

Robust and Speedy Multikernel Sparse Representation Technique for Image Classification in Remote Sensing

Bhuvaneswari. N. R^{#1}, Dr. Sivakumar.V.G^{*2}

^{#1}ECE Department, Sathyabama University, Semmenchery, Chennai-600 119, India
^{#1}nrbhuvaneswari@gmail.com

^{*2}ECE Department, Sathyabama University, Semmenchery, Chennai-600 119, India
^{*2}sivakumarvg2004@gmail.com

Abstract - In this article, a fast and reliable technique using Multikernel Sparse Representation for Image Classification in Remote Sensing. For higher dynamics remote detecting picture and the limitation of image segmentation, obtaining asatisfying Image classification result is difficult. In this paper, an evolution-based algorithm for contour detection has been developed to create templates from the original image. The templates thus extracted from the original image would act as inputs to the Particle Filter Framework (PFF). Features are extracted and fused using multiple kernels, the fused output is optimized with the Multikernel Sparse Representation technique. To achieve substantial reduction in execution time, template matching algorithm is used. The robustness of the implementation is focused to avoid misclassifications during variations in the illumination, size and direction of the image.

Keyword- Image Classification, Distance Regularization, Particle Filter Framework, Sparse Representation, Multikernel Sparse Representation

I. INTRODUCTION

Remote Sensing Classification is a complex process, due to several factors affecting the accuracy of the information extracted from the remote sensing data. Some of the factors influencing the accuracy of information extraction are; improper data selection, inadequate resolution, noises in the satellite data and cloud cover in the image.[1,2] Besides these factors, an efficient and effective performance of remote sensing image classification is surely a challenging task. The major steps to be followed for achieving an improved classification without much of information loss are; suitable classification system selection, better training sample selection, followed by image pre-processing and feature extraction, and finally the accuracy calculation.

In general, the classification techniques are broadly classified as either pixel-or object-based. Pixel-based classification is less preferred for high-resolution data because it produces some disagreeable classifications in extracting the targeted class. The pixel-based classification approach is proven to be thriving with low to moderate spatial-resolution data. In the case of high-resolution data, the approach produces numerous unacceptable classification accuracies. Also pixel by pixel based classification methods is highly time consuming and consumes more computational cost[3].

To overcome the limitation of the pixel-based classification approach in high-resolution remote sensing, the object-based classification approach is recommended in which the processing units are 'images'. Object-based classification is practical and broadly preferred in overcoming the object variation in remote sensing images which leads to pixel-based misclassification. Object-based classification method can take entire benefit of multi-dimensional features of objects, so it is appropriate for high-resolution remote sensing image[4]. Because of the complication of remote sensing data and the constraint of image segmentation, it is often difficult to attain acceptable segmentation result.

The evolution based algorithm for contour detection with distance regularization[5] is preferred to overcome the difficulty of segmentation. To achieve speed template matching algorithm which is based on DRLSE is used[6]. Sparse representation is an ideal procedure for image classification in remote sensing because of its transparent and powerful capacity to gain a precise information in depth from the data available[7]. To achieve robustness which is a major task in image classification Multikernel sparse representation is carried out.

II. PROPOSED METHOD

The work comprises of the following three steps:

Step 1: Segmentation of initial template

Step 2: Multikernel fusion

Step 3: Optimization Algorithm

The detailed flow of the proposed algorithm is provided below.

