**Supplemental Video 1 Script**

**0:00**: My name is Megan Coffer. I’m an ORISE postdoctoral fellow at the U.S. EPA Office of Research and Development.

**0:07**: The intent of these instructional videos is to visualize the methods outlined in the accompanying manuscript. While the manuscript classified seagrass at eleven coastal sites in the United States, these instructional videos will focus on a single study area at Back Sound, North Carolina, for which a satellite image and reference data are included in the supplemental material.

**0:25**: This first instructional video will show users how to select and download satellite data using web-based data archives.

**0:30**: In this study, we used imagery from Maxar’s WorldView-2 and WorldView-3 satellite platforms. Disclaimer: there is no endorsement of commercial products even if verbal. WorldView-2 and -3 imagery can be obtained through image archives, which have occasional satellite images collected over aquatic systems throughout the lifespan of each sensor.

**0:47**: To access this imagery, we’ll use two websites. The first is used to find imagery matching our selection criteria and the second is used to actually request and download the data.

**0:56**: First, we’ll navigate first to Maxar’s Archive Search & Discovery site at discover.maxar.com. To define our area of interest, we’ll zoom in to Back Sound and select the “Draw a rectangle” tool.

**1:08**: This will return several results, but we still want to apply additional filters before selecting an image. Under filters, we’ll select just “8-bands,” set our start and end date as the year our reference data was collected, our off nadir angle as less than 40°, and our sun elevation as greater than 20°. Finally, we’ll select “no automatic selection” and refresh our results.

**1:38**: This pairs down our list quite a bit and we’ll notice that imagery was collected in March 2013 which coincides with when our reference data was collected. One of them was collected on the 29th of March, but if we click on this image, we can see that there’s a good bit of cloud cover. The image collected on the 27th, however, seems to be pretty clear and we have a good view of the water. This middle image seems to be the one that contains the bulk of our reference data, so we’ll request this one. If we click on the plus sign, we can see additional metadata including the image ID which we’ll need for requesting our image, so we’ll go ahead and copy this to our clipboard.

**2:16**: In another tab, we’ll navigate to Maxar’s Global Enhanced GEOINT Delivery service at evwhs.digitalglobe.com/myDigitalGlobe/login. Once registered, login, and enter the Image ID from Maxar’s Archive Search & Discovery site in the search bar in the upper left and press enter.

**2:37**: You’ll notice our image appears in the bottom left of your screen and if you click on these three horizontal bars, it gives you the option of adding to cart which will create a new window with several parameters.

**2:48**: The first is “Order Name.” WorldView-2 and -3 Image ID’s are arbitrary; therefore, I’ll generate a more intuitive name for my records and name this Back Sound as well as the date the image was collected. We’ll also want to change our “Product Type” to “Basic All Bands” and “Clip Features to AOI” to “No.” We can add our image to cart, go to cart, and then submit our order.

**3:19**: You’ll receive three emails while your order is being processed: the first will state your archive order has been accepted, the second will state it is now in production, and the third will state it has been fulfilled. The entire process can take up to a day.

**3:31**: Once you receive the last email stating your archive order has been fulfilled, log back into Maxar’s Global Enhanced GEOINT Delivery service and click “My Imagery” and then “Library.”

**3:48**: This second order is an older order that I pre-requested for this video, and the status says completed so I can go ahead and download it.

**3:55**: Toggle the box to the left and then click “Download” which then gives you several options. I’m going to click this FTP option which will then provide server information for where the image is stored.

**4:08**: Then, in an FTP client such as FileZilla, you can login using your Maxar Global Enhanced GEOINT Delivery service username and password to download the zip file onto your local computer.

**Supplemental Video 2 Script**

**0:00**: This second instructional video will show users how to process satellite data from basic imagery to an analysis-ready product, using open-source Python programming.

**0:11**: Included in the supplemental material is a folder called WorldView seagrass which includes three folders, one of which is called Input data and one of which is called Python library.

**0:21**: The Input data folder contains two subfolders, one for image processing and one for image classification, which will be shown in the next instructional video. In the processing data folder, there’s a single subfolder, called Zipped data which contains a spatial subset of the satellite image requested in Supplemental Video 1.

**0:40**: The Python library folder contains the necessary scripts for image processing and classification, as well as the scripts associated metadata and documents specifying package requirements.

**0:51**: We’ll first open the Image processing Python file, which I’m editing here in Visual Studio Code. In this instructional video, I’ll walk through each line of the code but won’t actually run each step due to the processing time some steps can take to complete.

**1:05**: First, in lines 1-5, we import the required Python libraries.

**1:10**: Then, in lines 7-15, we’ll define all necessary input variables. These should be the only lines of this script that need to be changed by the user.

**1:19**: The first variable we will define is the name of the satellite image to be processed on line 9, which should match exactly with the name of the zipped satellite file that we have in Zipped data folder. So here, it’s going to be set to Back Sound as well as the date the satellite image was acquired.

**1:35**: Next, on line 11, we’ll set the coordinates of our area of interest to avoid processing unnecessary data. This extent should be specified in decimal degrees as the upper left longitude, upper left latitude, lower right longitude, and lower right latitude, in that order.

