the selection of proximity weights. These weights determine how neighboring cells or regions influence each other in the denoising process. The choice of proximity weights plays a significant role in the algorithm's performance. The proximity weights should be tailored to the specific image being processed. Different images may have different noise characteristics and structures, so the

weights should be optimized to achieve the best denoising results for that particular image. This customization can enhance the algorithm's ability to preserve image features while reducing noise [17].

Overall, addressing complex noise patterns, such as speckle noise, requires careful algorithm design and customization. Advanced CA algorithms and the fine-tuning of proximity weights are essential steps in developing effective noise reduction techniques. Additionally, understanding the unique characteristics of the noise in a specific image is crucial for optimizing the denoising process and achieving the best results.

3.3 Patch based Local Principal Component Analysis De-Noising Technique

Patch-based methods involve assessing the similarity between two regions or patches in the image. This similarity information informs the strategy for sharing patches, which aids in the denoising process.

In traditional batch PCA, the principal components are computed based on the eigenvectors of the covariance matrix. The optimal dimensionality "m" (number of principal components to retain) is determined once, typically when the entire dataset is available for analysis. However, this fixed dimensionality may not be suitable for data streams where the data distribution can change over time. In data streams, the optimal dimensionality "m" may continuously change due to variations in the data distribution. Existing approaches that allow for real-time adjustment of "m" often have limitations. They might permit increasing "m" by one for each incoming data point, which is insufficient to address sudden and significant changes in the data, potentially leading to information loss and suboptimal representation of the evolving data stream. The algorithm is designed to adaptively adjust the dimensionality "m" with an arbitrary step size while efficiently handling data streams. It incorporates elements from neural network-based PCA, which emphasizes efficient dimensionality reduction [18]. The algorithm acknowledges the ordering of eigenvalues in descending order and processes principal components sequentially, recognizing that variance is not evenly distributed across all components. By prioritizing the eigenvalues with the highest variability during training, it efficiently focuses on the most significant components first. The algorithm commences with "m = 2" for the initial processing cycle, indicating that it initially processes only the two most significant principal components. This approach efficiently reduces the system size and efficiently handles the most important components. This adaptability allows the algorithm to be responsive to changes in data distribution, making it a valuable tool for real-time applications where data distributions are dynamic. It addresses the challenge of dimensionality reduction in a flexible and efficient manner, making it suitable for a range of applications requiring adaptive data analysis.

By training a linear regression model with the logarithmic eigenvalues $\lambda i = \log(\lambda i)$, one can make predictions for additional eigenvalues. This approach leverages the observed linear behavior of the eigenvalues to estimate the behavior of the entire dataset, making it a useful tool for extrapolating information about the dataset beyond the observed values. The logarithmic eigenvalues in the set V are used to train a linear regression model. This model calculates the slope (α) and intercept (β) of a least-squares regression line in the logarithmic scale. This regression line is represented as $\lambda i = \alpha i + \beta$, where i ranges from m + 1 to n, predicting the missing eigenvalues based on the observed ones. The regression line is extrapolated to estimate the missing n - m eigenvalues in the logarithmic scale, resulting in a set of estimated eigenvalues $V^* = (\lambda m+1, \lambda m+2, ..., \lambda^* n)$. These estimated logarithmic eigenvalues are then combined with the initial set V, which contains the first m computed eigenvalues. The combined set, denoted as U, includes both the initial eigenvalues and the estimated missing eigenvalues. This procedure serves the purpose of estimating the remaining eigenvalues, which can be valuable in scenarios like dimensionality reduction or handling dynamic data distributions. By employing linear regression on the logarithmic eigenvalues, it efficiently predicts the missing eigenvalues, helping maintain a reduced dimensionality representation while preserving essential information from the original dataset.

$$\lambda i = \alpha i + \beta \text{ with } i \{m + 1, \dots, n\} \quad (1)$$

where λi represents the eigenvalues. The slope α and the intercept β of the regression line in the logarithmic scale are determined. The star symbol denotes values estimated by the linear regression line.

The estimated n − m log-eigen values,

$$V^* = (\lambda m+1, \lambda m+2, \dots, \lambda^* n) \quad (2)$$

generated as shown in Figure 1c, are combined with the initial set $V = \{\lambda 1, \lambda 2, \dots, \lambda m\}$ containing the first m computed eigenvalues:

