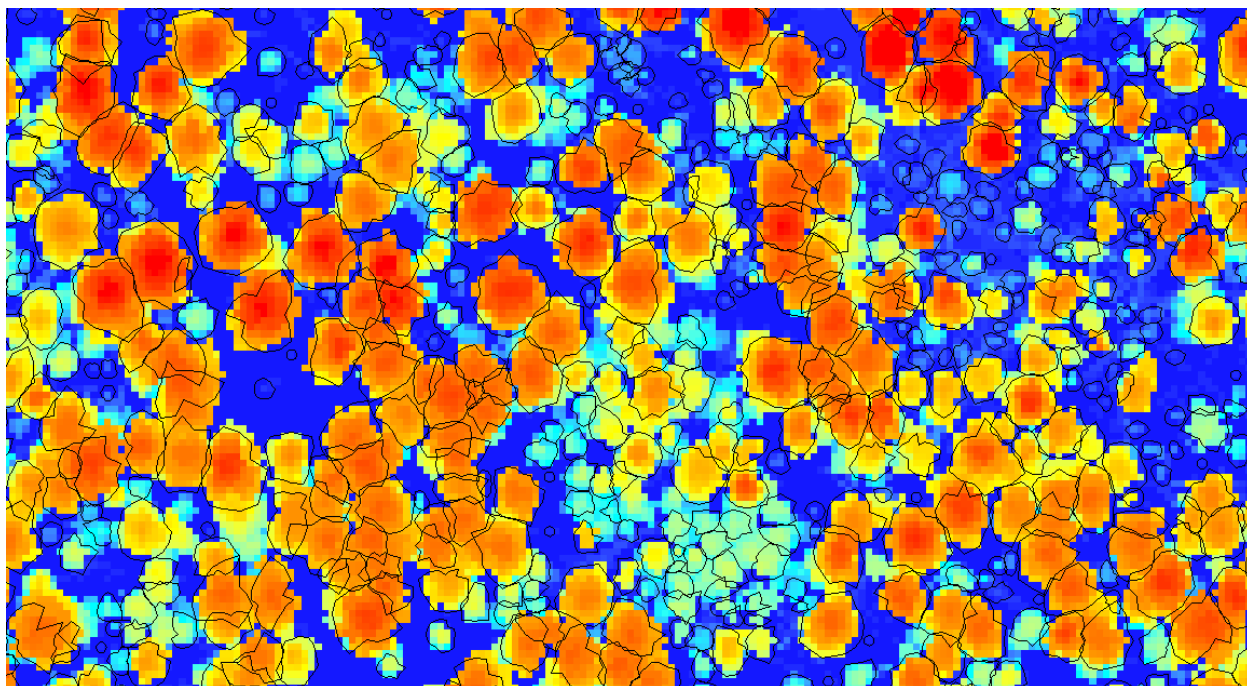


EXERCISE 5

Tree Segmentation



Introduction

The TreeSeg utility in FUSION operates on a canopy height model (raster derivative for which each pixel stores the height of the tree canopy) to produce individual tree segments. These segments can represent individual trees (in areas where the canopy is open) and/or clusters of trees (in areas where the canopy is more closed). While TreeSeg is very powerful, it must be run via command line, and formatting data for processing can be tedious. In this exercise, you will learn how to use a custom Python script to automate the process of performing tree segmentation on a CHM of any size.

Objectives

- Orient yourself to the workflow outlined in the Python script
- Set up your workspace for processing
- Edit the necessary variables and run the script

Required Data

- **Tree_Segmentation.py** – This script will automate a number of processing tasks, including some clipping and data conversions in ArcGIS, data conversions in FUSION, and the final tree segmentation operation in FUSION.



- **CHM_filled_not_smoothed_1p0METERS.tif**—Output from Area Processor that is available in the products folder or in the backup folder. This has been converted from DTM to TIFF for you.

