Hyperspectral Image Classification with Transfer Learning

by Devang Kiritbhai Savaliya, Shubhankar Kulkarni

Abstract

Hyperspectral images (HSI) contain tremendous amount of spectral and spatial information in large number of bands covering different wavelengths. This huge amount of information makes the task of hyperspectral image classification challenging with less training data available. Traditional image processing techniques along with convolutional neural networks (CNN) have limitations in improving the classification accuracy of hyperspectral images. To overcome these limitations caused due to a lack of labelled training data, techniques like transfer learning are explored to obtain better classification results. In this project, three popular HSI datasets, Salinas, Pavia University and Indian Pines are selected to conduct classification experiments amongst these datasets by comparing results with and without transfer learning. Modified version of 3D Res-Net was implanted to train on the source dataset with large amount of training samples and then the feature extraction part of this model was transferred to target dataset.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

The primary objective is to demonstrate the effect of transfer learning between two hyperspectral datasets. Although deep learning has made great progress in the field of hyperspectral image (HSI) classification, deep networks tend to overfit when the training data is limited and fail to generalize well to unseen data. To solve this problem, either more data can be collected of transfer learning can be used. Transfer learning involves pre-training the model on a dataset with large number of labelled samples and transferring the learned weights to a target data set for better classification results. Basically, training the feature extraction part of a larger but similar data set, and transferring those feature extraction weights to target model and fitting just the classification part of the target model. The proposed method seeks to examine whether the use of transfer learning between two hyperspectral images leads to an increase in classification accuracy over the model that was trained totally on a single dataset.

2. HSI classification

The abundant spectral and spatial characteristics of materials in a hyperspectral image help in classifying hyperspectral images [1]. Due to this, the classification of hyperspectral remote sensor images plays a crucial role in applications related to urban development and resource management. Existing methods for classifying hyperspectral images have focused on exploring the spectral information only, while the spatial information is left unused. Classifying these images by only using the spectral information is difficult due to high dimensional spectra combined with limited number of samples in the training set [1]. To improve the classification performance by considering both spectral and spatial characteristics of a hyperspectral image, a 3D Residual Network (3D ResNet) was used. As the training samples are limited, the deeper network will most probably overfit and fail to generalize to unseen data. Thus, it becomes necessary to combine the use of deep residual network with techniques like transfer learning.

2.1 ResNet & 3D separable ResNet (3D SR-Net)

Neural Networks have become popular in recent times with tremendous research and development done these days. Fields like Image Processing, Computer Vision and Natural Language Processing and other applications are flourishing amongst several others. Computational feasibility of computers has played a crucial role in this, enabling researchers to build deeper neural networks, with the state-of-the-art neural networks going from a few layers to over hundred layers. However, a drawback in training deeper neural networks is of vanishing gradients. Deep networks often have a backpropagated gradient signal that take values that go to zero very quickly, making the gradient descent very slow. A key approach in residual networks is to add a shortcut or a skip connection that allows information to flow easily from one layer to the N^{th} layer (skipping the layers in between), it will bypass data along with normal flow from one layer to the next layer. The architecture of a residual block is shown in figure 1. Some advantages of a residual block are that the addition of new layers using skip connections will not degrade the performance of the model as regularization will skip the newly added layers even if they are not useful. Deeper networks can be built by stacking residual blocks on one another. Basically, making a series of these units to make a complex model. Figure 1 also shows the separable residual block. Key difference here is that the 3d convolution layers in broken down into two layers, one only for spectral and another only for spatial information.

