During my internship as a Data Science Intern in the Yao-Ying Ma Laboratory at the Indiana University School of Medicine's Department of Pharmacology and Toxicology, I gained hands-on experience with the MiniAn Python miniscope analysis pipeline. My primary responsibility involved the processing of long videos captured using a miniscope, recording neural activity in freely moving mice *in vivo* by monitoring the fluorescence intensity changes of calcium-sensitive GCaMP8f fluorescent protein selectively expressed in excitatory neurons, to obtain accurate active neuron counts over time. Working with the data required substantial computational power and could take many hours even on high-performance systems. To manage these workloads efficiently, I was given access to the Indiana University Quartz supercomputer, which was capable of handling large-scale parallel processing.

Using the MiniAn pipeline, I first performed the preprocessing of video data by computing a minimum projection across all frames and subtracting it from each frame to normalize central illumination, which usually appears brighter due to video image vignetting. I then reduced noise introduced during central glow correction using a median filter, and in rare cases, Gaussian or anisotropic filters for comparison. Finally, I removed out-of-focus background regions that distorted cell detection by applying a morphological tophat transformation followed by dilation, which isolated only the in-focus neuronal structures.

After preprocessing, I performed motion correction to compensate for possible minor movements of the miniscope that may occur during recordings in freely moving mice. By calculating phase correlations between frames, I aligned the data to a consistent field of view, filling any missing areas with zeros to preserve structure. Next, I performed seed initialization to map potential neuron locations. To do so, I identified maximum-intensity projections (brighter regions), which are often referred to as local maxima or seeds. To ensure completeness, I used a function of overcomplete seed generation, with the window size being set to the largest expected neuron diameter. These identified seeds were refined using highpass and low-pass filters to separate the signal from noise and further validated using the Ks refinement algorithm based on brightness gradient assessment. Close and overlapping seeds were merged using a subroutine to identify the neuron positions. Finally, I constructed the spatial and temporal matrices from seed locations representing the identified neurons for further analysis.

The final stage involved applying Constrained Non-negative Matrix Factorization (CNMF) to extract neural signals from the processed data. In this model, the recorded fluorescence data (Y) were expressed as a combination of spatial components (A, neuron footprints), temporal components (C, activity traces), and background terms (b and f), with noise (E). Using an iterative optimization approach, I alternated between refining A while keeping C fixed, and refining C while keeping A fixed, to minimize reconstruction error and suppress noise. Before these updates, I estimated the spatial noise using Fast Fourier Transform (FFT) analysis to improve fidelity. Through successive spatial and temporal refinements, I separated overlapping neurons, enhanced spatial accuracy, and cleaned temporal traces of artifacts. This process produced high-quality neural activity maps and precise neuron counts.

Using this workflow, I have analyzed over 50 hours of session videos during my tenure as a Data Science intern. I have learned to collaborate effectively in an academic environment and have consistently provided my team with accurate neural activity maps and neuron counts.