FHIC: Fast Hyperspectral Image Classification Model Using ETR Dimensionality Reduction and ELU Activation Function

Dalal AL-Alimi, Zhihua Cai, and Mohammed A. A. Al-qaness

***Abstract*— Hyperspectral images (HSIs) are typically utilized in a wide variety of practical applications. HSI is replete with spatial and spectral information, which provides precise data for material detection. HSIs are characterized by a high degree of variations and undesirable pixel distributions, providing major processing challenges. This article introduces the fast hyperspectral image classification (FHIC) model, a rapid model for classifying HSIs and resolving their associated challenges. It uses the enhancing transformation reduction (ETR) method to address the HSI difficulties and enhance classes' differentiation. It also uses exponential linear units (ELU) to smooth and speed the classification processing. The structure of the FHIC model is designed to be very flexible and suitable for a range of HSIs. The model reduced execution time and RAM consumption and provided superior performance compared to seven of the most advanced analysis models, for three well-known HSIs. In some cases, it was 60% faster than other models. In addition, this work presents a new and highly effective method for measuring the performance of the compared models in terms of their accuracy and processing speed to provide an easy evaluation method. The code of the FHIC model is available at this link: https://github.com/DalalAL-Alimi/FHIC.**

***Index Terms*— hyperspectral image, classification, ETR, activation function, ELU, performance measurement**

I. INTRODUCTION

Hyperspectral imaging (HSI) uses specialized sensors to simultaneously collect data at various narrow wavelengths. The gathered data is organized into a "hyperspectral cube," which has three dimensions: two of which indicate the scene's spatial extent and the third its spectral content. The rapid advancement of remote-sensing technology has greatly increased the spatial resolution of HSI data sets, significantly enhancing the ability of HSI data sets to express unique objects accurately. This advancement has spread to the industrial, scientific, and military fields. The acquired images have many challenges and problems, making researchers face difficulties in extracting the HSI features and targets. These features and challenges of the HSI field make it attractive for many researchers. The enormous dimensionality of HSI brought redundant information and increased consumption of computing and resources. In addition, besides the redundant information, the HSI has challenges and many mixed pixels. Mixed pixels frequently correspond to multiple categories and cause significant challenges for the classification. Due to a flaw or issue with the detectors, dead pixels represent zero or missing values [1]–[3]. There are small ratios between the number of samples in many classes due to manually labelling HSI samples [4]–[6]. Moreover, due to atmospheric variability, HSI includes undesirable data such as outliers and noise [7], [8].

Developing convolutional neural networks (Conv) and deep learning (DL) methods has opened the door to refining the HSI classification and providing an excellent chance to deal with the HSI challenges. That has also allowed the researchers to adapt the DL methodologies to be suitable for HSIs and design many new classification models and techniques. In [9], the authors used in the beginning three Conv3D layers to extract the SSF, followed by one Conv2D to enhance the spatial information extraction. Also, they added dropout layers to avoid overfitting and decrease the performance time. In order to get a more flexible model that has the ability to deal with the HSI inter-class variation, a meta-learner is used in [10] to combine the output of two different models in the first level and enhance these outputs in the next level without going back into the previous level. This method of extraction increased the spatial information extraction and sped up the last level process. To handle the limitation of the training sample numbers and avoid the overfitting of the deep classification models, this study [11] used two different submodels in the flow of the main model with different kernel sizes. This structure improved the spectral-spatial feature extraction for three various HS datasets. In [12], residual networks, shortcut connections, and average and max pooling layers were employed to classify the HSIs. It created a sub-dictionary and a loss function for each CNN to increase the discrimination of extraction and another loss function (fisher discriminative loss) besides the cross-entropy loss to enhance the classification accuracy for the whole model. Recently, many studies have used a series of layers of Conv3D and Conv2D to enable the classification model to extract more spectral-spatial

This study is supported by the National Natural Science Foundation of China (61973285) and the Hubei Provincial Natural Science Foundation of China (2022CFB438). (Corresponding author: Zhihua Cai)

Dalal AL-Alimi is with the School of Computer Science, China University of Geosciences, Wuhan 430074, China, and with Faculty of Engineering, Sana'a University, Sana'a 12544, Yemen (e-mail: dalalm.ali@hotmail.com).

Zhihua Cai is with School of Computer Science, China University of Geosciences, Wuhan, 430074, China, and also with School of Computer Science and Artificial Intelligence, Wuhan University of engineering Science, No.8, Xiong Tingbi Road, Zhifang Street, Jiangxia District, Wuhan, 430200, China (e-mail: zhcai@cug.edu.cn)

Mohammed A. A. Al-qaness is with College of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua 321004, China, and also with Emirates International University, Sana'a 16881, Yemen (e-mail: alqaness@zjnu.edu.cn).

features, increase the inter-class variation, and reduce the number of learning parameters, speeding up the processing [13]–[16]. On the other hand, this method of processing may lead to unrecognizing the small number of classes' samples and losing them during training. Numerous studies have utilized transfer learning to address the limited sample size, as it transfers the parameters of the pre-trained models into a new network and randomly initializes the top layers' parameters to manage the new input dataset [10], [17], [18]. However, transfer learning requires multiple stages of training and the conformity of input datasets; otherwise, training the network from scratch is recommended [19]. Moreover, numerous other studies arranged the extraction layers in parallel lines before combining them into a single line, thereby increasing the variation and adapting the classification model to the differences between the HSI classes [20]–[27]. Although these DL models introduced a very high performance, at the same time, they have prolonged processing and consume the RAM and the processors.

Improving the input data before being inserted into the classification stage significantly accelerates and enhances the classification process. In [28], the different scales in the guided filter were applied to the output of the PCA to extract multi-scale spatial features before feeding them into the classification model. The classification model used is a simple model of Conv2D. Thus, this study concluded that the simple classification model is more effective for the HSI than the complex or deep one if the preprocessing enhances the spectral features. This study [29] divided the spatial and spectral information into small groups, which were then processed individually to reduce the phenomenon of spectral mixing. Using the mean of the transferred features, the highest features were chosen, and the data dimensionality was reduced. Finally, the classification process was done by SVM. This paper [30] proposed the discriminative sub-dictionary learning (DSDL) method, which can further enhance the discriminating ability within-class and minimize it between-class. The adaptive multi-scale superpixel (AMSP) part improved the features representation, which adaptive according to each HSI distribution. For classification, a simultaneous orthogonal matching pursuit was used. Its proposed methods improved the accuracy but took more time because of enhancing the feature transformation in the AMSP. The patch tensor-based sparse and low-rank graph (PT-SLG) was proposed in [31] to encapsulate HSI's local and global information. The clustering algorithm uses nonlocal similarity information to improve low-rank and sparse constraints. The PT-SLG concentrated on enhancing the correlation between adjacent pixels while ignoring distant or outlier pixels. In [5], five different feature descriptors were employed to deeply explore contextual information and conform to the spatial structure. A kernel sparse representation (AKSR) method is applied to process the problem of linear representation from the previous step. Additionally, multiple kernel learning was used before the final classification operation to evaluate the variation. Although many studies used a simple classifier model to speed up the classification, using many methodologies and passing the input data through many processes and calculations slowed down the preprocessing and the feature extraction.

In deep learning, the deep models get an effect by many factors and phenomena like the vanishing gradient problem (VGP), overfitting, processing time, and dead neurons. These factors badly affect model precision and its work. So, many methods have been used to address these phenomena in deep models, such as batch-norm, activation functions, and dropout layers. Batch-norm (Batch Normalization) is a kind of inner normalization used to stabilize the training and alleviate gradient vanishing/explosion problems, but it slows down the processing [10]. In each layer in the DL model, the activation function (AF) is considered the heart of each layer in the DL model. Just as loss function is significant for measuring the model processing, the activation function is more important to correct the generated neurons' weights and the whole model process. Many methods are used as an AF, like Sigmoid, ReLu, Leaky_ReLu, PReLu, RReLu, and Tanh [32]. ReLu is the most commonly used AF in DL because it speeds up the operation and somewhat solves the VGP.

On the other hand, ReLu is not the most effective function because this leads to dead neurons and may miss many discriminatory values [2]. Many recent studies have utilized the ELU activation function because it gives a better, more efficient, and faster process than the BN and ReLu activation functions. The authors used multi-layer CNNs with exponential linear units (ELU) AF to classify electroencephalography in [33], [34] and discovered that the ELU provided faster classification with higher accuracy than ReLu [35]. The ELU also demonstrated the best performance for optical property mapping by the artificial neural network (ANN) model in [36], for detecting the vigor of rice seeds by a deep CNN model in [37], for detecting rotten apples in a device of fruit sorting in [38], and for predicting the oil content in HSI of a single maize kernel in [39]. It was also utilized in numerous further studies [40]–[43].

In HSI, deep models (DM) slow the processing and lead to overfitting, so many models use the dropout method to avoid overfitting and speed up the processing. Moreover, the high dimension of HSI with a small quantity of labeled training data renders DMs worthless for HSI classification due to the need for many modifications during the training [2], [44], [45]. However, this leads to missing critical features that may help improve the classification because of the random stopping of neurons from working. The very complicated and deep models may not be flexible enough to handle the different HSIs and their varying complexities and sizes. Also, using a simple classification model with complicated preprocessing consumes time and resources. The best way is to find the balance between the preprocessing and classification stages. As a result, the tasks will be divided between the two stages, the HSI complications will be resolved in both, and the performance time will be reduced, by using the appropriate method in each stage. Consequently, this study considered these difficulties and provided a suitable classification model after understanding the input data well and choosing the best methods that can contribute to increasing the classification accuracy at a lower cost in terms of both time and resources. The following are the key contributions of this study:

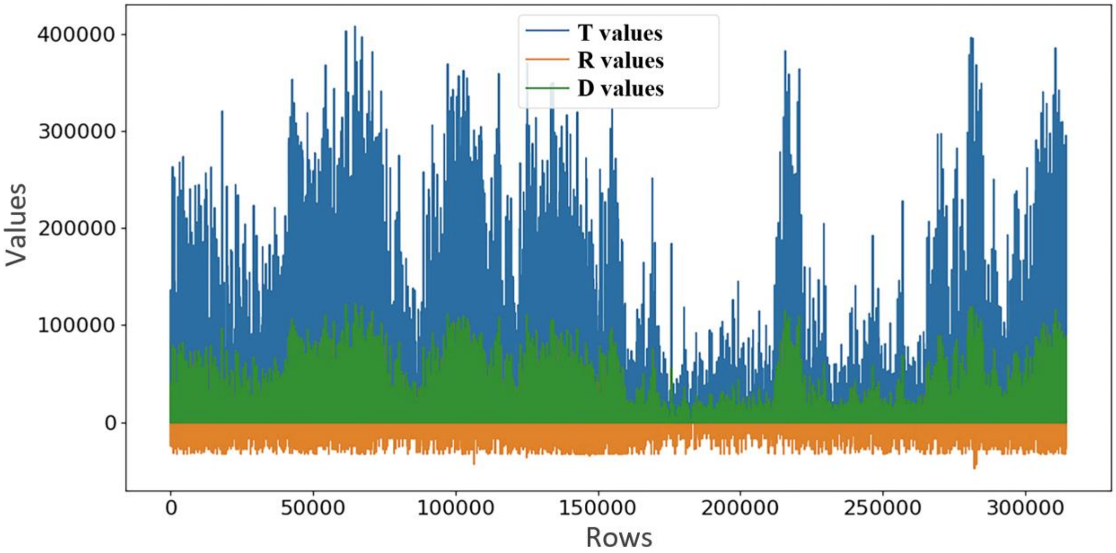


Fig. 1. The output of the internal process of enhancing transformation reduction (ETR) for the three generated matrix (T, R, and D).

- Since each HSI is unique in content and complexity, a method that can adapt to these variations is required. So, this study provided a flexible model that can handle the variation of the HSIs and give a stable performance for all different input datasets, called the fast hyperspectral image classification (FHIC) model.
- When examining the data of HSI classes closely, it is possible to identify numerous mixed values and noise, complicating the classification. The ETR method is used to rectify erroneous pixel placements and reduce noise and dimension [7].
- Driven by the input data, which contains a large number of negative and zero values, the ELU AF is chosen to accelerate the process, solve the problem of using ReLu, and provide a clear view of its impact on the processing.
- It introduces a new easy way to compare models' performances based on the relative importance of accuracy and execution time, which gives an easy way of comparing their performance and differentiating between them.
- The FHIC classification model continued to provide very high performance in terms of accuracy and time for all datasets.

