Vegetation Classification in Hyperspectral Image

Osman Cihan Kilinc - Kazim Ergun - Yuming Qiao - Fengjunyan Li

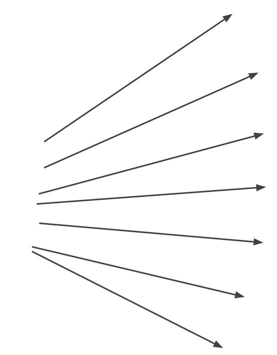
Introduction

Hyperspectral image collects and process information across electromagnetic spectrum. It is a efficient method to detect the segmentation and health of vegetation. In this project, we will classify vegetation segmentation in aerial hyperspectral imagery. The input will be a 145x145x224 image with 16 different vegetation classes. The 145x145 represents the image height and width, 224 is the number of spectral bands. Our goal is to train a classifier that can learn the 224 band patterns, and predict the classes for every data point. Different methods of data preprocessing are implemented, such as PCA and data augmentation. We propose SVM and convolutional neural network to solve the problem and implemented noise correction to boost the result. Our approaches can achieve relatively high accuracy in the satellite imaging. People can use our proposed method to track the growth of vegetation in a fast, precise, and low-cost manner.

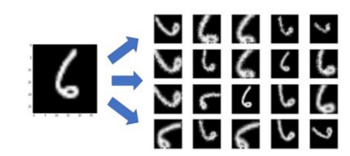
Dataset and Data Preprocessing

Indian Pines

- 145x145 pixels
- 16 classes
- 224 Spectral Bands

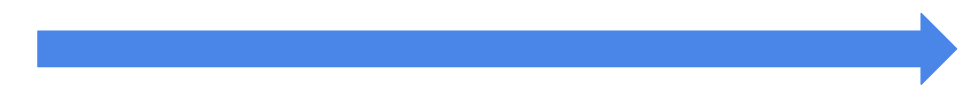


