Multimedia Tools and Applications (2022) 81:28207–28230
https://doi.org/10.1007/s11042-022-12628-2

A Novel Grey Wolf Optimisation based CNN Classifier for Hyperspectral Image classification

Sandeep Kumar Ladi[1,2] · **G. K. Panda**[3] · **Ratnakar Dash**[4] · **Pradeep Kumar Ladi**[5] · **Rohan Dhupar**[6]

Received: 12 September 2020 / Revised: 18 May 2021 / Accepted: 9 February 2022

Abstract
Hyperspectral image (HI) analysis is becoming popular in remote sensing applications due to its high spectral resolution along with high spatial resolution compared to a Multispectral image. Classification of pixel vectors in a hyperspectral image (HI) to their respective classes is challenging. HI classification is an elementary task for land cover mapping, mineral exploitation, precision agriculture, etc. Optimal parameters are required to reduce the losses in the Convolutional Neural Networks (CNNs) and provide the most accurate results possible. Studies in this regard, so far have been made with manual selection of optimal parameters using traditional trial-and-error methods like the selection of loss function, a number of convolution filters, optimizer function, etc., and found to be strenuous and time-consuming. To alleviate these challenges of selecting the hyperparameters and observing the accuracy until a competitive value is reached, this paper uses a novel mechanism to classify Hyperspectral Images using Convolutional Neural Network (CNN) where the 6-hyperparameters of CNN are optimized with Grey Wolf Optimizer (GWO). The proposed GWO-based-CNN-HI model exhibits better classification accuracy (99.95%, 99.96%, and 99.99%) on three benchmark HI datasets in comparison to traditional models. Thus the novel GWO-based-CNN-HI model finds its suitability in applications of land cover classification, crop stage detection, specially in remote applications with limited computing power.

Keywords Feature extraction · Stationary Wavelet Transform (SWT) · Principal Component Analysis (PCA) · Hyperspectral Image classification · Deep Learning (DL) · Convolutional Neural Network (CNN) · Grey Wolf Optimizer (GWO) · Optimal Hyper-parameters

All authors have contributed equally to the work.

✉ Sandeep Kumar Ladi
sandeepkmriacr@gmail.com

Extended author information available on the last page of the article.

1 Introduction

Hyperspectral images have both spatial and spectral representations of a scene. They are different from color images acquired with a traditional RGB color camera, which divides the light spectrum into broad overlapping red, green, and blue image channels. Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) or satellites like EO-1 with its hyperspectral instrument Hyperion; both from NASA are used to capture the HI of a scene. Handheld sensors are also available for capturing HIs for numerous studies. HI-based classifications are challenging and have been evolving in the research domain since their early developments during the 1990s. To its credit, some of the recent real-life applications with HI classifications includes food grade evaluation [17, 23], defense [31, 55], clinical diagnosis [14], geological [56], precision agriculture [5] etc. The HI classification predicts the class of a pixel vector, like land, water, crops, etc. A hyperspectral image is represented by a three-dimensional hypercube (x, y, λ). Spatial dimensions are represented by (x, y), where x and y denote rows and columns, respectively, and λ signifies the spectral bands. The λ entries for each pixel of the data cube are the reflectance values of the scene at different wavelengths [16, 28]. High spectral dimension with hundreds of bands improves the discriminative ability of the classifier. However, there may be redundant bands that are strongly correlated. In the context of HI classification [25], the author showed that, as the number of hyperspectral bands increases, the number of training samples required to maintain the least possible statistical confidence for reliable HI classification also rises exponentially, making the classification process more challenging. To alleviate the above-mentioned problem we need to reduce data redundancy and/or try to obtain large number of training pixels for each class. Data redundancy can be achieved through spectral Dimension reduction methods, the process in which the informative bands are included and remaining non contributing bands are removed. Feature selection and feature extraction are two spectral dimension reduction techniques. Feature selection chooses a group of spectral bands from the given hyperspectral image that produces acceptable classification accuracy [15]. In the feature extraction process, higher dimensional spectral space gets transformed to lower-dimensional spectral space keeping most of the vital information [21, 79].

Complex spatial structure of objects and spectral diversity are the main challenges for achieving accurate classification accuracy [67]. Several HI classification methods using Spatial features [8, 27, 34], spectral features or pixel vectors [6, 37, 49] and combined Spectral and spatial features [67] are proposed in the literature. In the spectral feature or Pixel-based classification approach, each pixel's reflectance spectra of an object in an image is considered its feature. It represents a quantitative characteristics of an object. The set of these features are used by Pattern recognition technique for prediction and classification of pixel to its respective classes to which it belongs [7].

In pixel-wise classification (spectral classification) dimensional reduction is carried out prior to classification process [20]. Different classifiers like Random Forest classifier [6], Support Vector Machine (SVM) [49] are available. SVM is an effective approach for hyperspectral image [37] but the parameter assignment problem poses a challenge for accurate classification [40]. The Spectral classification [52] of the hyperspectral image considers the pixel vector (spectral features) as the only features. These spectral features alone cannot exploit the underlying geometric structure of the objects in the HI [36].

Spatial data is also significant for classification because spectrally similar neighboring pixels presumably belong to the same class [53]. The two dimensional or three dimensional Patches [33] are extracted from Hyperspectral image as a feature engineering which are fed to classifier for classification. Apart from using raw hyperspectral spatial data as

discriminative features, we can also derive different spatial features from the raw hyperspectral information. Several spatial feature extraction techniques can be found in the literature. Local Binary patterns [27], Gradient Local Auto-Correlations [8], Morphological profiles [34], Extended Morphological profiles (EMPs) [3], DWT-PCA-KNN [62], are few examples of spatial feature extraction. Handcrafted features like EMPs are considered as shallow features that cannot adapt to different datasets. They don't perform well in situations where a lot of regional changes occur in an image within a short interval of time [75]. SVM classifiers have performed well for spatial data classification.

Deep learning classification models have increased attention as they achieve better performance by avoiding complex feature engineering. Due to the hierarchical architecture of the models, they can extract discriminating features from the image [48]. In an image, hierarchy starts from the pixels. Assembly of corresponding pixels gives the boundaries or edges. Edges combine to give parts. Finally, objects are obtained by assembling various parts. Hierarchical layers of the DL models learn these high-level features that are invariant to local changes from the image's complex structure. Remote sensing applications are now utilizing the DL models for many challenging tasks. Non-iterative ANN-based approaches [26, 57–59], Convolutional Neural Networks (CNNs) [30], 3-D CNNs [73], Multiscale CNNs [65], Autoencoders (AEs) [50], Deep belief networks [9], Joint Spectral and Spatial CNN [60], Deep Residual Networks [28], Recurrent Neural Networks [39], Spectral and Spatial LSTMs [78], Spectral Spatial Unified Network (SSUN) [67], Spectral–Spatial Residual Network (SSRN) [77], Random Patches Network (RPNet) [66], models are the typical DL models used for classifying Hyperspectral images. The patches of the images need to be flattened before input to models like SAEs [51], DBNs [36]. This flattening results in loss of spatial information [1]. CNN models are found to be a good alternative for this problem. Subsequently, 3-D CNN model was proposed for taking the spectral information in consideration along with spatial information. The convolution operation is carried out on 3-D patches. Results showed that this model failed to classify small objects and class boundaries. Authors proposed a model in [76] that incorporates balanced local discriminant embedding (BLDE) and CNN for spectral and spatial feature extraction. Feature extraction and classifier training are done separately, each having its objective functions. Learning the optimal features was not possible during classifier training and hence failed to achieve a stable accuracy. Authors in [76] proposed the spectral-spatial unified networks model that integrated spectral, spatial attribute extraction, and classification.

Machine Learning and Deep Learning models perform well after tuning their hyperparameters. The trial and error method of choosing the optimal hyperparameters is a tedious job and time-consuming [64]. This paper emphasizes using optimization methods for building an optimal Deep Learning model. Optimization plays a major role in making a system fully functional and as effective as possible. Practically some problems limit the performance of traditional optimization techniques. Human beings have learned many things from nature and have come out with efficient algorithms to solve the problems. These algorithms are motivated by nature and its processes. Nature-inspired algorithms (NIAs) [71] are the name given to such algorithms. The primary goal of NIAs is search and optimization. They are one of the branches of computational intelligence. There are two categories of Nature Inspired algorithms. They are: Evolutionary algorithm [4] and Swarm optimization [13]. Evolutionary algorithms like the Genetic algorithm mimic Darwin's theory of natural selection. Some key concepts taken from this theory are:

1. Offsprings are created by reproduction, mutation, etc.
2. Fittest Individuals are those who adapt to the environment survive. They reproduce and pass their genetic traits to offsprings.

3. Populations adapt to their environment and mutate. These mutations accumulate over time giving rise to new species.

On the other hand, swarm intelligence acts in accordance with natural and human-made systems comprised of multiple individuals coordinating using distributed control with self-organization. It focuses on the combined behaviors employing local interaction among individuals with one another and with their surroundings. Ant colonies, flocks of birds, herds of land animals are some cases of swarm intelligence. They are mostly used to find the best solution for a given problem. Grey wolf optimizer (GWO) [38] a swarm intelligence technique has shown competitive performance compared to other methods. This algorithm draws its inspiration from how the wolves strictly follow a nomenclature of social hierarchy and hunt for their prey in the group.

