HYPERSPECTRAL IMAGES CLASSIFICATION WITH DEEP BAYESIAN NEURAL NETWORKS

[1]MAHMOOD SIDDEEQ QADIR, [2]GOKHAN BILGIN

Department of Computer Engineering, Yildiz Technical University, Istanbul, Turkey
E-mail: [1]mahmood.qadir@std.yildiz.edu.tr, [2]gbilgin@yildiz.edu.tr

Abstract - The classification of Hyperspectral Image HSI is important in various fields where more discriminative characteristics are provided by the hundreds of narrow-band radiation information. Bayesian CNN is a very powerful method used in various difficult classification problems. In this work, Bayesian CNN model was built and applied on Pavia dataset. In order to compare the Bayesian results with other methods, two other approaches were applied. Machine learning methods are more frequently used, which uses both labeled and unlabeled data to fit the model. SVM and RF were used. Pretrained deep learning models were also applied. The results show that the Bayesian CNN method gives the best accuracy of 99%. Among pretrained deep learning networks, Xception gives the best 97%. SVM with the Radial Basis Function (RBF) kernel gives 96% accuracy.

Keywords - Hyperspectral Image, Pavia University Dataset, SVM, RF, Bayesian CNN, Pretrained Deep Learning Model, XCEPTION.

I. INTRODUCTION

The creation of hyperspectral sensors and the associated software to evaluate the picture data they produce is the most significant recent development in remote sensing. Hyperspectral image analysis has developed over the last ten years into one of the most potent and rapidly expanding technologies in the world of remote sensing. The word "hyper" in "hyperspectral" alludes to the numerous measured wavelength bands and signifies "over" as in "too many." The ability to detect and differentiate spectrally distinct materials is made possible by the fact that hyperspectral pictures are spectrally overdetermined. An especially significant picture type is the hyperspectral image (HSI). Each pixel in the image has a distinct spectral structure that may be utilized to recognize ground objects that are invisible to the human eye.

HSI offer spectral and spatial representations of objects, materials, and light sources. HSI are distinct from pictures taken with a typical RGB-color camera in two main things, the first is that the number of tiny picture slices that may be efficiently separated from the light spectrum by a hyperspectral camera varies on the camera and the application [1]. The second, RGB color camera separates the light spectrum into wide, overlapping red, green, and blue picture slices, which when combined seem realistic to the human eye. Although it may not be visible to the naked eye or an RGB camera, this fine-grained slicing in HSI exposes spectral structure that manifests in a variety of visual and optical phenomena, including as metamerism and color constancy [2]. Field and laboratory spectrometers often measure reflectance at a number of discrete wavelength bands that are spaced closely together, giving the resultant spectra the appearance of being continuous curves. When a spectrometer is included into an imaging sensor, the resultant pictures capture a reflectance spectrum for each pixel in the image. Each pixel's whole spectrum is produced as a consequence of measurements performed at several tiny, adjacent wavelength bands used to create the image. HSI may be seen as a cube with two spatial dimensions (pixels) and one spectral dimension (wavelength) [3]. There is a comprehensive intensity (grey-level) depiction of the scene's reflectance or radiance at each sample wavelength. As an example, shown in Figure 1, Pavia data could be presented as a traditional color image (left) and effective spectral reflectance at corresponding regions (right).

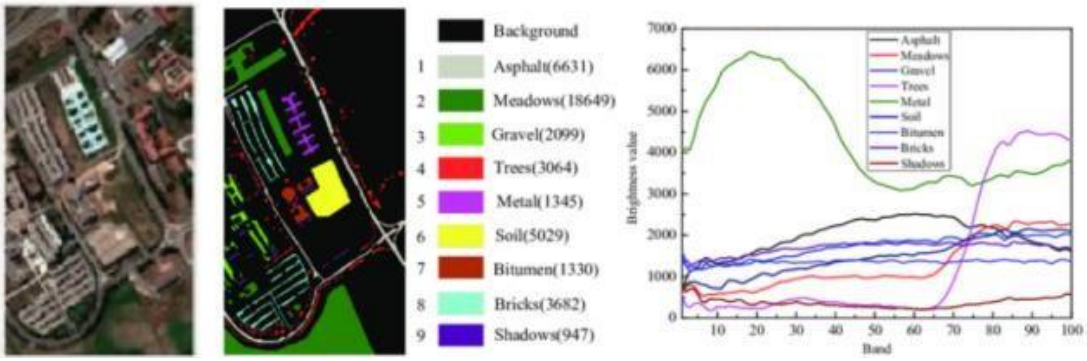

Figure 1. Spectral reflectance, bands of HSI (Pavia) [4]

Applications in the real world frequently deal with multi-view data. Color information and texture information are two distinct types of elements seen in photos and movies that may be thought of as two-view data [4]. One crucial aspect of HSI research is the categorization job. The main feature of the HSI is that the number of bands is typically greater than hundreds, which makes the categorization process challenging. Therefore, accurate categorization is a crucial part of examining the characteristics of ground objects.

