Image Segmentation and Analysis Using CellProfiler, Ilastik, and ImageJ Softwares.

Introduction

In this study, the segmentation of images of the **substantia nigra**, a region of the brain crucial in Parkinson's disease research, is achieved by designing a robust pipeline using **Ilastik**, **ImageJ**, and **CellProfiler**. This pipeline is specifically designed to improve the quality of binary images and facilitate accurate **Region of Interest (ROI)** calculation, such as the counting of cells within these regions. The goal is to enhance the precision of image analysis to support research in Parkinson's disease, where accurate quantification of cellular structures is vital for understanding disease progression and pathophysiology.

1.  **ImageJ** is a versatile image processing software that has become a standard tool in many scientific fields. While not inherently deep learning-based, ImageJ is highly extensible with plugins, enabling researchers to incorporate deep learning techniques and custom workflows for advanced image analysis.

2.  **Ilastik** offers a user-friendly interface for pixel and object classification, allowing researchers to apply both traditional machine learning and deep learning workflows to image data. It is particularly effective in segmenting complex biological structures, providing flexible options for different data types.

3.  **CellProfiler** is an open-source software primarily designed for automated analysis of cell images, allowing users to segment and extract quantitative data from large-scale image datasets. Although it primarily uses traditional image processing methods, it can integrate machine learning techniques to enhance its capabilities.

OVERVIEW OF STEPS

- **First, we will utilize the training capabilities in Ilastik to perform proper segmentation and masking of cells**. By manually annotating a subset of images, Ilastik will learn to distinguish between different cell structures and background, enabling the software to accurately segment and mask the cells in subsequent cells in image .
- **Next, we will export the segmented image to the desktop and use ImageJ to convert the segmented masks into binary masks.** In ImageJ, the segmented regions will be transformed into binary images, where the foreground (cells) is represented by one value (e.g., white) and the background by another (e.g., black)
- **Lastly, we will export the binary mask to CellProfiler, where we will set the appropriate diameters and thresholds for the segmentation.** By configuring a pipeline in CellProfiler, we will identify and count the ROIs (regions of interest), such as the number of cells, and generate a spreadsheet with the quantitative data

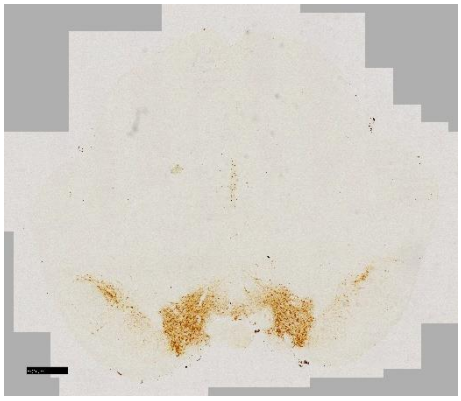
ORGANISING DATASETS

Firstly we will take 4 datasets images of substantia niagra and store it in a folder name **datasets** in desktop

