

Novel Image Classification technique using Particle Filter Framework optimised by Multikernel Sparse Representation

Bhuvaneshwari. N. R¹, Sivakumar.V.G^{2*}.

¹ Research scholar, Department of Electronics and Communication Engineering, Sathyabama University, Chennai;

² Faculty of Electrical and Electronics, Sathyabama University, Chennai.

ABSTRACT

The robustness and speed of image classification is still a challenging task in satellite image processing. This paper introduces a novel image classification technique that uses the particle filter framework (PFF)–based optimisation technique for satellite image classification. The framework uses a template-matching algorithm, comprising fast marching algorithm (FMA) and level set method (LSM)–based segmentation which assists in creating the initial templates for comparison with other test images. The created templates are trained and used as inputs for the optimisation. The optimisation technique used in this proposed work is multikernel sparse representation (MKSR). The combined execution of FMA, LSM, PFF and MKSR approaches has resulted in a substantial reduction in processing time for various classes in a satellite image which is small when compared with Support Vector Machine (SVM) and Independent Component Discrimination Analysis (ICDA)based image classifications obtained for comparison purposes. This study aims to improve the robustness of image classification based on overall accuracy (OA) and kappa coefficient. The variation of OA with this technique, between different classes of a satellite image, is only 10%, whereas that with the SVM and ICDA techniques is more than 50%.

Key words: Multikernel Sparse Representation, Image Classification, Sparse Learning, Level Set Method, Particle Filter Framework, Remote Sensing

* Author for correspondence: sivakumarv2004@gmail.com

INTRODUCTION

A satellite image is rich in information because of its advanced sensing devices, and it is vital in providing geographical information¹. The substantial qualitative and quantitative pieces of information available on remote sensing images cause a revolution in geo informatics, which primarily provides the ground data information of remote sensing images². The bulk of data gathering at data centres is massive and rising exponentially with the rapidly developing technology³. To understand and utilise satellite images, extricating and understanding valued knowledge from a vast set of satellite images is necessary. A satellite image classification is a powerful technique in extracting various pieces of information from a large number of satellite images. Higher dimension and rich spectral information of remote sensing images have been achieved. These would improve the image classification⁴. Given that remote sensing images accumulate more frequently at steady intervals, a speedy and robust image classification is necessary.

The most common techniques for image classification are supervised and unsupervised; parametric and nonparametric; object-oriented; sub-pixel, per-pixel and per-field; spectral, contextual and spectral–contextual classifiers; and hard and soft classifications⁵. The pixel-based approach cannot distinguish surface features which have different objects with similar spectral characteristics⁶. The object-based classification approach overcomes the disadvantages of the pixel-based classification approach, providing improved classification accuracies.

Forming and labelling the segments is vital⁷ in object-based classification, thereby reducing the overall computational complexity. The level set method (LSM) is a technique that uses the motion of the contour to develop a segment out of the image which is composed of different homogeneous regions⁸. The LSM primarily mentioned in the literature^{7,8} did not attract considerable attention. The LSM brought more applications such as computational geometry, fluid dynamics, image processing and computer vision⁹⁻¹⁴.

The object-based grouping technique can exploit multidimensional elements of items. Thus, it is appropriate for a high-resolution radar picture. The robustness of the multikernel sparse representation algorithms and the speed of template matching algorithms are carried out in the current implementation.

PROPOSED ALGORITHM

The proposed algorithm is composed of five important steps:

Segmentation for the Initial Template

The first step of the proposed algorithm is segmentation of the image. Under segmentation or over segmentation should be avoided to prevent any errors during segmentation which in turn produces classification errors. The segmentation algorithm selected in eliminating classification errors and creating initial templates is distance regularisation level set evolution (DRLSE). This LSE uses the motion of the contour to develop a segment out of the image which constitutes different homogeneous regions referred to as templates. The distance regularisation impact prevents the requirement for re-initialisation and maintains a strategic distance from its affected numerical mistakes. The initial templates are created with the help of DRLSE¹⁵.

