A Simple Data Preprocessing and Postprocessing Techniques for SVM Classifier of Remote Sensing Multispectral Image Classification

Manish Pratap Singh, V. Gayathri, and Debasis Chaudhuri, *Senior Member, IEEE*

***Abstract*—Present scenario of the remote sensing domain deals with how to utilize the data for different purposes like classification, target detection, disaster management, change detection, flood monitoring, deforestation, etc. Now due to improvements in the sensor technology very high spatial and spectral resolutions data are available. Over a decade, various new advanced research papers have been projected in the literature for spatial and spectral classification of such high-resolution remote sensing images. Thematic information investigation of the earth's surface image is possible by the classification technique and the most frequently used method for this purpose is multispectral classification using a supervised learning process. In the supervised learning process, the specialist challenges to discover exact sites in the remotely sensed data that represent homogeneous examples of the known land cover type. The most recommended method for the classification of remote sensing (RS) images is the support vector machine (SVM) because of its high accuracy but any classifier depends on good quality training samples. The collection of authentic training samples of different classes is a critical issue when the whole classification result is important. This article presents a preprocessing technique based on local statistics for generation-correction of training samples with quadrant division. A simple filter-based postprocessing technique is proposed for the improvement of classification accuracy. We study rigorously how the proposed preprocessing technique has affected the result of classification accuracy for different kernels SVM classifiers. Also, we have presented the comparison results between the proposed method and other different classifiers in the literature.**

***Index Terms*—Classification, remote sensing, spectral and spatial resolution, supervised learning, SVM, training sample.**

I. Introduction

CLASSIFICATION is the recognized method in the domain of image processing and pattern recognition [1], [2], [3], [4], [5] for classifying data. The earth surface remote sensing multispectral images are transmuted into information and the information is extracted using the classification technique. There exist two methods of multispectral classification technique, namely *supervised* and *unsupervised*. First, the training samples also called *training sites* of different classes like, concrete, water, vegetation, etc., are collected by field visits, maps, analysis of aerial photography, and photo interpreter for supervised classification technique. Next, the researchers classified each pixel of the image by a supervised classification algorithm using the training samples [6]. The unknown pixel of the image is assigned to the member of a class that has the highest likelihood. Smooth homogeneous training samples are important features for improving the classification accuracy [7].

Whereas, the ground truth or surface features are not well known in an unsupervised classification technique and the characteristics of land cover categories to be measured as classes within a scene are not generally known *a priori*. The pixel data of the image are grouped into different spectral classes based on some estimated statistical parameters and finally, the specialist has to label these clusters.

In the literature, various classification techniques [6], [17] such as Bayesian classifiers, ANN, *k*-NN classifiers, decision tree, Random Forest, minimum distance classifier, parallelepiped classifier, maximum likelihood classifier, multiseed voting approach [8], [9], and SVM [10], [11] exist for the prediction of accurate results. SVM gives better accuracy with optimized results in a sophisticated way among all these classifiers [11], [12], [13]. Recently, many researchers have published various SVM-based classification techniques [14], [15], [16], [36]. Lu and Weng [17] have reviewed and discussed in their survey paper on foremost cutting-edge classification techniques for improving accuracy. Su [18] proposed a likelihood class filter postclassification technique to remove nonhomogeneous classes for better accuracy. Kamiran and Calders [19] proposed data preprocessing techniques for classification problems. Nowadays machine learning algorithm is a growing advanced tool and researchers are involved to get a better idea for understanding the performance of this tool. A complete assessment of this tool is required for remote sensing image classification, which can complete the material classification by using the same land cover and the same satellite image. The final results of image classification are dependent on some issues—classification architecture, image data accessible, sample collection, preprocessing of the data with feature assortment and mining, classification algorithm, postprocessing methods, test sample gathering, and validation schemes [20]. Li et al. [21] proposed a performance comparison of 15 classification algorithms for the same set of data and

Manuscript received 22 January 2022; revised 7 April 2022 and 14 July 2022; accepted 15 August 2022. Date of publication 24 August 2022; date of current version 8 September 2022. This work was supported by Defenec Research & Development Organization, India. *(Corresponding author: Debasis Chaudhuri.)*

Manish Pratap Singh and V. Gayathri are with the DRDO Young Scientist Laboratory, Chennai 600113, India (e-mail: anurag.galib@gmail.com; gayathri.nvenkatesan@gmail.com).

Debasis Chaudhuri is with the Department of Computer Science and Engineering, Techno India University, Kolkata 700091, India (e-mail: deba_chaudhuri@yahoo.co.in).

Digital Object Identifier 10.1109/JSTARS.2022.3201273

fixed up the same other factors and applied the procedures on a pixel-by-pixel basis. Also, they have analyzed the procedures on object-based segments through preprocessing step in the image classification procedure.

Bernab´e et al. [33] proposed the procedure for spectral-spatial classification of multispectral RS images with the restriction of spectral resolution. Various literature is available in the context of several methodology comparisons in the classification domain. In remote sensing classification applications, data normality is required for maximum likelihood classifiers but the same is not required for ANN and SVM classifiers. Candade and Dixon [34] have compared the ANN and SVM classifiers and SVM achieved better than ANN even in a small number of training data. Also, SVM is enormously faster and simple than ANN. They have also observed that in the SVM classifier different kernels are not much different in the classification results. Another paper [35] has compared the performance between SVM and *k*-NN classifiers and the authors claimed that *k*-NN performed better than SVM in multispectral image classification. Wang et al. [37] proposed a hyperspectral image classification method using multiple kernels in an SVM classifier by combining spectral, spatial, and semantic information. Another way of performance analysis has been studied for SVM classifiers with different types of kernels. Savas and Dovis [38] have selected different kernels for scintillation detection using SVM. They have selected suitable kernels by investigating the preparation of data, cross-validation, and experimental test steps. They have observed that the complete accuracy of fine Gaussian SVM outclasses the linear, which has the lowest complexity and running time. Also, the third-order polynomial kernel gives a better result compared to other kernels in the SVM classifier but it increased the complexity. A good review paper [39] is reported in the domain of the SVM classification method and in this article, the authors have discussed the performance analysis between Random Forest and SVM classifiers.

Any supervised classification technique depends on the good quality of training samples of different land covers. But the training samples of a particular class may or may not be homogeneous due to various factors. It is essential to correct the training data so that it becomes homogeneous data. Very little literature [19], [22] is available for preprocessing the training data in the supervised classification domain. This aticle presents an adaptive-based statistical method for the generation and correction of training samples. At the same time, we will discuss elaborately how the proposed preprocessing technique has affected the result of classification accuracy for different kernels SVM classifiers. A postprocessing technique is introduced to increase the classification accuracy. Also, in this article, the classification results are compared in different combinations of preprocessing stages with SVM-based classifiers on different types of kernels, Decision Tree classifiers, *k*-NN, and Random Forest classifiers. The rest of the article is prepared as follows. A short-term overview of SVM is presented in Section II. The proposed SVM classification technique along with preprocessing and postprocessing techniques is delivered in Section III. Experimental results and future scope are described in Section IV. Finally, a summary is concluded in Section V.

II. Overview of Different Classifiers

The sensor technology has sufficiently improved in the remote sensing satellite for acquiring images of the earth's surface and the researchers are involved in extracting the information through advanced techniques from these data. That is the most challenging task of data analysis techniques from a new generation of sensors. In this context, the most effective technique is SVM, which is a supervised classification technique constructed on kernel methods that have been evidenced very active in solving compound classification complications in many different application fields. Also, the usefulness of SVM has increased in remote sensing applications in the last few years.

A. SVM Classifier

Training of SVM: The main challenge of the classification procedure of SVM is to isolate samples belonging to unlike classes by tracing maximum margin hyperplanes in the kernel space [see Fig. 1(a)], which is equivalent to minimizing $\boldsymbol{W}$. $\boldsymbol{W}$ is the distance from the training samples to the optimal decision hyperplane. In the example (see Fig. 1), yellow-squared samples are the *support vectors* for finding the optimal decision hyperplane.

SVM classifier is the optimal classification technique and the simplest scenario occurs when the SVM is applied to linearly separable data. Suppose $(x_i, y_i), i = 1, 2, \ldots k$ be the training samples and $x_i \in \mathcal{R}^N$ is the *N*-dimensional space. Here, $y_i \in \{-1, +1\}$ is denoted as class labels. These training samples are linearly separable if there exists a vector *w* and a scalar factor *b* such that the following equation is satisfied:

$$y_i (w \cdot x_i + b) - 1 \geq 0. \tag{1}$$

This hypothesis can be described using the following equation:

$$G_{w,b} = \operatorname{sign}(w \cdot x + b). \tag{2}$$

The SVM technique extracts the maximum distance separating hyperplanes between the classes by using the following optimization constraint:

$$\min_{w,b} \frac{1}{2} ||w^2||. \tag{3}$$

For the linearly separable classes, all the training samples of a specified class should lie on the same side of the optimum hyperplane and can be achieved with the help of *slack variable* α_i. The data which are not linearly separable can be handled to find out the optimum hyperplane using the following equation:

$$\min_{w,b,\alpha_1,\alpha_2,\ldots,\alpha_k} \left[\frac{1}{2} ||w^2|| + C \sum_{i=1}^{k} \alpha_i \right]. \tag{4}$$

Here in (4), *C* is a constant to control the difference between margin and misclassification error. The value of *C* varies as $0 < C < \infty$. For nonlinear decision surfaces, the feature vector $x \in \mathcal{R}^N$ is mapped into higher dimensional Euclidean space denoted as $\mathcal{F}$, i.e., $\psi : \mathcal{R}^N \mapsto \mathcal{F}$ [32] (as Fig. 2). Here, $\mathcal{F}$ can be rewritten by substituting x_i, x_j using the dot product of the functions $\psi(x_i), \psi(x_j)$ i.e., $\mathcal{F} = \psi(x_i) \cdot \psi(x_j)$. This $\mathcal{F}$ can be replaced

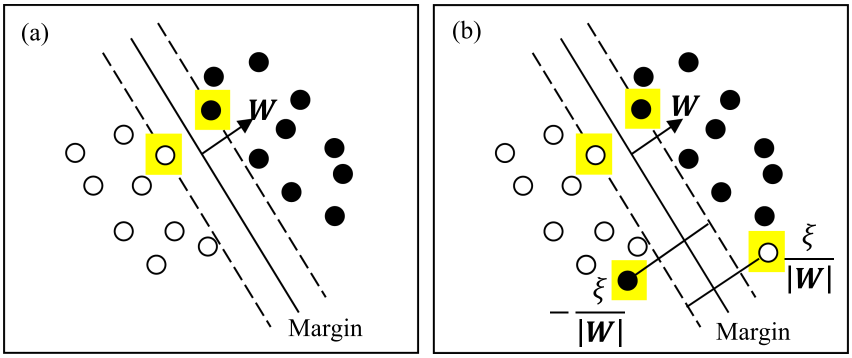


Fig. 1. (a) Representation of the optimal decision hyperplane in a linearly separable problem. (b) Linear decision hyperplanes in nonlinearly separable data can be handled by including slack variables ξ_i to allow classification errors.

