An Introduction to Multispectral Imagery and Supervised Classification with ArcGIS

*for v. 10.3 as of 12/16/2016*

*Contact: Sky Jones, 919 621 9137,* [*Sky.Jones@kci.com*](mailto:Sky.Jones@kci.com)

This tutorial provides an introduction to the concept and analysis of multispectral imagery as well as a tutorial on using multispectral imagery in conjunction with a supervised machine learning algorithm in ArcGIS to assess land cover type. Note that for all sections, the user should make sure they are naming output appropriately and saving it to where it can easily found later.

An Overview of Multispectral Imagery

Most aerial imagery we come across is collected in the visible spectrum (wavelengths of about 0.3 to 0.7 micrometers) since such images are essentially photographs. However, the unique signatures of materials in other parts of the electromagnetic spectrum can be exploited to create useful and interesting data. For example, data collected between about 10.4 to 12.5 micrometers is used to create thermal imagery, while wavelengths over a meter can be used to penetrate and map the near-subsurface.

In many cases, different materials may have similar signatures within a single portion (or *band*) of the EM spectrum, making them appear almost identical when mapped. If a scene is highly complex, there is probably no single band which can be used to accurately assess and discern all of the different types of ground cover. In such situations we use *multispectral imagery.* Though many materials may have similar signatures within one or two bands, most materials have a very unique signature across multiple bands. We can use this fact to create useful false-color maps that more accurately assess different types of groundcover, and even train algorithms to mathematically classify groundcover.

Most multispectral imagery contains at least three bands: blue (0.45-0.52μm), green (0.52-.60μm), and red (0.63-0.69μm). However, many missions collect data in even more bands. For example, the Landsat 8 mission collects data in the previously mentioned bands as well as: ultrablue, near infrared (NIR), two short wave infrared bands (SWIR), panchromatic, cirrus, and two thermal bands. For most of our analytical purposes, a dataset with blue, green, red and NIR bands is sufficient.

False Color Maps

Imagery is often distributed in the form of a raster. Each pixel in each band has an associated intensity value. When one band is displayed on its own, the image is greyscale, but because most computers use three color guns (red, green and blue) for each pixel, up to three bands can be linearly combined and displayed at once. When the bands match the color gun this gives a true color image, but different bands can be mixed to create useful false color images.

1. Open the *remote sensing sandbox.mxd* file and add in the *bloomington\_multispec.jp2* layer in the Imagery folder. This layer contains four bands: red, green, blue and NIR in that order. By default the image is true color (Figure 1).
2. Double click the *bloomington\_multispec.jp2* layer. Under Symbology, click on RGB Composite.
3. Assign Band\_4 (NIR) to the red channel, Band\_1 (red) to the green channel, and Band\_2 (green) to the blue channel. Also notice that there is an alpha channel; this controls layer transparency.
4. Click OK. The resulting image is a false color composite that emphasizes vegetation (red) and impermeable materials (cyan) (Figure 2).
5. Play around with different band combinations. Because this data is limited to just four bands, there are not any other particularly useful combinations. With more complete multispectral data, other combinations such as surface moisture, agricultural coverage, cloud penetration and land-water differentiation are possible.

Normalized Difference Vegetation Index

The Normalized Difference Vegetation Index (NDVI) is a way to quantify vegetative ground cover. The formula to calculate the NDVI value for any given pixel is

**NDVI = (NIR-RED) / (NIR+RED)**

The resulting normalized values show vegetation intensity in each pixel and account for differences in illumination, e.g., if some groundcover is in direct sunlight and other is under cloudcover. Detailed discussions of NDVI can be found online. An NDVI raster can be calculated as follows:

