# Knowledge-Based Morphological Deep Transparent Neural Networks for Remote Sensing Image Classification

Dasari Arun Kumar, *Senior Member, IEEE*

***Abstract*—Land use/land cover classification of remote sensing images provide information to take efficient decisions related to resource monitoring. There exists several algorithms for remote sensing image classification. In the recent years, Deep learning models like convolution neural networks (CNNs) are widely used for remote sensing image classification. The learning and generalization ability of CNN, results in better performance in comparison with similar type of models. The functional behavior of CNNs is unexplainable because of its multiple layers of convolution and pooling operations. This results in black box characteristics of CNNs. Motivated with this factor, a CNN model with functional transparency is proposed in the present study. The model is named as Knowledge Based Morphological Deep Transparent Neural Networks (KBMDTNN) for remote sensing image classification. The architecture of KBMDTNN model provides functional transparency due to application of morphological operators, convolutional and pooling layers, and transparent neural network. In KBMDTNN model, the morphological operator preserve the shape/size information of the objects through efficient image segmentation. Convolution and pooling layers are used to produce minimal number of features from the image. The operational transparency of proposed model is coined based on the mathematical understanding of each layer in the model instead of randomly adding layers to the architecture of model. The transparency of proposed model is also because of assigning the initial weights of NN in output layer of model with computed values instead of random values. The proposed KBMDTNN model outperformed similar type of models as tested with multispectral and hyperspectral remote sensing images. The performance of KBMDTNN model is evaluated with the metrics like overall accuracy (OA), overall accuracy standard deviation ($OA_{STD}$), producer's accuracy (PA), user's accuracy (UA), dispersion score (DS), and kappa coefficient (KC).**

***Index Terms*—Deep neural networks, granulation, knowledge encoding (KE), morphological operators, remote sensing image classification.**

## I. Introduction

LAND use/Land cover classification of remote sensing images provides information about the resources on the earth surface [1]. Availability of remote sensing images of various dimensionality in terms of spatial, spectral resolutions and temporal resolutions paved path for the origin of automated models to classify the images. The automated models such as statistical models are used for remote sensing image classification. The traditional statistical models like Baysian classifier depend on the statistical distribution of samples/pixels. In most of the remote sensing applications, the image data is not statistically well distributed. In the past decade, machine learning (ML) models are popularly used for image classification. The advantages of machine learning models are due to learning and generalization ability. Also, ML based classifiers produces better classification results with unevenly distributed data. The basic classifiers such as decision trees, random forests, K-nearest neighbour, minimum distance to mean, Bayesian classifier are used to classify the small size images on pixel by pixel approach. The performance of basic classifiers is not significant in large size images with multiple number of classes. With this motivation, advanced ML models such as neural networks (NNs), support vector machines (SVM), and genetic algorithms (GA) are widely used for image classification [2]. The learning and generalization ability of NNs can acquire the information from large size data during the training process [3]. The combination of GA and SVM was proposed by Salehi *et al.* [4] for urban land cover classification using Radarsat-2 PolSAR images. The model outperformed SVM and Wishart model in SAR image classification. The superiority of model is due to three important steps such as feature extraction, feature selection, and image classification in GA+SVM model. The performance of ML models is significant in classifying the images of temporal resolutions. Satalino *et al.* [5] used Kittler–Illingworth method to classify temporal SAR remote sensing data. ML models are also widely used in nonremote sensing image classification. Some of the major applications of ML models in other diversified fields include traffic signals detection [6], brain image classification [7], human pose estimation [8], pathological studies [9], multiple sclerosis (MS) disease detection in magnetic resonance images [10], tea category identification [11], text classification in images [12].

The ML based methods such as NNs work on pixel by pixel based approach during the image classification. The pixel based approach is confine to only minor details in the image. In the recent years, convolutional neural networks (CNNs) with object based image classification approach were proposed to overcome the limitations of pixel based NN model. The CNNs considers both major and minor level details of objects in the image during the classification stage. Due to this reason, CNNs performs better than NNs in image classification. CNNs

Manuscript received August 8, 2021; revised November 3, 2021; accepted February 8, 2022. Date of publication February 14, 2022; date of current version March 16, 2022. This work was supported by Kandula Sreenivasa Reddy Memorial College of Engineering through seed funded project 2021.

The author is with the Kandula Sreenivasa Reddy Memorial College of Engineering, Kadapa 516003, India (e-mail: arunkumar.mtech09@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2022.3151149

posses the advantages like learning and generalization during the classification. In the recent years, various type of CNN models were suggested for remote sensing image classification. CNN models are used in the diversified fields of remote sensing such as crop classification, land use/land cover classification, vegetation classification, and urban studies. CNNs are majorally applied on multispectral, hyperspectral, and synthetic aperture radar (SAR) remote sensing images. A deep belief networks (DBN) model for Urban mapping using SAR images was proposed by Lv *et al.* [13]. The DBN model was used to extract the effective contextual mapping feature from the image. The performance of DBN model was compared with SVM, NNs, and stochastic expectation maximization. The DBN model uses supervised learning approach during the classification approach. Zhong *et al.* [14], proposed large patch CNN (LPCNN) model with both supervised and semisupervised approach. The LPCNN model is used to generate large number of scene patches for the feature learning. In LPCNN model, fully connected (FC) network is replaced with global average pooling. The performance of LPCNN model was compared with similar type of model by testing with hyperspectral remote sensing images. Higher dimensionality and spare availability of labeled data are two important challenges with multispectral and hyperspectral remote sensing datasets. Motivated with this factor, Romero *et al.* [15], proposed greedy layer wise pretraining CNN model to extract efficient features from the image. The superiority of greedy layer CNN model over standard principal component analysis (PCA) and kernel based PCA models was justified with multispectral and hyperspectral remote sensing datasets. Extracting efficient features from the higher dimensional images is a challenging task in remote sensing image classification. Motivated with this factor, multiscale CNN (mCNN) model was suggested by Zhao *et al.* [16] to extract deep features from hyperspectral remote sensing images. The mCNN model transforms the original data into a pyramid structure such that each layer has spatial information at multiple scales. The mCNN model then automatically extracts high-level spatial features using multiscale training datasets. The superiority of mCNN model over similar type of models is justified by testing with hyperspectral and multispectral remote sensing images. The models such as hierarchial deep CNN [17], extreme learning based CNN [18], rotation equivariant CNN [19], markov random field CNN [20], hybrid spectral CNN [21], graph-based CNN [22], resnet [23], multitask CNN [24] were recently proposed for hyperspectral remote sensing image classification. The CNN models are also used in nonremote sensing image classification such as breast cancer detection in mamographic images [25] and object recognition in digital images [26].

The architecture of CNN consists of multiple layers like input layer, convolutional and pooling layers, and FC neural network (FCNN) [27]. The usage of multiple layers in CNN model results in transforming the input pixel values in an image to minimum number of features by applying convolution and pooling operators [28]. The increase in number of layers in CNN, results in lack of operational transparency where the functional mechanism of CNN cannot be explained in detail and thus, CNNs are called as black boxes. Operational transparency of deep learning model is based on interpretability of functional mechanism of each layer in deep learning model instead of randomly applying the layers. The interpretability of deep learning model is coined based on the mathematical understanding of each layer.

Morphological operators like erosion, dilation, opening, and closing considers the shape/size features in the remote sensing images [29], [30]. Usage of cascaded morphological operators produce shape/size features like lines, edges, and boundaries of objects in the image through segmentation [31], [32]. The cascaded operators are opening with closing and closing with opening [33]. The features extracted using morphological operators are fed as an input to the next layer to classify the objects in the remote sensing images.

A series of convolution and pooling layers are used to reduce the number of redundant features in the remote sensing image [34]. In the convolutional layer, a spatial convolution is implemented on the input pixels through moving average method [14]. Convolution process provides the information related to the edges, lines, and boundaries of objects in the image. Pooling layers are used to reduce the redundant information in the image [18]. The architecture of CNNs consists a series of convolution—pooling layers. The informative features extracted using series of convolution and pooling layers are fed as an input to a FCNN. Furthermore, the NN is trained with the labeled data using back propagation learning algorithm [16]. NNs are known as black boxes because of nontransparent architecture, where the functional mechanism is not clear.

An improved NN model named as transparent neural networks (TNNs) [also called as Granular neural networks (GNNs)] were suggested to overcome the black box nature of NNs [35], [36]. The architecture of TNNs is built based on the *If–Then* rules and the functional mechanism of TNNs can be explained based on the *If–Then* rules. TNNs operate on the information granules and the computations performed on the granules are called granular computing. A granule is defined as the group of objects or a class of objects with similarity. There exists various methods of remote sensing data granulation. Pal and Mitra [37] suggested class un-related (CUR) granulation in which the input pixel is granulated in to three linguistic variables called *low*, *medium*, and *high*. An improved method of data granulation called class related (CR) granulation of the input pixel was suggested by Pal [38], [39]. In CR granulation, the input pixel is granulated depending on the number of classes in dataset.

During the training of CNN, the performance of TNNs is based on the learning algorithm. The initial weights of the TNN model play an important role in effective learning. Traditionally, during the training stage of TNN, the initial weights of TNN are assigned with random values [40]. The better way of assigning the initial weights is based on the values, which are computed using soft sets like fuzzy sets [41]. The process of assigning the initial weights of TNN using the computed values is called knowledge encoding (KE) [40]. KE provides better learning ability in TNN. Banerjee *et al.* [41], suggested a fuzzy set based KE in NN. In this study, the required knowledge was computed using rough set based functional dependency. Avatharam *et al.* [42], suggested an improved method of computing initial weights using rough set based functional dependency.

Mellouli *et al.* [43], suggested morphological CNN (MCNN) model for image classification. The architecture of MCNN model consists of morphological operators, series of convolutional and pooling layers, and FCNN layer. In MCNN model, counter-harmonic mean morphology (CHMM) with basic morphological operators like erosion and dilation is used to preserve the shape/size features of the objects in the image. With erosion, the boundary details of the object get improved and with dilation the size of the object gets increased. Erosion and dilation can individually enhance specific features of the objects in the image. The individual application of erosion and dilation operators can only preserve the shape/size information partially. This is the first limitation of MCNN model. The limitation of MCNN model in using morphological operators like erosion and dilation can be handled with cascaded operators. The cascaded operators are combination of erosion, dilation, opening, and closing. The cascaded operators provide the scope to preserve the shape/size information of objects in the image than applying the operators individually [33]. Combination of erosion and dilation provides the scope to preserve the information related to shape/size features of objects (like lines, boundaries, and edges) than applying the operators individually [44]. Also, the output layer of CNN model consists of FCNN. The FCNN model lacks in operational transparency in its architecture. Lack of operational transparency in FCNN of MCNN model is the second limitation. Motivated with these two limitations of MCNN, a Knowledge Based Morphological Deep Transparent Neural Networks (KBMDTNNs) model is proposed in the present study for remote sensing image classification. The model consists of advantages like 1) cascaded morphological operators (closing followed by opening and opening followed by closing) used in the preprocessing stage of KBMDTNNs for better image segmentation and preserve the shape size features of the objects in the image, 2) The optimum number of convolutional and pooling layers and 3) The knowledge encoded (KE) TNN at the output layer. The layers used in the proposed model were selected based on the functional mechanism of each layer. Thus, the proposed model has optimum number of layers unlike in conventional DNN models, where the number of layers are randomly selected. The data granulation and KE in the proposed model supports better learning from the data in less number of iterations and reduces the computational time during the training stage in comparison with conventional DNN and MLP models. In general, the TNNs are built using the *if–then* rules extracted from the data. In the present study, the proposed model with TNN is a FC network and the initial weights of TNN are assigned with FRSD based method. The architecture of TNN in proposed model is not built using the *if–then* rules. The transparency in the proposed model is coined based on KE in the form of initial weights. The architecture of TNN considered in the present study is similar to fuzzy rough GNN suggested by Ganivada *et al.* [42]. The proposed model attains the transparency due to KE instead of *if–then* rules in conventional TNNs. These three advantages of KBMDTNNs model provide the operational transparency in the functioning mechanism of CNN model for remote sensing image classification. The superiority of proposed KBMDTNNs model over similar type of models is tested with both multispectral and hyperspectral remote sensing image classification.

