A Comparison of Machine Learning and Deep Learning in Hyperspectral Image Classification

Fady Mohamed Sadek[1], **Mahmud Iwan Solihin**[1], **Fahri Heltha**[1,2], **Lim Wei Hong**[1] **and M Rizon**[1]

[1] Faculty of Engineering, UCSI University, Kuala Lumpur 56000, Malaysia
[2] USK, Banda Aceh 23333, Indonesia
mahmudis@ucsiuniversity.edu.my

Abstract. In recent years, hyperspectral remote sensing has become popular in various applications. This technology can capture hyperspectral images with a large terrestrial data. In this paper, the feasibility of applying various machine learning and deep learning techniques to perform classification on hyperspectral images are investigated and compared. Particularly, a total of three popular machine learning classifiers namely supports vector machine (SVM), K-nearest neighbors (KNN) and artificial neural networks (ANN) are used for hyperspectral imagine classification, followed by another two deep architectures in convolutional neural networks (CNN). Three benchmarking datasets of hyperspectral images are used to evaluate the classification performances of suggested machine learning and deep learning techniques, namely: Indian Pines (IP) dataset, Salinas dataset, and Pavia University (PU) dataset. Extensive simulation studies reveal the excellent performance of 3D CNN deep learning in solving larger datasets with better classification accuracy despite the longer training time is required. However, it is not really the case when the dataset is not large enough. This is because deep learning is data-hungry architecture. Furthermore, the 3D CNN deep learning models employed in this study have shown more advantageous as compared to other machine learning models for having simplified pre-processing stages such as feature extraction in solving the classification problems of hyperspectral images.

Keywords: hyperspectral image classification; convolutional neural network (CNN); support vector machine (SVM); k-nearest neighbors (k-NN); artificial neural networks (ANN); fully convolutional network (FCN).

1 Introduction

Hyperspectral imaging (HSI) is a spectral detection technique in which an object is observed using various well defined, broad-spectral optical bands. Originally deployed on remote sensation satellite and airborne platforms, HSI has been used in various applications over the past two decades such as for agricultural and water resources management, military protection, art conservation and archeological, medical analyzes of crime scene information, forensics documentation, food quality control, and mini-images [1]. HSI encompasses a wide variety of imaging technologies, such as medical hyperspectral imagery, atmospheric sounds, and hyperspectral imagery at close range. However, it has also been developed for applications in mining and geology, deemed a reliable source of data for a variety of possible applications, such as mineralogy, climate tracking, precision farming, protection and privacy requests, chemical analysis, astronomy and biological sciences, as well as for products quality characterizing in food industry [2].

In the past few decades, hyperspectral image data classification algorithms using classical machine learning such ANN and SVM have been discussed. On the other hand, deep learning has been emerged as a new approach of image classification due to its automated capability in features extraction.

2 Hyperspectral Image

The scene rays are capturing in a layer by layer image and various wavelengths, as shown in a hyperspectral data cube in Fig.1. For each wavelength, the x-y plane shows the hyperspectral spatial data cube and the spectral contents of the z plane. Every hyperspectral band of images has a dimension that describes a digital number for every pixel that correlates to the radiance value gets by the sensor, where every band fits in a given wavelength. It is commonly depicted as the HSI data cube (3D hypercube) of $n_1 * n_2 * n_b$, where $n = n_1 * n_2$ represents the number of pixels and n_b represents the number of bands.

In a spectral area generated by the number of bands, each pixel is represented as a one-dimensional vector. The similar types of materials are then grouped by spectral, a mutually equivalent clustering algorithms. The famous clustering algorithms used for analysis of hyperspectral images are fuzzy c-clustering, K-means clustering, and unmixed spectral clustering approaches. Due to correlation is strong in a spectral space, the data is reflected in a lower dimensional space, which is less than spectral band count. Data reduction in dimensionality is achieved using other techniques [3] such as principal component analysis (PCA) [4], or independent component analysis (ICA) [5].