Supervised techniques, which are a subset of machine learning, are more typically employed for the HSI classification, although deep learning is also a significant subset of machine learning. Convolutional neural networks, in particular, show astonishing performance in the field of image identification using deep learning methods (CNN).

Numerous studies have been done to address the issue of insufficient training data. The solutions, according to [4], can be divided into three categories, the first is creating new designs or modifying existing architectures to incorporate techniques like regularizers and data augmentation to improve network performance in this situation [5], [6]. The second is reducing the feature vectors' dimensionality to provide the classifiers with more detailed information. The third is generating synthetic data artificially to produce more training data [4], [5], and [7]. Since the spectral channels in HSRS pictures have a great deal of redundancy, using dimensionality reduction techniques, such PCA, is quite successful. In [8], PCA is employed to reduce the dimensionality of edge- preserving filters. Before passing the filters to the classifier, their dimensionality is decreased. The authors demonstrate the potency and performance enhancement of such feature vectors. For example, enhanced multi-attribute profile (EMAP) [9] uses PCA after the feature vector to provide the model a more potent and informative feature vector [4], [5]. In their footsteps, we modify this strategy and employ PCA to enhance performance.

Recent years have seen a boom in the study of HIC methods based on the merging of spatial and spectral features. The extraction of spatial features frequently uses neighborhood spatial features, discrete Gabor transforms, and discrete wavelet transforms. There are several types of spatial spectrum feature fusion techniques. [6] suggested a PCA- based windowed wavelet transform hyperspectral decision fusion classification algorithm. For the dimensionality reduction of hyperspectral images, [7] use a modified tensor locality-preserving projection. Considering the deep learning topic of research. Convolutional neural networks were used to solve the HIC problem in [8, 9, 10], and the results were outstanding.

Several techniques were used on a variety of HIC datasets. [11] compared Support Machine Classifiers, k-Nearest Neighbor, and Random Forest for Land Cover Classification Using Sentinel-2 Imagery. Overall accuracy is 90–95%; SVM performs best, followed by RF, KNN, and then RF again. [12] used a variety of categorization techniques. According to his findings using data from the Pavia University, SVM performs best overall (88% accuracy), followed by 3-NN (84% accuracy). Support vector machine (SVM), K-Nearest neighbor (KNN), and convolutional neural network (CNN) comparative studies were performed by [13] on the Pavia University dataset. The findings suggest that SVM and KNN and CNN had an average accuracy of roughly 90%, 85%, and 96%, respectively. [14] employed several MLR techniques. One of them is SMLR, which provides Pavia University with (OA=78.78%).

Bayesian Convolutional Neural Networks (BCNN) is a particular kind of deep neural network. In BNNs, probability distributions are used in place of one or more layer's network parameters. A particular set of weights is sampled from these distributions for network inference. Consequently, a BNN may be thought of as a distribution of networks. The many inference processes used by this distribution enable ensembling. The ensemble results' variance can be used as an extra measure of uncertainty, and the prediction is then the ensemble results' mean.

Bayesian techniques have been used to analyze neural networks in several publications. The fundamental difficulty is that computing the real posterior probability distribution is hard. To compute the posterior, many approximation techniques have been examined and proposed. In their various maximum a-posteriori (MAP) algorithms for neural networks, [15] took second order derivatives into account while estimating prior probabilities. Other attempts to increase approximation quality while keeping the computation manageable and appropriate to contemporary applications have been made [16]. However, out of those, two are the most effective. The first is to build the Bayesian CNN network using the approximate variational inference techniques Dropout and Gaussian Dropout. The second method uses Backprop [17] to construct the Bayesian CNN based on variational inference. It takes into account the weights' Gaussian probability distribution, which is determined by the mean and variance of two parameters.

A dual-channel neural network design for the HSI classification approach based on DenseNet was described in [18]. A 1D DenseNet is used to extract spectral information from the proposed architecture, while a 2D DenseNet is used to retrieve spatial features. They were able to get greater classification accuracy, although the suggested method requires more training time.

II. METHODOLOGY

A. Deep Bayesian Convolutional Neural Networks

The major problem with Bayesian CNN approaches is that the true posterior probability distribution cannot be computed. To compute the posterior, many approximation techniques have been examined and proposed.