Prerequisites

- You have ArcGIS 10.7.x installed on your machine, which comes with Python 2.7 (the script will not run with older versions of ArcGIS – ArcGIS Pro has a different type of Python).
- Completion of Exercises 1-4



USDA Non-Discrimination Statement

In accordance with Federal civil rights law and U.S. Department of Agriculture (USDA) civil rights regulations and policies, the USDA, its Agencies, offices, and employees, and institutions participating in or administering USDA programs are prohibited from discriminating based on race, color, national origin, religion, sex, gender identity (including gender expression), sexual orientation, disability, age, marital status, family/parental status, income derived from a public assistance program, political beliefs, or reprisal or retaliation for prior civil rights activity, in any program or activity conducted or funded by USDA (not all bases apply to all programs). Remedies and complaint filing deadlines vary by program or incident.

Persons with disabilities who require alternative means of communication for program information (e.g., Braille, large print, audiotape, American Sign Language, etc.) should contact the responsible Agency or USDA's TARGET Center at (202) 720-2600 (voice and TTY) or contact USDA through the Federal Relay Service at (800) 877-8339. Additionally, program information may be made available in languages other than English.

To file a program discrimination complaint, complete the USDA Program Discrimination Complaint Form, AD-3027, found online at [How to File a Program Discrimination Complaint](#) and at any USDA office or write a letter addressed to USDA and provide in the letter all of the information requested in the form. To request a copy of the complaint form, call (866) 632-9992. Submit your completed form or letter to USDA by: (1) mail: U.S. Department of Agriculture, Office of the Assistant Secretary for Civil Rights, 1400 Independence Avenue, SW, Washington, D.C. 20250-9410; (2) fax: (202) 690-7442; or (3) email: program.intake@usda.gov.

USDA is an equal opportunity provider, employer, and lender.



Table of Contents

Part 1: Understanding TreeSeg.....	5
Part 2: Understanding the full tree segmentation workflow	6
Part 3: Edit and run the Python script	8
Part 4: Examine outputs in ArcMap.....	13

Part 1: Understanding TreeSeg

A. Open the FUSION manual

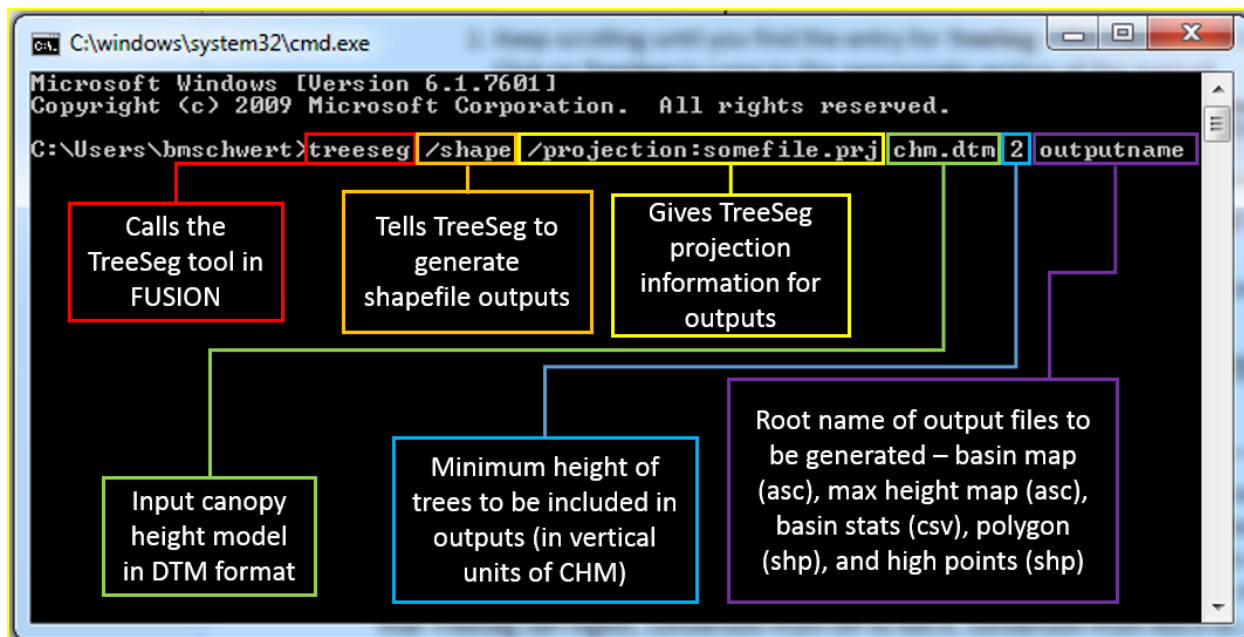
1. Navigate to the FUSION folder on your C drive (**C:\FUSION**) and open the **doc** folder, which contains the software documentation.
2. Double-click on **FUSION_manual.pdf** to launch to FUSION manual.

