2008 International Conference on Computer Science and Software Engineering

# Analysis and Classification of Remote Sensing, by using Wavelet Transform and Neural Network

Shaker K. Ali , *Zou Beijie
School of Information Science and Engineering, CSU
Changsha, 410083, China
shalynar@hotmail.com, *bjzou@vip.163.com

*Abstract*–**In this paper, we analysis textures of remote sensing images by taking two reference remote sensing images. We employ the wavelet transform and neural network for analysis and classification respectively. We use (symmlet5) and (cioflet1) mother functions for analyzing the two images, that contains water, forest and earth. The images are gray level and (128×128) size. The processing is carried out to divide each image into (16) blocks with size (32×32). Each block will be entered to the wavelet mother function, after trying several mother functions, we found that the (Coif1, Sym5) are the best choice. The results are passed to the features extraction (mean, standard deviation, and variance) and the output is then fed as input to the neural network(NN). Finally the result from NN with (Levenberg Marquardt (LM) algorithm) gives the type of texture (forest , earth ,and water ).**

***Keywords: remote sensing, wavelet transform, symmlet5, cioflet1, and LM algorithm***

## I. Introduction

Texture analysis plays a great role in remote sensing, military applications, and security. In the last ten years the wavelet has diffused in to most texture analysis. Wavelet transform (WT) is very powerful model for texture discrimination. The (WT) decomposes a texture image into a set of frequency channels with narrower band width in the low frequency regions. The WT is suitable for textures consisting primarily of smooth components, so that information is concentrated in the low frequency region [1].

The WT plays substantial role in multi-resolution technique, particularly in the last decade it was used in several digital image processing applications [2]. WT plays important role in the image processing analysis, cheaply in texture recognition of data, for its fine result when using multi-resolution modeling [3].

The importance of the wavelet as a multi-resolution technique results from its decomposition of image into multilevel of independent information with changing scale such as in a geographical map where image has non-redundant information due to changing scale [4,5]. This approach of decomposition process provided us with a number of unrealizable features in the original image which appear in their levels after the application of the transform that deal with image, sound, or any other pattern since it provides a powerful (time-frequency ) representation like a music tones. It points out the time that the observer has this tone or music

sound [6,7].We will discus the wavelet transform in section 2, the features extraction in section 3 neural network in section 4 , the obtained results in section 5, and conclusion are drawn in section 6.

## II. Wavelet Transform.

The DWT consists of applying a coefficient, first to a full data vector of length N, then to “smooth” vector of length N/2, then to “smooth-smooth” vector of length N/4, until only a trivial number of “smooth-……smooth” components (usually 2) remain [9].

The procedure is sometimes called “a pyramidal algorithm”, for obvious reasons. The output of the DWT consists of these remaining components and all the “detail” components accumulate along the way. If the length of the data vector is higher power of the two, there would be more stages of applying A ( or any other wavelet coefficients) and permutation. The end point will always be a vector with two S’s and a hierarchy of D’s, D’s , d’s, etc. It is noticed that once d’s are generated, they simply propagate through to all subsequent stages.

Mallat, developed the multiresolution architecture, which is very suitable for image analysis. Fig.1 show these subbands (channels). If take image of N*N and decompose it into 4 subimage of N/2*N/2 in the subbands decomposition of level 1 which are LL, LH, HL, and HH bands, the subimage of the LL band is the coarse image of the original image. Similarly, the LL band is decomposed into the LL, LH, HL and HH subbands of N/4*N/4 in level 2. Finally, the LL band in level 2 is decomposed into the LL, LH, HL, and HH subbands of N/8*N/8 in level 3.

Therefor, the LL band in level 3 is coarser than the LL band in level 2 and so on. The edge in level 3 is the coarsest edge and the edge in the level 1 is the finest edge. It is easy to know that the edge in the LH, HL, and HH band in level 3 should be highly correlated with

978-0-7695-3336-0/08 $25.00 © 2008 IEEE
DOI 10.1109/CSSE.2008.464

the edge estimated in the LL band in level 3. Therefore, the edges of the LL band in level 3 can be used to predict the edges of the HL, LH, and HH bands in level 3. The edges in the LH, HL, and HH band can also be used to predict the edges of their corresponding bands in level 2 and so on [5].

Since the scaling and wavelet functions fairly separable, each convolution breaks down into one_ dimensional convolution on the rows and columns of f(i,j) which is N*N dimension. At stage , the rows of the image f(i,j) first convolve with Low_ pass filter and with High_ pass filter. The columns of each of the N/2*N array are then convolved with Low_ pass filter and with High_ pass filter. The result is that the four N/2*N/2 arrays are required at the stage of the transform. The transform process can be carried to m stages, where the integer m<= min(log2(N)) for an N*N pixel image .

