International Journal of Engineering & Technology, *7 (3.4) (2018)* 127-132

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Complementary Approach for Image Edge Detection Using Rough Mirror Mapping in Neural Network

Nagaraj Bhat[1]*, U Eranna[2], Manoj Kumar Singh[3]

[1]Research Scholar, Department of Electrical & Electronics Engineering, VTU RRC, Belagavi
[2]Professor & Head, Department of Electronics & Communication Engineering BITM, Bellary
[3]Manuro Tech Research Pvt. Ltd., Bengaluru
**Corresponding author E-mail: [1]nbhat437@gmail.com ,[2]jayaveer_88@yahoo.com, [3]mksingh@manuroresearch.com*

Abstract

In this paper, a complementary approach has applied to obtain the available edges in the image. The complementary image has obtained by subtracting the rough mirror mapped image from the input image. The universal approximation capability of feedforward neural network has applied to define the rough mirror mapping. Multilayer perceptron network and radial basis function network have considered obtaining the mapping. Effect of better learning has also explored in both network by applying adaptivenesss in their transform function available in the active nodes. Single image based training has given for few number of iterations in the development of mapping process. It is observed that proposed method has self adjusted content aware oriented edge detection where as many existing methods like Sobel, Prewitt have shown their limitations in observing the edges associated with contents having similar shade in the surroundings.

Keywords: *complementary approach, edge detection, feed forward network, multilayer perceptron, adaptiveness.*

1. Introduction

Edges are huge neighborhood that alters image intensity. Edges normally happen on the point of confinement between two particular locales in an image. Objective of edge recognition is to get lines drawing of a picture. Critical highlights can be extricated from the edges of a picture (e.g., lines, curves and corners). These attributes are utilized by vision algorithms in larger computers (e.g., recognition). The different physical situations cause intensity changes in an image. Geometric occurrences are divided into object boundary and surface boundary. Object boundary is discontinuity in surface color and/or depth and texture. Surface boundary is discontinuity in surface orientation and/or surface color and texture. Non-geometric occurrences are divided into specularity and shadows. When light is reflected directly, it is known as Specularity, such as a mirror. Shadows are inter-reflection from same object or from the other objects.

Edges can be demonstrated by their intensity levels.

1. Step edge: The abrupt change of image intensity level from one part of the discontinuity to the other part with large values.
2. Ramp edge: Similar to step edge in which the intensity values change is not abrupt but occurs over a small distance.
3. Ridge edge: Within some short distance, the intensity suddenly changes value but reoccurs to the starting value (generated usually by lines).
4. Roof edge: Generated usually by the intersection of surfaces, (similar as ridge edge) in which the change in intensity is not abrupt but appear over a small distance.

The following are the steps for edge detection:

1. Smoothing: Suppresses the noise completely, without disturbing the true edges.
2. Enhancement: To improve the standard of the edges, apply a filter to the image (sharpening).
3. Detection: In this procedure, which pixels of edge should be retained and which should be discarded as noise are to be determined (normally, the criterion utilized for detection is thresholding).
4. Localization: Decides the correct area of an edge (assess the area of an edge to superior to the dispersing between pixels). In this step edge thinning and linking are usually required.
5. There are some practical issues exist with regular approach of edge recognition. The differential masks serve as high pass filters which will pass high frequency components from image i.e. noise. To reduce the impact of noise, the image should be smoothed first with a low pass filter.

More filter reduces the noise, worsens the localization and vice versa. This is the tradeoff between noise suppression and localization. Then how to select the threshold value? An Edge linking and thinning procedure are necessary to get best contours. Criteria for ideal image edge detection can be characterized as:

1. Good detection: The optimal detector must minimize the probability of detecting spurious edges caused by noise (false positives), as well as that of missing real edges (false negatives).

Good localization: The image edges are identified as proximately to the true edges. Single response constraint: The detector must return only one point just for each clear edge point; i.e., decrease the number of neighborhood maxima surrounding to the true edge (maxima generated by noise).

