Volume

1

Missouri State University-Computer Science Department

Leaf Disk Analyzer User Manual

© Missouri State University

901 S National Ave

Springfield, MO 65897

Phone 417.836.5000

# A Note from the Author(s):

Firstly, we would also like to thank Dr. Laszlo Kovacs for the ideal behind the creation of the software, and Dr. Razib Iqbal for giving us the opportunity to work on the software.

Also, we would like to make note that with version 2.0 of the Leaf Disk Analyzer, accuracy has been increased tremendously, however, there are still some cases that may give off a false positive if the mildew you’re wishing to look at is a similar color to the background or is of darker nature. This software works primarily on brighter types of mildew, but if enough warrant is needed it can be adjusted to work for both bright/dark mildews. If any bugs or problems arise or if you have any suggestions for the software, please feel free to reach out to us at the official hosting place of the software at following link: <https://github.com/KySarge23/LeafDiskAnalyzer/> or send us an email regarding one of those subjects at our email: [analyzeyourdisks@gmail.com](mailto:analyzeyourdisks@gmail.com). We will do our best to address any problems brought up and find solutions for a timely fix and report that back out to our users. We hope you find the software useful for your needs and recommend it to others looking for something of this caliber!

Table of Contents

[A Note from the Author(s): ii](#_Toc23161573)

[Installing the Python Interpreter 1](#_Toc23161574)

[Installing Python 1](#_Toc23161575)

[Installing the Dependencies 3](#_Toc23161576)

[About OpenPyXL 3](#_Toc23161577)

[Installing OpenPyXL 3](#_Toc23161578)

[About OpenCV 5](#_Toc23161579)

[Installing OpenCV 5](#_Toc23161580)

[Acquiring and setting up the Leaf Disk Analyzer 7](#_Toc23161581)

[How to Get the Leaf Disk Analyzer Software 7](#_Toc23161582)

[Setting up the Leaf Disk Analyzer 9](#_Toc23161583)

[Using the Leaf Disk Analyzer 12](#_Toc23161584)

[Booting up the software 12](#_Toc23161585)

[About the LDA GUI 13](#_Toc23161586)

[Interacting with the LDA Software 14](#_Toc23161587)

[Examining the Spreadsheet 20](#_Toc23161588)

[Navigating through the spreadsheet 20](#_Toc23161589)

[A Quick Look At the Code 22](#_Toc23161590)

[Opening the Source Code 22](#_Toc23161591)

[Index 24](#_Toc23161592)

Chapter

1

# Installing the Python Interpreter

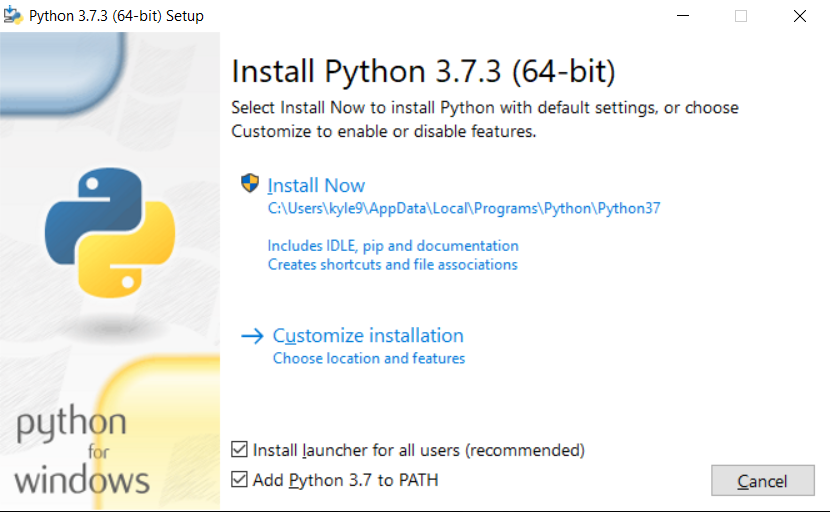
In this chapter, we will go over the installation process of the Python Interpreter and the dependencies of the Leaf Disk Analyzer software.

## Installing Python

B

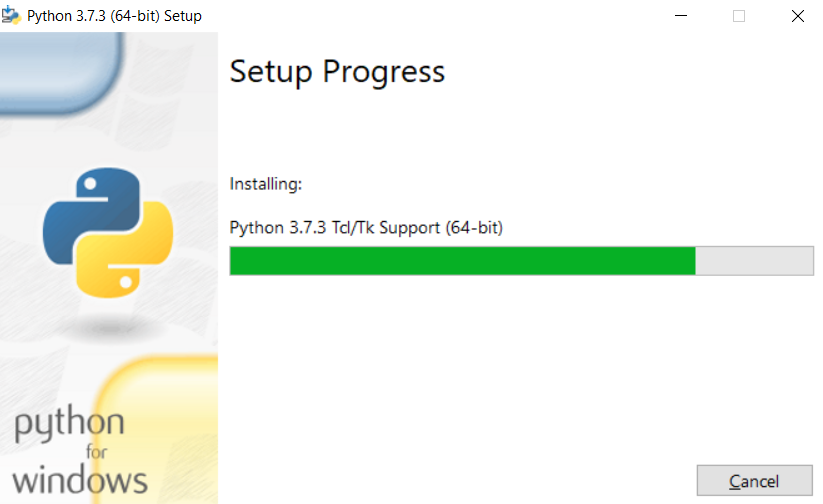
efore the software can be used, The user will need to download the most recent Python version (or any version greater than 3.4) onto their local machine. This will allow the user to install the dependencies necessary for the software once this set up has been completed. The link to download Python can be found at the following web address: <https://www.python.org/downloads/>. Once the you have reached this page, click the “download” button, After being redirected to the next page, select whichever version of the file works best for your machine (it is recommended that you do the “Windows x86-64 executable installer” version). When prompted to save the file, save it in any location of your choosing and leave the name as is.

After the downloading of the selected file has completed, navigate to where you have it saved and double-click the executable. Your screen should now look like the following:

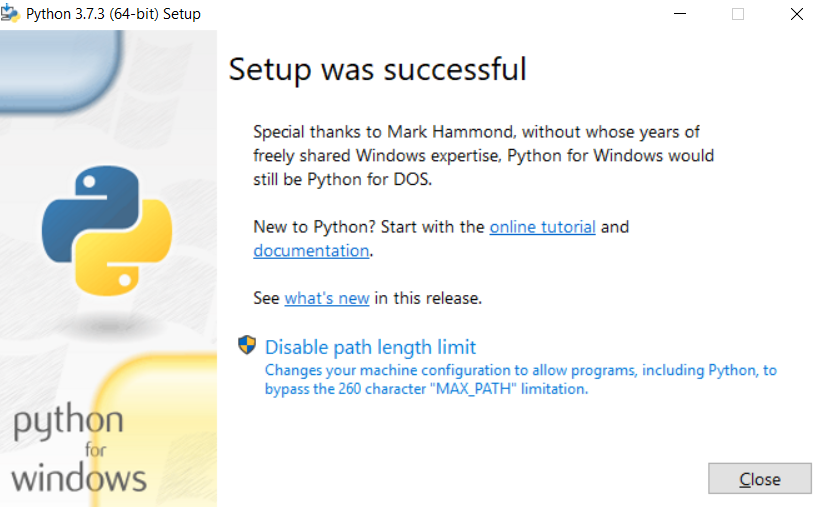


**Figure 1.1 Shown above is the 64-bit windows executable installer. This screen may be different depending on the file you’ve chosen to use.**