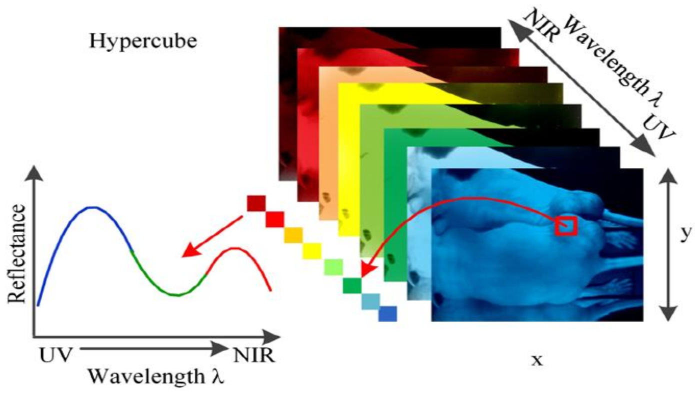


Fig. 1. Spectral Signature in Hyperspectral Data Cube.

3 Convolutional Neural Networks

Hyperspectral imaging contains different image lines. Many of classification methods are based on 2-D CNN. Against the context, the performance of HSI classification is highly dependent upon both spatial and spectral details. The 3-D CNN has been used by very few approaches due to the added computing complexity as shown in Fig. 2. This paper suggests a CNN (HybridSN) dual spectral for classification with HSI. The HybridSN is a typically spectral–spatial 3-D CNN accompanied by 2-D CNN [6].

Considering spectral–spatial data cube of the hyperspectral is given as the following set:

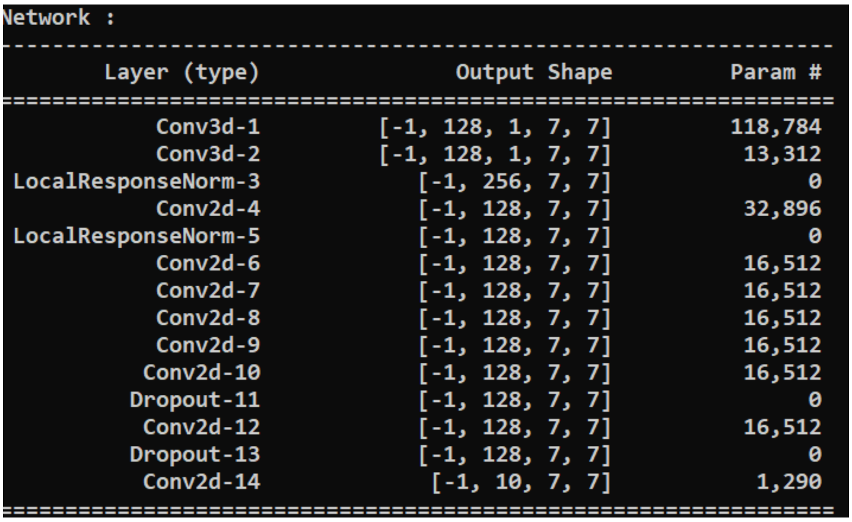
$$I \in R^{M\,x\,N\,x\,D} \tag{1}$$

where *I* is the initial signal, *M* is the width, *N* is the height, and *D* is the spectral bands/depth.

Every HSI pixel in *I* contains *D* measured spectral and forms a one-hot label vector expressed as:

