

Article

Fuzzy Inference Systems to Fine-Tune a Local Eigenvector Image Smoothing Method

Khleef Almutairi ^{1,2,*} , Samuel Morillas ¹  and Pedro Latorre-Carmona ³ 

¹ Instituto de Matemática Pura y Aplicada, Universitat Politècnica de València, Camino de Vera s/n, 46022 València, Spain; smorillas@mat.upv.es

² Mathematics Department, Faculty of Science, Albaha University, Al Baha 65779-7738, Saudi Arabia

³ Departamento de Ingeniería Informática, Universidad de Burgos, Avda. Cantabria s/n, 09006 Burgos, Spain; plcarmona@ubu.es

* Correspondence: kkalmuta@doctor.upv.es

Abstract: Image denoising is a fundamental research topic in colour image processing, analysis, and transmission. Noise is an inevitable byproduct of image acquisition and transmission, and its nature is intimately linked to the underlying processes that produce it. Gaussian noise is a particularly prevalent type of noise that necessitates effective removal while ensuring the preservation of the original image's quality. This paper presents a colour image denoising framework that integrates fuzzy inference systems (FISs) with eigenvector analysis. This framework employs eigenvector analysis to extract relevant information from local image neighbourhoods. This information is subsequently fed into the FIS system which dynamically adjusts the intensity of the denoising process based on local characteristics. This approach recognizes that homogeneous areas may require less aggressive smoothing than detailed image regions. Images are converted from the RGB domain to an eigenvector-based space for smoothing and then converted back to the RGB domain. The effectiveness of the proposed methods is established through the application of various image quality metrics and visual comparisons against established state-of-the-art techniques.

Keywords: colour image processing; fuzzy inference system; eigenvector analysis; Gaussian noise



Citation: Almutairi, K.; Morillas, S.; Latorre-Carmona, P. Fuzzy Inference Systems to Fine-Tune a Local Eigenvector Image Smoothing Method. *Electronics* **2024**, *13*, 1150. <https://doi.org/10.3390/electronics13061150>

Academic Editor: Manohar Das

Received: 24 February 2024

Revised: 13 March 2024

Accepted: 16 March 2024

Published: 21 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the visual information era image processing and computer vision dominate across many research fields. Denoising and restoring the quality of the distorted images is an essential procedure in the realm of computer vision systems. In the last decades, colour image processing has become a relevant research topic, and therefore colour image smoothing, a mandatory pre-processing step, is one of its important research branches.

Noise is almost unavoidable in digital images during the acquisition and transmission processes. This is particularly evident in the form of unpredictable variations of random motion within the image information or in the pixel brightness [1]. Among different noise sources, thermal noise (Johnson–Nyquist) is notably known as a result of the disruption of the charge-coupled device sensor (CCD) [2]. This type of noise can be characterized by a substantial amount of diverse phenomena and events, which commonly satisfy Gaussian distribution conditions [3]. Thus, this sort of noise is simulated by adding random values from a zero-mean Gaussian distribution into the initial values of each image channel separately, where the noise intensity is represented by the standard deviation, s , of the Gaussian distribution [4].

Several methods proposed in the literature for suppressing undesired Gaussian noise from the colour images share the same common aims [4–7]:

1. Homogeneous regions in the image should be completely smoothed.
2. Detailed regions in the image should be *cautiously* denoised, avoiding unnecessary sharpening, blurring or introducing other distortions.

3. The denoising process should not introduce any colour artifacts to the resulting images.

In Point 3, it is crucial to highlight that the introduction of artifacts in the form of new and/or non-existing colours in the images can be specially attributed to not properly taking the correlation among the image channels into account [4–6]. This explains why the channel-wise use of grey-level-based image smoothing methods is not appropriate for colour image denoising.

The first colour image denoising methods are based on linear assumption approaches. One of the renowned methods that use this approach is the arithmetic mean filter (AMF) [4]. AMF is effective in noise suppression due to the use of the zero-mean property. However, it introduces a blurring effect, which is particularly perceptible in the image details and structures. This limitation in linear approaches has motivated researchers to explore non-linear methods with the aim of preserving the edges from unwanted blurring by detecting the details of the image and therefore less smoothing in these important parts of the image.

Many non-linear methods have leveraged the demonstrated advantage of the zero-mean property and are still able to preserve the edges and improve noise reduction. A widely known approach is the bilateral filtering method (BF) [8], as well as some of the methods based on it [9–12].

On the other hand, there is a wide variety of non-linear methods based on different types of approaches. Some of them are based on the use of wavelets, which are used to decompose a signal into different scales or resolutions [13]. An example of this filtering method is the so-called collaborative wavelet filter (CWF) that is proposed in [14,15]. The CWF involves grouping similar 2D image fragments into 3D using what is called collaborative altering, containing jointly filtered grouped image blocks, reducing noise while preserving unique features. Another example uses data regularization and the wavelet transformation discussed in [16].

Another filtering method is based on analysing the structure of the local graph computed at each pixel using the neighbour pixel information. In this particular case, a local graph is used to assess the spatial relationship between the pixels within the image. An example of this approach can be found in the graph method for simultaneous smoothing and sharpening (GMS^3) and its normalised version ($NGMS^3$) [17].

Machine learning has garnered significant attention as an alternative for noise reduction. One such method leverages deep learning through a specific neural network type called DnCNN [18]. This network utilizes residual learning and batch normalization to achieve faster training and enhance performance. Notably, the DnCNN model can remarkably handle Gaussian noise without prior knowledge of its intensity (blind denoising). However, these neural network-based methods often face challenges in interoperability and explaining the underlying mechanisms behind their noise reduction capabilities.

Other/alternative methods are based on linear algebra mathematical roots. Among these, we can mention those based on principal component analysis (PCA) [19–21]. An alternative method is based on the eigenvector decomposition applied in the feature/color space instead of the image spatial domain. This method is called the eigenvector analysis filter (EIG) [22]. EIG effectively uses the weighted pixel averaging mechanism in order to mitigate the noise from the colour images. This method in particular is thoroughly discussed in the following section, as the proposed methods leverage the information extracted by EIG.

To improve EIG performance, we propose to include fuzzy inference systems to better manage the information extracted from the local neighbourhoods on which the eigenvector analysis is performed. Fuzzy logic was introduced by Zadeh in [23]. His groundbreaking work aimed at imitating the human thinking and decision process by incorporating the ability to reason when dealing with uncertain data. Fuzzy logic is able to deal with many possibilities by mapping the diverse responses onto a continuum. Moreover, fuzzy logic facilitates deriving meaningful conclusions even from imprecise or incomplete knowledge. This remarkable versatility has led to its widespread adoption across various scientific and technological fields. The field of computer vision has seen many studies

highlighting the crucial advantages of using fuzzy logic. Specifically, researchers have employed fuzzy inference systems (FIS) [24] to reduce noise from the colour images, as shown in studies [25,26]. For instance, in [27], authors implemented a two-step fuzzy approach in order to remove white additive noise (Gaussian noise). In [28], we studied a preliminary application of FIS in an image smoothing framework. In this work, we extend this study by including more accurate and complex approaches so that we really find a system that effectively eliminates the Gaussian noise. Moreover, we optimize the system to handle all three channels separately, taking advantage of non-correlation in the extracted data. In these methods, the image is transformed from the RGB domain to the local eigenvector space in order to perform the analysis of the correlation of the image channels. Subsequently, three descriptive statistic features are extracted from each 3×3 kernel vector field, which is therefore used as inputs of the FIS system.

It is critical to emphasize that the proposed methods diverge from EIG filtering methods in several aspects. A noteworthy distinction is that EIG relies on the normalized standard deviation for the smoothing process, whereas FIS also utilizes the local standard deviation for each component channel as input data for the inference systems. Moreover, the membership functions and the set of fuzzy rules are employed to determine the extent of smoothing in the three image channels based on the information within the channels. Consequently, if the channel within the homogeneous areas contains information that does not need to be preserved, i.e., it is identified as noise, FIS applies a higher smoothing intensity. Conversely, if the information resides within the detailed part of the images, FIS employs a gentle smoothing to retain these crucial image details. Section 3 provides a more detailed explanation of the whole filtering process. The extensive experiments and testing produced promising results, demonstrating the superiority of the proposed FIS compared to state-of-the-art methods. The main advantages of the proposed methods are that, by including a fuzzy inference system that decides the intensity of the smoothing performed, they are able to significantly improve the noise reduction capability of the EIG filter while keeping a good performance from the detail preservation point of view.