B. Navigate to the TreeSeg documentation

1. Scroll down to the **Table of Contents** and find the section on **Command Line Utility and Processing Programs** – these are all tools found outside of the FUSION graphical user interface (GUI) and can only be called via command line.
2. Keep scrolling until you find the entry for **TreeSeg** – the manual lists this tool on **page 138**. Click on **TreeSeg** to jump to the appropriate section of the manual.

Note: you can also press **CTRL + F** on your keyboard, type in **treeseg**, and then hit **enter** to find references to TreeSeg in the manual.

3. Read through the **Overview** section to gain insight into how TreeSeg works and the types of outputs that it generates.
4. Read through the **Syntax** section to learn how to format a TreeSeg command. An annotated screen capture of the TreeSeg command in a CMD window is provided below.



Note: we will only be using two switches, or optional parameters: **shape** and **projection**. The **shape** switch creates shapefile outputs, and the **projection** switch will assign projection information to all outputs so that we don't have to manually define the projections of tens or hundreds of files.

Part 2: Understanding the full tree segmentation workflow

While CHMs are widely available in either .tif or .img format, we learned in the previous section that the TreeSeg tool within FUSION requires input data to be formatted as .dtm files, or digital terrain models. This is a commonly used format in FUSION, but it is not widely used in standard desktop GIS software packages. In order to successfully generate tree segments, a number of steps are required of the user, including: tiling the CHM into rasters of a size that TreeSeg can ingest, conversion from tiff to ASCII, conversion from ASCII to digital terrain model, and finally tree segmentation.

While this workflow is simple enough for a user to initiate manually, having to do so increases the possibility of user error. Furthermore, the user would have to iterate through the steps of the workflow for each tile in the project, and depending on the size of the input CHM, there can be tens or hundreds of raster tiles to process. In order to automate the production of tree segments, we have created a Python script to handle both the ArcGIS and FUSION processing steps.

A. Open the Python script

1. From the **Start** menu, select **All Programs, ArcGIS, Python 2.7**, and then **IDLE (Python GUI)**.

Note: If you're unfamiliar with Python GUIs, the window that opens is a Python shell, which is basically a command prompt for Python. You can type commands directly into this prompt. However, we will be opening a completed script.

2. From the **File** menu, select **Open**.
3. Navigate to the directory where you've stored the Python script, **Tree_Segmentation.py**. Select it and click **Open with IDLE**.

B. Look over the first sections of the script

If you're unfamiliar with Python scripts, don't be afraid – you don't need to be able to read or write Python code to understand what this script is doing. Lines that begin with a pound symbol or hashtag (#) indicate that the line contains a comment.

