applied sciences

MDPI

Article

Image Noise Reduction by Means of Bootstrapping-Based Fuzzy Numbers

Reza Ghasemi [1,*,†], Samuel Morillas [2,†], Ahmad Nezakati [3,†] and Mohammadreza Rabiei [3,†]

1 Department of Statistics, Payame Noor University (PNU), Tehran 193953697, Iran
2 Instituto de Matemática Pura y Aplicada, Universitat Politécnica de Valéncia, 46022 Valencia, Spain
3 Department of Statistics, Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood 3619995161, Iran
* Correspondence: r.ghasemi62@pnu.ac.ir
† These authors contributed equally to this work.

Abstract: Removing or reducing noise in color images is one of the most important functions of image processing, which is used in many sciences. In many cases, nonlinear methods significantly reduce the noise in the image and are widely used today. One of these methods is the use of fuzzy logic. In this paper, we want to introduce a fuzzy filter by using the fuzzy metric for fuzzy sets. For this purpose, we define fuzzy color pixels by using the mean of neighborhoods. Due to the noise in the image, we use the bootstrap resampling method to reduce the effect of outliers. The concept of the strong law of large numbers for the bootstrap mean in fuzzy metric space helps us to use the resampling method.

Keywords: fuzzy metric space; image processing; fuzzy filter; noise reduction; bootstrap resampling

check for updates

Citation: Ghasemi, R.; Morillas, S.; Nezakati, A; Rabiei, M. Image Noise Reduction by Means of Bootstrapping-Based Fuzzy Numbers. *Appl. Sci.* **2022**, *12*, 9445. https://doi.org/10.3390/app12199445

Academic Editor: Byung-Gyu Kim

Received: 1 September 2022
Accepted: 18 September 2022
Published: 21 September 2022

1. Introduction

An essential task in image processing is the use of image noise reduction filters, which is necessary for any computer vision system that uses images for automated analysis. In fact, noise can cause problems in image processing tasks (such as edge detection, pattern detection, etc.). Therefore, we should try to reduce the noise.

In recent years, the interest in using color images in various applications such as medical diagnostics, GIS, etc., has increased significantly. Therefore, the use of filtering and image noise reduction has been considered as a new field for applied research. It is widely observed that there is a correlation between image channels. Therefore, color images should be processed taking into account these correlations.

The vector approach is one of the most famous methods used in this field. Among the methods that are used based on a strong statistical theory, we can mention the vector median filter (VMF) and vector directional filter (VDF), which show good performance in image filtering. Despite the good results in general, these techniques do not perform properly on the edges and some details of the image and cause blurring of the image in these areas. To solve this problem, recently, adaptive vector processing solutions have been used.

The concept of the fuzzy metric space was first defined by Kromasil and Michálek [1], which gave a generalization of Menger's probabilistic metric space [2]. This definition was modified by George and Viramani [3]. This concept has been used in applied sciences such as image noise reduction [4,5] and the perceptual color difference in [6,7]. Morillas et al. used a fuzzy metric as a distance measure to reduce image noise. This fuzzy metric uses two measures of similarity between color vectors and the spatial proximity of pixels in the image to compare two pixels simultaneously.

In recent years, much research has been performed in this field. Gregori et al. [8] used two different distance measures based on the fuzzy metric definition to filter image noise. Furthermore, they [9] reduced the effect of noise in the images by using fuzzy averaging.

Using fuzzy thresholding, Bandyopadhyay et al. [10] detected the noisy region of the image. The fuzzy T-metric is another algorithm that was introduced by Ralevic et al. [11] and showed good results in reducing image noise.

In this paper, we want to generalize Morillas's method. Our most important tool to achieve this goal is to convert image pixel values to fuzzy numbers. To do this, we used the mean of the neighborhoods per pixel (as will be described in Section 6). We used bootstrap resampling to calculate the mean to reduce the scatter created by the outlier data (noise entered in the image). The bootstrapping resampling method introduced by Efron is used for a variety of estimation problems. Since the strong law of large numbers (SLLN) is particularly important in the bootstrap method, many researchers have studied and researched this field (see Athreya [12] and Athreya et al. [13]). Ghasemi et al. [14] provided a fuzzy metric space for the fuzzy set, which is a generalization of the fuzzy metric space presented by George and Veeramani [3]. Furthermore, they generalized the SLLN into the fuzzy metric space of the bootstrap mean. We used the SLLN concept in the fuzzy metric space for the bootstrap mean introduced by Ghasemi et al. [14], in image noise reduction, which is a generalization of Morillas's approach [15].

In the following, some preliminaries are presented. In the next step, we introduce the space of the fuzzy metric. In Section 4, we study the bootstrap mean in the fuzzy metric space. The filtering images by the fuzzy metric is performed in Section 5. In the next step, the proposed filter is introduced and examples are provided to compare the performance accuracy with other methods.

2. Preliminaries

In this section, we introduce the concepts that we use in the following sections.

One of the definitions that is used in the fuzzy metric space is triangular norms (t-norms for short), which was first defined by Schweizer and Sklar [16].

Definition 1 ([17])**.** *A t-norm is a binary operation* $*\ : [0,\ 1] \times [0,\ 1] \to [0,\ 1]$*, such that, for all* $a,\ b,\ c,\ d\ \in [0,\ 1]$*, the following four conditions must be met:*

- $a * 1 = a$*;*
- $a * b\ \leq c * d$ *whenever* $a \leq c$ *and* $b \leq d$*;*
- $a * b = b * a$*;*
- $a * (b * c) = (a * b) * c$*.*

If the binary operation $*$ *is a continuous function on* $[0,\ 1] \times [0,\ 1]$*, then the t-norm is said to be continuous.*

A fuzzy set of $\mathbb{R}^p, p \geq 1$ is a function of $u : \mathbb{R}^p \to [0, 1]$. Here, supp u is the support of u, where $\{x \in \mathbb{R}^p \mid u(x) > 0\}$. Furthermore, the α-level set for each fuzzy set u is defined by

$$L_\alpha u = \{x \in \mathbb{R}^p \mid u(x) \geq \alpha\}, 0 < \alpha \leq 1.$$

Suppose $\mathcal{F}(\mathbb{R}^p)$ is the collection of those fuzzy sets on $\mathbb{R}^p$. Two fuzzy sets u and v are equal, written as $u = v$, if and only if $u(x) = v(x)$ for all x in $\mathbb{R}^p$ [18].