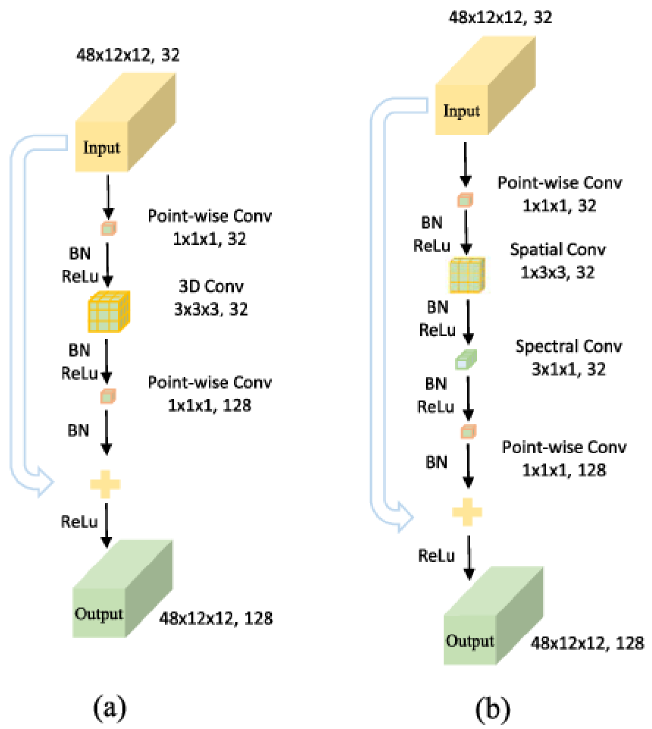


Figure 1. 3-D residual unit versus a 3-D separable residual unit (SR Unit) [1]

3D-ResNet was implemented by expanding the 2D convolution layer to 3D convolution layer. In comparison to a 2D-ResNet, a 3D-Resnet has huge number of parameters and to avoid overfitting for a 3D-ResNet, the spectral components are separated from the spatial components which make a separable residual (SR) unit. Figure 2 shows the 3-D-SRNet classification model. The architecture of a SR unit is compared with a conventional residual unit in figure 1. The 3D residual unit is replaced by a SR unit to form a 3D-SRNet with fewer parameters than a 3D-ResNet. The 3D-SRNet is the underlying model used for training the network on a source hyperspectral dataset, this is the pretraining stage in transfer learning [1]. The pretraining stage is followed by a fine-tuning stage where the entire model is transferred except the fully connected layers to the network implemented for the target hyperspectral dataset [1]. Figure 2 shows a complex version of what was implemented. For feature extraction 2 consecutive SR Unites were implemented and hyper parameters for SGD optimizer were, 1e-3, 1e-5, 0.9, glorot uniform for learning rate, decay, momentum and kernel initializer respectively. Batch size and number of epochs were selected based on the data size. Model was implemented by the modifying an ResNet available online [3].

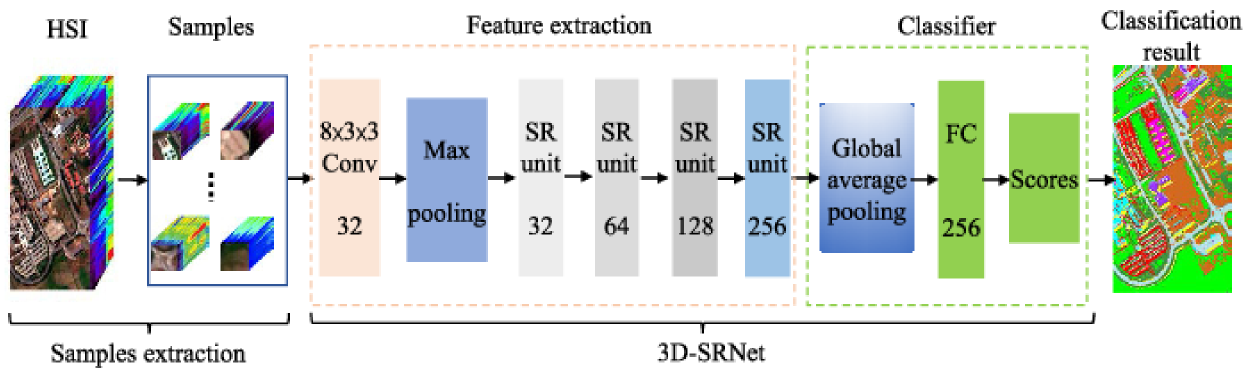


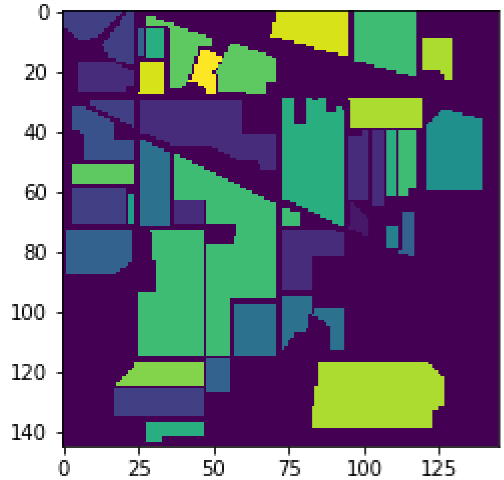
Figure 2. Block diagram of 3-D-SRNet classification model [1]

