Volume 2, No. 3, May-June 2011

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

PSO Tuned Neural Network for False Contour Reduction

Regina Manicka Rajam.G
II M.E. Applied Electronics,PSN
College Of Engg & Technology,
Anna University, Tirunelveli, India
georegi1@yahoo.co.in

Abstract: In this project, the adaptation of network weights using Particle Swarm Optimization (PSO) was proposed as a mechanism to improve the performance of Artificial Neural Network (ANN) in modelling a image false contour reduction Technique. The false contour reduction part has already two steps. They are NN processing and bi-directional filtering. False contours are reduced by pixel wise processing using NNs in the first step and bi-directional smoothing is applied to a neighbouring region of the false contour in the second step. PSO is proposed to allow automatic update of network weights to increase the adaptability to dynamic environment. The results obtained in this paper confirmed the potential of PSO-based ANN model to successfully model false contour reduction process. The results are explored with a discussion using the SNR illustrate the usability of the proposed approach. Finally, conclusions and future works are derived.

Keywords: Bi-directional smoothing, false contour reduction, decontouring, false contour detection, neural network

I. INTRODUCTION

These days display devices such as digital television (TV) and high definition TV are getting larger. The larger display devices are, the more artifacts become noticeable, where artifacts include false contours, block artifacts, charge coupled devices sensor noise, mosquito noise, and other types of noises. Thus, and reduction of these artifacts is one of important issues in the field of digital display devices, especially in large display Devices.

Among these artifacts, this paper focuses on reduction of false contours. False contours are observed over smooth regions, such as sky, water, skin, and a single-colored object in an image that are compressed or enhanced. These false contours are an eyesore. Accordingly, false contour reduction is needed in large displays for high-quality images or videos.

First of all, when we are talking about a neural network, we should more properly say "artificial neural network" (ANN), because that is what we mean most of the time. Biological neural networks are much more complicated than the mathematical models we use for ANNs. But it is customary to be lazy and drop the "A" or the "artificial". An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

ANNs have been applied to an increasing number of real-world problems of considerable complexity. Their most important advantage is in solving problems that are too complex for conventional technologies -- problems that do not have an algorithmic solution or for which an algorithmic solution is too complex to be found. In general, because of their abstraction from A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the biological brain, ANNs are well suited to problems that people are good at solving, but for which computers are not. These problems include pattern recognition and forecasting (which requires the recognition of trends in data).

Because gazing into the future is somewhat like gazing into a crystal ball, so it is better to quote some "predictions". Each prediction rests on some sort of evidence or established trend which, with extrapolation, clearly takes us into a new realm.

Neural Networks will fascinate user-specific systems for education, information processing, and entertainment. "Alternative ralities", produced by comprehensive environments, are attractive in terms of their potential for systems control, education, and entertainment. This is not just a far-out research trend, but is something which is becoming an increasing part of our daily existence, as witnessed by the growing interest in comprehensive "entertainment centers".

This "programming" would require feedback from the user in order to be effective but simple and "passive" sensors (e.g fingertip sensors, gloves, or wristbands to sense pulse, blood pressure, skin ionisation, and so on), could provide effective feedback into a neural control system. This could be achieved, for example, with sensors that would detect pulse, blood pressure, skin ionisation, and other variables which the system could learn to correlate with Neural networks, integrated with other artificial intelligence technologies, methods for direct culture of nervous tissue, and other exotic technologies such as genetic engineering, will allow us to develop radical and exotic life-forms whether man, machine, or hybrid.

Neural networks will allow us to explore new realms of human capabillity realms previously available only with extensive training and personal discipline. It is necessary in order to facilitate a man machine system interface.

A Radial Basis Function (RBF) neural network has an input layer, a hidden layer and an output layer. The neurons in the hidden layer contain Gaussian transfer functions whose outputs are inversely proportional to the distance from the center of the neuron.