2. Literature Survey

In image processing edge detection is an important process. Image identification, registration and segmentation are depending on edge detection. In [1], an information dependent method which has been utilized to acknowledge control strategies for previous years is suggested for edge detection. Few of the regular methods are utilized with specific parameters such as variance and threshold to implement edge detection. The restrictions about classical approach results in usually have fixed edge thickness. The control based method offers most points of interest, for example, offering consent to adjust a few parameters effortlessly. Considering the complete image, detection results using these operators when the threshold value is low includes too many noisy edge points, or they unsuccessful to add exact edges when the threshold is high. An algorithm for edge identification that consolidates global constraints with local contrast knowledge is presented in [2]. Edge detection algorithm which tends to two critical problems in long standing vision problem: (1) holistic image training and prediction; and (2) multi-scale and multi-level feature learning have been presented in [3]. Improved Canny edge detection algorithm to deal with existing issues in conventional algorithms has proposed in [4]. The anisotropic filter has applied to denoise original greyscale images and applied the genetic algorithm to search optimizing high and low thresholds used in canny operator. [5] Presented a verifying method and a framework in which edge detection is evaluated through boundary detection, i.e., the probability of recovering the full boundaries of object from the edge detection output. [6] Suggested the procedure of space edge detection, implements the space edge detection based SVM algorithm. Double threshold method of conventional Canny operator recognizes the edge depending on the knowledge of gradient magnitude, which has a lesser edge connectivity and insufficient image information. Focusing at this issue, [7] has proposed edge connection based edge detection algorithm, the HT-Canny (Hough Transform based Canny) edge detection algorithm. An improved canny edge algorithm by application of self adaptive filter has proposed in [8]. In [9], an object detection architecture based on edge computing has been proposed to attain distributed and efficient object detection through wireless communications for real-time surveillance applications. [10] has presented a MMFED (Multilevel Morphological Fuzzy Edge Detection) method for colour image edge detection method utilizing morphological operator. Depending on CNNs (cellular neural networks), neighbourhood radius image edge detection has been proposed in [11]. The edge information of crater for lander navigation has proposed in [12]. Histogram of image, adaptive Gaussian filters and adaptive canny algorithm has applied to detect edges of craters at different illumination. [13] has applied two more features along with gradient: the length and the directional change of the edges to detect the edges in the image in robust manner. In the automated manufacturing, the measurement and monitoring of tool condition are keys to the product accuracy. To meet this demand, [14] has proposed a apparatus wear checking approach based on the observed image edge detection.

3. Proposed Approach

In the image, edges are considered as the high frequency information where there is a start of transition in light intensity or colour variation or both takes place. Hence to detect the edges in the colour, if there is such kind of model which could pass the local approximated value of pixels, the obtained output image will be rough mirror image. The meaning of rough mirror image is the image which was formed by not clear glass, in result there is no clear image appeared. It can be observed that such kind of model can be achieved with use of neural network which carry the properties of universal approximation. Among the different possibilities of architecture, MLP and RBF are proved model for universal approximation. If the number of hidden nodes in the network is very small, then the developed output image will carry only the low frequency information i.e. output pixels will be smooth version of the input. The one possible architecture of MLP has shown in Fig.1. To get the edges in the image first a complementary version has to create which is the subtracted version between the input images and the obtained rough mirror image in the output. This complementary version will carry the information of input image which does not belong to the output image. Because the output images carry the smoother version of the input image, the complement image will have the high frequency information, i.e. edges available in the image. Thresholding has also applied further to make the sharper difference between the edge information and surrounding local regions. The whole process has shown in Fig.2.

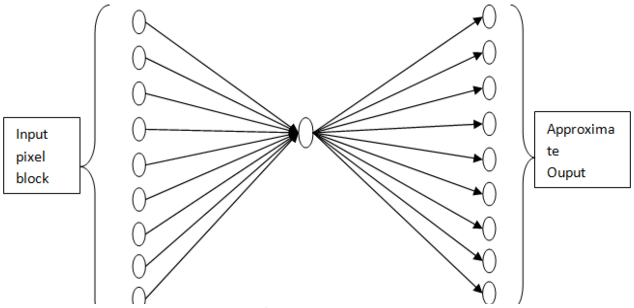


Fig. 1: MLP structure for rough mirror image creation

Four different possibilities have been explored to define the architecture. In first case, MLP with fixed slope sigmoid function has considered and here called as SFF indicate static feed forward. In second case the slope of sigmoid function in MLP is adaptive. This will increase the quality of learning. In third case a static RBF whose kernel function characteristic is constant and here called as SRBF as static RBF. In fourth case, kernel function parameters in RBF are adaptive to make the learning better. The algorithm for adaptivness in slope and kernel function has given below.