II. The Fast Hyperspectral Image Classification

This section provides an explanation of the fast hyperspectral image classification (FHIC) model of this study, which can be broken down into two sections. The first section, called the preprocessing section, describes the enhancing transformation reduction (ETR) approach in more detail. The classification model structure is explained in the second part.

A. Preprocessing Operations

HSI is an enormous data collection stored in pixels (thousands or millions of points). Each pixel usually correlates to its neighbours, providing rich, informative information to detect each class. Nevertheless, many challenges are faced because of the size, which consumes more RAM, storage space, and time. Furthermore, HSI data has several other issues, such as incorrect values, known as dead pixels and outliers [1], and abnormal data distribution. The undesired or incorrect values mostly happen because of changes in atmospheric conditions or sensor malfunction during data collection [2], [3], [8]. Dead pixels are presented as missing or zero values, which later complicate the analysis, so locating and handling dead pixels is a significant task to enhance the classification.

Generally, preprocessing processes aim to obtain informative data for more feature extraction and improved accuracy. Most feature extraction methods (FEM) like PCA, the most commonly used to transfer and extract the features and reduce the dimensionality of the HSIs, ICA, LDA, and SVD, have limitations in handling many issues like dead pixels, outliers, and data distribution [7]. Thus, enhancing transformation reduction (ETR) [7] successfully solved these problems and difficulties for HSIs. The ETR works to close the distance between the inter-class elements, reducing confusion and correcting the spatial elements' information. After the position correction, it also filters the data, which helps reduce mixed values and outliers. Moreover, ETR scaled the data many times and enhanced the data distribution; all of these helped speed up the classification and enhance the extraction accuracy.

In the first stage of ETR, after generating the covariance matrix (C) for the input data, the error matrix (Ɛ) is subtracted from C, whose values range from -1 to 1. Consequently, this operation improves the covariance matrix to correspond with the HSI variables and makes the variables closer together and more uniform, as shown in Equation (2).

$$\hat{C}_x = C_x - \mathcal{E}, \tag{1}$$

Then the top eigenvectors are taken from the $\hat{C}$ to get the weight matrix (W), whose number (k) is less than the number of input data bands. After that, the matrix of W is multiplied by the main input data (X):

$$\mathrm{T}_{\mathrm{d}\times\mathrm{k}} = \mathrm{X}_{\mathrm{d}\times\mathrm{k}} \times (\mathrm{W}_{\mathrm{k}\times 1})^{\mathrm{T}} \tag{2}$$

where d is the instance number in X.

So, in this stage, the dimension of input data is reduced and transferred into a new subspace.

The second stage of the ETR works to enhance the data distribution and eliminate outliers and death pixels. This stage depends on three factors: first-stage output (T), marker matrix (R), and weight constant (ρ). The R matrix is derived from T based on two conditions outlined in the following Equation, and the dimension of T and R is (d × k); where d is the number of instances, and k is the number of features (bands), respectively:

$$\mathrm{R} = \begin{cases} \sum_{\mathrm{d}=1}^{\mathrm{d}} \mathrm{T_d} \Big/ \mathrm{d} & , \text{if } \mathrm{R} \leq \mathrm{T} \\ \max(\mathrm{T_d}) \times \rho & , \text{if } \mathrm{R} > \mathrm{T} \end{cases} \tag{3}$$

1. After computing the mean of the instances of T, if the R-generated values are smaller than the values of T (R ≤ T), the first condition is met, and the correlation between related values is maximized.
2. If the values of R from the first calculation are bigger than the T values (R > T), the second process is used ($\max(\mathrm{T_d}) \times \rho$). Here, in order to build the R matrix, the maximum instance values of T are obtained first and multiplied by ρ, which is a constant value between zero and one. This step minimizes the top values and goes on with the same processes as in the previous condition to generate the R matrix.

After obtaining the R matrix, the morphological dilation (MD) technique is applied to improve the internal value correlation within each class and reduce the number of outliers and unrelated pixels.

$$\mathrm{D}_{\mathrm{T}}^{\delta}(\mathrm{R}) = \delta_{\mathrm{T}}^{\mathrm{n}}(\mathrm{R}) = \delta_{\mathrm{T}}^{1}\left(\delta_{\mathrm{T}}^{\mathrm{n}-1}(\mathrm{R})\right) \wedge \mathrm{T} \tag{4}$$

where the R matrix was dilated under the control of the T matrix, $T \geq R$, $n \geq 1$, δ denotes the MD, and ∧ indicates the pointwise minimum. The transferred data and its changed distribution for the three generated matrixes T, R, and D from Equations 1-3 are visualized in Fig. 1. It is evident how the D operation smooths and corrects the distribution and minimizes outliers and undesirable values in accordance with T and R, which will play a significant role in facilitating the classification. The final operation in ETR is to normalize the output of D using the Gaussian distribution, which helps speed up the classification training. More details about ETR is in [7].

After the ETR process and obtaining the most informative features, the order of these features is rearranged. This reorganization aims to guarantee that the focus is on the most important features and minimize loss during classification operations. As shown in Fig. 2, the reordering or shifting occurs as follows: the most informative features, which were in the beginning, are shifted to the centre, and the least informative features, which were in the last half part, are shifted to be distant from the centre. The authors in [15] employed a mathematical way to control the rearrangement loop of the features. Nonetheless, this way led to losing one of the most informative features (the second one), and the whole pixels of this feature or band was converted to zeros. To avoid this problem, we reordered the array of names of these features without their content and then recalled the contents according to the new order.

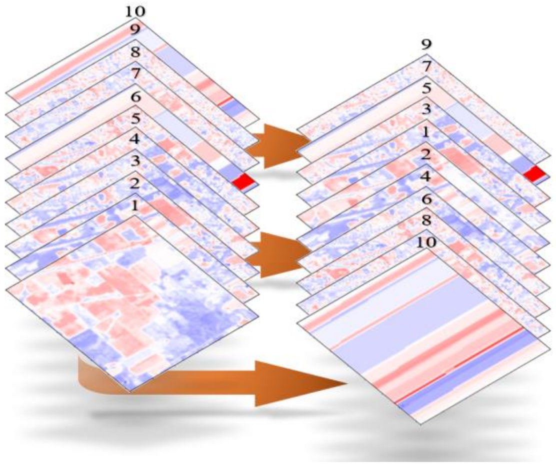


Fig. 2. Reorder the ETR features.

B. Fast hyperspectral image classification model

In the DL, the ReLu AF is utilized to overcome the problem of the vanishing gradient (VG) and accelerate the training since it converts all negative values to zeros ($max\ (0, x)$). Despite its advantages, ReLu decreases the accuracy of deep models due to the death of neurons during training as a result of the conversion of their output to zeros. Thus, this issue has prompted researchers to develop more complicated models to improve final accuracies, like residual networks (ResNet) [46], feature pyramid networks (FPN) [47], mixed CNN with covariance pooling (MCNN) [15], and meta-learner hybrid models (MLHM) [10]. This results in wasted time and RAM and makes the execution more difficult.

In the HSI, Conv2D and Conv3D have been used. The primary input data is divided into small cups of data whose size depends on the window size, and each window is padded by zeros to unify the size. Even though building the cube enables us to analyze spectral data in addition to spatial data, the classification model is fed with data that is comprised of zeros. Additionally, the preprocessing phase normalizes and scales the data to accelerate and streamline the categorization procedure. As a result, the negative numbers may equal the positive values after normalizing and scaling. Because of these existing negative and zero values, utilizing ReLu AF with them impacts the extraction

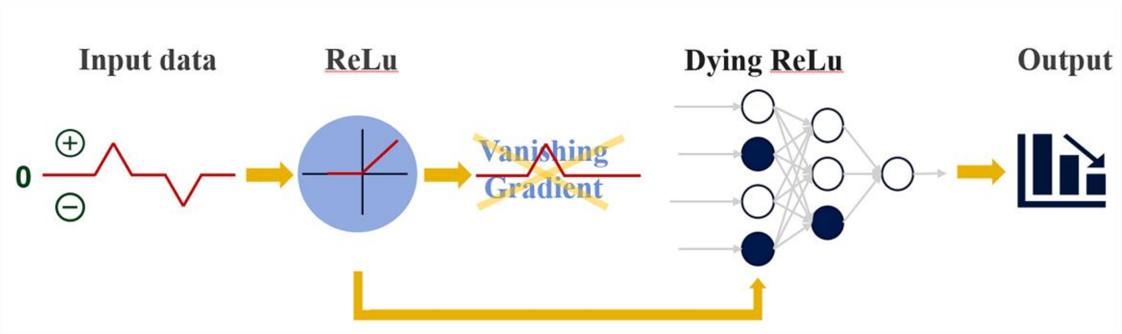


Fig. 3. The ReLu operation in the deep learning model and its effect.

and stops some neurons from working. Fig. 3 depicts the ReLu process in DL neurons and its positive and negative impacts on the training process and outcomes.

The exponential linear units (ELU) activation function has been introduced as an effective way to accelerate and smooth the training process and avoid the issue of dying neurons [33], [36], [37], [39]. Fig. 4 shows the differences between the output of the ReLu and ELU. ETR works to normalize the generated weight in each neuron, which helps speed up the training. It uses the exponential function for all negative values to make activations centred at zero, as a curve sloped closely to zero, as seen in the following Equation:

$$elu(\varkappa) = \begin{cases} \exp(\varkappa) - 1 & if\ \varkappa \leq 1 \\ \varkappa & if\ \varkappa > 1 \end{cases} \qquad (5)$$

where $\varkappa$ represents the input values in the activation function. The input data contains numerous zero values, which are increased by zeros padding, and countless negative values; therefore, ELU is preferable to ReLu as the classification model's activation function to avoid losing essential values to boost classification.

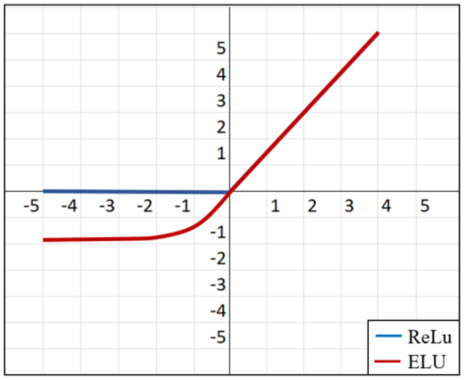


Fig. 4. The different outputs of the ReLu and ELU activation functions.

This study's classification model is a straightforward, fast hyperspectral image classification model (FHIC). It consists of two Conv2D layers with large numbers of filters (50 and 100) due to the amount of the input data, one MaxPooling layer with a 2×2 pooling window to accelerate training, a fully connected layer (FC), and a softmax layer to generate the final output, as seen in Table I. ELU is used as an activation function in all neurons of the CNN and FC layers.

TABLE I
THE STRUCTURE OF THE FAST HYPERSPECTRAL IMAGE CLASSIFICATION MODEL (FHIC).

Layer	(Filters) (kernel_size) (AF)
Conv2D	(50) (5, 5) (ELU)
Conv2D	(100) (5, 5) (ELU)
MaxPooling2D	(2, 2)
FC	(100) (-) (ELU)
FC	(NO. of classes) (-) (softmax)

III. EXPERIMENTS

This section covers the HSI datasets used to train and test the model of this study. In addition, it displays the outcomes of the fast hyperspectral image classification (FHIC) model and its efficacy, the outcomes of each dataset applied, and a comparison with eight well-known classification models.

A. Datasets

The study used three distinct types of datasets in its experiments. The first data set is the Indian Pines (IP), USA. The second dataset is Kennedy Space Center (KSC) in Florida. These two datasets were gathered by AVIRIS, the Airborne Visible InfraRed Imaging Spectrometer. Their details are listed in Table II and are accessible via this website[1].

The third dataset is called WHU-Hi-HongHu (HH)[2]. A 17-mm focal length Headwall Nano-Hyperspec imaging sensor was used to collect HH over Honghu City, Hubei province, China. The acquired region is a complex agricultural landscape with 22 varieties of crops. As seen in Fig. 5 (HH scene), it is a complex region since it has multiple varieties of the same crop, such as Chinese cabbage, cabbage, cotton, cotton firewood, Brassica Chinensis, and small Brassica Chinensis [48]. Table II describes the details of these datasets and shows the differences. Fig. 5 shows the scene and the ground truth (GT) for these three datasets. The number of samples that fall under each class is

[1] https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

[2] http://rsidea.whu.edu.cn/resource_WHUHi_sharing.htm

TABLE II
THE THREE USED DATASETS IN THIS STUDY.