RBF networks are similar to K-Means clustering and PNN/GRNN networks. The main difference is that PNN/GRNN networks have one neuron for each point in the training file, whereas RBF networks have a variable number of neurons that is usually much less than the number of training points. For problems with small to medium size training sets, PNN/GRNN networks are usually more accurate than RBF networks, but PNN/GRNN networks are impractical for large training sets. There are three layers.

Input layer – There is one neuron in the input layer for each predictor variable. In the case of categorical variables, N-1 neurons are used where N is the number of categories. The input neuron (or processing before the input layer) standardizes the range of the values by subtracting the median and dividing by the interquartile range. The input neurons then feed the valus to each of the neurons in the hidden layer.

Hidden layer – This layer has a variable number of neurons (the optimal number is determined by the training process). Each neuron consists of a radial basis function centered on a point with as many dimensions as there are predictor variables. The spread (radius) of the RBF function may be different for each dimension. The centers and spreads are determined by the training process. When presented with the x vector of input values from the input layer, a hidden neuron computes the Euclidean distance of the test case from the neuron's center point and then applies the RBF kernel function to this distance using the spread values. The resulting value is passed to the the summation layer.

Summation layer – The value coming out of a neuron in the hidden layer is multiplied by a weight associated with the neuron (W1, W2, ...,Wn in this figure) and passed to the summation which adds up the weighted values and presents this sum as the output of the network. Not shown in this figure is a bias value of 1.0 that is multiplied by a weight W0 and fed into the summation layer. For classification problems, there is one output (and a separate set of weights and summation unit) for each target category. The value output for a category is the probability that the case being evaluated has that category.

The following parameters are determined by the training process. The number of neurons in the hidden layer. The coordinates of the center of each hidden-layer RBF function. The radius (spread) of each RBF function in each dimension. The weights applied to the RBF function outputs as they are passed to the summation layer.

Various methods have been used to train RBF networks. One approach first uses K-means clustering to find cluster centers which are then used as the centers for the RBF functions. However, K-means clustering is a computationally intensive procedure, and it often does not generate the optimal number of centers. Another approach is to use a random subset of the training points as the centers.

DTREG uses a training algorithm developed by Sheng Chen, Xia Hong and Chris J. Harris. This algorithm uses an evolutionary approach to determine the optimal center points and spreads for each neuron. It also determines when to stop adding neurons to the network by monitoring the estimated leave-one-out (LOO) error and terminating when the LOO error beings to increase due to over fitting.

The computation of the optimal weights between the neurons in the hidden layer and the summation layer is done using ridge regression. An iterative procedure developed by Mark Orr (Orr, 1966) is used to compute the optimal regularization Lambda parameter that minimizes generalized cross-validation (GCV) error.

Previous works on false contour reduction include Mitsa and Parker's blue noise mask [1], Daly and Feng's decontouring [2]–[3], Lee *et al.*'s two-stage false contour reduction algorithm [4], and Choi *et al*.'s false contour reduction algorithm [5]. The Mitsa and Parker's blue noise mask is the unstructured mosaic pattern used to produce the high-quality image where the artifacts such as false contours are removed [1]. Daly and Feng presented a false contour reduction method, which applies a spatiotemporal dither to the lower bit-depth image and then performs requantization [3]. The algorithm proposed in [4] consists of two stages: false contour detection and reduction. The candidate pixels regarded as false contours are detected in the first stage and then reduced in the second stage using one-dimensional (1-D) adaptive-size directional smoothing filters. The algorithm presented in [5] expands the false contour region using directional dilation and reduces false contours using edge-preserving filtering. These works have some limitations. Daly and Feng's method can be applied to specific bit-depth, and Lee *et al*.'s method is not enough for false contour reduction. Choi *et al*.'s method produces blurring through bilateral filtering in texture region, such as skin and lawn. The other methods require *a priori* information about the cause of false contours and their characteristics.