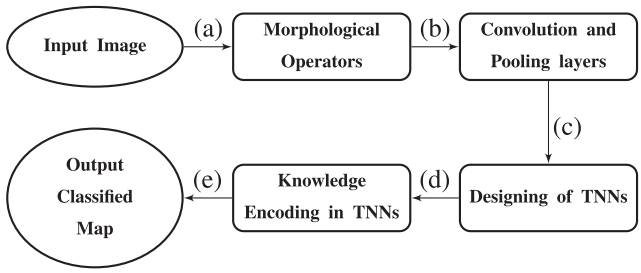


Fig. 1. KBMDTNN with input image, morphological operators, convolutional and pooling layers, TNN and KE in TNNs. (a) Cascaded morphological operators are applied on input image to extract the shape/size features. (b) The resultant image obtained after the morphological operations is passed through the series of convolution and pooling layers to obtain the finite details of edges, lines and boundaries. (c) The informative feature vectors obtained from the previous layer are used to design the architecture of TNN. (d) The initial weights of TNN are assigned with the values computed using FRSD. (e) During the testing, the class label is assigned to the input image.

The remaining sections of the present work is presented as follows. The proposed KBMDTNN model for remote sensing image classification is presented in Section II. Description of various type of remote sensing image datasets used in the study and the results and discussions of the work is given in Section III. The conclusions drawn from the work are given in Section IV.

## II. Proposed Model for Remote Sensing Image Classification

The proposed KBMDTNN model consists of four stages in its architecture such as input image, morphological operators, convolutional and pooling layers, and knowledge encoded TNN layer. The functional block diagram of KBMDTNN model is given in Fig. 1. The input layer consists the image used for training/testing the model. The morphological operators are used to extract the shape/size features from the input image. The convolutional and pooling layers are used to increase the information content in the image and to remove the redundancy features in the image, respectively. The TNN model is used to classify the input image. In the training stage, the initial weights of TNN model are assigned with the values computed using fuzzy rough set based feature dependency. The functional blocks of KBMDTNN model are given in following sections. The functional mechanism of each layer of proposed model (see Fig. 1) is explained mathematically in the following sections to make the architecture of KBMDTNN model as transparent.

### A. Input Image

The input image of KBMDTNN model consists of pixel values representing the reflectance from the objects on the earth surface. The dimensions of input image varies from monochromatic to hyperspectral remote sensing images. The dimensions of hyperspectral input image are more in comparison with the dimensions of monochromatic image.

### B. Morphological Operators

The input image is preprocessed using cascaded morphological operators to obtain the image segmentation with the features like edges, lines, and boundaries. The shape/size information in the form of feature values is obtained from the input image by applying the basic morphological operators like erosion, dilation, opening, and closing. Application of erosion operator on the input image preserves the information content of region of interest. The erosion operation on input image A by structuring element B is performed according to the following:

$$A \ominus B = \{z|(B)_z \subseteq A\} \tag{1}$$

where $(B)_z$ is structuring element B applied on pixel $z$. Dilation on the input image adds the informative features to the region of interest. Application of dilation operator (B—a sub image of input image) on input image A is implemented according to the following:

$$A \oplus B = \{z|[(\hat{B})_z \cap A] \neq \varnothing\} \tag{2}$$

where $z$ is current pixel in A on which the structuring element B is operating. $(\hat{B})_z$ symmetric rotation of structuring element over the pixel $z$. Opening operation is erosion followed by dilation. Opening eliminates the semiinformational features in the region of interest and preserves the core informational features in the region of interest. The opening operation on input image by structuring element is performed according to the following:

$$A \circ B = (A \ominus B) \oplus B. \tag{3}$$

Closing operation is dilation followed by erosion. Closing includes the semi-informational features to the core informational features in the region of interest. The closing operation of structuring element B on input image A is performed according to the following:

$$A \bullet B = (A \oplus B) \ominus B \tag{4}$$

Application of cascaded operators like opening followed by closing extracts the independent and the core information features of a class in the image. The cascaded operator opening followed by closing on input image is performed according to the following:

$$\text{Cas}(A, B) = (A \circ B) \bullet B. \tag{5}$$

In the present study, a cascaded morphological operator i.e., opening followed by closing is considered to obtain the finite shape/size features from the input remote sensing image. The resultant images obtained after applying basic morphological operators on IRS LISS III image are given in Fig. 2. The input IRS LISS III image of Mumbai city is given in Fig. 2(a). The resultant images obtained after applying multidimensional morphological operators like erosion, dilation, opening, and closing on IRS LISS III Image are given in Fig. 2(b)–(e), respectively.

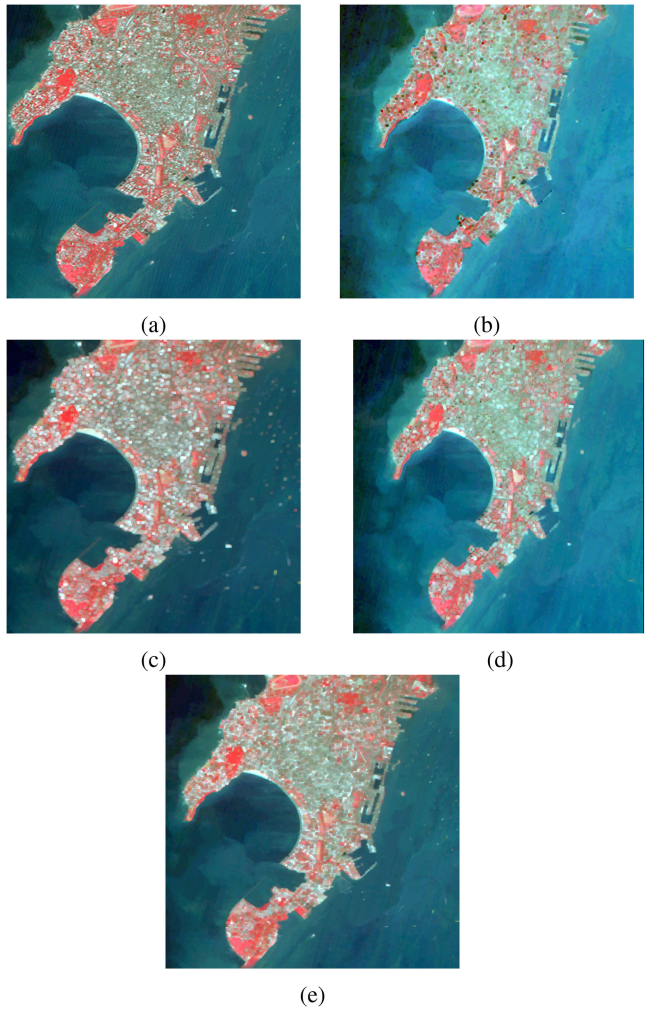


Fig. 2. Basic Morphological operations applied on IRS LISS III image (Mumbai City). (a) Input image (b) Erosion (c) Dilation (d) Opening and (e) Closing.

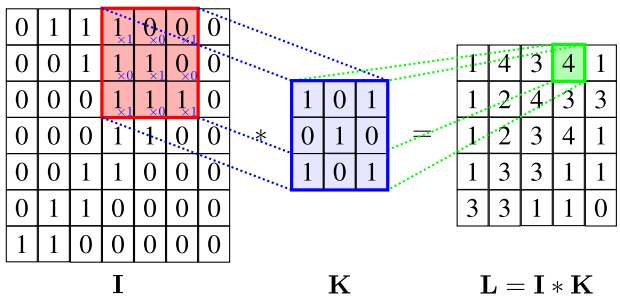


Fig. 3. Example–Convolution operation on input binary image (I, size 7 × 7) with binary structuring element (K, size 3 × 3) generating output image (L, size 7 × 7) [45].

### C. Convolutional and Pooling Layers

The output image of morphological operators is passed to the convolutional layer. In the convolutional layer, a moving window or kernel is operated on the image to extract the lines, edges, and boundaries in a image by moving average method. A symmetric kernel is operated on the image, where the mean of sum of products in the neighborhood of target pixel is replaced in the center pixel. In the present study, an example for convolution operation on input binary image (I) of size (7 × 7) with binary structuring element (K) of size 3 × 3) is considered to generate an output image (L) of size 7 × 7. The convolution operation on example image is given in Fig. 3. In the pooling layer, the redundant features are removed by applying strides and max/min/average operators. As an example, a 2 × 2 max

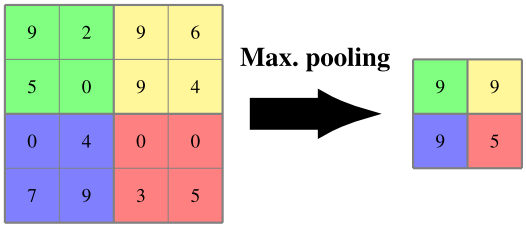


Fig. 4. Example—A 2 × 2 max pooling operation is applied on input image of size 4 × 4 to generate output image of size 2 × 2 [46].

operator is applied on example image of size 4 × 4 to produce the output image of size 2 × 2. The pictorial representation of applying 2 × 2 max pooling operation on input image is given in Fig. 4. In the proposed model, morphological layer preserves the shape/size features in the input image and the series of convolution and pooling layers are used to extract the features like edges, lines and boundaries in the input image. Combination of morphological operators, convolution, and pooling layers will provide the finite level information about the features of objects in the input image. In conventional CNN model, the informative features extracted from image using series of convolution and pooling layers are passed as an input to the FCNN. The limitation of FCNN model is due to the black box nature of the architecture where the functional mechanism of the network is not transparent. Motivated with this factor, a Knowledge based Transparent Neural networks (KBTNN) model is proposed in the present study. In the proposed model, the extracted features from the series of convolution and pooling layers are feed as input to the TNN. The detailed architecture of KBTNN model is given in following sections.

