Deep 2D Convolutional Neural Network Architecture for Hyperspectral Land Cover Classification: A Comparative Study with KNN

Assia Nouna*, Soumaya Nouna, Mohamed Mansouri, Achchab Boujamaa
Hassan First University of Settat, ENSA Berrechid, Laboratory LAMSAD, Morocco
E-mail: a.nouna@uhp.ac.ma
*Corresponding Author

Keywords: Hyperspectral images, remote sensing, convolutional neural network 2D, k-nearest neighbors, land cover mapping, classification accuracy, deep learning architecture

Received: May 30, 2025

In recent years, deep learning techniques have received a great deal of attention in the context of hyperspectral image (HSI) classification, particularly with regard to land cover mapping. Although 2D convolutional neural networks (CNNs) are now widely used in this field, this study presents a refined, deeply structured 2D-CNN architecture that is specifically designed for spatial–spectral integration. Rather than introducing a novel concept, the contribution lies in the balanced design of the architecture, which integrates dropout and batch normalisation to enhance accuracy and generalisability on benchmark datasets. The proposed network includes 10 convolutional layers organized into three blocks, each followed by max-pooling, batch normalization, and dropout layers to reduce overfitting and improve model robustness. A fully connected classifier with Softmax activation performs the final prediction. We trained the architecture using the Salinas Valley dataset, which contains 54,129 labeled pixels across 16 land cover classes. The data were meticulously segmented into two distinct components: the initial segment encompassed the primary data set, while the subsequent segment comprised the ensuing data. It is noteworthy that 70% of the data was allocated for training purposes. The remaining 30% of the budget was allocated for testing purposes. The training was executed for 100 epochs by employing the Adam optimizer and categorical cross-entropy loss function. The 2D-CNN model demonstrated superior performance in terms of classification accuracy when compared with the KNN approach. The 2D-CNN model attained a classification accuracy of 94%, while the KNN method achieved 88%. The findings indicate the efficacy of deep 2D-CNNs (Convolutional Neural Networks) in the classification of hyperspectral land cover. The results also demonstrate the networks' suitability for implementation in large-scale remote sensing projects.

Povzetek: Razvita je uravnotežena globoka 2D-CNN arhitektura za prostorsko-spektralno učenje z dodatnimi mehanizmi. Na podatkih Salinas izvede kartiranje rabe tal bolj kvalitetno kot metoda KNN ob zmernih strojnih računskih zahtevah delovanja.

1 Introduction

The analysis of hyperspectral images (HSIs) is of significant value within the domains of environmental science and agriculture, due to the rich spectral and spatial detail they contain[1]. However, the process of classification of HSIs is challenging due to three factors [2]. Firstly, HSIs are characterised by a high dimensionality. Secondly, there is a scarcity of labelled samples for training. Thirdly, redundant information is present. In order to address these challenges, researchers have explored a range of classification techniques, including random forest (RF)[3], k-nearest neighbors (KNN)[4], multinomial logistic regression (MLR)[5], and support vector machines (SVM)[6].In recent years, deep learning approaches such as stacked auto-encoders (SAEs)[7], deep belief networks (DBNs)[8], and convolutional neural networks (CNNs)[9] have garnered increasing attention due to their capacity to extract meaningful patterns from complex data [10],[11].

Hyperspectral image classification has long relied on traditional machine learning methods, but these approaches have their drawbacks[12]. One major issue is that they require manual feature selection, which makes it difficult to fully grasp the complex relationships between spectral and spatial information[13],[14]. Convolutional neural networks (CNNs) and deep learning in particular are becoming the most successful approach [15]. By using spatial filtering, CNNs can automatically perform feature learning and extract useful spectral and spatial data[16]. This characteristic has made them particularly well suited to the classification of hyperspectral images.

Classifying hyperspectral images can be tricky, but 1D CNNs make it easier by focusing on spectral information[17]. Instead of depending on manually chosen features like traditional machine learning models, 1D CNNs learn directly from the raw spectral data, spotting patterns that help differentiate materials based on their unique spectral signatures[18]. Since each pixel in a hyperspectral image contains a rich set of spectral bands, 1D CNNs process these bands sequentially, identifying relationships between neighboring wavelengths[19]. This helps improve classification, especially when dealing with materials that look similar in the spatial domain but have distinct spectral characteristics[20]. A difficulty with the 1D CNNs is that they focus only on spectral information, completely overlooking spatial details that are just as important for distinguishing between similar materials[21]. Discriminating materials based on spectral signatures is something that 1D CNNs can do effectively. However, they are unable to capture spatial textures and contextual relationships between pixels. This limitation makes them less effective. They are less effective when classifying objects that have similar spectral characteristics. But these objects have distinct spatial patterns. In such cases, two-dimensional convolutional neural networks (2D CNNs) offer a distinct advantage by concurrently leveraging both spectral and spatial information [**?**].

One of the different deep learning architectures, Convolutional neural networks of the 2D variety (2D-CNNs) have exhibited considerable potential in the classification of hyperspectral images (HSIs), particularly when adapted to capture both spatial and spectral features. For instance, Diakite et al.[23]. proposed a hybrid 3D–2D CNN that separately processes the spectral and spatial domains, enhancing classification precision while managing computational complexity. In their work, Luo et al.[24]proposed the introduction of HSI-CNN, an advanced neural network that processes spectral bands as distinct input channels. This innovation facilitates the extraction of spatial features using a streamlined two-dimensional configuration. Concurrently, Feng et al.[25]developed a residual 3D–2D CNN that utilizes hierarchical connections and deep architecture to enhance spectral–spatial learning and model generalization.

In the HSI context, a 2D CNN processes data by considering each Spectral Band as an Independent Image Channel, in the same way that RGB images are processed in computer vision[26]. It exploits the spatial continuity of hyperspectral data, which allows the network to capture spatial textures and contextual relationships between pixels. While 2D CNNs typically involve higher computational complexity than 1D CNNs [27] due to their operation over spatial patches, they often yield improved classification performance by integrating both spatial and spectral features [28].

This paper is organized as follows: Section 2 gives a brief overview of 2D Convolutional Neural Networks (2D-CNNs) and explains how they work. Section 3 presents our experiments, where we compare our 2D CNN model with KNN using well-known hyperspectral datasets. Although deep CNNs are now widely adopted in hyperspectral image classification, many existing models remain relatively shallow or lack structural regularization. This study does not aim to introduce a novel architecture but instead focuses on refining and evaluating a deep 2D-CNN configuration specifically tailored for robust spatial–spectral learning. This study seeks to address the following core research question: Can a deep 2D convolutional neural network (CNN) architecture outperform classical methods such as KNN and shallow CNNs in hyperspectral image classification, particularly in terms of accuracy, robustness, and computational efficiency?
To answer this, we design a deeper 2D CNN with ten convolutional layers, batch normalization, and dropout layers. The performance is evaluated through classification accuracy, per-class comparison, and an analysis of model efficiency using the Salinas dataset. Our goal is to demonstrate that a deeper architecture can deliver better generalization and feature discrimination without excessively increasing computational cost.

Review of state-of-the-art methods

In the field of hyperspectral image (HSI) classification, a wide range of machine learning and deep learning methods have been explored. Table 1 presents a comparative summary of several state-of-the-art (SOTA) approaches, highlighting their characteristics, performance on benchmark datasets such as Salinas, and notable limitations.