Both strategies offer ways to approximatively measure uncertainty, as documented in the literature. A comparison of the two methods is shown in [19], and it is demonstrated that they both perform similarly on the MNIST (Modified National Institute of Standards and Technology) dataset which is a large database of handwritten digits [7].

Frequentist neural networks have a propensity to make forecasts that are too confident. In addition, they are vulnerable to the overfitting issue when not given enough training data. These constraints can be overcome by combining the concept of Bayesian learning with conventional neural networks. By integrating across the distribution of potential models and the prior probability, Bayesian models provide predictions. This makes them more resilient to overfitting by enabling an internal regularization. We used a variational inference-based Bayesian Convolutional Neural network (BCNN).

By assuming the posterior of the model, Bayesian neural networks train a model. Even for moderately big models, accurate inference of the model posterior is computationally challenging and insoluble. As a result, the model posterior is often estimated. Variational inference is an efficient and well- liked approximation technique.

The function $f(x) = y$ estimates the output y from the inputs X given the input set $X = x_1, x_2, \dots, x_n$ and a matching output set $y = y_1, y_2, \dots, y_n$. Using Bayesian learning, one may extract the model posterior $p(f|X, y)$ in a logical manner. The posterior can only be calculated using two components.

- First, a prior distribution $p(f)$ that reflects an assumption made in the past on the estimator functions.
- Second, a probability function $p(y|f, X)$ that quantifies the likelihood that the model f will correctly forecast the output y in light of the observations X.

The posterior is specifically produced by integrating over all possible estimator functions f that are parametric models with parameter set θ given an unknown set of data (x*, y*),

$$p(y*|x*, X, y) = \int p(y*|f)\, p(f|x*, X, y) df = \int p(y*|f)\, p(f|x*, \theta)\, p(\theta|X, y)\, df d\theta \quad (1)$$

Due to the intractability of the distribution $p(\theta|X, y)$, this integral is unsolvable. Therefore, the variational technique involves using a variational distribution q(θ) to approximate $p(\theta|X, y)$. The original intractable distribution should be as similar to the contender q(θ) as is practicable. The Kullback- Leiber (KL) divergence may be used to determine how comparable $p(\theta|X, y)$ and q(θ) are [8]. Maximizing the log evidence lower bound with regard to the parameter set θ is comparable to minimizing the aforementioned KL divergence:

$$KL_{VI} = \int q(\theta) p(F|X, \theta) log_p(y|F) dF d\theta - KL(q(\theta)||p(\theta)) \quad (2)$$

A variational function that closely resembles the posterior is produced by maximizing KLVI.

Equation 1 becomes simpler using to the approximation q(θ).

$$q(y^*|x^*) = \int p(y^*|f)\; p(f|x^*, \theta)\; q(\theta) df\; d\theta \quad (3)$$

The network samples the network parameters from q(θ) while doing inference.

For training BCNN via back-propagation, the posterior distribution on the neural network weights is learnt in Bayes using Backprop [18], [20]. An estimated distribution $q_\alpha(\theta)$ comparable to the genuine distribution p(θ) is defined since the true posterior is frequently intractable. By identifying the ideal parameter, the training entails reducing the KL divergence of $q_\alpha(\theta)$ and the true intractable posterior p(θ). This is done by using n chosen samples to approximate the integral from Eq.(4):

$$F(D, \alpha) \approx \sum_{i=1}^{n} \log q_\alpha\left(\theta^{(i)}|D\right) - \log p\left(\theta^{(i)}\right) - \log p\left(D|\theta^{(i)}\right) \quad (4)$$

where D represents the training dataset. θ (i) is a sample from the variational distribution $q\alpha(\theta|D)$, which we set as a Gaussian distribution with mean and standard deviation as parameters. The cost function in Eq. 4 consists of three terms. First, log $q\alpha(\theta(i)|D)$ is the variational posterior with mean μ and standard deviation σ,

$$\log q_\alpha\left(\theta^{(i)}|D\right) = \sum_i \log N(\theta_i|\mu, \sigma^2) \quad (5)$$

Second, log *p(θ(i))* denotes the log prior, which is a zero-mean Gaussian distribution

$$\log p(\theta^{(i)} = \sum_i \log N(\theta_i|0, \sigma_p^2) \quad (6)$$

Third, the likelihood log $p(D|\theta(i))$ is the network output.