Once you’ve come across the screen displayed in Figure 1.1, be sure to click the Add Python 3.x to PATH and click the “Install Now” prompt. Your machine should ask if you want to give the application access to your machine to which you should click “yes”. At this point, you should see that the Python installation is being initialized before it begins. Let the installer run and eventually your screen should look something like this:

  
**Figure 1.2 This screen may be different depending on the file you’ve chosen to use and how far along the installation process is.**

Once the installer has finished, the screen should look like this:

 **Figure 1.3 Success screen of the python installer application.**

# Installing the Dependencies

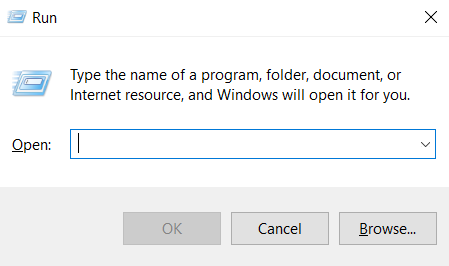
Now that the installation has been completed on your local machine, the Python environment necessary for the Leaf Disk Analyzer (LDA) software is now ready to be set up! This section will give a brief overview of the functionalities of each dependency as well as how to install it onto your machine.

## About OpenPyXL

openpyxl OpenPyXL, written by Eric Gazoni and Charlie Clark, will be your bridge between the Leaf Disk Analyzer and Microsoft Excel. It provides functionalities to the LDA software such as: creating, opening, or editing a workbook of your choosing; creating or editing a workbooks corresponding worksheet(s) and placing results found from the LDA software into cells within a designated workbook/worksheet and in its correct location. OpenPyXL provides a multitude of other functionalities not discussed above, but those other functionalities can be found here for reference: <https://openpyxl.readthedocs.io/en/stable/> .

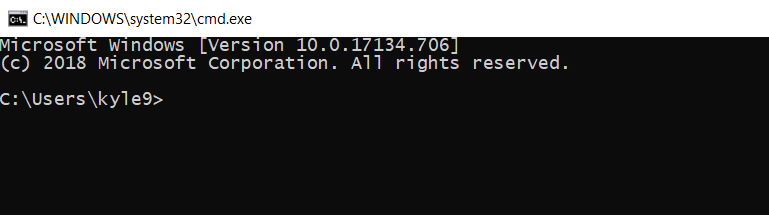
## Installing OpenPyXL

To install OpenPyXL on your local machine, press Win (“Win” key looks like this: thumb_WindowsKey) + R to bring up the “Run” prompt. Your screen should now have the following application on it:

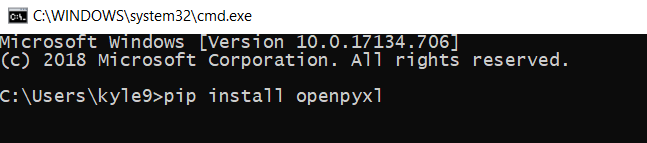


**Figure 1.4 Shown above is the run prompt found on windows. This will allow the user to enter in a name of an application they would like to run.**

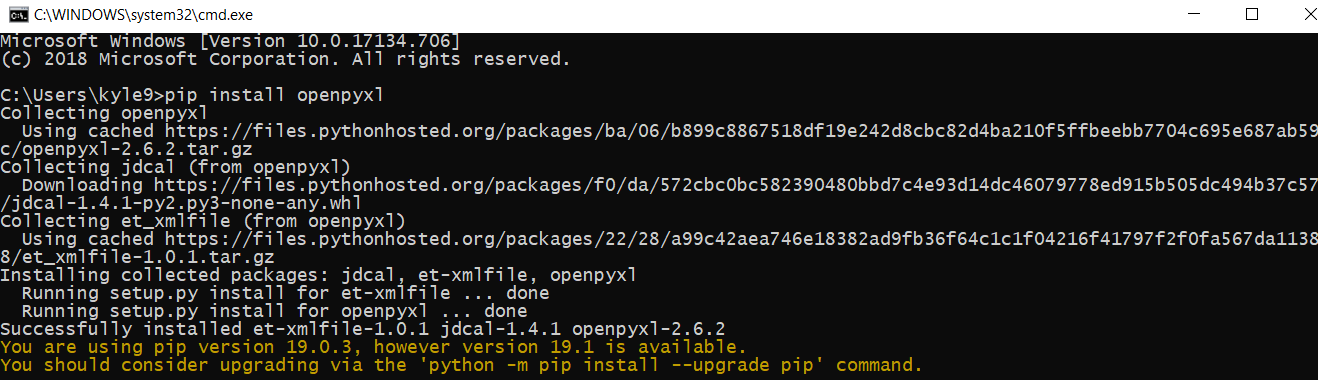
From this point, what you’ll want to do is type in “cmd” into the text box and hit the enter key. You should see a command line terminal pop up on your screen. Your screen should look like this:

** Figure 1.5 Shown above is the command line terminal for windows. This will allow the user to navigate through their system’s folders, directories, etc. Also, the user’s path displayed in the terminal will be different than that shown above.**

Once you see the screen as mentioned in **Figure 1.5,** what you’ll want to do is type in the following command: “pip install openpyxl” and then hit enter.

 **Figure 1.6 The command “pip install openpyxl” will call the pip installer, which is installed with python by default, to find and download the OpenPyXL modules onto your local machine.**

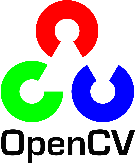
After hitting enter your screen should look something like this:

 **Figure 1.7 Once the command shown in Figure 1.6 is executed, it will pull the OpenPyXL files from an online source and download them onto your local machine.**

In **Figure 1.7,** OpenPyXL is already installed onto the machine. For your case, the command line should have more lines of “collecting…” and installing displayed.