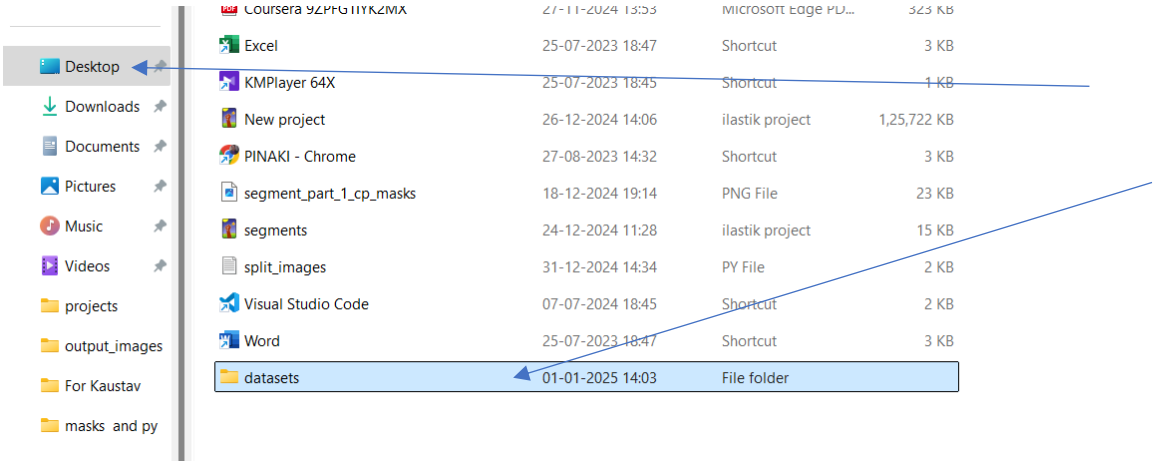
A

B

C



D



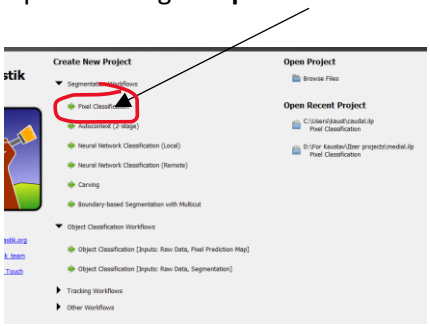
IMAGEJ (OPTIONAL)

For initial preprocessing(cropping) of image we use image j .

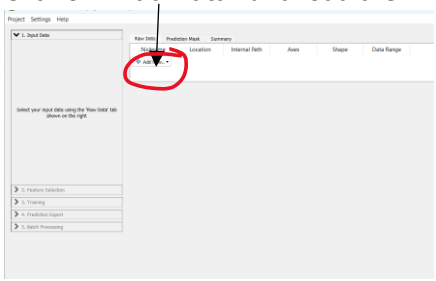
1. **Open the Image:**
   - Launch **ImageJ**.
   - Open your image by selecting **File > Open** and choosing the image you want to crop.
2. **Select the Area to Crop:**
   - Choose the **Rectangle Tool** from the toolbar (default tool).
   - Click and drag to define the rectangular area you want to crop.
3. **Crop the Image:**
   - Once the desired area is selected, go to **Image > Crop** from the menu. This will crop the image, keeping only the selected area.
4. **Save the Cropped Image with the Name "datasets":**
   - Go to **File > Save As**.
   - Choose the file format you wish (e.g., TIFF, PNG, JPEG).
   - In the save dialog, navigate to the location where you want to store the cropped image.
   - Enter **"datasets"** as the file name (without quotes).
   - Click **Save**.

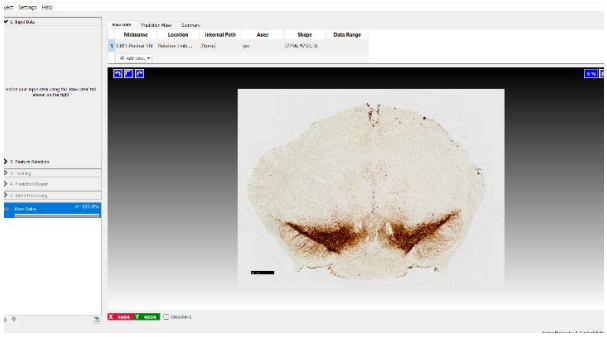
ILASTIK

- Open Ilastik go to **pixel classification** name **project , save.**



- Click on **"Add Data"** and load the image from datasets file that contains the cells you want to annotate.

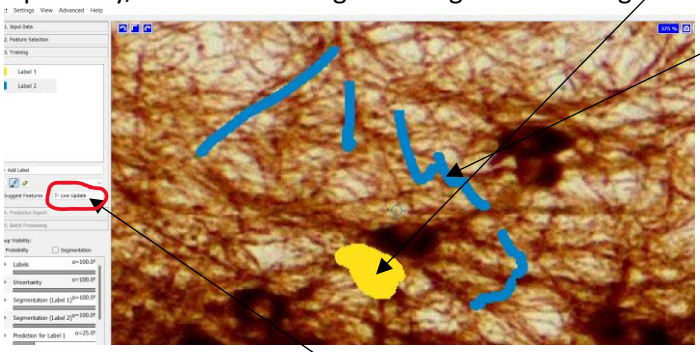




- Image will get loaded in dialogue box
- Click on **feature selection** and select all the cells and click on **OK** .