To include Bayesian learning in CNNs and solve the intractable posterior distribution problem via variational inference, fully connected layers and convolutional layers with a probability distribution over the weights as filter weights must be created. In this scenario, samples from the relevant distribution would serve as the weights. The distributions are Gaussian, as was described in the part before, and the mean and variance of each weight distribution serve

as its defining characteristics. Shridhar et al. employed the Local Re-parameterization approach on the convolutional layers to adopt this concept [21]. To convert the global uncertainty to the local uncertainty, it is just a matter of rewriting and re-parameterizing the equations above. This technique samples the activation maps b rather than the weights directly. Using this technique, the activation maps b are sampled rather than the weights directly, resulting in more effective and quicker computing.

Consider the *w* variable for network weights. The variational posterior $q_w(w_{ijhw}|D) = N(\mu_{ijhw}, \alpha\mu_{ijhw}^2)$, where *i, j* inputs, and *h, w* filter height, width. The activation of the convolutional layer for the associated receptive field R_i has the following effects:

$$b_j = R_i * \mu_i + \epsilon_j \odot \sqrt{R_i^2 * (\alpha_i \odot \mu_i^2)} \quad (7)$$

where $\epsilon_j \sim \mathcal{N}(0,1), \odot$ is the element-wise multiplication, and $*$ is the convolution operator. As shown, this approach splits the convolutional procedure that occurs within a layer into two operations: The Adam optimizer first treats the output of *b* as a frequentist output and updates it. The mean of the posterior is regarded as being this single-point estimate. The distribution's variance is discovered in the second process. This formula guarantees a positive variance that is not zero. In order to support both processes, Shridhar et al. developed the Soft-plus activation function [9].

B. Data preparation

In order to apply a deep learning or Bayesian models, some primary steps were implemented: PCA, channel-wise shift and patches creating. One of the most used unsupervised dimensionality reduction and feature extraction approaches is principal component analysis (PCA). Principal components (PCs) produced by PCA are ordered by variance in descending order and are linearly independent. Most of the information is located in the first few PCs. PCA, however, struggles to deal with the complicated nonlinear properties of HSIs since it is a linear orthogonal transform approach [16]. The eigen value decomposition of the covariance matrix of the HSI bands serves as the foundation for the PCA mathematical principle [17].

Kernel PCA (KPCA) is one of the nonlinear expansions of PCA, which nonlinearly maps the original data to a high- dimensional feature space [18]. Nonlinear dimensionality reduction is discretely performed by using PCA in the high- dimensional feature space. In addition to deriving a number of related features from PCA, KPCA is free of some of the practical issues that other nonlinear PCA extensions experience, such as nonconvergence or convergence to local minima [19]. For feature extraction and picture denoising, KPCA and its expansions have shown to be effective tools.

The channel-wise shift method is used between the PCA step and the first convolution layer to improve feature extraction capabilities and classification accuracy by emphasizing more significant spectral bands and suppressing less valuable ones. It was put forward in light of PCA's ranking fundamentals and convolution's margin impact, [22].

The channel-wise shift strategy is shown in Figure 2, and it aims to relocate the spectral bands that are comparatively more significant to a more central location for the most adequate convolutions. Instead, marginalizing the comparatively less significant spectral regions will help with information retrieval and processing efficiency.

This technique can preserve informative channels in the middle of the efficient receptive fields while increasing the number of spatial feature extraction times. We can guarantee that by performing this procedure, the crucial spectral bands will remain in the center of all the channels, allowing for additional convolution operations.

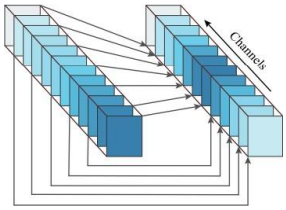

Figure 2. Channel-wise shift diagram. The more information in the spectral band, the deeper the color [23]

The hyperspectral image was divided into patches of size windowsize-by-windowsize pixels with DimReduction (80) channels after channel-wise shift was applied.

3-D convolution operations could be applied. The use of 3-D-kernels to extract spectral-spatial features is a sensible approach since HSI is a cubic data which could be thought as a 3-D tensor. The result of 3-D convolution includes spectral information.

Figure 3 demonstrates how the 2-D convolution method concentrates on creating hyperspectral data by just taking into account the spatial correlation of each channel in the provided image. When performing 3-D convolution operation, the correlation between various channels is also employed to create spectral-spatial feature maps, which enhance the ability of feature representation. So, the 3-D convolution can extract spectrum-spatial characteristics despite having a greater computational cost, whereas the 2-D convolution can extract spatial information but is unable to gather significant data in later spectral bands.