1. Read through the comments at the top of the script, which include the name, author, description, and script requirements. Before proceeding, it's important that you verify you meet the requirements.

```
# Name: Tree_Segmentation.py
#####
# Written by Brenna Schwert, bmschwert@fs.fed.us
# RedCastle Resources Inc., GTAC USFS
#####
# Description:
# Finds all TIFs in a given folder and converts
# them from TIF to ASCII format, then from ASCII to
# DTM, and then runs tree segmentation. When running tree
# segmentation, the script will identify the number
# of available processors in your machine and use one
# less than that number for processing.
#####
# Requirements:
# ArcGIS 10.3
# FUSION v3.6 (with FUSION added to your system's
#               environment variables)
#####
```

2. Read through the next section of comments, the **DEFINE VARIABLES** section. This is where you will make changes to the script and customize it to your data (you will do this in the next part of the exercise).
3. You will notice that after the DEFINE VARIABLES section, there is a comment block that warns the user not to edit anything past that point – this marks the end of your interaction with the script, but if you're curious, keep scrolling to learn more. If you don't care about the particulars of the script, **feel free to skip ahead to Part 3**.

C. Look over the latter sections of the script

1. SET UP ENVIRONMENT (line 47)

Hint: Press **Alt + G** on your keyboard to bring up the **Go to line number** prompt.

- i. Import system modules
 - (a) Python can do a lot, but it doesn't do everything natively. There are some tools that need to be imported in order to take advantage of them, like `arcpy`. Importing `arcpy` allows us to manipulate geospatial data in Python the same way that we can using tools in ArcGIS. Here, we import all of the tools necessary to accomplish our workflow.
- ii. Get time stamps
 - (a) This section will print time stamps to our console. This is very handy – it lets us know how long a script has been running and/or how long it took to complete the run.
- iii. Organize working directory
 - (a) The script will create a few subfolders in your working directory so that your workspace stays neat and tidy, including: **tif**, **asc**, **dtm**, and **treeseq**.
2. PREP RASTERS (line 78)
 - i. There are three main steps that need to be accomplished in this workflow, including two data conversions and tree segmentation. While processing these data isn't complicated, when running TreeSeg, we may run into tile size limits. In order to mitigate any potential errors, we need to check the size of our input data and tile them into smaller rasters if they exceed a certain size before we begin our workflow. The script will find all rasters in

the working directory and check the file size and properties of these data. If the data are too large, they will be tiled into smaller rasters.

3. CONVERT TO ASCII (line 137)

- i. After identifying and/or creating all eligible .tif tiles, the script will use arcpy to convert them to ASCII format. In this section, the script also does some prep work for the next sections (like pulling the number of processors available in your machine).

4. CONVERT TO DTM (line 185)

- i. In this section, the script will use FUSION to convert your ASCII files to .dtm format, which is the format that TreeSeg needs. Recall that FUSION operates via command line, and it cannot be called via Python the way processes in Arc can. The Python script will write a .bat (batch) file for this data conversion – a .bat file is basically a text file where each line represents a unique command to be called in a CMD window. In this case, the .bat file contains a line/command for each raster being converted from .tif to .asc. Once written, the script will then launch the .bat file. **When you run the script and it gets to this section, you will see a CMD window appear – do not close this window.** If you close the window, you will terminate the process, and data will not be converted.

5. TREE SEGMENTATION (line 199)

- i. In this section, the script will create a .bat file for TreeSeg. Again, you will have one line of code for every raster you wish to process. While each iteration of the DTM conversion takes a minute or two to run, each iteration of TreeSeg takes 5-10 minutes to run. In order to make better use of our computer's resources, the script will divide the processing among the available cores in your machine. If you have X cores available, your machine will take advantage of X-1 cores, which will result in X-1 CMD windows popping up on your screen. For example, if you have four available cores, the script will ensure that you use three of those cores, and as a result, three CMD windows will open. As with the DTM conversion process, **do not close these CMD windows, else you will terminate processing.**