$$Y = (y1, y2, \dots, yC) \in R^{1\times1\times C} \tag{2}$$

where *c* stands for types of ground cover [6].



```
Network :
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv3d-1        [-1, 128, 1, 7, 7]         118,784
            Conv3d-2        [-1, 128, 1, 7, 7]          13,312
 LocalResponseNorm-3           [-1, 256, 7, 7]               0
            Conv2d-4           [-1, 128, 7, 7]          32,896
 LocalResponseNorm-5           [-1, 128, 7, 7]               0
            Conv2d-6           [-1, 128, 7, 7]          16,512
            Conv2d-7           [-1, 128, 7, 7]          16,512
            Conv2d-8           [-1, 128, 7, 7]          16,512
            Conv2d-9           [-1, 128, 7, 7]          16,512
           Conv2d-10           [-1, 128, 7, 7]          16,512
          Dropout-11           [-1, 128, 7, 7]               0
           Conv2d-12           [-1, 128, 7, 7]          16,512
          Dropout-13           [-1, 128, 7, 7]               0
           Conv2d-14            [-1, 10, 7, 7]           1,290
================================================================
```

Fig. 2. Architecture of 3D CNN layer.

4 Support Vector Machines

Suppose that two class data sets are in the R^2 region, as shown in Fig. 3. It is obviously shown that the data is not linearly separable in the space. Classifying this data into 2D and rendering it in smaller dimensions becomes a complex task, a nonlinear classifier is required. Therefore, the data is mapped to space R^3. Initial space, in which the data is stored, is referred as the function space called the input space, and the higher dimensional space, in which the data is interpreted.

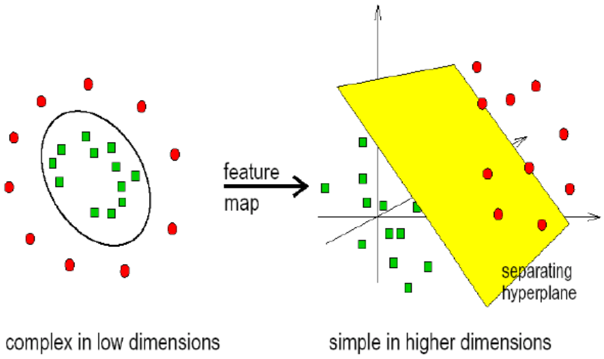


Fig. 3. Non-Linear data 2D is mapped to 3D.

SVMs are designed to classify a binary class data. However, multiclass data has to be dealt with in the HSI classification. We need to expand SVM effectively to manage multiclass data. Two available methods are currently available: to construct several binary classifiers and combine the results, and to frame an entire data optimization problem. The latter is computationally complex and costly [7].

5 K-Nearest Neighbors

The most widely used machine learning method in classification is K-neighbor grouping, where the best selected value depends heavily on the data. In general, it suppresses the effects of noise, but less distinguishes of the classification borders.

When the data is not distributed equally, the classification of radius-based neighbors may become a better alternative. The user specifies a fixed radius for using fewer closest neighbors to do classification in the sparser neighborhoods. This approach is less successful in high-dimensional parameter spaces due to the so-called 'dimensionality curse' [8].

The nearest basic neighbor classification uses uniform weights: i.e., a clear majority vote of the closest neighbors determines the value assigned to a question point. For certain cases, weighting of the neighbors is ideal, so that closer neighbors added to their fitness. This can be achieved by using the keyword weights. The default value, weights

= 'uniform,' gives each neighbor uniform weights. Weights = 'distance' assigns weight equal to the reverse of the question point size. Alternatively, weight can be determined by means of a user-defined distance function [8].

