Earth Science Informatics (2024) 17:593–616
https://doi.org/10.1007/s12145-023-01109-1

RESEARCH

A novel multi-class land use/land cover classification using deep kernel attention transformer for hyperspectral images

Ganji Tejasree[1] · Agilandeeswari L[1]

Received: 10 July 2023 / Accepted: 15 September 2023 / Published online: 18 December 2023

Abstract
Hyperspectral imaging is a prominent land use land cover (LULC)classification technology. However, due to fewer training samples, LULC classification using hyperspectral images remains complicated and labour-intensive. We have presented a Deep Kernel Attention Transformer (DKAT) to overcome these issues in classifying Land Use Land Cover classes. Before classifying the land cover, t-Distributed Stochastic Neighbouring Embedding (t-SNE) is exploited to extract the features from the LULC by applying the probability distribution function. To quantify the resemblance among the two points Kull Burk-Divergence (KL) is employed. Then, a searching-based band selection method is used to select the bands. The grey wolf optimization (GWO) technique is used in the searching-based band selection method to determine the informative bands. After choosing the informative bands from the hyperspectral data cube, we must classify the land cover. Experimental results are conducted by using five publicly available benchmark datasets. They are Indian Pines, Salinas, Pavia University, Botswana, and Kennedy Space Center. The classification accuracy is calculated using the overall accuracy, average accuracy, and kappa coefficient; we have achieved 99.19% overall accuracy, 99.32% average accuracy, and 99.14% kappa coefficient.

Keywords Hyperspectral images · Land use/land cover · Deep kernel attention transformer · t-distributed stochastic neighboring embedding · Grey wolf optimization

Introduction

As long as the significance of anthropogenic activities, variations in LULC are inconsistent. The abovementioned variations are like financial objectives, such as wood exploration, agriculture, and cattle ranching (Christovam et al. 2019). In particular, transformations in LULC will happen because of deforestation. Furthermore, certain expectations concerning deterioration as well as submergence circumstances will increase the LULC changes, and these problems raise an increase in greenhouse gas secretion and loss of biodiversity (Mas 1999; Mangan et al. 2022). Therefore, the data that the LULC has plays a vital role in climate and environmental change studies. Thus, the universal concern is to use all the requirements to get superior LULC maps. These maps will provide data for arranging and assessing natural resource management, developing sustainable practices, and modelling environmental variables (Adam et al. 2014). The images that are most prevalently used for LULC change classification and detection are Multispectral (MohanRajan and Loganathan 2021; 2022; 2023; MohanRajan et al. 2020), and Hyperspectral images (Navin and Agilandeeswari 2020a, b; Agilandeeswari et al. 2022). The main issues in the HSI over the past ten years have been spectral dimensionality and the requirement for particular spectral-spatial classifiers (Yadav et al. 2022).

Rapid, extensive changes in land cover are currently occurring in several areas of the world. Several of these nations, including Brazil, Columbia, Indonesia, Mexico, the Ivory Coast, Venezuela, and Zaire, are the focal points of most of this activity (Mas 1999). Due to the possible consequences of erosion, increased run-off and flooding, rising CO2 concentration, climatological shifts, and biodiversity loss, these changes in land cover, especially the destruction of tropical forests, have drawn attention (Fontan 1994). To map large areas and get multitemporal information from the large covered areas is a time-consuming and expensive task (Kavzoglu and Colkesen 2009). To overcome these issues, the satellite imagery technique is the most useful (Lv and Wang 2020). Since

✉ Agilandeeswari L
agila.l@vit.ac.in

Ganji Tejasree
ganji.tejasree2020@vitstudent.ac.in

[1] School of Information Technology and Engineering, Vellore Institute of Technology, Vellore 632014, Tamil Nadu, India

1970, remote sensing images have contributed to persistent as well as legitimate knowledge of the land surface area (Petitjean et al. 2012). These satellite images have the proficiency to grab the entire land (Puletti et al. 2016). Some of the satellite hyperspectral sensors are AVIRIS, ROSIS, Hyperion, and MODIS (Haq et al. 2021a; Haq et al. 2020; Haq et al. 2021b). The captured images extract the LULC information using image processing techniques. An aforementioned satellite community has emerged the improvements in image classification approaches throughout the period of time to map the LULC (Haq 2022a). Through digital image processing, there is an extent for mapping the LULC. However, as a result of numerous factors corresponding to the procedure, building the LULC maps using image processing techniques is a challenging task (Manandhar et al. 2009).

Hyperspectral remote sensing imagery emerged a few decades ago and can gently enhance LULC mapping. A large number of uninterrupted spectral bands, along with diminutive bandwidths could be accumulated for hyperspectral imaging (Bioucas-Dias et al. 2013). These images will give more details about spectral bandwidths collected by multispectral sensors. And it will provide new opportunities for LULC applications (Chutia et al. 2016). And also, it will give a large quantity of information that leads to the Hughes phenomenon. To help the difference between classes, the amount of predictor features may add data. Also, it will raise the complexity. A small dataset sample size may not be sufficient to characterize this complexity. To decrease the complexity, we have to add more features rather than increase the dimensions of the data (Maxwell et al. 2018). As a consequence of the confined proportion of training data specimens, in addition to the nonessential details exhibited across all features, hyperspectral image classification (HSI) is a difficult task. And also, the uncertainties accompanying the atmospheric and geographic effects present in the spatial resolution. Some machine learning algorithms like supervised, unsupervised, and semi-supervised classification methods are available to get high LULC classification accuracy using hyperspectral images (Haq et al. 2021c). Supervised classification will be used when the user has labeled input samples. Some supervised classification algorithms are random forest (RF) and support vector machine (SVM) (Ghamisi et al. 2017; Haq et al. 2022). Unsupervised classification is used when the user has unlabelled samples; the training model will take the labels from them. Some unsupervised classification algorithms are k-nearest neighbors (Cariou et al. 2020). Semi-supervised classification will be used when the user has more labeled and fewer unlabelled samples. Some of the semi-supervised classification algorithms are self-organizing algorithms. Each algorithm will give the best results based on the datasets we have used (Ma et al. 2016).

In recent years, for hyperspectral image classification (HSIC) use of deep learning-based algorithms has been increasing, and they are achieving good results in LULC classification (Haq 2022b). To get high LULC classification accuracy, we must perform feature extraction and band selection before classification. This helps us remove the redundancy among the features, avoids the Hughes phenomenon, and decreases the computational cost. When we have high-dimensional input features in the dataset, we have to extract the features from that data. Here are some of the feature extraction methods, PCA (principal component analysis) (Hotelling 1933), autoencoders (AE), LDA (linear discriminant analysis), and t-SNE (stochastic neighboring embedding). PCA will be used to find the subspace of principal components from the input feature vectors (Fauvel et al. 2009). To distinguish the low-dimensional hyperspace based on miscellaneous classes LDA is used (Chang and Ren 2000). LDA and PCA are known as global linear algorithms, that could never perform efficiently in nonlinear scattered data circumstances (Kambhatla and Leen 1993). As a consequence of these arguments, a couple of researchers have suggested nonlinear feature extraction algorithms for hyperspectral images (Zhang et al. 2008). They are ISOMAP (isometric mapping), LLE (local linear embedding) (Bachmann et al. 2005), and spherical stochastic neighboring embedding (SSNE) (Lunga and Ersoy 2012). Particularly, the above-mentioned nonlinear feature extraction approaches to handle an individual feature as input, i.e., spectral features (Segl et al. 2003). A multiple-feature extraction technique to subordinate a probability-preserving projection structure to get more features from the data, i.e., t-SNE. A probability distribution is fabricated based on t-SNE for each feature (Devassy and George 2020).

To perform the non-destructive diagnosis to get significant information from the different bands in hyperspectral images, t-SNE is used. a t-SNE-based dimensionality reduction method to analyze the ink. Nonlinear equivalence features among the spectra are used to extract the features and measure them through a lower dimension t-SNE. To validate ink spectral information graphically and measurably, t-SNE is giving good results compared to other feature extraction methods (Devassy and George 2020). A modified stochastic neighbour embedding (MNSE) in favour of numerous feature dimensionality reduction. This will build a probability distribution function based on t-SNE for the respective feature. Compared to additional ongoing dimensionality reduction techniques, the intended approach has been utilizing a hyperspectral image's spectral and spatial features. By using MNSE the hyperspectral image classification accuracy is also increased (Zhang et al. 2013).

Several studies have been conducted on the band selection strategy algorithms; a Minimum Noise band selection technique is suggested in (Bajwa et al. 2004). Depending on the characteristics of each band, the minimum noise method will work, that bands are represented by the high signal-to-noise ratio (SNR) and low correlation. A progressive band selection method has been proposed in (Ettabaa and Salem 2018), which differs from all the traditional band selection methods. This is going to be approving the more than a few picked bands; out of that, distinct endmembers are exploited for spectral unmixing. Researchers are focusing on optimization-based band selection methods to improve classification accuracy. For band selection global optimization algorithms are used; they are the GSA (gravity search algorithm) (Zhang et al. 2019), GA (genetic algorithms) (Saqui et al. 2016), PSO (particle swarm optimization) (Zhang et al. 2017), and FA (firefly algorithm) (Su et al. 2015). Due to low convergence speed, the gravity search algorithm does not satisfy the global search band selection strategy. Genetic algorithms have a more significant number of parameters; owing to that, the execution is complex; there is a possibility to fall into local optimum readily. This is also not giving good results for global search band selection. The band discovery rate is low when firefly algorithms are used. The accuracy rate is low, and the convergence speed is also very low. To overwhelm these issues, a contemporary heuristic algorithm is suggested. That is Grey Wolf Optimization (GWO), which was introduced in 2014. GWO has fewer modifications in parameters, high execution, and rapid convergence compared to other optimization methods, but it still has a few flaws. For example, while solving multimodal functions, the convergence speed is slow, as well as that is uncomplicated to descend victim to the drawbacks of local extremes (Wang et al. 2022). Because of this action, we have used the global search-based grey wolf optimization (GWO) band selection method to select the bands because GWO gives very few computational results compared with other existing optimization methods.

Semi-supervised band selection using an optimal graph (BSOG) is proposed in (Teng et al. 2022). It will perform the band selection as well as it will learn the local structuring bands. This intellectual similarity matrix is accommodative in contradiction to demonstrating the input similarity matrix to find out greater local configuration. A particular superlative band subset might be picked by assessing the gained projection matrix W. The proposed method is giving better results than the other existing methods. A modified Grey wolf optimization (MGWO) to select the number of bands from a scene is proposed in (Wang et al. 2021). The operations are performed based on the index of grey wolves. The intended technique is evaluating the variation of a particular band and calculating the worst fitness function. The suggested technique accomplished higher results for hyperspectral image classification than the other one. A band selection approach based on a modified cuckoo search optimization surrounded by correlations following the initialization procedure is demonstrated in (Sawant et al. 2019). The metaheuristic cuckoo search algorithm might drop into the local optimum solution. At the same time, they have initiated a strategy under interconnections to avoid the decline toward the local optimum solution.

Hyperspectral imaging is used to capture images; these images have rich spectral and spatial information. Spectral and spatial information is observed from the earth, and the generated image will be stored in the hyperspectral data cube. This data cube is 3D, two sides of the cube have spectral information, and one side has spatial information. The procedure of hyperspectral image classification is to establish the land-cover class of individual hyperspectral pixels, which are presented in hyperspectral images. As a consequence of the absence of accessibility of hyperspectral datasets that are available in public and the large size of the land cover classes, hyperspectral image classification is difficult. Since 2012, researchers have been focusing more on deep learning methods for LULC classification, and the classification results are promising (Otter et al. 2020). The deep learning techniques are enumerated towards spectra-spatial-based and spectral-based approaches. Spectral-based approaches are familiar with the spectral signatures of a single pixel for hyperspectral images. The spectral-spatial-based methods are used to explore the adjacent hyperspectral pixels of hyperspectral image classification. The convolutional neural network (CNN) as well as the fortunate deep learning algorithm are the most prevalent (Haq et al. 2023; Haq 2022c). Because it will use the hidden layers to extract the deep features for LULC classification (Lee and Kwon 2017). First, for HSI classification CNN is turned out to be used through various hidden layers. Then, to classify HSIs directly from their spectral domain, a deep convolutional neural network (deep-CNN) is employed. To withdraw the spectral features from hyperspectral images, 1D-CNN is exploited. But it involves input in a one-dimensional vector. 2D-CNN is also introduced for both spectral and spatial hyperspectral image classification, this combines the spectral and spatial features to give better results for classification. However, it is missing some information from spectral features (Roy et al. 2019).

Land use/ land cover (LULC) data obtaining is an essential stage because the source of information is utilized to receive the environmental variables, to improve the high-quality LULC maps LULC classification is used. Hyperspectral images have several issues: holding abundant spectral data, high dimensionality of information, and a smaller amount of training instances. Because of these issues, LULC classification is difficult. To overcome these issues, (Christovam et al. 2019) proposed supervised classification methods. They are random forest (RF), support vector machines (SVM), and spectral angular mapping (SAM). RF and SVM are determined along with 176 surface reflectance bands. PCA is used for the dimensionality reduction, and for classification, SVM and RF were used. The hyperspectral image classification has achieved good results using the SAM, SVM, and RF. To minimize the spectral shift produced by the adjacency factor, a correlation coefficient-weighted spatial filtering operation technique is suggested (Yang et al. 2022). To introduce the operation into the kernel collaborative representation method with Tikhonov regularisation (KCRT), the weighted spatial, spectral KCRT method is used to construct the land cover classes. The main problem of this proposed method is to label a pixel in hyperspectral images due to small-sized labeled samples.

