*International Journal of Research in Advent Technology, Vol.2, No.5, May 2014*
*E-ISSN: 2321-9637*

# Use of Artificial Neural Network for Restoration of Digitized Paintings (Review)

Rakshama J Bhingi [1], H.G.Virani [2]
*Dept of Electronics and Telecommunication Goa College of Engineering [1, 2, 3, 4], affiliated to Goa University [1, 2]*
*Email: rbhingi@gmail.com[1] , virani@gec.ac.in[2]*

**Abstract**- This paper is a review paper on the use of artificial neural network for restoration of digitized paintings. Here we have studied MRBF which is used to separate the detected cracks from brush strokes which are identified as cracks.

**Index Terms**- restoration , hue ,saturation,LVQ,MRBF; RBF; PSNR;

## 1. INTRODUCTION

The restoration of digitized paintings gives art historians and museum curators clues as to how the painting originally appeared. This can then be used as a nondestructive tool in actual restoration of the paintings. There three steps involved in the restoration of paintings: (1) detection of cracks, (2) separation of brush strokes that have been misidentified as cracks, and (3) filling of cracks (interpolation using filters. The filters interpolate the cracks using information from the neighboring pixels.This paper is a review on separation of cracks from brush strokes that are misidentified as cracks using artificial neural networks function like median radial basis function.

## 2. SEPARATION OF THE BRUSH STROKES FROM THE CRACKS

### 2.1 *Discrimination on the Basis of Hue and Saturation*

Hue H is associated with the dominant wavelength in a mixture of light wavelengths and represents the dominant color. In the HSV color model, hue is represented as the angle around the vertical axis, with red at 0, green at 120, and so on. Saturation S refers to the amount of white light mixed with a certain hue. Hue and saturation are defined similarly in other related color domains, e.g., in hue saturation intensity (HSI) or hue lightness saturation (HLS).

The hue of the cracks usually ranges from 0 to 60. On the contrary, the hue of the dark brush strokes varies, as expected, in the entire gamut [0 , 360 ]. Furthermore, crack saturation usually ranges from 0.3 to 0.7, while brush-stroke saturation ranges from 0 to 0.4. Thus, on the basis of these observations, a great portion of the dark brush strokes, falsely detected by the top-hat transform, can be separated from the cracks. This separation can be achieved by classification using a median radial basis function (MRBF) neural network, which is a robust, order statistics based, variation of radial basis function (RBF)networks

RBFs are two-layer feed forward neural networks , that model a mapping between a set of input vectors and a set of outputs. The network architecture is presented in Fig. 1. RBFs incorporate an intermediate, hidden layer where each hidden unit implements a kernel function, usually a Gaussian function

$$\phi_j(\boldsymbol{X}) = \exp\left[(\boldsymbol{\mu}_j - \mathrm{X})^T \boldsymbol{S}_j^{-1} (\boldsymbol{\mu}_j - \mathrm{X})\right], \quad j = 1, \ldots, L$$

Where $\boldsymbol{\mu}_j$ , $\boldsymbol{S}_j$ denote the mean vector and the covariance matrix for kernel and denotes the number of units (kernels) in the hidden layer. Each output consists of a weighted sum of kernels. In typical situations that involve pattern classification, the number of outputs equals to the number of classes. In such a setting, the current vector is assigned to the class associated with the output unit exhibiting the maximum activation (winner takes all approach). After the learning stage, the network implements the input-output mapping rule and can generalize it to input vectors not being part of the training set.

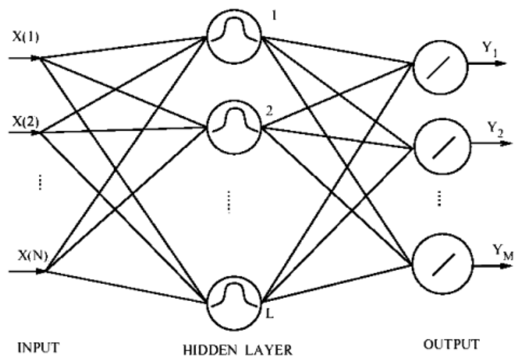

Fig.1

The parameters to be estimated (learned) in a RBF network are the center (mean) vector $\boldsymbol{\mu}_j$ and the covariance matrix $\boldsymbol{S}_j$

for each Gaussian function and the weights $w_{k,j}$ corresponding to the connections between neurons in the hidden layer and output nodes. A hybrid technique that has been frequently used for the training of such networks, has been adopted for the learning stage. According to this technique, training is performed in two successive steps: the hidden layer parameters are estimated using an unsupervised approach and, afterwards, the output layer weights are updated in a supervised manner, using the (now fixed) hidden layer parameters evaluated in the previous step.