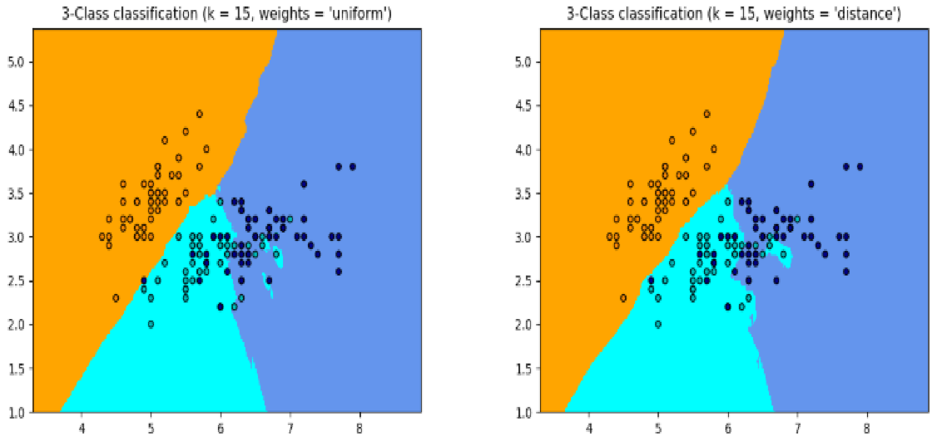


Fig. 4. Illustration of K-NN classification.

6 Artificial Neural Networks

Multi-layer perceptron (MLP) correspond as feedforward of artificial neural network (ANN). MLP is a supervised learning algorithm that learns a function:

$$F(.): R^m \;\; --> \;\; R^o$$

where *m* is the number of input dimensions and *o* the number of output dimensions c.

ANN has been used successfully in many applications of different field across discipline from environmental engineering [12-13], signal processing [14], robotics [15] and particularly image classification [16]. Fig. 5 shows the MLP hidden layer with several neurons.

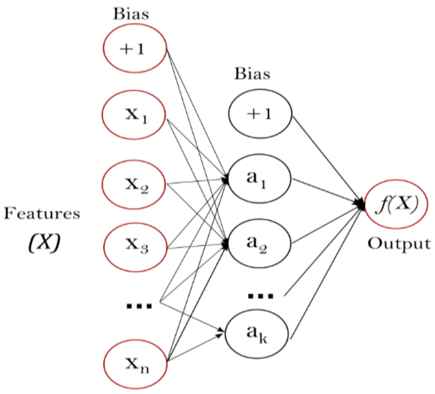


Fig. 5. One hidden layer MLP.

7 Hyperspectral Image Data Collection

The datasets used in this study has been downloaded from online platform. They have been taken with different spectrometer types such as ROSIS-3 and AVARIS sensors. The study is based on three different types of datasets, Pavia University (PU), Indian Pines (IP) and Salinas. The three different datasets will be used to evaluate the effectiveness of image classifier using the classification methods of machine learning and deep learning. They have respective ground truth labels to classify.

7.1 Pavia University Dataset

The PU datasets consist of 9 ground truth classes and each class has respective sample number of average of 4753 samples shown in Table 1. The object in the same class shows great differences in spatial structure. Fig.6 show the groundtruth of the data sample of PU.

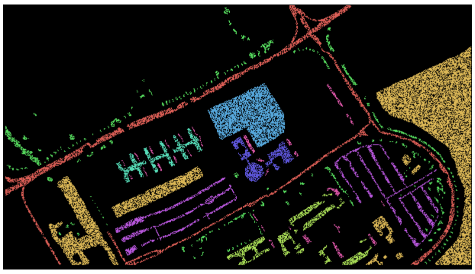


Fig. 6. Ground truth of PU

Table 1. Label of classes (ground truth) for PU dataset

#	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	1330
8	Self-Blocking Bricks	3682
9	Shadows	947
Total Samples	---------------	42776

7.2 Indian Pines Dataset

The IP dataset consists of 16 ground truth unbalanced class and has an average number of samples in each class of 640 samples as shown in Table 2. The average and minimum groups contain 2455 pixels and just 20 pixels, respectively. Fig.7 show the ground truth of the data sample of IP.