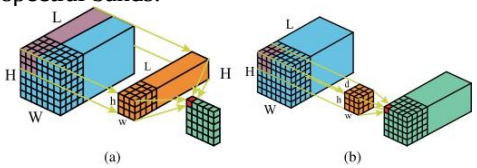

Figure 3. Models of 2-D and 3-D convolutions [22]

2-D and 3-D convolution was tested in Bayesian CNN model. 3D convolution required too much time to perform the third correlation operations. However, optimal results were obtained using Bayesian method. Whereas, in deep learning methods, the 2D convolutions are applied.

Deeper architecture is a potential method for extracting features from hyperspectral data since it can learn more abstract properties at higher levels. A deep 3-D-CNN is frequently computationally pricey, nevertheless. As a result, the PCA technique is frequently used to emphasize the key features of input data while simultaneously improving efficiency. The first-order statistics are always added using standard max or average pooling algorithms.

C. Machine learning methods

SVM is a popular linear classifier for HSI classification that is associated with kernel functions and optimization theory. Particularly in scenarios when there are more spectral bands and fewer training samples available, SVM outperforms traditional supervised classification techniques. SVM performs better in HSI than other methods of classical image categorization [23]. The core element of invention is the kernel trick. To transfer the input space to a high- dimensional feature space, SVM uses a kernel. As a result, we must select kernel functions based on the feature number, such as polynomial kernel functions and radial basis kernel functions. Another option is to employ a composite kernel structure that integrates spectral and spatial data.

Random forest is an approach for classifying data using several decision trees. Each tree in the random forest RF group of tree-based classifiers is trained using a bootstrapped set of training data. Each tree in the forest receives the classification-related data as an input. A "vote" for a certain class is the categorization that each tree provides. The categorization is made by the forest, which selects the class with the highest votes (over all the trees in the forest). A split in RF classification is found by looking through a random collection of variables at each node [23]. Two key aspects of RF are processing speed and a fair amount of precision. The final land cover map's accuracy can be impacted by the trees' independence or correlation, though.

D. Deep learning methods

The pretrained deep learning mode may be used by applying the dimension reduction using PCA, channel wise shift, and patches creation.

ResNet uses residual structure to address the "degradation" issue with deep neural networks. It employs several parameter layers to learn the representation of residuals between input and output, in contrast to VGGs networks, which use parameter layers to try to learn the mapping between input and output [16].

The ResNet network served as a model for the DenseNet network. All layers are connected by DenseNet using a dense connection technique. By allowing the feature map learnt by each layer to be communicated directly to all succeeding layers as input, this connection strategy makes it easier to train the network and increases the effectiveness of the features and gradient transmission [17].

Two areas are primarily where the **Inception-V3** network has been developed. The Inception Module is first optimized using the branch structure, and then the bigger two- dimensional convolution kernel is split into two one- dimensional convolution kernels. With less computing required, this asymmetric structure can handle more and richer spatial information [21].

Inception-V3 has been improved by **Xception**. The network suggests a unique Depthwise Separable Convolution that aligns them in columns, with space transformation and channel transformation serving as its main tenets. Xception is speedier and has less settings than Inception [16].

NasNet uses the two main functionalities are normal cell and cell for reduction. Normal cells specify the size of the feature map, whereas reduction cells return the feature map that has been shrunk by a factor of two in terms of height and breadth [21].

EfficientNet: Using a set of predetermined scaling coefficients, the EfficientNet scaling technique equally adjusts network width, depth, and resolution as opposed to the conventional approach, which scales these variables randomly [19].

III. EXPERIMENTAL RESULTS

A. Dataset Information

These are two scenes that the ROSIS sensor captured while flying above Pavia in northern Italy [15]. 103 spectral bands are present at Pavia University. Pavia University is 610*610 pixels, however some of the samples in the image are empty and must be eliminated before analysis since they lack information. The resolution in geometry is 1.3 meters.

#	Class	Samples
1	Asphalt	6631
2	Meadows	18649
3	Gravel	2099
4	Trees	3064
5	Painted metal sheets	1345
6	Bare soil	5029
7	Bitumen	13330
8	Self-Blocking Bricks	3682
9	Shadows	947

Table1. Ground truth classes for the Pavia University scene and their respective samples number

Nine classes of interest are considered, with respective samples number for each class in Table 1. Figure 1 shows Sample band of Pavia University dataset (left), Ground truth of Pavia University dataset.

B. Proposed method

The BCNN with previous classification methods were applied and implemented using the following steps which are illustrated in Figure 4 First, loading dataset from the University of Pavia and the ground truth.