A. Segmentation of Initial Template

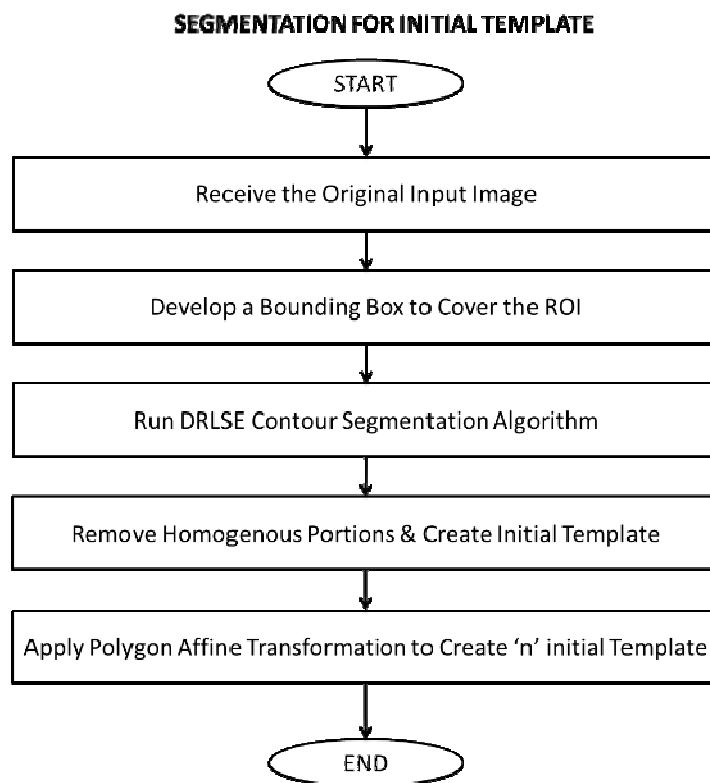


Fig 1: Segmentation of the Initial Template

The initialization stage involves the template creation stage, which would be used for template matching in the later stages by the use of particle filter framework. The Level Set Method (LSM) technique uses the motion of the contour to develop a segment out of the image, which constitute of different homogeneous regions [6]. The distance regularization term is defined with a potential function to maintain a desired shape of the level set function. This provides a new type of level set evolution called Distance Regularized Level Set Evolution (DRLSE). [6]. As the initial step of the particle filter framework the template thus obtained is populated to a group of initial templates that would be used for template matching using the regularization methods. Affine Transform is used to create the different templates by applying the geometrical variations like translation [8].

B.Multikernel Fusion

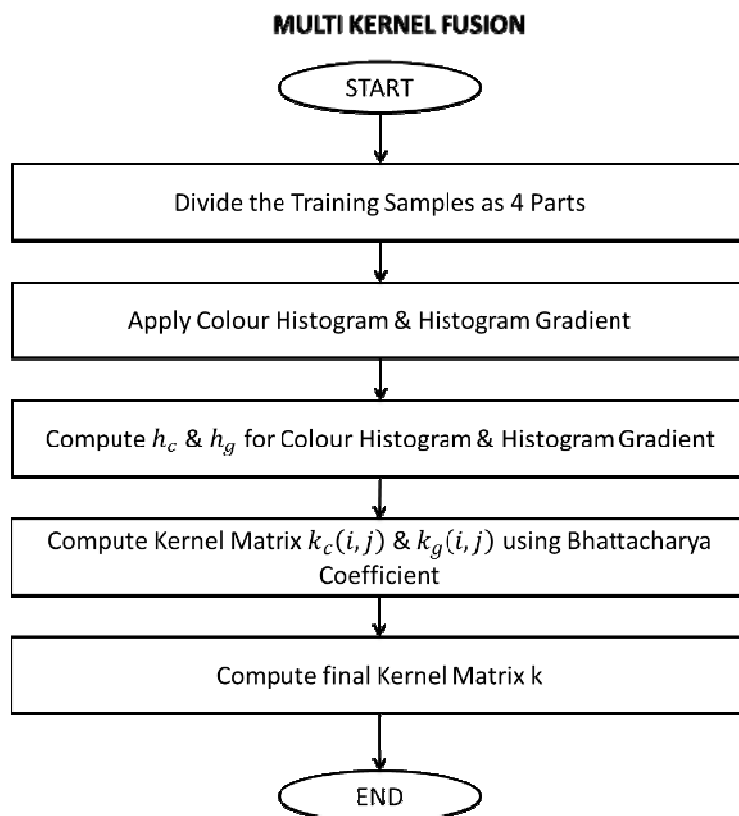


Fig 2.Multikernel Fusion

To speed up the computation time considerably and to achieve robustness, Kernels are preferred. Kernels are instance-based learners usually used in Support Vector Machines [9]. To simplify data representation and also to better visualization of kernels, we make use of the kernel matrix notation [10].