We denote $d(u, v)$ as a metric in $\mathcal{F}(\mathbb{R}^p)$, i.e., for $u, v, z \in \mathcal{F}(\mathbb{R}^p)$:

- $d(u, v) \geq 0$;
- $d(u, v) = 0$ if and only if $u = v$;
- $d(u, z) \leq d(u, v) + d(v, z)$.

Furthermore, the norm of u is defined $\|u\| = d(u, I_{\{0\}})$, which $I_{\{0\}}$ is the indicator function (the fuzzy set taking a value of one at 0 and zero for all $x \neq 0$).

Let $\mathcal{F}_c(\mathbb{R}^p)$ be the collection of those fuzzy sets $u : \mathbb{R}^p \to [0, 1]$ with the following properties:

- $L_\alpha u$ is compact for all $0 < \alpha \leq 1$;
- supp u is compact;
- $\{x \in \mathbb{R}^p | u(x) = 1\} \neq \emptyset$.

3. Introduction to Fuzzy Metric Space

As we know, one of the important concepts in data mining, image processing, and multivariate data analysis is the distance measurement criteria. Some distance measures of the exact number are well established in the literature. Distances are calculated in an imprecise framework, which creates logical problems due to ambiguity. In such cases, crisp numbers are converted to fuzzy numbers.

Various definitions of the fuzzy metric space have been proposed by various authors [19–21]. George and Viramani in [3] modified the concept of the fuzzy metric space introduced by Kramosil and Michalek [1] and defined a Hausdorff topology on this fuzzy metric space. In this definition, which we refer to in the following, $M(x,y,t)$ can be thought of as the degree of nearness between x and y with respect to t.

Assuming that $\mathscr{X}$ is an arbitrary non-empty set, M is a fuzzy metric (fuzzy set) on $\mathscr{X} \times \mathscr{X} \times (0,\infty)$ and $*$ is a continuous t-norm. In this case, the three-tuple $(\mathscr{X}, M, *)$ is said to be a fuzzy metric space [3]. Here, if $M(x,y,t)$ is symmetric with respect to x and y, continuous on t, and satisfies the following conditions for all:

- $M(x,y,t) = 1$ if and only if $x = y$;
- $M(x,z,t+s) \geq M(x,y,t).M(y,z,s)$.

$M(x,y,t)$ is called a fuzzy metric and indicates the degree of closeness or similarity of x and y according to t.

Example 1 ([3]). *Suppose that $(\mathscr{X}, d)$ is a metric space, and we define $a * b = \min\{a,b\}$ or $a * b = ab$ and $\forall x, y \in \mathscr{X}$,*

$$M(x,y,t) = \frac{kt^n}{kt^n + md(x,y)}, \quad k, m \in \mathbb{R}^+, n \geq 1.$$

Then, $(\mathscr{X}, M, *)$ is a fuzzy metric space. Let us say now that $M(x,y,t)$ is the standard fuzzy metric induced by the d metric, whenever $k = n = m = 1$:

$$M(x,y,t) = \frac{t}{t + d(x,y)}, \tag{1}$$

For the first time, the fuzzy normed space was defined by Saadati and Vaezpour [22]. Assuming that $\mathcal{X}$ is a vector space, N is a fuzzy set (fuzzy norm) on $\mathcal{X} \times (0,\infty)$, and $*$ is a continuous t-norm, then the three-tuple $(\mathcal{X}, N, *)$ is called a fuzzy normed space.

Detecting and reducing the effect of noise in an image can be considered one of the important applications of the fuzzy metric. Morillas et al. [15] used this metric to detect and reduce noise in images and showed that this method has a better effect than other methods. Actually, Morillas used the standard fuzzy metric as follows:

$$R(F_i, F_j, t) = \frac{t}{t + \|F_i - F_j\|},$$

where $F_i = (F_i^1, F_i^2, F_i^3)$ are red (R), green (G), and blue (B) for each pixel of the image.

Now, if we have a fuzzy view of the colors and consider the value of each pixel as a fuzzy number, the above method cannot be effective. In fact, the method presented by Morrilas et al. [15] is suitable for when the value of each pixel is a crisp number, and if the value of each pixel can be defined as a fuzzy number, this method cannot be used anymore. In this case, the fuzzy metric for fuzzy sets should be used.

Ghasemi et al. in [14] presented a definition of the fuzzy metric space for fuzzy sets, which are a generalization of the fuzzy metric space [3]. The definition is as follows:

Definition 2 ([14]). *Let $\mathscr{X}$ be an arbitrary non-empty set, $\mathcal{F}(\mathscr{X})$ be the collection of those fuzzy sets on $\mathscr{X}$, and $*$ be a continuous t-norm. The three-tuple $(\mathcal{F}(\mathscr{X}), M', *)$ is said to be a fuzzy metric* *space for fuzzy sets if M' is a fuzzy set (fuzzy metric for fuzzy sets) on $\mathcal{F}(\mathscr{X}) \times \mathcal{F}(\mathscr{X}) \times (0, \infty)$ satisfying the following conditions for all $u,\ v,\ z\ \in \mathcal{F}(\mathscr{X})$ and $t, s > 0$:*

- $M'(u, v, 0) > 0$;
- $M'(u, v, t) = 1$ for all $t\ >\ 0$ *if and only if* $u\ =\ v$;
- $M'(u, v, t) = M'(v, u, t)$;
- $M'(u, v, t) * M'(v, z, s) \leq M'(u, z, t + s)$;
- $M'(u, v, .) : (0, \infty) \to [0,\ 1]$ is continuous.