In the literature, we can find many applications of the Nature Inspired algorithm in Hyperspectral Image. In [21], authors proposed a constrained optimization method in which they have used the Genetic algorithm for feature selection in Hyperspectral image and achieved promising improvement in overall accuracy. The work proposed in [41], evaluated nature-inspired algorithms like gravitational search algorithm [44], harmony search [19], particle swarm optimization [68], Firefly algorithm [70], and Bat algorithm [69] for band-selection for Hyperspectral image.

HI classification using deep learning is one of the popular and efficient techniques in the current scenario. In this context, tuning the deep learning model's hyperparameters to obtain high performance is challenging. The main contribution in this paper is given below:

1. We propose a CNN model with its hyper-parameters optimized using Grey Wolf Optimizer(GWO). As per the authors' knowledge, GWO is used for the first time as an optimizer for the classification model of HI.
2. CNN hyper-parameters such as number of convolution layers, activation function, optimizer, batch size, number of epochs to run, etc. were optimized, as performance in CNN always lies in its choice optimal hyper-parameters.
3. This method reduces the time required for tuning and building a deep learning model as compared to using the trial and error method.
4. GWO has the least operators and can optimize multiple variable problems, and provides competitive results compared to other nature-inspired algorithms. Which in turn has the advantage of less Computational cost. In addition to this, it also has less space complexity.

The paper compares the classwise accuracies along with the Overall accuracies and Kappa accuracies with four state-of-the-art methods of SSUN [67], SSRN [77], RPNET-5 [66], and MCNN [32]. In SSUN [67], both the feature extraction and classifier frameworks have a unified objective function. In addition to the above-mentioned contribution, a spectral band grouping strategy is adopted to enhance the Spectral classification accuracy and a multi-scale CNN is introduced to prevent losses due to the convolution and pooling. The RPNET-5 [66] model focussed in reducing the time consumption for deep learning framework. The model uses the random patches that enables the classifier to train with limited training samples. RPNET-5 [66] utilizes both shallow and deep features for the classification process. SSRN [77] proposed the use of two separate channels for deep spectral and spatial residual networks to improve the accuracy and feature extraction process. In [62], authors proposed an effective supervised spatial feature extraction method using SWT [11, 30, 42], that focus on extraction of those features that improve the classification accuracy of the Hyperspectral images. The spectral dimension reduction is implemented using PCA [46] by selecting only first ten principal components. The extracted features after dimensional reduction process are fed to K-nearest neighbor classifier [12]. The authors in [62] showed that SWT-PCA

method outperformed the DWT-PCA, and conventional PCA algorithms. HI classification using MCNN model proposed by the authors in [32]. They introduced mapping layers for spectral–spatial feature learning in HI. They obtained the mapping kernels by decomposing the input HSI patches, and then combining these specialized layers for HI with 3D convolutional layers for deep feature extraction and classification. The MCNN model reduced the computation time and alleviated the problem of decreasing accuracy with deeper networks.

There are several methods for classification of HIs. Trial-and-error methods for setting the hyperparameters of a classifier is used to obtain the competitive accuracy. This is a strenous task. This paper incorporates GWO for finding the optimum hyper-parameters for designing a CNN model. The model is used for HI feature classification. In section II, Stationary wavelet transform (SWT), PCA, CNN, and GWO are elaborated. Section III outlines the proposed methodology. Experiments and analysis are described in Section IV, with Section V as the conclusion.

2 Proposed framework

In this section we elaborate the proposed GWO-based-CNN-HI framework. The block diagram and the algorithm of the proposed framework is illustrated in Fig. 1 and Algorithm 1 respectively.

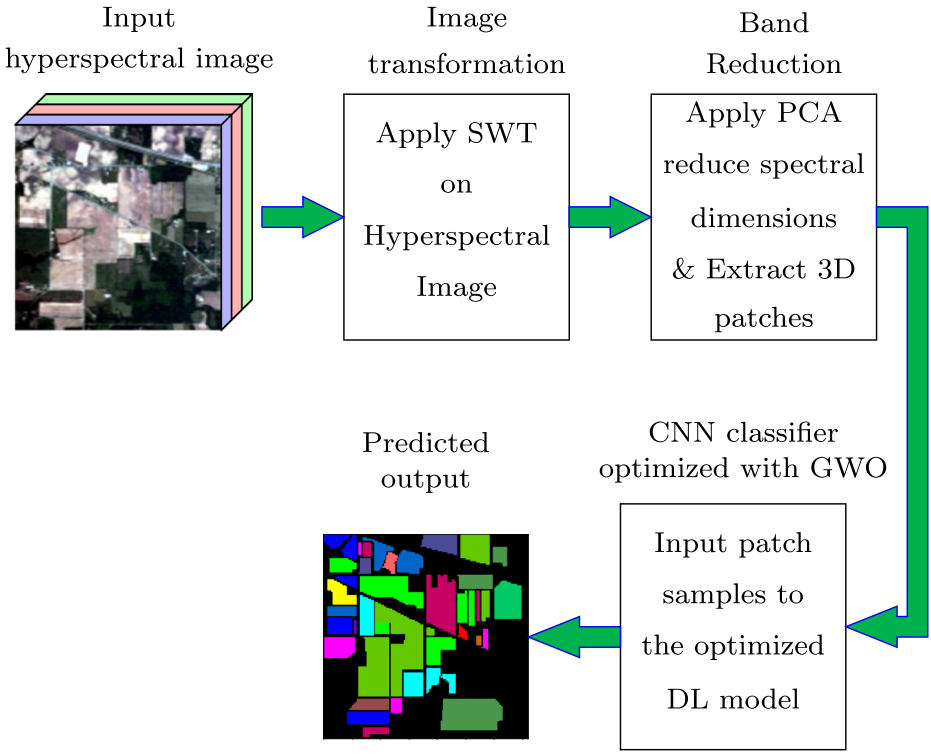


Fig. 1 SWT wavelet transform is carried out on HSI cube. The transformed image is given to PCA for band reduction. Patches extracted from this dimensionaly reduced image is given as input to the GWO optimized CNN

Algorithm 1 GWO_based_CNN_HI algorithm.

Input:
- Hyperspectral Image;
- A benchmark function(A CNN model in this case);
- Wavelet type, decomposition level for SWT;
- Number of Principal Components r;
- Patch window size w;
- Number of search agents S;
- Number of iterations i;
- Number of dimensions/hyperparameters to be tuned h;

Output: CNN model containing the optimal values of hyperparameters in X_α

Reshape the given input Hyperspectral Image, I'$^{h\times w\times d}$ to I$^{k\times d}$ where $k = h \times w$
for *Each row in image* $I^{k\times d}$ **do**
    Apply 1D SWT on row of length d to generate four coefficients cA_1, cD_1, cA_2, cD_2 which have the same length of the input row.
    Stack the features cA_1, cD_1, cA_2, cD_2
end
Reshape the coefficient matrix into 3D image, $I_{swt}^{h\times w\times 4d}$
Apply Principal Component Analysis on the transformed Image, I_{swt} for dimensional reduction to obtain $I_{pca}^{h\times w\times r}$ where r<4d
Extract N Patches P$_N^{w\times w}$ from I_{pca}. where N is the total number of labelled pixels of the given image. The groundtruth of the center pixel of the corresponding patch is considered as the groundtruth of the entire patch.
Split 80% of N patches into the train patches, P_{train} along with their corresponding labels y_train
Consider 20% of N test patches P_{test} along with their corresponding labels as test labels y_{test}.
As the dataset is of imbalance class, apply oversampling process on the train patches, p_train and their corresponding train labels y_train to oversample weak classes.
Design CNN benchmark model and select certain hyperparameters for obtaining their optimal values.
Pass the CNN benchmark function to the GWO() function as one argument. Other arguments are Number of hyperparameters(dimension) to be optimized, Randomly initialized q values of each of these hyperparameters called search space is generated that is of d dimensions, These values represents q positions of the search agents or wolves. The last argument is the total number of iterations,i.
Initialize the parameters a, A, and C of GWO algorithm.
Pass the q positions of the wolves are to cnnbenchmark() function evaluate the fitness. Take the best three results and denote them as X_α, X_β and X_δ that are best, second best and the third best search agents respectively.
while $t <$ *maximum iterations* **do**
    for *each search agent* **do**
        Modify the position of current search agent in accordance with equations 1,2,3 and 4;
    end
    Modify a,A, and C;
    Compute the fitness of all search agents;
    Update X_α,X_β and X_δ;
    Increment the value of t by 1;
end
return X_α;
Design the CNN model using the corresponding values of the X_α as the values of the hyperparameters and verify the accuracy of the CNN model.

2.1 Stationary Wavelet Transform (SWT)

Wavelet transforms play a major role in the signal processing domain. They carry rich information in both time and frequency. Wavelets are widely used in literature for image processing. Some wavelet transforms developed for digital signals like image are Discrete wavelet Transform (DFT), Stationary Wavelet Transform (SWT) etc. Wavelet transform using SWT for Image analysis has shown better performance in the field of denoising of image and edge detection [45]. SWT extracts the translation invariant features of an Image. In [30], authors have used Stationary Wavelet Transform (SWT) on Hyperspectral images for extracting spatial and spectral features and obtained promising results during their classification process using a CNN model. Flow diagram of SWT is shown in Fig. 2. The mathematical derivations are illustrated in [42]. Each pixel vector of a Hyperspectral image is a one dimensional signal, say S_i, where i is the ith pixel vector in an image. One dimensional Stationary wavelet transform with two level decomposition is applied on S_i using PyWavelets [24], an open source python toolbox for obtaining wavelet transforms. In the first level of the transformation process we obtain Approximation coefficients cA_1, and Detail coefficients, cD_1. Second level of transformation is applied on the cA_1 that further gives Approximation coefficients, cA_2 and cD_2. Thus, we obtain 4 coefficient vectors (cA_1, cD_1, cA_2, and cD_2) for each ith pixel vector, S_i. Finally these coefficients are stacked upon each other resulting in increasing the spectral features by four times. This transformation process carried out in this work is illustrated in Fig. 2.