Template Selection for PFF

The initial step of PFF is to populate the initial template into a group of initial templates that would be used for template matching. The polygon affine transform–

based fast matching algorithm has been used in achieving exact matching and speeding up execution time. Affine transform is used to create the different templates by applying the geometrical variations like translation. The affine transform method is ordinarily used for revising geometric distortions or disfigurements that happen with imperfect camera points. However, in this implementation, it is used to create different templates of the segmented portions of the image. Each segmented portion of the image would represent the classes. The template matching pattern recognition problem assists in finding the similar template in the image under testing. The classification problem could adopt this template matching method only if the higher detail of the template is known. The intricacies of the template should be compared for classification purposes.

Multikernel Fusion

Building an image classification efficiently using single feature extraction is difficult. For example, the colour feature is robust to partial occlusion and deformation but is sensitive to illumination variation. On the other hand, the edge feature is robust to illumination variation but fails in background clutter. Thus, combining multiple features and balancing them is a better approach. The multikernel sparse representation technique (MKSR) allows the fusion of two kernels to provide a hybrid feature for sparse representation which optimises the sparseness of the test template. MKSR will handle the intricacies in the template based on the kernel feature spaces selected. The kernel feature spaces selected are histogram of gradient (HOG) and colour histogram (CH).

A kernel is a function k that for all $x, y \in R^n$ satisfies Equation (1):

$$k(x, y) = \langle \varphi(x), \varphi(y) \rangle \quad (1)$$

where φ is defined as mapping and given as in Equation (2):

$$\varphi: x \rightarrow \varphi(x) \in F \quad (2)$$

The function that calculates the similarity between the two vectors is referred to as kernel. The value of kernel is higher for two vectors that are similar to each other. Kernels are instance-based learners usually used in SVMs. The primary benefit of kernel lies in its use in the same algorithm with different kernel functions and the same kernel functions with different algorithms.¹⁶

2.3.1. Kernel Sparse Representation

The learning in this paper is just a creation of the kernel matrix, as the kernels are instance-based learners. The kernel matrix documentation is utilised to streamline information representation. The kernel matrix K is given as follows:

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix} \text{ and} \\ K_{ij} = k(x_i, x_j), \text{ where } i, j = 1 \dots n$$

To reduce highly expensive computation, the proposed method introduces kernel sparse representation, which is a fast and simple method of sparse representation. It uses the kernel function on both the training samples (x) and the testing samples (y). A function called $\varphi(\cdot)$ is introduced which would map a feature vector into the kernel space. $\varphi(\cdot)$ satisfies $\varphi(x)^T \varphi(x) = 1$ when $\|X\|_2^2 = 1$, which is the condition for convexity.

The KSR can be expressed as in Equation (3):

$$\hat{\beta} = \min_{\beta} \left(\frac{1}{2} \|\sum_{i=1}^n \beta_i \varphi(X_i) - \varphi(y)\|_2^2 + \lambda \|\beta\|_1 \right) \quad (3)$$

Under the kernel method, to find the inner product, the formula is rewritten as in Equation (5):

$$\hat{\beta} = \min_{\beta} \left(\frac{1}{2} \beta^T K \beta + K(\cdot, y) \beta + \lambda \|\beta\|_1 \right) \quad (4)$$

Where K is an $n \times n$ kernel matrix, satisfying Equation (5):

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix} \quad (5)$$

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

The function $\varphi(\cdot)$ can be regarded as a feature extraction, which is an advantage of KSR on SR. $\varphi(\cdot)$ is also a sophisticated feature which is insensitive to occlusion and illumination variation. The application of multikernel fusion in KSR easily introduces multiple features which increase sparseness. One branch among all multikernel fusion methods is the weighted multikernel fusion, in which a weighted summation is used to obtain a kernel¹⁷ and is given in Equation (6):

$$K = \sum_{i=1}^n \omega_i K_i \quad (6)$$

Where K is the fused kernel, K_i is the kernel of the 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 can be expressed as follows:¹

$$K(\cdot, y) = \sum_{i=1}^n \omega_i K_i(\cdot, y) \quad (7)$$

2.3.2. Bhattacharyya Coefficient

The Bhattacharyya coefficient is used in determining the probability of classification errors arising out of the similarity between samples. It is a measurement of the degree of similarity between two statistical samples. This can be used to determine the relative similarity of the two samples being considered.

PFF and Kernel Coordinate Descent Algorithm

PFF uses MKSR for coding vector optimisation. Once the coding vector (β) is updated using the sparse representation optimisation, it would efficiently approximate the test and the sample images. The posterior probability model of the PFF explains how much of the test template is matching the sample template. The weight updating for each test template is conducted to identify the template that matches maximum with the sample templates.