In this paper, we propose a false contour reduction algorithm using neural networks (NNs) [6, 7, 8] and adaptive bi-directional smoothing. For image enhancement, NN filtering was used to restore blurred edges [9] and to reduce TV artifacts [10]. The proposed algorithm consists of two parts: false contour detection and reduction parts. The false contour reduction part is composed of two steps: NN processing and bi-directional filtering. In the first step, false contour are reduced by pixel wise processing using NNs. In the second step, a bi-directional smoothing filter is applied to a neighbouring region of the false contour.

II. PROPOSED ADAPTIVE BI-DIRECTIONAL FALSE CONTOUR REDUCTION ALGORITHM

This section describes an adaptive bi-directional false contour reduction algorithm. In the proposed algorithm, we first generate the binary false contour map using Lee *et al.*'s algorithm [4] and reduce false contours using NNs, however, which is not enough to remove false contours. False contours are still observed in neighbouring local smooth regions of the false contour region. Therefore, the proposed method further reduces false contours using an adaptive bi-directional smoothing filter not only in the false contour region but also in the neighbouring local smooth region of the false contour region.

A. False Contour Detection

False contours are detected in regions of an image, over which the gray level or color value at a pixel is smoothly changed, due to an insufficient number of gray levels or color levels in quantization, high image compression, image enhancement, image processing, and object motion or illumination changes [1–4].

In this paper, the proposed false contour detection algorithm is based on the two-stage false contour detection algorithm presented in [4]. In the first stage, smooth regions are removed by re-quantization whereas false contours, edges, and textures remain, where false contours are detected with the width of two pixels. In the second stage, false contours are separated from edges or texture regions using four directional contrast features .

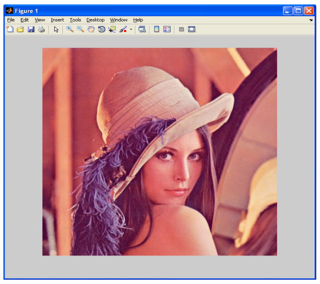

Figure.1.input image

B. False Contour Reduction

Intensity variation in the smooth region with false contours is larger than that in the smooth region without false contours. False contours detected by Lee *et al*.'s algorithm [4] just represent the positions of pixels at which the intensity changes more abruptly than those at neighbouring pixels. A small number of pixels detected as false contours make an image unpleasant to the eye and degrade the image quality badly. For example, a false contour line composed of false contour pixels divides the smooth region into more than two sub-regions, which looks divided and unnatural. Thus, it is not sufficient to modify pixel values only at false contours. The neighboring pixel values of false contours are also to be modified in order to look more natural.

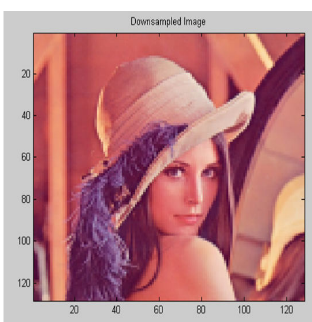

Figure .2.Down sampled image

To effectively deal with both the false contour pixels and their neighbouring pixels, the proposed false contour reduction algorithm composes of two steps: 1) pixel wise false contour reduction based on NN learning (using a number of test images and their synthetic images containing false contours) and 2) adaptive bi-directional smoothing for smoothing of both false contour pixels and their neighbouring pixels. We obtain weighting factors offline by NN learning in the first step of the false contour reduction with five test images used in experiments. With test images that contain false contours, we first find false contour points and then their directionality to use as input data of NN and compute weighting factors. The output image of the first step is used as an input to the second step. Following three subsections describe the procedures of NN learning and the two steps of the proposed false contour reduction algorithm.

a. NN Learning

The first step of the proposed false contour reduction algorithm employs NN learning, in which pixel wise Processing is performed. This subsection describes how to perform NN learning. As mentioned previously, false contours are artifacts that we can observe in smooth regions, where the colour tone or intensity varies smoothly. It is noted that there is a tendency in smooth regions to change with the directionality, where the direction is defined as the direction orthogonal to that of false contours.