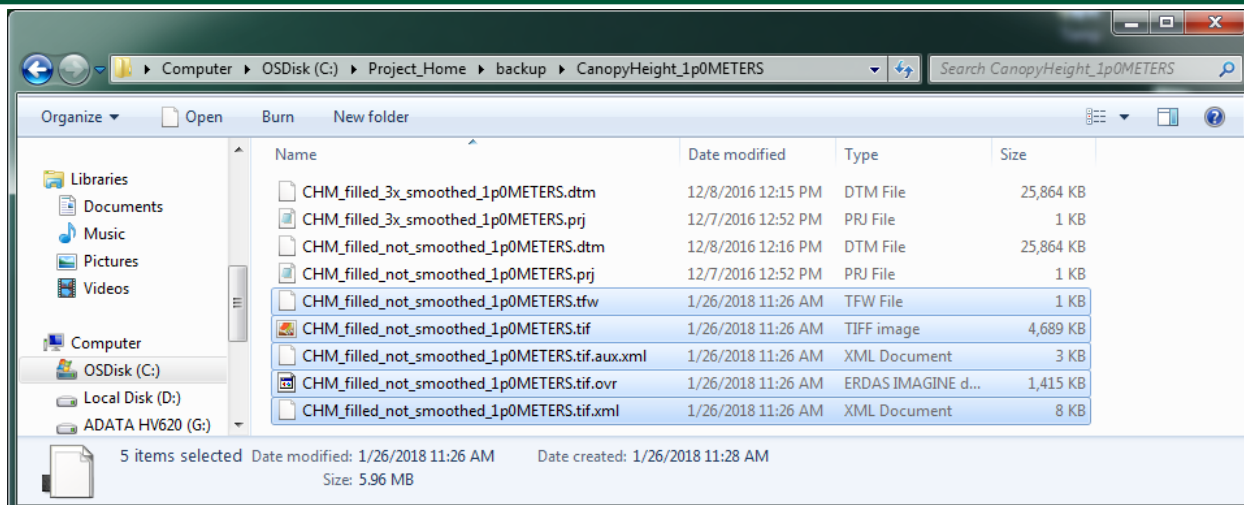
6. CLEAN UP ENVIRONMENT (line 246)

- i. The script will create a simple ReadMe file explaining what the folder contents are. It will also delete the .bat files, because they are no longer necessary. Finally, it will print the last time stamp and total runtime of the script.

Part 3: Edit and run the Python script

A. Create New Folder and Copy CHM

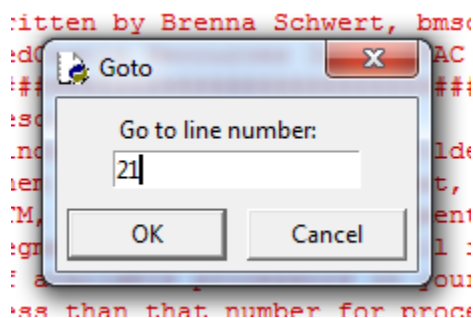
1. Navigate to the **Products** folder (c:\Project_Home\Products) and create a new folder named **Segmentation**.
2. Once you have created the new folder, navigate to the **CanopyHeight_1p0METERS** folder within the **backup** folder (C:\Project_Home\backup\CanopyHeight_1p0METERS).
3. Copy the TIFF file (.tif) and its four accompanying files located in that folder (see below), then paste them into the **Segmentation** folder (c:\Project_Home\Products\Segmentation) that you created.



Note: by default, the CHM in the CanopyHeight_1p0METERS folder is in a DTM format. In order to use those DTM outputs from the AreaProcessor in this python script, you would need to convert the DTM to a TIFF. The easiest way to do this is to use FUSION's DTM2TIFF utility, which is very simple to run, as it only requires the path to the DTM file and a path to the output TIFF file. For this course, a TIFF has already been created for you. Refer to the FUSION manual to learn more about the DTM2TIFF utility.

