

ENHANCED NON-LINEAR FILTERING SCHEME FOR EDGE DETECTION

HAIDER O. LAWEND*, ANUAR M. MUAD, A. HUSSAIN

Center for Integrated Systems Engineering and Advanced Technologies (INTEGRA),
Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia,
43600 UKM Bangi, Selangor, Malaysia

*Corresponding Author: haider_lawend@siswa.ukm.edu.my

Abstract

Edge detection plays an important role in image processing. Edge detectors have always been a compromising between information and noise. Since edge detection is a derivative operation, it tends to amplify noise. This means that increasing the amount of information may increase the noise as well. There are a variety of edge detectors or operators with different size of the kernel. In general, many established edge detectors focus on the gradient in grayscale image to detect edges. This paper proposed an improvement of edge detection algorithm by considering two edge features: gradient and length. In the proposed algorithm, the threshold value of the gradient was set to a value similar to a default value used in other existing edge detectors. The length of the edges feature was used to increase the robustness of the proposed algorithm towards the noise. The proposed algorithm was validated with synthetic and natural images with the inclusion of three types of noise: additive, multiplicative and impulsive noises. Results were compared with other established edge detectors whereby the proposed algorithm demonstrated its superiority in handling edges in low contrast regions and less sensitive towards the noise.

Keywords: Edge detection, Features extraction, Image processing, Non-linear filtering scheme, Noise reduction.

1. Introduction

Edge detection is one of the important and widely used low-level image processing technique and it is substantial for subsequent intermediate and high-level processes like segmentation, recognition, classification, and other stages in machine learning applications [1-4].

In edge detection, using operators with a kernel size of 2×2 pixels or 3×3 pixels like Robert, Sobel and Prewitt succeed in edge detection but they cannot produce good quality when applied to noisy images [5]. Using a larger kernel with the size of 5×5 pixels or more, such as Rotating Kernel Transformation (RKT) [6], is high resistance to noise. However, the larger the kernel is used, the slower and poorer it becomes in detecting edges especially low contrast and small edges like corners.

The operator of Canny Edge Detector (CED) [7, 8] reduces the noise by smoothing the image with a Gaussian filter, applies non-maxima suppression and uses hysteresis thresholding to detect the edges. Estimated Ground Truth edge detection algorithm (EGT) uses multiscale analysis of Canny to detect edges [9]. EGT is more robust to noise compare to the CED. However, EGT was criticized for being too slow and highly dependent on the initial set of values, which make it impractical [10]. Scale Multiplication of Canny (SMC) [11], is a robust algorithm towards noise but weak at detecting low contrast edges. Recently, the work of [12] proposed Snakelet edge detection (SED) which is an approach to connect the discontinuous edges applying snakelet. Their method was to track the edges first to initialize the snake. After that, the snake grows based on the internal and external energy. The internal energy is the length of the edge or the length of the snake itself. The external energy is the gradient of the image. SED was able to connect the edges and good at detecting the boundary of the object in less detailed images compared with CED. However, with the inclusion of noise in the image, or when the shapes of the edges tend to be irregular like a tree, the snake might also detect the noises and connect them to the real edges. This may reduce the accuracy of the edge detection process.

Edge detection algorithms based on morphological analysis were presented in [13-16]. Morphological edge detection algorithm depends highly on the shape and the direction of the edges, which may lead to non-continuous edges. It may also shift the location of the edges to a certain area especially when dilation and erosion operations were overly used, which reduce the edge localization accuracy.

Apart from the direct kernel convolution approach, other researchers use non-linear operators for edge detection. The work of [17] was a comparison of using different approaches like linear and non-linear operators in edge detection. Their work showed the advantages of using the non-linear approach as pre-filtering to smooth the image and prepare it for further image processing like edge detection. The researchers showed that using non-linear pre-filtering, in general, led to better results compared with linear filtering in noisy images. The researchers in [18] proposed a Non-Linear Filtering Scheme edge detector (NLFS) which uses non-linear operator and capable of suppressing noises. The image was smoothed using a Gaussian filter, then non-linear filter with local maxima suppression was applied to detect edges. The advantage of NLFS is in resisting noise. NLFS tends to shift the location of the edges towards the light area, which may help in detecting, estimating and tracking the edges in multiple scale image analysis.

The non-linear filter NLFS was also used to estimate the noise in the work of [19, 20]. The work of [21] showed the advantages of NLFS compared with CED. The researchers showed that NLFS tended to generate clearer and less noisy edges compared with CED. The work of [22] was another non-linear filtering scheme. In their work, Non-linear Noise Suppression Edge Detection (NNSSED) approach was proposed and compared with NLFS. NNSSED suppressed much noise compared with NLFS. NNSSED considered the image pixel to be an edge pixel if the slopes of the non-linear filters are the same (positive or negative).

Machine learning algorithms like structure forest and genetic algorithm are used to detect edges like in the works of [23-25]. Compared to the other edge detectors, machine-learning algorithms require samples to be trained before edges can be detected. Therefore, the accuracy of this type of algorithms depends on the quality and quantity of training samples. This type of algorithms is not preferred to detect edges because of its complexity, which makes it time-consuming. The work of [26] used machine learning to classify image based on the image features like the length and intensity. The work of [5] used an unsupervised backtracking search algorithm, which is a type of optimization algorithm. The work of [27] combined two learning algorithms like k-means clustering and genetic algorithms. However, using two iterative learning algorithms make the edge detector slow and complex. Despite the advantages of using a machine-learning algorithm in edge detection, this type of edge detectors is not preferred by the researchers. In the work of [12], the researchers criticized machine learning edge detectors of being slow. They also addressed another limitation, which is the difficulties to provide the samples for learning.

From the literature, researchers used many thresholding techniques to distinguish between true and false edges. For example, CED and SMC use hysteresis thresholding technique with two thresholds. In EGT, the researchers used Receiver Operating Characteristics (ROC) thresholding technique. OTSU adaptive thresholding is used in [28] to automatically determine the threshold of Canny edge detector. These thresholding techniques were not the best solution to deal with noise. Selecting the appropriate threshold may reduce the noise and give a better edge image. However, these techniques may not work with speckle and salt and pepper noises. This is because these types of noise create very large contrast changes in the image, which cannot be removed by just selecting the appropriate thresholds. Using their techniques, speckle noise in bright areas and salt and pepper noise may appear as high contrast edges.