$$U = V \cup \{\lambda m+1, \lambda m+2, \dots, \lambda^*\} \quad (3)$$

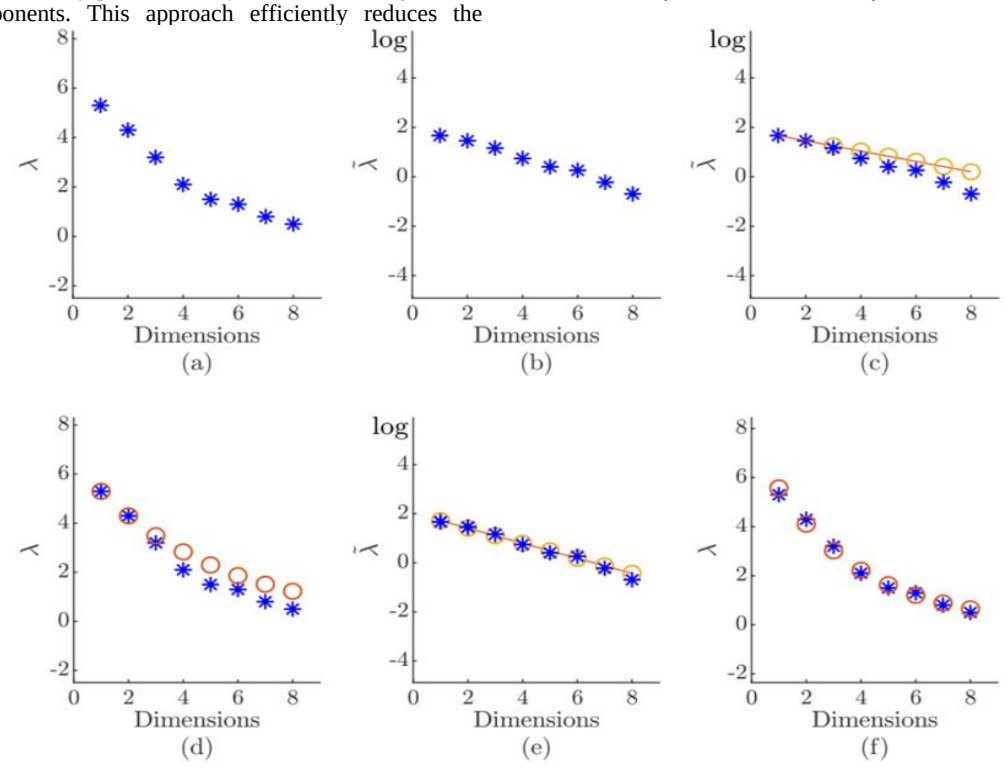

Fig. 2. Dimensionality reduction in PCA method

The introduced stopping criteria can be adapted using the proposed approach, which relies on the initially trained "m" eigenvalues (with an initial m = 2) and the regression parameters α and β derived from the log-eigenvalues. This adaptation allows for a more flexible and data-driven approach to determining the stopping criteria, enhancing the robustness of the dimensionality reduction method.

$$\lambda total = \sum_{i=1}^{n} \lambda i = \sum_{i=1}^{m} \lambda i + \sigma^2 \quad (4)$$

The dimensionality reduction process involves utilizing the "m" trained eigenvalues (λi) and the residual variance (σ^2), both of which are continuously updated in each PCA update step. This dynamic updating of eigenvalues and residual variance helps in maintaining an accurate representation of data distribution for effective dimensionality reduction.

$$m=|\{b \in U \mid b>1\}| \quad (5)$$

In these approaches, dimensionality is determined based on the extended set U rather than the fully trained set V. The condition for retaining eigenvalues greater than one or greater than the average simplifies the selection of relevant eigenvalues for further processing, effectively reducing dimensionality.

$$m=|\{ b \in U \mid b>\frac{1}{n} \lambda_{total}\}| \quad (6)$$

This approach uses a dynamic threshold for dimensionality reduction. Instead of a static threshold, as seen in the offline version (equation 3), it adjusts the average threshold with each PCA step. This moving threshold is calculated based on the average of the eigenvalues. The dynamic nature of this threshold accounts for changes in data distribution, making it more adaptable to evolving datasets. While the dynamic threshold provides greater adaptability, it introduces additional complexity compared to the static threshold (equation 3). The use of a moving average requires continuous updates with each PCA step, which can increase computational overhead. In contrast, the static threshold of equation 8 remains fixed throughout the process. Therefore, this dynamic approach, although more flexible, may be computationally more intensive than the eigenvalue-one threshold equation (equation 9) and the offline version of the threshold.

$$m=|\{b \in U \mid b> \eta \lambda_{total} \} \quad (7)$$

The factor η provides flexibility in adjusting the dimensionality threshold. However, this flexibility adds complexity due to the interactions between the moving total variance and the threshold η.

$$m=\arg\min \{z \in \{1,2,..,n\} | \sum\_(i=1)^z Ui \geq \lambda_{total} \quad (8)$$

The criterion [12] is based on the already trained "m" eigenvalues "λi" and their corresponding approximations derived from the regression line parameters "α" and "β." This criterion is a key part of the dimensionality adjustment process and can effectively handle a wide range of leading component distributions. It's particularly robust when the eigenvalues are arranged in descending order, as is common in hierarchical online PCA algorithms. The criterion's robustness becomes evident when dealing with different eigenvalue distributions. For example, in cases of perfectly symmetric data distributions with equal variances in all dimensions, the regression line with a slope "α" of zero can accurately estimate the remaining leading components. Conversely, in scenarios where all variance is concentrated in one dimension, a large negative slope "α," combined with the logarithmic function's behavior, can estimate values close to zero for the other leading components. This adaptability of the criterion, combined with the logarithmic scale representation, allows it to approximate a wide range of potentially occurring eigenvalue distributions effectively on the normal scale, making it a versatile tool in the dimensionality reduction process.

4. Algorithm 2: PCA Dimensionality Adjustment Procedure

1. In step 1, the code updates the PCA model with the new input data x using an online PCA method. It updates the current center c, eigenvectors W, and eigenvalues Λ.
2. Step 2 checks if a certain number of training cycles κ has exceeded a predefined value Γ. If it has, the dimensionality reduction procedure is initiated.
3. Step 3 (inside the "if" block) is the core of the dimensionality reduction procedure. It adjusts the dimensionality using the "Dimensionality Adjustment" procedure. This adjustment is based on the eigenvalues.
4. In Step 4, the code performs a log transformation (V) on the diagonal elements of the eigenvalues Λ.
5. Step 5 calculates the parameters α and β using linear regression on the transformed eigenvalues V.
6. Step 6 estimates the log eigenvalues U using the regression parameters α and β.
7. In Step 7, the code performs a normal transformation on the estimated eigenvalues.
8. Step 8 determines the updated dimensionality using a stopping rule. This rule is applied to the non-log transformed set U.
9. The procedure ends if κ exceeds the predefined value Γ.
10. Finally, the code increments the training cycle count (κ) in Step 11.

This code represents an adaptive dimensionality reduction algorithm that dynamically adjusts the dimensionality based on the data and uses online PCA techniques. It is particularly useful for scenarios where data distributions are dynamic and efficient real-time dimensionality reduction is required. The stopping rule in Step 8 plays a crucial role in determining the final dimensionality based on the observed eigenvalues and their linear regression.

