

*User*  
*Manual*

# Image Annotation Tool

Jingxiao Ma

## Table of Contents

<b>1.0 General Information.....</b>	<b>2</b>
1.1 System Overview.....	3
1.2 Organization of User Manual .....	3
<b>2.0 System Summary.....</b>	<b>4</b>
2.1 System Information .....	5
2.2 Platform Requirement .....	5
2.3 System Pre-requisite.....	5
<b>3.0 Introduction to User Interface.....</b>	<b>6</b>
<b>4.0 Using the System .....</b>	<b>9</b>
4.1 Open the System .....	10
4.2 Load Image(s) .....	10
4.3 Choose Color .....	10
4.4 Select Region .....	11
4.5 Label / Unlabel Region .....	12
4.6 Flood Fill .....	12
4.7 Output / Save.....	13
<b>[Appendix].....</b>	<b>15</b>
(a) How to Install Required Packages .....	16
(b) Bug Reporting .....	17

## 1.0 General Information

General information section explains in general terms the system and the purpose for which it is intended.

## **1.1 System Overview**

Image Annotation Tool is an application, which allows users to label items in an image. Though it is originally designed to annotate images of plantations, it is applicable to all kinds of images. The system takes a color image (in RGB format) as input, allows users to annotate regions, and generates an image with black background and colorful labels on it.

## **1.2 Organization of User Manual**

The user manual consists of five sections: general information, system summary, introduction to user interface, using the system and appendix.

General information section explains in general terms the system and the purpose for which it is intended.

System summary section outlines the basic information of the application, and the hardware / software requirements.

Introduction to user interface section gives a demonstration of graphical user interface of this application.

Using the system section provides a detailed description of system function.

Appendix provides a brief instruction of how to install required packages to your PC.

## 2.0 System Summary

System summary section outlines the basic information of the application, and the hardware / software requirements.

## **2.1 System Information**

Name: Image Annotation Tool

Version: 1.0

Language: Python 2

GUI tool: PyQt 4

Image processing tool: OpenCv 2

## **2.2 Platform Requirement**

- Windows: Windows XP or later versions
- Mac OS X: OS X 10.8 or later versions
- Linux / Unix