We can define two features that explain how the intensity varies at the position of false contours: 1) strength and 2) direction. For example, false contours can be regarded as contours in a height or elevation map, which tells variations of altitude. We can define the strength of intensity variation as a function of the distance between false contour lines, which is measured along the line perpendicular to false contour lines. If the false contour lines are close to each other, pixel intensity values change abruptly. While if the distance between neighbouring contour lines is large, pixel intensity values vary slowly in this region. We can also measure directions of intensity variation along a line perpendicular to false contour line. These two features show the main characteristics of false contours and can be used efficiently in false contour reduction processing.

Figure.3.False contour image

There are eight false contour directions that are considered in the proposed method. We assume that the pixel intensity in dark gray region is larger than that of bright gray region. Eight directions are determined by the intensity gradient vector at false contours and then eight NNs are constructed for eight different directions. We have eight figures and it shows the various direction.

b. First Step of False Contour Reduction: Pixel Wise

The first step is the pixel wise false contour reduction using weights obtained by NN learning. Intensity of pixels detected as false contours are replaced by the predicted or reconstructed ones by NN. With an $o \times o$ mask, an input data vector **t** is used in NN learning, whose size is $O = o \times o$. The input data vector **t** is constructed by raster scanning pixel values in the $o \times o$ mask, from top to bottom and from left to right. Computed pixel values using NN replace pixel values at false contour pixels. NN computation is as follows. To calculate this we apply input vector **t** to input layer.

Here we have to use the image with neural network and by this we have to reduce the false contour image.

535

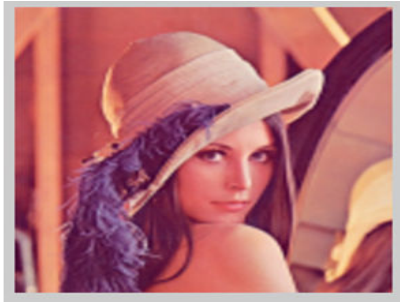

Figure.4.Image After Neural Network

c. Second Step of False Contour Reduction: Adaptive Bi-Directional Smoothing

In the first step, we reduce false contours at the points specified by false contour detection map obtained by Lee *et al*.'s algorithm [4]. False contours have characteristics different from those of the salt and pepper noise. The salt and pepper noise is pixel wise artifacts. Whereas false contours are observed over the neighbourhood, in which the intensity value at a pixel is smoothly changed. Thus, for better and natural false contour reduction we have to consider both the points detected as false contours as well as their neighbouring pixels.

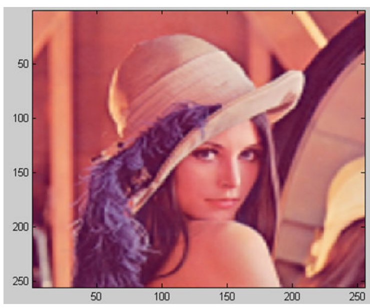

Figure.5.After bi directional smoothing

In the second step, we apply the proposed method that reduces the spreading artifacts shown over neighbouring pixels using an adaptive bi-directional smoothing filter.The way how to decide positions of pixels that are processed by an adaptive bi-directional smoothing filter. The regions with different color denote the different pixel intensity. The boundary of each color region depicts false contour. In Fig. 3, the red line represents object boundary and the pixel with orange color denotes a position determined as the pixel that will be filtered more than once in the second step by an adaptive bi-directional smoothing filter.

The process of the proposed adaptive bi-directional smoothing and the decision steps are as follows:

Step 1: At pixel position (i, j) detected as false contour two directions are selected. The two opposite directions are calculated which are orthogonal to the direction of the false contour.