An attention mechanism was also introduced to get high classification accuracy. Since CNN struggled to represent long-distance dependencies to gather global context data, the bulk of attention techniques for HSI classification now in use is based on the convolution layer. As a result, classification accuracy might be improved. Adding the channel attention mechanism to a squeeze-sand-excitation network (SENet) enhanced the classification accuracy (Hu et al. 2018). To enhance the feature maps using a squeeze operator and an excitation operation, a spatial-spectral squeeze-and-excitation network (SSSE) has been proposed. Additionally, including the attention mechanism is a well-known model, that can significantly enhance categorization performance. To compute the attention map, the sigmoid function, as well as the tanh function, was used (Wang et al. 2019). An extreme learning machine (ELM) ensemble method was proposed in (Su et al. 2017), to achieve good accuracy. By using correlation analysis, they have divided the spectral bands into Multiview. They have used random rotation to view the multiple coordinate spaces. Then, ensembled strategy pruning is designed for low complementary and, consequently, classification results.

Very recently, a model called transformers was introduced for hyperspectral classification. Transformers are developed for natural language processing (Mishra et al. 2023). Transformers will be working based on the self-attention mechanism (He et al. 2021). Transformers will be used attention to design a global dependency across a concatenation by time of input. To transform the input data from one sequence to another, a self-attention encoder and decoder are used. These decoder and encoder sequences are known as “tokens”, in the model, the tokens are represented as feature vectors in the primary data. The transformers are also used to extract the features from sequence data. Transformers will give the best weights to the initial data by using a multi-layer perceptron (MP). The transformer’s mapping considers it when a piece of exhaustive information is implemented to an image data. And it leads to a sudden intensification in the model’s size and a significant computational and training overhead. Therefore, image feature extraction is limited in transformers. To classify an image, we have various transformer models: vision transformers (ViT) (Palani and Loganathan 2023a, b; Aberna et al. 2023), Swin transformers (SwinT) (Agilandeeswari and Meena 2023), DeiT, and so on. The tokens are fixed scale, vision transformer will take all the tokens in a fixed size scale; because of this ViT is unstable for the vision applications. Swin transformers are also used for LULC classification, but their computational complexity will depend on the image size. And also, it is used to enable the development of hierarchical feature maps (Zhang et al. 2022). A multiscale convolutional embedding module with transformers for hyperspectral images is proposed in (Jia and Wang et al. 2022). To make use of the unlabelled samples, a self-supervised pre-training task is also developed. For the mask autoencoder, the proposed pre-training technique addresses the masks only on equivalent tokens away from the central pixel in the encoder.

The contribution of this paper is to classify the Land Use/ Land Cover classification using hyperspectral images with a deep kernel attention transformer algorithm to get high classified accuracy.

The remaining paper is arranged like this, The proposed method is elaborated in Section II, the dataset description is given in Section III, evaluation metrics are explained in Section IV, and Section V demonstrates the experimental results followed by limitations of the proposed method and future scope. Finally, the conclusions are drawn in Section VI.

Proposed method

There has been a relatively new attempt to apply deep learning approaches to classify the land use/land cover (LU/LC) using hyperspectral images. However, there are still some difficulties in this area due to limited labels. To overwhelm above mentioned difficulties, a deep kernel attention transformer (DKAT) is suggested to classify the land use/ land cover (LULC) classification using hyperspectral images. First, we’re extracting the features from the dataset using t-distributed stochastic neighboring embedding (t-SNE); this will apply the probability function to get the features. To pick the informative bands from the extracted features, a grey wolf optimizer (GWO) is applied. Then we apply the classification method on the selected bands, i.e., deep kernel attention transformer (DKAT), to get the accurate classified results. Figure 1 shows the proposed framework of the DKAT-based land use/land cover (LULC) classification. In satellite image classification approaches, the single-feature extractors will recognize the majority of selective features. Without taking into consideration, the spatial features of an adjacent pixel, single-feature extractors want to investigate respective pixels one at a time through their spectral features. To obtain efficient data across all the pixels, extracting complicated characteristics based on each spectral and spatial region is crucial in LULC classification using hyperspectral images. Here are the reasons to use the t-SNE in this work; the uppermost significant point is that t-SNE holds the ability to maintain both global as well as local forms of the data. An additional purpose of t-SNE is that it will be used to designate the probability equivalence in connection with the high- to low-dimensional feature interval. Eventually, t-SNE presents good results for both nonlinear as well as linear data. The reason for selecting the GWO, it is a global optimization algorithm that can find the best possible solution for the band selection problem, even if it’s non-convex. GWO is indeed an efficient algorithm. It can find the global optimal solution in a reasonable amount of time, which makes it a great tool for solving complex problems. GWO’s robustness is one of its greatest strengths, and it makes it suitable for a wide range of applications, including band selection problems where the initial conditions can be tricky to

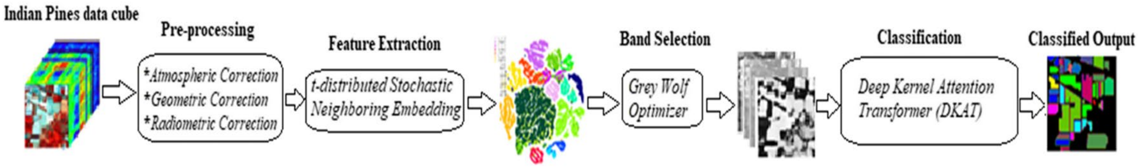


Fig. 1 The proposed architecture of DKAT-based classification

determine. With GWO, we can have confidence that we will get accurate results regardless of the initial conditions. The reason for selecting the Deep Kernel Attention Transformer is that it is a recently proposed classifier model for hyperspectral images. Learning the long-range dependencies between pixels is critical for hyperspectral images since the spectral bands contain information about different parts of the image. To overcome these flaws, the DKAT model is proposed, which can take the features from a kernel attention mechanism to capture the dependencies. DKAT is an excessive model for hyperspectral image classification because it is robust to noise. Even though hyperspectral images can be noisy, DKAT can ignore the noise and focus on the important features of the image to achieve high accuracy. Its kernel attention mechanism is particularly useful for capturing spectral band information, making it a promising classifier for improving hyperspectral image classification.

t-SNE (t- distributed stochastic neighbouring embedding)

Vander Maaten and Hinton developed the t-SNE algorithm in 2008. This algorithm can convert the higher dimensional values into lower dimensions. The elaborated divergence of traditional SNE (stochastic neighbouring embedding) is t-SNE, it is designed to meet the needs of a single-feature nonlinear dimension diminution. This is going to work in familiar with the standardization about the halved expanses as long as a combined probability dissemination beyond the contribution of representative pairs. High-dimensional intervals among data points towards the conditional probabilities will be transformed initially by SNE. In such a way that it will portray the common features. The resemblance of X_k to measurement point X_l is announced through the conditional probability $P_{k/l}$. And also, the probability of X_l will be identified as employing X_k as its neighbour, if the neighbours are incorporated in the same way to their probability density subject to the Gaussian centered at X_k. It is represented by

$$P_{k/l} = \frac{exp(-||X_k - X_l||^2/2\sigma^2)}{\sum_{k\neq l}\exp(-||-X_k - X_l||^2/2\sigma_i^2)} \tag{1}$$

Here, the variance of the Gaussian is represented as σ_i, it is concentrated explicitly on datapointX_k. Additionally, the t-SNE algorithm acknowledges the independent variables as the “perplexity”. It could be ascertained as an advance concerning more than a few efficient neighbours. Perplexity is determined scientifically as in the following equation

$$perp(P_i) = 2^{H(p_i)} \tag{2}$$

Here, $H(p_i)$ is represented as the Shannon entropy, P_i is represented to calculate in bits.

$$H(p_i) = -\sum_k P_{k/l} log_2 P_{k/l} \tag{3}$$

This approach relies on the pairwise intervals among the points; it will spontaneously identify the variance of σ_i, it ensures that the adequate number of neighbours matches the perplexity input by the user. To prevent overcrowding, the student t-distribution is used by t-SNE. It utilizes just one degree of freedom. The allocation of the probability at a lower dimension is $q_{l/k}$, it can be mathematically defined in the below equation

$$q_{l/k} = \frac{(1 + ||y_l - y_k||^2)^{-1}}{\sum_{k\neq l}(1 + ||y_k - y_l||^2)^{-1}} \tag{4}$$

The lower dimensional equivalents y_k and y_l of the high-dimensional data points X_k andX_l. To calculate the conditional probability Eq. (1) is used. If the lower-dimensional equivalents of y_k and y_l are used to model the high-dimensional data points X_k andX_l, then the conditional probability of $P_{k/l}$ and $q_{k/l}$ is equal. If a mismatch has occurred between the $P_{k/l}$ and $q_{k/l}$, SNE will find the lower-dimension representations to avoid the mismatch. To evaluate the correspondence among two distributions, Kullback-Leiber (KL) divergence is used. To underestimate the measurement of comprehensive data points of KL divergence, SNE will be using the gradient descent method. As long as the KL divergence is not symmetric, an immense volume of mistakes has been eventuated. To underestimate the KL divergence among the conditional distributions of $P_{k/l}$ and$q_{k/l}$, a joint probability distribution P is a single KL divergence in the lower-dimensional space. A joint probability distribution of Q in the lower dimension is diminished. The cost function has been provided as

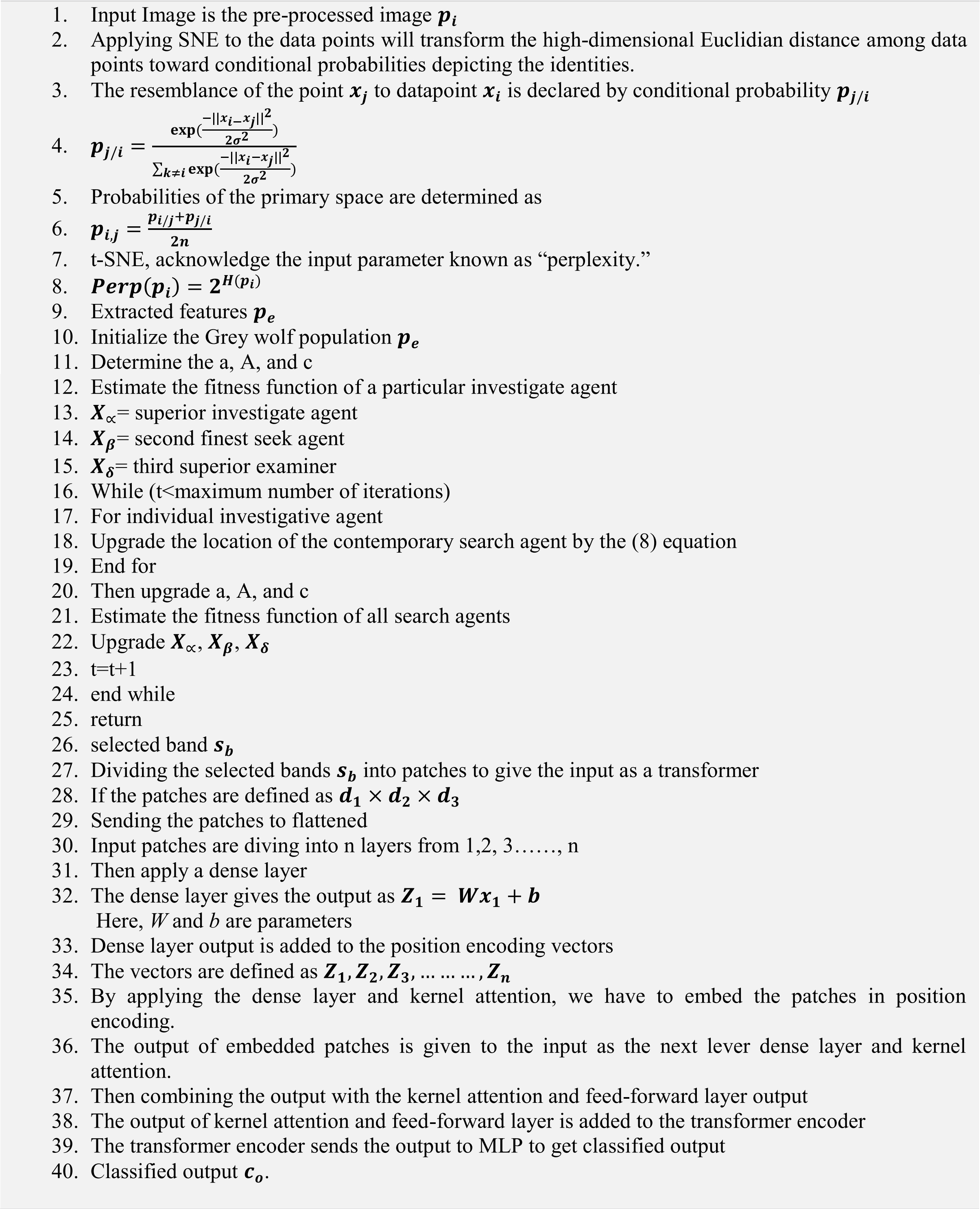
$$C = KL(P||Q) = \sum_l \sum_k p_{lk} \log \frac{p_{lk}}{q_{lk}} \tag{5}$$

where, p_{lk} is given by in Eq. (6) and q_{lk} can also be used the Eq. (4). Minimizing the cost function in Eq. (5) is now mentioned as a symmetric SNE.