1. Go to add *bloomington\_multispec.jp2* to your file, but double click it instead of adding it.
2. Individually add bands 1 (red) and 4 (NIR) to your map.
3. Open the raster calculator and enter the following expression: Float("bloomington\_multispec.jp2 - Band\_4"-"bloomington\_multispec.jp2 - Band\_1")/Float("bloomington\_multispec.jp2 - Band\_4"+"bloomington\_multispec.jp2 - Band\_1")
4. Run the tool.
5. The resulting raster shows the vegetative intensity at each pixel on a range of [-1,1] (Figure 3). More vegetated areas are more positive. Note that by selecting an arbitrary threshold you can create a binary vegetated vs. unvegetated layer. This can be done by using the raster calculator and entering “your\_ndvi\_layer” >= *[your threshold]*. For this study area I picked a threshold of 0 (Figure 4). For urban areas with little bare ground or water, this is also a good estimate of permeable vs. impermeable surfaces.
6. Note that there also exists a NDWI – a normalized *water* index, calculated using NIR and SWIR data and used to assess plant water stress.

Supervised Classification in Arc

False color maps are useful for qualitative assessments of a study area, but they don’t provide any quantitative information. Normalized indices can be used to break a study area into two classes (for example, vegetation and not-vegetation) based on some predetermined cutoff, but what if you need to break the study area into three or more classes? Using multispectral imagery we can train a machine learning algorithm to recognize the unique spectral signature of different groundcover types and then classify the remaining portions of the image for us. While this is a complex statistical technique, it is already implemented for us in Arc. We still need to train the algorithm, which is why the algorithm is called “supervised”. Arc has also implemented an unsupervised algorithm that automatically identifies potential classes. This is can useful for helping how to decide to train a supervised algorithm, but it is not discussed here.

1. Open the *bloomington\_multispec.jp2* layer again.
2. In the main toolbar, go to Customize>Toolbars and turn on the Image Classification toolbar.
3. The *bloomington\_multispec* layer should be automatically selected in the toolbar. If it isn’t make sure you’re working in the correct dataframe and select it.
4. Click on Draw Polygon, and then Training Sample Manager.
5. Begin drawing polygons on a single type of ground cover, e.g., vegetated fields. In general, the more area your samples cover the better. A good rule of thumb is about 5-10% of the image should be used to train your algorithm total.
6. Once you are done drawing polygons of a single type, highlight them all in the Training Sample Manager and select Merge training samples. You can rename the class if desired.
7. Repeat for as many classes as you’d like. However, because of their unique spectral signatures it is recommended that you train the algorithm to identify water, cloud cover and shadows if they have a significant presence in the image. In this case, I made classes for water, scrub/fields, pavement, forest, dirt/sand and building tops (Figure 5).
8. Once you are done training, save your Training Sample Manager as a shapefile.
9. Open the Train Support Vector Machine Classifier tool. Use the *bloomington\_multispec* layer and your training shapefile and run it.
10. Open the Classify Raster tool. Use the *bloomington\_multispec* layer and the output from the Train Support Vector Machine Classifer tool. Select okay, and allow the tool to run. This may take several minutes.
11. The result is a classification raster (Figure 6). Depending on how well you trained it, the result may be good or not so good. Selecting your training samples is an art as much as anything. Also note that the extent for this study area is quite large give the complexity of it (small residential roads with houses, etc.). Limiting extent of the study or coarsening the classification (e.g., urban instead of roads and building tops) can potentially improve results. An advanced technique called segmentation may also be used to improve results on complex study areas. While not discussed here, segmentation essentially allows the algorithm to classify the image in parts.
12. If the output is acceptable, the raster can be manipulated further, like converting it to a shapefile, combining classes (rooftops + roads= impervious surfaces) and more.
13. Note that the algorithm uses all available bands to classify to image. This means that having more bands in your raster will typically provide more accurate results

Using Landsat Data

Landsat data can be acquired through the USGS Earth Explorer. It can be downloaded as georeferenced TIFFs (GeoTIFF) and imported into ArcMap. However, the bands are not packaged together.

1. In ArcMap, open the Composite Bands tool.
2. Select the seven Landsat 5 bands in the Imagery Folder (half\_dome\_mono\_B*X*) and import them as a composite band.
3. The bands in this imagery are as follows: B1 – Blue, B2 – Green, B3 – Red, B4 – NIR, B5 – SWIR1 (shorter wavelength), B6 – Thermal, B7 – SWIR2 (longer wavelength). If you want a true color image, you will need to rearrange the bands.
4. Play around with different band combinations to see how they can be used to emphasize different scenery. See <http://web.pdx.edu/~emch/ip1/bandcombinations.html> for ideas.