Example 2. *Suppose that $(\mathcal{F}(\mathscr{X}), M', *)$ is a fuzzy metric space for fuzzy sets, and we define $a * b = \min\{a, b\}$ or $a * b = ab$ and $\forall u, v \in \mathcal{F}(\mathscr{X})$:*

$$M'(u, v, t) = \frac{kt^n}{kt^n + md(u, v)},\ \ k, m \in \mathbb{R}^+, n \geq 1.$$

It is easy to show that $(\mathcal{F}(\mathscr{X}), M', *)$ is a fuzzy metric space for fuzzy sets. Let us say now that $M'(u, v, t)$ is the standard fuzzy metric induced by the d metric, whenever $k = n = m = 1$:

$$M'(u, v, t) = \frac{t}{t + d(u, v)}, \tag{2}$$

4. Bootstrap Mean in Fuzzy Metric Space

Efron introduced the bootstrap resampling method in [23], which is used to estimate the statistical distribution based on independent observations. The bootstrap is a method that, regardless of many hypotheses, brings the sample conditions closer to the community conditions by creating many samples and obtains a more reliable estimate by considering all the cases of sample formation. The bootstrap will take a sample by substituting the original sample so that each sample taken by this method is independent, but has an equal distribution. In other words, the samples taken by the bootstrap method have an equal population distribution; however, each sample shall be independent of other samples.

Athreya in [12] presented the strong law for the bootstrap, then Athreya et al. [13] presented the law of large numbers for bootstrapped U-statistics. In addition, Csörgo [24] introduced the weak law of large numbers (WLLN) and the strong law of large numbers (SLLN) for bootstrap sample means under minimal moment conditions. In this section, we present an algorithm to remove or reduce noise in color images using the bootstrap in the fuzzy metric space.

Let $\{X_k | \, k \geq 1\}$ be an infinite sequence of independent and identically distributed fuzzy random variables defined on a probability space $(\Omega, \mathcal{A}, P)$ and X_1 be integrable. For each $n = 1, 2, \ldots,$ let $Y_{n,1}, Y_{n,2}, \ldots, Y_{n,m(n)}$ be the ordinary Efron bootstrap sample of size $m(n)$ where $\{m(n)\}$ is a sequence of positive integers. The variables $Y_{n,1}, Y_{n,2}, \ldots, Y_{n,m(n)}$ result from sampling m times the sequence $\{X_1, X_2, \ldots, X_n\}$ with replacement such that, at each stage, any one element has probability $\dfrac{1}{n}$ to be picked [24].

Suppose that $\bar{X}^*_n$ is the bootstrap sample mean where

$$\bar{X}^*_n = \frac{1}{m(n)} \sum_{i=1}^{m(n)} Y_{n,i}.$$

Klement et al. [25] provided limit theorems for fuzzy random variables. In fact, they showed

$$d_1\Big(\frac{1}{n} \sum_{i=1}^{n} X_i, E(co\ X_1)\Big) \to 0, \text{ a.e.}$$

where $E(co\,X_1)$ is the expectation of a convex fuzzy random variable. Now, if we apply this theorem for the bootstrap mean, then we have

$$d_1(\frac{1}{m(n)}\sum_{i=1}^{m(n)} Y_{n,i}, E(co\,X_1)) \to 0, \text{ a.e.}$$

Ghasemi et al. [14] showed with an example the use of the SLLN for the bootstrap mean in the fuzzy metric space. They believed that, sometimes, the expert's opinion may be important in determining the magnitude or smallness of the distance. In this case,

$$M'_{d_1}(\frac{1}{m(n)}\sum_{i=1}^{m(n)} Y_{n,i}, E(co\,X_1), t) \to 1, \text{ a.e.}$$

Due to this feature, in the next section, we provide an algorithm for reducing image noise using the bootstrap mean for pixels with a fuzzy value.

5. Filtering Images

Since the intensity of the darkness or lightness of the color can be fuzzy in nature, in this section, we want to have a fuzzy look at the color of the pixels. To use this method, which will be described below, we converted the crisp numbers for each pixel of the image to a triangular fuzzy number. On the other hand, to calculate the distance and similarity of these fuzzy pixels using a fuzzy metric, we need a metric that calculates the distance between two fuzzy numbers (as introduced in Section 3). In the following, we explain the required definitions and describe the approach.

The fuzzy metric filter developed by Morillas et al. [15] uses two operators to reduce image noise, which is denoted by *CFM*. It measures the similarity of the pixels with the help of an operator and uses the second operator to measure the spatial closeness of the pixels in detecting the replacement value of the noisy pixel. The algorithm introduced in this section has the same functionality as the previous method, except that our innovation is to use the fuzzy metric for fuzzy sets and bootstrapping. Hence, the similarity operator is the fuzzy metric for fuzzy sets, but the second operator, which measures the distance or proximity of pixels, is unchanged and similar to the previous filter.

5.1. Fuzzy Similarity Value

In this section using a fuzzy metric, we measure the amount of fuzzy similarity between the color vectors where the color in each pixel is expressed as a fuzzy number. This function is defined as follows.

