# Automated Image Analysis for Screening GEVIs in spiking HEK cells

S. K. Shaik

Delft University of Technology

TU Delft

# Automated Image Analysis for Screening GEVIs in spiking HEK cells

by

S. K. Shaik

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday July, 25 2024 at 9:30 AM.


| | | |
|---|---|---|
| Student number: | 5742218 | |
| Project duration: | December 1, 2023 – July 25, 2024 | |
| Thesis committee: | Dr. D. Brinks, | TU Delft, supervisor |
| | Dr. D. Maresca, | TU Delft |
| | Dr. N. Bhattacharya, | TU Delft |


*This thesis is confidential and cannot be made public until December 31, 2025.*



TU Delft

# Contents

# 1

# Abstract

Genetically Encoded Voltage Indicators (GEVIs) are tools to directly measure membrane voltages in cells through fluorescence. Spiking HEK cells, cells which can produce easily evocable voltage spikes, are useful in studying GEVIs. Populations of spiking HEK cells expressing GEVI variants can be used to identify the best GEVI variants in the population in terms of speed and sensitivity. To facilitate such screenings an automated image analysis pipeline is developed in this project. The pipeline corrects for motion artifacts, segments the single spiking HEK cell with an IoU of 0.881 compared to manual annotation and, extracts sensitivity, speed and membrane localization of the GEVIs expressed by these cells. When comparing sensitivity, speed and, membrane localization values extracted by the pipeline to manually calculated ground truth values, an error of 10.672%, 16.639% and, 13.107% is calculated in the averages of sensitivity, speed and, membrane localization, respectively. To demonstrate its functionality, the pipeline screens a population of spiking HEK cells expressing GEVI variants. From this screening, the pipeline identifies a single best GEVI variant with a sensitivity of 415.2% and a speed of 131.7/seconds.

**Keywords**
GEVIs, Spiking HEK cells, Voltage Imaging, Image Analysis, Signal Processing

# 2

# Introduction

*Neurons, the basic working unit of the brain, are electrically excitable cells. Neurons send electrical signals called action potentials (AP). These APs travel through individual neuronal cells. Neurons are typically connected to hundreds of other neurons, forming complex neural circuits. The electrical signals flowing through these complicated neural circuits give rise to higher order brain functions such as memory, emotions and, consciousness. The disturbances in the flow of these electrical signals is also a fundamental problem in many neuropsychiatric disorders. As a result, understanding these generated complex electrical signals has been a significant and challenging research area in neuroscience [1]. Understanding these electrical signals can help in answering active neuroscience research questions such as how memories are encoded, stored and, retrieved as well as what is the neural basis of consciousness. Additionally, these electrical signals can also help in understanding the neural mechanisms underlying psychiatric disorders.*

*Genetically Encoded Voltage Indicators (GEVIs) are protein based sensors that record membrane potential changes simultaneously in large numbers of cells. GEVIs can record the membrane potential of hundreds of cells with subcellular and millisecond scale spatio-temporal resolution [1, 2]. Therefore, GEVIs have become attractive and powerful tools for neuroscientists. GEVIs help neuroscientists in tracking the neural electrical signals by targeting subsets of neurons and distinct neuronal compartments [2]. Conventional methods such as functional magnetic resonance imaging (fMRI) and functional near-infrared spectroscopy (fNIRS) are only able to indirectly measure neural activity in the brain. Calcium ($Ca^{2+}$) imaging, on the other hand, directly measures neural activity but does not provide direct voltage information of the neurons. GEVIs enable direct recording of membrane voltage through measurement of fluorescence intensities. This potential of GEVIs to solidly couple fluorescent signals with the state of membrane potentials can help in producing a detailed examination of the brain mechanisms [1, 3, 4].*

*An action potential induced by neurotransmitters lasts 10 milliseconds (ms) at most. Hence, GEVIs expressing quick kinetics and large increase in fluorescence in response to increase in voltage are desirable to detect membrane voltage changes. Researchers have tried extensively to develop GEVIs that could follow these fast voltage changes [1]. However, due to the complex relationship between the structure of GEVIs and its fluorescent properties, improvements in GEVIs have been slow and labor-intensive [5]. Neurons, in general, are the best cells to characterize GEVIs in. But neurons are hard to use, especially in high quantities. Engineered Human Embryonic Kidney (HEK) cell lines, on the other hand, are swiftly growing and easily cultured. The spiking HEK cells are a subtype of the HEK 293 cells. Spiking HEK cells produce spontaneous optogenetically triggered electrical spikes. Therefore, the HEK 293 and spiking HEK cell lines allow researchers to easily evaluate several features of GEVIs. The spiking HEK cell lines can help in GEVIs screening in a high-throughput manner by identifying fast and sensitive variants of GEVIs [5, 6, 7, 8].*

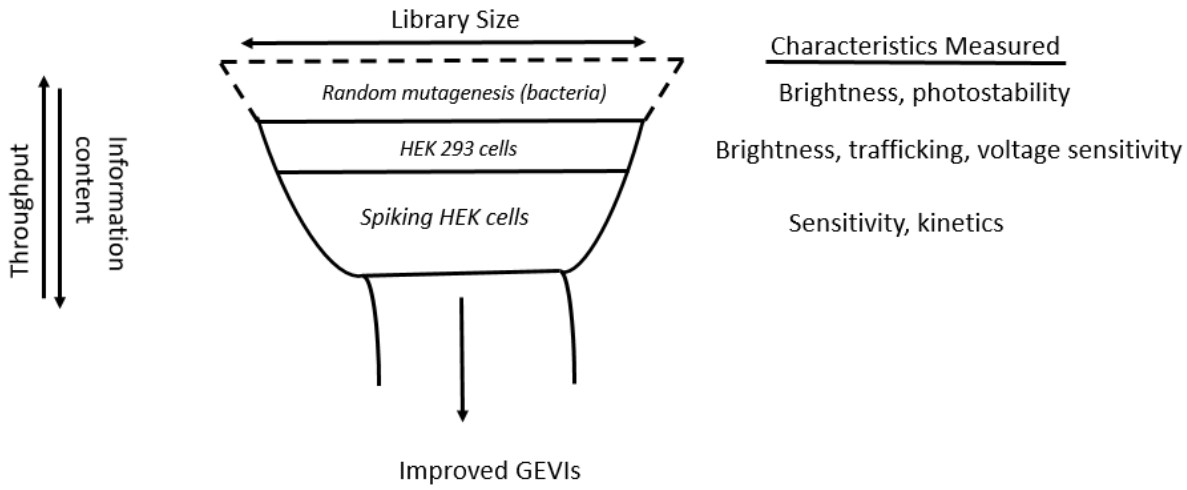



**Figure 2.1: GEVIs screening pipeline**. In bacteria large libraries of GEVIs can be screened but the parameters measured are only brightness and photostability. In spiking HEK cells although the throughput is relatively low compared to bacterial screening, more interesting parameters like trafficking, sensitivity and, kinetics can be measured. Copied from Xu et al. [5] and Park et al. [6]

*Figure 2.1 explains about the throughput with which GEVIs can be measured via different methods. Figure 2.1 also explains about the kind of information that these different measurements give [5, 6]. Spiking HEK cells enable extraction of several important features of GEVIs such as the speed of the GEVIs, sensitivity of the GEVIs and, the membrane intensity to whole cell intensity ratios. With the available information about the features of the GEVIs, researchers can quickly identify the best GEVIs among several of the variants of GEVIs. Therefore, screening of population of spiking HEK cells expressing libraries of GEVIs facilitates the development of improved fluorescent voltage indicators for neuroscience studies.*

*To compare the features of different GEVIs, signals of the spiking HEK cells expressing these GEVIs can be extracted by voltage imaging of the spiking HEK cells. Voltage imaging involves laser based excitation of the spiking HEK cells expressing GEVIs. After excitation, the spiking HEK cells show membrane potential changes. The GEVIs expressed by these spiking HEK cells produce fluorescent signals corresponding to the membrane potential changes of the spiking HEK cells, as shown in fig.2.2.*

*Voltage imaging using one-photon (1P) or two-photon (2P) excitation present the advantages of high spatial resolution, non-invasiveness, ease of operation and, high measurement throughput compared to electrode based techniques like patch clamp electrophysiology [2, 5]. However, the small signals generated by spiking HEK cells expressing GEVIs impose noteworthy challenges in the signal extraction of the GEVIs. This subsequently hampers the building of a novel screening pipeline for population of spiking HEK cells expressing GEVIs. GEVI imaging is a novel methodology with the image processing algorithms requiring reasonable optimization. The 'big data' produced requires computational power to extract the key information contained within [9].*

*Due to the complicated nature of the signals of GEVIs, analysis of voltage imaging movies is challenging. As a result, it is hard to extract the features such as sensitivity and speed of the GEVIs from their signals. In addition, the available image analysis pipelines are suited for analyzing patch clamp recordings of HEK cells expressing GEVIs and movies of neurons expressing GEVIs. Spiking HEK cells, on the other hand, are a different subset of cells. HEK cells differ from spiking HEK cells as HEK cells do not spike, thereby resulting in different signals as compared to spiking HEK cells. Neurons differ from spiking HEK cells in terms of their cellular morphology and in terms of voltage kinetics.*

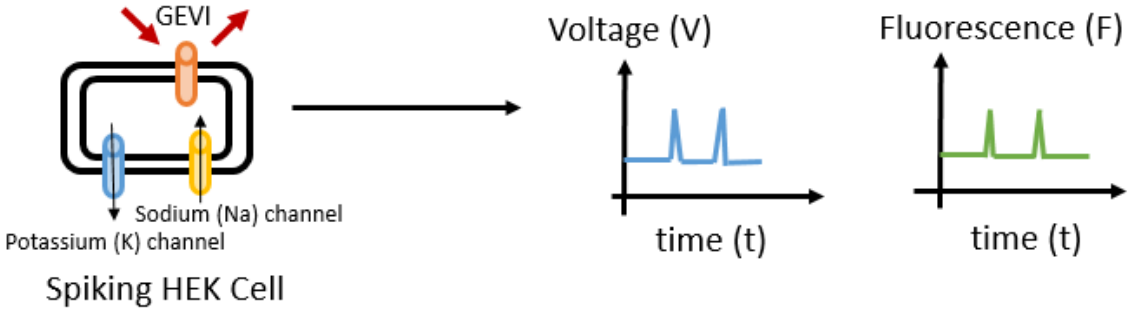



**Figure 2.2: Voltage Imaging of GEVIs**. The fluorescent fluctuations of GEVIs are detected by one-photon or two-photon microscopy. Spiking HEK cells are used to test for voltage sensitivity and kinetics of the GEVIs. Adapted from Xu et al. [5] and Roth et al. [9]

*Therefore, the available voltage image analysis pipelines are not well suited for analyzing signals of spiking HEK cells expressing GEVIs. This necessitates good quality image analysis pipelines that can automatically extract the information contained within the movies of spiking HEK cells expressing GEVIs.*

*Figure 2.3 depicts a generalised image analysis pipeline for the automated analysis of 1P voltage imaging datasets of spiking HEK cells expressing GEVIs. The first step in the image analysis pipeline is to acquire and load the time-lapse image videos of the dataset (Fig.2.3a). Next, the image analysis pipeline corrects for the small movement artifacts that occur in the movie frames (Fig.2.3b). These motion artifacts are corrected for by registering the individual image frames of the acquired movie to a common template image frame from the acquired movie. This ensures that the spiking HEK cells that are to be extracted occupy the same spatial footprint in all of the movie frames [9, 10]. After this, the image analysis pipeline identifies the individual spiking HEK cells of interest (Fig.2.3c). Finally, the image analysis pipeline extracts the fluorescence intensity distributions of the GEVIs expressed by the segmented spiking HEK cell (Fig.2.3d). The fluorescence intensity distributions of the GEVIs correspond to the voltage spikes of the spiking HEK cells. From the fluorescence intensity distributions, features of the GEVIs like speed and sensitivity can be calculated. These features help in filtering out best GEVIs from the large GEVI libraries, thereby facilitating the screening of population of spiking HEK cells expressing GEVIs. The quality control step of the image analysis pipeline is an additional step that allows the user to check the results of various components of the image analysis pipeline (Fig.2.3e). By evaluating the success of motion correction, segmentation and, signal extraction, the user can correct for any errors.*

*Therefore, successful application of GEVIs requires advances not just in GEVI development and hardware involved in GEVI imaging but also the software or image analysis pipelines. Good signal processing pipelines are essential for analyzing the stream of data that emerges. A good signal processing pipeline that can automatically and accurately extract various features of GEVIs is important for the protein engineering of the best GEVI for neuroscience studies. An ideal GEVI for neuroscience applications is the GEVI with high sensitivity and very fast kinetics.*

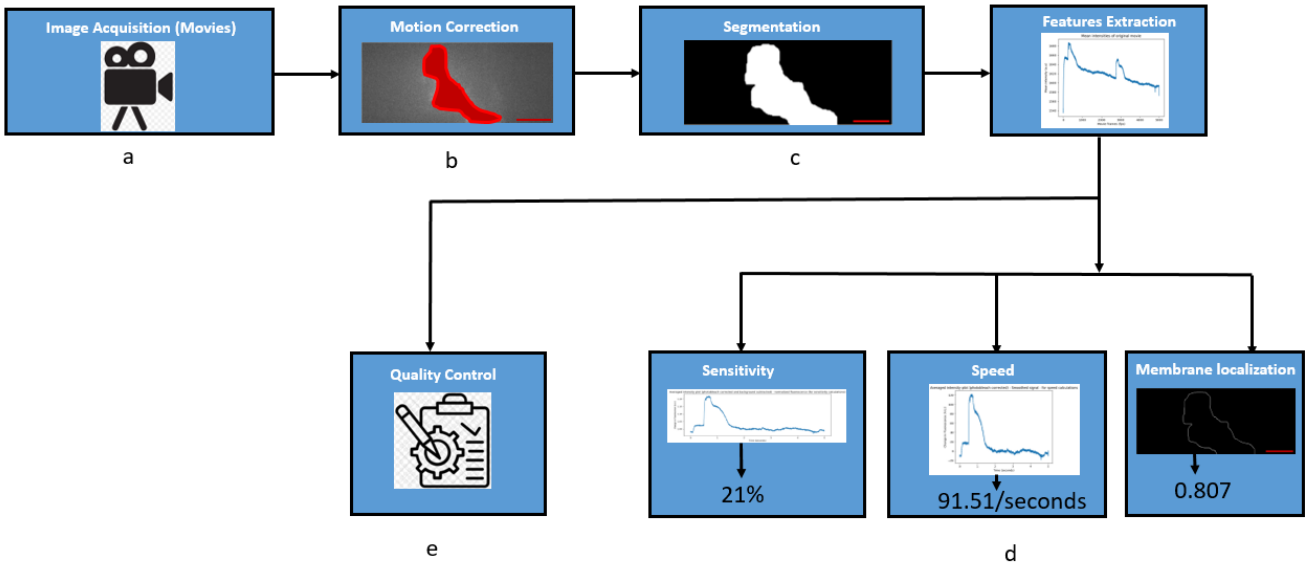



**Figure 2.3: Image analysis pipeline for the analysis of one-photon voltage imaging datasets of spiking HEK cells expressing GEVIs**. Steps b to d are the pre-processing steps required to extract fluorescent spikes of the GEVIs. These fluorescent spikes correspond to the voltage spikes of the spiking HEK cells. Step e is the post-processing step

*The objective of this research project is to develop an image analysis pipeline that can correct for motion artifacts, segment the spiking HEK cells and, extract the features of the GEVIs expressed by the segmented spiking HEK cells in 1P voltage imaging movies of spiking HEK cells expressing GEVIs. The ultimate goal of this research project is to improve the GEVI proteins by directed evolution of the GEVIs. Directed evolution is a method in protein engineering. It involves subjecting the GEVI gene to iterative rounds of mutagenesis, thereby, creating a library of GEVI mutants [11].*

*The research project comprises video based screening of large populations of spiking HEK cells expressing libraries of GEVIs. The screening methodology helps in identifying evolved GEVIs with better sensitivity and kinetics. The newly built image analysis pipeline is a component of the screening methodology. To demonstrate the working of this newly built image analysis pipeline, the pipeline is used for the automated analysis of the movies of single spiking HEK cells expressing variants of GEVIs. Libraries of mutants of GEVIs are created and single cell movies of the spiking HEK cell expressing the variants of GEVIs are obtained. These movies are analyzed by the built image analysis pipeline. The built image analysis pipeline extracts the sensitivity and speed values of the variants of the GEVIs. The best performing variants in terms of sensitivity and speed are then screened for from the population of spiking HEK cells expressing libraries of GEVIs.*

*This report is organised into 6 chapters. The next chapter provides an outline of the research project. Additionally, the chapter describes various components of the built image analysis pipeline. The subsequent chapters describe the results of the pipeline, pros and cons of the built pipeline and, further improvements that can be made to the built image analysis pipeline. Furthermore, the features of the newly engineered GEVIs that are extracted from the image analysis pipeline are also discussed. The supplementary chapter describes the various additional mini-projects that are carried out during the duration of this project. The report ends with a bibliography chapter which includes the list of various references that are used for this work.*

# 3

# Methods

The experimental set-up in this research project consists of a glass bottom dish of spiking HEK cells expressing variants of GEVIs, placed on the octoscope. The octoscope is a fluorescence microscope set-up. The instrument features imaging, 1P or 2P patterned illumination and, recording on a widefield camera or photomultiplier tube (PMT). Figure 3.1 depicts the octoscope design showing 1P illumination pathway. In this project, 1P patterned illumination and the widefield camera is used to record movies of spiking HEK cells expressing GEVIs. The widefield camera is a sCMOS camera (ORCA Flash4.0 V3, Hamamatsu) that can record movies of image sizes upto 2048x2048 pixels and 6.5$\mu$m pixel size [12].

The glass bottom dish containing cultured spiking HEK cells expressing GEVIs is illuminated by a laser. The octoscope has multiple lasers capable of producing light of different wavelengths. As shown in fig.3.1, the octoscope has 3 different lasers that can produce light of wavelengths 639nm, 532nm and, 488nm. These lasers can be individually turned on and off. The AOTF shown in fig.3.1 decides the excitation path of the lasers. For spiking HEK cells expressing GEVIs, the 639nm laser is used to visualize the fluorescence of the GEVIs whereas the 488nm laser is used to stimulate the voltage spikes of the spiking HEK cells.

The glass bottom dish containing cultured spiking HEK cells expressing variants of GEVIs is moved to different positions on the octoscope. At each position, a movie of the field of view (FOV) is recorded and saved. Therefore, movies from many FOVs of the glass bottom dish are recorded. The movies can be recorded at different frame rates (frames/seconds (fps)). The image pixel size of the movies is 233nm.

To select the spiking HEK cells expressing the best variants of GEVIs, features of the GEVIs such as sensitivity and speed of the GEVIs are extracted by analyzing the obtained movies. This analysis is performed using the built image analysis pipeline. The spiking HEK cells expressing the best variants of GEVIs in terms of sensitivity and speed are then traced back and picked for sequencing.

14 single cell movies of spiking HEK cells expressing different GEVIs are used to test the working of the built image analysis pipeline. The GEVIs expressed by these single spiking HEK cells are GR(V80D), GR(A242R) and, QuasAr6a. 3 movies of single spiking HEK cells expressing GR(V80D), 4 movies of single spiking HEK cells expressing GR(A242R) and, 7 movies of single spiking HEK cells expressing QuasAr6a are used. These 14 single cell movies of spiking HEK cells expressing different GEVIs are recorded at 500 fps on the octoscope. The movies have an image pixel size of 233nm. The movies are recorded for a duration of 10 seconds. The 488nm laser is stimulated twice, that is, every 5 seconds of the movie duration, in order to obtain 2 voltage spikes of the spiking HEK cells. The features of these 3 different GEVIs that are obtained from the built image analysis pipeline is discussed in the results chapter of the report.

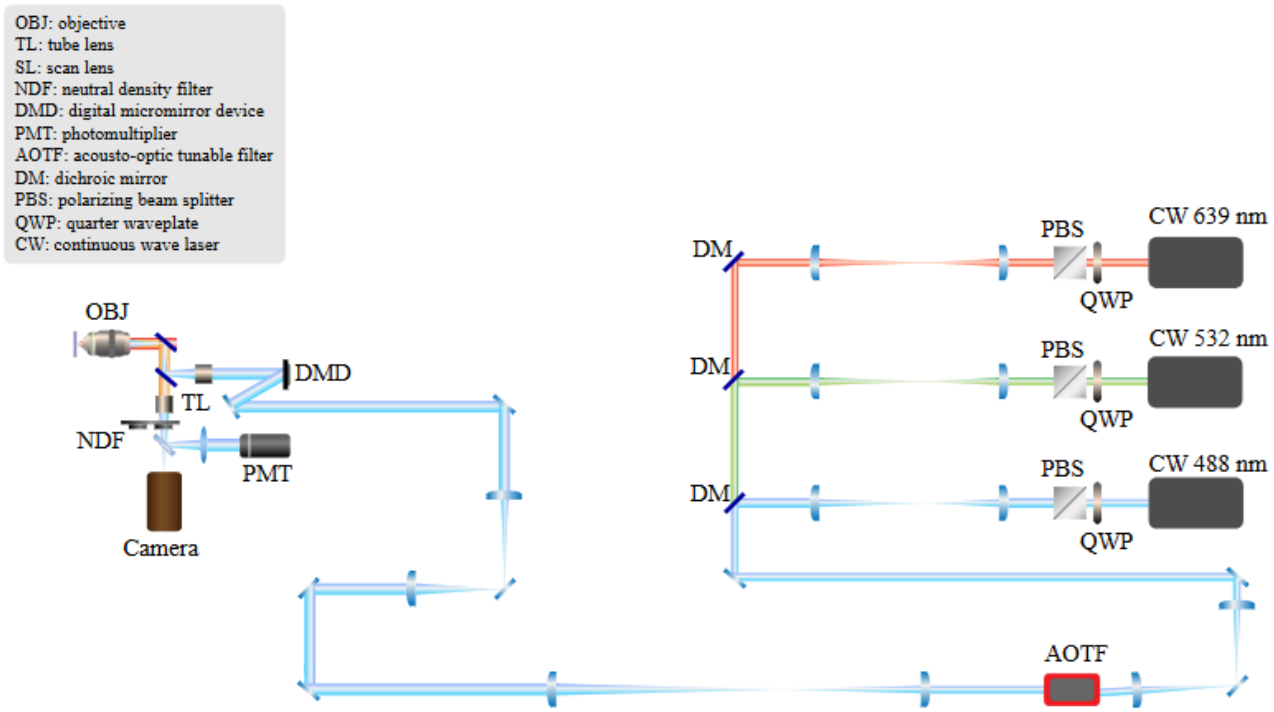



**Figure 3.1: The design of the octoscope set-up with one-photon illumination pathway**. Copied from Meng et al. [12]

The original code of the built image analysis pipeline is available online at `https://github.com/SafiaShaik123/Image-Analysis--1P-Voltage-datasets-of-Spiking-HEK-cells-expressing-GEVIs.git`.