Table 2. Label of classes (ground truth) for IP dataset

#	Class	Samples
1	Alfalfa	46
2	Corn no till	1428
3	Corn mantilla	830
4	Corn	237
5	Grass pasture	483
6	Grass Rees	730
7	Grass pasture mowed	28
8	Hay windrowed	478
9	Oats	20
10	Soybean no till	972
11	Soybean mint ill	2455
12	Soybean-clean	593
13	Wheat	205
14	Woods	1265
15	Buildings Grass Trees Drives	386
16	Stone-Steel-Towers	93
Total samples	---------------	10249

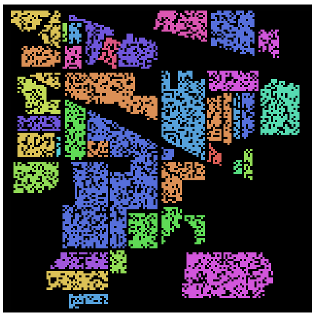


Fig. 7. Ground truth for IP

7.3 Salinas Dataset

The data set of Salinas has characteristics of two adjacent spatial classes and a bit of difference in spectral. These are a challenge for the methods of HSI classification. The datasets consist of 16 different classes with total of 54129 samples as shown in Table 3. Fig.8 shows the ground truth of the data sample of Salinas.

Table 3. Label of classes (ground truth) for Salinas dataset

#	Class	Samples
1	Broccoli green weeds_1	2009
2	Broccoli green weeds_2	3726
3	Fallow	1976
4	Fallow rough plow	1394
5	Fallow smooth	2678
6	Stubble	3959
7	Celery	3579
8	Grapes untrained	11271
9	Soil vineyard develop	6203
10	Corn senesced green weeds	3278
11	Lettuce romaine 4wk	1068
12	Lettuce romaine 5wk	1927
13	Lettuce romaine 6wk	916
14	Lettuce romaine 7wk	1070
15	Vineyard untrained	7268
16	Vineyard vertical trellis	1807
Total samples	------------------	54129

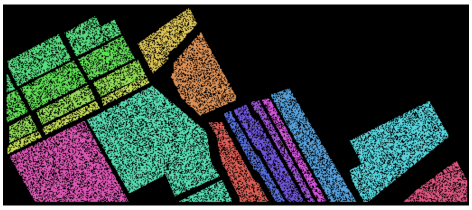


Fig. 8. Ground truth for Salinas.

8 Principal Component Analysis

In this study, the principal component analysis (PCA) is applied on several interrelated variables to minimize the dimensionality of spectral dataset. We represent the PCA reduced data cube by $X \in R^{M \times N \times B}$, where X is the modified input after PCA, M is the width, N is the height, and B is the number of spectral bands after PCA [10].

The steps involved in the PCA is to calculate the mean of all the dimension of dataset, except the labels, then scale the data as shown in equation 3.

$$Z = \frac{X - \mu}{\sigma} \tag{3}$$

Where Z is the scaled value, X is the initial, and μ and σ are mean and standard deviation, respectively.

To compute the covariance of two variable the equation 4 has to be used to find the covariance matrix.

$$Cov(X, Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - x)(Y_i - y) \tag{4}$$

Moving on by computing the eigenvectors and corresponding eigenvalue where the eigenvalue is the factor by which eigenvector is scaled. Last, chose the number of dimensions corresponding new datasets from the eigenvector with the largest eigenvalue [11]. The PCA algorithm was applied on the bands of the IP datasets, to show the difference of the bands before and after applying the PCA. The dimension of IP datasets was reduced to 8 dimension as in Fig.9 and Fig.10 shows the difference.

After visualizing the band some of the data in IP is missing which will lead to low accuracy, because hyperspectral data are covered by sensor with high spectral resolution which cannot be well described by the second order characteristics. Hence, the PCA is not effective tool for HSI classification since it deals only with second-order statistics [11].