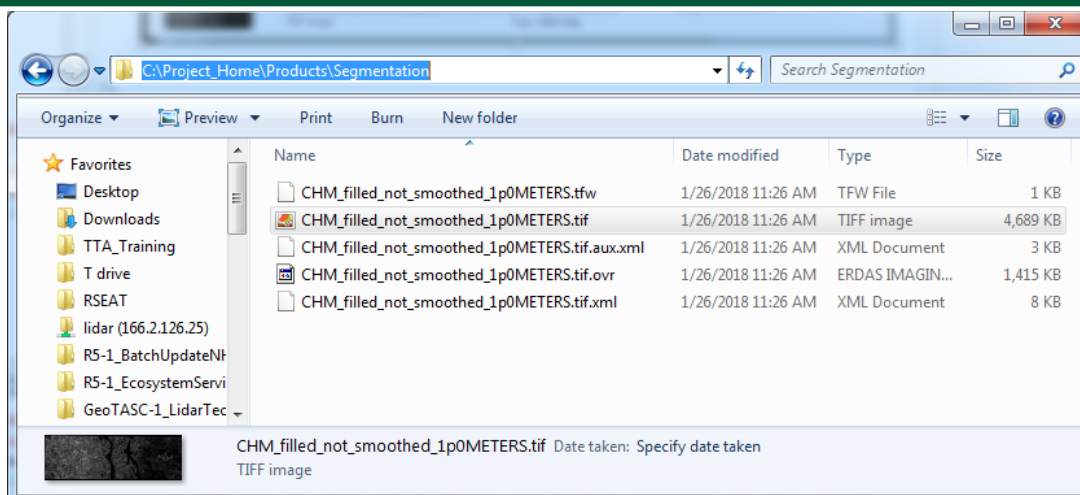
B. Edit the variables

1. Update the location of your working directory.
 - i. Press **Alt + G** – this will bring up the **Go to line number** prompt, allowing you to skip to a specific line of code in the script. Key in **21** and then hit **Enter**. This will bring you to the line that starts with **wd =**



Note: This line represents a variable, **wd** (short for working directory), that you need to set in order to run this script. This variable should be set to the Windows path of the folder that contains your canopy height model. It can be either a local path or a network path.

- ii. Press the **Windows Key + E** in order to bring up a new Windows Explorer window.
- iii. Navigate to the **Segmentation** folder where you pasted the CHM data.
- iv. Once you're in the folder that contains your CHM, select and copy the file path (shown below).



- v. Paste the path into line 21 of the script, making sure to leave the r character that precedes the path location in place (shown below).

```
#####
# DEFINE VARIABLES:
# Set working directory\
# Make sure to leave the 'r' before your working directory (wd) path below
wd = r'C:\Project_Home\Products\Segmentation' #the location of your CHM --
```

2. Set the properties of your CHM / lidar data:

- i. Line 24, **horizUnit**, represents the horizontal unit of your data. Set this to:
 - (a) 'm' for meters or
 - (b) 'f' for feet.
- ii. Line 25, **vertUnit**, represents the vertical unit of your data. Set this to:
 - (a) 'm' for meters or
 - (b) 'f' for feet.
- iii. Line 26, **coordSys**, represents the coordinate system of your data. Set this to:
 - (a) '1' for UTM,
 - (b) '2' for State Plane, or
 - (c) '0' for unknown.
- iv. Line 27, **zone**, represents the zone of your coordinate system. If you don't know the zone, set this to '0', otherwise set it to the zone number in single quotes.
- v. Line 28, **horizDatum**, represents the horizontal datum of your data. Set this to:
 - (a) '1' for NAD27,
 - (b) '2' for NAD83, or
 - (c) '0' if unknown.
- vi. Line 29, **vertDatum**, represents the vertical datum of your data. Set this to:
 - (a) 1' for NGVD29,
 - (b) '2' for NAVD88,
 - (c) '3' for GRS80, or

(d) '0' for unknown.

Note: The above parameters are the same ones you set in Exercise 2 when you used the ASCII2DTM utility. For this Kaibab data, the parameters should be set to **M M 1 12 2 2**.