Additionally, the built image analysis pipeline is also used to perform screening of a population of spiking HEK cells expressing variants of the GR(V80D) GEVI. These movies of the spiking HEK cells expressing GR(V80D) variants are recorded at 200 fps on the octoscope. The movies have an image pixel size of 233nm. The movies are recorded for a duration of 2 seconds. The 488nm laser is stimulated twice, that is, every 1 second of the movie duration, in order to obtain 2 voltage spikes of the spiking HEK cells. The results of this screening is described in the results chapter of the report.

The next section describes the various algorithms used to build the different components of the image analysis pipeline. This pipeline analyzes 1P voltage imaging datasets of spiking HEK cells expressing GEVIs. The programming language that is used to build the image analysis pipeline is python. Python is used so that the image analysis pipeline can be easily integrated into the already available software of the octoscope [12]. The octoscope uses the instrument control software Hokawo. This software is written in python. The use of python also ensures that anyone within the research group can use the entire pipeline or individual components of the pipeline, for example, the machine learning (ML) segmentation algorithm, for their image analysis tasks. Various python packages and libraries have been used to speed up the computation process of the built image analysis pipeline, as described in the next section. Due to the advantages such as straightforward writing and execution of code, interactive and user-friendly environment, the python code for the image analysis pipeline is written in the Jupyter Notebook platform.

## 3.1. Motion correction

Motion artifacts in the movies of spiking HEK cells expressing GEVIs, are corrected for using the NoRMCorre algorithm. This algorithm is chosen because of its computational efficiency and its ability to correct for both rigid and non-rigid motion artifacts [13]. Rigid motion artifacts arise when the shape and size of the object of interest is preserved but the pixels of interest have been rotated or translated. Non-rigid motion artifacts arise when the shape of the object of interest is preserved but the size of the object of interest is changed.

NoRMCorre operates by matching patches of FOV in each of the movie frames against a continuously updating template. The registration is performed at subpixel resolution by computing the cross-correlation coefficient between the template and the movie frame. The cross-correlation coefficient is a similarity measure, calculated by equation 3.2a [14, 15]

$$[\widehat{u}, \widehat{v}] = argmax_{u,v}\rho_{12}(u, v) \tag{3.2a}$$

where the offset [u,v] maximizes the normalized cross-correlation coefficient ($\rho$) between the template and the movie frame. The normalized cross-correlation coefficient is calculated by equation 3.2b [14, 15]

$$\rho_{12}(u, v) = \sigma_{g_1 g_2}(u, v)/\sigma_{g_1}(u, v)\sigma_{g_2} \tag{3.2b}$$

where $\sigma_{g_1 g_2}$ is the product of variations of intensities from mean in the template and the movie frame. $\sigma_{g1}$ is the standard deviation of intensity values of the movie frame in the area overlayed by the template. $\sigma_{g2}$ is the standard deviation of the intensity values of the template [14, 15].

The continuously updated template is based on the previously registered frames. To speed up the registration process, the cross-correlation of the template and the movie frame is obtained by Fast Fourier Transform (FFT) methods. The first template is taken as the median of the first few movie frames. The first set of 200 movie frames are then registered to this first template. The subsequent template is then calculated by taking the mean or median of the registered 200 movie frames. The next set of 200 movie frames are then registered to this newly updated template and so on. In this manner, NoRMCorre performs piecewise rigid motion correction. This type of motion correction corrects for rigid artifacts in local regions of the image.

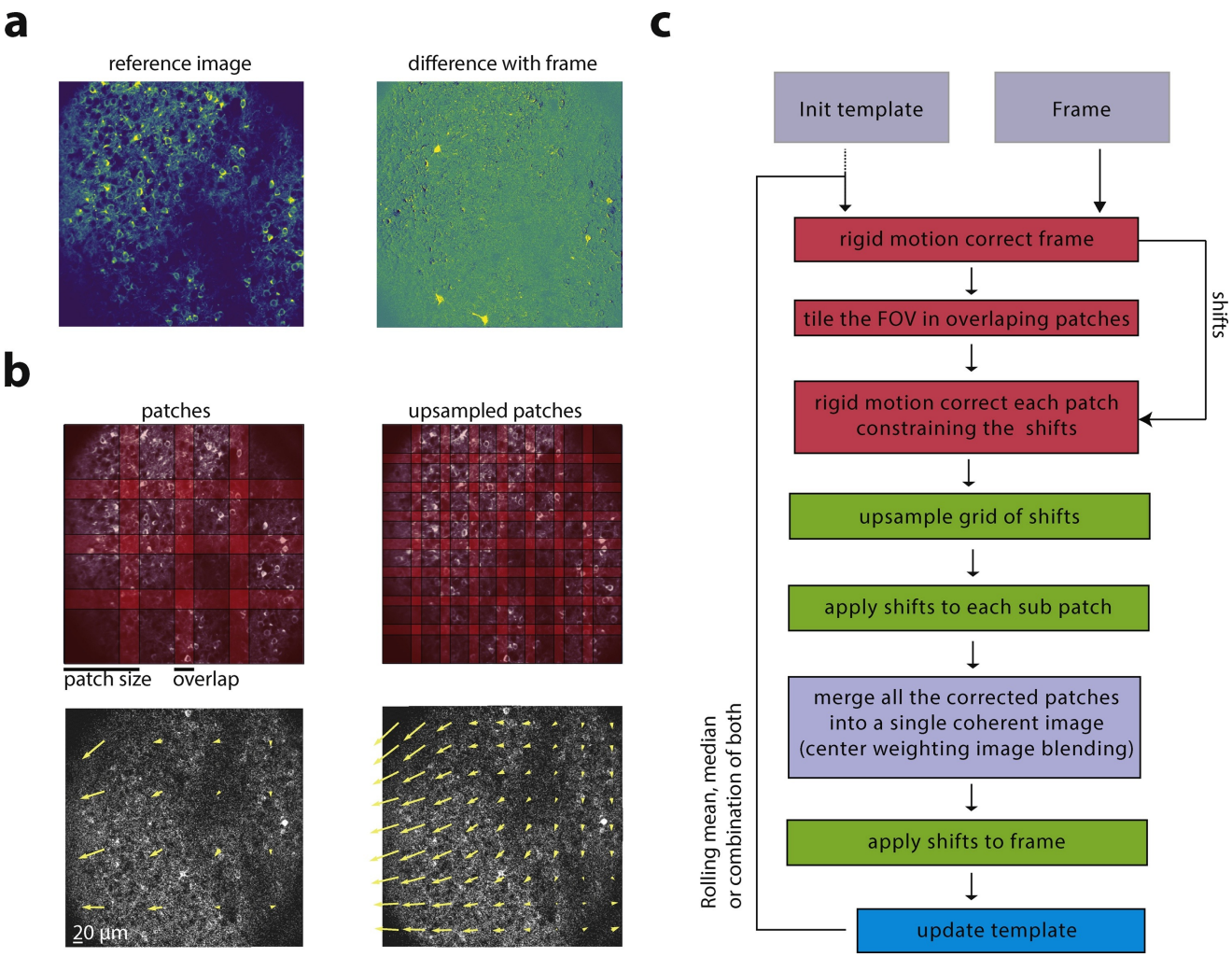



**Figure 3.2: NoRMCorre motion correction algorithm**. **a:** Effect of non-rigid motion artifacts on movie frames. Non-rigid motion causes the displaced cells (neurons in this image) to appear as rings. **b:** Generated overlapped patches and their displacement vectors. The yellow arrows depict the magnitude and direction of motion. **c:** Piecewise rigid motion correction performed by NoRMCorre algorithm. Copied from Pnevmatikakis and Giovannucci [13]

In addition, piecewise rigid motion correction can also correct for non-rigid motion artifacts by splitting the field of view into overlapping patches. These patches are then registered separately and merged by smooth interpolation. This is particularly useful to correct for non-rigid motion artifacts occurring along all axes and not just along one axes. Figure 3.2 describes the NoRMCorre motion correction algorithm.

In the built image analysis pipeline, CaImAn packages are used for motion correction and memory mapping [16].

CaImAn packages have efficient implementation of the NoRMCorre algorithm. There is also an option to set the piecewise rigid motion correction parameter (pw_rigid) to false. The registration is then performed on the entire movie frame rather than smaller independent regions of the movie frame. The decision to perform rigid or piecewise rigid motion correction can be made after visually inspecting the movies and getting a sense of motion in the movies. For the utilized 14 different single spiking HEK cell movies, to build the image analysis pipeline, rigid motion correction is performed. This is because significant motion is not seen in the spiking HEK cell movies thereby not necessitating piecewise rigid motion correction.

Memory mapped files are random portions of the movie read in any direction. This saves memory as the full movie doesn't need to be loaded into memory. After the original movies are motion corrected, memory mapped files of the motion corrected movie are created. These memory mapped files are used to obtain mean and correlation images of the motion corrected movie. A combination of the mean image and correlation image is used for segmentation of the spiking HEK cell in the motion corrected movie. This is described in the next section.

## 3.2. Segmentation

The memory mapped file of the motion corrected movie is used to obtain summary images of the motion corrected movie. The summary images include the mean image and the local correlation image of the motion corrected movie (Fig. 3.3) [17].

### Mean Image

The mean image of the motion corrected movie is obtained by averaging the movie across time for each pixel. This results in a 2D mean image of the motion corrected movie as shown in fig. 3.3b. The 2D mean image is then normalized by subtracting the mean of its pixels and dividing by the standard deviation of its pixels. This normalization step ensures that different datasets share the same intensity range as input to the segmentation model. The mean image of the motion corrected movie is obtained using the 'mean_image' function of python's caiman.summary_images module from the CaImAn library.

### Correlation Image

The local correlation image of the motion corrected movie is obtained by averaging the temporal correlation of each pixel with its 8 neighbour pixels [17, 18]. This results in a 2D correlation image of the motion corrected movie as shown in fig. 3.3c. The 2D correlation image is then normalized in the same manner as the 2D mean image. The local correlation image of the motion corrected movie is obtained using the 'local_correlations_movie_offline' function of python's caiman.summary_images module from the CaImAn library.

After obtaining the summary images, the next step is to prepare the data in such a way that the data can be input to a neural network for segmentation of the spiking HEK cell expressing GEVIs. Neural network is used for segmentation of the spiking HEK cell expressing GEVIs because simple segmentation algorithms like watershed segmentation, segmentation by otsu thresholding do not accurately identify the boundaries of the spiking HEK cell. Identifying the boundaries of a single spiking HEK cell becomes especially difficult when there are clusters of cells. This in turn makes it difficult to extract the features of the GEVIs expressed by the single spiking HEK cell. Since GEVIs are membrane proteins, it is necessary to accurately segment the single spiking HEK cell.

The neural network used in the built image analysis pipeline is the U-Net model. The U-Net network is fast and can be trained from very few images. In the built image analysis pipeline, the U-Net model pre-trained on the Common Objects in Context (COCO) dataset is used [19]. For segmentation of the spiking HEK cells expressing GEVIs, this pre-trained U-Net model is re-trained on single HEK cell images. Due to inavailability of sufficient data of spiking HEK cells, at the beginning of this project, the pre-trained U-Net model is re-trained on HEK cell images instead of spiking HEK cell images. HEK cells have cellular morphology similar to that of spiking HEK cells.

The input of the pre-trained U-Net network is a 3 channel image. Therefore, data from the mean image and the correlation image is prepared in python in such a way that it is compatible with the architecture and pre-trained weights of the pre-trained U-Net model. A combination of the mean and the correlation image is used to obtain a RGB image as shown in fig. 3.3d. This RGB mean + correlation image is compatible with the input layers of the pre-trained U-Net model. When this RGB mean + correlation image is input to the re-trained U-Net model, the model segments the single spiking HEK cell expressing GEVIs. The next section describes the U-Net model.

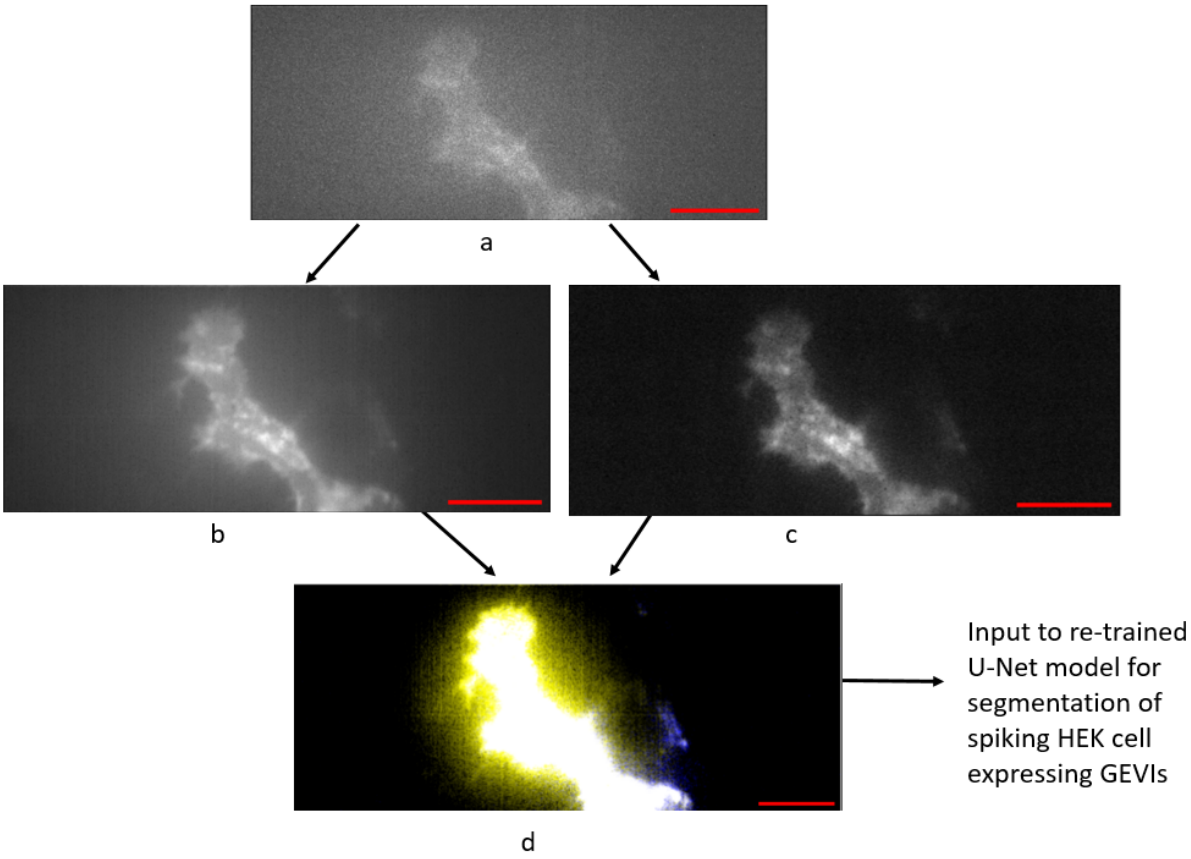



**Figure 3.3: Generated summary images**. **a:** Movie frame 4999 (last frame) of motion corrected movie of single spiking HEK cell. The frame is gray-scale image of size 500x208 pixels. **b:** Mean image of motion corrected movie. The image is gray-scale image of size 500x208 pixels. **c:** Correlation image of motion corrected movie. The image is gray-scale image of size 500x208 pixels. **d:** Final RGB mean + correlation image of size 500x208 pixels. Scale bars = 20$\mu$m

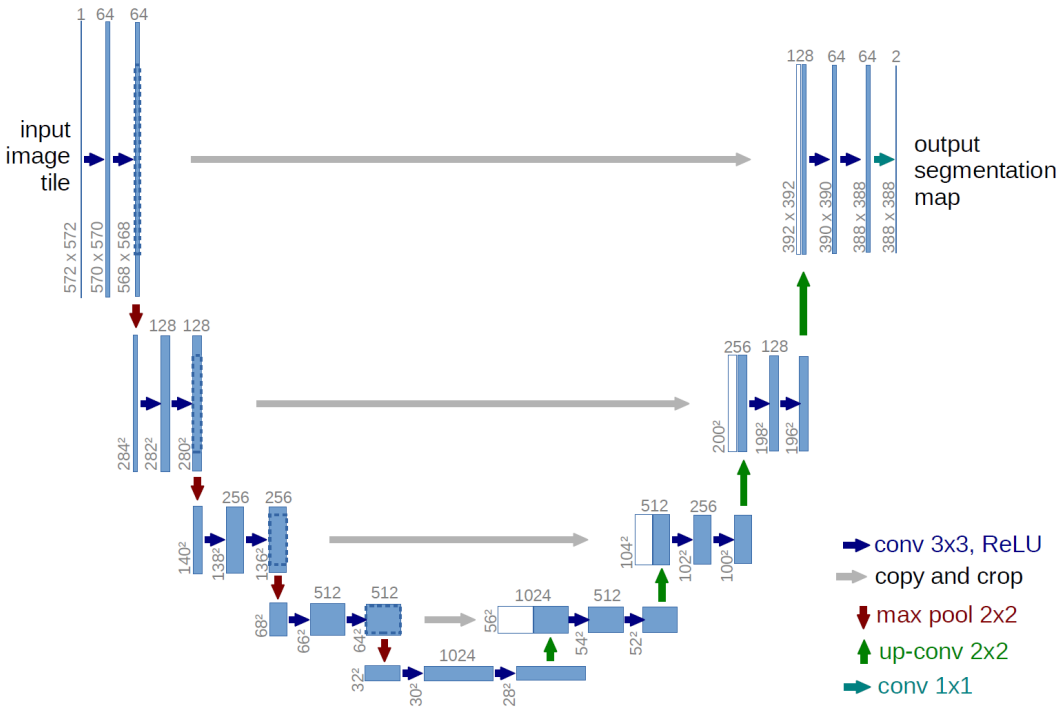



**Figure 3.4: U-Net architecture**. Copied from Ronneberger et al. [20]

### Working of U-Net

The main drawback of convolutional networks is the large amount of annotated datasets required to train the network. Ronneberger et al., in their paper titled 'U-Net: Convolutional Networks for Biomedical Image Segmentation', described the novel U-Net model [20]. The U-Net is a fully convolutional network where every input of the input vector influences every output of the output vector. The U-Net model has the expansive path (right side of Fig. 3.4) more or less symmetric to the contracting path (left side of Fig. 3.4), yielding a U-shaped architecture as shown in Figure 3.4. The 23 different convolutional layers in the U-Net model enable the model to see the context in images at high resolution. For segmentation of large images, U-Net follows an overlap-tile strategy. In this strategy, to predict the pixels at the borders of the images, the missing pixels are extrapolated by mirroring the input image. This overcomes the resolution limitations due to GPU memory for large images.

Additionally, U-Net can be trained on very little training data. Data augmentation by applying elastic deformations can be performed on the training datasets in order to increase the amount of training data. This also teaches the U-Net model to be shift and rotation invariant as well as to be robust to deformations and gray value variations.

The PyTorch implementation of the pre-trained U-Net model for image semantic segmentation is used in the built image analysis pipeline. PyTorch is a python package that offers tensor computation and building neural networks with strong GPU acceleration [21]. The loss function used to re-train the U-Net model in the built image analysis pipeline is BCEWithLogitsLoss. The BCEWithLogitsLoss combines a sigmoid layer with BCELoss in a single class making the loss function numerically stable. The unreduced loss can be described as given in equation 3.3a [22] .

$$l(x,y) = L = \{l_1,\dots,l_N\}^\top, l_n = -w_n[y_n log\sigma(x_n) + (1-y_n)log(1-\sigma(x_n))] \tag{3.3a}$$

where N is the batch size.

The optimization algorithm used to re-train the U-Net model in the built image analysis pipeline is RMSprop. The learning rate of the model is set to 0.0001. Since the learning rate of the model is small, RMSprop is a good optimization algorithm that can be used to re-train the model. RMSprop divides the stochastic gradient by a running average of its recent magnitude. RMSprop can be described as given in equation 3.3b [23].

$$MeanSquare(w,t) = 0.9MeanSquare(w,t-1) + 0.1(dE/dw(t))^2 \tag{3.3b}$$

Dividing the gradient by its $\sqrt{MeanSquare(w,t)}$ improves the learning rate of the model.

The other hyperparameters that are set to re-train the U-Net model are :
Number of epochs = 5
Batch size = 4

These hyperparameters are decided after testing many values. The set hyperparameters are found to be optimal in terms of computational speed and quality of segmentation. The results of a different number of epochs value set to the pre-trained U-Net model to re-train on HEK cell images is shown in the results chapter.

The next section describes the creation of training datasets to re-train the U-Net model and testing datasets to evaluate the re-trained U-Net model for segmentation of spiking HEK cell expressing GEVIs.

### Training and Testing Datasets

To re-train the U-Net model, 23 patch clamp movies of single HEK cells are used. These 23 movies are all of different image sizes and are recorded at 100 fps or 500 fps on the octoscope. 23 single cell RGB mean + correlation images of these HEK cells are obtained. These 23 single cell RGB mean + correlation images are all of different image sizes. In order to re-train the U-Net model, the image dataset should be of the same image size. Therefore, the 23 single cell RGB mean + correlation images are cropped to a common size of 496x109 pixels. The 496x109 pixels is the size of the smallest RGB mean + correlation HEK cell image with least memory requirements. Therefore, to save memory, the training dataset is cropped to the smallest image size of 496x109 pixels. This cropping of the images is performed by first saving the obtained single cell RGB mean + correlation image. Then the image is opened and cropped in FIJI.

Next, data augmentation is performed by rotating each of the cropped 23 single cell RGB mean + correlation images of HEK cells by 90, 180 and, 270 degrees. The rotation of the images is performed using the 'rotate' function of python's Pillow library. The images rotated by 90 and 270 degrees are of size 109x496 pixels. To bring these images back to the size of 496x109 pixels, constant padding with centering is performed on these images using python. This type of padding fills the extra space in the image with the constant color black while maintaining the original content and aspect ratio of the images.

Figure 3.5 shows the data augmentation process for one of the RGB mean + correlation images of single cell patch clamp HEK cell movies. Hence, a total of 92 single cell mean + correlation RGB images of HEK cells are used as the training dataset. The 92 single cell RGB mean + correlation images of HEK cells are manually annotated by a single annotator to obtain the ground truth masks. It roughly took 30 minutes to re-train 5 epochs on HP ENVY x360 Convertible 13-bd0xxx with 16 GB of RAM memory.