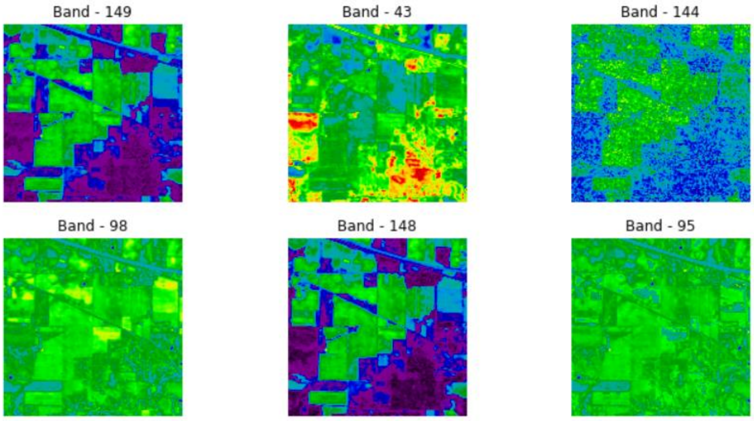


Fig. 9. Visualization of sample bands of IP.

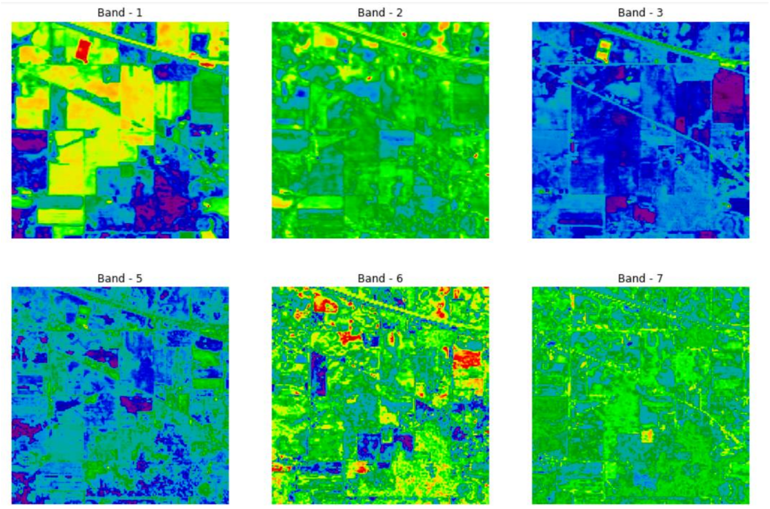


Fig. 10. Visualization of sample bands of IP after applying PCA.

9 Classification Results using Machine Learning

In this study we use three types of machine learning model, SVM, K-NN and ANN, and the three of them give different results in terms of accuracy. The three models have been tested twice where in the beginning, the training data was set to 30% and lastly was set to 90% to explain the difference of the accuracy when we increase the data training samples. Thus, the accuracy was evaluated based on respective 70% and 10% dataset. The data tested was IP, PU and Salinas.

Table 4. Classification accuracy for machine learning models

		Accuracy in testing		
Model	**Training samples**	***PU***	***IP***	***Salinas***
SVM	30%	83.90 %	66.718%	87.329%
	90%	90.603%	92.858%	90.098%
K-NN	30%	78.727%	72.864%	86.379%
	90%	79.149%	77.659%	86.200%
ANN	30%	88.08%	70.60%	85.326%
	90%	92.496%	92.585%	92.444%

Table 4 shows the performance testing accuracy among the proposed models with respective training data. The models trained with sample of 90% has shown better prediction comparing to 30%, especially for Indian Pines datasets due to it has 16 classes and lack of samples number those generate a very high prediction error when trained using 30% samples. By comparing IP and Salinas, we will notice that the performance of Salinas is higher, even they have the same number of classes. This difference performance is due to Salinas has 54129 samples and IP has 10229. Pavia university has good performance in both SVM and ANN model, except for K-NN, even we increased the number of samples it has almost gave the same classification result. It may be due to complexity of the data.

10 Classification Results using Deep Learning

The 3-D CNN deep learning model has been used throughout this study since the project is dealing with image processing datasets. This model has been trained first by using 30% of the datasets and then increased it to 90% of the datasets.

