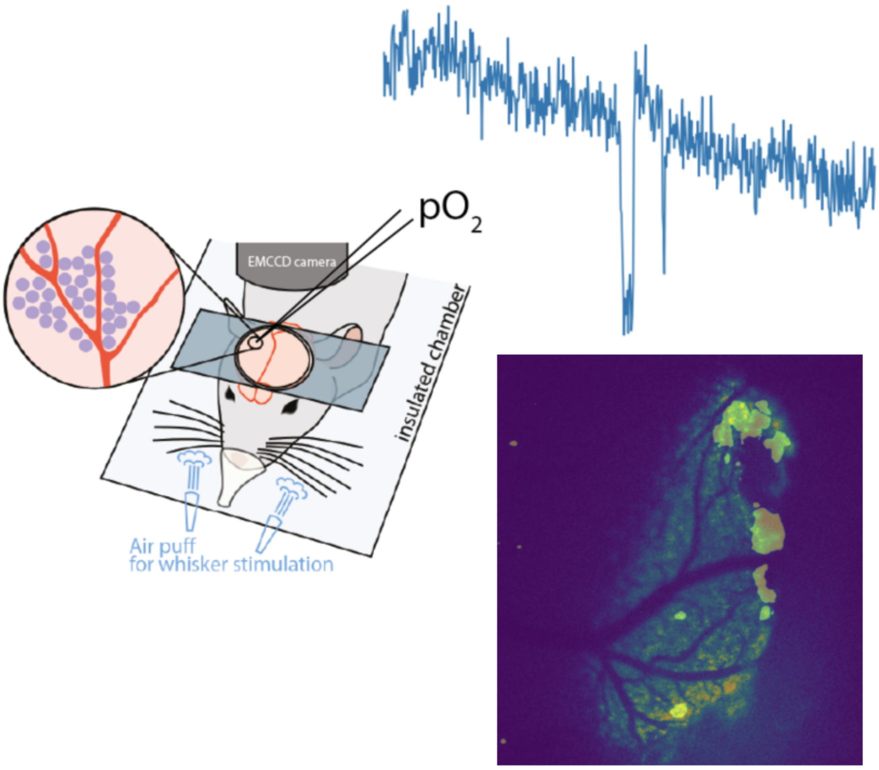
Fachhochschule Nordwestschweiz
Hochschule für Technik

Neuroscience Image Signal Segmentation and Extraction

Bachelor Thesis

Windisch, August 19, 2022



Students	Vincenzo Timmel Lukas Gehrig
Supervisor	Prof. Dr. Michael Graber
Clients	Dr. Felix Ralf Michael Beinlich Dr. Hajime Hirase Dr. Antonios Asiminas
Project Nr.	22FS_I4DS18

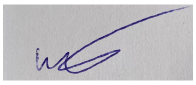
FHNW - University of Applied Sciences Northwestern Switzerland, School of Engineering

Declaration of Authenticity

We hereby declare that we have written this bachelor thesis independently, using only the sources indicated. The literal or taken literally or in terms of content from the sources listed are marked in the work as a quotation or paraphrase. This thesis has not yet been published. It has therefore neither been made available to other interested parties nor submitted to any other examination authority.

Windisch, August 19, 2022

Name: Vincenzo Timmel

Signature: 

Name: Lukas Gehrig

Signature: L. Gehrig

Acknowledgements

We thank Prof. Dr. Michael Graber for his valuable input and always putting us back on the right track.
We thank Dr. Felix Ralf Michael Beinlich, Dr. Hajime Hirase and Dr. Antonios Asiminas for giving us the opportunity for working in such an interesting domain and being available whenever needed.

We hope that our work will support the research of the team in Copenhagen.

Abstract

The brain constantly needs oxygen in order to be able to function consistently. This leads to the question, are all regions of the brain continually supplied with oxygen? Are there regions that are temporarily affected by oxygen deficiency? Neuroscientists at the University of Copenhagen have asked themselves these questions and have determined that such spatially and temporally limited oxygen dips exist on the basis of video recordings of the brains of mice, where oxygen is made visible by a chemical reaction which produces light when in contact with O^2. However, in order to study this phenomenon in more detail, these regions with the signal dips must first be identified. In this context, this thesis plays a central role which should help to find exactly these regions of interest (ROIs).

Currently, these signal dips are a relatively unstudied area and therefore not much is known about their characteristics in terms of signal dip intensity, duration, and general shape in space and time. This raises the question of how we can nonetheless identify these ROIs. While many sophisticated methods exist to detect and analyse neurons firing in the domain of calcium imaging, those do not deal with negative values of a signal and sometimes expect the signals to follow a specific pattern or do not deal with spatially overlapping ROIs. In addition, some of the algorithms take some assumptions which our data does not fulfil, e.g. that the spikes or dips follow an autoregressive model. Since the scientific community has not yet dealt with this type of problem as far as we know, it is up to us to identify suitable methods from related fields and adapt them to our problem.

In this thesis, we present a two-step process, consisting of signal processing to extract the dips and a matrix factorization approach for ROI segmentation. We show that our framework is capable of identifying and segmenting signal dips of different significance level regarding signal intensity and duration. For evaluation, we compare the predicted ROIs of selected videos with the ground truth provided by the University of Copenhagen quantitatively using various metrics. This is followed by a qualitative analysis of the predicted ROIs, where we comment on the predictions and compare them with each other and the ground truth.

Contents

Declaration of Authenticity **ii**

Abstract **iv**

1 Introduction **1**

2 Related Work **3**

3 Methods **4**

3.1 Data Set 4

3.2 ROI Evaluation 5

3.3 Signal Filtering 7

3.4 Detrending 9

3.5 Signal Amplification 13

3.6 Matrix Factorization 15

4 Data Processing **20**

4.1 ROI Analysis 20

4.2 Signal Smoothing 22

4.3 Detrending 25

4.4 Signal Amplification 32

5 Semi-Nonnegative Matrix Factorization **38**

5.1 Selecting Alpha 38

5.2 Overlapping ROIs 42

5.3 Multiple ROIs in a Single Prediction 42

6 Results **45**

6.1 Quantitative Analysis 45

6.2 Qualitative Analysis 50

7 Conclusion **52**

References **53**

List of Figures **53**

List of Tables **59**

Appendix **60**

1 Introduction

In neuroscience, understanding of physiological and computational processes can be gained from imaging the brain. The use of environment-sensitive indicator dyes allows recording movies which capture physiological processes over time. Recent studies at the center for translational neuromedicine of the University of Copenhagen investigate the use of oxygen sensitive dyes to study the brain's supply dynamics for oxygen. The movies acquired have interesting local oxygen characteristics of the brain which will be analyzed in this project.

Oxygen activity in the brain can be visualized with the bioluminescent protein GeNL. This protein is injected locally into an area of the brain to make areas with oxygen light up. Researchers at the University of Copenhagen have noticed that local dips occur over a period of time, signifying decreased local oxygen activity. However, these dips have not yet been studied and thus are not clearly characterized in terms of intensity, spatial extent, temporal extent and shape.

The goal of the thesis is to develop and evaluate a framework for automatically segmenting ROIs and extracting time courses from ROIs of relatively stable video. Furthermore, it is the goal of this project to build interactive tooling in Jupyter Notebooks which can be used by the neuroscientists in Copenhagen to analyze the videos/TIFF-Images on the basis of the developed algorithm by themselves.

This brings us to the question, how can we detect and extract such ROIs? What do the signals in ROIs look like? How can we distinguish interesting dips from unrelevant ones, given the characterization of dips is not entirely defined? How can we define and measure the efficiency of the developed framework? These are questions we intend to answer in the course of the thesis.

Methods mentioned in *Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data* (Pnevmatikakis et al., 2016), such as dictionary learning (Tošić and Frossard, 2011) or local correlations of neighboring pixels(S. L. Smith and Häusser, 2010), where tested and evaluated. However, local correlation was unable to deal with overlapping ROIs and dictionary learning is not able to produce a binary matrix for the spatial position of ROIs.

We have solved the detection and segmentation of these signal drops with a two-step procedure. First, we transform the video data into a form that allows the segmentation of the ROIs. For this we use techniques like smoothing (see chapter 3.3) and detrending (see chapter 3.4) to extract the signals of the dips. For particularly weak dips, we use a signal amplification method (see chapter 3.5) that brings these weaker signals to a comparable level of significance for different intense dips, so that they are significant enough for segmentation. Then, this processed data is passed to a matrix decomposition algorithm (see chapter 3.6) that performs the segmentation of the ROIs. The segmentation helps to distinguish different dip events and to assess their significance.

We show that our framework is capable of detecting and segmenting dips of different significance levels. The results (see chapter 6) are evaluated qualitatively and quantitatively and compared with a few ground truth videos obtained from the University of Copenhagen. On the quantitative level, the ground truth ROIs of different videos are evaluated whether they are found by the predicted ROIs. In addition, various metrics such as Intersection over Union (see equation 3.2), Precision (see equation 3.4), Recall (see equation 3.5) and F1-Score (see equation 3.6) are calculated on a pixel level and the results are compared with each other. On the qualitative level, the found ROIs are compared with the ground truth ROIs and discussed what ROIs are found that do not belong to the ground truth. In the figure 1.1 you can see an example of the predicted ROIs.

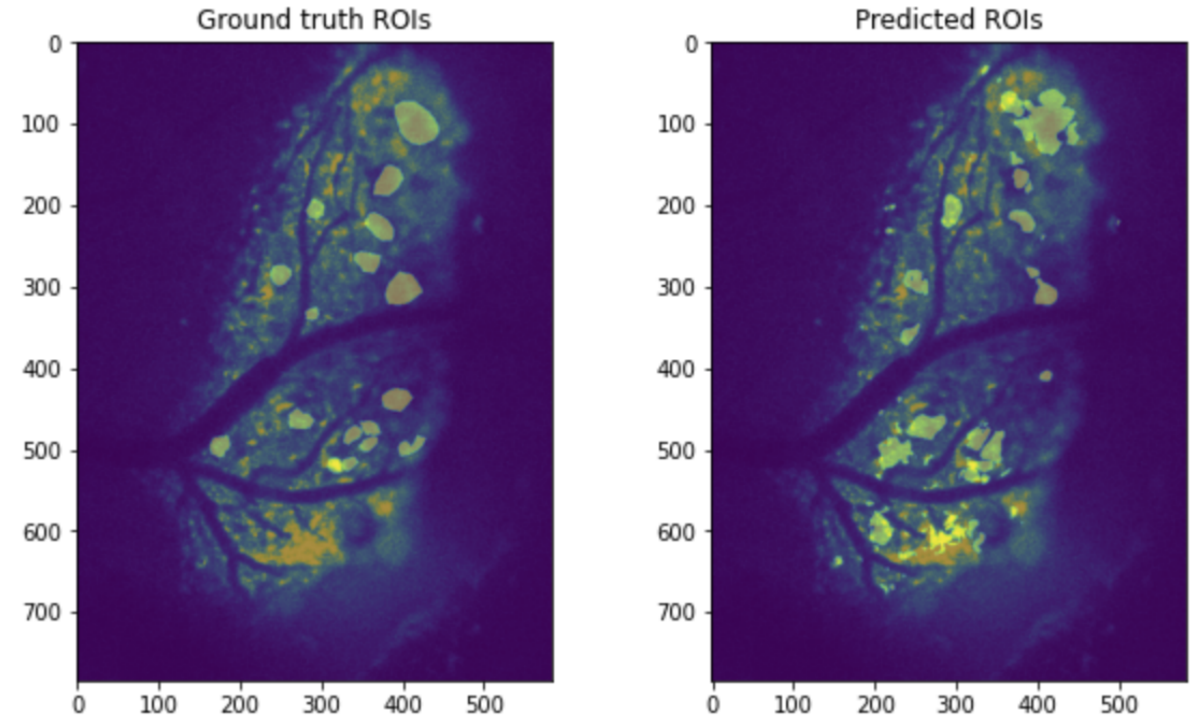


Figure 1.1: Here is an example of predicted ROIs on the left compared to the corresponding ground truth on the right of the video 2021_BLI-Baseline_KX/ nm200109_A2_GFAP-GeNL_awake/01001_mcor.tif. The segmentation was done on amplified signals presented in 3.5.

2 Related Work

Calcium imaging shares some similarities to our problem. Pnevmatikakis et al. say data analysts typically have difficulty identifying ROIs in the optical field, demixing spatially overlapping ROIs, and deconvolving (and denoising) the neuron activity from the much slower dynamics (Pnevmatikakis et al., 2014). They claim that ROI identification is usually based on spatial localization yielding methods for ROI selection based on local correlations of neighboring pixels or dictionary learning (Tošić and Frossard, 2011). Or a different approach stems from the observation that the spatiotemporal activity can be expressed as a product of two matrices where the first matrix encodes the spatial location of each neuron and the second matrix characterizes the calcium concentration evolution of each neuron.

The matrix factorization approach described above relies on a constrained nonnegative factorization that expresses the spatiotemporal fluorescence activity as the product of a spatial matrix that encodes the spatial footprint of each neuron in the optical field and a temporal matrix that characterizes the calcium concentration of each neuron over time. In addition, Pnevmatikakis et al. combined this framework with a constrained deconvolution approach that extracts estimates of neural activity from fluorescence traces, to create a spatiotemporal processing algorithm that requires minimal parameter tuning (Pnevmatikakis et al., 2016).

Another interesting method comes from astrocyte and neurotransmitter fluorescence dynamics for single-cell and population-level physiology (Wang et al., 2019). The authors Wang et al. proposed an analytical framework that releases astrocyte biologists from ROIs based tools using a new framework called AQuA. It is data-driven and based on machine learning principles. They defined an event as a cycle of a signal increase and decrease that coherently occurs in a spatially connected region defined by the fluorescence dynamics. Algorithmically, this definition is converted to the following: The temporal trajectory for an event contains only a single peak and adjacent locations in the same event have similar trajectories. So the strategy of event detection is to find peaks, which are used to specify the time window and temporal trajectory; explore the smoothness to group spatially adjacent peaks and finally apply machine learning and optimization techniques to iteratively refine the spatial and temporal properties of the events to best fit the data. For evaluation purpose they used spatial and temporal intersection over union (IoU) on a simulated dataset.

3 Methods

3.1 Data Set

We have 18 videos oxygen activity in a small area of a brain. Briefly, for each single mouse, two recordings have been made. Either awake and then under anesthesia or awake and again awake as a control group. Most of the videos have different numbers of pixels and frames. However, they were recorded at the same frame rate of 1 Hz and the pixel-size always refers to 2.83 um/pixel. These files are post-processed, meaning they are motion corrected as much as possible and saved as 8-bit uncompressed .tif files. The indicator dye becomes visible by adding the GeNL protein to a local site of the brain. The intensity of the indicator dye depends on the oxygen activity, where the value range of the recording is between 0 and 255. The signal was recorded with a EMCCD Camera Andor iXon. It is in the nature of this method that the visible intensity thereby is expected to steadily decreases over time since the protein is degraded over time. A video is assumed to be stable regarding spatial movement. An example of how a video looks like can be seen in figure 3.1 and an example dip of this video in figure 3.2. The dimension of the video is T x H x W.

- T: are the number of frames of the video.
- H: count of pixels in the vertical direction.
- W: count of pixels in the horizontal direction.

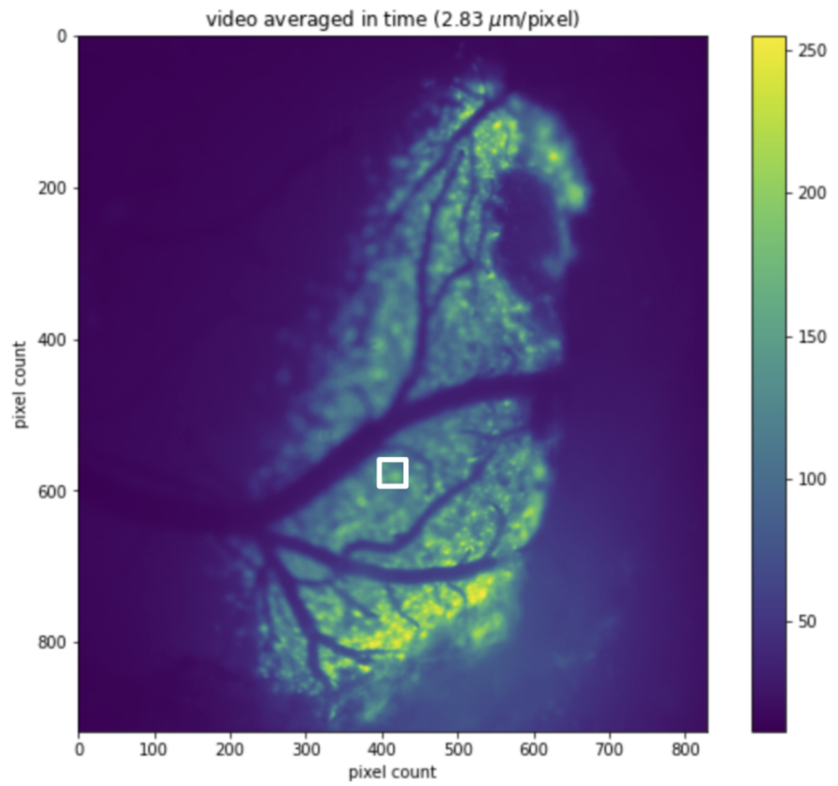


Figure 3.1: Here is an example image which represents an in time averaged video 2021_BLI-Baseline_KX/nm200109_A2_GFAP-GeNL_KX/01004.tif. The colorbar on the right displays the intensity values of this image. The marked region shows an instance of an ROI.

The question arises what actually characterizes a so-called ROI. An ROI is a spatial region in which a signal dip event over time occurs. These regions can be described with spatial polygons, where the pixels at the boundary and inside count towards the ROI. A specific signal is then extracted by averaging the spacial ROI region pixels over time. These signal dips are not yet well researched and therefore, not much is known about their characteristics. After some discussion with the scientists of Copenhagen, we have recorded the following dip characteristics:

- Abrupt, relatively steep signal dip followed by a relatively steep signal rise after unknown amount of time.
- Intensity change not specified exactly, but must be significant.
- Spatially limited event, but need not necessarily maintain exact shape over time.
- After a dip a small peak is assumed (see figure 3.2), but it does not have to be present.

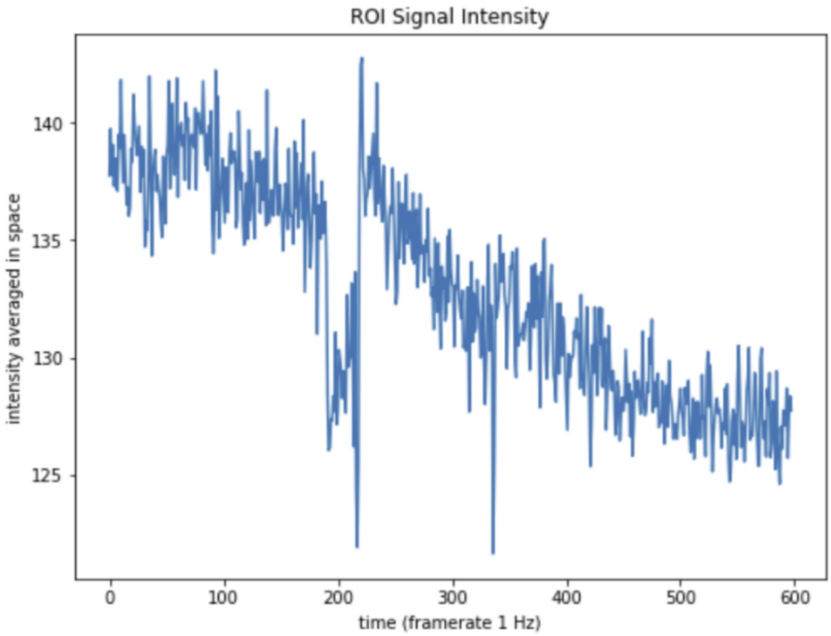


Figure 3.2: This signal describes the in space averaged intensity of the ROI depicted in figure 3.1. There is a sharp dip from frame 180 to 220. After the ascension, a small peak can be observed. In addition, a steadily decreasing trend over the whole signal can be noticed.