Step 2: Regions to be smoothed by a bi-directional filter are expanded along the two directions determined in step 1 up to X (or Y) pixels, which is defined as the maximum expanding length along the horizontal (or vertical) direction. We experimentally set X (or Y) to 10. If there are edges or textures, regions to be smoothed by a bidirectional filter are limited by edges or textures.

Pseudo code (Second step of false contour reduction: adaptive bi-directional smoothing)

Input: Reconstructed image by the first step (output of NN I_f')

Binary map $B_{f,d}$ denoting the position of the pixel to be filtered in the second step

Maximum expansion length X (or Y) along the x (or y) direction from the pixel detected as false contour

Local: Update term I_f' used in adaptive bi-directional smoothing

Output: False contour enhanced image $\hat{I}_f$ output of the second step: final result)

(Initial pixel value of $\hat{I}_f$ is the same as I_f')

FOR x upto X

FOR y upto Y

IF $B_{f,d}$ = TRUE **THEN**

$\hat{I}_f \leftarrow \hat{I}_{f\ +\Delta}$

ELSE

$\hat{I}_f \leftarrow I_f$'

ENDIF

ENDFOR

ENDFOR

Figure.6. Pseudo code for the second step of false contour reduction (adaptive bi-directional smoothing).

The problem of classifying a large number of economic activities descriptions from free text format every day is a huge challenge for the Brazilian governmental administration. This problem is crucial for the long term planning in all three levels of the administration in Brazil. Therefore, an either automatic or semi-automatic manner of doing that is needed for making it possible and also for avoiding the problem of subjectivity introduced by the human classifier.

To our knowledge, this is one of the first few initiatives on using probabilistic neural network for text categorization into a large number of classes as that used in this work and the results are very promising. One of the advantages of probabilistic neural network is that it needs only one parameter to be configured. In addition, the BBPSO employed is an almost parameter free algorithm, just the number of particles needs to be specified. A direction for future work is to boldly compare the probabilistic neural network performance against other multi-label text categorization methods. Examining the correlation on assigning codes to a set of descriptions of economic activities may further improve the performance of the multi-label text categorization methods. We are planning on doing that in future work.

III.WORKING

A. PSO Algorithm

In this work, we presented an experimental evaluation of the performance of Probabilistic Neural Network on multi-label text classification. We performed a comparative study of probabilistic neural network trained by PSO and BBPSO, using a multi-label dataset for the categorization of free-text descriptions of economic activities. The approach using PSO and BBPSO were compared with GA and it was noted that there weren't significant differences among them.

For each particle
Initialize particle
END
Do
For each particle
Calculate fitness value
If the fitness value is better than the best fitness value (pBest) in history set
current value as the new pBest

536

End
Choose the particle with the best fitness value of all the particles as the gBest
For each particle
Calculate particle velocity according equation (1)
Update particle position according equation (2)
End
While maximum iterations or minimum error criteria is not attained

The larger display devices, the more noticeable artifacts such as false contours, block artifacts, and other types of noises. This paper proposes a false contour reduction algorithm using neural networks (NNs) and adaptive bi-directional smoothing. In this project, the adaptation of network weights using Particle Swarm Optimization (PSO) was proposed as a mechanism to improve the performance of Artificial Neural Network (ANN) in modelling a image false contour reduction Technique.

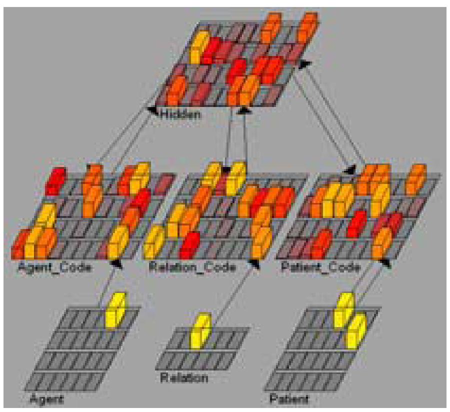

Figure. 7.Training a neural network using PSO

The false contour reduction part is composed of two steps: NN processing and bi-directional filtering. In the first step, false contours are reduced by pixel wise processing using NNs. In the second step, bi-directional smoothing is applied to a neighbouring region of the false contour. PSO is proposed to allow automatic update of network weights to increase the adaptability to dynamic environment. The results obtained in this paper confirmed the potential of PSO-based ANN model to successfully model false contour reduction process.