The given input HI, $\mathrm{I}^{h\times w\times d}$ is reshaped to $\mathrm{I}^{k\times d}$ where $k = h \times w$. Then, 1D SWT is applied on each row of length d to generate four coefficients cA_1, cD_1, cA_2, cD_2 which have the same length of the input row. These features cA_1, cD_1, cA_2, cD_2 are stacked and reshaped them into 3D image, $I_{swt}^{h\times w\times 4d}$.

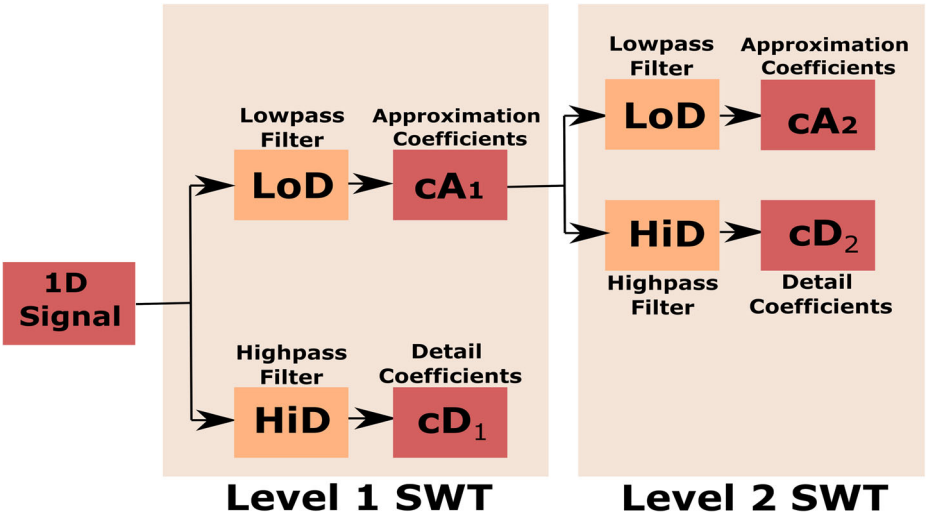


Fig. 2 Flow diagram of 1D SWT

2.2 Principal Component Analysis(PCA)

HI involves the processing of a large number of bands that increases the time needed and computational cost. Spectral Dimension reduction is the primary task carried out before any HI analysis. Principal components analysis (PCA) [46] is a dimensional reduction technique that uses the statistical properties in HI to identify correlations and band dependencies and

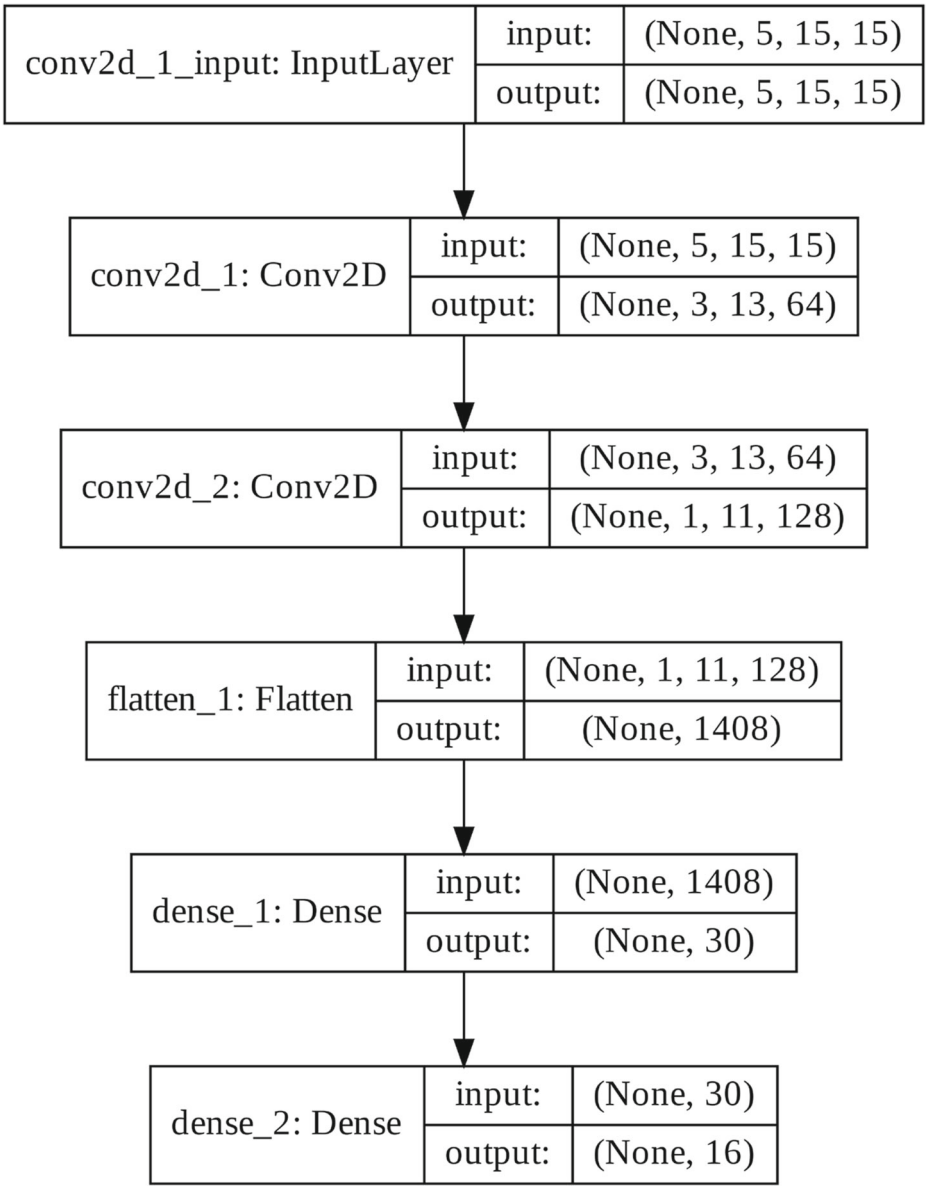


Fig. 3 CNN model for Indian pines dataset

applies an orthogonal transforms to transform it into a lower dimension feature space without loss of any important information. The mathematical derivations are illustrated in [22] and its brief illustrations can be found in [35].

We applied PCA on the features extracted from the SWT and reserved r principal components. This dimensionally reduced image is represented as $I_{pca}^{h \times w \times r}$.

