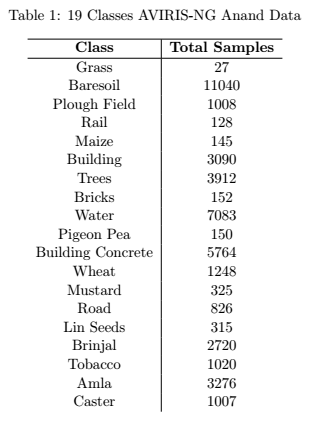
**Neural Network Architecture Code Design**

**Overview**

I will be making a design doc where I will show how the CNN algorithm performs in the case of hyperspectral datasets. This Hyperspectral Remote Sensing Dataset has big datacube and suffers from the curse of dimensionality.

The dataset of AVIRIS-NG is divided into 19 classes.



For classifying into each class, whether small or big, softmax classifier is used in the last layer.

To classify properly, the focus is on getting the data representation correct, and then to increase the complexity of the classification process by tweaking with the decision boundary over many layers.

The reason for leaving the task of feature extraction to the deep learning model is because we rarely know which features are important for the problem at hand.

**Merits of each python package:**

Keras: Easy Built-in API and easy to start

Tensorflow: Robust but difficult because of memory management

Pytorch: Easier to run computational graphs and proper dataflow in the code

The main pre-processing is to understand the feature extraction and patch extraction that can be done on the HSI datacube.

**Task 1: Implement the Keras/Pytorch model for few open datasets**

There are a lot of neural network architectures out there and developing models is essential to test out which one gives better accuracy.

Most of the data is either collected from the AVIRIS sensor or the ROSIS sensor.

**Task 2: Make a fully unsupervised NN network for the Anand HSI data**

On the frontend, command line execution using argparse library may be done for easy execution of code.

Also, to get reliable classification metrics many test runs are done and then averaged out.

The metrics were chosen to evaluate the model is learning curve (loss and accuracy), Error Image, Overall Accuracy, Kappa Coefficient, F1 score.

The test dataset is made by one-hot encoding the vector. If there is no ground truth, then the classification becomes more difficult.

**Offshoot Attempts**

Bayesian Optimization over Kernel-based methods

Composite Kernel and Fusion techniques to get better spatial spectral essence

Using TensorFlow and building it from scratch

Binary Partition Trees for getting thresholded profiles

Changing the Patch size and Window size analysis

Gabor filter + CNN

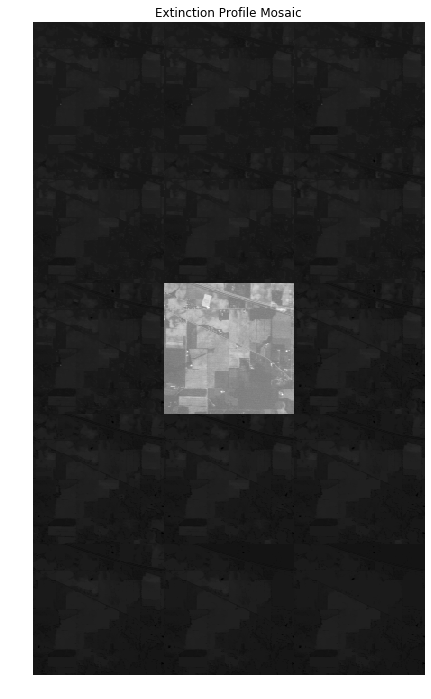
**Why implementing in Tensorflow for a Big unsupervised HSI dataset is hard?**

Tensorflow is nothing but computational graphs. Applying computational graphs to big image datasets which are also spectrally rich can be a daunting task.

Also, the tradeoff between image batch size and the number of epoch has to be addressed and implemented.

**Hypothesis 1: Does Neural Network Convolutional Feature Extraction method supersede the classical Morphological and Attribute profile approach.**

Attribute Profiles and Extinction Profiles were the state of the art method for enhancing classification in the research community during 2010-2012 for Remote Sensing Land Cover Classification Problems.

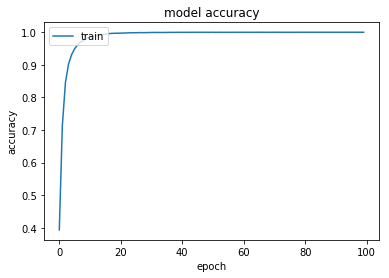
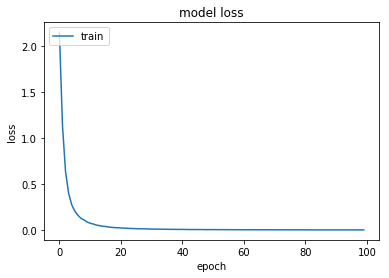


**Result 1: Indian Pines and Keras Simple CNN and 100 Epoch Training**

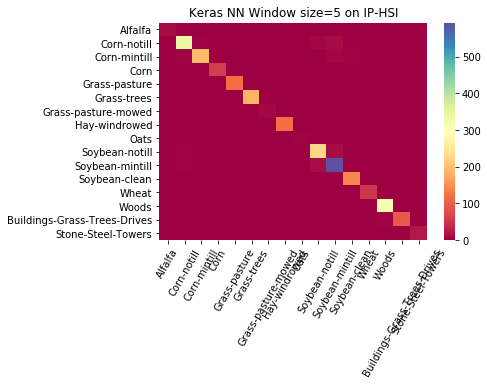
Indian Pine is a tricky Hyperspectral dataset. There are some small sized classes which have a chance of getting misclassified. The dataset is extremely small considered, the Remote Sensing data. Spatially, it is just 145x145 pixels, so to train effectively, we have to go with less sized patches. Even an 8x8 patch with Convolutional Autoencoder can yield just sub-par classification performance.

It is just plain preprocessing Keras code and standard data preprocessing to the HSI dataset, to get it ready to feed into the network.

On the window size and creating patches of 5x5x200, the model is trained on few 3x3 kernel CNN layers and later, dense and flattened layers and fed to the softmax which compares it against the ‘y’ label ground truth which is a one-hot encoded tensor. Some dropout is kept in between the network to prevent overfitting.



The overall accuracy is 97% over 100 epoch.



For dataset like Anand agriculture field, bigger patch sizes of the order 25x25 -30x30 have to be taken.

**Result 2: Keras code with different window size**

Prior to getting this result, there were many models which were not getting trained accurately because of the patch size creation leaving too many 0’s in the image. The patches are made overlapping and taken pixelwise over the entire scene.

**All Results and Documentation will be compiled here:**

<https://github.com/abunickabhi/CNN_Hyperspectral_Classification>

**Existing Open Sourced Solutions:**

<https://github.com/KGPML/Hyperspectral>

<https://github.com/jingge326/DeHIC>

DeepHyperX

Orfeo Toolbox