Computer Vision Emergency Response Toolkit

User Manual

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

This section's intended purpose is to provide a general overview of the application and its intended use as well as the layout of the user manual.

1.1 Application Overview

The Computer Vision Emergency Response Toolkit is an application that allows users to detect features of interest in high-resolution photos obtained from UAV fly-overs.

Users can run an analysis job by selecting a single image, a set of images, or a folder of images through the interface provided. Upon completion of the analysis of the image(s) provided, users can view the features of interest side-by-side with the original image, also through the interface provided. The ability to view jobs and their respective image heatmap pairs is available anytime, even while a job is running. In addition, there is a checkbox feature provided that enable users to keep track of the images they have reviewed and what is left to be done.

2.0 SYSTEM REQUIREMENTS

This section's intended purpose is to provide the user with the minimum set of system requirements needed to be able to run the application.

2.1 Hardware

Required:

- 64-bit processor
- Internet connection is required during installation only

Recommended:

- 8 gigabyte (GB) RAM or more
- 2.5 gigahertz (GHz) quad-core processor

2.2 Software

- Windows 7 64-bit
- Python 3.6.4 64-bit
- Microsoft .NET Framework 4.6.1

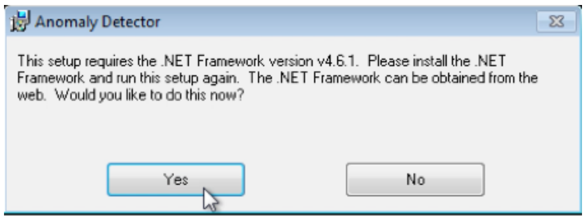
3.0 GETTING STARTED

This section's intended purpose is to help the user to get started using the application.

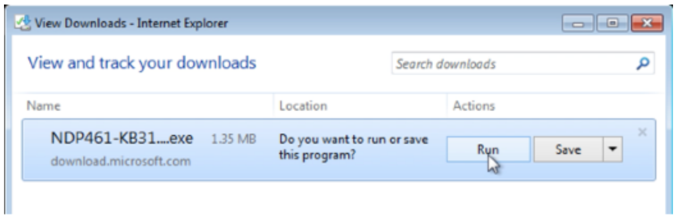
3.1 Installation

To install the program, follow these steps:

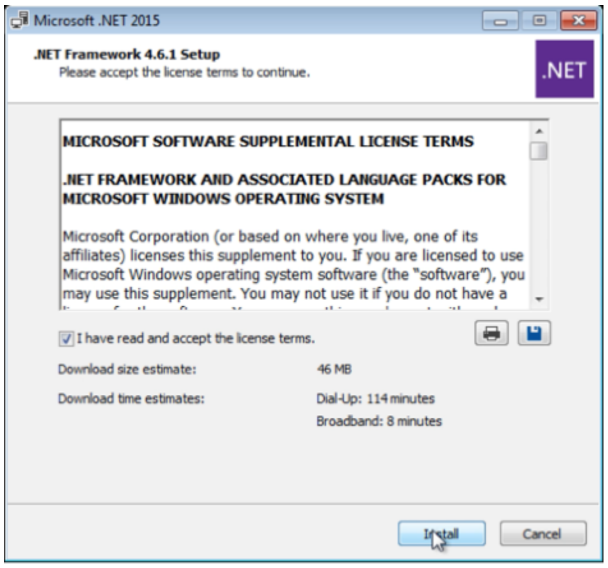
1. Open the installer file named "Setup.exe". This will begin the install process.
2. The installer will prompt you to install the .NET Framework 4.6.1 if your system does not already have it. Please select "Yes" to start downloading this component. Select "Run" and follow the instructions to install this component.



This setup requires the .NET Framework version v4.6.1. Please install the .NET Framework and run this setup again. The .NET Framework can be obtained from the web. Would you like to do this now?


View Downloads - Internet Explorer
View and track your downloads
Search downloads
Name
Location
Actions
Run
Save

NDP461-KB31....exe 1.35 MB download.microsoft.com Do you want to run or save this program?



.NET Framework 4.6.1 Setup
Please accept the license terms to continue.