2.2 Dataset description

For conducting classification experiments for classifying hyperspectral images using 3D-SRNet and transfer learning, three popular and widely used hyperspectral datasets were used. Indian Pines dataset, the Pavia University dataset and the Salinas Valley dataset. The Indian Pines dataset has images of size 145 pixels x 145 pixels with 224 spectral channels with wavelength ranging from 0.4 – 2.5 micrometers [2]. There are 16 land-cover classes in the ground truth image for the Indian Pines dataset. The Pavia University dataset has images of size 610 pixels x 340 pixels with a total of 115 spectral channels ranging from 0.43 – 0.86 micrometers [2]. The ground truth image for the Pavia University dataset has 9 land cover classes. 103 spectral bands are used for classification after the noisy bands were removed. The Salinas Valley dataset has images of size 512 x 217 pixels with 224 spectral bands with range of 0.4 to 2.5 micrometer [2]. The ground truth image has 16 classes for the Salinas Valley dataset. Salinas Valley and Indian Pines datasets are captured by same sensors and their spectral and spatial characteristics are similar [2]. As the Pavia University dataset is captured with a different sensor, classification experiments between Pavia University and Salinas Valley datasets are expected to obtain results that show the effect of transfer learning [2].

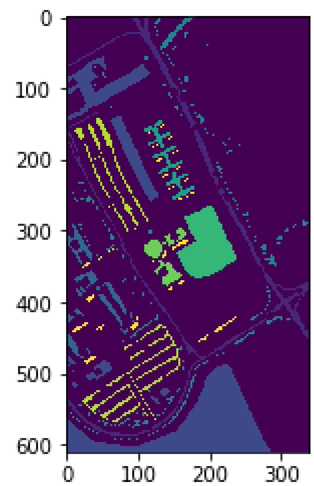
2.3 Data pre-processing

These hyperspectral images were single image with everything in one hyperspectral cube. To work with the 3D-SRNet small samples of 25x25x103(spectral bands) were selected from the image and the class of central pixel in the ground truth image was the class assigned to these cubes (sample). Multiple different datasets were generated based on the overlap ratio of this small cubes. Meaning, an overlap ration of 25% would mean, a sample belonging to a class will be selected only if next sample from same class dose not overlap more than 25% over this sample. Datasets for 25%, 50%, 75% and 95% overlap ratios were generated for Salinas and Pavia. For Indian Pines dataset overlap ratios were higher because of the size of the dataset. Although more overlap will give more data points, but it will also mean that the data will be repeating more compared to less overlap ratio. Goal is to get better accuracy on the data generated with less overlap ratio by transfer learning.



a) Indian Pines

b) Salinas



c) Pavia

Figure 3. Ground truth images for all three datasets [2]

Important thing to notice in figure 3 is that most of this ground truth images are background. If background is selected as class, then the data will have class imbalance. So, for that reason samples with center pixel as background were dropped. Indian Pines data is bigger in size but the way samples are selected, it ended up having less samples. Mainly because of the constrains to while selecting the sample (overlap ratio condition) Another way to assign the class would be to do that based on the majority pixel in the ground truth.

2.4 Model generation and transfer learning

Model described above was trained for all variation (different overlap ratios) of all three data sets. These are source models which include the classification part. To transfer only the feature extraction part, the fully connected layers were dropped before saving the model. Later these sub models can be called to predict the intermediate output which is then passed through new fully connected layers. These fully connected layers are then trained on the target data set. Figure 4 shows the block diagram of transfer learning. Here in the figure, source dataset is Salinas and target datasets are Indian pines and Pavia.