In general, many edge detectors can detect not only edges but amplify noises as well. In this paper, we proposed a new edge detection algorithm based on NLFS. This is because NLFS in the work of [18] was highly resistant to noise. The aim of this algorithm is to enhance NLFS toward noise (Gaussian, speckle and salt and pepper) and to reduce the trade-off between information and noise. This is done by tracking the edges of NLFS, measuring their lengths and removing the very short edges. This is because NLFS tends to create very short edges in a noisy part of the image. The algorithm does not affect the non-noisy parts of the image, which allows it to detect low contrast edges. The rest of this paper is organized as follow; NLFS and the proposed algorithm are presented in details in Section 2. Experiments on synthetic and natural images are presented in Section 3. The discussion is presented in Section 4. Finally, this paper ends with a conclusion in Section 5.

2. Methodology

NLFS algorithm proposed in [18] worked by smoothing the image with a Gaussian filter to reduce noise then applying non-linear filtering scheme to get the magnitude. Assume that an image I is smoothed by Gaussian filters horizontally N_x and vertically N_y , as in the Eq. (1), where S is the smoothed image.

$$S = I * N_x * N_y \tag{1}$$

After that, non-linear filters are applied to the smoothed image to get the horizontal and the vertical components S_H and S_V . To do this, NLFS convolves the smoothed image with non-linear horizontal filters H^+ and H^- to get S_H as in Eqs. (2)-(4) and with non-linear vertical filters V^+ and V^- to get S_V as in Eqs. (5)-(7).

$$S_H = S_{H^+} + S_{H^-} \tag{2}$$

$$S_{H^+} = S * H^+ | S_{H^+} \geq 0 \tag{3}$$

$$S_{H^-} = -(S * H^-) | S_{H^-} \leq 0 \tag{4}$$

where $H^+ = [0 \ 1 \ -1]$ and $H^- = [1 \ -1 \ 0]$

$$S_V = S_{V^+} + S_{V^-} \tag{5}$$

$$S_{V^+} = S * V^+ | S_{V^+} \geq 0 \tag{6}$$

$$S_{V^-} = -(S * V^-) | S_{V^-} \leq 0 \tag{7}$$

where $V^+ = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$ and $V^- = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$

After that, the gradient or magnitude M is calculated using Eq. (8). In this step, the magnitude is rescaled and normalized. In order to retain the width of the edge to one pixel and to get a better localization, local maxima of M is calculated. The researchers [18] did not suggest a thresholding mechanism for NLFS. For comparison reason of this work, we find NLFS edge image, I_e by using thresholding technique for NLFS based on determining 70% of the histogram in M . This thresholding technique is used in CED to determine the high threshold. The 70% value of the threshold is used based on the work of [8, 29] and they suggested that if more than 80% of the histogram is used, it may lead to non-continuous edges.

$$M = \sqrt{S_H^2 + S_V^2} \tag{8}$$

A good edge detection algorithm is an algorithm that is able to detect as many true edges as possible while removing the noise and false edges [30]. However, the presence of noise and some irregular structures in the image can cause the edge detector to misclassify noise as edges. In this case, these edges are unwanted. There are many types of noise that may degrade an image. Additive type noise like Gaussian noise is commonly found in images with low illumination. The equation of the image with Gaussian noise is $I+N(\mu_N, \sigma_N)$, where $N(\mu_N, \sigma_N)$ is a Gaussian random distribution with mean μ_N and variance σ_N . Multiplicative noise like speckle noise is usually found in satellite radar images and the equation of the image with speckle noise is $I+I \times U(\mu_U, \sigma_U)$, where $U(\mu_U, \sigma_U)$ is a uniform random distribution with mean μ_U and variance σ_U . Some images might also contain impulsive noise like salt and pepper noise. Besides the three famous types of noise, there are many other types of image degradation like aliasing, jpeg compression, blurring, and

illumination. However, the focus of this work is on the additive, multiplicative and impulsive noises. In general, the lengths of the noisy edges tend to be short, while for the true edges, their lengths are longer. This feature of the noisy edges provides a priori knowledge about noise so that they can be removed.

The proposed Enhanced Non-linear Filtering Scheme edge detector algorithm (ENLFS) is different from NLFS in which it included the length of the edge as additional feature together with the gradient. It is assumed that the true edges tend to be longer than noisy edges. The inclusion of the edge length makes the proposed algorithm more resistant to noise. The gradient edge feature G is the magnitude image M masked by the NLFS edge image I_e as in the Eq. (9).

$$G = \begin{cases} M, & I_e = 1 \\ 0, & I_e = 0 \end{cases} \quad (9)$$

The length edge feature L is calculated by tracking the edges of NLFS edge image I_e and counting the number of pixels for each edge. Figure 1 shows how to track and measure the length of the edges l . Figure 1 shows three types of edges. They are opened edge, closed edge and branched edge. The first step is to scan NLFS edge image I_e to find edge pixel, starting/ending edge pixel and branched edge pixel. The scanning is done under the assumption that the edges are one-pixel width and by considering eight connected neighbours. Starting/Ending edge pixel is an edge pixel with one neighbour edge pixel ignoring branched edge pixel. The branched edge pixel is an edge pixel with three or more neighbour edge pixels. To track an edge, the algorithm starts from the starting/ending edge pixel of an edge until it reaches the other starting/ending edge pixel. This tracking considers all possible directions like horizontal, vertical and diagonal. In each step of tracking, one neighbour edge pixel is counted until all edge pixels of the edge are counted. The length of the edge l is the number of edge pixels counted along the edge. This procedure is repeated for all open and branched edges. Finally, for the closed edges, the algorithm starts from any edge pixel and tracks the edge until it reaches the starting edge pixel. This tracking method tracks each edge once only.

It is important to mention that there are two main reasons why we selected the edges produced from NLFS algorithm as the initial edges to be tracked. The first reason is that NLFS operator is more resistant to noises than that of convolution type operators. The second reason is because NLFS uses non-linear filtering scheme, which can be biased to a certain area. In this operator, it is biased to the light area. This bias makes sure that the edge is continuous along one segment (light area segment) which reduces the possibility to track non-related edges from other segments.