## **2.3 System Pre-requisite**

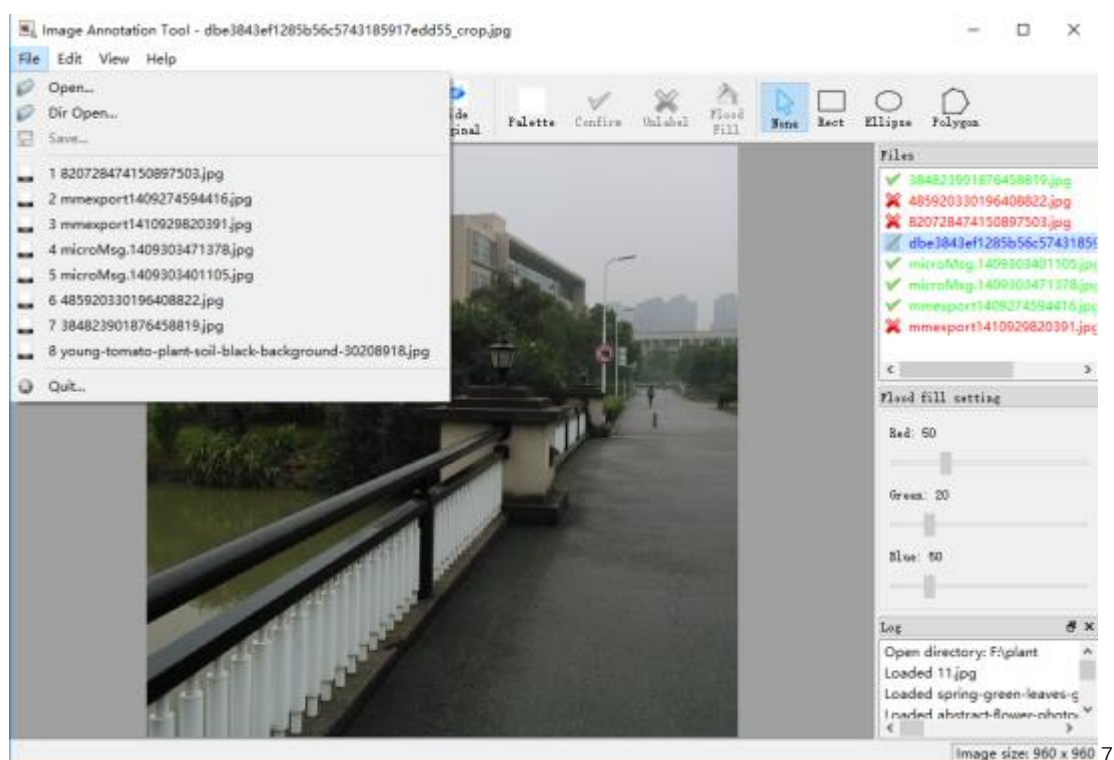
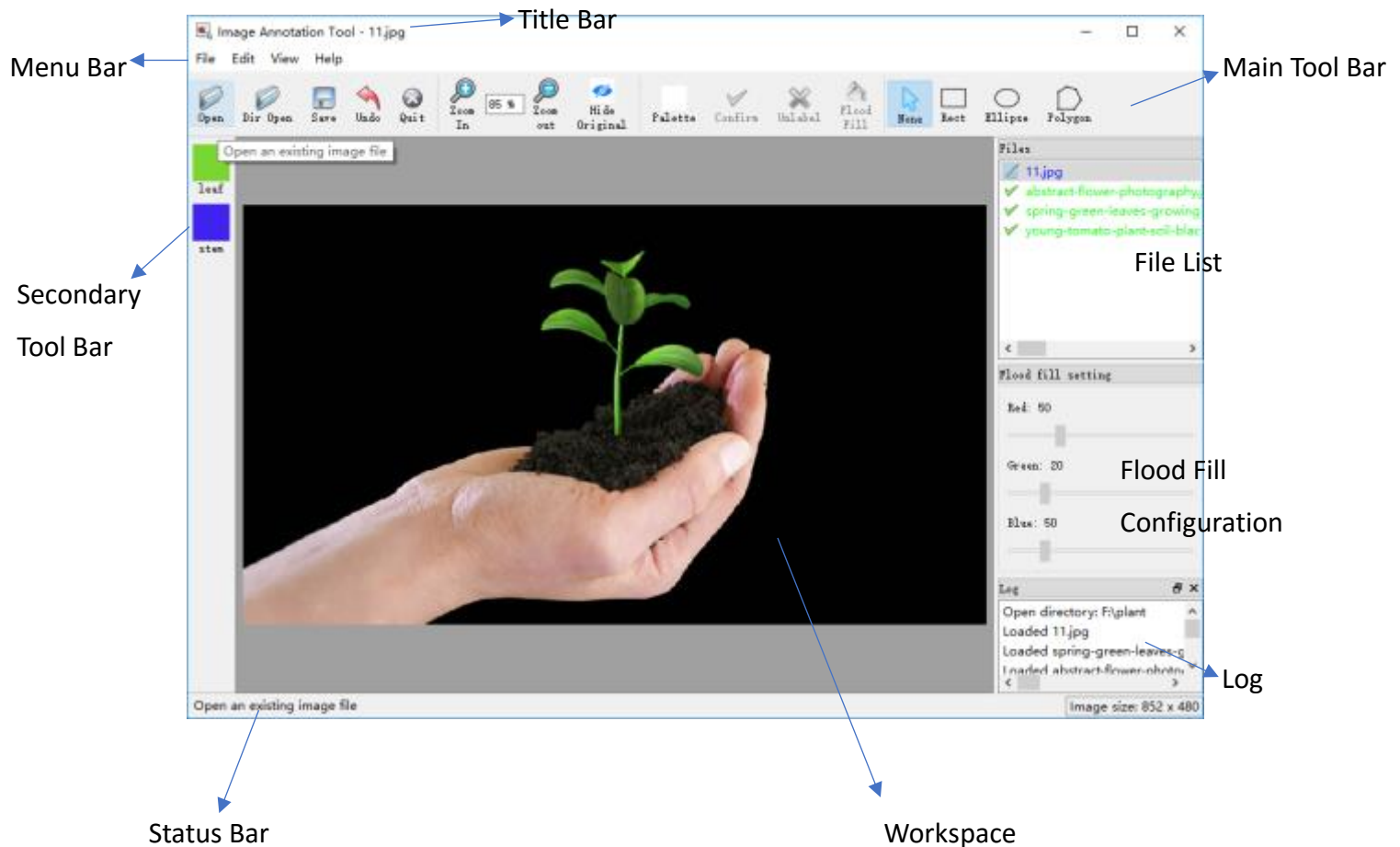
- Python 2: Python 2.3 or later versions, recommend Python 2.7
- PyQt 4: PyQt 4.4 or later versions, recommend PyQt 4.11
- OpenCv 2: Recommend OpenCv 2.4.13
- Numpy

