sciendo

DOI: 10.2478/ijasitels-2021-0004

Filtering Random Valued Impulse Noise from Grayscale Images through Support Vector Machine and Markov Chain

Arpad Gellert[1], Remus Brad[1], Daniel Morariu[1], Mihai Neghina[1]

[1]Computer Science and Electrical and Electronics Engineering Department, Faculty of Engineering, "Lucian Blaga" University of Sibiu, Romania

{arpad.gellert, reus.brad, daniel.morariu, mihai.neghina}@ulbsibiu.ro

Abstract

This paper presents a context-based filter to denoise grayscale images affected by random valued impulse noise. A support vector machine classifier is used for noise detection and two Markov filter variants are evaluated for their denoising capacity. The classifier needs to be trained on a set of training images. The experiments performed on another set of test images have shown that the support vector machine with the radial basis function kernel combined with the Markov+ filter is the best configuration, providing the highest noise detection accuracy. Our filter was compared with existing denoising methods, it being better on some images and comparable with them on others.

Keywords: random valued impulse noise, support vector machine, Markov filter, denoising

1 Introduction

Many images are affected by noise, either from the communication channel or mainly from the sensor; therefore, noise filtering is also nowadays one topic of interest. Image noise can be defined by the alteration of the intensity channel for grayscale or the RGB subpixels for color images. There are several types of image noise, as the Gaussian noise, which is a statistical noise with a normal image distribution, occurring in digital images during acquisition (e.g., noise sensor caused by low light and / or high temperature). Another type of noise affecting images is the impulse noise. The so called "salt-and-pepper" is a pulse-type noise, which shifts pixels into white or black. There is also a pulse-type noise that alters pixels with random valuesover the entire range of [0, 255]. Other types of noise are quantized noise, granularity noise or periodic noise.

In the present paper, we have used a Support Vector Machine (SVM) classifier to sort pixels from images affected by random valued impulse noise into noisy or unnoisy and a Markov filter for correcting them and getting a clearest picture. As mentioned, a binary classifier based on SVM called Support Vector Classification (C-SVC), with several classes of algorithms called kernel methods. The kernels used in this paper are linear, polynomial, Radial Basis Function (RBF) and sigmoid, which will be presented during this paper.

For the training of the SVM we have used the test images of Baboon and Lena affected with 50% noise. Each kernel method is trained with these two images, resulting in multiple model files that will be used to predict noisy pixels. The training images are compared with the original images, unaffected by noise, pixel by pixel, thus obtaining the training files. Testing and, implicitly the classification based on these kernel methods, was done on the Airplane, Cameraman and Boat images, on noise levels starting from 10%, increasing with 10% up to 90% noise.

The SVM was trained and tested based on the differences between the current / central pixel of a 3×3 window and its 8 neighbors in the noisy images. All these differences and the results obtained from them are scaled to the interval [–1,1]. The results have shown that the best kernel is the RBF (the polynomial kernel is slightly weaker). It also turns out that the sigmoid kernel does not performs well at all. The SVM with RBF kernel was used to select the noisy pixels processed then by our Markov or Markov+ denoising methods. Both filters were already applied in [6] and [7] for denoising images affected by salt-and-pepper noise, considering all the black and white pixels noisy, but also for image inpainting in [8]. To apply our filters to denoise images affected by random valued impulse noise, the additional classifier component was necessary. Finally, the SVM-based Markov filters were compared with other existing denoising methods functional on random valued impulse noise.

In the following section, a state-of-the-art in the field of denoising is presented, followed by the proposed method. The next section shows the obtained results and the corresponding comparison followed by the conclusions and further work directions.

2 Related Work

In signal processing, noise appears in two basic forms: as a modifier or as a replacer of existing sample values (or pixel values in the case of images). Additive or multiplicative noises from various distributions (Gaussian, uniform, etc.) are instances of modifier noise. Impulse noise, on the other hand, acts a replacer of existing sample values. The most studied example from this category is the extreme fixed valued impulse noise called "salt-and-pepper", usually encountered in the context of image processing, which gets its name from the effect of rendering random pixels white or black, similar to sprinkling the aforementioned spices on the image.

© 2021 Lucian Blaga University of Sibiu

More generally, random valued impulse noise can replace sample values with any value from the definition interval, not only the extremes. Many filters that work well with fixed valued impulse noise have trouble with random valued impulse noise, since it can be harder to confidently distinguish and remove.

Still, a great variety of approaches has been attempted for filtering such impulse noise. Among the many median filter (MF) based methods, Hamza et al. [10] use a relaxed median filter (RMF) to statistically balance noise elimination with preservation of edge and line details for highly corrupted inputs, while Jin et al. [12] use statistics in the absolute difference of rank-ordered pixels to adjust the weights of the filter. Banerjee et al. [3] detect and correct pixels based on the statistical properties and noise density of regions. In [26], Wang and Zhang introduced the progressive switching median filter (PSMF), which iteratively applies the impulse noise detection and filtering. Wang et al. [27] proposed a two-phase modified switching median filter. Also having two phases, Zhu et al. [29] use distinct impulse noise detection and correction phases, dealing separately with obviously disturbing noise pixels and attending to the "just noticeable differences" afterwards.

In [23], Srinivasan and Ebenezer introduced a decision-based algorithm (DBA), which applies the filter only on black and white pixels. Kunsoth et al. [14] use an improved decision-based median filter where the processing window side is increased as needed in order to get a confident decision. The Analysis Prior Algorithm (APA) applying the majorization-minimization principle was introduced and evaluated in [16].