MICROSOFT SOFTWARE SUPPLEMENTAL LICENSE TERMS

.NET FRAMEWORK AND ASSOCIATED LANGUAGE PACKS FOR MICROSOFT WINDOWS OPERATING SYSTEM

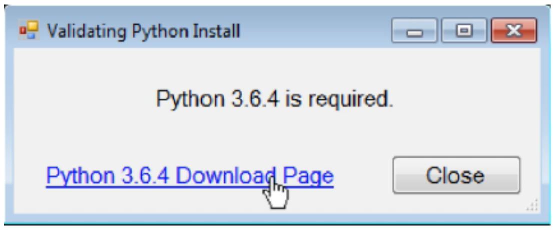
Microsoft Corporation (or based on where you live, one of its affiliates) licenses this supplement to you. If you are licensed to use Microsoft Windows operating system software (the "software"), you may use this supplement. You may not use it if you do not have a

☑ I have read and accept the license terms.

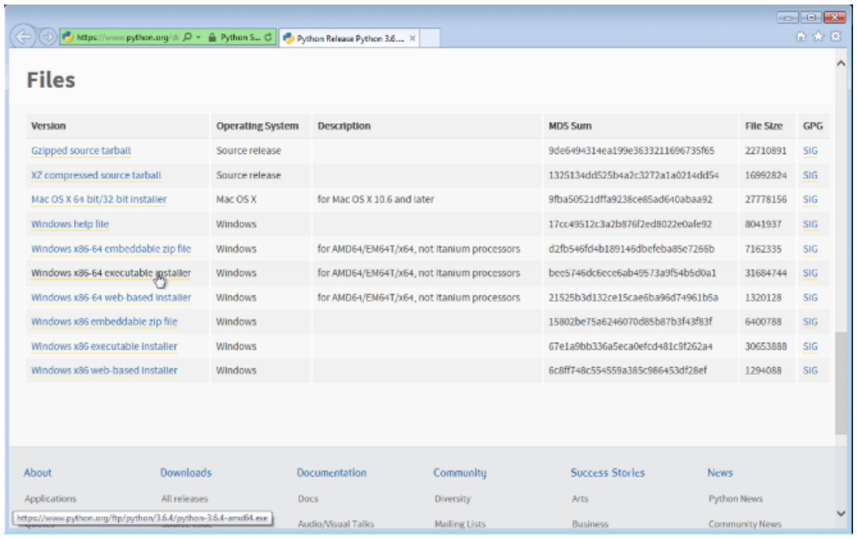
Download size estimate:	46 MB
Download time estimates:	Dial-Up: 114 minutes
	Broadband: 8 minutes

3. When the .NET Framework 4.6.1 has finished installing, please open the installer file named "setup.exe" and follow the instructions to continue with installing the application.
4. The program needs to be installed in a user controlled location such as the Desktop or the default location. Click the next button to proceed and follow the on screen instructions.

5. After installation, a shortcut to the program will be added both to the Windows start menu and to the Desktop.
6. When opening the program for the first time, it will check if Python 3.6 is installed. If the window shows “Python setup complete…” then you may proceed to step 10. If not, click on the link to download python. Then choose from one of the two Windows x86-64 installer at the bottom to download and run.



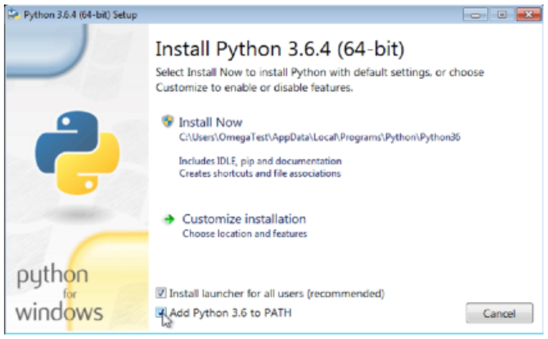
Python 3.6.4 is required.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		9de6494314ea199e3633211696735f65	22710891	SIG
XZ compressed source tarball	Source release		1325134dd525b4a2c3272a1a0214dd54	16992824	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	9fba50521dffa9238ce85ad640abaa92	27778156	SIG
Windows help file	Windows		17cc49512c3a2b876f2ed8022e0afe92	8041937	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64, not Itanium processors	d2fb546fd4b189146dbefeba85e7266b	7162335	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64, not Itanium processors	bee5746dc6ece6ab49573a9f54b5d0a1	31684744	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64, not Itanium processors	21525b3d132ce15cae6ba96d74961b5a	1320128	SIG
Windows x86 embeddable zip file	Windows		15802be75a6246070d85b87b3f43f83f	6400788	SIG
Windows x86 executable installer	Windows		67e1a9bb336a5eca0efcd481c9f262a4	30653888	SIG
Windows x86 web-based installer	Windows		6c8ff748c554559a385c986453df28ef	1294088	SIG

7. When installing python, make sure to check the box labeled “Add Python 3.6 to PATH”. Python needs to be installed in the default location. Click “Install Now” to use the default location.