The probabilities across the innovative distance are demonstrated in the following equation

$$P_{l/k} = \frac{(P_{l/k} + P_{k/l})}{2n} \quad (6)$$

Here, n represents the size of the dataset. Figure 2 illustrates the extracted features from the Pavia University and Salinas datasets.



```
Input Image is the pre-processed image $p_i$
Applying SNE to the data points will transform the high-dimensional Euclidian distance among data points toward conditional probabilities depicting the identities.
The resemblance of the point $x_j$ to datapoint $x_i$ is declared by conditional probability $p_{j/i}$
$p_{j/i} = \frac{\exp(\frac{-||x_i-x_j||^2}{2\sigma^2})}{\sum_{k\neq i}\exp(\frac{-||x_i-x_j||^2}{2\sigma^2})}$
Probabilities of the primary space are determined as
$p_{i,j} = \frac{p_{i/j}+p_{j/i}}{2n}$
t-SNE, acknowledge the input parameter known as "perplexity."
$Perp(p_i) = 2^{H(p_i)}$
Extracted features $p_e$
Initialize the Grey wolf population $p_e$
Determine the a, A, and c
Estimate the fitness function of a particular investigate agent
$X_\propto$= superior investigate agent
$X_\beta$= second finest seek agent
$X_\delta$= third superior examiner
While (t<maximum number of iterations)
    For individual investigative agent
        Upgrade the location of the contemporary search agent by the (8) equation
    End for
    Then upgrade a, A, and c
    Estimate the fitness function of all search agents
    Upgrade $X_\propto$, $X_\beta$, $X_\delta$
    t=t+1
end while
return
selected band $s_b$
Dividing the selected bands $s_b$ into patches to give the input as a transformer
If the patches are defined as $d_1 \times d_2 \times d_3$
Sending the patches to flattened
Input patches are diving into n layers from 1,2, 3……, n
Then apply a dense layer
The dense layer gives the output as $Z_1 = Wx_1 + b$
 Here, $W$ and $b$ are parameters
Dense layer output is added to the position encoding vectors
The vectors are defined as $Z_1, Z_2, Z_3, \dots\dots\dots, Z_n$
By applying the dense layer and kernel attention, we have to embed the patches in position encoding.
The output of embedded patches is given to the input as the next lever dense layer and kernel attention.
Then combining the output with the kernel attention and feed-forward layer output
The output of kernel attention and feed-forward layer is added to the transformer encoder
The transformer encoder sends the output to MLP to get classified output
Classified output $c_o$.
```

Algorithm for Hyperspectral image classification using DKAT

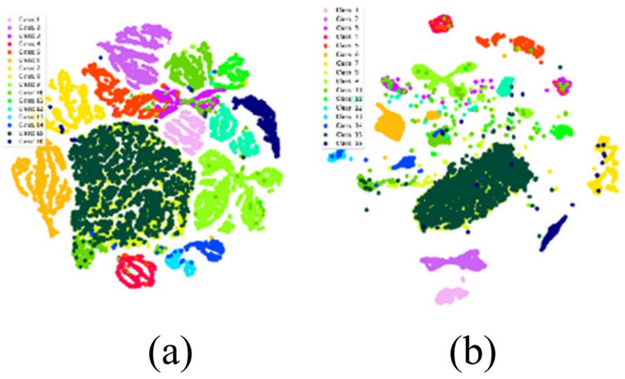


Fig. 2 **a** Pavia University, **b** Salinas datasets extracted features

Band selection using Grey Wolf Optimisation

Band selection is used to choose the informative bands from the hyperspectral data cube. By using the combinational optimization problem, band selection is formulated. Depending on the class separability evaluators and the classification accuracy in this paper, we have proposed a new fitness function. We have selected a new meta-heuristic method designated as a grey wolf optimization (GWO) to optimize that fitness function. Meta-heuristic means, a high-level independent algorithmic framework (developed for optimization algorithms). Mirjaliali Mohammed and Lewis developed grey wolf optimization in 2014 from Griffith University, Australia. This meta-heuristic algorithm has been used to solve many different fields. It is developed by taking the inspiration of nature and is also related to physical phenomena, evolutionary concepts, or animal behaviour. GWO is stimulated through the social hierarchy and hunting procedure of grey wolves. These grey wolfs are from the candidate family. Grey wolves are established in an immensely structured pack, and the volume of the pack is 5 to 12. Various ranks have been given to the pack of wolves, they are Alpha, Beta, Delta, and Omega wolves. Figure 3 represents (a) the Hierarchy of grey wolf organization, and (b) the grey wolf hunting process.

Alpha wolf The grey wolf family's captain is called the alpha wolf. A male alpha wolf and a female alpha wolf will represent the pack's leader. Other members will follow the instructions given by the Alpha. It is accountable for decision-making regarding sleeping places, hunting, time to wake up, and many others.

Beta wolf A beta wolf is represented as a secondary stage of the grey wolf hierarchy. The superior contender to lead the wolves is this wolf. The alpha wolf will benefit from it for both decision-making and other purposes. The alpha wolf will receive input from the beta wolf.

Delta wolf If the wolf is neither Alpha, Beta, nor Omega, it is Delta wolf. It dominates the omega wolf; to avoid the hazards, it will work for the pack. Delta wolf will deliver the food to the pack.

Omega wolf The lowest order of grey wolf is the Omega wolf. This will play the challenge of a scapegoat (a victim who is blamed for the mistakes or faults of others). Scouts, hunters, elders, and caretakers will come under this category. They are allowed to eat last, to avoid the internal conflicts and difficulties in the pack it will work.

Working principle of grey wolf optimization GWO algorithm impersonates grey wolves' leadership and hunting mechanism.

The primary stages for grey wolf hunting are:

1. Piercing for the prey
2. Chasing, tracking, and approaching the prey
3. Encircling, pursuing, and harassing the prey until it stops moving
4. Attacking the prey

Step 1: piercing for the prey: grey wolf targets the prey
Step 2: if the selected prey runs away, the pack will chase the prey to kill. Example: once the target is entered into the territory, i.e., wolf territory. A group of animals defines Wolf territory against another.

Chasing animal (i.e., the prey/ target) to other waiting wolves.

The following are the specifications for the mathematical model of the grey wolf optimization (GWO) social hierarchy:

The method of hunting is regarded as an optimization process,

- The optimum solution is determining the prey,
- Fitness solution as an alpha wolf (α)
- The second-greatest resolution is Beta wolf (β)
- The third finest interpretation is Delta wolf (δ), and
- Omega wolf (ω) follows these three wolves.

Mathematical model of encircling the prey Throughout the hunting, grey wolves encircle the prey. Encircling behaviour is modeled as:

$$\vec{D} = |\vec{C}.\vec{X_p} - \vec{X}(t)| \tag{7}$$

$$\vec{X}(t+1) = |\vec{X_p}(t) - \vec{A}.\vec{D}| \tag{8}$$

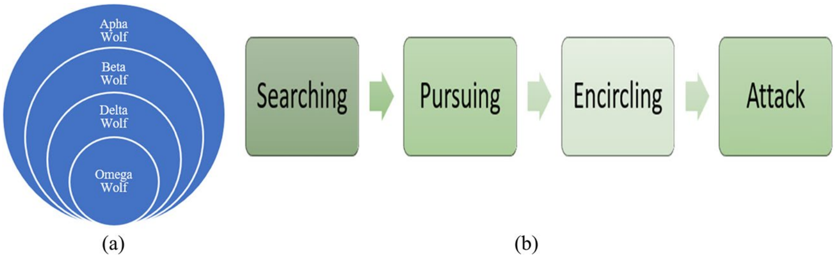


Fig. 3 **a** Hierarchy of Grey Wolfs, **b** Grey wolf hunting process

Here, the current iteration is represented as t, and the location of the prey is determined as $\overrightarrow{X_p}$, the position of the grey wolf is represented as $\vec{X}$, coefficient vectors are given as $\vec{A}, \vec{C}$, and $\vec{D}$ is the distance of the vector.

$\vec{A}, \vec{C}$ is calculated as:

$$\vec{A} = 2\vec{a}.\vec{r_1} - \vec{a} \tag{9}$$

$$\vec{C} = 2.\vec{r_2} \tag{10}$$

Here, the random vectors are represented as $\vec{r_1}, \vec{r_2}$, ranges from [0,1], component $\vec{a}$ is linearly decreased from 2 to 0 around reputations.

To improve the grey wolf location, the above equations are used. In accordance with the location of the grey wolf Eqs. (7) and (8) are various places approximately the best example of investigative agents might be able to concern about the contemporary location by regulating the standards of A & C. Equations (9), and (10) are used for calculating the vectors.

Step 3: grey wolf hunting: The grey wolf starts hunting by locating the prey. They will target the weak or elderly ones. Large animals like moose may stand on their ground and fight. Wolf may choose to try other prey rather than risk attacking large animals willing to fight.

Mathematical model of hunting the prey The hunting process is directed by Alpha; it is assumed that α, β, δ have enhanced understanding of the position of the prey (i.e., the optimum solution). Additional wolves will upgrade their positions according to the position of α, β, δ.

$$\overrightarrow{D_\alpha} = |C_1.\overrightarrow{X_\alpha} - \vec{X}(t)| \tag{11}$$

$$\overrightarrow{D_\beta} = |C_2.\overrightarrow{X_\beta} - \vec{X}(t)| \tag{12}$$

$$\overrightarrow{D_\delta} = |C_3.\overrightarrow{X_\delta} - \vec{X}(t)| \tag{13}$$

$$\overrightarrow{X_1} = |\overrightarrow{X_\alpha} - A_1.\overrightarrow{D_\alpha})| \tag{14}$$

$$\overrightarrow{X_2} = |\overrightarrow{X_\beta} - A_2.\overrightarrow{D_\beta})| \tag{15}$$

$$\overrightarrow{X_3} = |\overrightarrow{X_\delta} - A_3.\overrightarrow{D_\delta})| \tag{16}$$

Here, A_1, A_2, A_3 and C_1, C_2, C_3 are the coefficient vectors. $\overrightarrow{X_\alpha}, \overrightarrow{X_\beta}, \overrightarrow{X_\delta}$ are the positions of the vectors of α, β, δ.

The location of the grey wolf is upgraded by

$$\vec{X}(t+1) = \overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3} \Big/ 3 \tag{17}$$

Step 4: killing the prey: The alpha wolf will terminate the hunt by attacking the prey. Once the prey has been hunted, the alpha wolf will eat first.

A mathematical model for attacking the prey If the prey interrupts stimulating, the wolves should attack to complete the hunting. During the iteration, the model finishes by decreasing $\vec{a}$ from 2 to 0. As $\vec{a}$ decrease, $\vec{A}$ also decreases. $A < 1$ forces the wolf to attack the prey. |A|> 1 diverse from prey and locate greater prey. C vectors are random values; interval ranges from [0,2]. C prevents placing a few additional weights on the prey to make it complicated for the wolves to find it.

Fitness function The degree to which a particular design solution adheres to the stated objectives is summarised using a fitness function. It is a specific kind of objective function. In genetic programming and optimization problems, fitness functions are used to direct simulations toward the best possible design solutions. To select the bands using the grey wolf optimization, we are using the fitness function in three different ways; they are

i. Correlation coefficient Measure: It is used to find the correlation between the bands. The correlation coefficient measure between two neighboring bands is utilized as the decision criterion to determine whether the two bands are substantially correlated. The two bands must be merged if this correlation exceeds a particular threshold, at which point they are deemed redundant. The correlation is calculated by using the below formula

$$CorM(b_i, b_j) = \frac{\sum_{b_i,b_j}}{\sqrt{\sigma(b_i)\sigma(b_j)}} \tag{18}$$

Here, CorM (b_i, b_j) is the correlation coefficient between the b_i *and* b_j bands, $\sum_{b_i,b_j}$ is the covariance between the b_i *and* b_j bands, σ is the variance.

The correlation coefficient between the band varies from band $b_i = -1$ *and band* $b_j = +1$. If the correlation is close to + 1 and -1, it demonstrates the presence of a strong linear dependency between the two bands. And the two bands are supposed to be incorporated. Whereas 0 indicates no linear dependence.

ii. Distance Calculation: Distance calculation is used to calculate the distance between bands. Here, we consider the minimum distance between the band and to band. To calculate the distance between the bands and measure the class separability for selected bands, the Jeffries-Matusita (JM) distance is used. If we are considering the two classes i and j, then the JM distance between the *i* class and *j* class is provided in the following equation

$$J_{ij} = \sqrt{2(1 - e^{B_{ij}})} \tag{19}$$

Here, B_{ij} is the Bhattacharyya distance. It is defined as

$$B_{ij} = \frac{1}{8}(m_i - m_j)^T(\frac{\sum_i + \sum_j}{2})(m_i - m_j) + \frac{1}{2}In[\frac{|(\sum_i + \sum_j)/2|}{|\sum_i|^{1/2}|\sum_j|^{1/2}}] \tag{20}$$

Here, m_i, m_j are representing the mean vector of the classes and $\sum_i$, $\sum_j$ are representing the class covariance.

To decide which bands to use in a binary classification task, the J_{ij} distance is used. But for band assertions in a multi-class classification problem, we must discover the bands that give the average JM distance. The average distance is given in Eq. (21)

$$D(b_i, b_j) = \sum_{i=1}^{c} \sum_{j=1}^{c} p\left(\omega_i\right)p(\omega_j)J_{ij} \tag{21}$$

Here, J_{ij} is JM distance between the α, β, δ, and ω wolves, i and j are bands. p $\left(\omega_i\right)$ *and* $p(\omega_j)$ are representing the class prior probabilities. C represents the total number of classes.

iii. Maximum information Entropy (MIE): To select the bands with maximum information to improve the classification accuracy, maximum information entropy (MIE) is used. The optimum probability distribution is the one with the maximum information entropy. Shannon entropy is the fundamental unit of information in information theory. Let R be a vector; its information quality can be quantified discretely

$$E(R) = -\sum_i p(y_i) log_2 p(y_i) \tag{22}$$

Here, E(R) is the entropy of R, $p(y_i)$ is the probability and y_i is the component of R.

