

# Image Analysis for Plant Communities

February 10, 2016

Notes for developing plant categorisation for bee-communities using image analysis and FIJI.

## 1 Technology set-up and other stuff

These are just a few notes for you to think about. Open source packages are ALWAYS more flexible because they are designed to be. It's good to familiarise yourself with them now if you can.

### 1.1 Operating system

Linux is much more reliable than Windows, if you can stretch yourself, then covert all your systems over. Linux comes in a range of 'flavours' but Ubuntu is a common distribution. Most stuff that works on Windows will work with Ubuntu (<http://www.ubuntu.com/>).

### 1.2 WEB search engines

Google does not seem as reliable as it once was (e.g. see duckduck-go-6.pdf attached). Search results are less likely to be skewed with an engine like **DuckDuckGo** (DDG). Even better, you can be very specific about what you want with DDG by adding things like **f:pdf 'what ever your search is here'**. You can read more about it from the following sites:

1. Syntax <https://duck.co/help/results/syntax>
2. Extras <http://www.makeuseof.com/tag/6-really-cool-things-you-can-do-with-duckduckgo/>

There are some other good search engines as well (e.g. Ixquick). Really, the key is to try and be flexible; get used to using other types of search engines/ software and devices if you can?

### 1.3 Document processing

When you have a lot of images in your document - then Latex is a much better option for writing up. It can be a little difficult to get you head around initially, but at the end of your write-up you will appreciate the effort. TeXstudio is a good platform (<http://www.texstudio.org/>). This document was written in TeXstudio, all the files are included in the subfolder tex-doc.

### 1.4 Data entering/analysis and sharing

Apache open office (entering data) and R-Studio (analysis) work well together - both integrate with TeXstudio. For example, you can export an Apache document (.odt) to a .tex file for opening in TeXstudio.

GitHub is an easy way to share large amounts of *image* data online. You will need a repository manager. I used SmartGit, which has an easy to use desktop GUI. You can manage you GitHub account (e.g. pushing and pulling information from you PC to the online GitHub account) and keep data synchronised.

1. Apache Open Office <http://www.openoffice.org/>
2. RStudio <https://www.rstudio.com/>
3. SmartGit <http://www.syntevo.com/smartgit/>