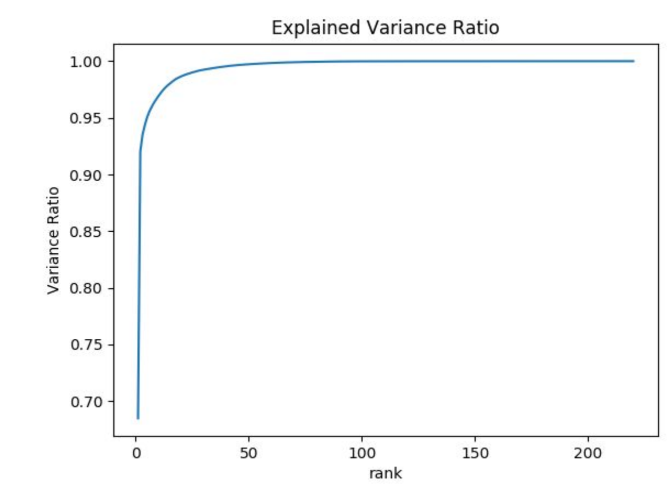
Oversampling



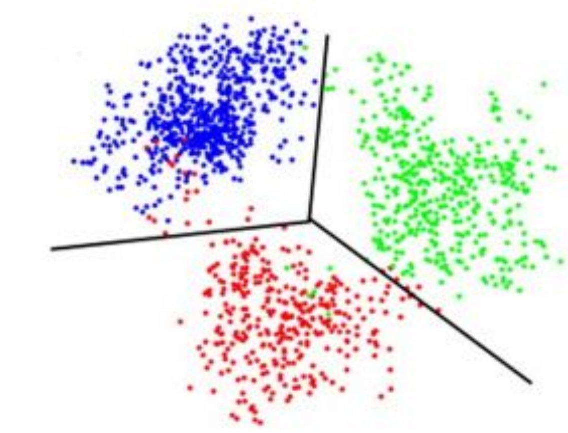
Data Augmentation

PCA & NMF





CNN & SVM

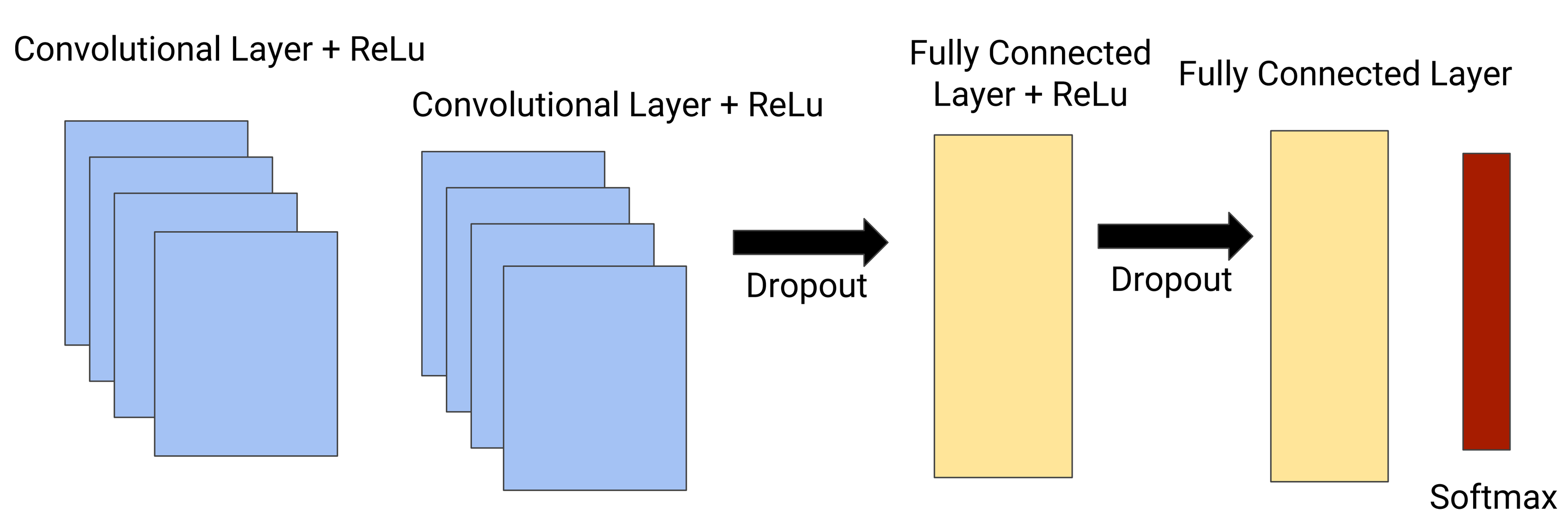


Data Preprocessing

Classification

In this project, we mainly use Indian Pines dataset. To reduce the dimensionality of the data, we implement PCA and pick first 30 principal components. We also oversample the weak classes, which have small number of samples. Finally, we use data augmentation methods such as rotation and vertical flips to introduce variability into our dataset.

CNN Architecture



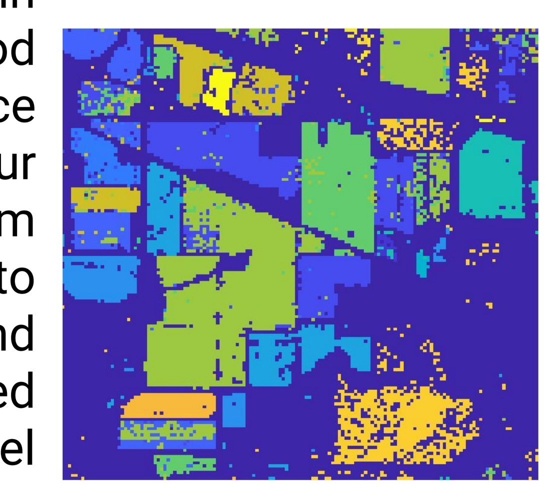
CNN Optimizations

We can further optimize and accelerate our model by layer compression. Each neuron in the fully connected layers is connected to all neurons in the next layer. We use PCA to determine the vital principal components and replace the first fully connected layer with its decomposition. Finally, we plotted the results for changing compression ratios. Given a fully connected layer with m inputs and n outputs. There are mxn parameters (weights). Given a compression ratio k, we reduce the number of parameters to kx(m+n). When k is small, the change is dramatic. This both accelerates our model and reduces the resources our model uses. The results are promising.