A straightforward way is directly collecting those features with high entropy to determine the most distinct feature subset, in which raw data information is taken notice of the maximum extent. If the feature subset has s features, the issue can be mathematically described as,

$$MIE(b_i, b_j) = \frac{1}{s} \sum_{i,j=1}^{s} E(R_{i,j}) \tag{23}$$

where $E(R_{i,j})$ is the correlation between the i^{th} and j^{th} bands.

Imagine, N grey wolves are inputted across the early population; this is going to be indispensable to establish an entropy of the particular grey wolf at the beginning of the population using the Eq. (22). correspondingly, to designate the grey wolves, the first three maximum entropies should be interchangeable, and are allocated as α, β, and δ wolf one after another. The leftover grey wolves are contemplated as ω wolves. Deficiency of the location of the prey might be useful to interchange the adequate location vector of the prey with the position of the α wolf.

Every grey wolf is accomplished to track down the prey and confine it. Surrounding the prey throughout the hunt can perhaps be the Eqs. (7) to (10) and setting a=2.

Consequently, every wolf is surrounded by the prey is essential to enhance at the right time in the context of the locations of α, β, and δ wolves to enclose the prey. The hunting process is fulfilled by calculating from Eqs. (11) to (17). Repeating the encircling and searching process frequently will lead to the finest result.

The fitness function for the grey wolf optimizer (GWO) is defined as

$$F_n = CorM(b_i, b_j) + D_b(b_i, b_j) + MIE(b_i, b_j) \tag{24}$$

Here, F_n is the fitness function of α, β, λ and ω. $CorM(b_i, b_j)$ is representing the correlation coefficient between the bands b_i *and* b_j. $D_b(b_i, b_j)$ is the distance calculation of the bands b_i *and* b_j. $MIE(b_i, b_j)$ is representing the maximum information entropy of the bands b_i *and* b_j. The selected band subset of the Salinas dataset is depicted in Fig. 4.

Deep kernel attention transformers

In this article, we have suggested a novel transformer-based model named Deep Kernel Attention Transformer (DKAT). Deep kernel attention transformer working flow is designed based on Visual Transformer (ViT). ViT is a model developed for image classification; it is used to apply transformers like architecture over image patches. Vision Transformers require partitioning the input image into patches of the same shape and vectorization of the patches. Vectorization means reshaping a tensor into a vector. If the patches are defined as $d_1 \times d_2 \times d_3$ tensors, and then the vectors $d_1 d_2 d_3 \times 1$ dimensional vectors. After the image is split into n layers from $1, 2, 3 \ldots \ldots .., n$, then we have to apply the dense layer. The output of the dense layer will be $Z_1 = Wx_1 + b$, here linear activation function is not applied. The dense layer is only the activation function. Here, W and b are the parameters; likewise, all the vectors will get the dense layer output. So, that dense layer has the same parameters. The Deep Kernal attention transformer (DKAT) and working procedure of DKAT are depicted in Fig. 5.

In addition, we need to add the position encoding to the vectors $Z_1, Z_2, Z_3, \ldots \ldots \ldots, Z_n$. An input image is broken down into n patches; each patch has a position that is an integer between 1 to n. Positional encoding maps an integer to a vector, and the shape of the vector is the same as Z. Add the positional encoding vectors to the Z vectors. In this way, the 'Z' vector captures a patch's content and position. X_1 to X_n are the vectorization of n patches. Let vectors Z_1 to Z_n be the best results of linear transformation and positional encoding. They are the representation of the n patches. They capture both the content and positions of the patches. Aside from the m patches, we use the cls token for classification. We are using the *cls* token because the output of the transformer is an embedding layer, it takes input as a *cls* token, and it will provide the output as Z_0. Z_0 has the shape of the other Z vectors. The output of the kernel Attention layers is a sequence of n + 1 vectors. And then adding another layer of kernel attention layer and dense layers. The kernel attention layer and dense layers constitute the transformer encoder network. The outputs are the sequence of n + 1 vectors denoting the kernel attention layers, feed-forward layers, and multi-layer perceptron (MLP). Network vector $c_{0,\ldots\ldots,n}$ are the output of the transformer. To perform the classification task, we need only c_0. It is the feature vector extracted from the image. Feed c_0 into a SoftMax classifier. The classification results will be based on c_0. The classifier output vector is P. P's shape, depending on the amount of classes in the input. For example, the Indian Pines dataset has 16 classes. So, vector P has 16 classes. These vectors also indicate the classification results. During training, we compute the cross entropy of vector P and the ground truth of the dataset. Then, we will add the gradient of cross-entropy loss concerning the model parameters and the performance of gradient descent to update the parameters. This allows DKAT to learn hierarchical context information and adjust hyperparameters like `N` to control the scope of self-attention calculation. Additionally, DKAT uses a hierarchical training strategy to improve its generalization capabilities, allowing it to learn more generalizable features. In the first stage of training, DKAT is trained on a small dataset of patches, and in the second stage, DKAT is fine-tuned on a larger dataset of patches. Table 1 shows the hyperparameter tuning of DKAT.

Dataset description

Dataset characteristics

When endeavouring to create a model for hyperspectral data cubes, one must take into account the various dataset characteristics that play a pivotal role in determining its

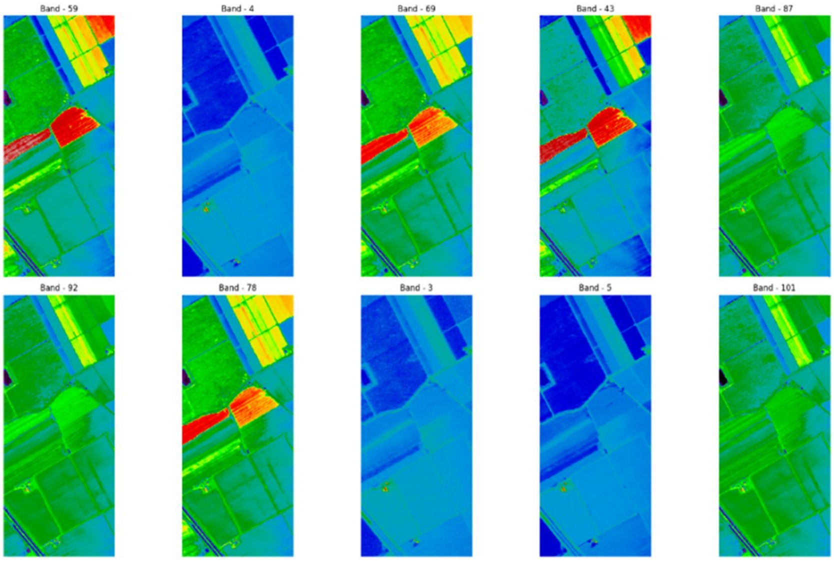
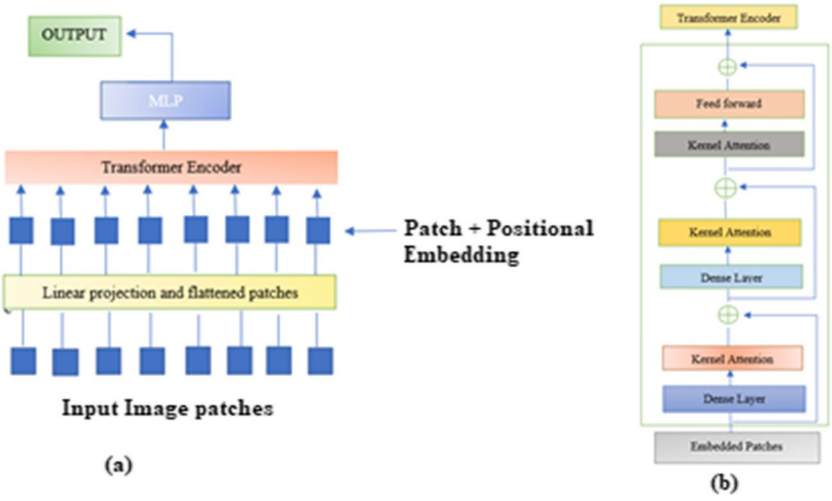


Fig. 4 Selected band subset from Salinas dataset

Fig. 5 **a** Deep Kernel Attention Transformer (DKAT), and **b** Working procedure of DKAT



performance and generalization. This is because of the factors such as the size of the hyperspectral data cubes, spatial and spectral resolutions, noise levels, and potential class imbalance can have a significant impact on the model's ability to accurately and reliably perform in real-world scenarios. Through the careful optimization of these key factors, we can ensure that our model is capable of delivering accurate, reliable, and effective results in a variety of real-world applications. The dataset characteristics table is given in Table 2.

Indian Pines (IP)

Using an AVIRIS sensor at a place with mixed vegetation in northwest Indiana, this dataset was collected. It includes 145 × 145 pixels, here individual pixel includes 224 spectral bands wavelengths extending from 0.4–2.5 µm. This IP dataset has 16 number classes. As an example, 'Oats' contain only 20 labeled samples, although 'Soyabean-mintill' has 2455 labeled pixels. In Fig. 6, (a) false-color composite (FCC) image, (b) ground truth reference map is shown, and (c) detailed classes, with their training and testing samples are described in Table 3. We can download the Salinas dataset by using the below link

https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Indian_Pines.

Salinas

Using AVIRIS (Airborne Visible/Infrared Imaging Spectrometer), the Salinas dataset was collected. The dimensions of every image are 512 × 217 pixels. It has a high spatial resolution of 3.7 m per pixel. This dataset encompasses 204 spectral bands, but some low signal-to-noise-ratio bands were eliminated. It also has 16 classes.

Table 1 Hyperparameter tuning of DKAT

Parameters	Size
Kernel size	3*3
Number of filters	16
Number of layers	10
Attention heads	8
Hidden units	64
Learning rate	0.001
Optimizer	Adam
Batch size	32
Epochs	25
Loss	Sparse Categorical Cross Entropy

In Fig. 6, (d) false-color composite (FCC) image, (e) ground truth reference map is specified as well as in (f) detailed classes with their training and testing samples are described. In Table 3, we can download the Salinas dataset by using the below link

https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Salinas.

Pavia University (PU)

In the location of northern Italy in 2001, using a Reflective Optics System Imaging Spectrometer sensor this dataset was acquired. The aforementioned dataset capability is 610 × 340 pixels. This dataset has nine (9) urban land-cover classes, the spatial resolution of each image is 1.3 m per pixel, as well as the wavelength range is from 430 to 860 nm. Pavia university has 103 spectral bands. In Fig. 6, (g) false-color composite (FCC) image, (h) ground truth reference map are given as well as in (i) detailed classes with their training and testing samples are described in Table 4**,** and we can download Pavia University dataset from below link

Table 2 Dataset characteristics

	Collection location	Acquisition equipment	Number of categories	Number of bands	Number of bands after denoising	Data size (Pixel)	Spatial resolution (m)	Spectral coverage (μm)
Indian Pines (IP)	Indiana, USA	AVIRIS	16	224	220	145*145	20	0.4 ~ 2.5
Salinas	Salinas Valley, California	AVIRIS	16	224	220	86*63	3.7	0.4 ~ 2.5
Pavia University (PU)	Northern Italy	ROSIS	9	103	103	610*601	1.3	NA
Botswana	Okavango Delta, Botswana	NASA E0-1	14	145	NA	1496*256	30	NA
Kennedy Space Center (KSC)	Kennedy Space Centre, Florida	NASA AVIRIS	13	176	NA	512*614	NA	0.4 ~ 2.5

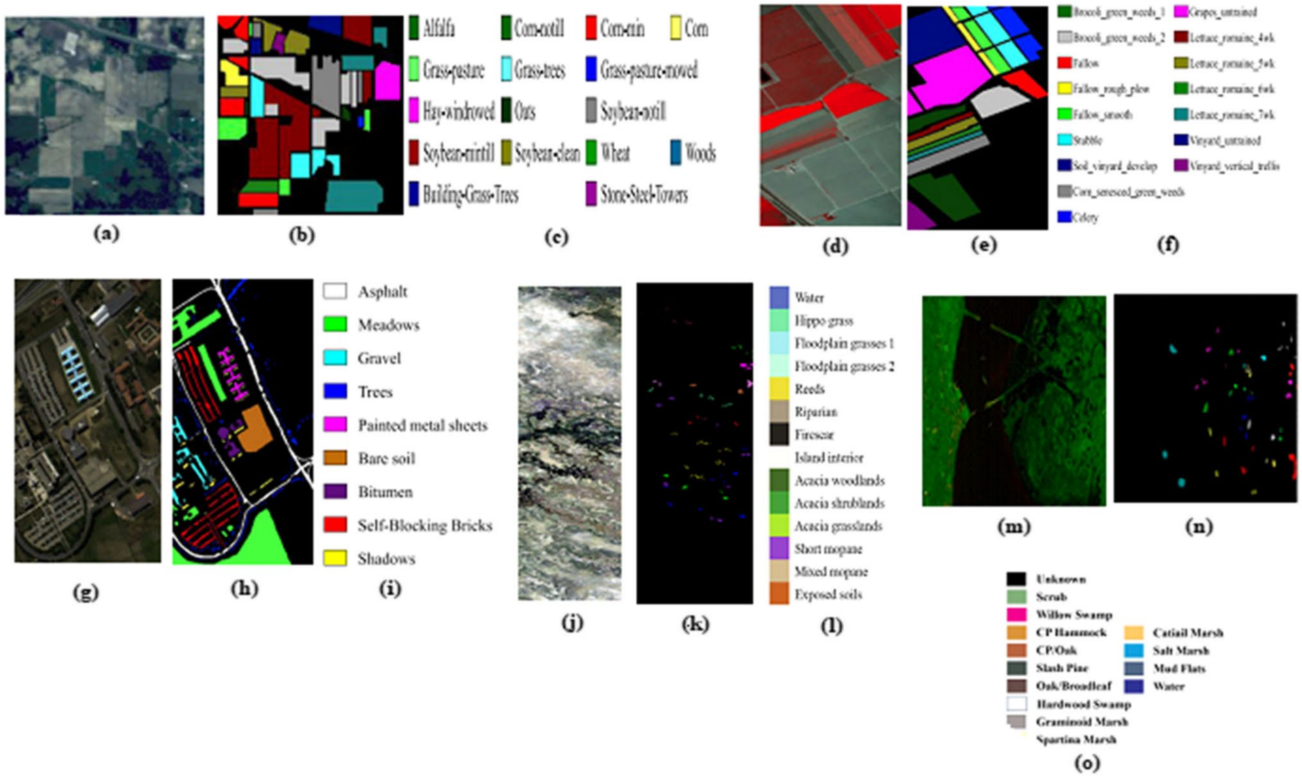


Fig. 6 **a** IP RGB image, **b** ground truth, **c** classes, **d** Salinas RGB image, **e** ground truth, **f** classes, **g** PU RGB image, **h** ground truth, **i** classes, **j** Botswana RGB image, **k** ground truth, **l** classes, and **m** KSC RGB image, **n** ground truth, and **o** classes

https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Pavia_University_scene

Botswana

This dataset is collected using NASA EO-1 sensors in Okavango Delta, Botswana. This has 145 bands with a pixel resolution of 1476*256. In Fig. 6, (j) false-color composite (FCC) image, (k) ground truth reference map is given as well as in (l) detailed classes with their training and testing samples are described in Table 4**,** This dataset is accessible in the lower link, so download it freely.

http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Botswana.