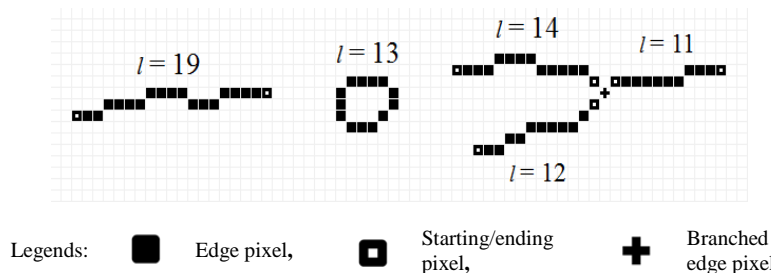


Fig. 1. Calculating edge length of opened, closed, and branched edges.

Finally, the edge image of the proposed ENLFS algorithm I_E is calculated in Eq. (10), where the gradient edge feature G and the length edge feature L contribute to detecting the true edges. The value of length threshold T_L for this work is set to be five pixels as we found in many experiments it was a suitable value for the ENLFS to differentiate between true edges and noises

$$I_E = \begin{cases} 1, & L/T_L + G \geq 1 \\ 0, & L/T_L + G < 1 \end{cases} \quad (10)$$

3. Experiments

The proposed algorithm ENLFS was compared with the most recent and related algorithms like NLFS, NNSD with Gaussian pre-filtering and SED. In order to make a better comparison, ENLFS is also compared with traditional and standard CED. Accuracy assessment was done using $F1$ value, which is the harmonic mean of the precision and recall whereby both are derived from true positive (TP), true negative (TN), false positive (FP), and false negative (FN) as in Eqs. (11)-(14) [26, 31, 32].

In more detail, the image is treated as a binary classification task between the pixels of edge image and the ground truth image. TP is the number of edge pixels in the edge image that match the edge pixels in the ground truth image. TN is the number of non-edge pixels in the edge image that match the non-edge pixels in the ground truth image. On the other hand, FP is the number of edge pixels in the edge image that do not match edge pixels in the ground truth image. FN is the number of non-edge pixels in the edge image that do not match non-edge pixels in the ground truth image. Several researchers [26, 32] used $F1$ value for accuracy assessment in the field of edge detection.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (11)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (12)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (13)$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (14)$$

Images that were used in the analyses included synthetic and natural images. The two synthetic images were generated on the lab computer. The two natural images were taken from [33]. These images are shown in Fig. 2 contain varying degrees of contrast, length, and shape of the edges. Results of the edge detectors like CED, NLSF, NNSD, SED and ENLFS are shown in Figs. 3-22. Qualitative analysis of the results can be performed by visually inspecting the results. Images were tested without noise and degraded with three types of noise like (Gaussian noise with mean, $\mu_N = 0$ and sigma, $\sigma_N = 0.01$), (Speckle noise with mean $\mu_U = 0$ and $\sigma_U = 0.05$) and (salt and pepper noise with a signal to noise ratio SNR = 0.05). The proposed edge detection algorithm ENLFS demonstrated its superiority as compared to others. Results show that ENLFS is able to detect small details and low contrast edge at the same time removing noise.

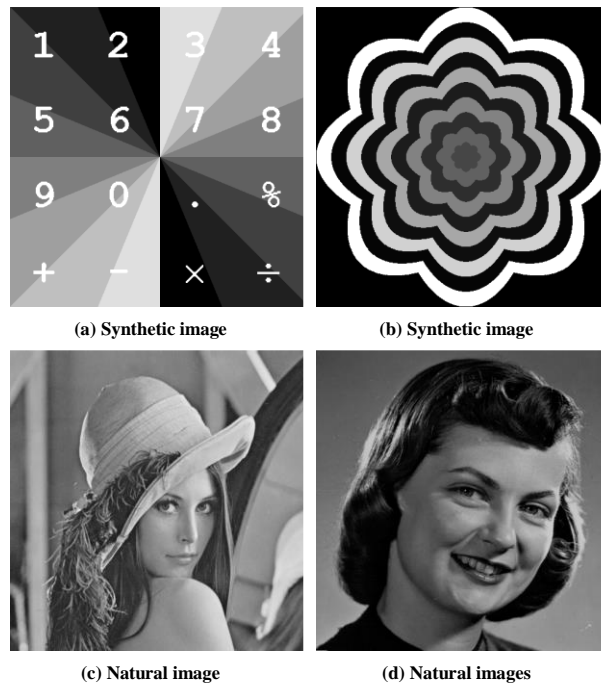


Fig. 2. Test images.

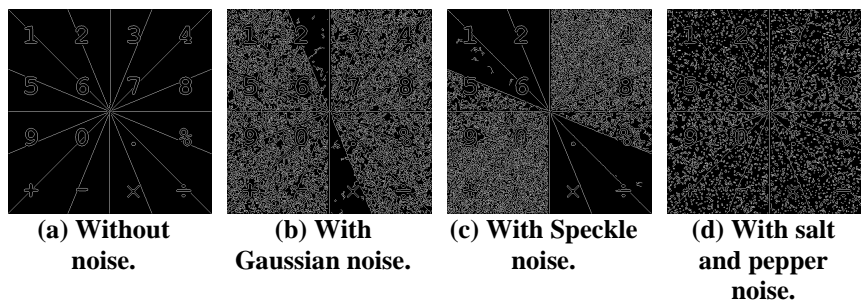


Fig. 3. Results of CED on the synthetic image of Fig. 2(a).

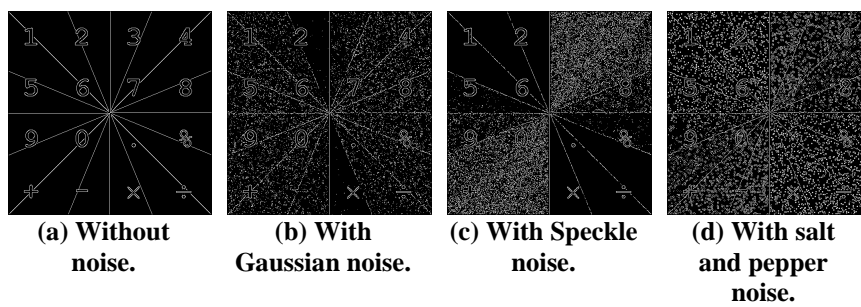


Fig. 4. Results of NLFS on the synthetic image of Fig. 2(a).