In order to explain the algorithm, which has been discussed above, the following example, is given: Taking a matrix of a finite set of N*N (in the matrix shown below of 8*8 pixels), where N is a power of 2, this value will be referred to as the input block size. Each row will pass through two convolution functions and each creates an output stream that is half the length of the input row. These convolution functions are filters; one half of the output is produced by the "low_ pass"

$$L_i = \frac{1}{2}\sum_{j=1}^{N} c_{j+1-2i} * f_j . i = 1,2..............N/2 \tag{1}$$

$Let\ N = 8 \Rightarrow i = 1,2,3,4.$

While the other half is produced by "high_ pass" filter function:

$$H_i = \frac{1}{2}\sum_{j=1}^{N} (-1)^{j+1} c_{2i-j} * f_j , i = 1,2,......... N/2 \tag{2}$$

Where

N: is the input row size.

C`s: are the coefficients (in this example c0=c1=1),

L, H: are the output functions.

In many situations, the Low_ pass output contains most of the "information content" of the input row. The High_ pass output contains the differences between the true input and the value of the reconstructed input, if it is to be reconstructed from only the information given in the Low_ pass (detail) output.

The output of the Low_ pass filter consists of the average of every two sample, and the output of the High_ pass filter consists of the difference of two samples.

## III. Feature extraction

Most systems perform feature extraction as a pre-processing step, in obtaining global image features like color histogram or descriptors like shape and texture.Texture features have been modeled on the marginal distribution of wavelet coefficients using generalized Gaussian distributions, in [11]

After the wavelet decomposition of the texture we can extract some important features that come from the use of ciof1 and sym5 by using Mean, Standard Deviation and Variance:

### *A. **Mean***

The mean is implemented as following [5].

$$mean = \frac{1}{RS}\sum_{r=0}^{R-1}\sum_{s=0}^{S-1} f(r,s) \tag{3}$$

### *B. **Standard Deviation (STD)***

The STD is implemented by using the equation below .

$$STD = \sqrt{\frac{\sum_{r=0}^{R-1}\sum_{s=0}^{S-1}(f(r,s) - mean)^2}{R * S}} \tag{4}$$

Where *f(r,s)* is the value of the pixel in this position.

### *C. **Variance***

The Variance is implemented by using this equation(Vr) .

$$Vr = STD^2 \tag{5}$$

Where R,S are the length of rows and columns of image respectively.

## IV. Neural Network

We use the neural network for recognizing the type of texture(output) after taking the input from the feature extraction, we use the Levenberg-Marquardt Algorithm[2] as training algorithm as shown below

### *A. **Algorithm***

*a) Initialize network (weights and biases).*

*b) For each training pair. Do (3-7) UNTILL performance criterion.*

*c) Sums weighted input and apply activation function to compute output signal.*

$$h_{oj} = \sum_{i=1} w_{ij} x_i + b_i \Rightarrow h_j = f(h_{oj}) \tag{6}$$

*where* $b_i$ *is biases*

*d) Compute the output of the network.*

$$yy = b_p + \sum w_{pi} h_i \Rightarrow y = f(yy) \tag{7}$$

*where* $w_{pi}$ *is weights*

*e) Calculate the error term*

$\varsigma = y - yd$ (8)

*where the y is Network output ,yd is desired output*

*f) Calculate correction term.*

$wb = [w_1 b_1 + w_2 b_2 + \ldots\ldots w_p b_p]$ (9-a)

$\Delta wb = (J^T * J + \eta I) * (-J^T \varsigma)$ (9-b)

*where J Jacobean matrix , η is learning factor*

*g) Update the weights and biases.*

$W_{ij}(new) = w_{ij}(old) + \Delta wb.$ (10)

*Where:*

- $J^{T:}$ *Jacobean matrix of proper dimension.*
- $J^T * J$: *covariance matrix of proper dimension.*
- $\Delta w$: *correction matrix of proper dimension.*
- $\eta$: *learning parameter* $1 >= \eta > 0$.
- *I: identity matrix.*
- $\varsigma$: *Difference between network output and desired output.*

## V. . Result

The texture image will be passed to the wavelet mother function which will segment the texture into several subbands, as shown in Fig.2 and Fig.3. This information will be fed to the feature extraction (FE) mean, standard deviation and variance (MSV) .The output of (MSV) will be input to the artificial neural network (ANN).The output of ANN will determine the type of texture (or the percentage of similarity ) from texture such as (2/3: earth texture), or ( 0: forset texture) or (1/3: water texture) depending on the illustruted results in table (I), the size of training NN is (3×3×3=27) inputs and (3×3×3=27) outputs, because the input comes from three (mean, std and variance ) and there are three textures and two hidden layers, we have three values of features extraction that will input to the ANN for three types of textures , but the values not always come from the same texture, so we must make combination between the textures this make 3×3×3=27 as show below:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 |
| 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| 2 | 2 | 2 | 3 | 3 | 3 | 1 | 1 | 1 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 |
| 1 | 2 | 3 | | | | | | | | | |
| 3 | 3 | 3 | | | | | | | | | |
| 3 | 3 | 3 | | | | | | | | | |