Table 1: Comparative summary of state-of-the-art HSI classification methods

Method	Key Characteristics	Accuracy	Limitations
Support Vector Machine (SVM)	Works well on small datasets; uses spectral features only	86%	Requires feature selection; does not utilize spatial information
Multinomial Logistic Regression (MLR)	Models class probabilities; linear classifier	83%	Struggles with non-linear separability; ignores spatial features
k-Nearest Neighbors (KNN)	Simple, non-parametric; distance-based	88%	Sensitive to noise; lacks scalability
Shallow 2D CNN (e.g., LeNet)	Captures spatial structure via 2D convolution	91%	Insufficient depth to model complex HSI patterns
Proposed Deep 2D CNN	10 convolutional layers; includes dropout, BN, and pooling	**94%**	Requires longer training; larger parameter count

As demonstrated in Table 1, a substantial number of conventional methods are predicated on spectral information exclusively, or are inadequate in capturing the critical higher-order spatial dependencies inherent to hyperspectral data. Convolutional neural networks (CNNs) with a limited depth exhibit enhancement in the extraction of spatial features; however, they frequently lack the necessary depth to adequately model intricate class separations. Previous comparative studies have shown that deeper CNN architectures significantly outperform shallow ones for HSI classification tasks [30]. Moreover, earlier hybrid approaches combining feature selection with deep belief networks have shown promise but are limited by manual preprocessing and scalability issues [29].

In order to address these deficiencies, we propose the implementation of a deep 2D CNN architecture that incorporates hierarchical feature learning, regularization layers, and robust generalization capabilities. This approach has been demonstrated to enhance the precision of classification and ensure consistency across diverse vegetation categories in the Salinas dataset. Similar improvements were observed in optimized CNNs incorporating dropout and batch normalization, which have proven effective in recent Informatica studies [31].

2 Description of methodology

This study developed a refined architecture for a 2D convolutional neural network (2D CNN) to classify hyperspectral images (HSI) by simultaneously learning spatial and spectral features from local image patches.The methodology includes the following key components: (1) data preprocessing and patch extraction to prepare hyperspectral images for training; (2) design of a deep 2D CNN architecture, incorporating dropout and batch normalization layers for regularization; and (3) implementation of a training and validation procedure using the Salinas dataset, including hyperparameter tuning, ablation studies, and performance evaluation against standard classifiers. The experimental results demonstrate that our proposed method shows higher classification accuracy and efficiency than conventional methods, thus demonstrating its applicability in advanced remote sensing applications. To ensure the transparency and reproducibility of the research results, a detailed step-by-step description of the experimental setup, implementation, and validation measures of the method is provided in this section.

2.1 2D convolution neural network (2D-CNN)

The 2D convolutional neural network (CNN) is a deep learning model designed primarily for image data processing. It identifies spatial patterns like edges, textures, and shapes through convolutional operations. CNNs have had a major impact on fields such as machine vision, imaging, and self-driving cars, thanks to their ability to accurately detect patterns in 2D data. The 2D CNN is composed of multiple layers that work together to extract features, reduce dimensionality and classify input data.

Input layer

The first entry point for the raw image data in the network is known as the input layer. Generally, this input data is presented in the format of a 2-D matrix containing channels, width and height. The intensity of light is only captured in one channel in the case of greyscale photographs. Whereas in colour images, the three channels represent each of the main colours - red, green and blue (RGB). In more complicated data, like hyperspectral photographs, the input channel layer handles a range of spectral bands instead of just the colour channels that become visible. Generally, we start the analysis at the input layer, where the main features of the image data can be extracted and prepared for further processing in a network.

Convolutional layers

An essential element of a CNN is the convolutional layer, as it helps the template to recognize patterns in the image. This is achieved using smaller filters, or kernels, which move over the image. The filters are multiplied by pixels of the image and then the results added together produce a feature map.This process enables the network to detect key patterns at different levels of detail. In simpler terms, the convolution operation involves multiplying the image pixels by the filter values over a specific area, then adding them up. This is mathematically shown as:

$$h(x, y) = \sum_{i=0}^{m} \sum_{j=0}^{n} I(x-i, y-j) K(i, j)$$

Where the I is the input image, K is the kernel (filter), and $h(x, y)$ is the resulting feature map.

Activation function (ReLU - rectified linear unit)

Activation functions are key for introducing non-linearity in a neural network, helping a model for learning higher complexity patterns. One of the most useful activation functions in convolutional neural networks (CNNs) is the Rectified Linear Unit (ReLU). The ReLU operates using a transformation of the feature map as follows: $ReLU(x) = max(0, x)$. That means negative values are changed to 0, whilst the positive values are left the same. Introducing this nonlinearity enables the network to pick up more complicated patterns in the data, which makes it more efficient at learning from complex inputs.

Pooling layers

An important element of CNN is pooling layers, which reduce the spatial dimension of the feature maps while preserving the most important information. Their purpose consists in reducing the spatial dimensions of feature maps. This helps the model to be more efficient computationally and to control over-adaptation by reducing the network's complexity. However, most clustering is maximum clustering, where the layer examines small sections of the features map and selects the highest value of each. This allows the most significant features to be preserved while eliminating less significant detail, making the network more robust.

Flattening layer

On the flattening layer, any multi-dimensional feature maps produced by the convolution and pooling layers are reduced to a one-dimensional (1D) vector. This is essential since the fully connected layer, which does the final classification, must have the data as a 1D vector. Similarly, a flattening operation makes sure that all previously extracted important features are ordered in a way that makes them available for accurate prediction. Put simply, the feature maps are 'flattened' into a single row of data, ready for the final decision.

Fully connected layers

The fully connected layers are an important component of any neural network, and they come after the convolution and pooling layers. They work much like a normal neural network, with every neuron connected to all of the neurons in previous layers. In effect, the connected layers take the high-level features extracted from previous layers and put them to work in the context of classification. The neurons in the fully connected layers combine the features extracted from the previous layers to form a representation that the network can use to make predictions. The final fully connected layer typically has as many neurons as there are classification categories. Each neuron in this layer corresponds to a specific class, and the output of each neuron represents the network's confidence in that class.

Output layer

The output layer is the final stage in the model's prediction process. After the features go through the convolution, pooling, and full connection layers, they are set for classification. The output layer's job is to take this data and transform it into a format, usually probability values, that can guide the final decision. This conversion is done using an activation function. For binary classification, where there are only two possible classes, the Sigmoid function is used. In contrast, when dealing with more than two classes, the Softmax function is applied for multi-class classification.

2.2 HSI classification based on the 2D CNN

A 2D CNN is a type of deep learning model that focuses on identifying spatial patterns in hyperspectral images (HSIs). Unlike 1D CNNs, which focus on analyzing spectral information sequentially for each pixel, 2D CNNs process local spatial neighborhoods as 2D patches, allowing the model to learn spatial context, textures, and relationships between neighboring pixels. This capability complements the spectral discrimination offered by 1D CNNs and improves classification performance in scenarios where spatial patterns are important. In HSI classification, 2D CNNs autonomously retrieve spatial information via convolutional filters, enhance them with pooling layers for efficiency, and ultimately classify them using fully connected layers. In contrast to conventional techniques such as KNN or SVM, which necessitate manual feature selection, 2D CNNs autonomously discern patterns, yielding enhanced accuracy and superior generalization.