The rest of the paper is organized as follows: Section 2 presents the general eigenvector-based smoothing framework our proposal stems from. Section 3 describes the FIS-based filtering methodology. Section 4 shows the experimental results and discusses its performance in relation to other state-of-the-art techniques. Finally, Section 5 summarizes the main outcomes and discusses potential future research avenues.

2. Eigenvector-Based Smoothing Framework

This section describes in some detail the eigenvector analysis-based filter [22] which may be considered a precursor idea of the research presented in this paper. Let us assume the colour image \mathbf{F} , which is represented in the RGB colour space, is processed using a sliding filtering window of size $N \times N$ where $N = 2n + 1$ and $n = 1, 2, \dots$. The sliding window is centered on each pixel to be processed, denoted by \mathbf{F}_0 , and defined by term (F_0^R, F_0^G, F_0^B) of its three RGB colour components. The rest of the neighbour pixels in the filtering window are denoted as $\mathbf{F}_i, i = 1, \dots, N^2 - 1$.

Using the colour component values of the pixels in the filtering window, we build a data matrix \mathbf{D} of size $N^2 \times 3$ where the columns of the matrix are associated to the colour components. Each row in this matrix represents each one of the pixels. The main novelty of the method is that an analysis of the \mathbf{D} matrix is used to (i) appropriately process the correlation among the image channel and (ii) conveniently smooth the noise in the image while preserving the original structures. We propose to perform an eigenvector analysis based on the information provided by matrix \mathbf{D} . For this, we find the eigenvectors, also called characteristic vectors or principal components, of $\mathbf{D}^T \mathbf{D}$, where T denotes matrix transpose. This procedure is behind well-known methods based on Principal Component Analysis (PCA) [29,30].

The method of principal components is based on a key result from matrix linear algebra: since $\mathbf{D}^T \mathbf{D}$ is a symmetric matrix, it may be reduced to a diagonal matrix \mathbf{L} by

premultiplying and postmultiplying it by a particular orthonormal matrix \mathbf{O} such that the diagonal elements of \mathbf{L} are called the characteristic roots, latent roots or eigenvalues, and the columns of \mathbf{O} are called the characteristic vectors, eigenvectors or latent vectors of $\mathbf{D}^T\mathbf{D}$ [29,30]. That is, vector \mathbf{v} is an eigenvector of $\mathbf{D}^T\mathbf{D}$ if and only if it satisfies

$$\mathbf{D}^T\mathbf{D}\mathbf{v} = \lambda\mathbf{v}, \quad (1)$$

where λ is a scalar called the eigenvalue corresponding to \mathbf{v} and, for convenience, \mathbf{v} is taken so that it is unitary. Eigenvalues λ_i of $\mathbf{D}^T\mathbf{D}$ can be obtained as the solutions of equation

$$\det(\mathbf{D}^T\mathbf{D} - \lambda\mathbf{I}) = 0, \quad (2)$$

where \det denotes the matrix determinant. Then, given the non-null eigenvalues λ_i , we can obtain [29,30] three associated eigenvectors \mathbf{v}_i from eigenvalue equations

$$(\mathbf{D}^T\mathbf{D} - \lambda_i\mathbf{I})\mathbf{v}_i = 0, \quad (3)$$

that can be considered as an alternative set of orthogonal coordinate axes. Transforming the original data by means of the coordinate axis provided by the eigenvectors implies transforming the original correlated variables into a new set of variables which are uncorrelated. Geometrically, this procedure is simply a principal axis rotation of the original coordinate axis about their means [29,30]. Therefore, if we denote by \mathbf{V} the 3×3 orthonormal matrix that has as columns the three eigenvectors of $\mathbf{D}^T\mathbf{D}$ denoted as \mathbf{V}^1 , \mathbf{V}^2 , and \mathbf{V}^3 , the mentioned transformation is performed by multiplying \mathbf{D} by \mathbf{V} so that

$$\mathbf{U} = \mathbf{D}\mathbf{V}, \quad (4)$$

where \mathbf{U} denotes the matrix containing the transformed data, also called the scores matrix, and each pixel $\mathbf{U}_i, i = 0, \dots, N^2 - 1$ is now represented by the term $\mathbf{U}_i = (U_i^1, U_i^2, U_i^3)$. Moreover, note that, since \mathbf{V} is orthonormal, it is fulfilled that

$$\mathbf{U}\mathbf{V}^T = \mathbf{D}. \quad (5)$$

Now, we can directly operate on the values of \mathbf{U} to reduce the noise. Notice that now the columns of \mathbf{U} are associated to a new set of uncorrelated variables that we denote as U^1 , U^2 , and U^3 , and which are associated to eigenvectors \mathbf{V}^1 , \mathbf{V}^2 , and \mathbf{V}^3 , respectively. This implies that we can safely apply component-wise methods to reduce the noise independently in each of the new variables.

We aim at applying a weighted averaging operation in order to smooth each component independently. Then, to smooth each component of the pixel represented by the term $\mathbf{U}_0 = (U_0^1, U_0^2, U_0^3)$, the operation given by the following expression is applied:

$$\hat{U}_0^i = \frac{\sum_{p=0}^{N^2-1} W_p^i U_p^i}{\sum_{p=0}^{N^2-1} W_p^i}, \quad i = 1, 2, 3. \quad (6)$$

where i refers to the colour channel and p to the pixel number in the neighbourhood window around a pixel.

According to above, weights W_p^i should be computed so that component U_0^i is less smoothed when $\sigma(U^i) \gg \sigma(U^j)$ and $\sigma(U^i) \gg \sigma(U^k)$, and more smoothed otherwise. For this, we define the *normalized* standard deviation σ_n of a variable U^i as

$$\sigma_n(U^i) = \frac{\sigma(U^i)}{\sum_{j=1}^3 \sigma(U^j)}. \quad (7)$$

To appropriately perform the averaging, weights W_p^i should be computed using a decreasing function on $|U_p^i - U_0^i|$ so that only U_p^i values close to U_0^i receive high weights. For this, we use the following exponential-based expression, but any other decreasing function could be used instead, as well:

$$W_p^i = \exp\left(-\frac{|U_p^i - U_0^i|\sigma_n(U^i)}{D}\right), \quad (8)$$

where $D > 0$ is a filter parameter that tunes the global smoothing capability of the method. It can be seen that larger values of D imply that values of W_p^i are closer to 1 and therefore the smoothing capability is higher. Conversely, for lower values of D , the smoothing capability decreases. The appropriate setting of $D = (5/6) \times s$ where s is the noise standard deviation. Note that the value given by $\sigma_n(U^i)$ is also related to the smoothing capability: for lower values of $\sigma_n(U^i)$, the smoothing capability increases, whereas for higher values of $\sigma_n(U^i)$, the smoothing performed is lower. Consequently, the desired behaviour is achieved.

Notice that this denoising method is based on computing the eigenvectors and eigenvalues of a data matrix and that this procedure, followed in Equations (1)–(3), is very sensitive to outliers in the data. Therefore, whereas this analysis is accurate in a context of white (zero-mean) noise, it is not expected to be useful for other types of image noise as impulse, salt-and-pepper or multiplicative noise unless a robust eigenvector analysis is performed.

3. Smoothing Colour Images Using a Fuzzy Inference System

The EIG filter has shown its effectiveness in eliminating unwanted noise while preserving important image information (spatial structure and details). However, EIG presents some limitations in the noise reduction capability when considering large homogeneous regions. This limitation arises from the use of the normalized standard deviations in Equation (8). When the original standard deviations are similar, their normalized counterparts all become close to $\frac{1}{3}$. This issue arises from the assumption that these standard deviations reflect meaningful information about the level of detail or the presence of edges within the image.