Suppose $\tilde{u}_k = (\tilde{u}_k^1, \tilde{u}_k^2, \tilde{u}_k^3)$ represents the color vector of the image pixels at position k comprising its R, G, and B components, where $\tilde{u}_k^i$ is a fuzzy number for each pixel and represents red, green, or blue with a fuzzy number. Furthermore, c is a positive real parameter used to control the spread of the function. This function, denoted by R, is as follows:

$$R(\tilde{u}_i, \tilde{u}_j) = \frac{c}{c + \|\tilde{u}_i - \tilde{u}_j\|}.$$

5.2. Fuzzy Spatial Closeness

For the case of fuzzy spatial closeness between pixels, we did exactly the same as Morillas et al. [15]. Let us consider the pixels in an $n \times n$ filter window W represented in Cartesian coordinates; thus, denote by $i = i_1, i_2 \in Y^2$ the position of a pixel $\tilde{u}_i$ in W

where $Y = \{0, 1, 2, \ldots, n-1\}$. We consider the standard fuzzy metric S deduced from the L_∞ metric [15] given by

$$S(i, j, t) = \frac{t}{t + \|i - j\|_\infty},$$

where $i, j \in Y^2$, $t > 0$, and $\|i - j\|_\infty = \max\{|i_1 - j_1|, |i_2 - j_2|\}$. It can be easily seen that pixels farther from the window center are less close overall to the rest of the pixels in the window. It has been experimentally observed that the L_∞ metric has a better result than the Euclidean metric [15]. Thus, $S(i, j, t)$ expresses the amount of fuzzy spatial closeness between color pixels $\tilde{u}_1$ and $\tilde{u}_2$ with respect to the value of t. Here also, the parameter t is used to adjust the importance given to the spatial closeness criterion. Now, considering that each pixel is displayed as a three-component color vector RGB and location, so to achieve our purpose, we used a combination fuzzy metric as follows:

$$BCFM(\tilde{u}_i, \tilde{u}_j, t) = R(\tilde{u}_i, \tilde{u}_j).S(i, j, t) = \frac{c}{c + \|\tilde{u}_i - \tilde{u}_j\|}.\frac{t}{t + \|i - j\|_\infty}. \tag{3}$$

If we identify each pixel $\tilde{u}_i$ with $\tilde{u}_i = (\tilde{u}_i^1, \tilde{u}_i^2, \tilde{u}_i^3, i_1, i_2)$, then it can be proven that $BCFM$ is a fuzzy metric for the fuzzy set on $X^3 \times Y^2$. This fuzzy metric expresses the degree of similarity of color and the closeness of pixels.

For example, suppose we have:

$$\|\tilde{u}_i - \tilde{u}_j\| = d_2(\tilde{u}, \tilde{v}) = (\int_0^1 f(\alpha) d^2(L_\alpha \tilde{u}, L_\alpha \tilde{v}) d\alpha)^{\frac{1}{2}},$$

and $\tilde{u}, \tilde{v}$ are two triangular fuzzy numbers as $\tilde{u} = (a, s_a)_T, \tilde{v} = (b, s_b)_T$, then

$$d_2(\tilde{u}, \tilde{v}) = (\int_0^1 \alpha d^2(L_\alpha \tilde{u}, L_\alpha \tilde{v}) d\alpha)^{\frac{1}{2}} = [(a-b)^2 + \frac{1}{6}(s_a - s_b)^2]^{\frac{1}{2}}. \tag{4}$$

so $BCFM$ will be equal to

$$BCFM(\tilde{u}_i, \tilde{u}_j, t) = \frac{c}{c + [(a-b)^2 + \frac{1}{6}(s_a - s_b)^2]^{\frac{1}{2}}}.\frac{t}{t + \|i - j\|_\infty}. \tag{5}$$

In the following example, we assume that we have a fuzzy image matrix (which is described in Section 6).

Example 3. *Suppose the following matrix (Figure 1) is equal to the pixel values of a part of an image on a gray scale. For each pixel, the amount of gray is expressed as a symmetric triangular fuzzy number.*

(44,0.5)	(45,0.3)	(159,0.7)
(161,0.8)	(165,0.4)	(159,0.6)
(157,0.7)	(162,0.1)	(154,0.4)

Figure 1. The 3×3 neighborhood for 165.

We want to use BCFM to estimate the value of 165 in the above matrix. Now, using Equation (5), we calculated the value of each pixel in the neighborhood of 165 and considered the pixel that has the largest BCFM as the best alternative to the value of 165.

Note that in Figure 2 the maximum value in matrix BCFM is 4.0756*, which corresponds to pixel number* $(3, 2)$*. Therefore, the value of this pixel,* $(162, 0.1)$*, can be chosen as the best alternative to the* 3×3 *neighborhood matrix center.*

1.6330	1.6375	3.7018
3.8509	3.8400	4.0105
3.7260	4.0756	3.7800

Figure 2. *BCFM* matrix.

To express the accuracy of this method, we assumed that a noise is added to the image so that the neighborhood matrix is as follows (Figure 3):

(44,0.5)	(45,0.3)	(159,0.7)
(161,0.8)	(0,0.4)	(159,0.6)
(157,0.7)	(162,0.1)	(154,0.4)

Figure 3. Matrix neighborhood with noise.

The maximum value in matrix BCFM (Figure 4) is 3.5294*, which corresponds to the pixel number* $(2,3)$*. Therefore, the value of this pixel,* $(159.0.6)$*, is the best estimate for the center of the* 3×3 *neighborhood matrix.*

1.6699	1.6726	3.2207
3.3116	1.1058	3.5294
3.3083	3.5130	3.0433

Figure 4. *BCFM* matrix.

6. Fuzzy Amount of Pixel's Color

One of the important points in using the algorithm presented in the previous section is to assign or calculate a fuzzy number for the color of each pixel. Since the color value of each pixel is expressed as a crisp number, different approaches to their fuzzy expression can be suggested, which can provide a field of study for future research. For a better explanation of the subject, it should be noted that just as the colors of a rainbow cannot be easily separated and the colors are slowly converted, the difference between the color of a pixel and the color of its neighborsis difficult. Fuzzy sets can easily do this, while in existing algorithms, a non-negative integer is considered for the color of each pixel.

Our approach in this paper is to use the neighborhoods of a pixel to express the fuzzy color of the desired pixel. In other words, we assumed that the color of each pixel is an asymmetric triangular fuzzy number, the color of each pixel being determined by the degree of membership 1 and its left or right width by the mean of k means generated by the bootstrap resampling method where k is the number of sampling repetitions of the desired pixel neighborhoods. We used the bootstrap resampling method to reduce the effect of the outliers in the selected window. One of the main reasons for using the bootstrap is that this sampling is based on the sample we have. Often, this sample is the only source we have for research, and this adds to the importance of the bootstrap method. Since the bootstrap reliability improves with the number of bootstrap repetitions, we can obtain a more reliable estimate of the mean value of the central pixel neighborhoods by increasing the number of samples.

