Fine Tuning the Pre-trained Convolutional Neural Network Models for Hyperspectral Image Classification Using Transfer Learning

Chapter · April 2023

DOI: 10.1007/978-981-19-7892-0_21

CITATIONS
0

READS
79

2 authors:

Manoj Kumar Singh

8 PUBLICATIONS **85** CITATIONS

SEE PROFILE

Brajesh Kumar

M.J.P. Rohilkhand University

37 PUBLICATIONS **456** CITATIONS

SEE PROFILE

Fine Tuning the Pre-Trained Convolutional Neural Network Models for Hyperspectral Image Classification Using Transfer Learning

Manoj Kumar Singh[1[0000-0003-3119-1244]] and Brajesh Kumar[2[0000-0001-8100-7287]]

Department of Computer Science & Information Technology[1,2]
MJP Rohilkhand University, Bareilly, India

Atal Center for Artificial Intelligence[2]
MJP Rohilkhand University, Bareilly, India
manojaswal1982@gmail.com

Abstract. Convolutional neural networks (CNNs) are an effective technique for hyperspectral image classification. Deep learning models require a large number of labeled and diverse samples to properly train a CNN model. But a training set is often not large enough. Transfer learning can help to overcome the need for training sets. In this paper, six pre-trained CNN models: EfficientNetB0, EfficientNetB7, ResNet50, VGG19, DenseNet121 and DenseNet201 are fine-tuned for hyperspectral image classification. The experiments are carried out on two benchmark images Houston and Kennedy Space Center (KSC). The results show that hyperspectral images can be classified with good accuracy by fine-tuned pre-trained CNN models. As compared to training a model from scratch, fine-tuning takes a small number of epochs. Thus, alleviating the requirement for high-end computing resources. Among the tested models, VGG19 achieves the best accuracy of 95.77% for the KSC image, while EfficientNetB0 performs better than others with 90.79% accuracy for the Houston image.

Keywords: Hyperspectral image classification, remote sensing, transfer learning, pre-trained networks, convolution neural networks.

1 Introduction

In recent years, hyperspectral image (HSI) [1] processing has become a rapidly evolving field in remote sensing and other applications. Hyperspectral imaging is concerned with the analysis and interpretation of spectra obtained from a specific scene by an airborne or satellite sensor at a short, medium, or long distance. With hundreds of spectral channels and spectral resolution of the order of 10 nanometers, this system can cover the wavelength range from 0.4 to 2.5 µm. As all materials have some chemical properties, hyperspectral signature is helpful to recognize their individual absorption properties [2]. It is therefore useful to identify fine changes in the reflectance of vegetation, soil, water, and minerals, etc. For more precise and detailed

information extraction, hyperspectral images provide adequate spectral information to identify and discriminate spectrally similar materials.

HSI classification is a technique for constructing thematic maps from remotely sensed hyperspectral images. A thematic map depicts the ground surface materials e.g. plants, soil, roof, road, water, concrete, etc. with distinguishable themes. A wide range of effective HSI classification schemes based on spectral and spatial information is available [3][4]. Deep learning-based classification models [5][6] have recently been brought to the hyperspectral community, and they appear to hold a lot of promise in the field of remote sensing classification. Convolutional neural networks (CNNs) have been proved to be an effective technique for hyperspectral image classification. A hybrid hyperspectral image classification model [7] was introduced in which the few initial convolutional layers extract location invariant middle-level features, followed by recurrent layers that extract spectral-contextual information. Ding et al. [8] cropped patches from 2D input images to train a 2D-CNN architecture that learns the data-adaptive kernels on its own. Chen et al. [9] integrated the Gabor filtering approach with 2D-CNN to tackle the overfitting problem caused by limited training data. The Gabor filtering substantially reduces the overfitting problem by extracting spatial features such as edges and textures. Ran et al. [10] introduced a spatial pixel pair feature as an improved pixel pair feature. Gao et al. [11] proposed a CNN architecture for HSI classification. They reduced high correlation among HSI bands by transforming the 1D spectral vector to a 2D feature matrix and cascading composite layers. Li et al. [12] extracted the first principal component with enhanced spatial information using PCA and then passed it to a full CNN framework for classification. Zhong et al. [13] proposed a supervised spectral-spatial residual network, which extracts discriminative joint representation using a sequence of 3D convolutions in the corresponding spectral and spatial residual blocks. Instead of assigning fixed weights to incorporate spatial features, Li et al. [14] used an adaptive weight learning technique to reflect the variations in spatial contexture in diverse hyperspectral patches. Roy et al. [15] investigated a new architectural design that can adaptively find adjustable receptive fields and then an upgraded spectral-spatial residual network for joint feature extraction to make the convolutional kernel more flexible. The considerable improvement in the performance achieved on benchmark data sets has contributed to their demand. Singh et al. [16] used a Markov random field to improve the classification map produced by 3D CNN. However, the need for large training sets and high-end computing resources is the major concern of the CNN-based methods. The concept of transfer learning can help to greatly reduce these requirements. A CNN model previously trained for a specific problem can be fine-tuned for a new purpose [17] [18]. Such previously trained networks are known as pre-trained models. The pre-trained models alleviate the problem of large training samples required by deep learning-based techniques [19]. The performance of a model increases with the use of the transfer learning technique as it makes use of previous knowledge. The information from a genuine/reliable source is shifted to the targeted domain to acquire the knowledge of the data that is unseen or unlabeled. This makes transfer learning most effective to use in areas where training data samples availability is not in abun-