### D. Knowledge Based Transparent Neural Networks

The feature vector generated by the series of convolutional and pooling layers is fed as an input to the KBTNN model. The KBTNN model process the granulated version of input feature vectors generated by the convolutional and pooling layers. The detailed architecture of KBTNN model is given in Fig. 5. The architecture of KBTNN is similar to the architecture of feed forward MLP neural network. The difference between KBTNN and MLP is based on the type of initial weights assigned during the training process. In MLP, the initial weights are assigned with random values. The initial weights of KBTNN are assigned with the values computed using FRSD. The architecture of KBTNN model consists of input layer, hidden layer, and output layer. Each layer of KBTNN consists of processing units called as nodes. In Fig. 5, the number of input layer nodes ($m$) is equal to the number of granulated input features applied to KBTNN. The detailed explanation of two important granulation methods is given in Section II-D1. The number of hidden layer nodes is greater than number of input layer nodes. The number of output nodes of KBTNN is equal to the number of classes in the dataset. In conventional neural network, the initial weights of the network are assigned with random values, which are updated during the training by back propagating the output error. Assigning the initial weights of the NN with the extracted

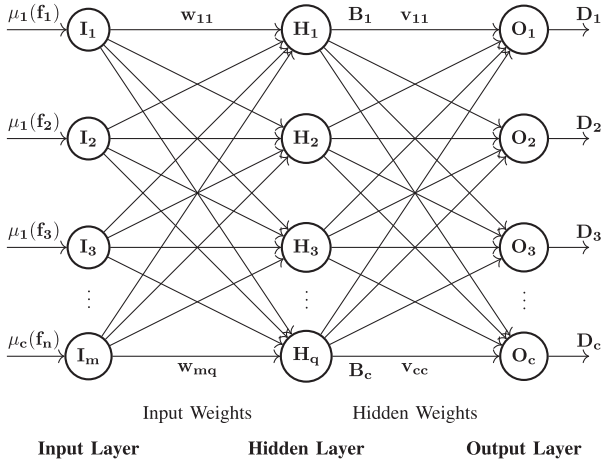


Fig. 5. Architecture of KBTNN. The architecture consists of (i) input layer, (ii) hidden layer and (iii) output layer. $I_1$,..., $I_m$ are the input layer nodes. $H_1$,...., $H_q$ are nodes of hidden layers. $O_1$, $O_2$,..., $O_c$ are the output layer nodes. $B_1$,..., $B_q$ are the outputs at hidden layer nodes and $D_1$,..., $D_c$ are the outputs at the output layer nodes.

knowledge from dataset would provide the better classification performance instead of assigning the weights of NN, randomly. Thus, the KBTNN model posses the advantages like operational transparency due to granulation of input feature vector and better learning due to KE in the form of initial weights in the architecture. The KE in KBTNN is used to learn informative features in less number of training iterations during the training process. Thus, the proposed model produced better classification results with less time complexity in comparison with conventional DNN models and MLP models. The detailed explanation of knowledge extraction and KE in KBTNN model is given in Sections II-D2 and II-D3, respectively.

*1) Type of Data Granulations:* Granulating the input feature vector provide the operational transparency in the functional mechanism of TNN unlike feed forwarding the feature vector directly to the FCNN in conventional CNN model. A granule is defined as the group of objects or a class of objects with similarity. The process of grouping the objects based on the similarity is called granulation. The concept of granulation was first coined by Zadeh [47]. In granulation, the feature values of a pixel are represented with fuzzy membership of pixel to the classes. The fuzzy membership of a pixel is the measure of possibility of input feature to the classes in the dataset. The number of granulated features of a feature vector is more in comparison with ungranulated feature vector. As each granulated feature vector represents the possibility of feature belongingness to the classes in datasets, the proposed KBMDTNN model learns from informative features in less number of iterations during the training inspite of more number of granulated features. Thus, the proposed model performs better classification in comparison with conventional DNN models and MLP models in less number of iterations and reduces the training time.

*a) Class Un-Related Granulation:* Pal *et al.* [37] suggested fuzzy granulation of input feature vector, which was named as CUR granulation. In CUR granulation, each feature of input feature vector is granulated in to three linguistic variables

called as *low*, *medium,* and *high*. The membership of a feature to the three fuzzy granules is computed based on $\pi$-type membership function. A pixel/pattern ($P$) with *n* number of features is given in the following:

$$P = \begin{bmatrix} f_1 & f_2 & . & . & . & . & f_n \end{bmatrix}. \tag{6}$$

The membership of feature value $f_n$ of a feature vector to the granules *low*, *medium*, and *high* is given as $\mu_{\text{Low}}(f_n)$, $\mu_{\text{Medium}}(f_n)$, and $\mu_{\text{High}}(f_n)$, respectively. The membership of a feature $f_i$ to a granule is given as follows:

$$\mu(f_i; c, \beta) = \begin{cases} 2(1 - \frac{\|f_i - c\|}{\beta})^2, & \text{for } \frac{\beta}{2} \le \|f_i - c\| \le \beta \\ 1 - 2(\frac{\|f_i - c\|}{\beta})^2, & \text{for } 0 \le \|f_i - c\| \le \frac{\beta}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

$\beta$ and *c* are radius and central point of granule, respectively. The membership of a feature $f_i$ is maximum value of 1 at the center of granule. The membership value gradually drops down to zero on either side of granule. The membership value is *0.5* at the crossover points. The parameters *c* and $\beta$ of three granules along the feature axis $f_i$ are computed by using maximum feature value ($f_{i_{\max}}$) and minimum feature value ($f_{i_{\min}}$), according to following equations:

$$\beta_{\text{Medium}}(f_i) = \frac{1}{2}(f_{i_{\max}} - f_{i_{\min}}) \tag{8}$$

$$c_{\text{Medium}}(f_i) = f_{i_{\min}} + \beta_{\text{Medium}}(f_i) \tag{9}$$

$$\beta_{\text{Low}}(f_i) = \frac{1}{\alpha}(c_{\text{Medium}}(f_i) - f_{i_{\min}}) \tag{10}$$

$$c_{\text{Low}}(f_i) = c_{\text{Medium}}(f_i) - \frac{1}{2}\beta_{\text{low}}(f_i) \tag{11}$$

$$\beta_{\text{High}}(f_i) = \frac{1}{\alpha}(f_{i_{\max}} - c_{\text{Medium}}(f_i)) \tag{12}$$

$$c_{\text{High}}(f_i) = c_{\text{Medium}}(f_i) + \frac{1}{2}\beta_{\text{high}}(f_i) \tag{13}$$

the extent of overlapping among the granules is controlled by the parameter $\alpha$. Thus, the pattern P with *n* number of features is granulated in to $3 \times n$ features such as

$$P_{ug} = \begin{bmatrix} \mu_{\text{Low}}(f_1), & \mu_{\text{Medium}}(f_1), & \mu_{\text{High}}(f_1) \\ \cdots, & \cdots, & \cdots, \\ \mu_{Low}(f_n), & \mu_{Medium}(f_n), & \mu_{High}(f_n) \end{bmatrix}. \tag{14}$$

*b) CR Granulation:* CUR granulation do not consider class wise membership of feature vector to the classes [42]. Motivated with this factor, Pal *et al.* [38] proposed CR granulation of feature vector. In CR granulation, each feature of a pattern/pixel is represented in terms of membership to the classes in dataset. The membership of feature $f_i$ to a class is given by

$$\mu(f_i; x, c, y) = \begin{cases} 0, & \text{for } f_i \le x \\ 2^{N-1}(\frac{f_i - x}{r - x})^N, & \text{for } x < f_i \le a \\ 1 - 2^{N-1}(\frac{c - f_i}{c - x})^N, & \text{for } a < f_i \le c \\ 1 - 2^{N-1}(\frac{f_i - c}{y - c})^N, & \text{for } c < f_i \le b \\ 2^{N-1}(\frac{y - f_i}{y - c})^N, & \text{for } b < f_i \le y \\ 0, & \text{for } f_i \ge y \end{cases} \tag{15}$$

*a* and *b* are the two crossover points with membership 0.5, and *c* is the center point of granule with membership value *1*. The

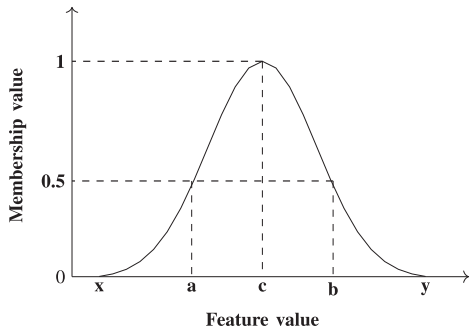


Fig. 6. $\pi$-type membership function for CR granulation.

values of *a*, *b*, and *c* are computed as *c=mean of feature values that belong to a particular class*, $a = c - (\frac{\max(F_n) - \min(F_n)}{2})$ and $b = c + (\frac{\max(F_n) - \min(F_n)}{2})$. The value of N lies between 0 to 1 and $min(F_n)$ and $max(F_n)$ are minimum and maximum values of a feature $F_n$. The points *x* and *y* are called extreme points, which are given as $x = c - (b - a)$ and $y = c + (b - a)$. The PI membership function is pictorially given in the Fig. 6. In CR granulation, a feature vector (P) with *n* number of features and the dataset with *c* number of classes is represented with $n \times c$ number of granulated features such as

$$P_g = \begin{bmatrix} \mu_1(f_1), & \mu_2(f_1), & \mu_c(f_1), \\ \mu_1(f_2), & \mu_2(f_2) & \cdots, \\ \mu_1(f_n), & \mu_2(f_n), & \mu_c(f_n) \end{bmatrix}. \tag{16}$$

$\mu_c(f_n)$ is the membership of feature $f_n$ of pixel P to the class *c*.

*2) Knowledge Extraction Using Feature Based Dependency:* The concept of knowledge extraction is based on the information extraction from the labelled dataset. Knowledge extraction is an attempt to find the reasoning between the input features and the corresponding class labels. The research on knowledge extraction from the dataset started with Yasdi [40]. In this method, a rough set based dependency (RSD) of each feature is computed and based on the RSD value, the initial weights of the NN architecture were computed. Banerjee *et al.* [41] used RSD method to extract the domain knowledge from a CUR granulated labeled dataset. Furthermore, Ganivada *et al.* [42] used fuzzy set based feature dependency for domain knowledge extraction from CR dataset. Meher [48] worked on knowledge extraction using neighborhood rough set theory. In the present study, a fuzzy rough set based dependency (FRSD) is used to quantify the relationship between the input CR granulated features and the output classes. The theoretical explanation of FRSD is provided in the literature [49]–[51]. A CR granulated feature vector $P_g$ with the membership values $\mu_1(f_1)$, $\mu_2(f_1)$, $\mu_c(f_1)$,...., and $\mu_c(f_n)$ is obtained from a feature vector (P) with features $f_1, f_2, f_3, \ldots, f_m$. The FRS based functional dependency (r) of granulated features $\mu_1(f_1)$, $\mu_2(f_1)$, $\mu_c(f_1)$,...., and $\mu_c(f_n)$ is obtained as

$$r = Y_{(\mu_1(f_1), \mu_2(f_1), \mu_c(f_1), \ldots, \mu_c(f_n))}/X. \tag{17}$$

where $Y_{(\mu_1(f_1), \mu_2(f_1), \mu_c(f_1), \ldots, \mu_c(f_n))}$ is the number of feature vectors, which can be distinguished with the n $\times$ c features i.e., $\mu_1(f_1)$, $\mu_2(f_1)$, $\mu_c(f_1)$,...., $\mu_c(f_n)$ in a given class, *X* is

TABLE I
EXAMPLE: THE DATASET WITH FOUR FEATURE VECTORS AND TWO CLASSES

| *Feature vector* | $\mu_1(f_1)$ | $\mu_2(f_1)$ | *Class label* |
|---|---|---|---|
| $P_{g1}$ | 45 | 70 | 1 |
| $P_{g2}$ | 45 | 33 | 1 |
| $P_{g3}$ | 38 | 68 | 2 |

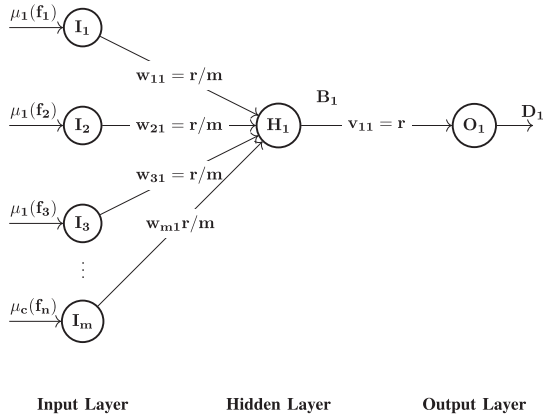


Fig. 7. KE in TNN with architecture having one output node, one hidden node and ($m$ = n × c) input nodes. The initial weights between the input layer nodes and hidden layer node $w_{11} = r/m, \cdots, w_{m1} = r/m$. The initial weights between the hidden layer node and the output layer node $v_{11} = r$.