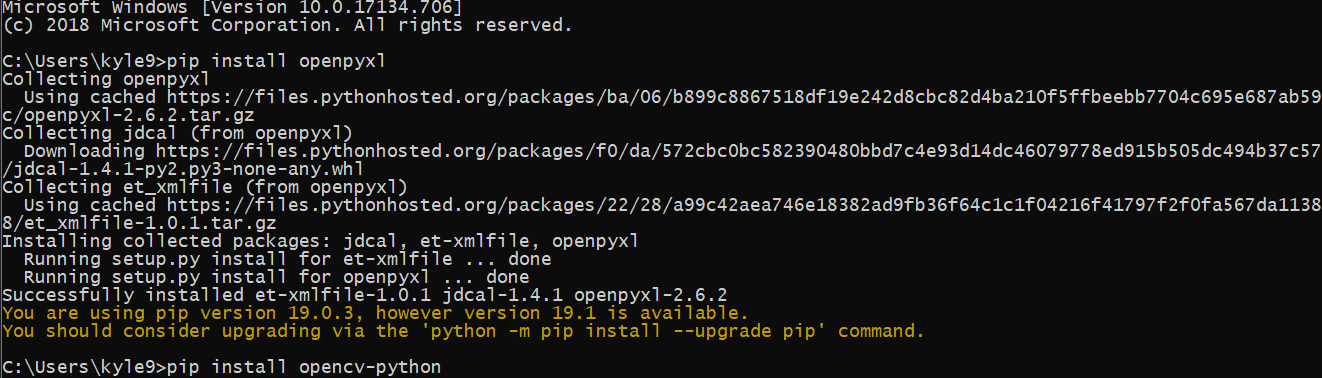
Once you receive the “Successfully installed x-files” line at the bottom of our command line, you’re all complete. OpenPyXL has now been installed on your machine. Also, keep the terminal window open. We’ll need it for the next dependency installation.

## About OpenCV

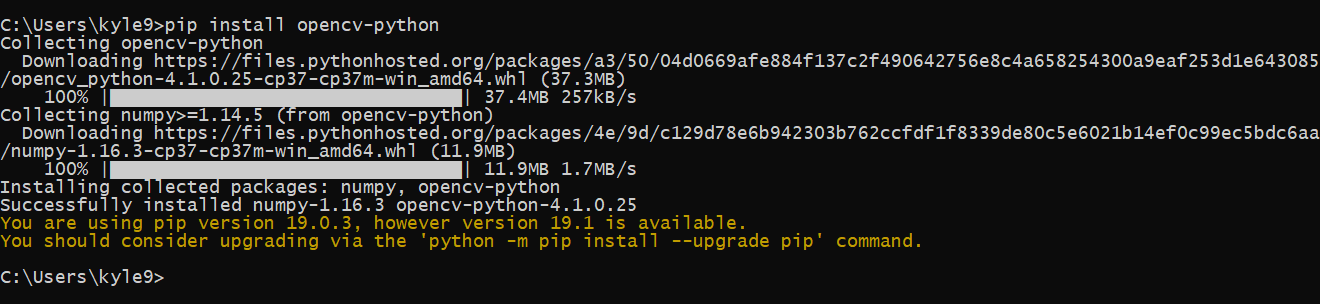
 OpenCV, or Open Source Computer Vision, originally designed by Intel, will be one of the Image Processing components of the LDA software. This dependency will bring in a library of functions that allow the LDA software to perform actions such as: reading in images off your machine, converting them to a different color scheme and more. This also has a ton of other functions that are best explained here at their home website: <https://opencv.org/>.

## Installing OpenCV

Installing OpenCV will have the same process as installing the last dependency of OpenPyXL. Since you should have the terminal still up (if you don’t, refer to the steps for installing OpenPyXL), enter the following command: “pip install opencv-python”. Your screen should now look like this:

**Figure 1.8 Once the command is executed, it will pull the OpenCV files from an online source and download them onto your local machine.**

Press the enter key and let the installation process finish. Upon completion your screen should look something like this:

 **Figure 1.9 This is what your screen should look like after the code in Figure 1.8 has been executed by your machine.**

At this point, if nothing has errored out, then all the dependencies have been installed. Also, don’t worry about the Numpy installation. OpenCV utilizes it for some functions and it’s also used in some background removal tasks in the code.

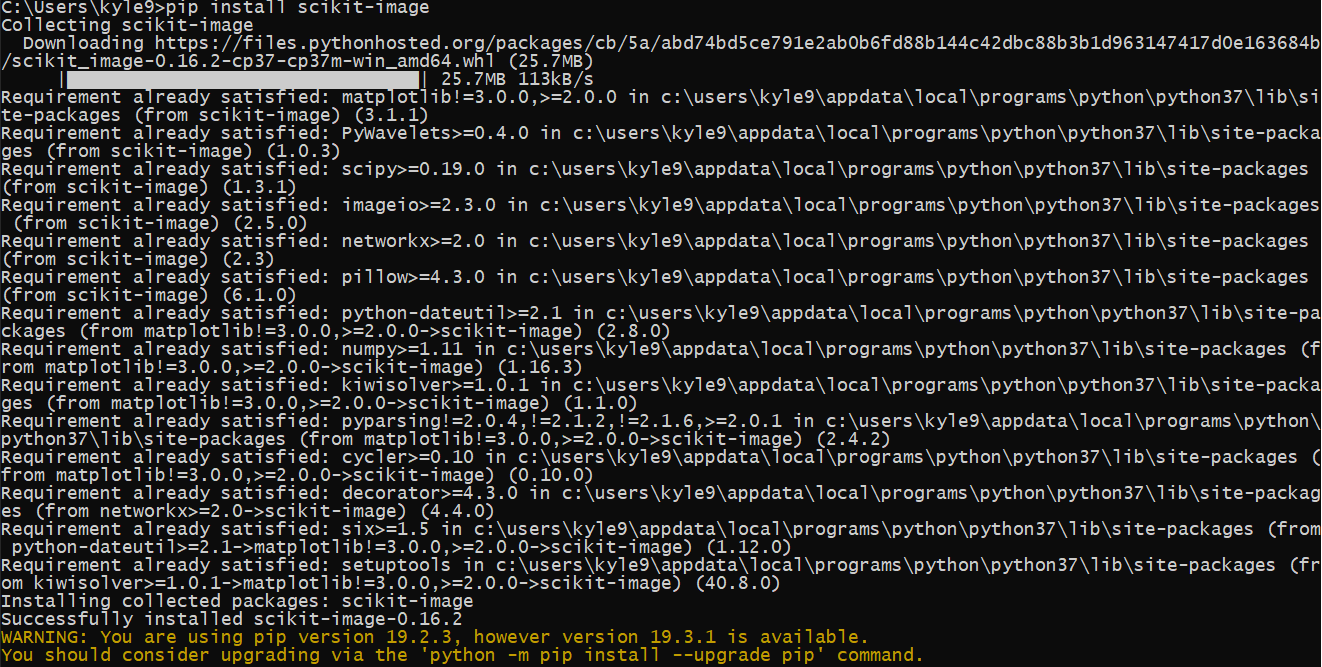
## About Scikit-Image

A close up of a logo

Description automatically generated Scikit-Image will be the other Image Processing component of the LDA software. This dependency will bring in a library of functions that allow the LDA software to perform actions such as: reading in images off your machine, conversions from color scheme to color scheme and the use of the Laplacian of Gaussian (LoG) Algorithm to detect mildew on the photo. This also has a ton of other functions that are best explained here at their home website: [[https://scikit-image.org/](https://opencv.org/)](https://scikit-image.org/).