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix} \text{ and} \tag{1}$$

$$K_{ij} = k(x_i, x_j), \text{ where } i, j = 1 \dots n$$

By using a limited number of samples the kernel could learn non-linear functions with lesser computational cost. Considering high dimensionality feature space, to obtain the desired output, the input vectors are to be linearly separable which is done by kernel method. In order to separate the data linearly we require a method that remaps the input space into the feature space[10].

Feature spaces with kernel methods are totally different. Firstly, the feature space is not handcrafted, but the kernel selected determines it. Different kernels specify different feature spaces implicitly [11]. Secondly, the data are linearly separable since the feature space is of higher dimensionality. The performance of the kernel method is known depending on the ease in linearly separating the data and the algorithm employed. The selection of the right kernel for a particular data distribution is essential, which makes the data linearly separable. The mapping functions or the feature spaces for this implementation are Histogram of Gradient h_g and Color Histogram h_c . The relative similarity of the two samples is determined using Bhattacharyya Coefficient.

C.Optimization Algorithm

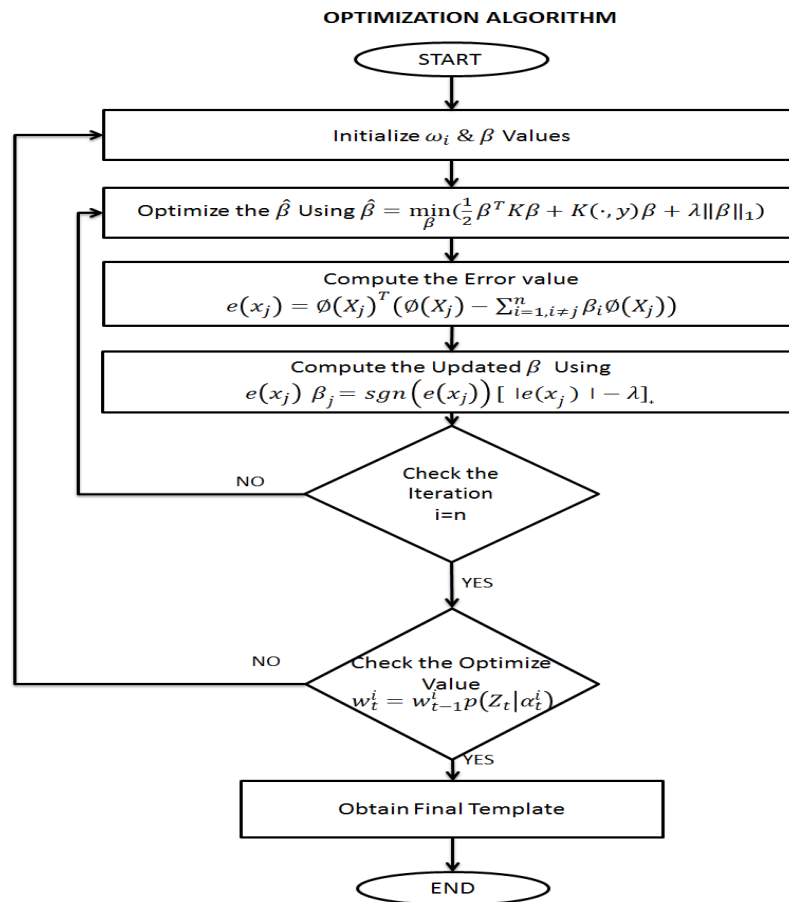


Fig3: Optimisation Algorithm

In order to reduce more expensive computation, we introduce kernel sparse representation, [12] which provides a very fast and simple method of sparse representation. This is achieved by kernel trick, a function referred as $\phi(\cdot)$ which would map a feature vector into the kernel space. $\phi(\cdot)$ which satisfies $\phi(x)^T \phi(x) = 1$ when $\|X\|_2^2 = 1$, which is the condition for convexity.

The KSR can be written as [1]

$$\hat{\beta} = \min_{\beta} \left(\frac{1}{2} \left\| \sum_{i=1}^n \beta_i \phi(X_i) - \phi(y) \right\|_2^2 + \lambda \|\beta\|_1 \right) \quad (2)$$

Recently, the KSR model is used for image classification [6]. In kernel method, we need to find inner product, so the formula can be rewritten as following [12]

$$\hat{\beta} = \min_{\beta} \left(\frac{1}{2} \beta^T K \beta + K(\cdot, y) \beta + \lambda \|\beta\|_1 \right) \quad (3)$$

where K is an $n \times n$ kernel matrix satisfying

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix} \quad (4)$$

and $K(i, y) = \phi(X_i)^T \phi(y)$ is an $n \times 1$ vector.