2. Annotate Sample Cells

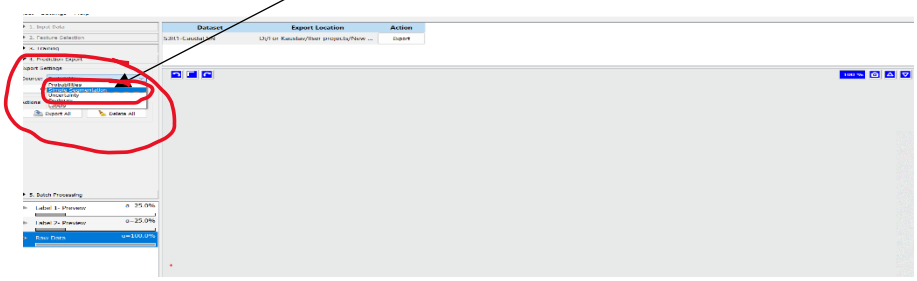
- In the "Training" tab, use the "Brush Tool" to manually annotate two or three cells of interest (e.g., positive examples).
    - Label them as "Foreground" (cells to be recognized) as yellow.
    - Optionally, mark some background regions as "Background" as blue .

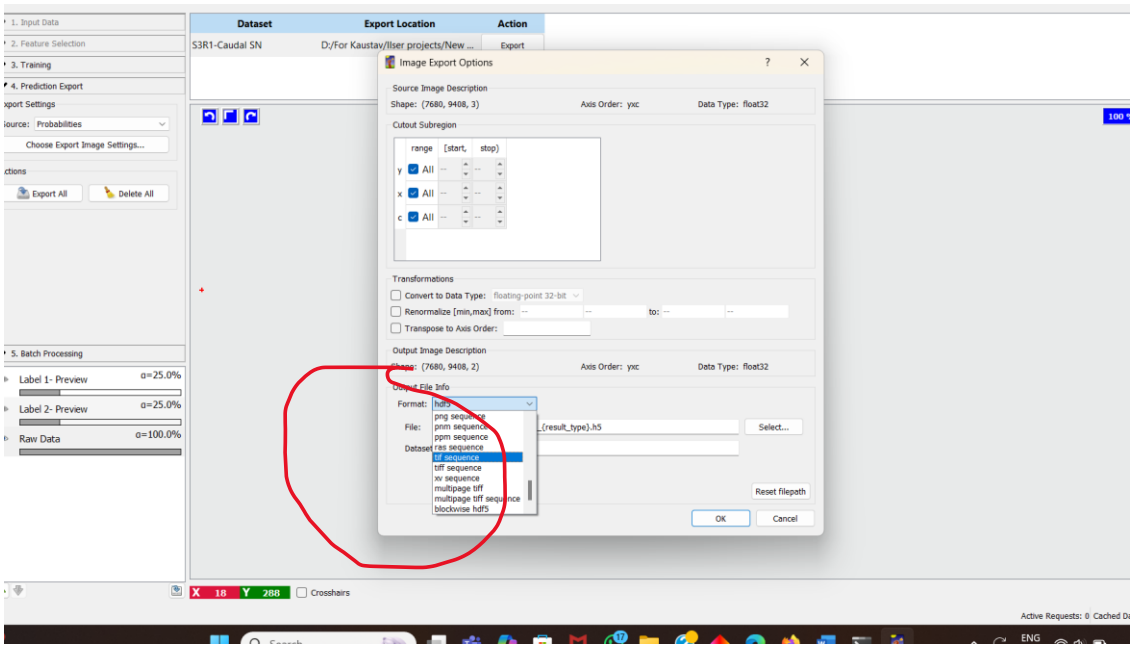


i. **NOTE- While training it is necessary to train all the types of cells of varying complexity .**

ii. **We can also click on live update to check our current update of training and correct accordingly .**

- After annotating the sample cells, click on the "Train" button. Ilastik will use these annotations to train a classifier.
- We can visualize the trained model in real-time as it starts to detect and classify the other cells in the image.
- Ilastik will automatically annotate the remaining cells based on the model's predictions.
  Go to prediction export and choose **simple segmentation** after that choose “Export image settings and save the file as {dataset1}.tif by choosing tif rom drop down menu



- 

This will actually save our masks in the form of .tif file in desktop.