Python 3.6.4 (64-bit) Setup

Install Python 3.6.4 (64-bit)

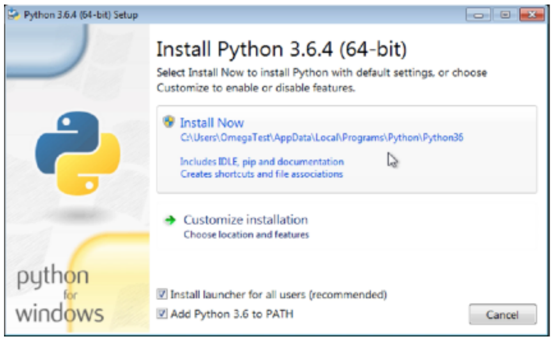
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\OmegaTest\AppData\Local\Programs\Python\Python36

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.6 to PATH



Python 3.6.4 (64-bit) Setup

Install Python 3.6.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

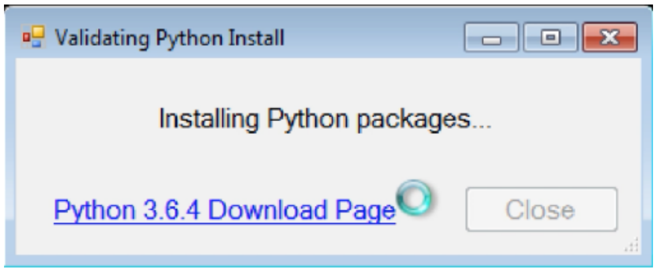
Install Now
C:\Users\OmegaTest\AppData\Local\Programs\Python\Python36

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.6 to PATH

8. Wait for the program to install the required python packages. This could take a few minutes to complete.



Validating Python Install

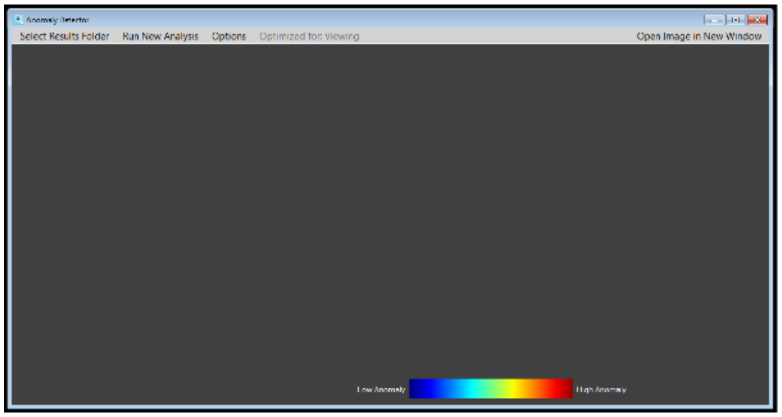
Installing Python packages...

9. Click close. Please close out of and restart the application.
10. The Missing Person Detection Toolkit is now fully installed.

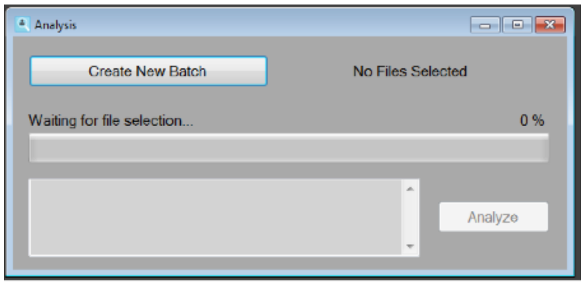
4.0 USAGE

4.1 Running an Analysis Job

1. Select the “Run New Analysis” button near the top left.



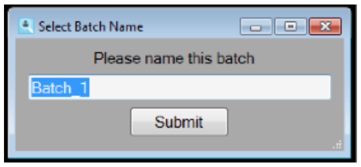
2. In the new window that pops up, select “Create New Batch”.