However, this assumption does not always hold true, particularly in large homogeneous regions. In homogeneous regions, where the data do not exhibit much variation, this results in similar standard deviation values. Consequently, the normalized standard deviation values in Equation (8) also become approximately equal to $\frac{1}{3}$. This situation poses a problem because Equation (8) uses these values to determine the denoising strength, and with all values close to $\frac{1}{3}$, the denoising effect becomes less intensive than desired, as in homogeneous regions, averaging can be safely applied. This means that the method may not effectively remove the noise, especially in homogeneous regions.

In this work, we propose the introduction of a new smoothing coefficient, C , in Equation (9) to address this limitation. The goal is to provide more control over the denoising strength and adapt it to different data characteristics, particularly in homogeneous regions. Smoothing coefficient C modulates the normalized standard deviation which leads to a stronger smoothing effect, particularly in homogeneous regions where all values are close to $\frac{1}{3}$. The new equation to set the weights is the following, instead of Equation (8):

$$W_p^i = \exp\left(-\frac{|U_p^i - U_0^i|\sigma_n(U^i)C_i}{D}\right). \quad (9)$$

where $\sigma_n(U^i)$ is given in Equation (7).

Adjusting smoothing coefficient C across different image elements is crucial. The key challenge lies in controlling C for smoothing intensity. In homogeneous areas with minimum details, we would like C to have low values (ideally, close to 0), allowing for a strong smoothing behaviour, in order to effectively remove noise, without compromising details. Conversely, areas with significant noise and edges require gentle smoothing (C close

to one) to preserve essential features while subtly reducing noise. However, intermediate areas (between zero and one) demand a degree of smoothing tailored to their specific characteristics. This reasoning effectively balances noise reduction and detail preservation, leading to improved image quality.

In order to determine smoothing coefficients C , we use local image features that capture the level of detail and noise in its surrounding area. Specifically, we rely on the standard deviations as inputs, denoted by $\sigma(U^i)$, $\sigma(U^j)$, and $\sigma(U^k)$, as described in Section 2, that fulfill $\sigma(U^1) < \sigma(U^2) < \sigma(U^3)$. In particular, we propose two different ways for this that in turn generate two filtering methods, as follows:

- **FuzzyEIG1:** It considers the absolute standard deviation of each channel as its input value. This aligns with the intuitive idea that higher standard deviations often indicate the presence of image structures to be preserved while lower values are associated to flat regions where we can apply more smoothing.
- **FuzzyEIG2:** It takes two input values given by $D1 = |\sigma(U^j) - \sigma(U^i)|$ and $D2 = |\sigma(U^k) - \sigma(U^j)|$. We also observed that when image structures are present, there are big differences between the standard deviation in the channels, while lower differences indicate flat regions.

According to this, the two sets of implication rules we propose to use are the following: In FuzzyEIG1, we consider two rules for each channel ($i = 1, 2, 3$):

- **IF** σ_i is *low*. **THEN** C_i is *low*.
- **IF** σ_i is *high* **THEN** C_i is *high*.

On the other hand, we specify the following set of two rules for FuzzyEIG2:

- **IF** $|\sigma_1 - \sigma_2|$ is *low* **AND** $|\sigma_2 - \sigma_3|$ is *low* **THEN** C_i is *low*.
- **IF** $|\sigma_1 - \sigma_2|$ is *high* **AND** $|\sigma_2 - \sigma_3|$ is *high* **THEN** C_i is *high*.

As these sets of rules include linguistic variables, the way to make them computational and finally obtain values for C_i is through the use of fuzzy theory, which handles the inherent vagueness and uncertainty of natural language. As described in [31], any fuzzy inference system (FIS) approach is formed by three key stages: (a) fuzzification, (b) inference, and (c) defuzzification. During fuzzification, FuzzyEIG1 considers the absolute values of $\sigma(U^i)$, $\sigma(U^j)$, and $\sigma(U^k)$ as input values.

These crisp values are then converted into linguistic terms (“low”, and “high”) using membership functions tailored to their specific ranges. Different linguistic terms are considered for each input due to their varying standard deviation ranges. Similarly, FuzzyEIG2 considers $D1$ and $D2$. These crisp values are also translated into linguistic terms using other membership functions. This process allows for FIS to reason about the different noise patterns detected through these different values.

Fuzzy rules work in conjunction with fuzzy sets, mimicking human reasoning by connecting linguistic terms like “when σ_i is low” to consequences like “the smoothing coefficient C_i is low” in the inference step. Finally, the defuzzification step converts the certainty of the consequents into crisp values of the output variable, C_i . For both systems and both input/output variables, Gaussian membership functions are considered, with two fuzzy sets (called low and high). While alternative membership functions could be used, Gaussian functions minimize the number of parameters involved (location and width), reducing the complexity as well.

Figure 1 illustrates the scheme of the proposed method visualizing the processing steps applied to a pixel. We begin by introducing the pixel, which is considered along with its neighboring pixels. Next, a data matrix, denoted by D in the equations above, is extracted. Eigenvector analysis is then performed to obtain the standard deviation for each color channel, $(\sigma_1, \sigma_2, \sigma_3)$. Then, $(\sigma_1, \sigma_2, \sigma_3)$ are used as inputs for the Fuzzy Inference System (FIS) which computes the smoothing coefficients (C_1, C_2, C_3) . Subsequently, weights W_p^i are calculated using Equation (9). And last, Equation (6) is employed to perform the smoothing

operation in the eigenvector space and the processed pixel is converted back from the eigenvector space to the RGB color space using Equation (5), resulting in smoothed pixel \hat{F}_0 .

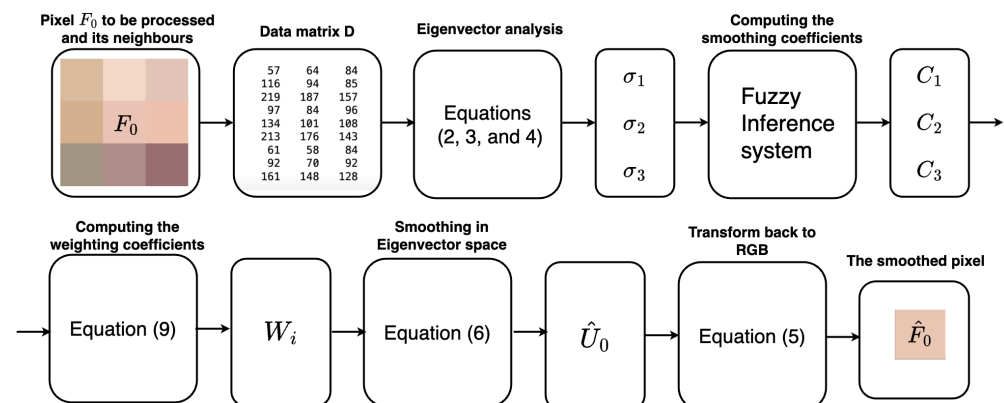


Figure 1. Flowchart of the Gaussian denoising proposed method that outlines the processing steps applied to a single pixel.

4. Experimental Setting of the Proposed Methods, Validation and Discussion

4.1. Optimization of the Membership Functions

Parameter optimization offers a standard and reliable tool to enhance the numerical accuracy of fuzzy inference systems while maintaining their interpretability [32]. There is a plethora of techniques that address this process, with genetic algorithms (GAs) [33] demonstrating considerable success in optimizing both structure (i.e., rule definition and system architecture) and membership function parameters within each fuzzy subset.

Genetic algorithms (GAs) represent a class of global stochastic optimization methods. Their foundation lies in the principles of Darwinian *survival of the fittest* and natural selection (in evolutionary theory applied to animal species). This well-established approach has shown its effectiveness as an optimization technique in challenging real-world problems, particularly in the domains of design and combinatorial optimization [34].

In this context, and for each generation, a specific number of individuals is randomly created, forming the so-called initial population of candidate solutions. Subsequent generations are created through the application of variation, selection, and inheritance principles. Algorithm 1 establishes how these methodologies work.