3.2 ROI Evaluation

For the evaluation of the ROIs, we received ROI lables for a few videos from our scientific collaborators at the University of Copenhagen. These lables consist of an ROI set of polygons in space created with the image tool Fiji. The evaluation is done in a qualitative and quantitative manner. For the quantitative part, using the ground truth, we now have the possibility to compute various metrics. The following metrics are calculated in a two-dimensional setting, since the ground truth provided is also two-dimensional.

On one side we want to investigate how well the predicted ROIs correspond to the ground truth in terms of each individual ground truth ROI. For this purpose, we classify for each ground truth event, if it is predicted using the coverage of the ground truth event A and all predicted events B compared to the size of the ground truth event (see equation 3.1). If the coverage exceeds a threshold of 0.5, we consider the ground truth ROI as successfully predicted. We decided against explicit ROI assignment of predicted and ground truth ROIs because a ground truth ROI may consist of multiple predicted ROIs and a predicted ROI may cover multiple ground truth ROIs, making assignment very difficult.

$$gt\_coverage = \frac{|A \cap B|}{|A|} \tag{3.1}$$

On the other hand, we want to calculate additional metrics at a pixel level to support the ground truth ROI classification. Otherwise, we could simply assume all pixels as predicted and thus classify all ground truth ROIs as predicted using the equation 3.1. Therefore, the following metrics, we consider the ground truth as well all predicted ROIs as a whole: IoU (see equation 3.2), accuracy (see equation 3.3), precision (see equation 3.4), recall (see equation 3.5), and the f1-score (see equation 3.6).

Intersection over Union

Intersection over Union (IoU), also known as Jaccard index is a popular metric in object detection. It tells how much a predicted region A matches another region B by dividing the intersection of both region by it's union. In our case, A are the spatial pixels of all ground truth ROIs and B are the spatial pixels of all predicted ROIs. IoU ranges from 0 to 1, where 1 indicates a perfect match of the ground truth and prediction.

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{|I|}{|U|} \tag{3.2}$$

Accuracy

Accuracy (Acc) measures how close or far off a given set of measurements are to their true value. It ranges from 0 to 1 where 1 means a perfect match of the predictions with the ground truth.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.3}$$

Precision

Precision (Prec) is the fraction of relevant elements among the retrieved elements. It ranges from 0 to 1 where 1 indicates that all predicted pixels belong to the ground truth ROIs.

$$Prec = \frac{TP}{TP + FP} \tag{3.4}$$

Recall

Recall (Rec) is the fraction of relevant elements that were retrieved. It ranges from 0 to 1 where 1 means that all ground truth pixels have been predicted.

$$Rec = \frac{TP}{TP + FN} \tag{3.5}$$

F1-Score

The F1-score combines the precision and recall into a single metric by taking their harmonic mean. It ranges from 0 to 1 where 1 indicates a perfect prediction.

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{3.6}$$

3.3 Signal Filtering

Wherever data is recorded from real sensors, white noise is typically involved. This is usually undesirable and distorts the actual signal of interest, making analysis difficult. In our case, we have spatial and temporal noise since we operate on videos. To improve signal to noise, we implemented Gaussian, mean and Fourier filtering methods. The following paragraphs describe mean, Gaussian, and Fourier filtering and what effect they have on the signals.

Mean Filtering

Mean or moving average filters are very common in signal processing. They work well as smoothing filters, but poorly as low-pass filters (S. W. Smith et al., n.d.). As the name implies, the mean filter operates by computing the mean on a defined time window moving along the signal. In equation form (S. W. Smith et al., n.d.) it is written as:

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j] \tag{3.7}$$

Where $x[]$ is the input signal, $y[]$ the output signal and M the window size. The simple mean filter is a convolution using a simple filter kernel $\mathbf{x} \in \mathbb{R}^M$ (for one dimensional case) where each element of $\mathbf{x}_i = \frac{1}{M}$. This can be applied to multiple dimensions by integrating for each dimension, where the kernel $\mathbf{x}$ is then a corresponding tensor.

Gaussian Filtering

Gaussian filtering is similar to mean filtering but with a Gaussian instead of a mean kernel. The distribution of a Gaussian kernel values follows a Gaussian distribution for each axis with mean $\mu = 0$ a corresponding standard deviation σ for each dimension. Since the kernel values approximate a Gaussian distribution, the sum of the kernel $\mathbf{x}$ results in 1, not mattering the dimension or the different σ. In figure 3.3 a Gaussian kernel can be seen with σ 3 and a kernel size of 9.

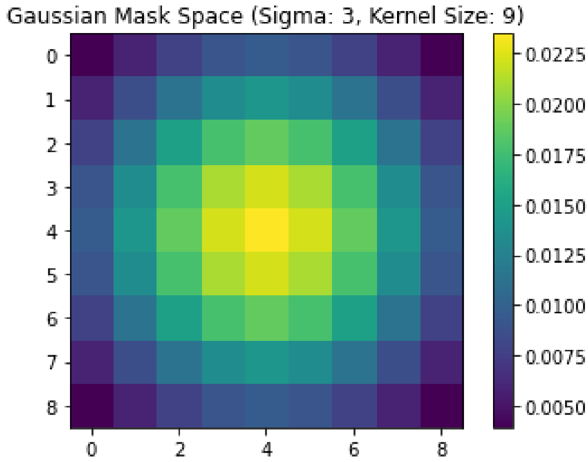


Figure 3.3: Image of the values of a spatial Gaussian kernel used for signal smoothing.

Fourier Filtering

Fourier transformation (see equation 3.8 and 3.9) is an important signal processing tool which is used to decompose signals into its sine and cosine components. The Fourier filter is a type of filter function based on manipulation of specific frequency components of the signal. It works by taking the Fourier transform of a signal, then attenuating or amplifying specific frequencies and inverse transforming it back. So using a Fourier filter, one is able to build a low- high- or band-pass filter. In the Fourier domain, white noise has equal amplitude at all frequencies giving it a constant power spectral density (PSD). After the transformation of a temporal signal into its Fourier components and thus get the PSD of the signal, one should be able to reconstruct the signal using only the Fourier components with a relatively high PSD where the rest can be considered as mainly white noise.

$$f(x) = \sum_{k=-\infty}^{\infty} c_k e^{ikx} \tag{3.8}$$

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-inx} dx \tag{3.9}$$

Convolution

Mean and Gaussian filtering can be efficiently implemented with convolution (see equation 3.10), where $f(\tau)$ is the kernel. The kernel as well as the integration can also be done over multiple dimensions. In our case, we have a three-dimensional convolution since the video is a three-dimensional tensor (no RGB channels). So we can define a 2-dimensional mean or Gaussian kernel on the spatial plane as well as another 1-dimensional mean or Gaussian kernel in the temporal dimension. However, instead of performing convolution on each frame in time and convolution on each spatial pixel, we can do this with a single 3-dimensional kernel, leading to the same result. If the spatial kernel is stacked n times in a row as the temporal kernel has elements and the temporal kernel stacks for each spatial element, two 3-dimensional tensors are formed. When these are multiplied element by element, this forms the 3-dimensional kernel. An example of what a performed convolution can be seen in figure 3.4.

$$f(t) * g(t) = \int_0^t f(\tau)g(t-\tau)d\tau \tag{3.10}$$

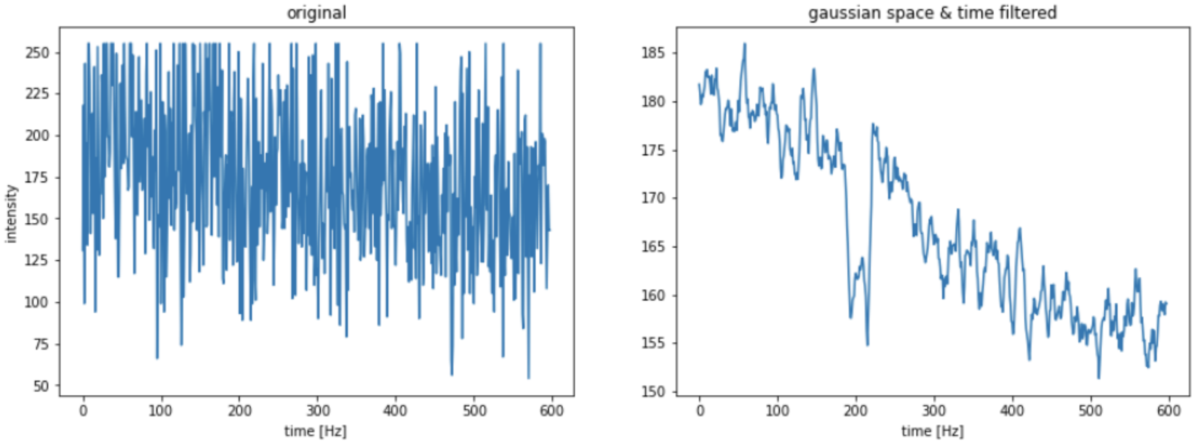


Figure 3.4: On the left is the original signal of a spatial pixel of a video in time. A Gaussian filtering in space and time with a kernel size of (7,9,9) TxHxW with a temporal σ of 5 and spacial σ of 3 is applied.

3.4 Detrending

In the domain of signal processing, there are often trends which are not of interest. These trends usually complicate further signal analysis because some methods assume some degree of stationarity. Detrending assumes a model that must be flexible to fit the trend, yet inflexible so to not absorb the signal of interest. It is in the nature of the data of our experiments that the luminosity of the signals is expected to decrease as the protein slowly degrades.

There are various methods to approximate the trend of a signal. Since we often expect a non-linear trend, the chosen method must be able to address the non-linearity. In addition, the method should be robust against outliers, especially against signal dips, since these are not part of the true trend of the signal. Therefore we have chosen a polynomial robust detrending (Holland and Welsch, 1977) method of unknown order. Unknown, because we want to address the possibly different trends of different data sets. In addition, we also have observed that the signal trends are spatially dependant (see figures 4.2 - 4.6). Thus, our detrending method has to address spatial locality.

Polynomial Detrending

We have chosen a polynomial detrending, because a polynomial is easy to understand and therefore therefore it is simpler to make assumptions from the signal into the order of the polynomial. In this particular case, we want to subtract a polynomial of n th-order, which has the following form:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + ... + \beta_d x_i^d + \epsilon_i \tag{3.11}$$

where ϵ_i is the error therm. The higher the order d of the polynomial, the more flexible the function y_i. The coefficients β can be easily estimated via the ordinary least squares (OLS) solution, since y_i is linear in β and thus just a standard linear regression model.

Robust Detrending

Detrending can fail if a glitch is mistaken for a trend, or a trend is mistaken for a glitch. To avoid this scenario, it may be useful to first use a low-order and then a higher-order model whose weights are constrained by the previous fit. Simple trends are easily fitted with a low-order polynomial or with just a few terms of a Fourier series. However, we wanted to have a method where we could control the robustness in a more defined way, and in addition, we wanted to be able to consider signal dips as outliers, but not necessarily signal peaks, since a dip can be interpreted as a signal between two peaks and the properties of signal dips are not really known well.

To achieve robust detrending we used a technique called iterative weighted regression (Holland and Welsch, 1977). This works by initially starting the data X with weight (usually 1) and then iteratively fitting a model where the weights are updated by a weight function H depending on the residuals, until a stopping criterion (usually convergence) is met (see algorithm below). First, a weight function H has to be defined. Since we were not sure in areas where alternating signal occurred whether they were dips or spikes, we would like to be able to treat the area between alternating signals as dips, meaning the assumed trend would lie more on the top and not in between. Thus, we have implemented a set weight functions which are able to treat positive and negative residuals differently.

Algorithm 1 Algorithm for robust detrending using asynchronous weighted residuals.

$\mathbf{w} \leftarrow [1, 1, .., 1, 1]$
$\mathbf{X} \leftarrow [1, 2, .., n-1, n]$
while $i < max\_iter$ **and** $\mathbf{w} \neq \mathbf{w}_{-1}$ **do**
    $\hat{\mathbf{y}} \leftarrow$ **polyfit(X, y, w)**
    $\mathbf{res} \leftarrow \mathbf{y} - \hat{\mathbf{y}}$
    $\mathbf{res\_rel} = \mathbf{res}/max(|\mathbf{res}|)$ ▷ **puts residuals in interval [-1,1]**
    $\mathbf{w} \leftarrow H(\mathbf{res\_rel})$
end while

As you can see above, we compute the relative residuals compared to the maximal residual while preserving the sings and put them into the interval [-1,1]. This allows to build functions H which operate on input ranges of [-1,1] for re-weighting. With this, one is able to treat relatively large or small residuals for the next polynomial fit as they like. We have implemented the following weight functions where s is a free parameter for shape control:

$$H_s(x) = (1 - |x|) \cdot s \tag{3.12}$$

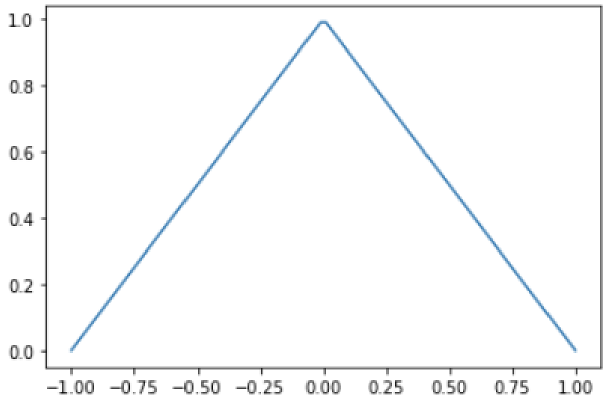


Figure 3.5: The hat function (see equation 3.12) treats positive and negative outliers equally where s controls the steepness of the slopes. The higher s the higher the the data with low residuals are weighted.

$$H_s(x) = \frac{tan(x \cdot tan^{-1}(\frac{s}{2}))}{s} + 0.5 \tag{3.13}$$

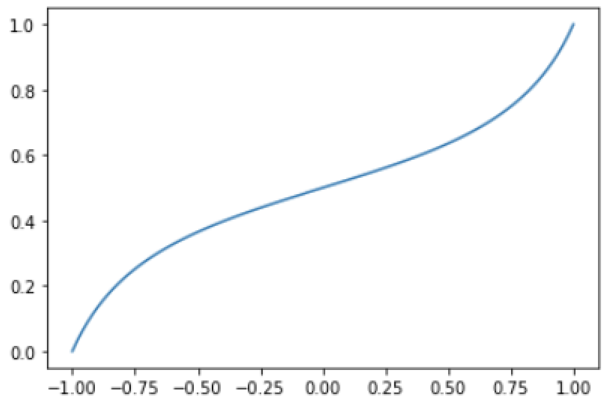


Figure 3.6: The tangential equation 3.13 tends to weight very low negative residuals weakly and very high, positive residuals strongly. Small residuals near zero tend to be weighted similarly, although this depends very much on s. Here, s was chosen to be 5. With larger s, the function becomes steeper at the ends and flatter in the middle.

$$H_s(x) = \frac{x+1}{2} \cdot s \tag{3.14}$$

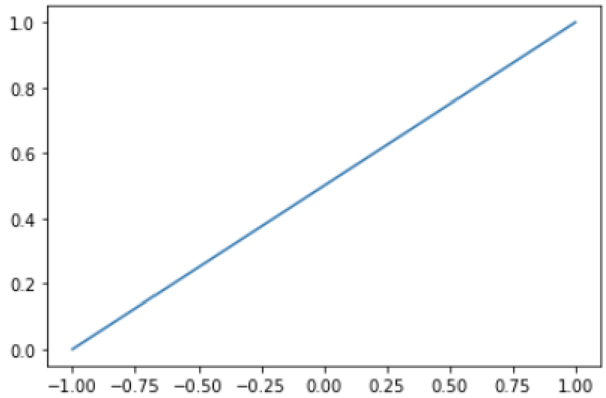


Figure 3.7: The equation 3.14 (here with $s = 1$) shows a linear weighting of the relative residuals. That is, the more positive the residuals, the more weight they receive. A linear behavior could also be achieved with the equations 3.15 and 3.13. However, an advantage is that s controls the slope of the linear behavior, i.e., the range of values can go from $[0,s]$, which gives signal intensities with large positive residuals an even much stronger weight for the next fit.

$$H_s(x) = \frac{1}{1 + e^{-x \cdot s}} \tag{3.15}$$

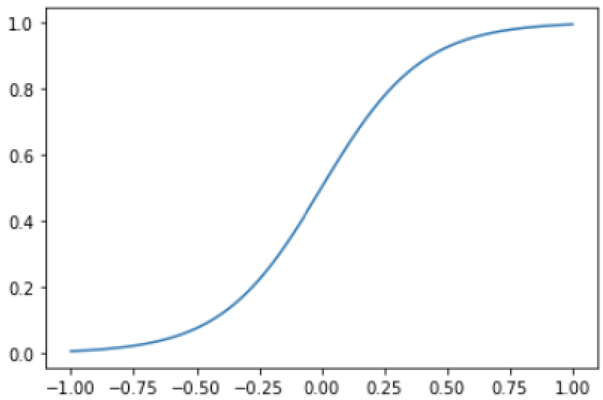


Figure 3.8: The logistic sigmoid function 3.15 has an S-shaped form (here $s = 6$). That is, signal points with negative residuals tend to have a small weight, but this weight increases rapidly as the residuals become more positive. How fast more positive residuals gain weight depends on s. The higher s is chosen, the more the function takes on an S- or step-shaped form. In the extreme case, data points with negative residuals could be almost completely excluded from the new fit of the next iteration.

Robust detrending can now be performed in space on a pixel-by-pixel basis. Pixel-wise, because locally different trends are present. In figure 3.9 the detrending is shown with an example. We have taken the smoothed signal of a space pixel and plotted the signal in time.