No Files Selected

Waiting for file selection... 0 %

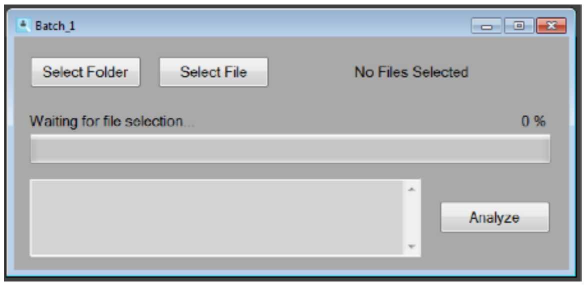
3. Next pick a name for the folder where the results will be stored and click “Submit”.



Please name this batch

Batch_1

4. Now you are ready to pick images to analyze. Click either the “Select Folder” or “Select File” button, to analyze all images in a folder or only those selected, respectively.

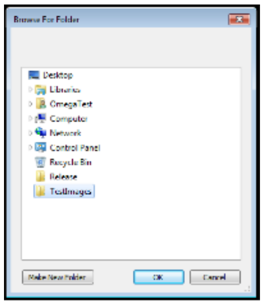


No Files Selected

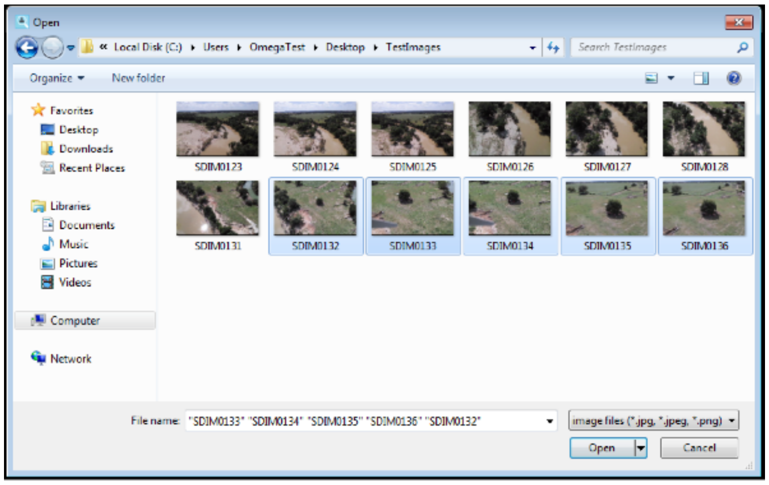
Waiting for file selection...

0 %

   a. “Select Folder”
      i. Navigate to the file/folder you want to analyze and click “Open” in the bottom right to continue.



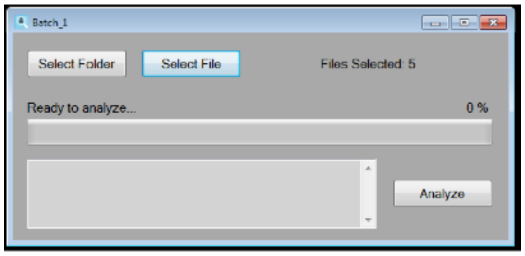
   b. “Select File”
      i. Select all images you would like to analyze and click “Open” in the bottom right to continue.

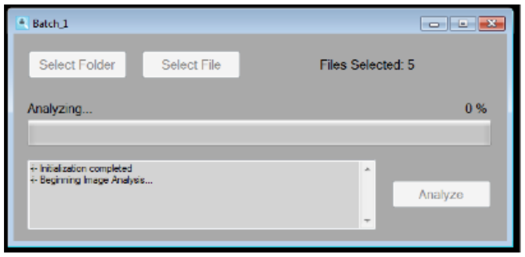


File name: "SDIM0133" "SDIM0134" "SDIM0135" "SDIM0136" "SDIM0132"

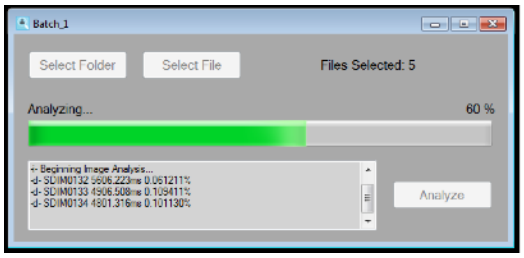
Image files (*.jpg, *.jpeg, *.png)

5. Now you're ready to run your analysis.
6. Click the "Analyze" button to begin the analysis process.
7. Once the job is complete, you can close the window by clicking the red 'X' in the top right corner.