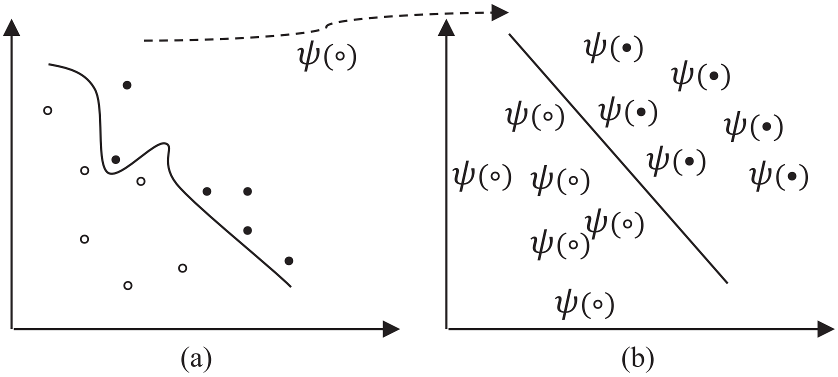


Fig. 2. Transformation of the input data by means of a kernel function into a high-dimensional feature space. (a) Input feature space. (b) Kernel induced high-dimensional feature space.

by Kernel function $\mathcal{K}$ for the reduction of computational cost and is defined as

$$\mathcal{K}\ (x_i, x_j) = \psi\,(x_i) \cdot \psi\,(x_j) \tag{5}$$

$$G_{w,b} = \text{sign}\left(\sum_i \lambda_i \cdot y_i \mathcal{K}\,(x_i, x_j) + b\right) \tag{6}$$

where λ_i is known as Lagrange multiplier.

The sign of $G_{w,b}$ is important for classification decision

$$\text{If } G_{w,b}\,(x) \begin{cases} \geq 0 \text{ then } x \in C_1 \\ < 0 \text{ then } x \in C_2. \end{cases} \tag{7}$$

The Kernel function $\mathcal{K}$ can be replaced using the following equations depending on the type of Kernel. Commonly used kernel functions are as follows:

For Linear kernel $(\boldsymbol{LK}) : \mathcal{K}\ (x_i, x_j) = x_i^T\ x_j$ (8)

For Polynomial kernel of power $P\,(\boldsymbol{PK}) :\ \mathcal{K}\ (x_i, x_j)$

$$= \left(1 + x_i^T x_j\right)^P\ ,\ \text{here } P\ =\ 2 \tag{9}$$

For Gaussian (Radial Basis Function) kernel $(\boldsymbol{GK})$:

$$\times\,\mathcal{K}\ (x_i, x_j) = exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right) \tag{10}$$

For Sigmoid kernel $(\boldsymbol{SK}) : \mathcal{K}(x_i, x_j) = \tanh\left(a x_i^T x_j + s\right)$. (11)

Here are two parameters: the scaling parameter of the input data is a and the shifting parameter is s, which controls the threshold of mapping. The best suitable combination for Sigmoid kernel is $a > 0$ and $s < 0$.

III. Proposed Methodology

In the last couple of years, there have been tremendous improvements in the sensor technology for remote sensing data analysis and information extraction of the earth's surface for civilian and defense operational applications. In this context, the most challenging aspect is to develop an advanced methodology for remote sensing data analysis. Recently, SVM is a very popular supervised classification technique, which is a kernel-based method that has been effectively used in solving various complicated classification problems. In the last few years, SVM's magnificence improvement in the remote sensing domain has increased [12], [23], [24], [25]. Fig. 3 shows the flowchart of the proposed algorithm. Next, we have described each step in the following sections.

A. Training Sample Collection

One of the most important features of multispectral data is *training sample selection*. A training sample to represent a land cover type is formed by grouping together a small representative number of those cells in the image files that are alike and

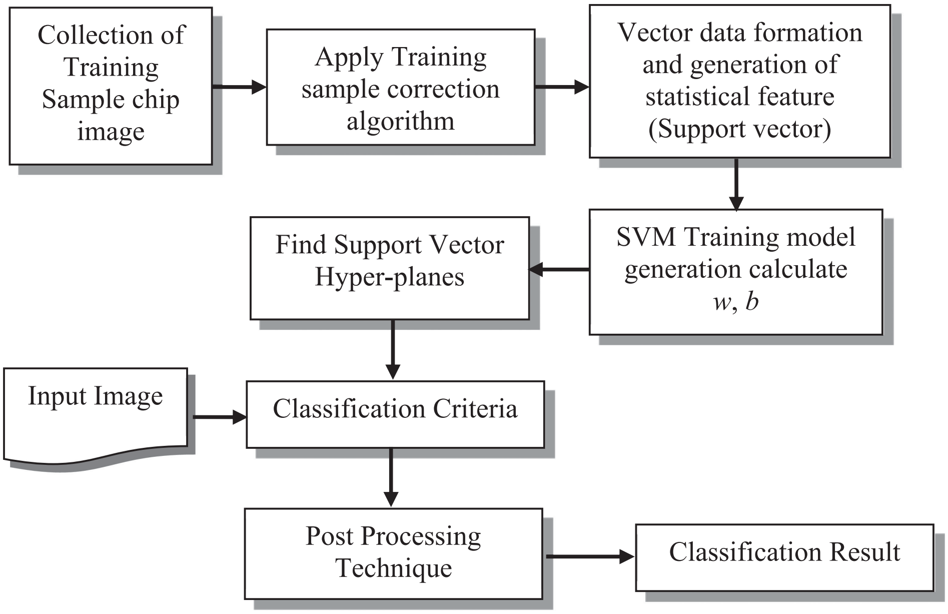

Collection of Training Sample chip image
Apply Training sample correction algorithm
Vector data formation and generation of statistical feature (Support vector)
SVM Training model generation calculate w, b
Find Support Vector Hyper-planes
Input Image
Classification Criteria
Post Processing Technique
Classification Result

Fig. 3. Flowchart of the proposed classifier.

represent a known land use or land cover class. The similarity of the cell accumulated together to characterize one class is quantified by statistical similarity in the radiance values recorded for those cells for a particular variable. So, a training set is a subsample whose credentials are known to some satisfactory level of accuracy and should be characteristic of each of the land use/land cover classes to be examined. The collection of training samples for RS image classification is always time-consuming and expensive. Few techniques are available for training sample collection based on active learning [48]. Foody et al. [55] proposed a per-pixel basis training samples collection procedure for decreasing redundancy and spatial-autocorrelation. They were cautiously chosen through image description with laborious field visits. It is habitually suggested that a training sample size for each class should not be smaller than 10–30 times the number of bands [56], [57].

In our project work, we have considered only four classes namely, water, vegetation, concrete, and others (rest class from water, vegetation, and concrete classes). Here concrete and vegetation are very broad classes i.e., buildings, roads bridges, etc., belong to the concrete class, and grass, forest, trees, etc., belong to the vegetation class. Perfect training sample generation of the defined classes is the most important part of any supervised technique. The major step is the former identification of training samples in straightforward supervised classification. First, we have collected the training chip image from the image of a particular class in the present location of the *region of interest* (ROI) and accepted the training chip sample image if both the constraints of the image chip are simultaneously satisfied. First, compute the maximum gray value (Max_g) , minimum gray value (Min_g), and mean gray value $(Mean_g)$ of the chip image in the present location of ROI. Second, compute $(Max_g - Mean_g)$ and $(Meanx_g - Min_g)$. The constraints are i) $(Max_g - Mean_g) \leq T$ and ii) $(Mean_g - Min_g) \leq T$, where T is prespecified tolerance level. The choice of the value of T is a critical issue. If it is too high then always heterogeneity chip image will be accepted as a training sample image and outlier rejection is not possible. At the same time, if it is too small then limited sample values will be considered and the judgment will not be unbiased because a genuine class sample may be deleted from the training dataset. We have seen that the gray value variation of a particular class depends on the area land cover type. If the area is plane land, then for example the water body gray value variation is different from the desert area. The water body in the desert area is brighter. In this case, the value of T for the desert area is more than the plane area. In our tested image we have taken $T = 6$. Here, both the constraints (i) and (ii) check the values on both sides of the mean value, which is required for balancing the homogeneity of the sample. If the present image chip does not satisfy both the constraints then we shifted the region of interest polygon over another region for that class in the image and followed the same procedure. Same regions have been rechecked by different band channels. Further, the collected training samples are also rechecked by field visits, or by using reference data such as topographic maps and aerial photographs. Finally, the training data are rechecked for further authenticity by a skilled photo interpreter.

Collected training sample images are refined by the training sample generation-correction method, which is described in Section II-B. Training sample generation-correction method is a second layer filtering process for rejection of outlier and conversion of homogeneous training sample image chips. Total accuracy of non-corrected random sample image chips of different four classes and after training sample generation-correction method are shown in Table I in the experimental results, which shows the effectiveness of our training sample generation-correction method.