Multiple features can be easily introduced in KSR by applying multikernel fusion. Weighted multikernel fusion is one of the most preferred method among all the multikernel fusion methods, in which a weighted summation is used to obtain a kernel [13-15]

$$K = \sum_{i=1}^n \omega_i K_i \quad (5)$$

Where K is the fused kernel; K_i is the kernel of i^{th} feature and ω_i is its corresponding weight, satisfying $\sum_{i=1}^n \omega_i = 1$ and $\omega_i \geq 0$. Similarly, the kernel vector [13-15]

$$\mathbf{K}(\cdot, \mathbf{y}) = \sum_{i=1}^n \omega_i K_i(\cdot, \mathbf{y}) \tag{6}$$

The particle filter framework is a strategy to make a decision to find out the posterior distribution of state variables that describe a dynamic framework. It provides an advantageous structure for evaluating and proliferating the posterior distribution density function of state variables, paying insignificant attention to the hidden distribution. It is composed of basically two stages: prediction and upgrade. In this method, execution of image classification is conducted within this framework[13-15].

In the classification algorithm, the classified portion is determined by the particle that has the maximum weight. The initial position of the object is manually chosen from the first frame and is used as the training sample. The other templates are generated by disturbing few pixels which lies in the corner points of the first template. The template number can be obtained by balancing the parameters: classifying efficiency and computational complexity. Each template that has the minimum coefficient of the currently classified object is updated for the changing target appearance. This will look as a continuous learning process as it is classifying.

III. RESULTS AND DISCUSSION

The Indian pines database images and Salinas database images are taken in to consideration, which is hyperspectral in nature. The templates are selected from the original image by comparing with the ground truth image. The ground truth image depicts the actual class present in the original image by means of different colours, each colour depicting one class. The different portions of the images are taken as the object or the template that would be trained sparsely using Multikernel fusion and Optimization Algorithm.

The overall accuracy of the classification is calculated by knowing the number of pixels in the object which is classified correctly and that which is classified wrongly. The overall accuracy for each of the classes is calculated and the results are tabulated. By the use of overall accuracy the Kappa coefficients are calculated and the execution time for the current system requirement and the Matlab version is also tabulated. The overall accuracy of classification from the implementation is calculated from the following formula,

$$OA = \frac{\sum_{i=1}^k n_{ii}}{|T|} \tag{7}$$

Where, n_{ii} is the number of pixels of the current class which is suitably traced by the template matching and T is the total number of pixels that the current class is using in the database image. And k is the number of isolated places where the current class is available on the database image. The kappa coefficient is evaluated based on the classifier performance from the position error. The kappa coefficient is described as $k = \text{probability of correct classification} - \text{probability of chance agreement} / (1 - \text{probability of chance agreement})$.

The illumination change is considered for the calculation, kernel and sparse representation procedures that are utilized are illumination independent. The usage is completed with two unique circumstances that are building up the calculation for database pictures with higher and lower illumination for both Indian Pines and Salinas datasets as shown in fig 4 and 5. To enhance the size variations in the templates, they were resized using affine transform. The global location during the size change was found, and then local template matching was carried out by changing the size of the template. The images that were resized and those that were tilted from the actual angle and those with different illumination conditions were considered to find the accuracy and robustness of the algorithm used. The overall accuracy (OA), kappa coefficient, execution time for various classes are ascertained and tabulated for illumination variation and size variation conditions.

The outcomes derived from the proposed calculation show that the calculation is exceedingly strong as the pictures. The classifying region on the pictures is rough on the grounds that the state of the classified space is constantly rectangular. A contour-based segmentation calculation can be executed but the goal to enhance this should be kept in mind. The output image for Indian Pines Database and Salinas database are shown in fig 5 and 6.