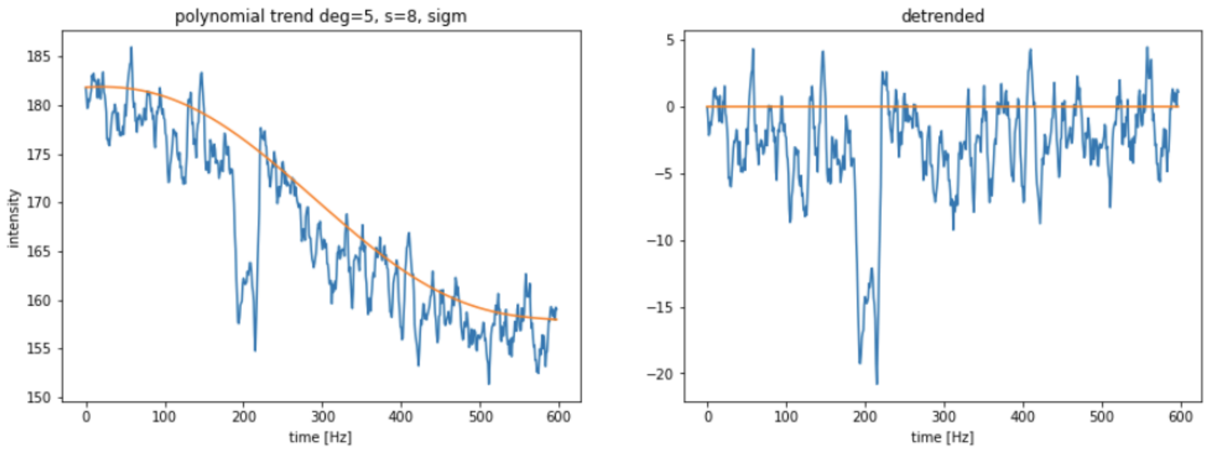


Figure 3.9: On the left is the smoothed signal and the assumed trend. On the right is the same signal where the trend was subtracted. For this a 5th degree polynomial was used with a sigmoid weight function (see equation 3.15) with $s = 8$. Here you can see very well how the trend tends to attach to the upper signal peaks.

3.5 Signal Amplification

Some regions have signal dips of relatively low intensity, i.e. with a much smaller signal to noise than other regions. These are barely visible to the eye. The relatively weak signal changes can also severely impact their detection with algorithms. Therefore, we have developed a signal transformation that makes a noise level estimate and attenuates signals that have been classified as noise and amplifies the rest. This can be applied on each space pixel in time assuming a smoothed and detrended video.

Noise Level Estimation

The estimation of the noise level is a crucial part of the signal amplification. We use block minima (i.e. dividing a timeseries in equal blocks and take the minimum of each block) with a constant block size and take the mean value. The signal intensity which is closer to 0 than the mean is then considered as noise. Other measures such as the median would is also possible, but we have not implemented this.

This method of noise level estimation assumes that there is not much signal activity in the signal because then the noise level will be estimated too large and thus rather small dips in this region will disappear. Furthermore the block size plays a central role. A too small block size (e.g. 1) estimates the noise level too high and a too large block size (entire signal) too low. Assuming that the noise level has been estimated correctly, we divide the signal by the level. In the negative spectrum of the signal, the noise lies between [0,-1] and the assumed signal < -1. It is important to note that we are not trying to estimate a special kind of noise like white noise here, but are only looking for the relevant portion, so that too small signal fluctuations are not unnecessarily amplified, which would worsen the signal to noise.

Amplification

After the noise level estimation we now have a noise level of the negative values of -1 everywhere, assuming the estimation is correct. The idea of signal amplification is to bring weaker signals

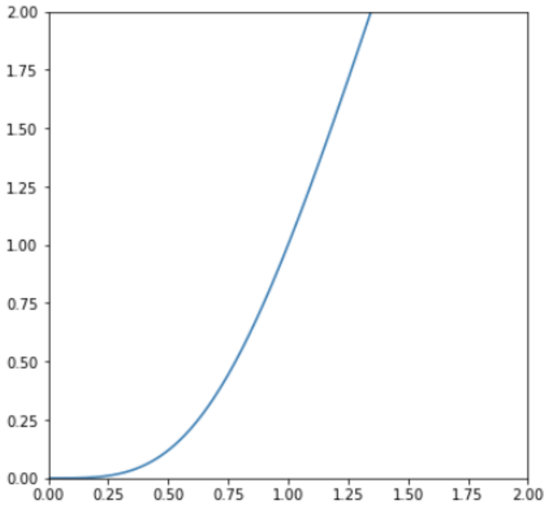


Figure 3.10: Here $f(x)$ is shown in the interval [0,2]. You can see the convex behavior between the interval [0,1] which leads to an attenuation of the noise. However, the closer x gets to 1, the less damping effect the function has.

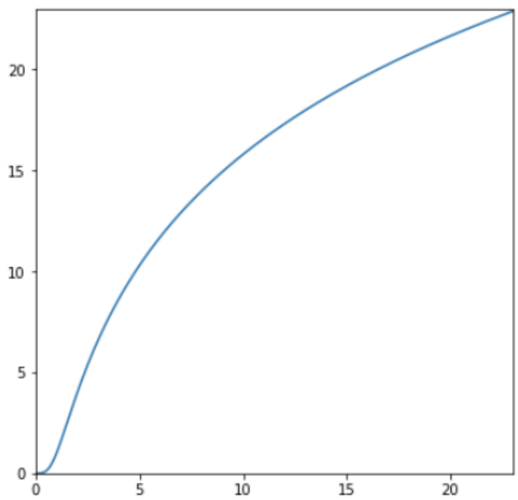


Figure 3.11: $f(x)$ where $x > 1$ becomes concave. Due to the already very steep slope, however, rather weaker signals are thus strongly amplified. However, this amplification decreases steadily. From $x = 23$ the signal is attenuated, but we do not expect signal to noise ratios of 23 on the videos.

closer to the level of stronger signals so that they have a comparable level. We achieve this with a function which is convex in the interval [0,1] and concave in the range greater than 1. If we then compute $(-1) \cdot f(-x)$ for all negative signal values x, the assumed noise is attenuated as well as the assumed negative signal dips are amplified. Now the question is what exactly $f(x)$ looks like. Based on empirical tests we have developed the following function 3.16:

$$f(x) = (ln(x^3 + 1) \cdot 1.442695)^{\frac{6}{5}} \tag{3.16}$$

The procedure amplifies (in a relative manner) weaker signal dips with an estimated signal to noise of > 1 and attenuates the estimated noise. Due to the concave behavior, already strong signal dips are still amplified, but not as much as weaker dips, which should bring the signal level of the dips to a more comparable level. Since the method depends very much on how well the noise level estimation was done and how much noise is still above this level, it happens that noise is also amplified (see figure 3.12). We are aware that this is a very experimental function, where the estimation of the noise level could be far better performed and the equation 3.16 could be better retrieved in an evaluated procedure.

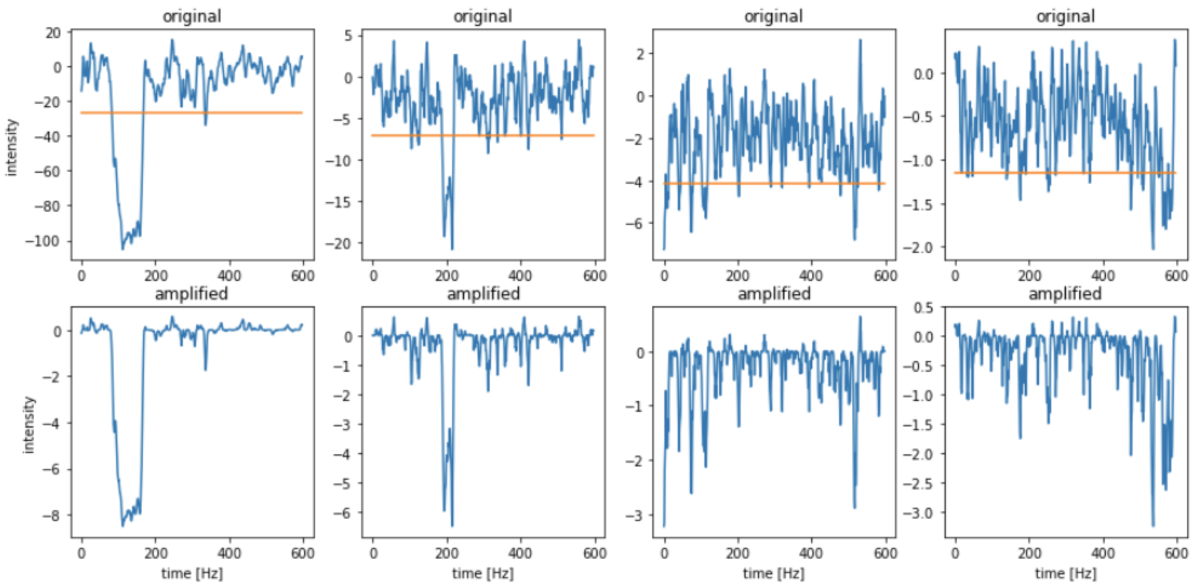


Figure 3.12: Above are four different detrended and smoothed signals from different space locations, where the orange line is the noise level estimation estimated with the mean of the block minimas with a block size of 20. The first two signals have dips, but they are quiet different regarding the signal to noise ratio. The latter two signals just look like noise, but with differences in signal intensity. Below we see that the signal intensity of the two dips have massively converged and the Noise has been attenuated. However, in all amplified signals there are also individual noise fragments introduced by this process.

3.6 Matrix Factorization

Introduction

A very promising approach for calcium imaging is NMF (Pnevmatikakis et al., 2016). However, because our signal is neither always positive nor negative, we have implemented a version of SNNMF (Semi-non-negative matrix factorization). The basic idea is two find two matrices SP and ST which when multiplying them reconstructs the initial matrix with the lowest possible matrix error, as defined in equation 3.17. When we minimize this error, we reduce the error iteratively by optimizing the two matrices alternately.

$$\|SMT - SP \cdot ST^T\|_\mathrm{F} = \sqrt{\sum_{i,j=1}^{n} |SMT_{ij} - (SP \cdot ST^T)_{ij}|^2} \tag{3.17}$$

To achieve this, we need to take some assumptions as defined below in 3.6.

- one of the matrix should be the positional, binary matrix and show where the dips are located.
- the other matrix should reconstruct the time series of the dips and allow all numeric values.

Shape of the Matrix

The matrix SMT is a representation of the video with two dimensions. Each column represents then one image and each row the time series of one pixel and is called the spacial temporal matrix $SMT_{d \times T}$ with d being the number of pixel in one frame and T being the number of frames in the video.

In the image 3.13 we can see that the number of rows corresponds to the number of total pixels (the “resolution” of the video, $d = m \times n$) per frame and the number of columns to the number of frames in the video.

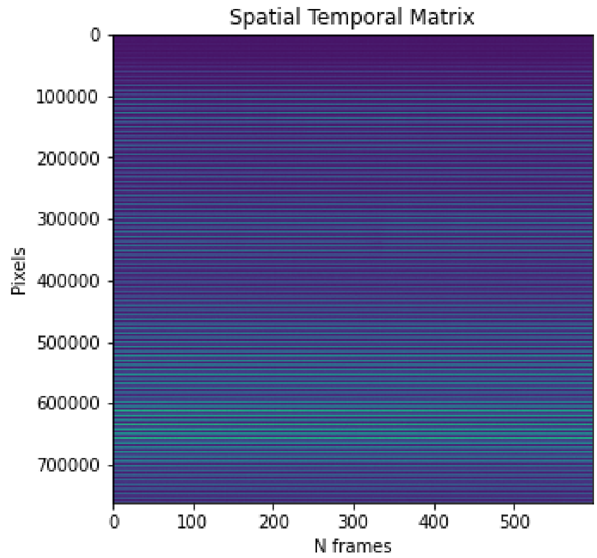


Figure 3.13: Spatial temporal matrix created from the video *01004.tif*. One column corresponds to one frame and one row corresponds to a pixel over time in the video.

The positional and boolean matrix $\mathcal{SP}_{g\times d}$ with g being the selected number of dips we would like to detect. Thus, one row represents the position of a dip. As it can be seen on the missing dimension T, we assume that a dip does not change position nor shape. The time series matrix $\mathcal{ST}_{T\times g}$ contains the time series of a dip. Then, we can reconstruct the matrix SMT as defined below in 3.18.

$$\underset{(d\times g)}{SP} \cdot \underset{(g\times T)}{ST^T} = \underset{(d\times T)}{SMT} \tag{3.18}$$

Initialization

For both matrices, we implemented and evaluated the following approaches:

- random initialization with possible values.
- initialization with a fixed value.

In addition, because the positional matrix SP should always have only positive values, we looked at proven methods from Non-Negative Matrix Factorization and implemented Nonnegative Double Singular Value Decomposition (NNDSVD) [Boutsidisa and Gallopoulosb, 2007] with the help of the Python Package Nimfa [Zitnik and Zupan, 2012].

In our experiments, initializing SP with NNDSVD led to the most sparse positional matrix and quickest convergence, which was also observed by Boutsidisa and Gallopoulosb (Boutsidisa and Gallopoulosb, 2007). For the time series matrix ST, a value in the middle of the average dips should be select but using the mean or median of the matrix to be factorized works well too.

Factorization

After initializing both matrices, the best decomposition for the reconstruction of the initial matrix has to be found. As proposed by (Pnevmatikakis et al., 2016), we use the Least Angle Regression (LARS) (Efron et al., 2004) where LARS is a "forward stepwise regression" method. With LARS, we update SP and ST alternately and converge to the two matrices, which best reconstruct the initial matrix.

In addition, to increase sparsity of the SP matrix, we use a lasso model with an l1 regularization factor $\alpha \neq 0$ which minimizes the objective defined in equation 3.19.

$$\min_{\beta \in \mathbb{R}^p} \left\{ \|y - X\beta\|_2^2 + \alpha \|\beta\|_1 \right\} \tag{3.19}$$

For the ST matrix, alpha is set to zero because sparsity is not desired.

For the SP matrix to be binary, we set all values in the positional matrix after each convergence of LassoLars to either 0 or 1 with the logic defined in equation 3.20 and restrict the LassoLars algorithm to coefficients with only positive values.

$$sp_{i,j} = \begin{cases} 1, & \text{if } sp_{i,j} \geq 0.5, \\ 0, & \text{else,} \end{cases} \tag{3.20}$$

We use LassoLars from scikit-learn [Pedregosa et al., 2011] for our project and thus the objective for the SP matrix is the objective of equation 3.21. As for the ST matrix, the objective is in equation 3.22.

$$SP = \min_{SP \in \mathbb{R}_+{}^p} \left\{ \left\| SMT^T - ST \cdot SP^T \right\|_2^2 + \alpha \|SP\|_1 \right\} \tag{3.21}$$

$$ST = \min_{ST \in \mathbb{R}^p} \left\{ \left\| SMT - SP \cdot ST^T \right\|_2^2 \right\} \tag{3.22}$$

An implementation for the step wise optimization for the SP and ST matrices can be found in the listing 1 below.

Listing 1: Step alternately finding the SP and ST matrices with LassoLars from scikit-learn.

```
def find_SP_matrix(ST, STM, alpha, only_01_values, ...):
    reg = linear_model.LassoLars(
        alpha=alpha,
        positive=True,
        ...
        )

    reg.fit(ST, STM.T)

    if only_01_values:
        reg.coef_[reg.coef_ < 0.5] = 0
        reg.coef_[reg.coef_ >= 0.5] = 1
```

```
    return reg.coef_

def find_ST_matrix(SP, STM, ...):
    reg = linear_model.LassoLars(
        alpha=0,
        ...
        )

    reg.fit(SP, STM)

    return reg.coef_
```

Those steps are then be applied alternately until the frobenius norm (3.17) between the reconstruction and the original matrix does converge. An implementation of the algorithm 2 is described below.

Algorithm 2 Semi-NonNegative Matrix Factorization algorithm with LassoLars.

SP ← *init*(...)
ST ← *init*(...)
max_iter ← ...
i ← 0
while $i < max\_iter$ **do**
  ST ← find_ST_matrix(**SP**, **STM**, ...)
  SP ← find_SP_matrix(**ST**, **STM**, ...)
  error ← $\|SMT - SP \cdot (ST)^T\|_F$
  if $error < best\_error$ **then**
    best_error ← *error*
    best_SP ← *SP*
    best_ST ← *ST*
  end if
  if early_stopping_condition_fulfilled **then**
    Stop
  end if
  i ← $i + 1$
end while

Currently, the algorithm stops when the reconstruction error does not change significantly after some iterations (early stopping). In figure 3.14 we can see how the reconstruction error changes over multiple iterations.

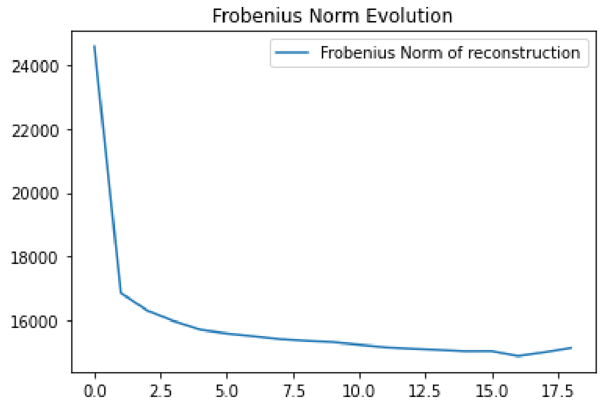


Figure 3.14: The evolution of the error term described in equation 3.17 over multiple iterations of the SNNMF-algorithm described in 2 applied on the detrended and denoised video *01004.tif*. As we can see, the reconstruction error drops with each iteration of SNNMF. The best iteration is number 16 and the algorithm is stopped at iteration 21 because of early stopping.

4 Data Processing

This chapter describes the practical implementation of the first part of the two-step ROI identification process. The first subchapter 4.1 **ROI Analysis**, describes the analysis of some regions which are extracted manually using Fiji. This is followed by **Signal Smoothing** (see chapter 4.2), which improves the signal to noise in space and time. We show how we found the parameters and what effect the filtering has on a video. In the next chapter 4.3 **Detrending** we explain how we take the trends from a video and what the result is. Last but not least, the chapter 3.5 **Signal Amplification** shows how we deal with less significant dips and what the effect is.

4.1 ROI Analysis

In general, it is difficult to identify a specific ROI because it is not always clear how a dip is characterized. To illustrate this and some other interesting features, we look at different ROIs extracted with Fiji and analyze them. We defined five different spatial rectangles (see figure 4.1) which are most likely actual ROIs. The intensity of the zones are then spatially averaged and plotted in time (see figures 4.2 - 4.6).