TABLE I
SPATIAL AND SPECTRAL CHARACTERISTICS OF KOMPSAT-3A SATELLITE

Mission Characteristic	**KOMPSAT-3A**
Design lifetime	4 years
Orbit altitude	528 Km
Swath width	≥ 12 Km (at nadir)
Gound sample distance	Pan: 0.55 m MS: 2.2 m
Spectral bands	Pan: 450-900 nm MS: Blue: 450-520 nm Green: 520-600 nm Red: 630-690 nm NIR: 760-900 nm
Radiometric resolution	14 bits

B. *Training Sample Generation-Correction Method*

Data preprocessing is an important step in remote sensing image classification problems. The most familiar terms in remote sensing image preprocessing are radiometric correction, geometric correction, noise removable, enhancement process, data reduction, etc. As per our knowledge, no such paper is available for training sample data preprocessing for remote sensing image classification problems. Few papers are available in the context of preprocessing technique but those are not in the application of remote sensing image classification problems. Here, we have discussed some papers related to preprocessing for other types of data. Kamiran and Calders [19] have offered the classification with a nondiscrimination restrictions problem. They have discussed three techniques for preprocessing the training data—*Massaging*, *Reweighing*, and *Sampling*, which eliminate discrimination from the training data, and then a classifier is trained on this unbiased data. They have concentrated on the case with only one binary sensitive attribute and a two-class classification problem. Data preprocessing is an important step in data mining from enormous structure operational data. Fan et al. [49] have reviewed data preprocessing of both conventional and cutting-edge data preprocessing techniques in existing literature, including missing value imputation, outlier detection, data scaling, data reduction, data transformation, and data partitioning. In their paper also, they have not discussed the preprocessing techniques for remote sensing data classification. Capodiferro et al. [50] proposed different kinds of data preprocessing, mainly noise removal procedures to improve SVM video classification, which is not related to our work. Salvi et al. [51] proposed another type of pre and postprocessing technique in the deep learning framework for medical image analysis. Three categories of preprocessing techniques have been described in their paper, namely—i) tissue and artifact detection, ii) stain color normalization algorithms, and iii) patch selection techniques. The method, which is not dependent on statistical measurement, is concentrated on medical imagery and not defined any training sets. In [20], the authors have discussed preprocessing that is only restricted to radiometric and geometric correction, feature selection, data reduction, and noise elimination. Also, they have discussed the training sample selection but not the preprocessing technique for improving the quality of training samples. Another paper [52] has considered normalization, discretization, and dimensionality reduction for data preprocessing on the accuracy of machine learning.

Perfect training sample generation of the defined classes is the most important part of any supervised technique. In the previous section, we have discussed the training sample image chip collection of different classes. The accuracy of the above training sample image chip is measured by the calculation of the standard deviation of the chip image. If the standard deviation is high the image chip is not homogeneous. The training sample chips are generated from the image and the image chips may or may not be homogeneous due to noise, lighting conditions, etc. In such a case, if we extract the training features from the heterogeneous sample chips and classify the unknown image using those features then definitely the accuracy of the classification result will be reduced. So, the training sample generation-correction step, which is a preprocessing step, is very essential to improve the accuracy. The proposed preprocessing technique is a simple and statistical-based method to make the burden of the training samples lighter as well as at the same time increase the accuracy level of the training samples and classification results. Very less work regarding preprocessing method is available for remote sensing image classification as per our knowledge. If the training sample chip is perfectly homogeneous then the classification result will be more accurate. But normally in remote sensing data, extraction of a perfect homogeneous training sample chip is a very difficult task for many reasons, like noise, weather conditions, moisture effect, etc. So, our aim is how to form a homogeneous training sample chip for getting better accuracy in the classification result. We used a few statistical parameters, such as standard deviation, mean, mode, etc., in the proposed preprocessing technique. Basic intuitions for these parameters are: i) standard deviation gives the measure of homogeneity, ii) mean gives the central tendency, and iii) mode gives the maximum occurrence probability of gray value in that region. We have converted the heterogenous training sample to homogeneous using these parameters in the statistical framework. Though sufficient number of training samples are usually advantageous, as they have a habit of more representative to the class population, a small number of training samples is obviously attractive for logistic reasons [55]. Total accuracy of random sample image chips of different four classes and after training sample generation-correction method are shown in Table II in the experimental results, which shows the effectiveness of our training sample generation-correction method. The algorithm in this aspect is described below.

The above training sample correction algorithm will give unimodal Gaussian distribution data, which is more acceptable than the raw training data, and also it can handle noisy data. This training sample generation-correction module is useful to classify the unknown data more accurately. The next step is vector data formation and generation of the statistical feature.

TABLE II
ERROR MATRICES OF RAW TRAINING SAMPLE AND CORRECTED TRAINING SAMPLE

	Class types determined from reference source (raw training sample)							Class types determined from reference source (corrected training sample)				
Class types determined from classified map		W	C	V	O	T	Class types determined from classified map		W	C	V	O	T
	W	43	6	0	0	49		W	48	1	0	0	49
	C	5	36	2	6	49		C	2	45	2	0	49
	V	0	7	38	4	49		V	0	1	47	1	49
	O	2	11	6	30	49		O	0	6	4	39	49
	T	50	60	46	40	196		T	50	53	53	40	196

C. Vector Data Formation and Generation of Statistical Feature (Support Vector)

SVM is a binary classification problem and the training process depends on the number of training data. But SVM classifier is much more attractive and faster than the ANN classifier. Several approaches have been published to speed up the classification procedures and most of them is to try to reduce the number of SVs. In terms of the speed-up process, SVMs are faster than ANN classifiers but still slower than the other many standard classifiers, such as, *k*-NN, decision tree, etc., especially for the large dataset. The computational complexity of this argument is presented in Section IV (see Table IV). The achievement of SVMs in machine learning logically leads to its possible allowance for large dataset classification. However, the theoretical concept of SVM is very strong but they are not as constructively used in large dataset classification. The main cause is that the training complexities of SVMs are highly reliant on the data size [26]. Recently to speed up the process of SVM classifier the researchers focused on the reduction of the decision trick. Traditional SVMs tend to not perform well as when trained with a complete large dataset than with a set of fine worth samples [27]. The sample selection or active learning methods of SVM is trying to find the intelligent training data for maximum performance, but unfortunately, many scans of the dataset are required [27], [28]. Lei and Govindaraju [28] proposed principal component analysis and recursive feature removal for a decrease of the feature space.

Another type of data reduction method is the clustering technique. Clustering is a process that separates unlabeled data into a finite and distinct group in such a way that the data in the same cluster are alike to each other and the data are dissimilar in different clusters as per certain similarity or proximity measures. Many data reduction methods have been reported by using clustering techniques in the literature [29], [30]. However, clustering-based techniques can decrease the calculation load of SVM, but they are themselves also very complex for large datasets. In this situation, the multiclass problem is more complicated than the binary class problem also in the case of the same training data, because the training time is larger than binary classification. Li et al. [31] proposed a random collection technique and a two-stage SVM classification method for large datasets. Random selection of data is quicker and preferred than other clustering techniques like the data nonpartitioned method. But there is a restriction in this method that the original data should be relatively uniform. Though this approach can work with much smaller training data than other SVM approaches, it requires twice the classifications.

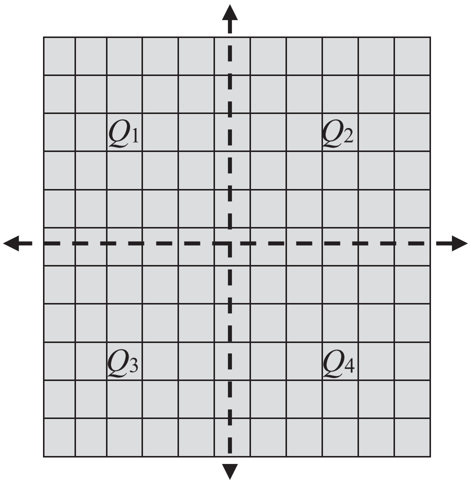


Fig. 4. 7 × 7 Image chip.

Already all the chip images are processed by the previous training sample generation-correction algorithm. The processed chip image is more homogeneous than the original raw chip image. Suppose we have a *K* number of chip images of a particular class. Now we will consider one chip and divide the chip image into four quadrants Q_1, Q_2, $Q_{3,}$ and Q_4 as shown in Fig. 4. Hence, each quadrant will be made of the same bands as the initial Image Chip, i.e., each pixel has four values for four bands in each quadrant. Find the mean value, which is a four-dimensional (4-D) value, of each band in each quadrant. If all the mean values are the same for the four quadrants, we only take those mean values once instead of taking them four times. Otherwise, data redundancy would occur. The number of support vectors will be reduced by this technique. Suppose for *K* number of chip images with $q \times q$ size the number of support vectors will be $K \times q \times q$ of a particular class. So, for all other classes, it is huge data. At the same time, by the proposed methods of training

```
Algorithm 1: Training Sample Generation-Correction.
  Step 1: Generate a maximum number of training sample
          chips of size, say $q \times q$ of those classes (concrete,
          water, vegetation, and unknown class) from the
          known images. We use NIR, G, B, and R band
          images of the high-resolution multispectral
          image.
  Step 2: The training chip sample may not be
          homogeneous due to noise. To test whether the
          chip image is homogeneous or not we must
          process the chip image. The procedure is as
          follows:
  • Find the number of distinct gray values of each band of
    the chip image. Next, find the frequency of the gray
    value of each band of the chip image. Let us suppose in
    a particular chip image there are $k_{nr}, k_g, k_b, k_r$ number
    of distinct gray values in NIR band, green band, blue
    band, and red band, respectively. Let
    $hnr_i, i = 1, 2, ..k_{nr}$ , $hg_i, i = 1, 2, ..k_g$,
    $hb_i, i = 1, 2, ..k_b$ and $hr_i, i = 1, 2, ..k_r$ are the
    frequency of NIR, green, blue, and red bands chip
    image.
  • Compute the mean value, i.e., $\left(\overline{nr}, \bar{g}, \bar{b}, \bar{r}\right) =$
    $\left(\frac{1}{\sum_i^{k_{nr}} hnr_i} \sum_{i=1}^{k_{nr}} nr_i \times hnr_i, \frac{1}{\sum_i^{k_g} hg_i} \sum_{i=1}^{k_g} g_i \times hg_i,\right.$
    $\left.\frac{1}{\sum_i^{k_b} hb_i} \sum_{i=1}^{k_b} b_i \times hb_i, \frac{1}{\sum_i^{k_r} hr_i} \sum_{i=1}^{k_r} r_i \times hr_i\right)$ where
    $(nr_i, g_i, b_i, r_i)$ are the gray values of the chip image.
  • Find the standard deviation $sd =$
    $\frac{1}{M} \sum_{i = 1}^{M} \sqrt{(nr_i - \overline{nr})^2 + (g_i - \bar{g})^2 + (b_i - \bar{b})^2 + (r_i - \bar{r})^2}$
    , where $M = q^2$ is total number of pixels of the chip
    image. In our case, $q = 7$.
  • If sd is very small, $sd < \varepsilon$ then the chip is perfectly
    homogenous, otherwise, apply the odd man out
    procedure. The odd man out procedure is as follows:
  √ Find the mode $(m_{nr}, m_g, m_b, m_r)$ , i.e., the maximum
      number of frequencies of each band of the chip image,
      i.e., $m_{nr} = {}_{i=1}^{k_{nr}} max\ \{hnr_i\}$, $m_g = {}_{i=1}^{k_g} max\ \{hg_i\}$,
      $m_b = {}_{i=1}^{k_b} max\ \{hb_i\}$ and $m_r = {}_{i=1}^{k_r} max\ \{hr_i\}$.
  √ Next, compute the distances between the mean gray
      value and all the pixel gray values of the chip image,
      i.e., $d_i =$
      $\sqrt{(nr_i - \overline{nr})^2 + (g_i - \bar{g})^2 + (b_i - \bar{b})^2 + (r_i - \bar{r})^2}$ , $i$
      $= 1, 2, \ldots, M$.
  √ If $d_i \geq \delta \times sd, i = 1, 2, \ldots, M$ (where $\delta$ is very
      small value, which is called Gaussian multiplier) then
      replace the ith pixel value by mode value of the image
      chip. Otherwise, no change for the ith pixel gray value
      of the chip image.
```