dance. Both the source and target domains are expected to be related but not identical in most cases. They may, however, have different distributions and the data in the two domains may differ due to different collection circumstances.

The goal of this work is to assess the efficacy of some pre-trained CNN models for HSI classification. Several pre-trained CNN networks including efficientNetB0, efficientNetB7, ResNet50, VGG19, DenseNet121, and DenseNet201 are fine-tuned and evaluated on two hyperspectral images Houston and KSC. The findings of this work will make it easier for remote sensing researchers to use powerful pre-trained CNN classifiers.

2 Methodology

Among the various deep learning methods, CNN is the method that has been the most popular in the field of computer vision and remote sensing. CNN method is highly effective for processing data represented in a grid structure, such as photographs. CNN is based on the anatomy of an animal's visual brain and is designed to automatically and adaptively learn spatial hierarchies of information, from low-level to high-level patterns. Convolutional neural networks (CNNs) have mainly three different types of layers: (1) convolution layer, (2) pooling layer, and (3) fully connected layers. The first two layers, the convolution and pooling layer, extract features, while the third, a fully linked layer, transfers those features into the final output, such as classification. The convolution layer, which is made up of a sequence of mathematical operations like convolution, is a key aspect of CNN. Convolution is a feature extraction process that uses a tiny array of numbers called a kernel to apply to the input. As input, a tensor of numbers is employed. At each point of the tensor, compute an element-wise product between each element of the kernel and the input tensor and add it to the output value at the corresponding place of the output tensor to produce a feature map. Convolution operation on a 2-D image I with a 2-D kernel K is performed as

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n) \quad (1)$$

Using several kernels, this process is repeated to generate an arbitrary number of feature maps that reflect various aspects of the input tensors. As a result, different kernels might be viewed as separate feature extractors. A stride is a distance between two consecutive kernel points, and it also specifies the convolution procedure. The stride of 1 is the most usual choice; however, the stride greater than 1 is occasionally used to accomplish feature map down-sampling. A nonlinear activation function is applied to the output of convolution that helps a model adapt to a variety of data. The most frequent activation function used is the rectified linear unit (ReLU).

To reduce the spatial amount of input, the pooling layer is typically employed after a convolution layer. It is applied separately to each depth slice of the input volume. A

kernel/size filter's is a matrix that is applied to all of the incoming data. The pooling layer reduces the size of feature maps,

$$(n_h - f + 1) / s * (n_w - f + 1)/s \; x \; n_c \qquad (2)$$

Where, n_h, n_w, and n_c are height, width, and the number of channels of the feature map respectively. Filter size is represented by f and s is stride length. Pooling is also advantageous for obtaining rotational and positional invariant dominant features, which aids the model's training process. Two typical pooling operations are maximum (Max) pooling and average pooling. The most frequently used max-pooling method selects the maximum value from the portion of the picture overlapped by the kernel. On the other hand, the average pooling operation calculates the average of all the values from the section of the image covered by the kernel. The number of such layers can be increased depending on the image complexity but at the cost of more processing resources. The output feature maps of the last convolution or pooling layer are usually flattened and turned into a one-dimensional (1D) array. Flattened features are projected onto one or more dense layers, which are fully connected. A learnable weight connects each input to each output in a dense layer. In the final fully linked layer, the number of output nodes is usually equal to the number of classes that produce the final output.

Transfer learning is a machine learning technique that seeks to apply knowledge from one task (source task) to a different but similar activity (target task). It is the process of improving learning in a new activity by transferring knowledge from a previously acquired related task. With the help of transfer learning, the need for a large number of training samples and high-end computing resources in CNN models can be reduced. Various pre-trained CNN models can be fine-tuned for hyperspectral image classification.