Please check appendix part (a) for more details of installation instruction.

## 3.0 Introduction to User Interface

Introduction to user interface section gives a demonstration of graphical user interface of this application.

The main interface shows below:





The title bar shows the names of application and current opened image.




The menu bar consists of four part: *File*, *Edit*, *View* and *Help*, each of which contains corresponding functionalities. In File menu, there is a recent file list.

The main tool bar provides basic operations over images, including opening, saving, zooming, choosing region, etc.

The secondary tool bar provides a quicker choice of labelling colors, which is specified by users in an external file. We will look at how to define user-specified colors at section 4.3.

The workspace is the central area of this application, which is used to display images and allow users to make annotations. Users can also zoom the images with mouse wheel and dragging in it.

The status bar shows the current operation and the size of current image.

The file list dock provides the list of images in current directory, if you open images with *Dir Open* operation. File name with red color and  icon indicates that the image is not annotated. File name with green color and  icon indicates that the image is annotated. File name with blue color and  icon indicates that the image is the current image.

Flood fill configuration dock provides a user setting of flood fill action. Please check section 4.6 for details.

Log dock keeps a record for each operation that user makes.

One more thing, users are able to adjust the size of application window, workspace, file list dock, flood fill configuration dock and log dock.

## 4.0 Using the system

Using the system section provides a detailed description of system function.

## 4.1 Open the system


To open the system, first make sure that all required packages have been installed. Then open the `cmd` in Windows (or `terminal` in Mac) and use `cd` command to get into the root directory of the application. Type


```
python ImgAnno/ImgAnnoTool.py
```



to open the system.

## 4.2 Load Image(s)

There are two approaches of loading image onto workspace.