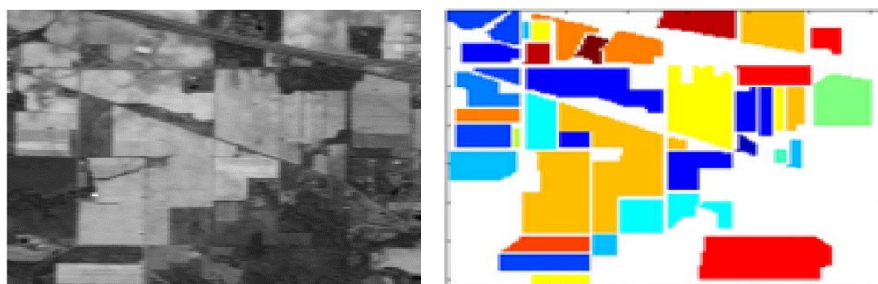




Fig 4: (a) Indian Pines Input Image—High Illumination, (b) Indian Pines Ground Truth Image and (c) Indian Pines 10-Class Classified Output Image—High Illumination

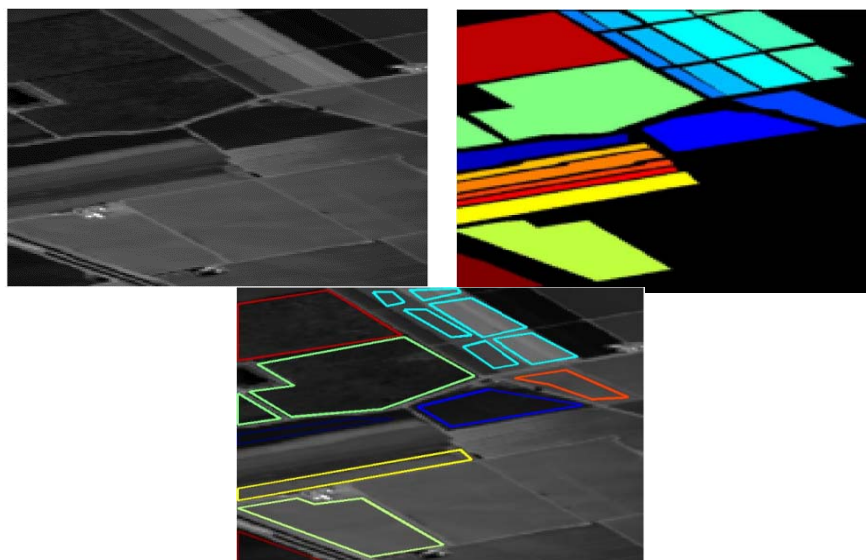


Fig 5: (a) SalinasInput Image—Low Illumination, (b) Salinas Ground Truth Image and (c) Salinas10-Class Classified Output Image—Low Illumination

The MKSR algorithm is applied on the Indian Pines and Salinas datasets for the original image . The OA and kappa coefficient are found to be approximately equal for all the classes in their respective datasets, which proves the robustness. The results are shown in Table 1.

TABLE 1. Performance comparison of the proposed MKSR image for various databases

Proposed MKSR With Original Aviris Data Set Images						
Class Number	Indian Pines Image			Salinas Database		
	Overall Accuracy in %	Kappa Coefficient in %	Execution time in secs	Overall Accuracy in %	Kappa Coefficient in %	Execution time In secs
Class1	90.64	90.46	12.386554	91.64	90.46	13.386554
Class2	90.47	90.35	6.778577	90.57	90.35	60.778577
Class3	89.07	88.61	10.849989	90.07	88.61	10.849989
Class4	94.21	93.69	7.056842	93.21	93.69	9.056842
Class5	85.28	85.18	8.126724	95.28	85.18	9.126724
Class6	82.85	82.83	3.850059	92.85	82.83	4.850059
Class7	81.41	81.38	6.705082	91.41	81.38	10.705082
Class8	83.40	83.39	3.924398	93.40	83.39	11.924398
Class9	89.52	89.50	2.425516	99.52	89.50	7.49516
Class10	81.41	81.38	7.212941	91.41	81.38	7.90

The MKSR algorithm is also applied on the two datasets under different illumination conditions. The OA and kappa coefficient under illumination variation condition are found to be approximately equal when compared with the original image without illumination variation. The results are shown in Table 2.