Table 5. Classification accuracy for deep learning models

		Accuracy in testing		
Model	**Training Samples**	***PU***	***IP***	***Salinas***
3D CNN	30%	83.90 %	66.718%	87.329%
	90%	94.60%	92.585%	92.296%
3D FCN	30%	83.629%	44.293%	88.116%
	90%	95.185%	64.585%	89.835%

Table 5 shows the performance testing accuracy for the proposed deep learning models with respective training data. PU and Salinas datasets have good performance comparing to Indian Pines. Using 30% of training sample in the Indian pines datasets, the prediction result shown very low comparing to 90% training samples. This is understandable as normally deep learning requires large dataset, i.e. data-hungry model.

11 Overall Results

The comparisons of the prediction results of both machine learning and deep learning for all discussed models are outlined in here. Table 6 shows the models' performance used in this study together with the training time taken for each executed model. Both machine learning and deep learning are good practical ways in solving image processing system problems. However, each datasets have different performance results in each model. In PU dataset, the model has shown the best performance result using 3D FCN with performance accuracy of 95%, while in other datasets it has not achieved the similar high-performance results. This is perhaps due to a smaller number of classes for PU dataset. In addition, it has taken a longer time of 1941 seconds for training process.

Table 6. Model performances and computation time

		Accuracy in testing					
		PU		***IP***		***Salinas***	
Model	**Training Samples**	Accuracy	Training Time (s)	Accuracy	Training Time (s)	Accuracy	Training Time (s)
SVM	30%	83.90 %	25	66.718%	31	87.329%	36
	90%	90.603%	29	92.858%	35	90.098%	48
K-NN	30%	78.727%	18	72.864%	24	86.379%	27
	90%	79.149%	21	77.659%	28	86.200%	32
ANN	30%	88.08%	240	70.60%	283	85.326%	311
	90%	92.496%	308	92.585%	347	90.444%	384
3D CNN	30%	83.90 %	901	66.718%	974	87.329%	1054
	90%	94.60%	1147	92.585%	1454	92.296%	1753
3D FCN	30%	83.629%	1098	44.293%	1274	88.116%	1346
	90%	95.185%	1941	64.585%	2107	89.835%	2584

For IP dataset, deep learning has not shown a good result as machine learning. The best performance in IP datasets found for SVM model and has taken 35 seconds in the training process.

Next, Salinas is a large dataset compared to others, and it has 92% accuracy of performance using 3-D CNN model. It is the highest one, but it has took 1753 seconds time for training process.

We conclude that the 3-D CNN deep learning method generally has higher accuracy comparing to machine learning because of the several convolutional layers that consist in both architecture 3-D CNN and 3-D FCN. Putting the scope on deep learning we will notice that this network has many non-linear components such as local response normalization (LRN), max pooling, linear rectified unit (ReLU) making this model a practical way for pixel classification such for HSI classification as it learns 5 x 5 pixel centered in individual pixel vector and applied for the entire test picture. That explains why 3-D CNN and 3-D FCN have longer training time comparing to the other models in machine learning.

12 Conclusions

This study aims to compare between machine learning and deep learning models applied for hyperspectral image classification datasets. The datasets used in this study are India Pines, Pavia University and Salinas. Several procedures have been taken starting with simple preparation of datasets, check their input types per channel and identify the kernel size for deep learning in each dataset.

It is found in the study that applying PCA is not a good option, since it reduces the performance of the prediction, therefore we use the original datasets and apply to the machine learning models namely SVM, K-NN and ANN. This process has been taken twice where we set the training sample to 30% and then we increase it to 90% to check the performance difference for the smaller and bigger training data. The same was applied to the deep learning models with 3D CNN and 3D FCN architecture.

Let's focus on the results of using 90% ratio in training datasets. In general, the classification accuracy of hyperspectral image classification using deep learning (3D CNN and 3D FCN) is better than that of machine learnings (SVM, KNN, ANN) for larger datasets only (Salinas and PU datasets). For smaller dataset, i.e. IP, machine learning accuracy is better than deep learning despite the training time is longer. This confirms the fact that deep learning architecture is data-hungry model as there are huge number of parameters to adapt.