Using the KCD algorithm, coding vector β can be obtained and optimised. KCD uses the coordinate descent approach^{17,18} because of its simplicity and efficiency. Differentiating $E(\beta)$ with respect to β_j and setting it to 0 result in Equation (8)¹.

$$\begin{aligned} e(x_j) &= j(X_j)^T \left(j(X_j) - \sum_{i=1, i \neq j}^n \beta_i j(X_i) \right) \\ &= K(x_j, y) - \sum_{i=1, i \neq j}^n \beta_i K(x_j, y) \end{aligned} \quad (8)$$

β_j is changed independently $\{\beta_i\}_{i=1, i \neq j}^n$.

When calculating β_j , $\{\beta_i\}_{i=1, i \neq j}^n$ is fixed. Hence, β_j is calculated as follows:

$$\beta_j = \text{sgn}(e(x_j)) [|e(x_j)| - \lambda]. \quad (9)$$

In the rundown, the coding vector β is upgraded iteratively by (11) by utilising the KCD calculation. The instatement of β is achieved by kernel ridge regression, as shown in Equation (11)^{17,18}

$$\beta_{\text{init}} = (K + \gamma I)^{-1} K(\cdot, y) \quad (10)$$

where γ is given to a small positive value. In practice, it is set to $\gamma = 2\lambda$.

Suppose α_t is the state variable at time t , which is used in characterising the state of object, such as position, size, speed and shape. Suppose Z_t is the observation at time t , while $Z_{1:t}$ represents all observations up to time t , namely, $Z_{1:t} = \{Z_1, Z_2, \dots, Z_t\}$. Prediction is provided in ^{17,18}

$$p(\alpha_t | Z_{1:t}) = \int p(\alpha_t | \alpha_{t-1}) p(\alpha_{t-1} | Z_{1:t-1}) d\alpha_{t-1}$$

And update is given by the equation (11)

$$p(\alpha_t | Z_{1:t}) = \frac{p(Z_t | \alpha_t) p(\alpha_t | Z_{1:t-1})}{p(Z_t | Z_{1:t-1})} \quad (11)$$

In particle filter, $p(\alpha_t | Z_{1:t})$ is approximated by a set of N particles $\{\alpha_t^i\}_{i=1}^N$ with importance weights $\{w_t^i\}_{i=1}^N$, namely ¹,

$$p(\alpha_t | Z_{1:t}) = \sum_{i=1}^N \delta(\alpha_t - \alpha_t^i) w_t^i$$

Where $\delta(\cdot)$ is the Dirac function. The weight of the i^{th} particle is iteratively redesigned by Equation (12) ^{17,18},

$$w_t^i = w_{t-1}^i p(Z_t | \alpha_t^i) \quad (12)$$

Where w_{t-1}^i is the weight of the i^{th} particle at previous time $t-1$, and $p(Z_t | \alpha_t^i)$ is the probability of the observation of the i^{th} particle. The weights $\{w_t^i\}_{i=1}^N$ are normalised. The classification process is governed by the observation model $p(Z_t | \alpha_t^i)$, where the probability of α_t^i is estimated by observing Z_t and the dynamical model between the two states.

RESULTS AND DISCUSSION

The Indian Pines database pictures are hyper spectral in nature. Around 220 pictures are found in the database. The ground truth picture portrays the genuine class present in the first picture by the method for various hues, every colour delineating one class. As the object is browsed, the picture in this technique goes under object-based classification. The accompanying figures portray the diverse class extraction from the first picture by taking the ground truth as the reference picture. These segments of the pictures are taken as the object or the format that would be trained using a sparse representation.

The OA of classification from the implementation is calculated by the following

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

formula:

where n_{ii} is the quantity of pixels and T is the aggregate number of pixels that the present class is utilising as part of the database picture. Moreover, k is the quantity of separated spots where the present class is accessible on the database picture. The kappa coefficient is the measure of the classifier execution from the position error.