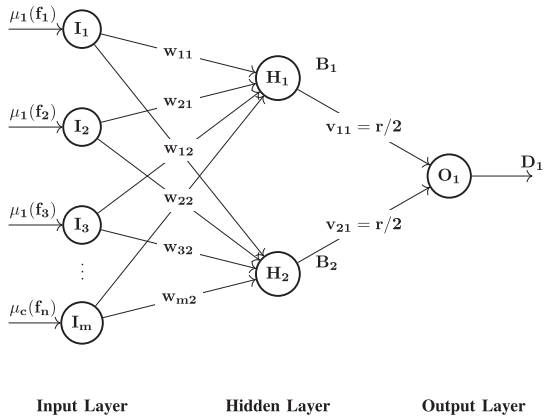


Fig. 8. KE in TNN with architecture having one output node, two hidden node and *m* input nodes. The initial weights between the input layer nodes and hidden layer nodes $w_{11} = r/2\,m$,...., $w_{m2} = r/2\,m$. The initial weights between the two hidden layer nodes and a output layer node $v_{11}$ = *r*/2 and $v_{21}$ = *r*/2, respectively.

the total number of feature vectors in the dataset. If all the ($n$ × $c$) granulated features are considered to compute FRSD, the value of $r = 1$, otherwise $r$<1. The computation of $r$ value is explained with an example as given in Table I. In Table I, the number of CR granulated feature vectors is $X = 3$ (i.e., $P_{g1}$, $P_{g2}$, and $P_{g3}$) with features $\mu_1(f_1)$, $\mu_2(f_1)$, and the number of classes is 2. The FRSD is computed for the features $\mu_1(f_1)$, $\mu_2(f_1)$, and ($\mu_1(f_1)$, $\mu_2(f_1)$). The $r$ value for the feature $\mu_1(f_1)$ is r = $Y_{\mu_1(f_1)}$/X = 1/3. In this case, the value of $Y_{\mu_1(f_1)} = 1$ because with the granulated feature $\mu_1(f_1)$, the vector $P_{g3}$ can only be identified. The feature vectors $P_{1g}$ and $P_{2g}$ cannot be uniquely identified with the feature $\mu_1(f_1)$. The FRSD value of feature $\mu_2(f_1)$ is $r = Y_{\mu_2(f_1)}$/ X = 3/3, the value of $Y_{\mu_2(f_1)}$ = 3 because with the feature $\mu_2(f_1)$, the feature vectors $P_{g1}$, $P_{g2}$, and $P_{g3}$ can be uniquely identified. Similarly, the FRSD of both the features in combine $\mu_1(f_1)$ and $\mu_2(f_1)$ is obtained as r = $Y_{(\mu_1(f_1),\mu_2(f_1))}$/X = 3/3. The value of $Y_{(\mu_1(f_1),\mu_2(f_1))} = 3$, because with $\mu_1(f_1)$, $\mu_2(f_1)$) the feature vectors $P_{g1}$, $P_{g2}$, and $P_{g3}$ can be identified. In the present study, all the CR granulated features were considered to represent the information content of images in the dataset with $r = 1$. Based on the FRSD, the initial weights of KBTNN model is assigned during the training stage.

*3) KE in TNN:* All the granulated feature vectors are fed as input to the TNN model. The FRSD value of dataset with all the granulated features is a maximum value 1 ($r = 1$). The initial weights of the TNN model are assigned such that the functional dependency at each output nodes ($O_1$, $O_2$,..., $O_c$) is 1. As an example, the initial weights of TNN model with one output node, one hidden node, and *m* input nodes are assigned based on the *r* value as given in Fig. 7. In Fig. 7, the weights between the hidden layer node $H_1$ and the output layer node $O_1$ is the maximum value $r = 1$. The weights between the input layer nodes and the hidden layer node are assigned with each weight value $w_{11} = r/m$,...., $w_{m1} = r/m$, where *m* is the number of CU/CUR granulated features. In the case, when the number of hidden layer and output layer nodes are more, the weights are uniformly distributed in the architecture of TNN. The initial weights of TNN model with *m* input nodes, two hidden nodes and one output node is given in Fig. 8. The weights between the hidden layer nodes and the output layer node are assigned with the value *r*/2. The weight values $v_{11}$ and $v_{21}$ = *r*/2 representing the two hidden nodes. The weight values $w_{12} = r/2\,m$,....., $w_{m2} = r/2\,m$ represent the uniform distribution of *r* value among the input layer nodes and the hidden layer nodes.

*4) Learning Process of KBTNN Model:* The learning mechanism of TNN is similar to the feedforward neural network [3]. The TNN model is trained using back propogation learning algorithm. During the training, morphological operators, series of convolutional, and pooling layers are applied on the the input image to obtain the minimal features from the image. These informative features are granulated to obtain the membership value of each feature to the class (explained in Section II-D1). The membership values of a features are provided as inputs to TNN. In the feedforward process, the membership value of each feature is multiplied with corresponding initial weight in the connecting link (the initial weights are computed using FRSD as given in Section II-D2) and the resultant value is feed forwarded through the architecture of TNN. The output at hidden node is the sum of product of the input feature values and the initial weights. Similarly, the output at the output layer node is the sum of product of output at hidden layer and the corresponding hidden weights. Later, the error at each node of output layer is computed. If the output error is equal to zero then the weights of TNN are not updated. In the case, the output error is a nonzero value, the weights between the layers of TNN are updated by back propagating the output error. Updation of weights is implemented by passing the input labeled image through the layers of KBMDTNN. The weights of the TNN

model are updated to reduce the overall cost function (CF). In the present study, the mean square CF is used to compute the overall error. The selection of mean square CF during the training is due to the usage of sigmoid activation function at each node of TNN in feed forward process. The CF of TNN for individual sample image ($x$) is given in the following:

$$C_x = \frac{1}{2}\sum_{k=1}^{c}(Z_k - D_k)^2 \tag{18}$$

where $c$ represents the number of classes in the dataset, $Z_k$ is desired value at the output node $O_k$. The overall CF of TNN for the entire dataset is given in the following:

$$\text{CF} = \frac{1}{P}\sum_{i=1}^{P} C_i \tag{19}$$

where P is the number of sample images in the dataset. Similarly, there exists various type of cost functions such as multiclass cross entropy loss, sparse multiclass cross entropy loss, and Kullback Leibler divergence loss. The mechanism of back propagating the output error and updation of weight parameters in TNN is similar to the conventional neural networks.

The initial weights of TNN in proposed model are computed from the dataset using FRDS method. During the training of KBMDTNN model, the images from the dataset are passed one by one and the initial weights are updated based on the output error. Initial weights are computed independently by considering the dataset and the weights of TNN in proposed model are updated during the training process. The detailed explanation about the computation of initial weights and training TNN model is given in the literature [37], [41], and [42].

## III. Results and Discussion

### A. Datasets Used

In the present study, the performance of CNN models was tested with six remote sensing image datasets. The images in the datasets have unique spatial and spectral resolutions. The collection of six datasets is made such that four datasets consists of multispectral remote sensing images and the remaining two datasets consists of hyperspectral remote sensing images. The panaromic view of remote sensing images used in the present study is given in Fig. 9(a)–(f). The detailed description of these datasets is given in the following sections.

*1) Hyperspectral Remote Sensing Images:* The hyperspectral remote sensing datasets like ROSIS and AIVIRS were considered in the present study. The detailed description of the datasets is given in following sections.

*a) ROSIS:* The hyperspectral ROSIS image covers the parts of Pavia University, Northern Italy. ROSIS sensor consists of spectral information in 103 bands with 610 × 610 pixels. The image has a spatial resolution of 1.3 m with nine land cover classes. The number of pixels per class in ROSIS dataset is given Table II.

*b) AIVIRS:* The AIVIRS dataset is also called as Indiana's Indian Pines dataset. The dataset consists of information related to 145 × 145 pixels in 200 spectral bands. The detailed description of the classes and the number of pixels in each class of AIVIRS dataset is given in Table II.

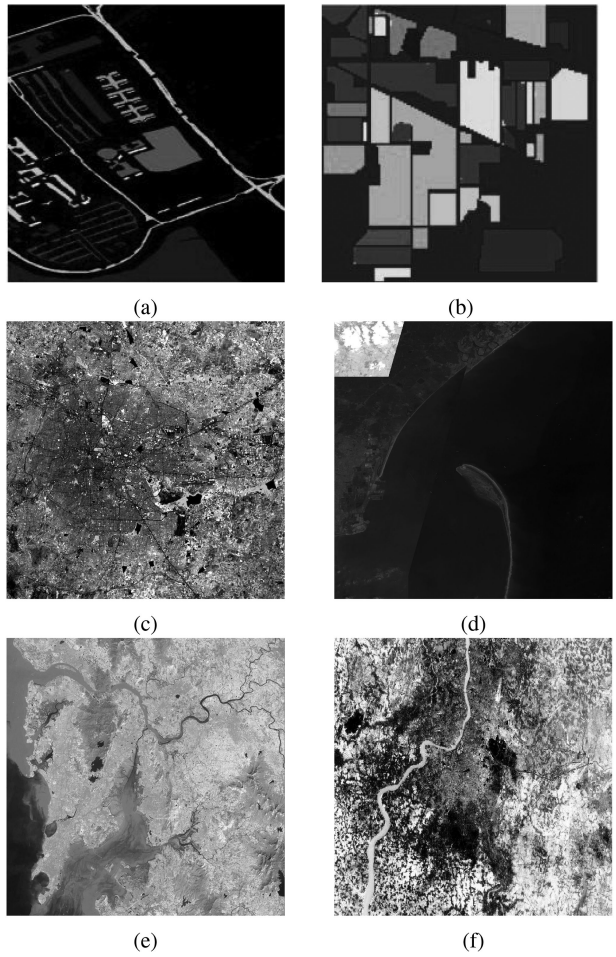


Fig. 9. (a) ROSIS image of Pavia University, (b) AVIRIS image of Indiana Pines, (c) Sentinel MSI image of Bangalore City, (d) IRS LISS IV image of Kakinada City, (e) IRS LISS III image of Mumbai City and (f) IRS LISS II image of Kolkatta city.

