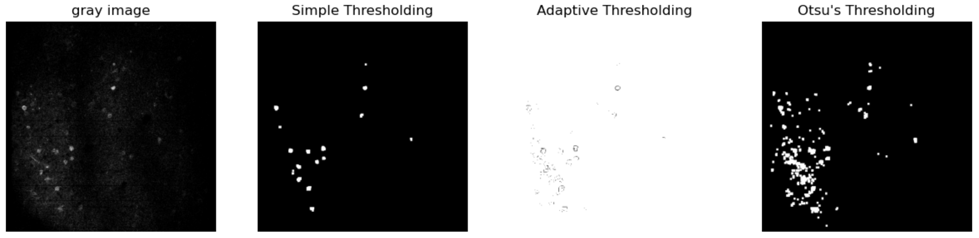
To perform the analysis, we decided to do thresholding, cell segmentation, and, afterward, measurements of calcium response as requested in the assignment.

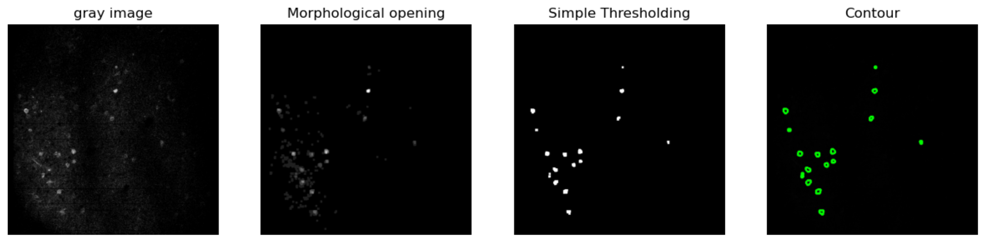
First, we evaluated simple, adaptive, and Otsu's thresholding for cell segmentation. Before the thresholding, we used morphological opening to reduce noise (Figure 1).

Figure 1- Example of one image: Thresholding after Morphological Opening



Contour-based segmentation offers advantages over methods that rely solely on geometric shapes like circles for recognizing cells, especially when cells do not have a perfect circular shape (Figure 2).

Figure 2- Example of one image: Contour Based Segmentation



To identify all cells across multiple time points, we aimed to construct a unified pattern encompassing all cells illuminated in different images. This pattern will facilitate the analysis of calcium response across all images (Figure 3-A). Since the illumination is progressive, we sought to exclude any contours nested within others since we're observing the same cell at different illumination stages (Figure 3B). Next, we apply more noise reduction to the contours by filtering based on the minimum area (figure 3C), which assists in distinguishing between individual cells. Also, adding an upper limit to the contour filter ensures that large structures, which may encompass multiple cells or non-cellular features, are excluded (Figure 3D).

Figure 3- Combined pattern of all images

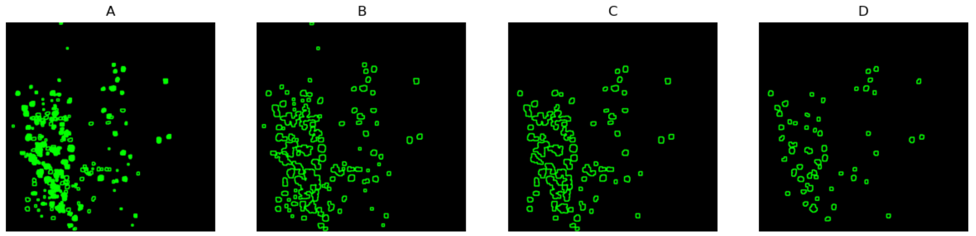
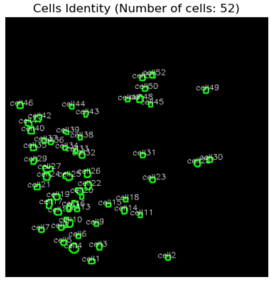
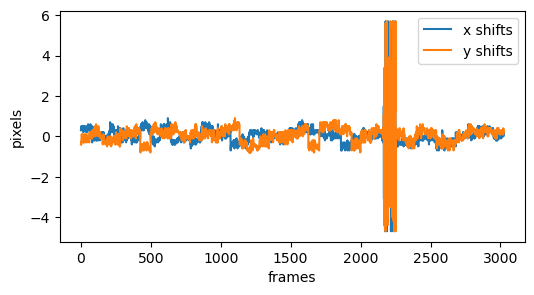


Figure 4- Numbering the cells:

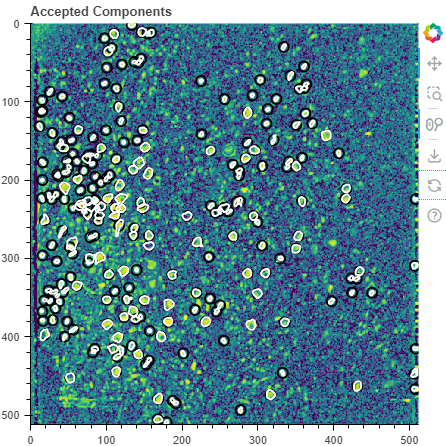


As we experimented with various parameters in the code to enhance cell identification, we observed that increasing the number of detected cells comes at the expense of distinguishing between cells (often resulting in multiple cells being considered as one). Consequently, we sought a more effective method for segmenting the cells. Therefore, we turned to the literature and consulted lab members on how they analyze their imaging data. We then turned to Python libraries suitable for cell activity image analysis, specifically for calcium imaging. We encountered a paper that improved the detectability of calcium imaging experiments. It was accompanied by a Python library, rich documentation, and examples that helped implement it in other data. This Python library, CaImAn, is specifically designed to analyze one-photon and two-photon calcium imaging data. In their paper, the authors explain the algorithms and methodologies used to reduce noise, improve cell segmentation accuracy, and provide an accurate measure of the calcium sensor response in an imaging session. Thus, we’ll briefly explain the logic behind the code and present some of the results we have obtained when implementing it on our dataset.