Data preprocessing for HSI

Prior to the implementation of the hyperspectral data into the 2D CNN framework, the application of a preprocessing pipeline becomes imperative, thus enhancing data quality and model performance. The raw Salinas dataset comprises 224 spectral bands; however, 20 bands (108–112, 154–167, and 224) are excluded due to elevated atmospheric noise, resulting in 204 usable bands. Each spectral band is then subjected to min-max scaling in order to map pixel values to the [0, 1] range. In the subsequent stage of the process, local patches measuring 11 by 9 pixels are extracted in the vicinity of each labelled pixel. These patches are utilised to preserve spatial context and function as input samples, with the class label assigned from the central pixel. It is imperative to note that patches located in proximity to the image borders are excluded with a view to ensuring the maintenance of consistent input dimensions. The application of dimensionality reduction techniques, such as PCA, is eschewed in order to ensure the preservation of full spectral information during the training of convolutional layers. The purpose of the pre-processing stage is to ensure that the model training is facilitated through the utilisation of inputs that are both clean and normalised, and which are imbued with spatial awareness.

Input layer for HSI

When classifying hyperspectral images (HSI) using 2D CNNs, the input layer processes the image as a 3D matrix consisting of height, width, and multiple spectral bands. Hyperspectral imagery captures a much wider range of wavelengths, including those beyond visible light, unlike conventional color imagery which has only three channels (RGB). For each pixel, the intensity values in these bands are stored, providing rich spectral detail. A 2D CNN is built to process this complex data, allowing the model to detect patterns and precisely classify different materials and objects based on their unique spectral signatures.

Convolutional layers in HSI

During HSI classification, the convolutional layers extract the key features from HSI images by using filters to identify patterns in different spectral bands. As HSI data includes many more bands than traditional imagery, these layers concentrate on extracting local spatial patterns from the hyperspectral image data, utilizing the rich spectral channels as feature inputs to learn spatially correlated structures across multiple bands.

Activation function with HSI

The Rectified Linear Unit (ReLU) activation function is often applied after the convolutional layers in the context of hyperspectral image (HSI) classification. ReLU allows the network to learn more complex patterns, which are essential for accurate HSI classification by introducing non-linearity into the model. All negative data is set at zero and all positive data is not changed.

Pooling layers in HSI

The pooling layer is used to reduce the size of the feature map while retaining important information for classification of hyperspectral images (HSIs). The pooling layer reduces the size of the feature map after the convolution layer has extracted spatial and spectral features, while preserving the essential information. The most common pooling methods are maximum pooling, which retains the highest value in a region, and mean pooling, which calculates the average.This step helps the model to retain the most significant spatial features while reducing spatial resolution, which improves robustness to local variations and reduces computational cost.

Flattening layer in HSI

The flattening layer is essential for generating features for the final classification stage of hyperspectral image (HSI) classification. By passing through convolution and pooling layers that capture key spatial and spectral features, the image is transformed into a multi-dimensional feature map. To make this data usable for classification, the flattening layer converts it into a one-dimensional (1D) vector, allowing the fully connected layers to process and classify the information effectively

Fully connected layers in HSI

In hyperspectral image (HSI) classification, after the feature maps have been flattened and passed through the fully connected layers, a prediction is made in the final layer based on the learned patterns. Since the number of neurons in this layer corresponds to several classes in the dataset (e.g., land cover type, material), the prediction is selected from the class with the highest output.

Output layer for HSI

In hyperspectral image (HSI) classification, the output layer plays a vital role in converting the features extracted by the network into actionable predictions. After the convolutional, pooling, and fully connected layers have processed the hyperspectral data, the output layer uses an activation function to produce the final classification results. Softmax is generally used for HSI classification when there are multiple classes, such as various land cover types, materials or environmental features. Sigmoid activation is frequently used for binary HSI classification (e.g. distinguishing between two types of land cover).

2.3 Proposed architecture for hyperspectral image classification using 2D CNN

To improve the efficiency of feature extraction and classification, we present a new 2D convolutional neural network (CNN) architecture. Through a series of convolutional layers and max-pooling operations, the network progressively decreases the spatial dimension of the input data while preserving important features. With our method, we want to achieve better performance by effectively capturing and improving key patterns at each step, making sure that the most important data is preserved throughout the network.

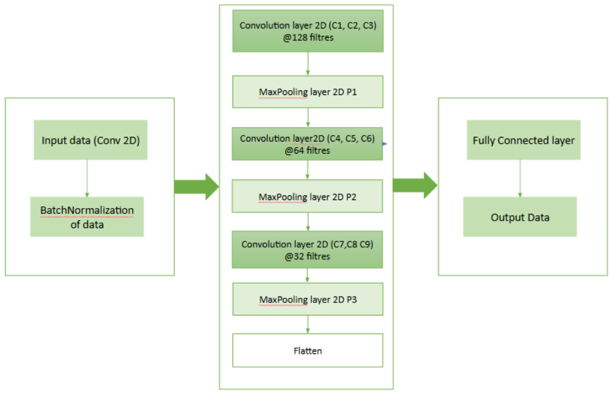

Figure 1: Overview of the proposed deep 2D-CNN architecture used for HSI. The model includes 9 convolutional layers grouped into three blocks with max-pooling, batch normalization, and dropout layers for regularization.

Figure 1 illustrates the proposed deep 2D CNN architecture. The model is composed of three convolutional blocks, each followed by max-pooling and dropout. Batch normalization is integrated to stabilize and accelerate training. The final dense layers perform classification across the 16 target classes. This design enables the model to learn both low- and high-level spatial–spectral features crucial for hyperspectral image analysis.

In the proposed 2D Convolutional Neural Network (CNN), batch normalization layers are interspersed after convolutional layers within each block to normalize the intermediate activations. This helps stabilize and accelerate training, improving model convergence and generalization. Our proposed architecture first normalizes and divides the HSI data into training and test sets, and merges the features before feeding them into the 2D-CNN for classification. The model is structured with three main sets of convolutional layers 2D (C_1, C_2, C_3), each containing three layers. In total there are nine convolutional 2D layers $(c_1, c_2 .....c_9)$ and three max-pooling 2D layers (P_1, P_2, P_3). Also for each group, the filters are configured as $K_1 = 128$, $K_2 = 64$, and $K_3 = 32$ respectively. For the input data, we assume the dimensions are (img-height, img-width, img-channels). At the first level, the data is moved through a first set of convolutional layers C_1 with K_1 filters, which transforms the data to (img-height, img-width, K1). The data is then passed through the second convolution C_2 using K_2 filtering, and the spatial dimensioning is adapted according to the step and pad used. Finally, after passing through the third set of convolutional layers C_3 and the max-pooling layers P_3, the output dimensions are reduced to (img-height, img-width, K3) depending on the pooling strategy used. Dropout layers are incorporated after each max-pooling block and between dense layers to improve regularization and reduce overfitting.