Kennedy Space Center (KSC)

This dataset is collected from the Kennedy space center in Florida using NASA AVIRIS sensors. This has 176

Table 3 Explanation of the Indian Pines (IP) and Salinas dataset

Indian Pines				Salinas		
Class number	Class	Training	Testing	Class	Training	Testing
Land Cover Classes				Land Cover Classes		
C1	Alfalfa	5	41	Fallow	11	1965
C2	Grass-trees	71	659	Stubble	22	3978
C3	Hay-windrowed	46	432	Broccoli_green_weeds_1	11	1998
C4	woods	123	1142	Broccoli_green_weeds_2	20	306
C5	Grass-pastures	47	436	Corn_senesced_green_weeds	18	3260
Land Use Classes				Land Use Classes		
C6	Corn-notill	139	1289	Celery	19	3560
C7	Corn	23	214	Lettuce_romainc_4wk	9	1059
C8	Corn-min	81	749	Lettuce_romainc_5wk	11	1916
C9	Oats	2	18	Lettuce_romainc_6wk	5	911
C10	Soyabean-notill	95	877	Lettuce_romainc_7wk	6	1064
C11	Soyabean-clean	58	535	Fallow_rough_plow	8	1386
C12	Soyabean-mintill	240	2215	Fallow_smooth	14	2664
C13	Grass-pasture-mowed	3	25	Vineyard_untrained	40	7228
C14	Wheat	20	185	Vineyard_vertical_trellis	10	1797
C15	Stone-Steel-Towers	9	84	Grapes_untrained	62	11209
C16	Buildings-Grass-Trees	38	348	Soil_vineyard_develop	34	6169
	Total	**1000**	**9249**	**Total**	**300**	**53829**

bands and 13 classes. 512*614 pixels in the wavelength 400-2500 nm of the electromagnetic spectrum. In Fig. 6, (m) false-color composite (FCC) image, (n) ground truth reference map are given as well as in (o) detailed classes, with their training and testing samples are described in Table 4. By using below link we can freely access the dataset

http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes#Kennedy_Space_Center_.28KSC.29.

Table 4 Description of Botswana, Kennedy Space Centre, and Pavia University dataset

Botswana				Kennedy Space Centre			Pavia University		
S.no	Class name	Training	Testing	Class name	Training	Testing	Class name	Training	Testing
Land Cover Classes				Land Cover Classes			Land Cover Classes		
C1	Exposed soils	1	93	Scrub	33	314	Tress	72	2992
C2	Hippo grass	2	97	Willow Swamp	23	220	Bare Soil	118	4911
C3	Floodplain grasses1	3	245	Cabbage Palm Hammock	24	232	Meadows	436	18213
C4	Floodplain grasses2	3	209	Cabbage Palm/Oak Hammock	24	228	Land Use Classes		
C5	Reeds1	3	263	Slash Pine	15	146	Gravel	49	2050
C6	Riparian	3	263	Oak/ Broadleaf Hammock	22	207	Shadows	22	925
C7	Fierscar2	3	253	Hardwood Swamp	9	96	Bitumen	31	1299
C8	Acacia woodlands	4	306	Graminoid Marsh	38	352	Asphalt	155	6476
C9	Acacia shrublands	3	242	Spartina Marsh	51	469	Painted metal sheets	31	1314
C10	Acacia grasslands	4	297	Cattail Marsh	39	365	Self-Blocking Bricks	86	3596
C11	Short mopane	2	177	Mud Flats	49	454	-	-	-
C12	Mixed mopane	3	262	-	-	-	-	-	-
Land Use Classes				Land Use Classes			**-**	**-**	**-**
C13	Water	3	264	Salt Marsh	41	378	-	-	-
C14	Island interior	3	197	water	91	836	-	-	-
	Total	40	3168	**Total**	**459**	**1297**	**Total**	**1000**	**41776**

Evaluation metrics

To understand the efficiency of the suggested algorithm for LULC classification, there are some evaluation metrics. They are Kappa coefficient (KC), overall accuracy (OA), and average accuracy (AA). The techniques mentioned above are exploited to test the pixels and confusion metrics $[T_{cc\prime}]$ are used to precise the given classifier.

1. **Overall accuracy:** Overall accuracy refers to the percentage of pixels correctly categorized into all the pixels. It is defined as

$$OA = \frac{1}{T} \sum_{c=1}^{C} T_{cc} \tag{25}$$

Here, T represents the confusion matrix of the chosen classifier, and T_{cc} is represents the number of testing pixels.

2. **Average accuracy:** Average accuracy calculates the average per-class classification accuracy, whereas the proportion of pixels in a specific class is correctly classified to all of the pixels in that class is known as per-class accuracy.

$$AA = \frac{1}{C} \sum_{c=1}^{C} \frac{T_{cc}}{\sum_{c'}^{C} T_{cc'}} \tag{26}$$

Here, T represents the number of difficult pixels, and Tcc' signifies the confusion matrix of a given classifier.

3. **Kappa coefficient: The** Kappa coefficient tries to fix OA by lowering its worth in the presence of an agreement that could be reached by chance.

$$KC = \frac{\frac{1}{T}\sum_{c} T_{cc} - \frac{1}{T^2}(\sum_{c\prime} T_{cc\prime})(\sum_{c\prime} T_{c\prime c})}{1 - \frac{1}{T^2}(\sum_{c\prime} T_{cc\prime})(\sum_{c\prime} T_{c\prime c})} \tag{27}$$

Experimental results

The purpose is to assess the anticipated method's accuracy, the outcomes are compared in this section to those of various land use/land change (LU/LC) classification algorithms that are currently in use. The existing methods used to compare the Pavia University, Indian Pines, and Salinas datasets are PCA-CNN (Chen et al. 2021), SVM (Chen et al. 2014), CNN-PPF (Li et al. 2016), GCN-CNN (Hong et al. 2020), DHC-Net (Zhu et al. 2018). Principal component analysis (PCA) is the dimensionality reduction technique for feature extraction and classification of convolutional neural networks (CNN). And the combination of the PCA-CNN gives better results for hyperspectral image classification. It's a classic machine learning approach used for training on small datasets. For SVM, the kernel function is used. Cost value and gamma values are used for the kernel and combinedly selected to get the average performance of the classified output. A CNN framework is developed based on PPF. PPFs are employed to increase the number of training samples. This can compensate for the lack of training sample data. It stands for graph convolutional network-convolutional neural network. It is a new concatenated fusion framework. In this method, the extracted features from CNN have been given to the GCN classifier. The combined use of CNNs and GCNs gives more accurate results for Hyperspectral image classifications. A deformable convolutional neural network is used to get the convolutional sampling locations. The size and shape of the hyperspectral images are composite because of spatial circumstances. Classifying land use/ land cover (LULC) classes using hyperspectral images is difficult by a reason of a lower amount of training samples. To overcome this issue, we have proposed a deep kernel attention transformer (DKAT) for classifying the land use/ land cover (LULC) classes (Tables 5 and 6).

Interpretability By understanding Deep kernel attention transformers (DKAT), we can unlock the potential for greater interpretability in our transformers. With kernels, we can measure the similarity between tokens in a sequence, providing us with valuable insights. DKAT takes this one step further, allowing us to easily visualize attention weights and comprehend why certain tokens are being emphasized. The traditional matrix approach can be clumsy, but with DKAT's kernel set, we can simplify the visualization process. Imagine the impact this can have on real-world applications, where accuracy and reliability are paramount. Embrace the power of DKAT and unlock the full potential of our transformers. With the use of DKAT, not only we can enhance interpretability, but we can also simplify the process of explaining the model's predictions. Traditional transformers can be challenging to comprehend as predictions are based on attention weights alone. However, DKAT goes a step further by taking into account kernels when making predictions. This allows us to identify the significant tokens for the prediction, leading to a clearer understanding of the model's prediction. This aspect of DKAT is particularly helpful in real-world scenarios where predictions need to be explained.

Training data The Deep Kernel Attention Transformer (DKAT) is a novel approach to hyperspectral image classification that overcomes the challenge of insufficient training data. DKAT leverages a kernel attention mechanism to capture extended relationships between pixels, resulting in enhanced accuracy in categorizing images even with a limited number of training examples. We have

Table 5 Computational complexity of DKAT for different datasets

Dataset	Indian Pines		Salinas		Pavia University		Botswana		Kennedy Space Centre	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Computational complexity	10^8	10^5	10^7	10^5	10^8	10^5	10^6	10^5	10^5	10^5

Table 6 Execution time in seconds for Proposed vs. state-of-the-art models

Method	Time in Seconds	Time in Seconds	Time in Seconds	Time in Seconds	Time in Seconds
PCA-CNN (Wang et al. 2022)	378	-	814	-	-
SVM (Chen et al. 2021)	156	-	533	-	-
CNN-PPF (Chen et al. 2014)	286	1259	1015	-	-
GCN-CNN (Li et al. 2016)	-	-	-	-	-
DHC-Net (Hong et al. 2020)	468	2820	2700	-	-
Proposed (DKAT)	327	235	330	239	233

learned that the kernel attention mechanism is an effective method for capturing extended relationships between pixels in hyperspectral image classification. It works by first learning a set of kernels that represent different spatial relationships between pixels, which are then used to attend to different parts of the image. This helps to capture long-range dependencies that are often important for classification, and enhances accuracy even when there is a limited amount of training data available. DKAT involves learning a set of kernels that represent different spatial relationships between pixels, which can help capture long-range dependencies and improve accuracy, even with limited training data.

Comparison with other algorithms To verify the classification performance of the proposed method, we have compared it with some of the state-of-the-art algorithms. The algorithms are PCA-CNN, SVM, CNN-PPF, GCN-CNN, are DHC-Net. Specifically, randomly we have taken 20% of training samples from each class to construct the training set, and the remaining samples are used to validate the effectiveness of the proposed method. By using the OA, AA, and KC we have obtained the per-class classification accuracy for the proposed method including state-of-the-art methods. For the Pavia University dataset, the proposed algorithm (DKAT) has been obtaining good classification accuracy. Compared with the SVM-based algorithms, the proposed method has achieved more OA, AA, and KC accuracy. When compared to the conventional CNN, the proposed algorithm yields higher accuracy. Figure 7 represents the Indian Pines dataset's original image, ground truth image, and existing methods results, along with the suggested method. Table 7 depicts the classification performance of the IP dataset with existing methods, and also the proposed method. Figure 8 depicts the (a) overall accuracy (OA), (b) average accuracy (AA), and (c) kappa coefficient (KC) of the existing methods and the proposed method for the IP dataset. Figure 9 represents the Salinas dataset's original image, ground truth image, and existing methods results, along with the suggested method. Table 8, classification performance of Salinas dataset with existing methods, and also proposed method. Figure 10 depicts the (a) overall accuracy (OA), (b) average accuracy (AA), and (c) kappa coefficient (KC) of the existing methods and the proposed method for the Salinas dataset. Figure 11 represents the PU dataset's original image, ground truth image, and existing methods results, along with the suggested method. Table 9, classification performance of the Salinas dataset with existing methods, and also the proposed method. Figure 12 depicts the (a) overall accuracy (OA), (b) average accuracy (AA), and (c) kappa coefficient (KC) of the existing methods and the proposed method for the PU dataset. Figure 13 represents the KCS dataset's original image, ground truth image, and existing methods results, along with the suggested method. Table 10 depicts the classification performance of the KSC dataset with existing methods, and also the proposed method. Figure 14 depicts the (a) overall accuracy (OA), (b) average accuracy (AA), and (c) kappa coefficient (KC) of the existing methods and the proposed method for the KSC dataset. Figure 15 depicts the (a) overall accuracy (OA), (b) average accuracy (AA), and (c) kappa coefficient (KC) of the existing methods and the proposed method for the Botswana dataset. Figure 16 represents the PU dataset's original image, ground truth image, and existing methods results, along with the suggested method. Table 11 depicts the classification performance of the Salinas dataset with existing methods, and also the proposed method.