Toh et al. [24] described a multi-stage noise adaptive fuzzy switching median filter (NAFSMF). More recently, Nadeem et al. [19] presented a method for dealing with random valued impulse noise in highly corrupted grayscale images through a spatial-directional adaptive filter with fuzzy rules. Another method for random valued impulse noise was proposed by Iqbal et al. [11], with the noise detection performed based on the computation of new and adaptive thresholds for each pixel and the denoising done with a weighted mean filter. Convolutional neural networks were used for random valued impulse noise filtering in [13]. Many other approaches involving artificial intelligence focus on neural networks in [2], [18] [22], [25] and [1].

Other context-based methods worth mentioning include the non-local approach of Buades et al. [4], correcting the noise with weighted averages of all pixels in the image, the stochastic method Estrada et al. [5], employing random walks beginning at the considered pixel as weights and the Markov-chain Monte Carlo sampling of Wong et al. [28]. Matsuoka et al. [17] even attempt to remove random valued impulse noise in color images treating pixels as vectors, thus using non-local vector median filtering.

In [15], Lin presented a filter which uses SVM for noise detection and fuzzy correction. The context-based filter proposed in our paper has two phases: the detection uses SVM, but the correction of the noise is done with a Markov model.

© 2021 Lucian Blaga University of Sibiu

3 SVM-Based Markov Filter

In this section we present the two components of the proposed method: the SVM-based noise detection and the context-based filter applied on the detected noisy pixels. The scheme of the method is presented in Figure 1.

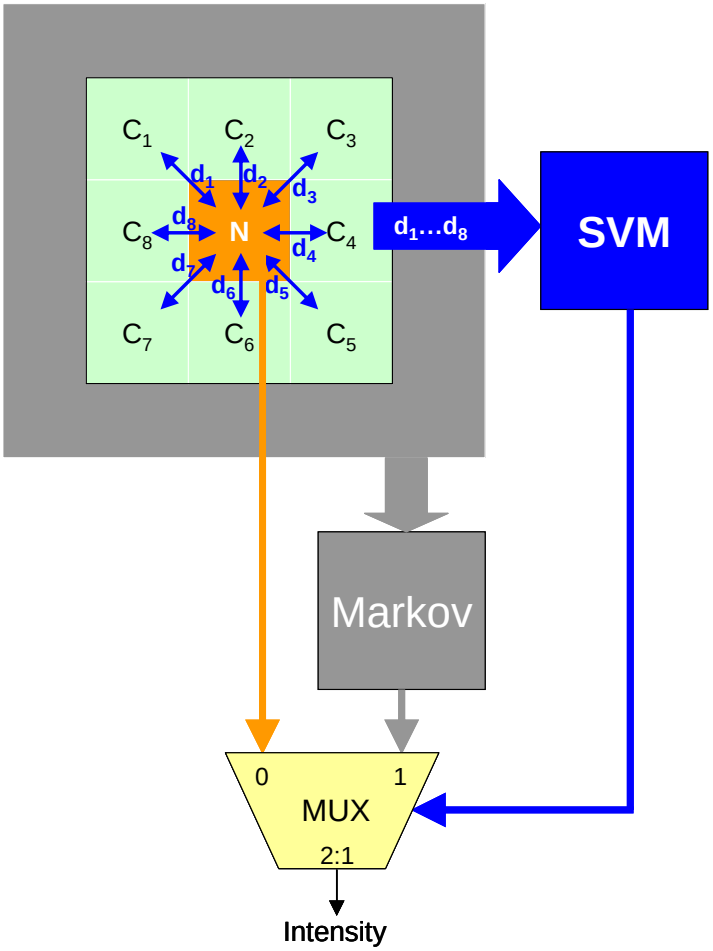

Figure 1. The scheme of the SVM-based Markov filter

As Figure 1 illustrates, the differences between the intensity of the current pixel and the intensities of the surrounding 8 pixels are used by the SVM classifier to decide if the current pixel is noisy or not. If the current pixel is classified as unnoisy, then its intensity is not changed. On the other hand, if it is classified as noisy, then its intensity is replaced with the intensity provided by the Markov filter. Next, this section details the two main components of our method: the SVM-based classifier and two Markov filter variants.

3.1 Random Valued Impulse Noise Detection with SVM

In this paragraph, the approach of detecting random valued impulse noise through SVM is presented. The SVM is used to classify pixels in two categories: noisy and unnoisy. In the training stage, the SVM receives as input characteristics the intensity differences between the current pixel and

© 2021 Lucian Blaga University of Sibiu

the 8 neighboring pixels, scaled to the [–1, 1] interval, and as target the value +1 in the case of a noisy pixel or –1 in the case of unnoisy pixel (determined by comparing the noisy image with the original clean image). In the testing stage, the SVM receives the same scaled differences and it returns the decision as +1 (noisy) or –1 (unnoisy).

3.1.1 Support Vector Machine

The SVM is a classification algorithm that knows to work only with two classes and uses a statistical learning theory [21] as learning technique. As advantages of this algorithm are: working with large datasets and the capability to learn with great success non-linear classification problems.