In the classical version of the adopted training technique, a variation of the learning vector quantizer (LVQ) algorithm is used for the unsupervised hidden layer parameter updating. Each input vector is assigned to the Gaussian kernel whose center is closer (in terms of either the Euclidean or the Mahalanobis distance) to this vector

$$if\ \|\mathbf{X}_i - \boldsymbol{\mu}_j\| = \min_{k=1}^{L} \|\mathbf{X}_i - \boldsymbol{\mu}_k\|\ then\ \mathbf{X}_i \in C_j$$

where $\|\ \|$ denotes either Euclidean or Mahalanobis distance and $C_j$ denotes the class of input vectors associated with $\phi_j$ kernel .Subsequently, the algorithm updates the center and covariance matrix of the winner kernel using running versions of the classical sample mean and sample covariance matrix formulas. On the other hand, the MRBF algorithm which has been used in our case is based on robust estimation of the hidden unit parameters.It employs the marginal median LVQ that selects the winner kernel using first equation and utilizes the marginal median of the input vectors currently assigned to this kernel for the update of the center vector (location parameter) $\boldsymbol{\mu}_j$ of the kernel

$$\boldsymbol{\mu}_j = \text{marginal\_median}\{\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_{n-1}\}$$

Where $X_{n-1}$ is the last vector assigned to kernel j . The update of the diagonal elements of the corresponding covariance matrix is performed using the median of the absolute deviations (MAD) of the inputs currently assigned to this kernel

$$\boldsymbol{\sigma}_j = \frac{\text{marginal\_median}\{\ \mathbf{X}_0 - \boldsymbol{\mu}_j|, \ldots, |\mathbf{X}_{n-1} - \boldsymbol{\mu}_j|\}}{0.6745}.$$

In the previous expression, $\boldsymbol{\sigma}_j$ denotes the vector containing the diagonal elements of the covariance matrix and $\mathbf{X}|$ denotes the vector obtained by taking the absolute value of each component of . The off-diagonal components of the covariance matrix are also calculated based on robust statistics. In order to avoid excessive computations the above operations can be applied on a subset of data extracted through a moving window that contains only the last W data samples assigned to the hidden unit j.

In the supervised part of the learning procedure, the weights of the output layer, which group the clusters found by the hidden layer into classes, are updated. The update mechanism for these weights is described by the following expression:

$$w_{k,j}(t+1) = w_{k,j}(t) + n_w\left(F_k(\mathbf{X}) - Y_k(\mathbf{X})\right) \times Y_k(\mathbf{X})\left(1 - Y_k(\mathbf{X})\right)\phi_j(\mathbf{X})$$

For k=1,…., M, j=1,…, L, and a learning factor $n_w$ ε (0,1]. In the previous formula $Y_k(\mathbf{X}),\ F_k(\mathbf{X})$ denote the actual and desired network output for input vector . The latter is given by

$$F_k(X) = \begin{cases} 1, & \text{if } \mathbf{X} \in C_k \\ 0, & \text{if } \mathbf{X} \notin C_k \end{cases}.$$

The update formula corresponds to the back propagation for the output weights of a RBF network with respect to the mean square error cost function.

In implementation, a MRBF network with two outputs was used. The first output represents the class of cracks while the second one the class of brush strokes. Input vectors were two-dimensional and consisted of the hue and saturation values of pixels identified as cracks by the top-hat transform. The number of clusters (hidden units) chosen for each class depends on the overlap between the populations of cracks and brush strokes. If there is a substantial overlap, the number should be increased, in order to reduce the classification error. In our implementation three hidden units have been incorporated. Training was carried out by presenting the network with hue and saturation values for pixels corresponding to cracks and crack-like brushstrokes. In order to select pixels corresponding to cracks and crack-like brush strokes the crack detection algorithm was applied on these paintings.

## 3. RESULTS

The results were generated using MATLAB R2012a.The MRBF algorithms were implemented on three images Below are the results

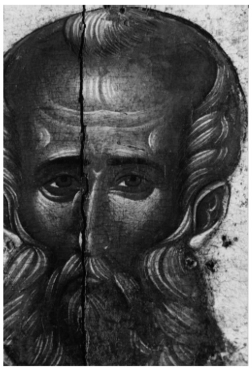

**Figure 1-Image with cracks**

Figure 1 –Top Hat transform result

Figure 1 –Separated brush strokes after MRBF application

## 4. CONCLUSION

This paper presents a review on the use of neural networks for digital restoration of images. Crack separation from brush strokes misidentified as cracks is the second step in restoring t digitized paintings. In this paper we have reviewed the MRBF function for separation of cracks and brush strokes. This methodology can thus be applied in virtual restoration of paintings.

## ACKNOWLEDGMENT

Rakshama J Bhingui would like to thank Dr. H.G.Virani and Prof. Shajahan Kutty for their continual support and guidance.

## REFERENCES

[1] Digital Image Processing Techniques for the Detection and Removal of Cracks in Digitized Paintings by Ioannis Giakoumis, Nikos Nikolaidis, Member, IEEE, and Ioannis Pitas, Senior Member, IEEE- IEEE transactions on image processing, vol. 15, no. 1,January 2006, pg 178-188

[2] M. Barni, F. Bartolini, and V. Cappellini, "Image processing for virtual restoration of artworks," IEEE Multimedia, vol. 7, no. 2, pp. 34–37, Jun 2000

[3] F. Abas and K. Martinez, "Craquelure analysis for content-based retrieval," in Proc. 14th Int. Conf. Digital Signal Processing, vol. 1, 2002, pp. 111–114.