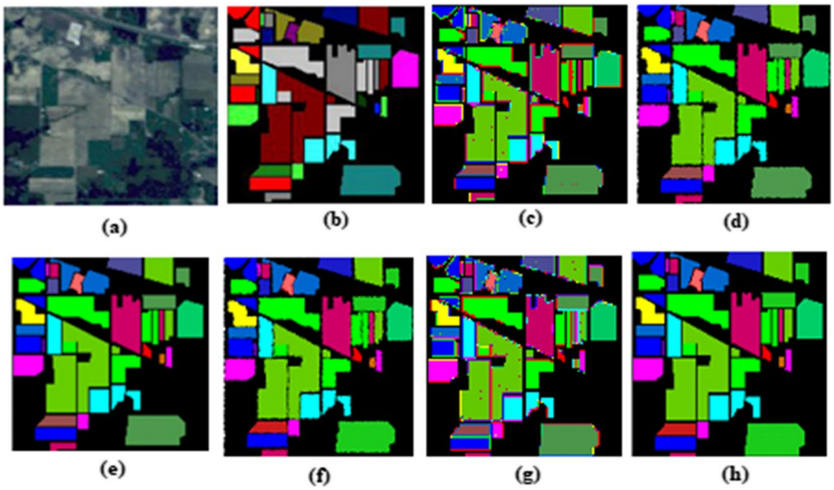


Fig. 7 Classification Results of Indian Pines dataset **a** Benchmark dataset, **b** Ground Truth, **c** PCA-CNN (Chen et al. 2021), **d** SVM (Chen et al. 2014), **e** CNN-PFF (Li et al. 2016), **f** GCN-CNN (Hong et al. 2020), **g** DHC-Net (Zhu et al. 2018), and **h** Proposed DKAT

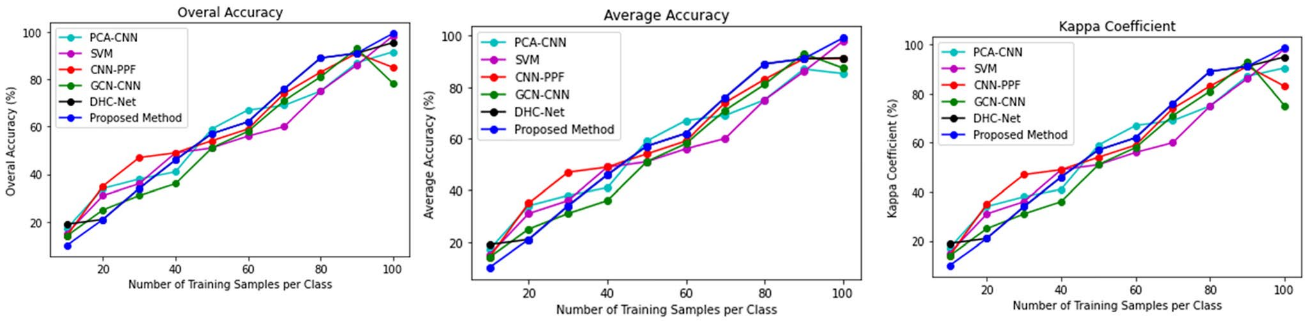
Table 7 Classification performance of different algorithms for the Indian Pines (IP) dataset

Class no	PCA-CNN	SVM	CNN-PPF	GCN-CNN	DHC-Net	Proposed method
Land cover classes						
C1	81.25	100	95.65	95.65	75.61	**100**
C2	86.36	98.84	85.24	626	94.09	**98.9**
C3	87.87	98.93	84.00	72.18	97.32	**99.13**
C4	84.29	96.19	94.22	92.09	84.98	**96.59**
C5	94.963	97.99	91.59	91.96	97.70	**98.12**
Land use classes						
C6	98.66	99.85	98.15	98.59	97.87	**99.95**
C7	60.87	100.0	82.86	97.14	60.00	**100**
C8	99.32	100.0	99.16	98.04	100.00	**100**
C9	33.33	94.44	96.00	100	94.44	**96.26**
C10	83.70	96.21	84.08	70.95	90.39	**97.37**
C11	96.04	98.56	72.08	73.74	99.28	**98.67**
C12	75.91	95.29	92.85	77.72	89.87	**96.37**
C13	100.0	98.42	100	99.87	100.00	**99.43**
C14	99.66	99.19	86.68	86.29	98.15	**99.35**
C15	86.55	99.42	94.35	83.10	81.56	**99.36**
C16	95.29	94.19	100	100	100.00	**95.26**
OA (%)	91.60	98.36	85.00	78.25	95.53	**98.42**
AA (%)	85.23	97.96	91.06	87.35	91.33	**99.13**
Kappa (%)	90.39	98.13	82.95	75.19	94.89	**98.62**

Bold indicates the maximum class-wise classification accuracy of the proposed model for Indian Pines (IP) dataset when compared to other existing systems

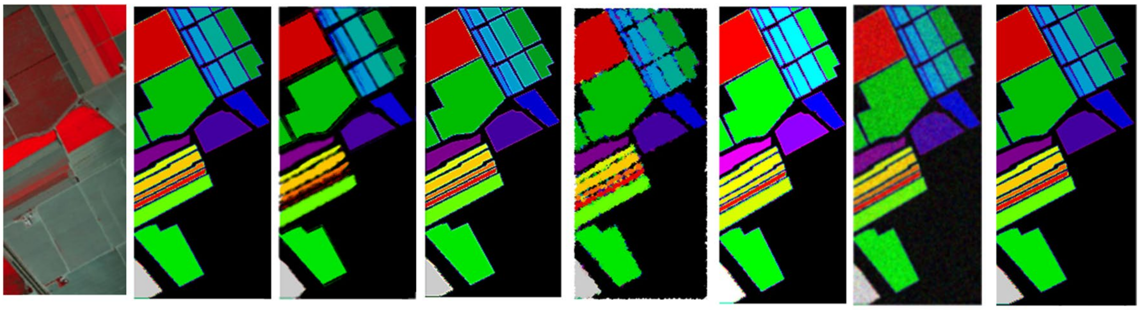
Computational complexity

We have achieved low computational complexity for all the algorithms used in our work because we have done each part individually. First, we extracted the features separately for each dataset using the t-SNE algorithm. In this, the high-dimensional intervals among data points towards the conditional probabilities will be transformed initially by SNE. Kullback-Leiber (KL) divergence is used, to evaluate the correspondence among two distributions. So that, we will get the useful features from the data. Then, we selected the bands using GWO from the extracted features. In GWO, we have defined a fitness function to select the appropriate bands, which are useful for classification. Here, the use of the fitness function is to select the relevant bands. After selecting a number of bands, which are having more information, the bands will be sent to the classifier model i.e., DKAT. Then the classifier model will classify the LU/LC cover classes. It will give the classification map as well as OA,



(a) (b) (c)

Fig. 8 Classification performance of existing methods along with the proposed method of the Indian Pines dataset, **a** Overall accuracy, **b** Average accuracy, and **c** Kappa coefficient



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 9 Classification results of Salinas dataset **a** Benchmark Data, **b** Ground Truth, **c** PCA-CNN, **d** SVM, **e** CNN-PPF, **f** GCN-CNN, **g** DHC-Net, and **h** proposed method

AA, and KC. We can also use this model for real-time applications and also for large-scale applications. But we have to select the number of features, and bands according to the problem with which we are going to work.

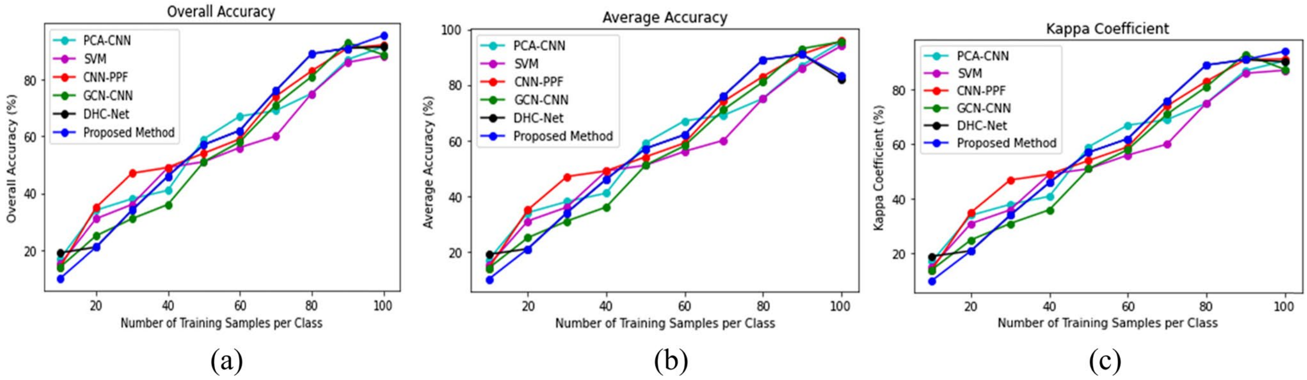
The time complexity of the DKAT model is calculated using the O(n*s*k). Here, n is the number of samples, s is the size of the image i.e., height, width, bands, and k is the attention heads. The Indian pines dataset has 1000 training samples, the image size is 100*100*10, and the number of attention heads in our model is 8, so the computational complexity of the Indian pines dataset is O(1000*100*100*10*8) = 10^8 The computational complexity for the deep learning models increases with the number of samples, the number of dimensions, and the number of hidden layers. The computational complexity of the DKAT model is shown in Table 5, and the computation time in seconds for different models is depicted in Table 6.

Limitations and future scope

The number of kernels used in DKAT can have a significant impact on the accuracy of its classifications. If the number of kernels is too small, it may not be able to capture the long-range spatial dependencies in the data, while if the number of kernels is too large, it may become too computationally expensive to train and run. It's important to find the right balance to ensure optimal performance. The size of the kernels used in DKAT can also affect the classification accuracy. If the kernels are too small, then the model may not be able to capture the important features in the data. If the kernels are too large, then the model may be less sensitive to noise. It's important to find the right balance to ensure optimal performance. The accuracy of DKAT's classification can be affected by the quality of its training data. If the training data does not accurately represent the test data, DKAT may not perform well in generalizing.

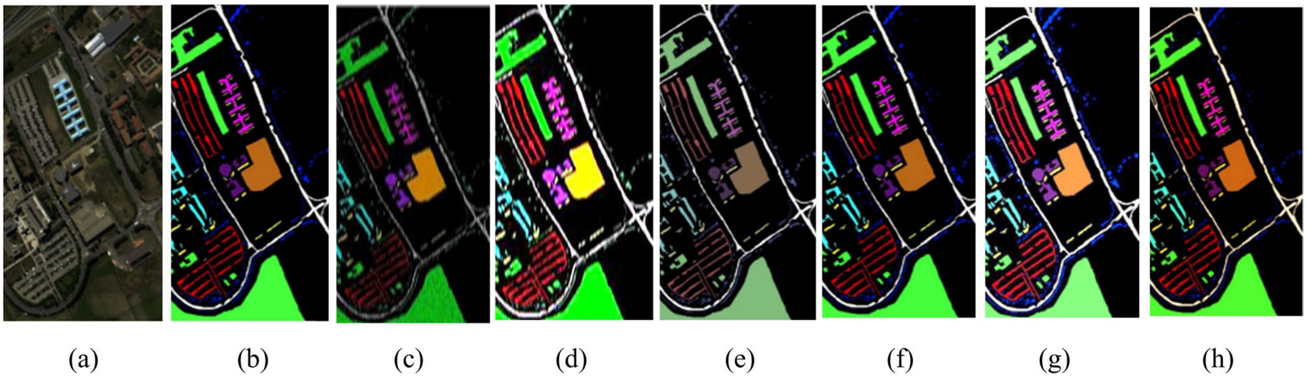
Table 8 Classification performance of different algorithms for the Salinas dataset

Class	PCA-CNN	SVM	CNN-PPF	GCN-CNN	DHC-Net	Proposed method
Land cover classes						
C1	95.50	98.81	98.66	99.20	65.85	**99.45**
C2	99.07	99.30	99.09	99.95	89.34	**98.13**
C3	88.45	96.96	99.17	97.38	92.50	**99.48**
C4	97.68	99.70	99.87	99.51	62.91	**97.14**
C5	99.48	96.18	95.30	98.55	95.85	**96.87**
Land use classes						
C6	99.62	99.60	99.57	99.98	99.24	**99.65**
C7	99.68	99.63	99.61	99.81	36.00	**97.81**
C8	85.26	74.50	84.84	66.84	96.74	**85.91**
C9	95.43	99.06	98.50	99.85	50.00	**99.16**
C10	97.24	87.74	90.90	93.71	68.73	**87.18**
C11	98.55	97.23	99.27	98.41	69.65	**98.42**
C12	99.99	98.96	99.99	99.94	76.74	**97.13**
C13	99.93	98.50	99.19	99.88	100.00	**99.26**
C14	99.39	93.43	94.71	98.37	97.01	**96.19**
C15	71.49	64.96	75.23	73.82	75.79	**77.59**
C16	98.70	98.84	98.93	97.94	91.57	**98.12**
OA (%)	91.48	88.31	92.14	88.72	91.46	**95.46**
AA (%)	95.34	93.96	95.80	95.50	82.06	**83.35**
Kappa (%)	90.51	87.11	91.24	87.47	90.24	**94.05**



(a) (b) (c)

Fig. 10 Classification performance of existing methods along with the proposed method of the Salinas, **a** Overall accuracy, **b** Average accuracy, and **c** Kappa coefficient



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 11 Classification results of PU dataset **a** Benchmark Data, **b** Ground Truth, **c** PCA-CNN, **d** SVM, **e** CNN-PPF, **f** GCN-CNN, **g** DHC-Net, and **h** proposed method

Table 9 Classification performance of different algorithms for the Pavia University dataset