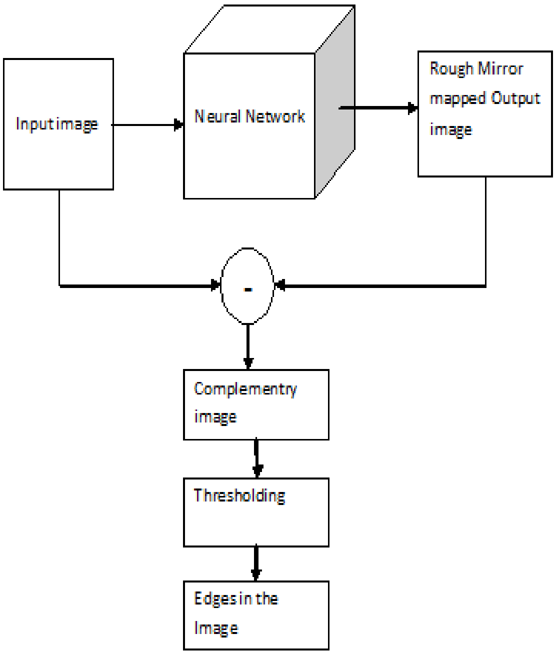


Fig. 2: Work Flow of Proposed Solution.

For learning purpose gradient based rule has applied for both networks. Weight update equation for back propagation, the rate of the update is proportional to the derivative of the nonlinear activation function. A typical activation function for neurons in MLP NN is of sigmoid type with bell shaped derivatives. In training of the network, the output of the combiner may appear in the constant region of the activation function. The derivative of the activation function in that region is very small and since the weight updates depends directly on the magnitude of the derivative hence the rate of learning becomes extremely slow. It may take many iterations before the output of the combiner moves out of the constant region. A possible solution of this problem is to increase the region of nonsaturated region by decreasing the slope of activation function. However decreasing the slope will make the system closer to linear model, which in effect reduce the advantage of having the multilayer network. Hence there is an optimum value of slope needed at each iteration as according to the landscape defined by the error function. Again the value of slope for activation function is not same for all the neurons. Complexity involved with MLP does not to have all the slopes values before training commence hence there is need to provide the adaptive mechanism which has to take care of slopes of activation function. The process for adaption of slopes can be derived simultaneously with weights optimization in terms of minimization of error function. Specifically, the slopes are to be chosen so as to minimize the performance criterion.

3.1. Learning Algorithm with Adaptive Activation function Slope

1. Initialize the weights in the network according to standard initialization process.
2. From set, the set of training data, derive the network response.
3. Compare the preferred network responses with the definite output of the network and the local error is calculated according to

For output layer$\delta_i^s = \left(d_q - x_{out,i}^s\right) g(u_i^s)$ (1)

For hidden layer:$\delta_i^s = \sum_{h=1}^{n2} \delta_h^{s+1} w_{hi}^{s+1}\, g(u_i^s)$ (2)

4. The weights of the network can be updated as

$w_{ij}^s(t+1) = w_{ij}^s(t) + \mu \delta_i^s x_{out,j}^s$ (3)

5. The slope of the activation function are updated according to

$k_i^s(t+1) = k_i^s(t) + \beta \delta_i^s + \alpha[k_i^s(t) - k_i^s(t-1)]$ (4)

6. Stop the iteration if network converged, else go back to step 2.

3.2. Functional Approach of RBF

In practice, the supervised training of the neural network can be observed as the curve fitting method. The network is given with training sets, every training sets comprising of a vector from an input vector space along with a required network response. By a learning algorithm definition, the network does the adjustments of its weights in order to minimize the error between the actual and desired response with respect to some optimization criteria. Once trained, the network carries out the interpolation in the output vector space. A nonlinear alignment between the input and the output vector spaces can be accomplished with RBF.

RBF NN architecture includes of three layers as shown in Fig3: an input layer, a single layer of nonlinear processing neurons known as hidden layer and the output layer. The output of RBF NN is computed by Equation (5).