The SVM algorithm tries to find a separation line (called hyperplane) that will optimally split the training dataset. The idea of the optimal hyperplane is: that hyperplane needs to maximize the distance between it and each class (positive and negative) taken in consideration the samples that are closer to the hyperplane. Based on the hyperplane, the algorithm uses a decision function like in equation (1) to make decision for unseen samples (denoted x in the equation).

$$f(x) = \operatorname{sgn}\left(\langle w, \Phi(x)\rangle + b\right) \tag{1}$$

The computed hyperplane is characterized by a weight vector denoted w, that is orthogonal to the hyperplane, a margin of the hyperplane denoted b and a function $\Phi(x)$ that is used to transfer the input data into a higher dimensional feature space. The output of the decision function is converted into +1 or –1 using a simple signum function denoted *sgn*. The weight vector w is normalized at the unit length (generally w will be scaled by $||w||$), then the dot product $<w, \Phi(x)>$ has the $\Phi(x)$ length in the direction of w. For learning, the algorithm needs to compute the vector w and the scalar b that correctly classifies all elements from the training dataset, and the decision line is called "optimal classification hyperplane". In the learning phase, the algorithm finds a lot of "good" hyperplanes that meet the classification requirements, but at the end the algorithm needs to choose the optimal one. The function f is constructed from empirical data and the algorithm is performed in a dot product space. For this we based on the two facts: first, from all the existing possible hyperplanes there is only one that maximizes the margins between them and the training points that are closer to the hyperplane (optimal hyperplane), and the second, the hyperplane's capacity to separate the classes decreases with the increasing of the margin. In the case of the input dataset for training, which is not separable in the input space (unfortunately in almost all cases), the SVM algorithm proposes to represent the training data into a higher-dimensional feature space using the function Φ and tries to find the optimal hyperplane in that space. Mapping the hyperplane found into the input space determines the hyperplane becoming a non-linear decision boundary in the input space. In the algorithm, this idea is a simplification because we use all vectors into a dot product and we can use the kernel trick,

© 2021 Lucian Blaga University of Sibiu

with one condition, to have a positive defined kernel function $\langle w, \phi(x) \rangle$. Thus, it is possible to compute the hyperplane into a higher dimensional space without representing the training dataset into that space [21]. In the algorithm, for the computation of the optimal hyperplane, we have to solve the function, called primal optimization function with restrictions, presented in the equation (2):

$$\underset{\mathbf{w}\in H, b\in \Re}{\text{minimize}} \ \tau(w) = \frac{1}{2}\|w\|^2 \tag{2}$$

subject to $y_i\left(\langle w, x_i \rangle + b\right) \geq 1$ for all i=1,...,m.

The constraints are included in the formula because we want the function $f(x_i)$ being +1 for $y_i = +1$ and −1 for $y_i = -1$. Even if there are efficient algorithms that solve this type of problems, these algorithms do not work correctly in all cases. Thus, we prefer to convert this type of problem in the so-called dual optimization problem introducing the Lagrange multipliers $\alpha_i \geq 0$ and the Lagrange [20] and including also the restrictions in the formula as in equation (3).

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i (y_i (\langle x_i, w \rangle + b) - 1) \tag{3}$$

The Lagrange function *L* must be minimized with respect to the primal variables *w* and *b* and needs to be maximized with respect to the dual variables α_i. Putting those conditions, we can obtain:

$$w = \sum_{i=0}^{m} \alpha_i y_i x_i \quad \text{and} \quad \sum_{i=0}^{m} \alpha_i y_i = 0 \tag{4}$$

Observe that the solution *w* is unique, due to the strict convexity of primal optimization problem, and the sum of α_i needs to be zero. Considering the Karush-Kuhn-Tucker (KKT) theorem, there are only some α_i that are non-zero that correspond only to samples considered saddle points. Thus, we can write those restrictions for all the samples from the training dataset:

$$\alpha_i \left[y_i \left(\langle x_i, w \rangle + b \right) - 1 \right] = 0 \tag{5}$$

The samples x_i for which the dual variables are different by zero are called in the algorithm "Support Vectors" and are samples that are very close to the margin of the hyperplane (decide the aspect of the hyperplane). All the samples from the training dataset that have α (dual variable) equal with zero are irrelevant and can be ignored when decide the position of the hyperplane. Eliminating the primal variables in the Lagrange function we obtain the formula that is usually solved in practice (equation 6) and we can re-write the decision function in the dual optimization problem representation (see equation 7).

$$\underset{\alpha \in \Re^m}{\text{maximize}} \ W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \tag{6}$$

$$f(x) = \text{sgn}\left(\sum_{i \in \Re} y_i \alpha_i \langle x_i, x \rangle + b \right) \tag{7}$$

© 2021 Lucian Blaga University of Sibiu

3.1.2 Used kernel functions

In the equation 7 we formulate the decision function including the vectors only in the dot product. This complication was made because we want to perform the algorithm in a higher dimensional feature space without representing the input data in that space. These initial complications will lead to a massive reduction in the calculations that need to be done in the testing phase. This occurs because using a defined positive kernel *k*, such that $k(x,x') = \langle x, x' \rangle$, we can compute the dot product into a higher space without representing the data in that space. This idea is referred in the literature as the kernel trick [21] and it is used to extend a linear classification to a nonlinear classification. Thus, the function Φ representing the input vector in the new space does not need to be computed. Taking into consideration those new aspects, the decision function can be written as:

$$f(x) = \mathrm{sgn}\left(\sum_{i \in \Re} y_i \alpha_i k(x, x_i) + b \right) \tag{8}$$

In this paper, we test and present results using all the kernel functions implemented in the libsvm library [30], and the formula for each kernel is presented below in eq. 9 to 12.