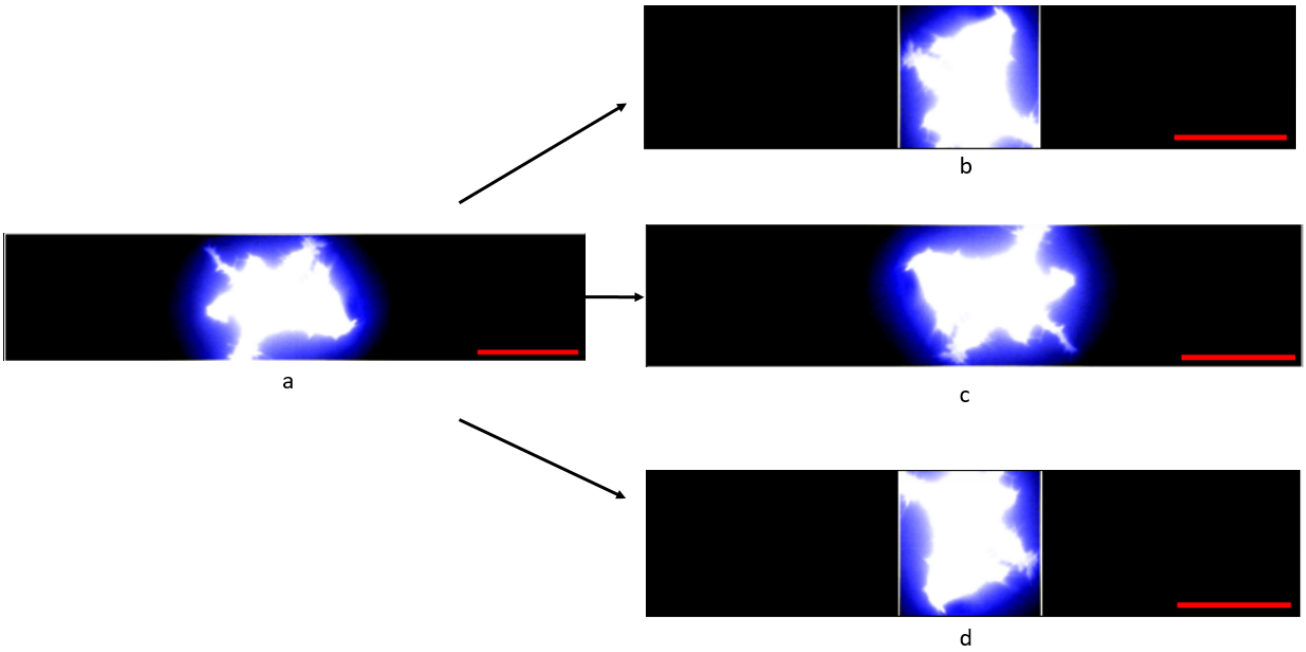



**Figure 3.5: Data augmentation**. **a:** Original zero degrees RGB mean + correlation image of single cell patch clamp HEK cell movie. Image of size 496x109 pixels. **b:** Image rotated by 90 degrees. Image of size 496x109 pixels. **c:** Image rotated by 180 degrees. Image of size 496x109 pixels. **d:** Image rotated by 270 degrees. Image of size 496x109 pixels. Scale bars = 25$\mu$m in the x-direction and 35$\mu$m in the y-direction

To test the re-trained U-Net model, a different set of 7 single cell patch clamp movies of HEK cells are used. These 7 single cell patch clamp movies of HEK cells do not belong to the 23 movies that are used for the training dataset preparation. 7 single cell RGB mean + correlation images of these HEK cells are obtained. These 7 single cell RGB mean + correlation images are cropped to a common size of 496x109 pixels. Data augmentation is then performed by rotating each of the 7 single cell RGB mean + correlation images of HEK cells by 90, 180 and, 270 degrees. The images rotated by 90 and 270 degrees are padded to bring them back to the size of 496x109 pixels. Hence, a total of 28 single cell RGB mean + correlation images of HEK cells are used as the testing dataset. The testing dataset is of size 496x109 pixels.

The re-trained U-Net model is saved as a .pth file. The re-trained U-Net model is loaded in the built image analysis pipeline using PyTorch tensors. Once loaded, the re-trained U-Net model outputs predicted binary masks of the input RGB mean + correlation image of single spiking HEK cell. The predicted binary mask of the single spiking HEK cell is used to estimate the features of GEVIs expressed by the single spiking HEK cell. This is described in the next section.

## 3.3. Features extraction

The predicted binary mask of the spiking HEK cell is of size 496x109 pixels. But the 14 different single spiking HEK cell movies used to build the image analysis pipeline are of different sizes. Therefore, the predicted binary mask is first re-sized in python to the size of the movie frame. The resizing is performed using the 'resize' function of python's OpenCV library. Then the predicted binary mask of the spiking HEK cell undergoes two different dilations resulting in two different dilated binary masks [24]. The first type of dilation is performed to overcome the issues faced when using the original predicted binary mask for feature extraction of the GEVIs. The original predicted binary mask does not accurately identify the membrane of the spiking HEK cell. Therefore, the first dilation can be avoided if the original predicted binary mask accurately segments the membrane of the spiking HEK cell. The second type of dilation is performed for local background estimation for sensitivity calculation of the GEVIs.

Dilation of the predicted binary mask is performed using the 'dilate' function of python's OpenCV library. A 5x5 sized kernel is used to perform both of the dilations. This kernel size is selected to enclose the boundary regions of the spiking HEK cell in all of the 14 different movies of the spiking HEK cells expressing GEVIs. The first dilation is repeated 5 times whereas the second dilation is repeated 4 times.

The original predicted binary mask fails to capture pixels of the segmented single spiking HEK cell that have high intensities. These pixels with high intensities are the pixels that react most to the voltage changes of the single spiking HEK cell. These pixels are necessary for accurate estimations of the features of the GEVIs expressed by the single spiking HEK cell. In order to check that the original predicted binary mask accurately captures all the pixels of interest of the single spiking HEK cell, the mean intensity of the segmented single spiking HEK cell in a particular motion corrected movie frame is calculated in python. This is compared to the calculated mean intensity of manually hand drawn spiking HEK cell region in the same motion corrected movie frame. The region of interest (ROI) is drawn and the mean intensity of the ROI is calculated in FIJI (Image). FIJI is used as the general ground truth to which every result obtained in the image analysis pipeline is compared to.

The mean intensity of the segmented single spiking HEK cell calculated in python is much lower as compared to the mean intensity of the ROI calculated in FIJI. The lower value of the mean intensity of the segmented single spiking HEK cell shows that the original predicted binary mask fails to capture the pixels of the single spiking HEK cell with high intensities.

The 4 times dilated binary mask ensures that the binary mask contains the pixels of the segmented single spiking HEK cell with higher intensities. Figure 3.6 shows the original predicted binary mask that is re-sized to the size of the movie frame, the 4 times dilated binary mask and, the ROI drawn in FIJI. The masks are overlayed on the last movie frame of motion corrected movie. The mean intensity of the pixels contained in these masks are calculated using the 'mean' function of python's NumPy library. These mean intensity values of the segmented single spiking HEK cell are compared to the mean intensity value obtained in FIJI. In FIJI, the ROI is manually drawn in the last movie frame of motion corrected movie. The mean intensity of the ROI is calculated using available features in FIJI.

On the other hand, the 5 times dilated binary mask ensures that the mask contains the region just outside the single spiking HEK cell but not too far away from the cell membrane of the single spiking HEK cell. The two dilated binary masks are subtracted from each other to get a background mask. Here, local background is of importance. This local background is the region just outside the single spiking HEK cell. It is the region where the background intensity is the highest. The background intensity is highest in regions close to the single spiking HEK cell because of the gaussian shaped signal of the illuminating laser. The laser is illuminated above the cells. This results in maximum background intensity in regions close to the cell. The signal decays in a gaussian manner in regions away from the cell. Figure 3.7 depicts the ideal segmented single spiking HEK cell and the ideal local background around the single spiking HEK cell. The ROIs depicted in fig. 3.7 are manually drawn in FIJI. Figure 3.8 depicts how the binary masks have been utilized in the image analysis pipeline in order to obtain the ROIs depicted in fig.3.7.

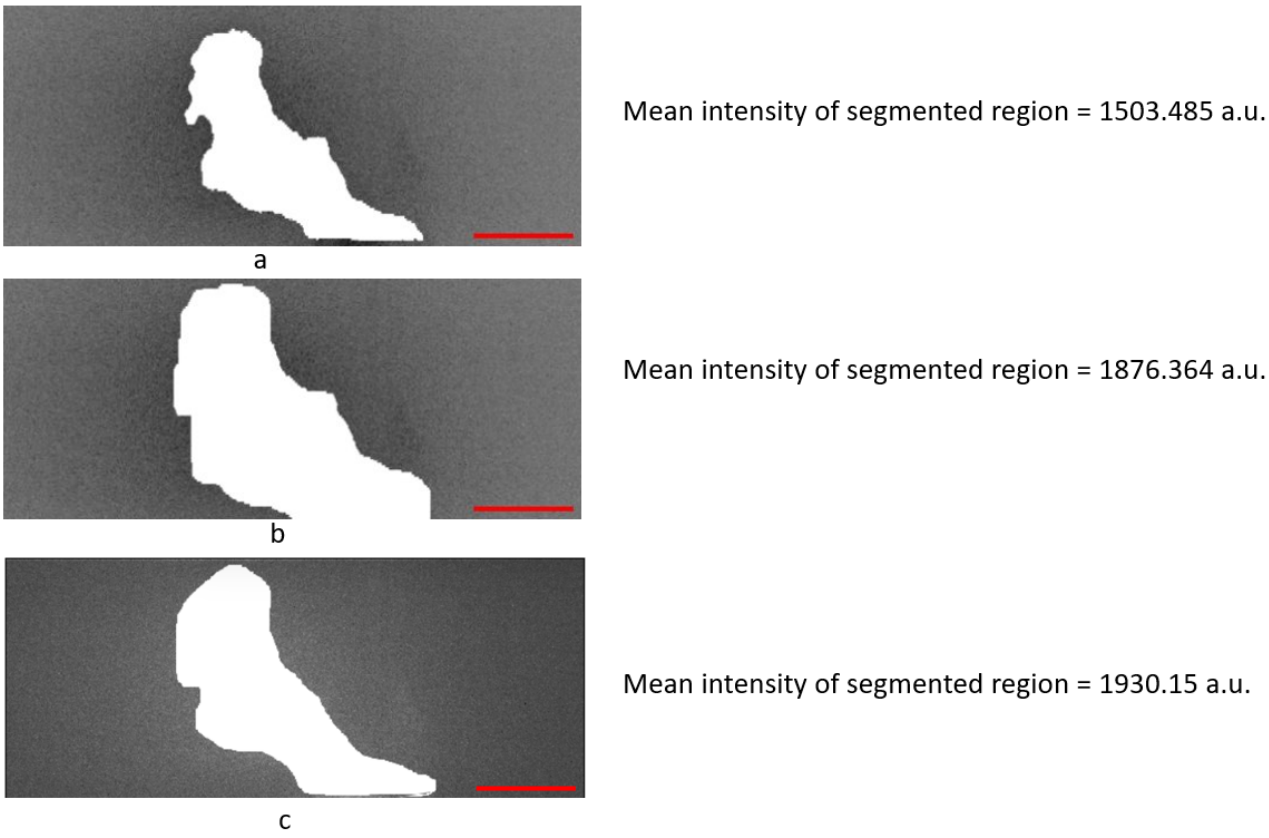



**Figure 3.6: Last frame (frame number 4999) of motion corrected movie showing the segmented single spiking HEK cell and the mean intensity of this region. a:** Original predicted binary mask showing the segmented single spiking HEK cell and the mean intensity of this segmented region. Image of size 500x208 pixels. **b:** 4 times dilated binary mask showing the segmented single spiking HEK cell and the mean intensity of this segmented region. Image of size 500x208 pixels. **c:** Region of interest drawn in FIJI (ImageJ) and mean intensity of this region calculated in FIJI (ImageJ). Image of size 500x208 pixels. Scale bars = 20$\mu$m

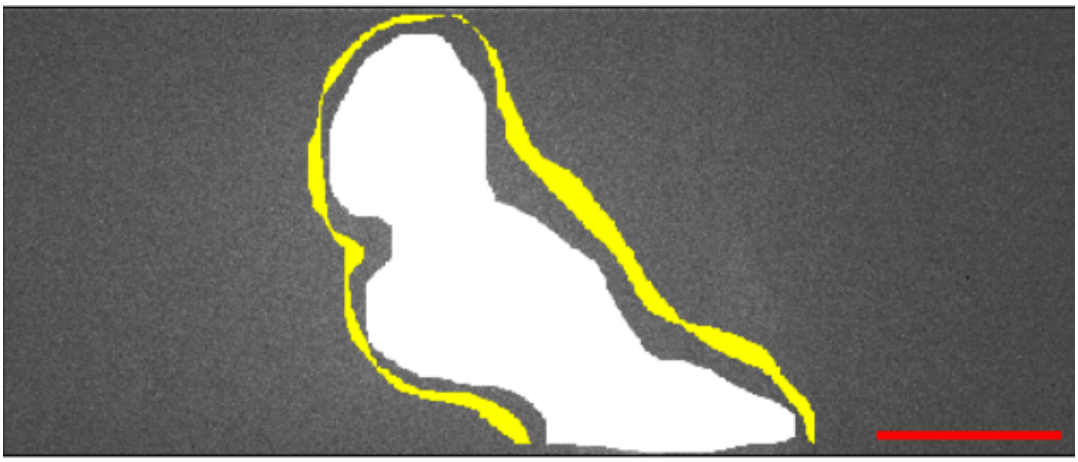


**Figure 3.7: Last frame (frame number 4999) of motion corrected movie showing the regions of interest**. The region in white is the single spiking HEK cell. The region in yellow is the local background around the single spiking HEK cell. Image of size 500x208 pixels. Scale bar = 20$\mu$m

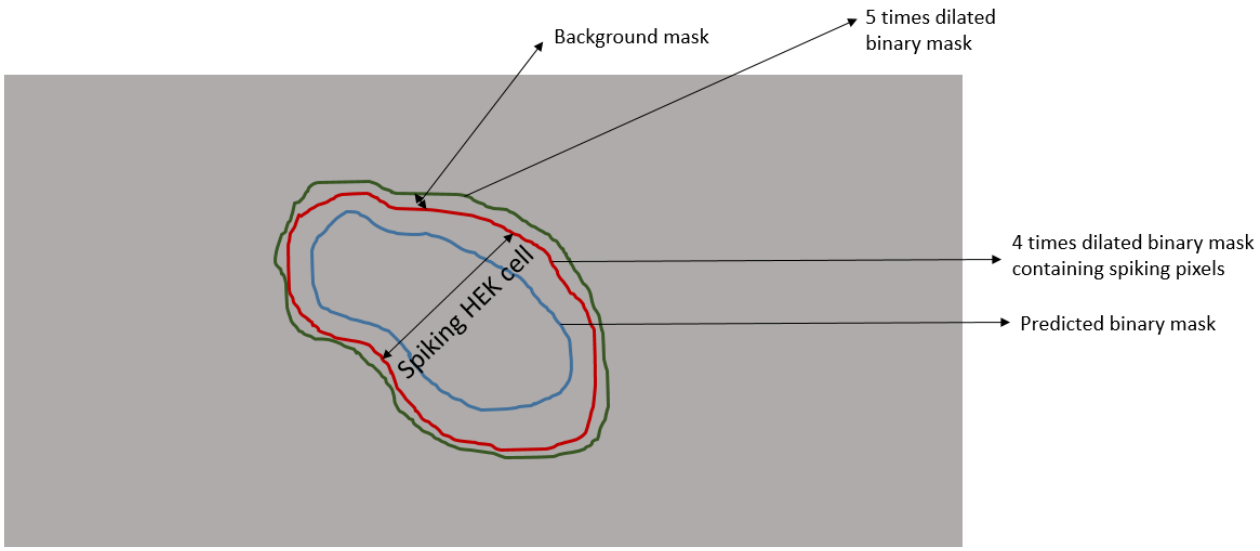



**Figure 3.8: Different binary masks used in the image analysis pipeline.** The spiking pixels are the pixels with high intensities

The 4 times dilated binary mask is now overlayed on the single spiking HEK cell in all of the movie frames of the motion corrected movie. This is to segment the single spiking HEK cell in all of the movie frames. Mean intensity of the segmented single spiking HEK cell is calculated in all of the movie frames and plotted against the number of movie frames. Similarly, the obtained background mask is overlayed on all of the movie frames. Mean intensity of the segmented background region is calculated in all of the movie frames and plotted against the number of movie frames.

The mean intensity plots across movie frames of the segmented single spiking HEK cell and the segmented local background are used to calculate the features of the GEVIs expressed by the segmented single spiking HEK cell. These features are described in the following sections, in the order of their importance.

### 3.3.1. Sensitivity

The sensitivity of GEVIs is the amount of fractional change in fluorescence for physiologically relevant voltage swings [6]. It is the relative change in fluorescence in reaction to a 100 mV change in membrane voltage of the spiking HEK cell typically measured between -70mV and +30mV. Therefore, the higher the sensitivity of the GEVIs, the greater fluorescence change the GEVIs shows in response to the 100mV membrane voltage changes of the spiking HEK cell. Sensitivity of GEVIs is an important parameter in selecting a good GEVI among the mutants of GEVIs expressed by the spiking HEK cells.

In the obtained mean intensity plot across movie frames of the segmented single spiking HEK cell, the signal is not of the GEVI alone. The signal also comprises signals due to autofluorescence effects and laser related artifacts. This signal constitutes the background signal. Additionally, the obtained mean intensity plot of the single spiking HEK cell also shows the photobleaching phenomenon. The photobleaching phenomenon alters the fluorophores in a manner that the fluorophores are not able to fluoresce anymore. This leads to decay in the signal of the GEVIs. For accurate estimations of the sensitivity of the GEVIs, it is necessary to correct for both the background signal as well as the effect of photobleaching.

In the built image analysis pipeline, the background signal is corrected by subtracting the mean intensity plot across movie frames of the background region from the mean intensity plot across movie frames of the segmented single spiking HEK cell. The photobleached signal, on the other hand, is removed using median filtering of the background corrected signal. The 'medfilt' function of python's scipy.signal module is used to smooth the background corrected signal. The medfilt function is a median filter that takes the median of the signal values within a window size. The filter then replaces all of the data points in the window size by the median value [25]. The obtained smoothed signal by median filtering of the background corrected signal is the photobleached signal.

The results of median filtering is compared to the results of moving average filtering. The moving average filter takes the average of a set of data points within a window size. It then replaces the data points within the window size with this average value [26]. Since moving average filtering is a type of convolution, the 'convolve' function of python's NumPy library is used to perform moving average filtering.

For sensitivity calculations, the background corrected signal is divided by the obtained photobleached signal. This normalizes the signal's baseline to 1 or 100 %. The signal is normalized across the duration of the movie. Normalizing the signal makes it easy to identify the peaks in the signal. These peaks correspond to the sensitivity of the GEVIs. In order to obtain a single value for the sensitivity of the GEVIs, the peaks in the plot are averaged to a single peak. From this single peak, sensitivity of the GEVIs is calculated as given in equation 3.4.1a

$$\text{Sensitivity (\%)} = \left( \frac{\text{HighVoltageStateFluorescence} - \text{BaselineFluorescence}}{\text{BaselineFluorescence}} \right) \times 100 \tag{3.4.1a}$$

where, the baseline fluorescence is equal to 1. The high voltage state fluorescence is estimated by averaging the fluorescence intensities between 0.54 to 0.75 seconds in the background-photobleaching corrected and averaged plot. This time range is decided after cross-checking the duration at which high voltage state fluorescence occurs in the 14 different single spiking HEK cell movies used to build the image analysis pipeline. This time range, however, is variable as it depends on the duration of the movie.

The next section describes how the speed of the GEVIs expressed by the segmented single spiking HEK cell is calculated in the built image analysis pipeline.

### 3.3.2. Speed

The speed of GEVIs expresses how quick the GEVIs can follow the voltage changes of the spiking HEK cell. The speed of the GEVI is the time the GEVI takes to reach a certain percentage of the maximum sensitivity. If GEVIs have fast kinetics then the fluorescence intensity trend of the GEVIs would be exactly the same as the voltage trend of the spiking HEK cell. This is because the GEVIs keep up with the quick voltage changes of the spiking HEK cell resulting in the same fluorescence signal trend as the voltage signal trend. However, if the GEVIs have slow kinetics then the fluorescence intensity trend of the GEVIs differs from the voltage signal trend at a particular time span of the signal. This time span is where the GEVIs cannot follow the quick voltage changes and take time to adjust to the voltage change.

For speed estimations in the image analysis pipeline, only the trend of the mean intensity plot across movie frames of the segmented single spiking HEK cell is looked at. Specifically, the trend in the mean intensity plot across movie frames of the segmented single spiking HEK cell, a couple of seconds before the steady state fluorescence starts, is looked at. This time span is where the GEVIs can or cannot keep up with the voltage changes of the single spiking HEK cell. Background signal removal is not necessary for speed calculations. This is because background signal does not affect the signal trend. However, photobleaching is still corrected for. Photobleached signal has affects on the signal trend.
To estimate the speed of the GEVIs, the obtained mean intensity plot of the segmented single spiking

HEK cell is first photobleaching corrected. The photobleached signal is removed in the same manner as described for sensitivity calculations. The only difference being, sensitivity calculations use the background corrected signal whereas speed calculations use the original mean intensity plot across movie frames of the segmented single spiking HEK cell in order to obtain the photobleached signal. To make speed calculation easier, the peaks in the photobleached corrected plot are averaged to a single peak across the duration of the movie. The next step is to check whether the GEVI is a very fast GEVI or a slow GEVI. A very fast GEVI follows the exact same trend as the voltage signal but a slow GEVI cannot keep up with the voltage change for a small duration. For the time span that a slow GEVI cannot follow the voltage change, the slow GEVI shows a different signal trend as compared to the voltage signal trend. To discriminate between a very fast and a slow GEVI, the photobleached corrected signal is smoothed using the 'savgol_filter' function of python's scipy.signal module. This function uses the savitzky-golay filter. The savitzky-golay filter smooths the data by fitting successive polynomial functions to local subsets of the data without distorting the signal tendency [27].

In the utilized 14 different single spiking HEK cell movies to build the image analysis pipeline, QuasAr6a is a very fast GEVI. On the other hand, the GEVIs GR(V80D) and GR(A242R) are slow GEVIs.

The smoothed signal trend of a very fast GEVI is different from that of a slow GEVI in the time span between 0.525 to 0.54 seconds. The excitation of spiking HEK cell starts at 0.5 seconds. This time span is decided by cross-checking with the 7 different movies of slow GEVIs and the 7 different movies of fast GEVIs used to build the image analysis pipeline. This time range, however, is variable as it depends on the duration of the movie.

The 'gradient' function of python's NumPy library is used to highlight the difference in smoothed signal trends of very fast and slow GEVIs. The gradient function computes the gradient of the data points. Second order accurate central differences is used to compute the gradient of the interior data points whereas either first or second order accurate one-sides (forward or backwards) differences is used to compute the gradient of the boundary points [28]. For very fast GEVIs, the gradient values are both positive and negative whereas for slow GEVIs, the gradient values are all positive in the specified time span. If the GEVI is identified as a very fast GEVI then the speed of the GEVI is not calculated. But if the GEVI is identified as a slow GEVI, the speed of the GEVI is calculated.

Once it is identified that the GEVI is a slow GEVI, best curve fit is now performed to the original averaged photobleaching corrected signal. A bi-exponential curve fit is performed to the signal in the time span between 0.515 to 0.54 seconds. This time span is where the slow GEVI does not keep up with the voltage change. The time span is decided by cross-checking with 14 different single spiking HEK cell movies. This time range, however, is variable as it depends on the duration of the movie. The 'curve_fit' function of python's scipy.optimize module is used to perform the curve fitting. The bi-exponential function is defined as given in equation 3.4.2a

$$a * e^{x/x1} + b * e^{x/x2} \tag{3.4.2a}$$

Bi-exponential function is selected after cross-checking with the 7 different single spiking HEK cell movies expressing the slow GEVIs GR(V80D) and GR(A242R). The bi-exponential function provides a better overall fit to the data points as compared to a mono-exponential function. To calculate the fast and slow time constants, equation 3.4.2a is co-related to equation 3.4.2b from [29]

$$F(t) = A * (C * e^{-t/t1} + (1 - C) * e^{-t/t2}) \tag{3.4.2b}$$

where $t_1$ is the fast time constant and $t_2$ is the slow time constant.