We have to train the neural network by using PSO algorithm and our main aim is to reduce the error in the image and the error is minimized by using the PSO algorithm.When we compare to the other areas like genetic algorithm and back propagation algorithm Pso gives minimum error criteria.

This is the figure which has minimum error ratio and here this is the advantageous one when we compare to other types of algorithm.

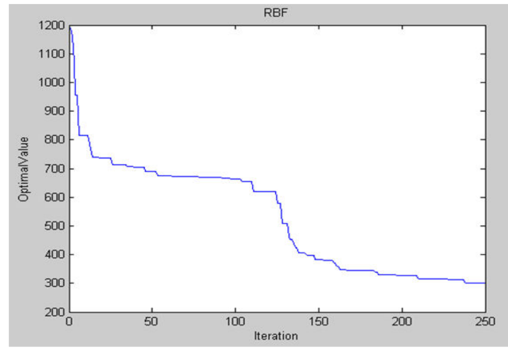

Figure.8. PSO tuned RBF

IV. RESULTS

We show the effectiveness of the proposed false contour reduction algorithm with PSO by using the following figures. The proposed method gives better performance than conventional false reduction methods in terms of result images, edge maps detected by Sobel masks using the same threshold, PSNRs, SSIMs, and computation times.

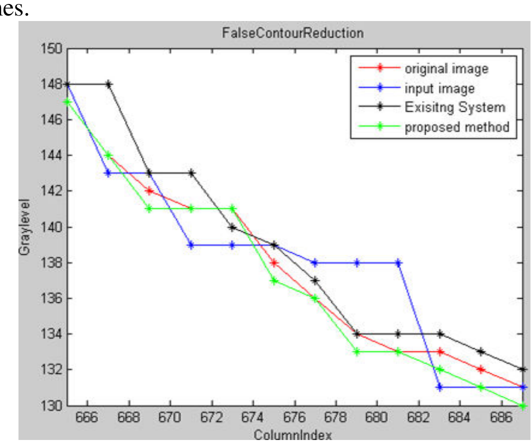

Figure. 9. False contour reduction ratio of gray level to column index

The above figure shows the ratio of false contour reduction gray level to column index. The red color shows the original image. The blue color shows the input image. The black color shows the existing system and the green color shows the proposed method.

Here we have to prove that our method has better performance when compare to the literature survey. RBF trains faster than a MLP and Another advantage that is claimed is that the hidden layer is easier to interpret than the hidden layer in an MLP. Although the RBF is quick to train. PSO particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position.

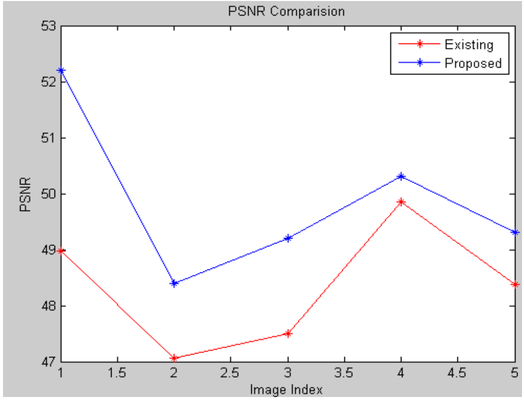

Figure.10. PSNR Comparision ratio

The above figure shows the comparison between PSNR with image index.It shows the results of both existing system and proposed system.