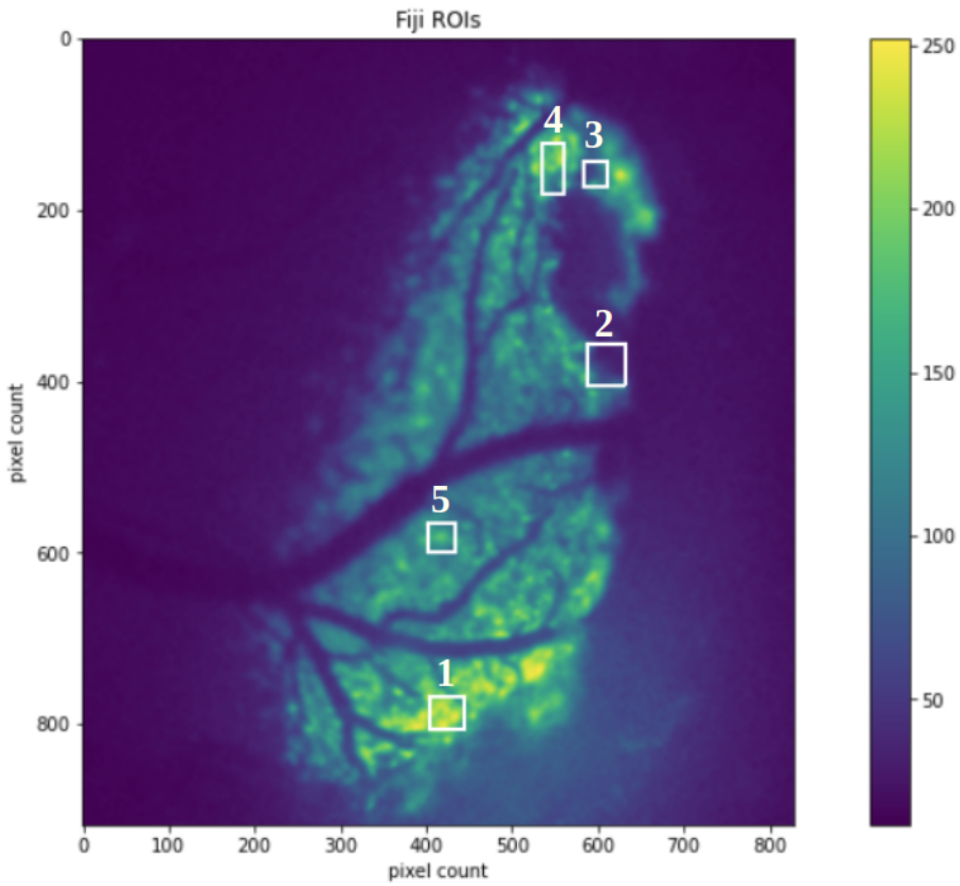


Figure 4.1: Here is the video averaged in time 2021_BLI-Baseline_KX/nm200109_A2_GFAP-GeNL_KX/01004.tif which roughly highlights the regions extracted with Fiji. These are all hand-selected regions because we think they show up various signal characteristics of the video very well.

The different zones show diverse apparent ROIs with very different characteristics. Some signal dips like those of zone 3 and 4 are very strong in terms of intensity and duration compared to those of zone 1 and 5. Additionally, it is not always clear which signal dips are of interest as

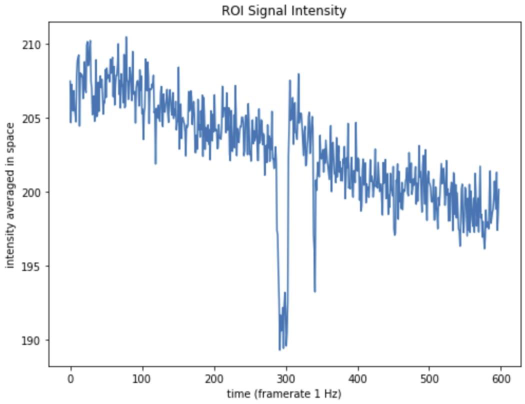


Figure 4.2: Zone 1: Around frame 300 there is a clear dip with an intensity change of about 15, which lasts a little over 10 frames. Shortly after the dip is over, a small peak appears. At frame 340 there is another dip, but much smaller in intensity and time. This zone seems to be clearly classified as an ROI based on the first dip.

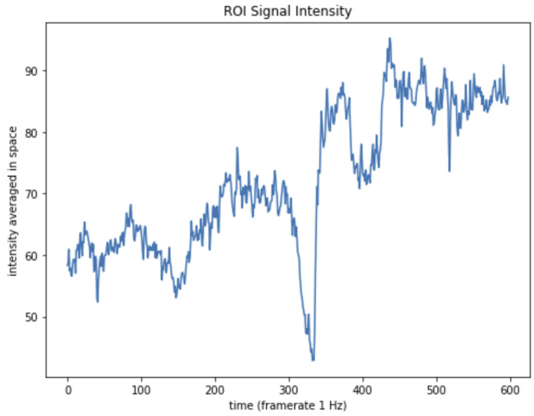


Figure 4.3: Zone 2: While there are a lot of signals dips visible, some of them can be perceived as signal peaks as the signal alternates in time. We discussed this with our scientific collaborators in Copenhagen and concluded that the dip between frames 310 and 330 is a signal dip of interest due to the steep behaviour and the relatively large intensity change of about 30. The dip from frame 390 to 450 seems to be more of a signal dip of interest rather than not since the dip and rise are rather abrupt.

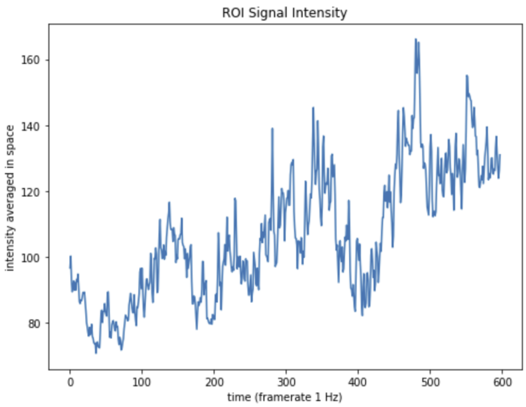


Figure 4.4: Zone 3: This zone looks complex. The signal alternates with a very high intensity, which makes it difficult to classify whether it is a dip or a spike. Assuming the signal dip between frames 350 and 450 is a dip, is the signal between frames 480 and 560 also a dip or is that the trend, which has two small spikes at both ends? Even looking at the signal changes earlier in time, it is difficult to tell if they are more like dips or spikes. Also, our scientific collaborators at the University of Copenhagen were not clear about the classification.

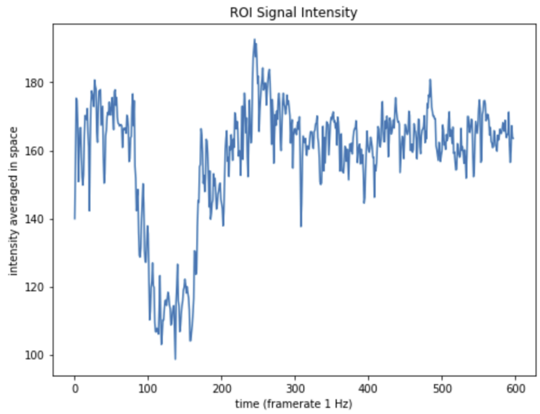


Figure 4.5: Zone 4: Here there are a few interesting characteristics to observe. Between frames 80 and 230 we have a strong signal dip. This is compared to other observed signal dips very high in intensity (around 70) as well as very long in time (more than 100 frames). It does not seem completely clear whether the small part between frames 180 and 200 is a separate event or not. The signal increase after the dip seems to occur again here.

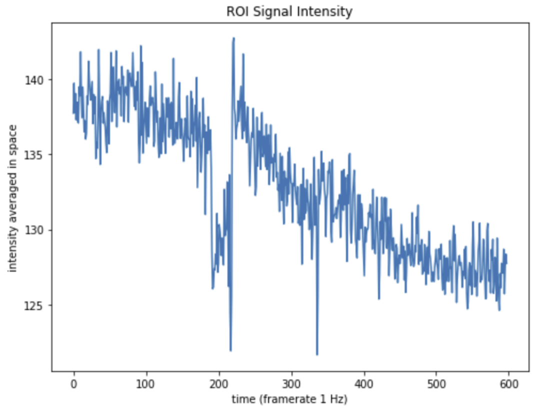


Figure 4.6: Zone 5: In this zone, a signal dip of interest is clearly visible between frames 190 and 215, which is followed by another short peak. The intensity change of the peak is about 10.

can be seen in zone 2 and 3. Furthermore, we can observe that there are locally different trends. Zone 1 and 5 show a downward trend, zone 2 and 3 show an upward trend, and zone 4 seems to be almost trendless.

4.2 Signal Smoothing

The goal of signal smoothing (also called denoising) is to improve the signal-to-noise ratio, especially the signal dips, which are more recognizable and suitable for further analysis. For this, we use filtering methods which are described in chapter 3.3. On our development video 2021_BLI-Baseline_KX/nm200109_A2_GFAP-GeNL_KX/01004.tif we perform Gaussian filtering in space and time. The kernel is shaped (7,9,9) where 7 represents the temporal kernel with a σ of 5 and the (9,9) represents the spatial kernel with a σ of 3 each. The choice of whether to use mean or Gaussian filtering and its exact configurations were experimentally evaluated by looking at how well the reference pixels of the zones matched the zone signal sampled with Fiji (see figures 4.7 - 4.11). Using a Gaussian kernel in space seemed reasonable because we intended to weight closer spatial regions higher to not disturb the spatial images. In addition, using a Gaussian kernel in time also appeared reasonable to weight closer signals higher and thus not interfere too much the short signal dips.

To illustrate the smoothing effect, we look at these reference pixels, which are approximately at the euclidean spatial centre of the zones shown in chapter 4.1 and two additional, different pixels (see figures 4.12 and 4.13) to show the effect on regions where we expect mainly noise. On the left is the original video signal of the reference pixel whereas on the right is the smoothed version of the same location. The pixel coordinates are shown in the title of the figures where the first value is the vertical and the second the horizontal spatial coordinate. The last figure 4.14 shows the spatial effect of smoothing the first frame of the video.