Algorithm 1 is designed for adaptive dimensionality reduction and takes several input parameters, including the current dimensionality "m," current eigenvectors "W," current eigenvalues "Λ," current mean "c," and new input data point "x." The algorithm is executed every time a new data point is presented. It goes through a complete training cycle and updates the PCA model parameters. The outputs of the algorithm include the updated dimensionality "m," updated eigenvectors "W," updated eigenvalues "Λ," and an updated mean "c.". The algorithm begins with an initial dimensionality of two, where the first two leading components are trained for a specific number of training cycles. After this initial phase, the dimensionality reduction method is initiated. Based on the logarithmic eigenvalues, linear regression parameters "α" and "β" are updated. These parameters are used to estimate the remaining eigenvalues in the logarithmic scale. Stopping criteria are applied to the non-log set "U," resulting in an updated dimensionality "m." The algorithm is designed to adapt to dynamic data distributions and efficiently reduce dimensionality while maintaining essential information. It is evaluated on diverse datasets, and its performance is benchmarked against an incremental PCA approach to assess its effectiveness.

4.1 Hazards in Patch based Local Principal Component Analysis

As the dimensions of image processing models increase, the task becomes more challenging due to the heightened complexity. Managing and processing high-dimensional data require substantial computational resources and can lead to increased processing times. This poses a challenge in maintaining efficiency and real-time processing. Many conventional filtering techniques primarily focus on noise removal and do not take into account the specific characteristics of the data samples in the image. This lack of consideration for data attributes can result in suboptimal outcomes, as different image regions may require different processing approaches. Consequently, filtering systems may not be well-suited for addressing the diverse needs of

complex image processing tasks, especially when working with high-dimensional data.

5. Hybrid Cellular Automata-Patch based Local Principal Component Analysis De-Noising Technique

A hybrid approach that combines Cellular Automata (CA) and Principal Component Analysis (PCA) has been employed to address image noise issues. In this CA-PLPCA hybrid, the process commences at the edge level by applying a CA Wiener filtering method, which operates based on certain rules of progression. The conventional approach employs a median filter, but it does not effectively enhance image denoising [7]. Therefore, in this hybrid approach, the Wiener filter within a CA framework is applied. Additionally, the patch size is limited to 3×3 in the PL-PCA phase.

Before undertaking this work, prior research used Stationary Wavelet Transform (SWT) as a thresholding technique, which is used to match the image in terms of grayscale. Choosing the right method for setting the threshold value in the image is essential for addressing various types of noise, such as Gaussian noise, salt and pepper noise, and speckle noise. The results indicate that the hybrid CA-PLPCA approach has improved metrics, including PSNR, ENL, COC, SSI, SMPI, SSIM, and MSE.

Stationary Wavelet Transform (SWT) is a wavelet transform technique that involves a set of fundamental functions. These functions are used to analyze signals simultaneously in both time and frequency domains. The wavelet transform is a critical tool due to its energy compaction property, which is used in denoising to resolve noise issues. Wavelet transforms provide a multiresolution representation of images and signals. This means that noise encompasses all frequencies, both low and high. Thanks to the higher frequencies, noise is periodic or non-smooth. The lower frequencies (low scale) suggest that the actual image, without noise, is smooth or piecewise smooth. Selecting the threshold value carefully is essential to ensure that the essential image information is preserved while reducing noise.

$$T = \sigma\sqrt{2logM} \quad (9)$$

$\sigma = Noisevariance$

$M = Imagelength$

$$\sigma = \frac{meadin\{|W_k|:k=1,2,\ldots n\}}{0.6745}$$

5.1 Modified Algorithm Procedure in Hybrid CA-PL-PCA

Step 1: Noisy PL-PCA Algorithm This step represents the initial part of the denoising process, which is patch-based PCA (PL-PCA). It typically involves clustering patches and denoising them using PCA.

Step 2: CA Inputs In this step, the algorithm takes the noisy pixels as inputs, denoted as yi for i = 1,..., M. These pixels are the elements that need to be denoised.

Step 3: Parameters The algorithm defines several key parameters, including the patch size (typically √N × √N), the number of clusters (K), and the maximum number of cycles (Niter).

Step 4: Output The ultimate output of this hybrid approach is the estimated image. The denoising process aims to enhance the quality of the noisy input image.

Step 5: Method This section outlines the method for denoising the image, combining Noisy PL-PCA and CA techniques.

Step 7: Patchization and Clustering The algorithm starts by creating a collection of patches from the noisy image Y. It then proceeds to cluster these patches using the K-means algorithm, generating K clusters. Each cluster, represented by a matrix Yk, contains Mk elements.

Step 8: Iterative Denoising within Clusters For each cluster, the algorithm performs an iterative denoising process. It initializes matrices U0 and V0, where U0 has dimensions Mk x l and V0 has dimensions l x N. It then enters a loop that iteratively updates the rows of U and the columns of V. The loop continues as long as certain conditions are met (t ≤ Niter and test > ε).

Step 9: Cluster Output After the loop, the algorithm computes Fk, a matrix representing the denoised cluster, using the updated U and V matrices. This process is applied to each cluster.

Step 10: Concatenation and Reprojection The algorithm combines the collection of denoised patches, F, and then performs a re-projection step where the various pixel estimates are averaged, taking overlaps into account. This results in an estimated image, F, which is the output of the entire hybrid denoising process.

This modified hybrid algorithm takes advantage of both patch-based PCA and cellular automata to effectively reduce noise in the input image. The two techniques are complementary and work together to improve denoising results.