sample correction and data formation, the maximum number of support vectors will be $K \times 4$ of that particular class, which is very less and useful for data reduction. This data file will be a single training data file for all classes in five dimensions. The first column is a class label (say, concrete, water, vegetation, and other class labels are 1, 2, 3, and 4, respectively), and the second, third, fourth, and fifth columns are mean gray values of four bands NIR, Green, Blue and Red of the corrected chip images. The proposed approach can work with much smaller training data than other SVM approaches.

D. Classification Criteria

In general, an SVM classifier has been designed for only binary classification by which the data are separated into two classes. But more expectation arises in the real-life domain for solving the multiclass problem and it is handled by breaking it down into multiple binary classification problems. There are two breakdown approaches for this:

1) *One-to-Rest* approach—In this approach, the classifier will use "*m*" SVMs if there are "*m*" classes.
2) *One-to-One* approach—In this approach, the classifier will use *m*(*m*-1)/2 SVMs for "*m*" classes.

In this article, we have reserved the One-to-Rest approach since the numbers of classes are less. Four SVM models are generated through training data samples – i) Water versus (Vegetation, Concrete, Others), ii) Vegetation versus (Water, Concrete, Others), iii) Concrete versus (Water, Vegetation, Others) and iv) Others versus (Water, Vegetation, Concrete). Four SVM models were trained separately by applying the linear, polynomial, radial basis function (RBF), and sigmoid kernels. The test dataset consisting of the mean values of all four bands is passed on to the SVM model and the prediction probability that the test data belongs to a particular class has been determined. The predicted class is the classifier that gave the highest probability. In this way, the SVM predicts the class label for each test data.

To determine the accuracy of this classification, we have constructed a confusion matrix by taking the columns as the true class and the rows as the predicted class. Four important metrics are computed from the confusion matrix namely true positive (*TP*), true negative (*TN*), false positive (*FP*), and false negative (*FN*). The accuracy is calculated as $\frac{TP+TN}{TP+TN+FP+FN}$.

E. Postprocessing Technique

The salt-and-pepper noise is a common characteristic in the classification image of remote sensing images due to several factors and to eliminate these noises this article emphasizes on postprocessing technique for pixelwise classification so that the final result should be homogeneous. Many postprocessing classification techniques are available to improve the final classification result in the literature. In Su [18] postclassification technique, a kernel of size 3 × 3 has taken and LCF is computed for the central pixel in the neighbor. Second, edge pixels are not considered in the classification map because many neighboring pixels cannot be an element at the present position of the kernel. Ouma and Tateishi [53] proposed a simple filter-based postclassification technique for the elimination of salt-and-pepper noise. Tu et al. [54] proposed a combination of several filters such as a weighted median filter, bilateral filter, and fast median filter for smoothing the edges, occlusions, and other smooth regions of the flow field. Large window size will give a better result but at the same time many meaningful classes will be merged and as a result, they might be removed. Also, overall accuracy will be decreased and computationally expensive.

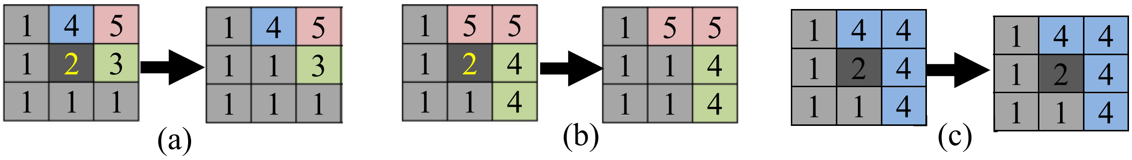


Fig. 5. Class transfer using a 3 × 3 window: (a) Case 1: $No_{MCV} \geq 5$, (b) Case 2: No_{MCV} is greater than the number of other classes occurrence and (c) Case 3: No_{MCV} is the same for the number of other class occurrence.

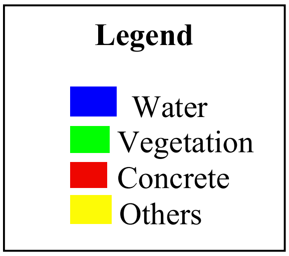


Fig. 6. Color legend of four classes.

The classification result can be improved by the majority class-based filtering method. The result of a particular class region may not be smooth for the noise and illumination effect of the original image. Each distinct class has a unique value and limited pixels within a class may be signified as dissimilar class values for noise. These misclassification results will be modified over a window-based smoothing operator. The smoothing technique is based on class values of all pixels within the a $w \times w$ moving window for the class value transformation of the center pixel. First, the majority class value, is defined as: within a $w \times w$ window if more than $[\frac{w \times w}{2}] + 1$ (where $[a]$ means the greatest integer $\leq a$) pixels class values are same then the class value is called *majority class value* (*MCV*) within the window. *MCV* is used in the smoothing technique. Let No_{MCV} is the number of pixels having majority class value within the window. Fig. 5 shows the different cases of class transfer of the center pixel within a 3 × 3. The postprocessing technique is described below.

Step 1: Take a moving window of size $w \times w$. In our case w = 3.

Step 2: Do not operate the window postprocessing operation in the border pixels of the classified image, i.e., for w = 3 exclude the operation for the first row, first column, last row, and last column.

Step 3: Find the class value of all pixels from the classified image within the window.

Step 4: Find the MCV at the present position of the window. If $No_{MCV} \geq [\frac{w \times w}{2}] + 1$ (for w = 3, $No_{MCV} \geq 5$) replace the class value of the center pixel position by MCV (Fig. 5(a): *Case 1:* Confirm majority Case: center pixel with class value 2 have 5 neighboring pixels with class value 1).

Step 5: If $No_{MCV} < ([\frac{w \times w}{2}] + 1)$ (for w = 3 it is 5) then check any one of the cases, which one is satisfied follow that case.

If No_{MCV} is greater than the number of other classes occurrence within the window then replace the class value of the center pixel position with MCV (Fig. 5(b): Case 2).

If both the number of majority class values (No_{MCV}) and the number of other class occurrences are the same then no change for the center pixel class (Fig. 5(c): Case 3).

Step 6: Continue Steps 3–5 for all pixels of the classified image.

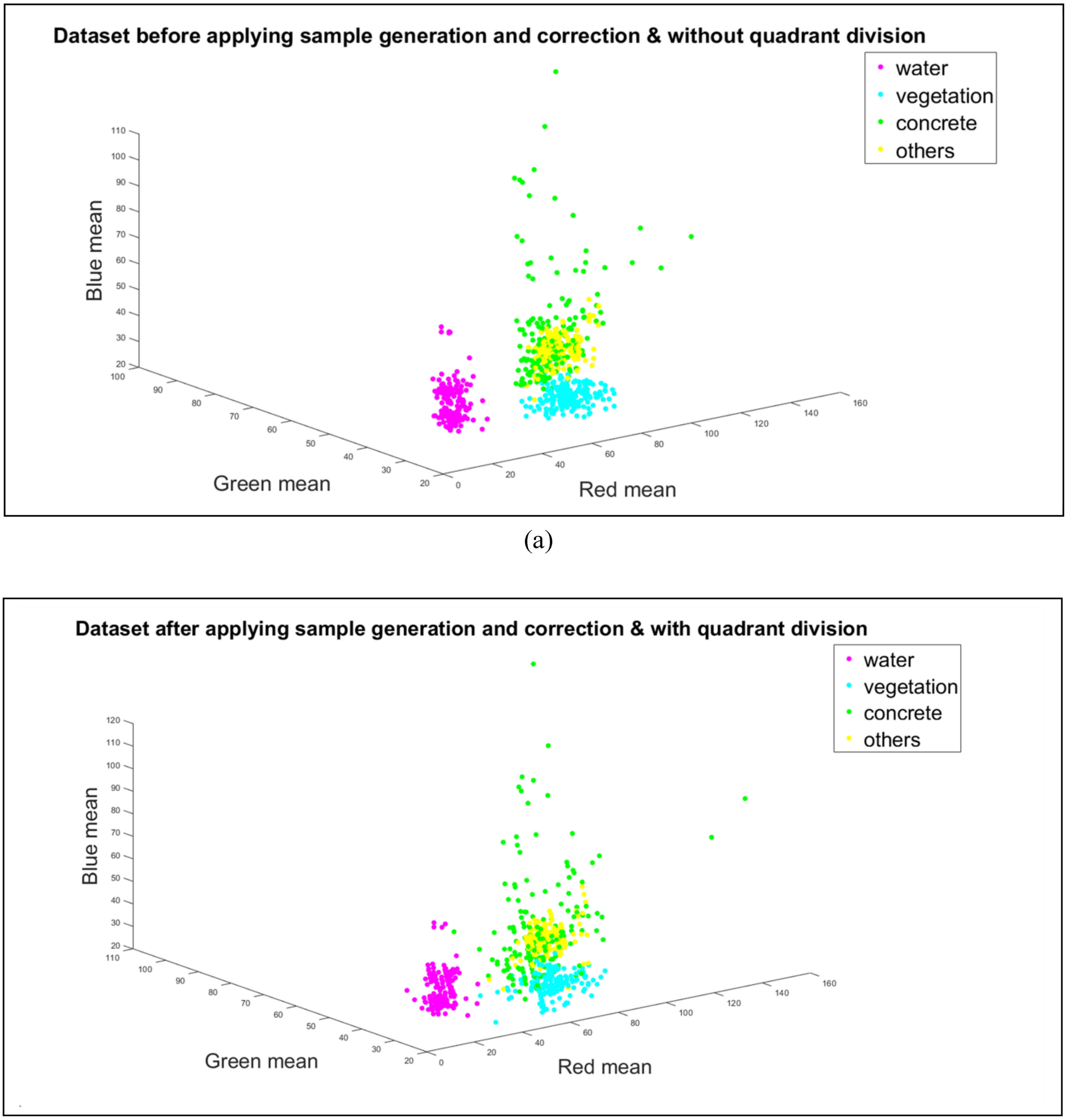
IV. Experimental Results and Discussion

We have established the projected algorithm with numerous scenes obtained from Kompsat-3A multispectral image. We are not using the panchromatic image for our classification procedure. We have used only those four bands in the multispectral mode for classification purposes.