## Installing Scikit-Image

Installing Scikit-Image will have the same process as installing the last dependency of OpenCv. Since you should have the terminal still up (if you don’t, refer to the steps for installing OpenCV), enter the following command: “pip install scikit-image”. Your screen should now look like this:

  
**Figure 1.10** Our screen after installing sci-kit image. Yours may look different if you don’t have some of the other dependencies that Sci-kit will install.

After this, you’ll want to type in “pip install scipy” which will install another extension of Scikit-image that will be necessary for the LDA software to work called SciPy. Follow the same steps for installing Scikit-Image and once it’s done, you’re all set to move onto the next part of the setting up process!

Chapter

2

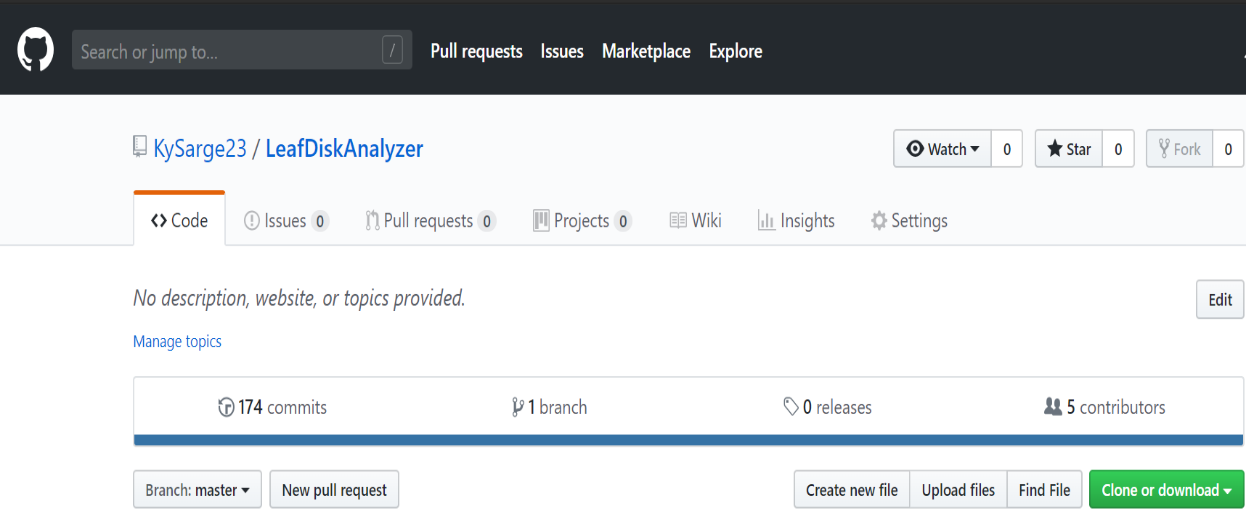
# Acquiring and setting up the Leaf Disk Analyzer

In this second chapter, we will go over how to get the Leaf Disk Analyzer Software and how to set it up properly.

## How to Get the Leaf Disk Analyzer Software

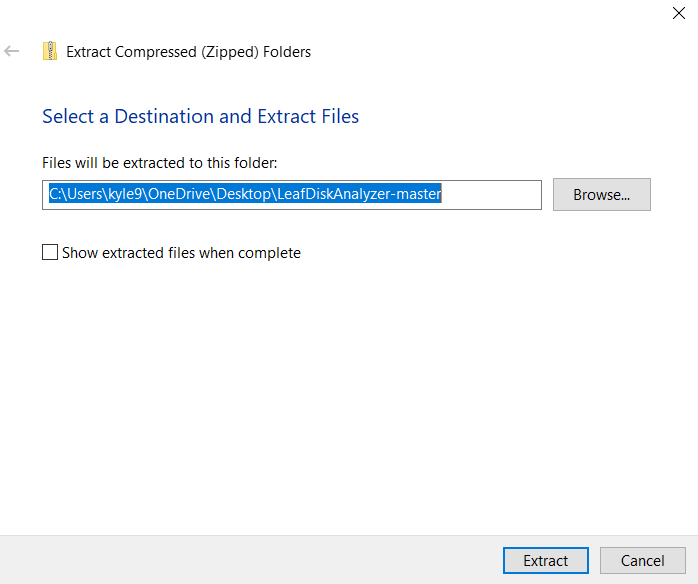
B

y this point, you should have both OpenPyXL and OpenCV installed on your machine. If not, please review chapter 1 for information on how to install those two modules. Acquiring the LDA software is a simple method. To obtain it, follow the following link: <https://github.com/KySarge23/LeafDiskAnalyzer>. This will be the main storage location to the LDA software. When you click on the link provided you should read a page that looks like this:

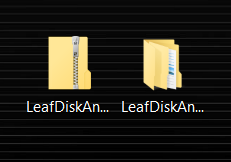


**Figure 2.1 This is the GitHub Repository where all changes to the LDA software will be held. This will be the central place for the code to live amongst all its changes.**

Once you’ve reached this page, you will want to click on the “Clone or download” button and click the “Download Zip” button as well. Save the .zip file wherever you please, but make sure it is in an easy to remember or easy to access spot (this is recommended for ease of access). After the downloading process has completed, navigate to where you saved the .zip file, right-click on it and click “Extract All”. The following should appear on your screen:

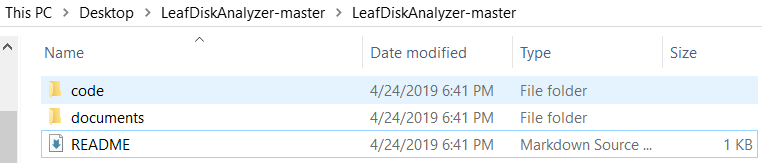
  
 **Figure 2.2 This is the windows extraction tool for Zipped Folders. This will allow you to extract the contents of the zipped folder for easier access.**

After you’ve reached the screen displayed in **Figure 2.2**, you choose to may save the extracted files wherever you please, and then click the extract button. This will extract all the files within the LeafDiskAnalyzer-master folder. When the extraction process is completed, you should now have two folders as such:

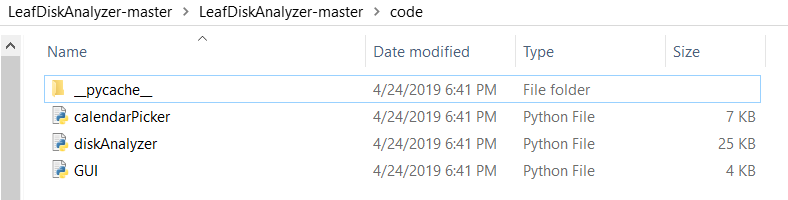
  
**Figure 2.3 This figure shoes the zipped and unzipped folders after extraction.**

## Setting up the Leaf Disk Analyzer

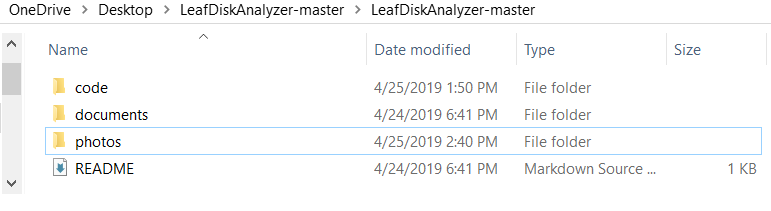
At this point, you can delete the zipped file and keep the non-zipped version. Now, you have the entire software! If you double-click on the non-zipped folder however, you should see a subfolder labeled “Leaf Disk Analyzer”. In this folder, you should see folders named “code” and “documents” and a readme.md file. It should look like this:

 **Figure 2.3 The contents of the LDA-master root folder are shown above.**

Feel free to delete the documents folder. Otherwise you may use it for storing any spreadsheets you create from this software. Now, double-click on the folder named “code”. Once you’ve done this, your screen should now look like this:

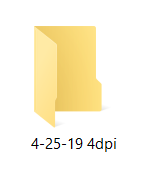
**Figure 2.4 The contents of the LDA-code folder are shown above**

Within your current view, you should see three python files and a folder. One named GUI, another named calendarPicker, and, most importantly, one named diskAnalyzer. Don’t worry about the “\_pycache\_” folder can just be left alone; it helps the software run faster. Now that we know where the code is located, exit out of the code folder and return to the view displayed in **Figure 2.3**. Once you’ve done that, create a new folder named “photos”. Your view should now look like this:

**Figure 2.5 The contents of the LDA-master root folder after photos folder was created.**

Navigate into the photos folder you just created, and perform the following tasks:

1. Create folder(s) with a date from when samples were collected (if the month is a singular digit, do not put the leading 0 in the name of the folder). You can attach the days past inoculation (DPI) attached at the end, but it is not necessary. The folder(s) should look something like the following:

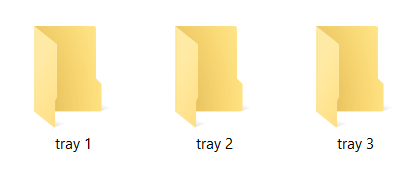


**Figure 2.6 Created folder after following Step 1.**

Note

For the folders you create, their dates `must be unique for the software to automatically detect the correct folder.

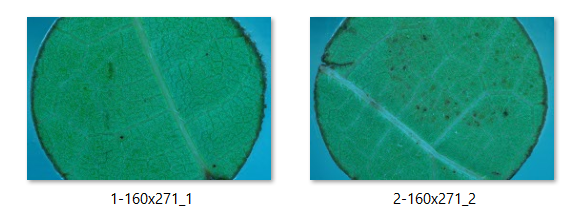
1. Navigate into the folder(s) created from step 1 and create more folder(s) for each tray of samples you collected. These folders should look something like the following:



**Figure 2.7 Created sub-folder(s) after following Steps 1 and 2.**

Within those tray folders is where you’ll want to place any pictures that you wish to have analyzed.

The last and final step to setting up this software is to name your photos as shown below:

**Figure 2.8 Sample photos. The photos you have may look different than the ones shown above**

Your photos must have the same naming format as the ones shown in **Figure 2.8.** Each leading number must be unique. This way, the software will be able to separate the photos from one another and access them automatically.

Recommendation

We recommend that before you run the software you reduce the size of your photos to 423x280. The software will do this automatically for you, but if you pass in bigger photos, you have the chance to encounter an OutOfMemoryError if running the software on a non-high-grade machine. So, to combat that, we recommend that you reduce them down to the referred size above.

Once you’ve gotten the photos’ names converted to the format displayed above, you’re all set to start using the Leaf Disk Analyzer Software. Make sure you have one named   
1-(anything here) and one named 2-(anything here) and that your photos have any one of the valid photo extensions (.jpg, .jpeg, .gif, .tif, .tiff, etc.).

Chapter

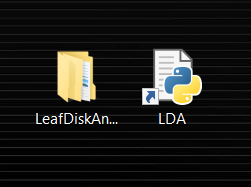
3

# Using the Leaf Disk Analyzer

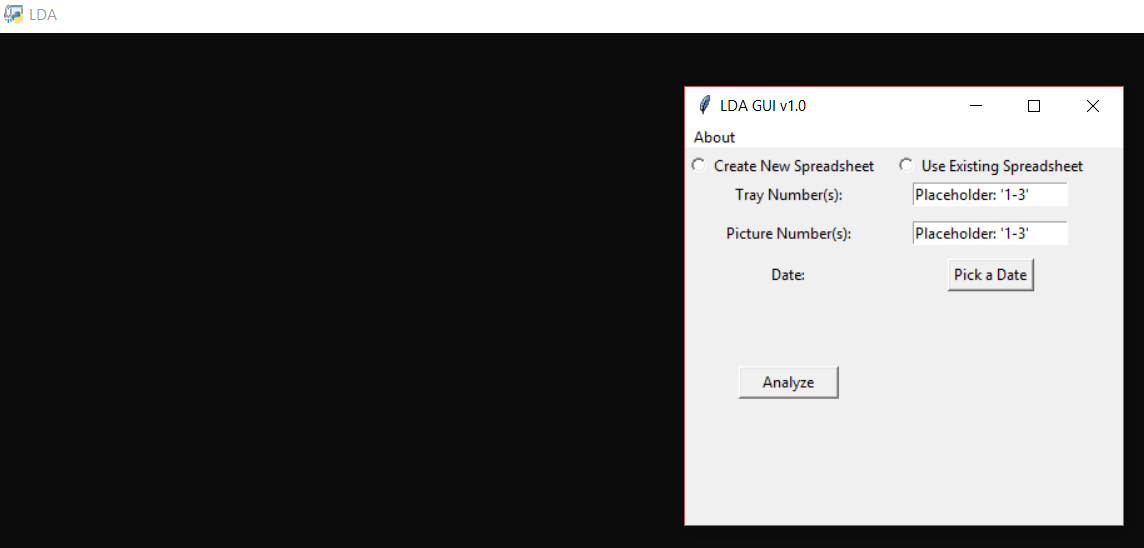
In this third, we will go over how to use the LDA software and what you should expect as an outcome from it.

## Booting up the software

Now that everything has been installed and set up for the software, it’s time to test it out! Navigate back to the code folder where the diskAnalyzer python file is. If you want an easier time accessing this file, right-click on it and then click on “create shortcut”. From here, drag the shortcut onto the desktop so that you don’t need to navigate into the LDA-code folder every time you wish to use the software. You should see something like this:

  
**Figure 3.1 The shortcut after being placed on the desktop. For our purposes, we renamed the shortcut to “LDA”. To do the same, press the F2 key and type in the new name.**

Once you’d done that, or if you chose to use the file within the LDA-code folder, double click the diskAnalyzer file. You should see a terminal pop up as well as the GUI for the Leaf Disk Analyzer software. On the first run, loading up may take a little bit so don’t worry if it doesn’t load right away. You should see something like this on your screen now:

**Figure 3.2 The GUI and the terminal that will pop up after double clicking on the diskAnalyzer.py file.**

## About the LDA GUI

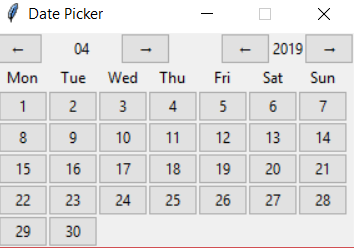
The terminal will be where you’ll see any outputs from the software first. You do not need to interact with this terminal either, just the GUI. Since we’ve got the GUI up now, lets go over what each individual field on the GUI will allow you to do.