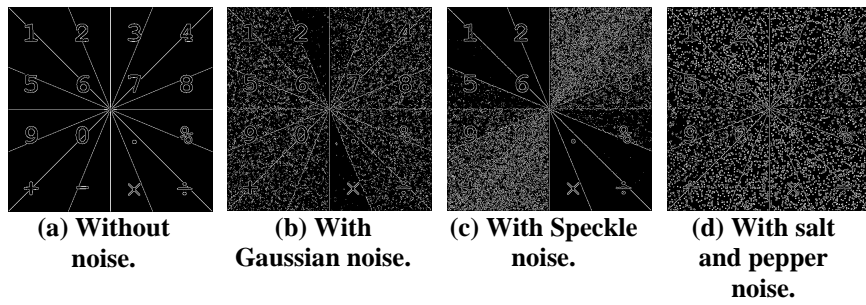


Fig. 5. Results of NNSD on the synthetic image of Fig. 2(a).

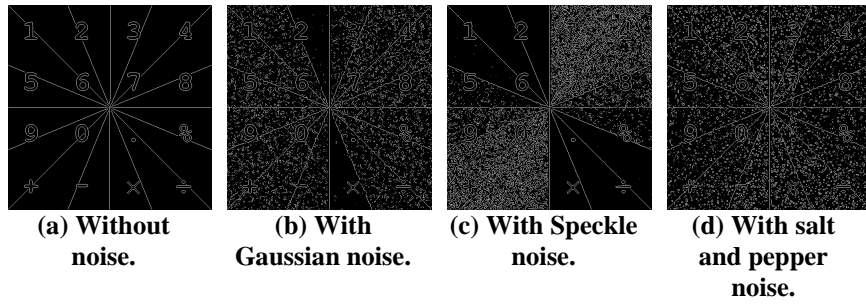


Fig. 6. Results of SED on the synthetic image of Fig. 2(a).

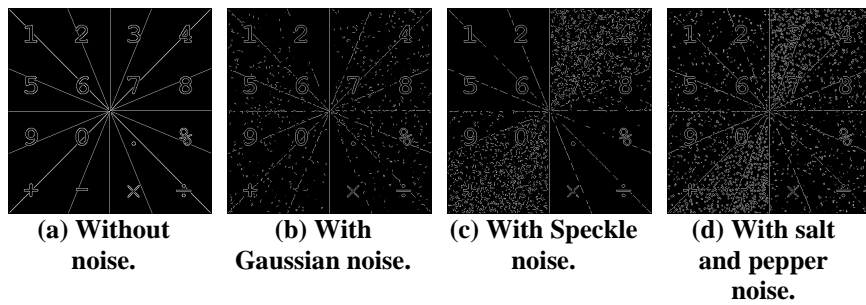


Fig. 7. Results of ENLFS on the synthetic image of Fig. 2(a).

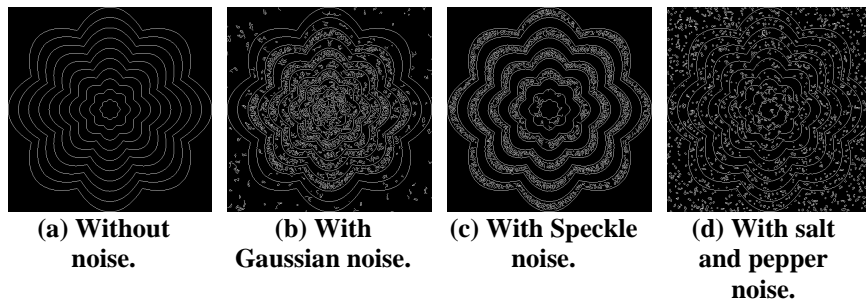


Fig. 8. Results of CED on the synthetic image of Fig. 2(b).

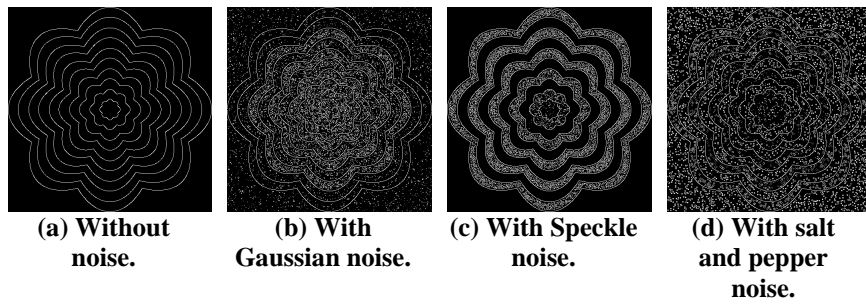


Fig. 9. Results of NLFS on the synthetic image of Fig. 2(b).

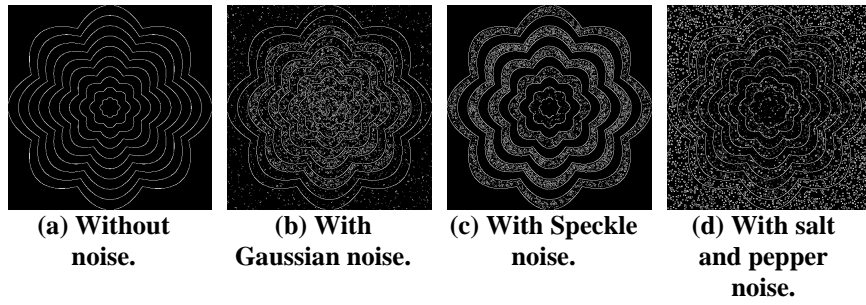


Fig. 10. Results of NNSD on the synthetic image of Fig. 2(b).

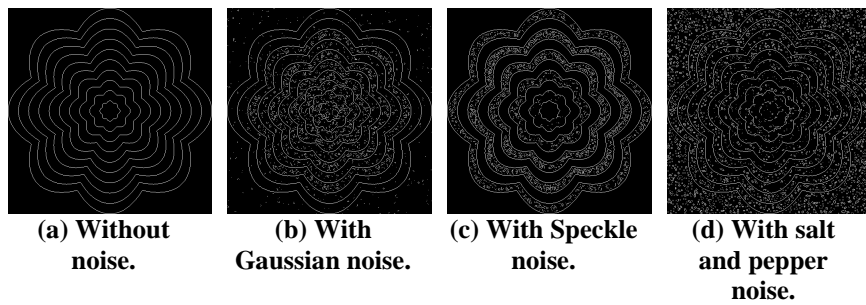


Fig. 11. Results of SED on the synthetic image of Fig. 2(b).

