Spatial-Temporal Deep Neural Networks for Land Use Classification of LandSat Images

Land cover, the physical material that convers the surface of our planet, is an important form of data that is essential to global environment sustainability. Its use has many applications in the fields of environmental science, land management and geography. Because of how important land cover information is, there have been many efforts to generate large amounts of land cover datasets. But simply having large amounts of the data is not enough; the data must be processed and observed. This is generally achieved through ground surveys and/or remote sensing. Ground surveys, although very accurate, are limited by logistic constraints; making it not a feasible option to cover most of our large planet. Remote sensing, on the other hand, is the favored method because it takes advantage of cameras and sensors that are installed on satellites and observe large areas of land surface of Earth over time. This allows an efficient and affordable method to map land cover patters. Luckily, there is an abundance of publicly available and free of charge remote sensing data at medium-resolution such as Landsat imagery (USGS, 2016a). This data includes aerial photographs, satellite imagery, thermal imagery, hyper-spectral imagery, radar and lidar datasets; which vary in spatial, spectral, radiometric and temporal resolutions. Using the vast amount of data available, both visual interpretation and computer-based digital classification can be used to extract information about land cover. Generally, digital pattern classification is preferred over visual interpretation for mapping land cover in large areas. Now that there is remote sensing data with improved spectral, spatial and temporal resolutions available to the community, the real problem is to effectively and accurately use this data for image classification for land cover use.

There has been many advances with conventional methods such as maximum likelihood classifier (MLC) (Strahler, 1980) and clustering (Huang, 2002); it has now moved towards more advanced techniques such as decision trees (Xu, Watanachaturaporn, Varshney, & Arora, 2005), random forests (RF) (Pal, 2005), neural networks (NN) (Kavzoglu & Mather, 2003; Mas & Flores, 2008), support vector machines (SVM) (Gidudu, Hulley, & Marwala, 2007; Mountrakis, Im, & Ogole, 2011; Pal & Mather, 2005), convolutional neural networks (CNN) (Castelluccio, Poggi, Sansone, & Verdoliva, 2015; Romero, Gatta, & Camps-Valls, 2016), and recently, patch-based convolutional neural networks (Sharma, 2017) to preform classification on the remote sensing data. Although there has already been research done using CNNs, they have been using smaller datasets with high-resolution images. This is because classification of high-resolution images is similar to object recognition in computer vision. As a result, these CNNs have shown to work well on high-resolution remote sensing images. While this is a great achievement, the architecture is limited in that it depends on high-resolution images; which currently is hard to obtain high-resolution images that not only cover all areas of our planet, but also span a very large area. On the other hand, there is plenty of medium-resolution remote sensing data that is available from the Landsat satellite that is publicly available for free to the community. It also provides the longest continuous observations of Earth’s surface from space. The Landsat system provides copious amounts of highly calibrated, multi-spectral data of global coverage. If there exists an accurate form of classifying medium resolution imagery, this would be a large advantage as it combines the power of CNNs along with the abundance of ready-to-use medium-resolution images. There have been previous forms of using medium-resolution images, such as Sharma’s patch-based approach, however, the accuracy is only 85.60% after 149,999 iterations (Sharma, 2017). It would be ideal to have a different architecture that can achieve better accuracy. Such a system would be able to provide more reliable and efficient classification of remote sensing data over large areas.

The proposed research method involves beginning with Sharama’s patched-based CNN architecture. This is because the model has been adapted for medium-resolution multidimensional data. His architecture features five convolutional layers for feature extraction, along with a fully connected dense layer for classification. Sharama’s proposed architecture uses patched based samples by acknowledging the spatial relation of a pixel in regard to its neighboring pixels. Using this same approach, patches of size 5x5x8 will initially be extracted from the Landsat dataset as part of our training set. Because this is a supervised learning problem, to train the network, we also need to be provided labels. Luckily, The Florida Cooperative Land Cover Map (CLC), partnered with the Florida Fish and Wildlife Conservation Commission (FWC) and Florida Natural Areas Inventory (FNAI), has provided a reference map for the purpose of land cover analysis. Included in the reference map is a set of all the classes and super-classes. We can use this reference map as our labels by aligning the Landsat images with projection coordinates. Now that the dataset is ready, and the reference map has been aligned, the architecture must be created. In my attempt, I will first begin with the same architecture as Sharama because his has shown to work to up to 85.60% accuracy. Once implementing the architecture, we will use it to gather preliminary results. Afterwards, the architecture must be improved upon in attempt to increase the validation accuracy. There are many different approaches we can take in attempt to improve the accuracy. One of these includes using different size patches per class. Some classes, like a body of water, it would be ideal to use smaller patched samples. This is because \_\_\_\_\_\_\_\_\_\_\_\_\_. On other cases, like Low intensity urban, it would be better to use larger parched samples as these areas are harder to classify. Making the architecture dynamic in input size is the challenge. One idea on how to allow the architecture to support is to have the input size to be the largest size, for example 11x11x8, and using skip connections in the architecture used to skip layers in which are not necessary for some classes.