A. Class Data Collection of RS Image

Already we have mentioned that four classes, e.g., concrete, water, vegetation, and others are considered as per the requirement of the project. The training image chips are collected as above in Section III-A. In our case, we have considered the training sample image chip of size 7 × 7 for each of the four classes and taken 200 such training sample chips for each class. The parameters ε and δ for training sample generation and corrections have taken 1.0 and 0.1, respectively, for all chips. In a particular class, there are $7 \times 7 \times 200 = 9800$ training samples in 4-D (since four bands image) and so for four different classes, the total number of training samples is 39 200.

We have estimated the error matrices of the raw training sample chip image and the sample generation-correction chip image for the judgment of the effect of the training sample correction method. We have taken 7 × 7 image chips for each class and so, 49 samples are considered for each class. The total number of samples for all 4 classes is 196. Notation of classes are as follows Water: W, Concrete: C, Vegetation: V, Other: O, and Total: T in Table II, which is the error matrices of the proposed classification method by using raw training sample chip image and the sample generation-correction chip image. As we know that the two sets of classes should be identical and the diagonal line (in gray color) shows the pixels that have the same land cover class in both datasets. Off-diagonals have been misclassified. We have seen from Table II that the overall accuracy for raw training sample data is 75% and the overall accuracy for a corrected training sample is 91.3%. But the overall accuracy measure is insufficient because this is an average value, which does not reflect that the error was distributed evenly across all the classes. Even also one maximum misclassified class will



(a)

(b)

Fig. 7. Scattered plot of training samples. (a) Original training sample plot. (b) Reduced training sample plot.

affect the overall accuracy. So, we have computed also the *user's accuracy* and *producer's accuracy*. The user's accuracy for water, concrete, vegetation, and other classes with raw training samples are 88%, 73.5%, 77%, and 61%, respectively, whereas the producer's accuracy of the same order classes with raw training samples is 86%, 60%, 83%, and 75%, respectively. Similarly, the user's accuracy for water, concrete, vegetation, and other classes with corrected training samples are 98%, 92%, 96%, and 80%, respectively, whereas the producer's accuracy of the same order classes with corrected training samples are 96%, 85%, 89%, and 98%, respectively. In this context, we can conclude from all types of accuracies that the corrected training samples give better classification accuracy than the raw training samples.

B. Results of RS Multispectral Image

We have used all four multispectral bands blue, green, red, and NIR for our classification technique and all four Kernels linear, polynomial, radial basis function (RBF), and sigmoid as (8), (9), (10), and (11), respectively, in the SVM model of the proposed technique. Also, we have compared the results elaborately by various combinations: i) using four different types of Kernels with and without quadrant division, ii) the effect of

TABLE III
COMPARISON OF THE CLASSIFICATION ACCURACY

Case → Kernel ↓	C1 In %				C2 In %				C3 In %				C4 In %			
	W	V	C	O	W	V	C	O	W	V	C	O	W	V	C	O
LK	100	97.5	80	77.5	100	96.9	79.8	78.8	100	93.4	74.7	76.9	100	98.1	81.6	78.9
PK	100	100	85	85	100	97.8	86.3	90.0	100	95.3	84.3	85.3	100	100	95.2	92.5
GK	100	100	88	87.2	100	98.3	89.2	91.1	100	96.7	85.3	86.6	100	100	97.8	95.1
SK	97.5	96.3	77.5	76.3	90	87.5	65.0	67.5	88.8	88.7	67.2	67.8	98.4	97.2	78.7	79.1

results with and without training sample corrections, and iii) the effect of results before and after the postprocessing technique. Also, we have elaborately compared the proposed method with different other classifiers like K-NN, Decision Tree (DT), and Random Forest (RF) with/without four combinations of sample generation-correction methods and with/without quadrant divisions, i.e., C1: without sample generation-correction and without quadrant division, C2: with sample generation-correction and without quadrant division, C3: without sample generation-correction and with quadrant division and C4: with sample generation-correction and with quadrant division (proposed algorithm).

The classes *C*, *W*, *V* and *O* are labeled with the color legend as shown in Fig. 6 for all classified images. Already we have discussed the training sample image chip in this section as above. Also, we have discussed the parameters ε and δ in the same section. A total of 39 200 training samples are considered in our case study. But by the proposed Algorithm 1 and quadrant division methods (Section III-C), the size of the total training samples after reduction from the original training samples is 3200. Though we have considered all four bands for classification and other procedures but due to constraints of graphical representation for four bands, we have shown the scattered plot of the training samples in 3-D (blue, green, and red). The scattered plot of the original training samples 39 200 and modified training samples 3200 for blue, green, and red bands are shown in Fig. 7(a) and (b), respectively. Each dot of Fig. 7(a) and (b) are represented as 49 pixels for each 4 class, respectively. If we closely look then we can see the training sample pattern distributions of the modified training samples are almost the same for all classes as the original and as a result, the classification result will not be affected. At the same time, the training time will be less for the classification technique by using modified training samples.

Out of 200 image chip samples per class, 90% i.e., 180 image chip samples are used for training, and the remaining 10%, i.e., 20 image chip samples for testing. We have applied the proposed algorithm to these samples for different kernels. Table III shows that the classification accuracies of the proposed algorithm (C4) for different kernels are much better than the rest other conditions. Also, it is noticed that the Radial basis function (Gaussian) kernel gives better accuracy than other kernels in the proposed SVM-based classification approach.

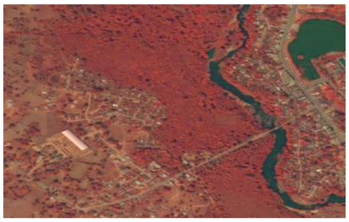


Fig. 8. Original multispectral image of size 512 × 512.

Here we are considering the Bijapur area, Chhattisgarh State, India (LAT: 18.805760, LONG: 80.804810), which is mainly forest, small hilly, small concrete structures with road, river/water body, etc. classes. Fig. 8(a) shows the original multispectral image of size 512 × 512. As we have discussed earlier regarding the four different kernels LK, PK, GK, and SK, and four different cases C1, C2, C3, and C4. The proposed algorithm is case C4 with a postprocessing technique. Since there are four kernels (LK, PK, GK, and SK) without/with postprocessing steps (two stages) so a total of eight sets of classification results in a particular case. The training sample data file is the same for all classification results. The training samples of all classes are generated by the previous steps as already discussed in Section IV-A. Now we have discussed all four cases' results with different combinations in separate sections as below.

1) Case C1: Without Sample Generation-Correction and Without Quadrant Division: First, the classification results of the original image Fig. 8 by using different kernels LK, PK, GK, and SK for case C1 without postprocessing technique are shown in Fig. 9(a), (b), (c) and (d), respectively. The final corresponding results by applying the postprocessing technique are shown in Fig. 9(e), (f), (g) and (h), respectively, for case C1.

2) Case C2: With Sample Generation-Correction and Without Quadrant Division: Next, we are representing the classification results of the original image Fig. 8 by using different kernels LK, PK, GK, and SK for case C2. Fig. 10(a), (b), (c), and (d) shows the classification results without a postprocessing technique by using kernels LK, PK, GK, and SK, respectively. Finally, Fig. 10(e), (f), (g), and (h) show the corresponding classification results with the postprocessing technique by applying different kernels LK, PK, GK, and SK, respectively, for case C2.

3) Case C3: Without Sample Generation-Correction and With Quadrant Division: Here we are discussing the results of case C3. The classification results of Fig. 8 using different kernels LK, PK, GK, and SK for case C3 without and with postprocessing technique are shown in Fig. 11. Fig. 11(a), (b),

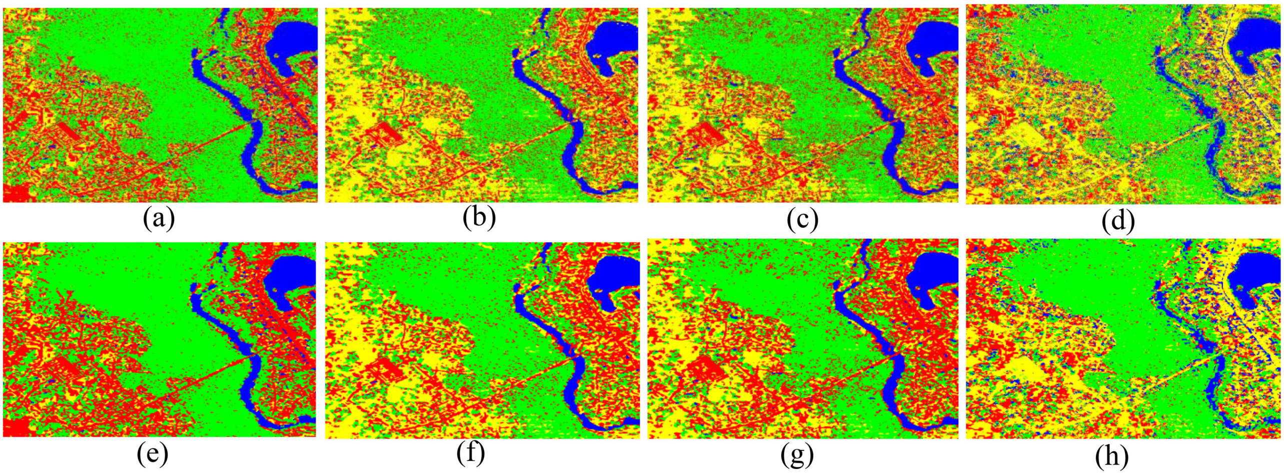


Fig. 9. Classification results of Fig. 8. Without postprocessing technique of case C1. (a) Using kernel LK. (b) Using kernel PK. (c) Using kernel GK. (d) Using kernel SK. Classification results with postprocessing technique of case C1: (e) using kernel LK, (f) using kernel PK, (g) using kernel GK, and (h) using kernel SK.