TABLE II
Hyperspectral Datasets and Number of Samples

| *ROSIS* | | *AVIRIS* | |
|---|---|---|---|
| Pavia University | Samples | Indian Pines | Samples |
| *Asphalt* | *6631* | *Corn-notill* | 1428 |
| *Meadows* | *18649* | *Corn-mintll* | 830 |
| *Gravel* | *2099* | *Grass-pature* | 483 |
| *Trees* | *3064* | *Grass-trees* | 730 |
| *Painted metal sheets* | *1345* | *Hay-Windrowed* | 478 |
| *Bar soil* | *5029* | *Soybean-notill* | 972 |
| *Bitumen* | *1330* | *Soybean-mintill* | 2455 |
| *Self blocking bricks* | *3682* | *Soybean-Clean* | 593 |
| *Shadows* | *947* | *Woods* | 1265 |
| | | *Building-grass-trees-drives* | 386 |

*2) Multispectral Remote Sensing Images:* In support to the hyperspectral remote sensing images, multispectral remote sensing images acquired from four sensors like 1) sentinel multispectral instrument (MSI), 2) indian remote sensing linear imaging

self scanning sensor (IRS LISS IV), 3) IRS LISS III and 4) IRS LISS II were considered to evaluate the performance of KBMDTNN model. The four sensors are unique interms of the characteristics like spatial and spectral resolutions. A total of five classes were considered in the study area. The classes are named as (i) Urban Dense (ii) Urban Sparse (iii) Water (iv) Agriculture and (v) Forest.

*a) Sentinel MSI–Bangalore City:* The sentinel MSI sensor has the spectral information of the objects in 13 spectral bands. Out of 13 spectral bands, four bands has the spatial resolution of 10 m, six bands has the spatial resolution of 20 m and three bands has the spatial resolution of 60 m. In the present study, MSI image of Bangalore city, India is considered for evaluation.

*b) IRS LISS IV–Kakinada City:* IRS LISS IV is a mutlispectral sensor with operational spectral bands over the range (0.52–0.86 $\mu$m). The spatial resolution of the sensor is 5.8 m. The IRS LISS-IV image of Kakinada city, India is considered in the present study to test the performance of CNN models.

*c) IRS LISS III–Mumbai City:* IRS LISS III image of Mumbai city, India is used to test the performance of CNN models. LISS III image has the spectral information of the classes in four operating wavelengths ranging from 0.52 to 1.7 $\mu$m. The spatial resolution of IRS LISS III image is 23.5 m.

*d) IRS LISS II–Kolkatta City:* IRS LISS II image of Kolkatta region, India is considered to test the performance of CNN models. The spatial resolution of IRS LISS II is 36.25 m and the sensor operates in four spectral bands ranging 0.46–0.86 $\mu$m. In the present study, various type of CNN models were considered for performance analysis. The performance of these CNN models was evaluated using the metrics like OA, UA, PA, KC, DS, $OA_{\text{STD}}$, and NOI. The detailed description of the CNN models is given in following sections.

### B. Model Description

In the present study, seven deep learning models were considered to compare the performance of proposed KBMDTNN model. The models were tested with six remote sensing image datasets. The selection of models was based on the type of morphological operator used to extract the lines, edges, and boundaries in image, the type of granulation method used for feature vectors, method of computing the initial weights of the NN, and the type of NN model for classifying the image.

1) Model : Morphological operator + Type of Granulation + Initial weights + Type of Neural network
2) Model 1 $(M_1)$ : Counter-Harmonic Mean + Ungranulated + Randomly + Conventional NN [43]
3) Model 2 $(M_2)$ : Closing + CUR + FRSD + TNN
4) Model 3 $(M_3)$ : Opening + CUR + FRSD + TNN
5) Model 4 $(M_4)$ : Opening + CR + FRSD + TNN
6) Model 5 $(M_5)$ : Cascaded operator + CR + FRSD + TNN
7) Model 6 $(M_6)$ : Cascaded operator + CR + FRSD + TNN (Proposed model)
8) Model 7 $(M_7)$ : Cascaded operator + Ungranulated + Randomly + Conventional NN

Model 1 $(M_1)$ is a CNN model with CHM morphological operator at the morphological layer to extract the informative features. The feature vectors generated by the convolutional and pooling layers of $M_1$ are directly fed in to the output layer without granulation. The output layer of model 1 consists of FCNN which is trained with random initial weights. Model 2 $(M_2)$ is CNN model with closing operator on the input image. The input feature vector of $M_2$ is granulated using CUR method and the granulated feature vector is processed using TNN. Model 3 $(M_3)$ is similar to $M_2$ with the only difference in applying the Opening operator on the input image. Model 4 $(M_4)$ is a CNN model similar to $M_3$ and with CR granulated feature vector. Model 5 $(M_5)$ is CNN model with the cascaded operator (opening followed by closing) applied on the input image. The initial weights of model $M_2$, $M_3$, and $M_4$ are assigned with the values computed using FRSD. The initial weights of model 5 are computed using using RSD method. Model 6 $(M_6)$ is similar to $(M_5)$ and the required knowledge for the model 6 is computed using FRSD method. The output layer of the models $M_2$, $M_3$, $M_4$, $M_5$, and $M_6$ consists of TNN to process the granulated feature vectors. Model 7 $(M_7)$ is CNN model with cascaded operator in morphological layer and the features generated by the series of convolution and pooling layers are fed to conventional NN in the output layer.

A five facet approach is chosen in the present study to justify the superiority of proposed model $M_6$ over the models $M_1$, $M_2$, $M_3$, $M_4$, and $M_5$. In the first facet, comparison between $M_1$ (base model) and $M_2$ is performed to show the improvement in the performance of the CNN model due to 1) CUR granulation of feature vector and 2) KE in the TNN. In the second facet of comparison, $M_2$, $M_3$, and $M_4$ are compared to demonstrate the superiority of $M_4$ over $M_2$, $M_3$ is due to efficient feature extraction using opening operator in morphology layer and CR granulation of input data. In the third facet, $M_4$ and $M_5$ are considered to show the superiority of $M_5$ over $M_4$ due to the use of cascaded operator (opening followed by closing operator). In the fourth facet, the performance of $M_5$ and $M_6$ are compared to know the superiority of proposed model over $M_5$ due to FRSD based KE in TNN. In the fifth facet of comparison, $M_6$ is compared with $M_7$ to justify the superiority of KBMDTNN model over $M_7$ because of transparency obtained due to FRSD based KE.

In the present study, the size of input image for all models is considered as 500 $\times$ 500 $\times$ B, where B is the number of spectral bands in the remote sensing image. The architecture of proposed model comprise of thirteen layers with one input image, cascaded operator, six convolution layers (Conv), four pooling layers (Pool), and a transparent neural network. The detailed architecture of KBMDTNN model is given as : Input image–Cascaded operator–Conv1–Conv2–Pool1–Conv3–Conv4–Pool2–Conv5–Conv6–Pool 3–Pool 4–KBTNN. The KBTNN model in the output layer consists of three layer transparent neural network. The architecture of TNN is given as (28 $\times$ 28) 784 : 100 : 5, where 784 is the number of pixels received from series of convolution and pooling layers, the number of hidden layer nodes is taken as 100 and the number

TABLE III
EXPERIMENTAL RESULTS OF MODELS FOR *ROSIS* DATASET

| | Training | | | | Performance Metrics | | |
|---|---|---|---|---|---|---|---|
| *Model* | *10%* | *30%* | *50%* | *70%* | *OA%* | *KC* | *NOI* |
| $M_1$ | 90.45 | 91.78 | 92.01 | 93.20 | 91.86 | 0.842 | 50 |
| $M_2$ | 90.58 | 91.98 | 92.78 | 94.62 | 92.48 | 0.867 | 50 |
| $M_3$ | 91.83 | 93.02 | 94.46 | 95.81 | 93.78 | 0.881 | 50 |
| $M_4$ | 92.28 | 93.87 | 95.68 | 96.41 | 94.56 | 0.910 | 50 |
| $M_5$ | 93.12 | 94.76 | 96.88 | 97.96 | 95.68 | 0.927 | 50 |
| $M_6$ | **94.22** | **96.45** | **97.96** | **98.73** | **96.84** | **0.932** | **50** |
| $M_7$ | 90.50 | 91.86 | 92.36 | 94.12 | 92.21 | 0.856 | 50 |

of output nodes is equal to number of classes (no. of classes = 5 for LISS III dataset).

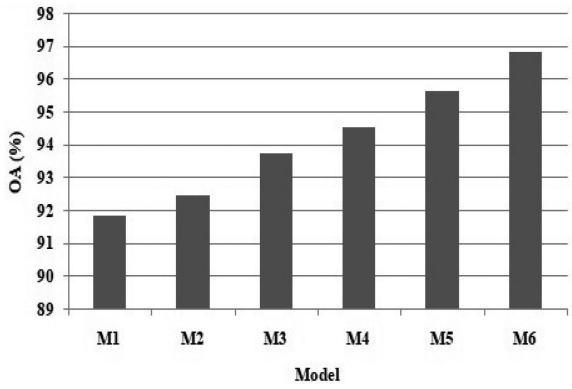


Fig. 10. Performance of $M_1$ to $M_6$ in terms of OA with ROSIS DATASET.

### C. Criteria for Training and Testing

The remote sensing image pixels are labeled based on the ground truth and furthermore, the labeled pixels are used to prepare the dataset. The entire dataset is divided into two disjoint sets with the first set of pixels named as training set and the second part of the dataset is named as testing set. The parameters of the models are computed from the training dataset. The division of dataset is based on the selection of 10%, 30%, 50%, and 70% pixels in to training sets and the remaining 90%, 70%, 50%, and 30% pixels of the dataset in to test sets. The selection of sub images in to training and testing sets is done randomly. The results of models are obtained through tenfold cross validation and the average accuracy of all the folds is taken as OA.

### D. Performance Measurement Index

After implementing the training and testing phases, the performance of models was evaluated using the performance metrics. Six performance indices like OA, PA, UA, DS, KC, NOI, and $OA_{\text{STD}}$ were used in the present study. The indices are computed using confusion matrix (CM). CM is a table of actual classified output classes assigned by a model. The sum of diagonal elements of the CM divide by total number of pixels is termed as OA. The limitation of OA lies in considering the overall agreement between actual and predicted class labels because of which the decision making capability of the model will be reduced. As a solution to this problem, other metrics like DS, UA, PA, KC, NOI, and $OA_{\text{STD}}$ were considered to obtain the detailed performance of the models. The probability of a reference pixel being correctly classified is called as UA. UA is defined as correctly classified pixels of a class to the total number of pixels which are classified to the class. The performance of PA is better than UA. DS is the measure of distribution of classified pixels among the classes in a dataset. DS is used to understand the characteristics like intersection of various class boundaries. Kappa coefficient is used to measure the performance of a model depending on the individual class-based agreement.