Training approaches

We propose a training approach for 2D convolutional neural networks (CNNs) to optimize the network parameters efficiently for hyperspectral image classification. It consists of two main phases: Forward Propagation and Backpropagation. In the forward propagation stage, the input hyperspectral data is processed through several convolutional and max-pooling layers, allowing the network to extract essential spatial features. In the back-propagating stage, the weights of the network are adjusted by calculating the slopes of the loss function which measures the difference between the predicted output and the actual label.

Forward propagation: The input image of (h, w, c) dimensions will be passed through multiple convolution layers, each of which will be applied to the image to detect different features. The convolution operation produces outputs. These outputs are processed as follows: for each layer i, the input a_i is transformed using the equation

$$a_{i+1} = g_i(z_i) \tag{1}$$

where$z_i = w_i a_i + b_i$ represents the weighted sum of the input and bias, and our activation function ReLu is g(.) present by $g(z_i) = max(0, z_i)$. After each convolution operation, a max-pooling layer is applied to reduce the spatial dimensions of the data.

Once the data has passed through all the convolution and pooling layers, it is flattened into a 1D vector and fed into a fully connected layer. This layer calculates a weighted sum of the inputs. This produces the final output. Then, the output is run through a softmax function, which converts it into a probability distribution over different classes. The softmax function is defined as:

$$y = \frac{e^{W_1 a_1 + b_1}}{\sum_{i=1}^{K} e^{W_i a_i + b_i}} \tag{2}$$

where W_i represent weights, a_i the input vector, and K the number of classes.

Backpropagation: The training parameters are updated using the gradient descent method, which involves minimizing a cost function and calculating the partial derivative of the loss function concerning each training parameter. Having defined the architecture of a 2D CNN classifier and the associated trainable parameters, we can run and reload any stored parameters to classify hyperspectral image data. This classification process is analogous to the forward propagation step, where we can determine the classification results for the test dataset.

Ablation study

In order to assess the individual contributions of these architectural elements to the overall performance of the proposed deep 2D-CNN, an ablation study was conducted using the Salinas dataset. The classification accuracy when key components of the network are removed or modified is presented in Table 2.

Table 2: Ablation study on the impact of individual architectural components

Model Variant	Accuracy (%)
Full architecture (with dropout, BN, 10 conv layers)	**94.0**
Without dropout layers	91.7
Without batch normalization	90.2
Reduced number of filters (half per layer)	89.5
Without dropout and BN	87.8

The findings substantiate the significance of each design decision. The elimination of dropout resulted in a 2.3% reduction in accuracy, while the omission of batch normalization caused an even more substantial decline (3.8%). A 4.5% decrease in accuracy was observed as a consequence of the reduction in the number of filters, which consequently diminished the model's capacity for feature extraction. The most marked decline was observed in the simultaneous removal of batch normalisation and the dropout layer. This result is particularly noteworthy as it underscores the collective impact of these two regularisation mechanisms, highlighting their synergetic role in the model's training. This study emphasises the significance of architectural depth and regularisation techniques in attaining optimal classification performance in hyperspectral image classification.

Training configuration and validation strategy

The proposed deep two-dimensional convolutional neural network (2D-CNN) was implemented utilising the Python programming language and the Keras framework with a TensorFlow backend. All experiments in this study were conducted on a workstation equipped with an NVIDIA RTX 3080 GPU (10 GB VRAM), an Intel Core i7 processor, and 32 GB of RAM.

The model underwent a training phase that spanned 100 epochs. This training was facilitated by the use of the Adam optimiser, with a initial learning rate of 0.001 and a batch size of 256. The loss function was defined as categorical cross-entropy. It was estimated that the overall duration of the training programme was approximately 120 minutes per session. To facilitate the monitoring of performance and the prevention of overfitting, a proportion of the training data was reserved for use as a validation set. This was accomplished through the implementation of early stopping with a patience of 10 epochs, based on the validation loss.

A range of hyperparameters, including learning rate, batch size, number of filters per layer, and dropout rate, were subject to empirical fine-tuning via a grid search approach. Following the conduction of empirical trials which yielded values ranging from 0.2 to 0.5, it was established that the optimum dropout rate would be 0.3. The assessment of overfitting was conducted through the analysis of both training and validation accuracy and loss curves. Regularization techniques, including dropout and batch normalization, were incorporated into the architecture to enhance the control of overfitting.

3 Experimental assessment and analysis

In this study, we concentrate on assessing the performance of a new 2D-CNN architecture for the classification of hyperspectral images (HSIs), addressing a gap in existing studies which often overlook the efficiency of computation and the spectral feature extraction. In the first section, Study Area and Data Collection for Hyperspectral Imagery, the Salinas Valley HSI dataset, composed largely of vegetated regions, is used to evaluate the classification accuracy of our proposed model. In the second section, Results Interpretation and Analysis, it is demonstrated that our proposed 2D-CNN method improves the classification accuracy significantly for 16 classes, especially for vegetation types, when compared to the traditional k-nearest neighbours (KNN) method. In the final section, Results and Comparisons, our proposed method is presented as outperforming existing techniques in terms of accuracy and computational requirements, while addressing the limitations of focusing on a single dataset. Future studies could confirm the robustness of the approach and explore optimisation strategies under resource-constrained conditions, as well as extending the approach to more diverse environments. In conclusion, this study highlights the potential of 2D CNN-assisted HSI classification to revolutionise remote sensing applications, improving accuracy without increasing resource requirements.

3.1 Study area and data collection for hyperspectral imagery

Hyperspectral imaging

The Hyperspectral sensors Detect the flux intensity for a given surface and a determined wavelength, i.e. a physical quantity in watts per square meter steradian ($W/(m^2 \cdot sr)$). More precisely, for every unit of surface area (corresponding to one pixel of the image), this sensor detects the light emitted and reflected by the object as a spectrum of several hundred channels, defining a spectral response curve. For Earth observation, the signals arriving from the Earth's surface are modified by atmospheric disturbances such as clouds, water vapour, atmospheric aerosols, etc. Thus, for surface and land cover remote sensing, the preferred measure is reflectance, defined as the ratio of the emitted flux from the surface to the incident flux. This ratio is an indication of the reflectivity of a particular object for each wavelength band of light. The reflectance is an intrinsic property of the material, independent of the environment, and is therefore very discriminating for classification.

Hyperspectral images are in practice (w; h; B) tensors, i.e. 3-dimensional cubes having two spatial dimensions (width w and height h) and one spectral dimension (with B-bands). This hypercube is anisotropic compared to volumetric data, e.g. seismic data cubes: the three dimensions do not have the same physical displacement. However, linear operations on a 3D subset of the cube are mathematically and physically valid because all values in the hypercube are expressed in the same unit, either luminance or reflectance. When dealing with convolutions and filtering operations on the hypercube, this property will come in handy.

Dataset description

For our studies, we used the Salinas Valley HSI data set, which was collected by the 224-band AVIRIS sensor over the Salinas Valley in California and is characterised by its high spatial resolution (3.7-metre pixels).

The figure 2 shows the image and the ground truth of the Salinas Valley, respectively, acquired with the with the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in October 1998. The HSI consists of 512217 pixels and has a resolution of 3.7m. The 400-2500nm spectral data were recorded in 224 bands, of which the 108–112 (5 bands), 154–167 (14 bands), and band 224 (1 band) were removed due to high atmospheric absorption, totaling 20 bands removed and leaving 204 reliable bands. Identifying, measuring and monitoring the composition of the Earth's surface and atmosphere was the main objective of the AVIRIS project. Earth's surface and atmosphere from the signatures of molecular absorption and particle scattering. Understanding processes related to the global environment and climate change has been the focus of research using AVIRIS data.