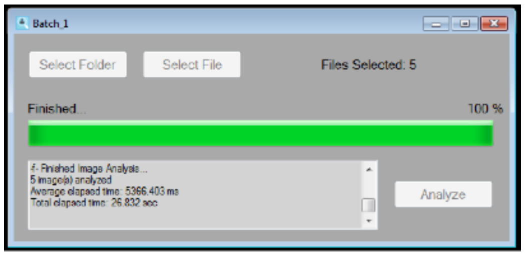




```
-i- Initialization completed
-i- Beginning Image Analysis...
```



```
-i- Beginning Image Analysis...
-d- SDIM0132 5606.223ms 0.061211%
-d- SDIM0133 4906.508ms 0.109411%
-d- SDIM0134 4801.316ms 0.101130%
```

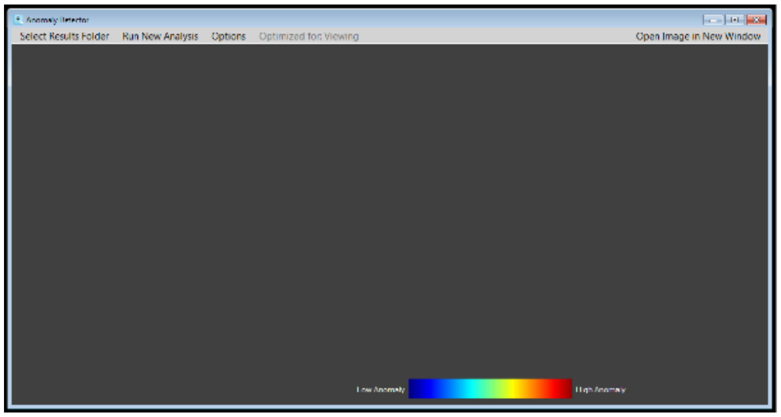


```
-f- Finished Image Analysis...
5 image(s) analyzed
Average elapsed time: 5366.403 ms
Total elapsed time: 26.832 sec
```

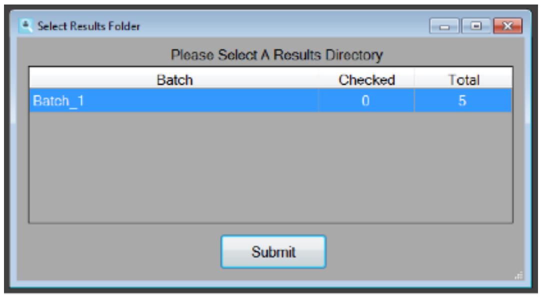
8. You have just completed your first image analysis.
9. The next section will show you how to review the results.

4.2 Viewing Results

1. Click the “Select Results Folder” button at the top left.



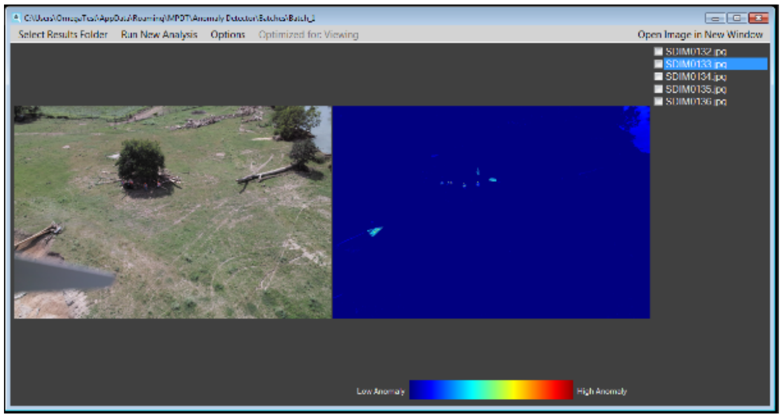
2. From the new window, select the batch, from which you would like to view the results. If there are no batches shown in the list, you will need to run a new analysis job. Please refer to Section 4.1 for instructions.
   Note: You can sort the list by clicking on one of the column headers.
3. Once you have selected a batch to view, click “Submit”. This will open up the analyzed images in a side-by-side view.
4. On the right of the application will be the names of the images that have been flagged as