### E. Performance of Models With ROSIS Dataset

The performance of CNN models with ROSIS dataset is given in Table III. In Table III, $M_6$ performed better than the remaining CNN models with the OA of 96.84%. The superiority of proposed KBMDTNN model over other models is due to the use of cascaded morphological operator and also due to FRSD based KE in the TNN. In the first facet of comparison, the OA of $M_2$ is 0.62% better the base model $M_1$. The superiority of $M_2$ is due to the use of closing morphological operator, which considers the finite information in the image and FRSD based KE in the TNN. In the second facet of comparison, $M_4$ outperformed $M_2$ and $M_3$ with 2.08% and 0.78%, respectively. The superiority of $M_4$ over $M_2$ and $M_3$ is due to the CR granulation of feature vectors and application of morphological opening operator. In a similar comparison, $M_3$ better than $M_2$ because in $M_3$ closing operator preserves the semiinformation of objects in the image while the opening operator in $M_2$ eliminates the semiinformation features and preserves the major information in the region of interest. In the third facet of comparison, $M_5$ is better than $M_4$ with 1.12% OA. The superiority of $M_5$ is due to the use of cascaded operator that extracts efficient features from the image. In the fourth facet of comparison, $M_6$ performed better than $M_5$ with 1.24% improvement in the OA. The superiority of $M_6$ is due to the FRSD based KE in KBMDTNN model. The proposed KBMDTNN model is 4.97% better than the base model $M_1$ [43]. The NOIs were taken as 50 for the CNN models in the present study. The superiority of KBMDTNN over other models is justified with highest KC value of 0.932. The pictorial representation of the performance for $M_1$ to $M_6$ is given in Fig. 10. The classified images of ROSIS sensor for KBMDTNN model ($M_6$) (best classifier) and ($M_1$) (worst classifier) are given in Fig. 12(a) and (b), respectively. The class labels are given in Fig. 12(b). In the fifth facet of comparison, $M6$ is 4.25% better than $M7$ in terns of OA. The superiority of $M_6$ over $M_7$ is due to KE in TNN and CR data granulation. The superiority of $M_6$ over $M_7$ is justified with KC as $M_6$ has KC value of 0.932 and $M_7$ has KC value of 0.856.

In support to the performance metrics like $OA$ and KC, the metrics such as UA, PA, and DS were considered in the present study. The characteristics of UA, PA, and DS is that these metrics provide class based analysis of the labeled images. The performance of $M_1$, $M_3$, and $M_6$ were analyzed with the metrics UA, PA, and DS. The results of $M_1$, $M_3$, and $M_6$ in terms of three metrics UA, PA, and DS are given in Table IV. The performances of the models with UA, PA, and DS are given

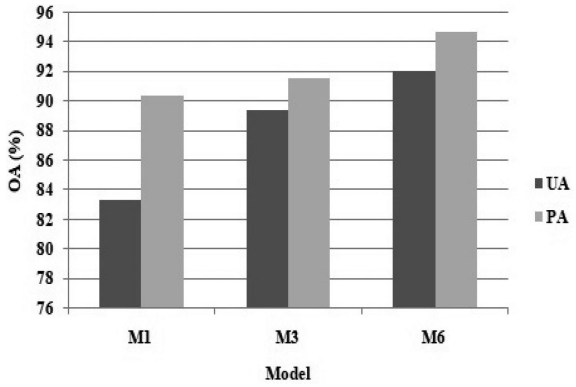


Fig. 11. Performance of $M_1$, $M_3$, and $M_6$ in terms of UA (%) with ROSIS dataset.

TABLE IV
NINE CLASSES OF ROSIS DATASET WITH UA, PA AND DS FOR $M_1$, $M_3$, AND $M_6$ FOR 50% TRAINING DATA

| | Model 1 | | | Model 3 | | | Model 6 | | |
|---|---|---|---|---|---|---|---|---|---|
| *Class* | *UA* (%) | *PA* (%) | *DS* | *UA* (%) | *PA* (%) | *DS* | *UA* (%) | *PA* (%) | *DS* |
| 1 | 83.36 | 90.44 | 0.593 | 89.45 | 91.59 | 0.372 | 92.12 | 94.77 | 0.324 |
| 2 | 81.22 | 88.56 | 0.522 | 86.34 | 91.44 | 0.384 | 90.33 | 95.34 | 0.294 |
| 3 | 83.65 | 92.73 | 0.477 | 87.77 | 92.55 | 0.308 | 91.22 | 93.68 | 0.255 |
| 4 | 71.76 | 79.84 | 0.468 | 73.39 | 80.34 | 0.266 | 80.24 | 80.45 | 0.202 |
| 5 | 72.68 | 81.83 | 0.488 | 78.83 | 84.46 | 0.284 | 81.26 | 85.54 | 0.199 |
| 6 | 73.74 | 79.34 | 0.444 | 80.88 | 83.33 | 0.277 | 83.36 | 86.64 | 0.184 |
| 7 | 74.86 | 75.77 | 0.789 | 73.68 | 85.68 | 0.484 | 81.24 | 87.36 | 0.332 |
| 8 | 76.95 | 80.54 | 0.565 | 83.38 | 83.37 | 0.334 | 90.46 | 86.78 | 0.225 |
| 9 | 77.88 | 72.06 | 0.314 | 82.24 | 80.98 | 0.226 | 84.58 | 81.47 | 0.164 |

in Fig. 11. In Table IV, $M_6$ produced better UA and PA values as compared with $M_1$ and $M_4$ in all the classes. The UA of $M_6$ for class 1 is 92.12%, UA of $M_3$ for class 1 is 89.45% and UA value for model 1 is 83.36%. This indicate a significant improvement in UA from $M_1$ to $M_6$. In a similar analysis, an improvement in PA is noticed from $M_1$ to $M_6$. $M_6$ outperformed $M_1$ and $M_3$ in terms of DS. The superiority of $M_6$ over $M_3$ and $M_1$ is justified with less value of DS for $M_6$.

*1) Comparison of KBMDTNN Model With Other Basic Classifiers for ROSIS Dataset:* The performance of proposed KBMDTNN model is compared with some of the basic CNN models like large patch CNN (LPCNN), multiscale deep CNN (MSDCNN), CNN with overfeat pretrained model (CNNOPM), CNN with extreme learning machine (CNNELM), deep random-scale stretched CNN (SRSCNN), and deep-local-global feature fusion framework (DLGFF). The selected models have deep layers of convolution and pooling operators. The performance of proposed model in comparison with similar type of CNN models with the metrics like OA, KC and $OA_{\text{STD}}$ is given in Table V. The proposed KBMDTNN model is 4.46%, 3.78%, 2.49%, 1.96%, 1.38%, and 0.9% better than the models LPCNN, MSDCNN, CNNOPM, CNNELM, and DLGFF, respectively. The superiority of the proposed model over similar type of basic CNN classifiers is due to the usage of cascaded operators, CR granulation of feature vector and knowledge encoded TNN in the output layer. The graphical representation of the performance of KBMDTNN model and other CNN models is given in Fig. 13.

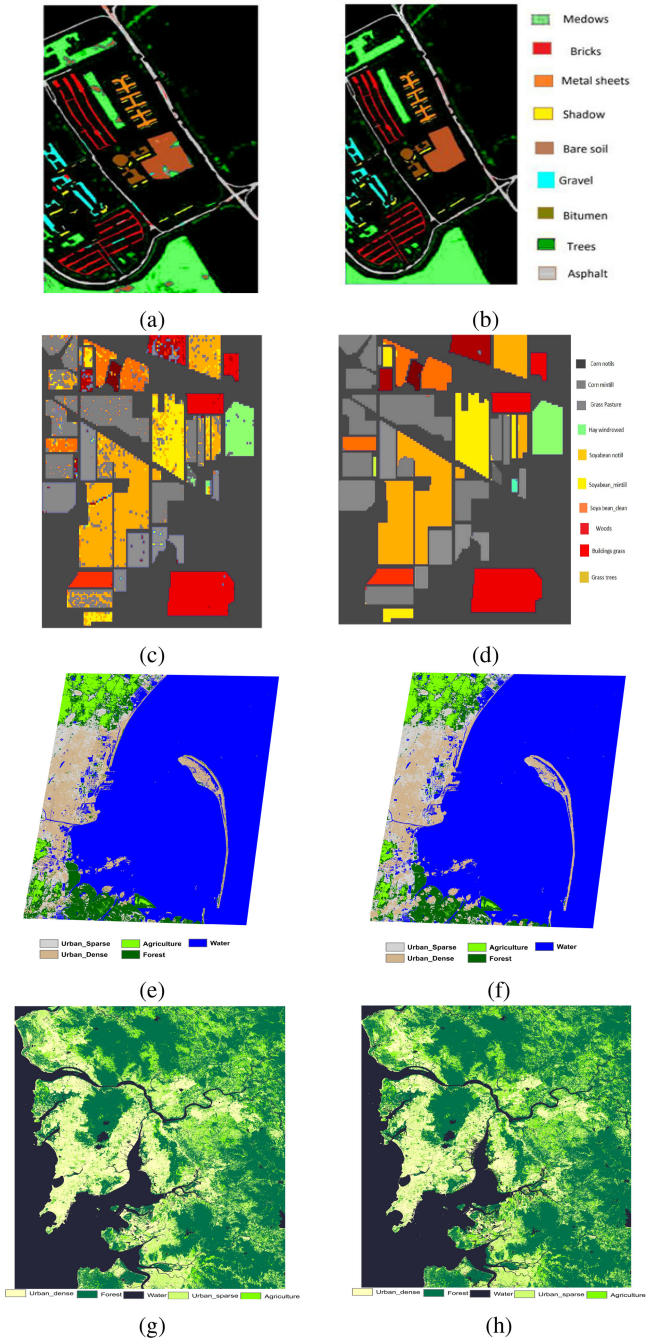


Fig. 12. Output classified images of (a) ROSIS sensor with $M_1$ (worst classifier), (b) ROSIS sensor with $M_6$ (best classifier), (c) AVIRIS sensor with $M_1$, (d) AVIRIS sensor $M_6$, (e) LISS IV sensor with $M_1$, (f) LISS IV sensor with $M_6$, (g) LISS III sensor with $M_1$ and (h) LISS III sensor with $M_6$.