After performing a bi-exponential fit to the data points in the time range 0.515 to 0.54 seconds, the values of a, $x_1$, b and, $x_2$ of equation 3.4.2a are extracted. The slow time constant is taken as the minimum of $x_1$ and $x_2$. The fast time constant is taken as the maximum of $x_1$ and $x_2$. The variable C from equation 3.4.2b is calculated as given in equation 3.4.2c

$$c = a/(a+b) \tag{3.4.2c}$$

where c is the contribution of the time constants. a and b are the variables defined in equation 3.4.2a.

After cross-checking with the 7 different single spiking HEK cell movies expressing the slow GEVIs GR(V80D) and GR(A242R) used to build the image analysis pipeline, the following conditions are set for speed calculations :-

1. If c $<$ 0.1 or c = 1, then speed of the slow GEVI is calculated from fast time constant only as given in equation 3.4.2d

$$Speed(seconds) = 1/FastTimeConstant \tag{3.4.2d}$$

2. If c $>$ 1, then speed of the slow GEVI is calculated from slow time constant only as given in equation 3.4.2e

$$Speed(seconds) = 1/SlowTimeConstant \tag{3.4.2e}$$

3. If conditions 1 and 2 are not met, then speed of the slow GEVI is calculated as given in equation 3.4.2f

$$Speed(seconds) = 1/((FastTimeConstant * (1-c)) + (SlowTimeConstant * c)) \tag{3.4.2f}$$

where c is the contribution of slow time constant and 1-c is the contribution of fast time constant to speed calculations.

The next section describes how the membrane localization feature of the GEVIs expressed by the segmented single spiking HEK cell is calculated in the built image analysis pipeline.

### 3.3.3. Membrane Localization

An additional feature of the GEVIs, extracted in the image analysis pipeline is the membrane intensity to whole cell intensity ratio. The whole cell comprises both the membrane and the cytoplasm of the spiking HEK cell. The membrane localization feature of the GEVIs expresses how well the GEVIs is localized to the membrane of the spiking HEK cell. Since GEVIs are membrane proteins, the membrane localization feature of the GEVIs is ideally higher than 1. This shows that most of the calculated intensity is from the membrane of the cell. Thereby, this indicates that the GEVIs are localized to the membrane of the spiking HEK cell.

In the RGB mean + correlation images of the single spiking HEK cells, the membrane of the single spiking HEK cell cannot be identified. In order to identify the true membrane of the spiking HEK cell, a time averaged image of the motion corrected movie is obtained. This time averaged image is a 2-D image of a 3-D dataset. A 2-D image of the motion corrected movie could also be obtained by taking, for example, the median intensity or, minimum intensity of the 3-D dataset [30]. In the image analysis pipeline, the time averaged image of the motion corrected movie is obtained using the 'mean' function of python's NumPy library. This method of obtaining a 2-D image of the motion corrected movie is chosen after cross-checking the membrane visibility in all of the 14 different single spiking HEK cell movies used to build the image analysis pipeline.

The next step is to remove the background in the entire time averaged image. Background subtraction from the entire image is done to enhance the visibility of the spiking HEK cell while preserving the edges in the image. These edges correspond to the membrane of the spiking HEK cell. In the image analysis pipeline, background subtraction of the time averaged image is performed using the 'bilateralFilter' function of python's OpenCV library. This function uses the bilateral filter.

The bilateral filter is an edge preserving filter that smooths the image while keeping the edges in the image fairly sharp. The bilateral filter works by replacing each pixel's value with a weighted average of the nearby pixels. The weights depend on both the spatial distance and the intensity difference between the pixels [31]. A filter size of 75 is used. This filter size is robust with all of the 14 different single spiking HEK cell movies used to build the image analysis pipeline.

The results of bilateral filtering is also compared to the results of another edge preserving filter, that is, the median filter. The median filter calculates the median of the data points contained within the specified window size. It then replaces each entry within the specified window size by the calculated median value of the data points [32].

Since the 4 times dilated binary mask contains the pixels of high intensities, this mask is used to get the membrane of the spiking HEK cell. As stated before, GEVIs are membrane proteins. Therefore, the pixels with high intensities belong to where the GEVIs are localized in the single spiking HEK cell. The background subtracted time averaged image is of the same image size as the movie frames. Therefore, the 4 times dilated binary mask is overlayed on the background subtracted time averaged image. Now, the mean intensity of the pixels contained in the mask is calculated. This mean intensity corresponds to the whole cell mean intensity of the single spiking HEK cell.

To get the membrane of the single spiking HEK cell, 'findContours' and 'drawContours' functions of python's OpenCV libary are used. The 'findContours' function finds the contour of the 4 times dilated binary mask. This contour corresponds to the membrane of the single spiking HEK cell. The found contour from the 4 times dilated binary mask is of 1 pixel thickness and 4 points connected. A contour of 1 pixel thickness is chosen to get the outer contour of the 4 times dilated binary mask. However, due to time constraints and lesser importance of the membrane localization feature, this pixel thickness value is not explored further. On the other hand, a 4 points connected contour is chosen due to lesser memory requirements of a 4 points connected contour as compared to a greater than 4 points connected contour.

The 'drawContours' function creates a new binary mask of the same size as the 4 times dilated binary mask. This newly created binary mask contains the pixels of the found contour. The newly created binary mask containing pixels of the membrane of the single spiking HEK cell is overlayed on the background subtracted time averaged image. Now, the mean intensity of this region is calculated. This mean intensity corresponds to the mean intensity of the membrane of the single spiking HEK cell. Finally, the membrane localization is calculated as given in equation 3.4.3a

$$MembraneLocalization = MeanIntensityOfMembrane/MeanIntensityOfWholeCell \quad (3.4.3a)$$

The next chapter provides the results of the various components of the image analysis pipeline and the results of the GEVIs used to build the image analysis pipeline. In addition, the results of the screening of a population of spiking HEK cells expressing variants of GEVI is also presented.

# 4

# Results

## 4.1. Results of the Image Analysis Pipeline

Figure 4.1 shows the segmentation results of one of the single cell spiking HEK cell movies expressing the GEVI GR(A242R). Fig. 4.1a shows the RGB mean + correlation image of the single spiking HEK cell fed to the re-trained U-Net model. Fig. 4.1b shows the predicted binary mask. Fig. 4.1c, d are the dilated binary masks. Fig. 4.1e is the background mask obtained by subtracting Fig. 4.1c from Fig. 4.1d.

The 4 times dilated binary mask (fig. 4.2a) is then overlayed on the motion corrected movie frames to obtain the mean intensity plot across movie frames of the segmented single spiking HEK cell. This plot is shown in Fig. 4.2b. Figure 4.2b comprises signal from the GEVIs, background signal and, the photobleached signal. Similarly, the background binary mask (fig. 4.3a) is overlayed on the motion corrected movie frames to obtain the mean intensity plot across movie frames of the local background (fig. 4.3b). Figure 4.3b depicts the background signal.

Figure 4.4 depicts the different intensity plots across movie frames obtained in order to calculate the sensitivity of the GR(A242R) GEVI expressed by the segmented single spiking HEK cell. Figure 4.4a is the background corrected signal obtained by subtracting the signal in Fig. 4.3b from the signal in Fig. 4.2b. Figure 4.4b depicts the photobleached signal present in Fig. 4.4a. Figure 4.4c is the normalized fluorescence intensity plot obtained by dividing the signal in Fig. 4.4a and 4.4b. The peaks in Fig. 4.4c are averaged to obtain a single peak as shown in Fig. 4.4d. This averaging of the signal is done across the duration of the movie for simpler calculation of the sensitivity of the GEVIs. The sensitivity from Fig. 4.4d is calculated as described in section 3.4.1. The sensitivity of this particular GR(A242R) GEVI is obtained as 21.17%.

Figure 4.5 depicts the different intensity plots across movie frames obtained in order to calculate the speed of the GR(A242R) GEVI expressed by the segmented single spiking HEK cell. Figure 4.5a is the mean intensity plot across movie frames of the segmented single spiking HEK cell, same as the one shown in Fig. 4.2b. Figure 4.5b is the photobleached signal present in Fig. 4.5a. Figure 4.5c is the photobleaching corrected signal obtained by subtracting Fig. 4.5b from Fig. 4.5a. The peaks in Fig. 4.5c are averaged to get a single peak as shown in Fig. 4.5d. This averaging of the signal is done across the duration of the movie for simpler calculation of the speed of the GEVIs. The signal in Fig. 4.5d is smoothed as described in section 3.4.2. Smoothing the signal in Fig. 4.5d gives Fig. 4.6a.

Figure 4.6 shows the difference in the signal trends of a very fast and a slow GEVI. The region circled in red in Fig. 4.6. is the region where very a fast GEVI can keep up with the voltage change (Fig. 4.6b) and a slow GEVI cannot keep up with the voltage change (Fig. 4.6a). The gradients of the circled region are estimated to distinguish between a very fast and a slow GEVI.

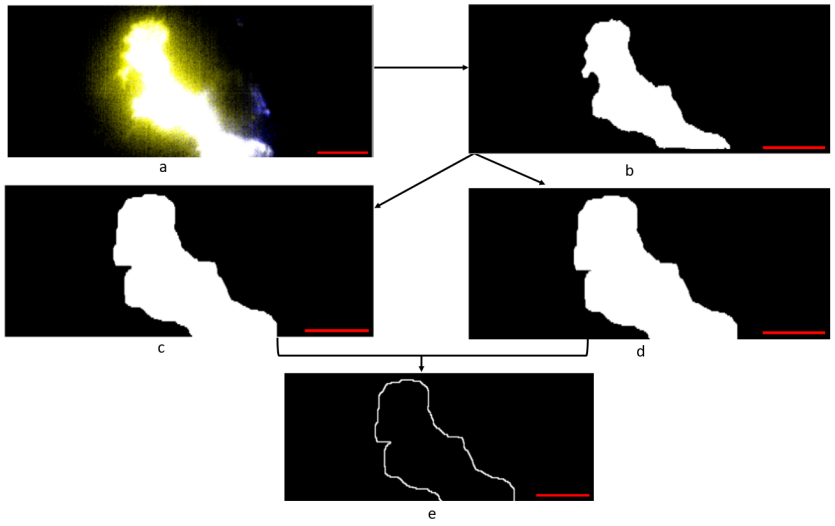



**Figure 4.1: Segmentation Results. a:** The RGB mean + correlation image of a spiking HEK cell. Image of size 500x208 pixels. **b:** The predicted binary mask of the spiking HEK cell. Image of size 500x208 pixels. **c:** The 4 times dilated binary mask. Image of size 500x208 pixels. **d:** The 5 times dilated binary mask. Image of size 500x208 pixels. **e:** The local background binary mask. Image of size 500x208 pixels. Scale bars = 20$\mu$m

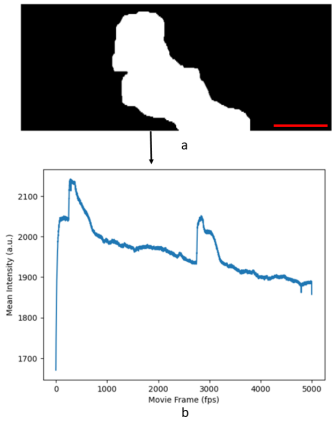



**Figure 4.2: Mean intensities across movie frames plot of segmented spiking HEK cell. a:** 4 times dilated binary mask of size 500x208 pixels. **b:** Mean intensity across movie frames plot of segmented spiking HEK cell. Scale bar = 20$\mu$m

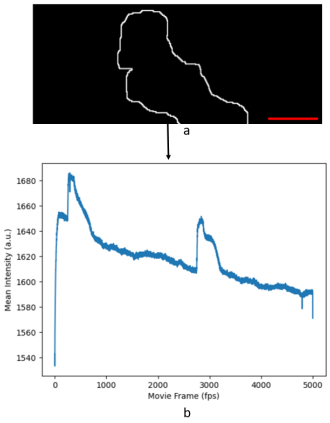



**Figure 4.3: Mean intensities across movie frames plot of local background. a:** Local background binary mask of size 500x208 pixels. **b:** Mean intensity across movie frames plot of background. Scale bar = 20$\mu$m

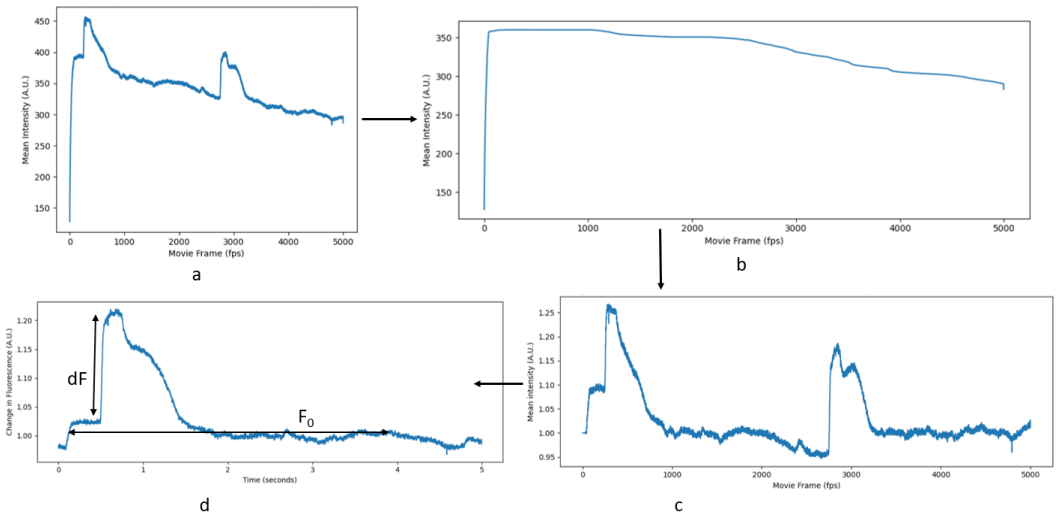



**Figure 4.4: Intensity plots across movie frames and movie duration for sensitivity calculation. a:** Background corrected signal across movie frames **b:** Photobleached signal across movie frames **c:** Background and photobleaching corrected signal across movie frames **d:** Averaged background and photobleaching corrected signal across duration of the movie. Sensitivity $(\%)$ is calculated as $dF/F_0$

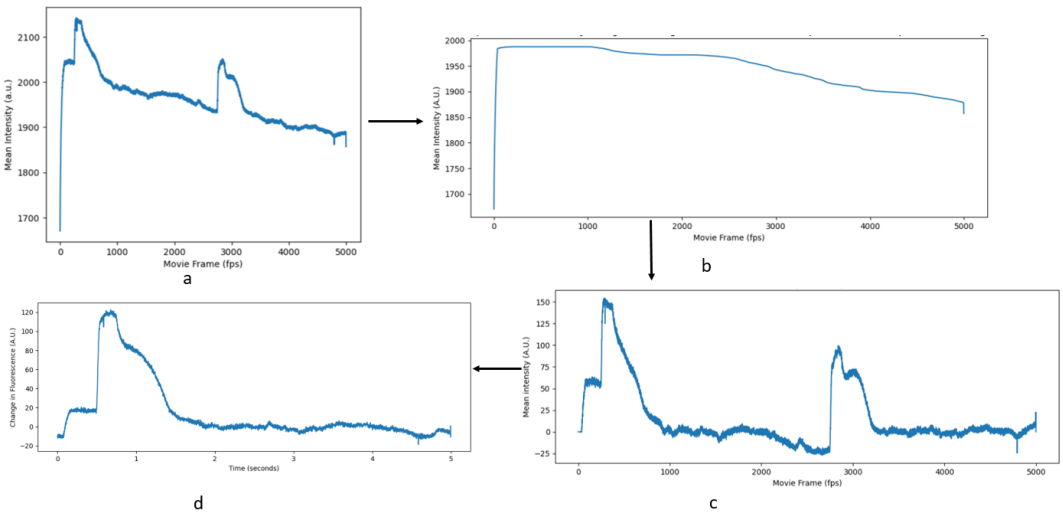



**Figure 4.5: Intensity plots across movie frames and movie duration for speed calculation. a:** Original mean intensity plot across movie frames of segmented single spiking HEK cell **b:** Photobleached signal across movie frames **c:** Photobleaching corrected signal across movie frames **d:** Averaged photobleaching corrected signal across duration of the movie

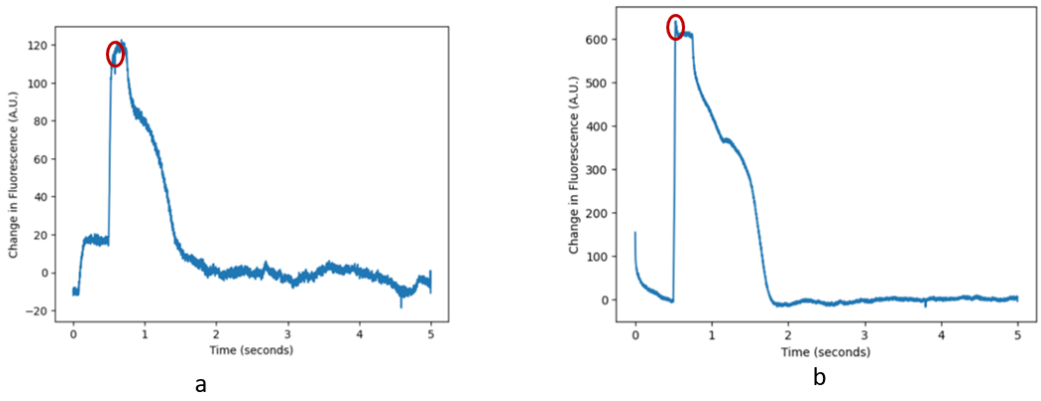



**Figure 4.6:** Signal trends of GEVIs. **a:** Signal trend of a slow GEVI (GR(A242R) GEVI) **b:** Signal trend of a very fast GEVI (QuasAr6a GEVI). The circled region in red highlights the difference in signal trends. In the circled region, QuasAr6a shows a sharp peak whereas GR(A242R) doesn't

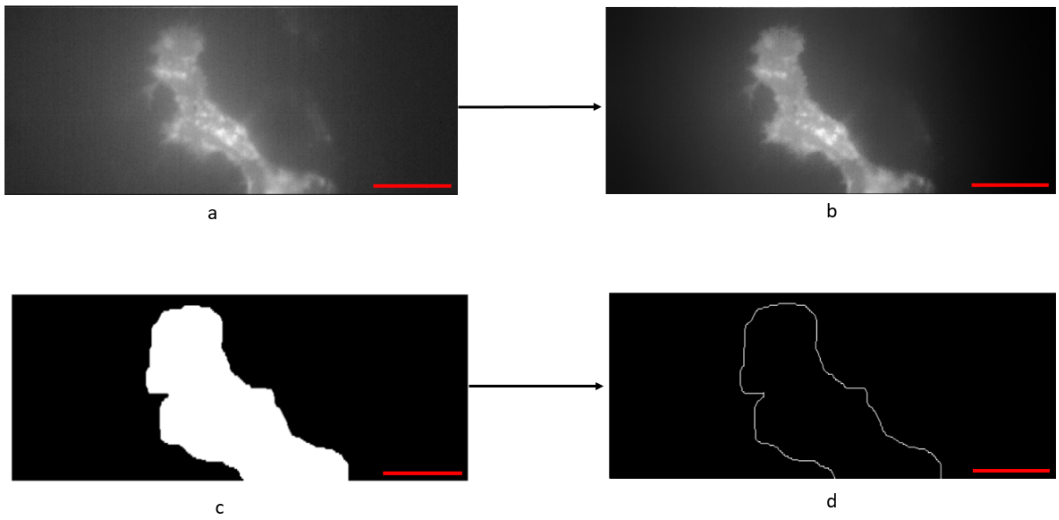



**Figure 4.7: Membrane localization calculation. a:** Time averaged 2-D image of motion corrected movie. Image of size 500x208 pixels. **b:** Background subtracted time averaged image. Image of size 500x208 pixels. **c.** Whole cell. Image of size 500x208 pixels. **d.** Membrane of spiking HEK cell. Image of size 500x208 pixels. Scale bars = 20$\mu$m

GR(A242R) is a slow GEVI. To estimate the speed of this GR(A242R) GEVI, a bi-exponential fit is performed in the circled region in red of Fig. 4.6a. Based on the conditions described in section 3.4.2, the speed of this particular GR(A242R) GEVI is estimated as 91.51/seconds.

Figure 4.7 depicts the calculation of membrane localization feature of the GR(A242R) GEVI. Figure 4.7a is the time averaged 2-D image of the motion corrected movie. This time averaged image undergoes bilateral filtering to get the background subtracted time averaged image (Fig. 4.7b). The 4 times dilated binary mask (Fig. 4.7c) is overlayed on Fig. 4.7b to calculate the mean intensity of the whole cell. Contours of Fig. 4.7c are found to get the membrane mask (Fig. 4.7d). This membrane mask is overlayed on Fig. 4.7b to calculate the mean intensity of the membrane. The ratio of the obtained mean intensities is calculated as given in equation 3.4.3a. The membrane localization of this particular GR(A242R) GEVI is calculated as 0.807.

The next section shows the results of different choices and algorithms other than the ones used to build the current image analysis pipeline.

### 4.1.1. Other Choices and Algorithms

This section shows the results of wrong hyperparameter values set to the pre-trained U-Net model to re-train on HEK cell data. In addition, the section also shows the results of using moving average filter for obtaining the photobleaching signal in spiking HEK cell movies, performing mono-exponential fit when calculating the speed of the GEVIs expressed by the single spiking HEK cell, different methods of obtaining a 2-D image from the 3-D spiking HEK cells datasets and, results of median filter for background subtraction in time averaged images of the spiking HEK cell movies for membrane localization feature calculations.

Figure 4.8 depicts the predicted binary mask (Fig. 4.8b) of a re-trained U-Net model trained on RGB mean images (Fig. 4.8a) of the motion corrected movie instead of RGB mean + correlation images. These RGB mean images of the motion corrected movies are obtained from patch clamp movies of single HEK cells. The re-trained U-Net model is not able to predict pixels with low intensity values resulting in the binary mask shown (Fig. 4.8b).

Figure 4.9 depicts the predicted binary mask (Fig. 4.9b) of a re-trained U-Net model trained on RGB mean + correlation images (Fig. 4.9a) of the motion corrected movie. However, the number of epochs hyperparameter set for this re-trained model is different than the one described in section 3.3. The number of epochs set for this re-trained model is 10. Figure 4.9 shows that the re-trained U-Net model experiences over-fitting. Therefore, the re-trained U-Net model is now able to predict only an underlying pattern, that is, the contour of the spiking HEK cell as shown in fig. 4.9b.

