Using GLCM and Gabor filters for classification of PAN images

Conference Paper · May 2013

DOI: 10.1109/IranianCEE.2013.6599565

2 authors:

Fardin Mirzapour
Tarbiat Modares University

Hassan Ghassemian
Tarbiat Modares University

DOI: 10.1109/IranianCEE.2013.6599565

Using GLCM and Gabor Filters for Classification of PAN Images

Fardin Mirzapour*, Hassan Ghassemian**
Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran
*f.mirzapour@modares.ac.ir
**ghassemi@modares.ac.ir

***Abstract*— In the present research we have used GLCM and Gabor filters to extract texture features in order to classify PAN images. The main drawback of GLCM algorithm is its time-consuming nature. In this work, we proposed a fast GLCM algorithm to overcome the mentioned weakness of traditional GLCM. The fast GLCM is capable of extracting approximately the same features as the traditional GLCM does, but in a really much less time (in the best case, 180 times faster, and in the worst case, 30 times faster). The other weakness of the traditional GLCM is its lower accuracies in the region near the class borders. As Gabor filters are more powerful in border regions, we have tried to combine Gabor features with GLCM features. In this way we would compensate the latter mentioned weakness of GLCM. Experimental results show good capabilities of the proposed fast GLCM and the feature fusion method in classification of PAN images.**

Keywords: *Texture feature, GLCM, fast algorithm, Gabor filters, image classification, satellite images*

I. INTRODUCTION

Features which are commonly used in classification of remote sensing images are categorized in two main groups: spectral features and spatial features. A spectral feature vector of a pixel is a vector whose elements are the reflected energy, from a point in the scene corresponding to the pixel, recorded in different spectral bands. So, spectral feature vector is defined only for colored, multispectral (MS), or hyperspectral (HS) images. On the other side, spatial features of a pixel are the ones which are obtained from processing the gray level values of the pixel and its neighbors in a single-band image. So, this kind of features can be defined for single-band images, such as PAN images, as well as, individual bands of colored, MS, or HS images. Spatial features used in image processing can be divided into two main categories: texture and shape features. Texture features act as a measure of coarseness, size, and directionality of image details, while the latter assesses the shape of these details. However, there is not a clear distinction between the two categories, e.g. features extracted from GLCM matrices are known as texture features while they could be used as shape measures too [1,2].

In the present research we try to utilize two kinds of spatial features to classify single-band images: (1) statistical features extracted from GLCM matrices, and (2) features obtained using Gabor filters.

The outline of the remainder of this paper is as follows. In section2, we review GLCM matrices and Gabor filters, and introduce a fast algorithm for GLCM calculations. In section 3, GLCM and Gabor features are fused to make it possible to use their strength at the same time. Finally, section 4 provides our conclusion.

II. FEATURE EXTRACTION ALGORITHMS

A. GLCM

One of the simplest statistics of a two dimensional image is the information obtained from its one-dimensional histogram, i.e. probability of gray level occurrence. One-dimensional histogram does not consider the relationship between pixels exactly, thus it is not a good texture measure. To overcome this weakness, two-dimensional histogram was introduced [3] which is in fact the probability of occurrence of two different gray levels in the neighborhood of the pixel under investigation. In this approach, the relationship between pixels is considered more accurately, but it is very time consuming. Again to solve this problem, a new approach was proposed in which between-pixel relationships were considered only in a few predefined directions and distances. To be more accurate, for a pixel with (x_c, y_c) coordination placed at the center of its neighborhood window $W(x_c, y_c)$, the (i, j) element of its $GLCM_{d,\theta,(x_c,y_c)}$ matrix is defined as the number of occurrences of pixels with gray levels of j, at the distance d, and at the direction θ of pixels with gray level of i. All these pixels are in the neighborhood window $W(x_c, y_c)$.

$$\begin{cases} f(x_1, y_1) = i \ , \qquad f(x_2, y_2) = j \\ \qquad in\ which \\ (x_2, y_2) = (x_1, y_1) + (d\cos\theta, d\sin\theta) \\ (x_1, y_1)\ and\ (x_2, y_2) \ \in\ W(x_c, y_c) \end{cases} \tag{1}$$

So, for every distance-direction couple, (d, θ), a $G \times G$ $GLCM_{d,\theta,(x_c,y_c)}$ matrix is obtained, in which G is the number of gray levels in the image. After obtaining GLCM matrices for every pixel, some statistics such as mean, standard deviation, and entropy is extracted from the GLCM matrices and are dedicated to the pixel.