3. Set the tree segmentation variables

- i. Line 32, **minTreeHeight**, represents the minimum height threshold for trees to be processed by FUSION. The default setting for the height threshold is '2', which in this case means 2 m. If your vertical unit is feet, this would represent 2 ft., so be mindful of double-checking your script before running it. Leave the minTreeHeight at its default of 2.
- ii. Line 33, **prj**, represents the path of a .prj file that can be used to assign projection information to the output shapefiles. Update this with a filepath of the same .prj file you used for the Area Processor (C:\Project_Home\deliverables\vector\workshop-area.prj).
- iii. Line 34, **overwriteValue**, tells the script whether to overwrite existing files.
 - (a) **True** means overwrite is on.
 - (b) **False** means overwrite if off.

4. Choose what to do with intermediate files.

- i. Line 37, **delDTM**, determines whether the script should delete the intermediate DTM files. If set to 'n' the script will leave the DTM folder intact. If set to 'y' the script will delete the DTM folder.
- ii. Line 38, **delTIF**, determines whether the script should delete the intermediate TIF files. If set to 'n' the script will leave the TIF folder intact. If set to 'y' the script will delete the TIF folder.

Note: There is no option to save the intermediate ASC files. While the tiled TIF and DTM files may be useful in the future (e.g., for viewing a small portion of the CHM or for doing other processing in FUSION), the ASC files serve no purpose other than for data conversion – they are deleted.

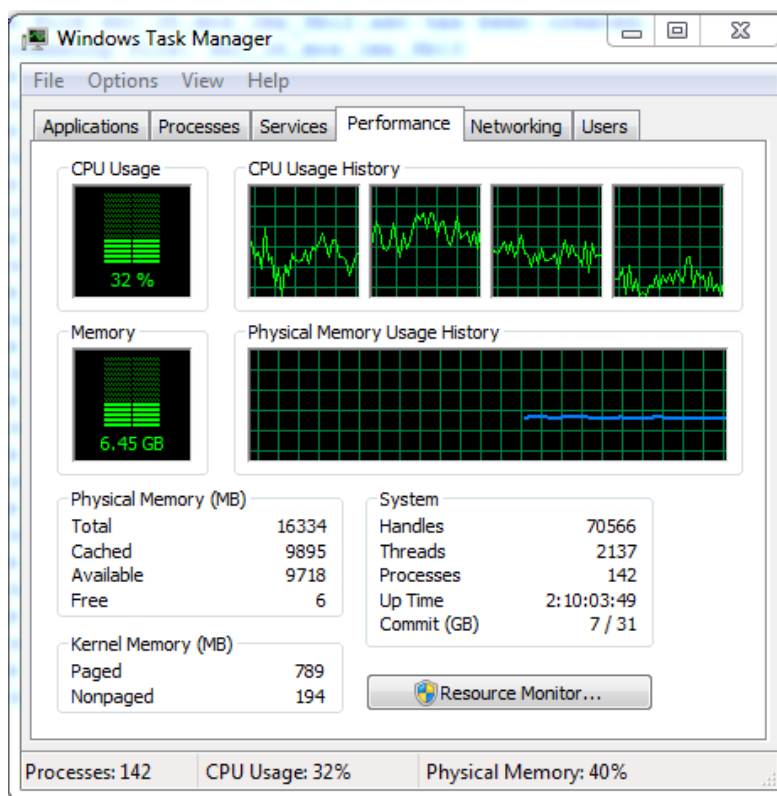
C. Save and run script

1. Press **CTRL + S** to save your script.
2. Press **f5** to run the script.
 - i. Your console will move to the front of your screen, and you will see that the script has begun to run.