This values of input combination makes the following output:

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 10 | 10 | 10 | 25 | 0 | 10 | 0 | 75 | 10 | 25 | 0 |
| 25 | 55 | 25 | 25 | 0 | 25 | 75 | 10 | 0 | 75 | 0 | 25 |
| 75 | 75 | 555 | | | | | | | | | |

The output 5 means that the inputs have 3 ones [1 1 1], the output 10 means that there are two ones [1 $\phi$ 1] or [$\phi$ 1 1] or [1 1 $\phi$] , (*where* $\phi$ *is any number*) the output 55 means that there are three 2 input [ 2 2 2], the output 25 means that there are two 2 input value[2 2 $\phi$] or [2 $\phi$ 2] or [$\phi$ 2 2], the output 0 means that there are three different values [1 2 3] or [3 2 1] or [2 1 3], the output 75 means that there are two 3 input value [$\phi$ 3 3] or [3 $\phi$ 3] or [3 3 $\phi$] and the output 5 5 5 means that there are three 3 input values [3 3 3] . The training of the NN has (20,49s) of time for each training, the training needs: 32/250 epochs the sum-squared error (SSE) = (5.00565e-015), and the number of flops = (325097666).

When we used the mean as a (FE) we found that $LL_1$, $LL_2$, $LL_3$ are only positive and have greater values, but the other sub-bands have negative, but when we use the (STD) as a (FE) Here only $LL_1$, $LL_2$ and $LL_3$ not used as feature for classification, because in these sub-band its values are greater from the other positive that may cause interference between the types of textures $HL_1$, $HH_2$ and $HH_3$ are used for features classification, since these sub-bands are only stable as shown in Fig.4 and Table (II).

The result from two texture (forest and earth ) using coiflte1 and symylte5 as shown in table (I).