TABLE 2. Performance comparison of the proposed MKSR for various databases with illumination variation

Proposed MKSR With Illumination Variation Images For Aviris Dataset						
Class Number	Indian Pines Image			Salinas Database		
	Overall Accuracy in %	Kappa Coefficient in %	Execution time in secs	Overall Accuracy in %	Kappa Coefficient in %	Execution time In secs
Class1	90.635179	90.457566	12.163336	91.7	90.457566	19.99
Class2	90.471513	90.348052	6.705922	90.471513	90.348052	9.7112
Class3	89.072848	88.611308	10.735412	89.072848	88.611308	19.5412
Class4	94.210729	93.687681	6.980669	94.210729	93.687681	60.00066
Class5	85.282523	85.176564	8.076532	85.282523	85.176564	9.0732
Class6	82.845188	82.833085	3.786678	82.845188	82.833085	10.786678
Class7	81.413613	81.380076	6.651305	81.413613	81.380076	8.605
Class8	83.404255	83.392935	3.813821	83.404255	83.392935	9.821
Class9	89.518414	89.502268	2.310441	89.518414	89.502268	9.3141
Class10	81.413613	81.380076	6.810555	81.413613	81.380076	9.855

The proposed MKSR algorithm is applied on the two datasets under different size variation conditions, and the OA and kappa coefficient under size variation condition are found to be approximately equal when compared with the original image without size variation. The results are shown in Table 3.

TABLE 3. Performance comparison of the proposed MKSR for various databases with size variation

Proposed MKSR With Size Variation Images For Aviris Dataset						
Class Number	Indian Pines Image			Salinas Database		
	Overall Accuracy in %	Kappa Coefficient in %	Execution time in secs	Overall Accuracy in %	Kappa Coefficient in %	Execution time in secs
Class1	81.64	91.46	12.386554	93.7	92.7566	18.89
Class2	82.57	94.37	69.7577	99.41513	99.34052	9.712
Class3	85.03	89.61	130.849989	88.2848	87.68	18.512
Class4	88.22	92.69	9.956842	93.2129	92.681	62.0669
Class5	98.29	88.19	9.1724	86.2523	89.176564	8.8732
Class6	91.85	83.83	5.059	86.888	81.8485	11.78
Class7	81.41	84.38	13.70	91.413	91.076	9.605
Class8	94.40	89.39	12.924398	93.405	93.395	8.821
Class9	98.8	87.54	9.46	90.577	99.50	8.3
Class10	81.41	84.33	8.91	83.413	83.389	10.88

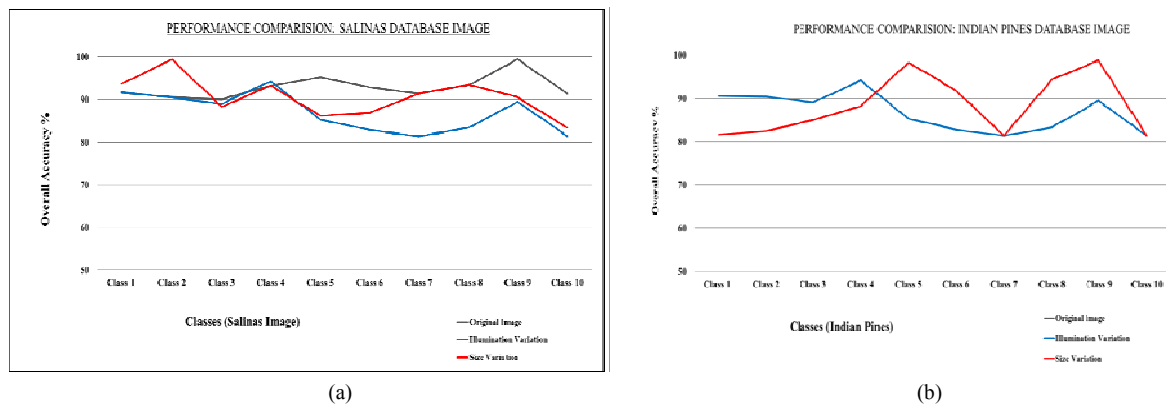


Fig6: Performance Comparison of Database Images: (a) Salinas Image and (b) Indian Pines Image

The other benefits of the proposed MKSR method are as follows: (1) The OA and kappa coefficient for different classes do not face much deviation by which robustness, which is a major parameter for image classification, is achieved. (2) Under illumination variation condition, the OA and kappa coefficient are similar to the original image without illumination. (3) Under size variation condition, the OA and kappa coefficient are similar to the original image without illumination.