Figure 4.10 compares the results of median filtering and moving average filtering in order to obtain the photobleached signal. Fig. 4.10a is the signal of the segmented single spiking HEK cell, same as the one shown in Fig. 4.2b. When a median filter is applied to the signal in Fig. 4.10a, the photobleached signal is directly obtained as shown in Fig. 4.10b. When a moving average filter is applied to Fig. 4.10a, the signal is heavily smoothed as shown in Fig. 4.10c. Subtracting the signal in Fig. 4.10c from the signal in Fig. 4.10a for photobleaching corrections only leads to significant loss in signal. Therefore, a moving average filter is not effective in photobleaching corrections of spiking HEK cell movies expressing GEVIs.

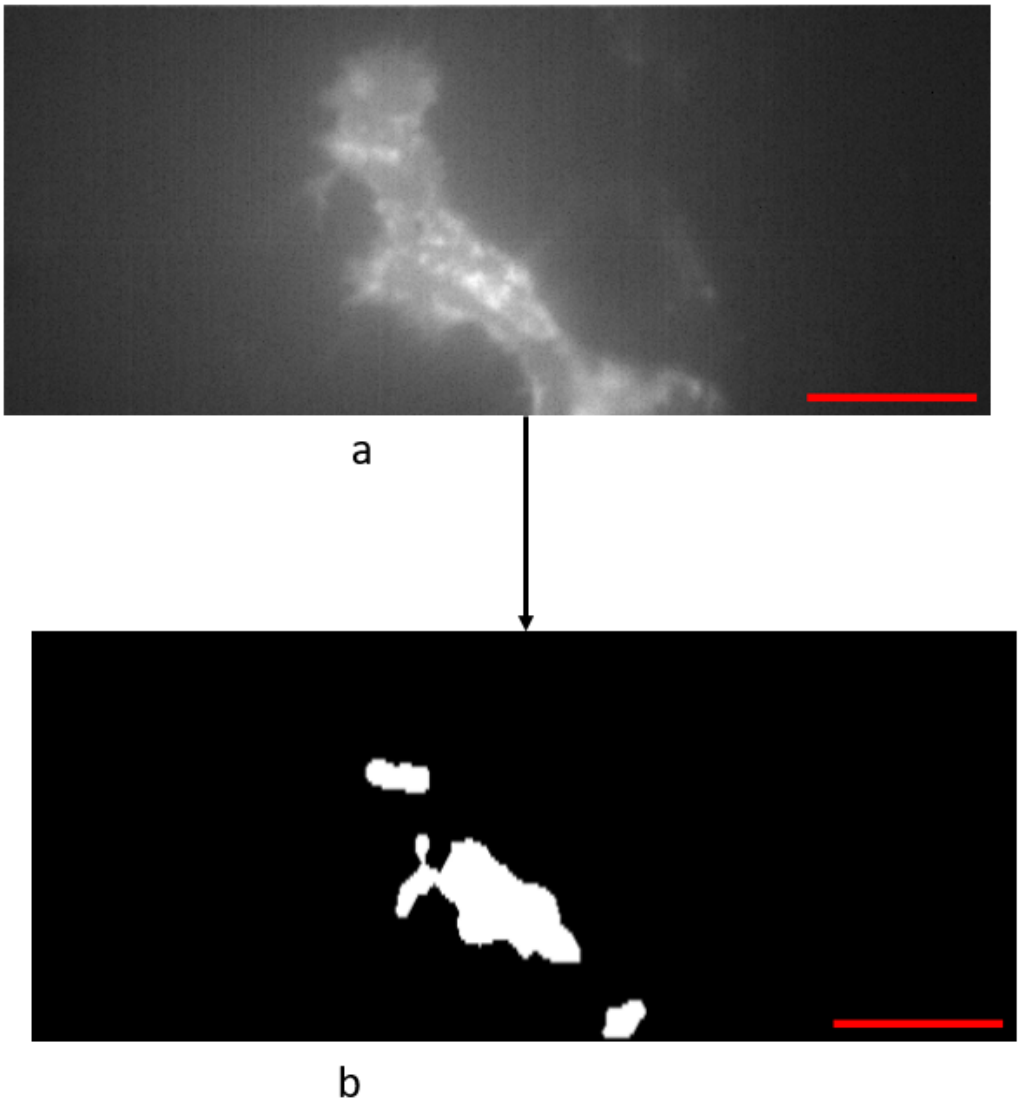



**Figure 4.8: Predicted binary mask of a re-trained U-Net model trained on RGB mean images of motion corrected patch clamp HEK cell movies**. **a:** Input image to the re-trained U-Net model. Image of size 500x208 pixels. **b:** Predicted binary mask. Image of size 500x208 pixels. Scale bars = 20$\mu$m

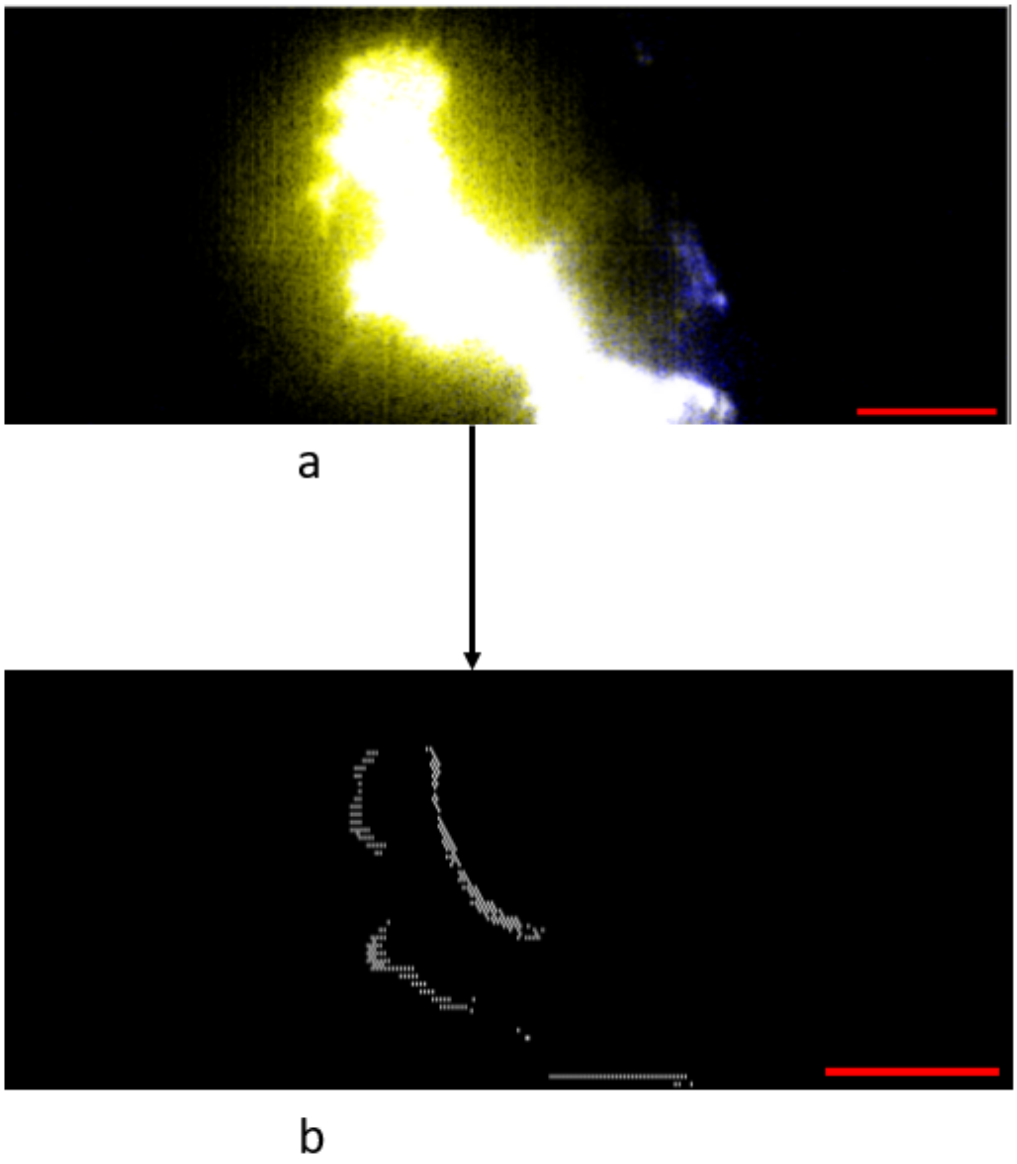



**Figure 4.9: Predicted binary mask of a re-trained U-Net model trained with number of epochs as 10**. **a:** Input image to the re-trained U-Net model. Image of size 500x208 pixels. **b:** Predicted binary mask. Image of size 500x208 pixels. Scale bars = 20$\mu$m

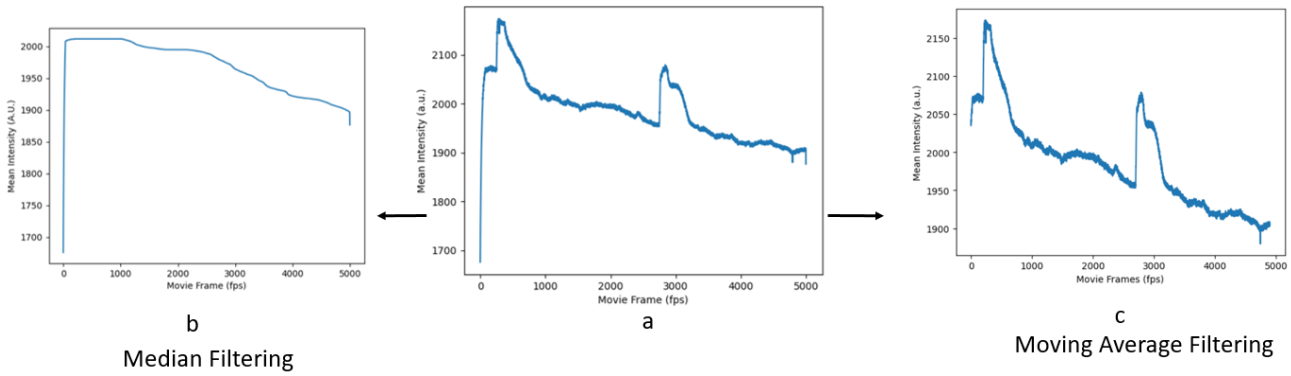



**Figure 4.10: Signal filtering using 2 different filters for obtaining the photobleached signal**. **a:** Original signal of segmented single spiking HEK cell expressing GEVIs. **b:** Signal after applying a median filter. **c:** Signal after applying a moving average filter

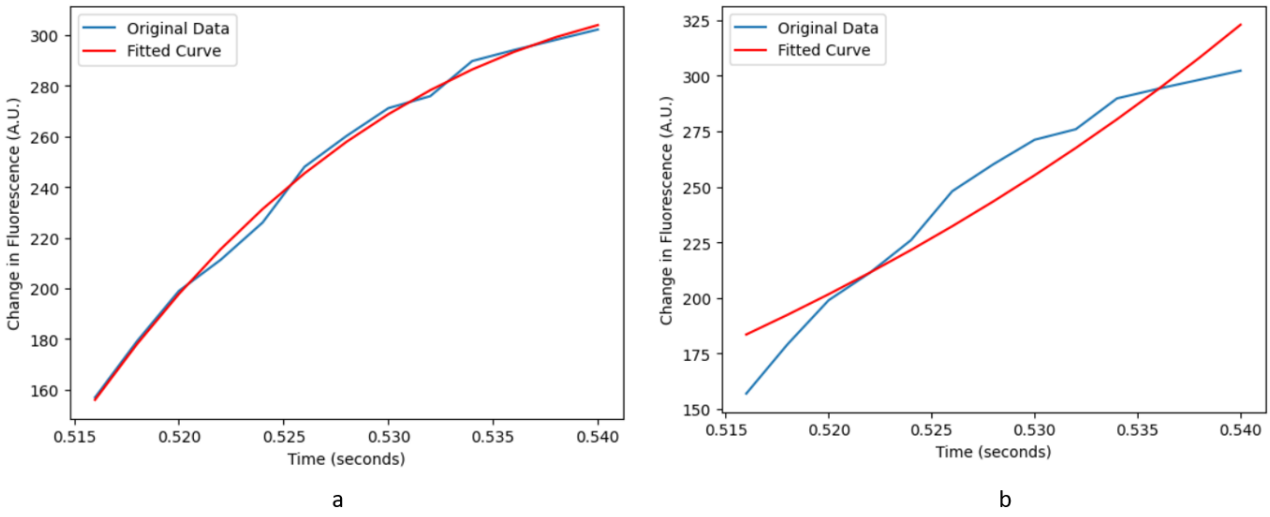



**Figure 4.11: Exponential fits performed for speed calculation of a slow GEVI**. **a:** Bi-exponential fit **b:** Mono-exponential fit

Figure 4.11 compares the results of a bi-exponential fit (Fig. 4.11a) versus a mono-exponential fit (Fig. 4.11b) for the speed calculation of a slow GEVI (GR(V80D) in this figure). Bi-exponential fit (fig. 4.11a) fits all of the data points required to calculate the speed of the GR(V80D) GEVI. The mono-exponential fit (fig. 4.11b), on the other hand, is performed sub-optimally. Despite trying out various parameter settings for mono-exponential fitting of the data points, the mono-exponential fit doesn't fit all of the data points in the selected time range for speed calculations of the 7 different single spiking HEK cell movies expressing slow GEVIs. Due to shortage of time, mono-exponential fitting is not optimized any further.

Figure 4.12 compares the results of different methods of obtaining a 2-D image of the motion corrected movie. There is very little difference in the intensity variations of the 2-D images obtained by taking the mean intensity of the motion corrected movie (Fig. 4.12a) and by taking the median intensity of the motion corrected movie (Fig. 4.12b). However, for this particular motion corrected movie (Fig. 4.12), obtaining the 2-D image by taking the median intensity of the dataset takes 38 seconds whereas obtaining the 2-D image by taking the mean intensity of the dataset takes 3 seconds on the HP ENVY x360 Convertible 13-bd0xxx system. On the other hand, taking the minimum intensity of the dataset (Fig. 4.12c) to obtain a 2-D image of the 3-D dataset is not suitable as the membrane of the spiking HEK cell is not quite visible in the image.

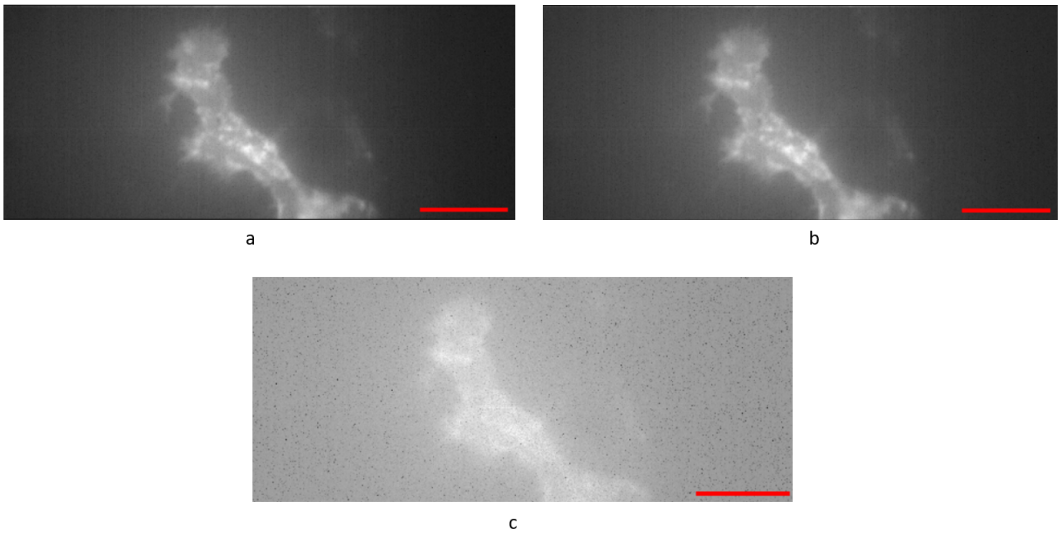



**Figure 4.12: 2-D images of the motion corrected movie.**. **a:** 2-D image obtained by taking the mean intensity of the 3-D dataset. Image of size 500x208 pixels. **b:** 2-D image obtained by taking the median intensity of the 3-D dataset. Image of size 500x208 pixels. **c:** 2-D image obtained by taking the minimum intensity of the 3-D dataset. Image of size 500x208 pixels. Scale bars = 20$\mu$m

Figure 4.13 compares the results of bilateral filtering (Fig. 4.13a) and median filtering (Fig. 4.13b) for background subtraction in the time averaged image. The region circled in green shows the part of the cell membrane of the spiking HEK cell that is sharper in the bilateral filtered image as compared to the median filtered image. However, this difference is very small.

The next section describes how the results of the components of pipeline have been evaluated in order to estimate the quality of the image analysis pipeline.

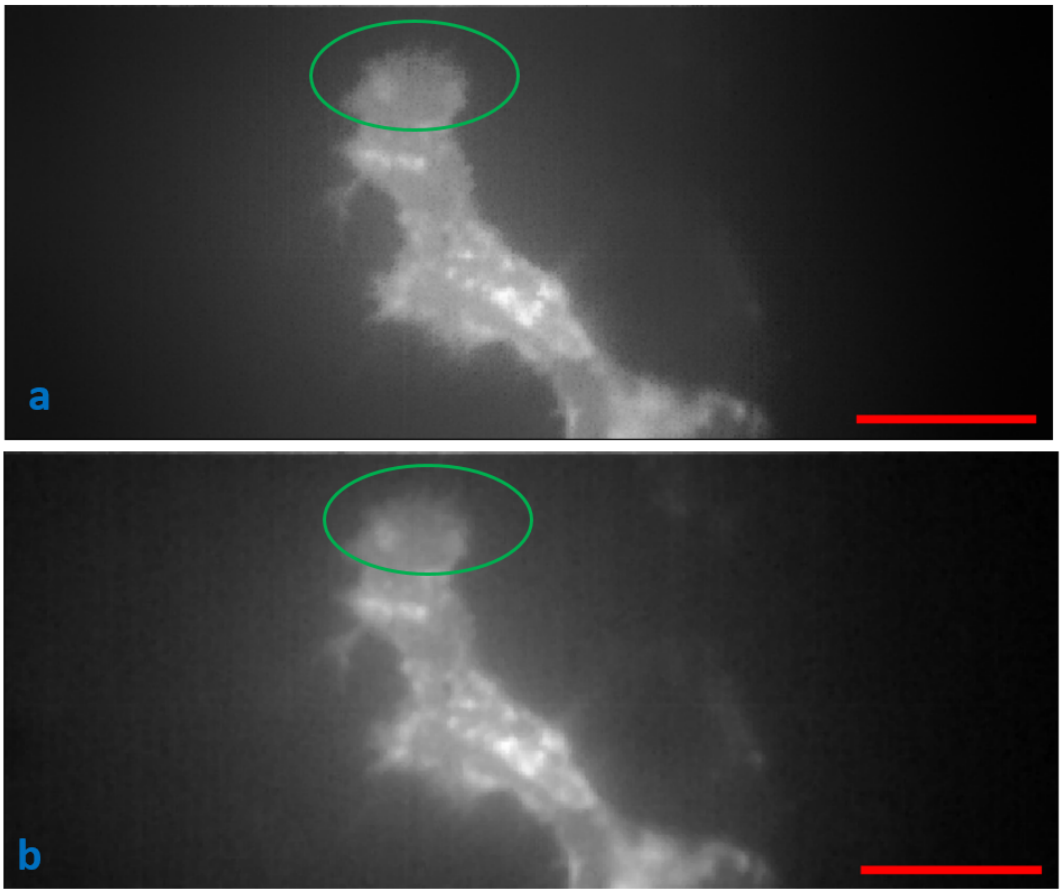



**Figure 4.13: Edge preserving filters used for background subtraction of 2-D images**. **a:** Bilateral Filter. Image of size 500x208 pixels. **b:** Median Filter. Image of size 500x208 pixels. Region circled in green shows the difference in edges. Scale bars = 20$\mu$m

### 4.1.2. Quality of the Image Analysis Pipeline

#### Motion correction

To estimate the quality of performed motion correction, the pearson product-moment correlation coefficients evaluation metric is used. This evaluation metric is a widely used metric to compute the degree of similarity between two different images. The pearson product-moment correlation coefficient is calculated as given in equation 4.1.2a [33]

$$r = \Sigma(x_i - \bar{x})(y_i - \bar{y}) / \sqrt{\Sigma(x_i - \bar{x})^2 (y_i - \bar{y})^2} \tag{4.1.2a}$$

where $x_i$ and $y_i$ are the data points and $\bar{x}$ and $\bar{y}$ is the mean of the x- and y- data points. r = 1, represents a perfect correlation between the two images. For the single cell spiking HEK cell movies, the first movie frame is taken as the reference image. The correlation coefficient between the first movie frame and all other frames of the original movie of one of the single cell spiking HEK cell movies, is computed. The 'corrcoef' function of python's NumPy library is used for this computation. The average of all of the correlation coefficients is then computed using the 'mean' function of python's NumPy library. This average value of correlation coefficients between the first movie frame and all other frames of the original movie is obtained as 0.9645. Similarly, the pearson product-moment correlation coefficients between the first movie frame and all other frames of the motion corrected movie is computed. The average of all of the correlation coefficients is then computed. This average value of correlation coefficients between first movie frame and all other frames of the motion corrected movie of the single spiking HEK cell is obtained as 0.9657. This shows that motion correction improves the stability and consistency of pixels across all of the movie frames relative to the first frame of the movie. However, the difference between the two obtained averaged correlation coefficients is small.

#### Segmentation

To estimate the quality of segmentation of the re-trained U-Net model, intersection over union (IoU) evaluation metric is used. IoU is a popular evaluation metric for object detection. It is the ratio of intersection of two boxes' areas to their combined areas. IoU is calculated as given in equation 4.1.2b [34]

$$IntersectionOverUnion(IoU) = A \cap B / A \cup B \tag{4.1.2b}$$

The IoU between the manually annotated ground truth mask and the predicted binary mask of the re-trained U-Net model is compared with the IoU between the ground truth mask and the masks obtained by cellpose segmentation algorithm and the available mask creation algorithm on the octoscope. The cellpose segmentation algorithm is a widely used generalist algorithm for cellular segmentation [35]. This comparison is made to check whether the newly re-trained U-Net model performs better than the already available segmentation algorithm on the octoscope and also if it performs better than a widely used segmentation algorithm in microscopy labs.