The VGG is a 19-layer deep convolutional neural network. (Karen Simonyan and Andrew Zisserman) [20] built and trained it at the University of Oxford in 2014. The VGG-19 network was trained using over 1 million pictures from the ImageNet collection. VGG19 network has been trained to classify up to 1000 items. Color images with a resolution of 224x224 pixels were used to train the network. Total trainable parameters in VGG19 are 143 million. In 2019, Google trained and released EfficientNet, a cutting-edge convolutional neural network, to the general public. There are eight different implementations of EfficientNet (B0 to B7). The number of layers in efficientnetB0 is 237 and in EfficientNetB7 the total layer comes out to 813. EfficientNetB0 has a total of 53 million trainable parameters. Total trainable parameters in EfficientNetB7 are 66 million. ResNet50 is a CNN with 50 layers of depth. Microsoft designed and trained it in 2015. In a feed-forward approach, DenseNet connects each layer to every other layer. Dense layers are linked by dense circuitry, in which each dense layer is capable of receiving feature maps from previous layers and passing them on to subsequent layers. In DenseNet121 there are 121 layers and 8

million trainable parameters while denseNet201 has 201 layers and 20 million trainable parameters.

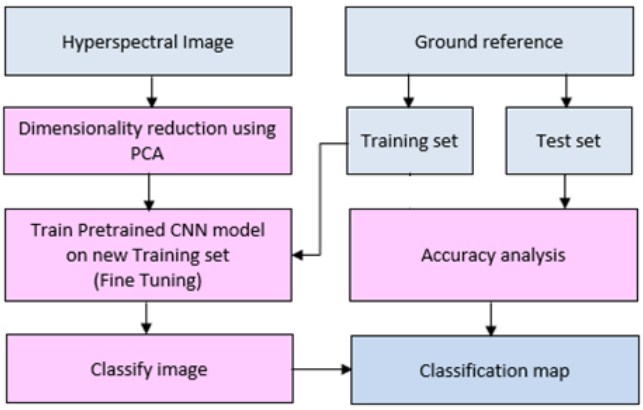


Fig. 1 Proposed classification Scheme

In this paper, a pre-trained CNN model-based classification scheme is proposed for hyperspectral imagery as shown in Fig. 1. The pre-trained networks accept only three band images. Therefore, the input hyperspectral image is first reduced using principal component analysis. The reduced image is then sent to the CNN model, which is fine-tuned with the training samples from the ground reference image. Once the CNN model is fine-tuned, it can classify the input image. The results are evaluated on the test set.

3 Experiments and Results

The experiments are performed to evaluate the performance of the six different pre-trained CNN models: EfficientNetB0, EfficientNetB7, ResNet50, VGG19, DenseNet121, and DenseNet201. All tests are performed on a computer with a Xeon 4.5 GHz processor, 64 GB RAM, and an 8 GB Quadro P4000 GPU. The code is written in Python and uses a variety of frameworks including Keras and TensorFlow.

3.1 Data Sets

Two well-known hyperspectral images, Houston University and KSC are used in the experiments. An aerial ITRES-Compact Airborne Spectrographic Imager 1500 hyperspectral imager was used to capture the image of the University of Houston campus and the nearby urban area. The image has spatial dimensions of 1905×349 pixels with a spatial resolution of 2.5m per pixel. It consists of a total of 15 classes. The FCC and ground reference of the Houston University image is given in Fig. 2. There are two standard sample sets for Houston University. Both these sets are combined in this work to form the ground reference as shown in Fig. 2(b).

The second image is acquired using the AVIRIS instrument over the KSC in Florida, USA, in 1996. After removing noisy bands, the final image contains 176 bands with a 512 x 614 size, ranging from 400 to 2500 nm, and 20m spatial resolution. The ground truth available contains 13 classes. Fig. 3 shows the FCC and ground reference of the KSC image. The classwise number of training and test samples for both images are shown in Table 1.