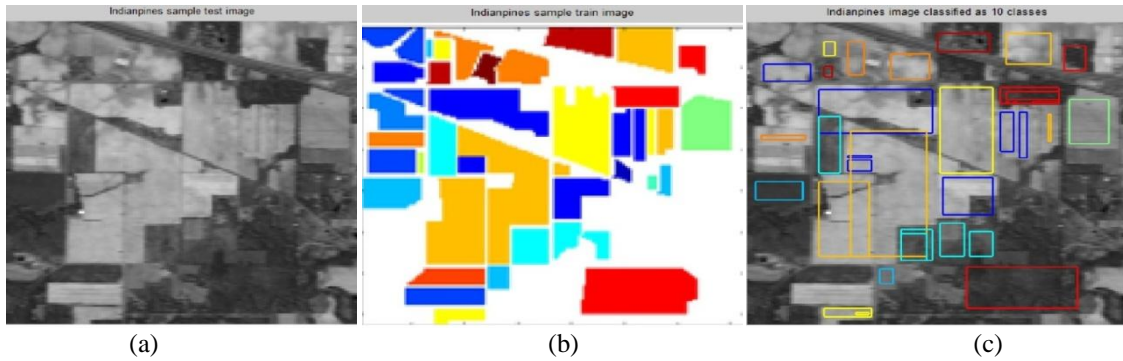


Figure 1: (a) Indian Pines Image, (b) Ground Truth Image and (c) Indian Pines 10-Class Classified Output Image.

The kappa coefficient is characterised as $k = \text{probability of correct classification} - \text{probability of chance agreement} / (1 - \text{probability of chance agreement})$.

The proposed MKSR method is used in classification, and the results are shown in figure 1 which is compared with those obtained with SVM- and ICDA-based image classifications¹⁹. Table 1 and figure 2 shows that the OA and average accuracy provided by MKSR are better than those provided by SVM and ICDA techniques for both datasets. In general, object-based classification has the following drawbacks: (1) Under segmentation introduces classification errors because pixels are overlapped in image objects and assigned to the same class. (2) From this under segmented or over segmented image objects which consist of classification errors, the features extracted under such a condition do not represent the properties of real objects. In this proposed MKSR method, classification errors are avoided using the DRLSE and template matching algorithms, which in turn improve OA and average accuracy. The improvement in the kappa coefficient in the proposed MKSR method is more than the SVM- and ICDA-based image classifications.

Table 1. Performance comparison of the proposed MKSR image classification with SVM- and ICDA-based classifications with original images

Class Number	SVM [19]	ICDA [19]	MKSR
	Overall Accuracy in %	Overall Accuracy in %	Overall Accuracy in %
Class1	84.44	83.33	90.64
Class2	36.79	55.97	90.47
Class3	40.67	51.61	89.07
Class4	72.31	78.80	94.21
Class5	80.40	84.87	85.28
Class6	78.93	92.53	82.85
Class7	95.38	96.15	81.41
Class8	76.11	90.43	83.40
Class9	100.00	100.00	89.52
Class10	53.80	57.25	81.41
Average Accuracy	70.10±0.87%	76.15±1.02%	86.83±1.25%
Processing time	580 secs	420 secs	108 secs

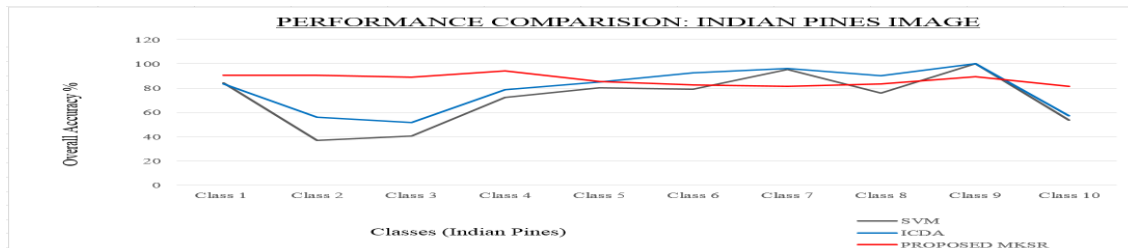


Figure 2: Performance comparison graph of the proposed MKSR and SVM- and ICDA-based classifications with original images

The computational complexity of SVM, ICDA, and KSR has been analysed. The processing time of SVM depends on the training samples. When a large number of training samples are used, the processing time is long. The training time in ICDA is short because of the fast computation of density estimations of the training samples. By contrast, because of the calculation of the densities for each of the testing samples, the time taken for testing is long. The size variation which is the drawback of SVM and the density estimation variation which is the drawback of ICDA are overcome by combining PFF and multikernel optimisation technique in the proposed MKSR method which reduces the computational complexity.