The bootstrap method provides an estimate of a quantity of a population. This is achieved by collecting small samples repeatedly, calculating the statistics, and averaging the calculated statistics. This procedure can be summarized in the following terms:

- Select a number of bootstrap samples to perform;
- Select a sample size;
- For each bootstrap sample:
  a Draw a sample with replacement with the chosen size;
  b Calculate the statistic on the sample;
- Calculate the mean of the calculated sample statistics.

For example, if the image is grayscale, we have only one color number for each pixel. Consider a 3×3 window of its neighborhoods of each pixel (except in the pixels of the image margin), and calculate the mean of the bootstrap means.

The obtained number is a triangular fuzzy number with a right or left width. For example, suppose the pixel color is 50 and the mean of the bootstrap means is 60. Consider the color of a pixel as a fuzzy number $(50, 0, 10)$, with the left width of zero (because the mean of the bootstrap means is larger than the desired pixel) and the right width of 10 (which is equal to the difference between the pixel value and the mean of the bootstrap means). In the next example (Morillas et al. [15]), we give a more detailed description of the algorithm.

Example 4. *Suppose the following matrix is equal to the values of the pixels in the image on a grayscale (Figure 5).*

[a]

[b]

40	45	47	50	47	45	45
40	50	49	60	48	45	50
45	45	44	45	159	50	50
45	51	161	165	159	155	45
45	152	157	162	154	155	150
159	154	156	153	151	150	155
155	155	153	156	151	150	155

Figure 5. (**a**) image on a grayscale. (**b**) Matrix of image.

Now, if there is noise in the image, we can eliminate the noise using the fuzzy approach. Suppose we replace the middle pixel of the image (165) *with the value of* 0*, what is shown in Figure 6.*

40	45	47	50	47	45	45
40	50	49	60	48	45	50
45	45	44	45	159	50	50
45	51	161	0	159	155	45
45	152	157	162	154	155	150
159	154	156	153	151	150	155
155	155	153	156	151	150	155

Figure 6. Pixel matrix with noise.

To convert the color matrix of pixels, it is necessary to select each pixel and its nearest neighbors (a 2×2*, or* 2×3*, or* 3×2 *matrix for the side pixels and a* 3×3 *matrix for the other pixels, similar to Figure 7). Then, using bootstrap sampling, we obtain the mean of the neighborhoods. Now, if the mean of the neighborhoods is smaller than the center pixel, the pixel becomes a triangular fuzzy number the left width of which is equal to the difference of the pixel with the mean and the right width of which is equal to* 0*, and if the mean is greater, it is the product of the fuzzy number with the right width. After calculating the left or right width of each pixel, the results can be seen in Figures 8 and 9.*

45	50	49	50	48	45	45
50	50	49	60	48	45	45
45	45	44	45	159	50	50
45	51	161	0	159	155	155
45	152	157	162	154	155	155
154	154	156	153	151	150	150
155	155	156	156	151	150	150

Figure 7. Neighbors of each pixel.

The left width is as follows.

0	0	0	0	0	0	0
0	4	0	0	1	0	3
1	0	0	0	92	0	0
0	0	32	0	2	35	0
0	56	25	23	0	14	68
4	0	2	0	0	0	0
0	0	0	2	0	0	4

Figure 8. Left width of the fuzzy matrix.

9	1	6	0	1	2	1
3	0	0	11	0	15	0
0	41	25	42	0	23	16
19	56	0	91	0	0	90
55	0	0	0	2	0	0
0	1	0	0	0	3	0
1	1	1	0	2	1	0

Figure 9. Right width of the fuzzy matrix.

Now, we eliminated the noise by performing the algorithm described in this section. We know that

$$\|\tilde{u}_i - \tilde{u}_j\| = d_2(\tilde{u}, \tilde{v}) = \left(\int_0^1 f(\alpha) d^2(L_\alpha \tilde{u}, L_\alpha \tilde{v}) d\alpha\right)^{\frac{1}{2}},$$

and $\tilde{u}, \tilde{v}$ are two asymmetric triangular fuzzy numbers as follows:

$$\tilde{u} = (a, sl_a, sr_a)_T,$$
$$\tilde{v} = (b, sl_b, sr_b)_T,$$

where sl is the left width and sr is the right width. Then,

$$\begin{aligned} d_2(\tilde{u}, \tilde{v}) = \left(\int_0^1 \alpha d^2(L_\alpha \tilde{u}, L_\alpha \tilde{v}) d\alpha\right)^{\frac{1}{2}} &= \Big(\frac{1}{4}(sl_b - sl_a)^2 + \frac{1}{2}[(a-b) + (sl_b - sl_a)]^2 \\ &- \frac{2}{3}(sl_b - sl_a)[(a-b) + (sl_b - sl_a)] \\ &+ \frac{1}{4}(sr_b - sr_a)^2 + \frac{1}{2}[(a-b) - (sr_b - sr_a)]^2 \\ &+ \frac{2}{3}(sr_b - sr_a)[(a-b) - (sr_b - sr_a)]\Big)^{\frac{1}{2}}. \end{aligned}$$

and BCFM will be equal to:

$$BCFM(\tilde{u}_i, \tilde{u}_j, t) = \frac{c}{c + d_2(\tilde{u}, \tilde{v})} \cdot \frac{t}{t + \|i - j\|_\infty}. \tag{6}$$

Now, the estimate of the noisy pixel for 10 *repetitions in bootstrap sampling is shown in Figure 10:*

45	50	49	50	48	45	45
50	50	49	60	48	45	45
45	45	44	45	159	50	50
45	51	161	162	159	155	155
45	152	157	162	154	155	155
154	154	156	153	151	150	150
155	155	156	156	151	150	150

Figure 10. Image matrix after noise reduction.