Dataset	**Sensor**	**Band Numbers**	**Spatial Dimensions**	**Spatial Resolution**	**Classes Number**
IP	AVIRIS	200	145 × 145	20m	16
KSC		176	512 × 614	18m	13
HH [48]	Headwall Nano-Hyperspec	270	940 × 475	0.043m	22

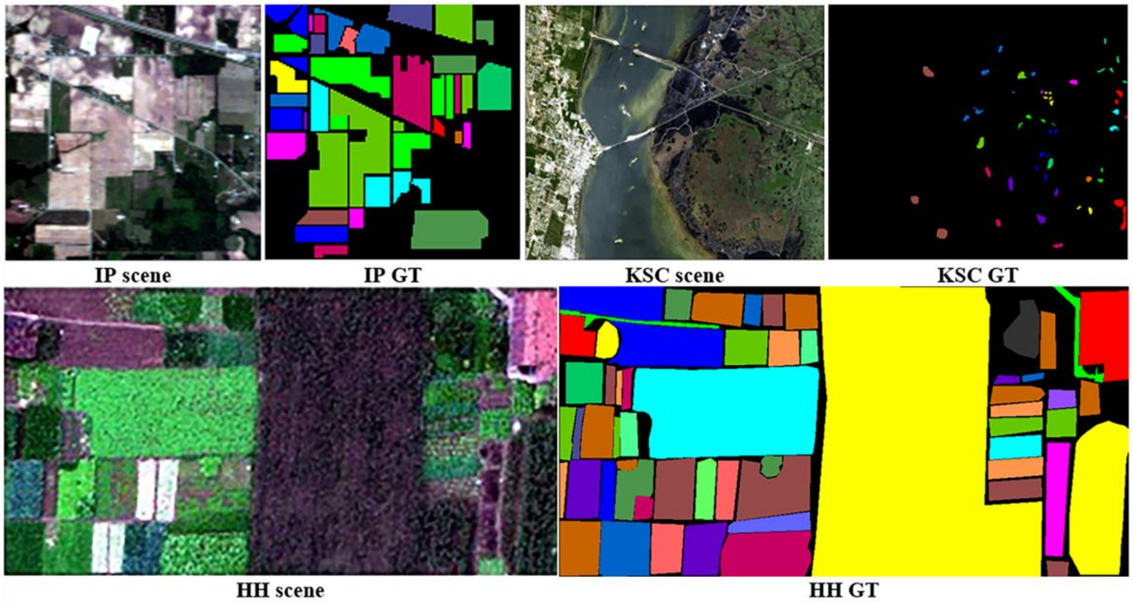


Fig. 5. The scene and the ground truth for the tree used images.

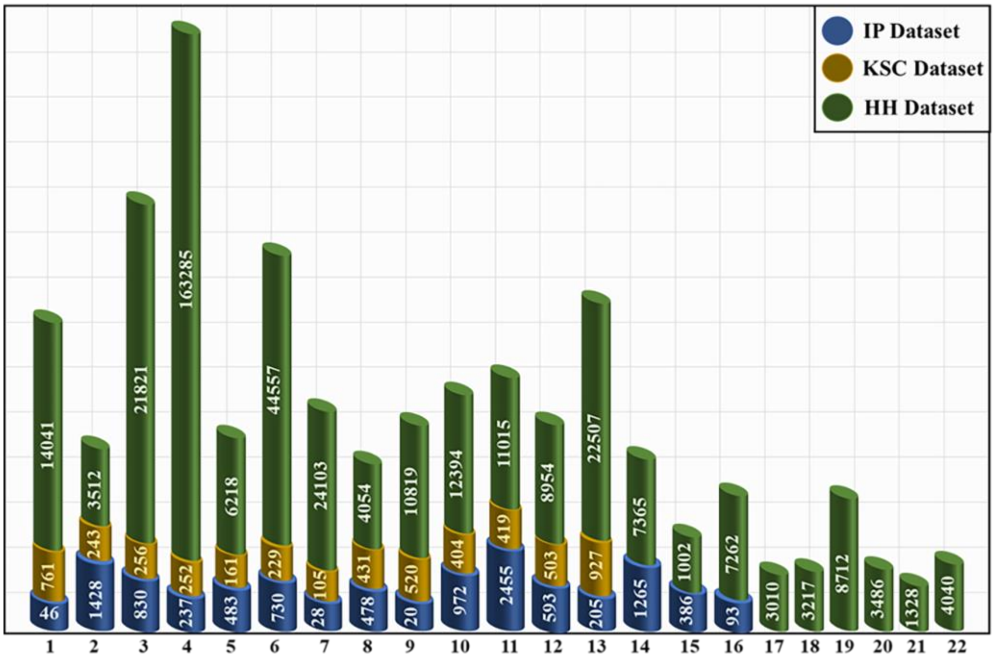


Fig. 6. The number of classes in each dataset and how many samples in each class are.

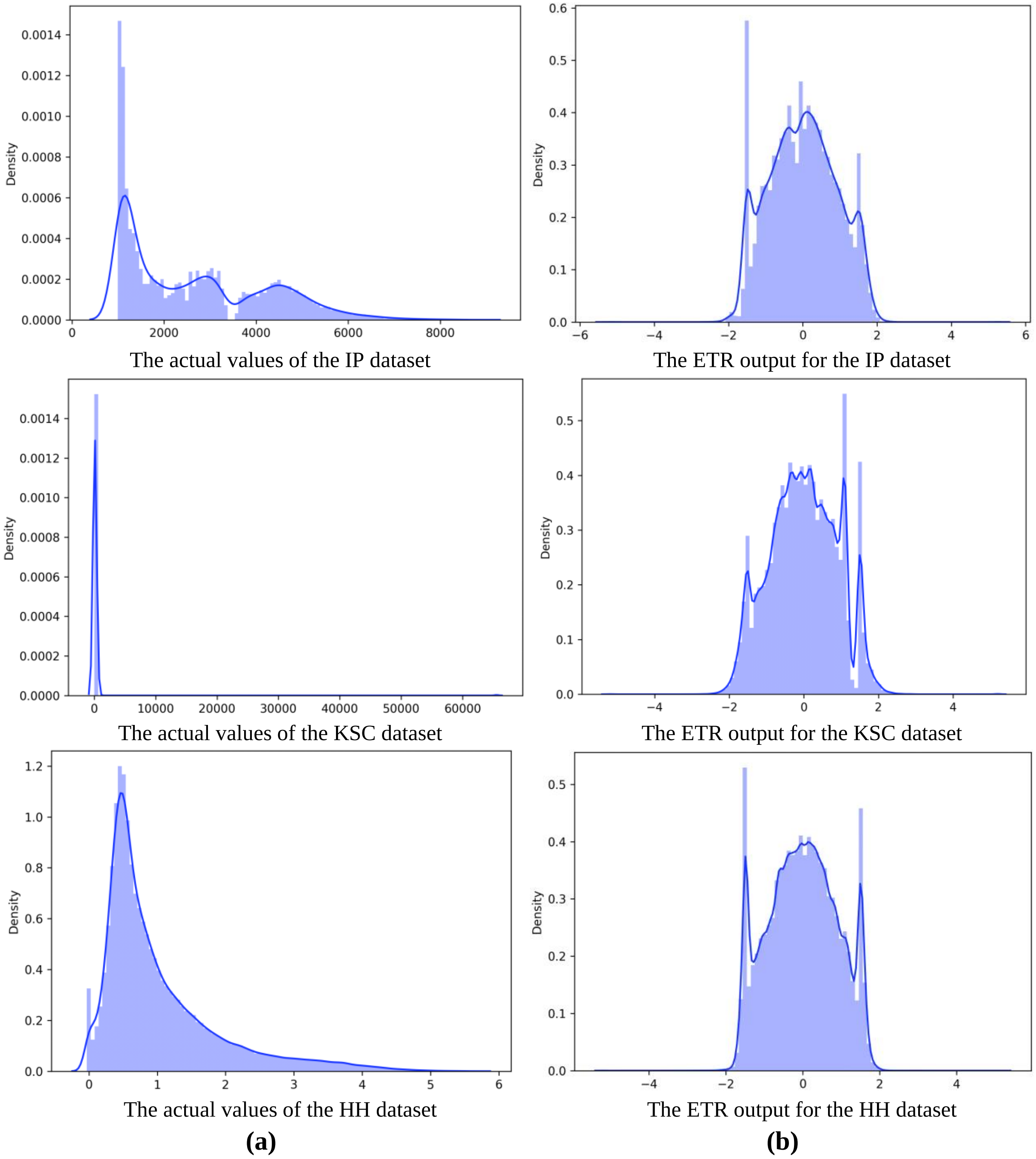
depicted in Fig. 6, which compares the three different datasets.

B. The Experimental Results

This section presents the findings of the fast hyperspectral image classification (FHIC) model and a comparison to the most well-known classification models in the field of HSI. This study evaluated the efficacy of the FHIC using three common HSI datasets: Indian Pines (IP), Kennedy Space Center (KSC), and WHU-Hi-HongHu (HH) datasets.

The HSI needs to be filtered, enhanced in the data distribution, and reduced its dimension before being fed into the classification model. This work used ETR, a dimension-reduction technique, to reduce the HSI difficulties and enhance the classification. According to Table II and Figs. 5 and 6, the three datasets have distinct characteristics and were collected by different sensors in distinct regions. They vary in the number of bands and classes and the number of samples within each class. In addition, their data distribution and complexities are diverse, as seen in Fig. 6, which depicts the data distribution of the first band in each dataset.

As a consequence of these reasons, the ETR method was utilized as a preprocessing to simplify these HSI complexities. It assigned all the input datasets into 15 bands or features. The IP dataset values match the first condition of Equation (3) in the second part of ETR. However, the values of the KSC and HH datasets met the second condition. Based on Table III, it was determined that 0.8 and 0.3 should be used for the KSC and HH datasets after conducting numerous experiments with various $\boldsymbol{\rho}$ values and the FHIC model; these values provided the highest degree of accuracy. Along with reducing dimensions, the ETR also reduced noise and undesired values. These aid in accelerating the categorization process, improve accuracy concurrently—as seen in the comparison—and decrease the drain on the RAM and storage space. The reduced dataset by ETR is then partitioned into small cubes with a window size of 15 for all datasets.



The actual values of the IP dataset

The ETR output for the IP dataset

The actual values of the KSC dataset

The ETR output for the KSC dataset

The actual values of the HH dataset

The ETR output for the HH dataset

Fig. 7. The data distribution for the first band of each dataset. (a) represents the actual values, and (b) represents the output of the ETR method for the first band.

TABLE III.
THE KAPPA ACCURACY OF DIFFERENT (ρ) VALUES BY FHIC MODEL FOR THE TWO DATASETS, KSC AND HH.

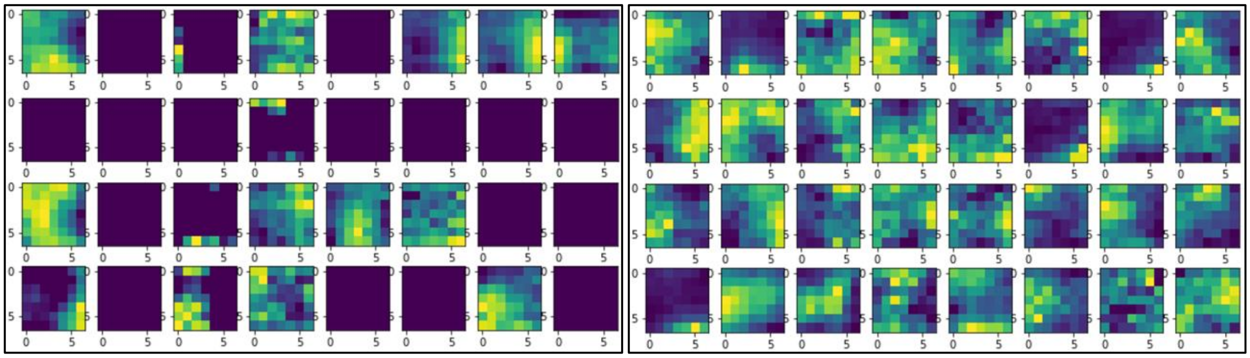
ρ (%)	KSC	HH
20	96.77	99.48
30	97.97	**99.86**
50	98.72	99.57
70	98.48	99.80
80	**98.88**	98.49

All the models utilized in this article were repeated several times to decrease the impact of random initialization. The performance of the various approaches was then compared using the mean and standard deviation of Kappa accuracy (KA), overall accuracy (OA), and average accuracy (AA) for all executions. The learning rate is 0.001 with the Adam optimizer, with 100 epochs and 256 batches. 20% of each dataset was used for training and 80% for testing. The size of the window was 15. All experiments were conducted on a 64-bit Windows 10 system with 128GB of RAM, 89GB of GPU space, and Python as the programming language.

