Impulse noise threshold detection and Noise removal using Non-local Grid Network

Myung-Jae Lim[1], Young-Chae Kim[2], Jeong-Eun Kim[3], Kyu-Ho Kim[4], Dong-Keun Chung[5*]

[1,2,3,4,5]*Department of Medical IT, Eulji University, SeongNam city, Korea. E-mail: lk04@eulji.ac.kr*
Corresponding Author: Dong-Keun Chung

Abstracts: The method of removing the existing Impulse noise was to find the threshold and apply the median filter to remove the noise. In this paper, we try to find reference values in neural networks. In order to find noise, we used a non-local method to consider not only the information of the region but also the entire region. We then utilized Yolo's grid cell structure to make stronger predictions considering various sizes and different areas. Accordingly, qualitative and quantitative evaluation results show better performance than conventional median filters.

Keywords: Impulse noise, Median filter, Non-local Filter, Grid cell

1. INTRODUCTION

In the era of the 4th Industrial Revolution, the field of image processing systems is drawing attention. Image processing is to improve picture information for human interpretation or to change the nature of the image to better express it for automation machine recognition.

In the process of transmitting an image, the deterioration of the image signal due to external disturbance is called noise, but the image contaminated with this noise needs to be restored. Image restoration focuses on eliminating or reducing degradation that occurs in the process of acquiring the image [1].

Types of image noise include Impulse noise, Gaussian noise, Speckle noise, and Periodic noise [1]. Among them, Impulse Noise may be caused by sudden disturbance in an image signal. In addition, the appearance of this noise appears irregularly spread with white and black pixels throughout the image.

Methods for restoring images with this impulse noise include low pass filtering, Median filtering, and Rank-order filtering. Median filtering, which finds a medium-sized value among these methods, is effective in removing Impulse noise.

This paper tries to implement neural networks through non-local neural networks and YOLO grid cells proposed to supplement the structure where only local areas of existing CNNs can be seen after adding noise to remove Impulse noise added to images. In conclusion, we want to find the reference value through deep learning using PSNR, an indicator that restores the image and compares the quality of the image by applying a media filter to the pixels of the image with noise added.

2. RELATED WORK

2.1. Non-local Neural Networks

Non-local Neural Networks present a non-local block to complement the structure in which existing CNN can only see local areas.

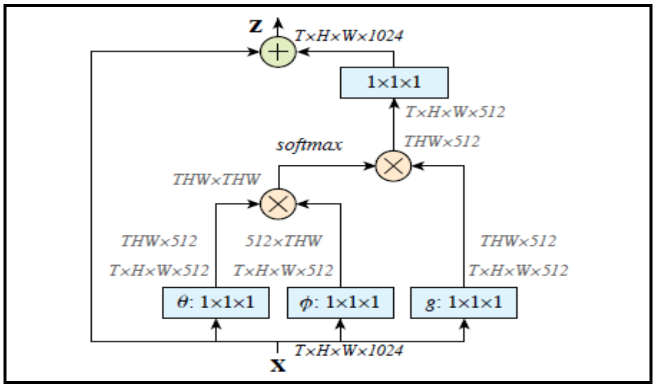

Figure 1. Non-local block Structure.

Figure 1 shows that the non-local block extracts important information in the input feature using 1X1X1 filter and finally derives Output with a size such as Input size (T x H x W x 1024) [2].

Non-local Neural Networks can be learned in consideration of the entire area by utilizing a variety of 1x1 convolutional filters rather than local areas in the data. Also, since the Input size and Output size are the same, they can be easily applied to existing CNN network structures and have a structure similar to Self-Attention, the prediction results can be interpreted.

In this study, we intend to use information of various areas and sizes using Non-local Filter. In order to use various information with robustness to noise, the structure of Non-local Filter was used by dividing the image into grids of various sizes.

2.2. YOLO

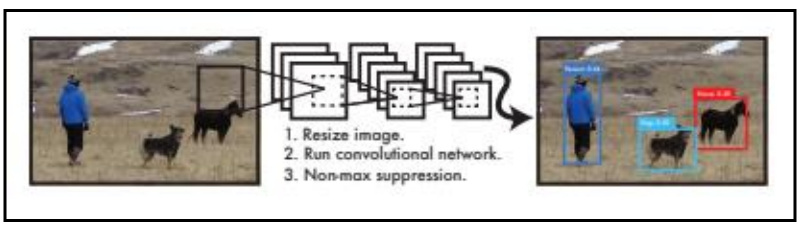

Figure 2. YOLO Detection System.

