Impulse Noise Filtering using MLMVN

Olivia Keohane
Manhattan College
Riverdale, NY, USA
okeohane01@manhattan.edu

Igor Aizenberg
Manhattan College
Riverdale, NY, USA
igor.aizenberg@manhattan.edu

Abstract— **In this paper, we consider how a complex-valued neural network, the multilayer neural network with multi-valued neurons (MLMVN), can be efficiently used for impulse noise filtering. It is shown that MLMVN with only a single hidden layer of neurons can restore an image corrupted by random-valued impulse noise while carefully preserving edges and boundaries. MLMVN processes overlapping patches, which to a noisy image should be broken down. A network shall be trained using a robust learning set created from patches randomly picked up from many images. Then, the trained network should be used for actual filtering. Final intensities in pixels of a resulting image are obtained by averaging the intensities over all overlapping patches. This approach becomes highly efficient for images corrupted by random impulse noise with a low corruption rate of 5-10%. It makes it possible to preserve the smallest image details very carefully and avoid the smoothing of edges. MLMVN outperforms sophisticated filters with detectors of impulse noise for a lower corruption rate and shows comparable results for a higher corruption rate.**

Keywords— Complex-Valued Neural Networks, Multi-Valued Neuron, Multilayer Neural Network with Multi-Valued Neurons, MLMVN, Intelligent Filtering, Impulse noise

I. Introduction

Complex-valued neural networks (CVNN) demonstrate their high efficiency when solving various problems of pattern recognition, classification, and prediction. Their flexibility, generalization capability, and their higher functionality in comparison with their real-valued counterparts, are well known and are crucial when it is necessary to deal with highly nonlinear input/output mappings. A comprehensive review of these important features of CVNNs is given, for instance, in [1]-[3].

As it was already pointed out above, CVNNs reveal their high efficiency and superiority in solving real-world problems in a variety of significant areas. For example, forecasting of wind profiles [4] and productivity of oil wells [4], detection of landmines [6], analysis of EEG and signals decoding in brain-computer interfaces [5], and medical imaging [7].

In this paper, we consider how a complex-valued neural network, the multilayer neural network with multi-valued neurons (MLMVN), can be efficiently used for impulse noise filtering. Intelligent image filtering has attracted researches ever since neural networks and fuzzy techniques became commonly and widely used tools in the 1980s. During a long period of time, intelligent filtering was mostly applied by using a feedforward neural network with a single hidden layer and a single output neuron. In such a case, a neural network performs as a low pass filter. Therefore, a feedforward network with a single hidden layer is de-facto a low pass filter. However, this approach was essentially reduced to the intelligent "replication" of classical spatial domain filters. It was based on the processing of a 3x3 or 5x5 local window around a pixel of interest and creation of a single output that is a filtered intensity value in the pixel of interest. This approach was more experimental rather than useful because it was not capable of outperforming traditional spatial domain filters.

Impulse noise filtering is a very specific kind of filtering. Unlike additive noise, which distorts image intensities by adding noisy additive components, impulse noise completely substitute original intensities. It is important to note that while any additive or even multiplicative noise corrupts an entire image, impulse noise corrupts only some of its pixels. A level of this corruption is characterized by the corruption rate – a percentage of pixels corrupted by impulses. For a long period of time, a classical median filter was considered the best tool for impulse noise removal. In fact, this filter is as powerful as it is simple. It can efficiently smooth impulses. However, a great disadvantage is while impulse noise does not corrupt an entire image, median filter is applied to all pixels of an image and thereby smoothing not only impulses, but everything else as well. This results in the heavy loss of image sharpness due to the smoothing of edges, boundaries and all small details. Because of this, there became a very high demand for the design of impulse detectors. A possibility to detect impulses prior to the use of median filter (or any other filter), makes it possible to apply filtering only to those specific pixels, which were marked as noisy, and thereby not affecting pixels which were detected clear. Many various detectors were designed. We can address the reader, for example to [9]-[14].

Intelligent tools were also used for impulse noise filtering because of their ability to solve pattern recognition problems and thus recognize, or detect, noisy pixels. We should mention, for instance approaches presented in [15]-[17].

In the 1970s-early 2000s, most efforts of the image processing community in the impulse noise filtering area were focused on the filtering of highly corrupted images. However, it became clear that while it is possible to remove noise even from heavily corrupted images (50-90% corruption rate), the quality of such filtering is low because an image loses its sharpness and all small details become smoothed and mostly indistinguishable. It is important to mention that a special interest was paid to the case of images corrupted by impulse noise with a low corruption rate. On the one hand, this case is of special interest because an accurate detection of impulses in such images makes it possible to remove noise without smoothing an image, its edges and boundaries. On the other hand, modern equipment used in image acquisition does not produce a heavy impulse noise. This makes the case of lower corruption rate a subject of a special attention. With this regard,

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

it is possible to distinguish, for example filters presented in [13] and [14], which are especially efficient for filtering of noise with a corruption rate 5-20%.

A very powerful BM3D filter was proposed in [18]. Although this filter is extremely efficient (and perhaps even the best) for filtering additive and multiplicative noise, it is not able to filter impulse noise. BM3D filtering is based on the processing in the frequency domain where impulse noise is generally indistinguishable.