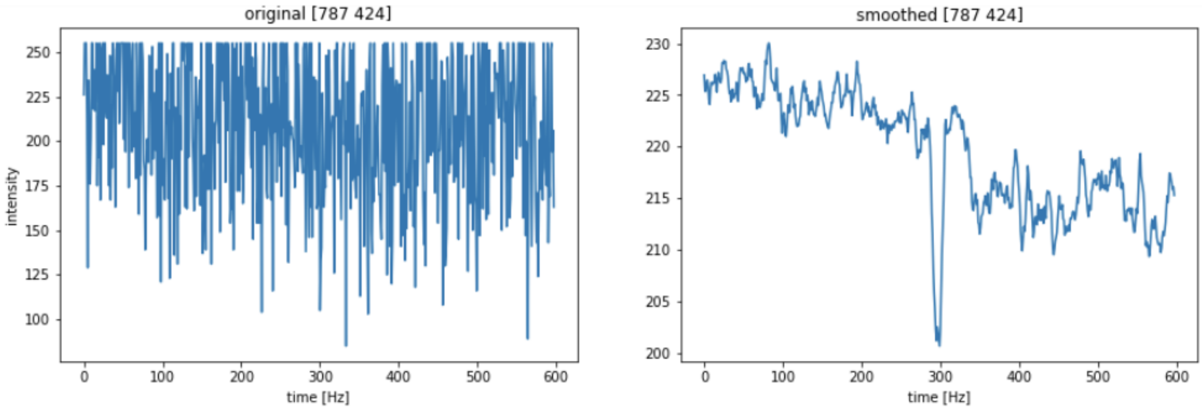


Figure 4.7: Reference pixel of **Zone 1**

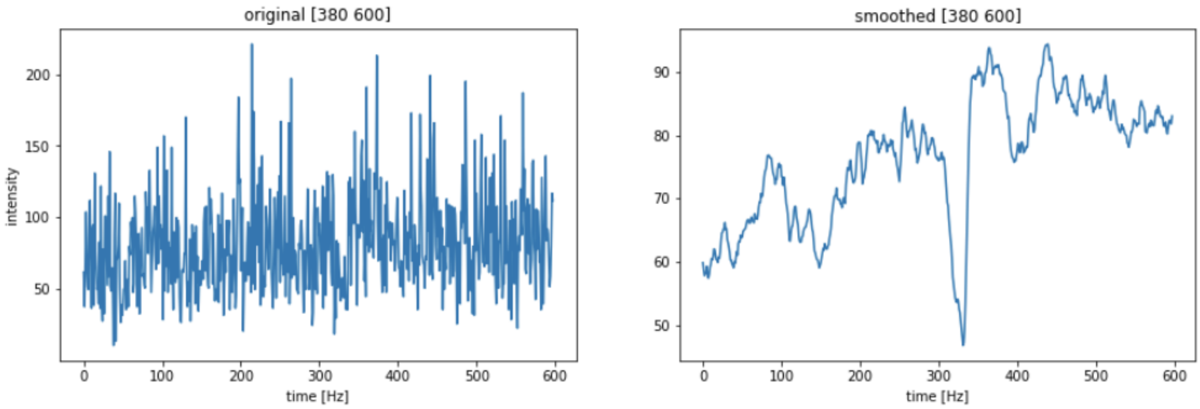


Figure 4.8: Reference pixel of **Zone 2**

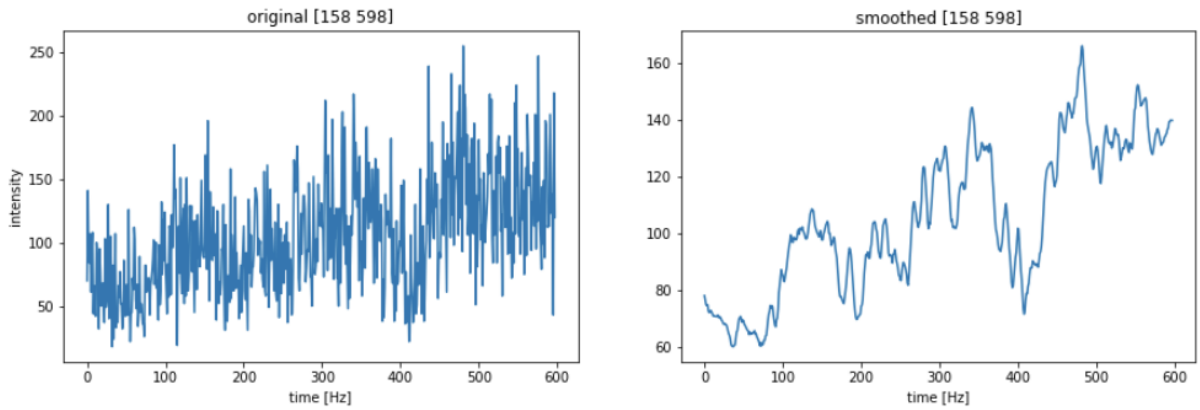


Figure 4.9: Reference pixel of **Zone 3**

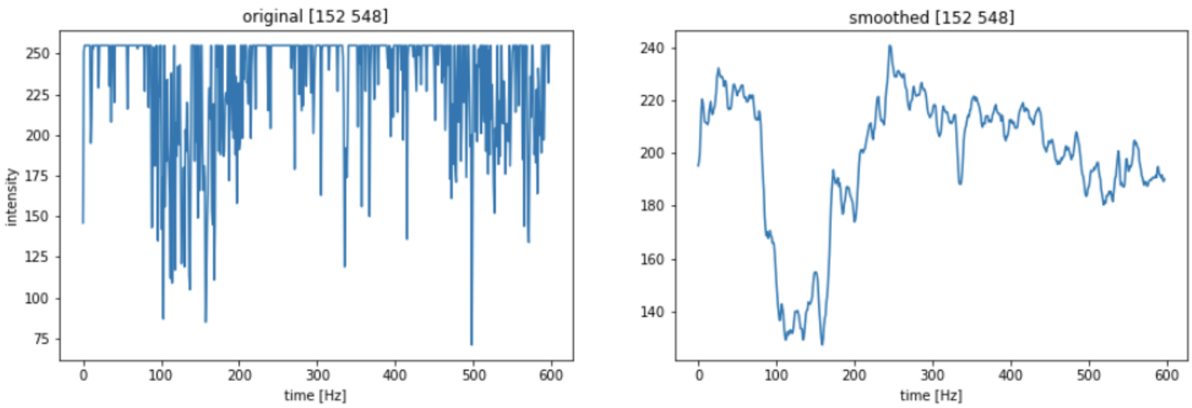


Figure 4.10: Reference pixel of **Zone 4**

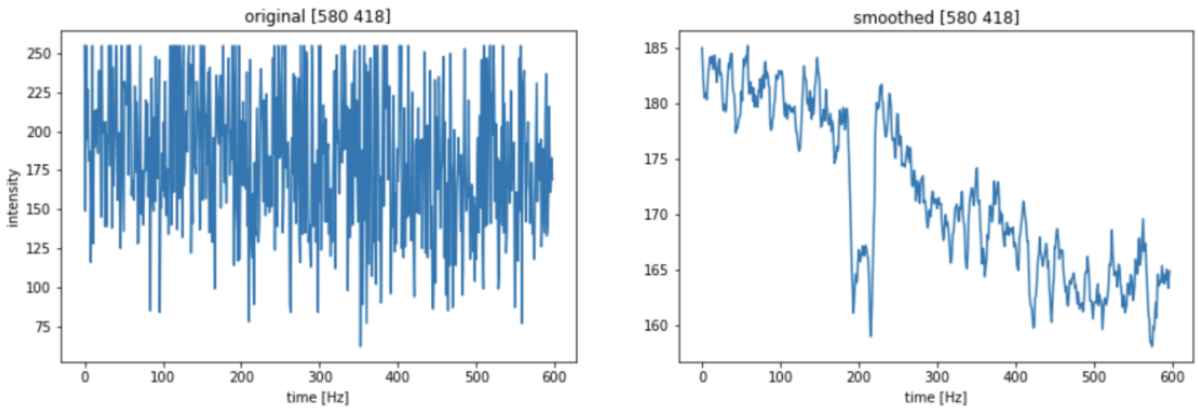


Figure 4.11: Reference pixel of **Zone 5**

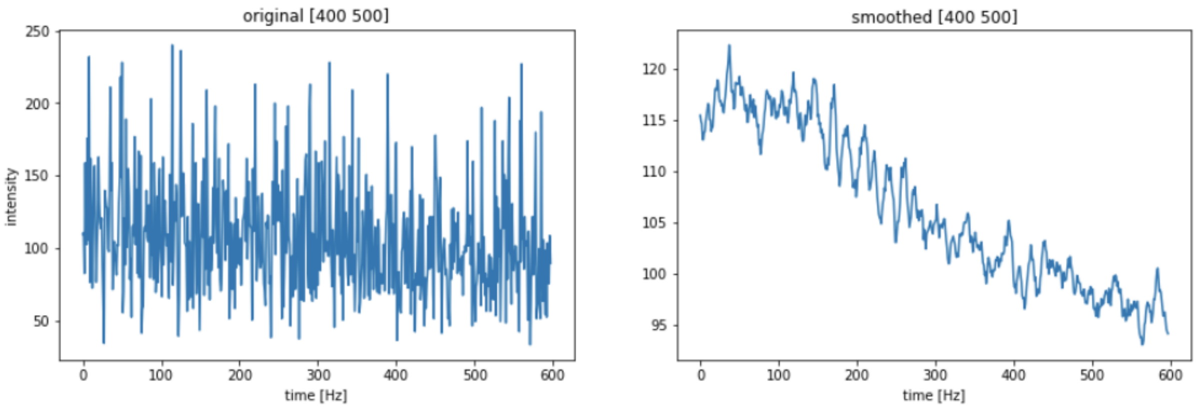


Figure 4.12: Pixel with oxygen activity but no signal dip

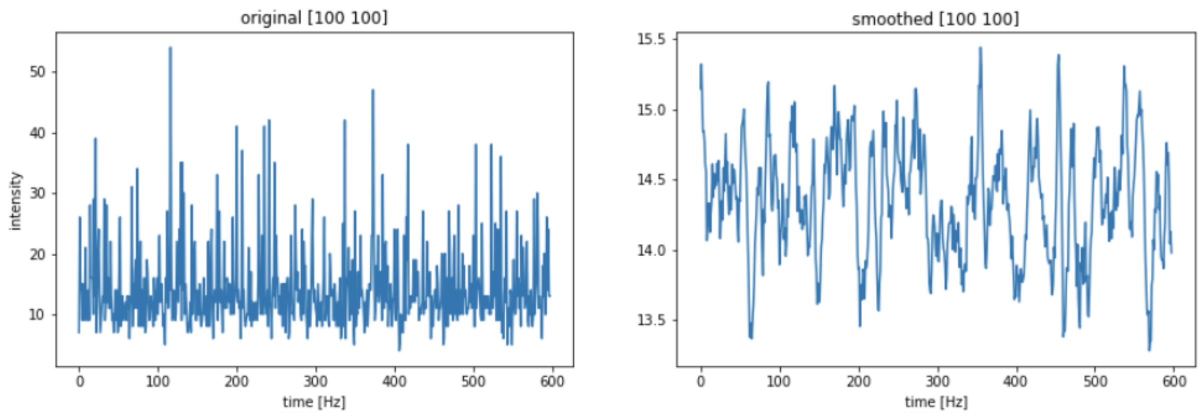


Figure 4.13: Pixel outside of oxygen activity where just noise is expected

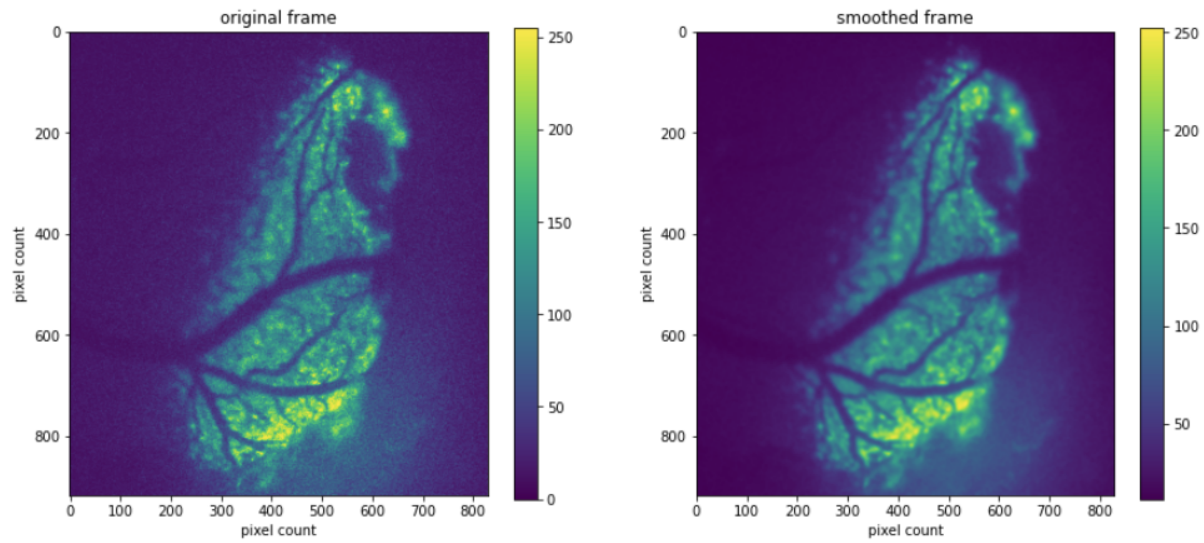


Figure 4.14: On the left is a video frame of the original signals. On the right, spatial and temporal smoothing was applied.

If we compare the reference pixels of the zones 1-5 (see figure 4.7 - 4.11) with the corresponding ROI zones (see figure 4.2 - 4.6), we notice that they are relatively similar. The signals of the reference pixels are even a bit smoother because they are also smoothed in time, which is not the case for the Fiji ROIs. Therefore, we expect that in the closer vicinity of a pixel the neighbouring signals are very similar. The relevant signal features, the signal intensities, and the localized trends seem to agree to some extent. For the figure 4.12, as expected, there is no signal dip and the trend is steadily decreasing. The figure 4.13 shows that even for regions with virtually no signal, smoothing suppresses noise. There is also no trend and the signal fluctuations are very small. The effect of spatial smoothing is also clearly visible in figure 4.14. The spatial noise seems to decrease, but the frame looks a bit more blurry.

4.3 Detrending

After signal smoothing, the signal to noise improved, but the signal dips are still not really detectable yet by further algorithms. This is because dips are relative signals in time, meaning the signal level overwhelms the signal of interest. In addition, an existing trend also makes further signal analysis more difficult, as more advanced algorithms do not handle trends well. The trends and levels represent a slowly changing component in time of a video, which we call background video (example see figure 4.22). We obtain the background video by using the detrending method described in chapter 3.4 for each spatial pixel. We use a pixel per pixel detrending to address the locality of the trends (described in chapter 4.1). On our development video, we performed polynomial detrending of 5th degree, using the sigmoid function with $s = 8$ as the weight function of the residuals (see equation 3.15). The degree of the polynomial as well as which weight function with which s we evaluated experimentally by looking at some reference pixels of assumed ROIs (see figures 4.15 - 4.21) and what the effect polynomial degree, the weight function H and the corresponding choice of s was. The choice of these parameters can influence the assumed trend on a large scale. We have chosen the above depicted parameters based on what we thought could or should be dips based on the characteristics depicted in chapter **Data Set** 3.1. The next few figures 4.15 - 4.21 show the effect of detrending at the in chapter 4.2 defined reference pixels. On the left, each figure shows the filtered signal and the corresponding estimated trend. On the right side the corresponding background video has been subtracted from the smoothed video.

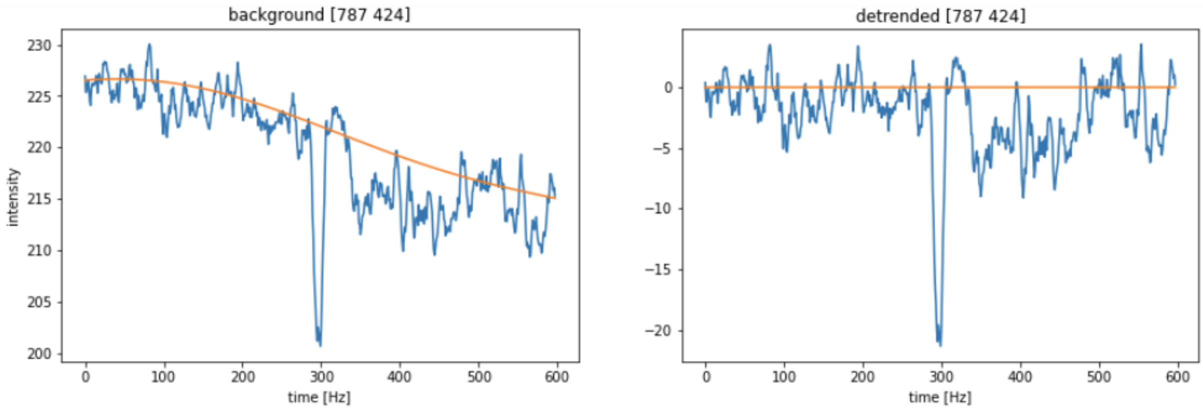


Figure 4.15: Reference pixel of **Zone 1**

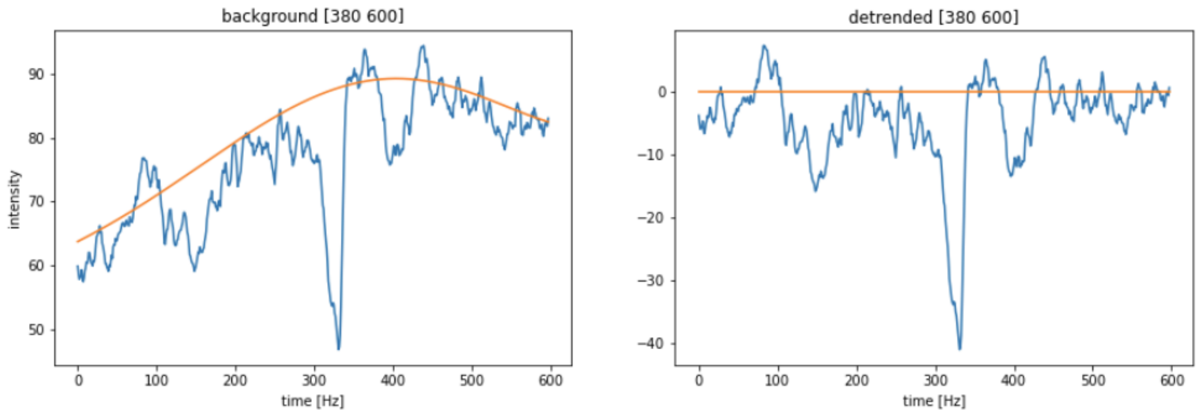


Figure 4.16: Reference pixel of **Zone 2**

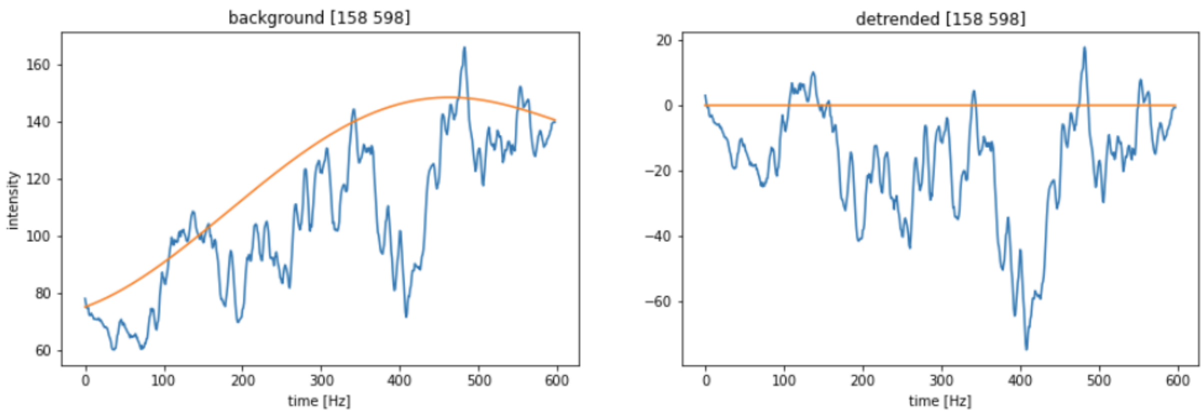


Figure 4.17: Reference pixel of **Zone 3**

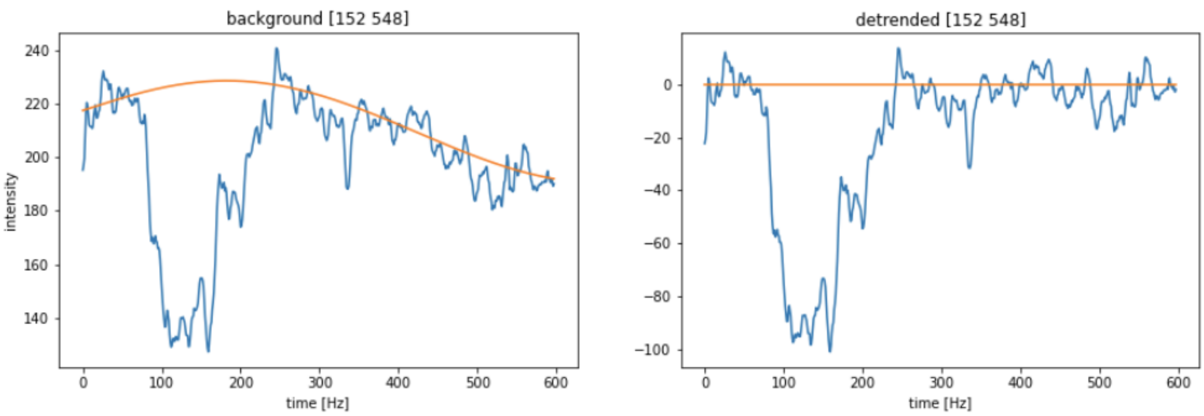


Figure 4.18: Reference pixel of **Zone 4**

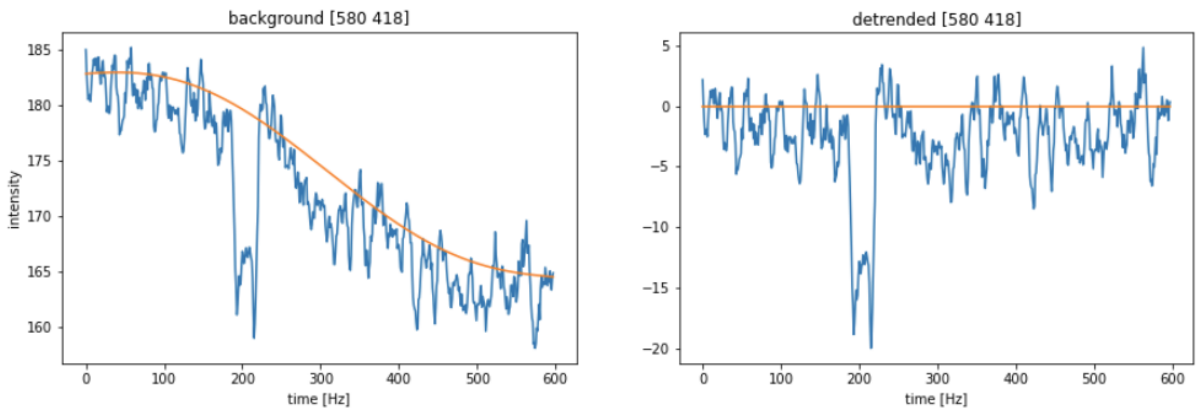


Figure 4.19: Reference pixel of **Zone 5**

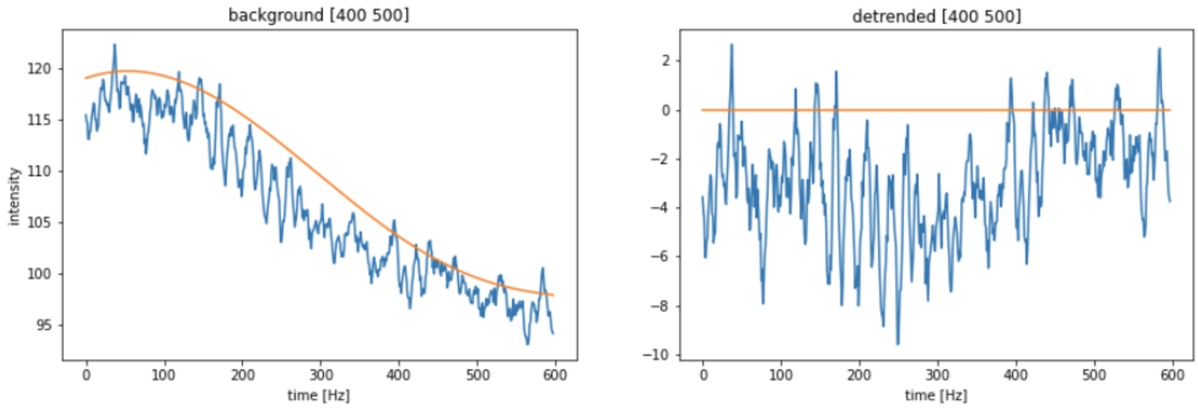


Figure 4.20: Pixel with oxygen activity but no signal dip

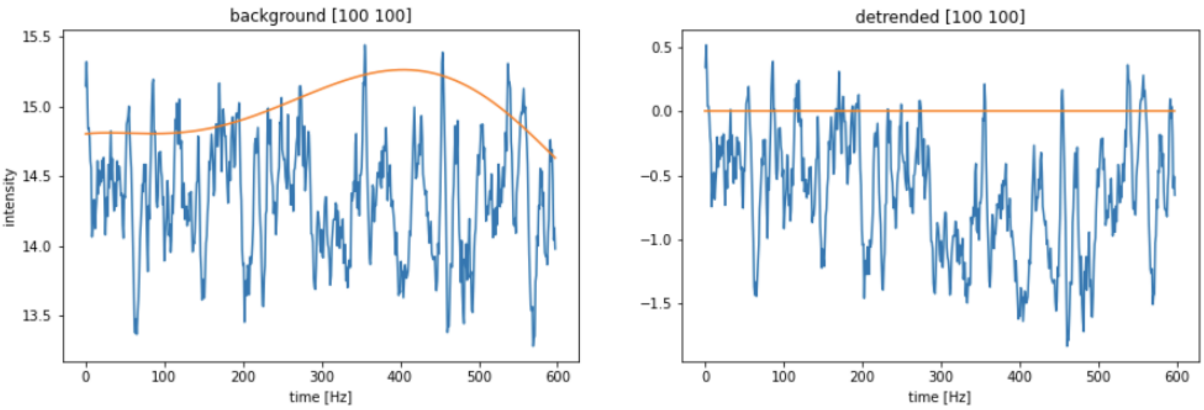


Figure 4.21: Pixel outside of oxygen activity where just noise is expected

We see that the robust detrending shown at the reference pixels works as hoped for the most part. The polynomials do not try to converge to long-lasting dips, as we see in figure 4.18. Without robustness, the polynomial has always approached the dip in past experiments, which has also negatively affected the rest of the trend estimation. It can also be observed in each figure that the trend tends to stick to the upper signal peaks. This forces assumed signal dips more into the negative value range, making them easier to identify for further analysis. In addition, we can observe that for the noise regions (see figure 4.20 and 4.21) the trend also tends to stick to the top. This forces the noise into negative territory, which is an undesirable side effect of our detrending method.

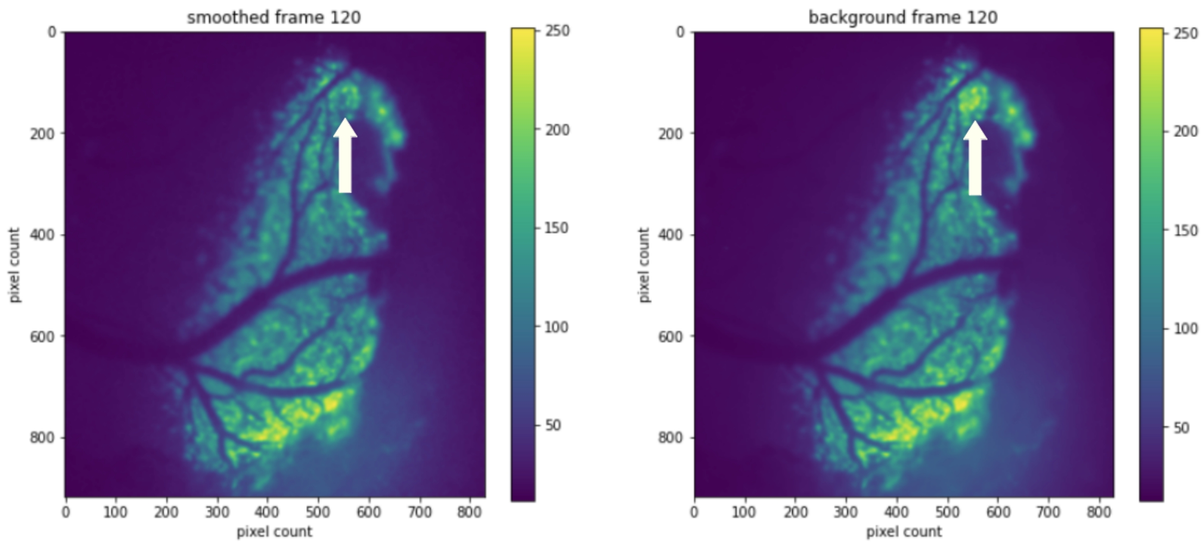


Figure 4.22: On the left is the frame 120 of the smoothed video. On the right is the same frame of the assumed background video where zone 4 (see figure 4.5) is much brighter.