The GLCM features are very close to human inference from texture and describe image texture well, but they need strong processing resources. Many approaches are proposed to face this problem. The simplest one is reducing G by resampling the gray level of the image, which causes the dimensions of GLCM matrices to reduce.

Another approach is to consider less distance-direction couples, (d, θ). It is shown that features extracted from GLCM matrices corresponding to "$(d, \theta) = (1, 0)$" are enough to describe texture of most images [3]. Although GLCM concept and features were introduced in more than three decades ago, it is a strong method to extract texture features and still many attempts are being made to improve the speed and performance of the algorithm, or combine it with new methods.

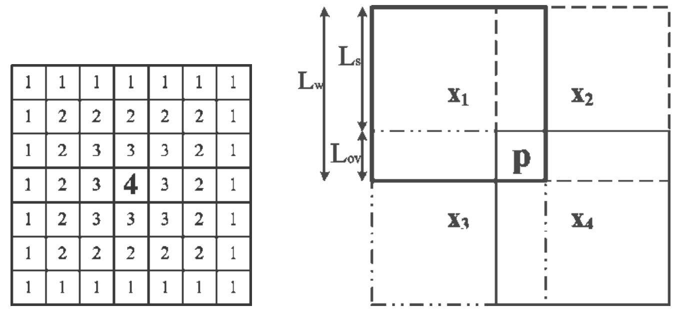
Here, we propose a new way to conquer the resource consuming nature of GLCM while maintaining the strength of the extracted features. In other words, we make it faster while the accuracy of the image classification using these features (as a criterion to assess the quality of the extracted features) is not affected.

The proposed method is based on the concept of correlation of the features of spatially-close pixels, i.e. the pixels which belong to the same neighborhood. The simplest type of dependency would be the linear. With assumption of this type of correlation, we can calculate GLCM matrix (and then extract the related features from it) only for a few pixels by skipping pixels with the step size of L_s in row and column. Then the extracted features for these pixels are assigned to all pixels in their neighborhood with a pyramidal weight matrix ($L_w \times L_w$) as in Fig. 1a. Finally the feature vector associated to each pixel is the sum of all weighted feature vectors from all the weighting windows which include the pixel:

$$f_j(p) = \sum_{i=1}^{n} a_i(p).\ f_j(x_i) \qquad (2)$$

In which $f_j(.)$ is j-th element of the feature vector. x_i is the central pixels of each of the overlapping weighting windows which include the pixel p, and $a_i(p)$ is the weight associated to p point in the window centered at x_i (Fig.1). The number of overlapping windows is n ($n = 4$ in Fig. 1b).

In order to evaluate the proposed GLCM method, we implemented it on two different data: (1) a single-band, 1024x1024-pixel image, containing 5 different textures synthetized from Brodatz set, and (2) a single-band, 512x512-pixel image, containing 8 different textures from a PAN satellite image of Tehran/Iran (Fig. 2). Table I shows the parameters used in the implemented fast GLCM.



(a) (b)

Figure 1. (a) Linear (pyramidal) weighting window for a window length of $L_w = 7$, and (b) four $L_w \times L_w$ weighting windows with overlap of length L_{ov} (features of pixel p are the weighted sum of features of x_i pixels)

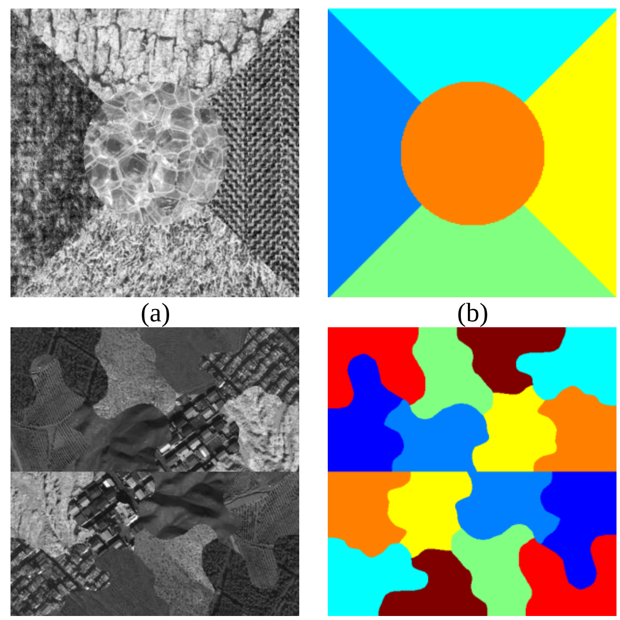

(a) (b) (c) (d)