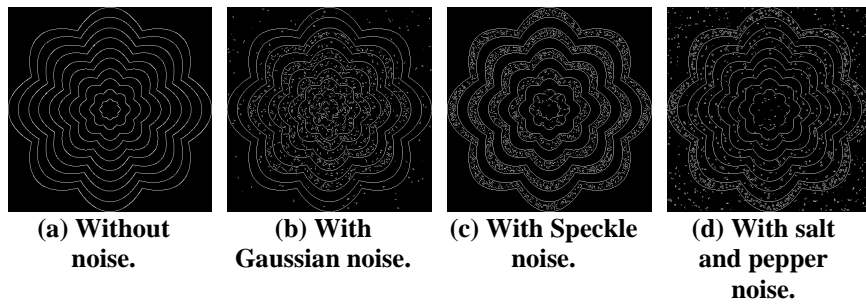


Fig. 12. Results of ENLFS on the synthetic image of Fig. 2(b).

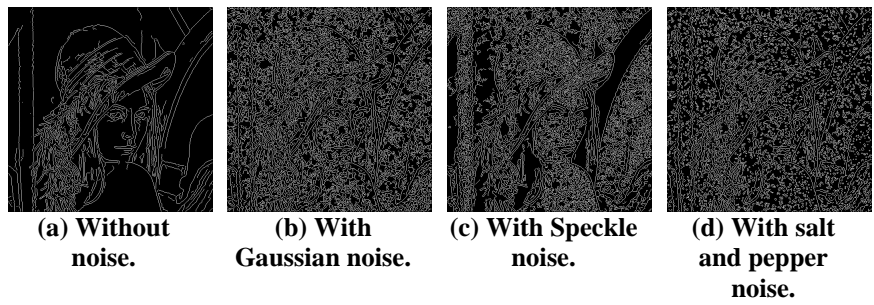


Fig. 13. Results of CED on the natural image of Fig. 2(c).

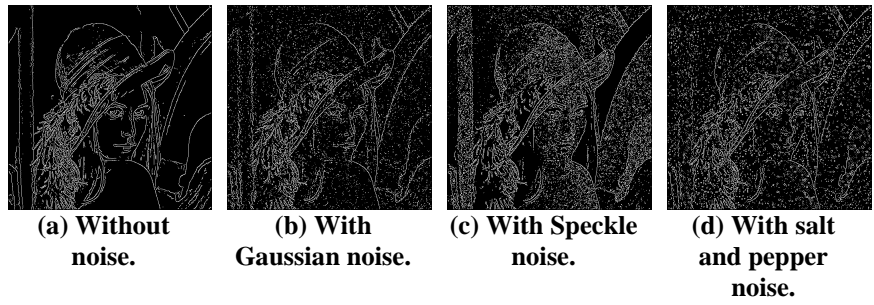


Fig. 14. Results of NLFS on the natural image of Fig. 2(c).

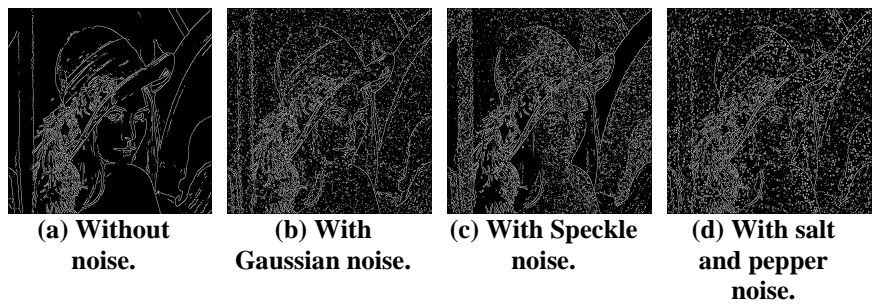


Fig. 15. Results of NNSD on the natural image of Fig. 2(c).

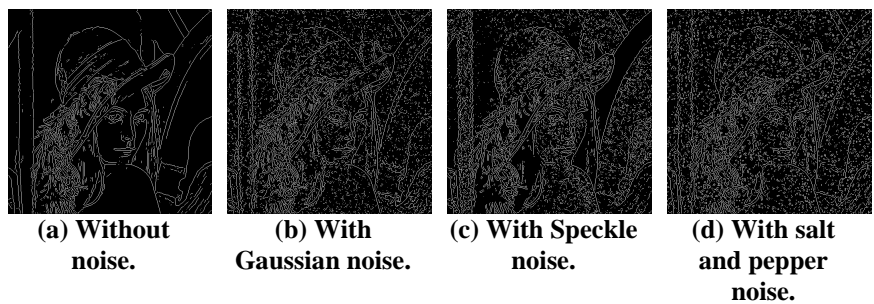


Fig. 16. Results of SED on the natural image of Fig. 2(c).

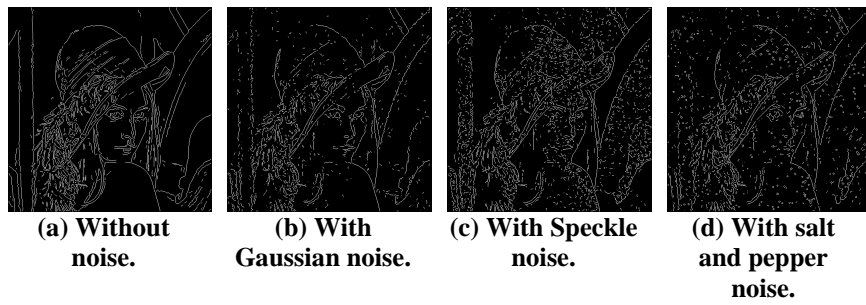


Fig. 17. Results of ENLFS on the natural image of Fig. 2(c).