$$y_i = f_i(x) = \sum_{k=1}^{N} W_{ik}\phi_k(x, c_k) = \sum_{k=1}^{N} W_{ik}\phi_k\left(\|x - c_k\|_2\right) \quad (5)$$

i=1, 2,….m

Where $x \in \Re^{n\times 1}$ is an input vector, $\phi_k(.)$ is a function from $\Re^+$ to $\Re$, $\|\cdot\|_2$ denotes the Euclidean distance, W_{ik} are the output layer weights. Number of neurons in the hidden layer is N and $c_k \in \Re^{n\times 1}$ is the RBF centers in the output space. In the hidden layer for each neuron, the Euclidean distance is computed between inputs to the network and its associated centers. The output of the hidden layer is a nonlinear function of the Euclidean distances. The output of the network is calculated by equation (5). The function $\phi_k(.)$ is assumed to be given and is mostly Gaussian function as given by Eq. 6

$$\phi(x) = \exp\left(-x^2/\sigma^2\right) \quad (6)$$

where σ parameter controls the width of RBF and is generally referred as spread parameter. The centers are defined points that are assumed to perform an adequate sampling of the input vector space. They are generally considered as a subset of the input data. In the case of the Gaussian RBF, the spread parameter σ is commonly set according to the following heuristic relationship

$$\sigma = \frac{d_{\max}}{\sqrt{k}} \quad (7)$$

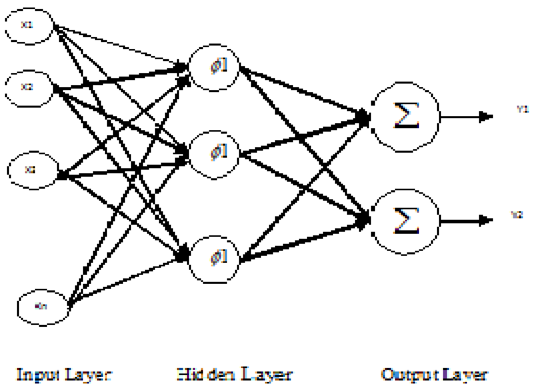


Fig. 3: RBF Neural Network Architecture.

where $d_{\max}$ is the maximum Euclidean distance between the selected centers and K is the number of the centers. Using Eq.7 the RBF of a neuron in the hidden layer of the network is given by

$$\phi(x, c_k) = \exp\left(-\frac{k}{d_{\max}^2}\|x - c_k\|^2\right) \quad (8)$$

Conventionally the center values are randomly sampled from the data set and the standard deviation is measured using the Euclidean distance available. This approach is appropriate only when there is highly concentrated data set available as very little variation exists. The performance can be improved by providing the optimal value of centers and corresponding standard deviations. The training of the parameters is a crucial part. Each parameter is updated based on the error in the output. Approach based on gradient mechanism is applied for the updating during each iteration.

3.3. Adaptive RBF Neural Network

In the fixed center based RBF NN, there is only one adjustable parameter of network available and it is weights of the output layer. This approach is simple, however to achieve sufficient sampling of the input, a substantial number of centers must be chosen from the input data set. This produces a relatively very large network.

In proposed method there are possibilities to adjust all the three set of network parameters i.e. weights, width of the RBF and location of the RBF centers. Hence, with the weights in the output layer, spread parameter in the hidden layer and position of the centers undergoes supervised training. The foremost step in the development is to define instantaneous error cost function as

$$J(n) = \frac{1}{2}|e(n)|^2 = \frac{1}{2}\left[y_d(n) - \sum_{k=1}^{N} w_k(n)\phi\{x(n), c_k(n)\}\right]^2 \quad (9)$$

When the chosen RBF is Gaussian, Eq.5 becomes

$$J(n) = \frac{1}{2}\left[y_d(n) - \sum_{k=1}^{N} w_k(n)\exp\left(-\frac{\|x(n) - c_k(n)\|_2^2}{\sigma_k^2(n)}\right)\right]^2 \quad (10)$$

The following equations is for updating the network parameters are shown by Eq.11 to Eq.13

$$w(n+1)=w(n)-\mu_w \frac{\partial}{\partial w} J(n) \quad \rangle_{w=w(n)} \tag{11}$$

$$c_k(n+1)=c_k(n)-\mu_c \frac{\partial}{\partial c_k} J(n) \quad \rangle_{c_k=c_k(n)} \tag{12}$$

$$\sigma_k(n+1)=\sigma_k(n)-\mu_k \frac{\partial}{\partial c_k} J(n) \quad \rangle_{\sigma_k=\sigma_k(n)} \tag{13}$$