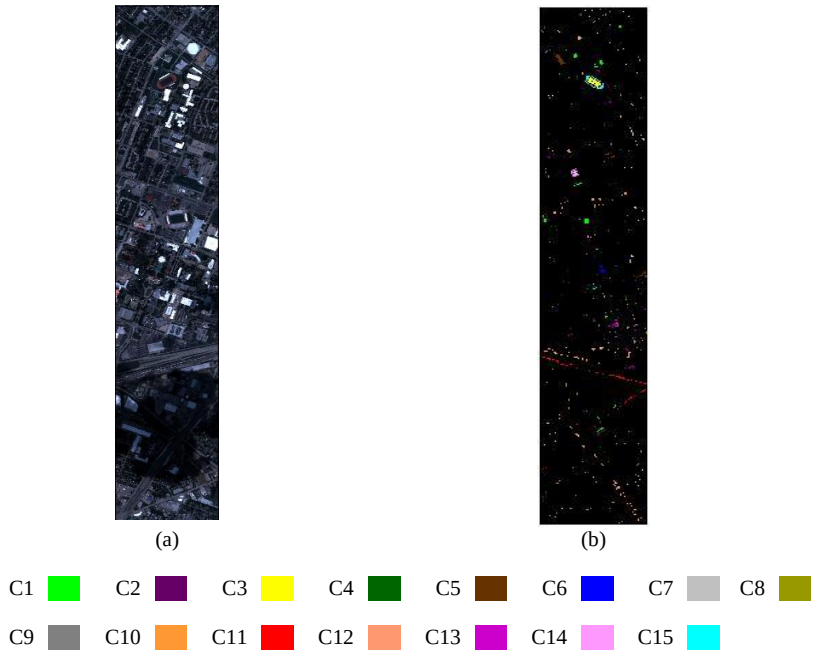


Fig. 2 Houston University (a) FCC (b) ground reference

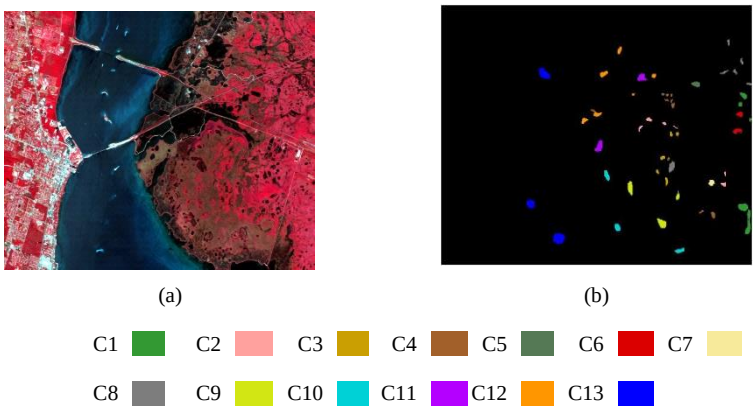


Fig. 3 KSC (a) FCC (b) ground reference

Table 1. Details of information classes in hyperspectral images

Houston University			Kennedy Space Center		
Class	**Training**	**Testing**	**Class name**	**Training**	**Testing**
Healthy grass	125	1126	Scrub	76	685
Stressed grass	125	1129	Willow swamp	24	219
Synthetic grass	70	627	CP hammock	26	230
Trees	124	1120	Slash pine	25	227
Soil	124	1118	Oak/Broadleaf	16	145
Water	33	292	Hardwood	23	206
Residential	127	1141	Swamp	11	94
Commercial	124	1120	Graminoid marsh	43	388
Road	125	1127	Spartina marsh	52	468
Highway	123	1104	Cattail marsh	40	364
Railway	124	1111	Salt marsh	42	377
Parking Lot 1	123	1110	Mud flats	50	453
Parking Lot 2	47	422	Water	93	834
Tennis Court	43	385			
Running Track	66	594			

3.2 Results and Discussion

The accuracy of a classification and other parameters are examined in the findings. The accuracy of classification is measured in terms of overall kappa and overall accuracy (OA). The ground reference is used to select the training pixels randomly. For Houston University, 5% of samples are used for training and the remaining samples are retained for testing. The evolution of training and validation accuracy as a function of the number of epochs is depicted in Fig. 4 and Fig. 5 for all the tested models for Houston University and KSC images respectively. It can be very clearly observed that the training and validation accuracy curves are very similar indicating that pre-trained models are well fit for the problem considered in this work.

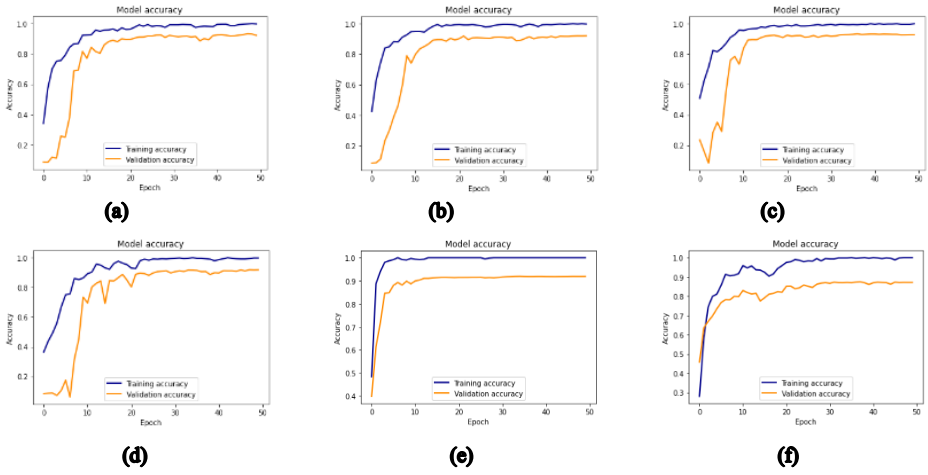


Fig. 4 Training accuracy and validation accuracy for Houston image (a) Densenet121 (b) Densenet201 (c) EfficientnetB0 (d) EfficientnetB7 (e) Resnet50 (f) VGG19