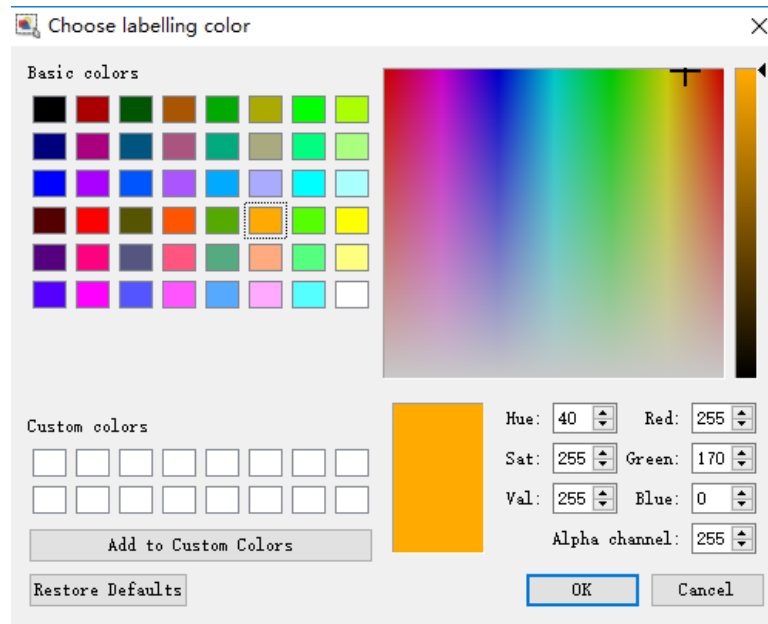
The first one is loading a single image, by clicking  Open or File > Open. Users can choose a single file in all kinds of image formats in the opening file dialog. Then the chosen image will be displayed in workspace.

The second approach is loading a directory, or a root of image files, by clicking  Open Dir or File > Open Dir. Users will see a dialog that allows them to open a directory. After that, all image files in that directory will be loaded to the file list dock on the right side of the interface (for the meanings of colors of file names, please refer to section 3). Users are able to open an image by double-clicking the file name on the file list dock.

After an image is opened, it will be auto-sized according to the size of workspace. Users can also zoom the image by rolling the mouse wheel, clicking  Zoom In or  Zoom Out button, or changing the value of the spin box.

## 4.3 Choose Color

Users are able to choose the colors for different labels. There are two ways of choosing colors.



The first way is using the Palette, which can be found at tool bar or Edit menu. As shown above, users can either choose a color or generate a color by entering RGB values.


Users can also define user-specified color-label pairs on the secondary tool bar, as shown in section 3. They can be defined in the file called `label.txt`, which should be put in the root directory of the image(s). Each line of file defines a color-label pair, which should be in this format:

`50,200,100,Leaf`


First three integers are in the range of  $[0, 255]$ , which accordingly specify the red, green, blue values of the label color. The forth component specifies the name of that label. For the detail of this implementation, please check the function `getLabelColor(imgPath)` in `config.py`.


## 4.4 Select Region


After that, users are able to select the region and manipulate it. This operation is done with the four right-most buttons on the main tool bar.

The button  No Action means not choosing any area. When this button is

checked, the mouse actions are only used to view the image.

The button  Rect is for choosing an rectangle area. When this button is checked, users are able to draw a rectangle by pressing and moving the mouse. The top-right and bottom-left points are specified by users.


The button  Ellipse is for choosing an ellipse area. When this button is checked, users are able to draw an ellipse, which is inscribed to the corresponding rectangle.


Users can also choose an irregular polygon area by choosing the button  Polygon. Users should first choose a starting point of the polygon by left-clicking a point, and then repeat choosing vertices of the polygon by left-clicking points by points. For the final vertex of the polygon, just double-click at that point and finish choosing the polygon area.

To cancel the selection or re-select, you can just right-click at any point of the image.

## 4.5 Label / Unlabel Regions


After choosing the area, users can do three operations to the selected region: label, unlabel and flood-fill. In this section, we will look at the first two operations.

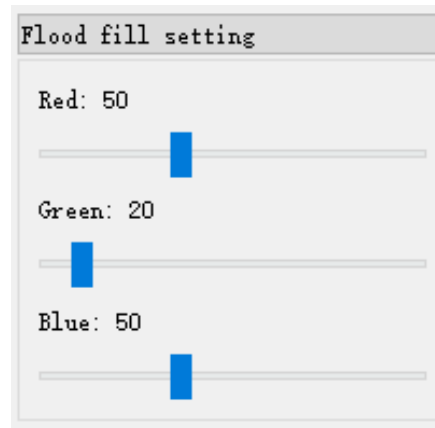
Label action means filling the selected area with the chosen color. Users can do this by clicking  Confirm button after choosing the region.