4. Experimental Results

In this paper the proposed principle has applied for edge detection in image having varying characteristics. In facts having more inclination towards application in computer vision where edges are caused of various actions. To prove the principle experimentally different images are considered, which has a size of 512*512 pixels .Preprocessing is applied to each image in terms normalization and dividing the image in number of subplocks having size of 3*3. Larger sizes of blocks have the low frequency spatial information in complementary image. The size of architecture in MLP is [9 1 9] while in RBF is [9 10 9].It has seen that when there is lesser number of hidden nodes in RBF, there is very poor learning. A single image “Lena” is taken for training purpose and total 5 iterations were allowed to minimize the error. More number of iterations will make mirror image better in result lesser information of edges in complementary image. Initialization of all weights is defined as random number by uniform distribution in the range of [-1 1].Once learning has been finished, the obtained parameters have applied at the time of test image. Threshold value has defined as the function of mean and standard value of complementary image pixels. For all cases there is nonweighted addition of mean and standard deviation is considered as threshold value. The finer tuning in getting better appearance of edges can be achieved with forming a weighted sum.In the Fig.4 and Fig.5 the convergence characteristics of learning for MLP and RBF have shown. It is clear from both figures that because of adaptiveness there is lesser error in learning for the same number of iterations. The obtained corresponding complementary images with all the four architecture are shown Fig6. It can be observed that AFF has lesser information of edges in compare to SFF. The same observation have achieved with ARBF in compare to SRBF. This is because of better mapping approximation in the case adaptiveness. After thresholding the obtained edges in the image are shown in Fig.7.There is better performance has obtained with SFF in compared to others. Hence SFF has considered as final model for edge detection in other test cases.

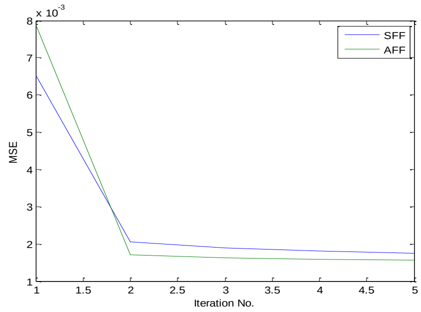


Fig. 4: Convergence characteristics for MLP.

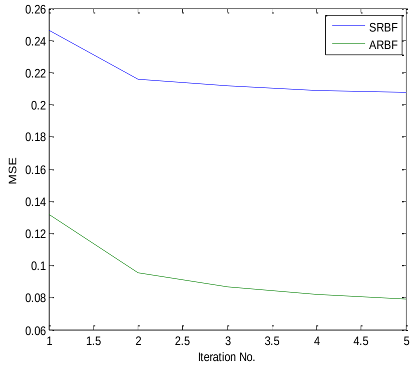
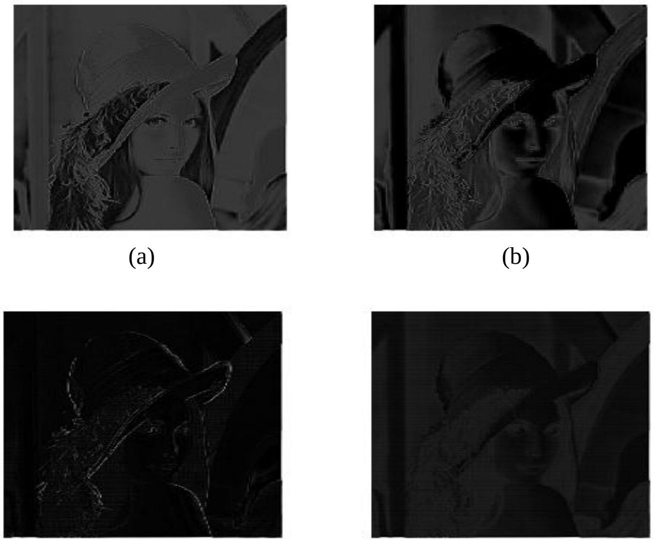


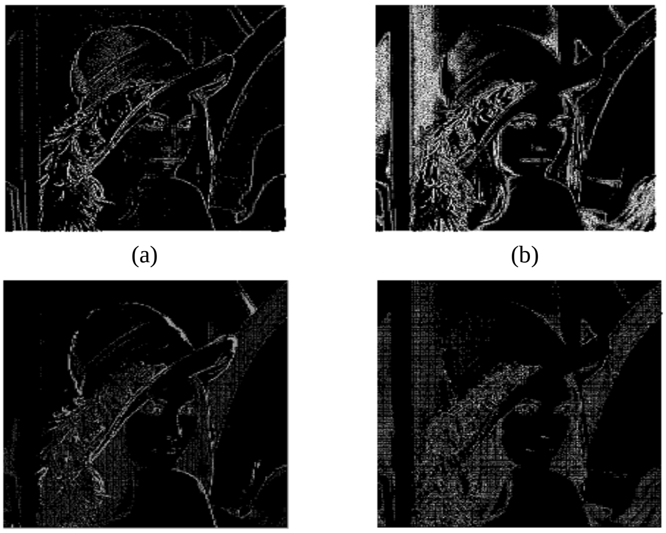
Fig. 5: Convergence characteristics for RBF.