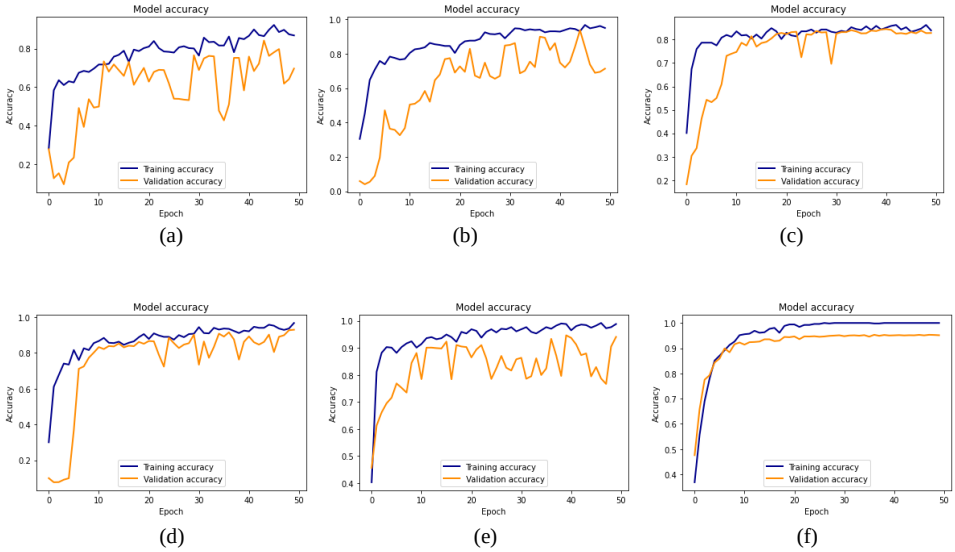


Fig. 5 Training accuracy and validation accuracy for KSC image (a) Densenet121 (b) Densenet201 (c) EfficientnetB0 (d) EfficientnetB7 (e) Resnet50 (f) VGG19

Table 2. Classification accuracy (OA in % and kappa) for Houston University.

Class	Pretrained CNN Models					
	ENB0	**ENB7**	**RN50**	**DN121**	**DN201**	**VGG19**
Healthy grass	89.37	87.93	77.94	88.81	86.57	88.73
Stressed grass	88.12	79.59	87.72	87.48	77.35	78.39
Synthetic grass	95.12	96.41	95.55	97.13	97.85	92.97
Trees	81.75	87.54	74.28	88.34	77.09	69.37
Soil	95.73	95.81	93.80	94.69	96.62	93.48
Water	83.38	67.69	60.31	59.69	57.23	60.62
Residential	94.56	88.41	86.99	84.38	86.36	68.38
Commercial	86.17	85.29	84.65	83.12	83.04	79.82
Road	88.26	92.49	87.54	83.47	91.69	84.98
Highway	99.51	95.68	96.58	97.80	97.88	89.81
Railway	99.43	94.09	99.27	97.89	97.73	92.39
Parking Lot 1	87.75	86.78	91.08	95.13	94.00	86.29
Parking Lot 2	91.26	87.42	82.94	89.34	92.54	66.31
Tennis Court	100	98.13	99.77	99.77	100	96.03
Running Track	78.64	92.88	67.88	87.73	82.42	87.88
Overall OA (%)	90.79	89.54	87.00	89.89	88.69	83.14
Overall Kappa	0.9004	0.8869	0.8593	0.8907	0.8776	0.8175

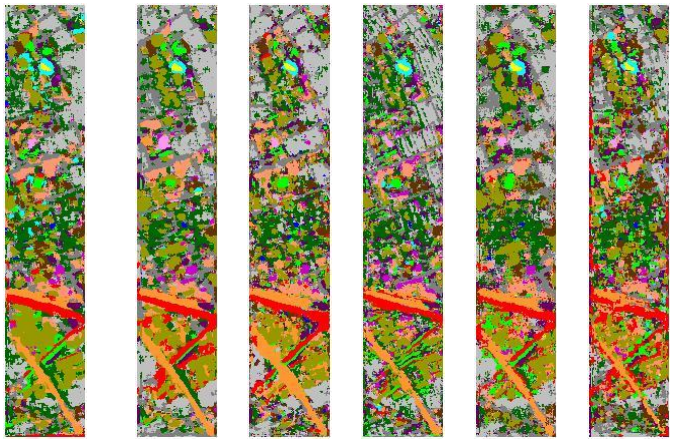


Fig. 6 Houston University classification maps: (a) denseNet121, (b) denseNet201, (c) efficientNetB0, (d) efficientNetB7, (e) resNet50, (f) vgg19