As the multikernel-based strategy is actualised next to the sparse learning technique, the OA and kappa coefficient, which are ascertained in percentage, are in average higher than the SVM- and ICDA-based strategies used for image classification. Thus, the precision can be moved forward by further increasing the number of templates.

CONCLUSION

A multikernel sparse representation-based image classification implementation was implemented, and the results were tabulated. The results showed that the accuracy of the MKSR-based implementation is on average better than the SVM and ICDA implementations, and further improvement of the kernel features can result in a robust method with higher accuracy. Given the robust nature of the algorithm used, the implementation on images with lesser illumination also performed better as the HOG and CH kernel spaces were fused. The template matching, if conducted using a variable size template, could further improve the accuracy. The computational complexity was considerably reduced in the proposed MKSR method by overcoming the size variation drawback of SVM and the density estimation variation of ICDA with the combined use of PFF and multikernel optimisation technique.

REFERENCES

1. Muhammad, S., Aziz, G., Aneela, N. and Muhammad, S., Classification by Object Recognition in Satellite Images by using Data Mining. In Proceedings of the World Congress on Engineering (WCE 2012), 2012, 1, 4 -6.
2. Chaichoke, V., Supawee, P., Tanasak, V. and Andrew, K, S., A Normalized Difference Vegetation Index (NDVI) Time-Series of Idle Agriculture Lands: A Preliminary Study. Engineering Journal, 2011. 15(1), 9-16.
3. Zheng, X., Sun, X., Fu, K. and Hongqi Wang, Automatic Annotation of Satellite Images via Multifeature Joint Sparse Coding With Spatial Relation Constraint, IEEE Geoscience and Remote Sensing Letters, 2013,10(4), 652-656.
4. H. Ghassemian, On-board satellite image compression by object- feature extraction. 20th congress of International Society of Photogrammetry and Remote Sensing Archives, 2004 35(B3), 820-825.
5. 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, 2(1), 1005-1009.
6. 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, 4, 177-195
7. H. Ghassemian and D. Landgrebe, An unsupervised feature extraction method for high dimensional image data. IEEE Proc. on System, Man and Cybernetics, 1987, 3, 540-544.

8. S. Osher and J. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.* 1988, 79(1), 12–49.
9. A. Dervieux and F. Thomasset, A finite element method for the simulation of Rayleigh-Taylor instability. *Lecture Notes Math.*, 1980, 771, 145–158.
10. A. Dervieux and F. Thomasset, Multi fluid incompressible flows by a finite element method. *Lecture Notes Phys.*, 1980, 141, 158–163.
11. V. Caselles, F. Catta, T. Coll, and F. Dibos, A geometric model for active contours in image processing, *Numer. Math.*, 1993, 66(1), 1–31.
12. R. Malladi, J. A. Sethian, and B. C. Vemuri, Shape modeling with front propagation: A level set approach. *IEEE Trans. Pattern. Anal. Mach. Intell.*, 1995, 17(2), 158–175.
13. M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active contour models. *International Journal of Computer. Vision.*, 1987, 1(4), 321–331.
14. S.-C. Zhu and A. Yuille, Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Trans. Pattern. Anal. Mach. Intell.*, 1996, 18(9), 884–900.
15. Chunming Li, Chenyang Xu, Distance Regularized Level Set Evolution and Its Application to Image Segmentation, *IEEE transactions on image processing*, 2010, 19(12), 3243 – 3254.
16. Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia, Sparse Representation With Kernels, *IEEE Transactions on Image Processing*, 2013, 22(2), 423 - 434
17. 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, 24(7), 1132-1141.
18. X. Mei and H. Ling, Robust, Visual Tracking and Vehicle Classification via Sparse Representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011, 33(11), 2259–2272.
19. Alberto Villa, Jon Atli Benediktsson, Jocelyn Chanussot, Christian Jutten, Hyperspectral Image Classification With Independent Component Discriminant Analysis. *IEEE transactions on Geoscience and Remote Sensing*, 2011, 49(12), 4865-4876

Received: February 03, 2016;

Accepted: July 14, 2016