After the background video has been subtracted from the smoothed video (we call it signal video), the signals should also become more visible on the spatial level, since the signal should now be close to 0 for regions without a signal dip. We now look at this on selected frames where the signal dips of zone 1 - 5 (see figures 4.2 - 4.6) occur and try to identify them spatially. The frame identification can be seen in the title of the respective figures. In addition, the zones (see figure 4.1) are marked where their signal dips occur and are accordingly relatively visible.

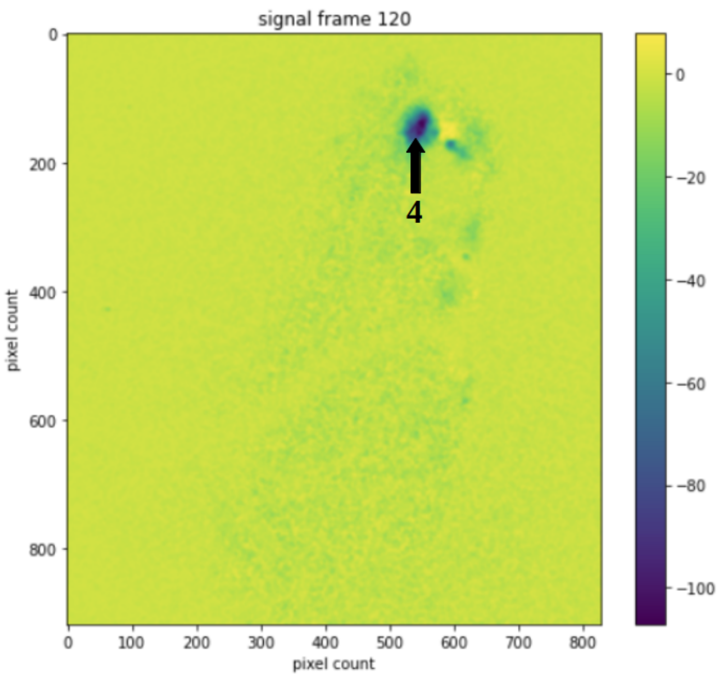


Figure 4.23: The signal dip of zone 4 (see 4.5) can be seen very strongly at frame 120.

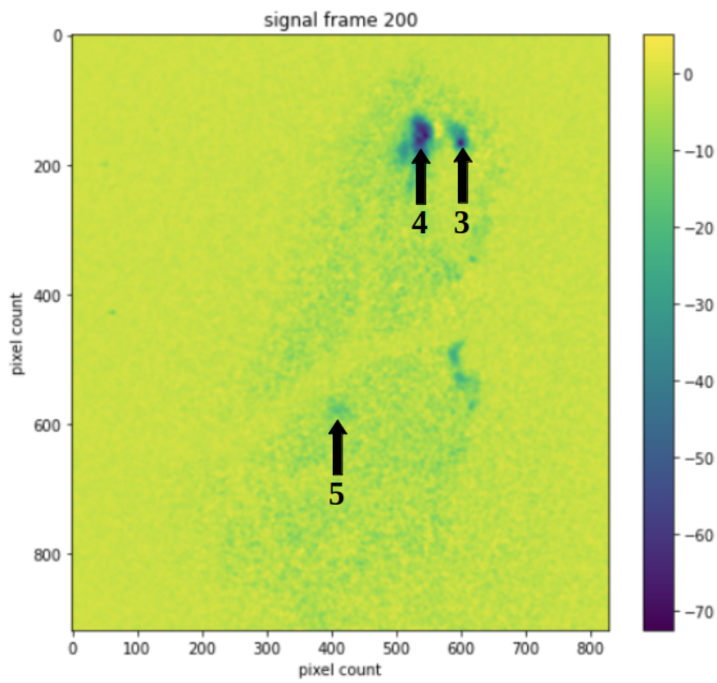


Figure 4.24: Here you can see the signal video in frame 200. Zone 3 (see figure 4.4) and zone 4 (see figure 4.5) are clearly visible. Zone 5 (see figure 4.6) also has a signal dip here. However, this is very weak relative to zones 3 and 4 in terms of intensity, which is reflected in the visibility of this image.

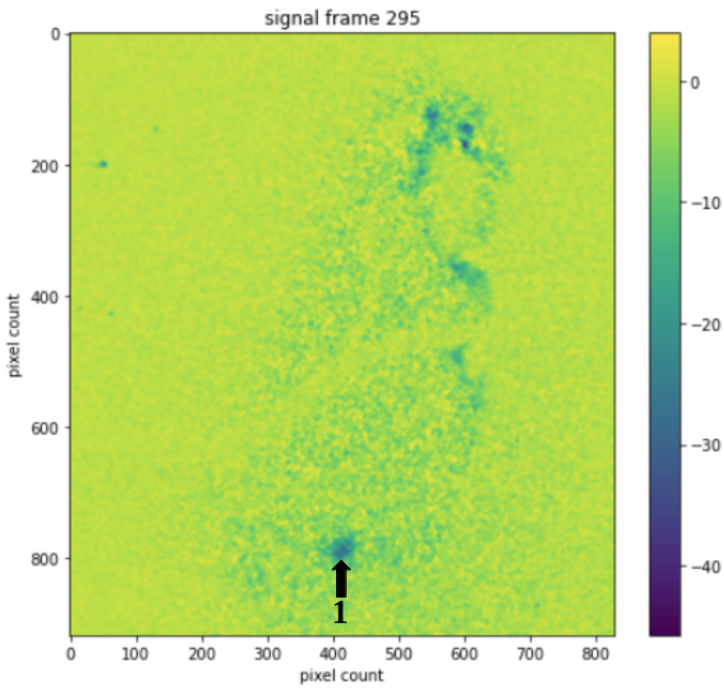


Figure 4.25: In frame 295 the dip of zone 1 (see figure 4.2) is visible. However, compared to the dip of zone 5 (see figure 4.24), this dip is relatively weak.

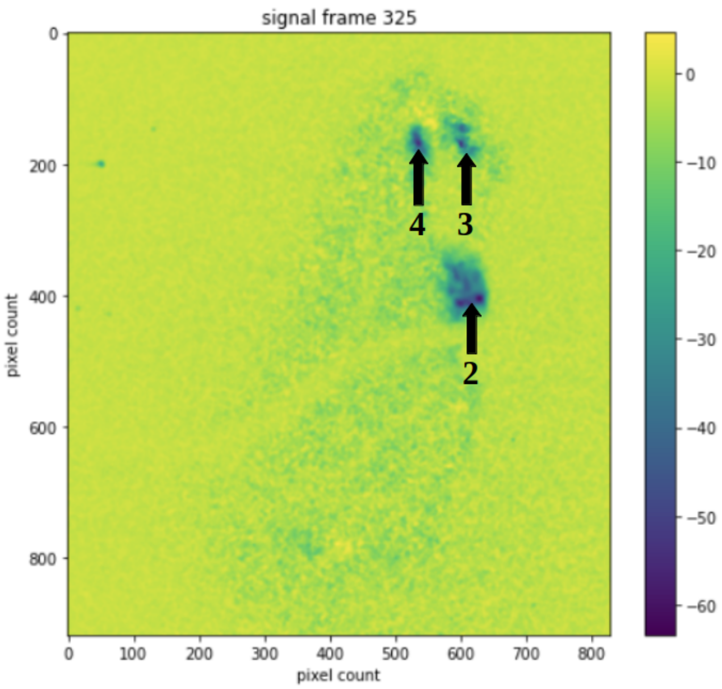


Figure 4.26: In this frame we see the signal dip of zone 2 (see figure 4.3) relatively pronounced. In addition, zones 3 (see figure 4.4) and 4 (see figure 4.5) are still visible, although they are not in focus in this frame.

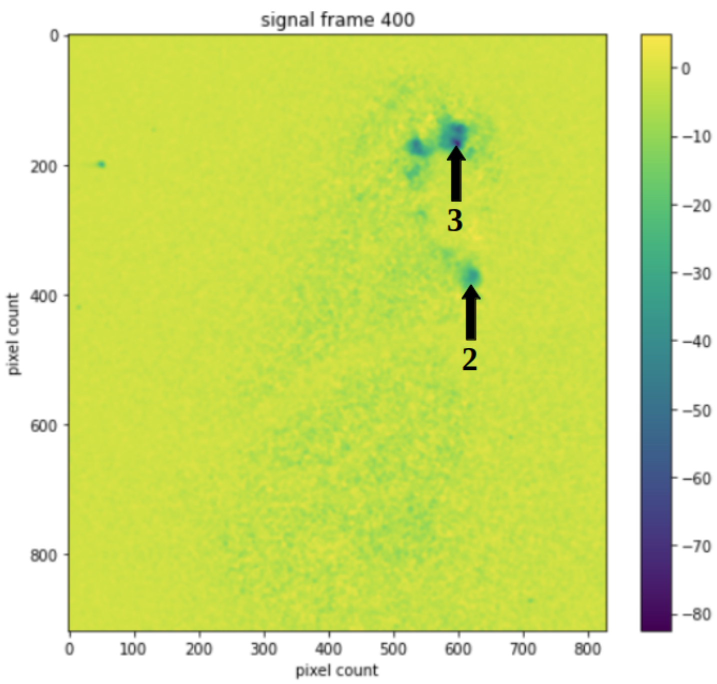


Figure 4.27: In frame 400 we have two dips that stand out. On the one hand, we have zone 3 (see figure 4.4), where the dip reaches its low, and zone 4 (see figure 4.5), where the last significant signal dip can be seen.

We have observed that the signal video extracts the assumed dips and we can use this for further algorithms. However, weaker dips in terms of signal intensity (see figures 4.24, 4.25 and 4.27) have difficulty reaching a comparable significance level as the larger dips. This can make their detection very difficult. We think that for relatively strong signal dips this is probably sufficient preprocessing and we can use it to segment the dip events.

4.4 Signal Amplification

As discussed in chapter 3.4, we have regions like zone 1 (see figure 4.2) and zone 5 (see figure 4.6) which are very weak in terms of signal intensity as well as temporal duration. Since this massively complicates their identification (see chapter 6), we try to bring the signal dips to a more comparable level using the method described in chapter 3.5 for each spatial pixel. We call the resulting video "amplified video". For the noise level estimation, we chose a window size of 20 for the block minimas. This size seemed reasonable considering the fluctuations of the reference pixels. For the verification of the noise level estimation and what the amplification did, we look at the reference pixels considered so far (see figures 4.28 - 4.34). On the left is the noise level estimation depicted using an orange line whereas on the right is the amplified signal.