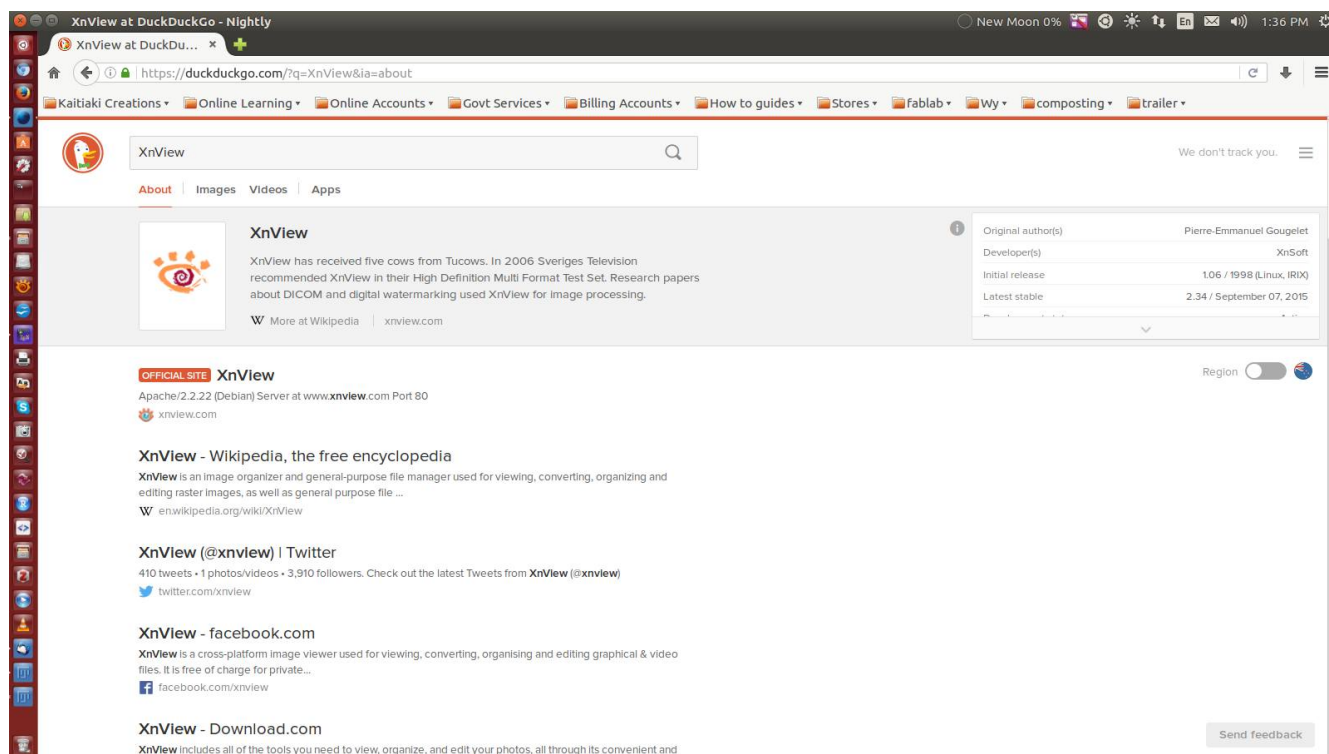


Figure 1: DuckDuckGo and XnView

## 2 File and image data backup software

Image files can be time consuming to back-up or copy, unless your using a good package. Grsync (<http://www.opbyte.it/grsync/>) is easy to use and once set up you can quickly synchronise data between drives.

## 3 Select and download image management/pre-processing software

Here are three. I tend to use XnView, especially for file management, naming, re-naming, cropping etc.

- XnView <http://www.xnview.com/en/> (see figure 1 below)
- GIMP <http://www.gimp.org/>
- InkScape <https://inkscape.org/en/>

## 4 Download ImageJ/FIJI imaging software

Download FIJI from here: <http://fiji.sc/Fiji>. Follow installation instructions.

## 5 General image file handling procedures

You won't need this if you only have a few image files. When processing images, data gets large quickly. It can be very difficult to manage properly; these are just some suggestions to help.

- Import raw image data.
- If possible, save to 3x backups (e.g. 2x external HD, 1x internal partition disk) using Grsync (see section 2 above).

- Set-up a *working folder* structure. Try to anticipate some of the processing steps. Name the folders accordingly. For example, folder names could be: raw-images, sorted-date-location-images, rename, crop, normalise, segmented, etc...Once the structure has been decided, you can shorten the folder names.
- Process a sample of images as per their folder ID - complete an entire run of the processing cycle:
  - Step 1 FOLDER rename - rename files in Xnview.
  - Step 2 FOLDER crop - crop images to a grid.
  - Step 3 FOLDER normalise - set the saturation level of images.
  - Step 4 FOLDER segmented - classified images. See [section 7](#) below.
  - Step 5 FOLDER binary - convert classified images into binary images and post-process.
  - Step 6 FOLDER analysis - count the number of, or assess the area of, target objects.
- Repeat and refine the procedures. Check that results make sense.

## 6 Verifying image analysis results.

- The most reliable/straightforward way to do this, is to manually estimate image quantifications. For example, if you have counted five objects using the automatic classification method, then you should be able to count five objects in the images by manual inspection.