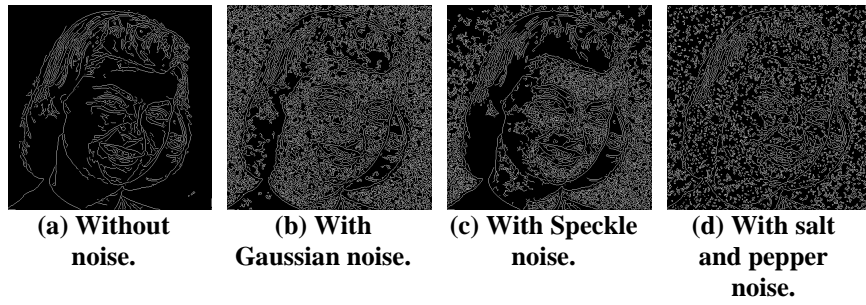


Fig. 18. Results of CED on the natural image of Fig. 2(d).

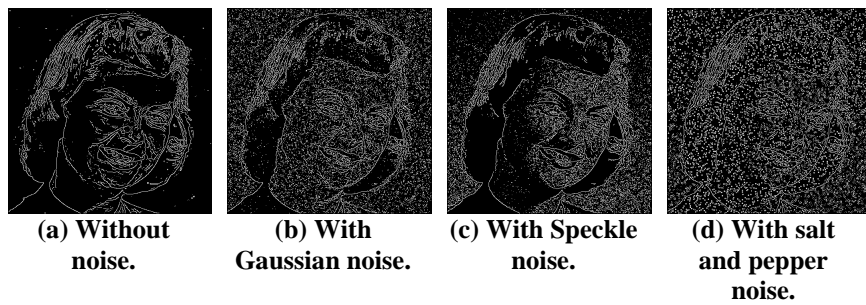


Fig. 19. Results of NLFS on the natural image of Fig. 2(d).

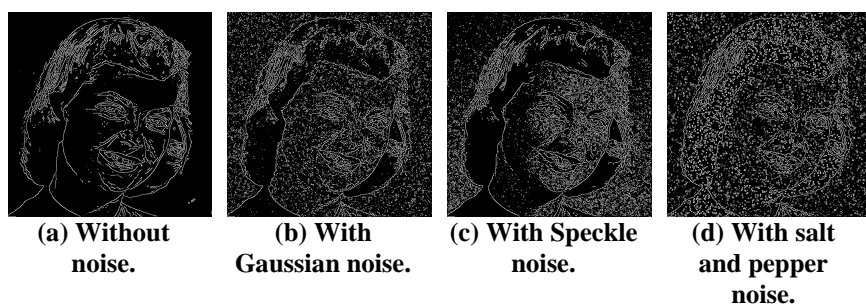


Fig. 20. Results of NNSD on the natural image of Fig. 2(d).

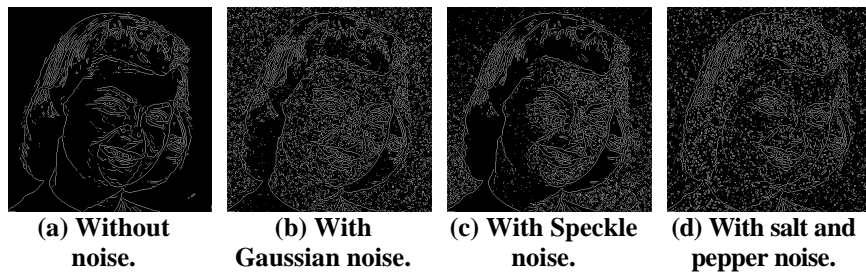


Fig. 21. Results of SED on the natural image of Fig. 2(d).

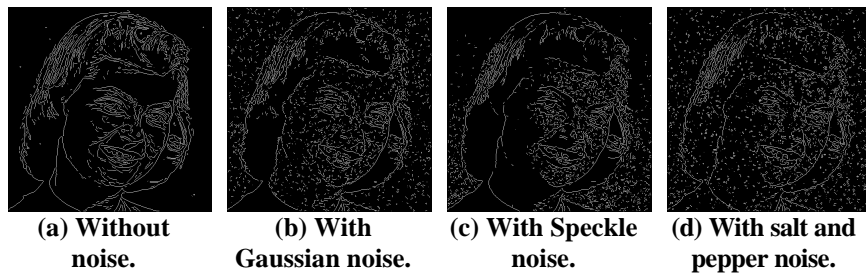


Fig. 22. Results of ENLFS on the natural image of Fig. 2(d).

Quantitative analysis was done by calculating the *FI* value, accuracy, precision, and recall. Only the two synthetic images were considered in the quantitative test because it is difficult to generate the ground truth from natural images and it is subjected to the researchers. Table 1 shows the average *FI* values taken from the two synthetic images without noise and with Gaussian, speckle and salt and pepper noises. The proposed algorithm produced the highest *FI* values in images without noise, with Gaussian, speckle, salt and pepper. Table 2 shows the average accuracies on the synthetic images. The proposed algorithm produced the highest accuracy in all the cases. Table 3 shows the average precisions. Here, the proposed algorithm produced the highest precisions. Finally, Table 4 shows the average recalls, where CED tends to produce the highest recalls. In general, the proposed algorithm produced the best results.

Table 1. Average *FI* values of the two synthetic images.

	No noise	Gaussian	Speckle	Salt and pepper	Overall Average
CED	1.0000	0.3531	0.3892	0.4079	0.5375
NLFS	1.0000	0.5012	0.4494	0.3820	0.5831
NNSD	1.0000	0.4999	0.4774	0.3913	0.5917
SED	1.0000	0.5812	0.5127	0.4606	0.6386
ENLFS	1.0000	0.7351	0.5764	0.5764	0.7233

Table 2. Average accuracies of the two synthetic images.

	No noise	Gaussian	Speckle	Salt and pepper	Overall Average
CED	1.0000	0.7704	0.8508	0.8702	0.8728
NLFS	1.0000	0.9029	0.8870	0.8261	0.9040
NNSD	1.0000	0.8918	0.8832	0.8297	0.9011
SED	1.0000	0.9423	0.9184	0.9070	0.9419
ENLFS	1.0000	0.9790	0.9569	0.9528	0.9722

Table 3. Average precisions of the two synthetic images.

	No noise	Gaussian	Speckle	Salt and pepper	Overall Average
CE	1.0000	0.2235	0.2490	0.2621	0.4336
NLFS	1.0000	0.3378	0.2926	0.2379	0.4671
NNSD	1.0000	0.3392	0.3202	0.2468	0.4765
SED	1.0000	0.4239	0.3587	0.3025	0.5213
ENLFS	1.0000	0.6300	0.4287	0.4284	0.6218

Table 4. Average recalls of the two synthetic images.