1) The ETR Method Outputs

Observing the input data distribution before processing is preferable, as this provides a clear picture of the type of data available, and the type of processing required. Fig. 7 (a) shows the data distribution for each dataset's first band. Due to the spatial resolution of the IP and KSC datasets, the probability of having noise and mixed pixels is very high, further complicating the classification process. Moreover, because the variation between the IP and KSC dataset classes is very narrow and the wavelength range in the several bands (spectral dimension) is very small, the average value spread is minimal, as illustrated in Fig. 7 (a). Fig. 7 (a) demonstrates that the IP and KSC data distributions are not normal and that there is a big gap between small and high values. In addition, the KSC dataset has the most complex data distribution due to the high degree of similarity among its classes and their narrow wavelength range. The HH dataset is the least complex dataset, but it has a huge number of classes. These difficulties and those outlined in the introduction will complicate and slow down the classification process.

Fig. 7(b) illustrates the output of the ETR method and the differences between the actual and ETR values in the first band for each of the three datasets. The differences are significant in the data distribution and dimensionality. The ETR method enhances the variation between classes and their features to select the most informative bands,



(a) (b)

Fig. 8. The output of the second 2CNN layers in R-FHIC and FHIS models. a) represents the ReLu outputs, and b) is the ELU outputs.

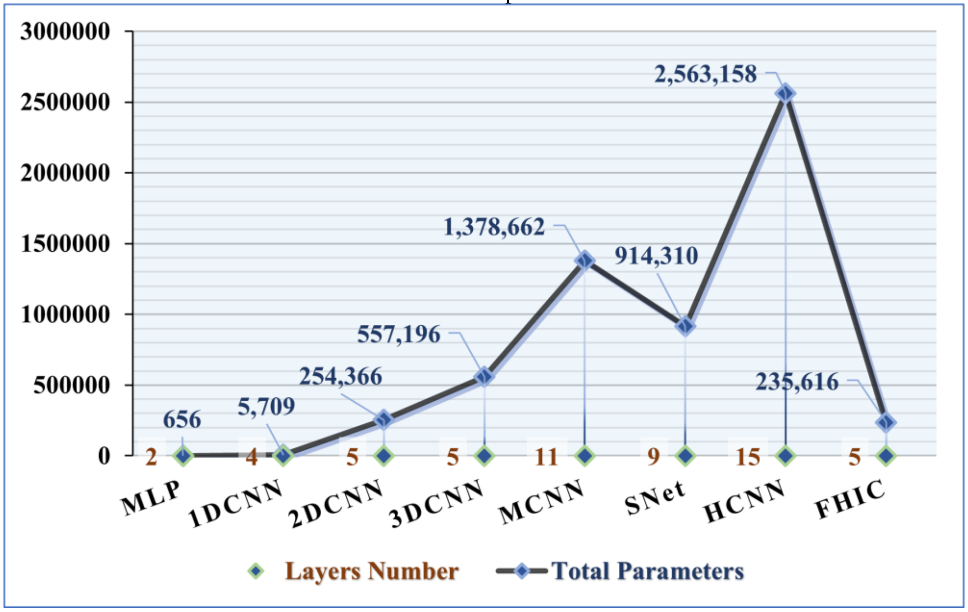


Fig. 9. The comparison between these models according to their layer numbers and parameter numbers for the IP dataset.

ensuring the noise and the mixed pixels are reduced. The second part of ETR reduced the distance between the related values in the same class. After using the ETR, the datasets have a more normal data distribution, and the range of data distribution becomes longer, increasing the variation. In addition, the ETR method reduced the gap between the large and small values, enhancing the range of the values' density and making it more balanced in distribution. Scaling the data in a specific range with the correct allocation helps speed up the classification processing, as seen in the following sections. As seen in Fig. 7, the ETR reduced the complexity and enhanced the data distribution to be more efficient.

2) The Activation Function Outputs

This study initiated experiments by passing input data through the enhancing transformation reduction (ETR) dimension reduction method. The reasons behind using the ETR are as follows: 1) ETR has a O(n) complexity, making it a speedy and efficient dimension reduction approach. 2) After increasing the diversity between the classes entered in the first part, it selects the most informative features. 3) It corrects the characterization of pixels within each class by eliminating the gaps and boosting the correlation between them, hence decreasing noise and undesired pixels/values and enhancing the differentiation between classes and classification accuracy. 4) ETR scales and normalizes the data gathered, simplifying and accelerating the classification process. Thus, all input data was transferred according to the processes of ETR and reduced to 15 features. After completing the ETR process and identifying the most critical subset of the transferred features, the top half of these features were moved to the center, as explained in the preprocessing operation subsection.

During training, the pixels on the edge lose efficacy due to the forward and backward propagation operations, and only the pixels in the middle achieve the required accuracy. Consequently, utilizing the zeros padding to cover the created cubes efficiently ensures that all pixels can be treated equally and improve extraction efficiency. Thus, after shifting, the data was divided into small cubes with 15 window-size dimensions and then covered by zeros padding. After that, the data was inserted into the FHIC model for classification. The FHIC model is very straightforward and consists of just five layers: two Conv2D layers for extracting the feature maps, a max-pooling layer to speed up the process, a fully connected layer for flattening the extracted feature maps and enhancing the extraction, and a Softmax layer for obtaining the results.

As previously mentioned, after the ETR normalization, scaling, and the creation of the cubes, the input data contains many negative and zero values. This section shows the effectiveness and differences between using ReLu and ELU AFs. In order to compare the performances of ReLu and ELU, the FHIC model was trained with ReLu AF (referred to as the R-FHIC model) and ELU AF (the FHIC model). The output of the second Conv2D layer in the R-FHIC and FHIC models was visualized to evaluate and see the efficacy of these two activation functions during the training period in Fig. 8. As depicted in Fig. 8(a), the amount of zero representation in ReLu is enormous and far larger than in ELU AF. Their outputs differ significantly; therefore, the ReLu has a negative effect on model performance and slows down the

TABLE IV

THE RESULTS OF THE NINE COMPARED CLASSIFICATION MODELS FOR THE IP DATASET.

#	Training Samples	Testing Samples	MLP	CNN1D	CNN2D	CNN3D	SNet	MCNN	HCNN	R-FHIC	FHIC
1	9	37	0.08	0	0.46	0.62	1	1	1	0.84	0.92
2	286	1142	0.45	0.64	0.72	0.94	0.96	0.94	0.99	0.99	0.97
3	166	664	0.12	0.31	0.67	0.98	0.97	0.99	1	0.98	1
4	47	190	0.13	0.07	0.42	0.97	0.98	0.99	0.99	1	1
5	97	386	0.32	0.78	0.88	0.99	0.99	1	0.99	0.97	0.98
6	146	584	0.80	0.98	0.91	0.98	0.97	1	1	1	1
7	5	23	0.25	0	0.52	0.70	0.95	1	1	0.91	1
8	96	382	0.93	0.99	0.98	1	1	1	1	1	1
9	4	16	0.18	0	0.38	0.88	0.88	1	0.86	1	0.88
10	194	778	0.45	0.63	0.76	0.94	0.97	0.99	0.98	0.98	0.98
11	491	1964	0.73	0.76	0.82	0.95	0.96	0.99	1	1	1
12	119	474	0.07	0.26	0.51	0.97	0.93	0.96	0.99	0.97	0.95
13	41	164	0.91	0.91	0.91	0.99	1	1	1	1	1
14	253	1012	0.83	0.96	0.95	1	1	1	1	1	1
15	77	309	0.08	0.28	0.75	0.87	0.77	0.99	1	0.99	0.98
16	19	74	0.57	0.84	0.91	0.92	0.96	0.93	1	0.96	0.97
	KA (%)		46.23±4.27	64.38±2.83	75.68±2.20	94.52±1.34	95.10±7.09	98.10±0.36	99.31±0.36	97.71±2.80	98.65±0.17
	OA (%)		53.47±3.56	69.15±2.39	78.69±1.94	95.19±1.18	95.69±6.26	98.34±0.31	99.39±0.32	98.00±2.44	98.82±0.15
	AA (%)		42.24±5.22	55.78±3.97	68.58±3.81	90.54±3.43	94.61±6.29	97.74±1.26	99.06±0.61	96.80±3.83	98.14±0.74
	TT(s)		18.33	29.76	39.15	59.23	14.46	20.96	35.03	12.22	9.96
	ST(s)		0.43	0.75	0.87	1.12	1.18	1.42	2.09	1.03	1.03

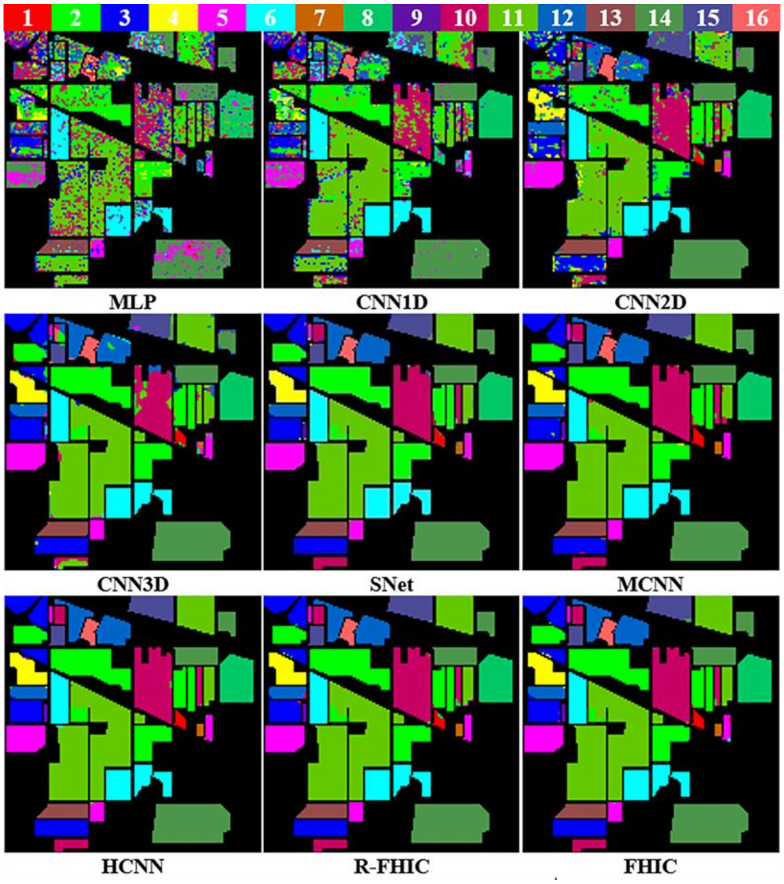


Fig. 10. The output of the nine used classification models for the IP dataset.

process. In contrast, the ELU activation function extracted more features and ran without being affected by negative or zero values, as shown in Fig. 8(b).

The experiments of these two models were run many times to get a more accurate observation to measure the accuracy of each model. From Tables IV to VI can be observed the FHIC model with ELU activation function obtained higher accuracy than R-FHIC with ReLu AF in the three used datasets. Besides the higher accuracy, the ETR improved the performative time too much, especially in the HH dataset, which was 86 seconds faster in the training time (TT) than using ReLu and 12 seconds in the test time (ST). We learned from these experiments that understanding the input data helps choose the correct activation function.

3) The Experimental and Comparison Results

In this section, the proposed model is compared with many different well-known classification models. These models are multi-layer perceptron (MLP) [2], CNN1D [2], CNN2D [2], CNN3D [2], HybridSN (SNet) [9], mixed convolutions and covariance pooling (MCNN) [15], and HybridCNN (HCNN) [49]. These models are different in complexity and structure. Fig. 7 shows the number of layers and the parameters in each model. These models, with their layer types, are divided into three data processing types. Spectral extraction models: MLP and CNN1D, spatial extraction model: CNN2D, and spectral-spatial extraction models: CNN3D, SNet, MCNN, and HCNN. These two models of MLP, which includes one layer of FC, and CNN1D, which has one layer of Conv1D, focus on extracting spectral information. The CNN2D model with two Conv2D layers is the best to classify spatial information. The CNN3D model includes two Conv3D layers, so it works to extract spectral-spatial information at the same time. The SNet, MCNN, and HCNN models each comprise a unique series of CNN3D and CNN2D layers to improve the accuracy of spectral-spatial data extraction.