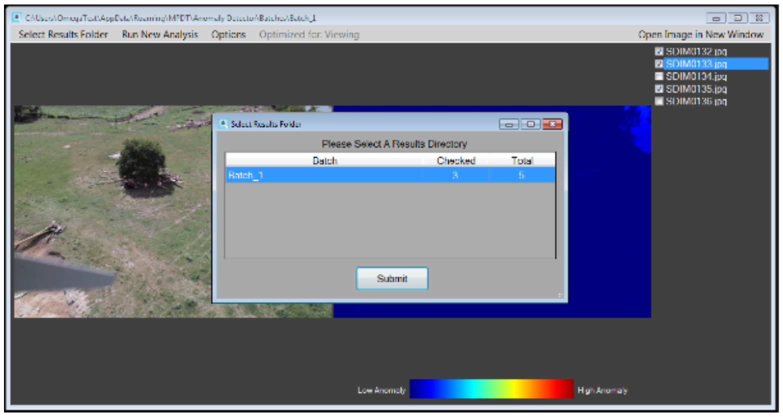
Batch	Checked	Total
Batch_1	0	5

containing features of interest. Images that were below the threshold will not be shown.

5. Click on an image’s name in the list to view it.



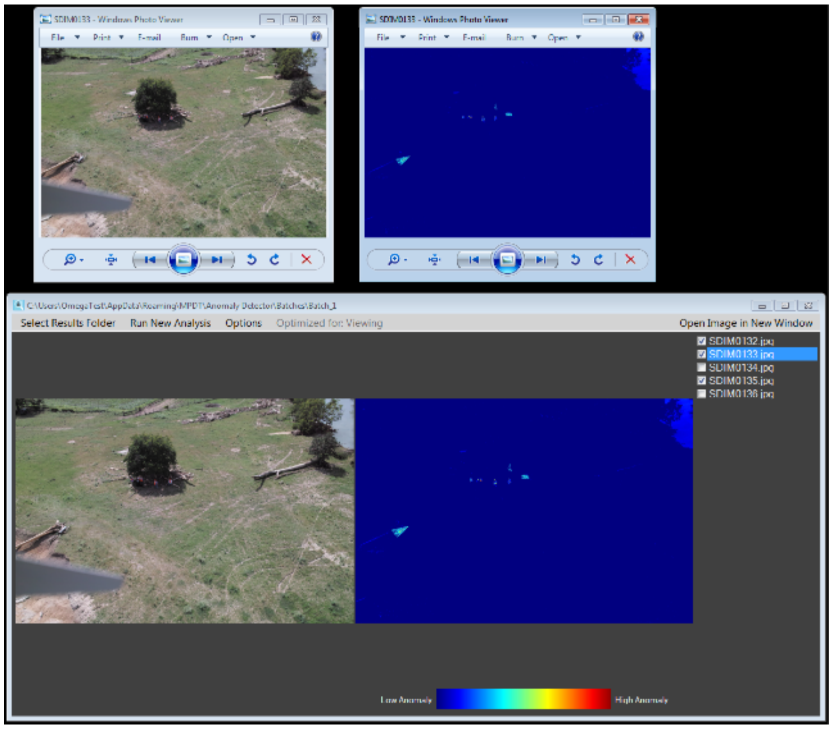
6. You can toggle the checkbox next to an image name in the list by selecting the image name and pressing the Spacebar on your keyboard. The state of these checkboxes will be saved for later viewing sessions. The number of selected checkboxes in a batch can be seen from the batch selection window.
7. In the top right of the application, there is a button labeled “Open Images in New



☑ SDIM0132.jpg
☑ SDIM0133.jpg
☐ SDIM0134.jpg
☑ SDIM0135.jpg
☐ SDIM0136.jpg

Batch	Checked	Total
Batch_1	3	5

Window”. Clicking this button, uses Windows’ default image viewer to open the currently selected images in a new window. This allows you to perform more advanced tasks with each image.