Algorithm 1: Genetic Algorithm

- 1 **Initialization:** $t = 0$. Generate a random population $P(t)$ of n chromosomes.
Define the crossover probability P_{cross} and the mutation probability P_{mut}
 - 2 **Selection:** Select two parent chromosomes, p_1 and p_2 from $P(t)$.
 - 3 **Crossover (with probability P_{cross}):** If random value $\in [0, 1] \leq P_{\text{cross}}$, perform crossover operation on p_1 and p_2 to create offspring o_1 and o_2 .
 - 4 **Mutation (with probability P_{mut}):** For each bit in o_1 and o_2 If random value $\in [0, 1] \leq P_{\text{mut}}$, flip the bit value.
 - 5 **Evaluation:** Calculate the fitness of o_1 and o_2 .
 - 6 **Selection (for replacement):** Select two chromosomes c_1 and c_2 from $P(t) \cup \{o_1, o_2\}$.
 - 7 **Replacement:** Replace c_1 and c_2 in $P(t)$ with o_1 and o_2
 - 8 **Increment generation counter:** $t = t + 1$
 - 9 **Output:** Best chromosome found in any generation
-

In our case, a genetic algorithm (GA) is applied in order to optimize the input and output membership functions independently, for each one of the FIS methods, with the aim to enhance the noise reduction performance of Fuzzy Inference System (FIS) methods. To achieve this, we select a diverse set of three images *Pills* (50×50), *Headphone* (200×200), and

Parrots (80×80) shown in Figure 2. These images are subjected to varying levels of Gaussian noise, $s = 10, 20, 30$. Meanwhile, the optimization process itself focuses on a single noise level at a time for the three images. In order to assess the quality of noise reduction achieved by the optimized membership functions, we employ the Peak Signal-to-Noise Ratio (PSNR) metric [4]. The key advantage of this independent optimization lies in its ability to account for the inherent differences in value ranges across various input and output variables. For example, the concept of “low” might have significantly different interpretations for each level of noise. By optimizing independently, we ensure that the membership functions adapt effectively to these diverse ranges of noise levels. The optimization process itself uses a fitness function that aims to minimize the mean squared error (MSE) between the original and denoised pixel colour values. The optimization process starts with the third channel, which typically holds the most part (the most important) information, followed by the second channel using the smoothing coefficients obtained from the optimization of the third one. Finally, the first channel is optimized using the smoothing coefficients from both the third and the second channels. This sequential approach ensures that channels with more prominent information are the guide for the noise reduction process in subsequent channels. The input membership functions in both FuzzyEIG1 and FuzzyEIG2 methods are restricted to $[0-200]$ for both the location centre and width parameters. Output membership functions are in the $[0-1]$ interval for both the centre and the width parameters.



Figure 2. Group of images selected for the training stage. (a) Pills (50×50); (b) Parrots (80×80); (c) Head-Phone (200×200).

The parameters corresponding to the optimized membership functions for FuzzyEIG1 and FuzzyEIG2 are shown in Tables 1 and 2, respectively. These tables can be used as references for parameter setting when using the proposed methods.

In general, the membership function parameters change substantially across different noise levels, somehow *revealing* an adaptation to the noise level. The main tendency is related to membership functions increasing in overlapping as the noise increases, which reflects the adaptation that the method shows as the uncertainty in the data increases. This uncertainty is related to the location of the membership functions being closer or the width being enlarged.

This adaptive behaviour allows for the system to deal with higher standard deviation values that might be present in the data. However, the impact of noise varies across inputs, with the first input being more affected due to its association with less correlated data variations. Table 2 shows, for FuzzyEIG2, that noise also shifts the location of the output functions. This leads to higher smoothing coefficients, which is a desirable outcome for noise reduction.

Table 1. Optimized parameters of the FuzzyEIG1 membership functions for the input and output stages.

Optimized Parameters for the Inputs												
SD	Input ($\sigma(U^i)$)/Output (C1)				Input ($\sigma(U^i)$)/Output (C2)				Input ($\sigma(U^k)$)/Output (C3)			
MF	Low		High		Low		High		Low		High	
Params.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.
s = 10	86.73	97.82	158.06	14.28	16.64	134.95	41.91	82.78	16.44	6.93	17.02	34.66
s = 20	106.96	3.84	22.52	84.22	112.00	103.57	24.75	93.79	11.61	31.19	6.58	46.54
s = 30	63.04	152.90	15.68	72.15	133.34	168.30	27.79	116.87	26.25	18.22	14.84	85.29

Optimized Parameters for the Outputs												
Params.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.
s = 10	0.16	0.07	0.08	0.40	0.060	0.04	0.26	0.90	0.03	0.09	0.02	0.94
s = 20	0.01	0.01	0.09	0.63	0.07	0.19	0.87	0.72	0.10	0.16	0.15	0.65
s = 30	0.04	0.05	0.40	0.47	0.086	0.07	0.80	0.55	0.06	0.04	0.05	0.55

Table 2. Optimized parameters of the FuzzyEIG2 membership functions for the input and output stages.

Optimized Parameters for the Inputs																								
Inputs	$D1 = \sigma(U^j) - \sigma(U^i) $				$D2 = \sigma(U^k) - \sigma(U^i) $				$D1 = \sigma(U^j) - \sigma(U^i) $				$D2 = \sigma(U^k) - \sigma(U^i) $											
MFs	Low		High		Low		High		Low		High		Low		High		Low		High					
Params.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.				
s = 10	139.56	46.26	70.18	148.93	36.52	188.33	14.23	26.41	113.56	173.52	36.07	105.37	2.26	22.37	53.28	164.90	79.45	184.07	69.54	43.52	8.81	3.03	162.23	70.76
s = 20	50.85	106.26	21.33	180.43	2.84	14.13	45.23	188.19	10.78	8.17	7.07	82.04	69.23	52.90	34.71	193.97	36.63	172.34	1.25	143.91	15.30	0.80	27.77	83.69
s = 30	79.24	95.38	56.10	189.90	83.90	123.27	28.42	128.19	108.21	197.57	49.94	181.76	41.50	29.98	20.84	185.57	133.52	67.14	24.51	77.90	23.27	3.15	15.46	54.90

Optimized Parameters for the Outputs												
Outputs	C1				C2				C3			
MFs	Low		High		Low		High		Low		High	
Params.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.	Loc.	Wid.
s = 10	0.158	0.095	0.158	0.671	0.146	0.311	0.055	0.755	0.017	0.221	0.060	0.970
s = 20	0.027	0.136	0.148	0.747	0.164	0.077	0.004	0.468	0.009	0.033	0.292	0.879
s = 30	0.032	0.099	0.511	0.333	0.140	0.082	0.066	0.622	0.097	0.0009	0.202	0.788

4.2. Denoising Results and Discussion

Following the optimization stage, we tested the methodology by using a validation set of four images with different sizes: *Beach* (100 × 100), *Lena* (90 × 90), *Grass* (200 × 200), and *Micro* (51 × 51) shown in Figure 3. These validation images, similar to the training images, were corrupted with varying levels of Gaussian levels: $s = \{10, 20, 30\}$. To comprehensively assess the performance of our method, we employed five different metrics:

- Mean absolute error (MAE) [4]: It is particularly aimed at assessing detail preservation.
- Peak signal-to-noise ratio (PSNR) [4]: It measures noise cancellation.
- Normalized color difference (NCD) [4]: It evaluates colorimetric preservation.
- Fuzzy color structural similarity (FCSS) [35]: It is a color extension of the Structural Similarity measure, SSIM [36], that focuses on evaluating perceptual similarity.
- Perceptual difference inspired by iCAM (iCAMd) [37]: It measures colour changes from a perceptual point of view.

