

# Ilastik for the Segmentation of Histological Section Images

## Contents

1. Introduction - what is ilastik? .....	2
2. General Information .....	2
3. Analysis approach for series of rodent brain section image .....	3
4. Pixel classification workflow .....	4
5. Object classification workflow .....	7
6. FAQ and Troubleshooting .....	8
Which pixel classification features should I select? .....	8
Which images should I upload in the Input Data applet? .....	8
What are training images? .....	8
How many classes should I use? .....	8
Which part of the image, and how much, should I label? .....	9
How do I test the applicability of the trained classifier to the whole series? .....	9
Which export settings should I select? .....	9
My output images are black (or white). What's gone wrong? .....	10
Help! Ilastik keeps crashing. I have very large images. What do I do? .....	12

## 1. Introduction - what is ilastik?

---

- Ilastik is a versatile image analysis tool specifically designed for the classification, segmentation and analysis of biological images based on supervised machine learning algorithms.
- It can be used on its own or in combination with other image analysis software, for the effective analysis of large numbers of images.
- Ilastik has many functions, packaged into 8 unique workflows. However, as this manual describes the use of ilastik for the analysis of series of images of labelled rodent brain sections, only workflows relevant to this process will be mentioned.

## 2. General Information

---

The ilastik program can be downloaded here: <http://ilastik.org/download.html>

For online user instructions see: <http://ilastik.org/documentation/index.html>

Note that the user instructions are written for an outdated version of ilastik. This can be confusing, but the general principles have not changed.

All the workflows, with the exception of the density counting workflow, build on pixel classification. The pixel classification workflow is therefore the starting point for most analyses.

### 3. Analysis approach for series of rodent brain section image

---

There are two main approaches for the analysis of rodent brain section images.

1. Pixel classification only (with two or more classes)
2. Pixel classification with two classes (*immunoreactivity* and *background*), followed by object classification with two classes (*objects-of-interest* and *artefact*).

#### Which approach is best for my dataset?

- The first approach is easier and quicker, but does not always give the best result.
- The second approach is more time consuming, but often gives a better result. This is not always the case however, and will depend on the appearance of the labelling in the section images.

As a general rule, pixel classification is suitable for images in which there are clear differences in the colour, intensity and/ or texture of the feature-of-interest versus the background and other structures. If there is non-specific labelling in the image that is very similar in appearance to the labelling-of-interest, object classification may allow the non-specific labelling to be filtered out based on object level features such as size and shape. The best approach is determined by trial and error.

## 4. Pixel classification workflow

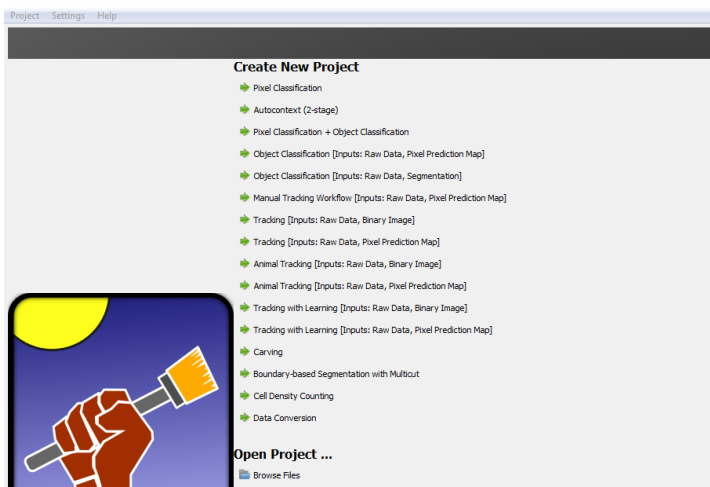
---

For a quick introduction, watch:

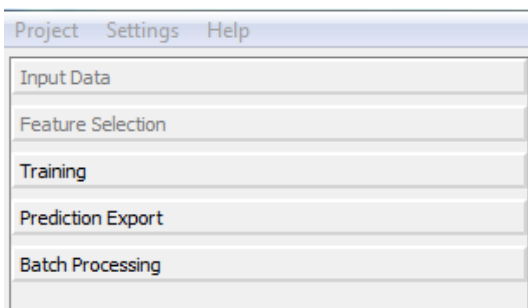
<https://www.youtube.com/watch?v=5N0XYW9gRZY&feature=youtu.be>