References

1. P. S. Thenkabail, M. K. Gumma, P. Teluguntla, and A. Irshad, PE & RS, vol. 80, no. 8. (2014).
2. G. ElMasry, M. Kamruzzaman, D. W. Sun, and P. Allen, "Principles and Applications of Hyperspectral Imaging in Quality Evaluation of Agro-Food Products: A Review," Crit. Rev. Food Sci. Nutr., vol. 52, no. 11, pp. 999–1023 (2012).
3. H. Huang, G. Shi, H. He, Y. Duan, and F. Luo, "Dimensionality Reduction of Hyperspectral Imagery Based on Spatial-Spectral Manifold Learning," IEEE Trans. Cybern., vol. XX, no. X, pp. 1–13 (2019).
4. Q. Du and J. E. Fowler, "Low-complexity principal component analysis for hyperspectral image compression," Int. J. High Perform. Comput. Appl., vol. 22, no. 4, pp. 438–448 (2008).
5. J. Wang and C. I. Chang, "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis," IEEE Trans. Geosci. Remote Sens., vol. 44, no. 6, pp. 1586–1600 (2006).
6. S. K. Roy, S. Member, G. Krishna, S. R. Dubey, and B. B. Chaudhuri, "HybridSN : Exploring 3-D – 2-D CNN Feature Hierarchy for Hyperspectral Image Classification," vol. 17, no. 2, pp. 277–281 (2020).
7. C. Chang and C. Lin, "LIBSVM : A Library for Support Vector Machines," pp. 1–39, 2019.
8. R. K. Nowicki, "Rough Nearest Neighbour Classifier," Stud. Comput. Intell., vol. 802, pp. 133–159 (2019).
9. Lokman, G., Çelik, H., & Topuz, V. Hyperspectral Image Classification Based on Multilayer Perceptron Trained with Eigenvalue Decay. Canadian Journal of Remote Sensing, 1–19 (2020).
10. Roy, S. K., Member, S., Krishna, G., Dubey, S. R., & Chaudhuri, B. B. HybridSN : Exploring 3-D – 2-D CNN Feature Hierarchy for Hyperspectral Image Classification. 17(2), 277–281 (2020).
11. Wall, M. E., Rechtsteiner, A., & Rocha, L. M. Singular Value Decomposition and Principal Component Analysis BT - A Practical Approach to Microarray Data Analysis (D. P. Berrar, W. Dubitzky, & M. Granzow (eds.); pp. 91–109, Springer US (2003).
12. G. Hayder, M. I. Solihin, and H. M. Mustafa, "Modelling of River Flow Using Particle Swarm Optimized Cascade-Forward Neural Networks: A Case Study of Kelantan River in Malaysia," Appl. Sci., vol. 10, no. 23, p. 8670, (2020).

13. G. Hayder, M. I. Solihin, and K. F. Bin Kushiar, “A Performance Comparison of Various Artificial Intelligence Approaches for Estimation of Sediment of River Systems,” J. Ecol. Eng., vol. 22, no. 7, pp. 20–27, (2021).
14. W. Astuti, S. Tan, M. I. Solihin, R. S. Vincent, and B. Michael, “Automatic Voice-Based Recognition For Automotive Headlights Beam Control,” Int. J. Automot. Mech. Eng., vol. 18, no. 1, pp. 8454 – 8463–8454 – 8463, (2021).
15. L. William, A. Winda, D. Satrio, T. Sofyan, and M. I. Solihin, “Automotive start-stop engine based on face recognition system,” in E3S Web of Conferences, (2019).
16. Z. H. Ang, C. K. Ang, W. H. Lim, L. J. Yu, and M. I. Solihin, “Development of an artificial intelligent approach in adapting the characteristic of polynomial trajectory planning for robot manipulator,” Int. J. Mech. Eng. Robot. Res., vol. 9, no. 3, pp. 408–414 (2020).