6. Experimental Results and Discussion

6.1. PSNR (Peak – Signal to Noise Ratio)

Signal-to-noise, typically denoted as PSNR (Peak Signal-to-Noise Ratio), is a fundamental term used to quantitatively measure the relationship between the maximum possible power of a signal and the power of degrading noise that affects the fidelity of its representation. While a higher PSNR often indicates that the representation is of higher quality, it may not always be the case. One should be very cautious about the limited applicability of this measurement; it is only truly valid when used to compare results from the same codec (or codec type) and identical content [19]. The mathematical representation of PSNR is as follows:

$$PSNR = 20log_{10}(\frac{MAX}{\sqrt{MSE}}) \quad (10)$$

$$PSNR(dB) = 10log_{10}(\frac{255^2}{MSE}) \quad (11)$$

6.2. MSE (Mean Squared Error)

PSNR (Peak Signal-to-Noise Ratio) is commonly defined using the Mean Squared Error (MSE). The MSE is calculated as follows:

$$\text{MSE} = \frac{1}{mn}\sum_0^{m-1}\sum_0^{n-1}||f(i,j) - g(i,j)|| \quad (12)$$

This can also be represented in a text-based format as:

$$\text{MSE}=(1/(m*n))*$$

$$sum(sum(f-g).\text{^}2))$$

$$PSNR = 20*$$

$$log(max(max(f)))/(MSE)\text{^}0.5)$$

Where f: represents the network information of our original, high-quality image. It contains the pixel values of the known, pristine image.

-g: Refers to the network information of the image in question, which may be corrupted or degraded in some way.

-m: Denotes the number of columns or pixels in a row (line) of the images. It's essentially the width of the image.

- I: Represents a list or array that holds the pixel values within a specific row or line of the image.

- n: Indicates the number of segments or columns of pixels in the image, often representing its height.

- j: Corresponds to a list or array that stores pixel values within a particular column or segment of the image.

- max(f): This term signifies the highest signal value present in our original, known-to-be-good image.

These variables are commonly used in image quality assessment and metrics like Peak Signal-to-Noise Ratio (PSNR) that involve comparing a processed or degraded image (referred to as 'g') to the original, high-quality image ('f'). The PSNR calculation takes into account the Mean Squared Error (MSE) between these images, helping to quantify the level of noise or degradation in 'g'

concerning the 'f' reference, with 'max(f)' being the peak possible signal value [20].

6.3. ENL (Equivalent Number of Looks)

Greater ENL effectiveness in a filter corresponds to increased efficiency in mitigating speckle noise across uniform image regions.

$$ENL = (\frac{mean}{standarddeviation})^2 \quad (13)$$

6.4. SSI (Speckle Suppression Index)

Speckle noise typically manifests as clusters of relatively small groups of pixels. Suppressing this noise can sometimes lead to the loss of fine image details. Therefore, the ability to preserve the essential content of an image is quantified by SSI.

$$SSI = \frac{\sqrt{var(I_f)}}{mean(I_f)} \times \frac{mean(I_o)}{\sqrt{var(I_o)}} \quad (14)$$

6.5. SMPI: Speckle Suppression and Mean Preservation Index (SMPI)

ENL and SSI may not be reliable when the filter excessively alters the mean value. A new metric called the Speckle Suppression and Mean Preservation Index (SMPI) has been developed. In the case of this index, lower values indicate a better performance of the filter in terms of preserving the mean value and reducing noise.

$$SMPI = Q \times \frac{\sqrt{var(I_f)}}{\sqrt{var(I_o)}} \quad (15)$$

$$Q = R + |mean(I_o) - mean((I_f)|$$

Where $R = \frac{Max\left(mean(I_f)\right) - Min(mean(I_f))}{mean(I_o)}$ (16)

6.6. SSIM (Structural Similarity Index)

The SSIM (Structural Similarity Index) score is a method for quantifying the similarity between two images. The SSIM score can be interpreted as a measure of how well one of the images being compared matches the other image, assuming the latter is considered of excellent quality.

$$SSIM = \frac{(2\mu_a\mu_b + C_1)(2\sigma_{ab} + C_2)}{(\mu_a^2 + \mu_a^2 + C_1)(\sigma_a^2 + \sigma_a^2 + C_2)} \quad (17)$$

6.7. COC: Correlation Coefficient

Correlation is a statistical measure of how changes in the value of one variable predict changes in the value of another.

$$(r) = [n\Sigma xy - (\Sigma x)\,(\Sigma y)/Sqrt([n\Sigma x^2 - (\Sigma x)^2[n\Sigma y^2 - (\Sigma y)^2])] \quad (18)$$

Fig. 3. Sample Images with De-noising Outputs

Table 1. Hybrid CA-PL-PCA- Salt and Pepper Noise

IMAGE	PSNR	MSE	ENL	SSI	SMPI	SSIM	Coc
Barbara	51.3834	0.3736	15.4675	0.9954	-0.0608	0.9935	0.989
Boat	54.2704	0.1916	15.4243	0.9934	-0.0622	0.9936	0.9943
Bridge	53.7978	0.2127	12.3879	0.9936	-0.0609	0.9942	0.9942
Cameraman	51.0836	0.4013	7.921	0.945	-0.0608	0.9932	0.9944

Couple	54.668	0.1731	11.4546	0.9962	-0.0543	0.9922	0.9947
Flintstones	44.4162	1.8641	28.5436	0.9709	-0.0842	0.9816	0.9932
Hill	55.8175	0.1326	32.1319	0.9801	-0.0544	0.9946	0.9941
House	54.7212	0.1721	13.0095	0.9952	-0.0635	0.9942	0.9945
Lena	53.0681	0.2524	10.2481	0.0079	-0.0553	0.9935	0.9943
Man	54.8663	0.1642	14.1562	0.0065	-0.0543	0.9943	0.9936
Bio-Medical							
Fingerprint	39.5435	5.7338	34.1258	0.9134	-0.0823	0.9357	0.9654
Gulcoma	60.8471	0.0423	15945	0.9989	-0.0329	0.9943	0.9993
Palm	57.8292	0.0824	4.4257	1.0038	-4.46E-02	0.9945	0.9943