Figure 2. Single-band images used to evaluate the proposed fast GLCM algorithm: (a) a single-band, 1024x1024-pixel image, containing 5 different textures synthetized from Brodatz set, and (b) its ground truth map; (c) a single-band, 512x512-pixel image, containing 8 different textures from a PAN satellite image of Tehran/Iran, and (d) its ground truth map

TABLE I. PARAMETERS USED IN EVALUATING FAST GLCM ALGORITHM

directions and distances used in GLCM matrix calculations	$(x, y) \rightarrow (x+1, y): (d, \theta) = (1, 0)$ (Equation 1)
number of gray levels of image (G)	$2^{5\text{ bits}}$ = 32 levels GLCM matrices: 32x32
feature extraction from GLCM matrix	Primary features (6 features): Contrast, Correlation, Energy, Homogeneity, Entropy, Variance
	Final features: First, applying PCT to the primary features and then selecting the first 5 components
GLCM extraction window size	33x33
Weighting window (Fig. 1)	33x33 pyramid
Skip length	$L_s = 1 \dots 16$

In Fig. 3, the average ML classification accuracy and the relative processing time are illustrated versus different skip lengths (L_s). The processing times are normalized to the processing time for the case of $L_s = 16$. As can be seen, fast GLCM algorithm can significantly reduce the processing time (by the order of ${L_s}^2$) while preserving GLCM features quality. We have used average ML classification accuracy as the quality measure of features.

A point that should be mentioned here, is the effect of the minimum size of objects (connected areas of the same texture) in an image, on the maximum value of the algorithm parameter (skip length, L_s). It is obvious that the skip length should be less than the dimensions of the smallest connected area of the same texture in the image. Thus a prior knowledge about the image is required to select an appropriate value for L_s. This can be seen comparing the classification accuracy trends of images of Fig. 2a and 2c illustrated in Fig. 3 and 4 respectively: the ratio of the *"the average classification accuracy for the case of $L_s = 16$"* to *"the mean average classification accuracy for all cases of L_s"*, for the image in Fig. 2a is bigger than the ratio for the image in Fig. 2b (99.6% versus 96.9%).

B. Gabor filters

Gabor filters have been used in different areas of image processing such as texture classification, edge detection, fingerprint identification, and image coding [4, 5, 6]. Also, different methods have been developed to use Gabor filters in image classification [5]. In [7], Gabor wavelets are utilized to extract image texture features. The idea is based on detecting linear directional elements.

In this method, a set of wavelets is generated using a mother wavelet as in (3).Then the whole image is used as the input to the wavelet set (Fig. 5). Since the filters output values are complex, we use the magnitude of these values. In order to reduce the inter-class variance, and consequently reducing classification errors, a Gaussian LPF is applied to these values [7].

$$\varphi(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{-\frac{1}{2}\left(\frac{x^2}{{\sigma_x}^2}+\frac{y^2}{{\sigma_y}^2}\right) + 2\pi j U_h x\right\} \quad (3)$$

$$h_{s,d}(x,y) = \left(\frac{U_h}{U_l}\right)^{\frac{-s}{S-1}} .\ \varphi(\mathrm{X},\mathrm{Y}) \quad (4)$$

$$X = \left(\frac{U_h}{U_l}\right)^{\frac{-s}{S-1}} \left[(x-x_0)\cos\left(\frac{d\pi}{N_d}\right) + (y-y_0)\sin\left(\frac{d\pi}{N_d}\right)\right]$$

$$Y = \left(\frac{U_h}{U_l}\right)^{\frac{-s}{S-1}} \left[-(x-x_0)\sin\left(\frac{d\pi}{N_d}\right) + (y-y_0)\cos\left(\frac{d\pi}{N_d}\right)\right]$$

In the above equations, $s = 1 \dots N_s$ and $d = 1 \dots N_d$ are the scale and direction parameters of wavelets (Fig. 6); (x_0, y_0) couple is the filter center coordination in the spatial domain; U_l and U_h are respectively the minimum and maximum center frequency of filters on u axis.