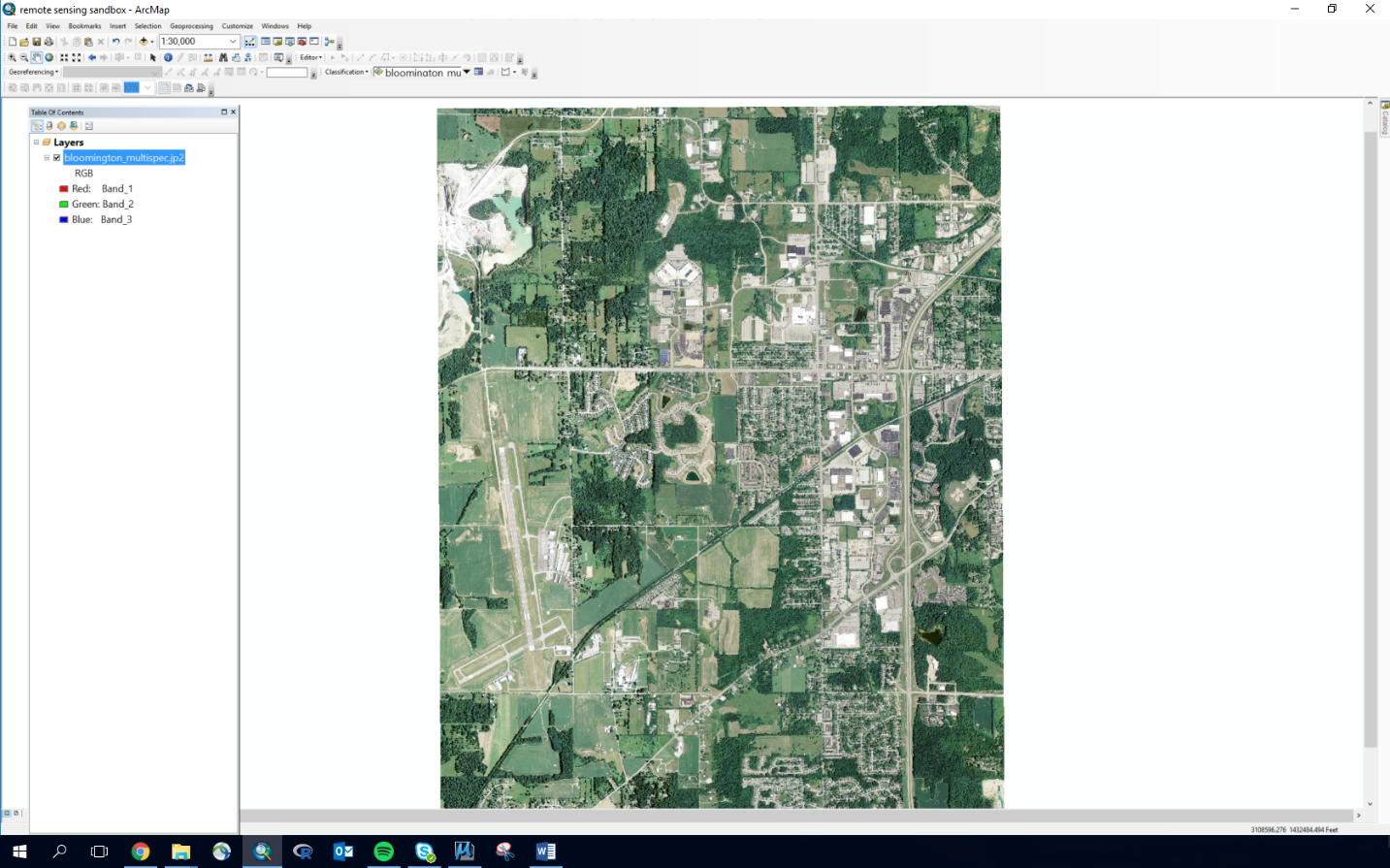


Figure 1. Aerial imagery of Bloomington, IN.