Table 2. Hybrid CA-PL-PCA- Gaussian Noise

IMAGE	**PSNR**	**MSE**	**ENL**	**SSI**	**SMPI**	**SSIM**	**Coc**
Barbara	51.6673	0.3435	15.097	0.9949	-0.0594	0.9934	0.9943
Boat	54.8555	0.1665	14.73	0.9943	-0.0611	0.9942	0.9944
Bridge	54.2623	0.1916	11.9864	0.9934	-0.0587	0.9943	0.9945
Cameraman	51.5373	0.3595	5.4795	0.9966	-0.0630	0.9932	0.9974
Couple	55.0855	0.1587	11.2469	0.9962	-0.0526	0.9943	0.9964
Flintstones	44.5474	1.8114	22299	0.9829	-0.0837	0.9822	0.987
Hill	56025	0.1111	31.041	0.9846	-0.0542	0.9942	0.9943
House	54.0571	0.1599	12.8895	0.9956	-0.0632	0.9941	0.9942
Lena	53.4203	0.221	9.9503	0.9942	-0.0543	0.9934	0.9943
Man	55.2416	0.1515	13.6589	0.9954	-0.0551	0.9943	0.9945
Bio-Medical							
Fingerprint	39.5416	5.7401	33.1327	0.9111	-0.0814	0.9395	0.9687
Gulcoma	61.8675	0.0316	14.7128	0.9945	-0.0303	0.9945	0.9945
Palm	57.9152	0.0825	4.1803	0.9938	-4.60E-02	0.9942	0.9958

Table 3: Hybrid CA-PL-PCA- Speckle- Noise

IMAGE	**PSNR**	**MSE**	**ENL**	**SSI**	**SMPI**	**SSIM**	**Coc**
Barbara	50.5462	0.2618	15.363	.9955	-0.05	0.9944	0.998
Boat	53.017	0.2037	15.75	0.9974	-0.0612	0.978	0.9991
Bridge	52.4513	0.2233	12.293	0.9964	-0.0594	0.9963	0.9993
Cameraman	49.6041	0.3494	5.8415	0.9973	-0.0654	0.9935	0.9991
Couple	52.9508	0.207	11.937	0.0059	-0.0553	0.9959	0.999
Flintstones	43.1707	0.977	27.384	0.9824	-0.0843	0.9805	0.9956
Hill	54.2772	0.1432	33.977	0.9817	-0.0569	0.9944	0.9965
House	53.2427	0.1844	13.544	0.9957	-0.0635	0.9968	0.9974
Lena	51.9624	0.2411	10.764	0.978	-0.0552	0.9954	0.9987
Man	52.3606	0.2232	14.393	0.9973	-0.0544	0.9961	0.9988
Bio-Medical							
Fingerprint	39.8604	5.1899	34.998	0.9036	-0.0814	0.9427	0.9689
Gulcoma	57.6019	0.0857	146	0.9995	-0.0317	0.9979	0.9976
Palm	54179	0.1148	4.6048	.9936	-0.0448	0.9976	0.9987

7. Performance Evaluation

7.1. PSNR (Peak – Signal to Noise Ratio) Comparison De-noised Evaluation

This work considered the accompanying significant presentation measurements for the assessment by reenactment.