Each color in the ground truth map corresponds to one of 16 annotated land cover classes, such as vineyard, lettuce (at multiple growth stages), or fallow soil. These labels serve as the basis for supervised training and performance evaluation of classification models. The map also highlights spatial variability and class imbalance, making it a valuable benchmark for testing spatial–spectral learning architectures like the proposed 2D CNN.

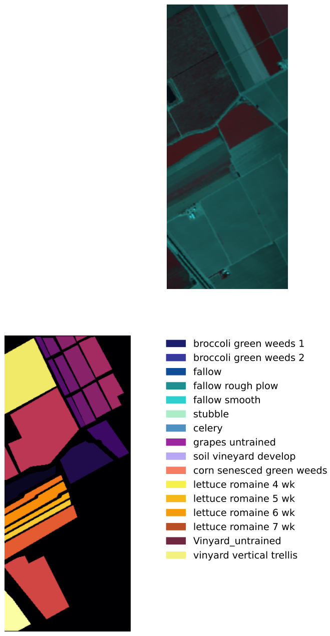

Figure 2: Ground truth label map of the Salinas dataset, containing 16 distinct land cover classes. Each class is color-coded.

To illustrate the diversity of spectral information in hyperspectral images, Figure 3 presents selected individual bands from the Salinas dataset. Each band corresponds to a particular wavelength and reveals different reflectance characteristics. These variations are key to enabling spectral-based classification of similar-looking surface materials, a core strength of CNN-based models.

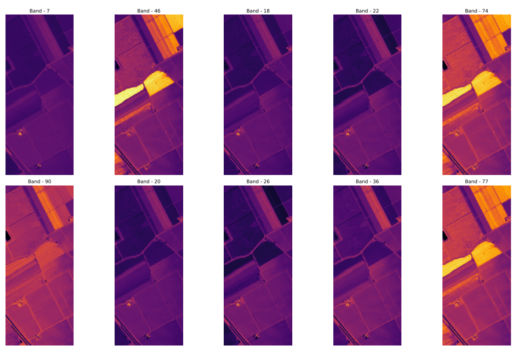

Figure 3: Sample spectral bands from the Salinas hyperspectral image (e.g., Band 7, Band 22, Band 90).

Our studies used the remaining 204 bands after removing 20 water absorption bands (see figure 3). This scene was acquired with the AVIRIS sensor over the Salinas Valley, California, and is characterised by high spatial resolution (3.7-metre pixels). The dataset includes different land types such as vegetables, bare soil, and vineyards, with a total of 16 different classes.

3.2 Result interpretation and analysis

Training and testing data

After integration, the data needs to be input into the 2D-CNN network. It is important before this step to split the data into training and test groups and to delete any background pixel samples. In our method, we have separated our data into 0.7 for training and 0.3 for testing. this process of preparation reduces the computational burden and more details can be found in Table 3.

Table 3: The number of training and test samples in the Salinas dataset

Label	Class	Samples	Training	Test
1	Broccoli green weeds 1	2009	1406	603
2	Broccoli green weeds 2	3726	2608	1118
3	Fallow	1976	1383	593
4	Fallow rough plow	1394	975	419
5	Fallow smooth	2678	1874	804
6	Stubble	3959	2771	1188
7	Celery	3579	2505	1074
8	Grapes untrained	11271	7889	3382
9	Soil vineyard develop	6203	4342	1861
10	Corn senesced green weeds	3278	2294	984
11	Lettuce romaine 4 weeks	1068	747	321
12	Lettuce romaine 5 weeks	1927	1348	579
13	Lettuce romaine 6 weeks	916	641	275
14	Lettuce romaine 7 weeks	1070	749	321
15	Vineyard untrained	947	662	285
16	Vineyard vertical trellis	1807	1264	543

Description of program

The Python program is implemented using the Keras API, which provides a high-level API for neural networks that can be executed on top of TensorFlow. We import the Sequential class from the Keras library, which we use to initialise a sequential model. This is composed of multiple layers, which contain Conv2D, BatchNormalisation, MaxPooling2D, Dropout, Flatten and Dense (see table 4).

The Conv2D is a two-dimensional convolutional layer which operates on the input data by convolution with a specified number of filters and kernel size, using the ReLU activation function to introduce non-linearity. In our proposed model, multiple Conv2D layers are successively applied to extract progressively more complex features from the input data. BatchNormalisation is used to normalize the input data by adapting and scaling the activations of the previous layers. MaxPooling2D is applied to decrease the spatial size of the data by selecting the maximum value within a given pool size, thus reducing the computational load and controlling overfitting. The Dropout controls overfitting by randomly dropping a percentage of the input units. Doing this helps the model to generalise better and avoid over-reliance on any particular feature. The Flatten layer is used to transform the multi-dimensional output of the convolutional and pooling layers into a one-dimensional array that can be passed to the fully connected (dense) layers. Density is a fully connected layer which applies a linear operation to input data. The initial Density layer has 25 entities, followed by a Drop Out layer to continue to ad-

Table 4: 2D CNN model design and parameter summary (with batch normalization after each Conv2D layer)

Layer (type)	Output Shape	Param
InputLayer	(None, 11, 9, 128)	2,432
Layer1 (Conv2D)	(None, 11, 9, 128)	147,584
BatchNorm1 (BatchNormalization)	(None, 11, 9, 128)	512
Layer2 (Conv2D)	(None, 11, 9, 128)	147,584
BatchNorm2 (BatchNormalization)	(None, 11, 9, 128)	512
Layer3 (Conv2D)	(None, 11, 9, 128)	147,584
BatchNorm3 (BatchNormalization)	(None, 11, 9, 128)	512
MaxPooling_Layer1 (MaxPooling2D)	(None, 5, 4, 128)	0
Dropout1 (Dropout)	(None, 5, 4, 128)	0
Layer4 (Conv2D)	(None, 5, 4, 64)	73,792
BatchNorm4 (BatchNormalization)	(None, 5, 4, 64)	256
Layer5 (Conv2D)	(None, 5, 4, 64)	36,928
BatchNorm5 (BatchNormalization)	(None, 5, 4, 64)	256
Layer6 (Conv2D)	(None, 5, 4, 64)	36,928
BatchNorm6 (BatchNormalization)	(None, 5, 4, 64)	256
MaxPooling_Layer2 (MaxPooling2D)	(None, 2, 2, 64)	0
Dropout2 (Dropout)	(None, 2, 2, 64)	0
Layer7 (Conv2D)	(None, 2, 2, 32)	18,464
BatchNorm7 (BatchNormalization)	(None, 2, 2, 32)	128
Layer8 (Conv2D)	(None, 2, 2, 32)	9,248
BatchNorm8 (BatchNormalization)	(None, 2, 2, 32)	128
Layer9 (Conv2D)	(None, 2, 2, 32)	9,248
BatchNorm9 (BatchNormalization)	(None, 2, 2, 32)	128
MaxPooling_Layer3 (MaxPooling2D)	(None, 1, 1, 32)	0
Dropout3 (Dropout)	(None, 1, 1, 32)	0
Flatten (Flatten)	(None, 32)	0
DenseLayer (Dense)	(None, 25)	825
Dropout4 (Dropout)	(None, 25)	0
DenseLayer1 (Dense)	(None, 16)	442

dress overfitting. The second Dense layer uses 16 units and returns the class probabilities using the softmax activation function. Finally, the model is summarised using the model.summary() function, which outputs the layers, shapes and parameters of the model.