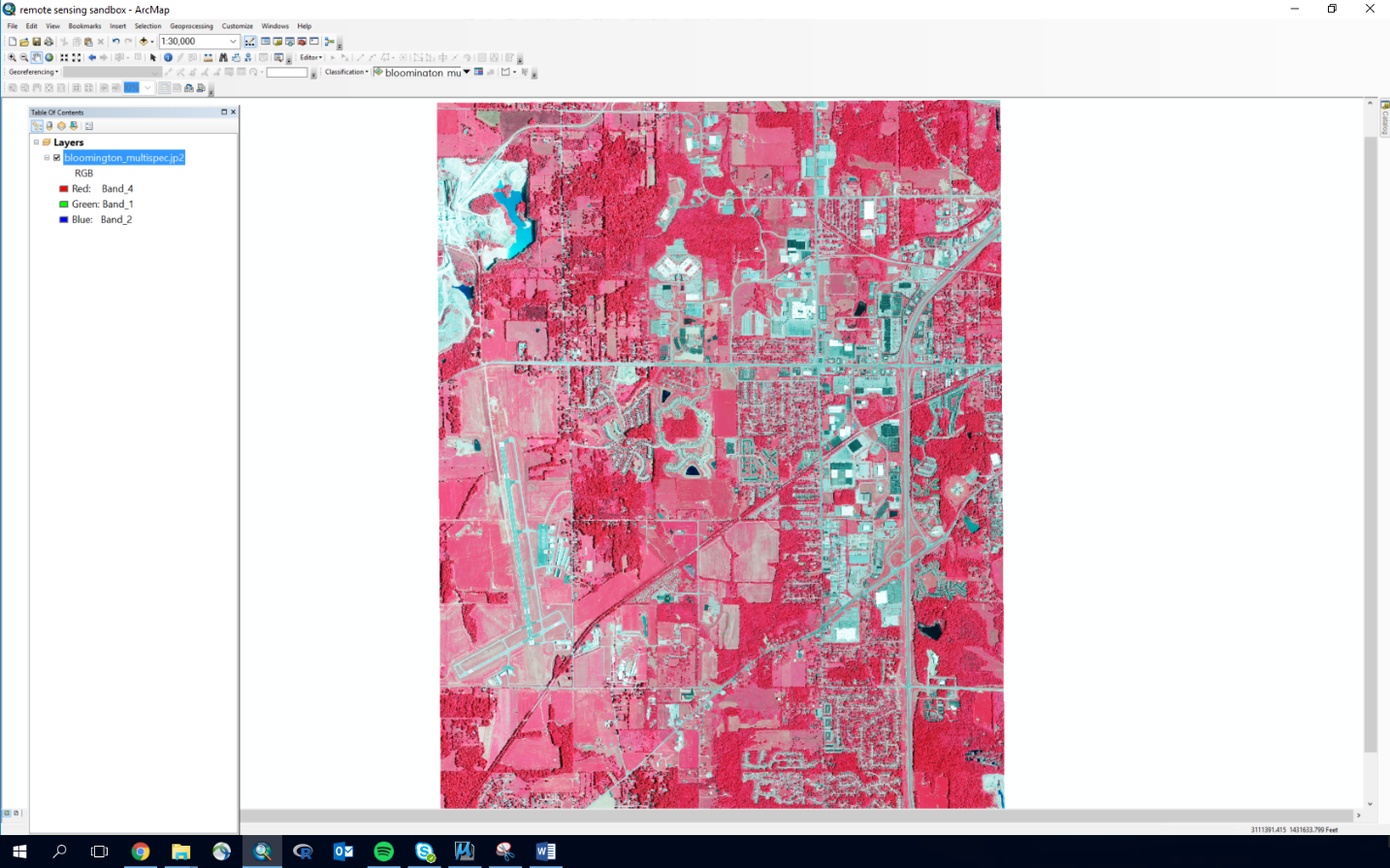


Figure 2. False color imagery emphasizing vegetation (red) and impervious cover (cyan).