## 7 Procedure for Trainable WEKA Segmentations (TWS)

1. Open FIJI.
2. Load image.

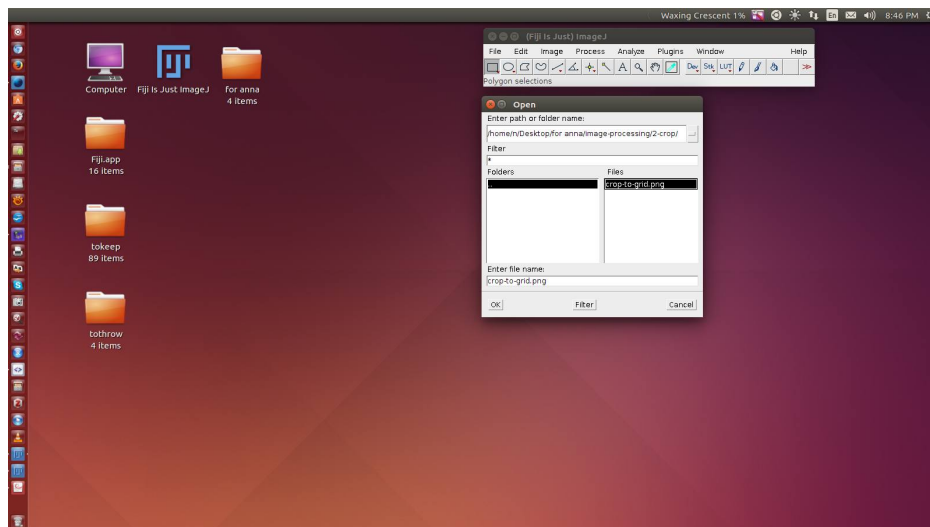


Figure 2: TWS Open file

### 3. Load TWS plug-in.

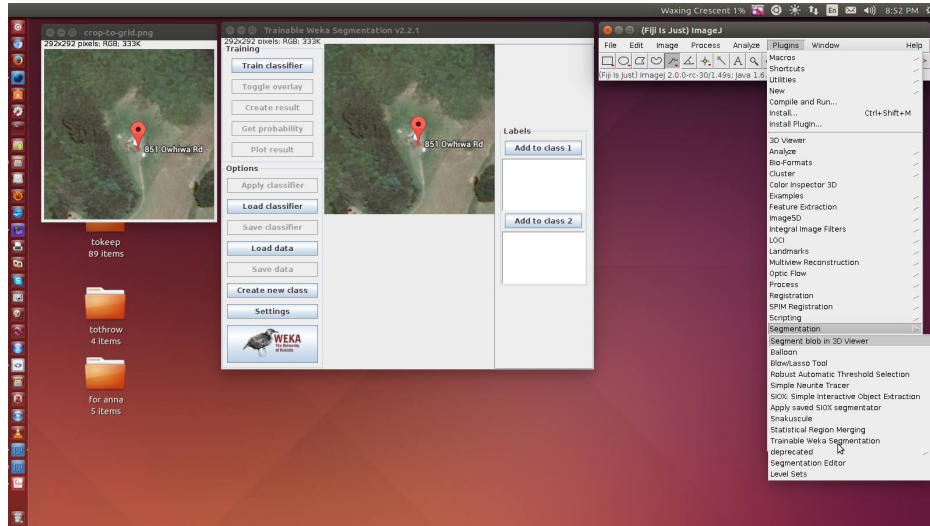


Figure 3: Load TWS plug-in.

### 4. Set-up initial training.

- A. De-select training features.
- B. Set: Gaussian blur, Mean, Minimum, Maximum, Medium for initial runs.
- C. Set Maximum sigma to 4.
- D. Set Random Forest paramters to 50 trees, 7 random features.
- E. For example, the Classifier options for this should now be: `hr.irb.fastRandomForest.FastRandomForest -I 50 -K 7 -S -188830655 -threads 2`.