Classification performance. The classification accuracy of all six pre-trained models for the Houston University image is presented in Table 2. Both classwise and overall accuracy values are reported for all six pre-trained models. It can be seen that the EfficientNetB0 gives the best results. The overall kappa and OA values for EfficientNeB0 are 0.9004 and 90.79% respectively. Most of the classes except Trees, Water, and Running track provides good classification accuracy. EfficientNetB7 and DenseNet121 are other models that exhibited similar performance. VGG19 is the model with the lowest accuracy of 83.14%. VGG19 produces its maximum accuracy of 96.03% for the Tennis Court class. For class water, all models except EfficientNetB0 give an accuracy of less than 70%. Based on the results it is highly recommended to use EfficientNetB0 for performing classification on images containing large areas with water and if training samples of water are very less. Similarly, for class Tree, DenseNet121 and EfficientNetB7 should be selected for classifying images having trees in a large area as they give the highest accuracy of 88.34% and 87.54% respectively. A similar type of observation can be done on other classes which assist in selecting the effective model for performing classification on the targeted image. The classification maps for Houston University are shown in Fig. 6 for visual inspection.

The classification results for the KSC image are reported in Table 3. For the KSC image, the VGG19 produces the highest OA 95.77%. It gives a global kappa value of 0.9530. Interestingly DenseNet121, which is one of the best performing models for Houston Image, does not perform well for KSC. For Graminoid marsh, DenseNet121 gives a very poor accuracy of 35.50%. All six models perform well for Scrub, Saltmarsh, Mudflats, and water classes. For class Cattail marsh, EfficientNetB0, EfficientNetB7 and DenseNet201 gives 100% accuracy. Fig. 7 shows the classification maps for the KSC image. Pre-trained models perform well for the hyperspectral images but different models perform differently for different images.

Table 3. Classification accuracy (OA in % and kappa) for Kennedy Space Center

Class	Pretrained CNN Models					
	ENB0	**ENB7**	**RN50**	**DN121**	**DN201**	**VGG19**
Scrub	98.16	95.27	97.90	92.51	98.82	97.77
Willow swamp	78.19	82.72	97.94	73.66	64.20	92.18
CP hammock	71.48	85.94	66.80	38.67	79.30	93.75
Slash pine	88.89	91.67	84.92	89.29	88.89	81.75
Oak/Broadleaf	97.52	96.89	87.58	93.79	100	100
Hardwood	53.28	100	96.07	74.24	76.86	92.58
Swamp	77.14	100	95.24	63.81	76.19	98.10
Graminoid marsh	50.12	59.16	87.24	35.50	93.50	87.70
Spartina marsh	50.96	98.85	99.62	97.69	94.62	96.92
Cattail marsh	100	100	99.50	85.40	100	95.05
Salt marsh	99.76	100	98.57	95.47	99.28	99.05
Mud flats	98.41	98.61	98.81	96.42	100	99.20
Water	99.14	98.38	99.46	99.89	99.78	99.35
Overall OA (%)	84.84	93.39	95.12	84.66	93.93	95.77
Overall Kappa	.8299	.9265	.9456	.8288	.9324	.9530

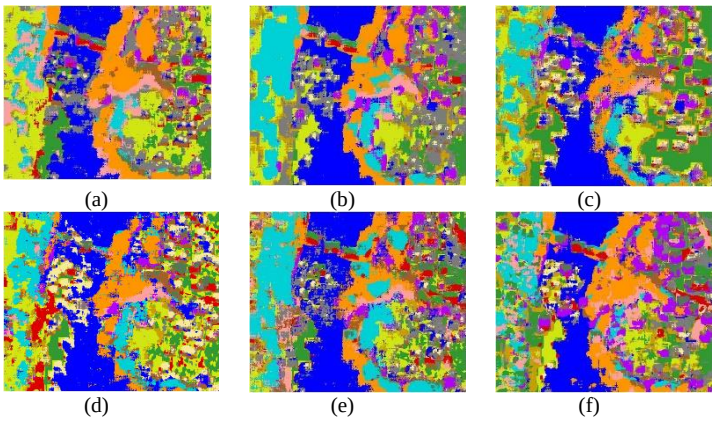
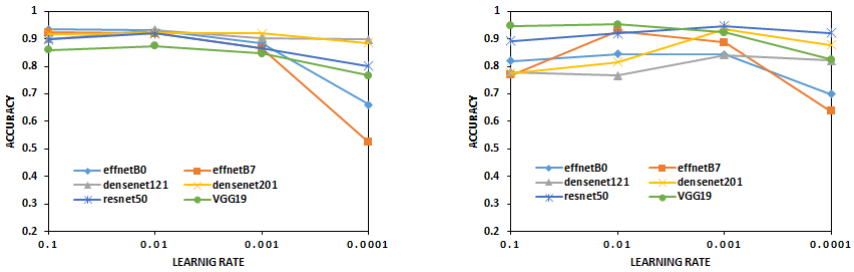