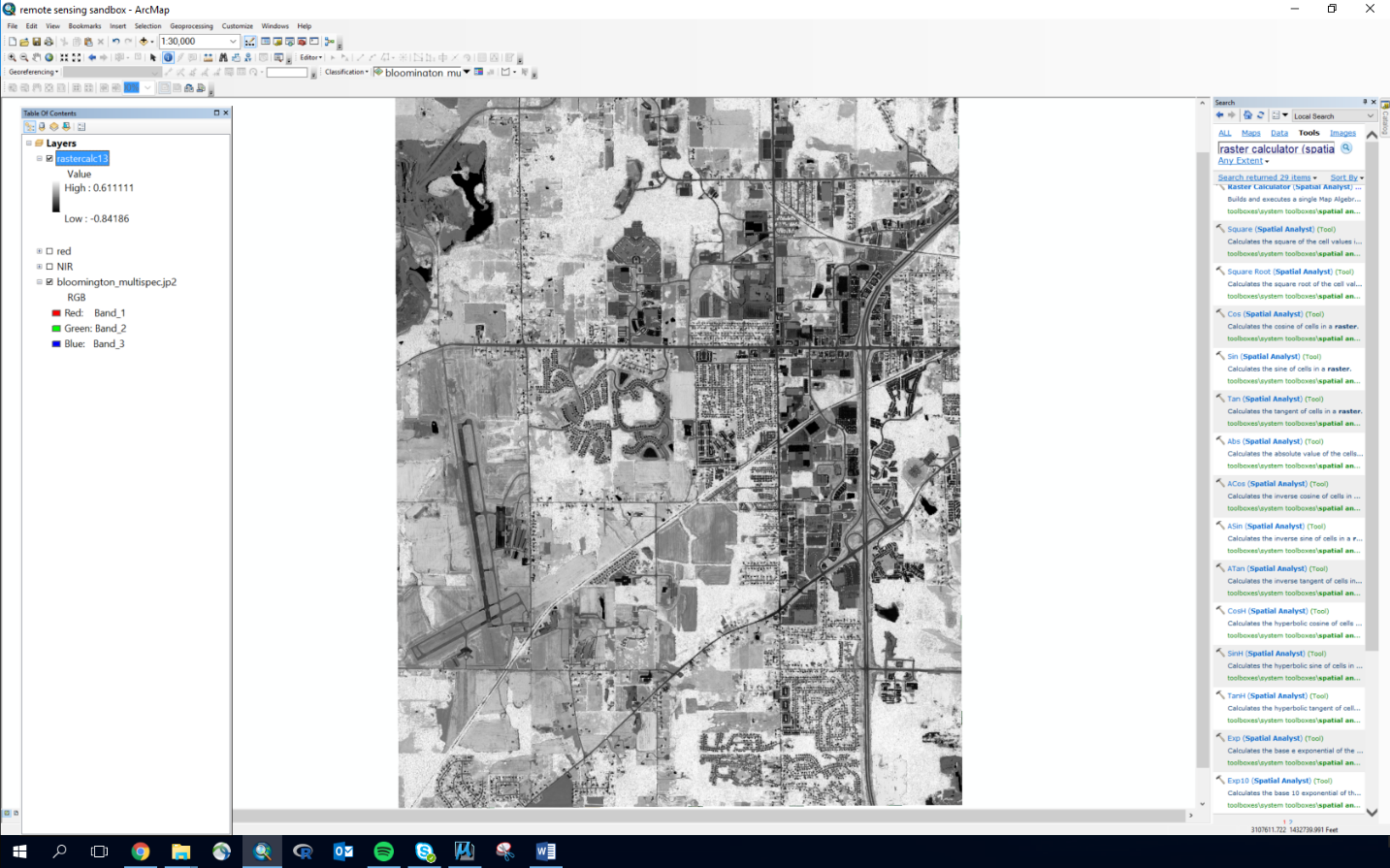


Figure 3. NDVI for Bloomington.