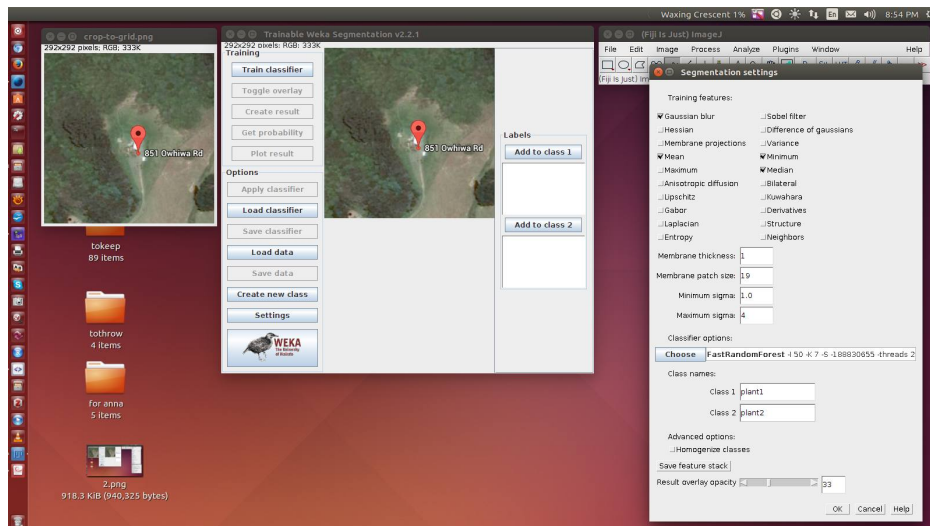


Figure 4: Set-up training and classifier parameters.

5. Add more categories and supply trace samples.
  - A. There must be *at least* one region of interest (RIO) example representing the target and background.
  - B. Use any of the ROI tools to add samples.
  - C. When providing ROI samples, LESS is more - the objective is to train the machine learner.
  - D. The algorithm needs to learn from a few *key* samples. Otherwise, it is memorising your traces and regurgitating.

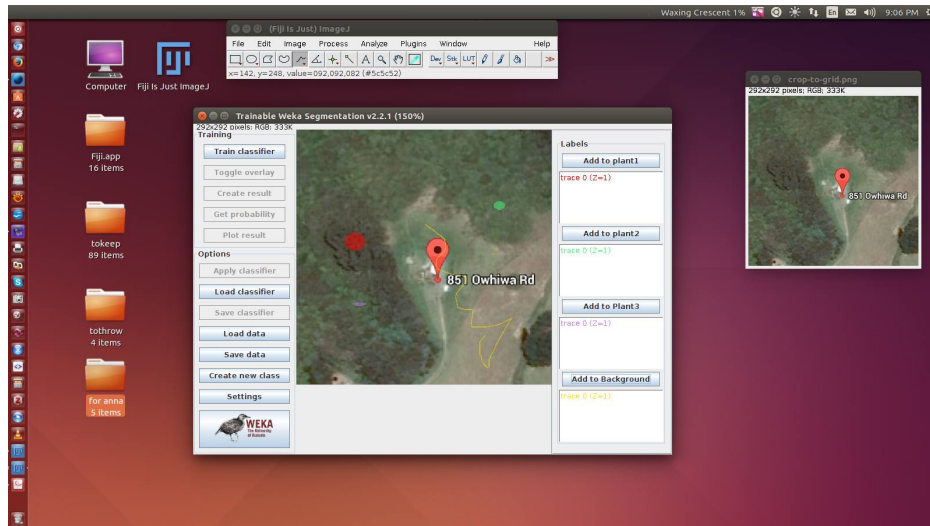


Figure 5: Add more categories if required. Use the ROI tools to add traces representing the target areas.

6. Train the initial classifier.

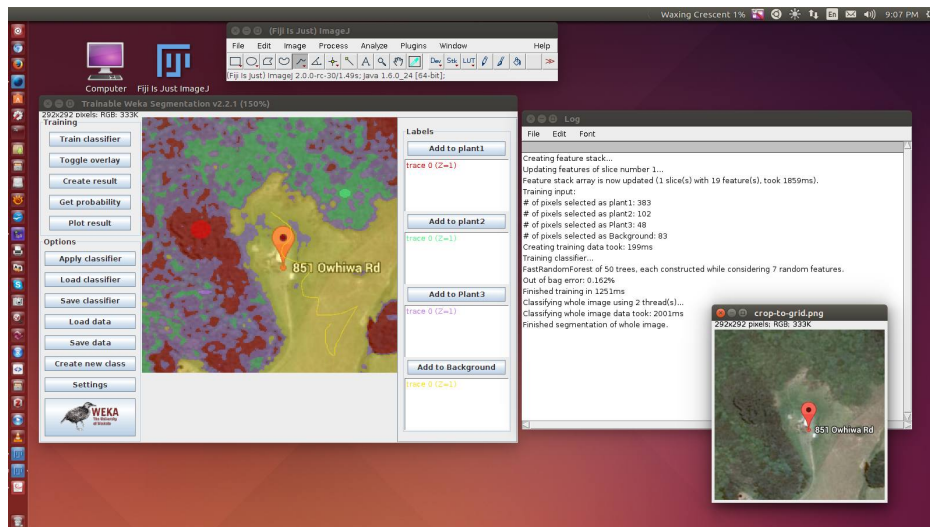


Figure 6: Initial classifier training and results.