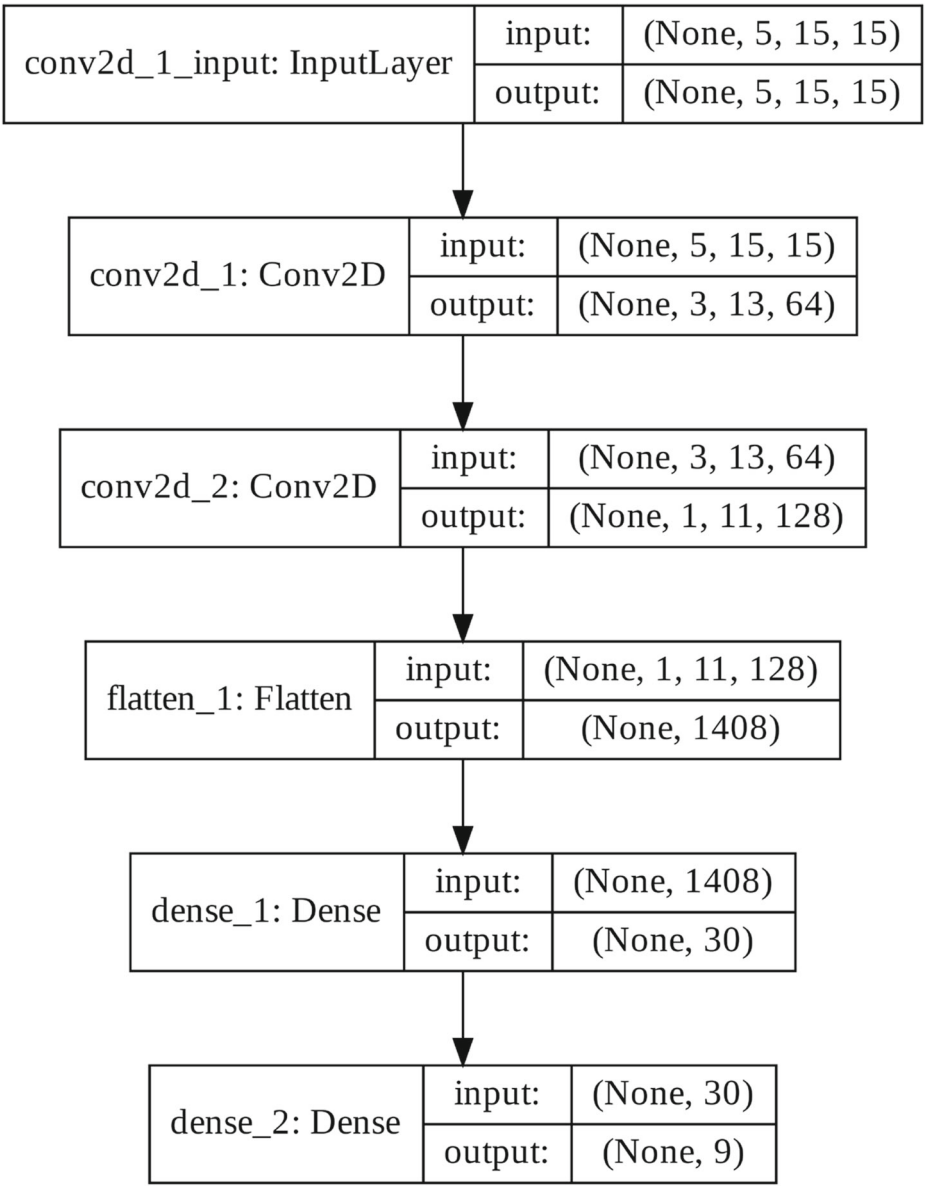


Fig. 4 CNN model for Pavia University dataset

2.3 Convolutional Neural Networks (CNNs)

CNN [74] is a popular deep learning technique widely used for image classification. In the case of a CNN, the neuron in a layer is associated with a small area of the layer before it. CNN draws its inspiration from the functioning of the visual cortex in animals [36]. The visual cortex has a small region of cells where some cells are [33] fired by horizontal edges, some get fired by vertical edges, and some by diagonal edges of the image.

Hyperspectral image classification is a pixel classification [33] technique that is also called semantic segmentation. It is shown that instead of giving the entire image, patches of image give better performance in classification [10, 72]. The patches are treated as the samples for the classification. For the supervised classification process, a label corresponding to the the center pixel of a patch is considered as the label for the entire patch. In case of a normal image classification the set of images along with their labels are split into two separate sets called training sets and testing sets. In the case of hyperspectral images, the patches and their corresponding class labels are divided into training and testing samples. This paper also follows the same methodology for samples and labels for the purpose of classification.

The structure of our proposed CNN model used for classification of the three benchmark datasets are illustrated in Figs. 3, 4 and 5. There are two 2D convolutional layers. The training patches each of dimensions $5 \times 15 \times 15(channels \times height \times width)$ are input to the first 2D Convolutional layer. This convolutional layer employs 64 filters of dimension 3×3 and the activation function is determined by the GWO algorithm. The convolutional layer is followed by a Dropout layer to prevent the model from overfitting by setting the fraction value (0 to 1) representing the percentage of neurons that will be dropped. The fraction value is determined by the GWO algorithm. The Dropout layer is followed by a second 2D convolution layer that has 128 filters of shape 3×3 and the activation function is determined by the GWO algorithm. The output features obtained from the Dropout layer are flattened by reshaping process in Flatten layer before passing it to the first Dense layer. The linear operation is carried out in the first Dense layer that has $6 \times$ number of principal components number of neurons. The activation function in this layer is determined by the GWO algorithm. The output from the first dense layer is fed to the second Dropout layer with the fraction determined by the GWO. The output from the Dropout layer is input to the final dense layer with output neurons equal to the number of classes, with Softmax activation function. Softmax function takes the vector of real numbers as input vector and normalizes it into a probability distribution consisting of n (Number of classes) probabilities. A probability distribution implies the likelihood of the input feature vector belonging to all the classes. This resultant vector sums up to 1. It determines the target class of the given input vector.

2.4 Grey Wolf Optimizer (GWO)

GWO is an optimization algorithm that imitates the leadership positions exhibited by a batch of grey wolves popular for their group hunting strategy. The hunting process comprises of three stages: looking for the prey, blocking and attacking the prey. There are four categories of wolves based on their fitness. They are as follows:

1. α: It is the leader wolf that may be male or female. It takes the decision for hunting, sleeping place etc. Other category wolves acknowledge α by holding their tails down.
2. β: They are the subordinates of α and they help the latter in decision making. They consider the best candidate to become α when the existing α dies or becomes old. β ensures the discipline from its subordinates and gives feedback to the α.

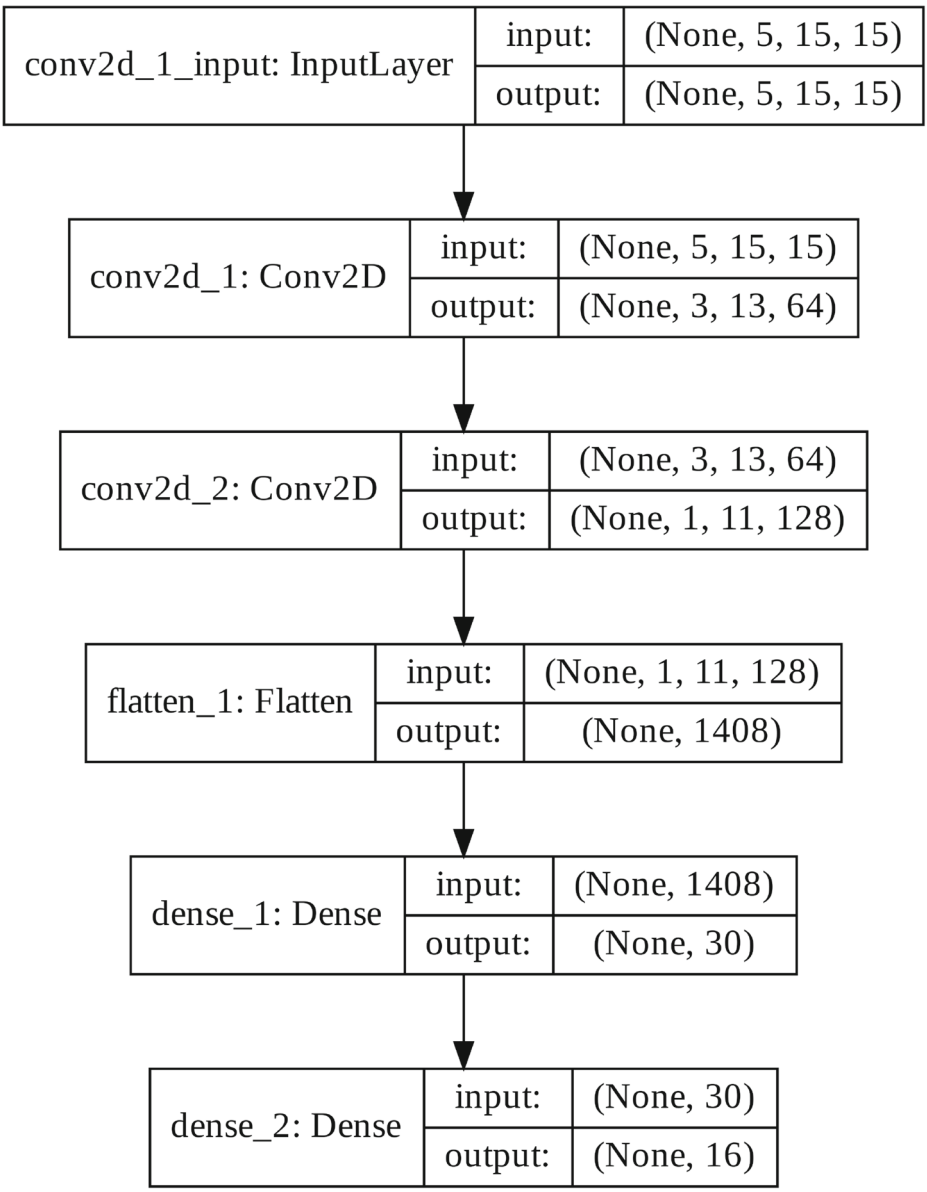


Fig. 5 CNN model for Salinas dataset

3. δ : This is a subordinate that are superior to ω and report to α and β. δ wolves keep a watch at boundaries, protect the pack, assist α and β in hunting, and take care of ill, weak and wounded wolves.
4. ω: They are the weakest wolves in the social hierarchy. They are like scapegoats in the pack. They are the last ones permitted to eat.