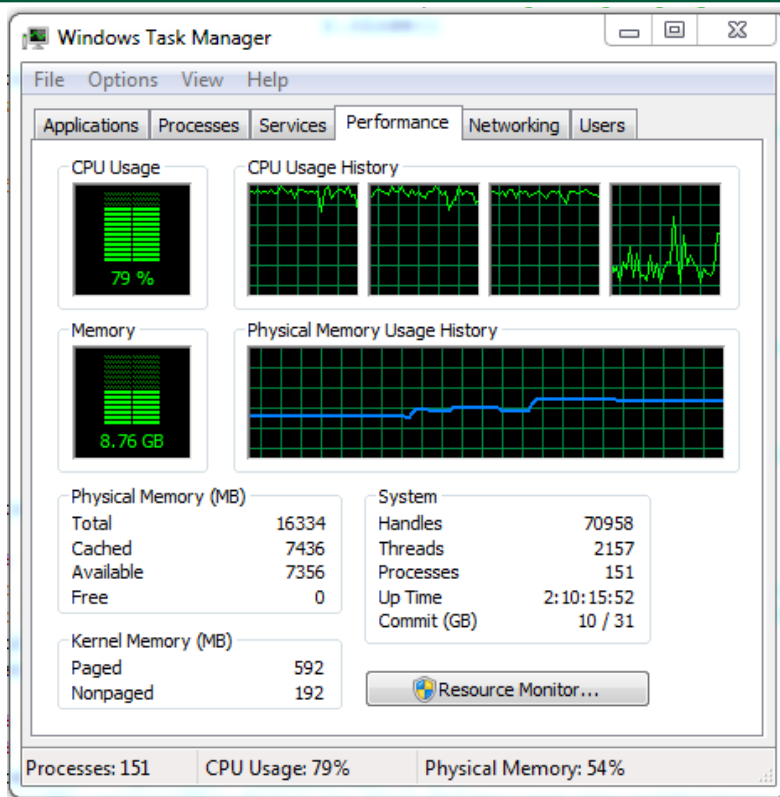
```
Python 2.7.8 (default, Jun 30 2014, 16:03:49) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Process began at 2017-10-03 10:52:59.119000.
```

```
Creating folders in working directory...
Processing files in C:\R3_TreeSeg
Generating a list of input rasters...
Checking if rasters meet size criteria...
Splitting rasters that don't meet size criteria...
```

- ii. As the script moves through parts of the workflow, you will be updated with print statements in the console.
3. Press the **Ctrl + Shift + Esc** to launch the Task Manager.
4. Click on the **Performance** tab to monitor how your computer's resources are being used.
 - i. In the steps before tree segmentation, it will be obvious that your computer is working, but it shouldn't be using anywhere near your computer's full processing power. In the graphic below, you can see that the machine has four available processors (the number of boxes shown in the CPU Usage History section), but only one is actually being used for these processing tasks. CPU usage is at about 32%, indicating that the computer has a lot more resources at its disposal.



- ii. During the tree segmentation processing, you will notice that your machine's resources will be more taxed. In the graphic below, you can see that the CPU usage is much higher, and there is more memory being used. This is because the tree segmentation process is using three of the four available processors. As a result, processing time will be shortened.



5. When the script is done, you will see your CPU usage return to normal levels, you will see completion statements printed to the Python console, and there will be three greater than symbols (>>>) in the console (shown below).

```
Process ended at 2017-10-03 11:59:17.903000.
Process took 3978.78400016 seconds to run.
>>>
```

Part 4: Examine outputs in ArcMap

A. Load CHM and Tree Segments

1. The outputs from the segmentation process are now stored in this folder:
C:\Project_Home\Segmentation\treeseg.
2. Open ArcMap and add the following data:
 - i. CHM_filled_not_smoothed_1p0METERS.tif
 - ii. Chm_filled_not_smoothed_1p0METERS_HighPoints.shp
 - iii. Chm_filled_not_smoothed_1p0METERS_Polygons.shp
3. Adjust the symbology of the polygons to a distinct color that contrasts well with the Canopy Height raster.
4. Examine the accuracy of the tree segmentation shapefiles relative to the Canopy Height raster and open the attribute tables to see what type of attribute information is available.
5. How accurate do the CHM polygons look? What are some applications for this type of data?



Congratulations! You have finished this exercise and successfully generated tree segments.