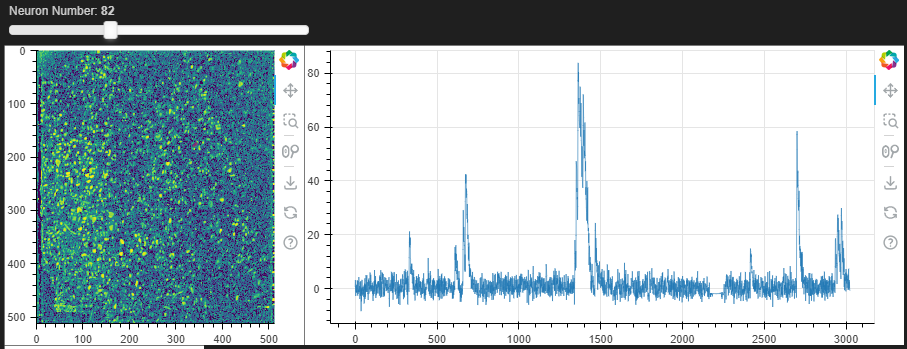
The first algorithm we implemented in our analysis, the NoRMCorre algorithm, is of paramount importance. It plays a pivotal role in our analysis as it rectifies rigid motion artifacts that may arise due to the natural movement of cells in living animals. This is achieved by estimating motion vectors and smoothing the motion field within our field of interest (FOV). The result is a still, clearer image of the cells in our FOV, thereby enhancing the accuracy of our analysis. The following figure shows the pixel compensation made by the motion correction algorithm, showing that it did have some effect on the displayed imaging session.



After motion correction, source extraction was performed. We used the framework of non-negative matrix factorization (CNMF). Essentially, this algorithm extracts components with spatial overlapping projections. This allows for better cell segmentation performance relative to the method we tried previously. We applied source extraction algorithms to our data, which were designed to handle one-photon data, as we assumed it was the source of our dataset. As can be seen in the following figure, cell segmentation was significantly more accurate.

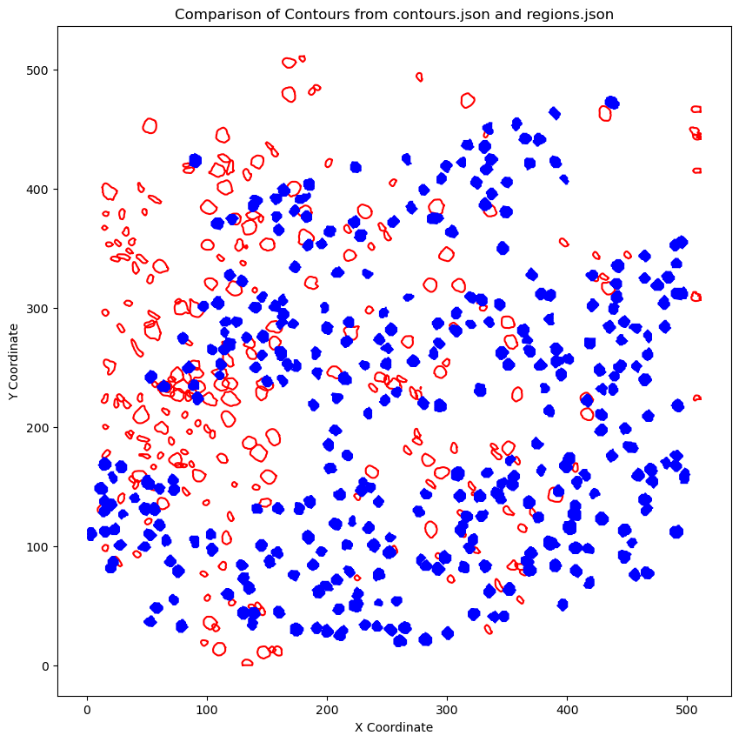


Finally, deconvolution was performed—sparse non-negative deconvolution using the OASIS algorithm and convex optimization framework. Deconvolution aims to recover accurate spike timing, improve the temporal resolution of spikes, reduce noise, and allow the quantification of activity levels of the various neurons identified by the source extraction phase. We applied this and quantified the neuronal activity levels of the cells in our dataset, as presented below. We can see spikes emerging from distinct neurons, which validated our analysis's accuracy.



A graph showing different colored lines

Description automatically generated

Finally, we compared the contours we obtained to the regions.json file provided. As shown below, we had significant differences, including cells that we missed and some that we captured and did not mark in the regions.json file (Blue – regions.json file, Red –our analysis).