It can be seen that the value of this pixel (162) *is the best estimate for the center of the matrix, which was* 165 *as the actual value. That is, while using the median filter, the alternative value will be equal to* 157. *Of course, it should be noted that to achieve a convergence in estimation, it is necessary to increase the number of iterations.*

In the following, as a real example, we removed the noise from the gray and color images. First, for the gray image, we examined the lung CT scan image related to the COVID-19 virus. Then, we used the images of Lenna and Baboon for the color images. For each image, we compared the result of the filter presented in this paper with the median filter and *CFM* using the mean absolute error (*MAE*) and peak signal-to-noise ratio (*PSNR*).

Example 5. *A lung CT scan is a common imaging tool for diagnosing pneumonia. CT scans provide high-quality images of lung tissue, and the radiologist can quickly determine the extent of lung involvement with the disease. A CT scan of the lung reveals common radiological features in patients with COVID-19 pneumonia. Noise in medical images can make it difficult to diagnose COVID-19. In this section, we remove the noise from the lung CT scan image in a patient with the COVID-19 virus by using the BCFM filter. To show the power of the BCFM filter for fuzzy sets in noise removal or reduction, we first added noise (salt and pepper of 5 and 10 percent) to the initial image.*

After using the filter, the results are in Figure 11 and Table 1. As we can see in Table 1, the BCFM filter has the lowest MAE and the highest PSNR among the three filters BCFM, CFM, and Median. Furthermore, as the number of bootstrap samples increases, MAE and PSNR will converge to the minimum and maximum values, respectively.

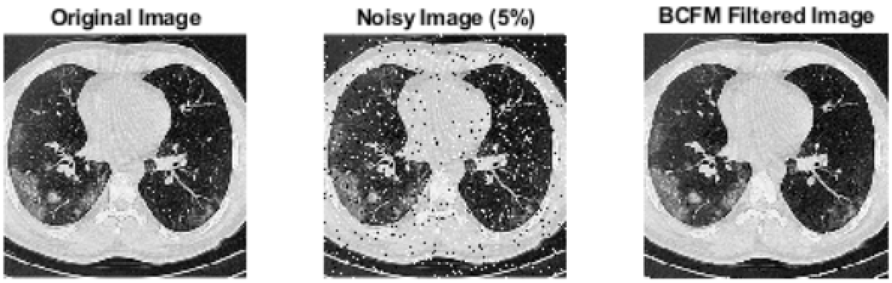

Figure 11. 1. Original COVID-19 image. 2. Noisy image after adding noise. 3. BCFM filtered image for $n = 10$ bootstrap resampling.

To better compare the two methods CFM and BCFM, we can use the probability density graph of images. Figure 12 shows the density of the original image in black, CFM in green, and BCFM in red. Whichever density of green or red is closer to the density of black indicates the better performance of the method.

Careful examination of the densities shows that, except in the range of 0 *to about* 30, *the red density is closer to black, and in general, it can be said that the red density performs better in most places or is similar to the green density. To better illustrate this point, we see a slice of the graph in Figure 13.*

Table 1. Comparison of the performance of noise reduction filters for the COVID-19 image.

Noise Percentage	Filter Evaluation	Median Filter	CFM Filter	*BCFM* Filter		
				$n = 10$	$n = 20$	$n = 50$
5	MAE	5.0347	1.009	0.7990	0.7948	0.7935
5	PSNR	22.5748	24.2157	24.7815	24.7672	24.7862
10	MAE	5.4470	1.7251	1.5781	1.5621	1.5610
10	PSNR	21.8976	22.7346	23.0609	23.0460	23.0580

Note: n is the number of bootstrap resampling.

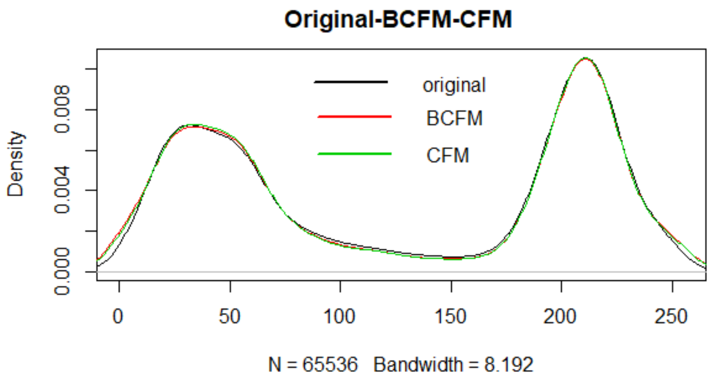

Figure 12. Density graph of the original, CFM, and BCFM images.

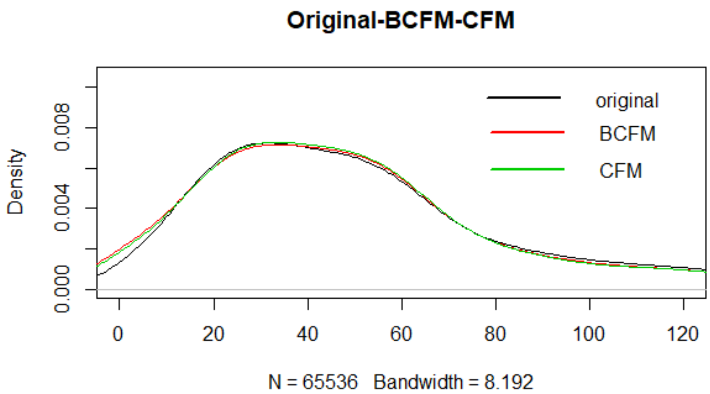

Figure 13. A slice density graph of the original, CFM, and BCFM images.

By observing the image details and examining the pixels, it is observed that BCFM has a softer approach to image details than CFM. In other words, the modified image loses more detail by the CFM method, and this can be a reason for the better performance of BCFM.