The SNet model has a series line of three Conv3D layers and one Conv2D layer, and all have different kernel sizes to enhance the extraction and dropout layers to speed up the process. The MCNN model consists of three Conv3D layers and one Conv2D layer with the exact kernel sizes, followed by Covariance Pooling to create multiple stacked features from the final layer and add supplementary information. The HCNN model has three parallel lines, and each line has three Conv3D layers. The output of the combined three lines was fed into two layers of Conv2D, and all the layers have different kernel sizes. These three models are deep and intricate classification models designed to improve the efficiency of spectral-spatial feature extraction.

The epochs number, batch size, and window size were 100, 256, and 15 in all models, except in the HCNN model, were 50, 200, and (15, 13, 11). All these models used the ReLu as an activation function. PCA was utilized

This article has been accepted for publication in IEEE Transactions on Geoscience and Remote Sensing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TGRS.2023.3314619

TABLE V
THE RESULTS OF THE NINE COMPARED CLASSIFICATION MODELS FOR THE KSC DATASET.

#	Training Samples	Testing Samples	MLP	CNN1D	CNN2D	CNN3D	SNet	MCNN	HCNN	R-FHIC	FHIC
1	152	609	0.80	0.94	0.92	0.78	0.86	0.92	0.95	1	1
2	49	194	0.51	0.30	0.16	0.70	0.78	0.33	0.92	0.97	0.96
3	51	205	0	0	0.03	0.91	0.83	0.07	0.85	0.98	0.95
4	50	202	0	0	0.17	0.42	0.58	0.25	0.65	0.82	0.88
5	32	129	0.01	0	0.36	0.80	0.89	0.20	0.85	0.92	0.98
6	46	183	0.40	0	0.31	0.97	0.40	0.03	0.77	0.99	1
7	21	84	0	0	0.07	1	0.58	0	0.88	1	1
8	86	345	0.34	0.38	0.76	0.80	0.86	0.89	0.96	0.99	0.99
9	104	416	0.04	0.74	0.65	0.73	0.91	0.19	0.76	1	1
10	81	323	0.03	0.00	0.40	0.64	0.87	0.80	0.90	1	1
11	84	335	0.76	0.85	0.65	0.93	1.00	0.95	1	1	1
12	101	402	0.18	0.22	0.37	0.99	0.91	0.68	0.95	0.98	1
13	186	741	0.89	0.98	0.72	1	1	1	1	1	1
	KA (%)		32.95±13.46	44.42±1.70	48.84±9.36	81.79±6.64	84.63±2.87	60.51±1.57	89.71±2.07	95.69±8.58	98.63±0.17
	OA (%)		40.45±12.26	51.80±1.47	54.61±8.01	83.56±6.11	86.24±2.55	65.08±1.37	90.76±1.86	96.11±7.77	98.77±0.15
	AA (%)		27.30±8.78	32.67±1.37	43.04±8.66	81.28±4.87	80.64±3.25	49.49±2.07	87.67±2.61	95.33±7.44	98.01±0.33
	TT (s)		12.19	21.29	25.33	38.01	8.50	15.97	19.71	7.67	6.15
	ST (s)		0.28	0.52	0.61	0.73	0.55	0.80	1.02	0.56	0.54

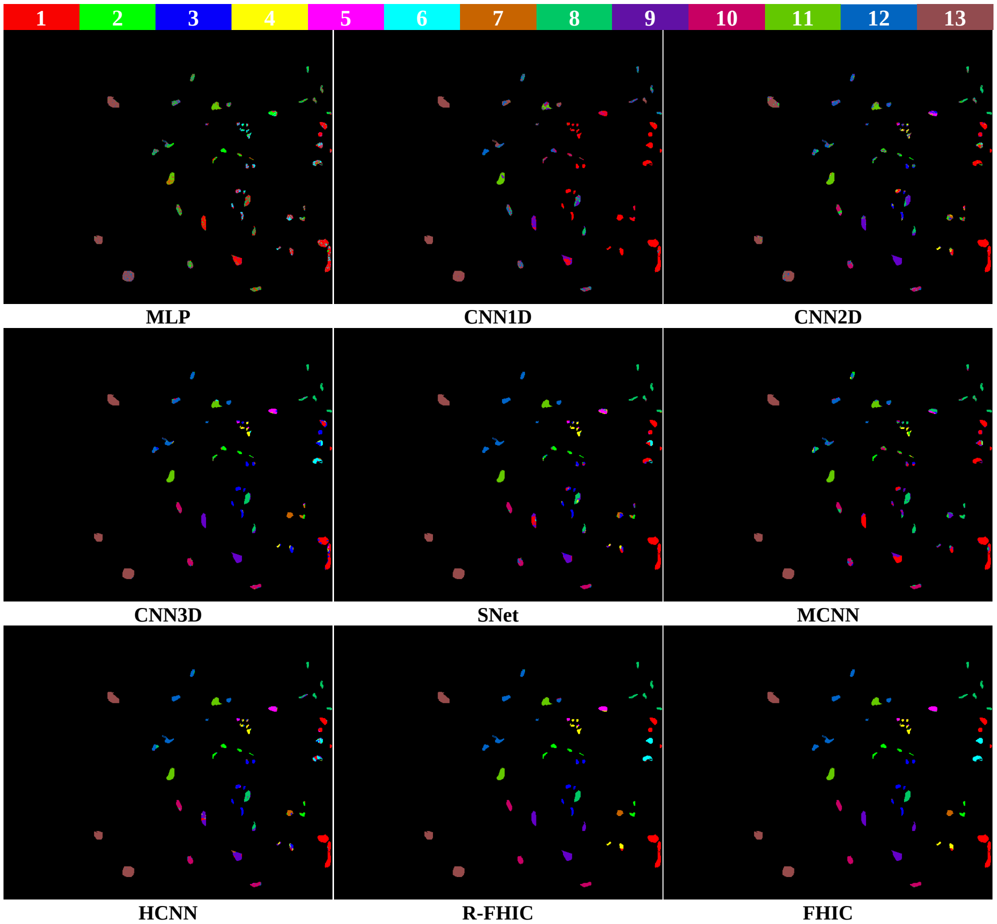


Fig. 11. The output of the nine used classification models for the KSC dataset.

to reduce the dimension into 30 features for CNN1D and CNN3D and 15 for others. Fig. 9 depicts a comparison of the various used models based on their layer and parameter numbers for the IP dataset. The FHIC model is the third model with the fewest parameters after the MLP and 1DCNN models.

4) The Indian Pines Dataset Results

As seen in Table II, the IP dataset has the smallest spatial dimension and the lowest resolution, with 16 classes. In addition, Fig. 6 demonstrates that the classes of the IP dataset contain a varying number of samples, with a wide variation between them. In addition to these complications, it has an abnormal data distribution with many noises and many outliers, as seen in Fig. 7. All these characteristics make it extremely difficult to extract its features.

From Table IV, it can be seen that the spectral models (MLP and 1DCNN) achieve less accuracy, and 2DCNN, focusing on spatial information, was better than them. On the other side, the most accurate model in the spectral-spatial extraction models was HCNN, whose parallel structure; was better than MCNN and SNet models. The FHIC was the second most accurate model, after HCNN, with a tiny difference (only 0.66%), to classify the IP dataset. In terms of execution time, the longest training time was for the 3DCNN model because it did not employ any techniques to speed up the process, such as a dropout layer like in SNet, and it also used BN, which slowed down the process even more. The MCNN model obtained very high accuracy but needed a long training time because of the Covariance Pooling function. Due to its structure, the processing time of the HCNN model was longer than the MCNN model. The FHIC was the best performing on the IP dataset in terms of accuracy; it also provided the shortest execution time (training and testing time), as shown in Table IV. Fig. 10 depicts the categorization output visualization for the whole collection of models applied to the IP dataset.

5) The Kennedy Space Center Dataset Results

The KSC dataset has longer dimensions and a narrower wavelength range than the IP dataset, as seen in Fig. 7. It is challenging to differentiate land cover due to the similarities of their spectral signatures, whose wavelength ranges are very close. In addition, Fig. 7 demonstrates that the data range is extremely narrow, making this the most complex dataset. Because the ETR method corrected the position of pixels, reduced the complexity, and normalized the data distribution, that helped the FHIC model obtain the best accuracy while the other models failed to do the same. The second-best result after FHIC was for the R-FHIC model, which means the ETR enabled it to obtain more features, as seen in Table V. Additionally, during the execution time, the ELU activation function was more beneficial than ReLu; it contributed to increasing the precision of the FHIC model. The ELU AF succeeded in normalizing the generated weight of each neuron, which helped speed the processing, and dealt with the negative and zero values

TABLE VI

THE RESULTS OF THE NINE COMPARED CLASSIFICATION MODELS FOR THE HH DATASET.

#	Training Samples	Testing Samples	MLP	CNN1D	CNN2D	CNN3D	SNet	MCNN	HCNN	R-FHIC	FHIC
1	2808	11233	0.93	0.94	1	1	1	1	1	0.99	1
2	702	2810	0.77	0.82	0.99	0.99	0.99	0.97	0.98	0.97	0.97
3	4364	17457	0.91	0.91	1	1	1	1	0.99	1	1
4	32657	130628	0.98	0.98	1	1	1	1	1	1	1
5	1244	4974	0.40	0.60	1	1	1	0.99	1	0.98	1
6	8912	35645	0.89	0.91	1	1	1	1	0.99	1	1
7	4821	19282	0.74	0.78	1	1	1	1	0.99	1	1
8	811	3243	0.18	0.24	1	1	0.99	0.99	0.99	0.97	1
9	2164	8655	0.93	0.94	1	1	1	1	1	1	1
10	2479	9915	0.55	0.61	1	1	1	1	1	0.99	0.99
11	2203	8812	0.41	0.50	1	1	1	1	0.99	0.99	0.99
12	1791	7163	0.51	0.55	1	1	1	0.99	0.99	0.99	0.99
13	4502	18005	0.71	0.72	1	1	1	0.99	1	0.99	1
14	1471	5885	0.64	0.73	1	1	0.99	1	0.99	0.99	0.99
15	200	802	0.38	0.41	0.97	1	0.98	0.99	0.98	0.99	0.98
16	1452	5810	0.83	0.85	1	1	1	1	0.99	1	1
17	602	2408	0.59	0.72	1	1	1	1	1	0.99	1
18	643	2574	0.49	0.59	1	1	1	1	0.98	0.95	0.99
19	1742	6970	0.76	0.75	1	1	0.99	0.99	0.99	0.99	0.99
20	697	2789	0.71	0.78	0.99	0.99	0.99	0.99	1	0.99	1
21	266	1062	0	0.11	0.99	1	1	1	0.99	0.99	0.99
22	808	3232	0.46	0.56	1	1	1	0.99	0.99	0.99	1
	KA (%)		79.27±0.29	81.92±0.09	99.84±0.01	99.93±0.01	99.71±0.11	99.68±0.08	99.30±0.65	99.38±0.59	99.65±0.54
	OA (%)		83.78±0.22	85.81±0.07	99.87±0.01	99.94±0.01	99.77±0.08	99.74±0.06	99.45±0.52	99.51±0.47	99.73±0.43
	AA (%)		62.20±0.78	67.67±0.40	99.66±0.07	99.86±0.02	99.49±0.15	99.41±0.12	98.99±1	98.83±0.62	99.24±1.18
	TT (s)		523.12	693.69	892.12	1673.30	427.73	560.73	1108.36	359.66	273.31
	ST (s)		10.94	19.56	27.13	37.38	46.62	46.34	76.71	43.40	31.35