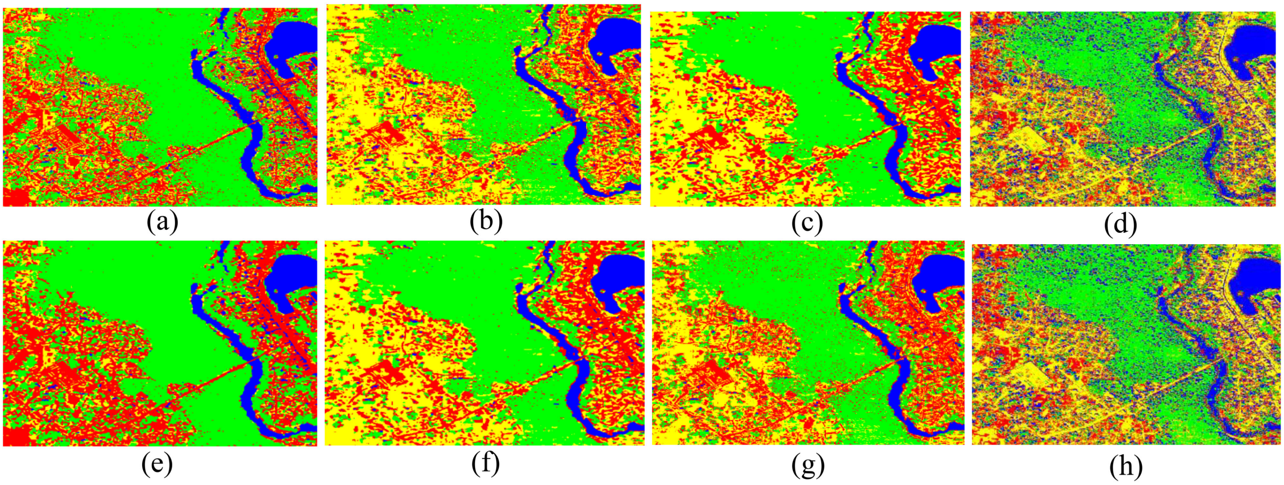


Fig. 10. Classification results of Fig. 8: without postprocessing technique of case C2. (a) Using kernel LK. (b) Using kernel PK. (c) Using kernel GK. (d) Using kernel SK. Classification results with postprocessing technique of case C2. (e) Using kernel LK. (f) Using kernel PK. (g) Using kernel GK. (h) Using kernel SK.

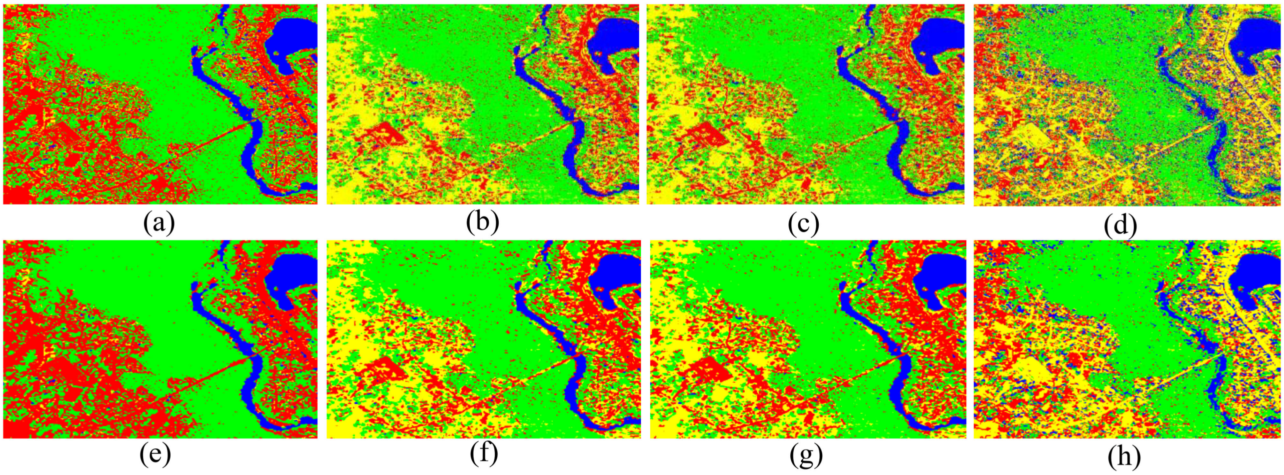


Fig. 11. Classification results of Fig. 8: without postprocessing technique of case C3. (a) Using kernel LK, (b) Using kernel PK. (c) Using kernel GK. (d) Using kernel SK. Classification results with postprocessing technique of case C3. (e) Using kernel LK. (f) Using kernel PK. (g) Using kernel GK. (h) Using kernel SK.

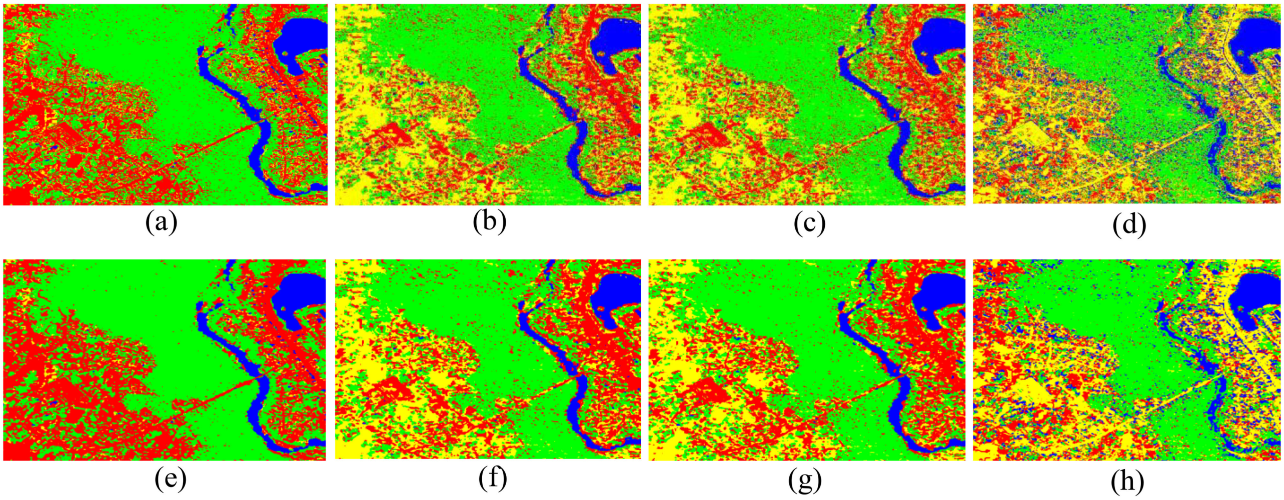


Fig. 12. Classification results of Fig. 8: without postprocessing technique of case C4. (a) Using kernel LK. (b) Using kernel PK. (c) Using kernel GK. (d) Using kernel SK. Classification results with postprocessing technique of case C4. (e) Using kernel LK. (f) Using kernel PK. (g) Using kernel GK. (h) Using kernel SK.

TABLE IV
COMPARISON OF THE CLASSIFICATION ACCURACY WITH VARIOUS CLASSIFIERS

Classifier → Case ↓	K-NN classifier In %				DT classifier In %				RF Classifier In %				SVM with RBF kernel In %			
	W	V	C	O	W	V	C	O	W	V	C	O	W	V	C	O
Case 1 (C1)	98.1	96.8	88.8	88.8	100	95	80	82.5	100	100	86.2	86.2	100	100	88	87.2
Case 2 (C2)	100	96.3	78.8	82.5	100	95	80	85	100	96.2	86.2	90	100	98.3	89.2	91.1
Case 3 (C3)	100	95.3	81.9	83.4	100	95.6	84.1	84.7	100	95.3	84.1	86.8	100	96.7	85.3	86.6
Case 4 (C4)	99.4	97.2	90.3	91.9	100	96.3	92.5	93.8	100	96.9	86.2	87.5	100	100	97.8	95.1

TABLE V
COMPUTATIONAL COMPLEXITY OF DIFFERENT CLASSIFIERS

Cases	Computational Cost (in Sec.)				
	SVM		DT	KNN	RF
Case 1: C1	LK	14.905	14.491	14.563	19.448
	PK	14.773			
	GK	14.729			
	SK	30.903			
Case 2: C2	LK	14.376	14.386	14.1	19.39
	PK	14.544			
	GK	14.938			
	SK	32.434			
Case 3: C3	LK	15.219	14.803	14.2	19.881
	PK	18.115			
	GK	16.19			
	SK	55.154			
Case 4: C4	LK	15.181	14.633	14.282	19.694
	PK	16.866			
	GK	16.899			
	SK	54.22			

(c), and (d) are the classification results by using different kernels LK, PK, GK, and SK, respectively, without postprocessing technique. At the same time, Fig. 11(e), (f), (g), and (h) are the classification results by using different kernels LK, PK, GK, and SK, respectively, with a postprocessing technique.

4) Case C4: With Sample Generation-Correction and With Quadrant Division (Proposed Algorithm): Finally, we are representing the classification results of the original image Fig. 8 by our proposed algorithm without/with combinations of postprocessing techniques. The classification results by using different kernels LK, PK, GK, and SK for case C4 without a postprocessing technique are shown in Fig. 12(a), (b), (c) and (d), respectively. The final corresponding results by applying the postprocessing technique are shown in Fig. 12(e), (f), (g), and (h), respectively, for case C4.

After discussions about the results of different combinations, we have seen closely that in many places like the left bottom corner patch, the road structures of the original image are not properly classified by liner kernel and sigmoid kernel, respectively. But the RBF kernel better classified all those places. Also, by the field visit, we have observed that the proposed algorithm gives an improved result than any other cases, and also RBF kernel shows a better result than any other kernel.