Example 6. *In this example, we compared the performance of the BCFM filter in color images with other methods. For the performance comparison, we used the MAE and PSNR for different densities of noise for the test images.*

Here, we added 5 *and* 10 *percent noise (salt and pepper) to the images of Lenna and Baboon. The results of the BCFM filter with the number of bootstrap samples being 10, 20, and 50 are reported in Figures 14 and 15 and Tables 2 and 3.*

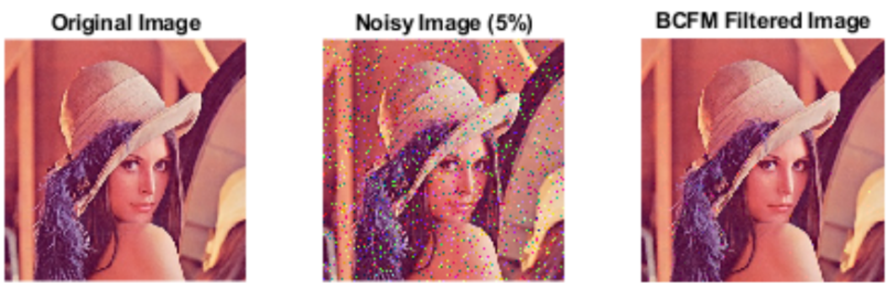

Figure 14. 1. Original Lenna image. 2. Noisy image after adding noise. 3. BCFM filtered image for $n = 10$ bootstrap resampling.

Table 2. Comparison of the performance of the noise reduction filters for the Lenna image.

Noise Percentage	Filter Evaluation	Median Filter	CFM Filter	*BCFM* Filter		
				$n = 10$	$n = 20$	$n = 50$
5	MAE	2.2008	0.3244	0.2871	0.2862	0.2851
5	PSNR	28.2372	32.2956	32.5741	32.6900	32.6968
10	MAE	2.3839	0.5766	0.5513	0.5500	0.5498
10	PSNR	27.2823	30.3462	30.5876	30.6003	30.6152

Note: n is the number of bootstrap resampling.

Also, for the image of Baboon, the results are as follows:

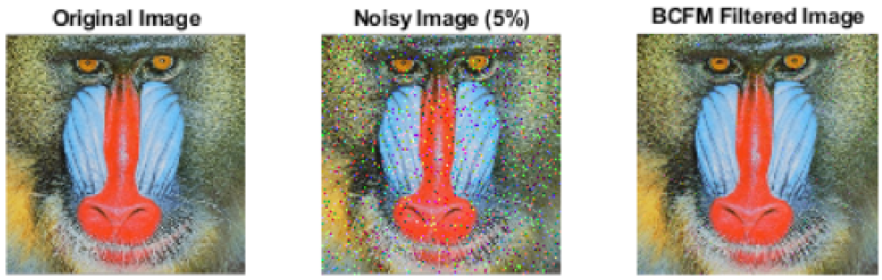

Figure 15. 1. Original Baboon image. 2. Noisy image after adding noise. 3. BCFM filtered image for $n = 10$ bootstrap resampling.

Table 3. Comparison of the performance of the noise reduction filters for the Baboon image.

Noise Percentage	Filter Evaluation	Median Filter	CFM Filter	*BCFM* Filter		
				$n = 10$	$n = 20$	$n = 50$
5	MAE	4.5121	1.1825	1.0452	1.0422	1.0406
5	PSNR	24.7003	27.9888	28.3880	28.3888	28.4157
10	MAE	4.6989	1.5336	1.3862	1.3841	1.3799
10	PSNR	24.3602	27.0842	27.4901	27.4877	27.4813

Note: n is the number of bootstrap resampling.

Figure 16 shows how MAE and PSNR converge to the minimum and maximum values as the sample size increases, respectively.

Clearly, Tables 2 and 3 also confirm the results shown in Example 5 for color images.

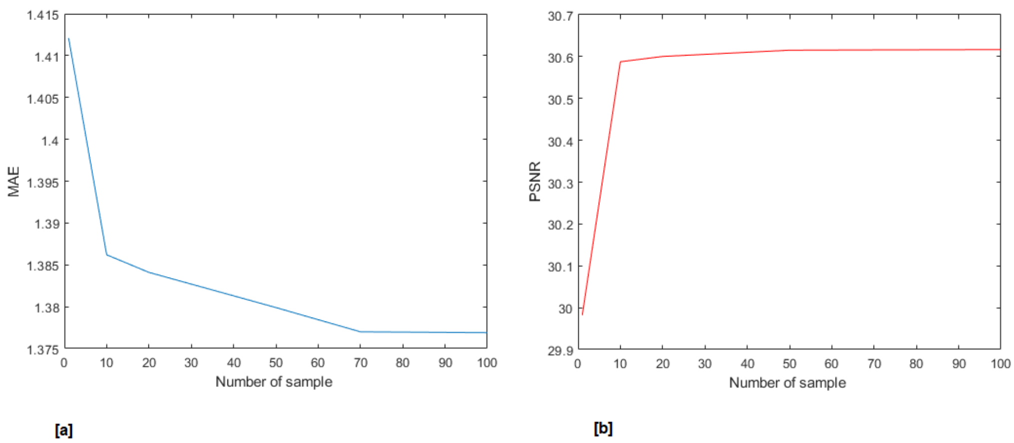

Figure 16. (**a**) MAE of the Baboon image with 10 percent noise. (**b**) PSNR of Baboon image with 10 percent noise

7. Conclusions

Colors are one of the places where the fuzzy concept can be greatly used. If we express the color of each pixel as a fuzzy number in an image, we can separate the color of the pixels more accurately. In this paper, we introduced an approach to determine the fuzzy color value of each pixel and reduce the noise of an image by using convergence in the mean bootstrap (according to the *SLLN* in the fuzzy metric space [14]).

In fact, the *BCFM* filter relies on the *SLLN* for the bootstrap mean [14], eliminates noise, and as shown in the results, performs better than the *CFM* and *Median* filters.