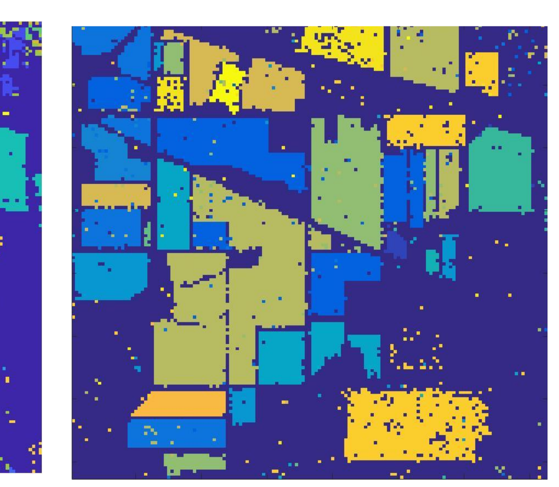
SVM Method & Results

Support Vector Machine is a classification method under supervised learning. SVM looks for the hyperplane that maximize the margin between different classes. Kernel method transforms the raw data to feature space using a user-specified feature map. In our case, the PCA reduces the dimension from 224 to 30. Then we use different kernels to project the reduced data to feature space and run the training and classification. We tested different kernels, and the Polynomial Kernel (order = 3) produces the best result.

Gaussian Kernel
Accuracy = 80.7%



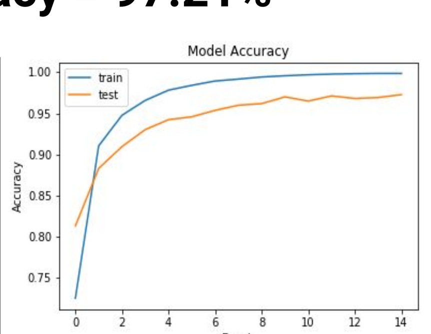
Polynomial Kernel (p=3)
Accuracy = 82.7%



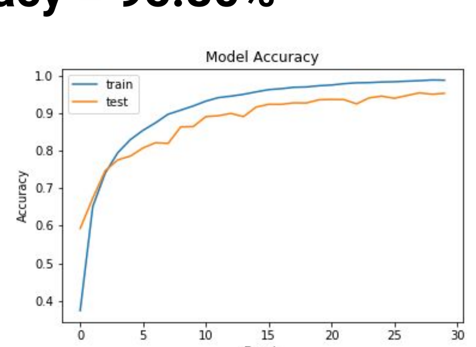
CNN Results

Convolutional neural networks has gained popularity for image classification tasks, since they automatically construct and utilize hidden features. Using the first 30 principal components, we reach 97.21% accuracy with only after 10 epochs. We use dropout layers to prevent overfitting and Adam for gradient descent optimization. Results can be seen from the figures below.