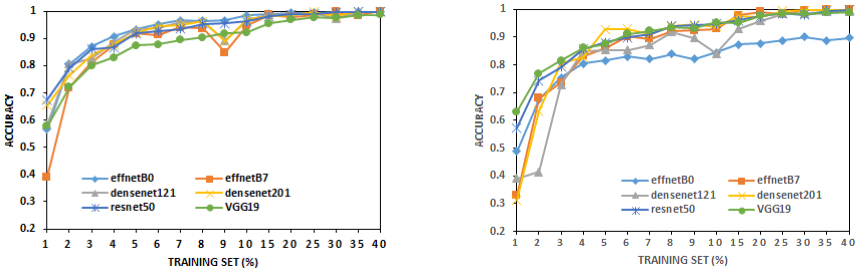
Fig. 7 KSC classification maps: (a) denseNet121, (b) denseNet201, (c) efficientNetB0, (d) efficientNetB7, (e) resNet50, (f) vgg19

The classification accuracy is plotted against the number of epochs in Fig. 4 and Fig.5. It is observed from the graphs that classification accuracy becomes stable after a small number of epochs. In most cases, only ten epochs are good enough for fine-tuning a pre-trained CNN model for hyperspectral image classification. Therefore, it also alleviates the need for high-end computing resources.



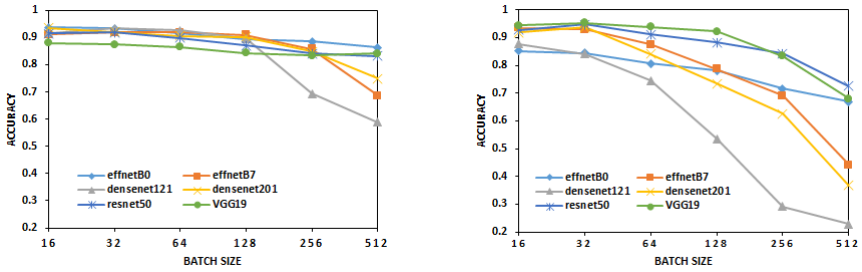
(a) (b)

Fig.8 Effect of learning rate (a) Houston (b) KSC



(a) (b)

Fig. 9 Effect of training samples (a) Houston (b) KSC



(a) (b)

Fig. 10 Effect of batch size (a) Houston (b) KSC

Effect of learning rate. The effect of the learning rate on classification accuracy can be observed in Fig. 8, where classification accuracy is plotted against the learning rate. The learning rate varies from 0.0001 to 0.1. It is observed from the figure that most of the models give their best accuracy at 0.1 or 0.01. For both images, accuracy sharply decreases at a learning rate of 0.0001.

Impact of training samples. Experiments are performed to observe the effect of training samples on classification accuracy. Training set size is varied from 1% to 40%. The impact of training set size on classification accuracy is shown in Fig. 9. It can be observed that accuracy increases with the increase in training sample size. It can be seen that accuracy increases sharply between 1% and 4% training size, the rate of increase in accuracy is moderate between 5% and 10%, and increase in accuracy becomes very slow between 10% and 40% training sample size. Overall, it can be concluded from this experiment that pre-trained CNN models can give good classification accuracy with a smaller training set.

Impact of batch size. Fig. 10 shows the impact of batch size on accuracy. Experiments are carried out to observe how classification accuracy gets affected by the change in the batch size. Batch size is varied from 16 to 512. It is selected as the power of 2. The output of the experiment shows that all models produce the highest accuracy at batch size 32. As batch size increases accuracy started decreasing, and the lowest accuracy is obtained at batch size 512. Therefore, it is observed that a smaller batch size gives better results for classification.