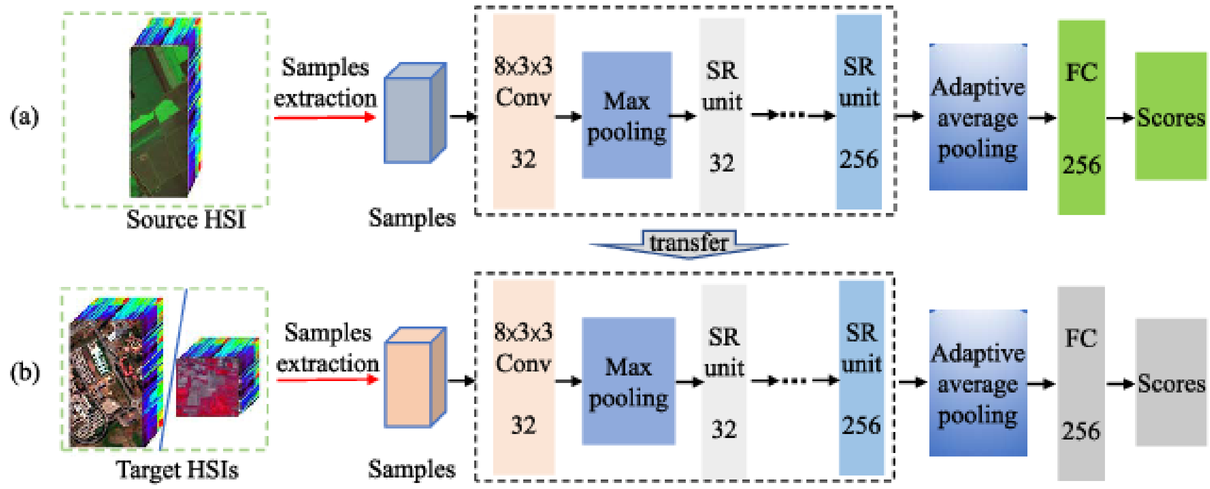


Figure 4. Block diagram of transfer learning [1]

Considering all the combinations 144 models were generated (4 source model for a dataset transferred to 4 different target datasets, time 9 combination for 3 unique datasets). Table 1 below shows the accuracies of the source models. Obvious trend is seen here, as the overlap ratio increases the test accuracies are increasing. As mentioned, every possible combination of source model from this table (12 different source model) were transferred to all the variations (overlap ration) of these three datasets and were saved. This report will focus on the cases were the target dataset was with the least overlap ratio (less data points) in each dataset (Salinas, Pavia and Indian pines). Source model were the once with highest overlap ratio (more data points). Transfer learning by definition, is a technique to transfer learning from the model trained on huge datasets to the model to be trained on less but similar dataset.

Salinas		Pavia		Indian Pines	
Overlap Ratio	**Test Accuracy**	**Overlap Ratio**	**Test Accuracy**	**Overlap Ratio**	**Test Accuracy**
25%	76.4706%	25%	71.9512%	50%	40.9091%
50%	79.1209%	50%	61.5385%	75%	63.9344%
75%	88.5417%	75%	71.7105%	85%	60.3550%
95%	94.5182%	95%	84.9275%	95%	77.0701%

Table 1. Source model test accuracies

2.5 Results and accuracies

Expectation was to improve test accuracy for the datasets with least overlap ratio. Table 2 below shows the effect of transfer learning for combinations of source and target dataset. Table 2 also includes the transfer learning for source and target being from same dataset. Only two of these combinations didn't show improvement in the test accuracy. All other combinations either gave the same results or were improved by transfer learning. Accuracies are highlighted based on the performance with transfer learning.

Source model (Overlap 95%)	Target Dataset					
	Indian Pine (Overlap 50%)		Pavia (Overlap 25%)		Salinas (Overlap 25%)	
	Without transfer learning	With transfer learning	Without transfer learning	With transfer learning	Without transfer learning	With transfer learning
Indian Pines		63.6364%		71.9512%		70.5882%
Pavia	40.9091%	54.5455%	71.9512%	81.7073%	76.4706%	84.3137%
Salinas		40.9091%		54.8781%		96.0784%

Table 2. Transfer learning test accuracies

3. Summary & conclusion

Table 2 also shows some cases for which accuracy with transfer learning did not improve at all, an example case was for Pavia as the target dataset when source dataset was Indian pines or Salinas. For Indian pines as target, improvement can be seen with source being better model of itself or source being Pavia, and the same goes for Salinas. Source model Pavia performed comparatively better than others. This doesn't not imply that Indian pines and Salinas didn't give a good performance. It just indicates that Pavia dataset turns out to be a good source dataset. Looking at test accuracies from many other combinations which are not mentioned here, there is still some room for improvement in the way samples are selected and the way class labels are assigned.

Considering the results in table 2, this technique surely has great potential. Basic concept of transfer learning is not that hard to grasp, learning feature extraction on a similar but bigger dataset and learning information about class prediction on target dataset. Going forward this project can easily be scaled to another set of datasets with some minor changes.

4. References

1. Y. Jiang, Y. Li and H. Zhang, "Hyperspectral Image Classification Based on 3-D Separable ResNet and Transfer Learning," in IEEE Geoscience and Remote Sensing Letters.
doi: 10.1109/LGRS.2019.2913011

2. http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

3. https://github.com/pantheon5100/3D-CNN-resnet-keras/blob/master/Cre_Model.py