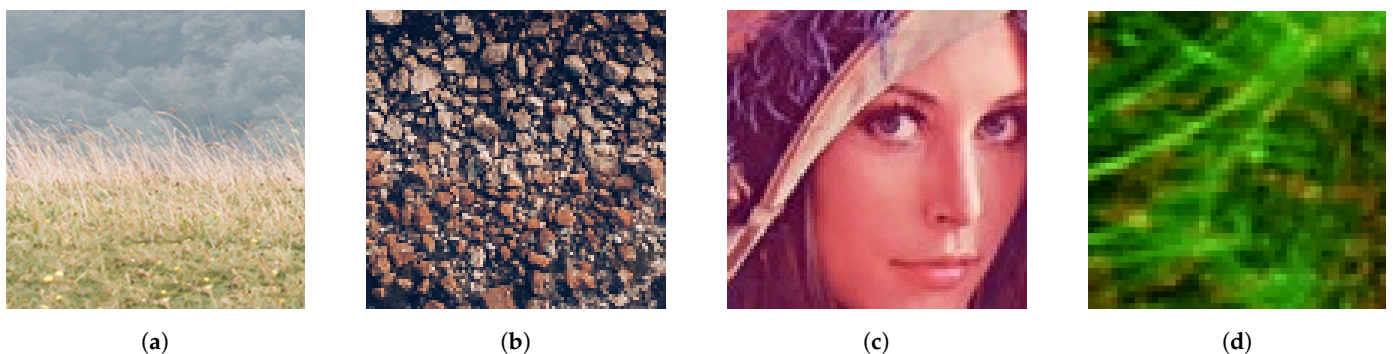


Figure 3. Group of images selected for the validation stage. (a) Grass (200 × 200); (b) Beach (100 × 100); (c) Lena (90 × 90); (d) Micro (51 × 51).

We compared the performance of our proposed approaches, FuzzyEIG1 and FuzzyEIG2, with several state-of-the-art denoising methods: the collaborative wavelet filter (CWF) [15], the eigenvector analysis method (EIG) [22], the method of Feed-forward denoising convolutional neural networks(DnCNN) [18], and the Graphs-based methods for simultaneous smoothing and sharpening (GMS3) as well as the Normalized graph-method for simultaneous smoothing and sharpening (NGMS3) [17].

All filters used a 3×3 window with the authors’ recommended settings. Notably, the sharpening processes involved in GMS^3 and $NGMS^3$ were ignored for a fair comparison. The experimental results are shown in Table 3, where the best and second-best results are highlighted in blue and red colours, respectively. Additionally, some of the method outputs are visualized in Figures 4 and 5, along with squared error maps for each output.

Table 3. MAE, PSNR, NCD ($\times 10^2$), FCSS ($\times 10$) and iCAMd results for FuzzyEIG1, FuzzyEIG2, and 5 other denoising methods, on images of different sizes contaminated with varying levels of Gaussian noise (s). For each noise level and metric, the best performing method is highlighted in blue, while the second-best is highlighted in red.

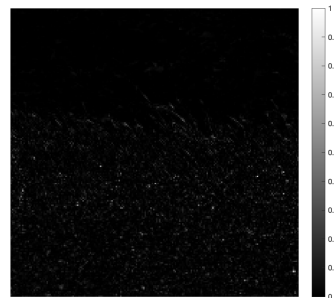
Filter	s = 10					s = 20					s = 30				
	MAE	PSNR	NCD	FCSS	iCAMd	MAE	PSNR	NCD	FCSS	iCAMd	MAE	PSNR	NCD	FCSS	iCAMd
Grass Image (200 × 200)															
None	7.82	28.22	5.37	9.38	3.38	15.56	22.24	10.75	8.69	6.71	23.23	18.79	16.16	7.97	10.04
GMS ³	5.23	31.16	2.91	9.51	2.72	8.94	26.71	5.65	9.19	4.93	14.68	22.53	9.85	8.70	7.70
NGMS ³	5.52	30.59	2.88	9.48	2.80	8.11	27.35	4.66	9.18	4.72	10.92	24.96	6.86	8.98	6.70
DnCNN	4.84	31.87	2.69	9.45	2.55	7.11	28.40	3.63	9.15	3.93	8.56	26.74	4.10	8.95	4.80
CWF	3.82	33.90	1.82	9.63	1.91	5.80	30.08	2.48	9.38	2.93	7.35	27.92	2.97	9.14	3.76
EIG	4.82	32.24	2.80	9.58	2.56	8.64	27.20	5.33	9.27	4.82	12.21	24.22	7.85	8.98	7.02
FuzzyEIG1	4.57	32.54	2.45	9.55	2.50	7.51	28.31	4.22	9.29	4.46	9.11	26.43	4.68	9.00	5.66
FuzzyEIG2	4.59	32.51	2.46	9.55	2.50	7.55	28.26	4.22	9.27	4.48	9.13	26.41	4.69	9.00	5.68
Beach Image (100 × 100)															
None	3.88	34.35	6.88	9.68	3.33	7.61	28.51	13.12	9.42	6.31	11.22	25.13	18.90	9.17	8.91
GMS ³	6.51	29.62	6.70	9.16	4.60	7.87	28.08	9.27	9.07	5.89	9.70	26.32	12.41	8.95	7.41
NGM ³	10.94	24.86	9.52	8.68	6.72	11.90	24.27	11.30	8.61	7.58	13.21	23.47	13.44	8.49	8.81
DnCNN	9.79	25.93	8.97	8.77	6.57	11.47	24.71	11.55	8.58	7.79	13.43	23.44	14.15	8.36	9.17
CWF	4.23	33.40	6.29	9.66	3.52	8.40	27.25	10.75	9.30	6.37	12.74	23.61	14.59	8.79	9.10
EIG	3.69	34.60	4.94	9.65	3.02	6.33	30.00	8.06	9.34	5.01	9.04	26.90	11.11	9.02	6.83
FuzzyEIG1	3.94	33.98	5.13	9.63	3.11	7.38	28.56	8.71	9.18	5.44	11.21	24.88	12.29	8.71	7.64
FuzzyEIG2	3.82	34.28	5.03	9.64	3.06	7.18	28.84	8.56	9.20	5.40	10.46	25.56	11.82	8.79	7.41
Lenna Image (90 × 90)															
None	7.64	28.33	9.11	9.34	5.19	14.88	22.54	17.49	8.63	10.25	21.85	19.17	25.75	7.95	15.39
GMS ³	4.87	31.62	4.95	9.49	3.31	8.53	26.99	9.18	9.20	6.24	13.72	22.89	15.25	8.68	10.26
NGMS ³	5.08	31.16	4.90	9.45	3.31	7.71	27.62	7.74	9.22	5.47	10.45	25.07	11.01	8.98	8.01
DnCNN	4.50	32.61	4.87	9.55	3.10	6.60	29.19	6.69	9.33	4.60	8.55	26.94	8.57	9.15	6.02
CWF	3.38	35.23	3.01	9.64	2.17	5.07	31.51	4.25	9.47	3.19	6.87	28.91	5.91	9.30	4.24
EIG	4.56	32.64	4.67	9.56	3.20	8.17	27.57	8.48	9.24	5.96	11.40	24.63	12.14	8.95	8.72
FuzzyEIG1	4.15	33.35	4.05	9.57	2.86	6.59	29.05	5.79	9.24	4.22	8.64	26.80	7.91	9.07	5.87
FuzzyEIG2	4.17	33.32	4.07	9.57	2.87	6.65	28.96	5.82	9.23	4.24	8.68	26.75	7.94	9.07	5.90
Micro Image (51 × 51)															
None	6.54	29.13	10.19	9.41	5.40	12.61	23.29	18.55	8.86	10.38	18.28	20.03	26.26	8.32	15.69
GMS ³	5.61	30.42	7.67	9.38	4.86	8.51	26.85	11.56	9.00	7.02	10.99	24.61	15.81	8.78	9.71
NGMS ³	5.91	30.00	7.77	9.35	5.09	8.37	26.98	11.85	9.04	6.77	10.76	24.88	15.94	8.81	9.61
DnCNN	5.69	31.26	9.51	9.36	4.13	8.61	27.72	13.35	9.00	6.16	11.08	25.57	17.63	8.84	8.92
CWF	4.70	32.49	7.27	9.49	3.82	8.11	27.93	11.94	9.09	6.16	10.99	25.48	16.62	8.78	8.97
EIG	4.97	31.75	7.30	9.47	4.19	7.95	27.88	11.35	9.08	6.25	10.27	25.73	15.84	8.94	9.03
FuzzyEIG1	5.03	31.68	7.13	9.38	4.08	7.93	27.90	11.62	9.08	6.12	10.44	25.57	16.27	8.96	8.89
FuzzyEIG2	5.06	31.63	7.16	9.38	4.09	7.93	27.88	11.62	9.09	6.20	10.43	25.58	16.33	8.98	8.92



(a) Original image

(b) Noisy image $s = 30$ 

(c) Output from CWF



(d) Normalized error map CWF



(e) Output from EIG



(f) Normalized error map EIG



(g) Output from FuzzyEIG1



(h) Normalized error map FuzzyEIG1

Figure 4. Visual comparison of different denoising methods for the Grass image (200×200). Original image (a), noisy image (b), denoised images (CWF (c), EIG (e), FuzzyEIG1 (g)), corresponding normalized squared error maps (d,f,h).