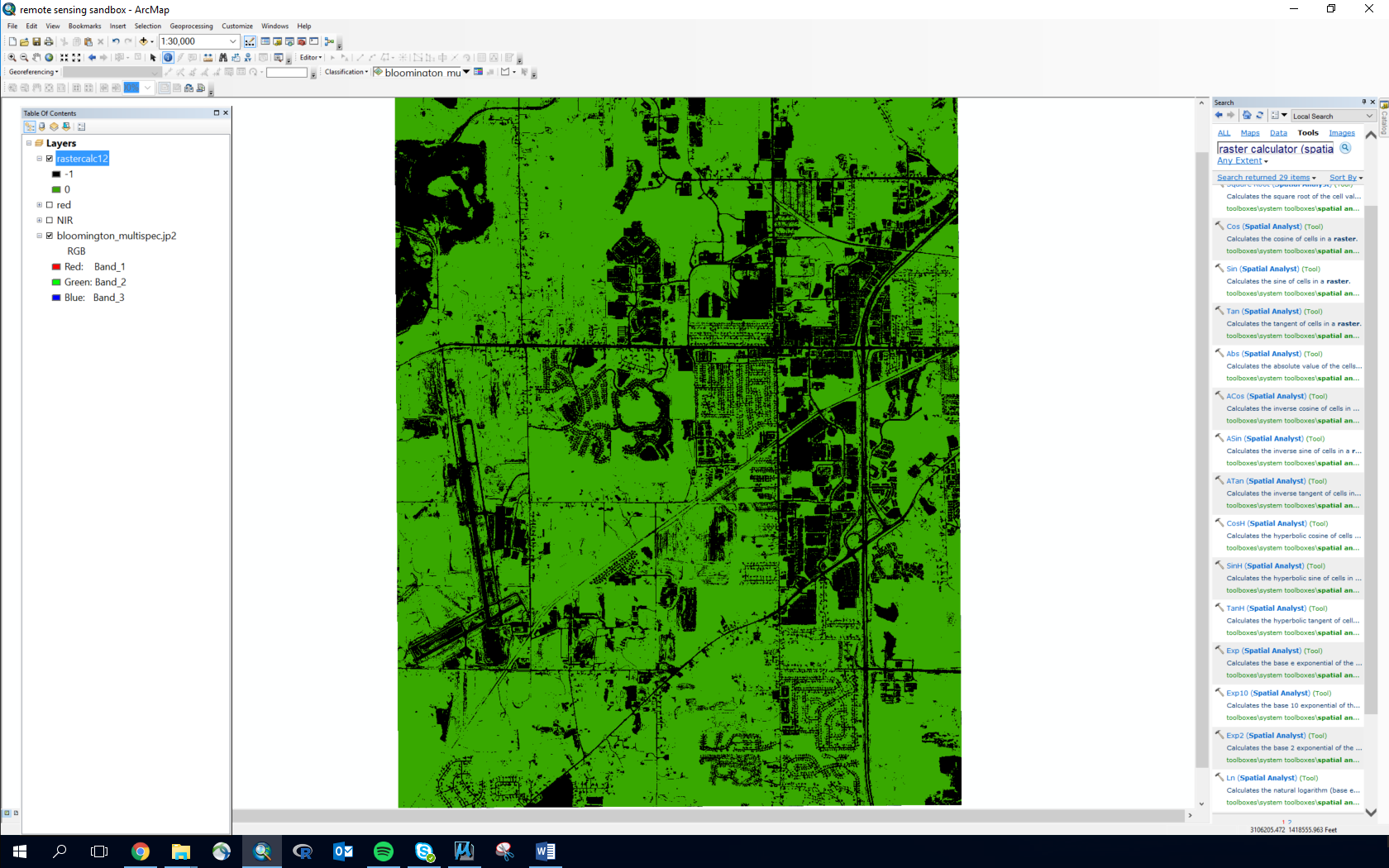


Figure 4. Logical/binary NDVI for Bloomington. Vegetation in green, all other classes in black.