5) Comparison With Different Classifiers: Next, we have compared the proposed algorithm with different classifiers SVM, K-NN, decision tree (DT), and random forest (RF). We have applied the different classifiers, K-NN, decision tree, and SVM with RBF kernel (proposed algorithm) and RF with the same data samples. Table IV shows the classification accuracies of the proposed algorithm, as well as the different classifiers and it has been observed that the proposed algorithm under the condition of SVM classifier with RBF kernel gives better results than the rest of the other cases (C1, C2, and C3). Also, it is

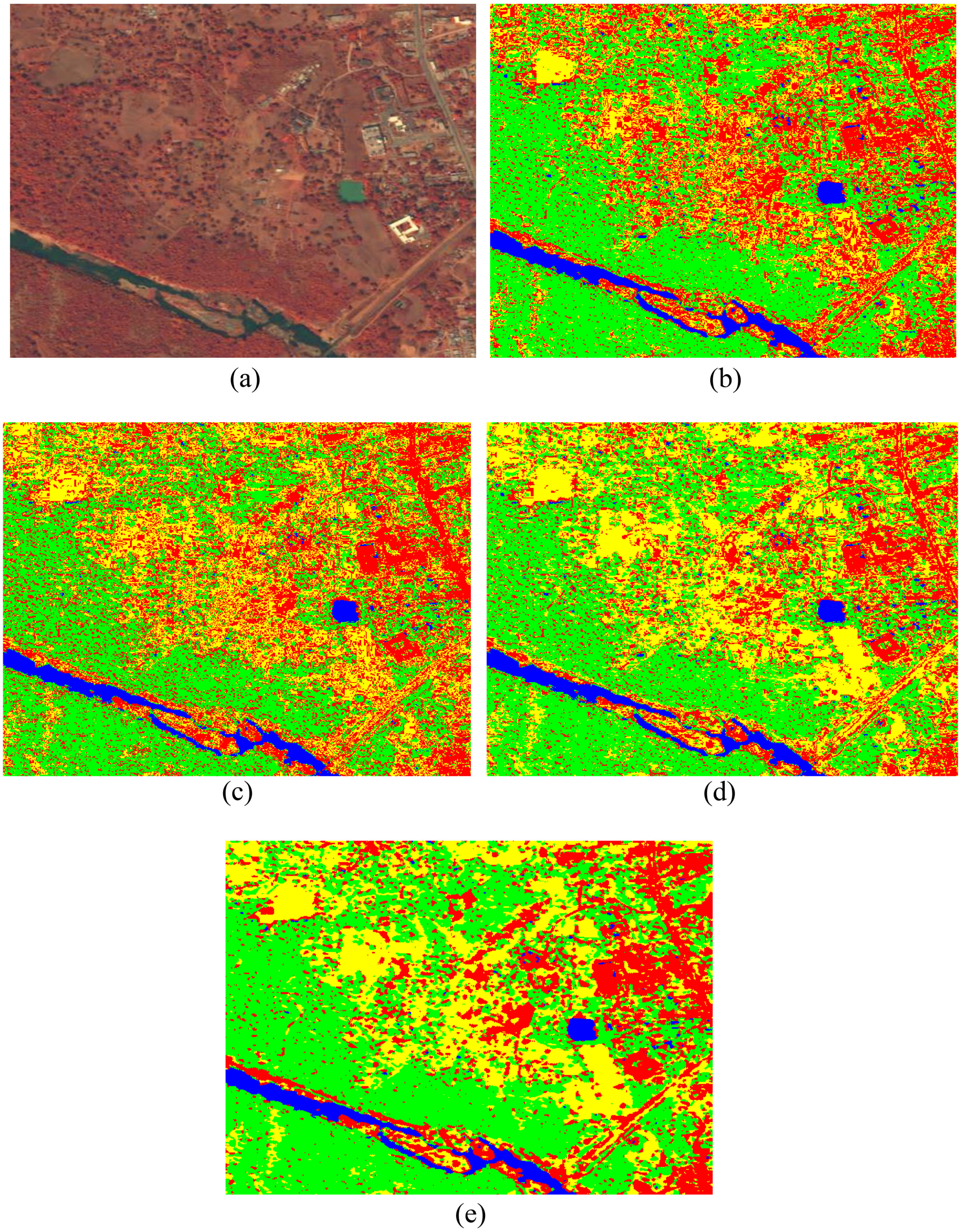


Fig. 13. Classification results for different classifiers. (a) Original image. (b) Classification map by K-NN classifier. (c) Classification map by Decision Tree classifier. (d) Classification map by RF classifier. (e) Classification map by the SVM classifier with RBF kernel by the proposed algorithm.

noticed that the radial basis function (Gaussian) kernel-based SVM classifier gives better accuracy than other classifiers in the environment (Case 4: C4) of the proposed algorithm.

We have tested the proposed algorithm with the same satellite multispectral data by using different classifiers. Here, Fig. 13(a) shows another Bijapur area, Chhattisgarh State, India original multispectral image of size 512 $\times$ 512, which contains mostly the center/road, river/water body, vegetation, and mud of unknown class. The classification results by different classifiers K-NN, DT classifier, RF classifier, and SVM with RBF kernel by the proposed technique are shown in Fig. 13(b), (c), (d) and (e), respectively. We have verified the classification results with Survey of India maps and it is noticed that the proposed algorithm by SVM with RBF kernel classifier gives a better result than the other reported classifiers.

The time complexity is another important factor in the selection of an algorithm. All algorithms are implemented on an HP workstation 64-bit operating system, X64-based processor, Intel(R) Core (TM) i7-7700 CPU @ 3.60 GHz, 16.0 GB RAM (15.9 GB usable). Table V shows the computational complexity of different classifiers. SVM is enormously faster and simple than ANN [34]. We have seen that the computational complexity of the Sigmoid kernel-based SVM classifier is much more expensive in all cases than any other kernels and classifiers. Also, it has been observed from Table IV that the computational complexity of the SVM classifier using the other three kernels in all cases is less expensive than the RF classifier. At the same time, the computational complexities of the SVM classifier using the other three kernels in all cases are a little bit higher than DT and *K*-NN classifiers. Overall, the proposed algorithm has much better accuracy and is computationally not much expensive as other reported classifiers.

V. Conclusion and Future Scope

SVM is a very suitable method with sufficient strong theoretical background for the classification of RS data. Though SVM is the maximum suggested method for remote sensing data classification but a good quality training sample data set is required to train the classifier. We proposed an effective preprocessing technique, training sample generation-correction method with quadrant division for classification of high-resolution RS images. The training sample generation correction is dependent on local statistics. We have elaborately described the training sample generation-correction method, which does not change the distribution of the original pattern. Finally, we have applied a simple filter-based postprocessing technique to improve the classification accuracy. We have studied and compared the SVM classifier with different kernels by the proposed technique for showing its effectiveness. Also, we have compared the results with other classifiers and we have observed that the proposed technique with SVM classifier using RBF kernel shows better results than other classifiers. The methodology can be extended in many different directions. In this case, we have to develop a faster SVM classifier by reducing support vectors from the training dataset. Nowadays high-resolution remote sensing image is available and, in the future, we are planning to develop the classifier in a new direction based on CNN. First, we segmented and extracted the features; then classification of each segment is performed on its feature vector.

References

[1] M. R. Anderberg, *Cluster Analysis for Application*. New York, NY, USA: Academic, 1973.

[2] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.

[3] R. F. Ling, "A probability theory of cluster analysis," *J. Amer. Statist. Assoc.*, vol. 68, no. 3, pp. 159–164, 1973.

[4] A. K. Jain, "Cluster analysis," in *Handbook of Pattern Recognition and Image Processing*, T. Y. Young and K. S. Fu, Eds. New York, NY, USA: Academic, 1986.

[5] R. Dubes and A. K. Jain, "Validity studies in clustering methodologies," *Pattern Recognit.*, vol. 11, no. 4, pp. 235–254, 1979.

[6] J. R. Jensen, *Introductory Digital Image Processing, a remote Sensing Perspective*. Hoboken, NJ, USA: Prentice Hall, 1986.

[7] T. Gilmour, "Image smoothing as an aid to classification in advances in digital image processing," in *Proc. Annu. Conf. Remote Sens. Soc.*, Nottingham, U. K., 1987, pp. 56–54.

[8] D. Chaudhuri and B. Chaudhuri, "A multi-seed non-hierarchical clustering technique for data analysis," *Int. J. Syst. Sci.*, vol. 26, no. 2, pp. 375–385, 1995.

[9] D. Chaudhuri and B. B. Chaudhuri, "A novel multi-seed nonhierarchical data clustering technique," *IEEE Trans. Syst. , Man , Cybern. , B (Cybern.)*, vol. 27, no. 5, pp. 871–876, Oct. 1997.

[10] K. Tan and P. Du, "Combined multi-kernel support vector machine and wavelet analysis for hyper-spectral remote sensing image classification," *Chin. Opt. Lett.*, vol. 9, no. 1, 2011, Art. no. 011003.

[11] G. M. Foody and A. Mathur, "Toward intelligent training of supervised image classifications: Directing training data acquisition for SVM classification," *Remote Sens. Environ.*, vol. 93, no. 1/2, pp. 107–117, 2004.

[12] V. N. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed., Berlin, Germany: Springer, 1995.

[13] G. M. Foody and A. Mathur, "A relative evaluation of multiclass image classification by support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 6, pp. 1335–1343, Jun. 2004.

[14] C.-F. Tsai, "Training support vector machines based on stacked generalization for image classification," *Neurocomputing*, vol. 64, pp. 497–503, 2005.

[15] C.-T. Lin, C. M. Yeh, S.-F. Liang, J.-F. Chung, and N. Kumar, "Support-vector-based fuzzy neural network for pattern classification," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 1, pp. 31–41, Feb. 2006.

[16] R. Strack, V. Kecman, B. Strack, and Q. Li, "Sphere support vector machines for large classification tasks," *Neurocomputing*, vol. 101, pp. 59–67, 2013.

[17] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *Int. J. Remote Sens.*, vol. 28, no. 5, pp. 823–870, 2007.

[18] T. C. Su, "A filter-based post-processing technique for improving homogeneity of pixel-wise classification data," *Eur. J. Remote Sens.*, vol. 49, pp. 531–552, 2016.

[19] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowl. Inf. Syst.*, vol. 33, pp. 1–33, 2012.

[20] P. Gong and P. J. Howarth, "An assessment of some factors influencing multispectral land-cover classification," *Photogrammetric Eng. Remote Sens.*, vol. 56, pp. 597–603, 1990.