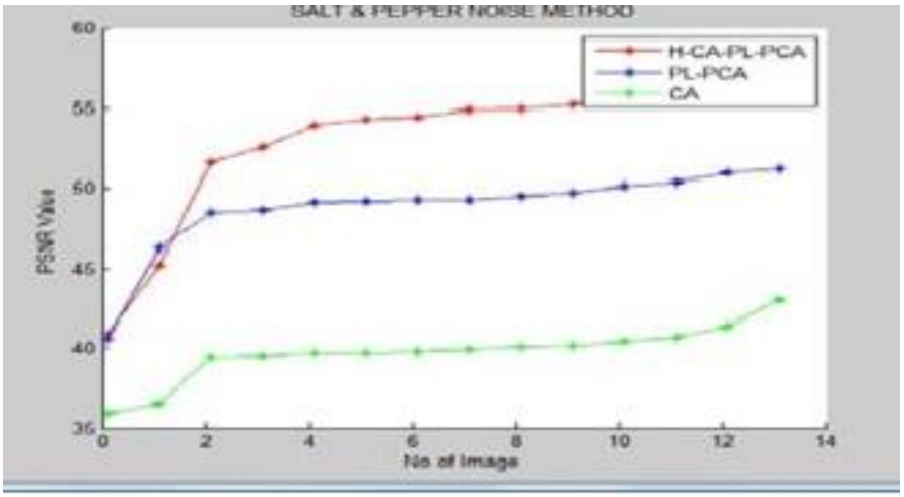

Fig. 4. Salt and Pepper Noise Method for PSNR Value

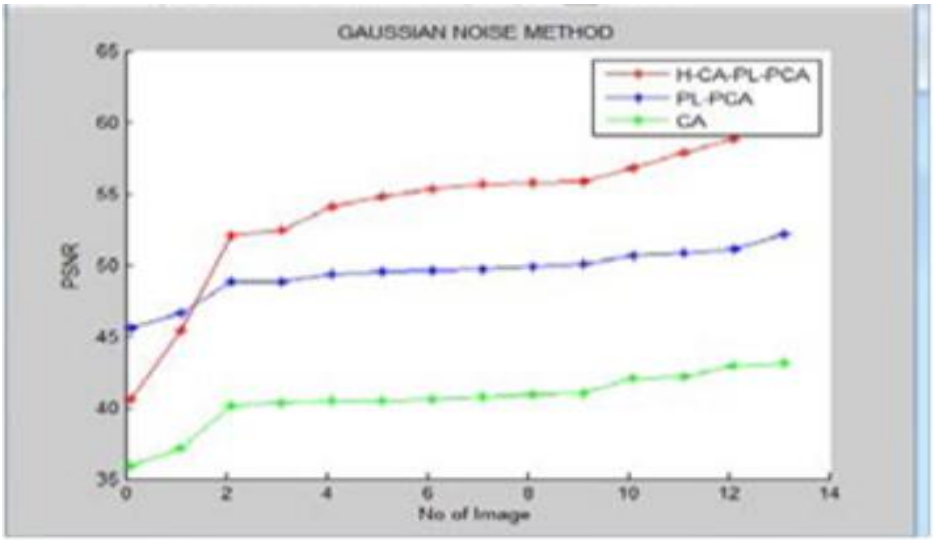

Fig. 5. Gaussian Noise Method for PSNR Value

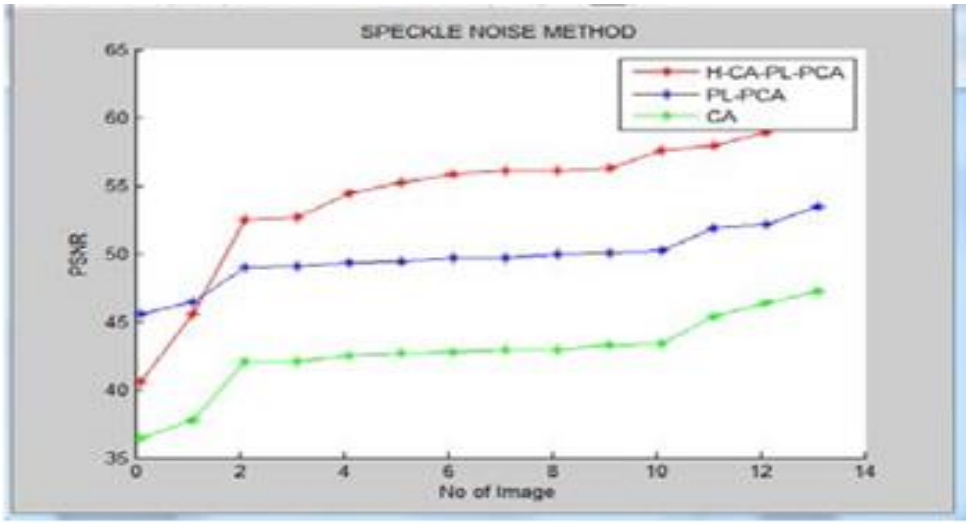

Fig. 6. Speckle Noise Method for PSNR Value

Fig 4, 5 and 6 shows the relationship between PSNR ratios on standard images highlights that the H-CA-PL-PCA approach outperforms other CA and PL-PCA approaches, yielding significantly higher PSNR ratios. This superior performance is

evident across 14 different images when applying the image denoising technique.

The PSNR (Peak Signal-to-Noise Ratio) is a quality metric used to assess the fidelity of denoised images by comparing them to their original versions. A higher PSNR value signifies that the denoised image closely resembles the original image, indicating superior quality in the denoising process. Consequently, when it comes to the image denoising process, achieving a high PSNR ratio is indicative of superior performance. The proposed method in this study demonstrates its effectiveness by consistently delivering higher PSNR ratios across a range of images. Importantly, this superior performance is not limited to just a few images but holds true even as more images are included in the evaluation.