The wolves explore for the prey across all the boundaries. Upon discovering the prey, they surround it from all boundaries. When the prey is unable to move, it is attacked by the

wolves. α wolf is responsible for attacking the prey. Positions are updated so that the wolves move towards their goal of reaching the prey. These wolves will encircle the prey and keep updating their positions with respect to α, β, and δ search agents so that they can move closer to the goal. Finally, the best position of α, β, and δ are considered as the three best solutions and they are saved. The remaining wolves are obligated to change their positions conforming to the position of α, β, and δ. Grey wolves mainly search depending on the location of α, β, and δ. Figure 6 describes the hunting process of grey wolves. In GWO it is modeled by utilizing a vector A, which is an arbitrary value generated by a wolf. A has arbitrary values more than 1 or smaller than -1. This causes the search agent to diverge from the prey. When $| A | > 1$ the wolves search for the better prey. The grey wolves encircle the prey during the hunt. Mathematically, encircling is given by the equations below.

$$\vec{\mathbf{D}} = | \vec{\mathbf{C}}.\vec{\mathbf{X}}_{\mathbf{p}}(t) - \vec{\mathbf{X}}(t) | \tag{1}$$

$$\vec{\mathbf{X}}(t+1) = \vec{\mathbf{X}}_{\mathbf{p}}(t) - \vec{\mathbf{A}}.\vec{\mathbf{D}} \tag{2}$$

t denotes the current iteration, $\vec{\mathbf{A}}$, $\vec{\mathbf{C}}$ are coefficient vectors.$\vec{\mathbf{X}}_{\mathbf{p}}(t)$ and $\vec{\mathbf{X}}$ are the position vectors of the prey and wolf respectively. $\vec{\mathbf{A}}$ and $\vec{\mathbf{C}}$ are found as below.

$$\vec{\mathbf{A}} = 2.\vec{\mathbf{a}}.\vec{\mathbf{r}}_1 - \vec{\mathbf{a}} \tag{3}$$

$$\vec{\mathbf{C}} = 2.\vec{\mathbf{r}}_2 \tag{4}$$

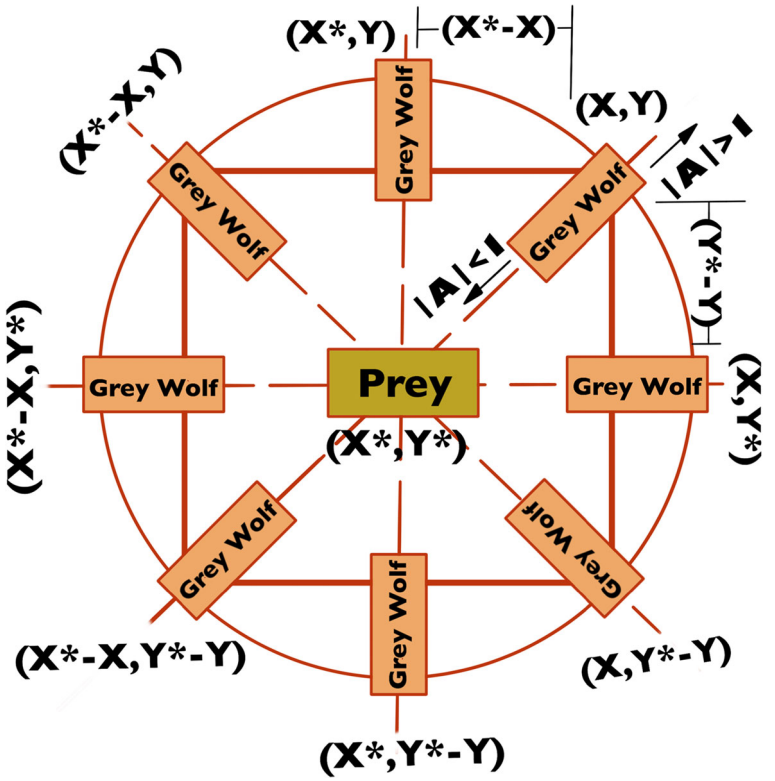


Fig. 6 Figure showing the hunting strategy of Grey wolves [38, 43]

Table 1 Complexity of the proposed algorithm

Method	Complexity	Description
GWO	$\mathcal{O}(P \times d \times maxIter)$ [18]	P=Population size, d = Dimension of the problem, maxIter= Maximum number of Iterations
PCA	$\mathcal{O}\left(N \times D \times min(N, D) + D^3\right)$ [54]	N = Number of samples, D = Number of features
SWT	$\mathcal{O}\left(n \log_2 n\right)$ [63]	n = Number of datapoints in the signal
All Convolutional layers of CNN	$\mathcal{O}\left(\sum_{l=1}^{d} n_{l-1}.s_l^2.n_l.m_l^2\right)$ [29]	l = Index of the convolution layer, n_l = Number of filters in the l^{th} layer, n_{l-1} = Number of intput channels of the l^{th} layer, s_l = Spatial size of the filter, m_l = Spatial size of the output feature map

Elements of $\vec{\mathbf{a}}$ are repeatedly reduced linearly from 2 to 0, $\vec{\mathbf{r}}_1$ and $\vec{\mathbf{r}}_2$ are the random vectors in [0,1]. The complexity of the proposed algorithm is show in the Table 1.

3 GWO-based-CNN-HI Model

3.1 Methodology

The hyperparameters of the proposed CNN model that are to be optimized are considered as the dimensions of the problem. The proposed CNN function is the benchmark function for our optimization problem. The GWO algorithm is applied to search and hunt for the target accuracy that is treated as the prey. On the completion of all iterations, the GWO algorithm stops and it returns the Accuracy which is denoted by X_α and is considered as the fittest wolf. Further, the algorithm returns the tuned hyper-parameters that have resulted in obtaining the highest accuracy.

The applications of GWO are forecasting commodity price, image processing, training multilayer perceptrons etc. This paper shows the use of GWO for selecting the optimal

Table 2 Optimal hyperparameters obtained from GWO algorithm

Hyper parameters	Optimal Value
Activation function of CNN layer	'relu'
Activation function of Dense layer	'relu'
Optimizer	'adam'
Dropout neurons	0
Batch size	32
Number of epochs	30

hyperparameters for proposed CNN model to boost the performance. Hyperparameters of a CNN model are the number of layers, hidden units in a layer, the activation function in a layer, the filter size for a layer, the positioning of these layers within the network, etc. For this experiment, activation function for CNN layer, activation function for a feedforward layer, optimizer for training the model, batch size, number of epochs, number of dropout layers Table 2 illustrates the optimal values obtained from the proposed method are chosen as the hyper-parameters.

4 Experimental results and discussion

4.1 Experimental data sets

Experiments are performed on three benchmark hyperspectral image datasets of Indian pines [47], Pavia University [47] and Salinas [47].

Indian pines dataset was captured by AVIRIS sensor in the month of June. They have Spatial dimension of 145x145 with 224 spectral bands with each having bandwidth of 0.4-2.5μm. Two-third of image contains agriculture area and rest is forest area. Dual-line highways, small roads, railway line, houses, buildings are also the parts of the scene. Crops like soyabean, corns constitute 5% of the image. Indian pines dataset possesses 16 classes which can be found in the groundtruth matrix provided as a separate file along with the data. There is another dataset file which contains 200 spectral bands obtained by removing water absorption bands.

PaviaU dataset contain the images of Pavia in North Italy obtained from the ROSIS sensor. The images have the dimension of 610x610 with 103 spectral bands. Groundtruth of the image is also avaliable that has 9 classes.

Salinas dataset belongs to Salinas Valley, California captured by AVIRIS Hyperspectral sensor containing 224-band and with 512 lines by 217 samples.

4.2 Framework settings

This section illustrates the parameters used for the experiment. The Indian pines, PaviaU and Salinas images undergo Stationary Wavelet Transformation (SWT) process resulting in spectral-spatial feature extraction for classification purpose. The two level 1D-SWT is used with the db1 wavelet. Principal Component Analysis (PCA) removes redundant spectral dimensions of the features. In this experiment the number of principal components chosen as 5 for Indian pines datasets, Pavia University and Salinas datasets. The patches of window size = 15 are extracted from the datasets. Patches are extracted from these image features that are now the samples for classification. In this paper samples are split into 20% tesing data and 80% training data for training and evaluation of the proposed CNN model. The training patches belonging to weak classes are oversampled to alleviate the problem of class imbalance. The label for a given patch is the groundtruth value corresponding to its centre pixel. In this experiment GWO algorithm available in NiaPy [61] python library is used. The GWO algorithm is used to get the optimal number of convolution layers, dropout percentage, activation function to be used for convolution layer, activation function for the feed forward layer, optimizer to be used to fit the model, number of epochs to run, and the batch size for training in the CNN model that gives the best accuracy. Learning rate for the proposed CNN model is set to 0.000001 with a factor of 0.9 and patience as 25. The parameters Activation function of Convolutional layer, Activation function of Dense layer,

Optimizer, Dropout neurons, Batch size, Number of epochs along with the CNN model is input to the GWO algorithm. The number of dimensions for GWO is set to 9 representing the number of hyperparameters to be optimized and the number of iterations is set to 100. The population parameter is initialized to 40 wolves. These population try to explore the target accuracy that is the prey. In the first step the fitness of the entire population is computed using the benchmark function. The wolf with highest fitness is considered as the current X_α or best search agent, second fittest as the current X_β or second best search agents, and the third fittest as the current X_δ or third best search agent. The remaining 37 search agents are considered as omega search agents. In the second step, third parameter of the GWO, the number of function evaluation, updates the position of 37 omega search agents with respect to the centroid of the positions of best search agents. This update is done 40 times for each iteration. The values a, A and C will be updated using random values. The value A is decreased from 2 to 0 linearly, after all search agents positions gets updated. If $| A | < 1$ then a new prey is found. Else if$| A | > 1$ then this search is continued on this prey.