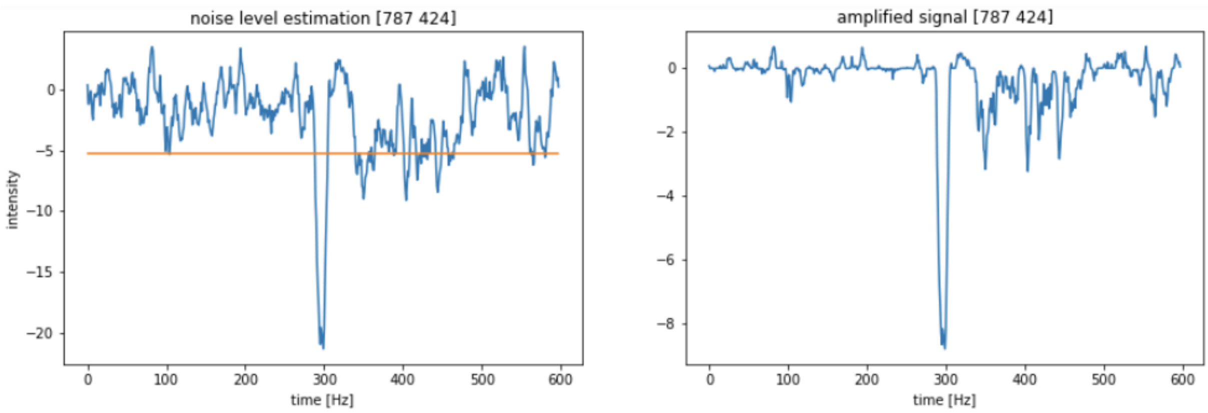


Figure 4.28: Reference pixel of **Zone 1**

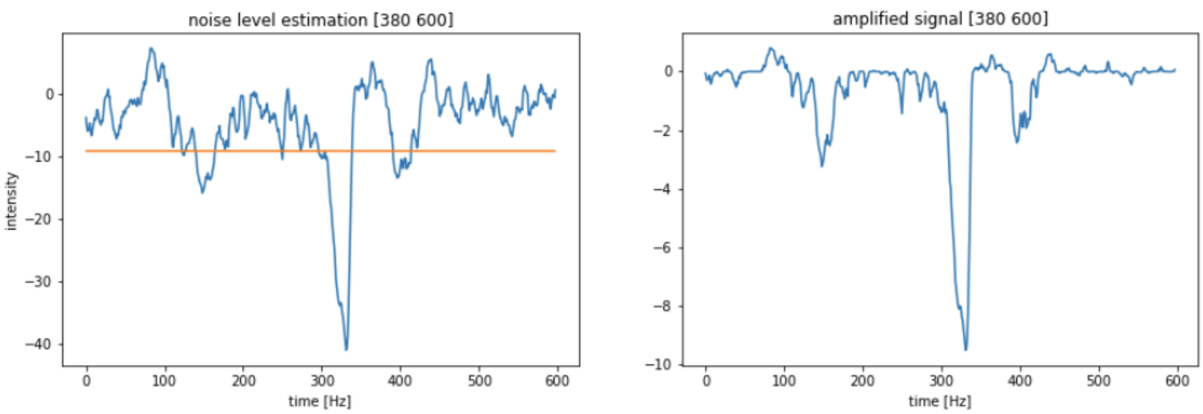


Figure 4.29: Reference pixel of **Zone 2**

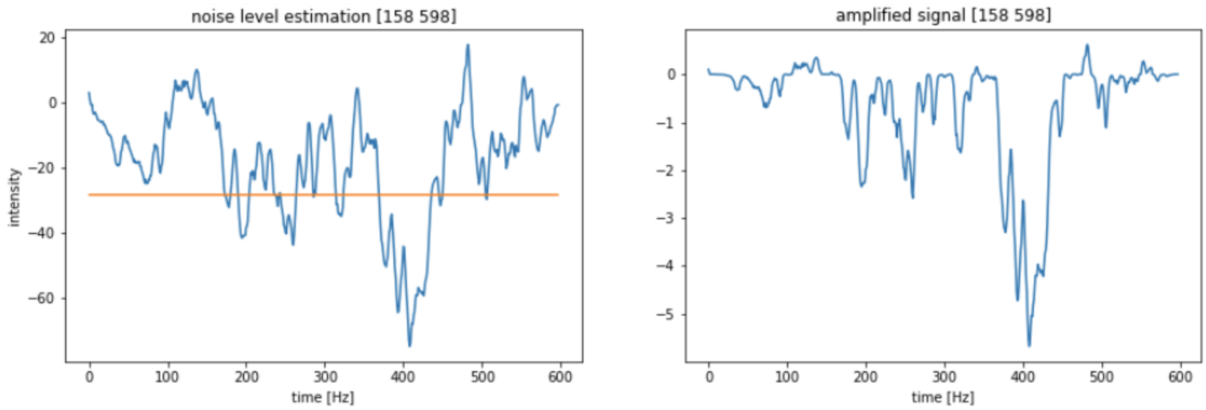


Figure 4.30: Reference pixel of **Zone 3**

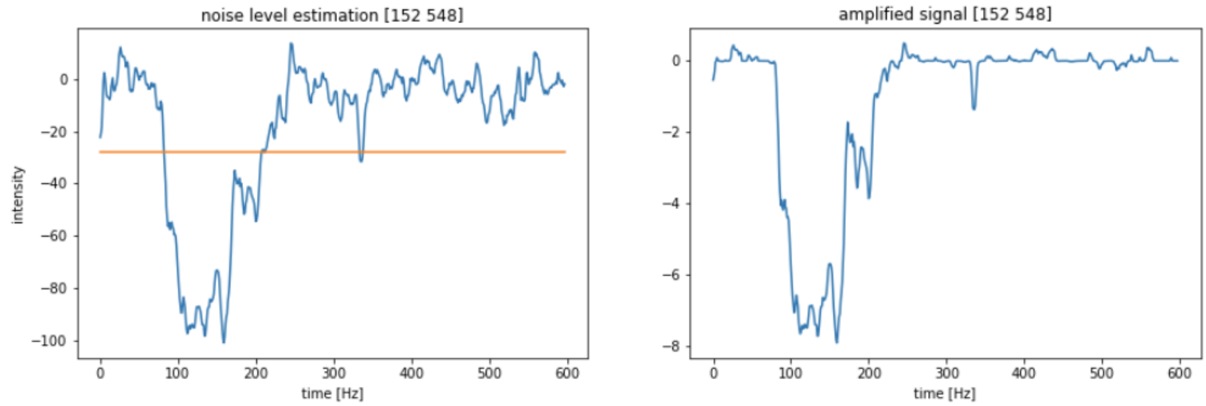


Figure 4.31: Reference pixel of **Zone 4**

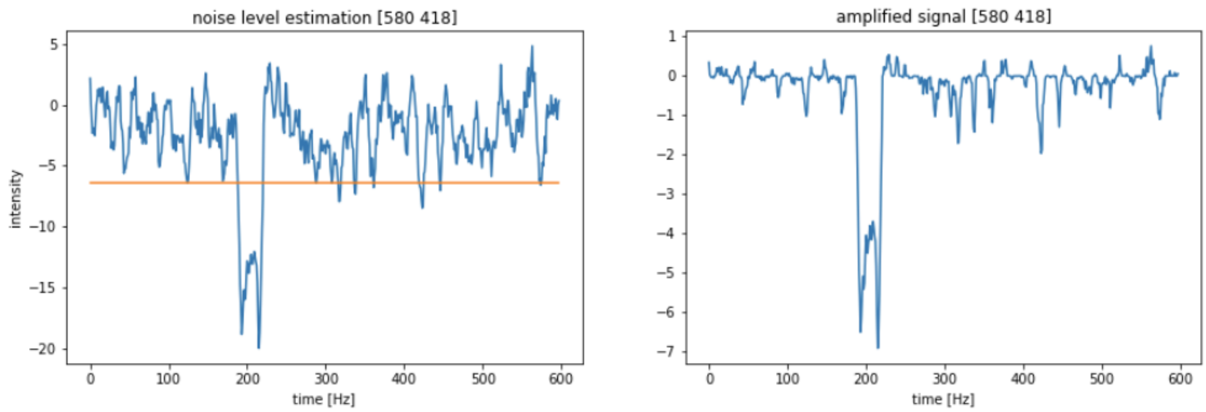


Figure 4.32: Reference pixel of **Zone 5**

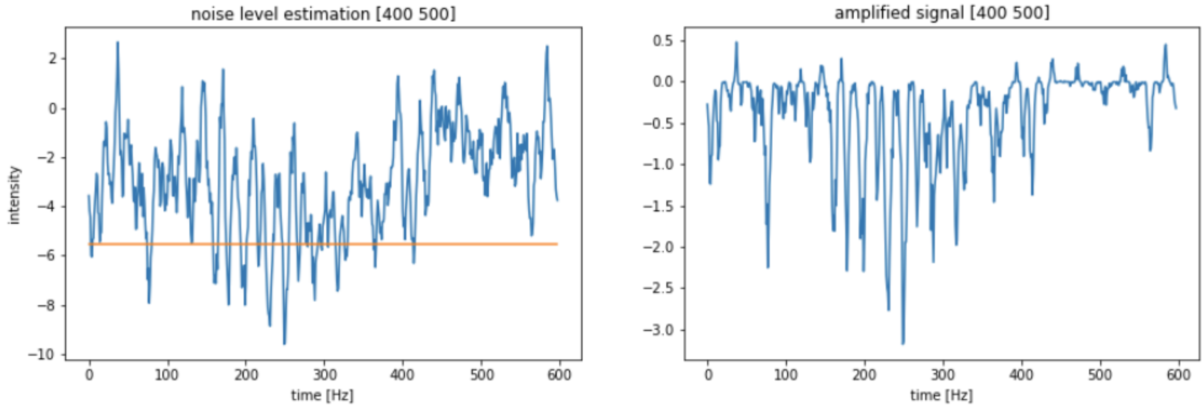


Figure 4.33: Pixel with oxygen activity but no signal dip

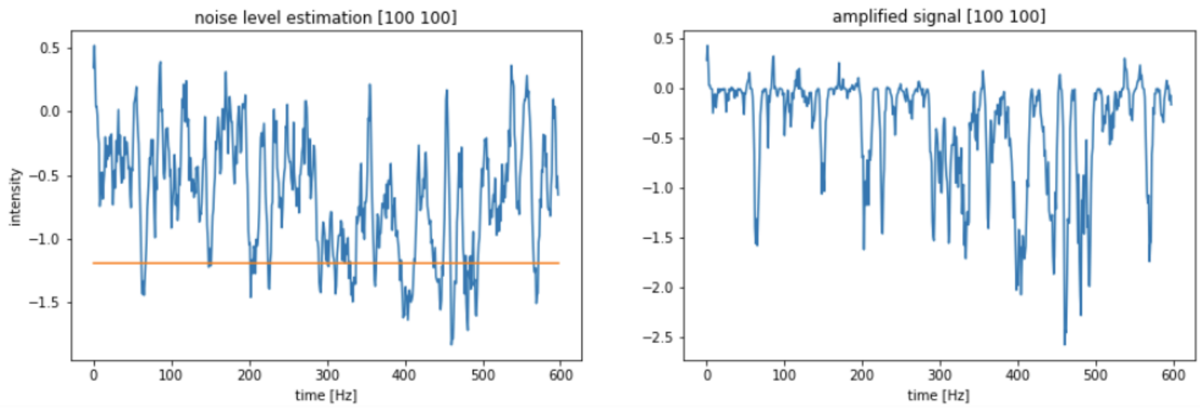


Figure 4.34: Pixel outside of oxygen activity where just noise is expected

In all reference pixels of zones 1 - 5 we see that the signal amplification transforms the signal dips to a more comparable level. In figure 4.29 and 4.30 we see that for more active regions the noise level is estimated rather low, which only transforms the strongest dip as hoped. This is obviously due to the lower noise level estimation since these are pixels of very active regions. However, the reference pixels of zones 1 and 5 are the most interesting, as the intent of the method is precisely to amplify the signal of these relatively weak dips. For these regions, the method seems to work relatively well because the noise level estimation seems to estimate the level as intended. Finally, we have the two reference pixels at regions where we did not detect any signal dips and therefore only noise should be present. Here we can observe that the noise level estimation is not optimal and therefore amplifies the noise at a few locations in time, which introduces additional noise.

In figures 4.28 - 4.34, there are already some strengths and weaknesses of the method recognizable. The method seems to work relatively well as long as the noise level is well estimated. For the relatively weak dips (see figure 4.28 and 4.32) this turned out well. For the reference pixels which have stronger signal dips (see figure 4.29, 4.30 and 4.31) the dips were mainly transformed to a comparable level as the weaker dips because the noise level estimation turned out to be relatively large. Assuming for these three reference pixels the noise level would be much smaller, the amplify function 3.16 should probably be more attenuated for stronger signals, perhaps even explicitly constrained with a maximum. Otherwise, the signal level of stronger and weaker dips would get closer in signal intensity, but the difference would probably be still too large. In addition, one can see in figure 4.33 and 4.34 that this method introduces additional noise by amplifying some of its noise. In summary, we think that this method should work as intended, but depends heavily on the noise level estimation as well as the amplifier function. After this temporal analysis, we look at how the signal dips look spatially by looking at the same frames as in chapter 4.3 (see figures 4.35 - 4.39). On the left are the signal video frames and on the right are the corresponding amplified frames.

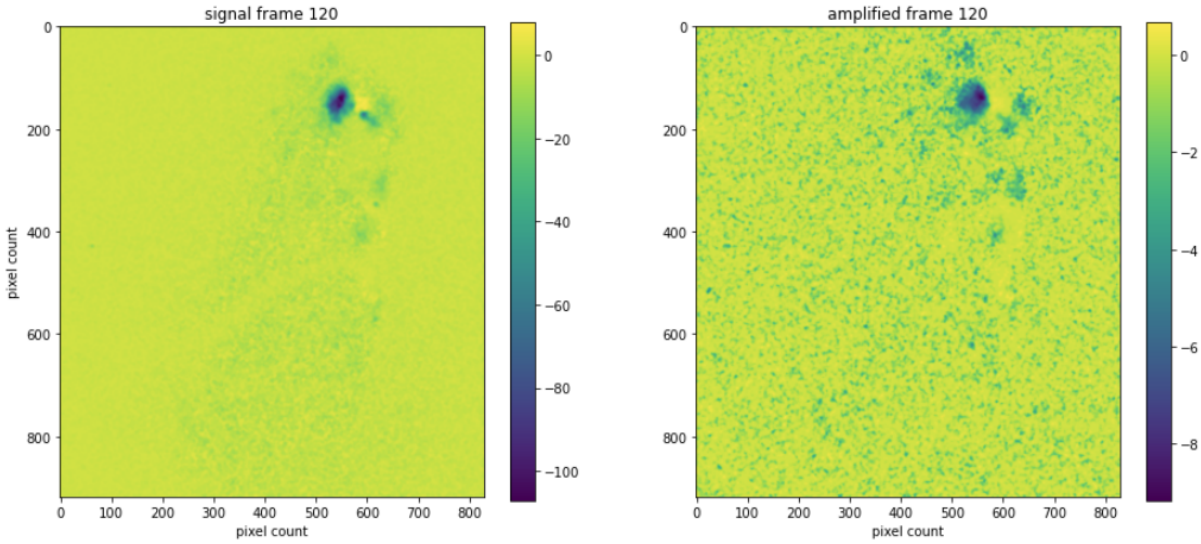


Figure 4.35: The signal dip of zone 4 (see 4.31) can be seen very strongly at frame 120 on both, the signal and amplified videos.

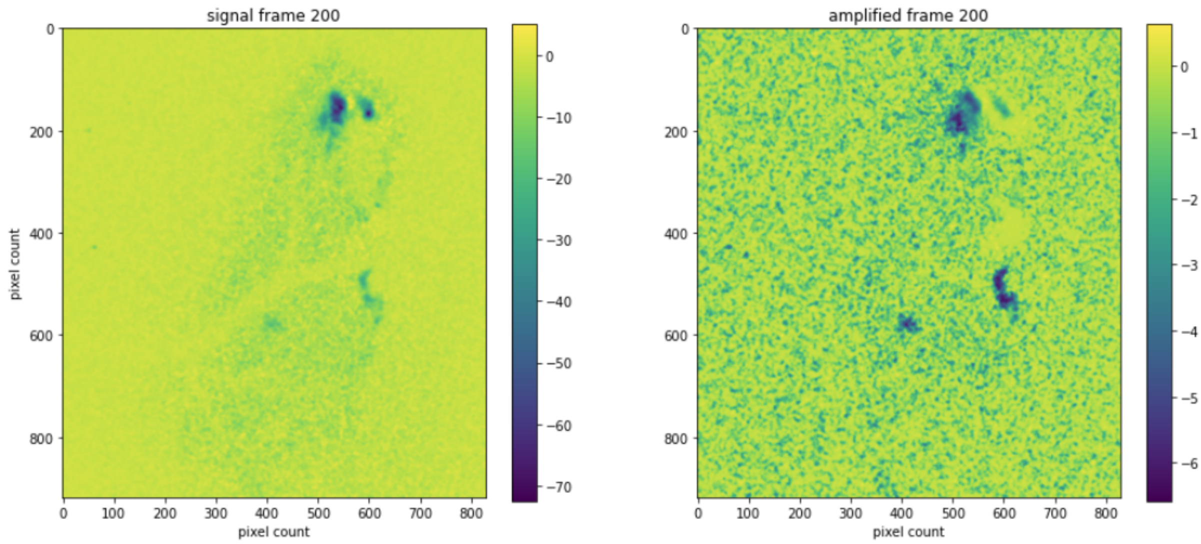


Figure 4.36: At frame 200, at the top of the signal frame, zone 3 (see figure 4.30) and zone 4 (see figure 4.31) are clearly visible. However, zone 3 seems to vanish on the amplified frame. On the other hand, zone 5 (see figure 4.32) at [580,418] (left, bottom) gets much more visibility on the amplified frame compared to the signal frame.

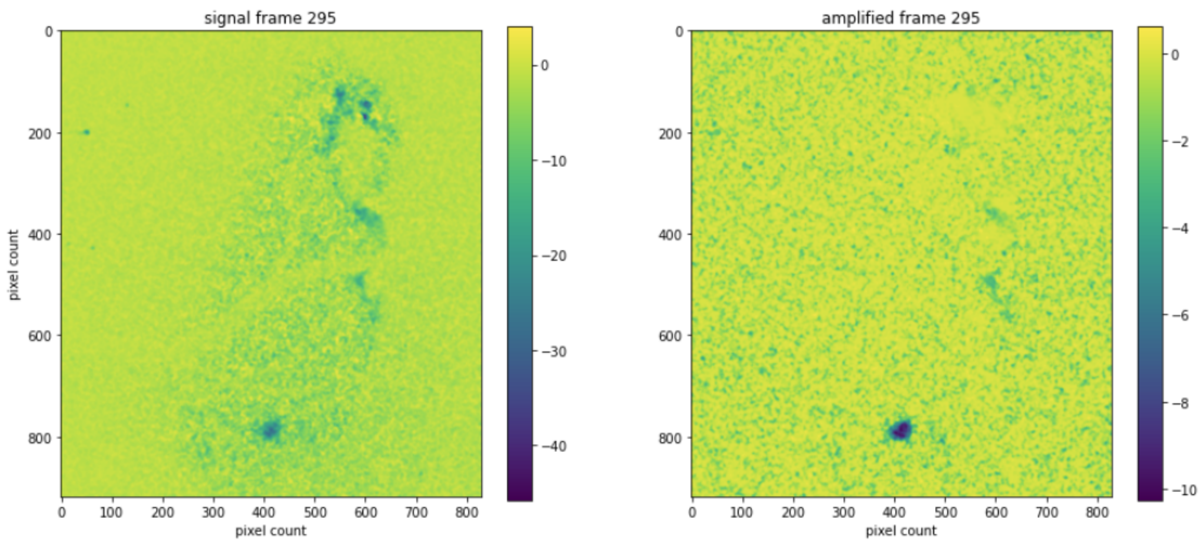


Figure 4.37: Here at frame 295, the dip of zone 1 (see figure 4.28) of the signal frame is visible, but with low intensity compared to stronger dips. On the right figure, however, the dip of zone 1 has now a comparable signal level.

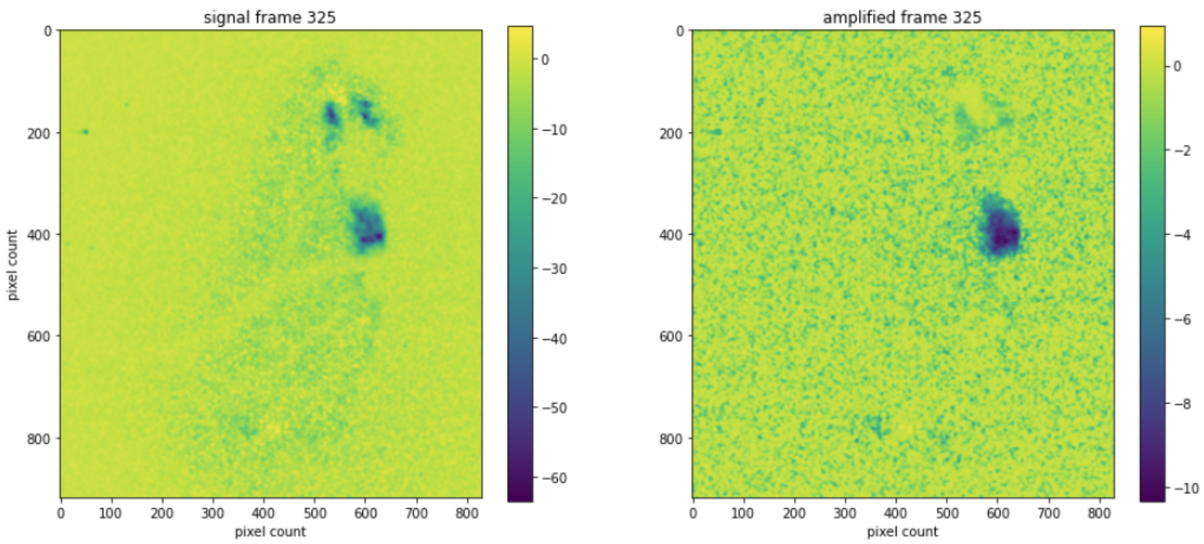


Figure 4.38: On the amplified frame, the dip of zone 3 (see figure 4.30) and zone 4 (see figure 4.31) vanish as expected based on the reference pixels. The spatially largest dip of zone 2 (see figure 4.29 remained.

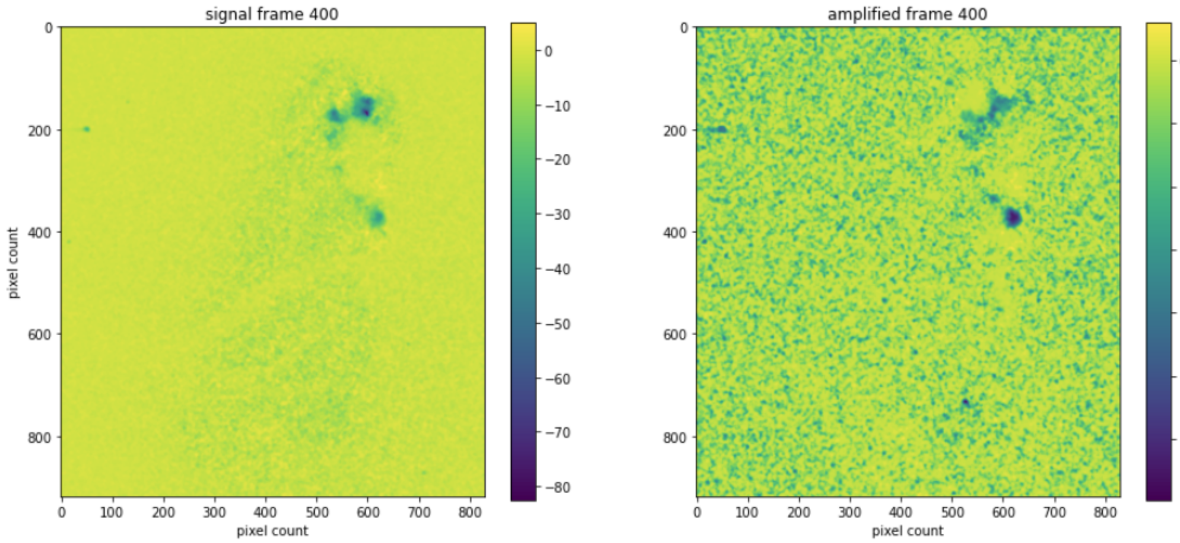


Figure 4.39: Frame 400 is interesting with respect to zone 2 (see figure 4.29) and zone 3 (see figure 4.30). The second signal dip from zone 2 is visible in a small spot on [370,610]. This is probably because we declared zone 2 to be relatively large (see figure 4.1), but the second event at frame 400 is effectively not the size of zone 2 at all and therefore hurt the dip intensity of the second event. Interesting is also zone 3, which does not disappear completely, but with a value range of -4 to -5 turns out relatively small, although the signal dip has reached the low point approximately in this frame.

The spatial analysis has shown that the signal level of the amplified dips has approached a comparable level for the most part and seems to have some consistency in space. However, it can be observed that noise is partially amplified relative to the signal as expected from the temporal analysis. Thus, the signal dips of different intensity scales tend to become more comparable, but the signal to noise seems to get worse. As shown in the temporal analysis and now visible in the spatial analysis (see figure 4.39), the signal dips partially disappear where the noise level estimation was estimated to be large. In a further step, it is probably useful to increase the signal to noise using mean or Gaussian filtering, so that further algorithms have fewer problems with the noise.

5 Semi-Nonnegative Matrix Factorization

In this chapter, we first discuss the choice of alpha for factorization and then analyse how our matrix factorization deals with overlapping ROIs, since this was one reason for choosing the matrix factorization method. At the end, we look at why certain dips found have multiple ROIs and how they are related to each other.

5.1 Selecting Alpha

To quickly compare different alphas, a simple class is implemented in our framework to do exactly that. The optimization procedure used warm start, which means that the solution of the previous fit was used as initialization of the matrices. This we can do because a higher alpha leads to more regularization and thus a higher reconstruction error. Which means that more values of the positional matrix SP are pushed to 0 as alpha gets higher where only the most significant dip events per temporal vector should remain (see figure 5.1). In this chapter, we will shorten the video name from 2021__BLI-Baseline __KX/nm200109__A2__GFAP-GeNL__KX/01004__mcor.tif to *01004.tif*.

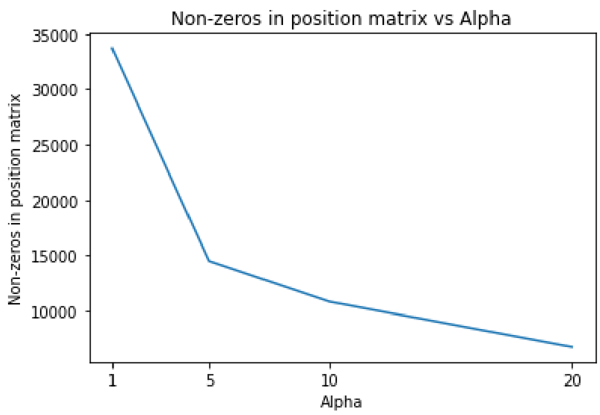


Figure 5.1: Number of non-zeros in the positional matrix per alpha of the factorization of the detrended and denoised video *01004.tif*.

Because we have less non-zeros, the reconstruction error is also higher, as seen in figure 5.2.