Figure 4.14 shows the manually drawn ground truth mask (Fig. 4.14b), the mask predicted by the re-trained U-Net model (Fig. 4.14c), the mask predicted by the cellpose segmentation algorithm (Fig. 4.14d) and the mask predicted by the available mask creation algorithm on the octoscope (Fig. 4.14e) along with their IoU values. The RGB mean + correlation image (Fig. 4.14a) used is that of a HEK cell. Figure 4.14a is one of the images of the testing dataset used to test the re-trained U-Net model. The IoU of different masks is computed for all of the 28 images of the testing dataset. Table 4.1 shows the average IoU values and their standard error of the mean for different masks of the testing dataset.

| Masks Used | Average IoU |
|---|---|
| Ground Truth Masks and Re-trained U-Net Predicted Masks | $0.864 \pm 0.014$ |
| Ground Truth Masks and Cellpose Algorithm Predicted Masks | $0.532 \pm 0.039$ |
| Ground Truth Masks and Predicted Masks of the Mask Creation Algorithm on the octoscope | $0.772 \pm 0.034$ |

**Table 4.1: Average IoU and their standard error of mean for different masks**

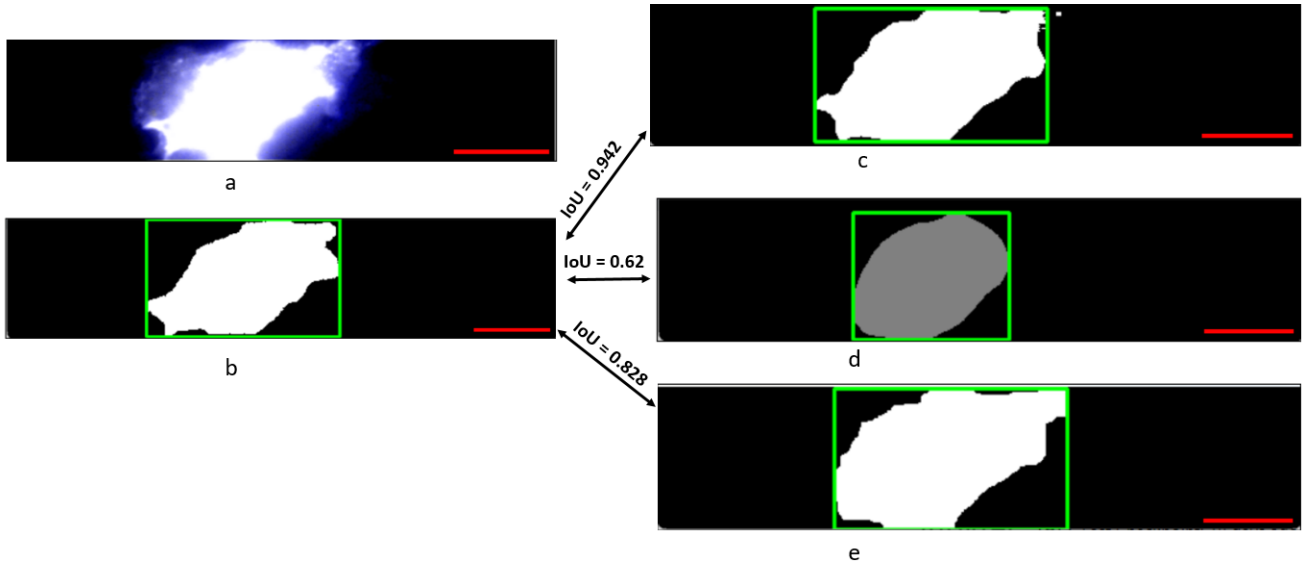



**Figure 4.14: IoU of different masks. a:** RGB mean + correlation image of single HEK cell. Image of size 496x109 pixels. **b:** Ground truth mask. Image of size 496x109 pixels. **c.** Re-trained U-Net predicted mask. Image of size 496x109 pixels. **d.** Cellpose predicted mask. Image of size 496x109 pixels. **e:** Predicted mask of the mask creation algorithm on the octoscope. Image of size 496x109 pixels. Scale bars = 25$\mu$m in the x-direction and 35$\mu$m in the y-direction

Additionally, one-way ANNOVA is performed on the 28 IoU values of different masks. The 'f_oneway' function of python's scipy.stats module is used for one-way ANNOVA computations. The obtained F-statistic value is 29.53 and the obtained P-value is $2.322e^{-10}$. This p-value indicates that there is a significant difference between the 3 different IoU means. Therefore, the re-trained U-Net model outperforms cellpose segmentation algorithm and the available mask creation algorithm on the octoscope for spiking HEK cell segmentation.

### Features extraction

To estimate the accuracy of the values of the features extracted by the image analysis pipeline, the values of sensitivity, speed and, membrane localization are compared to the values of these features extracted in FIJI. For sensitivity and speed, a ROI is drawn on the motion corrected movie in FIJI. The mean intensity profile of the ROIs is obtained in FIJI. This mean intensity profile is the mean intensity of the ROI calculated in each of the motion corrected movie frames. The obtained data is saved as a .csv file. This .csv file is then read in python. From the read data, the mean intensity plots of the spiking HEK cell and the local background are obtained for 14 different single cell spiking HEK cell movies. The algorithms described in section 3.3.1 and 3.3.2 are then used to calculate the sensitivity and speed values of the 14 different single cell spiking HEK cell movies expressing different GEVIs. As mentioned before, 4 single cell movies of spiking HEK cell expressing the GEVI GR(A242R), 3 single cell movies of spiking HEK cell expressing the GEVI GR(V80D) and 7 single cell movies of spiking HEK cell expressing the GEVI QuasAr6a are used.

Figure 4.15 shows the sensitivity values of different GEVIs. These values are obtained from the predicted binary masks of the image analysis pipeline and drawn ROIs in FIJI.

Figure 4.16 shows the sensitivity values of all of the 14 different single cell spiking HEK cell movies expressing different GEVIs. These values are obtained from the predicted binary masks of the image analysis pipeline and the drawn ROIs in FIJI.

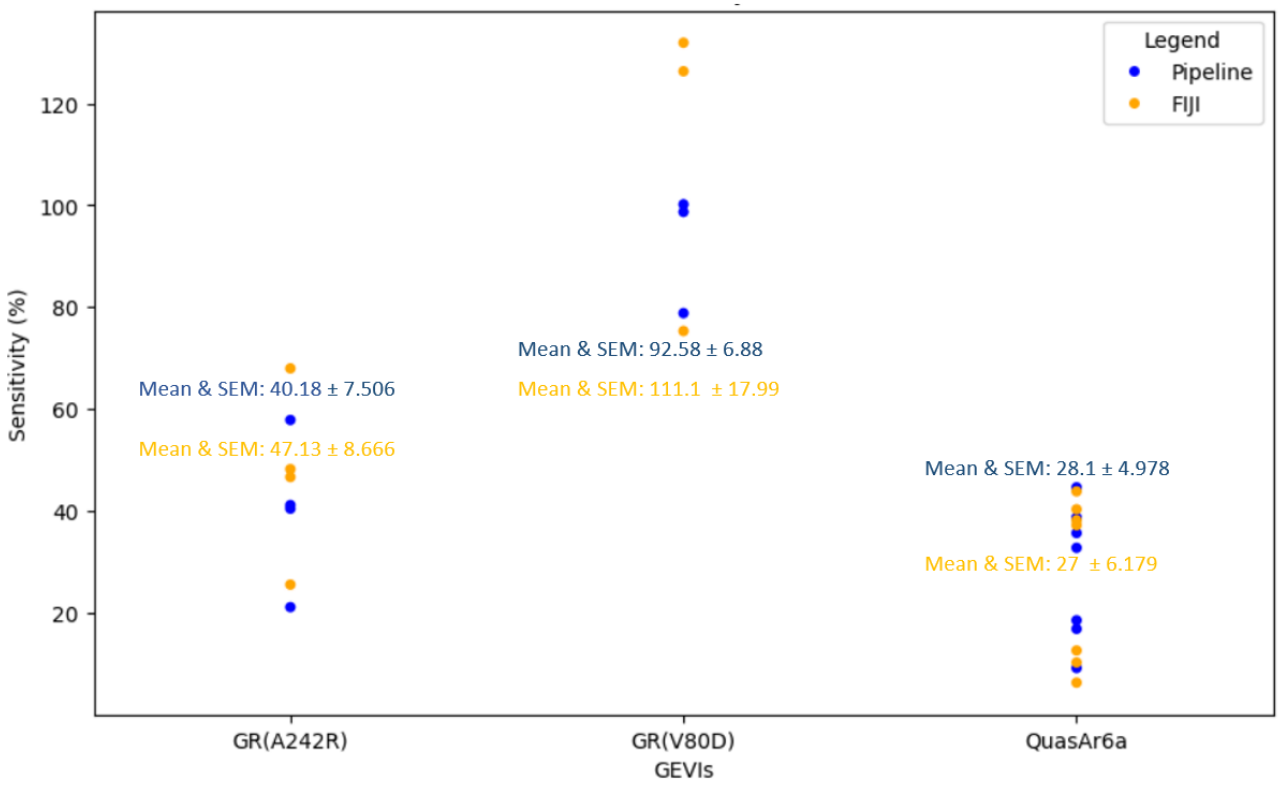



**Figure 4.15: The sensitivity values of different GEVIs obtained from the predicted binary masks of the image analysis pipeline compared to the sensitivity values of different GEVIs obtained by drawing ROIs in FIJI. The values in blue are the values obtained from the image analysis pipeline and the values in orange are the values obtained from the drawn ROIs in FIJI. SEM in the figure stands for standard error of the mean**

Table 4.2 depicts the percentage error in the averages of sensitivities of different GEVIs. This percentage error in the averages is calculated as given in equation 4.1.2c

$$PercentageError(\%) = (MeasuredValue - TrueValue)/TrueValue \tag{4.1.2c}$$

where measured value is the mean value obtained from the image analysis pipeline results. True value is the mean value obtained from the FIJI results.

The percentage error in averages of sensitivity values of all of the 14 different single cell spiking HEK cell movies expressing different GEVIs is 10.672%. The sensitivity values of the GEVIs are underestimated by the built image analysis pipeline.

| GEVIs | Percentage Error (%) |
|---|---|
| GR(A242R) | 14.746 |
| GR(V80D) | 16.669 |
| QuasAr6a | 4.074 |

**Table 4.2: Percentage error in averages of sensitivities of different GEVIs**

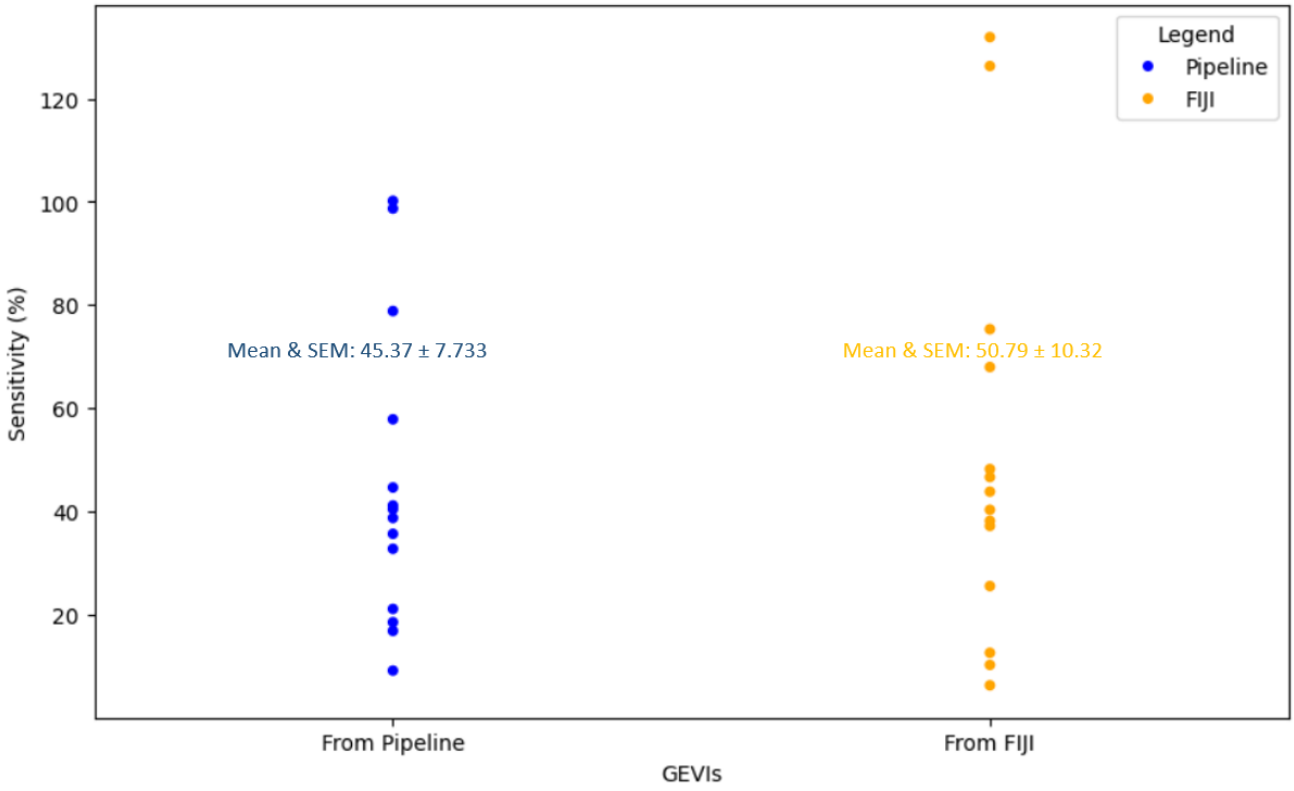



**Figure 4.16: The sensitivity values of 14 different GEVIs obtained from the predicted binary masks of the image analysis pipeline compared to the sensitivity values obtained from drawn ROIs in FIJI. The values in blue are the values obtained from the image analysis pipeline and the values in orange are the values obtained from the drawn ROIs in FIJI. SEM in the figure stands for standard error of the mean**

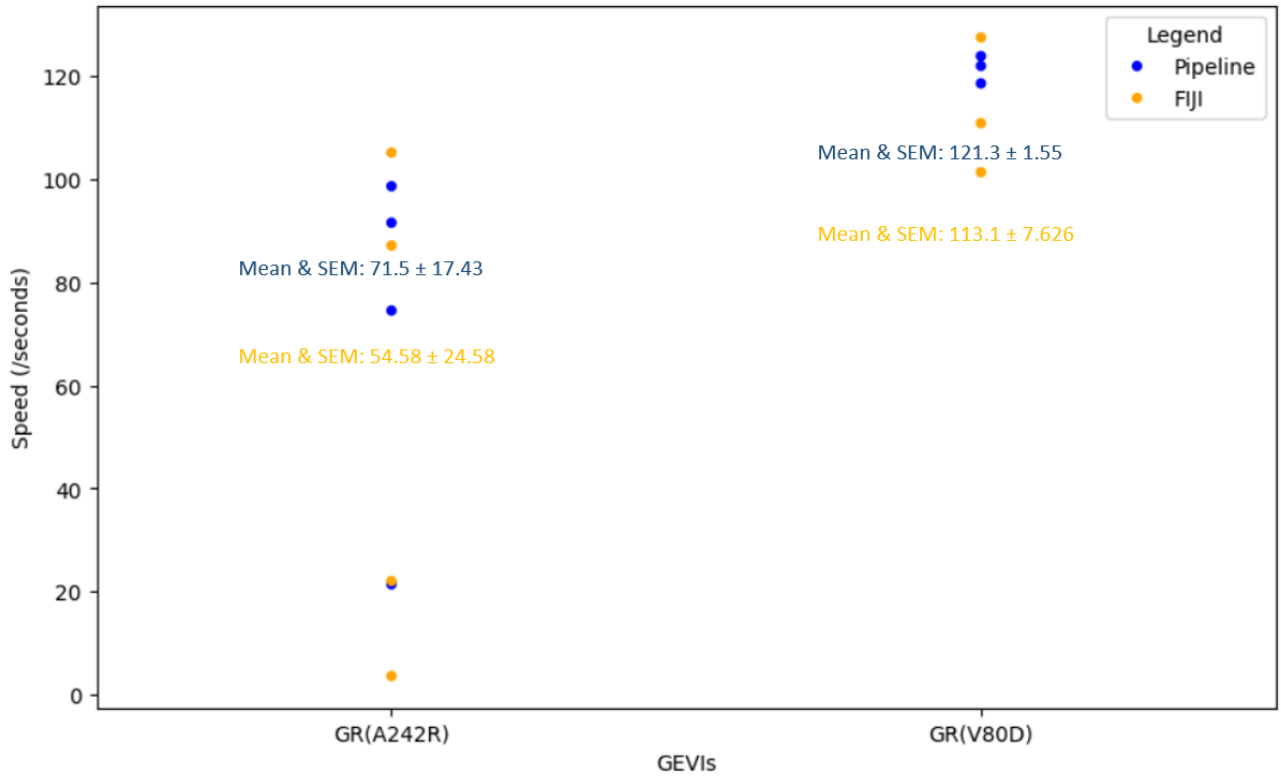



**Figure 4.17: The speed values of GR(A242R) and GR(V80D) GEVIs obtained from the predicted binary masks of the image analysis pipeline compared to the speed values obtained by drawing ROIs in FIJI. The values in blue are the values obtained from the image analysis pipeline and the values in orange are the values obtained from the drawn ROIs in FIJI. SEM in the figure stands for standard error of the mean**

Similar comparisons are made for speeds of the different GEVIs obtained from the predicted binary masks of the image analysis pipeline and the ROIs drawn in FIJI. QuasAr6a is a very fast GEVI, therefore, speed of QuasAr6a is not estimated. GR(A242R) and GR(V80D) are slow GEVIs and speed of these GEVIs is estimated.

Figure 4.17 shows the speed values of GR(A242R) and GR(V80D) GEVIs. These values are obtained from the predicted binary masks of the image analysis pipeline and by drawing ROIs in FIJI.

Figure 4.18 shows the speed values of all of the 7 different single cell spiking HEK cell movies expressing the GEVIs GR(V80D) and GR(A242R). These values are obtained from the predicted binary masks of the image analysis pipeline and by drawing ROIs in FIJI.

Table 4.3 depicts the percentage error in averages of speeds of GR(A242R) and GR(V80D) GEVIs. This percentage error is calculated as given in equation 4.1.2c.

| GEVIs | Percentage Error (%) |
|---|---|
| GR(A242R) | 31 |
| GR(V80D) | 7.25 |

**Table 4.3: Percentage error in averages of speeds of GR(A242R) and GR(V80D) GEVIs**

The percentage error in the averages of speeds of all of the 7 different single cell spiking HEK cell movies expressing the GEVIs GR(A242R) and GR(V80D) is 16.639%. The speed values of the GEVIs are over-estimated by the built image analysis pipeline.

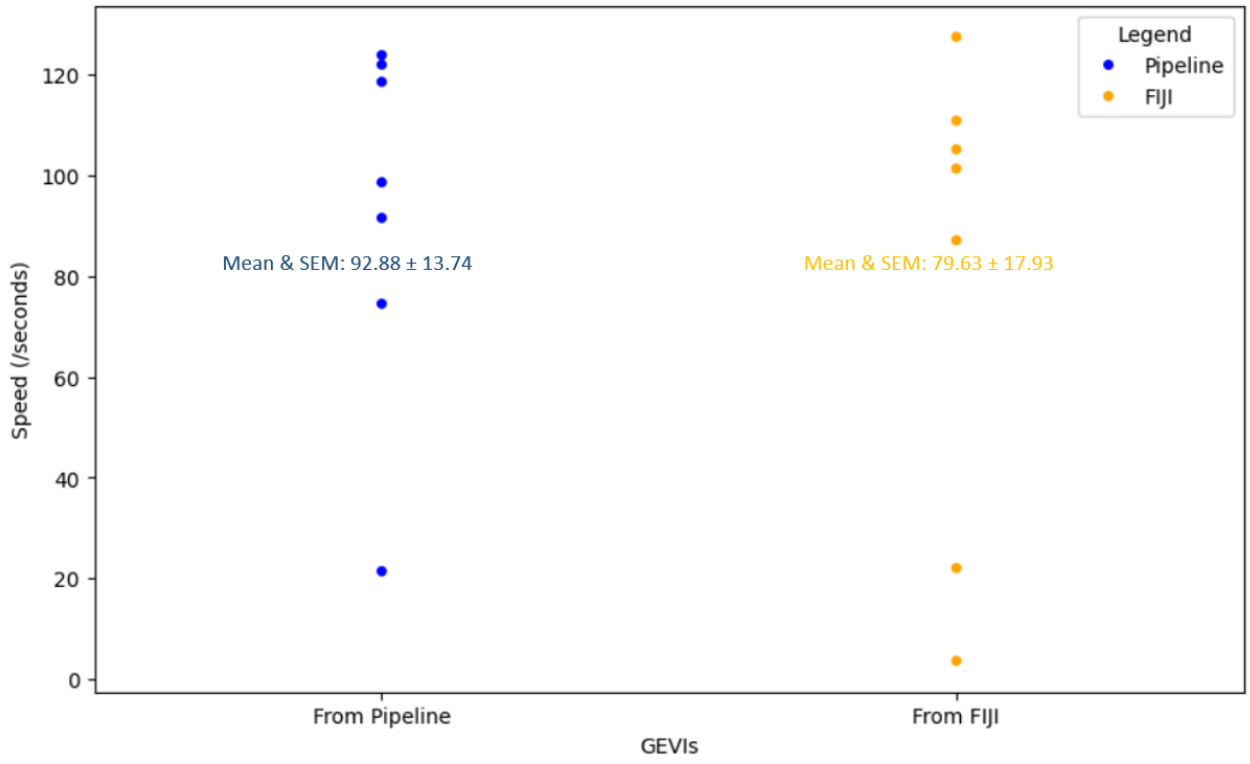



**Figure 4.18: The speed values of 7 different GEVIs obtained from the predicted binary masks of the image analysis pipeline compared to the speed values obtained from drawn ROIs in FIJI. The values in blue are the values obtained from the image analysis pipeline and the values in orange are the values obtained from the drawn ROIs in FIJI. SEM in the figure stands for standard error of the mean**

Additionally, out of the 14 different single cell spiking HEK cell movies expressing the GEVIs GR(A242R), GR(V80D) and, QuasAr6a, 13 movies are correctly identified as slow or very fast GEVIs by the image analysis pipeline. However, one of the QuasAr6a movie is wrongly identified as slow GEVIs. Therefore, the estimated accuracy of the image analysis pipeline when identifying GEVIs as slow or very fast GEVIs is 92.8%. This accuracy is estimated as given in equation 4.1.2d

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{4.1.2d}$$

where TP is true positives, TN is true negatives, FP is false positives and FN is false negatives. Here, TP is 6 as 6 very fast GEVIs are correctly identified as very fast GEVIs by the image analysis pipeline. TN is 7 as 7 slow GEVIs are correctly identified as slow GEVIs by the image analysis pipeline. FP is 0 as no slow GEVI is identified as a very fast GEVI by the image analysis pipeline. FN is 1 as 1 very fast GEVI is wrongly identified as a slow GEVI by the image analysis pipeline.

The estimated sensitivity of the image analysis pipeline when identifying a GEVI as a slow or a very fast GEVI is 0.857. This sensitivity is estimated as given in equation 4.1.2e

$$Sensitivity = TP/(TP + FN) \tag{4.1.2e}$$

where TP is true positives, FN is false negatives. In this case TP is 6 and FN is 1.