4.3 Classification results and analysis

We compared our proposed GWO-based-CNN-HI framework with with four state of art Deep learning frameworks, SSUN [67], SSRN [77], RPNET-5 [66], and MCNN [32]. Tables 3 and 4 show the individual class accuracy along with Overall accuracy (OA), and Kappa coefficient (KA) obtained from the experiment on Indian pines, Pavia University data respectively. Tables 5, 6, and 7 show the classwise accuracy, Average accuracy (AA), Overall accuracy (OA), and Kappa accuracy (KA) of the proposed model for Indian pines,

Table 3 Classification results for Indian Pines Data Set [47]

Class	SSUN [67]	SSRN [77]	RPNET-5 [66]	MCNN [32]	GWO-based-CNN-HI
1	100	97.82	96.25	98.3	100
2	97.26	99.17	94.02	96.2	98
3	99.36	99.53	96.59	95.3	98
4	99.85	97.79	99.20	94.2	97
5	99.56	99.24	99.00	98.2	100
6	99.75	99.51	99.69	98.7	100
7	99.58	98.70	91.67	96.4	100
8	100	99.85	99.91	100	100
9	100	98.50	100	92.2	75
10	98.15	98.74	96.36	97.8	98
11	97.15	99.30	93.08	99.6	98
12	99.52	98.43	97.72	97.2	100
13	99.81	100	99.70	100	100
14	99.39	99.31	99.21	99.7	99
15	99.05	99.20	95.76	98.8	100
16	100	97.82	98.37	95.8	100
OA	98.40±0.37	99.19±0.26	96.09±0.66	98.3±0.2	99.2
KA x 100	98.14±0.43	99.07±0.30	95.46±0.76	98.0±0.3	99.2

Table 4 Classification results for Pavia University Data Set [47]

Class	SSUN [67]	SSRN [77]	RPNET-5 [66]	MCNN [32]	GWO-based-CNN-HI
1	99.32	99.92	98.35	99.6	100
2	99.36	99.96	99.52	99.4	100
3	99.48	98.46	98.97	99.4	98
4	99.64	99.69	99.41	99.8	99
5	99.93	99.99	99.99	99.9	100
6	99.74	99.94	99.88	99.8	100
7	99.86	99.82	99.47	98.8	99
8	99.34	99.22	99.20	98.9	99
9	99.91	99.95	99.90	98.4	100
OA	99.46±0.32	99.79±0.09	99.34±0.12	99.5±0.2	99.86±0.04
KA x 100	99.26±0.44	99.72±0.01	99.10±0.17	99.3±0.2	99.84±0.04

Salinas and Pavia University Data Set respectively. Figure 7a, b and c shows groundtruth images of the Indian pines and Pavia University and Salinas respectively. The corresponding predicted images are shown in Fig. 8a, b, and c respectively. The plots for accuracy and

Table 5 Training and Test samples used in Indian Pines Dataset [47]

Class	Training	Testing	Accuracy SWT-PCA-CNN [30]	Accuracy GWO-based-CNN-HI
1	1961	9	100	100
2	2284	286	97	98
3	1992	166	95	98
4	1900	47	95	97
5	1930	97	98	100
6	1752	146	100	100
7	1955	5	100	100
8	1910	96	100	100
9	1968	4	100	75
10	2334	194	100	98
11	1964	491	95	98
12	1896	119	100	100
13	1968	41	97	100
14	2024	253	99	99
15	1854	77	98	100
16	1998	19	100	100
AA			98.56	97
OA			99.07	99.2
KA x 100			97.30	99.2

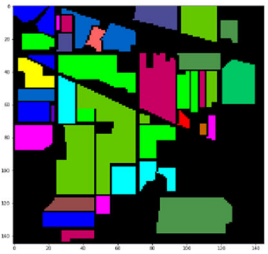
Table 6 Number of Training patches and Test patches used in Salinas Data Set [47]

Class	Training Samples	Testing Samples	Accuracy (%) GWO-based-CNN-HI
1	9642	402	100
2	8943	745	100
3	9486	395	100
4	8920	279	100
5	8568	536	100
6	9501	792	100
7	8589	716	100
8	9017	2254	100
9	9924	1241	100
10	7866	656	99.8
11	9405	213	100
12	9252	385	100
13	8796	183	100
14	9416	214	100
15	11628	1454	99.9
16	8676	361	100
AA			99.98
OA			99.31
KA x 100			99.03

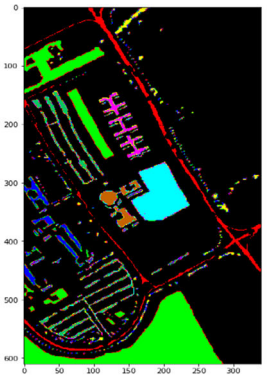
loss for v/s epoch obtained from the experiment for Pavia University, Salinas and Indian pines datasets are given in Figs. 9, 10 and 11 respectively.

Table 7 Number of Training patches and Test patches used in Pavia University Data Set [47]

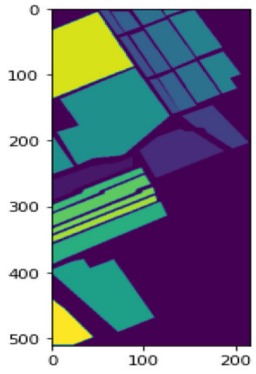
Class Number	Training Samples	Testing Samples	Accuracy SWT-PCA-CNN [30]	Accuracy GWO-based-CNN-HI
1	15915	1326	99.62	100
2	14919	3730	99.67	100
3	15111	420	93.80	98
4	14706	613	100.00	99
5	15064	269	100.00	100
6	16092	1006	99.30	100
7	14896	266	98.49	99
8	14725	737	96.20	99
9	15160	189	100.00	100
AA			98.56	99.4
OA			99.07	99.81
KA x 100			98.6	99.80



(a) Groundtruth image for indian pines dataset

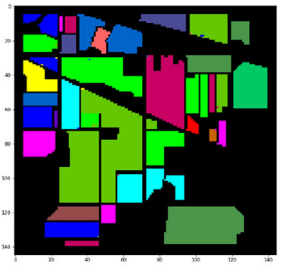


(b) Groundtruth image for PaviaU dataset

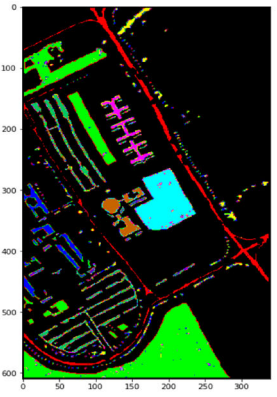


(c) Groundtruth image for Salinas dataset

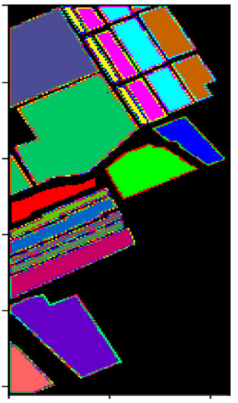
Fig. 7 Groundtruth images of three hyperspectral datasets



(a) Predicted image of indian pines dataset

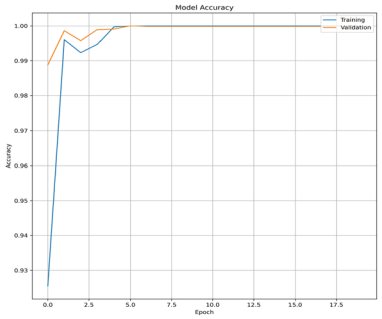


(b) Predicted image of paviaU dataset

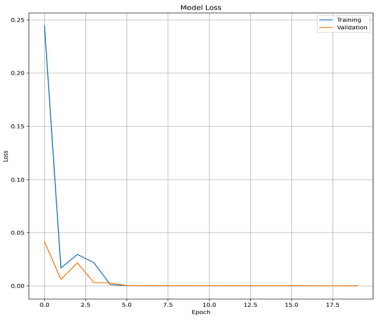


(c) Predicted image of Salinas dataset

Fig. 8 Predicted images of three hyperspectral images

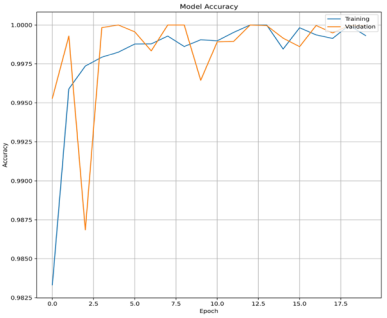


(a) Accuracy v/s epochs by our proposed GWO-based-CNN-HI framework for Indian Pines

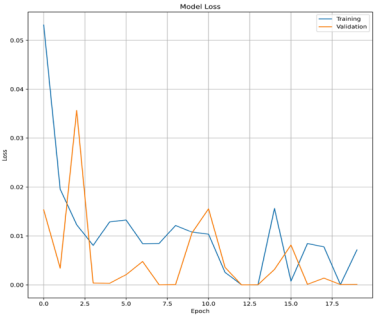


(b) Training and Testing Loss v/s epochs for Indian pines dataset after achieving the optimal hypermaters for the proposed GWO-based-CNN-HI framework

Fig. 9 Plot of loss and accuracy v/s epochs for Indian Pines dataset

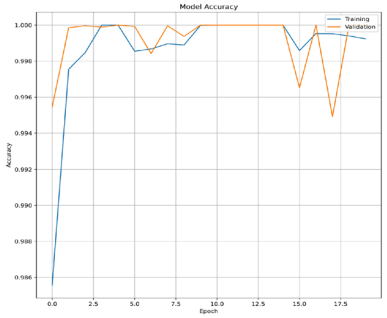


(a) Accuracy v/s epochs plot of the proposed GWO-based-CNN-HI framework for Salinas dataset

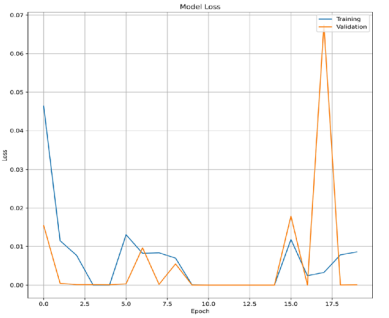


(b) Training and Testing Loss v/s epochs for Salinas dataset after achieving the optimal hypermaters for the proposed GWO-based-CNN-HI framework