Linear kernel:

$$k(x,x') = (x,x') \tag{9}$$

Polynomial kernel:

$$k(x,x') = \left(coef_0 + gamma * (x,x')\right)^{\mathrm{deg}ree} \tag{10}$$

Radial Basis function kernel:

$$k(x,x') = e^{(-gamma\ *|x-x'|)^2} \tag{11}$$

Sigmoid kernel:

$$\tanh(coef_0 + gamma * (x * x')) \tag{12}$$

3.2 Random Valued Impulse Noise Filtering with Markov Chain

A general prediction algorithm with Markov chain is given in [9]. The principle of the Markov filter introduced in [6] is depicted in Figure 2. A noisy pixel intensity should be replaced with the intensity which has the highest probability to appear in contexts which are similar with the noisy pixel's context. The probability of an intensity to appear in a context is directly proportional with the frequency of the intensity in that context. Thus, in our filter, the intensity of a pixel *N* classified as being noisy by the SVM is replaced with the most frequent noise-free intensity occurring in similar contexts within the surrounding area limited by a certain search distance (SD). The context pixels *C* form a square having the side *CS* around the noisy pixel.

© 2021 Lucian Blaga University of Sibiu

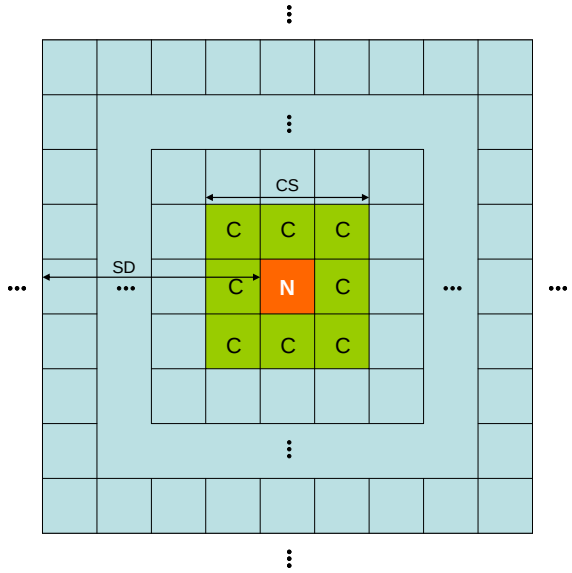

Figure 2. The principle of the Markov filter

The Markov+ filter introduced in [7], is using a context in form of the plus symbol having the size *CS* (see Figure 3). The search for the context is performed in form of the 8-spoked asterisk symbol. The intensity of a noisy pixel is replaced with the most frequent intensity met in similar contexts.

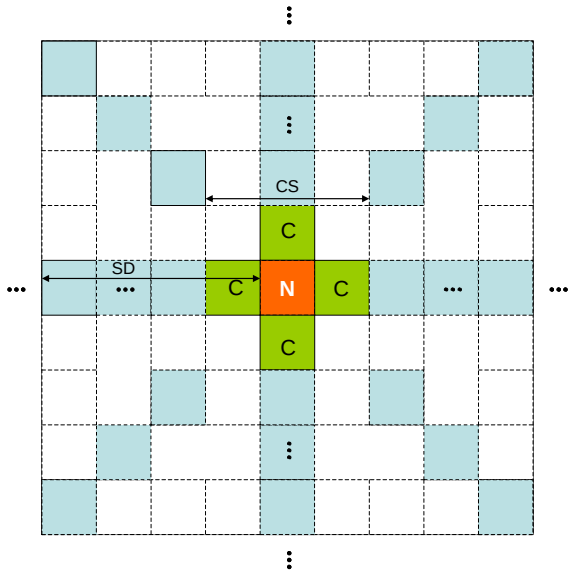

Figure 3. The principle of the Markov+ filter

In both Markov filter variants, the context of the noisy pixel is compared with other blocks of the same form using the sum of absolute differences SAD metric. If the SAD value is less than a certain threshold value *T*, then the compared blocks of pixels are considered similar. Obviously, the optimal value of *T* depends on the number of pixels from the context. As larger is the context, as higher is the similarity threshold *T*.

© 2021 Lucian Blaga University of Sibiu

4 Experimental Results

The filters used for comparisons were available in Matlab. The SVM was used from the LIBSVM library and the Markov filters were implemented in C#. For the Markov filters we kept the optimal parameter values established in [6] and [7]. Thus, the *CS* is 3, the *SD* is 4, whereas the threshold *T* is 500 for the Markov filter and 300 in the case of the Markov+ filter.

The SVM was trained on the Baboon and Lena images with 50% of random valued impulse noise. We chose a noise level of 50% for these training images to assure the same conditions to learn both the noise and noise-free cases. For evaluations, we used the Airplane, Boat and Cameraman test images with random valued impulse noise of levels between 10% and 90%. All the images are 512×512 grayscale PNGs.

As metrics we used the noise detection accuracy and the mean squared error (MSE). The noise detection accuracy is the percentage of pixels correctly classified into the noisy and unnoisy categories. The MSE shows the difference between a filtered image *F* and the original unnoisy image:

$$MSE = \frac{\sum_{i=1}^{W}\sum_{j=1}^{H}\left(F(i,j)-O(i,j)\right)^2}{W \cdot H} \tag{13}$$

where *W* and *H* are the width and the height of the images, respectively.