However, the invention of BM3D filter inspired new interesting developments in intelligent image filtering. In the series of publications [19]-[21], it was proposed to use a multilayer perceptron (MLP) with a few hidden layers to filter images in the following wat. A noisy image should be broken down in overlapping patches, then each patch should be processed using MLP and a resulting image should be produced by averaging the results over all overlapping patches. To train MLP composed of, as it was mentioned, a few (3-4) hidden layers and an output layer containing the number of neurons equal to the number of pixels in a patch, a learning set should contain noisy patches as inputs and clear patches as desired outputs. The results of this approach were very good, MLP was able to approach and even somewhere outperform BM3D filter for filtering additive Gaussian noise.

In [22], it was suggested to use MLMVN for filtering additive Gaussian noise. In that work, the same approach to process overlapping patches as the one employed in [19]-[21], was used. However, superiority of MLMVN in its learning and generalization capability when compared to MLP made it possible to use a network with a single hidden layer. The results were very attractive. Filtering of additive noise using MLMVN either was a little bit more efficient than using BM3D filter or comparable to it (depending on corresponding images).

In this work, we use the same approach, which was suggested in [22], to filter additive Gaussian noise. We will further develop it and use it to filter random-valued impulse noise. It will be shown that MLMVN is as highly efficient for removal of impulse noise as it is efficient for removal of additive noise. It will also be shown that this approach is especially efficient for images with lower corruption rate. We will see that MLMVN outperforms highly efficient filters, which are based on noise detection. It is possible to preserve edges, boundaries and small details in an image more accurately using the approach which is presented in this paper.

II. Some Keynotes on MLMVN

MLMVN (multilayer neural network with multi-valued neurons (MVN)) is a complex-valued neural network with a classical feedforward organization, but based on multi-valued neurons operating with complex-valued input/output mappings, complex-valued weights and a complex-valued activation function.

After it was introduced in [23], MLMVN and its learning algorithm were comprehensively presented in [24]. Theoretical foundations behind MVN, this neural network, and its derivative-free backpropagation learning algorithm based on the generalization of the classical F. Rozenblat's error-correction learning rule are presented and discussed in detail in [3].

MLMVN has a number of advantages over MLP. It has superior learning and generalization capabilities, as well as the ability to solve challenging applied problems better than MLP and somewhat even more efficiently than, for example, using SVM. These superiorities are confirmed by a number of successful applications. Some of them are, for example prediction of oil production [5], forecasting of the railway equipment reliability [25], restoration of blurred images [26], system identification and fault tolerance design [27], prediction of soil moisture [28], EEG analysis and brain-computer interfaces (interpretation of signals) [7], filtering of additive Gaussian noise from images [22].

All of these advantages and successful applications of MLMVN follow from the properties of MVN. This neuron was initially introduced as "an element of multiple-valued threshold logic" by N.N.Aizenberg *et al*. in 1973 [29]. MVN – multi-valued neuron as a term was suggested in [30]. MVN inputs can be arbitrary complex numbers, but its outputs are located on the unit circle. An activation function of MVN depends only on the argument of its weighted sum. A k-valued discrete activation function of MVN is

$$P(z) = \varepsilon_k^j = e^{i2\pi j/k}, \text{ if } 2\pi j/k \le \arg z < 2\pi (j+1)/k,$$

where $z = w_0 + w_1 x_1 + \ldots + w_n x_n$ is a weighted sum of inputs $x_1, \ldots, x_n$ with the weights $w_0, w_1, \ldots, w_n$, and $\varepsilon_k = e^{i2\pi/k}$.
A continuous activation function of MVN is as follows $P(z) = e^{i\mathrm{Arg}(z)} = z/|z|$.

Hidden neurons in MLMVN always employ a continuous activation function, while output neurons may have either a continuous or discrete activation function, depending on what kind of output should be produced.

In [33], a batch learning algorithm for MLMVN was suggested. This algorithm performs adjustments of the weights for the entire learning set instead of separately for each learning sample where the error occurred. This means that actual outputs shall be found for the entire learning set using current weights, and then the errors shall be calculated. To find adjustments, which shall be added to the weights to correct them, an overdetermined system of linear algebraic equations for these adjustments shall be solved. In [34], this algorithm was further developed for a network with more than one hidden layer and also employing soft margins [35]. In this paper, we will use a batch learning algorithm as it was presented in [34]. This batch learning algorithm makes it possible to speed up a learning process, to use a big network containing thousands of hidden neurons, and to work with learning sets containing tens of thousands of learning samples.

III. Filtering using MLMVN

In this work, we will adapt an approach, which was used earlier for additive noise filtering.

As it was mentioned above, to remove impulse noise efficiently, it is very important to correct intensities only in the pixels corrupted by noise and not touching the ones which are noise-free. However, instead of the use of any noise detection as the first stage of filtering, we would like to detect impulses

indirectly – that is we want a neural network to make a decision whether a certain pixel is noisy or not, and correct its intensity accordingly. At the same time, if a pixel is not noisy, its intensity should not be changed or at least such a change should be negligible. We would like to achieve this in the following way. A neural network should process not a single pixel based on the information from its neighborhood, but simultaneously all pixels from a central m x m area taken from an n x n patch (thus $m < n$). This means that a network shall take an n x n patch as its input and produce an m x m patch as its output (see Fig. 1 where this process is shown).