The performance of proposed KBMDTNN model is compared with the recently proposed DNN models such as Resnet, VGGnet, alexnet, mobilenet, and darknet. The performances of these DNN models are given in Table VI. In Table VI, the performance of proposed model is 0.72 %, 0.60%, 1.16%, 2.52%, and 0.8% better than ResNet, VGGNet, alexnet, mobilenet, and darknet models, respectively. The superiority of proposed model

TABLE V
PERFORMANCE COMPARISON OF KBMDTNN MODEL WITH OTHER CNN CLASSIFIERS FOR ROSIS DATASET WITH THE METRICS LIKE $OA$, KC, AND $OA_{\text{STD}}$

| *Model* | $OA(\%)$ | *KC* | $OA_{STD}$ | NOI |
|---|---|---|---|---|
| *LPCNN [14]* | 92.38 | 0.836 | 5.89 | 50 |
| *MSDCNN [52]* | 93.06 | 0.843 | 4.48 | 50 |
| *CNNOPM [53]* | 94.35 | 0.856 | 3.436 | 50 |
| *CNNELM [18]* | 94.88 | 0.868 | 2.789 | 50 |
| *SRSCNN [54]* | 95.46 | 0.875 | 1.657 | 50 |
| *DLGFF [55]* | 95.94 | 0.884 | 1.045 | 50 |
| $M_6$ **(KBMDTNN)** | 96.84 | 0.968 | 0.786 | 50 |

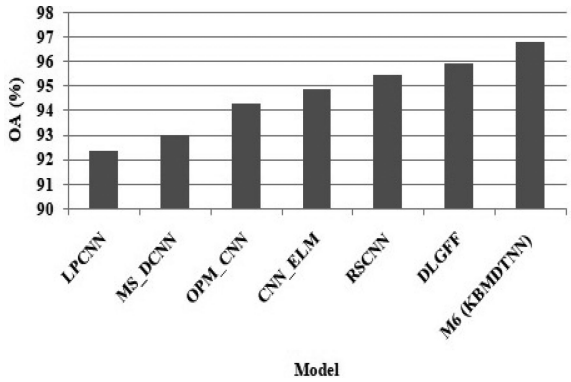


Fig. 13. Performance of KBMDTNN ($M_6$) and other classifiers in terms of OA (%) with ROSIS dataset.

TABLE VI
PERFORMANCE COMPARISON OF KBMDTNN MODEL WITH MOST RECENTLY USED CNN CLASSIFIERS AND MLP FOR ROSIS DATASET

| *Model* | $OA(\%)$ | *KC* | $OA_{STD}$ | NOI |
|---|---|---|---|---|
| *ResNet* | 96.12 | 0.902 | 0.942 | 50 |
| *VGGNet* | 96.24 | 0.936 | 0.954 | 50 |
| *AlexNet* | 95.68 | 0.880 | 1.424 | 50 |
| *MobileNet* | 94.32 | 0.850 | 3.781 | 50 |
| *DarkNet* | 96.04 | 0.924 | 0.954 | 50 |
| *MLP* | 86.42 | 0.732 | 7.34 | 700 |
| $M_6$ **(KBMDTNN)** | 96.84 | 0.968 | 0.786 | 50 |

over these models is due to the KE and data granulation during the training process. The superiority of proposed model over the recent CNN models is justified in terms of other metrics also. In Table VI, the NOI of proposed model is 50 and NOI of MLP is 700. Model 6 performed better than MLP with additional advantage of less number of iterations. KE and data granulation in KBMDTNN model provides the understandable learning process using possibility and convergence of lowest gradient descent value in less number of iterations. Due to this reason, the proposed model produces better results than MLP with less time complexity.

The performance of proposed model is compared with other basic TNN models such as decision trees (DT), random forest (RF), fuzzy multilayer perceptron (FMLP) [37], rough fuzzy multilayer perceptron (RFMLP) [41], fuzzy rough GNN (FRGNN) [42]. These models were considered in the present study because of operational transparency due to KE (assigning initial weights with computed values). The results of TNNs

TABLE VII
PERFORMANCE COMPARISON OF KBMDTNN MODEL WITH OTHER TNN CLASSIFIERS FOR ROSIS DATASET

| *Model* | *NOI* | $OA(\%)$ | NOR | KE |
|---|---|---|---|---|
| *DT* | - | 84.78 | 35 | - |
| *RF* | - | 85.68 | 55 | - |
| *FMLP* | 200 | 87.26 | FC | FSD |
| *RFMLP* | 200 | 88.46 | 25 | RSD |
| *FRGNN* | 200 | 89.68 | FC | FRSD |
| $M_6$ **(KBMDTNN)** | 50 | 96.84 | FC | FRSD |

TABLE VIII
EXPERIMENTAL RESULTS OF ALL THE MODELS FOR *AVIRIS* DATA WITH 60% TRAINING DATA

| *Model* | *OA%* | KC | $OA_{STD}$ |
|---|---|---|---|
| $M_1$ | 90.24 | 0.812 | 6.42 |
| $M_2$ | 91.36 | 0.835 | 5.84 |
| $M_3$ | 92.48 | 0.857 | 4.86 |
| $M_4$ | 93.78 | 0.874 | 3.67 |
| $M_5$ | 94.86 | 0.895 | 2.78 |
| $M_6$ | 96.69 | 0.912 | 1.89 |

models are given in Table VII. In Table VII, the performances of models were evaluated based on the number of rules (NOR), NOI, OA, and KE procedure. DT with 35 rules produced 84.78% OA. RF with 55 rules produced 85.68% OA, FMLP with FC architecture and FRSD based KE produced 87.26% OA. RFMLP with 25 rules and RSD based KE produced 88.46% OA. FRGNN model with FC architecture and FRSD based KE produced 89.68% OA. The proposed model produced 96.84% OA, which is 12.06% better than the DT (lowest OA of 84.78%). Similarly, the proposed model outperformed other TNN models in OA. The superiority of proposed model over other TNN models is due to deep transparent architecture and FRSD based KE.

### *F. Performance of Models With AVIRIS Dataset*

The superiority of $M_6$ over similar type of CNN models in terms of OA, KC and $OA_{\text{STD}}$ is justified with AVIRIS dataset. The results of six models for AVIRIS dataset is given in Table VIII. In Table VIII, $M_6$ outperformed other models with OA of 96.69%, KC value of 0.912, and $OA_{\text{STD}}$ of 1.89. The performance of $M_1$ is lowest with OA of 90.24%, KC value of 0.812, and $OA_{\text{STD}}$ of 6.42. The superiority of M6 over other models is due to 1) The cascaded morphological operator that preserves the shape/size features of the objects in the image, 2) Fuzzy rough set based KE in TNN at the output layer that provides the best possible solution during the gradient descent. The performance of $M_1$–$M_6$ for AVIRIS dataset is pictorially given in Fig. 14. The classified images of AVIRIS sensor for KBMDTNN model ($M_6$) (best classifier) and ($M_1$) (worst classifier) are given in Fig. 12(c) and (d), respectively. The 10 class labels of AVIRIS images is given Fig. 12(c) and (d).

### *G. Performance of Models With Sentinel MSI, LISS IV, LISS III, and LISS II Datasets*

The performance of six models were tested with the datasets such as Sentinel MSI, LISS IV, LISS III, and LISS II. The results

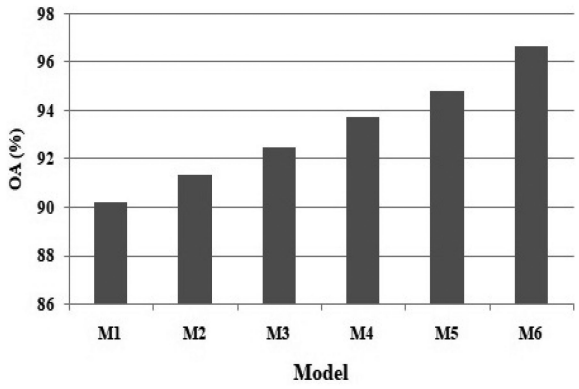


Fig. 14. Performance of $M_1$ to $M_6$ for AVIRIS dataset.

TABLE IX
PERFORMANCE OF $M_6$ (BEST CLASSIFIER) AND $M_1$ (WORST CLASSIFIER) IN TERMS OF OA, KC AND $OA_{\text{STD}}$ FOR SENTINEL MSI, LISS IV, LISS III, AND LISS II DATASETS

| *Dataset* | *Model* | *OA%* | KC | $OA_{STD}$ |
|---|---|---|---|---|
| Sentinel MSI | $M_1$ | 82.45 | 0.823 | 5.789 |
| | $M_6$ | 91.86 | 0.914 | 0.986 |
| LISS IV | $M_1$ | 88.46 | 0.886 | 6.567 |
| | $M_6$ | 96.84 | 0.924 | 0.784 |
| LISS III | $M_1$ | 89.58 | 0.814 | 7.214 |
| | $M_6$ | 95.68 | 0.894 | 0.845 |
| LISS II | $M_1$ | 87.96 | 0.874 | 6.724 |
| | $M_6$ | 96.74 | 0.945 | 0.543 |

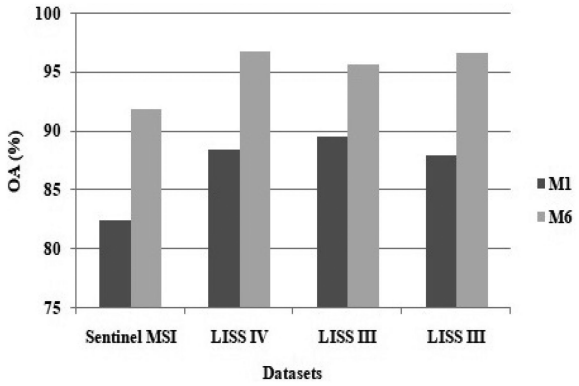


Fig. 15. Performance of $M_1$ and $M_6$ interms of OA (%) for MSI, LISS IV, LISS III, and LISS II datasets.

of $M_6$ (Best classifier) and $M_1$ (Worst classifier) for the four datasets is given in Table IX. In the case of MSI dataset, $M_6$ is having the best performance with OA of 91.86%, KC value of 0.914, and $OA_{\text{STD}}$ of 0.986. $M_1$ is having the least performance with OA of 82.45%, KC value of 0.823, and $OA_{\text{STD}}$ of 5.789. The performance of $M_6$ and $M_1$ is similar in the case of other datasets like LISS IV, LISS III, and LISS II. The performance of models $M_1$ and $M_6$ for MSI, LISS IV, LISS III, and LISS II is pictorially represented in Fig. 15. The classified images of LISS IV and LISS III sensors for KBMDTNN model ($M_6$) (best classifier) and CHM based MCNN ($M_1$) (worst classifier) are given in Fig. 12(e)–(h), respectively. The five land use/land cover classes of IRS LISS III and LISS IV images such as urban dense, urban sparse, water, forest and agriculture are given in Fig. 12(h), Fig. 12(g) and Fig. 12(e), Fig. 12(f), respectively.

## IV. CONCLUSION

A KBMDTNN model is suggested in the present study for remote sensing image classification. The model uses cascaded morphological operators to extract the information related to lines, edges, and boundaries of objects from the input image. The model uses optimum number of convolutional and pooling layers in the remote sensing image classification. The output layer of KBMDTNN model consists of TNN with FRSD based initial weights. The KBMDTNN model provides the operational transparency in all the layers starting from morphological operators to the output layer. Thus, the model overcomes the black box nature of conventional CNN. The superiority of KBMDTNN model over similar type of deep learning models is evaluated with multispectral and hyperspectral remote sensing image classification. The model produced an highest OA of 96.84% with 4.97% better than the MCNN as tested with ROSIS dataset. The proposed model produced similar type of classification results with other datasets. Furthermore, the proposed model can be used for multisensor multiresolution remote sensing image classification.