Next, we present the experimental results. First, we measured the noise detection accuracy of the SVM considering different kernels: linear, polynomial, RBF and sigmoid. We are interested to find out which of these kernels is the best in classifying the pixels as noisy or unnoisy. Figures 4-6 depict the results obtained for the Airplane, Boat and Cameraman test images, respectively.

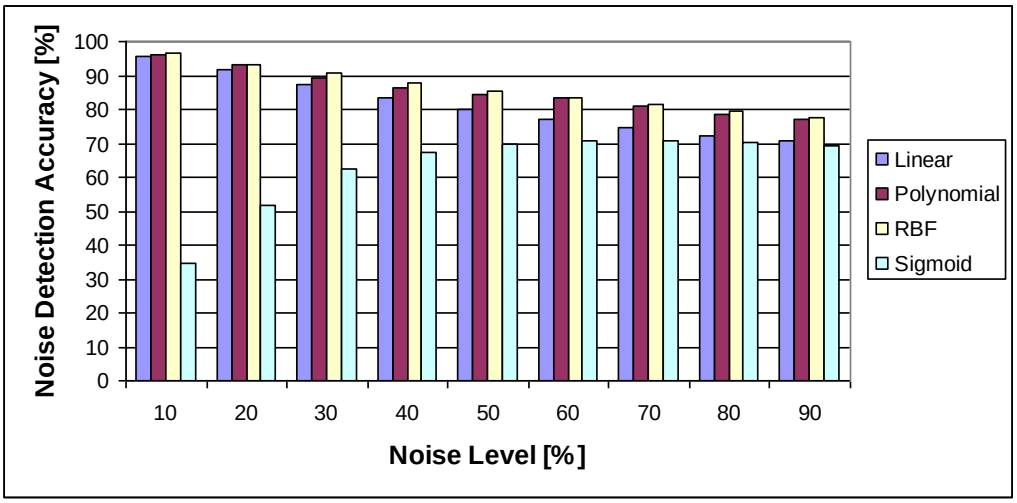

Figure 4. Noise detection accuracy obtained on the Airplane test image with SVM considering different kernels

© 2021 Lucian Blaga University of Sibiu

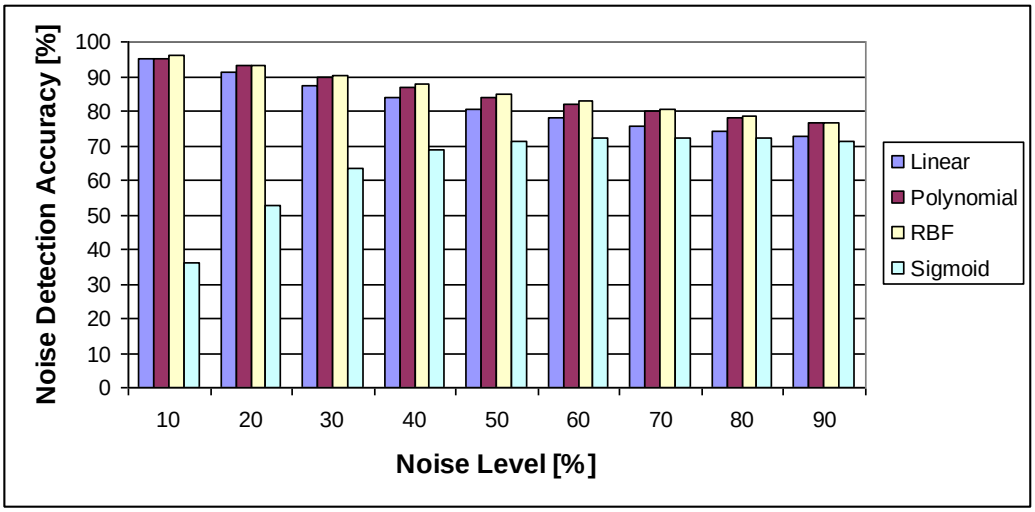

Figure 5. Noise detection accuracy obtained on the Boat test image with SVM considering different kernels

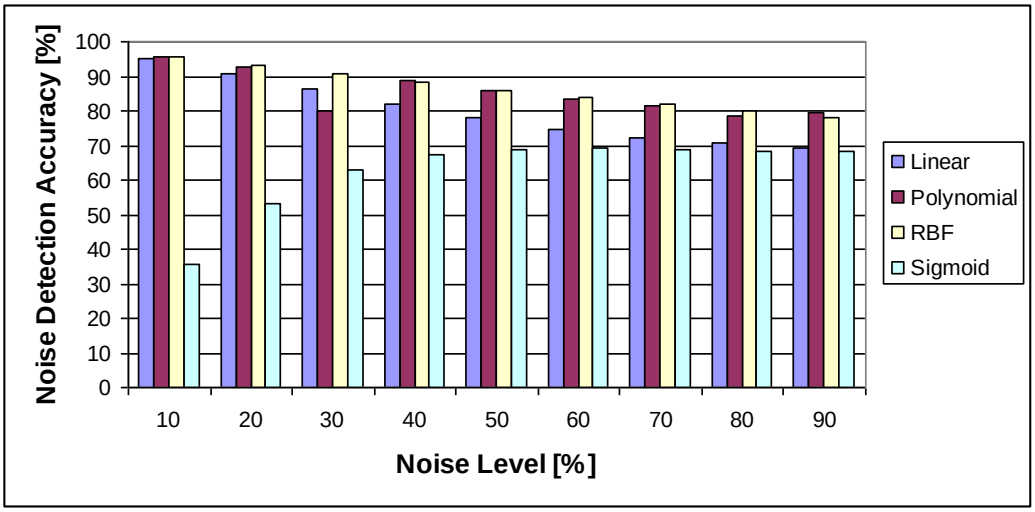

Figure 6. Noise detection accuracy obtained on the Cameraman test image with SVM considering different kernels

As the results show, in almost all the cases the SVM with the RBF kernel performed better. The polynomial kernel is also good, providing results close to those obtained with the RBF kernel. The linear kernel has good results only on low noise levels. The sigmoid kernel is the weakest one, producing the lowest prediction accuracy in all the cases, but especially in the case of low noise densities. Consequently, we choose the RBF kernel for the SVM.