### 1. Open the ilastik programme.

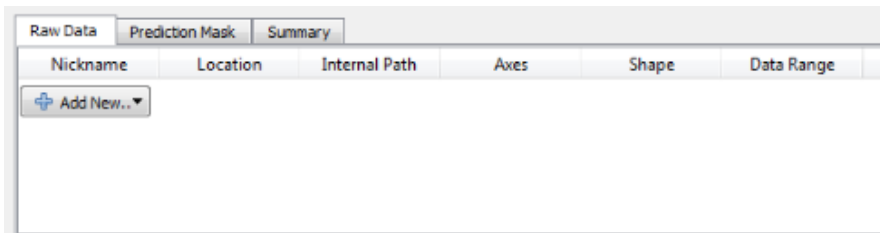
Under ‘Create New Project’ select ‘Pixel Classification’. Save the project under a new file name in the same location as the images for analysis (create a new folder).



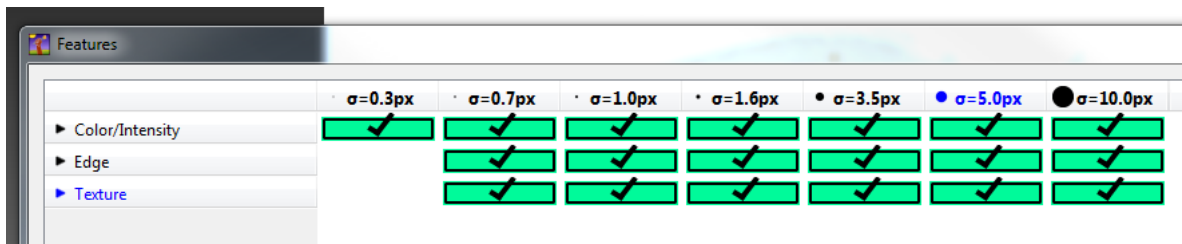
### 2. On the left hand side of the screen there are five input applets.



In the **Input data** applet, select 'Add New'. Add one or more images for the purpose of training the classifier (*training images*). Convert the format of the images to HDF5 to increase the processing speed. To do this, highlight the uploaded images, select 'storage' and change from 'relative link' to 'copied to project file'. Save the project.



3. Select the **Feature Selection** applet and click 'Select Features'.



Select the features and scales that can be used to discern the objects or classes-of-interest: for most datasets, all the features should be selected. [See FAQ for advice on selecting good features.](#)

4. Select the **Training** applet. To scroll around the image, press **shift** and use the mouse wheel to navigate. To zoom, press **ctrl** and use the mouse wheel to zoom in and out. Click 'add label' to create two or more classes. [See FAQ for advice on the number of classes to use.](#)
5. Label some example pixels of each class with the paintbrush, and remove labels with the eraser. Select 'live update' to begin the machine learning and prediction process. Turning on the *uncertainty* overlay, by clicking on the uncertainty eye, will help in the labelling process as it identifies pixels of which ilastik is unsure of the class. By correctly labelling these pixels, the prediction rapidly improves. [See FAQ for advice on placing labels.](#)
6. The 'probability' and 'segmentation' overlays should be turned on to inspect the final result.
7. On completion of training, select the **Prediction Export** applet. Export the probability maps in HDH5 format, and simple\_segmentation images in .png format, with the default settings.

Do not alter the export location. The files will automatically save in the same location as the input files. Click on the export button to export the image.

8. For batch processing of images with the trained classifier, select the **Batch Processing** applet. Upload the images to be analysed, and select 'process all files'. The time taken to process the files will depend on the size and number of files selected.
9. Save the ilastik file before closing.

NOTE: Save the ilastik file frequently during the annotation process.