Data preparation stage includes the PCA approach to extract features and reduce dimensions, then applying channel wise shift technique. Patch creating was applied for BCNN and transfer learning models. Building the classifiers models and following the establishment of specific parameters was done. Using 5-fold cross validation, the stage of training and testing was performed using the necessary functions in order to classify the HSI image to obtain its labels and classes. Using the differences between predicted and groundtruth labels, performance metrics were evaluated.

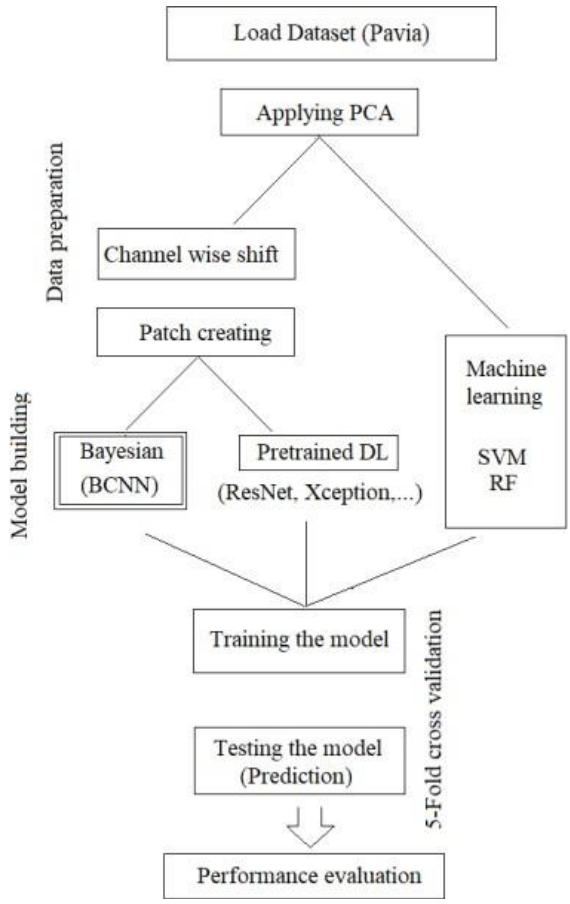

Figure 4. Flowchart of the proposed approach

C. Performance

By comparing the previously trained models with various metrics, the performance of the suggested method is measured. By evaluating how well the learning algorithms perform on the testing dataset, one may judge the overall quality of the algorithms [24]. Performance and productivity of a confusion matrix are affected by four variables. The true positives, true negatives, false positives, and false negatives are used to gauge these characteristics. The following formulae can be used to demonstrate how accurate the measurement was.

$$Sensitivity = \frac{TP}{TP + FN} \tag{8}$$

Values of type TP stand for true positive (correctly identified), type FP for false positive (mis-classified), type FN for false negative, and type TN for true negative [24]. Specificity is connected to the conditional probability of a true negative which has been given a secondary class. As such, it estimates the likelihood of a negative labeling. It is denoted by:

$$Specificity = \frac{TN}{TN + FP} \tag{9}$$

where TN is the number of true negatives, or negative cases that are indeed negative and are classified as negative, and FP is the number of false positives, or negative instances that are wrongly classified as positive cases. Accuracy represents the most common stat for measuring the model's ability to categorize. As such, accuracy was extremely important within the experimentation and was determined after every 20 iterations. It also relates to how many samples were properly identified. It was calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

Classification models are evaluated on a short data sample using a resampling approach called cross-validation.

The model determines how many groups should be created from every dataset using a single parameter (called k). Because of this, the model is usually referred to as k-fold cross-validation [25]. When choosing a particular value for k, this figure can be substituted for k within the model. In order words, k=10 equates to 10-fold cross-validation.

D. Bayesian CNN results

Applying the Bayesian approach, as previously reviewed, gives an accuracy about 99%. Table 2 shows that the accuracy of most classes are around 0.99, for the third class 0.98, for the ninth class 0.97. Noting little number of misclassified samples between the class "Gravel" and the class "Self-Blocking Bricks".

	Class Name	Precision	Recall	F1-score	Support
1	Asphalt	0.998	0.987	0.992	1194
2	Meadows	0.998	1.000	0.999	3357
3	Gravel	0.984	0.992	0.988	378
4	Trees	0.995	0.985	0.990	551
5	Painted metal sheets	1.000	0.996	0.998	242
6	Bare Soil	0.996	1.000	0.998	906
7	Bitumen	0.996	1.000	0.998	239
8	Self-Blocking Bricks	0.985	0.986	0.986	663
9	Shadows	0.977	1.000	0.988	170

Table.2 Bayesian results

Window size significantly affects on the classification accuracy, as shown in Table 3. Increasing the window size gives better performance and speeds up the training process.