Figure 2 shows that the YOLO model is designed in a single CNN structure, where convolutional layers extract features from images, and fully connected layers predict class probabilities and coordinates of bounding boxes [3].

YOLO divides the input image by S x S grid, and each grid cell predicts the confidence score for B bounding boxes and their bounding boxes. When predicting, YOLO looks at the entire image and learns and processes not only the information about the shape of the class but also the surrounding information.

In this study, YOLO's grid cell was used to make it possible to utilize patterns in large areas and patterns in small areas with robustness to noise.

2.3. Median Filter

Median filter is a filter that sorts the values of surrounding pixels in order of magnitude in an input image and replaces the pixel value with the value in the center [4]. If the values of the surrounding pixels are even, the Median value is a value obtained by averaging the two values in the middle. This Median filter is known to be suitable for effectively removing Impulse noise. When calculating the Median value, the Median operation usually replaces the noise value with the value closest to it, because values that are very large or very small are located at the beginning or end of the sorting list.

In this study, 5X5 Median filter is applied to pixels where Impulse noise occurs to operate firmly even when there is a lot of noise, maintain the characteristics of the image, and denoise the image.

2.4. PSNR

The peak signal-to-noise ratio (PSNR) is the maximum signal-to-noise ratio, which represents the power of noise to the maximum power a signal can have, and is mainly used to evaluate image quality loss information in video or video loss compression.

$$\begin{aligned} PSNR &= 10\log_{10}\left(\frac{MAX_I^{\,2}}{MSE}\right) \\ &= 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20\log_{10}(MAX_I) - 10\log_{10}(MSE) \end{aligned}$$

Figure 3. PSNR formula.

Figure 3 shows the PSNR equation, where MAX is the maximum value of the signal, and for 8-bit gray scale images, MAX becomes 255 [5]. As shown in the formula, PSNR is inversely proportional to the MSE.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,\ j)-K(i,\ j)]^2$$

Figure 4. MSE formula.

Figure 4 shows the Mean Squared Error (MSE) as the mean square error, which is the mean of the difference between the algorithm's predicted value and the actual correct answer [5].

In this work, we intend to use PSNR for performance comparison with the proposed model [6].

3. PROPOSED NETWORK

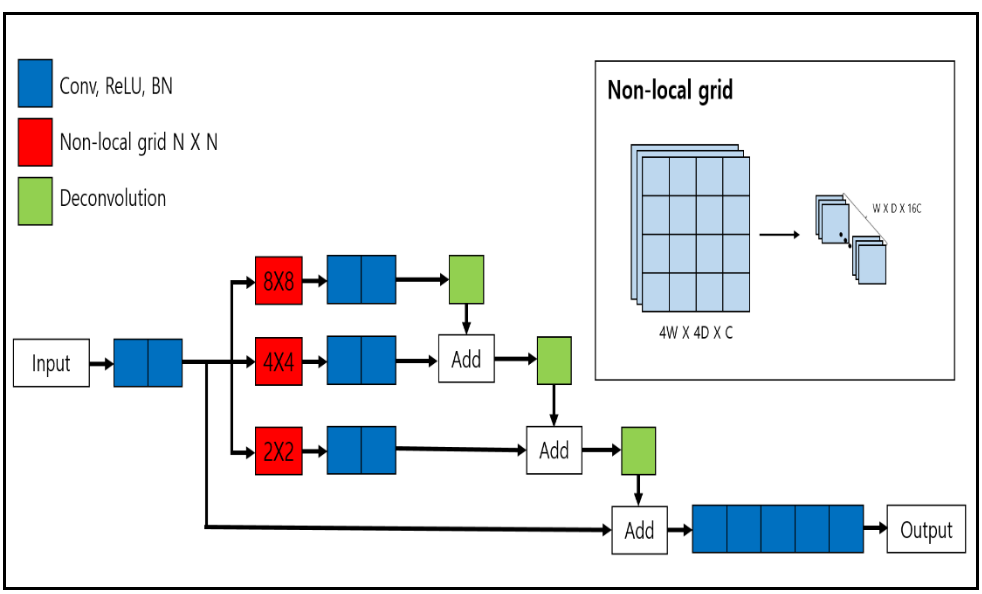

Figure 5. Proposed Network.

Figure 5 shows the proposed neural network structure, with a structure that divides the grid and a Residual structure that adds the calculated one.