(a) (b)

(c) (d)

Fig. 6: Complementary image obtained with (a) SFF (b) AFF (c) SRBF (d) ARBF



(a) (b)

(c) (d)

Fig. 7: Detected edges after tresholding with (a) SFF (b) AFF (c) SRBF (d) ARBF.

5. Comparison with Sobel, Prewitt, Laplacian and Canny Method

To understand the relative performance and benefits , relative comparisons are given between obtained edge image from the proposed method to other well established methods like Sobel method, Prewitt method, Laplacian method and Canny method. Morphological operation cleaning is applied to remove the isolated pixels. For the Lena image the obtained performance is shown in Fig.8.With the other two images which have high density of curve edges, performances have shown in Fig9 and in Fig.10.It is clear with observed edges that Sobel and Prewitt methods have missed some of the valuable edges available in the images while Laplacian and Canny methods were too noisy. Even though proposed method does not include any edge sharpening morphological operation, quality of edges were very appreciable and does not require the tuning of optimal threshold.

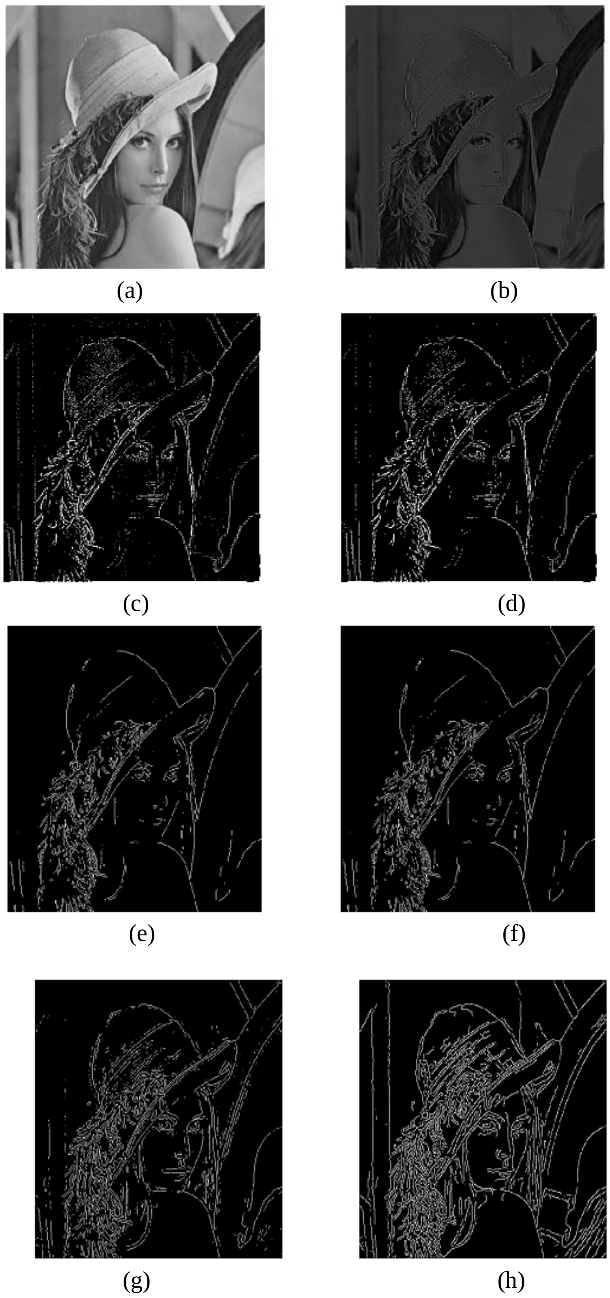
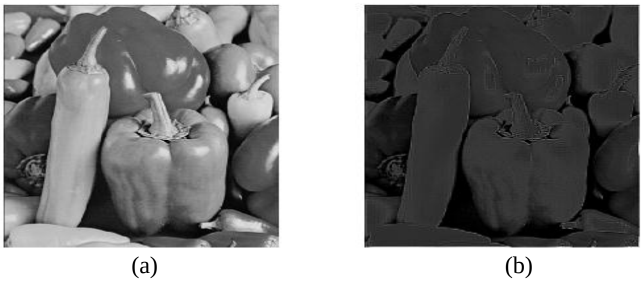