Class	PCA-CNN	SVM	CNN-PPF	GCN-CNN	DHC-Net	Proposed method
Land cover classes						
C1	96.28	95.58	95.19	81.32	98.51	**98.89**
C2	91.31	96.10	85.98	94.90	99.97	**99.99**
C3	95.56	98.68	89.46	85.53	93.23	**97.58**
Land use classes						
C4	99.21	94.17	90.98	94.75	98.93	**99.15**
C5	100.0	100.0	99.91	99.88	100.0	**99.43**
C6	100.0	100.0	92.87	89.62	99.62	**99.75**
C7	99.54	99.63	94.58	90.42	97.94	**98.67**
C8	97.51	99.12	87.36	78.95	98.46	**99.33**
C9	100.0	97.79	99.13	100	100.0	**99.98**
OA (%)	93.35	96.95	89.66	90.46	99.11	**99.19**
AA (%)	97.71	97.90	91.30	90.60	98.52	**99.32**
Kappa (%)	93.35	95.91	86.54	87.41	98.82	**99.14**

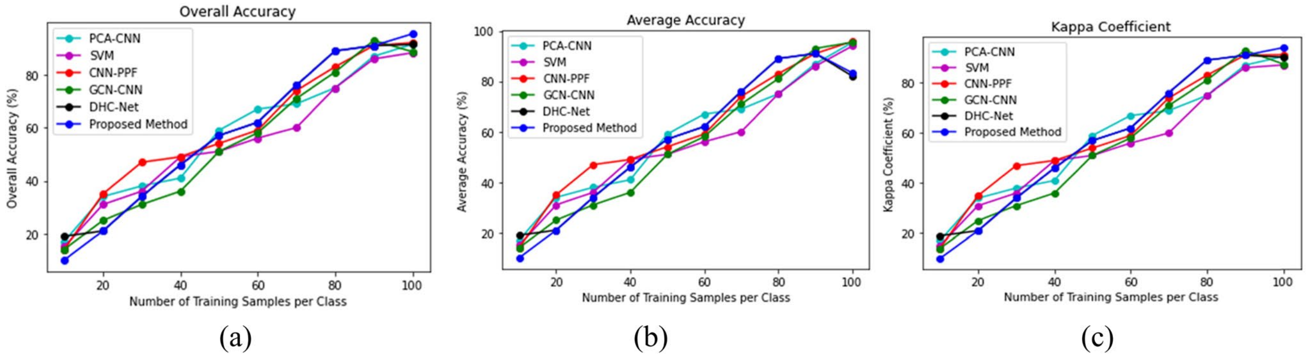


Fig. 12 Classification performance of existing methods along with the proposed method of Pavia University, **a** Overall accuracy, **b** Average accuracy, and **c** Kappa coefficient

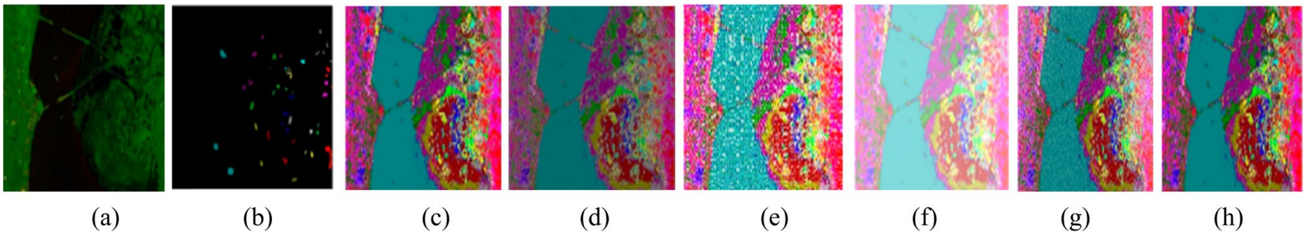
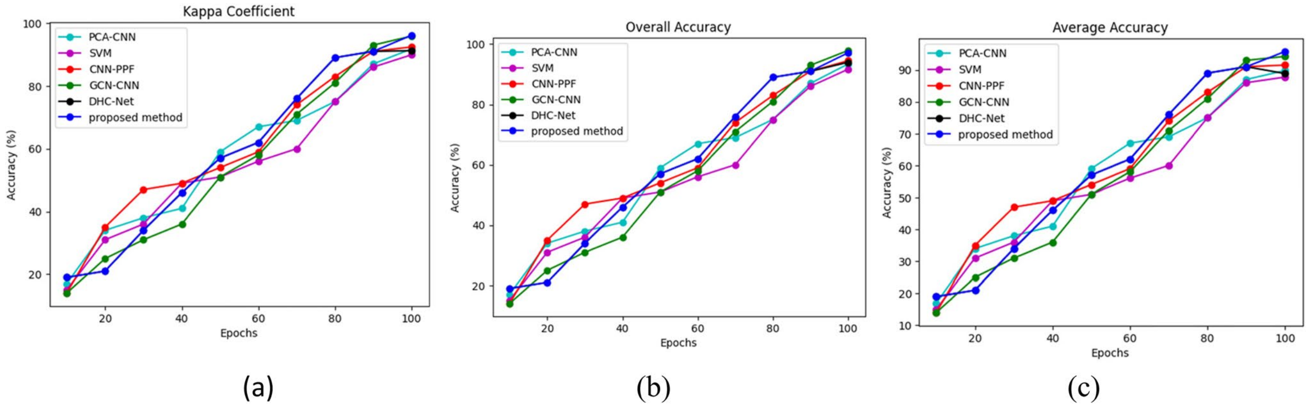


Fig. 13 Classification results of KSC dataset, **a** Benchmark Data, **b** Ground Truth, **c** PCA-CNN, **d** SVM, **e** CNN-PPF, **f** GCN-CNN, **g** DHC-Net, and **h** proposed method

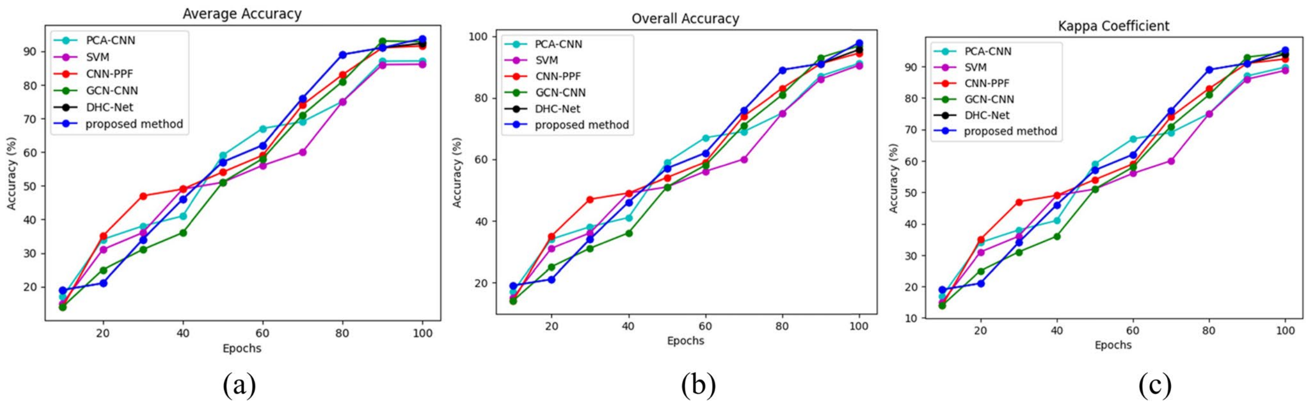
Table 10 Classification performance of different algorithms for the Kennedy Space Centre dataset

Class	PCA-CNN	SVM	CNN-PPF	GCN-CNN	DHC-Net	Proposed method
Land cover classes						
C1	92.23	90.71	99.18	99.63	89.57	**99.63**
C2	92.81	85.7	100	90.15	85.07	**90.15**
C3	93.39	88.39	98.39	89.64	88.46	**99.64**
C4	86.54	77.69	95	87.23	79.81	**87.23**
C5	77.5	70	93.44	99.06	99.38	**99.06**
C6	89.66	87.93	87.66	95.14	92.17	**95.14**
C7	92.86	87.14	93.75	99.52	95.22	**99.52**
C8	91.93	84.89	87.48	90.03	87.03	**90.03**
C9	89.72	97.72	91.69	99.91	99.84	**99.91**
C10	95.02	89.61	85.56	99.29	91.34	**99.19**
Land use classes						
C11	87.63	88.54	85.77	92.34	85.04	**92.34**
C12	88.53	97.33	79.9	95.22	86.12	**95.22**
C13	89.93	95.2	92.03	87.31	75.88	**97.31**
OA (%)	93.41	91.67	94.49	97.87	93.97	**97.11**
AA (%)	89.82	87.75	91.52	94.23	88.85	**95.77**
Kappa (%)	91.78	89.91	92.40	95.85	91.21	**96.12**



(a) (b) (c)

Fig. 14 Classification performance of existing methods along with the proposed method of KSC dataset **a** Overall accuracy, **b** Average accuracy, and **c** Kappa coefficient



(a) (b) (c)

Fig. 15 Classification performance of existing methods along with the proposed method of Botswana dataset **a** Overall accuracy, **b** Average accuracy, and **c** Kappa coefficient

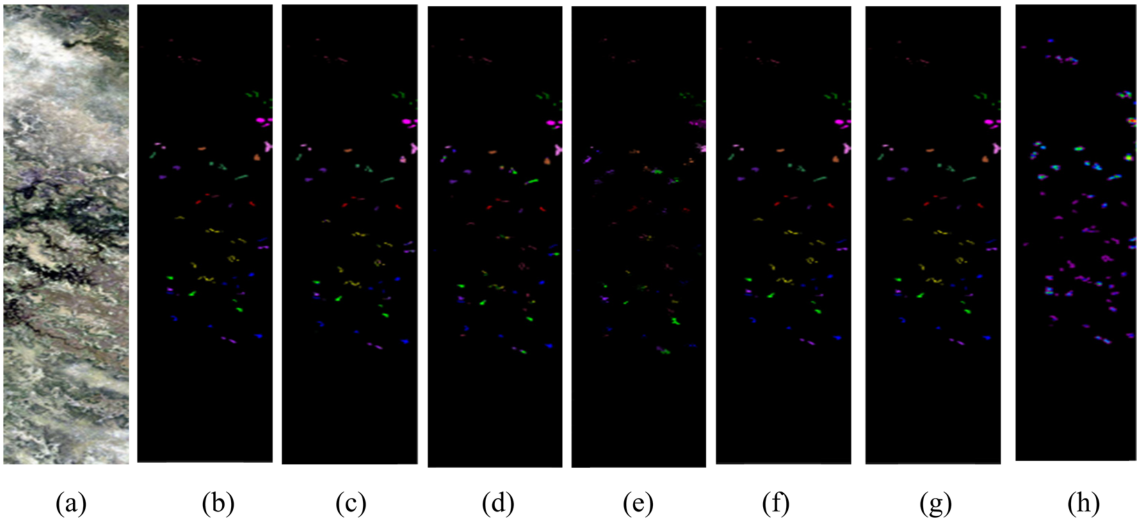


Fig. 16 Classification results of Botswana dataset, **a** Benchmark Data, **b** Ground Truth, **c** PCA-CNN, **d** SVM, **e** CNN-PPF, **f** GCN-CNN, **g** DHC-Net, and **h** proposed method

Table 11 Classification performance of different algorithms for the Botswana dataset

Class	PCA-CNN	SVM	CNN-PPF	GCN-CNN	DHC-Net	Proposed method
Land cover classes						
C1	90.32	92.18	99.18	95.87	79.31	**97.87**
C2	94.15	65.72	99.02	92.32	98.19	**92.32**
C3	91.39	98.39	98.39	79.64	86.29	**79.64**
C4	83.54	73.93	100	88.21	99.81	**89.21**
C5	67.91	77.54	93.44	100	99.86	**100**
C6	81.16	82.99	89.62	98.24	94.17	**99.24**
C7	72.66	87.85	93.75	89.29	97.32	**89.29**
C8	81.93	84.95	87.48	87.98	96.18	**87.98**
C9	99.42	99.12	91.93	97.74	97.17	**97.74**
C10	97.02	85.41	78.62	100	98.74	**100**
C11	89.32	78.94	81.75	95.26	89.94	**95.26**
C12	87.31	99.73	73.91	97.19	91.62	**97.19**
Land use classes						
C13	89.39	99.19	96.45	99.11	85.18	**99.11**
C14	93.66	79.21	99.89	87.59	78.95	**87.59**
OA (%)	91.13	90.53	94.49	96.87	95.63	**97.93**
AA (%)	87.08	86.08	91.52	92.81	92.33	**93.74**
Kappa (%)	89.81	88.75	92.40	94.37	93.94	**95.37**

In the future, classifying land use land cover using hyperspectral images with Multiple attentions in Deep kernel attention transformer (DKAT) is looking very promising. Even though it is a new method, DKAT has already shown impressive results. As research in this area progresses, DKAT may become even more capable and adaptable. One possible improvement is making the kernels more flexible to capture even more intricate spatial relationships. Another way to optimize the kernel attention mechanism is to reduce its computational cost and make it more efficient.

Conclusion

This article proposes a Deep Kernel Attention Transformer (DKAT) for land use/land cover (LU/LC) classification using hyperspectral images. To classify LULC, we have used two methods. One is feature extraction; the other is band selection. To extract the features from the land cover, t-distributed stochastic neighbouring embedding is used. Searching-based band selection using grey wolf optimizers (GWO) is used for selecting the informative bands from the dataset. After selecting the bands from the land cover, we have classified the land cover by using the DKAT. In DKAT, we have used multiple dense networks and kernel attention layers to enhance classification accuracy. We have calculated the accuracy using overall accuracy (OA), average accuracy (AA), and kappa coefficient (KC). The proposed method performance has been checked with five publicly available datasets, namely Indian Pines, Salinas, Pavia University, Botswana, and Kennedy Space Centre. For real-world scenarios, our proposed method gives robust results. In the real world, by using our proposed method can work with any dataset for classification. Coming to LU/LC change classification, we can know how much land cover classes have been spread or decreased around the land, and how much the land use classes have been used. As a result, we got the best classification accuracy compared with the state-of-the-art methods.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by [Ganji Tejasree], and [L. Agilandeeswari]. The first draft of the manuscript was written by [Ganji Tejasree] and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding The authors did not receive support from any organization for the submitted work.