V. CONCLUSIONS

In this paper,a false contour reduction algorithm using neural networks (NNs) and adaptive bi-directional smoothing.The adaptation of network weights using Particle Swarm Optimization (PSO) was proposed as a mechanism to improve the performance of Artificial Neural Network (ANN) in modeling a image false contour reduction Technique. The false contour reduction part is composed of two steps: NN processing and bi-directional filtering. In the first step, false contours are reduced by pixelwise processing using NNs.

In the second step, bi-directional smoothing is applied to a neighboring region of the false contour. PSO is proposed to allow automatic update of network weights to increase the adaptability to dynamic environment. Experimental results in various aspects show that the proposed algorithm efficiently reduces false contours in both real and synthetic images. The proposed algorithm can be applied to artifact reduction caused by quantization in various display devices.The results obtained in this paper confirmed the potential of PSO-based ANN model to successfully model false contour reduction process.

VI.ACKNOWLEDGMENT

This work has been produced as a part of my Master of Engineering project, for the award of degree of ME in Applied Electronics, Anna University, Tirunelveli. This work has been done under the guidance of Ms. A.Selva Anuja, Lecturer, and PSNCET.

VII. REFERENCES

[1] T. Mitsa and K. J. Parker, “Digital halftoning using a blue noise mask,”in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, vol.4, pp. 2809–2812, Toronto, Ont., Canada, Apr. 1991.

[2] S. Daly and X. Feng, “Bit-depth extension using spatiotemporal microdither based on models of the equivalent input noise of the visual system,” in Proc. SPIE Color Imaging VIII, vol. 5008, pp. 455–466, Santa Clara, CA, Jan. 2003.

[3] S. Daly and X. Feng, “Decontouring: Prevention and removal of false contour artifacts,” in Proc. SPIE Human Vision and Electronic Imaging IX, vol. 5259, pp. 130–149, San Jose, CA, June 2004.

[4] J. W. Lee, B. R. Lim, R.-H. Park, J.-S. Kim, and W. Ahn., “Two-stage false contour detection using directional contrast features and its application to adaptive false contour reduction,” IEEE Trans. Consumer Electronics, vol. 52, no. 1, pp. 179–188, Feb. 2006.

[5] H.-R. Choi, J. W. Lee, R.-H. Park, and J.-S. Kim, “False contour reduction using directional dilation and edge-preserving filtering,” IEEE Trans. Consumer Electronics, vol. 52, no. 3, pp. 1099–1106, Aug. 2006.

[6] D. de Ridder, R. P. W. Duin, P. W. Verbeek, and L. J. van Vliet, “The applicability of neural networks to non-linear image processing,” Pattern Analysis and Applications, vol. 2, no. 2, pp. 111–128, June 1999.

[7] S. Zhang and E. Salari, “A neural network based nonlinear filter for image enhancement,” Int. Journal of Imaging Systems and Technology, vol. 12, no. 2, pp. 56–62, Apr. 2002.

[8] M. Egmont-Petersen, D. de Ridder, and H. Handels, “Image processing with neural networks–A review,” Pattern Recognition, vol. 35, no. 10, pp. 2279–2301, Oct. 2002.

[9] K. L. Dinh and C.-T. L. Dinh, “Cascaded neural network-based S-VHS restoration,” IEEE Trans. Consumer Electronics, vol. 53, no. 2, pp. 513–518, May 2007.

[10] H.-S. Lee, R.-H. Park, and S. Kim, “Adaptive video enhancement using neural network,” IEEE Trans. Consumer Electronics, vol. 55, no. 3, pp. 1637–1644, Aug. 2009.

[11] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Imagequality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.

[12] Z. Wang and A. C. Bovik, “A universal image quality index,” IEEE Signal Process. Letters, vol. 9, no. 3, pp. 81–84, Mar. 2002.

[13] A. M. Eskicioglu and P. S. Fisher, “Image quality measures and their performance,” IEEE Trans. Communications, vol. 43, pp. 2959–2965,Dec. 1995.

538