Feature maps are extracted using Convolution, Batch Normalization (BN), and Corrected Linear Unit (ReLU) active functions in the Input image. The extracted feature map is divided into grids of sizes 8X8, 4X4, and 2X2, and then the size is small and the number of feature maps is increased. After two convolution, BN, and ReLU operations for each grid-sized feature map, increase the size through Deconvolution and add it to the feature map with a larger grid. For the last 2X2 grid, add the grid to the feature map before dividing it. To obtain the output for the noise position, set it to be the same size as the input value and one channel.

Normalization is performed by changing the pixel value of the output image to 1 if there is noise and 0 if there is no noise. Then, apply the Median filter to the location where noise is expected.

4. EXPERIMENT

Figure 6 shows the Cameraman image, the original from the top left to the bottom, and the Impulse noise is applied at 10%, 30%, and 50%, respectively. In the experiment, these noise levels are learned and tested.

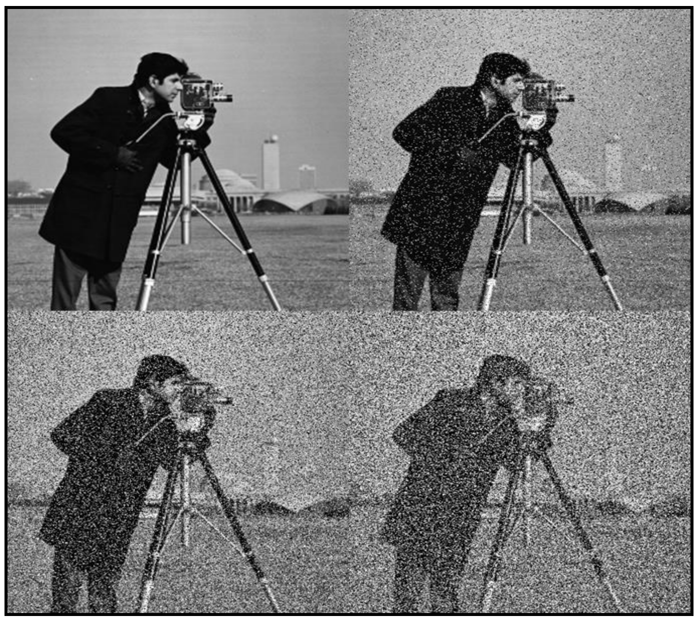

Figure 6. Original and Impulse noise 10%, 30%, 50% applied Image.

We set the detailed conditions of the experiment as follows. First, BSD 200 is used for the data to be learned, and BSD 100 is used for the test. At the time of learning, the patch size was 8, the epoch was 4,000 times, the loss function was Mean Square Average (MSA), and the learning method was Adam optimizer. In order to perform qualitative and quantitative evaluation, one image was visualized and the PSNR value was measured using BSD100 data.

Figure 7. Original and Denoising 10%, 30%, 50% applied Image.

Figure 7 shows the Cameraman image, which appears when Impulse noise denoises 10%, 30%, and 50%, respectively, compared to the original image from the top left to the bottom left.

Table 1 shows the results of Median Filter (MF) and Proposed using PSNR, a quality evaluation index.

Table 1. Test PSNR values for BSD100 data.

Noise density	MF	Proposed
10	27.36	32.54
30	26.43	31.42
50	22.54	28.23

5. CONCLUSION

This paper proposed a non-local grid network to consider various sizes and various areas, and through this, the location of impulse noise was found and the damaged image was denoised through the Median filter.

PSNR and data were visualized to evaluate the method. As a result, we show that Impulse noise had better PSNR values than Median Filter for 10%, 30%, and 50% and that the texture of the image is better represented when visualized.

When noise was misjudged during the experiment, it had a worse effect than the existing method. Therefore, in future studies, we intend to propose a loss function suitable for this to prevent erroneous discrimination.

REFERENCES

[1] Alasdair McAndrew, A Computational Introduction to Digital Image Processing, 2/E, Taylor & Francis, pp 193-194, 2015.

[2] X. Wang, R. Girshick, A. Gupta and K. He, "Non-local Neural Networks", 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7794-7803, 2018.

[3] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 779-788, 2016.

[4] Hwang Sun-kyu, 「Visual C++ Image Processing Programming」, gilbut, pp. 383-386, 2015.

[5] Fardo, F. A., Conforto, V. H., de Oliveira, F. C., & Rodrigues, P. S., "A Formal Evaluation of PSNR as Quality Measurement Parameter for

Image Segmentation Algorithms", arXiv preprint arXiv:1605.07116, pp 3-4, 2016.

[6] R. Kunsoth and M. Biswas, "Modified decision based median filter for impulse noise removal," 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 1316-1319, 2016.

DOI: https://doi.org/10.15379/ijmst.v10i1.1450