Fig. 10 Plot of loss and accuracy v/s epochs for Salinas dataset



(a) Accuracy v/s epochs by our proposed GWO-based-CNN-HI framework for Pavia University dataset



(b) Training and Testing Loss v/s epochs for pavia University dataset after achieving the optimal hypermaters for the proposed GWO-based-CNN-HI framework

Fig. 11 Plot of loss and accuracy v/s epochs for Pavia University dataset

Table 8 Performance of the proposed model on the three datasets with Window size=15 and number of Principle Components=5

Dataset	Test ratio	Kappa score*100	Overall Accuracy(%)
Salinas	0.4	99.99	99.99
	0.6	99.98	99.98
	0.8	99.82	99.84
Indian Pines	0.4	99.88	99.90
	0.6	99.33	99.41
	0.8	97.25	97.58
Pavia	0.4	99.93	99.95
	0.6	99.90	99.92
	0.8	99.58	99.68

Table 9 Overall Accuracy(%) of our proposed framework with different patch sizes with 5 principal component bands

Patch Size	Indian Pines	Pavia University	Salinas
7X7	97.90	99.62	99.58
11X11	99.21	99.89	99.99
15X15	99.90	99.96	100

With 30 epochs of training, the proposed model gives competitive results in accuracy. The optimal values of the hyper-parameters obtained from the GWO method are as given in the Table 2. The structure of the proposed CNN models are shown in Figs 3, 4 and 5.

To test the validity and distinguishable performance of the model towards diverse training samples of all the three benchmark datasets, 60%, 40% and 20% of labeled samples are randomly selected for training the model. Table 8 shows the Overall accuracy (OA) and Kappa coefficient (KA) for all the above mentioned training samples percentages. The experiment on all the three benchmark datasets was carried out with patches of window size = 7,11 and 15. The outcome showed that the accuracy increases with increase in the window size as illustrated in the Table 9.

5 Conclusion

This paper attempts to increase the performance of HI classification by using GWO-based-CNN-HI framework. This framework consists of following steps, SWT has been used to extract the spatial and spectral features, PCA for spectral dimension reduction and retaining the most informative bands, CNN for classification, GWO to find the optimal hyper-parameters for the CNN for HI classification. The process of obtaining optimal hyper-parameters for CNN using trial and error method is a tedious task. To solve this issue we have utilized the GWO (Grey Wolf Optimizer) to find the optimal set of hyperparameters. The proposed model has achieved better performance in classification. The GWO-based-CNN-HI framework is validated using standard datasets and experimental results show that Optimized CNN model achieves an improved performance over the state of the art techniques.

The hyperspectral image classification is a very important technique for many applications. Since optimized model is an essential for achieving promising classification accuracy, this paper mainly focuses on finding optimal parameters of the model. The proposed GWO-based-CNN-HI framework can be further extended to other variants of CNNs. Though the proposed framework gives improved overall accuracy, there are misclassified pixels when compared to the ground truth. Some post-processing may be required to correct the misclassifications. The reported accuracy is also not guaranteed when the input image is contaminated with noise. Further study on this is required.

References

1. rnn vs mlp analyzing 3 types of neural networks. Retrieved April 20, 2020, from https://www.analyticsvidhya.com/blog/2020/02/cnn-vs\\discretionary-rnn-vs-mlp-analyzing-3-types-of-neural-networks-\in-deep-learning/
2. Antoniadis A, Oppenheim G (2013) Wavelets and statistics. lecture notes in statistics. Springer, New York, NY

3. Beirami BA, Mokhtarzade M (2020) Band grouping superpca for feature extraction and extended morphological profile production from hyperspectral images. IEEE Geosci Remote Sens Lett 17:1953–1957. https://doi.org/10.1109/LGRS.2019.2958833
4. Back T (1996) Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, Inc., USA
5. Ben-Dor E, Patkin K, Banin A, Karnieli A (2002) Mapping of several soil properties using dais-7915 hyperspectral scanner data-a case study over clayey soils in Israel. Int J Remote Sens 23(6):1043–1062
6. Breiman L (2001) Machine learning, volume 45, number 1 - springerlink. Mach Learn 45:5–32. https://doi.org/10.1023/A:1010933404324
7. Chang C-I (2002) Target signature-constrained mixed pixel classification for hyperspectral imagery. IEEE Trans Geosci Remote Sens 40(5):1065–1081. https://doi.org/10.1109/TGRS.2002.1010894
8. Chen C (2015) Hyperspectral image classification using gradient local auto-correlations. pp 454–458
9. Chen Y, Zhao X, Jia X (2015) Spectral-spatial classification of hyperspectral data based on deep belief network. IEEE J Sel Top Appl Earth Obs Remote Sens 8(6):2381–2392. https://doi.org/10.1109/JSTARS.2015.2388577
10. Chen Y, Jiang H, Li C, Jia X, Ghamisi P (2016) Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. IEEE Trans Geosci Remote Sens 54(10):6232–6251
11. Coifman RR, Donoho DL (1995) Translation-invariant de-noising Volume 103 of Antoniadis.A. [2]. pp 125–150
12. Dasarathy BV (1991) Nearest neighbor (nn) norms: Nn pattern classification techniques. IEEE Computer Society Tutorial
13. del Valle Y, Venayagamoorthy GK, Mohagheghi S, Hernandez J, Harley RG (2008) Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Trans Evol Comput 12(2):171–195
14. Dicker DT, Lerner J, Van Belle P, Guerry D, Herlyn M, Elder DE, El-Deiry WS (2006) Differentiation of normal skin and melanoma using high resolution hyperspectral imaging. Cancer Biol Ther 5(8):1033–1038
15. Ding X, Li H, Yang J, Dale P, Chen X, Jiang C, Zhang S (2020) An improved ant colony algorithm for optimized band selection of hyperspectral remotely sensed imagery. IEEE Access 8:25789–25799. https://doi.org/10.1109/ACCESS.2020.2971327
16. ESA European space agency (esa). Retrieved April 20, 2020, from http://www.esa.int/SPECIALS/Eduspace_EN/SEMPNQ3Z2OF_0.html
17. Feng Y-Z, Sun D-W (2012) Application of hyperspectral imaging in food safety inspection and control: a review. Crit Rev Food Sci Nutr 52(11):1039–1058
18. Fu Y, Jianzhong X, Kumchol Y (2019) Dynamically dimensioned search grey wolf optimizer based on positional interaction information, Hindawi
19. Geem ZW (2009) Music-inspired harmony search algorithm: Theory and applications, vol 191
20. Ghamisi P, Yokoya N, Li J, Liao W, Liu S, Plaza J, Rasti B, Plaza A (2017) Advances in hyperspectral image and signal processing: a comprehensive overview of the state of the art. IEEE Geosci Remote Sens Mag 5(4):37–78. https://doi.org/10.1109/MGRS.2017.2762087
21. Gharaati E, Nasri M (2015) A new band selection method for hyperspectral images based on constrained optimization
22. Gonzalez R, Woods R (1993) Digital Image Processing. Reading, Massachusetts, Addison-Wesley Publishing Company
23. Gowen AA, O'Donnell CP, Cullen PJ, Downey G, Frias JM (2007) Hyperspectral imaging–an emerging process analytical tool for food quality and safety control. Trends Food Sci Technol 18(12):590–598
24. Gregory L, Ralf G, Filip W, Kai W, Aaron (2019) Pywavelets: a python package for wavelet analysis. J Open Sour Softw 4:1237. https://doi.org/10.21105/joss.01237
25. Hughes G (1968) On the mean accuracy of statistical pattern recognizers. IEEE Trans Inf Theory 14(1):55–63. https://doi.org/10.1109/TIT.1968.1054102
26. Izonin I, Tkachenko R, Kryvinska N, Greguš M (2019) Multiple linear regression based on coefficients identification using non-iterative sgtm neural-like structure
27. Jia S, Deng B, Zhu J, Jia X, Li Q (2018) Local binary pattern-based hyperspectral image classification with superpixel guidance. IEEE Trans Geosci Remote Sens 56(2):749–759. https://doi.org/10.1109/TGRS.2017.2754511
28. Jonnadula H, Kumar LS, Panda GK, Dash R, Kumar LP (2020) Hyperspectral image classification bi-dimensional empirical mode decomposition and deep residual networks. In: 2020 International Conference on Artificial Intelligence and Signal Processing (AISP), pp 1–6