1. Create New Spreadsheet radio button – This button will tell the software to allow for you to create a file with the ‘.xlsx’ extension. The file created will later be written to after the analyzing process has completed for the photos selected.
2. Use Existing Spreadsheet radio button – This button will tell the software to allow for you to select an existing spreadsheet of your choosing for writing to after the analyzing process has completed for the photos selected.
3. Tray Number(s) entry field (right of the Tray Number(s) label) – This is where you will enter the tray number(s) you wish to select photos from. You must put in numbers in one of three formats.
   1. A range of numbers in the form of 1-3 or 4-8.
   2. A selection of numbers in the form of 1,2,3 or 3,5,6.
   3. A mixture of (a) and (b) separated by a space.
4. Picture Number(s) entry field (right of the Picture Number(s) label) – Similar to the tray number(s) entry field, you will enter the picture numbers you wish to have analyzed from the trays entered above. These must be input in one of the same formats as noted in (3) above.
5. Pick a Date button – This will display a drop-down calendar for you to pick the date of the samples (the folder name) that you wish to pick the trays and pictures from.
6. Analyze button – This button will be what you press once you’ve entered in all the data. This will collect all the data found entered in the GUI, validate them and then run the processing on the photos selected if it passes the validation portion
7. About taskbar button – this will allow you to access two more menu-buttons. One for viewing a rundown about the “LDA Software” within a pop-up box and another for viewing the copyright for the Leaf Disk Analyzer Software within a popup-box.

## Interacting with the LDA Software

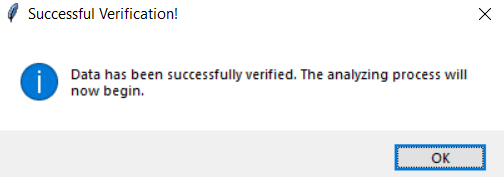
Now that we know what each portion of the GUI does, it’s time to finally test the software. Follow the steps below to see how the LDA software works:

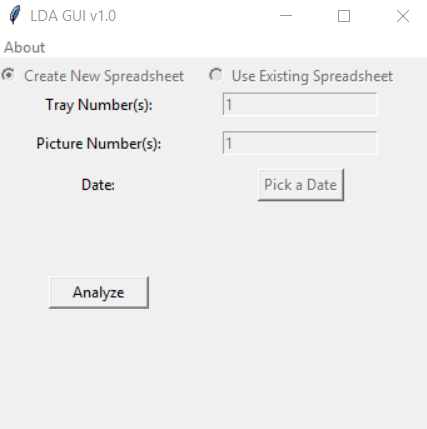
1. Click on the “Create New Spreadsheet” radio button.
2. Enter the number “1” into both tray number(s) and picture number(s) entry fields.
3. Click on the “Pick a Date” button. You should see something like the following pop up on your screen:

  
**Figure 3.3 The date picker that is created upon clicking the “Pick a Date button”.**

1. Select the date of the samples that you wish to analyze (**Reminder: this will be the name of a folder you created in the “photos” folder.)** We chose 4-25-2019 for the sake of this example.
2. Press the “Analyze” button.

At this point, your GUI fields and buttons should turn gray and a message displaying “Successful Validation” should appear on your screen like so:

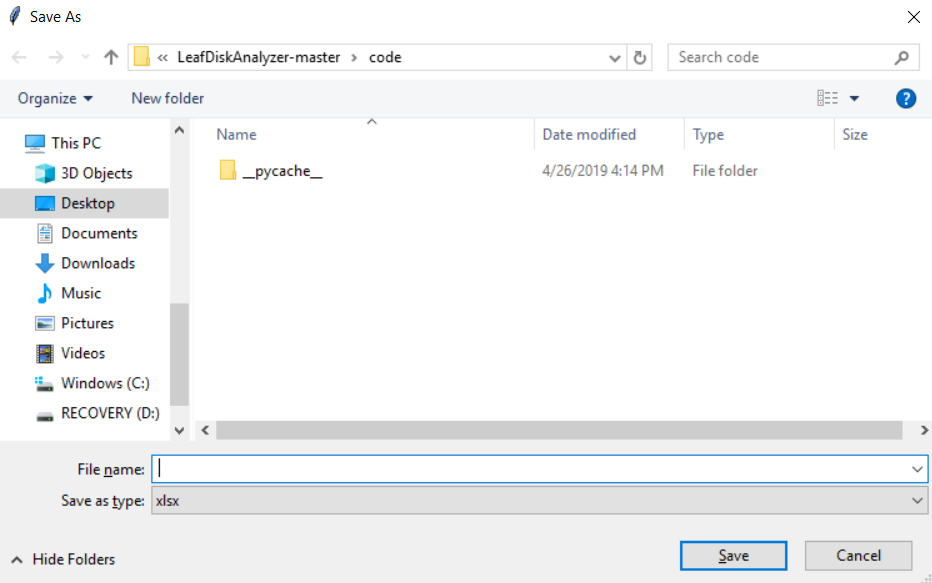
  
**Figure 3.4 The success message that is returned upon successful data validation.**

  
**Figure 3.5 The GUI fields and radio buttons should be greyed out, dictating they’ve been disabled from use.**

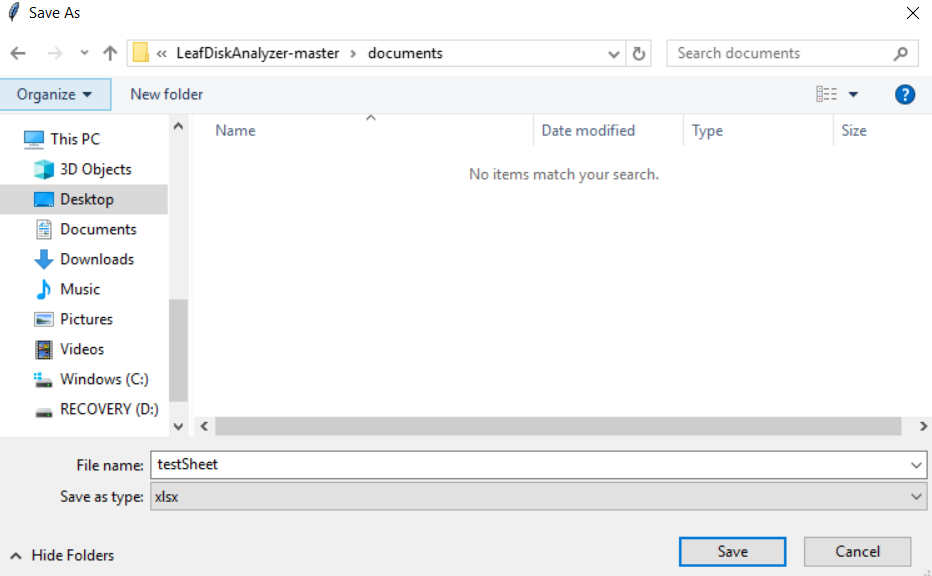
Your terminal at this point should have printed out the date for you as well so you know which date folder it will be searching for. It should look something like this:

  
**Figure 3.6 The date is selected via the drop-down calendar is printed to the screen after data validation has completed.**

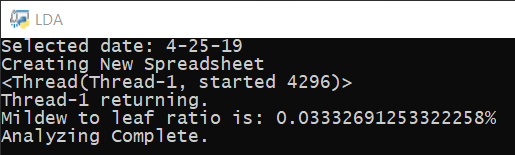
Click on the “OK” button as depicted in **Figure 3.4**, and you should now see another line, reading “Creating New Spreadsheet” or “Going to Existing”, being printed to the terminal. This will change depending on the selection of the radio buttons atop of the GUI window. If you see “Creating New Spreadsheet” then you should see this “Save As” window pop up:

**Figure 3.7 The Save As window that allows the user to create a new spreadsheet, if the option was chosen.**

You can save the file under whatever name you’d like and wherever you’d like. This is what we named our test spreadsheet:

**Figure 3.8 Printed statements from the terminal, letting the user know what is happening at what point of the software.**

Click the “Save” button and you should now see more lines being printed onto the terminal. The terminal should read something like this:



**Figure 3.9 Printed statements from the terminal, letting the user know what is happening at what point of the software.**

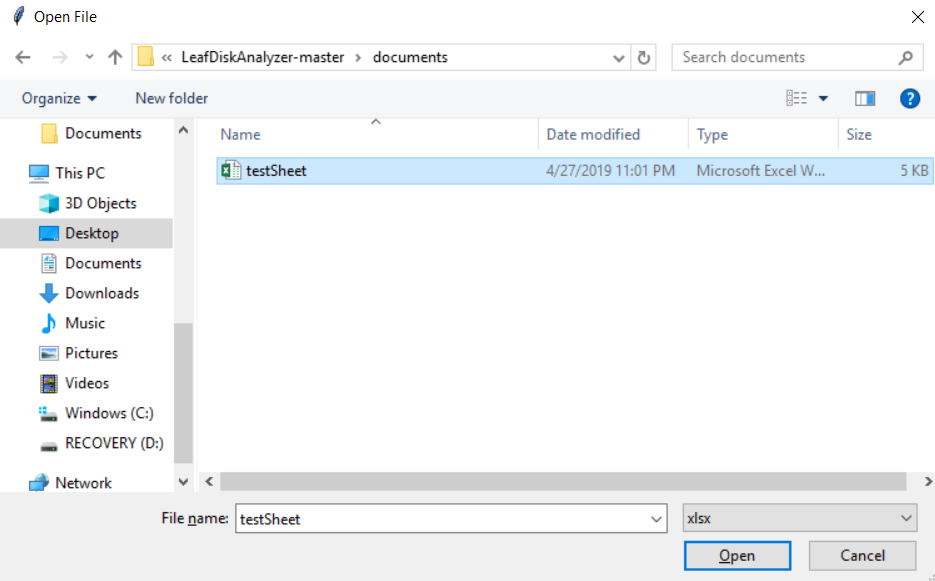
At this point, you should now see the fields of the GUI fields become enabled and be able to change the field(s) entries. We’re going to run this again with choosing an existing spreadsheet, but before we do that, let’s go over the new lines displayed in **Figure 3.8:**

1. <Thread(Thread-1, started 4296)> – This line details the thread that has started, its thread number, and CPU process number. This line isn’t too important to you, but it’s useful to know which threads are being spawned.
2. Thread-1 returning – This line details the thread that is returning at this point in the code. The more threads you have, the more of this line and the line discussed in (1) you will have.
3. Mildew to leaf ratio is: 0.03332691253322258% - This line is the calculated mildew to leaf ratio. It should be interpreted as “0.0333…22258 % of the pixels within the detected leaf disk circle have been identified as mildew”.
4. Analyzing Complete. – This line indicates that the analyzing process has completed. After this line is printed, you should be able to interact with the GUI fields once again.

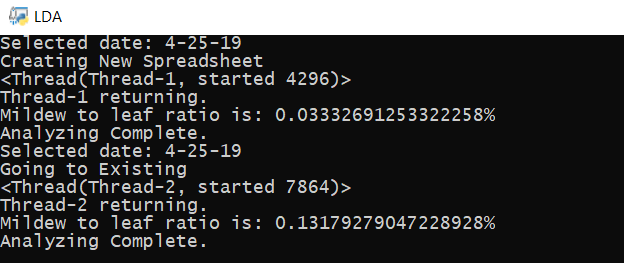
Now that we know what the new lines mean, lets test the software again only this time using an existing spreadsheet (the one we just created from the previous test). With the software still up and running from its previous run, follow the steps below:

1. Click on the “Use Existing Spreadsheet” radio button.
2. Leave the tray number(s) entry field as is
3. Enter “2” into the picture number(s) entry field
4. Do not select a new date.
5. Click the “Analyze” button.

After completing the steps above, you should see the same screens as displayed in **Figure 3.4, 3.5 and 3.6.** Once you’ve progressed past those screens, you should now see an “Open File” window. Navigate to where you saved the spreadsheet from the last run and click on the file. Your screen should now look like this:

**Figure 3.10 Open File window. This is what the use will see if they select the “Use Existing Spreadsheet” radio button.**

Click on the “Open” button and let the analyzation process complete. Your terminal should now look like this:

  
**Figure 3.11 Terminal after creating the new spreadsheet and writing to the existing spreadsheet.**

With that being done, you’ve successfully used the LDA Software! However, there is one more aspect to look at before you’re fully equipped to use the software. Close out the software by clicking the red X, or hitting the “esc” key, in the top-right corner of the LDA GUI and you’ll get a pop-up box that will ask if you want to close the software; hit yes.