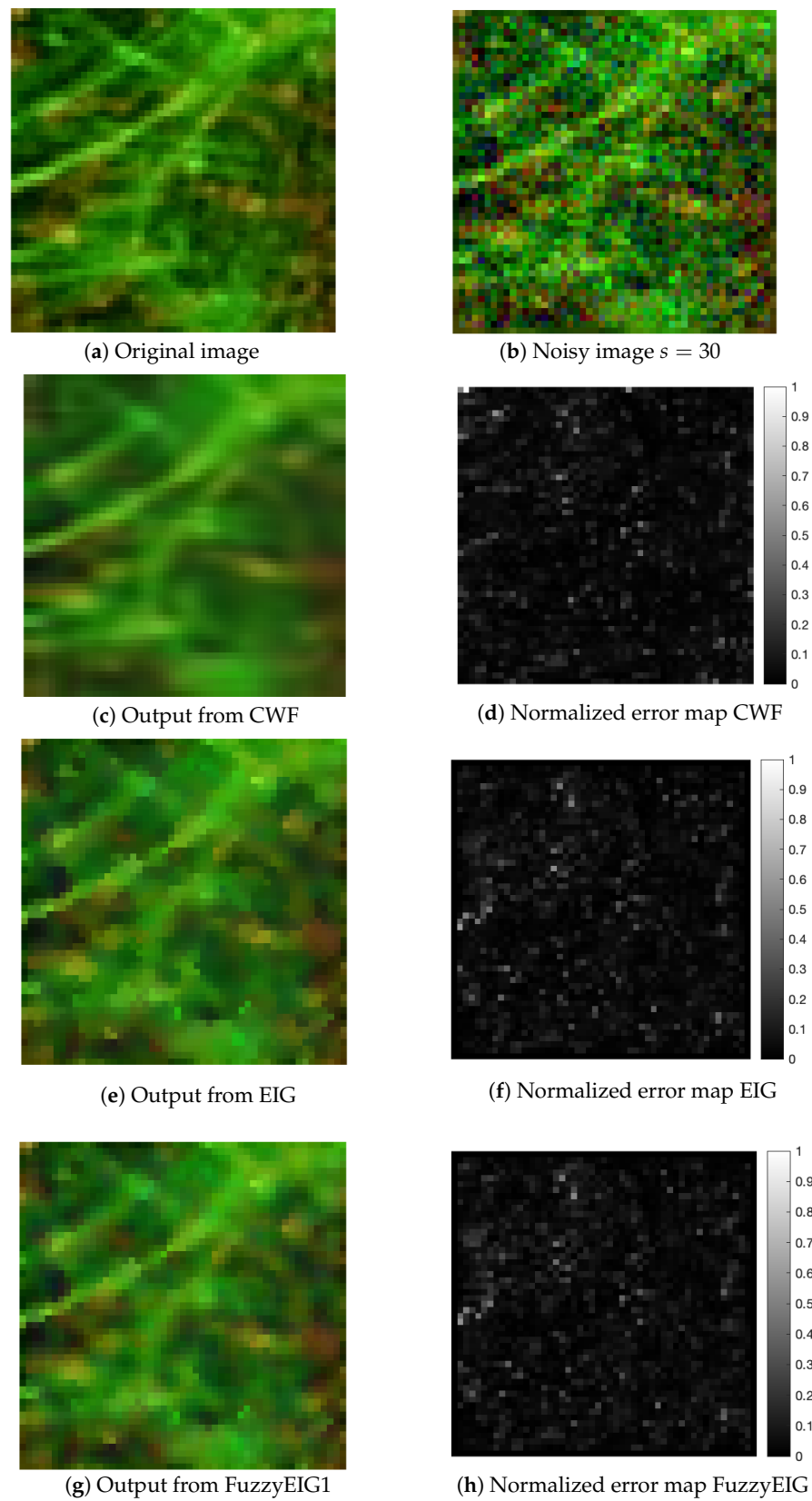


Figure 5. Visual comparison of different denoising methods for the Micro image (5×51). Original image (a), noisy image (b), denoised images (CWF (c), EIG (e), FuzzyEIG1 (g)), corresponding normalized squared error maps (d,f,h).

By closely examining the tables and figures, we observed that certain methods consistently outperform others across different noise levels and image sizes while others have a high variable performance. In particular, probably the highest variability in performance was shown by the DnCNN method, which we associated to training bias. Machine learning-based denoising methods are difficult to fully understand and are biased by the training dataset used to set their parameters. Consequently, it is hard to fully trust them in any scenario. On the other hand, the other methods in the experimental comparison are interpretable. Notably, the FuzzyEIG1 and FuzzyEIG2 methods demonstrated competitive performance, often ranking among the top two performers across multiple metrics. This is primarily due to their ability to enhance the smoothing capability of the EIG method in image areas with greater homogeneity.

However, it is important to clarify that they did not outperform CWF in such areas. CWF indeed achieved the best results in homogeneous regions due to its non-local block-matching approach, which is significantly more computationally expensive. This is understandable because its block-matching approach effectively finds more matches in these areas at an expense of higher computational load. Moreover, for example, in the Lenna images with noise level $s = 30$, we saw a 2-unit improvement in the PSNR of the proposed methods over the EIG method. However, there were slight variations in performance for detailed images, where FuzzyEIG1 and FuzzyEIG2 lost some performance with respect to EIG but still outperformed the non-local method CWF. For instance, in the Beach image with noise level $s = 10$, the proposed methods achieved around a 1 unit higher PSNR than CWF while they were only 0.3 below EIG. So, the benefit provided by the proposed methods is an increased performance in noise reduction with respect to EIG while keeping a good level of detail preservation although a bit lower than EIG.

Visually, in Figure 4, the CWF-processed “Grass” image exhibits some blurring in the top cloud areas compared to FuzzyEIG1 and EIG. While FuzzyEIG1 and EIG leave some noise to preserve details, CWF prioritizes smoothness, leading to this visible loss of sharpness. This trend is further evident in Figure 5. Here, the “Micro” image processed by CWF is noticeably blurred compared to FuzzyEIG1, which retains details closer to the EIG output.

A detailed analysis of the performance revealed that the limitation of the proposed methods arises during the defuzzification phase. The highest smoothing is achieved for coefficients $C_i = 0$, whereas EIG level detail preservation happens for $C_i = 1$. However, after defuzzification, C_i values never reach the extremes of 0 or 1. This is shown in Figure 6. Here, we can see, particularly for C_3 , that higher values are associated to image edges and detail regions to be preserved but they are far from the top value of 1. Conversely, neither 0 is reached for maximum smoothing in flat regions. This results in two key trade-offs: (1) Incomplete Smoothing in Homogeneous Regions: Since C cannot reach 0, maximum noise reduction in flat areas is limited. (2) Sub-optimal Detail Preservation: this limitation prevents C from reaching 1, although it only leads to small detail loss compared to EIG. Despite this limitation, the proposed methods excel at denoising flat regions compared to EIG filtering. This is because they effectively assign low enough C values in these areas, achieving strong noise reduction while maintaining details.

This improved performance is also seen in the histogram plots presented in Figures 7 and 8, where the proposed method performance lies in between CWF and EIG. These histograms plot, in the x axis, the normalized error, where the normalisation factor corresponds to the maximum error for any pixel in the denoising methods that are compared (CWF, EIG, FuzzyEIG1 and FuzzyEIG2). We only show CWF, EIG, and FuzzyEIG1 for interpretability purposes. We can see that CWF has a higher number of pixels with low error values when the image contains large homogeneous regions. This is the case, for instance, of the histograms for Grass (Figure 7). However, when there are more edges, the EIG and FuzzyEIG1 methods outperform CWF, as seen in the Beach histograms (Figure 8).