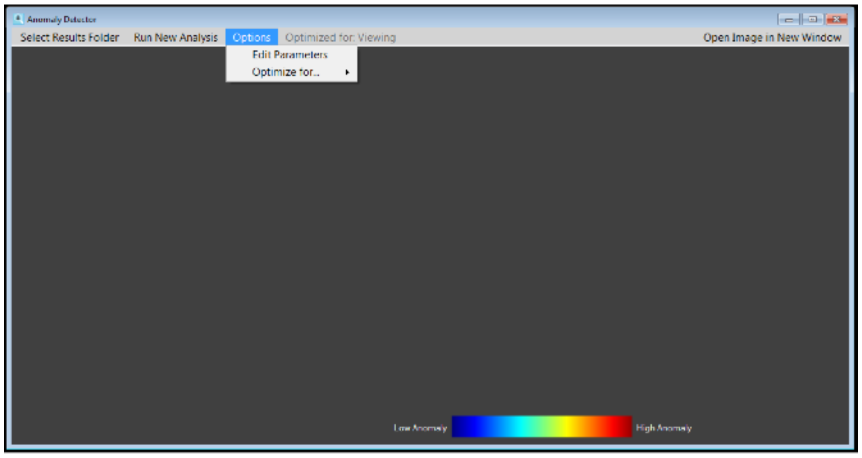


☑ SDIM0132.jpg
☑ SDIM0133.jpg
☐ SDIM0134.jpg
☑ SDIM0135.jpg
☐ SDIM0136.jpg

8. As part of the Installation process, a shortcut to the Batches folder will be created on the desktop to allow you to review images that were not flagged, they will be saved inside the “Other” folder within its respective batch. Otherwise, the Batches folder can be found in your “Documents\Computer Vision Emergency Response Toolkit\Batches” folder

4.3 Advanced Options

1. In the “Options” menu at the top, you can click on the “Edit Parameters” button to change the detection settings. Selecting this option opens up a new window, which allows you to change the parameters each analysis will use.



Below are descriptions of all the parameters and their effect on the analysis:

a. RxThreshold
   i. This is the threshold value a single pixel must meet in order to flag the entire image as containing a feature of interest.
   ii. A lower value will result in more images being flagged.
   iii. Increase this value to make it harder for images to be flagged as containing a feature of interest.
   iv. The lowest accepted value is 0.

b. RxChiThreshold
   i. This value determines the percentage of pixels filtered out.
   ii. The default value is 0.999 (99.9%), which means the top 0.1% of pixel values are allowed through.
   iii. The accepted values range from 0.0 to 1.0.
   iv. Changing this value does not change whether an image is flagged, It only changes how much detail/noise is allowed in the final heatmap. Decreasing this value will make the heatmap look closer to the original image, but may make finding the features of interest harder.
   v. Reducing this value can make it easier to compare the original image and the heatmap image.

c. LineGaussianIter
   i. The kernel standard deviation in both the X and Y directions. If left at 0, these values are calculated by the application and applied automatically. The larger the value, the higher the degree of smoothing when applying a Gaussian Blur on the image. This value should be a positive number. NOTE: this values only applies to the “Line Detection” portion of the algorithm.

d. LineDilationIter
    i. Dilation is used to add an extra layer of pixels on the edges of lines, making them stand out more. This variable determines the number of iterations that the dilation is applied to the image. The higher the number, the more pronounced edges will be. This value should be an integer greater than 0. Also, the higher the number, the longer the application will take to process images. NOTE: this values only applies to the “Line Detection” portion of the algorithm.
e. LineBilatBlurColor
    i. This value filters sigma in the color space. A larger value means that farther colors within the pixel neighborhood (determined by [f] LineBilatBlurSpace) will be mixed together, resulting in larger areas of semi-requal color. The higher the number, the more differing colors are grouped when applying a bilateral filter.
    ii. For simplicity, this value should be the same as [f] LineBilatBlurSpace. If both these values are small (< 10), then the filter will not have much effect. On the other side, if they are large (> 150), the filter will be strong resulting in an image that is cartoonish and thus, edge detection will suffer. Recommended range of values is [25, 125]. NOTE: this values only applies to the “Line Detection” portion of the algorithm.
f. LineBilatBlurSpace
    i. This value filters sigma in the coordinate space. A larger value means that further pixels will influence each other as long as their colors are close enough (determined by [e] LineBilatBlurColor).
    ii. For simplicity, this value should be the same as [e] LineBilatBlurColor. If both these values are small (< 10), then the filter will not have much effect. On the other side, if they are large (> 150), the filter will be strong resulting in an image that is cartoonish and thus, edge detection will suffer. Recommended range of values is [25, 125]. NOTE: this values only applies to the “Line Detection” portion of the algorithm.
g. LineCannyEdgeLowerBound
    i. The first threshold for the hysteresis procedure of the Canny algorithm to find edges. This is the lower threshold.
    ii. If a pixel gradient that is detected is lower than this value, it is rejected. If the pixel gradient is in between [g] LineCannyEdgeLowerBound and [h] LineCannyEdgeThreshold, it is only accepted if it is connected to pixels above the upper threshold, [h] LineCannyEdgeThreshold. If a pixel gradient is above the upper threshold, it is accepted as an edge.
    iii. The smallest value between [g] LineCannyEdgeLowerBound and [h] LineCannyEdgeThreshold is used for edge linking. The largest value is used to find initial segments of strong edges.
    iv. An upper:lower ratio between 2:1 and 3:1 is recommended.