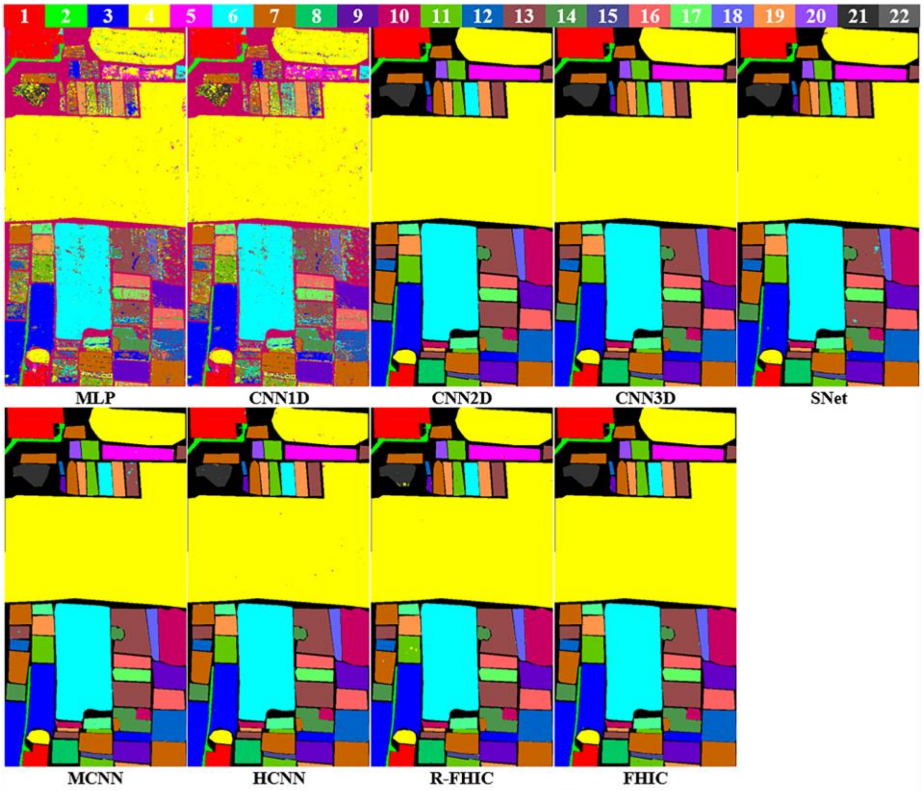
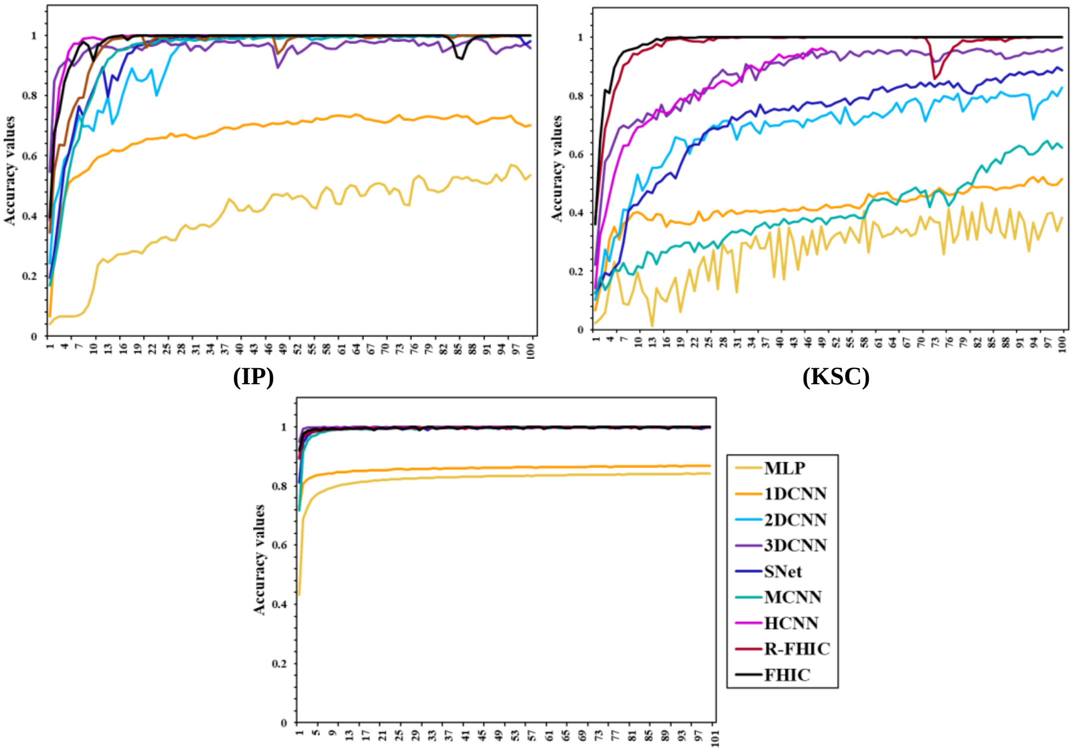


Fig. 12. The output of the nine used classification models for the HH dataset.



(IP) **(KSC)**

(HH)

Fig. 13. The accuracy range during the training time in the nine compared classification models for the three datasets.

enabled the model to be fast and more accurate, Table V.

The accuracy and training time of the 3DCNN model were superior to those of the MCNN model since the input data of the 3DCNN model had 30 bands rather than 15. Although the MLP model had the shortest test duration, its accuracy was the lowest. The HCNN model did not provide high performance because it used different sizes of windows, which led to the loss of some spatial information and increased the calculation time. All other models were less accurate than the FHIC model. According to the comparison of all models, the FHIC model was the most accurate and fastest in execution time.

The classification output visualization for all models applied to the KSC dataset is depicted in Fig. 11.

6) The WHU-Hi-HongHu Dataset Results

The HH dataset is totally different from the IP and KSC datasets. It is a massive dataset with a greater number of bands and classes, longer dimensions, and higher resolution. In addition, Fig. 7 reveals that the data in the HH dataset has a normal distribution. Therefore, it is a very rich dataset with a large number of pixels. Because of its enormous size, it drained the RAM and time more than the others.

Because the 3DCNN model accepts at least 30 bands of input data, it obtained the highest accuracy and the longest training time. The 2DCNN model with 15 bands of input data was the most accurate. The second model was SNet, which completed training in half the time of 2DCNN due to the dropout approach, followed by the MCNN model with the longest training time, and after that, the FHIC model with the best training time. The difference in precision between these four models was very tiny, but the FHIC introduced a great speed, significantly impacting the execution time, as observed in Table VI. Regarding the training time, the FHIC model was 60% faster than the most accurate model, the 2DCNN model, and 75% faster than the slowest model, the HCNN. The classification output visualization of all models for the HH dataset is in Fig. 12.

From the training time of the R-FHIC model in Tables IV-VI for the three datasets, we can realize the contribution and effect of the ETR method in enhancing the speed of the classification process. The R-FHIC was the second fastest model after the FHIC model. Moreover, when we look at the performance of the FHIC model for the three datasets, the ELU AF's advantages in improving the accuracy and time also can be noted. Because of the ETR method and ELU AF, the FHIC model was the top fastest model for all datasets.

7) Compared FHIC With Recently Published Models

In the last section, the proposed model was compared to known spectral extraction models (MLP and CNN1D), spatial extraction models (CNN2D), and spectral-spatial extraction models (CNN3D, SNet, MCNN, and HCNN). This section compares the FHIC model to the spectral-spatial extraction models that were recently published in 2022 and 2023. These models are spatial pooling graph convolutional network (SPGCN-21) [50], IDA-HybridCNN (IHCNN) [51], multi-direction network-attentional spectral prior (MDN-ASP) [52], SST-M [53], adaptive hash attention mechanism and a lower triangular network (AHA-LT) [54], multi-hybrid deep learning model (MHDL) [8], ELUSNet [55], hybrid spectral CNN with ETR (ETRSN) [7], pyramidal coordinate attention and weighted self-distillation (PCA-WSD) [56], subspace classifier and feature transformation (SSFT) [57], and fast dynamic graph convolutional network and CNN (FDGC) [58].

TABLE VII
THE COMPETITION WITH THE RECENTLY PUBLISHED SPECTRAL-SPATIAL EXTRACTION MODELS FOR THE IP DATASET. THE BLACK-BOLD STYLE REPRESENTS THE BEST RESULTS, THE GREEN-BOLD STYLE REPRESENTS THE SECOND-BEST, AND THE BLUE-BOLD STYLE MEANS THE THIRD-BEST.

Models	**KA**	**OA**	**AA**	**TT(s)**	**ST(s)**
SPGCN-21 [50]	98.53+0.40	97.89±0.89	98.71+0.35	1117.67	44.34
IHCNN [51]	94.54±0.38	95.22±0.33	90.77±1.29	942.26	21.18
MDN-ASP [52]	93.09±0.22	93.99±0.31	94.55±0.52	725.41	21.63
SST-M [53]	**98.95±0.36**	**99.08±0.31**	**99.01±0.33**	658.14	3.43
AHA-LT [54]	97.56±0.19	97.86±0.17	97.89±0.20	317.4	13.51
MHDL [8]	89.73±0.64	91.03±0.56	82.63±1.93	212.57	4.97
ELUSNet [55]	98.57±0.33	98.75±0.29	97.23±0.21	208.97	5.02
ETRSN [7]	93.96±9.58	94.75±8.30	92.61±10.92	200.77	4.79
PCA-WSD [56]	**99.27±0.11**	**99.36±0.09**	**99.44±0.15**	166.56	4.63
SSFT [57]	83.31±1.74	85.21±1.58	77.44±2.11	**142**	**2.7**
FDGC [58]	98.00±0.33	98.27±0.28	96.81±1.02	**20.57**	**0.55**
FHIC	**98.65±0.17**	**98.82±0.15**	**98.14±0.74**	**9.96**	**1.03**

Table VII was sorted from longest to shortest training time. Evidently, the proposed model (FHIC) is the fastest. The FHIC model is ten seconds faster than the next-fastest model (FDGC). In addition, the accuracy of the best three models (PCA-WSD, SST-M, and FHIC) is virtually identical; there are no significant differences between them. In contrast, the difference in speed between FHIC and the most accurate model is more than 156 seconds. Therefore, compared with the other 11 hyperspectral image classification models, the proposed model is optimal for achieving an equilibrium between extremely high accuracy and processing speed.

Fig. 13 depicts the accuracy situation for the three datasets across the processing of the nine classification models throughout the training period. In contrast to the other models, the FHIC model was the most stable and provided high accuracy from the beginning to the end, especially with the IP and KSC datasets, which are more complex than the HH dataset.