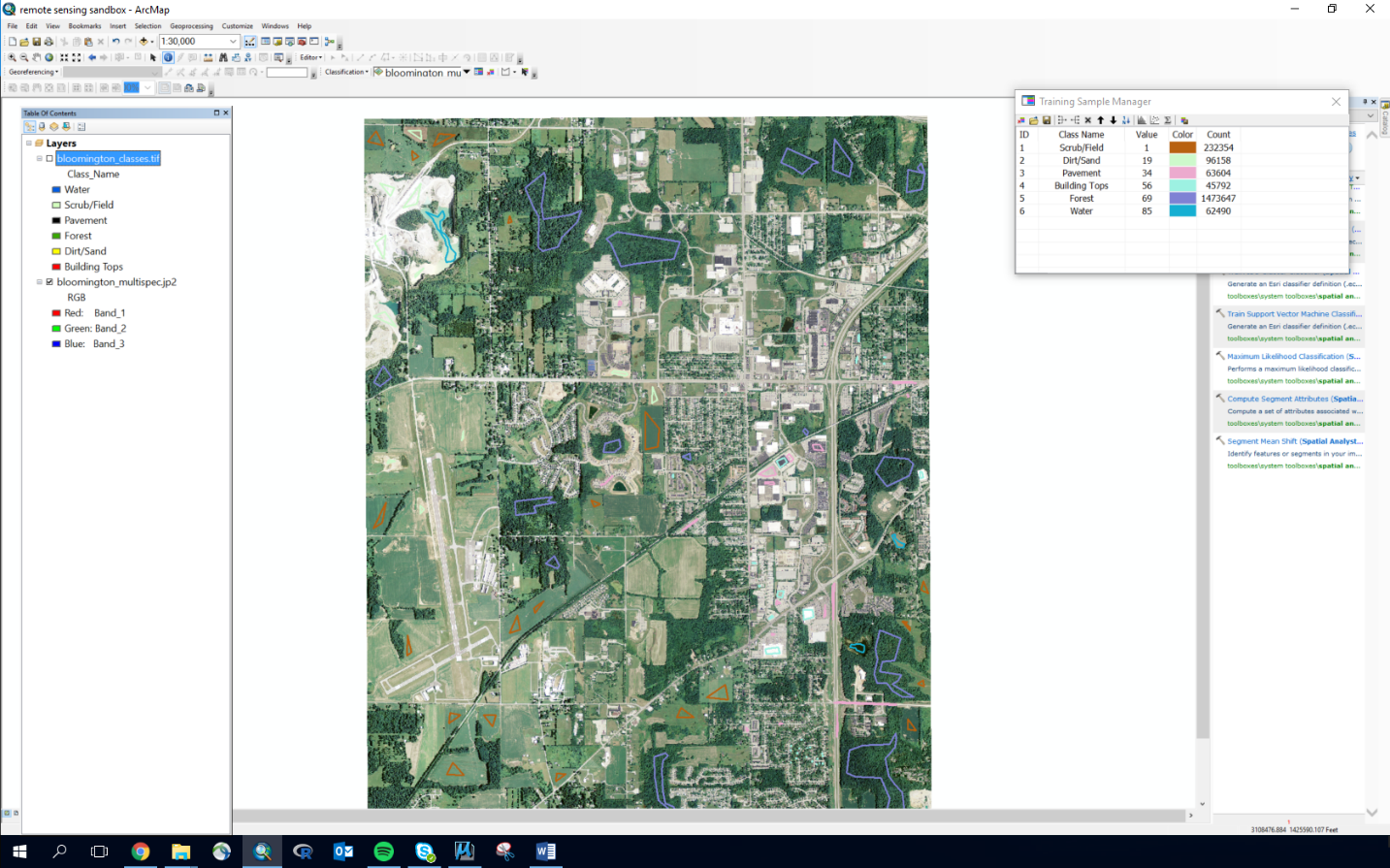


Figure 5. Image showing training classes and polygons used to create Figure 6.