## 5. Object classification workflow

---

1. There are three options on the ilastik start up page for running Object Classification. Choose the *Object Classification with Raw Data and Pixel Prediction Maps as input*. It is not advisable to use Pixel Classification + Object Classification. This file type has a tendency to corrupt.
2. Save the object classification file in the same folder as the raw images for analysis. If the images are moved after the ilastik file is created, the link between the ilastik file and the images may be lost, resulting in a corrupted file.
3. In the **Input Data** applet, upload the original images and their respective probability maps in HDH5 format (output from the pixel classification).
4. In the **Threshold and Size Filter** applet, select:
  - The *simple* method.
  - The input channel that corresponds to the label of interest.
  - Smoothing factor for the x and y axis. In general, the same value should be selected for each. Determine the most appropriate factor by trial and error: the goal is to achieve object shapes that are most representative of the real data. Zero is often advisable – in which case no smoothing filter is applied.
  - Threshold. The probability threshold can range from 0 to 1: with zero representing no exclusion of pixels; and 1 representing exclusion of all pixels except those with 100% probability of belonging to the class-of-interest. In reality only the pixels that were manually annotated in the Pixel Classification workflow have a 100% probability of belonging to the class-of-interest. A good compromise is 0.4.
5. In the **Object Feature Selection** applet, select all the features (except those relating to location within the image).
6. In the **Object Classification** applet, create two classes (label and artefact) and label some example objects of each class. Tick the live update box. Continue annotating until you are happy with the predictions.
7. In the **Object Information Export** applet, export “Object Predictions” in PNG format. Do not change the default export location.
8. For batch processing, use the **Batch Processing** applet. Upload the raw images and corresponding prediction maps and “process all files”.

## 6. FAQ and Troubleshooting

---

### Which pixel classification features should I select?

The features and scales to select are those that distinguish the different classes in the image. As it is not always obvious which features and at what scales the classes are distinguished, it is best to select all the features and scales for the pixel classification in the first instance. Selecting fewer features and scales may speed up the analysis process, so refining the selection may be helpful at a later stage.

Note that the scale corresponds to the pixel diameter of the feature. For example, if a textural pattern has a pixel size of 4, the scale of the texture has a sigma of 4. As the maximum scale of the features available in ilastik is 10, ilastik is not capable of recognising objects based on edge if the objects are larger than approximately 60 pixels (the whole object should be visible in 10 x 10 pixel window).

### Which images should I upload in the [Input Data](#) applet?

Only training images should be uploaded in the **Input Data** applet (~10 is good).

### What are training images?

Training images are a subset of the whole image series that are annotated in the training phase. Choose images that contain labelling that is representative of the labelling in the whole series. It is good to select images that span the full volume, as labelling often varies in different anatomical regions (for example, every 4<sup>th</sup> section). The same subset can be used for the pixel and object classification workflows.

### How many classes should I use?

The number of classes to annotate will depend on the classification approach. See section 3 for a description of the two approaches.



- For Pixel Classification only, create 3 or more classes, depending on the number of visually distinct classes. For example: labelling, non-specific labelling, artefact (e.g. marks on the coverslip, etc) and background.
- For Pixel Classification with Object Classification, annotate two classes in each classification step (see section 3).

### **Which part of the image, and how much, should I label?**

Start by zooming-in and annotating a few pixels of each class that clearly belong to their respective class. Turn the 'live update' on to visualise the predictions. The 'uncertainty' overlay can be switched on to identify pixels with uncertain class prediction (it identifies these pixels in bright blue). By specifically annotating these pixels, the prediction quickly improves.

Note that even just a few pixels incorrectly annotated can disrupt the prediction. If in doubt, it is better to delete annotations and start again, rather than continuing with the annotation. By ticking the 'segmentation' box you can visualize the final segmentation based on the classifier. When you are happy with this, stop annotating and test the applicability of the trained classifier on the next training image (see the next FAQ).

### **How do I test the applicability of the trained classifier to the whole series?**

To test the ability of the trained classifier to segment a new image, select 'current view' in the Training applet and choose a new training image from the drop-down menu. Press 'live update' and view the 'segmentation' overlay. If you are not happy with the classification you can annotate pixels on the new image to improve the prediction. When happy with the result, the trained classifier can be tested on a third image. Continue this processes until you are satisfied that the classifier is optimally trained for the image series. You are now ready for batch processing.