Next, we compare in terms of MSE both our optimal SVM-based Markov filters with the MF, PSMF, RMF and APA filters which are functional on random valued impulse noise. The NAFSMF and DBA denoising methods are not working on random valued impulse noise. The comparisons are shown in Figures 7-9 on the Airplane, Boat and Cameraman test images, affected by different levels of random valued impulse noise.

© 2021 Lucian Blaga University of Sibiu

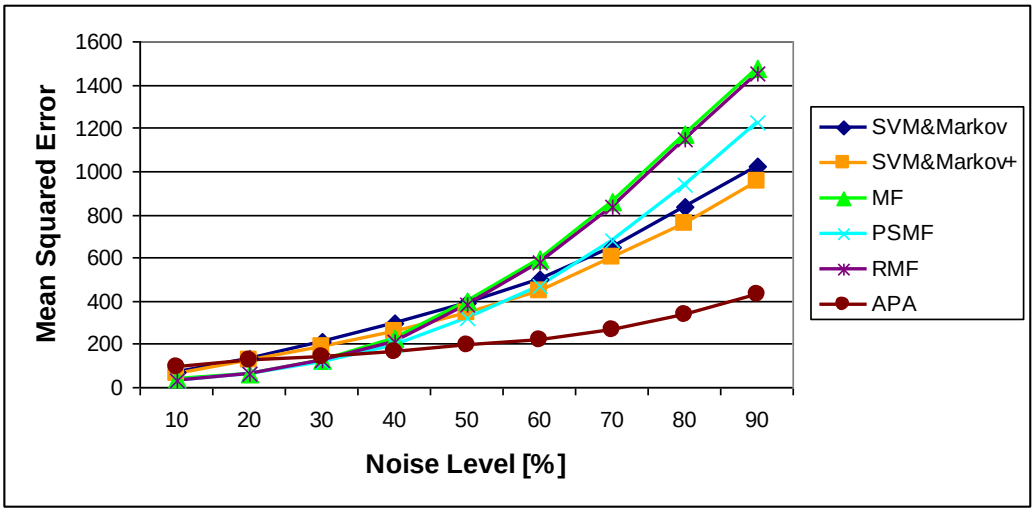

Figure 7. The MSE obtained using the studied filters on the Airplane test image

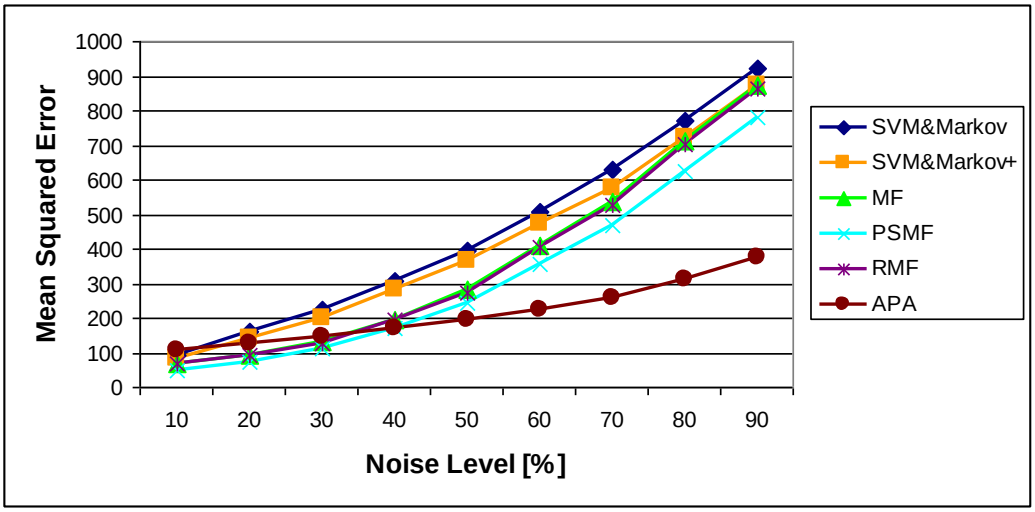

Figure 8. The MSE obtained using the studied filters on the Boat test image

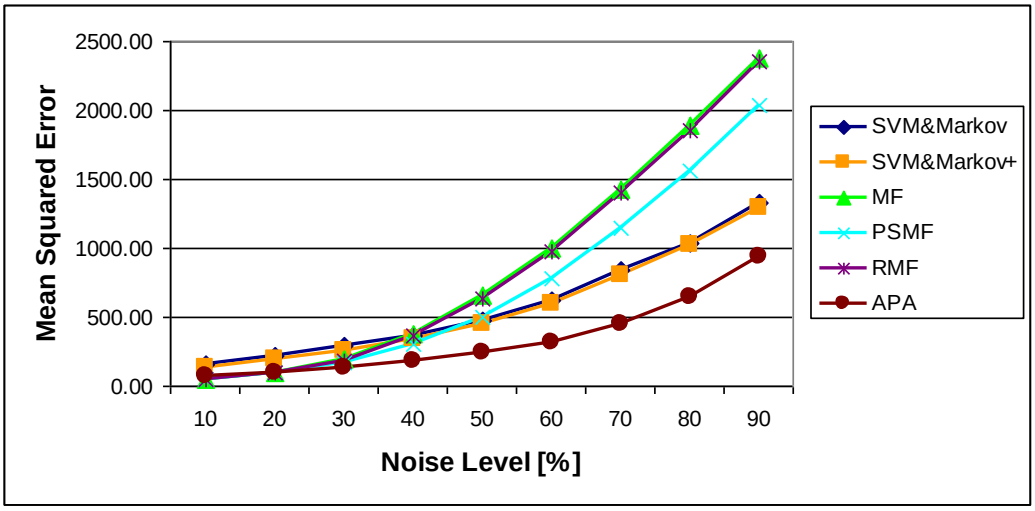

Figure 9. The MSE obtained using the studied filters on the Cameraman test image

© 2021 Lucian Blaga University of Sibiu

As we expected, the Markov+ filter outperforms the Markov filter on images affected by random valued impulse noise, as it did in [7] in the case of salt-and-pepper noise, too. The SVM-based Markov+ filter is the second best after the APA on the Airplane and Cameraman test images, however, it outperforms the APA on low noise levels. On the Boat test image, the SVM-based Markov+ filter has similar results with the MF and RMF especially in the case of high noise densities, but it is clearly outperformed by the APA denoising method on high noise levels and by the PSMF on all the noise levels.

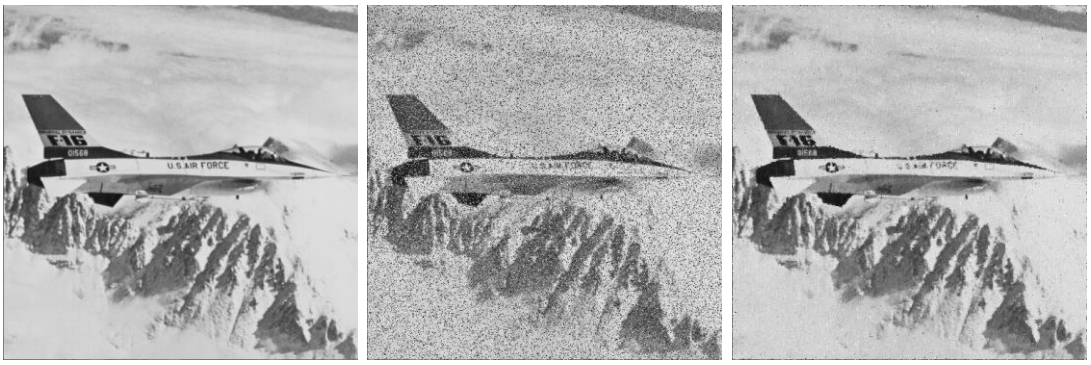

Figure 10. The Airplane test image: original, noisy (30%), denoised with the SVM-based Markov+ filter

Figure 11. The Boat test image: original, noisy (60%), denoised with the SVM-based Markov+ filter

Figure 12. The Cameraman test image: original, noisy (90%), denoised with the SVM-based Markov+ filter

© 2021 Lucian Blaga University of Sibiu

Finally, Figures 10-12 present the original, noisy and denoised Airplane, Boat and Cameraman images. Taking into account that it is a random valued impulse noise, the proposed SVM-based Markov+ filter provides quite good results.