To reduce the number of features and also to reduce information redundancy due to overlapping filters, principal component transform (PCT) is usually applied to the outputs of Gabor wavelets.

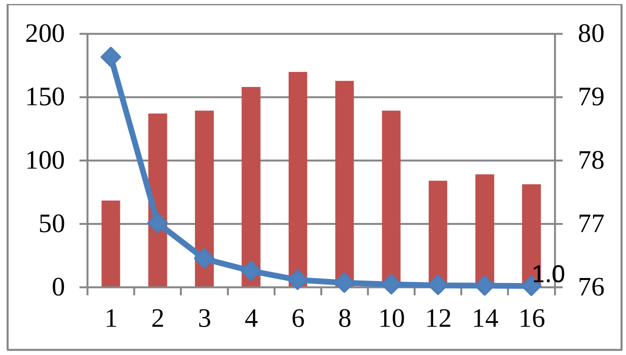


Figure 3. Average ML classification accuracy for the image shown in Fig. 2a using fast GLCM features (the bars with values shown on the right side), and the relative feature extraction time (the curve with values shown on the left side); the horizontal axis shows the fast GLCM algorithm parameter, i.e. skip length (L_s)

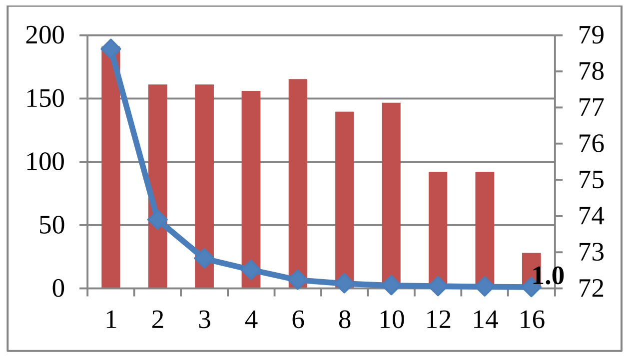


Figure 4. The same as Fig. 3 for the image shown in Fig. 2c

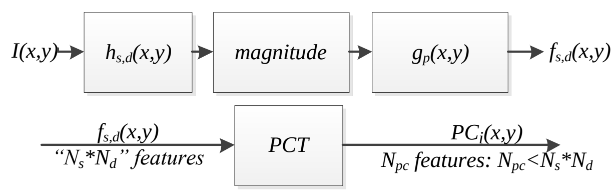


Figure 5. Gabor features extraction process

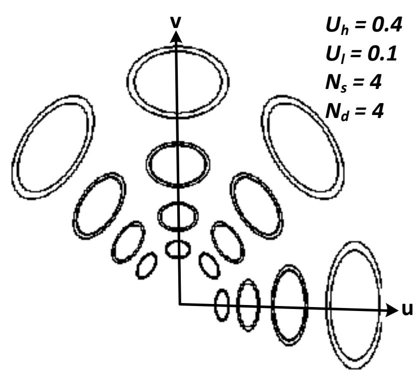


Figure 6. 3-dB cross section of Gabor filters in spatial frequency domain (u,v); for 4 scales and 4 directions