	No noise	Gaussian	Speckle	Salt and pepper	Overall Average
CE	1.0000	0.9943	0.9948	0.9931	0.9362
NLFS	1.0000	0.9710	0.9769	0.9886	0.9841
NNSD	1.0000	0.9845	0.9902	0.9811	0.9881
SED	1.0000	0.9899	0.9902	0.9811	0.9920
ENLFS	1.0000	0.8962	0.9902	0.9362	0.9274

Figure 23 shows the average *F1* values of the two synthetic images degraded with Gaussian, speckle, and salt and pepper noises. The proposed algorithm produced the highest *F1* values in all cases. Figure 23 also shows that when the level of noise for all the Gaussian, speckle and salt and pepper noises were $\sigma_N = \sigma_U = \text{SNR} = 0$, the *F1* values for all tested algorithms were approaching 1. However, when the level of noise increased, the *F1* values for all the algorithms were reduced. The *F1* value of the CED demonstrated sharp reduction, while the *F1* value for the proposed algorithm decreased steadily. Based on this trend, the *F1* value for the proposed algorithm maintained as the highest in varying level of noises, suggesting that the algorithm was highly resistant to noises.

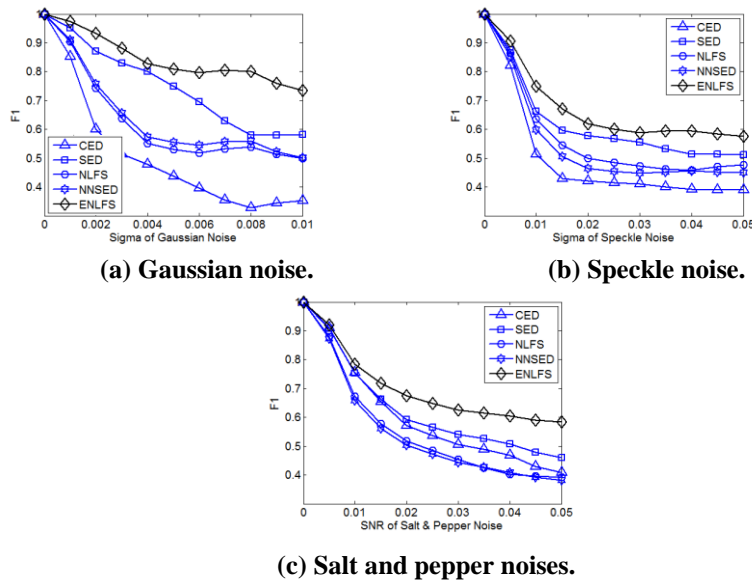


Fig. 23. Average *F1* values tested on the two synthetic images with varying levels of sigma of Gaussian noise, Speckle noise and SNR of salt and pepper noises.

4. Results and Discussion

This paper proposed a new edge detection algorithm ENLFS based on NLFS operator. Apart from the gradient, this algorithm enhanced NLFS operator by considering the length of the edges as an additional feature. The threshold of the gradient was determined by 70% of the histogram of the magnitude image. Using non-linear filtering scheme enabled the proposed algorithm to detect low contrast edges and resist noise. Similar to the NLFS operator, the proposed algorithm used non-maxima suppression, which improved the edge localization. The advantage of this feature enabled sharp edges and angles to be detected clearly.

This paper presented two contributions in this work. Firstly, the proposed algorithm ENLFS was sufficient to detect low contrast and small edges and highly resistant to additive, multiplicative and impulsive noises. Comparison between the tested algorithms demonstrated that the proposed edge detection algorithm was superior over them. Secondly, apart from the gradient feature of the edge, an additional feature of the edges like the length was considered. This approach enabled the proposed algorithm to focus on long edges. The reason for this is that the longer edges tend to have a high probability to be considered as true edges. Therefore, informative edges can be extracted with less noise. Understanding the noise by analyzing the edge features is a key success of the proposed algorithm. In the future work, more edge features and the optimum threshold of the edges, will be considered to suit with different types of applications.

5. Conclusion

In this paper, a new enhanced non-linear scheme edge detection algorithm ENLFS based on NLFS operator is proposed. Incorporating the length feature in addition to the gradient feature is the key success toward removing noise. ENLFS is tested considering images without noise and with different types of noise. It is also tested with synthetic and natural images. Results demonstrated that ENLFS is more robust to noise and at the same time able to detect low contrast and short edges compare to CED, NLFS, NNSD and SED. This suggests that the trade-off between noise and information is low.

Nomenclatures

G	Gradient edge feature
H^+, H^-	Non-linear horizontal filters (positive and negative)
I	Input original image
I_E, I_e	Edge images of ENLFS and NLFS
L, l	Length edge feature and length of single edge
M	Gradient or magnitude image
N_x, N_y	Horizontal and vertical Gaussian filters
S	Smoothed image by Gaussian filters
S_H, S_V	Horizontal and vertical non-linear components
T_L	Length threshold
V^+, V^-	Non-linear vertical filters (positive and negative)

Greek Symbols

μ_N, μ_U	Means of Gaussian and Speckle noises
σ_N, σ_U	Standard deviation of Gaussian and Speckle noises
Abbreviations	
CED	Canny Edge Detector
EGT	Estimated Ground Truth Edge Detector
ENLFS	Enhanced Non-Linear Filtering Scheme Edge Detector
FN	False Negative
FP	False Positive
SMC	Scale Multiplication of Canny Edge Detector
NLFS	Non-Linear Filtering Scheme Edge Detector
NNSED	Non-Linear Noise Suppression Edge Detector
RKT	Rotating Kernel Transformation Edge Detector
ROC	Receiver Operating Characteristics thresholding technique
SED	Snakelet Edge Detector
SNR	Signal to Noise Ratio
TN	True Negative
TP	True Positive