PCA - 30 Components
Accuracy = 97.21%

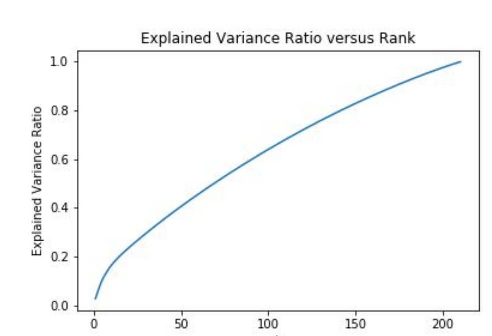


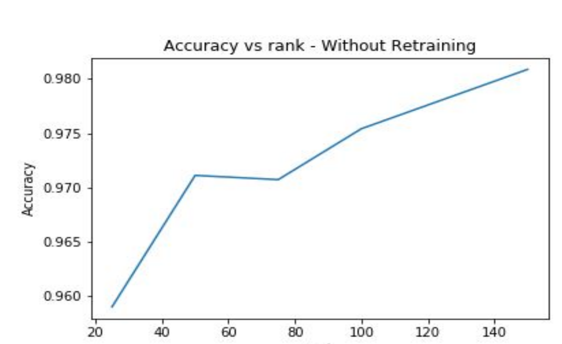
NMF - 30 Components
Accuracy = 95.80%

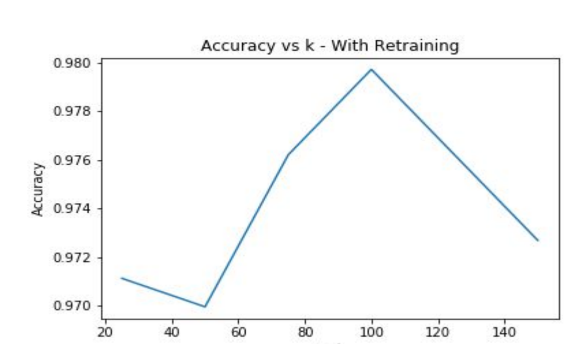


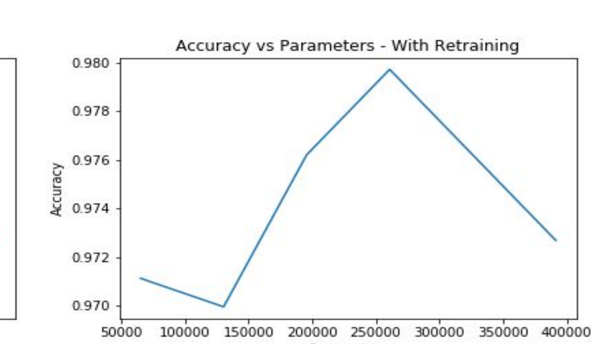
CNN Optimization Results

Layer compression decreases accuracy very little, since it provides a compressed representation of the weight matrix. Moreover, it prevents overfitting, which provides a %0.5 accuracy gain. Although the explained variance ratio plot is nearly linear, compression is still useful.





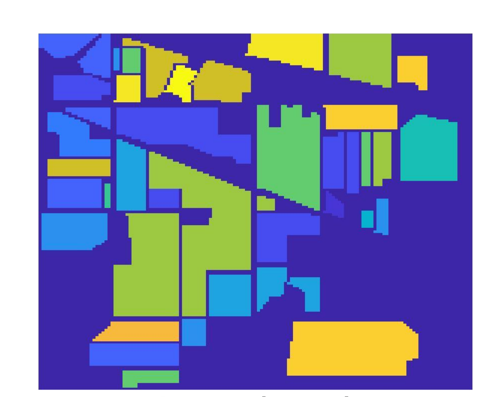




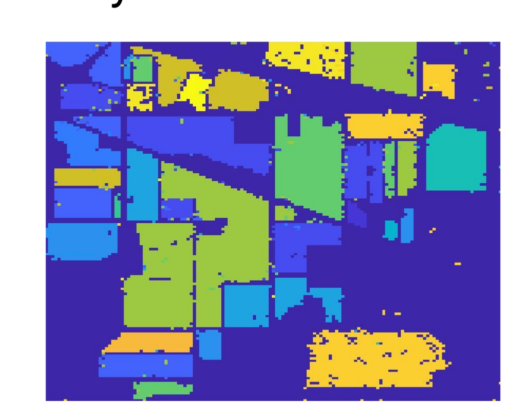
Conclusion

In order to overcome the noise and boost the result, we apply noise correction at the end by filtered out the dangling points - point whose neighbours all belong to the same other class. As the image shows, this corrects a large number of misclassified points in SVM. This method boosts the SVM results by ~1.2%, and the CNN result by 0.2%.

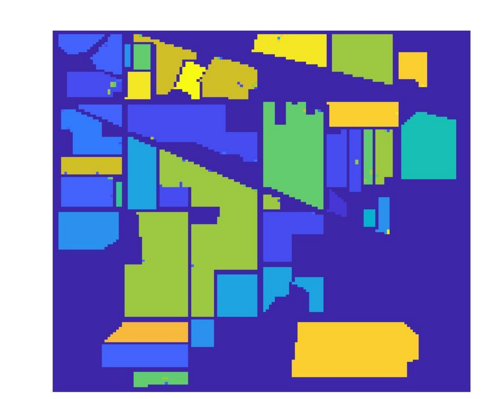
CNN perform significantly better than SVM in hyperspectral image classification tasks, and requires more resources. There are many optimization methods that can be used to further increase the accuracy. Using hyperspectral images taken by satellites and deep learning, vegetation can be successfully classified.



Ground Truth



SVM After Denoise (83.9%)



CNN After Denoise (97.4%)