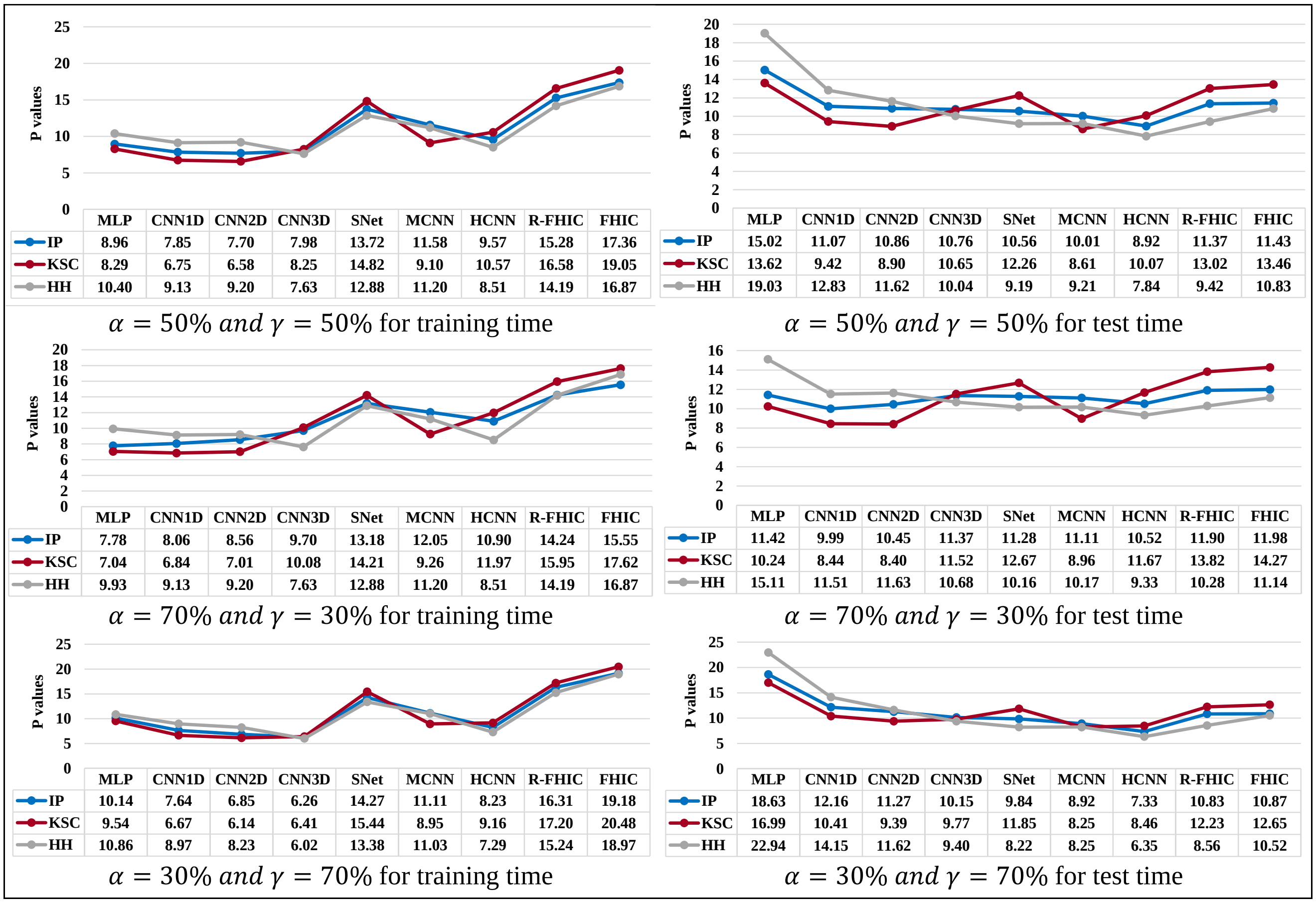
IV. EFFECTIVENESS MEASURE

As is known, the best model is a model that provides the best accuracy in a short time compared to the others. So, this paper provides an Equation that compares used models according to their accuracy and training/testing time. As can be seen from Tables IV-VI, some models achieved very low accuracy over a very short training time, like MLP and CNN1D. CNN3D and HCNN gave very high accuracy over a very long training time. The SNet, MCNN, R-FHIC, and FHIC models produced different values for accuracy and training time, complicating their differentiation. This Equation is an easy and quick measure to evaluate the models' efficacy and compare them quickly. We will deal with the accuracy and time as numbers in this measurement and ignore their units. To make the comparison as reasonable as possible, we must first scale the comparison's parameters. The accuracy values are scaled using the following Equation:

$$A = \left(\Lambda_i / \sum_{i=1}^{i} \Lambda_i\right) \times 100\ \% \qquad (6)$$

where Λ is the model accuracy, i is the model's number.

After that, because the execution time should be short and the



	MLP	CNN1D	CNN2D	CNN3D	SNet	MCNN	HCNN	R-FHIC	FHIC
IP	8.96	7.85	7.70	7.98	13.72	11.58	9.57	15.28	17.36
KSC	8.29	6.75	6.58	8.25	14.82	9.10	10.57	16.58	19.05
HH	10.40	9.13	9.20	7.63	12.88	11.20	8.51	14.19	16.87

$\alpha = 50\%$ *and* $\gamma = 50\%$ for training time

	MLP	CNN1D	CNN2D	CNN3D	SNet	MCNN	HCNN	R-FHIC	FHIC
IP	15.02	11.07	10.86	10.76	10.56	10.01	8.92	11.37	11.43
KSC	13.62	9.42	8.90	10.65	12.26	8.61	10.07	13.02	13.46
HH	19.03	12.83	11.62	10.04	9.19	9.21	7.84	9.42	10.83

$\alpha = 50\%$ *and* $\gamma = 50\%$ for test time

	MLP	CNN1D	CNN2D	CNN3D	SNet	MCNN	HCNN	R-FHIC	FHIC
IP	7.78	8.06	8.56	9.70	13.18	12.05	10.90	14.24	15.55
KSC	7.04	6.84	7.01	10.08	14.21	9.26	11.97	15.95	17.62
HH	9.93	9.13	9.20	7.63	12.88	11.20	8.51	14.19	16.87

$\alpha = 70\%$ *and* $\gamma = 30\%$ for training time

	MLP	CNN1D	CNN2D	CNN3D	SNet	MCNN	HCNN	R-FHIC	FHIC
IP	11.42	9.99	10.45	11.37	11.28	11.11	10.52	11.90	11.98
KSC	10.24	8.44	8.40	11.52	12.67	8.96	11.67	13.82	14.27
HH	15.11	11.51	11.63	10.68	10.16	10.17	9.33	10.28	11.14

$\alpha = 70\%$ *and* $\gamma = 30\%$ for test time

	MLP	CNN1D	CNN2D	CNN3D	SNet	MCNN	HCNN	R-FHIC	FHIC
IP	10.14	7.64	6.85	6.26	14.27	11.11	8.23	16.31	19.18
KSC	9.54	6.67	6.14	6.41	15.44	8.95	9.16	17.20	20.48
HH	10.86	8.97	8.23	6.02	13.38	11.03	7.29	15.24	18.97

$\alpha = 30\%$ *and* $\gamma = 70\%$ for training time

	MLP	CNN1D	CNN2D	CNN3D	SNet	MCNN	HCNN	R-FHIC	FHIC
IP	18.63	12.16	11.27	10.15	9.84	8.92	7.33	10.83	10.87
KSC	16.99	10.41	9.39	9.77	11.85	8.25	8.46	12.23	12.65
HH	22.94	14.15	11.62	9.40	8.22	8.25	6.35	8.56	10.52

$\alpha = 30\%$ *and* $\gamma = 70\%$ for test time

Fig. 14. The Effectiveness Measure of all models for all used datasets with different significance coefficients.

accuracy should be high, the relationship between them is reciprocal, as seen in the following Equation:

$$t_i = 1/T_i \tag{7}$$

where t is the reciprocal relationship of the execution time T, now the scaling time can be taken as follows:

$$B = \ (t_i/\sum_{i=1}^{i} t_i) \times 100\ \% \tag{8}$$

Finally, after scaling the accuracy values and time, the percent of linear combination can be calculated for A and B as follows:

$$P = (\alpha \times A) + (\gamma \times B), \qquad where \quad \alpha + \gamma = 100\ \% \tag{9}$$

Where P represents the standard measurement evaluation between the models according to their accuracy A and time B, α is the significance coefficient for accuracy, and γ is the significance coefficient for time, α and γ values are between zero and one. Equation (9) aims to find an independent approach to evaluate the effectiveness of any model compared to others.

First, the effectiveness was tested with the same significance coefficients for the accuracy and time (training and testing), where α and γ = 50%. The second evaluation was α = 70% and γ = 30%; here, more attention was given to accuracy than time. Third, the effectiveness was evaluated with more attention to time rather than accuracy, where α = 30% and γ = 70%.

If we evaluated the effectiveness according to the accuracy and training time, the FHIC model was the best whatever the values of α and γ, as seen in Fig. 14. On the other hand, evaluating the effectiveness according to accuracy and test time, with 50% for α and γ, the FHIC got the second position after MLP. With less attention to the test time (γ = 30%), the FHIC took the top position in the IP and KSC datasets. Finally, with less attention to the accuracy (α = 30%), the FHIC is in the second position with the KSC dataset and fourth with the IP and HH datasets. Furthermore, from the results in Fig. 14, it can be noted that the less complex models like MLP, 1DCNN, 2DCNN, and HFIC were more flexible and provided more stable results with the different HSIs than the deep and complex models like MSCNN and HCNN.

In the field of artificial intelligence, the efficacy and speed of the model are of utmost importance. The advantage of this evaluation method is that it provides an easy measure for testing the effectiveness of models based on their performance, not just their accuracy, and for identifying the models with the best results and time.

V. Conclusion

This study aimed to demonstrate that pre-enhancing the input data and application of appropriate methods are more crucial for achieving excellent performance than employing a complex classification model. It employed a novel diminution reduction method, the enhancing transformation reduction (ETR) method, to reduce the size of the HSIs, and an efficient activation function, the exponential linear units (ELU) AF, to optimize the classification process inside every neuron. The classification part was a simple network to extract feature maps using the Conv2D approach. The study model is called the fast hyperspectral image classification model (FHIC) model. The process of the classification depends on the ETR to simplify and reduce the HSI complexity and on the ELU AF to normalize the inter-process and improve the extraction. We have shown that many well-known classification models were used to assess and demonstrate the effectiveness of the proposed model through three different HSI datasets. This research also introduced a new effective measure method to provide an easy way for the researchers to evaluate the effectiveness of their models against others according to significance coefficients. From the experiments and comparison results, notes can be summarized as the following: 1) The content and the complexities are different from one HSI to the other, so a flexible method that can deal with the different complexities of the different HSIs is needed. 2) The ETR is a very quick diminution reduction method, and it introduced an impressive performance in correcting the classification of pixels and enhancing their correlation; it also enhanced classification model performance. 3) Due to the high number of negative and zero values in the input data, the FHIC model operated more quickly and efficiently with the ELU activation function than with the Relu. 4) The less complex models are more flexible in dealing with the different HSIs than the deep and complex models. 5) The proposed model reduced the use of RAM and saved time. 6) The proposed model outperformed the other state-of-the-art models and gave the best performance according to the effectiveness measure in terms of accuracy and execution time for the three HSIs. Future work will focus on improving the test time.

References

[1] M. Vidal and J. M. Amigo, “Pre-processing of hyperspectral images. Essential steps before image analysis,” *Chemom. Intell. Lab. Syst.*, vol. 117, pp. 138–148, Aug. 2012, doi: 10.1016/j.chemolab.2012.05.009.

[2] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, “Deep learning classifiers for hyperspectral imaging: A review,” *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 279–317, 2019, doi: https://doi.org/10.1016/j.isprsjprs.2019.09.006.

[3] B. Rasti, P. Scheunders, P. Ghamisi, G. Licciardi, and J. Chanussot, “Noise Reduction in Hyperspectral Imagery: Overview and Application,” *Remote Sens.*, vol. 10, no. 3, p. 482, Mar. 2018, doi: 10.3390/rs10030482.

[4] W. Ma *et al.*, “Hyperspectral image classification based on spatial and spectral kernels generation network,” *Inf. Sci. (Ny).*, vol. 578, pp. 435–456, Nov. 2021, doi: 10.1016/j.ins.2021.07.043.

[5] D. Li, Q. Wang, and F. Kong, “Adaptive kernel sparse representation based on multiple feature learning for hyperspectral image classification,” *Neurocomputing*, vol. 400, pp. 97–112, Aug. 2020, doi: 10.1016/j.neucom.2020.03.022.

[6] J. Fang and X. Cao, “Multidimensional relation learning for hyperspectral image classification,” *Neurocomputing*, vol. 410, pp. 211–219, Oct. 2020, doi: 10.1016/j.neucom.2020.05.034.

[7] D. AL-Alimi, Z. Cai, M. A. A. Al-qaness, E. Ahmed Alawamy, and A. Alalimi, “ETR: Enhancing transformation reduction for reducing dimensionality and classification complexity in hyperspectral images,” *Expert Syst. Appl.*, vol. 213, p. 118971, Mar. 2023, doi: 10.1016/j.eswa.2022.118971.

[8] D. AL-Alimi, Z. Cai, M. A. A. Al-qaness, A. Dahou, E. A. Alawamy, and S. Issaka, “Compression and reinforce variation with convolutional neural networks for hyperspectral image classification,” *Appl. Soft Comput.*, vol. 130, p. 109650, Nov. 2022, doi: 10.1016/j.asoc.2022.109650.

[9] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, “HybridSN: Exploring 3-D–2-D CNN Feature Hierarchy for Hyperspectral Image Classification,” *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 277–281, Feb. 2020, doi: 10.1109/LGRS.2019.2918719.

[10] D. AL-Alimi, M. A. A. Al-qaness, Z. Cai, A. Dahou, Y. Shao, and S. Issaka, “Meta-Learner Hybrid Models to Classify Hyperspectral Images,” *Remote Sens.*, vol. 14, no. 4, p. 1038, Feb. 2022, doi: 10.3390/rs14041038.

[11] F. Cao and W. Guo, “Cascaded dual-scale crossover network for hyperspectral image classification,” *Knowledge-Based Syst.*, vol. 189, p. 105122, Feb. 2020, doi: 10.1016/j.knosys.2019.105122.

[12] W. Wang, Y. Han, C. Deng, and Z. Li, “Hyperspectral Image Classification via Deep Structure Dictionary Learning,” *Remote Sens.*, vol. 14, no. 9, p. 2266, May 2022, doi: 10.3390/rs14092266.

[13] S. K. Roy, S. R. Dubey, S. Chatterjee, and B. Baran Chaudhuri, “FuSENet: fused squeeze-and-excitation network for spectral-spatial hyperspectral image classification,” *IET Image Process.*, vol. 14, no. 8, pp. 1653–1661, Jun. 2020, doi: 10.1049/iet-ipr.2019.1462.