4 Conclusion

In this paper, six pre-trained CNN models were fine-tuned for hyperspectral image classification. The experiments were carried out on Houston University and KSC images. It was observed from the experimental results that EfficientNetB0 produced the highest accuracy of 90.79% for the Houston image and VGG19 yielded the highest accuracy of 95.79% for the KSC image. It is found that only a small number of epochs are needed to fine-tune the pre-trained model for hyperspectral classification to get good accuracy. The pre-trained models can perform well with a smaller training set and give effective results using transfer learning. One single model cannot be attributed as the best model that can be applied to all types of images. Instead, a model can be selected based on the image content. The experiment results shown in this paper will be useful for researchers in selecting the pre-trained networks according to the image content which is to be classified. For example, the pre-trained network ENB7 should be selected instead of ENB0 for classifying the image containing mostly hardwood material, as we can see in Table 3, ENB0 gives only 53.28% accuracy while ENB7 produces 100% accuracy for the hardwood class.

References

1. X. Cao et al., "Hyperspectral image classification with convolutional neural network and active learning," IEEE Trans. Geosci. Remote Sens. 58(7), 4604–4616 (2020).
2. Kumar, B., Dikshit, O.: Spectral contextual classification for hyperspectral imagery with probabilistic relaxation labeling. IEEE Transactions on Cybernetics. 47(12), 4380-4391 (2017).

3. Kumar, B.: Hyperspectral image classification using three-dimensional geometric moments. IET Image Processing. 14(10), 2175–2186, (2020).
4. Kumar, B., Dikshit, O., Gupta, A., and Singh, M. K.: Feature extraction for hyperspectral image classification: a review. International Journal of Remote Sensing. 41(16), 6248-6287, (2020).
5. Paoletti, M., Haut, J., Plaza, J. Plaza, and Plaza, A.: Deep learning classifiers for hyperspectral imaging: A review. ISPRS Journal of Photogrammetry and Remote Sensing. 158, 279–317, (2019).
6. Li, S. et al.: Deep learning for hyperspectral image classification: an overview. IEEE Trans. Geosci. Remote Sens. 57(9), 6690–6709 (2019).
7. Wu, H., Prasad, S.: Convolutional recurrent neural networks for hyperspectral data classification. Remote Sensing. 9(3), 298, (2017).
8. Ding, C., Li, Y., Xia, Y., Wei, W., Zhang, L., Zhang, Y.: Convolutional neural networks based hyperspectral image classification method with adaptive kernels. Remote Sensing. 9(6), 618, (2017).
9. Chen, Y., Zhu, L., Ghamisi, P., Jia, X., Li, G., Tang, L.: Hyperspectral images classification with gabor filtering and convolutional neural network. IEEE Geoscience and Remote Sensing Letters. 14(12), 2355–2359 (2017).
10. Ran, L., Zhang, Y., Wei, W., Zhang, Q.: A hyperspectral image classification framework with spatial pixel pair features. Sensors, 17(10), 2421, (2017).
11. Gao, H., Yang, Y., Li, C., Zhou, H., Qu, X.: Joint alternate small convolution and feature reuse for hyperspectral image classification. ISPRS International Journal of Geo-Information. 7(9), 349, (2018).
12. Li, J., Zhao, X., Li, Y., Du, Q., Xi, B., Hu, J.: Classification of hyperspectral imagery using a new fully convolutional neural network. IEEE Geoscience and Remote Sensing Letters, 15(2), 292–296 (2018).
13. Zhong, Z., Li, J., Luo, Z., Chapman, M.: Spectral–spatial residual network for hyperspectral image classification: A 3-d deep learning framework. IEEE Transactions on Geoscience and Remote Sensing, 56(2), 847–858 (2018).
14. Li, S., Zhu, X., Liu, Y., Bao, J.: Adaptive spatial-spectral feature learning for hyperspectral image classification. IEEE Access. 7, 61534– 61547 (2019).
15. Roy, S. K., Manna, S., Song, T., Bruzzone, L.: Attention-based adaptive spectral-spatial kernel resnet for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing. 59(9), 7831–7843 (2021).
16. Singh, M. K., Mohan, S., Kumar, B.: Hyperspectral image classification using deep convolutional neural network and stochastic relaxation labeling. J. Appl. Rem. Sens. 15 (4), 042612, (2021).
17. Marmanis, D., Datcu, M., Esch, T., and Stilla, U.: Deep learning earth observation classification using ImageNet pretrained networks. IEEE Geosci. Remote Sens. Lett. 13(1), 105–109, (2016).
18. Romero, A., Gatta, C., Camps-Valls, G.: Unsupervised deep feature extraction for remote sensing image classification. IEEE Trans. Geosci. Remote Sens. 54(3), 1349–1362 (2016).
19. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 3320–3328.
20. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: 3rd International Conference on Learning Representations, pp. 1-14. dblp, San Diego, CA, USA (2015).