	Window Size						
Model	**5 × 5**	**7 × 7**	**9 × 9**	**11 × 11**	**13 × 13**	**15× 15**	**50 × 50**
BCNN	77.4%	84.5%	93.4%	97.8%	98.3%	99.4%	99%
EfficientNet	68.5%	62.6%	61.8%	64.5%	65%	67.3%	80%
Inception	83.3%	90%	91.3%	90.9%	95.1%	91.8%	92%
NasNet	41.3%	42.7%	44.8%	41.6%	43.3%	41.0%	66%
ResNet	79.8%	80.3%	83.9%	83.4%	84.7%	84.3%	81%
Xception	81.2%	85.2%	89.9%	94%	96%	97%	92%
SVM	96 %						
RF	96 %						

Table.3 Window size effect

E. SVM results

Applying SVM classifier with RBF kernel gives an accuracy about 96%. In details, the performance for every class can be seen in Table 4. Noting that the third class “Gravel” has the lowest accuracy 0.87. However, the average accuracy is good. Misclassified samples size is relatively big, comparing with Bayesian.

	Class Name	Precision	Recall	F1-score	Support
1	Asphalt	0.96	0.97	0.97	1346
2	Meadows	0.97	0.99	0.98	3685
3	Gravel	0.87	0.86	0.86	420
4	Trees	0.98	0.98	0.98	604
5	Painted metal sheets	1.00	1.00	1.00	248
6	Bare Soil	0.96	0.91	0.93	1034
7	Bitumen	0.93	0.88	0.91	255
8	Self-Blocking Bricks	0.91	0.92	0.92	762
9	Shadows	1.00	1.00	1.00	201

Table.4 SVM results

F. Xception results

Pretrained models take more time comparing with SVM and Bayesian CNN. However, Xception network gives the best average accuracy of about 97%.

	Class Name	Precision	Recall	F1-score	Support
1	Asphalt	0.870	1.000	0.931	1213
2	Meadows	0.996	1.996	0.996	3461
3	Gravel	0.846	1.000	0.917	408
4	Trees	0.980	0.923	0.950	590
5	Painted metal sheets	1.000	1.000	1.000	239
6	Bare Soil	1.000	1.000	1.000	947
7	Bitumen	0.905	1.000	0.950	244
8	Self-Blocking Bricks	1.000	0.774	0.873	721
9	Shadows	1.000	0.643	0.783	197

Table.5 Xception results

IV. CONCLUSION

In this study, Bayesian CNN was applied for the classification of Pavia hyperspectral image dataset. Two other approaches were applied and viewed, machine learning SVM RF methods, pertained deep learning.

PCA was applied in all methods for dimensionality reduction and feature extraction. 5-cross validation was also applied as a good technique to apply the training and testing process covering randomly the whole dataset. Knowing that the terminal spectral bands in Pavia HSI are more informative, the channel-wise shift technique was applied in Bayesian and deep learning methods. Splitting the hyperspectral image into patches of size m by n pixels was done using patches creating.

Bayesian CNN gives the best results of 99% accuracy. However, it was not easy to implement this method. Machine learning methods such as SVMs, RFs have been widely used in the hyperspectral analysis community. Using PCA for dimension reduction and features extraction had enhanced the performance of classifier. SVM with RBF kernel gives good result (96% accuracy). Pretrained deep learning networks take long time for classification. However, Xception model gives an accuracy of about 97%.

Applying more advanced techniques require powerful computational capabilities. However, more tests could be done using other HSI dataset (Indian Pines, Salinas).

REFERENCES

[1] Deng. Y. J, Li. H. C, Pan. L, et al, Modified tensor locality preserving projection for dimensionality reduction of hyperspectral images, IEEE Geosciences and Remote Sensing Letters, vol. 15, no. 2, pp. 277-281, 2018.

[2] Davari. A, Ozkan. H. C, Maier. A, and Riess.C, Fast and efficient limited data hyperspectral remote sensing image classification via gmmbased synthetic samples, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 12, no. 7, pp. 2107–2120, 2019.

[3] Zheng. J , Feng. Y, Bai. C, and Zhang. J, Hyperspectral image classification using mixed convolutions and covariance pooling, IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 1, pp. 522-534, 2020.

[4] Van Der Meer. F, The effectiveness of spectral similarity measures for the analysis of hyperspectral imagery, International Journal of Applied Earth Observation and Geoinformation, vol. 8, no. 1, pp. 3-17, 2006.