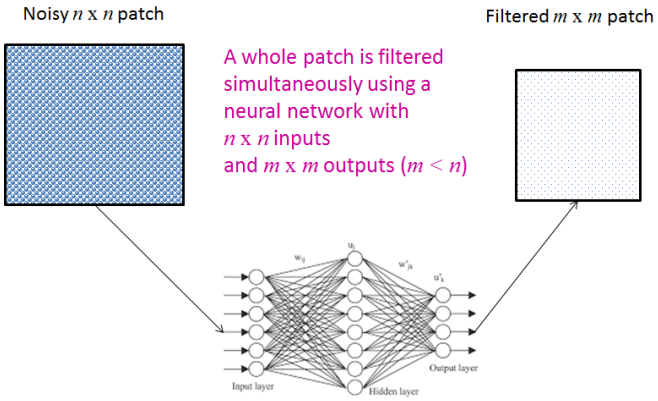

Fig. 1 A process of the n x n patch to m x m patch filtering

To train a network how to filter impulses, while preserving intensities in all pixels not affected by noise, it is necessary to create a robust learning set. A network should be able to learn from this set how to create a noise-free patch from a noisy one, correcting intensities in noisy pixels and preserving them in noise-less pixels. This should be utilized through taking a clear image, corrupting it by impulse noise and then by randomly picking noisy patches from a corrupted image as input learning samples and taking corresponding patches from a clear image as desired outputs. This process is shown in Fig. 2.

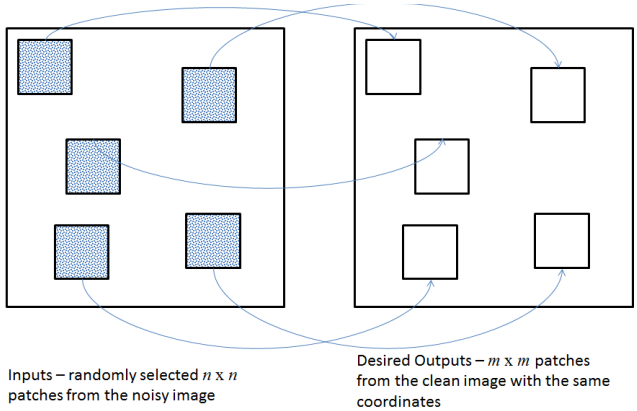

Fig. 2 Learning concept: a network should learn how to transform corrupted patches into clear patches

To create a robust learning set, many images should be taken and input/output patches should be randomly picked from all of them (see Fig. 3).

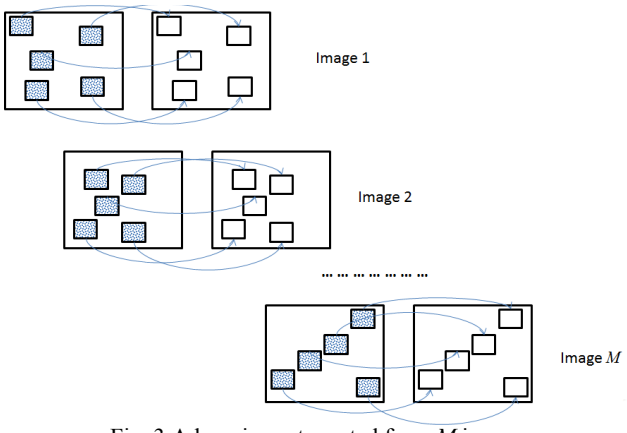

Fig. 3 A learning set created from M images

To ensure that a network is well trained, it is important to use a wide variety of images – various scenes from different areas and containing multiple edges within. The latter is especially important to train a network to differentiate pixels belonging to edges from the ones corrupted by impulses.

A learning process may continue until a certain targeted value of the root mean square error (RMSE) has been reached. This targeted value shall be determined by a desirable goal of reaching a certain value of RMSE or PSNR (peak signal to noise ratio) during filtering.

We employed a batch learning algorithm for MLMVN presented in [33] to train a neural network. This algorithm converges faster than a standard serial algorithm and makes it possible to use a learning set containing thousands and even tens of thousands of learning samples in conjunction with a network containing thousands of hidden neurons.

After a neural network is trained, it should be used to remove noise from images, which did not participate in the learning process. An image to be filtered shall be split into overlapped n x n patches taken with a minimal offset (that is, it should be equal to 1) starting from the origin located in the top-left corner. Every single patch has to be filtered using a neural network. This process is shown in Fig. 4.

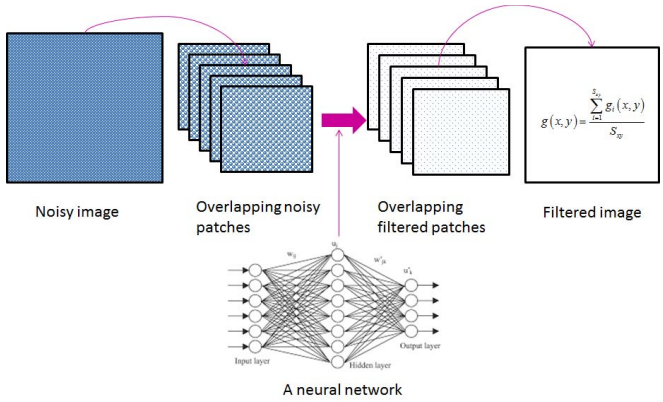

Fig. 4 Learning concept: a network should learn how to transform corrupted patches into clear patches