Data Availability The datasets which are used in this study is openly available in GRUPO DE INTELIGENCIA COMPUTATIONAL (GIC) at https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes, (Graña et al. n.d.).

Declarations

Competing interests The authors declare no competing interests.

References

Aberna P, Agilandeeswari L, Bansal A (2023) Vision transformer-based watermark generation for authentication and tamper detection using Schur decomposition and hybrid transforms. Int J Comput Inf Syst Ind Manag Appl 15:107–121

Adam E, Mutanga O, Odindi J, Abdel-Rahman EM (2014) Land-use/cover classification in a heterogeneous coastal landscape using RapidEye imagery: evaluating the performance of random forest and support vector machines classifiers. Int J Remote Sens 35(10):3440–3458

Agilandeeswari L, Meena SD (2023) SWIN transformer based contrastive self-supervised learning for animal detection and classification. Multimed Tools Appl 82(7):10445–10470

Agilandeeswari L, Prabukumar M, Radhesyam V, Phaneendra KL, Farhan A (2022) Crop classification for agricultural applications in hyperspectral remote sensing images. Appl Sci 12(3):1670

Bachmann CM, Ainsworth TL, Fusina RA (2005) Exploiting manifold geometry in hyperspectral imagery. IEEE Trans Geosci Remote Sens 43(3):441–454

Bajwa SG, Bajcsy P, Groves P, Tian LF (2004) Hyperspectral image data mining for band selection in agricultural applications. Trans ASAE 47(3):895–907

Bioucas-Dias JM, Plaza A, Camps-Valls G, Scheunders P, Nasrabadi N, Chanussot J (2013) Hyperspectral remote sensing data analysis and future challenges. IEEE Geosci Remote Sens Mag 1(2):6–36

Cariou C, Chehdi K, Moan SL (2020) Improved nearest neighbor density-based clustering techniques with application to hyperspectral images. In: ICASSP. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 2020–2020

Chang C-I, Ren H (2000) An experiment-based quantitative and comparative analysis of target detection and image classification algorithms for hyperspectral imagery. IEEE Trans Geosci Remote Sens 38(2):1044–1063

Chen C, Li W, Su H, Liu K (2014) Spectral-spatial classification of hyperspectral image based on kernel extreme learning machine. Remote Sens 6(6):5795–5814

Chen H, Miao F, Chen Y, Xiong Y, Chen T (2021) A hyperspectral image classification method using multifeature vectors and optimized KELM. IEEE J Sel Top Appl Earth Obs Remote Sens 14:2781–2795

Christovam LE, Pessoa GG, Shimabukuro MH, Galo ML (2019) Land use and land cover classification using hyperspectral imagery: evaluating the performance of spectral angle mapper, support vector machine and random forest. Int Arch Photogramm Remote Sens Spat Inf Sci 42:1841–1847

Chutia D, Bhattacharyya DK, Sarma KK, Kalita R, Sudhakar S (2016) Hyperspectral remote sensing classifications: a perspective survey. Trans GIS 20(4):463–490

Devassy BM, George S (2020) Dimensionality reduction and visualisation of hyperspectral ink data using t-SNE. Forensic Sci Int 311:110194

Ettabaa KS, Salem MB (2018) Adaptive progressive band selection for dimensionality reduction in hyperspectral images. J Indian Soc Remote Sens 46:157–167

Fauvel M, Chanussot J, Benediktsson JA (2009) Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas. EURASIP J Adv Signal Process 2009:1–14

Ghamisi P, Plaza J, Chen Y, Li J, Plaza AJ (2017) Advanced spectral classifiers for hyperspectral images: A review. IEEE Geosci Remote Sens Mag 5(1):8–32

Graña M, Veganzons M, Ayerdi B (n.d.) Hyperspectral remote sensing scenes. (GRUPO DE INTELIGENCIA COMPUTATIONAL (GIC))

Haq MA (2022a) CDLSTM: A novel model for climate change forecasting. Comput Mater Contin 71(2)

Haq MA (2022b) CNN based automated weed detection system using UAV imagery. Comput Sys Sci Eng 42(2)

Haq MA (2022c) Planetscope nanosatellites image classification using machine learning. Comput Syst Sci Eng 42(3)

Haq MA, Baral P, Yaragal S, Rahaman G (2020) Assessment of trends of land surface vegetation distribution, snow cover and temperature over entire Himachal Pradesh using MODIS datasets. Nat Resour Model 33(2):e12262

Haq MA, Alshehri M, Rahaman G, Ghosh A, Baral P, Shekhar C (2021a) Snow and glacial feature identification using Hyperion dataset and machine learning algorithms. Arab J Geosci 14:1–21

Haq MA, Baral P, Yaragal S, Pradhan B (2021b) Bulk processing of multi-temporal modis data, statistical analyses and machine learning algorithms to understand climate variables in the Indian Himalayan region. Sensors 21(21):7416

Haq MA, Rahaman G, Baral P, Ghosh A (2021c) Deep learning based supervised image classification using UAV images for forest areas classification. J Indian Soc Remote Sens 49:601–606

Haq MA, Ahmed A, Khan I, Gyani J, Mohamed A, Attia E-A, Pandi D (2022) Analysis of environmental factors using AI and ML methods. Sci Rep 12(1):13267

Haq MA, Ahsan A, Gyani J (2023) Implementation of CNN for plant identification using UAV imagery. Int J Adv Comput Sci Appl 14(4)

He X, Chen Y, Lin Z (2021) Spatial-spectral transformer for hyperspectral image classification. Remote Sens 13(3):498

Hong D, Gao L, Yao J, Zhang B, Plaza A, Chanussot J (2020) Graph convolutional networks for hyperspectral image classification. IEEE Trans Geosci Remote Sens 59(7):5966–5978

Hotelling H (1933) Analysis of a complex of statistical variables into principal components. J Educ Psychol 24(6):417

Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. Proc IEEE Conf Comput Vis Pattern Recognit:7132–7141

Jia S, Wang Y (2022) Multiscale convolutional transformer with center mask pretraining for hyperspectral image classification. arXiv preprint arXiv:2203.04771

Kambhatla N, Leen T (1993) Fast non-linear dimension reduction. Adv Neural Inf Proces Syst 6

Kavzoglu T, Colkesen I (2009) A kernel functions analysis for support vector machines for land cover classification. Int J Appl Earth Obs Geoinf 11(5):352–359

Lee H, Kwon H (2017) Going deeper with contextual CNN for hyperspectral image classification. IEEE Trans Image Process 26(10):4843–4855

Li W, Wu G, Zhang F, Du Q (2016) Hyperspectral image classification using deep pixel-pair features. IEEE Trans Geosci Remote Sens 55(2):844–853

Lunga D, Ersoy O (2012) Spherical stochastic neighbor embedding of hyperspectral data. IEEE Trans Geosci Remote Sens 51(2):857–871

Lv W, Wang X (2020) Overview of hyperspectral image classification. J Sens 2020

Ma X, Wang H, Wang J (2016) Semisupervised classification for hyperspectral image based on multi-decision labeling and deep feature learning. ISPRS J Photogramm Remote Sens 120:99–107

Manandhar R, Odeh IO, Ancev T (2009) Improving the accuracy of land use and land cover classification of Landsat data using post-classification enhancement. Remote Sens 1(3):330–344

Mangan P, Pandi D, Haq MA, Sinha A, Nagarajan R, Dasani T, Alshehri M (2022) Analytic hierarchy process based land suitability for organic farming in the arid region. Sustainability 14(8):4542

Mas J-F (1999) Monitoring land-cover changes: a comparison of change detection techniques. Int J Remote Sens 20(1):139–152

Maxwell AE, Warner TA, Fang F (2018) Implementation of machine-learning classification in remote sensing: an applied review. Int J Remote Sens 39(9):2784–2817

Mishra G, Sethi N, Agilandeeswari L, Hu Y-C (2023) Intelligent abstractive text summarization using hybrid Word2Vec and Swin transformer for long documents. Int J Comput Inf Syst Ind Manag Appl. 15:212–226

MohanRajan SN, Loganathan A (2021) Modelling spatial drivers for LU/LC change prediction using hybrid machine learning methods in Javadi Hills, Tamil Nadu, India. J Indian Soc Remote Sens 49:913–934

Mohanrajan SN, Loganathan A (2022) Novel vision transformer–based bi-LSTM model for LU/LC prediction—Javadi Hills, India. Appl Sci 12(13):6387

MohanRajan SN, Loganathan A (2023) A novel fuzzy Harris hawks optimization-based supervised vegetation and bare soil prediction system for Javadi Hills, India. Arab J Geosci 16(8):478

MohanRajan SN, Loganathan A, Manoharan P (2020) Survey on land use/land cover (LU/LC) change analysis in remote sensing and GIS environment: techniques and challenges. Environ Sci Pollut Res 27:29900–29926

Navin MS, Agilandeeswari L (2020a) Comprehensive review on land use/land cover change classification in remote sensing. Journal of spectral. Imaging 9

Navin MS, Agilandeeswari L (2020b) Multispectral and hyperspectral images based land use/land cover change prediction analysis: an extensive review. Multimed Tools Appl 79(39–40):29751–29774

Otter DW, Medina JR, Kalita JK (2020) A survey of the usages of deep learning for natural language processing. IEEE Trans Neural Netw Learn Syst 32(2):604–624

Palani A, Loganathan A (2023a) Multi-image feature map-based watermarking techniques using transformer. Int J Electr Electron Res 11:339–344

Palani A, Loganathan A (2023b) Semi-blind watermarking using convolutional attention-based turtle shell matrix for tamper detection and recovery of medical image. Expert Syst Appl 121903

Petitjean F, Kurtz C, Passat N, Gançarski P (2012) Spatio-temporal reasoning for the classification of satellite image time series. Pattern Recogn Lett 33(13):1805–1815

Puletti N, Camarretta N, Corona P (2016) Evaluating EO1-hyperion capability for mapping conifer and broadleaved forests. Eur J Remote Sens 49(1):157–169

Roy SK, Krishna G, Dubey SR, Chaudhuri BB (2019) HybridSN: exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification. IEEE Geosci Remote Sens Lett 17(2):277–281

Saqui D, Saito JH, Lucio AD, Ferreira EJ, Lima DC, Herrera JP (2016) Methodology for band selection of hyperspectral images using genetic algorithms and gaussian maximum likelihood classifier. In: 2016 international conference on Computational science and Computational intelligence (CSCI), IEEE

Sawant SS, Prabukumar M, Samiappan S (2019) A band selection method for hyperspectral image classification based on cuckoo search algorithm with correlation based initialization. In: 2019 10th workshop on hyperspectral imaging and signal processing: evolution in remote sensing (WHISPERS). IEEE

Segl K, Roessner S, Heiden U, Kaufmann H (2003) Fusion of spectral and shape features for identification of urban surface cover types using reflective and thermal hyperspectral data. ISPRS J Photogramm Remote Sens 58(1–2):99–112

Su H, Yong B, Du Q (2015) Hyperspectral band selection using improved firefly algorithm. IEEE Geosci Remote Sens Lett 13(1):68–72

Su H, Tian S, Cai Y, Sheng Y, Chen C, Najafian M (2017) Optimized extreme learning machine for urban land cover classification using hyperspectral imagery. Front Earth Sci 11:765–773

Teng W, Zhao J, Bai X (2022) Improved graph-based Semisupervised hyperspectral band selection. In: IGARSS. IEEE International Geoscience and Remote Sensing Symposium, IEEE, pp 2022–2022

Wang L, Peng J, Sun W (2019) Spatial–spectral squeeze-and-excitation residual network for hyperspectral image classification. Remote Sens 11(7):884

Wang M, Liu W, Chen M, Huang X, Han W (2021) A band selection approach based on a modified gray wolf optimizer and weight updating of bands for hyperspectral image. Appl Soft Comput 112:107805

Wang Y, Zhu Q, Ma H, Yu H (2022) A hybrid gray wolf optimizer for hyperspectral image band selection. IEEE Trans Geosci Remote Sens 60:1–13

Yadav CS, Pradhan MK, Gangadharan SM, Chaudhary JK, Singh J, Khan AA, Haq MA (2022) Multi-class pixel certainty active learning model for classification of land cover classes using hyperspectral imagery. Electronics 11(17):2799

Yang R, Zhou Q, Fan B, Wang Y (2022) Land cover classification from hyperspectral images via local nearest neighbor collaborative representation with Tikhonov regularization. Land 11(5):702

Zhang T, Tao D, Li X, Yang J (2008) Patch alignment for dimensionality reduction. IEEE Trans Knowl Data Eng 21(9):1299–1313

Zhang L, Zhang L, Tao D, Huang X (2013) A modified stochastic neighbor embedding for multi-feature dimension reduction of remote sensing images. ISPRS J Photogramm Remote Sens 83:30–39

Zhang M, Ma J, Gong M (2017) Unsupervised hyperspectral band selection by fuzzy clustering with particle swarm optimization. IEEE Geosci Remote Sens Lett 14(5):773–777

Zhang A, Ma P, Liu S, Sun G, Huang H, Zabalza J, Lin C (2019) Hyperspectral band selection using crossover-based gravitational search algorithm. IET Image Process 13(2):280–286

Zhang Z, Ma Q, Zhou H, Gong N (2022) Nested transformers for hyperspectral image classification. Journal of Sensors

Zhu J, Fang L, Ghamisi P (2018) Deformable convolutional neural networks for hyperspectral image classification. IEEE Geosci Remote Sens Lett 15(8):1254–1258

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.