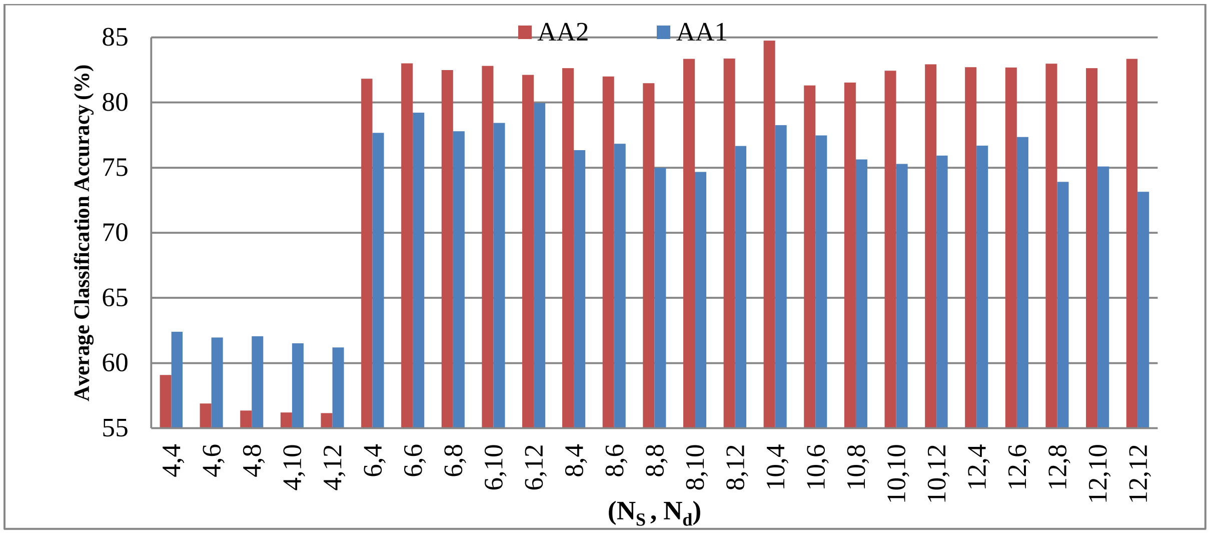


Figure 7. Average ML classification accuracy for the images shown in Fig. 2a and 2c, using Gabor features , for different number of scales (N_s) and directions (N_d)