Fig. 8: (a) Lena Image (b) Complementary Image (c) detected edge by SFF (d) after cleaning ‘c’ (e) Sobel method (f) Prewitt method (g) Laplacian method (h) Canny method



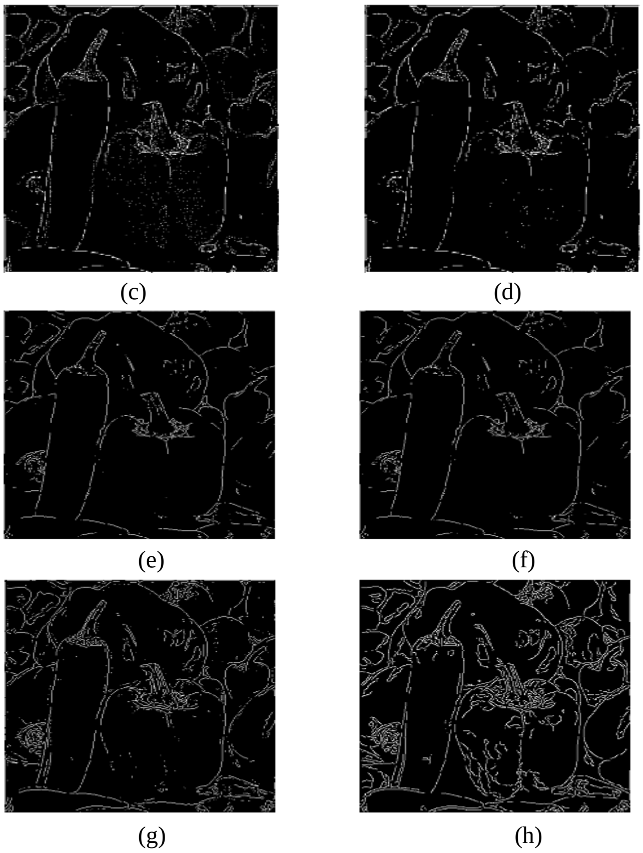


Fig. 9: (a) Vegetable Image (b) Complementary Image (c) detected edge by SFF (d) after cleaning ‘c’ (e) Sobel method (f) Prewitt method (g) Laplacian method (h) Canny method

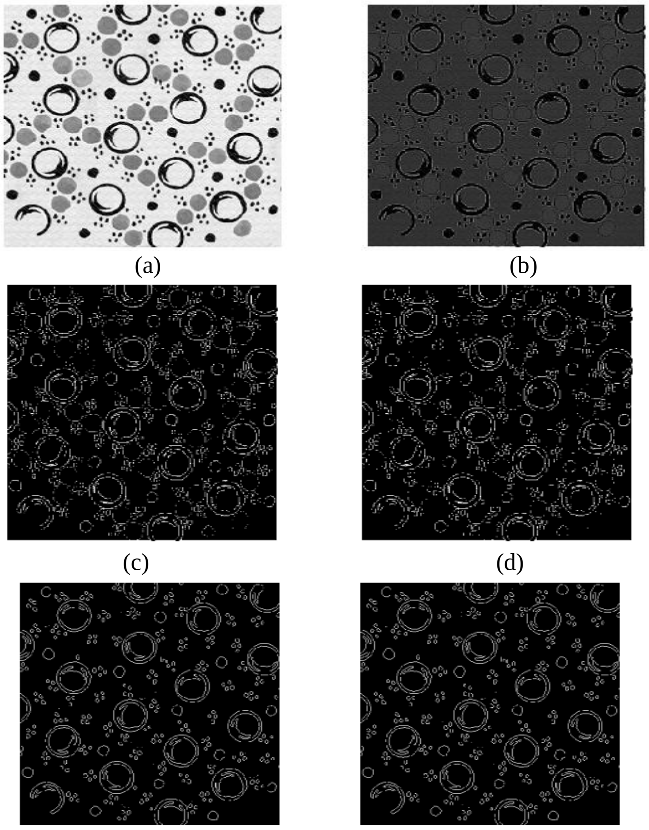


Fig. 10: (a) Circle pattern Image (b) Complementary Image (c) detected edge by SFF (d) after cleaning ‘c’ (e) Sobel method (f) Prewitt method