To correctly process the first and last rows and columns of a noisy image and adjacent areas ($n/2$ rows and columns adjacent to image boundaries accordingly), the image should

be extended by mirroring these areas. After all patches are processed, we will have as many estimations $g_i(x,y)$ for a corrected intensity in the pixel whose coordinates are (x,y) as the number S_{xy} of appearances of this pixel in the corresponding overlapped patches. To find a final estimation $g(x,y)$ for the filtering result in the pixel (x,y), we shall average estimations $g_i(x,y)$ of this output intensity over all overlapping patches ($i = 1,...,S_{xy}$)

$$g(x,y) = \sum_{i=1}^{S_{xy}} g_i(x,y) / S_{xy}.$$

This process is also illustrated in Fig. 4.

Since MLMVN works with complex-valued inputs and creates complex-valued outputs, we need to transform integer input intensities into complex-valued inputs of the network and transform complex-valued outputs of the network to the integer output intensities, accordingly. We used here the same procedure, which was used many times in earlier works. We set $k = 288$ in the discrete MVN activation function and used this value to transform integer intensities into complex-valued ones. The use of $k = 256$ may lead to occasional creation of impulses in a resulting image because "white" (0) and "black" (255) would be "neighbors" on the unit circle. Thus if the actual intensity $j \in \{0,1,...,255\}$ a MLMVN input is $e^{i2\pi j/288}$. Since output neurons of MLMVN employed here have a discrete activation function, each one of them creates an output $e^{i2\pi j/288}$ where $j \in \{0,1,...,288\}$. To be sure that all resulting intensities r do not exceed 255, the following adjustment to them is made if necessary:

$$255 < j < 272 \rightarrow r = 255,$$
$$272 \le j < 288 \rightarrow r = 0.$$

In all experiments presented in this paper, we used MLMVN with a single hidden layer containing H neurons. If n x n is the size of an input patch and m x m is the size of an output patch, then MLMVN has n^2 inputs, H hidden neurons in a single hidden layer and m^2 output neurons creating the same amount of outputs.

IV. Simulation Results

Our goal was to train MLMVN with a single hidden layer to filter impulse noise as described above. After MLMVN was trained we used it to filter impulse noise from 10 images, which did not participate in the training process. The results were compared to the ones obtained using highly efficient filters with noise detectors – differential rank impulse detector [13] and threshold Boolean filter with noise detecting Boolean functions [14]. We compared the results in terms of the filtering quality evaluated based on peak signal to noise ratio (PSNR) and root mean square error (RMSE). We also evaluated quality based on the ability to preserve image edges from smoothing. To test the approach presented above and find the most appropriate input and output patch sizes and H (the number of hidden neurons), we have done a number of experiments.

In our experiments, we used the same image dataset, which was used in [22] for Gaussian additive noise filtering. This dataset contains a number of widely used test images like "Airplane F16", "Cameraman", "Mandrill" and others along with about 100 images from a dataset designed and available in [36] and about 300 images of various scenes from the authors' collection.

To create four learning sets corresponding to four different corruption rates, we used 400 images. They were manually corrupted with random impulse noise using different corruption rates – 5%, 10%, 20%, and 30%. To create each learning set, 100 patches were picked, starting from randomly generated coordinates from each of 400 images, totaling an amount of 40,000 learning samples.

Our test set consisted of 10 images, which we also manually corrupted with random impulse noise using the same four different corruption rates.

MLMVN was trained using each of the four learning sets corresponding to our four corruption rates.

It was found experimentally that the best filtering results can be obtained for $n = 5$ and $m = 3$, that is for the input patch size of 5 x 5 and the output patch size of 3 x 3.

The learning process was controlled by the learning RMSE, which should be lowered to some certain level. The following results were obtained. The learning process for the network 25→2048→9 (25 inputs corresponding to a 5 x 5 input patch, 2048 hidden neurons and 9 output neurons corresponding to a 3 x 3 output patch) converged after 542 iterations with RMSE 5.0584 for the learning set created from images corrupted with 5% noise. The learning process for the network 25→2048→9 converged after 356 iterations with RMSE 6.196 for the learning set created from images corrupted with 10% noise. The learning process for the network 25→3072→9 converged after 212 iterations with RMSE 9.2993 for the learning set created from images corrupted with 20% noise and after 203 iterations with RMSE 10.8455 for the learning set created from images corrupted with 30% noise. The number of hidden neurons was found experimentally and was considered optimal if it was possible to lower a learning RMSE.

Thus, our learning sessions resulted in the four sets of weights corresponding to the four corruption rates. We filtered each 5%, 10%, 20%, and 30% corrupted test images with these four sets of weights resulted from the learning process for all four sets of images. So, images with a 5% corruption rate were filtered with weights determined during 5% corruption filtering, 10% corruption filtering, 20% corruption filtering, and 30% corruption filtering. Same process was used for 10% and 20% corrupted images. The results from this process are presented in Table I, Table II and Table III. It follows from these data that filtering of images with a 5% corruption rate returned the highest PSNR when they were filtered with the weights obtained from the 10% corruption learning set. Filtering of images corrupted with 10% and 20%

corrupted rates returned the highest PSNR when filtered with the 20% corruption weights.