### **Which export settings should I select?**

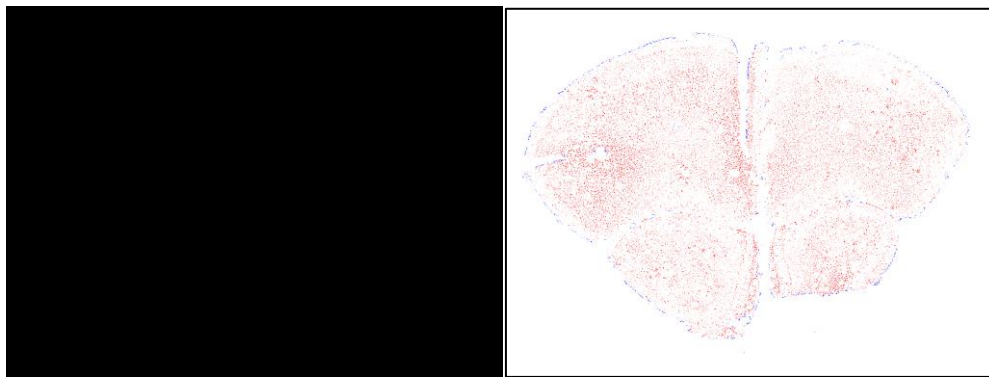
The file type to export will depend on the plan for the next step of analysis.

- In the Pixel Classification workflow, export Simple\_Segmentation.png to visualise the segmentation, or Prediction\_maps.H5 to continue with Object Classification.
- In the Object Classification workflow, export Object\_Predictions.png.

- The PNG images should be export as unsigned 8-bit images.
- Do not alter the output location. The default export location is the folder in which the ilastik file is located. If the output location is altered, the file will fail to export. This is a bug in the system!

### My output images are black (or white). What's gone wrong?

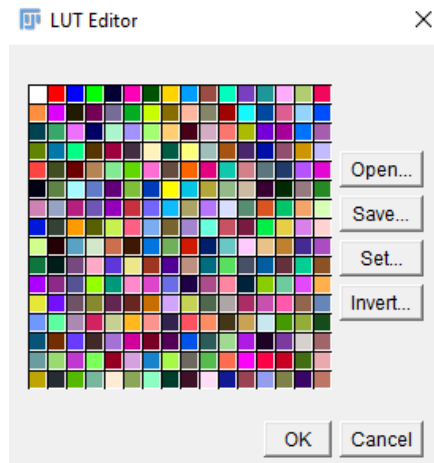
Nothing has gone wrong. Simple\_Segmentation and Object\_Prediction .png images are always black in appearance (the classes are displayed in black by default).



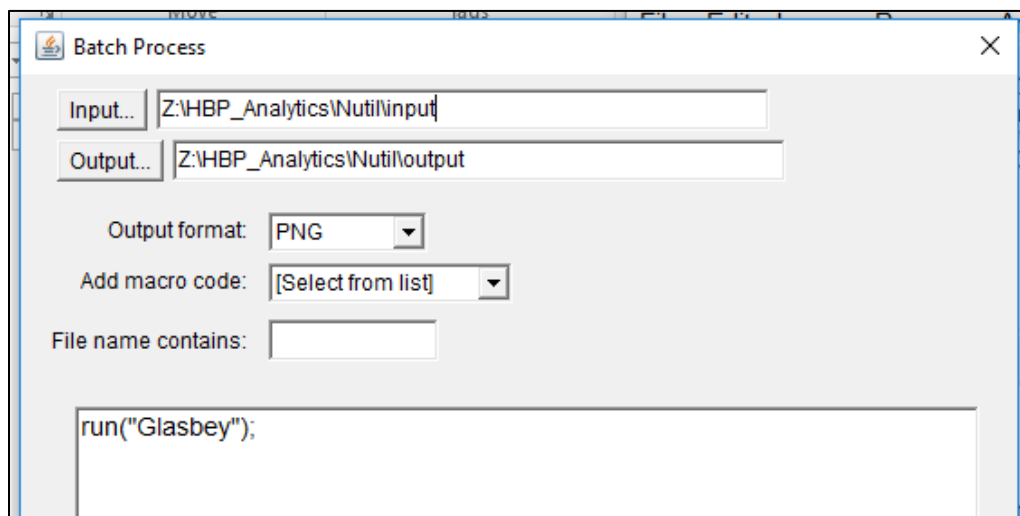
**Figure showing ilastik output with (right) and without (left) LUT applied.**

To visualise the segmentation, edit the image with NIH ImageJ in a post-processing step. Apply the Glasbey lookup table by selecting **Image > Lookup Tables > Glasbey**. This assigns a different colour to each channel.