6. Conclusion

The universal approximation capability of neural network has been exploited to develop a very simple and effective approach of edge detection. Proposed method does not require any extensive training of neural model over set of edges instead training over a single raw image for few iterations were good enough. It is observed with extensive experiments that in comparison to other existing and well known methods they were very sensitive towards associated parameters while the proposed method does not require any tuning. It was also observed that betterment of neural model were not fruitful hence there was no need to involve the more complicated architecture. Proposed method not only offer the detection of high quality of edges but is also a simple process that makes the process computational efficient.

References

[1] Becerikli Y., Karan T.M. (2005) A New Fuzzy Approach for Edge Detection. In: Cabestany J., Prieto A., Sandoval F. (eds) Computational Intelligence and Bioinspired Systems. IWANN 2005. Lecture Notes in Computer Science, vol 3512. Springer, Berlin, Heidelberg.

[2] Chang Y., Lee DJ., Hong Y., Archibald J. (2008) Edge Detection from Global and Local Views Using an Ensemble of Multiple Edge Detectors. In: Bebis G. et al. (eds) Advances in Visual Computing. ISVC 2008. Lecture Notes in Computer Science, vol 5359. Springer, Berlin, Heidelberg.

[3] Saining Xie,Zhuowen Tu," Holistically-Nested Edge Detection," International Journal of Computer Vision,December 2017, Volume 125, Issue 1–3, pp 3–18.

[4] Wang M., Jin J.S., Jing Y., Han X., Gao L., Xiao L. (2016) The Improved Canny Edge Detection Algorithm Based on an Anisotropic and Genetic Algorithm. In: Tan T. et al. (eds) Advances in Image and Graphics Technologies. IGTA 2016. Communications in Computer and Information Science, vol 634. Springer, Singapore.

[5] Song Wang,Feng Ge,Tiecheng Liu," Evaluating Edge Detection through Boundary Detection", EURASIP Journal on Advances in Signal Processing, December 2006, 2006:076278.

[6] Meng F., Lin W., Wang Z. (2011) Space Edge Detection Based SVM Algorithm. In: Deng H., Miao D., Lei J., Wang F.L. (eds) Artificial Intelligence and Computational Intelligence. AICI 2011. Lecture Notes in Computer Science, vol 7004. Springer, Berlin, Heidelberg.

[7] Renjie Song,Ziqi Zhang,Haiyang Liu," Edge connection based Canny edge detection algorithm" ,Pattern Recognition and Image Analysis,October 2017, Volume 27, Issue 4, pp 740–747.

[8] Bing Wang, Shashong Fan, "An improved Canny edge Algorithm" 2nd international workshop on computer science and engineering 2009.

[9] Ju Ren; Yundi Guo; Deyu Zhang; Qingqing Liu; Yaoxue Zhang ," Distributed and Efficient Object Detection in Edge Computing: Challenges and Solutions ",IEEE Network ,Year: 2018, (Early Access) ,Pages: 1 – 7.

[10] Eswaran Perumal; Pramila Arulandhu ,"Multilevel morphological fuzzy edge detection for color images (MMFED)" , 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT) , Pages: 269 – 273.

[11] Wang Xue; Xu Wenxia; Li Guodong ," Image Edge Detection Algorithm Research Based on the CNN's Neighborhood Radius Equals 2",2016 International Conference on Smart Grid and Electrical Automation (ICSGEA) ,Year: 2016 ,Pages: 115 – 119.

[12] Suketu M. Saheba; Trushit K. Upadhyaya; Ritesh Kumar Sharma ," Lunar surface crater topology generation using adaptive edge detection algorithm",IET Image Processing ,Year: 2016, Volume: 10, Issue: 9 ,Pages: 657 – 661.

[13] Haider o Lawend, Anur M Muad, Aini Hussain, "Robust edge detection based on Canny algorithm for Noisy images", Journal of theoretical and applied information technology, Vol95, Issue 19,2017.

[14] Yu x, Lin X, Dai Y, " Image Detection based tool condition monitoring with morphological component Analysis", ISA TRANS 2017,pp 315-322, doi: 10.1016/j.isatra.2017.03.024. Epub 2017 Apr 5.