TABLE I
SIMULATION RESULTS: SUMMARY OF LEARNING AND FILTERING COMPARING MLMVN WITH NOISE DETECTING FILTERS FOR 5% CORRUPTED IMAGES

TEST IMAGES		1	2	3	4	5	6	7	8	9	10	AVERAGE
MLMVN (WEIGHTS OBTAINED FROM 10% CORRUPTED IMAGES)	PSNR	**33.90**	**30.96**	**29.87**	**30.33**	27.04	**28.06**	**35.07**	**33.03**	**34.38**	**35.94**	**31.86**
	RMSE	**5.15**	**7.22**	**8.18**	**7.76**	11.34	**10.08**	**4.50**	**5.69**	**4.87**	**4.07**	**6.89**
THRESHOLD BOOLEAN FILTERING WITH NOISE DETECTING BOOLEAN FUNCTIONS [14]	PSNR	32.71	29.04	27.75	26.81	**28.85**	28.03	33.19	30.21	32.66	32.06	30.13
	RMSE	5.90	9.01	10.45	11.64	**9.20**	10.11	5.58	7.87	5.93	6.36	8.21
FILTER WITH DIFFERENTIAL RANK IMPULSE DETECTOR [13]	PSNR	33.42	29.08	27.58	28.19	28.31	27.98	33.58	30.32	33.31	33.98	30.58
	RMSE	5.44	8.96	10.66	9.93	9.79	10.17	5.34	7.77	5.51	5.10	7.87

TABLE II
SIMULATION RESULTS: SUMMARY OF LEARNING AND FILTERING COMPARING MLMVN WITH NOISE DETECTING FILTERS FOR 10% CORRUPTED IMAGES

TEST IMAGES		1	2	3	4	5	6	7	8	9	10	AVERAGE
MLMVN (WEIGHTS OBTAINED FROM 10% CORRUPTED IMAGES)	PSNR	**31.72**	**28.33**	**27.25**	**27.66**	24.34	25.87	**32.64**	**30.50**	**32.21**	**33.80**	**29.43**
	ST. DIV.	**6.62**	**9.77**	**11.06**	**10.56**	15.48	12.98	**5.95**	**7.61**	**6.25**	**5.21**	**9.15**
THRESHOLD BOOLEAN FILTERING WITH NOISE DETECTING BOOLEAN FUNCTIONS [14]	PSNR	30.74	27.11	25.89	24.91	26.87	26.05	31.64	28.50	30.78	30.08	28.26
	ST. DIV.	7.40	11.25	12.94	14.48	11.56	12.70	6.67	9.58	7.37	7.99	10.20
FILTER WITH DIFFERENTIAL RANK IMPULSE DETECTOR [13]	PSNR	**32.31**	27.61	26.23	26.04	**27.06**	**26.60**	32.63	29.17	32.15	32.03	29.18
	ST. DIV.	**6.17**	10.62	12.44	12.72	**11.31**	**11.92**	5.96	8.87	6.29	6.38	9.27

TABLE III
SIMULATION RESULTS: SUMMARY OF LEARNING AND FILTERING COMPARING MLMVN WITH NOISE DETECTING FILTERS FOR 20% CORRUPTED IMAGES

TEST IMAGES		1	2	3	4	5	6	7	8	9	10	AVERAGE
MLMVN (WEIGHTS OBTAINED FROM 20% CORRUPTED IMAGES)	PSNR	27.76	26.03	24.58	**25.13**	21.47	23.16	30.09	**28.21**	29.30	**30.56**	26.63
	ST. DIV.	10.43	12.74	15.04	**14.12**	21.54	17.73	7.98	**9.91**	8.74	**7.56**	12.58
MLMVN (WEIGHTS OBTAINED FROM 30% CORRUPTED IMAGES)	PSNR	29.20	**26.19**	24.81	25.00	21.90	23.67	30.46	**28.21**	29.67	**31.06**	27.02
	ST. DIV.	8.84	**12.49**	14.64	14.33	20.47	16.70	7.64	**9.91**	8.36	**7.13**	12.05
THRESHOLD BOOLEAN FILTERING WITH NOISE DETECTING BOOLEAN FUNCTIONS [14]	PSNR	28.58	25.34	24.16	23.46	25.07	24.20	30.21	27.03	28.65	28.23	26.49
	ST. DIV.	6.28	7.60	9.19	7.11	8.28	8.83	4.77	5.74	3.95	4.21	12.47
FILTER WITH DIFFERENTIAL RANK IMPULSE DETECTOR [13]	PSNR	**30.25**	26.11	**24.91**	**24.38**	**25.69**	**25.07**	**31.04**	**27.82**	**30.12**	**29.78**	**27.52**
	ST. DIV.	**7.84**	12.62	**14.48**	**15.41**	**13.25**	**14.22**	**7.16**	**10.36**	**7.95**	**8.27**	**11.16**

Filtering of images corrupted by 30% noise was performed only using the weights corresponding to this corruption rate.

These results were compared to the ones obtained using the filters with detectors presented in [13] and [14]. All results are summarized in Table I - Table III.