CONCLUSION

A Multikernel Sparse Representation technique, which is the sparse coding technique in a high dimensional feature space for image classification is implemented. Experimental results for Indian Pines and Salinas datasets show that our Multikernel Sparse Representation achieves an improved performance for image classification. Moreover, a template matching algorithm is used in the proposed method which speeds up the execution time. Robustness is achieved using a combined execution of DRLSE, PFF, Multikernel fusion. The images that were resized and those that were tilted from the actual angle and those with different illumination conditions were considered to find the accuracy and robustness of the algorithm used.

REFERENCES

- [1] Michael Elad, "Sparse and Redundant Representations From Theory to Applications in Signal and Image Processing", Springer, 2010
- [2] D. Tuia, F. Ratle, F. Pacifici, M. Kanevski, and W. J. Emery, "Active learning methods for remote sensing image classification," IEEE Trans. on Geoscience and Remote Sensing, vol.47, no.7, pp. 2218 -2232, 2009.
- [3] Pooja Kamavisdar, Sonam Soluja, Sonu Agarwal, "A Survey on Image Classification Approaches and Techniques. International Journal of Advanced Research in Computer and Communication Engineering", 2013, vol2, no 1, 1005-1009.
- [4] Shridhar D. Jawak, Prapti Devliyal, Alvarinho J. Luis, "A Comprehensive Review On Pixel Oriented and Object Oriented Methods for Information Extraction from Remotely Sensed Satellite Images With a Special Emphasis on Cryospheric Applications". Advances in Remote Sensing, 2015, vol4, 177-195
- [5] H. Ghassemian and D. Landgrebe, "An unsupervised feature extraction method for high dimensional image data.", IEEE Proc. on System, Man and Cybernetics, 1987, vol 3, pp540-544
- [6] Chunming Li, Chenyang Xu, "Distance Regularized Level Set Evolution and Its Application to Image Segmentation", IEEE transactions on image processing, vol. 19, no. 12, december 2010
- [7] Zheng Zhang, Yong xu, Jian Yang, Xuelong Li and David Zhang, "A survey of sparse representation: Algorithms and Applications," IEEE ACCESS, vol.3, 2015.
- [8] Y. T. Chi, M. Ali, M. Rushdi, and J. Ho, "Affine-constrained group sparse coding and its application to image-based classifications," Proc. IEEE CVPR, pp.681- 688, 2013.
- [9] F. Bovolo, L. Bruzzone, and M. Marconcini, "A novel context-sensitive SVM for classification of remote sensing images," Proc. IEEE Int. Geosci. Remote Sens. Symp. pp. 2498-2501, 2006.
- [10] Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia, "Sparse Representation With Kernels", IEEE transactions on Image Processing, 2013, 22(2), 423 - 434
- [11] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, and J. Calpe-Maravilla, "Composite Kernels for hyperspectral image classification, IEEE Geosci. Remote Sens. Lett., vol. 3, no. 1, pp. 93-97, 2006.
- [12] Shenghua Gao, Ivor W Tsang, Liang-Tien Chia, "Kernel sparse representation for image classification and face recognition", Inbook: Computer Vision - ECCV, pp. 1-10, Sep 2010
- [13] Lingfeng Wang, Hongping Yan, KeLv, and Chunhong Pan, "Visual Tracking via Kernel Sparse Representation with Multikernel Fusion", IEEE transactions on circuits and systems for video technology, 2014, vol 24, no 7, pp 1132-1141.
- [14] X. Mei and H. Ling, Robust, "Visual Tracking and Vehicle Classification via Sparse Representation", IEEE Trans. Pattern Anal. Mach. Intell., 2011, vol33, (11), 2259-2272.
- [15] Jayaraman J. Thiyagarajan, Karthikeyan Natesan Ramamurthy and Andreas Spanias, "Multiple kernel sparse representations for Supervised and Unsupervised Learning," IEEE Trans on Image Processing, Oct 2013.