[21] C. Li, J. Wang, L. Wang, L. Hu, and P. Gong, "Comparison of classification algorithms and training sample sizes in urban land classification with Landsat thematic mapper imagery," *Remote Sens.*, vol. 6, pp. 964–983, 2014.

[22] D. Phiri, J. Morgenroth, C. Xu, and T. Hermosilla, "Effects of preprocessing methods on Landsat OLI-8 land cover classification using OBIA and random forests classifier," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 73, pp. 170–178, 2018.

[23] F. Melgani and L. Bruzzone, "Classification of hyper-spectral remote sensing images with support vector machines," *IEEE Trans. GRS*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[24] C. W. Hsu and C. J. Lin, "A comparison of methods for multi-class support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[25] L. Cheng, J. Zhang, J. Yang, and J. Ma, "An improved hierarchical multi-class support vector machine with binary tree architecture," in *Proc. Int. Conf. Internet Comput. Sci. Eng.,* 2008, pp. 106–109.

[26] C. C. Chang and C. J. Lin, "Training -support vector classifiers: Theory and algorithms," *Neural Comput.*, vol. 13, pp. 2119–2147, 2001.

[27] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 2000, pp. 999–1006.

[28] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," in *Proc. Int. Conf. Mach. Learn.*, 2000, pp. 839–846.

[29] X. Z. Liu and G. C. Feng, "Kernel bisecting *k*-means clustering for SVM training sample reduction," in Proc. *IEEE 19th Int. Conf. Pattern Recognit.*, 2008, pp. 1–4.

[30] L. Su, "Optimizing support vector machine learning for semi-arid vegetation mapping by using clustering analysis," *ISPRS J. Photogrammetry Remote Sens.*, vol. 64, no. 4, pp. 407–413, 2009.

[31] X. Li, J. Cervantes, and W. Yu, "Two-stage SVM classification for large data sets via randomly reducing and recovering training data," in Proc. *IEEE Int. Conf. Syst., Man Cybern.*, 2007, pp. 3633–3638.

[32] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 144–152.

[33] S. Bernab´e, P. Reddy Marpu, A. Plaza, M. D. Mura, and J. A. Benediktsson, "Spectral–spatial classification of multispectral images using kernel feature space representation," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 288–291, Jan. 2014.

[34] N. Candade and B. Dixon, "Multispectral classification of LANDSAT images: A comparison of support vector machine and neural network classifiers," in *Proc. ASPRS Annu. Conf.*, May 2004.

[35] M. R. Kushalatha, H. S. Prashantha, and B. R. Shetty, "Comparison of image classification techniques and its applications," *Asian J. Convergence Technol.*, vol. VII, no. 1, pp. 58–62, 2021.

[36] G. Mountrakis, J. Im, and C. Ogole, "Support vector machines in remote sensing: A review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 66, pp. 247–259, 2011.

[37] Y. Wang, W. Yu, and Z. Fang, "Multiple kernel-based SVM classification of hyperspectral images by combining spectral, spatial, and semantic information," *Remote Sens.*, vol. 12, 2020, Art. no. 120.

[38] C. Savas and F. Dovis, "The impact of different kernel functions on the performance of scintillation detection based on support vector machines," *Sensors*, vol. 19, 2019, Art. no. 5219.

[39] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, "Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 6308–6325, 2020.

[40] E. Blanzieri and F. Melgani, "Nearest neighbor classification of remote sensing images with the maximal margin principle," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1804–1811, Jun. 2008.

[41] G. Alimjan, T. Sun, Y. Liang, H. Jumahun, and Y. Guan, "A new technique for remote sensing image classification based on combinatorial algorithm of SVM and KNN," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 7, 2018, Art. no. 1859012.

[42] C. Vasilakos, D. Kavroudakis, and A. Georganta, "Machine learning classification ensemble of multitemporal sentinel-2 images: The case of a mixed mediterranean ecosystem," *Remote Sens.*, vol. 12, 2020, Art. no. 2005.

[43] L. Samaniego and K. Schulz, "Supervised classification of agricultural land cover using a modified *k*-NN technique (MNN) and Landsat remote sensing imagery," *Remote Sens.*, vol. 1, pp. 875–895, 2009.

[44] P. T. Noi and M. Kappas, "Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery," *Sensors*, vol. 18, pp. 18, 2018.

[45] A. Kulkarni and A. Shrestha, "Multispectral image analysis using decision trees," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, pp. 11–18, 2017.

[46] L. Xuerong, X. Qianguob, and K. Lingyan, "Remote sensing image classification method based on evidence theory and decision tree," in *Proc. Multispectral, Hyperspectral, Ultraspectral Remote Sens. Technol., Techn., Appl., III*, vol. 7857, 2010 Art. no. 78570Y.

[47] R. Sharma, A. Ghosh, and P. K. Joshi, "Decision tree approach for classification of remotely sensed satellite data using opensource support," *J. Earth Syst. Sci.*, vol. 122, no. 5, pp. 1237–1247, 2013.

[48] Q. Lu, Y. Ma, and G. S. Xia, "Active learning for training sample selection in remote sensing image classification using spatial information," *Remote Sens. Lett.*, vol. 8, no. 12, pp. 1211–1220, 2017.

[49] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A review on data pre-processing techniques toward efficient and reliable knowledge discovery from building operational data," *Front. Energy Res.*, vol. 9, 2021, Art. no. 652801, doi: 10.3389/fenrg.2021.652801.

[50] L. Capodiferro, L. Costantini, F. Mangiatordi, and E. Pallotti, "Data pre-processing to improve SVM video classification," in *Proc. 10th Int. Workshop Content-Based Multimedia Indexing*, 2012, pp. 1–4, doi: 10.1109/CBMI.2012.6269801, 2012.

[51] M. Salvi, U. R. Acharya, F. Molinari, and K. M. Meiburger, "The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis," *Comput. Biol. Med.*, vol. 128, 2021, Art. no. 104129.

[52] H. S. Obaid, S. A. Dheyab, and S. S. Sabry, "The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning," in *Proc. 9th Annu. Inf. Technol., Electromechanical Eng. Microelectron. Conf.*, 2019, pp. 279–283, doi: 10.1109/IEMECONX.2019.8877011.

[53] Y. O. Ouma and R. Tateishi, "Urban-trees extraction from Quickbird imagery using multiscale spectex-filtering and non-parametric classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 63, pp. 333–351, 2008.

[54] Z. Tu, A. N. van der, C. V. Gemeren, and R. C. Veltkamp, "A combined post-filtering method to improve accuracy of variational optical flow estimation," *Pattern Recognit.*, vol. 47, no. 5, pp. 1926–1940, 2014.

[55] G. M. Foody, A. Mathur, C. Sanchez-Hernandez, and D. S. Boyd, "Training set size requirements for the classification of a specific class," *Remote Sens. Environ.*, vol. 104, pp. 1–14, 2006.

[56] P. M. Mather, *Computer Processing of Remotely-Sensed Images*, 3rd ed., Chichester, U.K.: Wiley, 2004.

[57] T. Van Niel, T. McVicar, and B. Datt, "On the relationship between training sample size and data dimensionality: Monte Carlo analysis of broadband multi-temporal classification," *Remote Sens. Environ.*, vol. 98, pp. 468–480, 2005.

Manish Pratap Singh received the B.Tech. degree in computer science and engineering from Institute of Engineering and Technology Lucknow, Lucknow, India, in 2007, and the M.Tech. degree in computer science and engineering from Indian Institute of Technology Kharagpur Kharagpur, India, in 2016.

He Joined DEAL, DRDO in 2008 and worked in the area of Polarimetric SAR Image Processing. He is currently the Director of DRDO Young Scientist's Laboratory-Cognitive Technologies at Chennai. His area of interest includes pattern recognition, satellite image processing, and cognitive surveillance.

V. Gayathri received the B.Tech. degree in computer science and engineering from Prince Shri Venkateshwara Padmavathy Engineering College, Chennai, India, in 2014, and the M.Tech. degree in computer science and engineering from Madras Institute of Technology, Chennai, in 2016.

She is working as a JRF in DRDO Young Scientist's Laboratory-Cognitive Technologies at Chennai. Her research interest includes machine learning, image processing, and pattern recognition.

Debasis Chaudhuri (Senior Member, IEEE) received the B.Sc. degree from Visva-Bharati University, Santiniketan, Bolpur, West Bengal, India, in 1984, the M.Sc. degree in applied mathematics from Jadavpur University, Kolkata, India, in 1987, and the Ph.D. degree in image processing and pattern recognition from Indian Statistical Institute (ISI), Kolkata in 1995.

He was a Project Scientist at Indian Statistical Institute in ISI-ADRIN Project (collaborated by ISI and Department of Space, Govt. of India). He was a Research Associate of Council of Scientific and Industrial Research (CSIR) at ISI, Kolkata. He was a Scientist at Defence Electronics Applications Laboratory, Dehradun, India, from July 1996 to December 2000 and December 2006 to middle of July 2014. He was an Associate Professor of Defence Institute of Advance Technology (DIAT) (Deemed university), Pune during 2001–2003 (July). He was a Visiting Professor with the University of Nebraska, USA, in the Department of Computer Science and Engineering during 2003–2004. He was posted at Integrated Test Range (ITR), Chandipur, Odisha, for a missile tracking project by EOTS system. He was DGM & DMS at DRDO Integration Centre, Panagarh, West Bengal, and he has retired from there. He is currently a Professor at Computer Science and Engineering Department, Techno India University, Kolkata.

He also visited many other Universities and Institutes within India to deliver invited lectures. He has been a member of the program/organizing committees of many national and international conferences. He also served as a Session Chair of international conference (5th International Conference on Signal, Image Processing, Chennai. He made an influential contribution in automatic target detection from multisensor satellite images for Scene Correlation Area Navigation of AGNI Missile guidance system, generated the information map and transmitted it to military intelligence during Kargil war and developed prototype applications for strategic defense intelligence information extraction. He has got lot of experience in advising and guiding others in related fields. He has extensive experience in the field of automatic target detection from satellite imagery. He got many awards from DRDO and other organizations. He has guided many Ph.D. students, M.Tech./M.S. Student Thesis and B. Tech Students Thesis. He has authored or coauthored of many papers in international journals and conferences in the area of his domain expert. He is an Associate Editor and a Reviewer in many international journals. He is a fellow of IETE. His research interests include image processing, pattern recognition, computer vision, remote sensing and target detection from satellite, SAR, Thermal and MMW imageries.