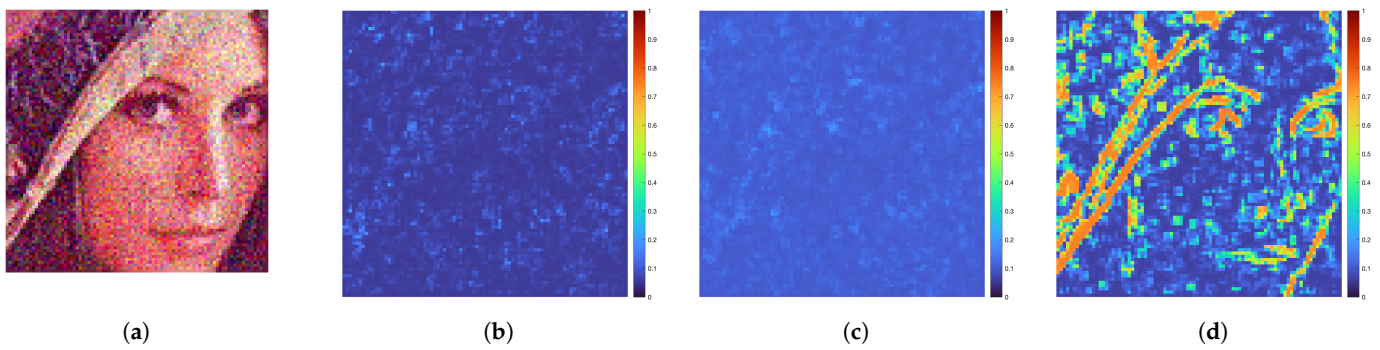


Figure 6. Maps of the smoothing coefficients for each colour channel in the Lenna image with noise level $s = 30$. (a) Noisy image $s = 30$; (b) C1; (c) C2; (d) C3.

In general, FuzzyEIG1 stands out for its ability to achieve low error values while balancing detail preservation and noise reduction. This performance lies in its approach to the trade-off between smoothing and detail retention. CWF prioritizes smoothness, which can be ideal in specific scenarios. However, this emphasis can lead to over-smoothing and loss of fine details in other cases/images types. FuzzyEIG methods are able to balance between smoothing and detail preservation. This makes them suitable for a wider range of image types, despite limitation in the defuzzification where the smoothing coefficients never reaching 0 or 1 (indicating full or minimal smoothing). While this suggests further potential for improvement, the current balance allows for FuzzyEIG methods to preserve details better than CWF while achieving comparable or better denoising in homogeneous regions. This balanced approach makes FuzzyEIG methods competitive across various metrics, even if they may not always surpass CWF in strictly homogeneous regions.

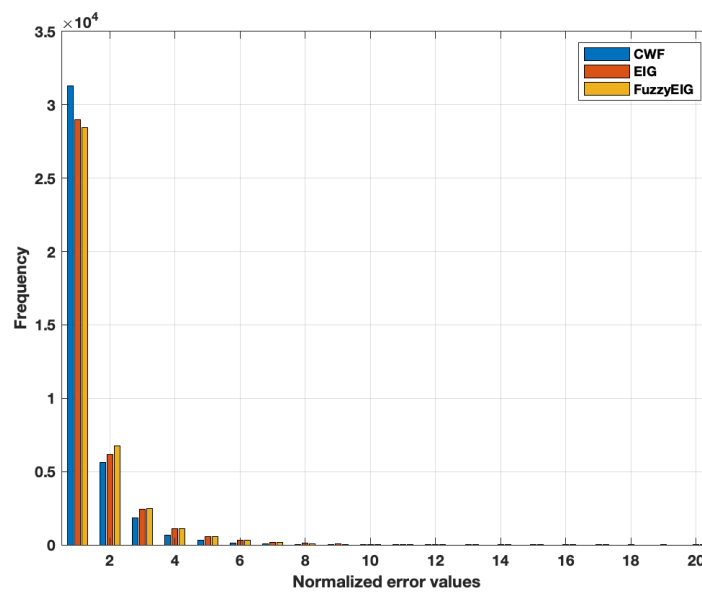


Figure 7. The combined histogram visualizes the distributions of normalized squared error values obtained from three denoising methods applied to an image with “Grass 200×200 ” noise level $s = 10$. The x -axis represents the range from 0 (perfect reconstruction) to 1 (maximum error), and the y -axis represents the frequency of each error value.

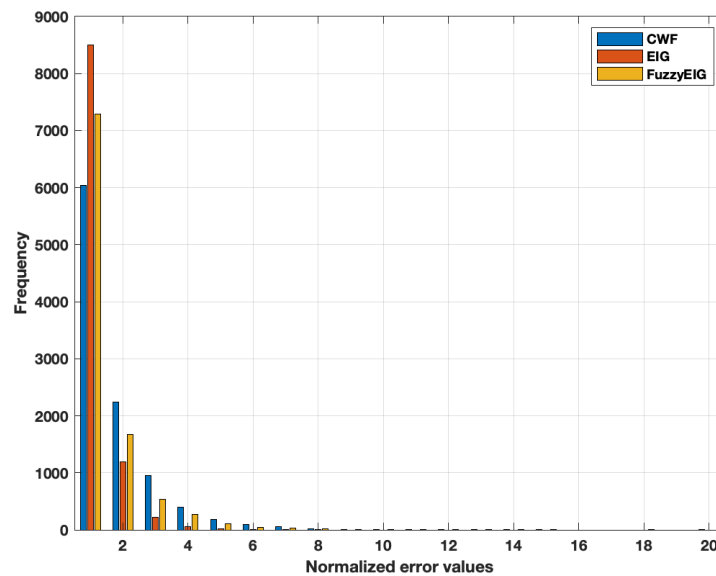


Figure 8. The combined histogram visualizes the distributions of normalized squared error values obtained from three denoising methods applied to an image with “Beach 100×100 ” noise level $s = 30$. The x -axis represents the range from 0 (perfect reconstruction) to 1 (maximum error), and the y -axis represents the frequency of each error value.

5. Conclusions and Future Work

Two novel image-denoising methods, FuzzyEIG1 and FuzzyEIG2, are presented in this paper, both of them based on an previous eigenvector-based strategy, EIG. While both methods achieve promising results in terms of noise reduction and detail preservation, their limitations are identified. FuzzyEIG1 and FuzzyEIG2 show competitive performance across various metrics and present a more scalable behaviour. Their main advantage is an increased performance in noise reduction with respect to EIG while keeping a good level of detail preservation although a bit lower than that of EIG.

However, the methods’ performance in both noise smoothing and detail preservation is not optimal. This happens because of the defuzzification process being unable to provide values close enough to one when higher detail preservation is needed or to zero when more smoothing is appropriate.

One promising avenue for future research is the integration of neural networks (NNs) for this defuzzification (last) stage. NNs can be trained to learn the optimal parameters for the defuzzification process, potentially improving both efficiency and accuracy. This hybrid approach would combine the strengths of both methods, leveraging the data-driven learning capabilities of NNs with the adaptability of FuzzyEIG methods. Ultimately, continuous exploration and refinement are crucial for developing robust and adaptable image-denoising techniques suitable for real-world applications.

Author Contributions: Conceptualization, K.A., S.M. and P.L.-C.; methodology, K.A, S.M. and P.L.-C.; software, K.A.; validation, K.A., S.M. and P.L.-C.; formal analysis, K.A., S.M. and P.L.-C.; investigation, K.A., S.M. and P.L.-C.; resources, S.M.; data curation, K.A., S.M. and P.L.-C.; writing—original draft preparation, K.A., S.M. and P.L.-C.; visualization, K.A., S.M. and P.L.-C.; supervision, S.M. and P.L.-C.; project administration, S.M.; funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Generalitat Valenciana under grant CIAICO/2022-051 IMaLeVICS and Spanish Ministry of Science under grant PID2022-140189OB-C21.