29. Kaiming H, Jian S (2015) Convolutional neural networks at constrained time cost. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5353–5360
30. Ladi SK, Dash R, Panda GK, Ladi PK, Dhupar R (2019) Hyperspectral image classification using swt and cnn. In: 2019 International Conference on Information Technology (ICIT), pp 172–177
31. Letalick D, Renhorn I, Steinvall O (2009) Multi-optical mine detection system (moms) final report
32. Li R, Pan Z, Wang Y, Wang P (2020) A convolutional neural network with mapping layers for hyperspectral image classification. IEEE Trans Geosci Remote Sens 58(5):3136–3147. https://doi.org/10.1109/TGRS.2019.2948865
33. Li T, Zhang J, Zhang Y (2014) Classification of hyperspectral image based on deep belief networks. In: 2014 IEEE International Conference on Image Processing (ICIP), pp 5132–5136
34. Liao W, Ochoa D, Gao L, Zhang B, Philips W (2019) Morphological analysis for banana disease detection in close range hyperspectral remote sensing images. In: IGARSS 2019 - 2019 IEEE international geoscience and remote sensing symposium, pp 3697–3700
35. Lillesand TMM, Kiefer R (2000) Remote sensing and image interpretation
36. Liu B, Yu X, Zhang P, Yu A, Fu Q, Wei X (2018) Supervised deep feature extraction for hyperspectral image classification. IEEE Trans Geosci Remote Sens 56(4):1909–1921. https://doi.org/10.1109/TGRS.2017.2769673
37. Mercier G, Lennon M (2003) Support vector machines for hyperspectral image classification with spectral-based kernels. In: Igarss 2003. 2003 ieee international geoscience and remote sensing symposium. proceedings (ieee cat. no.03ch37477), vol 1, pp 288–290 vol. 1
38. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
39. Mou L, Ghamisi P, Zhu XX (2017) Deep recurrent neural networks for hyperspectral image classification. IEEE Trans Geosci Remote Sens 55(7):3639–3655
40. Mountrakis G, Im J, Ogole C (2011) Support vector machines in remote sensing: a review. ISPRS J Photogramm Remote Sens 66:247–259. https://doi.org/10.1016/j.isprsjprs.2010.11.001
41. Nakamura R, Fonseca L, dos Santos J, Yang X-S, Papa J, Torres R (2013) Nature-inspired framework for hyperspectral band selection, vol 1
42. Nason GP, Silverman BW (1995) The stationary wavelet transform and some statistical applications. Volume 103 of Antoniadis.A. [2]
43. Niu P, Niu S, liu N, Chang L (2019) The defect of the grey wolf optimization algorithm and its verification method, vol 171
44. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) Gsa: a gravitational search algorithm. Inf Sci 179:2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
45. Rathod M, Khanapuri J (2017) A comparative study of transform domain methods for image resolution enhancement of satellite image
46. Rodarmel C, Shan J (2002) Principal component analysis for hyperspectral image classification, vol 62
47. Romay D Hyperspectral remote sensing scenes. Retrieved March 1, 2020, from http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes
48. Shaheen F, Verma B, Asafuddoula M (2016) Impact of automatic feature extraction in deep learning architecture. In: 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp 1–8
49. Smola A, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14:199–222
50. Su Y, Li J, Plaza A, Marinoni A, Gamba P, Chakravortty S (2019) Daen: Deep autoencoder networks for hyperspectral unmixing. IEEE Trans Geosci Remote Sens 57(7):4309–4321. https://doi.org/10.1109/TGRS.2018.2890633
51. Tao C, Pan H, Li Y, Zou Z (2015) Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyprspectral imagery classification, vol 12
52. Tarabalka Y, Benediktsson JA, Chanussot J (2009) Spectral-spatial classification of hyperspectral imagery based on partitional clustering techniques. IEEE Trans Geosci Remote Sens 47(8):2973–2987. https://doi.org/10.1109/TGRS.2009.2016214
53. Tarabalka Y, Benediktsson JA, Chanussot J, Tilton JC (2010) A multiple classifier approach for spectral-spatial classification of hyperspectral data. In: 2010 IEEE International Geoscience and Remote Sensing Symposium, pp 1410–1413
54. Tarek E, Mohamed H (2015) Analysis of pca algorithms in distributed environments. arXiv preprint arXiv:1503.05214
55. Thomas AM, Cathcart JM (2010) Applications of grid pattern matching to the detection of buried landmines. IEEE Trans Geosci Remote Sens 48(9):3465–3470
56. Ting-ting Z, Fei L (2012) Application of hyperspectral remote sensing in mineral identification and mapping. In: Proceedings of 2012 2nd international conference on computer science and network technology, pp 103–106

57. Tkachenko R, Doroshenko A, Izonin I, Tsymbal Y, Havrysh B (2019) Imbalance data classification via neural-like structures of geometric transformations model: Local and global approaches
58. Tkachenko R, Izonin I (2019) Model and principles for the implementation of neural-like structures based on geometric data transformations
59. Tkachenko R, Izonin I, Tsymbal Y (2018) Learning-based image scaling using neural-like structure of geometric transformation paradigm. pp 537–565
60. Vaddi R, Manoharan P (June 2020) Hyperspectral image classification using CNN with spectral and spatial features integration. Infrared Physics and Technology 107:103296. https://doi.org/10.1016/j.infrared.2020.103296
61. Vrbančič G, Brezočnik L, Mlakar U, Fister D, Fister Jr. I (2018) NiaPy: Python microframework for building nature-inspired algorithms. Journal of Open Source Software 3
62. Wang Y, Cui S (2014) Hyperspectral image feature classification using stationary wavelet transform. In: 2014 International conference on wavelet analysis and pattern recognition, pp 104–108
63. Wirsing K (2020) Time frequency analysis of wavelet and fourier transform. In: Wavelet theory. IntechOpen
64. Wu J, Chen X-Y, Zhang H, Xiong L-D, Lei H, Deng S-H (2019) Hyperparameter optimization for machine learning models based on bayesian optimization. J Electron Sci Technol 17:26–40. https://doi.org/10.11989/JEST.1674-862X.80904120
65. Xia H, Zhu F, Li H, Song S, Mou X (2020) Combination of multi-scale and residual learning in deep cnn for image denoising. IET Image Process 14(10):2013–2019
66. Xu Y, Du B, Zhang F, Zhang L (2018) Hyperspectral image classification via a random patches network. Isprs J Photogramm Remote Sens 142:344–357
67. Xu Y, Zhang L, Du B, Zhang F (2018) Spectral-spatial unified networks for hyperspectral image classification. IEEE Trans Geosci Remote Sens 56(10):5893–5909
68. Yang H, Du Q, Chen G (2012) Particle swarm optimization-based hyperspectral dimensionality reduction for urban land cover classification. IEEE J Sel Top Appl Earth Obs Remote Sens 5(2):544–554
69. Yang X-S (2010) A new metaheuristic bat-inspired algorithm. 284, pp 65–74
70. Yang X-S (2013) Multiobjective firefly algorithm for continuous optimization. Eng Comput. 29:1–10
71. Yang X-S (2020) Nature-inspired optimization algorithms: challenges and open problems. J Comput Sci 46:101104. https://doi.org/https://doi.org/10.1016/j.jocs.2020.101104. https://www.sciencedirect.com/science/article/pii/S1877750320300144 20 years of computational science
72. Ying L, Haokui Z, Qiang S (2017) Spectral–spatial classification of hyperspectral imagery with 3d convolutional neural network. Remote Sens 9(1):67
73. Yu C, Han R, Song M, Liu C, Chang CI (2020) A simplified 2d-3d cnn architecture for hyperspectral image classification based on spatial-spectral fusion. IEEE J Sel Top Appl Earth Obs Remote Sens 13:2485–2501. https://doi.org/10.1109/JSTARS.2020.2983224
74. Yu S, Jia S, Xu C (2017) Convolutional neural networks for hyperspectral image classification. Neurocomputing 219:88–98. https://doi.org/https://doi.org/10.1016/j.neucom.2016.09.010. https://www.sciencedirect.com/science/article/pii/S0925231216310104
75. Zhang L, Zhang L, Du B (2016) Deep learning for remote sensing data: a technical tutorial on the state of the art. IEEE Geosci Remote Sens Mag 4:22–40. https://doi.org/10.1109/MGRS.2016.2540798
76. Zhao W, Du S (2016) Spectral-spatial feature extraction for hyperspectral image classification: a dimension reduction and deep learning approach. IEEE Trans Geosci Remote Sens 54(8):4544–4554. https://doi.org/10.1109/TGRS.2016.2543748
77. Zhong Z, Li J, Luo Z, Chapman M (2018) Spectral-spatial residual network for hyperspectral image classification: A 3-d deep learning framework. IEEE Trans Geosci Remote Sens 56(2):847–858. https://doi.org/10.1109/TGRS.2017.2755542
78. Zhou F, Hang R, Liu Q, Yuan X (2018) Hyperspectral image classification using spectral-spatial lstms. Neurocomputing, 328. https://doi.org/10.1016/j.neucom.2018.02.105
79. Zortea M, Haertel V, Clarke R (2007) Feature extraction in remote sensing high-dimensional image data. IEEE Geosci Remote Sens Lett 4(1):107–111. https://doi.org/10.1109/LGRS.2006.886429

Affiliations

Sandeep Kumar Ladi[1,2] · G. K. Panda[3] · Ratnakar Dash[4] · Pradeep Kumar Ladi[5] · Rohan Dhupar[6]

Pradeep Kumar Ladi
ladi.pradeep@giet.edu

G. K. Panda
drgkpmail@gmail.com

Ratnakar Dash
ratnakar.dash@gmail.com

Rohan Dhupar
rohandhupar18@gmail.com

1 GITAM Deemed to be University Visakhapatnam, Visakhapatnam, Andhra Pradesh, India

2 Department of ECE, Biju Patnaik University of Technology (BPUT), Chhend Colony, Rourkela, Odisha 769015, India

3 MITS Bhubaneswar, Bhubaneswar, Odisha, India

4 NIT Rourkela, Rourkela, Odisha, India

5 GIET University Gunupur, Gunupur, Odisha, India

6 Bharti Airtel, Gurugram, Haryana, India