MLMVN trained with a learning set created from images corrupted by 5% noise slightly yields to filters [13] and [14] when it used for filtering of noise from test images. However, MLMVN trained with a learning set created from images corrupted by 10% noise outperforms both filters [13] and [14] in terms of PSNR and RMSE for filtering of noise from images corrupted with 5% noise and 10% noise. It is also very important that MLMVN better preserves edges in images. A disadvantage of noise detectors is a high level of false positive detection of noise when pixels belonging to sharp edges often can be misclassified as noisy. MLMVN preserves edges and small details much better than filters with noise detectors. This is demonstrated by examples shown in Fig. 5-16. Particularly we would like to draw the reader's attention to Fig. 9-10 and Fig. 15-16 showing enhanced differences between an original image and filtered ones. When there is a stark contrast - as discussed with the cockpit (Fig. 5-6) where there is dark writing on top of a much lighter base, the network determines and corrects this more accurately and precisely than the detecting filters. The same is clearly visible from Fig. 15-16 showing enhanced differences between another original image and filtered images. An original image (Fig. 11) contains many sharp edges and small details. They are significantly better preserved by MLMVN than by noise detecting filters.

MLMVN also shows results comparable to the ones by edge detecting filters for noise with a 20% corruption rate. However, MLMVN yields to noise detecting filters (when the latter ones are applied iteratively) when it is dealing with a 30% corruption rate. A very attractive idea for the future work is to change the learning strategy and train a network not to minimize its learning error, but to minimize its testing error. This will be a good subject for our future work and hopefully this will make it possible to improve MLMVN performance for higher corruption rates.

V. Discussion and Conclusions

The main result obtained in this paper – MLMVN, a complex-valued neural network can successfully be used to filter impulse noise from images. MLMVN is especially efficient for filtering of noise with pretty low corruption rate of 5%-10%. This case is more interesting from the practical point of view because traditional sophisticated noise detectors – being efficient for higher corruption rates, have a very high level of false positive detections of sharp edges in images corrupted by noise with a lower corruption rate. As a result, edges and small details appear smoothed in filtered images. Intelligent filtering using MLMVN makes it possible to preserve edges and small details in such images much more accurately.

All software simulations were performed in Matlab. The corresponding Matlab functions along with the data used for simulations in this paper can be found at https://www.freewebs.com/igora/Downloads.htm right after publication of this paper.

VI. Acknowledgement

The authors appreciate a possibility to use computer facilities at the Kakos Center for Scientific Computing of the School of Science at Manhattan College to perform all simulations presented in this paper.

References

[1] A. Hirose, *Complex-Valued Neural Networks, 2nd Edn.*, Springer, Berlin, Heidelberg, 2012.

[2] T. Nitta, *Complex Valued Neural Networks: Utilizing High-Dimensional Parameters*, Hershey, New York, IGI Global, 2009.

[3] I. Aizenberg, I., *Complex-Valued Neural Networks with Multi-Valued Neurons*. Berlin: Springer-Verlag Publishers, 2011.

[4] S. L. Goh, M. Chen, D. H. Popovic, K. Aihara, D. Obradovic and D. P. Mandic, "Complex Valued Forecasting of Wind Profile," *Renewable Energy*, vol. 31 , Sep. 2006, pp. 1733-1750.

[5] Aizenberg I., Sheremetov L., Villa-Vargas L., and Martinez-Muñoz J., "Multilayer Neural Network with Multi-Valued Neurons in Time Series Forecasting of Oil Production", *Neurcomputing*, vol. 175, part B, pp. 980-989, Jan. 2016.

[6] Y.Nakano, A.Hirose, "Improvement of Plastic Landmine Visualization Performance by use of ring-CSOM and Frequency-Domain Local Correlation," *IEICE Transactions on Electronics*, vol. E92-C, Issue 1, Jan. 2009, pp. 102-108.

[7] N.V.Manyakov, I. Aizenberg, N. Chumerin, and M. Van Hulle, "Phase-Coded Brain-Computer Interface Based on MLMVN", book chapter in *Complex-Valued Neural Networks: Advances and Applications* (A. Hirose – Ed.), Wiley, 2012, pp. 185-208.

[8] A. Handayani, A.B.Suksmono, T.L.R.Mengko, and A.Hirose, "Blood Vessel Segmentation in Complex-Valued Magnetic Resonance Images with Snake Active Contour Model," *International Journal of E-Health and Medical Communications*, vol. 1, issue 1, Jan. 2010, pp. 41-52.

[9] H.-L. Eng and K.-K. Ma, "Noise adaptive soft-switching median filter," *IEEE Transactions on Image Processing*, vol. 10, pp. 242-251, 2001.

[10] K.Kondo, M.Haseyama and H.Kitajima "An Accurate Noise Detector for Image Restoration", *Proc. of 2002 IEEE International Conference on Image Processing*, vol. 1, pp. 321-324, 2002.

[11] L. Khriji and M. Gabbouj, "Median Rational Hybrid Filters," *IEEE International Conference on Image Processing,* ICIP'98, pp.853-857, Chicago, Illinois, USA, October 4-7, 1998.

[12] S. Zhang and M. A. Karim, "A New Impulse Detector for Switching Median Filters," *IEEE Signal Processing Letters*, vol. 9, no. 11, pp. 360-363, Nov 2002.

[13] I. Aizenberg and C. Butakoff, "Effective Impulse Detectors Based on Rank-Order Criteria", *IEEE Signal Processing Letters*, Vol. 11, No 3, March, 2004, pp. 363-366.