7. Check results and refine classifier.

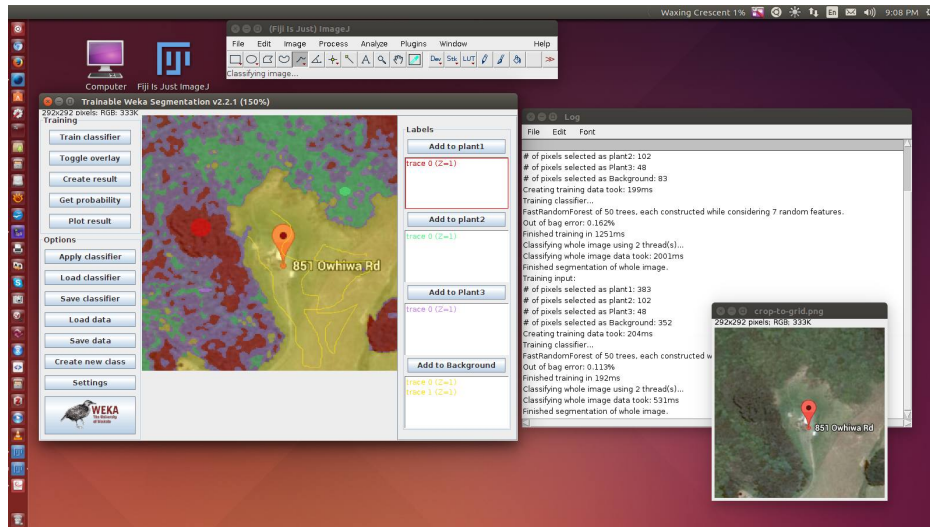


Figure 7: Final classifier training and results.

8. Save results - classified image, probability maps, data, model and log files (as per your folder set-up).

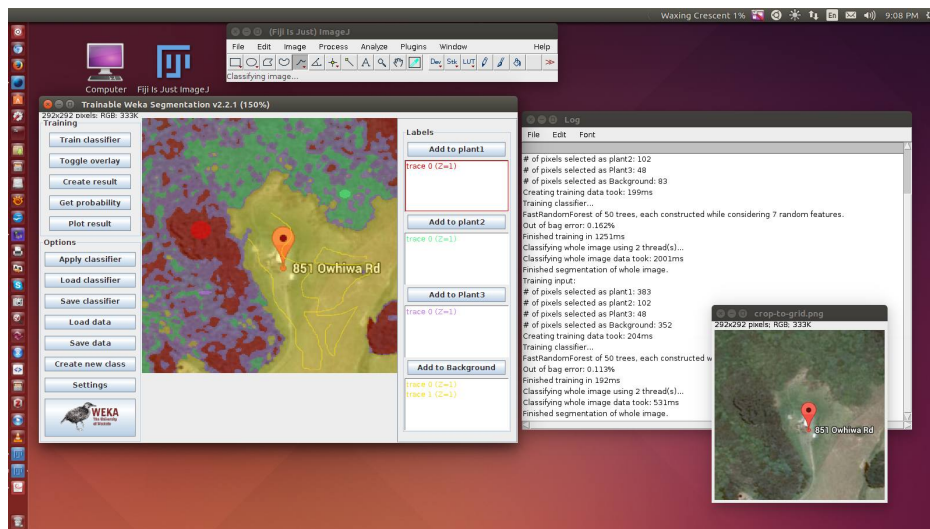


Figure 8: Save data.

## 8 Post-processing

See FIJI or ImageJ sites for methods to quantify data from binary images. This step can be refined, 1) when your more sure about what you need, 2) and as you become more familiar with the techniques. I've included some literature that might be helpful.