IMAGEJ

Load the TIF file:

- Open ImageJ and load your TIF file. Initially, the image will appear black.

Adjust the contrast:

- Click on **"Image"** in the menu bar → Select **"Adjust"** → Click on **"Contrast"**.
- In the contrast window, click on **"Auto"** to automatically adjust the contrast.
- Afterward, click **"Set"** to apply the changes, then click **"Apply"** to finalize.

Convert to Binary Mask:

- After adjusting the contrast, go to **"Image"** → Select **"Type"** → Choose **"8-bit"** to convert the image to grayscale if needed.
- Then, go to **"Process"** in the menu bar → Select **"Binary"** → Click on **"Make Binary"**.
- This will automatically convert the image to a binary mask, where areas of interest are shown in white, and the background is black.

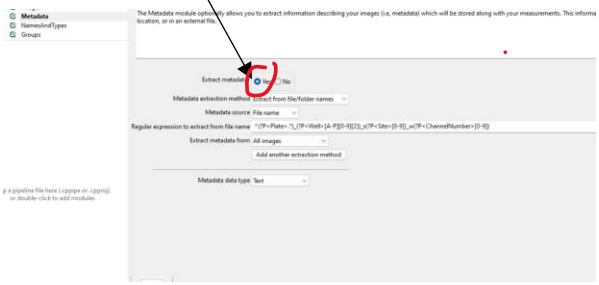
CELLPROFILER

(WITHOUT BATCH PROCESSING)

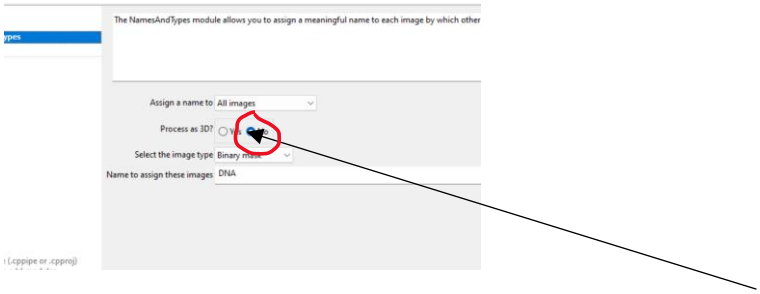
1. **Open CellProfiler**: Launch the CellProfiler application on our system.

2. **Drag and Drop the Binary Masks**: Locate the binary masks we created in ImageJ. Drag and drop them into CellProfiler.

3. **Extract Metadata**: Once the images are loaded, a prompt will appear asking if we want to extract metadata. Click **Yes** to proceed.



4.**Select Image Type**: In the "NamesAndTypes" module, find the dropdown menu under "Image Type." Here, we need to delete the current selection and choose **Binary Images** as the type.



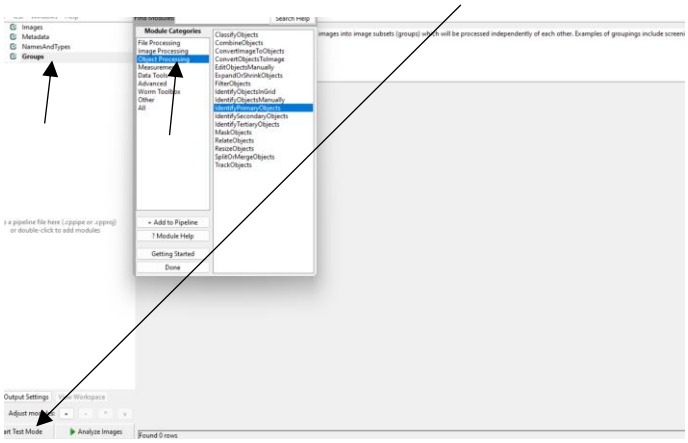
5. **Group Settings**: Navigate to the "Groups" module and click **No** when prompted.