**1:51**: On line 13, we’ll define the EPSG code for our area of interest. Spatialreference.org has information on all valid spatial reference systems, and websites such as projfinder.com can be useful to determine an EPSG code from a set of coordinates.

**2:08**: On line 15, we’ll define the working directory where our WorldView seagrass folder is located.

**2:15**: Lines 17-18 install required libraries using the requirements text file located in the Python library folder.

**2:24**: Lines 20-23 import another file located in this folder called seagrass lib which is a Python file containing all the functions used in this script and the classification script. The seagrass lib metadata word document also contains information on each of these functions including required inputs, optional inputs, outputs, and example code.

**2:45**: In lines 25-27, we increase the cache size to avoid memory constraints.

**2:51**: And finally, in lines 29-31, we actually start getting into the image processing. First, we list all multispectral image files from the zipped data folder and, in lines 34-36, we unzip just these image files and the associated image metadata files from the zipped data folder.

**3:09**: You’ll notice in your file explorer that this creates a new folder called Unzipped data with a subfolder for the image currently being processed. Similarly, for each of the following processing steps, a new folder will be created in this Processing data folder named to describe each respective processing step.

**3:28**: Each image can contain multiple image tiles which are generated arbitrarily by Maxar to aid in data storage and transfer. The remainder of the script will loop through all tiles in a single image to apply each processing step. Note, the satellite image included with this manuscript contains only a single image tile.

**3:46**: In lines 38-44, we loop through each multispectral image tile contained in our image and apply radiometric calibration which converts digital numbers provided in the raw satellite data to reflectance values.

**4:01**: In lines 46-52, we apply atmospheric correction using dark object subtraction to remove the influence of the atmosphere and convert our data to remote sensing reflectance.

**4:12**: Next, in lines 55-59, we project our image using the ESPG code defined at the beginning of the script.

**4:20**: And, finally, as our last step in image processing, in lines 61-66, we clip our image using the input coordinates also defined at the beginning of the script.

**4:31**: This clipped image is then what is classified in the Image classification Python script which is shown in the next instructional video.

**Supplemental Video 3 Script**

**0:00**: This third instructional video will show users how to define input data for our machine learning classifier using ArcGIS Pro and how to apply this machine learning classifier using open-source Python programming.

**0:14**: First, in ArcGIS Pro, we’ll open the fully processed image file from the previous instructional video. For this example, we will generate polygons of known input data, called regions of interest or ROIs, for 6 classes.

**0:28**: To generate our ROIs, we’ll first select the Imagery tab, and then under classification tools, we’ll select the training samples manager. We will create a new schema, and then right click the schema to add a new class. We’ll name our first class “Seagrass” and start indexing our values a 0. The next class will have a value of 1, the class after that will have a value of 2, and so on. We can then click “Okay” and then we can use this rectangle tool to draw a polygon within our image that contains entirely the class we are planning to classify. We continue to generate a few polygons for each class and when we’re done, we’ll click “Save As” and then export this new schema as a shapefile by specifying the .shp extension.

**1:18**: When we’re done, our ROIs will look something like this shapefile that has ROIs for each of the classes we intend to classify.

**1:26**: Along with this instructional video is a shapefile specifying ROIs for each of these classes which is included for brevity.

**1:34**: Once we’ve generated polygons ROIs for each of our classes, we’ll navigate to the “Python\_library” folder and open the image classification Python script in Visual Studio Code. Just like in the previous instructional video, I’ll walk through each line of the code but won’t actually run each step due to the processing time some steps can take to complete.

**1:52**: First, in lines 1-4, we’ll import the required Python libraries.

**1:56**: Then, in lines 6-12, we’ll define all necessary input variables. These should be the only lines of the script that need to be changed by the user.

**2:06**: First, in line 8, we’ll specify the name of the processed satellite tile which we created in the previous instructional video. While the image processing script processed all tiles within a given satellite scene, the image classification script processes only a single tile at a time. However, if it’s of interest, a user can easily mosaic all tiles within a scene to generate a single image before generating ROIs and running the image classification.

**2:31**: In line 10, we’ll define the name of the satellite image to be processed which, just like the last instructional video, should match exactly the name of the zip file we downloaded in Supplemental Video 1 but without the .zip extension.

**2:44**: Finally, in line 12, we’ll define the working directory where the WorldView seagrass folder is located.

**2:50**: Lines 14-15 install required libraries using the requirements text file located in the Python library folder, and lines 17-20 import the seagrass library Python file which contains all necessary functions.

**3:04**: Next, we increase the cache size to avoid memory constraints in lines 22-24.

**3:10**: In lines 26-29, we read in the ROIs shapefile we created at the beginning of this video and prepare them for the classification function.

**3:18**: In lines 31-35, we extract the spectral information contained within each ROI to use as known information for the deep convolutional neural network classifier.

**3:29**: In lines 37-41, we use this extracted information to train the deep convolutional neural network, and in lines 43-47 we classify the image on a pixel-by-pixel basis.

**3:42**: The classification is then exported as a single layer tif file which can be opened and viewed in programs such as ArcGIS Pro, and it’s saved in our WorldView seagrass folder under classification data and then classified image.