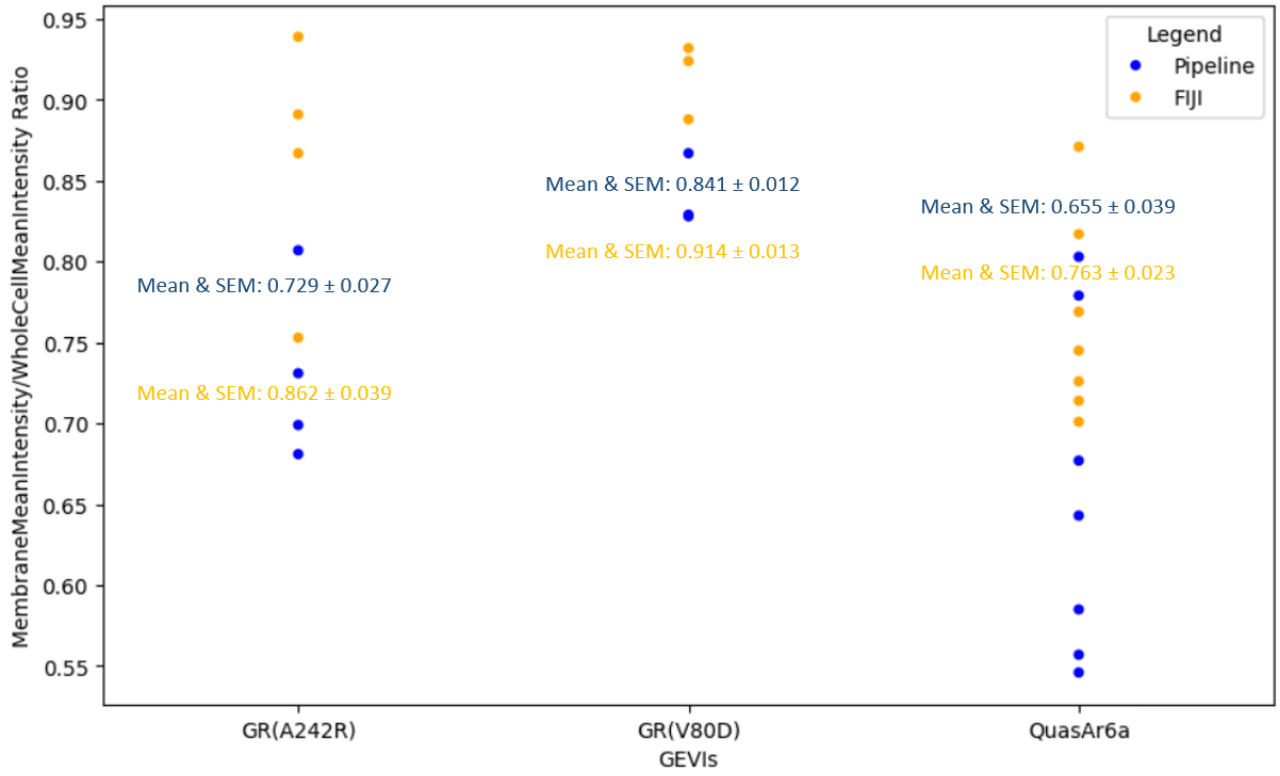



**Figure 4.19: The membrane localization values of different GEVIs obtained from the predicted binary masks of the image analysis pipeline compared to the membrane localization values obtained by drawing ROIs in FIJI. The values in blue are the values obtained from the image analysis pipeline and the values in orange are the values obtained from the drawn ROIs in FIJI. SEM in the figure stands for standard error of the mean**

For membrane localization comparisons, a time averaged 2-D image is obtained using features of FIJI. On this image, ROIs of the membrane and whole cell are drawn in FIJI. Then the mean intensity of these ROIs is calculated in FIJI.

Figure 4.19 shows the membrane localization values of different GEVIs. These values are obtained from the predicted binary masks of the image analysis pipeline and drawn ROIs in FIJI.

Figure 4.20 shows the membrane localization values of all of the 14 different single cell spiking HEK cell movies expressing different GEVIs. These values are obtained from the predicted binary masks of the image analysis pipeline and by drawing ROIs in FIJI.

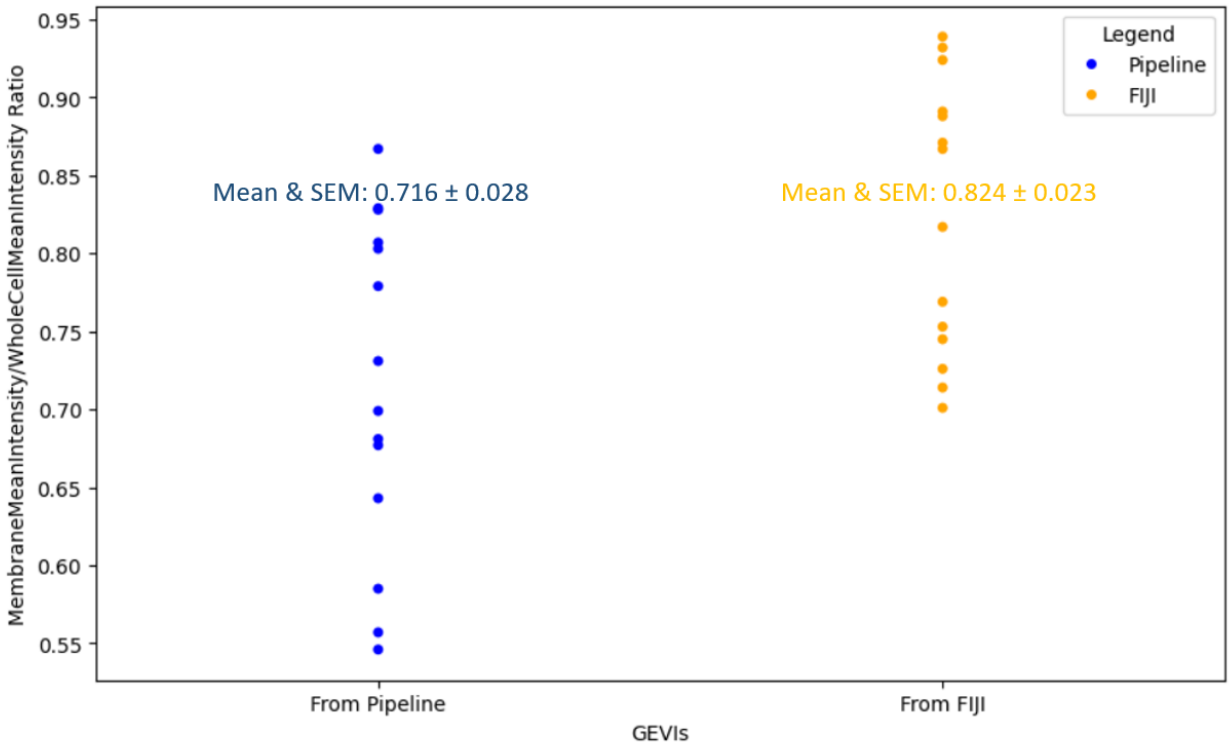



**Figure 4.20: The membrane localization values of 14 different GEVIs obtained from the predicted binary masks of the image analysis pipeline compared to the membrane localization values obtained from the drawn ROIs in FIJI. The values in blue are the values obtained from the image analysis pipeline and the values in orange are the values obtained from the drawn ROIs in FIJI. SEM in the figure stands for standard error of the mean**

| GEVIs | Percentage Error (%) |
|---|---|
| GR(A242R) | 15.429 |
| GR(V80D) | 7.987 |
| QuasAr6a | 14.155 |

**Table 4.4: Percentage error in averages of membrane localization values of different GEVIs**

Table 4.4 depicts the percentage error in averages of membrane localization values of different GEVIs. This percentage error is calculated as given in equation 4.1.2c.

The percentage error in averages of membrane localization values of all of the 14 different single cell spiking HEK cell movies expressing different GEVIs is 13.107%. The membrane localization values are under-estimated by the built image analysis pipeline

The next section describes the results of the features of the 3 different GEVIs obtained from the image analysis pipeline. The section also describes the results of screening a population of spiking HEK cells expressing variants of GR(V80D) GEVI.

## 4.2. Results related to GEVIs and screening of population of spiking HEK cells expressing variants of GEVI

The image analysis pipeline identifies GR(V80D) as the GEVI with best sensitivity and membrane localization features in comparison to QuasAr6a and GR(A242R) GEVIs. On the other hand, QuasAr6a is identified as the GEVI with least sensitivity and membrane localization features in comparison to GR(V80D) and GR(A242R) GEVIs. The image analysis pipeline identifies QuasAr6a as a very fast GEVI. Additionally, the image analysis pipeline identifies GR(V80D) GEVI to be faster than GR(A242R) GEVI. These results of features comparison of the 3 different GEVIs obtained from the built image analysis pipeline match with the ground truth results obtained in FIJI.

Figure 4.21 describes how the screening of a population of spiking HEK cells expressing variants of GR(V80D) GEVI is performed with the help of the built image analysis pipeline. Movies from every FOV of the glass bottom dish containing population of spiking HEK cells are recorded on the octoscope. Figure 4.21a shows the first movie frame of one of the movies recorded from a certain FOV of the glass bottom dish. These movies are of size 2000x1000 pixels. For faster computation, these movies are cropped into smaller movies of single spiking HEK cells. Figure 4.21b shows the first movie frame of one of the 4 smaller movies obtained from Fig. 4.21a. This cropping into smaller movies is performed manually in FIJI. A total of 56 smaller movies of single spiking HEK cells expressing variants of GR(V80D) GEVI are obtained from the individual movies recorded from every FOV of the glass bottom dish. Each of the obtained smaller movie is analyzed by the built image analysis pipeline. Figure 4.21c-1 shows the binary mask of the single spiking HEK cell. This binary mask is predicted by the re-trained U-Net model. Figure 4.21c-2 shows the background mask obtained for sensitivity calculations of the GEVIs expressed by the segmented spiking HEK cell. From the different binary masks, the image analysis pipeline extracts the sensitivity and speed values of the variants of GR(V80D) GEVI expressed by the segmented single spiking HEK cell. For this screening, sensitivity and speed features of the variants of the GEVI are of importance. This is because the goal is to find a variant of GEVI with sensitivity and kinetics that can help with the neuroscience applications.

Figure 4.22 shows the results of screening of population of spiking HEK cells expressing variants of GR(V80D) GEVI. The sensitivity and speed values (Fig. 4.22a) of all the variants of GR(V80D) GEVI expressed by the 56 segmented single spiking HEK cells are obtained from the built image analysis pipeline. From these sensitivity and speed values of all the variants of GR(V80D) GEVI, the best performing variants of GR(V80D) GEVI are selected by applying 2 conditions. The first condition is to select the variants with top 5 score values (Fig. 4.22b). This score to the variants is assigned as given in equation 4.1.2f

$$score = (0.3 * speed) + (0.7 * sensitivity) \tag{4.1.2f}$$

The weight-age to the sensitivity and speed features in equation 4.1.2f is given based on their order of importance.

From the selected top 5 variants of GR(V80D) GEVI, the final best performing variants are selected by applying a second condition. This second condition is to select the variants that have speed greater than 10/seconds and sensitivity greater than 20%. In this screening of a population of spiking HEK cells expressing variants of GR(V80D) GEVI, a single best forming variant of GR(V80D) GEVI is identified. This variant has a speed of 131.7/seconds and a high sensitivity of 415.2% (Fig. 4.22c).

Therefore, in this manner the built image analysis pipeline is used for screening of a population of spiking HEK cells expressing variants of GEVI with best sensitivity and kinetics.

The next chapter summarizes this research project by highlighting the pros and cons of the built image analysis pipeline and further improvements that could be made to the built image analysis pipeline.

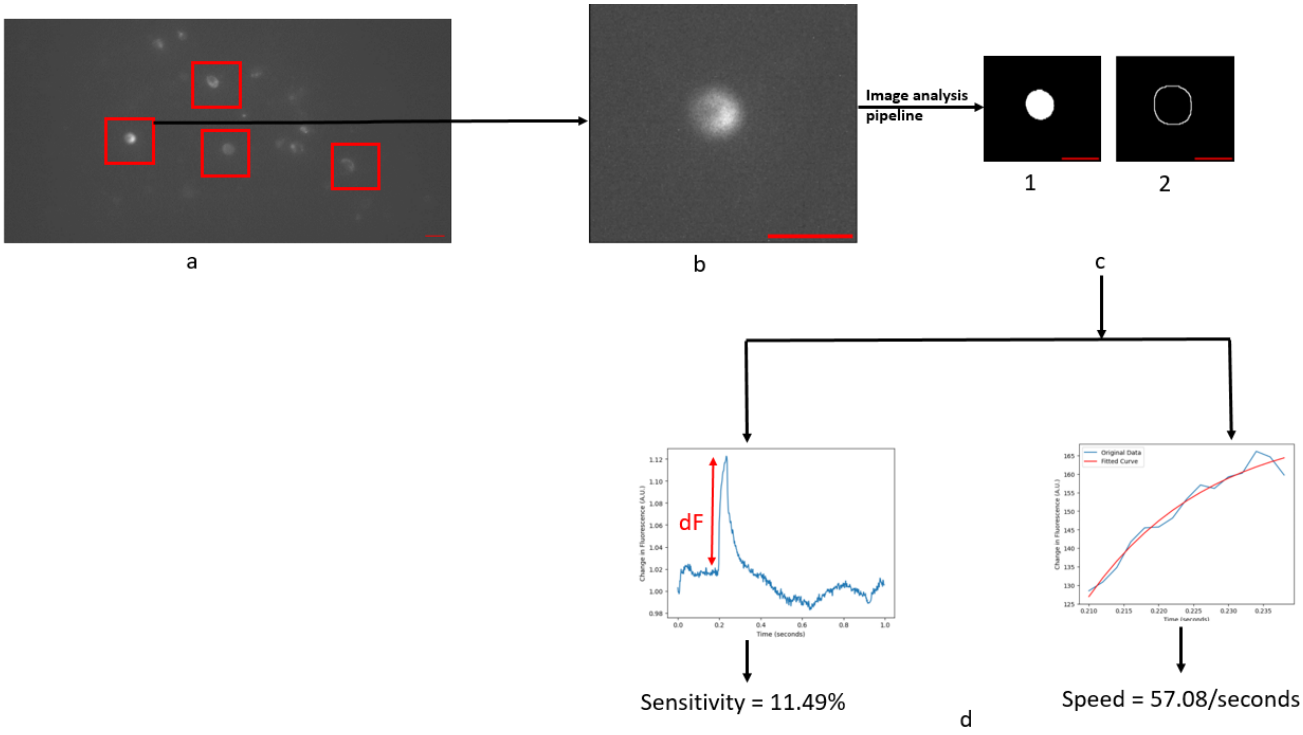



**Figure 4.21: Image analysis of movies of a population of spiking HEK cells expressing variants of GR(V80D) GEVI. a: First movie frame of one of the movies recorded from a certain FOV of the glass bottom dish. Image of size 2000x1000 pixels. b: Smaller movie of single spiking HEK cell. Image of size 272x244 pixels. c: Obtained binary masks from the built image analysis pipeline. Images of sizes 272x244 pixels. d: Obtained sensitivity and speed values from the built image analysis pipeline. Scale bars = 20$\mu$m**

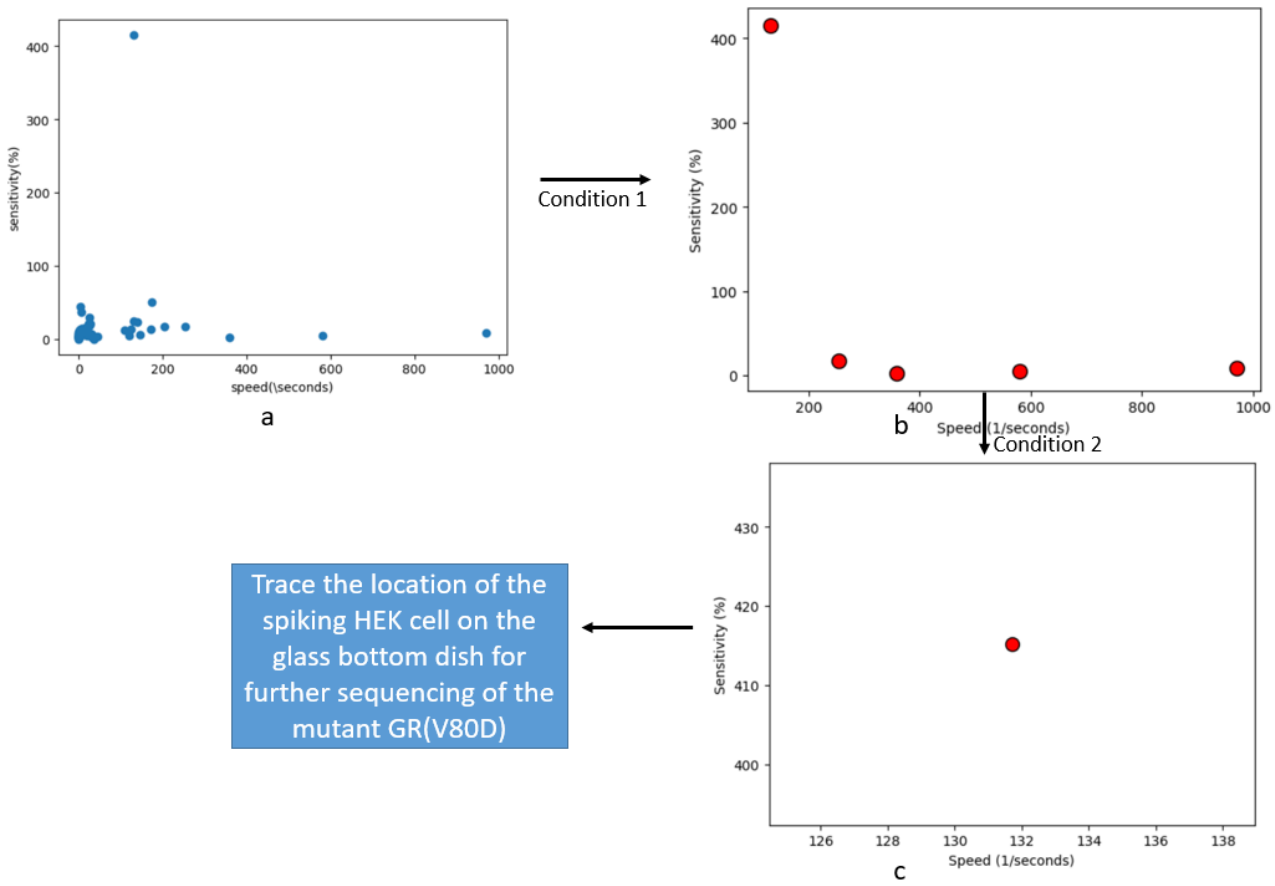



**Figure 4.22: Screening of a population of spiking HEK cells expressing variants of the GR(V80D) GEVI. a: Sensitivity and speed values of all the variants of GR(V80D) GEVI. b: Top 5 best performing variants of GR(V80D) GEVI. c: Final best performing variant of GR(V80D) GEVI**

# 5

# Discussion

*This report describes an image analysis pipeline built for automated analysis of 1P voltage imaging datasets of populations of spiking HEK cells expressing variants of GEVI. The results in this report show that the built image analysis pipeline can aid in quick screening of best performing variants of GEVI among several of its variants in spiking HEK cells. The screening is performed by extracting features such as sensitivity and speed of the variants of GEVI. These features are extracted by the built image analysis pipeline.*

*However, further improvements can be made to the built image analysis pipeline in order to extract accurate values of the features of the GEVIs expressed by the spiking HEK cells. For this, the binary mask prediction by the used segmentation model is the most important step. Firstly, although HEK cells and spiking HEK cells have similar cellular morphology, there are still differences in the signals of the GEVIs expressed by these 2 subsets of cells. The pre-trained U-Net model can be re-trained on spiking HEK cells datasets rather than HEK cells datasets in order to increase the IoU of segmentation of spiking HEK cells. Additionally, accurate annotations of the membrane of the spiking HEK cells is necessary for extraction of features of the GEVIs expressed by these spiking HEK cells. This is because GEVIs are membrane proteins. For accurate annotations, more than one annotator can manually annotate the training datasets. The pre-trained U-Net model can be re-trained on these multiple annotated datasets of spiking HEK cells and the results of the predicted binary masks can be compared with their IoU values. The re-trained U-Net model with the highest IoU can then be used for analysis of movies of spiking HEK cells expressing GEVIs. If the predicted binary mask is accurate enough then the step in the built image analysis pipeline where the predicted binary mask is 4 times dilated can be removed. The predicted binary mask and the 5 times dilated binary mask can be used for obtaining the background mask for sensitivity calculations of the GEVIs. Furthermore, datasets of clusters of spiking HEK cells are not considered in this project. The performance of the re-trained U-Net model on clusters of spiking HEK cells needs to be evaluated in order to confirm if U-Net can accurately distinguish between 2 different spiking HEK cells. If the performance of U-Net decreases when analysing clusters of cells, other neural networks based segmentation models such as MaskRCNN can used for segmentation of spiking HEK cells.*

*Secondly, for calculations of the speed of the GEVIs, mono-exponential fits is not explored due to time constraints. If mono-exponential fits can be optimized better by setting robust parameter values then the speed of the GEVIs can be calculated from one time constant only. This simplifies speed calculations of the GEVIs.*

*Thirdly, the membrane localization feature is of least importance in this project. Therefore, instead of using a bilateral filter for background subtraction in the time averaged images, the median filter can be used. This would decrease the computational time of the built image analysis pipeline. The bilateral filter takes 2 seconds for a single time averaged spiking HEK cell image on the HP ENVY x360 Convertible 13-bd0xxx system. On the other hand, the median filter takes less than a second for a single time averaged spiking HEK cell image on the HP ENVY x360 Convertible 13-bd0xxx system.*

*Fourthly, to make the screenings completely automated, the built image analysis pipeline needs to be set-up on the desktop of the octoscope. The movies recorded from a certain FOV by the octoscope can either be directly analyzed by the image analysis pipeline or the movies can be automatically cropped to smaller movies of single spiking HEK cells. These smaller movies can then be analyzed by the built image analysis pipeline.*

*Finally, computational speed of the image analysis pipeline is an important parameter in performing screenings of GEVIs in spiking HEK cells. Currently, the built image analysis pipeline takes less than 20 seconds to extract sensitivity and speed values from a single movie of a spiking HEK cell. This single movie is manually cropped from one of the movies recorded from certain FOV of the glass bottom dish. In order to increase the computational speed of the built image analysis pipeline even more, the motion correction algorithm can be removed from the image analysis pipeline. This is because there isn't significant motion seen in the 1P voltage imaging datasets of spiking HEK cells expressing GEVIs.*

*The current built image analysis pipeline discriminates spiking HEK cells expressing GEVIs with high sensitivity and speed values from spiking HEK cells expressing GEVIs with low sensitivity and speed values. With the improvements suggested in this section, the built image analysis pipeline can fully aid in automated and fast screening of populations of spiking HEK cells expressing variants of GEVI with good sensitivity and speed values. Thereby, the built image analysis pipeline can help in engineering improved GEVIs for neuroscience applications.*

# 6

# Acknowledgement

I would like to express my gratitude towards my main supervisor, Daan Brinks, for all the advice related to image analysis and creating an image analyst out of me. I had approached Daan in the beginning of my master's and asked him for a project related to image analysis. A year later, Daan not only provided me with a topic of my interests but also tuned it to the master's level graduation project. I am extremely grateful to Daan for making this happen.

I would like to express my gratitude and appreciation towards my daily supervisor, Marco Post, for all the lectures related to protein engineering and cellular biology. Without the daily crash courses, I would not be able to extract the information required for this research project. The daily supervision and guidance from Marco not only helped me develop as a scientist and a researcher but also made me appreciate science for its beauty and pitfalls.