[14] S. Ghaderizadeh, D. Abbasi-Moghadam, A. Sharifi, N. Zhao, and A. Tariq, “Hyperspectral Image Classification Using a Hybrid 3D-2D Convolutional Neural Networks,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 14, pp. 7570–7588, 2021, doi: 10.1109/JSTARS.2021.3099118.

[15] J. Zheng, Y. Feng, C. Bai, and J. Zhang, “Hyperspectral Image Classification Using Mixed Convolutions and Covariance Pooling,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 522–534, Jan. 2021, doi: 10.1109/TGRS.2020.2995575.

[16] A. Paul, S. Bhoumik, and N. Chaki, “SSNET: an improved deep hybrid network for hyperspectral image classification,” *Neural Comput. Appl.*, vol. 33, no. 5, pp. 1575–1585, Mar. 2021, doi: 10.1007/s00521-020-05069-1.

[17] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao, “Active Transfer Learning Network: A Unified Deep Joint Spectral–Spatial Feature Learning Model for Hyperspectral Image Classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 3, pp. 1741–1754, Mar. 2019, doi: 10.1109/TGRS.2018.2868851.

[18] Y. Liu, L. Gao, C. Xiao, Y. Qu, K. Zheng, and A. Marinoni, “Hyperspectral Image Classification Based on a Shuffled Group Convolutional Neural Network with Transfer Learning,” *Remote Sens.*, vol. 12, no. 11, p. 1780, Jun. 2020, doi: 10.3390/rs12111780.

[19] N. Wambugu *et al.*, “Hyperspectral image classification on insufficient-sample and feature learning using deep neural networks: A review,” *Int. J. Appl. Earth Obs. Geoinf.*, vol. 105, p. 102603, Dec. 2021, doi: 10.1016/j.jag.2021.102603.

[20] L. Yang, F. Zhang, P. S.-P. Wang, X. Li, and Z. Meng, “Multi-scale spatial-spectral fusion based on multi-input fusion calculation and coordinate attention for hyperspectral image classification,” *Pattern Recognit.*, vol. 122, p. 108348, Feb. 2022, doi: 10.1016/j.patcog.2021.108348.

[21] R. Hang, Z. Li, Q. Liu, P. Ghamisi, and S. S. Bhattacharyya, “Hyperspectral Image Classification With Attention-Aided CNNs,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2281–2293, Mar. 2021, doi: 10.1109/TGRS.2020.3007921.

[22] S. Pande and B. Banerjee, “HyperLoopNet: Hyperspectral image classification using multiscale self-looping convolutional networks,” *ISPRS J. Photogramm. Remote Sens.*, vol. 183, pp. 422–438, Jan. 2022, doi: 10.1016/j.isprsjprs.2021.11.021.

[23] Y. Feng, J. Zheng, M. Qin, C. Bai, and J. Zhang, “3D Octave and 2D Vanilla Mixed Convolutional Neural Network for Hyperspectral Image Classification with Limited Samples,” *Remote Sens.*, vol. 13, no. 21, p. 4407, Nov. 2021, doi: 10.3390/rs13214407.

[24] L. Huang and Y. Chen, “Dual-Path Siamese CNN for Hyperspectral Image Classification With Limited Training Samples,” *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 3, pp. 518–522, Mar. 2021, doi: 10.1109/LGRS.2020.2979604.

[25] R. Chen and G. Li, “Spectral-spatial feature fusion via dual-stream deep architecture for hyperspectral image classification,” *Infrared Phys. Technol.*, vol. 119, p. 103935, Dec. 2021, doi: 10.1016/j.infrared.2021.103935.

[26] C. Pu, H. Huang, and L. Yang, “An attention-driven convolutional neural network-based multi-level spectral–spatial feature learning for hyperspectral image classification,” *Expert Syst. Appl.*, vol. 185, p. 115663, Dec. 2021, doi: 10.1016/j.eswa.2021.115663.

[27] M. Bandyopadhyay, "Multi-stack hybrid CNN with non-monotonic activation functions for hyperspectral satellite image classification," *Neural Comput. Appl.*, vol. 33, no. 21, pp. 14809–14822, Nov. 2021, doi: 10.1007/s00521-021-06120-5.

[28] Y. Guo, H. Cao, J. Bai, and Y. Bai, "High Efficient Deep Feature Extraction and Classification of Spectral-Spatial Hyperspectral Image Using Cross Domain Convolutional Neural Networks," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 12, no. 1, pp. 1–12, 2019, doi: 10.1109/JSTARS.2018.2888808.

[29] S. Ghanbari Azar, S. Meshgini, T. Yousefi Rezaii, and S. Beheshti, "Hyperspectral image classification based on sparse modeling of spectral blocks," *Neurocomputing*, vol. 407, pp. 12–23, Sep. 2020, doi: 10.1016/j.neucom.2020.04.138.

[30] X. Tu, X. Shen, P. Fu, T. Wang, Q. Sun, and Z. Ji, "Discriminant sub-dictionary learning with adaptive multiscale superpixel representation for hyperspectral image classification," *Neurocomputing*, vol. 409, pp. 131–145, Oct. 2020, doi: 10.1016/j.neucom.2020.05.082.

[31] J. An, X. Zhang, H. Zhou, J. Feng, and L. Jiao, "Patch Tensor-Based Sparse and Low-Rank Graph for Hyperspectral Images Dimensionality Reduction," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 11, no. 7, pp. 2513–2527, Jul. 2018, doi: 10.1109/JSTARS.2018.2833886.

[32] Z. Qiumei, T. Dan, and W. Fenghua, "Improved Convolutional Neural Network Based on Fast Exponentially Linear Unit Activation Function," *IEEE Access*, vol. 7, pp. 151359–151367, 2019, doi: 10.1109/ACCESS.2019.2948112.

[33] S. U. Amin, M. Alsulaiman, G. Muhammad, M. A. Mekhtiche, and M. Shamim Hossain, "Deep Learning for EEG motor imagery classification based on multi-layer CNNs feature fusion," *Futur. Gener. Comput. Syst.*, vol. 101, pp. 542–554, Dec. 2019, doi: 10.1016/j.future.2019.06.027.

[34] L. Yang, W. Chen, W. Liu, B. Zha, and L. Zhu, "Random Noise Attenuation Based on Residual Convolutional Neural Network in Seismic Datasets," *IEEE Access*, vol. 8, pp. 30271–30286, 2020, doi: 10.1109/ACCESS.2020.2972464.

[35] M. Z. Alom *et al.*, "A State-of-the-Art Survey on Deep Learning Theory and Architectures," *Electronics*, vol. 8, no. 3, p. 292, Mar. 2019, doi: 10.3390/electronics8030292.

[36] Z. Sun, L. Xie, D. Hu, and Y. Ying, "An artificial neural network model for accurate and efficient optical property mapping from spatial-frequency domain images," *Comput. Electron. Agric.*, vol. 188, p. 106340, Sep. 2021, doi: 10.1016/j.compag.2021.106340.

[37] N. Wu, S. Weng, J. Chen, Q. Xiao, C. Zhang, and Y. He, "Deep convolution neural network with weighted loss to detect rice seeds vigor based on hyperspectral imaging under the sample-imbalanced condition," *Comput. Electron. Agric.*, vol. 196, p. 106850, May 2022, doi: 10.1016/j.compag.2022.106850.

[38] S. Fan *et al.*, "On line detection of defective apples using computer vision system combined with deep learning methods," *J. Food Eng.*, vol. 286, p. 110102, Dec. 2020, doi: 10.1016/j.jfoodeng.2020.110102.

[39] L. Zhang, D. An, Y. Wei, J. Liu, and J. Wu, "Prediction of oil content in single maize kernel based on hyperspectral imaging and attention convolution neural network," *Food Chem.*, p. 133563, Jun. 2022, doi: 10.1016/j.foodchem.2022.133563.

[40] L. E. C. La Rosa, C. Sothe, R. Q. Feitosa, C. M. de Almeida, M. B. Schimalski, and D. A. B. Oliveira, "Multi-task fully convolutional network for tree species mapping in dense forests using small training hyperspectral data," *ISPRS J. Photogramm. Remote Sens.*, vol. 179, pp. 35–49, Sep. 2021, doi: 10.1016/j.isprsjprs.2021.07.001.

[41] L. Pang, L. Wang, P. Yuan, L. Yan, and J. Xiao, "Rapid seed viability prediction of Sophora japonica by improved successive projection algorithm and hyperspectral imaging," *Infrared Phys. Technol.*, vol. 123, p. 104143, Jun. 2022, doi: 10.1016/j.infrared.2022.104143.

[42] Y. Dixit, M. Al-Sarayreh, C. R. Craigie, and M. M. Reis, "A global calibration model for prediction of intramuscular fat and pH in red meat using hyperspectral imaging," *Meat Sci.*, vol. 181, p. 108405, Nov. 2021, doi: 10.1016/j.meatsci.2020.108405.

[43] Z. An, J. Zhang, Z. Sheng, X. Er, and J. Lv, "RBDN: Residual Bottleneck Dense Network for Image Super-Resolution," *IEEE Access*, vol. 9, pp. 103440–103451, 2021, doi: 10.1109/ACCESS.2021.3096548.

[44] M. Ahmad *et al.*, "Hyperspectral Image Classification—Traditional to Deep Models: A Survey for Future Prospects," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 15, pp. 968–999, 2022, doi: 10.1109/JSTARS.2021.3133021.

[45] D. Erhan, A. Courville, and P. Vincent, "Why Does Unsupervised Pre-training Help Deep Learning ?," *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, 2010, doi: 10.1145/1756006.1756025.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[47] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 936–944, 2017, doi: 10.1109/CVPR.2017.106.

[48] Y. Zhong, X. Hu, C. Luo, X. Wang, J. Zhao, and L. Zhang, "WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF," *Remote Sens. Environ.*, vol. 250, p. 112012, Dec. 2020, doi: 10.1016/j.rse.2020.112012.

[49] A. Mohan and M. Venkatesan, "HybridCNN based hyperspectral image classification using multiscale spatiospectral features," *Infrared Phys. Technol.*, vol. 108, p. 103326, Aug. 2020, doi: 10.1016/j.infrared.2020.103326.

[50] X. Zhang, S. Chen, P. Zhu, X. Tang, J. Feng, and L. Jiao, "Spatial Pooling Graph Convolutional Network for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022, doi: 10.1109/TGRS.2022.3140353.

[51] D. AL-Alimi, M. A. A. Al-qaness, Z. Cai, and E. A. Alawamy, "IDA: Improving distribution analysis for reducing data complexity and dimensionality in hyperspectral images," *Pattern Recognit.*, vol. 134, p. 109096, Feb. 2023, doi: 10.1016/j.patcog.2022.109096.

[52] B. Xi *et al.*, "Multi-Direction Networks With Attentional Spectral Prior for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022, doi: 10.1109/TGRS.2020.3047682.

[53] J. Bai *et al.*, "Hyperspectral Image Classification Based on Multibranch Attention Transformer Networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022, doi: 10.1109/TGRS.2022.3196661.

[54] Z. Ge, G. Cao, Y. Zhang, X. Li, H. Shi, and P. Fu, "Adaptive Hash Attention and Lower Triangular Network for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–19, 2022, doi: 10.1109/TGRS.2021.3075546.

[55] D. AL-Alimi, M. A. A. Al-qaness, and Z. Cai, "Speeding Up and Enhancing the Hyperspectral Images Classification," M. Abd Elaziz, M. Medhat Gaber, S. El-Sappagh, M. A. A. Al-qaness, and A. A. Ewees, Eds. Cham: Springer Nature Switzerland, 2023, pp. 53–62. doi: 10.1007/978-3-031-28106-8_4.

[56] R. Shang *et al.*, "Hyperspectral Image Classification Based on Pyramid Coordinate Attention and Weighted Self-Distillation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022, doi: 10.1109/TGRS.2022.3224604.

[57] J. Bai *et al.*, "Few-Shot Hyperspectral Image Classification Based on Adaptive Subspaces and Feature Transformation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022, doi: 10.1109/TGRS.2022.3149947.

[58] Q. Liu, Y. Dong, Y. Zhang, and H. Luo, "A Fast Dynamic Graph Convolutional Network and CNN Parallel Network for Hyperspectral Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022, doi: 10.1109/TGRS.2022.3179419.