Two reasons can be considered the main factors improving the results. The first reason can be seen in the use of the fuzzy color pixel approach, which, according to the expressed triangular fuzzy number for each pixel, finds a more accurate mean of the central pixel neighborhoods of each window. The second reason for the performance improvement can be expressed by using the bootstrap mean to calculate the mean of the neighbors, which, by increasing the sample, reduces the effect of outliers (noisy pixels) on the mean. In 2007, Morillas et al. [15] compared the *CFM* filter to other methods, and the results showed that *CFM* performed better than the other filters. In this paper, we compared the *BCFM* filter with *CFM*, which shows the better performance of the *BCFM* filter compared to the other methods.

Author Contributions: All authors contributed to the study conception and design. The material preparation, data collection, and analysis were performed by R.G., S.M., A.N. and M.R. The first draft of the manuscript was written by R.G., and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding: Samuel Morillas acknowledges the support of the Generalitat Valenciana under Grant AICO2020-136.

Data Availability Statement: Data will be made available upon reasonable request.

Conflicts of Interest: The authors have no conflict of interest to declare that are relevant to the content of this article.

Code Availability: The codes developed during the current study are available from the corresponding author upon reasonable request.

References

1. Kramosil, I.; Michálek, J. Fuzzy metrics and statistical metric spaces. *Kybernetika* **1975**, *11*, 336–344.
2. Menger, K. Statistical metrics. *Proc. Natl. Acad. Sci. USA* **1942**, *28*, 535–537. [CrossRef] [PubMed]
3. George, A.; Veeramani, P. On some results in fuzzy metric spaces. *Fuzzy Sets Syst.* **1994**, *64*, 395–399. [CrossRef]
4. Morillas, S.; Gregori, V.; Peris-Fajarnés, G.; Latorre, P. A fast impulsive noise color image filter using fuzzy metrics. *Real-Time Imaging* **2005**, *11*, 417–428. [CrossRef]
5. Morillas, S.; Gregori, V.; Hervás, A. Fuzzy peer groups for reducing mixed gaussian-impulse noise from color images. *IEEE Trans. Image Process.* **2009**, *18*, 1452–1466. [CrossRef]
6. Morillas, S.; Gómez-Robledo, L.; Huertas, R.; Melgosa, M. Fuzzy analysis for detection of inconsistent data in experimental datasets employed at the development of the ciede2000 colour-difference formula. *J. Mod. Opt.* **2009**, *56*, 1447–1456. [CrossRef]
7. Gregori, V.; Miñana, J.J.; Morillas, S. Some questions in fuzzy metric spaces. *Fuzzy Sets Syst.* **2012**, *204*, 71–85. [CrossRef]
8. Gregori, V.; Morillas, S.; Sapena, A. Examples of fuzzy metrics and applications. *Fuzzy Sets Syst.* **2011**, *170*, 95–111. [CrossRef]
9. Gregori, V.; Morillas, S.; Roig, B.; Sapena, A. Fuzzy averaging filter for impulse noise reduction in colour images with a correction step. *J. Vis. Commun. Image Represent.* **2018**, *55*, 518–528. [CrossRef]
10. Bandyopadhyay, S.; Das, S.; Datta, A. A hybrid fuzzy filtering-fuzzy thresholding technique for region of interest detection in noisy images. *Appl. Intell.* **2020**, *50*, 1112–1132. [CrossRef]
11. Ralević, N.M.; Karaklić, D.; Pištinjat, N. Fuzzy metric and its applications in removing the image noise. *Soft Comput.* **2019**, *23* 12049–12061. [CrossRef]
12. Athreya, K. Strong law for the bootstrap. *Stat. Probab. Lett.* **1983**, *1*, 147–150. [CrossRef]
13. Athreya, K.B.; Ghosh, M.; Low, L.Y.; Sen, P.K. Laws of large numbers for bootstrappedU-statistics. *J. Stat. Plan. Inference* **1984**, *9*, 185–194. [CrossRef]
14. Ghasemi, R.; Rabiei, M.R.; Nezakati, A. Strong Law of Large Numbers for Fuzzy Random Variables in Fuzzy Metric Space. *Int. J. Fuzzy Log. Intell. Syst.* **2020**, *20*, 278–289. [CrossRef]

15. Morillas, S.; Gregori, V.; Peris-Fajarnes, G.; Sapena, A. New adaptive vector filter using fuzzy metrics. *J. Electron. Imaging* **2007**, *16*, 033007. [CrossRef]
16. Schweizer, B.; Sklar, A. Statistical metric spaces. *Pac. J. Math.* **1960**, *10*, 313–334. [CrossRef]
17. Klement, E.P. ; Mesiar, R.; Pap, E. *Triangular Norms*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 2000.
18. Zadeh, L. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
19. Deng, Z. Fuzzy pseudo-metric spaces. *J. Math. Anal. Appl.* **1982**, *86*, 74–95. [CrossRef]
20. Kaleva, O.; Seikkala, S. On fuzzy metric spaces. *Fuzzy Sets Syst.* **1984**, *12*, 215–229. [CrossRef]
21. Erceg, M.A. Metric spaces in fuzzy set theory. *J. Math. Anal. Appl.* **1979**. *69*, 205–230. [CrossRef]
22. Saadati, R.; Vaezpour, S. Some results on fuzzy banach spaces. *J. Appl. Math. Comput.* **2005**. *17*, 475–484. [CrossRef]
23. Efron, B. *Bootstrap Methods: Another Look at the Jackknife Breakthroughs in Statistics*; Springer: New York, NY, USA, 1992; pp. 569–593.
24. Csörgo, S. On the law of large numbers for the bootstrap mean. *Stat. Probab. Lett.* **1992**. *14*, 1–7. [CrossRef]
25. Klement, E.; Puri, M.L.; Ralescu, D. Limit theorems for fuzzy random variables. *R. Soc.* **1986**, *407*, 171–182. [CrossRef]