7.2. SSIM (Structural Similarity Index) Comparison for de-noised Evolution

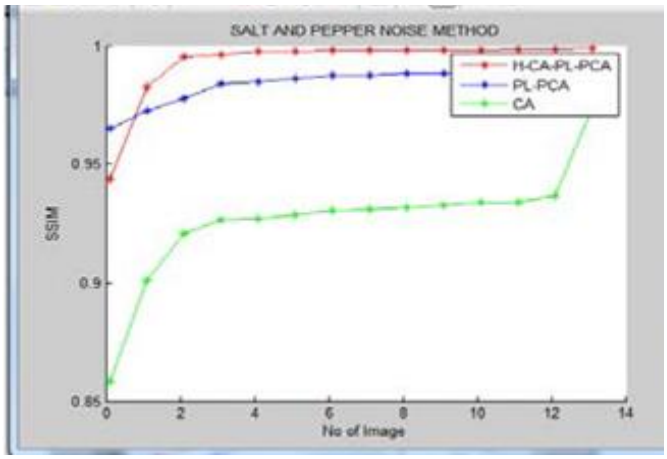

Fig. 7. Salt and Pepper Noise Method for SSIM Value

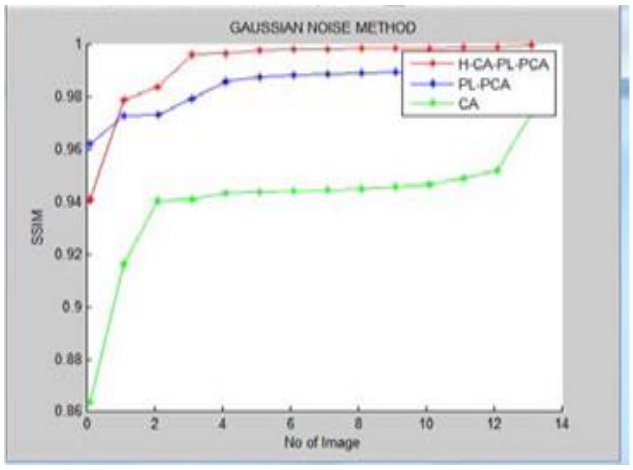

Fig. 8. Gaussian Noise Method for SSIM Value

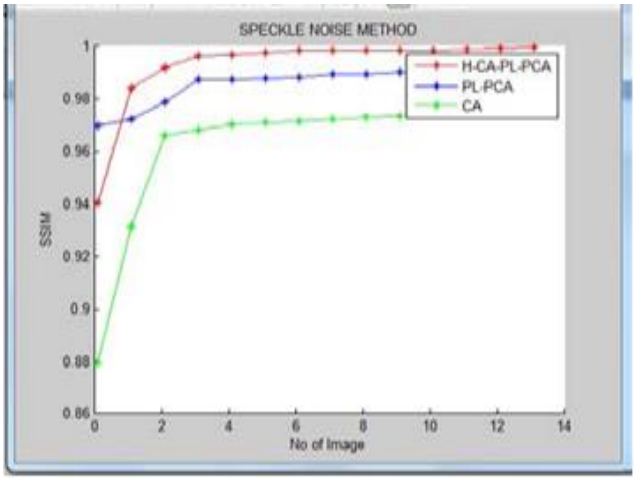

Fig. 9. Speckle Noise Method for SSIM Value

Fig 7, 8 and 9, shows the relationship between SSIM (Structural Similarity Index) proportions on standard images reveals that the proposed H-CA-PL-PCA approach consistently yields higher SSIM proportions when compared to other CA and PL-PCA techniques. This trend is evident across 14 different images subjected to the image denoising process.

SSIM serves as a valuable metric for quantifying the similarity between two images. It operates as a full-reference metric, relying on an undistorted reference image for quality assessment. In the context of image denoising, a high SSIM score is indicative of superior performance, and the proposed method clearly demonstrates this.

Notably, the performance of SSIM proportions across images continues to excel, even as the number of evaluated images increases.

7.3. MSE (Mean Squared Error) comparison for de-noised evolution

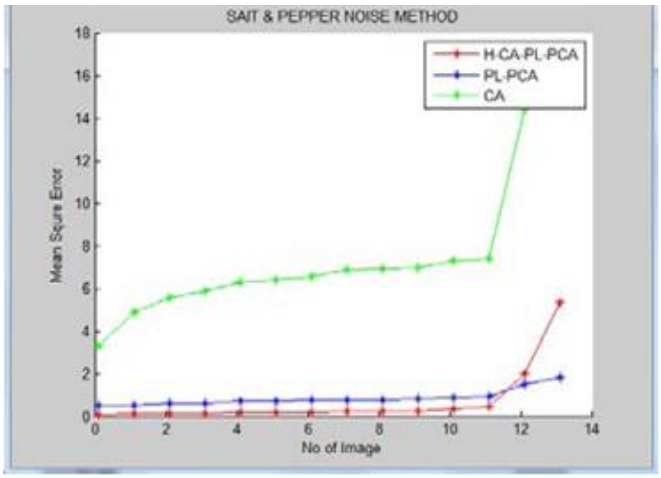

Fig. 10. Salt and Pepper Noise Method for MSE Value

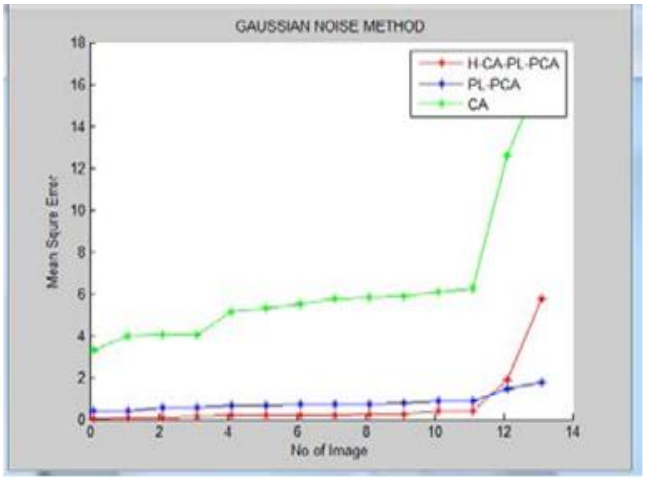

Fig. 11. Gaussian Noise Method for MSE Value

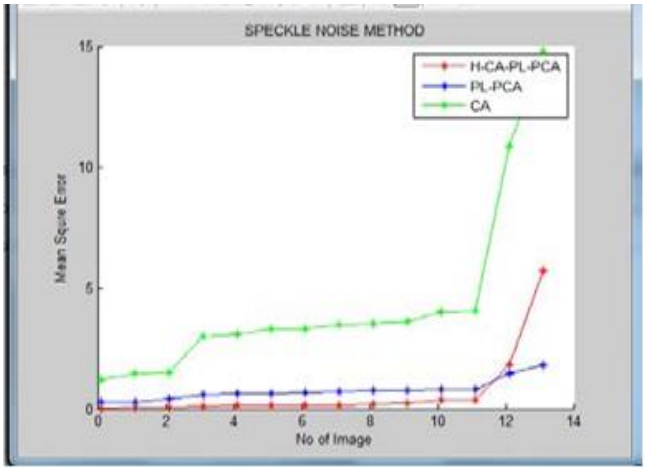

Fig. 12. Speckle Noise Method for MSE Value

Fig 10, 11 and 12 shows the relationship between MSE (Mean Squared Error) proportions in the context of standard images reveals that the proposed H-CA-PL-PCA approach consistently results in lower MSE proportions when compared to other CA and PL-PCA methods. This trend is observed across 14 different images subjected to the image compression process. MSE represents the average of the squared errors or deviations, which measures the disparity between the actual and estimated values. It's a risk function, corresponding to the expected value of the squared error loss. The discrepancies arise due to variations that might lead to a more accurate estimation. In the context of image denoising, a low MSE ratio indicates superior performance, and the proposed method effectively demonstrates this. Importantly, the performance of MSE proportions across images continues to improve, even as more images are included in the evaluation.