Data Availability Statement: The data and the code will be available upon a reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, W.; Lin, W. Additive white Gaussian noise level estimation in SVD domain for images. *IEEE Trans. Image Process.* **2013**, *22*, 872–883. [[CrossRef](#)]
2. Swenson, G.W.; Thompson, A.R. Radio Noise and Interference. In *Reference Data for Engineers*, 9th ed.; Middleton, W.M., Van Valkenburg, M.E., Eds.; Newnes: Oxford, UK, 2020; pp. 34-1–34-13. [[CrossRef](#)]
3. Price, J.; Goble, T. Signals and Noise. In *Telecommunications Engineer's Reference Book*; Mazda, F., Ed.; Butterworth-Heinemann: Oxford, UK, 1993; pp. 10-1–10-15. [[CrossRef](#)]
4. Plataniotis, K.N.; Venetsanopoulos, A.N. *Color Image Processing and Applications*; Springer: Berlin/Heidelberg, Germany, 2000.
5. Lukac, R.; Smolka, B.; Martin, K.; Plataniotis, K.N.; Venetsanopoulos, A.N. Vector Filtering for Color Imaging. *IEEE Signal Process. Mag.* **2005**, *22*, 74–86. [[CrossRef](#)]
6. Lukac, R.; Plataniotis, K.N. A Taxonomy of Color Image Filtering and Enhancement Solutions. In *Advances in Imaging and Electron Physics*; Hawkes, P.W., Ed.; Elsevier Academic Press: Amsterdam, The Netherlands, 2006; Volume 140, pp. 187–264.
7. Buades, A.; Coll, B.; Morel, J.M. Nonlocal Image and Movie Denoising. *Int. J. Comput. Vision* **2008**, *76*, 123–139. [[CrossRef](#)]
8. Tomasi, C.; Manduchi, R. Bilateral filter for gray and color images. In Proceedings of the IEEE International Conference Computer Vision, Bombay, India, 4–7 January 1998; pp. 839–846.
9. Elad, M. On the origin of bilateral filter and ways to improve it. *IEEE Trans. Image Process.* **2002**, *11*, 1141–1151. [[CrossRef](#)]
10. Kao, W.C.; Chen, Y.J. Multistage Bilateral Noise Filtering and Edge Detection for Color Image Enhancement. *IEEE Trans. Consum. Electron.* **2005**, *51*, 1346–1351.
11. Garnett, R.; Huegerich, T.; Chui, C.; He, W. A Universal Noise Removal Algorithm with an Impulse Detector. *IEEE Trans. Image Process.* **2005**, *14*, 1747–1754. [[CrossRef](#)] [[PubMed](#)]
12. Morillas, S.; Gregori, V.; Sapena, A.F. Fuzzy Bilateral Filtering for Color Images. *Lect. Notes Comput. Sci.* **2006**, *4141*, 138–145.
13. Mehra, M.; Mehra, V.K.; Ahmad, M. *Wavelets Theory and Its Applications*; Springer: Singapore, 2018.
14. Dabov, K.; Foi, A.; Katkovnik, V.; Egiazarian, K. Image Denoising by Sparse 3D Transform-Domain Collaborative Filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095. [[CrossRef](#)] [[PubMed](#)]
15. Dabov, K.; Foi, A.; Katkovnik, V.; Egiazarian, K. Color Image Denoising via Sparse 3D Collaborative Filtering with Grouping Constraint in Luminance-Chrominance Space. In Proceedings of the IEEE International Conference on Image Processing (ICIP 2007), San Antonio, TX, USA, 16–19 September 2007; pp. 313–316.
16. Hao, B.B.; Li, M.; Feng, X.C. Wavelet Iterative Regularization for Image Restoration with Varying Scale Parameter. *Signal Process. Image Commun.* **2008**, *23*, 433–441. [[CrossRef](#)]
17. Pérez-Benito, C.; Jordán, C.; Conejero, J.A.; Morillas, S. Graphs-based methods for simultaneous smoothing and sharpening. *MethodsX* **2020**, *7*, 100819. [[CrossRef](#)] [[PubMed](#)]
18. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [[CrossRef](#)] [[PubMed](#)]
19. Oja, E. Principal components, minor components, and linear neural networks. *Neural Netw.* **1992**, *5*, 927–935. [[CrossRef](#)]
20. Takahashi, T.; Kurita, T. Robust de-noising by kernel PCA. In Proceedings of the ICANN2002, Lecture Notes in Computer Science, Madrid, Spain, 28–30 August 2002; Volume 2145, pp. 739–744.
21. Park, H.; Moon, Y.S. Automatic Denoising of 2D Color Face Images using Recursive PCA Reconstruction. In Proceedings of the Advanced Concepts for Intelligent Vision Systems ACIVS06, Lecture Notes in Computer Science, Antwerp, Belgium, 18–21 September 2006; Volume 4179, pp. 799–809.
22. Latorre-Carmona, P.; Miñana, J.-J.; Morillas, S. Colour Image Denoising by Eigenvector Analysis of Neighbourhood Colour Samples. *Signal Image Video Process.* **2020**, *14*, 483–490. [[CrossRef](#)]
23. Zadeh, L.A. Fuzzy Sets. *Inf. Control* **1965**, *8*, 338–353. [[CrossRef](#)]
24. Sabri, N.; Aljunid, S.A.; Salim, M.S.; Badlishah, R.B.; Kamaruddin, R.; Malek, M.A. Fuzzy Inference System: Short Review and Design. *Int. Rev. Autom. Control* **2013**, *6*, 441–449.
25. Schulte, S.; De Witte, V.; Kerre, E.E. A Fuzzy Noise Reduction Method for Color Images. *IEEE Trans. Image Process.* **2007**, *16*, 1425–1436. [[CrossRef](#)] [[PubMed](#)]
26. Schulte, S.; Witte, V.D.; Nachtegaal, M.; M'elange, T.; Kerre, E.E. A New Fuzzy Additive Noise Reduction Method. In Proceedings of the International Conference Image Analysis and Recognition, Montreal, QC, Canada, 22–24 August 2007; pp. 12–23.
27. Van De Ville, D.; Nachtegaal, M.; Van der Weken, D.; Kerre, E.E.; Philips, W.; Lemahieu, I. Noise Reduction by Fuzzy Image Filtering. *IEEE Trans. Fuzzy Syst.* **2003**, *11*, 429–436. [[CrossRef](#)]
28. Almutairi, K.; Morillas, S.; Latorre-Carmona, P. *Fuzzy Inference System in a Local Eigenvector Based Color Image Smoothing Framework*; Scitepress: Prague, Czech Republic, 2023.
29. Dillon, W.R.; Goldstein, M. *Multivariate Analysis: Methods and Applications*; Wiley: Hoboken, NJ, USA, 1984.
30. Jackson, J.E. *A User's Guide to Principal Components*; Wiley: Hoboken, NJ, USA, 2003.
31. Mendel, J.M. Fuzzy Logic Systems for Engineering: A Tutorial. *Proc. IEEE* **1995**, *83*, 345–377. [[CrossRef](#)]
32. Casillas, J.; Cord'ón, O.; Herrera, F.; Magdalena, L. Interpretability Improvements to Find the Balance Interpretability-Accuracy in Fuzzy Modeling: An Overview. In *Interpretability Issues in Fuzzy Modeling*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 3–22.
33. Moallem, P.; Mousavi, B.; Naghibzadeh, S.S. Fuzzy Inference System Optimized by Genetic Algorithm for Robust Face and Pose Detection. *Int. J. Artif. Intell.* **2015**, *13*, 73–88.

34. Abualigah, L.M.Q.; Hanandeh, E.S. Applying genetic algorithms to information retrieval using vector space model. *Int. J. Comp. Sci. Eng. Appl. (IJCSEA)* **2015**, *5*, 19–28. [[CrossRef](#)]
35. Grecova, S.; Morillas, S. Perceptual similarity between color images using fuzzy metrics. *J. Visual Commun. Image Represent.* **2016**, *34*, 230–235. [[CrossRef](#)]
36. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
37. Fairchild, M.D.; Johnson, G.M. ICAM Framework for Image Appearance, Differences, and Quality. *J. Electron. Imaging* **2004**, *13*, 126–138. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.