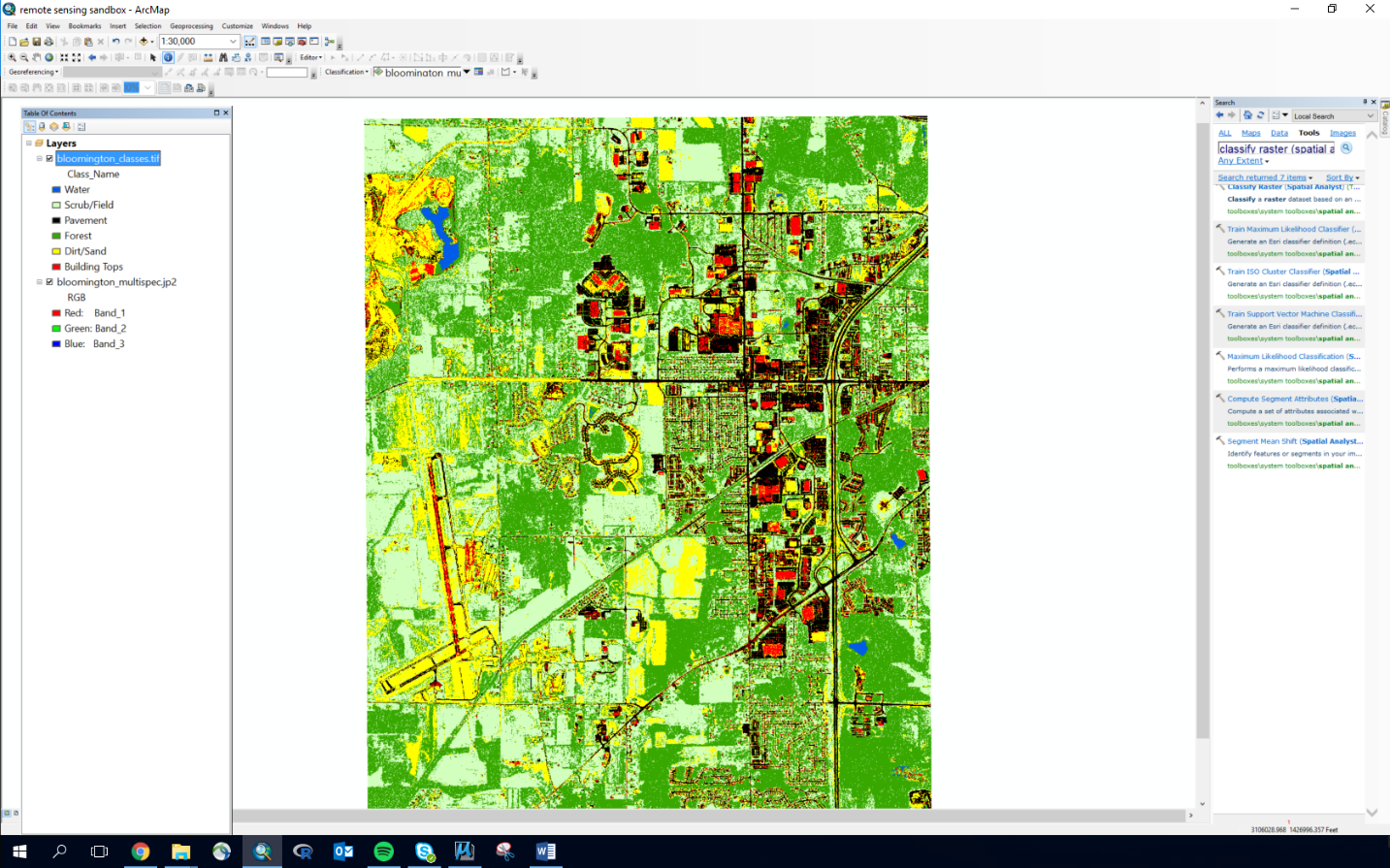


Figure 6. Output raster from a supervised training algorithm. Note the minor misclassifications.