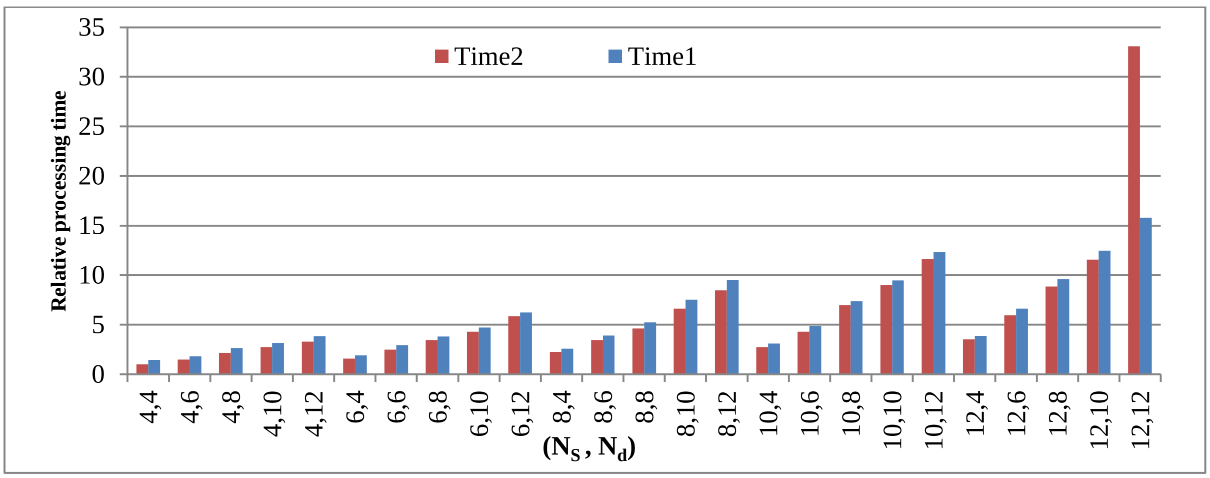


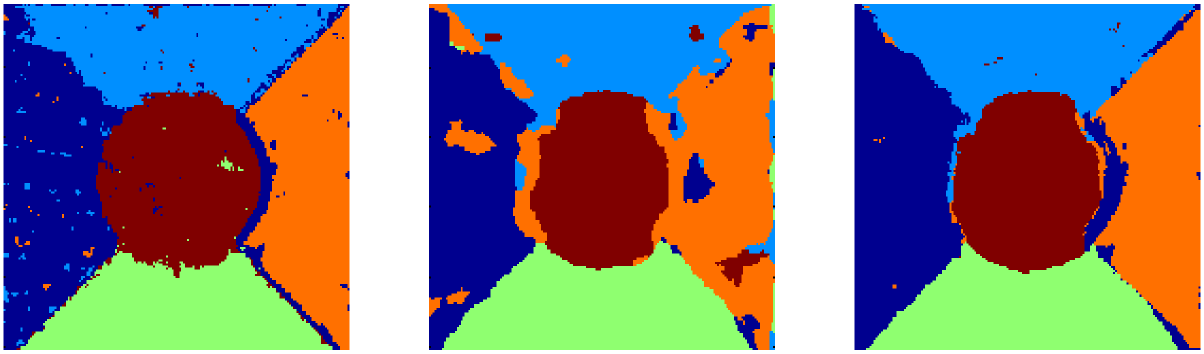
Figure 8. Relative processing time required for extracting Gabor features from the images shown in Fig. 2a and 2c, for different number of scales (N_s) and directions (N_d)

The main characteristic of PCT is that the output components are uncorrelated. Typically, after applying this transform to input feature vectors, a number of the output components (elements of the output vector) which their cumulative sum of energy is bigger than a threshold are preserved and the rest are discarded. The number of preserved components depends on the selected threshold and of course on the input data characteristics.

Two significant parameters in Gabor filters are the number of scales and directions, N_s and N_d. To find the best values for these parameters we classified images shown in Fig. 2a and 2c using Gabor features extracted for different values of N_s and N_d, and applying ML classifier. The average classification accuracy was used as a selection criterion. According to Fig. 7, although the optimal choice of N_s and N_d depend on the type of data, there is an obvious distinction between accuracy values for $N_d \geq 4$ and $N_s \geq 6$. So, the boundary values of $N_d = 4$ and N_s=6, would be an appropriate choice regardless of the type of input data.

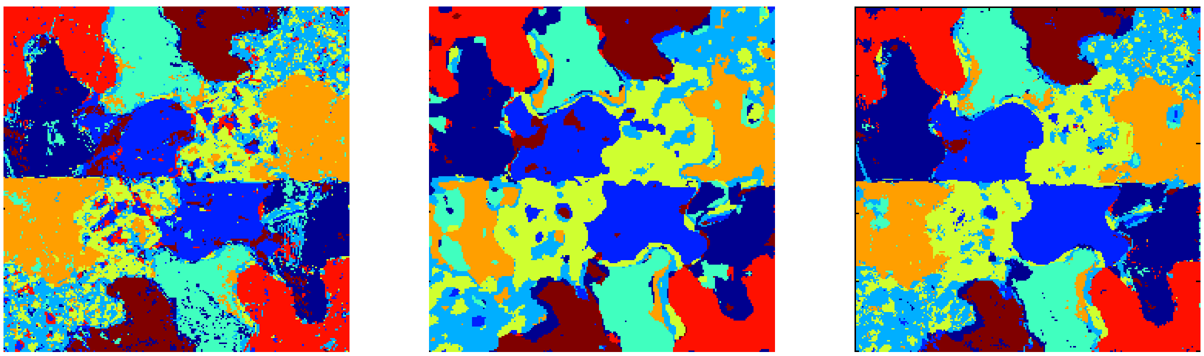
TABLE II. PARAMETERS USED IN GABOR FEATURE EXTRACTION

filter parameters	21x21 window, $U_l = 0.01$, $U_h = 0.49$
number of filters	$N_d \times N_s$; N_d directions and N_s scales
extracted features	Primary features: $N_d \times N_s$ features
	Final features: By applying PCT to the primary features and using a threshold level of 95% (the number of final features, N_{PC}, depends on the selected threshold and characteristics of input features)
Gaussian LPF	21x21 ($\sigma x = \sigma y = 10$)



(a) (b) (c)

Figure 9. ML classification results for the image shown in Fig. 2a, using (a) Gabor features, (b) fast GLCM features, and (c) fused features



(a) (b) (c)

Figure 10. ML classification results for the image shown in Fig. 2c, using (a) Gabor features, (b) fast GLCM features, and (c) fused features