[14] I. Aizenberg, C. Butakoff C. and D. Paliy, "Impulsive Noise Removal using Threshold Boolean Filtering based on the Impulse Detecting Functions", *IEEE Signal Processing Letters*, Vol.12, No 1, January 2005, pp. 63- 66.

[15] Chang-Shing Lee, Chin-Yuan Hsu, and Yau-Hwang Kuo, "Intelligent Fuzzy Image Filter for Impulse Noise Removal," *Proceedings of the 2002 IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 431-436, 2002.

[16] M. E. Yüksel, "A Hybrid Neuro-Fuzzy Filter for Edge Preserving Restoration of Images Corrupted by Impulse Noise", *IEEE Transactions on Image Processing*, Vol. 15, No 4, April 2006, pp. 928-936.

[17] Sheng-Fu Liang, Shih-Mao Lu, Jyh-Yeong Chang, and Chin-Teng (CT) Lin, "A Novel Two-Stage Impulse Noise Removal Technique Based on Neural Networks and Fuzzy Decision" *IEEE Transactions on Fuzzy Systems*, Vol. 16, No 4, August 2008, pp. 863-873.

[18] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image Denoising by sparse 3-D Transform-Domain Collaborative Filtering". *IEEE Transactions on Image Processing*, vol. 16, no 8, pp. 2080–2095, 2007.

[19] H. C. Burger, C. J. Schuler, and S. Harmeling, “Image denoising: Can plain Neural Networks compete with BM3D?”, *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR-2012), pp. 4321-4328, 2012.

[20] H. C. Burger, C. J. Schuler, and S. Harmeling, “Image Denoising with Multi-Layer Perceptrons, Part 1: Comparison with Existing Algorithms and with Bounds”, available online at http://arxiv.org/pdf/1211.1544.pdf, 2012.

[21] H. C. Burger, C. J. Schuler, and S. Harmeling, “Image Denoising with Multi-Layer Perceptrons, Part 2: Training Trade-offs and Analysis of their Mechanisms, available online at http://arxiv.org/pdf/1211.1552.pdf, 2012.

[22] I. Aizenberg, A. Ordukhanov, and F. O’Boy, “MLMVN as an Intelligent Image Filter”, *Proceedings of the 2017 IEEE International Joint Conference on Neural Networks* (IJCNN 2017), Anchorage, May, 2017, pp. 3106-3113.

[23] I. Aizenberg, C. Moraga, and D. Paliy, "A Feedforward Neural Network based on Multi-Valued Neurons", In *Computational Intelligence, Theory and Applications. Advances in Soft Computing* , XIV, (B. Reusch - Ed.), Springer, Berlin, Heidelberg, New York, 2005, pp. 599-612.

[24] I. Aizenberg, and C. Moraga, “Multilayer Feedforward Neural Network based on Multi-Valued Neurons (MLMVN) and a backpropagation learning algorithm,” *Soft Computing*, 11, issue 2, , Jan. 2007, pp. 169-183.

[25] O. Fink, E. Zio, U. Weidmann, “Predicting Component Reliability and Level of Degradation with Complex-Valued Neural Networks”, *Reliability Engineering & System Safety*, vol. 121, 2014, pp. 198–206.

[26] I. Aizenberg, D. Paliy, J. Zurada, and J. Astola, "Blur Identification by Multilayer Neural Network based on Multi-Valued Neurons", *IEEE Transactions on Neural Networks*, vol. 19, No 5, May 2008, pp. 883-898.

[27] I. Aizenberg, A. Luchetta, S. Manetti., and C. Piccirilli., “System Identification using FRA and a modified MLMVN with Arbitrary Complex-Valued Inputs”, *Proceedings of the 2016 IEEE International Joint Conference on Neural Networks* (IJCNN 2016), Vancouver, July, 2016, pp. 4404-4411.

[28] R. Ji, S. Zhang, L. Zheng, Q. Liu, “Prediction of Soil Moisture based on Multilayer Neural Network with Multi-Valued Neurons”, *Transactions of the Chinese Society of Agricultural Engineering*, Vol. 33, Supl. 1, March 2017, pp. 126-131.

[29] N. N. Aizenberg, Yu. L. Ivaskiv, D. A. Pospelov, and G.F. Hudiakov, "Multivalued Threshold Functions. Synthesis of Multivalued Threshold Elements", Cybernetics and Systems Analysis, vol. 9, No 1, January 1973, pp. 61-77.

[30] N.N. Aizenberg and I.N. Aizenberg, "CNN Based on Multi-Valued Neuron as a Model of Associative Memory for Gray-Scale Images", *Proceedings of the Second IEEE International Workshop on Cellular Neural Networks and their Applications,* Munich, October 14-16, 1992, pp.36-41.

[31] I. Aizenberg, A. Luchetta and S. Manetti, S, “A modified learning algorithm for the multilayer neural network with multi-valued neurons based on the complex QR decomposition,” *Soft Computing*, vol. 16, issue 4, 563-575, Apr. 2012.

[32] I. Aizenberg, “MLMVN with Soft Margins Learning”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, No 9, September 2014, pp. 1632-1644.