[5] He. X and Chen. Y, Transferring CNN ensemble for hyperspectral image classification, IEEE Geosciences and Remote Sensing Letters, vol. 18, no. 5, pp. 876–880, 2020.

[6] Audebert. N, Saux. B. L, and Lefevre. S, Deep learning for classification ` of hyperspectral data: A comparative review, IEEE Geoscience and Remote Sensing Magazine, vol. 7, no. 2, pp. 159- 173, 2019

[7] Cao. X, Yao. J, Xu. Z, and Meng. D, "Hyperspectral image classification with convolutional neural network and active learning, IEEE Transactions on Geosciences and Remote Sensing, vol. 58, no. 7, pp. 4604–4616, 2020.

[8] Joshaghani. M, Davari. A, Hatamian. F. N, Maier. A, Riess. C, Bayesian convolutional neural networks for limited data hyperspectral remote sensing image classification, arXiv preprint arXiv: 2205.09250, 2022 May 19.

[9] Huang. X and Zhang. L. A comparative study of spatial approaches for urban mapping using hyperspectral rosis images over Pavia city, northern Italy. International Journal of Remote Sensing, vol. 30, no. 12, pp. 3205–3221, 2009.

[10] Ye Z, He M. PCA and windowed wavelet transform for hyperspectral decision fusion classification. Journal of Image & Graphics, vol. 20, no. 1, pp. 0132-0130, 2015.

[11] Sigirci, I.O. and Bilgin, G., Spectral-Spatial Classification of Hyperspectral Images Using BERT-Based Methods With HyperSLIC Segment Embeddings, IEEE Access, vol.10, pp.79152- 79164, 2022.

[12] Mei S, Ji J, Hou J, et al. Learning Sensor-Specific Spatial-Spectral Features of Hyperspectral Images via Convolutional Neural Networks[J]. IEEE Transactions on Geoscience & Remote Sensing, vol. 55, no. 8, pp.4520-4533, 2017.

[13] Noi. P. T and Kappas. M, Comparison of rrandom forest, k-Nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery, Sensors, vol. 18, no. 1, p. 18, 2018.

[14] Tarabalka. Y, Classification of hyperspectral data using spectral- spatial approaches, Ph. D dissertation, 2010.
[15] Jiang, J.; Ma, J.; Chen, C.; Wang, Z.; Cai, Z.; Wang, L. Super PCA: A superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery. IEEE Trans. Geosci. Remote Sens, vol. 56, no. 8, pp. 4581–4593, 2018.
[16] Zhuo. S, Guo. X. S and Wan.J , et al., Fast classification algorithm for polynomial kernel support vector machines, Jisuanji Gongcheng/ Computer Engineering, vol. 33, no. 6, pp. 27-29, (2007).
[17] Lu. D and Weng. Q, A survey of image classification methods and techniques for improving classification performance, Int. Jour. Remote Sens, vol. 28, no. 5, pp. 823–870, 2007
[18] Zhao. C, Gao. X, Wang. Y, and Li. J, "Efficient multiple-feature learning-based hyperspectral image classification with limited training samples," IEEE Trans. Geosci. Remote Sens., vol. 54, no. 7, pp. 4052–4062, July 2016.
[19] Tuia. D and Camps-Valls. G, "Semisupervised remote sensing image classification with cluster kernels," IEEE Geosci. Remote Sens. Lett., vol. 6, no. 2, pp. 224–228, 2005.
[20] Castillo. C, Chollett. I, and Klein. E, Enhanced duckweed detection using bootstrapped SVM classification on medium resolution RGB MODIS imagery, Int. J. Remote Sens., vol. 29, no. 19, pp. 5595–5604, 2008.
[21] Krishnapuram. B, Carin. L, Figueiredo. M, and Hartemink. A, Sparse multinomial logistic regression: Fast algorithms and generalization bounds, IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 6, pp. 957–968, 2005.
[22] Chen. Y, Nasrabadi. N. M, and Tran. T. D, Hyperspectral image classification using dictionary-based sparse representation, IEEE Transactions on Geoscience and Remote Sensing, vol. 49, no. 10, pp. 3973–3985, Oct. 2011
[23] Tao. C, Pan. H, Li. Y, and Zou. Z, Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification, IEEE Geoscience and Remote Sensing Letters, vol. 12, no. 12, pp. 2438–2442, 2015.
[24] Chen. Y, Zhao. X, and Jia. X, "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. 2381–2292, 2015.
[25] Huang. G. B, Zhou. H, Ding. X, and Zhang. R, Extreme learning machine for regression and multiclass classification, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 42, no. 2, pp. 513–529, 2012.

★★★