h. LineCannyEdgeThreshold
    i. The second threshold for the hysteresis procedure of the Canny algorithm to find edges. This is the upper threshold.
    ii. If a pixel gradient that is detected is lower than this value, it is rejected. If the pixel gradient is in between [g] LineCannyEdgeLowerBound and [h] LineCannyEdgeThreshold, it is only accepted if it is connected to pixels above the upper threshold, [h] LineCannyEdgeThreshold. If a pixel gradient is above the upper threshold, it is accepted as an edge.
    iii. The smallest value between [g] LineCannyEdgeLowerBound and [h] LineCannyEdgeThreshold is used for edge linking. The largest value is used to find initial segments of strong edges.
    iv. A upper:lower ratio between 2:1 and 3:1 is recommended.

i. CornerGaussianIter
    i. This is the same as [c] LineGaussianIter but for the corner detection portion of the application.

j. CornerErosionIter
    i. Erosion strips the outermost layer of pixels in a structure, making corners more prominent. This variable determines the number of times that the erosion is applied to the image before processing. The higher the number, the more pronounced corners will be. This value should be an integer greater than 0. Also, the higher the number, the longer the application will take to process images. NOTE: this value only applies to the “Corner Detection” portion of the algorithm.

k. CornerBilateralColor
    i. This is the same as [e] LineBilatBlurColor but for the corner detection portion of the application.

l. CornerBilateralSpace
    i. This is the same as [f] LineBilatBlurSpace but for the corner detection portion of the application.

m. CornerMaxDistance
    i. This parameter determines how far the algorithm looks pixel-wise for other corners detected to connect to.
    ii. The lower the number, the less corners will be connected because of their proximity to one another, leading to less features of interest being returned. A value too low will be useless, as corners with correlation will not be connected because they don’t meet the distance required.
    iii. The higher the number, the more corners will be connected because of their proximity to one another, leading to more features of interest being returned. A value too high will be useless, as corners that have no correlation will be connected due to a high pixel distance checked by the algorithm.
    iv. A recommended value is between [25, 100].

    n. CornerNumPoints
        i. This value determines the number of corners that a connected polygon needs to count as a feature of interest.
        ii. A polygon is formed from the connection of corners that are near each other distance-wise, as determined by the value of [n] CornerMaxDistance.
        iii. Only polygons that have a minimum of [o] CornerNumPoints connected will be returned as a feature.
        iv. The lower the value, the more junk is returned because of a low threshold.
        v. The higher the value, the less features of interest will be returned because of the high threshold.

2. Also in the “Options” menu, there is the “Optimize for…” submenu. In this submenu you are able to select from two optimization levels. These tell the program which optimization it should use. The selected option is saved for the next time you open the program.
    a. Selecting “Optimize for...” → “Viewing” will optimize performance for viewing images.
    b. Selecting “Optimize for...” → “Analysis” will optimize the application to get better speed when analyzing images.

Warning: When the “Optimize for Analysis” is selected, the program will use all available CPU resources to perform the analysis as fast as possible. You should expect your system to have a reduced responsiveness while an analysis is running. This option is only recommended if you will not be using the system for anything other than running an analysis. Once the analysis has completed all CPU resources will be freed up and your system will perform as normal. In some instances, attempting to use the system while it is running a job can result in an extreme reduction is system wide responsiveness. In this instance the system may behave as if it has frozen. It is up to the user to decide whether they should halt the analysis or allow it to continue.

5.0 TROUBLESHOOTING

5.1 Running a New Job

1. If The Run New Analysis button is greyed out, this means that the program could not detect an installation of python. Uninstall python and reinstall in the default location. Make sure to check “Add Python 3.6 to PATH” and click “Install Now”.

5.2 Viewing a Job

1. If you are encountering issues with viewing batches, make sure that the “Optimize for Viewing” option is enabled.