[33] I. Aizenberg, A. Luchetta, and S. Manetti, “A modified Learning Algorithms for the Multilayer Neural Network with Multi-Valued Neurons based on the Complex QR Decomposition”, *Soft Computing*, vol. 16, April 2012, pp. 563-575.

[34] E. Aizenberg and I. Aizenberg, “Batch LLS-based Learning Algorithm for MLMVN with Soft Margins”, *Proceedings of the 2014 IEEE Symposium Series of Computational Intelligence (SSCI-2014)*, December, 2014, pp. 48-55.

[35] I. Aizenberg, “MLMVN with Soft Margins Learning”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, No 9, September 2014, pp. 1632-1644.

[36] Test Images. University of Granada Image Data Base. Available Online http://decsai.ugr.es/cvg/dbimagenes/

Fig. 5. The original image “Airplane F-16” (Image 10 in Tables I-III - was not used in the learning sets)

Fig. 6. The “Airplane F-16” image from Fig. 5 corrupted by additive impulse noise with 5% corruption rate.

Fig. 7. The image from Fig. 6 filtered with MLMVN 25→2048→9 trained using a learning set created from 10% corrupted images. PSNR=35.94

Fig. 8. The image from Fig. 6 filtered using TBF [14]; PSNR=32.06.

Fig. 9. The enhanced difference between the original image (Fig. 5) and the one filtered with MLMVN (Fig. 7) Image edges are well preserved.

Fig. 10 The enhanced difference between the original image (Fig. 5) and the one filtered with TBF (Fig. 8). Many edges are smoothed because their pixels were misclassified by TBF as noisy

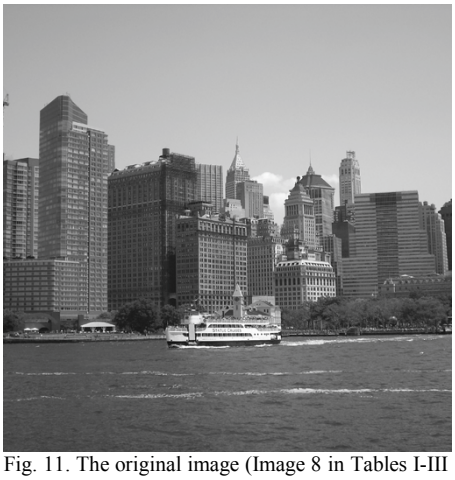

Fig. 11. The original image (Image 8 in Tables I-III - was not used in the learning sets)

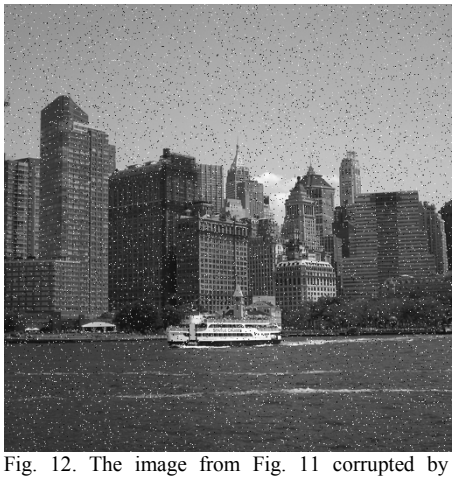

Fig. 12. The image from Fig. 11 corrupted by additive impulse noise with 5% corruption rate

Fig. 13. The image from Fig. 12 filtered with MLMVN 25→2048→9 trained using a learning set created from 10% corrupted images. PSNR=33.03

Fig. 14 The image from Fig. 12 filtered using TBF [14]; PSNR=32.06.

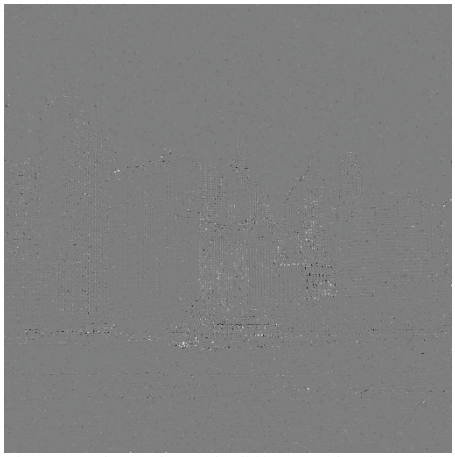

Fig 15. The enhanced difference between the original image (Fig. 11) and the one filtered with MLMVN (Fig. 13) Image edges and small details are well preserved.

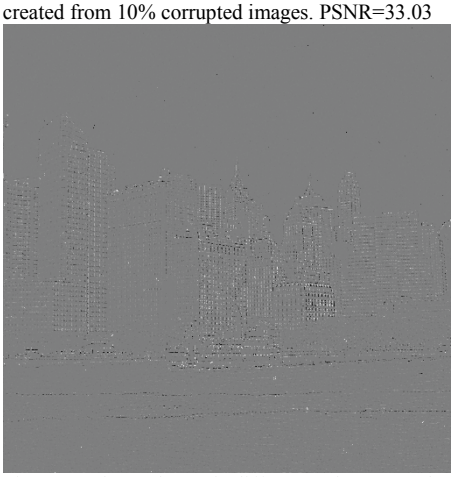

Fig 16. The enhanced difference between the original image (Fig. 11) and the one filtered with TBF (Fig. 14). Many edges and small details were smoothed because pixels belonging to them were misclassified by TBF as noisy