5 Conclusions and Further Work

Without a noise detection component, our Markov filter could be used only on salt-and-pepper noise, being applied on all the black and white pixels. In this paper, we developed a SVM classifier for our Markov and Markov+ filters, which now can be applied to eliminate random valued impulse noise from grayscale images. The SVM classifier detects the noisy pixels, which then are processed by any of our Markov filter variants. The C-SVC with linear, polynomial, RBF and sigmoid kernels have been evaluated, the RBF kernel being the most efficient in terms of noise detection accuracy. After the best classifier has been decided, it was used as noise detection component for our Markov and Markov+ filters. The SVM-based Markov and Markov+ filters were then compared in terms of MSE with existing denoising methods. The experiments have shown that the SVM-based Markov+ filter is outperformed only by the APA denoising method on two test images.

As a further work direction, we intend to evaluate different neural methods for detecting random valued impulse noise.

Acknowledgment

We thank our former B.Sc. student Alin-Nicolae Popescu for his help in applying the SVM for noise detection.

References

[1] F. Agostinelli, M.R. Anderson, H. Lee, *Adaptive multi-column deep neural networks with application to robust image denosing*, Advances in Neural Information Processing Systems 26, pp. 1493–1501, Lake Tahoe, NV, USA, 2013.

[2] I. Aizenberg, G. Wallace, *Intelligent detection of impulse noise using multilayer neural network with multi-valued neurons*, SPIE Proceedings, Vol. 8295, p. 82950S, 2012.

[3] S. Banerjee, A. Bandyopadhyay, A. Mukherjee, A. Das, and R. Bag, *Random Valued Impulse Noise Removal Using Region Based Detection Approach*, Engineering, Technology & Applied Science Research, Vol. 7, No. 6, pp. 2288–2292, 2017.

[4] A. Buades, B. Coll, J.-M. Morel, *A non-local algorithm for image denosing*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 60–65, San Diego, CA, USA, 2005.