6. **Add Modules**: Click on **Add Modules** to include additional processing steps.

7. **Object Processing**: From the available modules, select **Object Processing**.

8. **Identify Primary Objects**: Within the Object Processing section, click on **Identify Primary Objects**. This will set up the object identification step using the binary masks.

9.**Start Test Processing**: Click on **Start Test Processing** to begin processing and testing your setup.



CELLPROFILER(WITH BATCHPROCESSING)

1. **Load Images from Dataset File**

We start by loading all the images from the dataset file by dragging and dropping them into the CellProfiler interface.

2. **Extract Metadata**

After loading the images, we proceed to extract metadata. This step is crucial for organizing and processing the data efficiently.

3. **Select Binary Images**

We choose the binary images required for further processing.

4. **Group Settings**

Navigate to the "Groups" module. When prompted, click **No** to skip automatic grouping.

5. **Add Modules**

Click on **Add Modules** to include additional processing steps necessary for our analysis.

6. **Object Processing**

From the available modules, select **Object Processing** to work with image objects.

7. **Identify Primary Objects**

- Within the **Object Processing** section, click on **Identify Primary Objects** to set up object identification using binary masks.
- Adjust the **diameter** parameter, which can be measured using ImageJ.
- Set the **threshold lower bound** to **0.5** and the **upper bound** to **1** for accurate object detection.

8. **Export as Spreadsheet**

Add a module named **Export as Spreadsheet** to save the results. The specific file path for the spreadsheet will be set by default.

9. **Analyse Images**

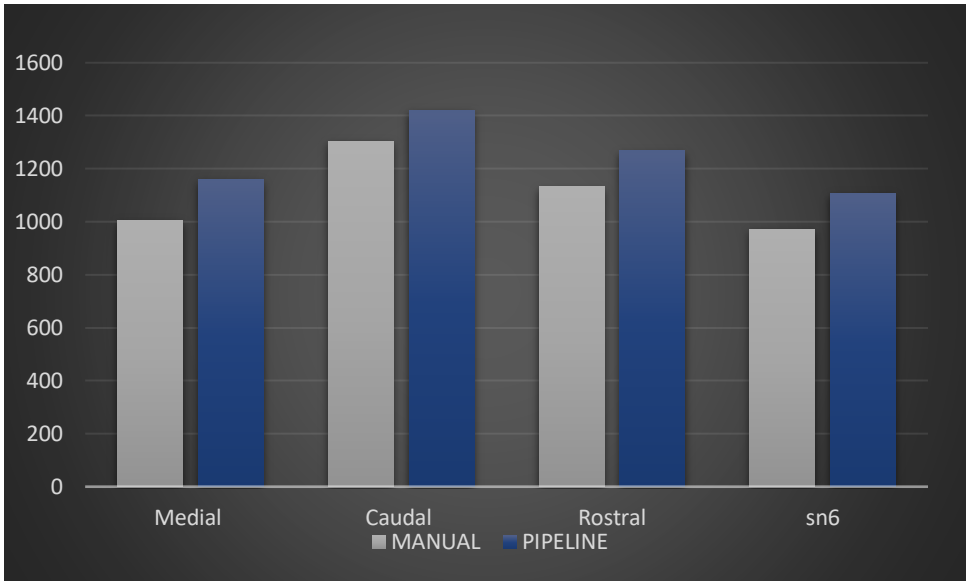
Finally, click on **Analyse Images** to process the data. The analysis will provide the specific number of counts, which will be saved in the spreadsheet file

RESULTS :

Required results are as follows in the following table which has been represented in excel format :-

ANALYSIS OF NUMBER OF CELLS AND ACCURACY OF PIPELINE

				RADIUS	
Cell Types			Number of cells	Lower limit	Upper limit
	CELL	MANUAL	PIPELINE		
1	Medial	1005	1160	EIGHT	62
2	Caudal	1303	1419	9	30
3	Rostral	1132	1267	10	30
4	sn6	972	1107	6	30



CONCLUSION

Hence, we have successfully counted the regions of interest (ROI) using the designed software pipeline in CellProfiler. This streamlined approach ensures accurate analysis and efficient data export, ready for further interpretation.