Chapter

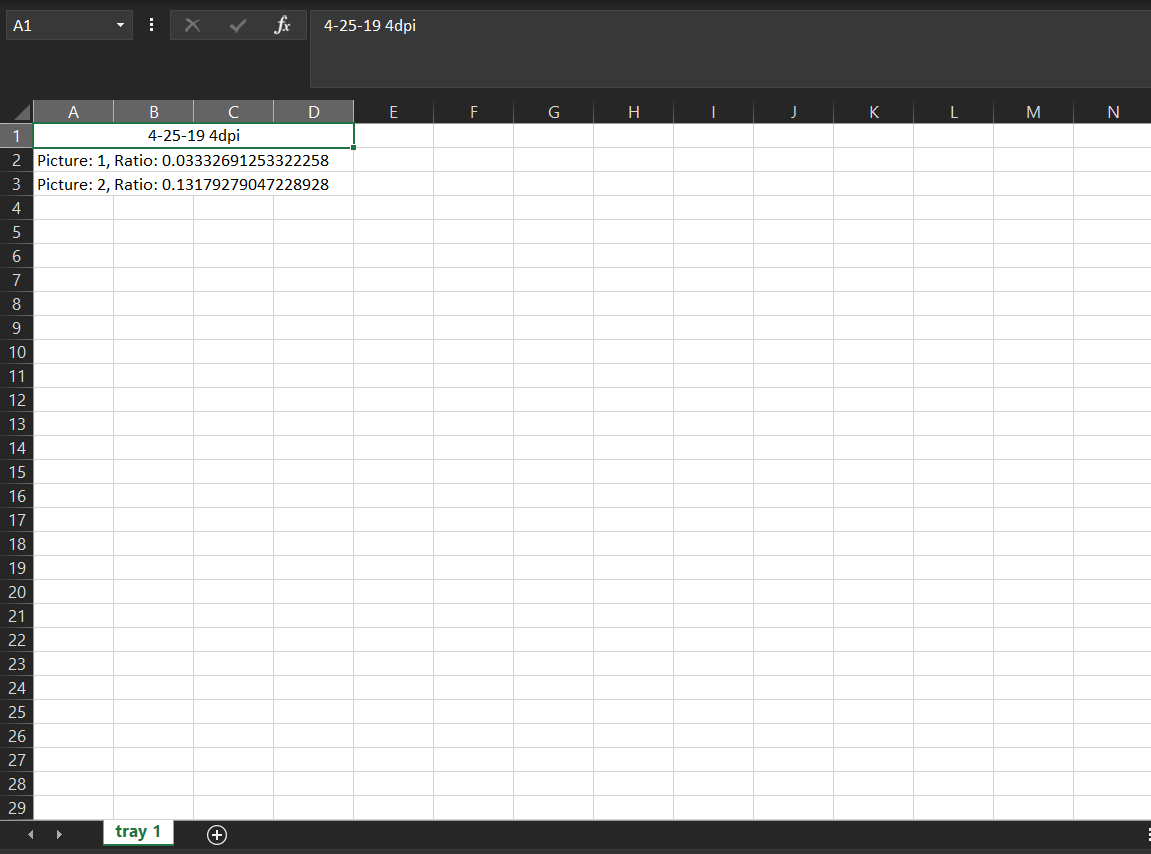
4

# Examining the Spreadsheet

In this fourth chapter, we explain the aspects of the spreadsheet you created in the last chapter.

## Navigating through the spreadsheet

To begin, let’s make our way back to where we saved the spreadsheet we created and used in the previous chapter. Once you’re there, double-click on it to open it up. You should see something that looks similar to this:

**Figure 4.1 The output of the spreadsheet created in chapter 3.**

As you may have expected, you can see the date chosen (the folder name you wished to use samples from) and the picture number(s) you chose along with its respective mildew to leaf ratio. Also, you may notice that the sheet name is the tray that the picture is from. Each input tray will have a sheet created for it within the spreadsheet. If it is a new spreadsheet, then all tray numbers will have a sheet created. If it is an existing spreadsheet, then only new tray numbers will have a sheet created. Otherwise it will write to the already existing sheet that corresponds to that tray. This prohibits samples from different trays being written into an incorrect tray’s sheet thus giving incorrect information for that tray.

As for the ratios you see, they are indexed by their picture number + 1 since the date resides in the first row. So, picture 1 will be placed in row 2, picture 2 will be placed in row 3, so on and so forth. This allows for a consistent placing on each sheet for each corresponding photo. If you need to find the third photo on three different trays, then you’ll only need to look at the fourth row.

Now that we’ve gone over all aspects of the LDA software, you’re ready to use it to its fullest potential!

Chapter

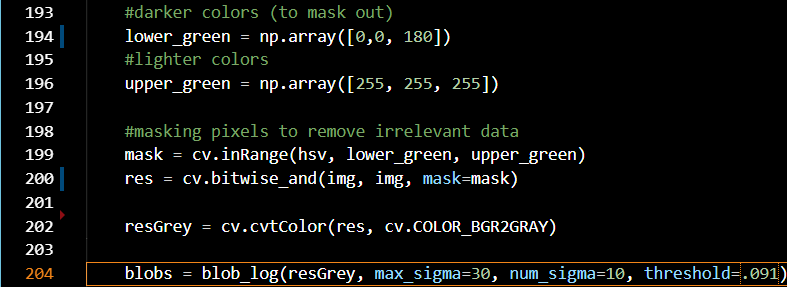
5

# A Quick Look At the Code

In this fifth and final chapter, we take a quick look at the code so that you can adjust the program to your liking.

## Opening the Source Code

To begin, navigate to where you have the “diskanalyzer.py” script saved at. Right-click the script and select “Open with >” and choose the IDE (Integrated Development Environment) or code editor of your choosing. We chose VS Code, for its accessibility and ease of use, but any other will do. Once the script has opened, you want to navigate to the “calculateMildewArea” function. You’ll want to look for the lines as shown below:

**Figure 5.1 The lines that are of interest to us for this chapter. These lines can be found in the calculateMildewArea function of the script.**

However, we really want to focus on lines 194 and lines 204. You may be wondering why these lines are important, let alone relevant, but this is one of the first lines that affects the outcome of the program. Specifically, the last number of this line, within the brackets, is the one we truly care about. Increasing the last number of the line causes more of the photo to become masked out during the mildew detection process. Decreasing that number causes less of the photo to become masked out during the mildew detection process, so if you think that there’s some spots that are mildew but aren’t being picked up, or they aren’t mildew and are being picked up as well, this is the first place you want to look to adjust that outcome.

For line 204, we also want to focus primarily on the last number denoted by “threshold”. This number translates to how many circles the Laplacian of Gaussian (LoG) algorithm

can detect based off the kernels it has used to detect key points within the masked photo (hence why line 194 is important). For more information on the LoG Algorithm, we recommend that you look here: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm>. Much like that of line 194, a higher number reduces the total amount of circles that may be detected, whereas a smaller number can increase the total amount of circles that may be detected within the photo.

So, now you can now adjust the code to whatever seems to work best for your samples! Just keep in mind that after every time you alter a number, you must save and restart the LDA software so that the changes take effect.

# Index

cmd – 3

date – 9, 13, 14, 15, 17, 20

date picker – 13

folder – 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20

gui – 8, 12, 13, 14 ,15, 17, 18

ide – 22,

install – 1, 2, 3, 4, 5, 6, 11

log – 6, 22, 23

lda – 3, 5, 6, 8, 11, 12, 13, 14, 19, 21, 23

leaf disk analyzer – 1, 3, 6, 8, 10, 11, 12, 13

opencv – 5, 6

openpyxl – 3, 4, 5, 6

path – 1, 4

photos – 5, 6, 10, 12, 13, 14

picture number – 13, 17, 20

python – 1, 2 , 4, 8, 11

ratio – 5, 17, 20

row – 20

scikit-image – 6

scipy - 6

spreadsheet – 11, 12, 13, 15, 17, 18, 19, 20

thread – 17

tray number – 12, 13, 17, 20