Unlabel action means cancelling all the labels in the selected area. The area will be colored with background color of output image, i.e. black. Users can do this by clicking  Unlabel button after choosing the region.

## 4.6 Flood Fill

This application also supports flood-fill operation, which means the algorithm that determines the area connected to the given pixel with the similar color values. It can be used to quickly select an item out of the image.


After choosing the region, users can go into the flood-fill mode by checking  Flood Fill button. Then users can press at a point / pixel inside the selected region and the image on workspace will show the result of applying flood-fill algorithms. Users are also able to change result by clicking other points in that region. **Please note that the flood-fill operation will be only performed in the selected region.**



Users can also manually configure flood-fill action, by setting the difference values between the selected pixel (or the seed point) and its connected component. In the “flood fill configuration dock”, users are able to set the difference values, which consist of red, green and blue different values. The less the value is, the more sensitive that color is. This application is designed to annotate plantation images, so the value of green is low while the ones of red and blue are relatively higher (red=50, green=20, blue=50). Users are able to change those value by moving the slider bars. And the result of changing the value will be shown on workspace.

This setting is implemented in the script `ImgAnno/FloodFillConfig.py`.

## 4.7 Output / Save

As mentioned in section 1, the output is an image with black background and colorful labels on it. Users are able to view the (current) output by checking the  Hide Origin button. The workspace will show the output instead of the binding image of input and output.

If users click the  save button, the current generated output will be saved to

the `[root_of_image_file]/Output_Images/` directory. The file name will be the same as original input while the format of image will be `.png` (for the lower level of compression).

The saving directory is defined in the `outputDir()` function and the `outputFile()` function of `ImgAnno/config.py`.

[Appendix]



## (a) How to Install Required Packages

To run the system, several packages have to be installed in PC in advance. In this section, we will mainly focus on how to install those packages on Windows operating system.

First of all, we shall install Python2.7 from <https://www.python.org/downloads/>. Also remember to add it to the System Variable `Path`. Extremely recommend to install pip because you can easily install many packages with the help of pip later on. Download pip from <https://pypi.python.org/pypi/pip>, unzip it, open command line tool `cmd`, use `cd` command to get into the unzipped folder, and then run `python setup.py install` to install pip.

Then you can install PyQt4 package. You can find PyQt4.11.4 from <https://sourceforge.net/projects/pyqt/files/PyQt4/PyQt-4.11.4/>. Find the correct version and download the executable ( `.exe` ) file. Run the installation program and install PyQt4. Note if you want to use higher version of PyQt, you might need to install SIP in advance (from <https://www.riverbankcomputing.com/software/sip/download> ).

After then you can install OpenCv 2 for python. OpenCv for python needs a package named numpy. You can install numpy with the help of pip. Type

```
pip install -U numpy
```

in `cmd`. Then download opencv 2 from <http://opencv.org/releases.html>. Run the installation application and get `cv2.pyd` from

`[installation_directory]/build/python/2.7/x86/cv2.pyd` (or `x64/cv2.pyd` if your system is 64 bits). Copy `cv2.pyd` into

`[python_installation_directory(normally C:/Python27)]/Lib/site-packages`. Then you should be able to successfully run the system according to the section 4.1.

If you want to do some unit tests, download nose by the command

```
pip install -U nose
```

If you want to install this application to the python repository (e.g. C:/Python27), download setuptools by the command

```
pip install -U setuptools
```

For Mac users, all those packages will be easily installed with the help of homebrew. For the details of homebrew, please check <https://brew.sh>.

## **(b) Bug Reporting**

If anyone has technical issue or spots a bug, please contact Jingxiao Ma by emailing [psyjm7@nottingham.ac.uk](mailto:psyjm7@nottingham.ac.uk).