[5] F. Estrada, D. Fleet, A. Jepson, *Stochastic image denoising*, British Machine Vision Conference, p. 117, London, 2009.

[6] A. Gellert, R. Brad, *Context-Based Prediction Filtering of Impulse Noise Images*, IET Image Processing, Vol. 10, Issue 6, pp. 429–437, 2016.

© 2021 Lucian Blaga University of Sibiu

[7] A. Gellert, R. Brad, *Studying the influence of search rule and context shape in filtering impulse noise images with Markov chains*, Signal, Image and Video Processing, Springer London, Vol. 12, Issue 2, pp. 315–322, 2018.
[8] A. Gellert, R. Brad, *Image Inpainting with Markov Chains*, Signal, Image and Video Processing, Vol. 14, Issue 7, pp. 1335–1343, 2020.
[9] A. Gellert, A. Florea, *Investigating a New Design Pattern for Efficient Implementation of Prediction Algorithms*, Journal of Digital Information Management, Vol. 11, Issue 5, pp. 366–377, 2013.
[10] A.B. Hamza, P. Luque-Escamilla, J. Martínez-Aroza, R. Román-Roldán, *Removing noise and preserving details with relaxed median filters*, Journal of Mathematical Imaging and Vision, Vol. 11, Issue 2, pp. 161–177, 1999.
[11] N. Iqbal, S. Ali, I. Khan, B. M. Lee, *Adaptive Edge Preserving Weighted Mean Filter for Removing Random-Valued Impulse Noise*, Symmetry, Vol. 3, Issue 11, 2019.
[12] Q. Jin, L. Bai, J. Yang, I. Grama, Q. Liu, *A New Method for Removing Random-Valued Impulse Noise*, Neural Information Processing. ICONIP 2014. Lecture Notes in Computer Science, Vol. 8836. Springer, Cham, 2014.
[13] C. Junqing, Z. Guizhen, X. Shaoping, Y. Haiwen , *A Blind CNN Denoising Model for Random-Valued Impulse Noise*, IEEE Access, Vol. 7, pp. 124647–124661, 2019.
[14] R. Kunsoth, M. Biswas, *Modified decision based median filter for impulse noise removal*, 2016 International Conference on Wireless Communications, Signal Processing and Networking, pp. 1316–1319, 2016.
[15] T.C. Lin, *SVM-based filter using evidence theory and neural network for image denoising*, Journal of Software Engineering and Applications, Vol. 6, Issue 3B, pp. 106–110, 2013.
[16] A. Majumdar, R.K. Ward, *Synthesis and Analysis Prior Algorithms for Joint-Sparse Recovery*, IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3421–3424, 2012.
[17] J. Matsuoka, T. Koga, N. Suetake and E. Uchino, *Random-valued impulse noise removal in color images by using switching non-local vector median filter*, 2015 International Symposium on Intelligent Signal Processing and Communication Systems, pp. 11–16, 2015.
[18] S.K. Mishra, G. Panda, S. Meher, *Chebyshev functional link artificial neural networks for denosing of image corrupted by salt and pepper noise*, International Journal of Recent Trends in Engineering, Vol. 1, No. 1, pp. 42–46, 2010.
[19] M. Nadeem, A. Hussain, A. Munir, M. Habib, M.T. Naseem, *Removal of random valued impulse noise from grayscale images using quadrant based spatially adaptive fuzzy filter*, Signal Processing, Volume 169, p. 107403, 2020.
[20] C. Nello, J. Swawe-Taylor, *An introduction to Support Vector Machines*, Cambridge University Press, 2000.
[21] B. Schoslkopf, A. Smola, *Learning with Kernels*, MIT Press, London, 2002.
[22] P.L.B. Soares, J.P. Silva, *Neural networks applied for impulse noise reduction from digital images*, INFOCOMP J. Comput. Sci., Vol. 11, Issue 3–4, pp. 7–14, 2012.
[23] K.S. Srinivasan, D. Ebenezer, *A New Fast and Efficient Decision Based Algorithm for Removal of High Density Impulse Noise*, IEEE Signal Processing Letters, Vol. 14, Issue 3, pp. 189–192, 2007.
[24] K.K.V. Toh, N.A.M. Isa, *Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction*, IEEE Signal Processing Letters, Vol. 17, Issue 3, pp. 281–284, 2010.
[25] I. Türkmen, *Removing random-valued impulse noise in images using neural network detector*, Turkish Journal of Electrical Engineering and Computer Science, Vol. 22, Issue 3, pp. 637–649, 2014.

© 2021 Lucian Blaga University of Sibiu

[26] Z. Wang, D. Zhang, *Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images*, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 46, Issue 1, pp. 78–80, 1999.
[27] G. Wang, D. Li, W. Pan, Z. Zang, *Modified switching median filter for impulse noise removal*, Signal Processing, Vol. 90, Issue 12, pp. 3213–3218, 2010.
[28] A. Wong, A. Mishra, W. Zhang, P. Fieguth, D.A. Clausi, *Stochastic image denoising based on Markov-chain Monte Carlo sampling*, Signal Processing, Vol. 91, Issue 8, pp. 2112–2120, 2011.
[29] Z. Zhu, X. Zhang, *A random-valued impulse noise removal algorithm via just noticeable difference threshold detector and weighted variation method*, International Journal of Computers and Applications, 2020.
[30] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

© 2021 Lucian Blaga University of Sibiu