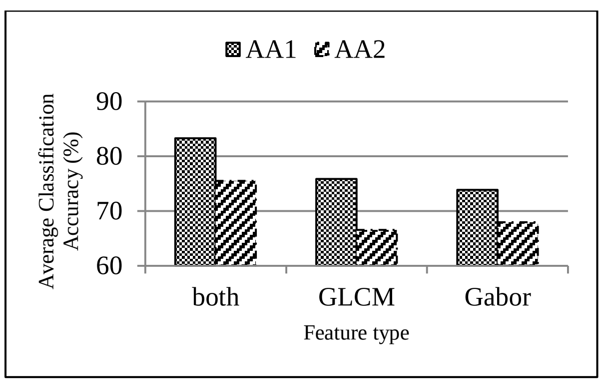


Figure 11. Average classification accuracy of the images shown in Fig. 2a (AA1), and Fig. 2b (AA2); using Gabor, GLCM, and fused features

III. Implementation

In the previous section, we saw that by utilizing the proposed fast GLCM algorithm we were able to overcome the main drawback of GLCM (its slow nature) while benefiting from its strength in extracting texture features. Also, according to the diagrams in Fig. 3, the optimal value for skip length in fast GLCM was $L_s = 6$. If faster processing is required, any L_s value up to 16 could be a good option at the expense of less but acceptable accuracy. Also, we saw that selecting the number of directions and scales of $N_d = 4$ and N_s=6 for Gabor filters could be an appropriate choice.

Then fast GLCM and Gabor features were extracted from the image shown in Fig. 2a. The parameters were selected as shown in tables I and II, and as mentioned in the above paragraph. The results are illustrated in Fig. 9a and 9b. As seen in these figures, Gabor features have good ability to find class boundaries, while the GLCM features are more accurate in areas within the classes. So, it seems that by combining these two types of features, we may be able to use the advantages of both. To verify this idea, we fused both feature vectors by simply stacking them. Then we classified the data using the stacked feature vectors. The results shown in Fig. 9c and 11 confirmed this idea.

Then we tried to implement this fusion technique on the PAN image shown in Fig. 2c. Again, the results were satisfactory and the fusion idea was confirmed.

IV. CONCLUSION

In this paper we tried to utilize two well-known methods for extracting texture features from single-band satellite images: GLCM and Gabor filters. GLCM method, even though have good performance in texture feature extraction, is very time consuming. Here we proposed a fast GLCM algorithm which significantly improved the speed of GLCM: about 180 times faster for images with large connected areas of the same texture (for skip length of $L_s = 16$), and at least 30 times faster for images with small connected areas of the same texture (for skip length of $L_s = 6$). This increase in speed was obtained while preserving the quality of extracted features. The average ML classification accuracy using the extracted features was used as a measure of the quality of the features. The classification results showed that Gabor features are more powerful in areas close to class borders, while GLSM features are preferable in the areas within classes. Using these findings, we could test the idea of fusing these two types of features in order to benefit the advantages of both. The implementation results confirmed the idea.

REFRENCES

[1] Materka and M. Strzelecki, “Texture Analysis Methods – A Review,” Brussels , COST B11 report, 1998

[2] Yang Mingqiang and et al., “A survey of shape feature extraction techniques,” Pattern Recognition, Peng-Yeng (ED.), July 2008

[3] R. M. Haralic, “Statistical and Structural Approaches to Texture,” in Proceedings to IEEE, vol. 67, No.5, 1979, pp. 786-804

[4] Weldon, T.P. & Higgins, W.E. & Dunn, D.F., “Efficient Gabor filter design for texture segmentation,”PREPRINT of Pattern recognition, 1996

[5] Weldon, T.P. & Higgins, W.E. & Dunn, D.F., “Gabor filter design for multiple texture segmentation,” Opt. Eng., Vol. 33, No.10, OCT. 1996

[6] Dunn, D.F. & Higgins, W.E. & Wakeley, J., “Texture segmentation using 2-D Gabor elementary functions,” IEEE Trans. Pattern Anal. And Machine Intell., Vol. 16, No. 2, pp. 130-149, 1994

[7] Mirzapour, F, and Ghassemian, H, “Texture Feature Extraction in Satellite Images using Gabor Wavelets,” MVIP2003, Tehran, IRAN, 2003- PRINTED IN PERSIAN