8. Conclusion

Image denoising is a crucial aspect of improving the quality of images in various image processing applications. In both cases, the examination of two algorithms demonstrates their effectiveness across all scenarios. The Cellular Automata (CA) algorithm is typically applied to image edges, enhancing their sharpness and contributing to overall image quality. On the other hand, the Patch-Level Principal Component Analysis (PL-PCA) algorithm is employed to cover specific regions of the image, effectively reducing noise and enhancing image clarity. By combining these two approaches, both edge and interior image disturbances can be efficiently removed. The hybrid approach employs various techniques, including the Wiener filter, Stationary Wavelet Transform (SWT) for boundary delineation, and the application of a 3x3 patch grid at the input stage. This combination proves highly effective in achieving superior denoising results for images.

Acknowledgments

We thank Suresh A to develop the algorithms. We are grateful to Ranjith Kumar S and Nithya N.S, for modifications in Algorithms. We thank Jothi.K R for her valuable corrections, editing and reviews of this manuscript.

References

[1] V. R. Mol and P. U. Maheswari, "A Survey on Restoration of Paintings," 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2020, pp. 102-108, doi: 10.1109/ICCSP48568.2020.9182422.

[2] P. L. Rosin, "Training Cellular Automata for Image Processing," IEEE Transactions On Image Processing, vol. VOL. 15, pp. 1-12, 2006, http://dx.doi.org/10.11 0 9/TIP.2006.877040.

[3] Zhang, M., & Gunturk, B. K. (2008). Multiresolution bilateral filtering for image de-noising. IEEE Transactions on image processing, 17(12), 2324-2333.

[4] Liu, Y. L., Wang, J., Chen, X., Guo, Y. W., & Peng, Q. S. (2008). A robust and fast non-local means algorithm for image de-noising. Journal of Computer Science and Technology, 23(2), 270-279.

[5] Zeng, W. L., & Lu, X. B. (2011). Region-based non-local means algorithm for noise removal. Electronics letters, 47(20), 1125-1127.

[6] Gorsevski, P. V., Onasch, C. M., Farver, J. R., & Ye, X. (2012). Detecting grain boundaries in deformed rocks using a cellular automata approach. Computers & Geosciences, 42, 136-142.

[7] Wang, X., & Luan, D. (2013). A novel image encryption algorithm using chaos and reversible cellular automata. Communications in Nonlinear Science and Numerical Simulation, 18(11), 3075-3085.

[8] Zhang, Y., Liu, J., Li, M., & Guo, Z. (2014). Joint image de-noising using adaptive principal component analysis and self-similarity. Information Sciences, 259, 128-141.

[9] Dolui, S., Patarroyo, I. C. S., & Michailovich, O. V. (2014). Generalized non-local means filtering for image de-noising. In IS&T/SPIE Electronic Imaging (pp. 90190B-90190B). International Society for Optics and Photonics.

[10] Zhong, S., &Oyadiji, S. O. (2007). Crack detection in simply supported beams without baseline modal parameters by stationary wavelet transform. Mechanical Systems and Signal Processing, 21(4), 1853-1884.

[11] Wongthanavasu, S., & Tangvoraphonkchai, V. (2007). Cellular automata-based algorithm and its application in medical image processing. In Image Processing, 2007. ICIP 2007. IEEE International Conference on (Vol. 3, pp. III-41). IEEE.

[12] Zhang, L., Dong, W., Zhang, D., & Shi, G. (2010). Two-stage image de-noising by principal component analysis with local pixel grouping. Pattern Recognition, 43(4), 1531-1549.

[13] S. Saladi and N. Amutha Prabha, "Analysis of de-noising filters on MRI brain images", International Journal of Imaging Systems and Technology, vol. 27, no. 3, pp. 201-208, 2017.

[14] S. Suhas and C R Venugopal, "An efficient MRI noise removal technique using linear and nonlinear filters", International Journal of Computer Applications, vol. 179, no. 15, pp. 17-20, January 2018.

[15] Ran, Q., Xu, X., Zhao, S. et al. Remote sensing images super-resolution with deep convolution networks. Multimed Tools Appl 79, 8985–9001 (2020). https://doi.org/10.1007/s11042-018-7091-1

[16] van Zijl, L. (2014). Content-Based Image Retrieval with Cellular Automata. In: Rosin, P., Adamatzky, A., Sun, X. (eds) Cellular Automata in Image Processing and Geometry. Emergence, Complexity and Computation, vol 10. Springer, Cham. https://doi.org/10.1007/978-3-319-06431-4_8

[17] E. Luo, S. H. Chan and T. Q. Nguyen, "Adaptive Image Denoising by Targeted Databases," in IEEE Transactions on Image Processing, vol. 24, no. 7, pp. 2167-2181, July 2015, doi: 10.1109/TIP.2015.2414873.

[18] Migenda N, Möller R, Schenck W. Adaptive dimensionality reduction for neural network-based online principal component analysis. PLoS One. 2021 Mar 30; 16(3):e0248896. doi: 10.1371/journal.pone.0248896. PMID: 33784333; PMCID: PMC8009402.

[19] H. M. Ali, 'MRI Medical Image Denoising by Fundamental Filters', High-Resolution Neuroimaging - Basic Physical Principles and Clinical Applications. InTech, Mar. 14, 2018. doi: 10.5772/intechopen.72427
[20] Bizhani, M., Ardakani, O.H. & Little, E. Reconstructing high fidelity digital rock images using deep convolutional neural networks. Sci Rep 12, 4264 (2022). https://doi.org/10.1038/s41598-022-08170-8