I would like to thank Rowan, Bram, Lise, Loek, Boyd, Ian and all the members of the BRINKS lab as well as the MInT group, for all the laughter and discussions related to science. The daily talks in the student room of ImPhys, TU Delft, are a memory that I deeply treasure. I am also grateful to all the cakes that I had with everyone. The cakes were my pre-workouts and helped me lift a lot more at the gym helping me deal with frustration when my code wasn't working.

I would like to thank my friends Nicky, Jeanine, Qihao, Josephine and, Melissa for listening to my everyday rants about the project. I am beyond blessed and grateful to have shared my master's journey with them.

I would like to thank my medical physics classmates and my power-lifting club members for making my master's a memorable journey.

Lastly, I would like to thank my family who supported me not only for my graduation project but for the entire duration of my master's despite being miles away. Without them cheering me on everyday, I would not be where I am today both in my life and career.

# A

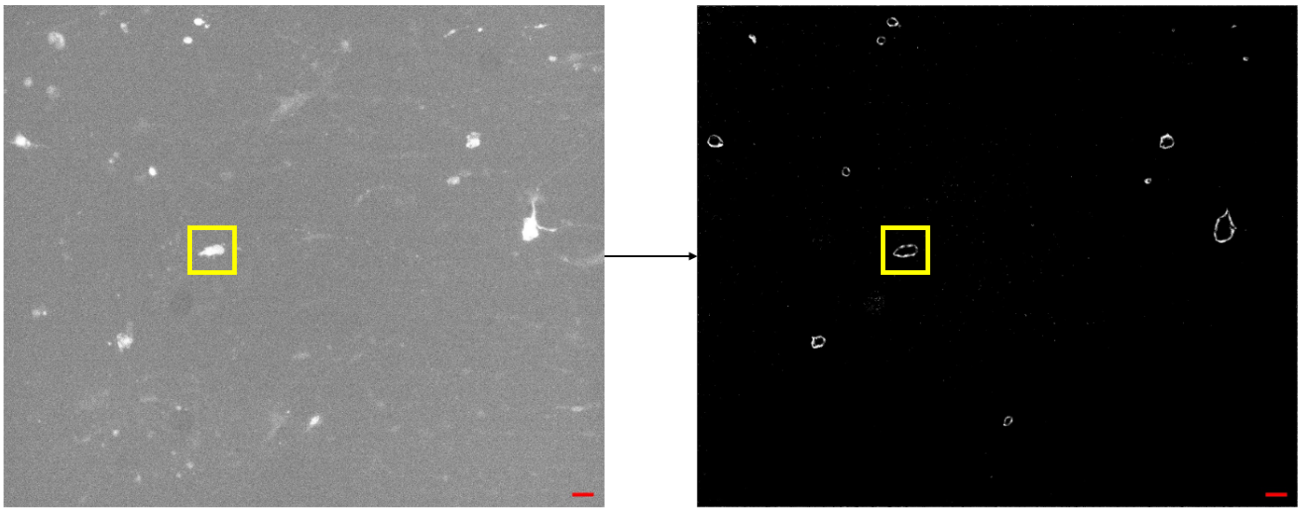
# Supplementary Information

In order to check how the pre-trained U-Net model performs in segmenting other cell types other than the HEK cells and spiking HEK cells, a mini-project is carried out. This project is a small part of another bigger project that uses NPC neurons expressing GEVIs for a different scientific study. The pre-trained U-Net model is re-trained on 35 RGB NPC neuron images. These NPC neuron images are RGB images acquired on the inverted fluorescence microscope (Mshot MF53-N). All the 35 RGB NPC neuron images are of size 2448x2048 pixels and have an image pixel size of 345nm. It roughly took greater than 24 hours to re-train 5 epochs on Intel(R) Xeon(R) CPU E5-1650 v4 with 256 GB of RAM memory. The ground truth masks of these 35 RGB NPC neuron images are manually annotated. The ground truth masks contain outlines of neuronal cell bodies and neuronal dendrites. Two different pre-trained U-Net models are re-trained. The hyperparameters chosen to re-train these U-Net models are same as the ones described in section 3.2. One of the re-trained U-Net model segments outlines of neuronal cell body structures whereas the other re-trained U-Net model segments outlines of neuronal dendrite structures.

Figure A.1 shows the results of the re-trained U-Net model to segment outlines of neuronal cell body structures. The re-trained U-Net model is able to segment a single neuronal cell body with an IoU of 0.807. The regions depicted in yellow in fig. A.1a and b are the regions between which the IoU is calculated.

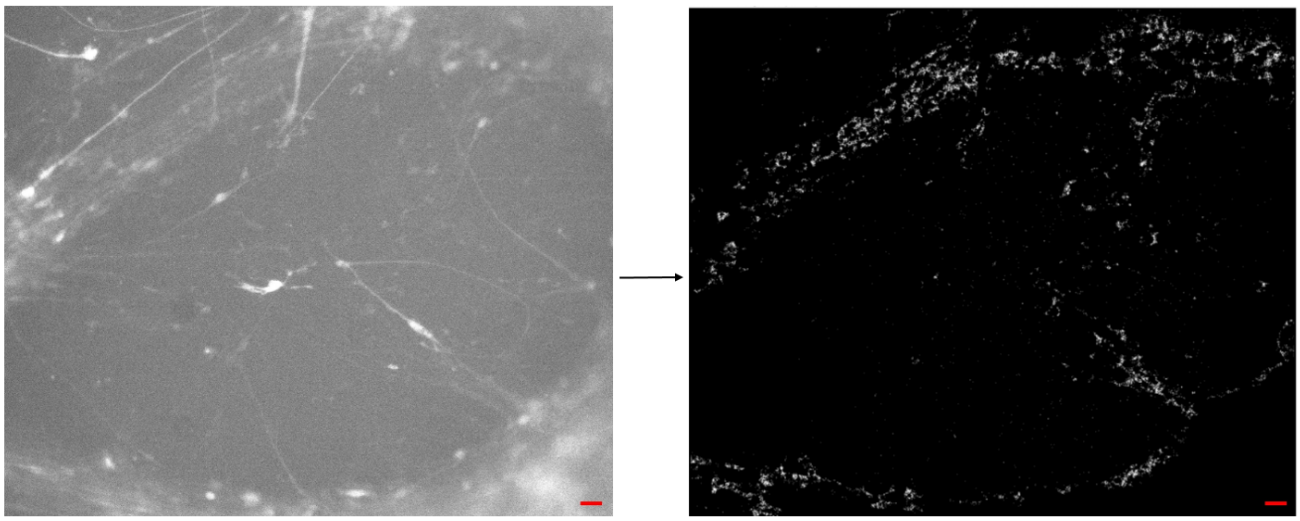
Figure A.2 shows the results of the re-trained U-Net model to segment outlines of neuronal dendrite structures. The re-trained U-Net model fails to segment neuronal dendrites. Rather than segmenting the fine dendritic structures, the model segments clusters of objects as shown in fig.A.2.

Additionally, in order to check the cellular morphology of the single spiking HEK cells that are segmented with greater than 0.7 IoU by the re-trained U-Net model to segment single spiking HEK cells, total bending energy of the single HEK cells is calculated. Total bending energy provides an idea about the contour of the cells. It is the energy required to deform a circular geometry to a particular geometry [36]. Total bending energy is a good feature in understanding the cellular morphology. Figure A.3 shows the plot of total bending energy of the HEK cells vs their IoU. The total bending energy and the corresponding IoU values of different single HEK cells shown in the plot belong to the 28 single HEK cell images taken as the testing dataset (section 3.3) and one additional image of a single HEK cell with comparatively higher bending energy than the rest of the single HEK cells. The IoU is calculated between the ground truth manually annotated mask and the predicted binary mask of the re-trained U-Net model to segment single spiking HEK cells. Single HEK cells with a total bending energy in the range 1.5 to 2.25 have IoU in the range 0.85 to 0.95. These single HEK cells have elliptical like contours. The single HEK cells with a bending energy lower than 0.5 and a bending energy greater than 3 are segmented with an IoU lesser than 0.6 by the re-trained U-Net model trained to segment single spiking HEK cells.



a b

**Figure A.1: Predicted binary mask of re-trained U-Net model to segment neuronal cell bodies**. **a:** Input image to the re-trained U-Net model. Image of size 2448x2048 pixels. **b:** Predicted binary mask of the re-trained U-Net model. Image of size 2448x2048 pixels. The IoU between the bounding boxes represented in yellow is 0.807. Scale bars = 30$\mu$m



a b

**Figure A.2: Predicted binary mask of re-trained U-Net model to segment neuronal dendrites**. **a:** Input image to the re-trained U-Net model. Image of size 2448x2048 pixels.**b:** Predicted binary mask of the re-trained U-Net model. Image of size 2448x2048 pixels. Scale bars = 30$\mu$m

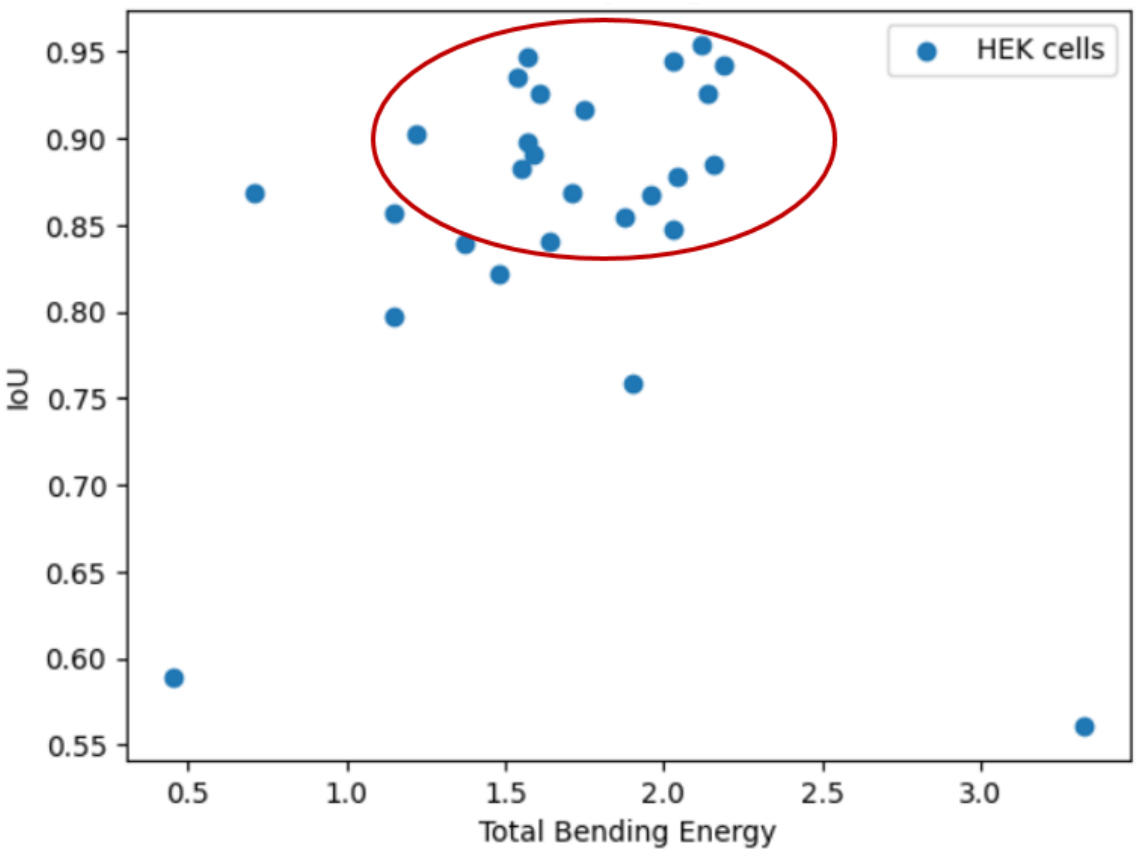



**Figure A.3: Total bending energy VS IoU of single HEK cells. The region circled in red shows the total bending energy of single HEK cells that are segmented with an IoU greater than 0.8 by the re-trained U-Net model**

Fig. A.4a is the RGB mean + correlation image of a single spiking HEK cell. The movie of this single spiking HEK cell is cropped from one of the recorded movies of certain FOV of a population of spiking HEK cells expressing variants of GR(V80D) GEVI. Figure A.4 shows the IoU results computed between the manually annotated mask (Fig. A.4b) of the single spiking HEK cell (Fig. A.4a) and predicted mask (Fig. A.4c) of the single spiking HEK cell (Fig. A.4a). Fig. A.4b is the binary mask manually annotated on Fig. A.4a. Fig. A.4c is the binary mask predicted by the re-trained U-Net model trained to segment spiking HEK cells. The average IoU between the manually annotated masks of 10 single spiking HEK cells and predicted masks of 10 single spiking HEK cells is 0.807 with a standard error of mean of 0.028. These 10 single spiking HEK cell movies are cropped from 10 different movies of population of spiking HEK cells expressing variants of GR(V80D) GEVI recorded from different FOVs on the octoscope.

Apart from the above described mini-projects related to U-Net segmentation, another mini-project is carried out related to patch clamp recordings of HEK cells. The mini-project is to compare the computational speed of best curve fit method and low pass filtering method for photobleaching signal corrections in patch clamp movies of HEK cells. Both best curve fitting and low pass filtering of the original signal help in removal of the photobleached signal in patch clamp movies of HEK cells. The low pass filtering of the original signal is performed using the moving average filter. However, for 12 different patch clamp movies of HEK cells, curve fitting took on an average 0.0034 seconds on the HP ENVY x360 Convertible 13-bd0xxx whereas low pass filtering the signal took on an average 0.00031 seconds on the HP ENVY x360 Convertible 13-bd0xxx. Therefore, low pass filtering the original signal using moving average filter for photobleaching signal removal in patch clamp movies of HEK cells is computationally faster.

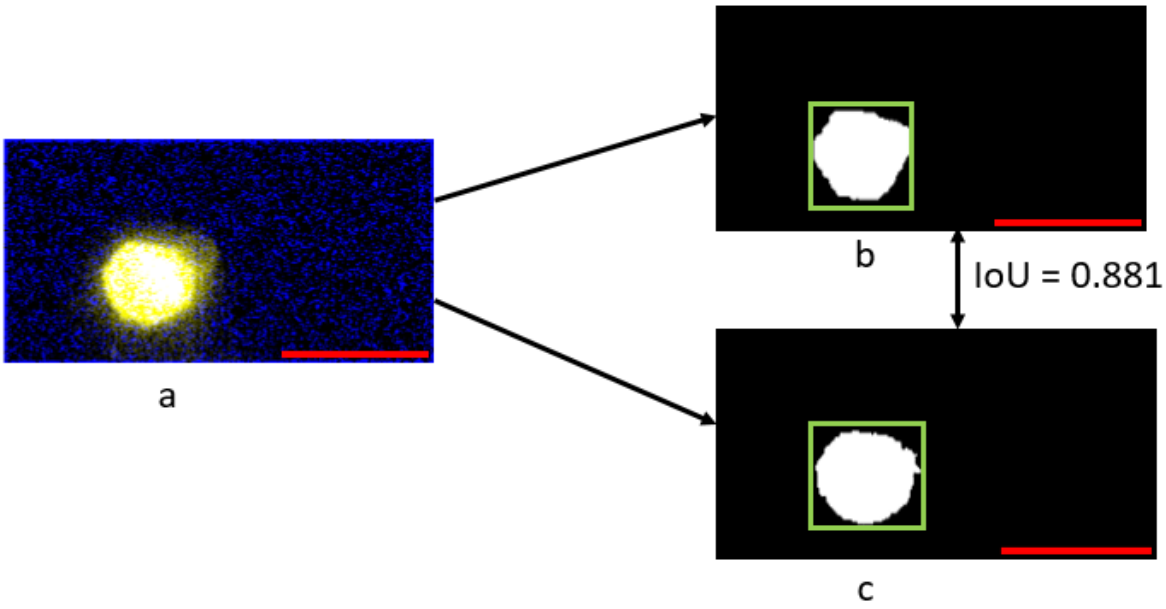



**Figure A.4: IoU between manually annotated mask and predicted mask of single spiking HEK cell. a: RGB mean + correlation image of single spiking HEK cell. Image of size 252x132 pixels. b: Manually annotated ground truth binary mask. Image of size 252x132 pixels. c: Binary mask predicted by the re-trained U-Net model. Image of size 252x132 pixels. Scale bars = 20$\mu$m**

# Bibliography


[1] Shigenori Inagaki Takeharu Nagai. Current progress in genetically encoded voltage indicators for neural activity recording. *Current Opinion in Chemical Biology*, 33:95–100, 2016. doi: https://doi.org/10.1016/j.cbpa.2016.05.023.

[2] Yoav Adam. All-optical electrophysiology in behaving animals. *Journal of Neuroscience Methods*, 353, 2021. doi: https://doi.org/10.1016/j.jneumeth.2021.109101.

[3] Silapetere A. Hwang S. Hontani Y., et al. Quasar odyssey: the origin of fluorescence and its voltage sensitivity in microbial rhodopsins. *Nat Commun*, 13, 2022. doi: https://doi.org/10.1038/s41467-022-33084-4.

[4] Knöpfel T. Song C. Optical voltage imaging in neurons:moving from technology development to practical tool. *Nat. Rev. Neurosci.*, 20:719–727, 2019. doi: 10.1038/s41583-019-0231-4.

[5] Yongxian Xu Peng Zou Adam E Cohen. Voltage imaging with genetically encoded indicators. *Current Opinion in Chemical Biology*, 39:1–10, 2017. doi: https://doi.org/10.1016/j.cbpa.2017.04.005.

[6] Park J Werley CA Venkatachalam V, et al. Screening fluorescent voltage indicators with spontaneously spiking hek cells. *PLoS One*, 8, 2013. doi: https://doi.org/10.1371/journal.pone.0085221.

[7] Zhang H Reichert E Cohen AE. Optical electrophysiology for probing function and pharmacology of voltage-gated ion channels. *Elife*, 5, 2016. doi: https://doi.org/10.7554/eLife.15202.

[8] McNamara HM Zhang H Werley CA Cohen AE. Optically controlled oscillators in an engineered bioelectric tissue. *Phys. Rev. X*, 6, 2016. doi: https://link.aps.org/doi/10.1103/PhysRevX.6.031001.

[9] Richard H. Roth Jun B. Ding. From neurons to cognition: Technologies for precise recording of neural activity underlying behavior. *BME Front.*, 2020. doi: 10.34133/2020/7190517.

[10] Hadas Bensity Alexander Song Gal Mishne Adam S. Charles. Review of data processing of functional optical microscopy for neuroscience. *Neurophotonics*, 9, 2022. doi: https://doi.org/10.1117/1.NPh.9.4.041402.

[11] Lutz S. Beyond directed evolution—semi-rational protein engineering and design. *Current Opinion in Biotechnology*, 21:734–743, 2010. doi: 10.1016/j.copbio.2010.08.011.

[12] Xin Meng et al. A compact microscope for voltage imaging. *Journal of Optics*, 2022. doi: 10.1088/2040-8986/ac5dd5.

[13] Eftychios A. Pnevmatikakis Andrea Giovannucci. Normcorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. *Journal of Neuroscience Methods*, 291:83–94, 2017. doi: https://doi.org/10.1016/j.jneumeth.2017.07.031.

[14] Szeliski. Computer vision: Algorithms and applications. *Chapter 4*. doi: http://mesh.brown.edu/engn1610/szeliski/04-featuredetectionandmatching.pdf.

[15] Förstner. Scriptum photogrammetrie 1. *Chapter "Matching / Kreuzkorrelation"*. doi: https://www.ipb.uni-bonn.de/html/teaching/photo12-2021/2021-pho1-09-matching-cc.pptx.pdf.

[16] Andrea GiovannucciJohannes FriedrichPat GunnJérémie KalfonBrandon L BrownSue Ann KoayJiannis TaxidisFarzaneh NajafiJeffrey L GauthierPengcheng ZhouBaljit S KhakhDavid W TankDmitri B ChklovskiiEftychios A Pnevmatikakis. Caiman an open source tool for scalable calcium imaging data analysis. *eLife*, 8, 2019. doi: https://doi.org/10.7554/eLife.38173.

[17] Changjia Cai Johannes Friedrich Amrita Singh et al. Volpy: Automated and scalable analysis pipelines for voltage imaging datasets. *PLoS Comput Biol*, 17(4), 2021. doi: https://doi.org/10.1371/journal.pcbi.1008806.

[18] Smith SL Häusser M. Parallel processing of visual space by neighboring neurons in mouse visual cortex. *Nature Neuroscience*, 13(9):1144–1149, 2010. doi: https://doi.org/10.1038/nn.2620.

[19] Lin TY et al. Microsoft coco: Common objects in context. *In: European conference on computer vision. Springer*, page 740–755, 2014. doi: https://doi.org/10.1007/978-3-319-10602-1_48.

[20] Ronneberger O. et al. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention*, 9351, 2015. doi: https://doi.org/10.1007/978-3-319-24574-4_28.

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Curran Associates, Inc., 2019. doi: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[22] PyTorch. (n.d.). BCEWithLogitsLoss — PyTorch 1.8.1 documentation. https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html.

[23] Geoffrey Hinton with Nitsh Srivastava and Kevin Swersky. Neural networks for machine learning, lecture 6a, overview of mini-batch gradient descent. doi: https://medium.com/@fernando.dijkinga/the-rmsprop-optimizer-78f02efb63e9.

[24] Jean Serra. Image analysis and mathematical morphology. 1982. doi: ISBN0-12-637240-3.

[25] Huang et al. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27:13–18, 1979. doi: 10.1109/TASSP.1979.1163188.

[26] Rob J Hyndman. Moving averages. 2009. doi: https://robjhyndman.com/papers/movingaverage.pdf.

[27] A. Savitzky and M.J.E Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–39, 1964. doi: 10.1021/ac60214a047.

[28] Quarteroni A. et al. Numerical mathematics (texts in applied mathematics). *New York: Springer*, 2007. doi: https://doi.org/10.1007/b98885.

[29] Piatkevich K.D. Jung E.E. Straub, C. et al. A robotic multidimensional directed evolution approach applied to fluorescent voltage reporters. *Nat Chem Biol*, 14:352–360, 2018. doi: https://doi.org/10.1038/s41589-018-0004-9.

[30] Collins T. J. Imagej for microscopy. *BioTechniques*, 43, 2007. doi: 10.2144/000112517.

[31] R Tomasi, C; Manduchi. Bilateral filtering for gray and color images. *Sixth International Conference on Computer Vision. Bombay*, page 839–846, 1998. doi: 10.1109/ICCV.1998.710815.

[32] Huang et al. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27:13–18, 1979. doi: 10.1109/TASSP.1979.1163188.

[33] Correlation coefficient: Simple definition, formula, easy steps. *Statistics How To*. doi: https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/.

[34] Rezatofighi et al. Generalized intersection over union. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. doi: 10.1109/CVPR.2019.00075.

[35] Stringer C. Wang T. Michaelos, M. et al. Cellpose: a generalist algorithm for cellular segmentation. *Nat Methods*, 18:100–106, 2021. doi: https://doi.org/10.1038/s41592-020-01018-x.

[36] Young I. et al. An analysis technique for biological shape. i. *Information and Control*, 1974. doi: 10.1016/S0019-9958(74)91038-9.