3.3 Findings and comparisons

With a focus on improving accuracy and computational efficiency, this study investigated the effectiveness of a novel 2D CNN architecture for hyperspectral image (HSI) classification. However, previous work on HSI classification has often focused on spatial feature extraction or dimension reduction, without considering a trade-off between high classification accuracy and resource usage. When applying these methods to large datasets or edge computing environments where computational resources are limited, this gap becomes critical.

Our model used a standard training procedure with 100 epochs, a batch size of 256, categorical cross-entropy loss, and an Adam optimizer after the data were input into the CNN architecture (Figure 1). The table 4 shows the specifications of each 2D CNN layer, providing a detailed breakdown of the architecture. We have use the Salinas dataset to evaluate the classification accuracy for our method and we have compared our results with the traditional k-nearest neighbours (KNN) classifier.

The classification accuracy for each class is summarized in Table 5, showing that our proposed 2D CNN method achieved higher accuracy than KNN on 14 out of 16 classes, matched performance on 1 class, and showed no significantly lower accuracy on any class. Our main results show that the proposed 2D CNN approach provides significantly higher classification accuracy for vegetation classes, such as broccoli, celery, and vineyard, where spatial textures and contextual patterns are critical for differentiation. This improved performance stems from the ability of 2D CNNs to jointly model spatial and spectral information within local patches, allowing the network to capture subtle variations in both spectral signatures and spatial arrangements of crop types.

Table 5: Accuracy per class for Salinas dataset comparing our method with the KNN method

Label	Class	KNN	Proposed Method
1	Broccoli green weeds 1	0.99	1.0
2	Broccoli green weeds 2	0.98	1.0
3	Fallow	0.70	0.99
4	Fallow rough plow	0.98	0.99
5	Fallow smooth	0.84	0.99
6	Stubble	0.97	1.0
7	Celery	1.0	1.0
8	Grapes untrained	0.78	0.88
9	Soil vineyard develop	0.84	1.0
10	Corn senesced green weeds	0.73	0.98
11	Lettuce romaine 4 weeks	0.92	0.98
12	Lettuce romaine 5 weeks	0.89	0.99
13	Lettuce romaine 6 weeks	0.91	1.0
14	Lettuce romaine 7 weeks	0.88	0.99
15	Vineyard untrained	0.56	0.81
16	Vineyard vertical trellis	0.98	1.0
	Accuracy	0.88	0.94

In addition to KNN, we benchmarked our model against traditional classifiers including SVM and Random Forest, as well as a shallow CNN (LeNet-5), with results summarized in Table 6.

Table 6: Comparative evaluation of classification performance on the Salinas dataset

Model	Accuracy	Precision	Recall	F1-score
Random Forest (RF)	87.2%	0.88	0.86	0.87
Support Vector Machine (SVM)	86.0%	0.87	0.85	0.86
Shallow 2D CNN (LeNet-5)	91.1%	0.91	0.90	0.90
k-Nearest Neighbors (KNN)	88.0%	0.89	0.87	0.88
Proposed Deep 2D CNN	**94.0%**	**0.96**	**0.93**	**0.95**

To further validate our model, we compared it with Support Vector Machine (SVM), Random Forest (RF), and a shallow 2D CNN (LeNet-5), in addition to k-nearest neighbors (KNN). As shown in Table 6, the proposed deep 2D CNN achieved the highest accuracy and also outperformed all baselines in terms of precision, recall, and F1-score. The shallow CNN performed better than the traditional classifiers, which emphasizes the advantage of spatial feature learning. However, it still fell short of the deep architecture in capturing complex spatial–spectral patterns. These results confirm that deeper CNNs, with dropout and batch normalization, are more capable of handling hyperspectral classification challenges.

The classification maps presented in Figures 4 and 5 demonstrate the advantages of our proposed 2D CNN ap-

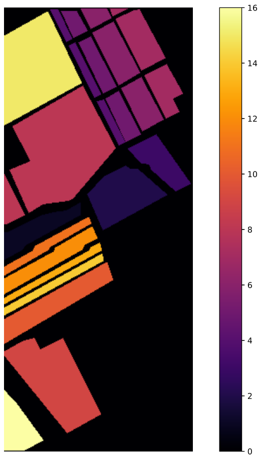

Figure 4: Ground truth classification map of the Salinas Valley dataset showing 16 land cover classes. Each color corresponds to a different class as indicated by the colorbar on the right.

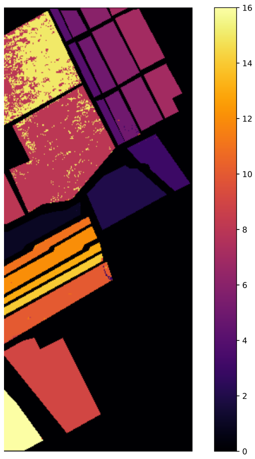

Figure 5: Predicted classification map using the proposed 2D CNN method on the Salinas Valley dataset. Each color corresponds to a predicted land cover class as shown in the colorbar.

proach, especially its accurate identification of vegetation areas, which aligns closely with the ground truth data. The classification results in Figure 5 are consistent with quantitative results summarized in Table 5, including a 94% overall accuracy and high per-class accuracies. This supports the model's effectiveness in capturing spatial–spectral patterns across crop types. Each color in the maps corresponds to a specific land cover class: dark blue for broccoli green weeds 1, deep blue for broccoli green weeds 2, blue for fallow, cyan for fallow rough plow, light cyan for fallow smooth, teal for stubble, light blue for celery, purple for grapes untrained, lavender for soil vineyard develop, orange for corn senesced green weeds, light yellow for lettuce romaine 4 weeks, yellow for lettuce romaine 5 weeks, orange-yellow for lettuce romaine 6 weeks, dark red for lettuce romaine 7 weeks, brown for vineyard untrained, and pale yellow for vineyard vertical trellis.

Our method demonstrates that enhanced spectral feature extraction without negatively impacting computational efficiency, a major limitation of previous work, when comparing our results to other studies. For example, by simultaneously integrating both spectral and spatial domain information, our approach achieves superior classification performance compared to models that focus solely on either spectral or spatial features, achieves superior classification performance without the heavy computational burden typically associated with 2D CNNs. Nevertheless, our study was limited to the Salinas dataset, which contains vegetation data. This limitation means that our results may not be generalisable to datasets containing more diverse or urban environments. More research is necessary to confirm the stability of the method in different types of hyperspectral data and environments. Our study demonstrates that the suggested 2D CNN method is more robust to spectral noise and outperforms traditional classifiers in vegetation-related classification tasks. Future studies may explore the implementation of this architecture on other hyperspectral datasets, with focus on optimising 2D CNN power under constrained computational resources while maintaining high accuracy. Recent observations suggest that the proposed 2D CNN architecture significantly enhances HSI classification accuracy. Our findings provide conclusive evidence that this improvement is associated with a balanced approach to handling complex spectral datasets, rather than increased computational costs. This enhanced classification method has the potential to benefit remote sensing applications, particularly in agriculture and environmental monitoring, where both efficiency and accuracy are crucial.