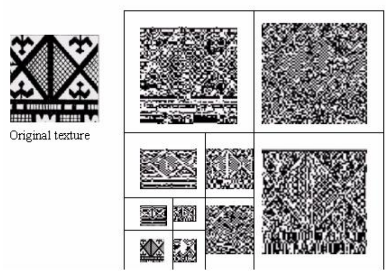


Figure 1. Illustration the 3rd level of wavelet decomposition

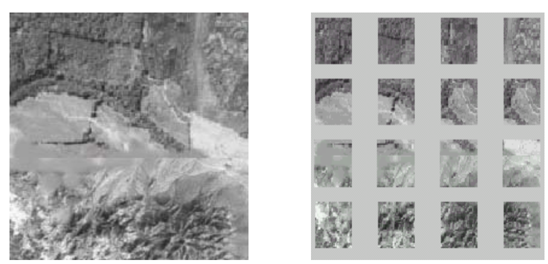

Image1 divided image1

Figure 2. Illustration of r emote sensing image

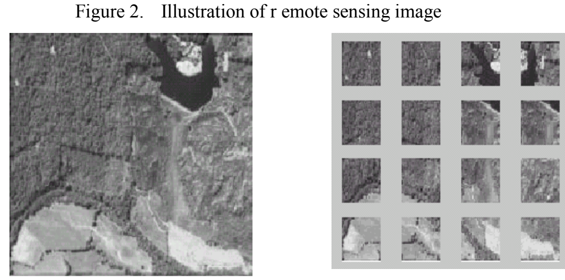

Image2 divided image2

Figure 3. Illustration of remote sensing image

TABLE I. LLUSTRATION THE RESULT FOR FOREST AND EARTH TEXTURES BY USING CIOF1AND SYM

| Level No | result for forest texture by using ciof1 | | result for forest texture by using sym5 | | result for earth texture by using ciof1 | | result for earth texture by using sym5 | |
|---|---|---|---|---|---|---|---|---|
| | ***Mean*** | ***STD*** | ***Mean*** | ***STD*** | ***Mean*** | ***STD*** | ***Mean*** | ***STD*** |
| LL1 | 176.2523 | 25.0155 | 165.7322 | 47.0475 | 156.1266 | 60.4618 | 264.6184 | 133.4736 |
| LH1 | 0.0128 | 6.0211 | 0.0121 | 8.8891 | 0.0114 | 6.8784 | 0.3788 | 28.1443 |
| HL1 | -0.0599 | 5.2934 | -0.0564 | 10.1572 | -0.0531 | 7.413 | -0.2002 | 25.566 |
| HH1 | -0.0172 | 2.7644 | -0.0162 | 2.0746 | -0.0152 | 1.7193 | -0.0531 | 8.1892 |
| LL2 | 352.5046 | 46.6532 | 294.6651 | 133.9215 | 249.9756 | 163.1532 | 423.6827 | 298.7879 |
| LH2 | 0.7405 | 12.9801 | -0.5964 | 18.3971 | 0.5128 | 16.7984 | 3.3059 | 79.4351 |
| HL2 | 0.1306 | 9.7305 | -0.1987 | 20.1013 | 0.0826 | 18.5026 | -1.0597 | 67.1886 |
| HH2 | -0.0144 | 8.043 | -0.0355 | 7.0712 | -0.0067 | 6.8261 | 0.4354 | 41.4367 |
| LL3 | 705.0093 | 89.3336 | 451.2059 | 337.4852 | 341.1765 | 349.5428 | 578.2587 | 602.551 |
| LH3 | 0.8188 | 22.9198 | 0.7419 | 37.793 | -0.5235 | 42.829 | -5.2331 | 138.2198 |
| HL3 | -1.4312 | 10.7463 | -0.684 | 45.5915 | 0.6478 | 54.065 | -3.9223 | 114.9183 |
| HH3 | -0.895 | 10.3379 | -0.197 | 9.6429 | -0.343 | 9.1041 | 3.8606 | 66.7374 |

TABLE II. ILLUSTRATED THE VALUES THAT EXTRACT FROM THE FE,

| Level No | Mean | STD |
|---|---|---|
| LL1 | 176.2523 | 25.0155 |
| LH1 | 0.0128 | 6.0211 |
| HL1 | -0.0599 | 5.2934 |
| HH1 | -0.0172 | 2.7644 |
| LL2 | 352.5046 | 46.6532 |
| LH2 | 0.7405 | 12.9801 |
| HL2 | 0.1306 | 9.7305 |
| HH2 | -0.0144 | 8.043 |
| LL3 | 705.0093 | 89.3336 |
| LH3 | 0.8188 | 22.9198 |
| HL3 | -1.4312 | 10.7463 |
| HH3 | -0.895 | 10.3379 |

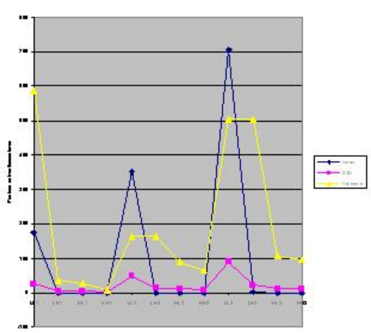

Figure 4. Illustrate the carve for values in table (II)

## VI. . Conclusion

Our experiments reveal that coif1is very efficient to use in remote sensing field as it out performs the result obtained from other filters. However if we want to work on natural texture (the images that are not remote) we can use the Sym5. The Sym5 is better than (Haar, Symmelet2, Coiflet2, Daubechies1)

Using the mean as a (FE) we found that LL1, LL2, LL3 only positive and have greater values, but the other sub-bands have negative. Whereas, using the (STD) as a

(FE) made only LL1, LL2 and LL3 not to be used as feature for classification, this is because in these sub-bands, its values are greater than the other positive and this may cause interference between the types of textures. This, HL1, HH2 and HH3 are used for features classification, because these sub-bands are the only stable ones.

### *References:*


[1] Feacher, I., and Jevis, B., W., "Digital Signal Processing, Apractical Approach" Addison-Wasley com. 1996.

[2] Unser, M., "Texture Classification and Segmentation Using Wavelet Frames" IEEE transaction on image processing, Vol.4, No11, pp.1549-1560, November 1995.

[3] Jianguo Z., and Tieniu T., " Brief of Invariant Texture Analysis Method" Pattern Recognition, Vol.35, pp.735-747, Mach 2002.

[4] Abhinav Mathur, "Wavelet Based Texture Recognition of Forest Canopies", Internet Explorer, 2002.

[5] Wilson, R., "Multiresolution Image Modeling" Electronic and communication Journal, pp.90-96, April 1997.

[8]Hideki Noda, Mahdad N. Shirazi and Eiji Kawaguchi, "MRF-Based Texture Segmentation Using Wavelet Decomposed Images", Pattern Recognition, Vol.35,pp.771-782, April 2002.

[9] William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery, " Numerical Recipes in C ", 2nd Ed., Cambridge university press, 1992

[10] Negia, leasrter S.H. and Jonas sjoberg, "Efficient Training of Levenberg-Marquardt Algorithm" Internet Explorer,1999.

[11] M.N. Do and M. Vetterli," Wavelet –based Texture Retrieval Using Generalized Gaussian Density and Kullback- Leibler Distance" IEEE Trans. Image Processing, 11(2):146-158, 2002.