References

1. Melin, P.; Gonzalez, C.I.; Castro, J.R.; Mendoza, O.; and Castillo, O. (2014). Edge-detection method for image processing based on generalized type-2 fuzzy logic. *IEEE Transactions on Fuzzy Systems*, 22(6), 1515-1525.
2. Russo, F.; and Lazzari, A. (2005). Color edge detection in presence of Gaussian noise using nonlinear prefiltering. *IEEE Transactions on Instrumentation and Measurement*, 54(1), 352-358.
3. Pellegrino, F.A.; Vanzella, W.; and Torre, V. (2004). Edge detection revisited. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(3), 1500-1518.
4. Torre, V.; and Poggio, T.A. (1986). On edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(2), 147-163.
5. Gunen, M.A.; Atasever H.A.; and Besok, E. (2017). A novel edge detection approach based on backtracking search optimization algorithm (BSA) clustering. *Proceedings of the 8th International Conference on Information Technology (ICIT)*. Amman, Jordan, 116-122.
6. Wang, S.; Sun, S.; Guo, Q.; Dong, F.; and Zhou, C. (2014). Image edge detection based on rotating kernel transformation. *Proceedings of the 7th International Congress on Image and Signal Processing*. Dalian, China, 397-402.
7. Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 679-698.
8. Canny, J.F. (1983). Finding edges and lines in images. *Report AI-TR-720, MIT Artificial Intelligence Laboratory*, Cambridge, Massachusetts.
9. Yitzhaky, Y.; and Peli, E. (2003). A method for objective edge detection evaluation and detector parameter selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8), 1027-1033.

10. Medina-Carnicer, R.; Carmona-Poyato, A.; Munoz-Salinas, R.; and Madrid-Cuevas, F.J. (2010). Determining hysteresis thresholds for edge detection by combining the advantages and disadvantages of thresholding methods. *IEEE Transactions on Image Processing*, 19(1), 165-173.
11. Bao, P.; Zhang, L.; and Wu, X. (2005). Canny edge detection enhancement by scale multiplication. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9), 1485-1490.
12. Bastan, M.; Bukhari, S.S.; and Breuel, T. (2017). Active canny: Edge detection and recovery with open active contour models. *IET Image Processing*, 11(12), 1325-1332.
13. Zhu, S. (2011). Edge detection based on multi-structure elements morphology and image fusion. *Proceedings of the IEEE 2nd International Conference on Computing, Control and Industrial Engineering (CCIE)*. Wuhan, China, 406-409.
14. Bhateja, V.; and Devi, S. (2011). A novel framework for edge detection of micro calcifications using a non-linear enhancement operator and morphological filter. *Proceedings of the 3rd International Conference on Electronics Computer Technology*. Kanyakumari, India, 419- 424.
15. Jiang, J.-A.; Chuang, C.-L.; Lu, Y.-I.; and Fahn, C.-S. (2007). Mathematical-morphology-based edge detectors for detection of thin edges in low-contrast regions. *IET Image Processing*, 1(3), 269-277.
16. Pawar, K.B.; and Nalbalwar, S.L. (2016). Distributed canny edge detection algorithm using morphological filter. *Proceedings of the IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology*. Bangalore, India, 1523-1527.
17. Mostaghim, M.; Ghodausi, E.; and Tajeripoor, F. (2014). Image smoothing using non-linear filters a comparative study. *Proceedings of the Iranian Conference on Intelligent Systems*. Bam, Iran. 1-6.
18. Laligant, O.; and Truchetet, F. (2010). A nonlinear derivative scheme applied to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2), 242-257.
19. Laligant, O.; Truchetet, F.; and Fauvet, E. (2012). The noise estimator NOLSE. *Report paper ID hal-00904176*.
20. Laligant, O.; Truchetet, F.; and Fauvet, E. (2013). Noise estimation from digital step-model signal. *IEEE Transactions on Image Processing*, 22(12), 5158 -5167.
21. Chen, Y.-S.; Chang, Y.-M.; and Chen, C.-H. (2012). Comparing nonlinear derivative method and Canny algorithm to tongue diagnosis in traditional Chinese medicine. *Proceedings of the IEEE Symposium on Robotics and Applications*. Kuala Lumpur, Malaysia, 407-409.
22. Krishna, A.S.; Reddy, B.E.; and Pompapathi, M. (2014). Nonlinear noise suppression edge detection scheme for noisy images. *Proceedings of the International Conference on Recent Advances and Innovations in Engineering*. Jaipur, India. 1-6.
23. Dollar, P.; and Zitnick, C.L. (2015). Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8), 1558-1570.

24. Fu, W.; Johnston, M.; and Zhang, M. (2014). Low-level feature extraction for edge detection using genetic programming. *IEEE Transactions on Cybernetics*, 44(8), 1459-1472.
25. Qasim, M.; Woon, W.L.; Aung, Z.; and Khadhirkar, V. (2012). Intelligent edge detector based on multiple edge maps. *Proceedings of the International Conference on Computer Systems and Industrial Informatics*. Sharjah, United Arab Emirates, 1-6.
26. Flores-Vidal, P.A.; Gomez, D.; Montero, J.; and Villarino, G. (2017). Classifying segments in edge detection problems. *Proceedings of the 12th International Conference on Intelligent Systems and Knowledge Engineering*. Nanjing, China, 1-6.
27. ElAraby, W.S.; Madian, A.H.; Ashour, M.A.; Farag, I.; and Nassef, M. (2017). Fractional edge detection based on genetic algorithm. *Proceedings of the 29th International Conference on Microelectronics*. Beirut, Lebanon. 1-4.
28. Zhou, P.; Ye, W.; Xia, Y.; and Wang, Q. (2011). An improved canny algorithm for edge detection. *Journal of Computational Information Systems*, 75(5), 1516-1523.
29. Forghani, A. (2000). Semi-automatic detection and enhancement of linear features to update GIS files. *Proceedings of the XIXth ISPRS Congress*. Amsterdam Netherlands, 289-296.
30. Demigny, D. (2002). On optimal linear filtering for edge detection. *IEEE Transactions on Image Processing*, 11(7), 728-737.
31. Sokolova, M.; and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45(4), 427-437.
32. Ren, J.; Jiang, J.; Wang, D.; and Ipson, S.S. (2010). Fusion of intensity and inter-component chromatic difference for effective and robust colour edge detection. *IET Image Processing*, 4(4), 294-301.
33. Rao, K.R. (2018). Retrieved March 1, 2018, from <http://www.uta.edu/faculty/krrao/dip/Courses/EE5356/index.htm>.