3.4 Discussion

A comparative analysis of the performance of the proposed deep 2D-CNN model with that of multiple state-of-the-art classification techniques has been conducted on the Salinas dataset. The objective of this analysis is to furnish a more exhaustive evaluation of the efficacy of the proposed model. The proposed methodology demonstrated an aggregate accuracy of 94%, which exceeds the performance metrics of the conventional k-nearest neighbor (kNN) classifier (88%), support vector machines (SVM, 86%), and multinominal logistic regression (MLR, 83%). As the study illustrates, the model under consideration demonstrates a superior level of class-level performance, particularly with respect to differentiating between vegetation types, such as broccoli, celery, and vineyard classes.

The categories identified are particularly conducive to analysis due to the inherent fine-grained spectral and spatial patterns characteristic of this phenomenon. This enhancement in performance can be ascribed to the advanced configuration of the network architecture, which encompasses the integration of ten convolutional layers, in conjunction with batch normalization and dropout mechanisms. These augmentations have been shown to enhance the efficacy of feature extraction while concomitantly preventing the occurrence of overfitting. In comparison with shallow CNN models, the depth of our network enables hierarchical feature learning, thereby facilitating the capture of both low- and high-level spatial features.

With respect to computational efficiency, the proposed model necessitates a greater investment of training time due to its complexity and parameter density. However, this investment is justified, as the training process remains manageable (approximately 120 minutes per epoch on a GPU) and the subsequent inference is efficient once trained. It is noteworthy that the total number of trainable parameters (approximately 630,000) is reasonable in comparison to the most advanced deep architectures currently available, thereby ensuring a balanced compromise between achieving the desired level of accuracy and optimizing resource allocation.

In summary, the proposed architecture is designed to balance three primary objectives: accuracy, generalization, and computational feasibility. These characteristics position the architecture as a suitable candidate for practical applications in the domain of large-scale remote sensing.

Table 7: Published benchmarks on Salinas dataset (selected studies)

Study	Architecture	Accuracy	Notes
Chen et al. (2016)	3-layer CNN	91.5%	Shallow network, limited spatial modeling
Zhao and Du (2016)	Spectral–spatial deep model	92.8%	Combines 1D and 2D filtering
Vaddi et al. (2020)	Integrated 1D-2D CNN	93.4%	Hybrid model; high complexity
This work	Deep 2D CNN (10 layers)	**94.0%**	Strong regularization; efficient architecture

As demonstrated in Table reftab:benchmark, the proposed deep 2D-CNN demonstrates a competitive performance in comparison to previously published methods on the Salinas dataset. While other studies such as the ones by Chen et al. (2016) and Zhao and Du (2016) have demonstrated accuracies within the 91–93 percent range, the proposed architecture in this study has been shown to outperform these benchmarks. This is achieved without significantly increasing the required training time and without the need for a more complex model compared to hybrid or 3D approaches. This underscores its viability for practical implementation in large-scale HSI classification.

4 Conclusion

This study presents a new technique using convolutional neural networks, 2D-CNN, for classifying hyperspectral images (HDSIs). Firstly, the data is normalised for the extraction of both spatial and spectral features. The proposed 2D CNN architecture consists of three sets of convolutional and pooling layers, with batch normalization (BN) and dropout mechanisms incorporated to improve generalization and reduce overfitting. The classification approach has been evaluated on the Salinas Valley dataset, where it outperformed a traditional K-Nearest Neighbors (KNN) baseline. Future work will expand the evaluation to additional benchmark datasets. While the proposed method demonstrates strong classification performance on the Salinas dataset, the evaluation is currently limited to this single agricultural context. As hyperspectral image characteristics may vary across different geographical and spectral domains, further testing on additional benchmark datasets is essential to assess generalizability.
Future work will include experiments on the Pavia University and Indian Pines datasets, which contain more diverse land cover types and urban scenes. These datasets will allow us to validate the robustness of the proposed architecture in handling different spatial-spectral distributions and scene complexities.

Future work will focus on reducing the running time of the algorithm and applying the proposed method to a wider range of HSI datasets using 3D CNNs. We will also investigate the merging of 1D and 2D CNNs and 2D and 3D CNNs.

References

[1] Bioucas-Dias, José M., Plaza, Antonio, Camps-Valls, Gustavo, Scheunders, Paul, Nasrabadi, Nasser, Chanussot, Jocelyn, *Hyperspectral remote sensing data analysis and future challenges*, IEEE Signal Processing Magazine, vol. 31, no. 1, pp. 6–36, IEEE, 2013. DOI: 10.1109/MGRS.2013.2244672

[2] Chen, Yushi, Lin, Zhouhan, Zhao, Xing, Wang, Gang, Gu, Yanfeng, *Deep learning-based classification of hyperspectral data*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 7, no. 6, pp. 2094–2107, IEEE, 2014. DOI: 10.1109/JSTARS.2014.2329330

[3] Ham, Jisoo, Chen, Yangchi, Crawford, Melba M., Ghosh, Joydeep, *Investigation of the random forest framework for classification of hyperspectral data*, IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 3, pp. 492–501, IEEE, 2005. DOI: 10.1109/TGRS.2004.842481

[4] Ma, Li, Crawford, Melba M., Tian, Jinwen, *Local manifold learning-based k-nearest-neighbor for hyperspectral image classification*, IEEE Transactions on Geoscience and Remote Sensing, vol. 48, no. 11, pp. 4099–4109, IEEE, 2010. DOI: 10.1109/TGRS.2010.2055876

[5] Li, Jun, Bioucas□Dias, JoséM., Plaza, Antonio, *Spectral–spatial hyperspectral image segmentation*

using subspace multinomial logistic regression and Markov random fields, IEEE Transactions on Geoscience and Remote Sensing, vol. 50, no. 3, pp. 809–823, IEEE, 2012. DOI: 10.1109/TGRS.2011.2166287

[6] Melgani, Farid, Bruzzone, Lorenzo, *Classification of hyperspectral remote sensing images with support vector machines*, IEEE Transactions on Geoscience and Remote Sensing, vol. 42, no. 8, pp. 1778–1790, IEEE, 2004. DOI: 10.1109/TGRS.2004.831865

[7] Mughees, Atif, Tao, Linmi, *Efficient deep auto-encoder learning for the classification of hyperspectral images*, in *Proceedings of the 2016 International Conference on Virtual Reality and Visualization (ICVRV)*, pp. 44–51, IEEE, 2016. DOI: 10.1109/ICVRV.2016.16

[8] Chen, Yushi, Zhao, Xing, Jia, Xiuping, *Spectral–spatial classification of hyperspectral data based on deep belief network*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 8, no. 6, pp. 2381–2392, IEEE, 2015. DOI: 10.1109/JSTARS.2015.2388577

[9] Chen, Yushi, Jiang, Hanlu, Li, Chunyang, Jia, Xiuping, Ghamisi, Pedram, *Deep feature extraction and classification of hyperspectral images based on convolutional neural networks*, IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 10, pp. 6232–6251, IEEE, 2016. DOI: 10.1109/TGRS.2016.2584107