## ACKNOWLEDGMENT

The author would like to thank Prof. Landgrebe and Prof. P. Gamba for providing the AVIRIS Indiana's Indian Pines and ROSIS hyperspectral remote sensing data, respectively. The author would also like to thank K. Madan Mohan Reddy, Vice-Chairman, Kandula Sreenivasa Reddy Memorial College of Engineering and A. Mohan, Former Director, Kandula Group of Institutions for establishing Machine Learning group at Kandula Sreenivasa Reddy Memorial College of Engineering, Kadapa, Andhra Pradesh, India - 516003.

## REFERENCES

[1] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis: An Introduction*, 4th ed. Berlin, Germany: Springer, 2006.
[2] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ, USA: Wiley, 2000.
[3] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. London, U.K.: Pearson, 2009.
[4] M. Salehi, M. R. Sahebi, and Y. Maghsoudi, "Improving the accuracy of urban land cover classification using Radarsat-2 polSAR data," *IEEE J. Sel. Topics Appl. Earth Observ.*, vol. 7, no. 4, pp. 1394–1401, Apr. 2014.
[5] G. Satalino, A. Balenzano, F. Mattia, and M. W. Davidson, "C-band SAR data for mapping crops dominated by surface or volume scattering," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 2, pp. 384–388, Feb. 2014.
[6] S. Wang, H. Pan, C. Zhang, and Y. Tian, "RGB - D image-based detection of stairs, pedestrian crosswalks and traffic signs," *J. Vis. Commun. Imag. Representation*, vol. 25, no. 2, pp. 263–272, 2014.
[7] Y.-D. Zhang, S. Chen, S.-H. Wang, J.-F. Yang, and P. Phillips, "Magnetic resonance brain image classification based on weighted-type fractional Fourier transform and nonparallel support vector machine," *Int. J. Imag. Syst. Technol.*, vol. 25, no. 4, pp. 317–327, 2015.
[8] M. Ding and G. Fan, "Articulated and generalized Gaussian kernel correlation for human pose estimation," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 776–789, Feb. 2016.
[9] Z. Lu, S. Lu, G. Liu, Y. Zhang, J. Yang, and P. Phillips, "A pathological brain detection system based on radial basis function neural network," *J. Med. Imag. Health Informat.*, vol. 6, no. 5, pp. 1218–1222, 2016.

[10] S. Wang, J. Yang, G. Liu, S. Du, and J. Yan, "Multi-objective path finding in stochastic networks using a biogeography-based optimization method," *Simulation*, vol. 92, no. 7, pp. 637–647, 2016.
[11] S. Wang, P. Phillips, A. Liu, and S. Du, "Tea category identification using computer vision and generalized eigenvalue proximal SVM," *Fundam. Informaticae*, vol. 151, no. 1–4, pp. 325–339, 2017.
[12] X. Bai, B. Shi, C. Zhang, X. Cai, and L. Qi, "Text/non-text image classification in the wild with convolutional neural networks," *Pattern Recognit.*, vol. 66, pp. 437–446, 2017.
[13] Q. Lv, Y. Dou, X. Niu, J. Xu, J. Xu, and F. Xia, "Urban land use and land cover classification using remotely sensed SAR data through deep belief networks," *J. Sensors*, vol. 2015, pp. 1–10, 2015, doi: 101155/2015/538063.
[14] Y. Zhong, F. Fei, and L. Zhang, "Large patch convolutional neural networks for the scene classification of high spatial resolution imagery," *J. Appl. Remote Sens.*, vol. 10, no. 2, pp. 1–20, 2016.
[15] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.
[16] W. Zhao and S. Du, "Learning multiscale and deep representations for classifying remotely sensed imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 113, pp. 155–165, 2016.
[17] P. Li, P. Ren, X. Zhang, Q. Wang, X. Zhu, and L. Wang, "Region-wise deep feature representation for remote sensing images," *Remote Sens.-Basel*, vol. 10, no. 6, pp. 1–14, 2018.
[18] Q. Weng, Z. Mao, J. Lin, and W. Guo, "Land-use classification via extreme learning classifier based on deep convolutional features," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 704–708, May 2017.
[19] D. Marcos, M. Volpi, B. Kellenberger, and D. Tuia, "Land cover mapping at very high resolution with rotation equivariant CNNs: Towards small yet accurate models," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 96–107, 2018.
[20] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, and J. Paisley, "Hyperspectral image classification with Markov random fields and a convolutional neural network," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2354–2367, May 2018.
[21] N. Audebert, B. Le Saux, and S. Lefévre, "Deep learning for classification of hyperspectral data: A comparative review," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 159–173, Jun. 2019.
[22] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.
[23] S. K. Roy, S. Manna, T. Song, and L. Bruzzone, "Attention-based adaptive spectral-spatial kernel ResNet for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7831–7843, Sep. 2021.
[24] S. Liu and Q. Shi, "Multitask deep learning with spectral knowledge for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 12, pp. 2110–2114, Dec. 2020.
[25] X. Yu and S.-H. Wang, "Abnormality diagnosis in mammograms by transfer learning based on ResNet18," *Fundam. Informaticae*, vol. 168, no. 2–4, pp. 219–230, 2019.
[26] Z. Yuan and J. Zhang, "Feature extraction and image retrieval based on AlexNet," in *Proc. Int. Conf. Digit. Image Process.*, 2016, Art. no. 100330E.
[27] H. Liang and Q. Li, "Hyperspectral imagery classification using sparse representations of convolutional neural network features," *Remote Sens.-Basel*, vol. 8, no. 2, pp. 1–16, 2016.
[28] H. Liang and Q. Li, "Hyperspectral imagery classification using sparse representations of convolutional neural network features," *Remote Sens.-Basel*, vol. 8, no. 2, pp. 1–25, 2016.
[29] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 4, pp. 532–550, Jul. 1987.
[30] P. Maragos, R. W. Schafer, and M. A. Butt, *Mathematical Morphology and Its Applications to Image and Signal Processing*, vol. 5. New York, NY, USA: Springer, 2012.
[31] J. Serra and P. Soille, *Mathematical Morphology and Its Applications to Image Processing*, vol. 2. New York, NY, USA: Springer, 2012.
[32] P. Soille, *Morphological Image Analysis: Principles and Applications*. New York, NY, USA: Springer, 2013.
[33] D. B. S. Sagar, Q. Cheng, and F. Agterberg, *Handbook of Mathematical Geosciences: Fifty Years of IAMG*. Berlin, Germany: Springer, 2018.
[34] D. Tuia, R. Flamary, and N. Courty, "Multiclass feature learning for hyperspectral image classification: Sparse and hierarchical solutions," *ISPRS J. Photogramm. Remote Sens.*, vol. 105, pp. 272–285, 2015.
[35] D. Stathakis and A. Vasilakos, "Satellite image classification using granular neural networks," *Int. J. Remote Sens.*, vol. 27, pp. 3991–4003, 2006.
[36] D. Stathakis and I. Kanellopoulos, "Global elevation ancillary data for land use classification using granular neural networks," *Photogrammetric Eng. Rem Sens.*, vol. 74, pp. 55–63, 2008.
[37] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 683–697, Sep. 1992.
[38] S. K. Pal, S. K. Meher, and S. Dutta, "Class-dependent rough-fuzzy granular space, dispersion index and classification," *Pattern Recognit.*, vol. 45, pp. 2690–2707, 2012.
[39] S. K. Pal and S. K. Meher, "Natural computing: A problem solving paradigm with granular information processing," *Appl. Soft. Comput.*, vol. 13, pp. 3944–3955, 2013.
[40] R. Yasdi, "Combining rough sets learning- and neural learning-method to deal with uncertain and imprecise information," *Neurocomputing*, vol. 7, pp. 61–84, 1995.
[41] M. Banerjee, S. Mitra, and S. K. Pal, "Rough fuzzy MLP: Knowledge encoding and classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 9, no. 6, pp. 1203–1216, Nov. 1998.
[42] A. Ganivada, S. Dutta, and S. K. Pal, "Fuzzy rough granular neural networks, fuzzy granules, and classification," *Theor. Comput. Sci.*, vol. 412, pp. 5834–5853, 2011.
[43] D. Mellouli, T. M. Hamdani, J. J. Sanchez-Medina, M. B. Ayed, and A. M. Alimi, "Morphological convolutional neural network architecture for digit recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2876–2885, Sep. 2019.
[44] D. B. S. Sagar, *Mathematical Morphology in Geomorphology and GISci*. Boca Raton, FL, USA: CRC Press, 2013.
[45] P. Velickovic, "2D convolution," 2016. [Online]. Available: https://github.com/PetarV-/TikZ/blob/master/2D/20Convolution/2d_convolution.tex
[46] M. Thoma, "Max-pooling," 2016. [Online]. Available: https://github.com/MartinThoma/LaTeX-examples/tree/master/tikz/max-pooling
[47] L. A. Zadeh, "Fuzzy sets," *Inform. Control*, vol. 8, pp. 338–353, 1965.
[48] S. K. Meher, "Knowledge-encoded granular neural networks for hyperspectral remote sensing image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2439–2446, Jun. 2015.
[49] R. Jensen and Q. Shen, "Fuzzy-rough attribute reduction with application to web categorization," *Fuzzy Sets Syst.*, vol. 141, pp. 469–485, 2004.
[50] R. Jensen and Q. Shen, "Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 12, pp. 1457–1471, Dec. 2004.
[51] R. Jensen and Q. Shen, *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*. Hoboken, NJ, USA: Wiley, 2008.
[52] X. Suhui, M. Xiaodong, Z. Peng, and M. Ji, "Scene classification of remote sensing image based on multi-scale feature and deep neural network," *Acta Geod. et Cartogr. Sin.*, vol. 45, no. 7, pp. 7109–7121, 2016.
[53] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning earth observation classification using ImageNet pretrained networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 105–109, Jan. 2016.
[54] Y. Liu, Y. Zhong, F. Fei, Q. Zhu, and Q. Qin, "Scene classification based on a deep random-scale stretched convolutional neural network," *Remote Sens.-Basel*, vol. 10, no. 3, pp. 1–23, 2018.
[55] Q. Zhu, Y. Zhong, Y. Liu, L. Zhang, and D. Li, "A deep-local-global feature fusion framework for high spatial resolution imagery scene classification," *Remote Sens.-Basel*, vol. 10, no. 4, pp. 1–22, 2018.

**Dasari Arun Kumar** (Senior Member, IEEE) received the Ph.D. degree in spatial information technology from Jawaharlal Nehru Technological University (JNTU), Kakinada, India, in 2019.

He is currently working as an Associate Professor with the Department of Electronics and Communication Engineering and Center for Research and Innovation, KSRM College of Engineering, Kadapa, Andhra Pradesh, India. He worked as a Project Fellow with the Indian Statistical Institute, Bangalore, India. He has an experience of seven years and four years in teaching and research, respectively. He published research papers in reputed publishers like Springer, Elsevier and IEEE. His research of interests are pattern classification, granular neural networks, deep learning and remote sensing.