To alter the colour selection, select **Image, Color, Edit LUT**. The LUT applies colours from the top left hand corner (first colour is ignored). Click on each panel to change the colours. Then Save the customised LUT.



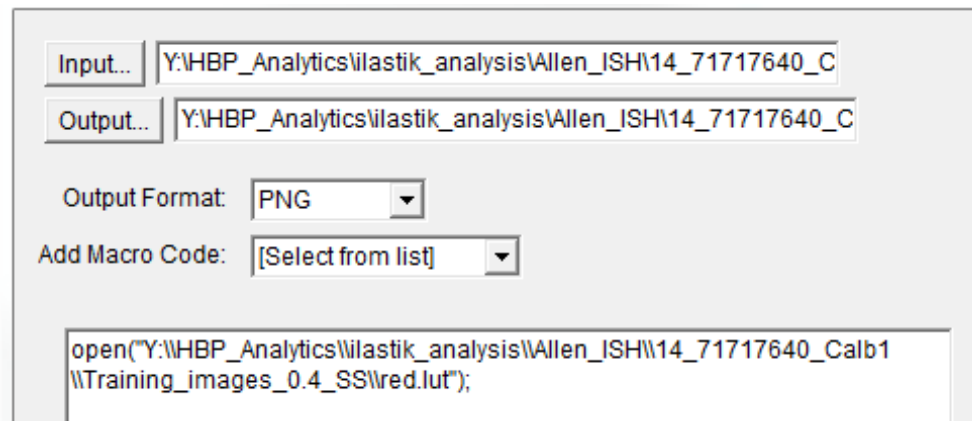
To apply the Glasbey lookup table to a whole folder of segmented images, select **Process > Batch > Macro**; select the input and output folders, required file type, and type the following code in the macro box: `run("Glasbey");`



To apply the customized LUT to a folder of images, first save the customized LUT as a .lut file. Apply to a whole folder of images with the Batch Processing feature. Select: **Process > Batch > Macro**. Select the input and output directories and the file type, and type the following macro:

```
open("C:\\.....\\...\\...\\filename.lut");
```

*Note: Make sure to update the directory so it locates the customized.lut file and ensure the macro is written with double back slashes.*



The screenshot shows the ilastik software interface. It has a light gray background. At the top, there are two text boxes: 'Input...' and 'Output...'. Both contain the path 'Y:\HBP\_Analytics\ilastik\_analysis\Allen\_ISH\14\_71717640\_C'. Below these, there is a dropdown menu for 'Output Format' set to 'PNG'. To the right of this is another dropdown menu for 'Add Macro Code' with the text '[Select from list]'. At the bottom, there is a larger text box containing the macro code: `open("Y:\HBP_Analytics\ilastik_analysis\Allen_ISH\14_71717640_Calb1\\Training_images_0.4_SS\red.lut");`

### Help! Ilastik keeps crashing. I have very large images. What do I do?

- Whilst ilastik has the computational power to process very large images, the viewer in the ilastik user interface is not able to process whole images that are very large in the “live” mode. For large images in the training phase, it is therefore important to remain zoomed-in in the viewer when the live update is switched on. This is especially true if many classes are labelled and many features selected. As a general rule of thumb, keep the portion of the image that is visible in the viewer to below 3000 x 3000 pixels. The absolute value will depend on the number of classes and features selected.
- For very large images, be more selective with the features for classification, and label as few classes as possible.
- If all else fails, it is possible to split large images into tiles and process tiles individually.
- Note that exportation of the segmented images will take time. One large image (e.g. 30,000 x 30,000 pixels) may take 2 hours to export. Image analysis can be run overnight in the batch mode.