[10] Farooq, Adnan, Hu, Jiankun, Jia, Xiuping, *Weed classification in hyperspectral remote sensing images via deep convolutional neural network*, in *Proceedings of IGARSS 2018 – IEEE International Geoscience and Remote Sensing Symposium*, pp. 3816–3819, IEEE, 2018. DOI: 10.1109/IGARSS.2018.8518541

[11] Gómez□Chova, Luis, Tuia, Devis, Moser, Gabriele, Camps□Valls, Gustau, *Multimodal classification of remote sensing images: A review and future directions*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 103, no. 9, pp. 1560–1584, IEEE, 2015. DOI: 10.1109/JSTARS.2015.2399651

[12] Ahmad, Muhammad, Shabbir, Sidrah, Roy, Swalpa Kumar, Hong, Danfeng, Wu, Xin, Yao, Jing, Khan, Adil Mehmood, Mazzara, Manuel, Distefano, Salvatore, Chanussot, Jocelyn, *Hyperspectral image classification—Traditional to deep models: A survey for future prospects*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 15, pp. 968–999, IEEE, 2021. DOI: 10.1109/JSTARS.2021.3051290

[13] Kong, Yi, Wang, Xuesong, Cheng, Yuhu, *Spectral–spatial feature extraction for HSI classification based on supervised hypergraph and sample expanded CNN*, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 11, no. 11, pp. 4128–4140, IEEE, 2018. DOI: 10.1109/JSTARS.2018.2849526

[14] Yuan, Yuan, Zhu, Guokang, Wang, Qi, *Hyperspectral band selection by multitask sparsity pursuit*, IEEE Transactions on Geoscience and Remote Sensing, vol. 53, no. 2, pp. 631–644, IEEE, 2015. DOI: 10.1109/TGRS.2014.2331215

[15] Sharma, Vivek, Diba, Ali, Tuytelaars, Tinne, Van Gool, Luc, *Hyperspectral CNN for image classification & band selection, with application to face recognition*, Technical Report KUL/ESAT/PSI/1604, KU Leuven, ESAT, Leuven, Belgium, 2016. DOI: 10.3390/pr11020435

[16] Zhao, Wenzhi, Du, Shihong, *Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach*, IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 8, pp. 4544–4554, IEEE, 2016. DOI: 10.1109/TGRS.2016.2543748

[17] Yu, Shiqi, Jia, Sen, Xu, Chunyan, *Convolutional neural networks for hyperspectral image classification*, *Neurocomputing*, vol. 219, pp. 88–98, Elsevier, 2017. DOI: 10.1016/j.neucom.2016.09.010

[18] Vaddi, Radhesyam, Manoharan, Prabukumar, *Hyperspectral image classification using CNN with spectral and spatial features integration*, *Infrared Physics & Technology*, vol. 107, pp. 103296, Elsevier, 2020. DOI: 10.1016/j.infrared.2020.103296

[19] Lee, Hyungtae, Kwon, Heesung, *Going deeper with contextual CNN for hyperspectral image classification*, IEEE Transactions on Image Processing, vol. 26, no. 10, pp. 4843–4855, IEEE, 2017. DOI: 10.1109/TIP.2017.2716361

[20] Bera, Somenath, Shrivastava, Vimal K., Satapathy, Suresh Chandra, *Advances in hyperspectral image classification based on convolutional neural networks: A review*, *CMES – Computer Modeling in Engineering & Sciences*, vol. 133, no. 2, pp. [check pages], 2022. DOI: 10.32604/cmes.2022.014019

[21] Huang, Jiadong, He, Hongyuan, Lv, Rulin, Zhang, Guangteng, Zhou, Zongxian, Wang, Xiaobin, *Non-destructive detection and classification of textile fibres based on hyperspectral imaging and 1D□CNN*, Analytica Chimica Acta, vol. 1224, pp. 340238, Elsevier, 2022. DOI: 10.1016/j.aca.2022.340238

[22] Nouna, Soumaya, Nouna, Assia, Mansouri, Mohamed, Tammouch, Ilyas, Achchab, Boujamaa, *Two-dimensional Klein-Gordon and Sine-Gordon*

numerical solutions based on deep neural network, *International Journal of Artificial Intelligence*, vol. 14, no. 2, pp. 1548–1560, 2022. DOI: 10.11591/ijai.v14.i2.pp1548-1560

[23] Farokhah, Lia, Sarno, Riyanarto, Fatichah, Chastine, *Simplified 2D CNN architecture with channel selection for emotion recognition using EEG spectrogram*, IEEE Access, vol. 11, pp. 46330–46343, IEEE, 2023. DOI: 10.1109/ACCESS.2023.3269138

[24] Diakité, Alou, Jiangsheng, Gui, Xiaping, Fu, *Hyperspectral image classification using 3D–2D CNN*, International Journal of Applied Earth Observation and Geoinformation, vol. 15, no. 5, pp. 1083–1092, Wiley, 2021. DOI: 10.1080/17538947.2021.1879621

[25] Luo, Yanan, Zou, Jie, Yao, Chengfei, Zhao, Xiaosong, Li, Tao, Bai, Gang, *HSI□CNN: A novel convolution neural network for hyperspectral image*, in *Proceedings of the 2018 International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 464–469, IEEE, 2018. DOI: 10.1109/ICALIP.2018.8628313

[26] Feng, Fan, Wang, Shuangting, Wang, Chunyang, Zhang, Jin, *Learning deep hierarchical spatial–spectral features for hyperspectral image classification based on residual 3D–2D CNN*, Remote Sensing, vol. 19, no. 23, pp. 5276, MDPI, 2019. DOI: 10.3390/rs19235276

[27] Zhang, Haitao, Meng, Lingguo, Wei, Xian, Tang, Xiaoliang, Tang, Xuan, Wang, Xingping, Jin, Bo, Yao, Wei, *1D□convolutional capsule network for hyperspectral image classification*, *arXiv preprint* arXiv:1903.09834, 2019. (No DOI available)

[28] Nguyen, Duc Huy, Chao, Paul C.-P., Chung, Chih□Chieh, Horng, Ray□Hua, Choubey, Bhaskar, *Detecting atrial fibrillation in real time based on PPG via two CNNs for quality assessment and detection*, IEEE Sensors Journal, vol. 22, no. 24, pp. 24102–24111, IEEE, 2022. DOI: 10.1109/JSEN.2022.3216517

[29] Zhang, J., Chen, F., Liu, Q., *A hybrid approach for hyperspectral image classification using deep belief networks and feature selection*, Informatica, vol. 45, no. 3, pp. 345–355, 2021. DOI: 10.31449/inf.v45i3.3473

[30] Blefari-Melazzi, T., Ricciardi, S., Nardini, F.M., *A comparative study of deep learning architectures for hyperspectral image classification*, Informatica, vol. 44, no. 4, pp. 567–576, 2020. DOI: 10.31449/inf.v44i4.3182

[31] Kostadinov, M., Petrova, N., *Optimized deep CNN for spectral–spatial HSI classification with dropout and batch normalization*, Informatica, vol. 46, no. 2, pp. 178–189, 2022. DOI: 10.31449/inf.v46i2.3877