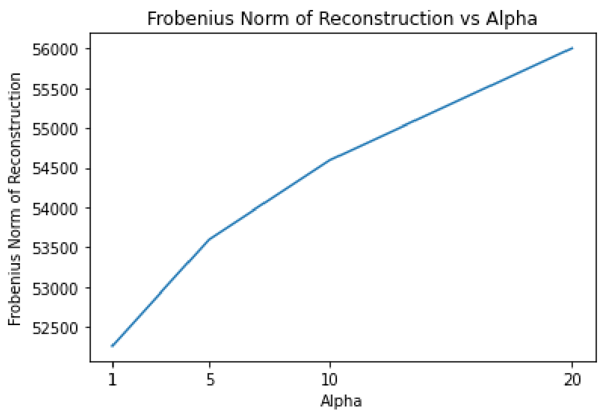


Figure 5.2: Frobenius norm of the reconstruction per alpha of the factorization of the detrended and denoised video *01004.tif*. We can observe that the reconstruction error increases when alpha increases.

In addition, we can also look at how the detected ROIs vary with a different alpha in figure 5.3.

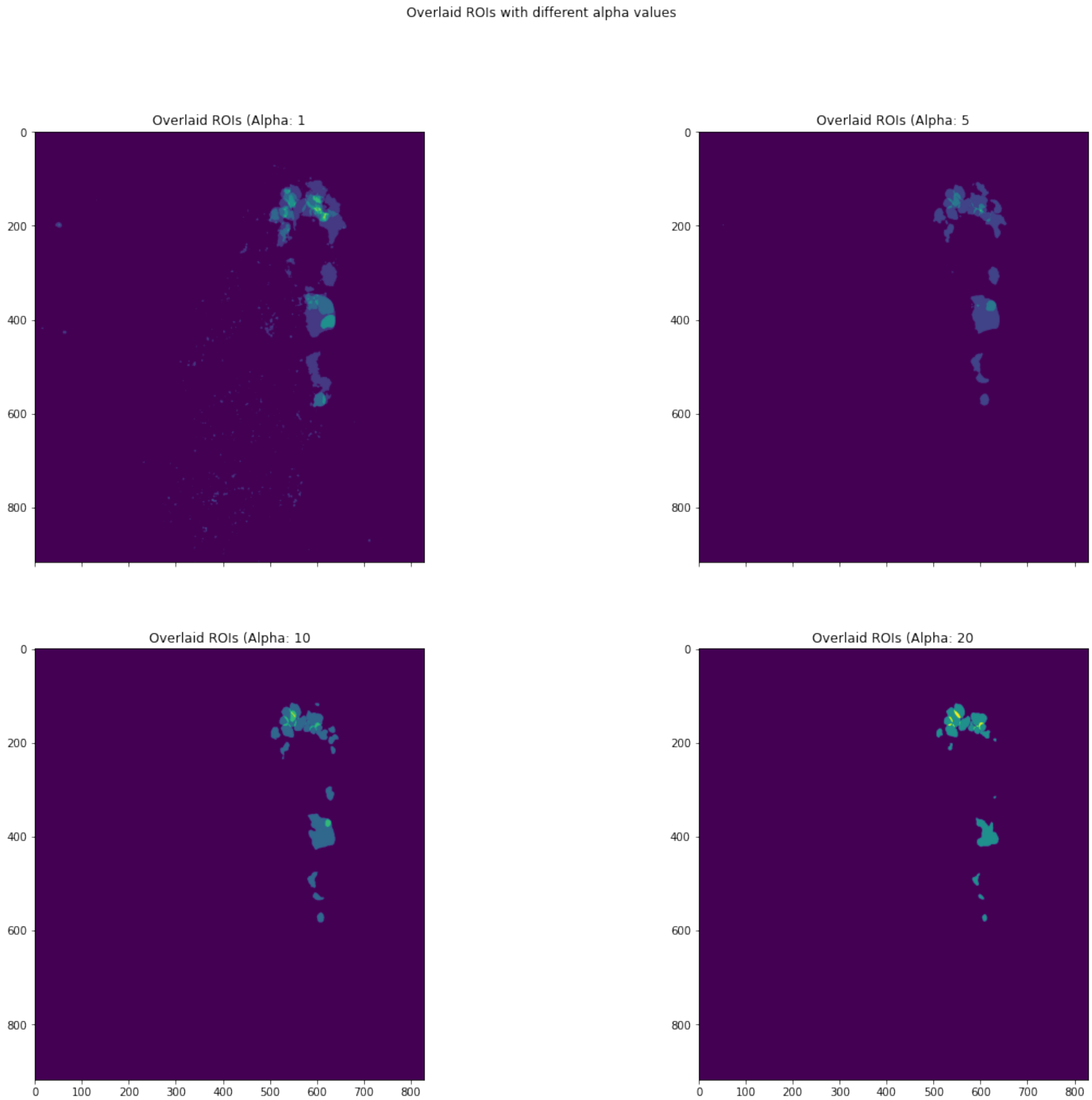


Figure 5.3: Overlaid predicted ROIs with different alphas on the matrix-factorized video *01004.tif*. A higher alpha leads to more regularization and thus to more zero pixels in the positional matrix. Also, brighter regions mean that there are overlapping ROIs because those figures are constructed by adding all positional, binary matrices together of the ROIs. Generally, we can observe that with a higher alpha the ROIs get more compact and less ROIs are found.

To quickly compare multiple alphas, we implemented the class *OptimizeSNNMF* in our framework, which returns the reconstruction error, the fitting time, the detected ROIs and the number of iterations of the algorithm. From this, a decision for the best alpha can be taken, especially when comparing the results visually, as shown in figure 5.4 vs figure 5.5.

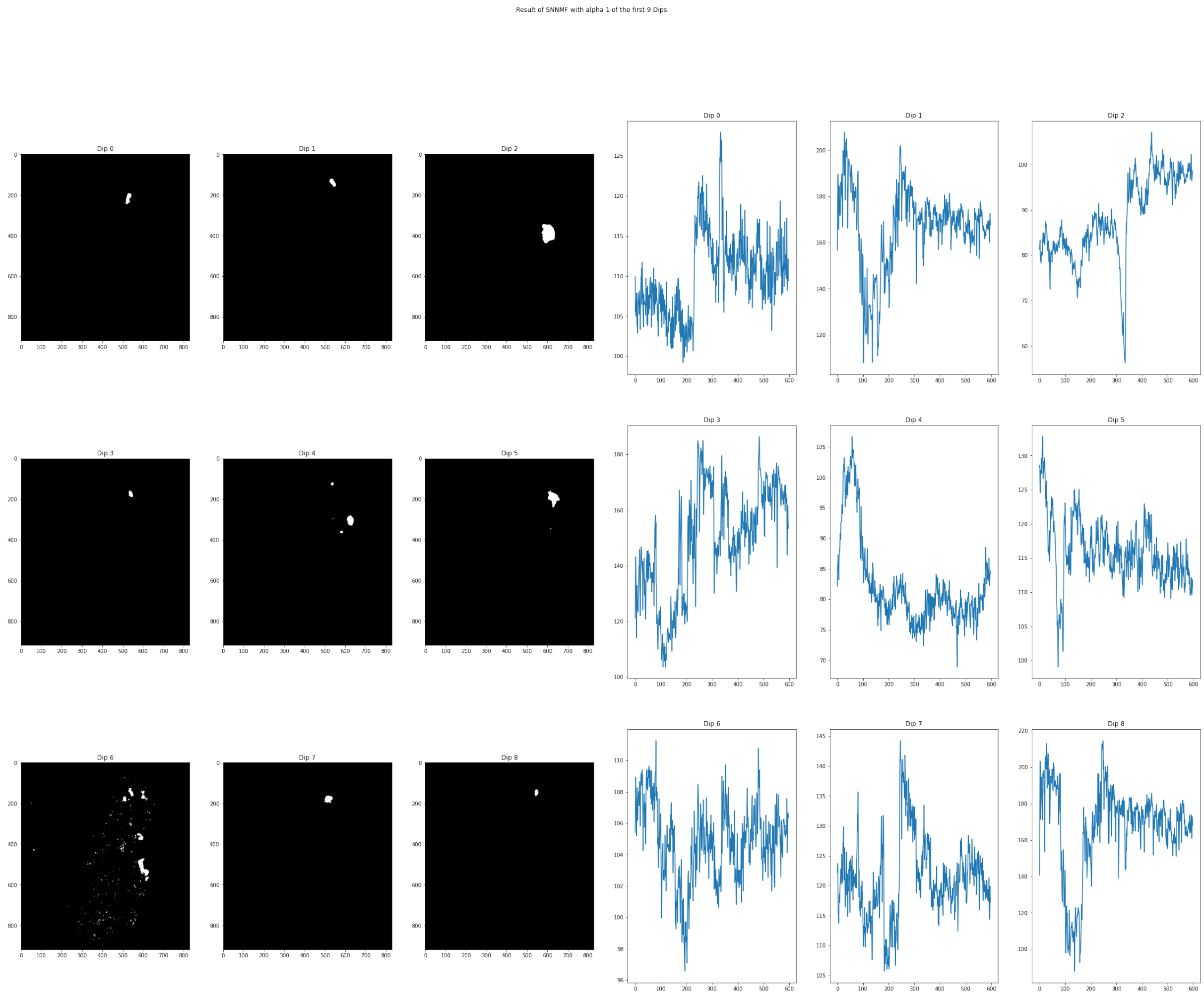


Figure 5.4: Different, selected ROIs with their extracted, unedited signal of the original video (*01004.tif*). Left we see the ROIs in the position matrix and on the right the spatial averaged signals of the corresponding ROI. As it can be seen, the alpha is probably too conservative because in Dip 6, we see that the ROI is spanning over the whole image and thus expected to not be an actual ROI.

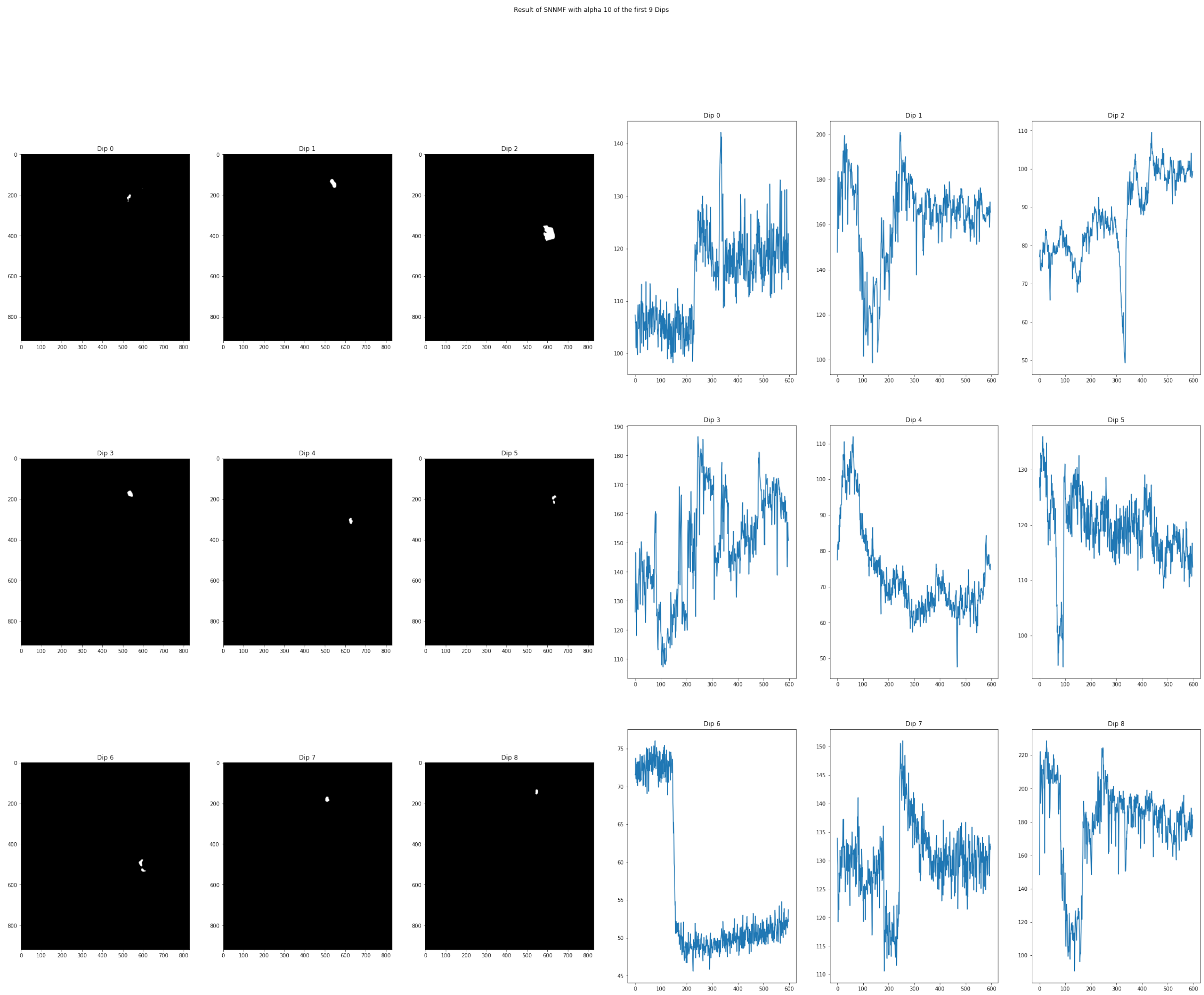


Figure 5.5: Different, selected ROIs with their extracted, unedited signal of the video mentioned in figure 5.4. Left we see the ROIs in the position matrix and on the right the signals of the corresponding ROI. This alpha seems to be better than the alpha chosen in figure 5.4 because no ROI is expected to span over the whole spatial region.

Currently, matrix factorization is slow, as seen in figure 5.6. Also, those numbers are with warm-start, which means that it uses the result of the previous alpha as initialization for the two matrices, so the numbers are skewed to the lower side.

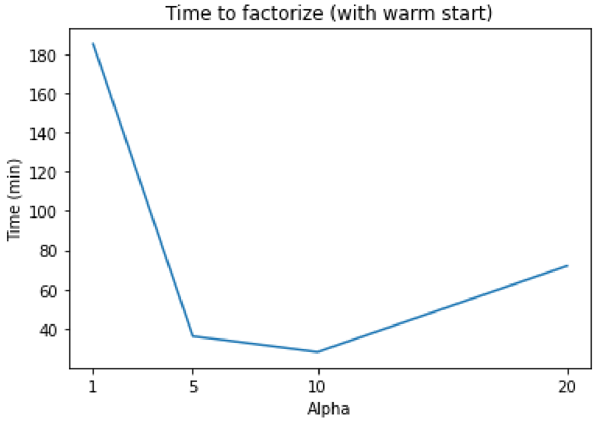


Figure 5.6: Time to factorize per alpha (with warm start) of the denoised and detrended video *01004.tif*, processed on an **Intel Core i7-9700KF (3.60GHz / 12MB).**

Our implementation *SNNMF* is slow. For LARS, there is an experimental implementation in cuML (Raschka et al., 2020), which could probably be expanded to make it useable for SNNMF.

5.2 Overlapping ROIs

A reason we decided for a matrix-factorization approach was that it is able to deal with spatially overlapping ROIs. When looking at a subset of *01004.tif*, which contains overlapping ROIs (as seen in figure 5.7), we see that the extracted time series for each ROI varies greatly.

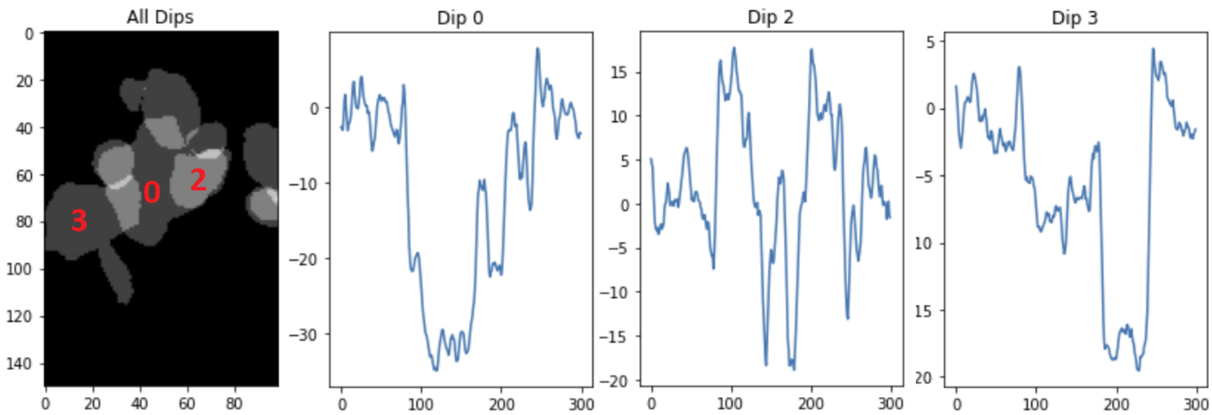


Figure 5.7: Extracted signals with a different signal intensity scaling of overlapping ROIs. As we can see, the extracted signals vary per ROI.

5.3 Multiple ROIs in a Single Prediction

Sometimes, we have multiple ROIs in one prediction, as seen in figure 5.8 of the factorization of *01004.tif*.

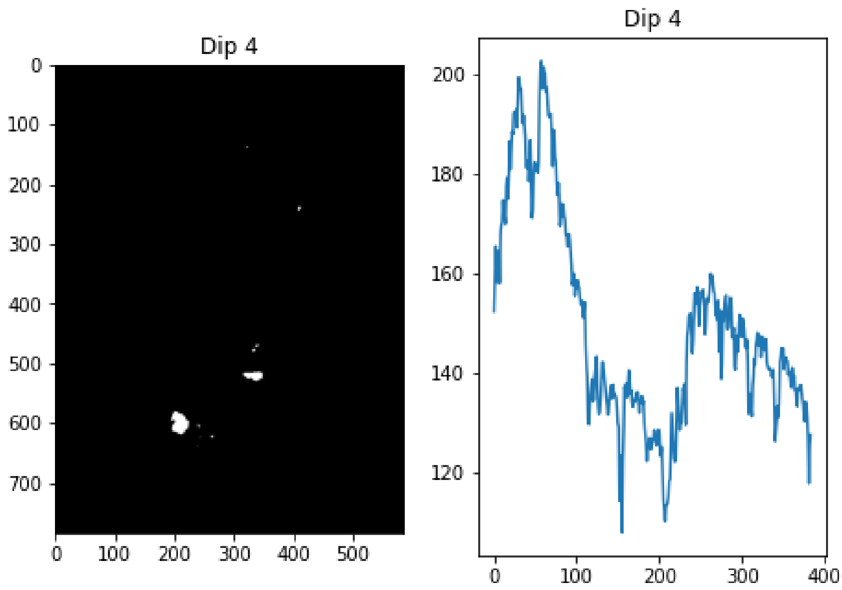


Figure 5.8: Multiple ROI per area with the corresponding to dip 4 of figure .9 found in the appendix. The left figure shows the segmented spatial ROIs where the right figure depicts the temporal in space averaged signal of both ROIs.

When we analyse those ROIs separately (see figure 5.9), we can see that the signals are fairly similar. When looking at the detrended and denoised signal, we can look at the Pearson correlation between them. The Pearson correlation is defined in equation 5.1, with σ being the standard deviation and cov being the covariance.

$$\rho = \frac{\text{cov}(X, Y)}{\sigma_x \sigma_y} \tag{5.1}$$

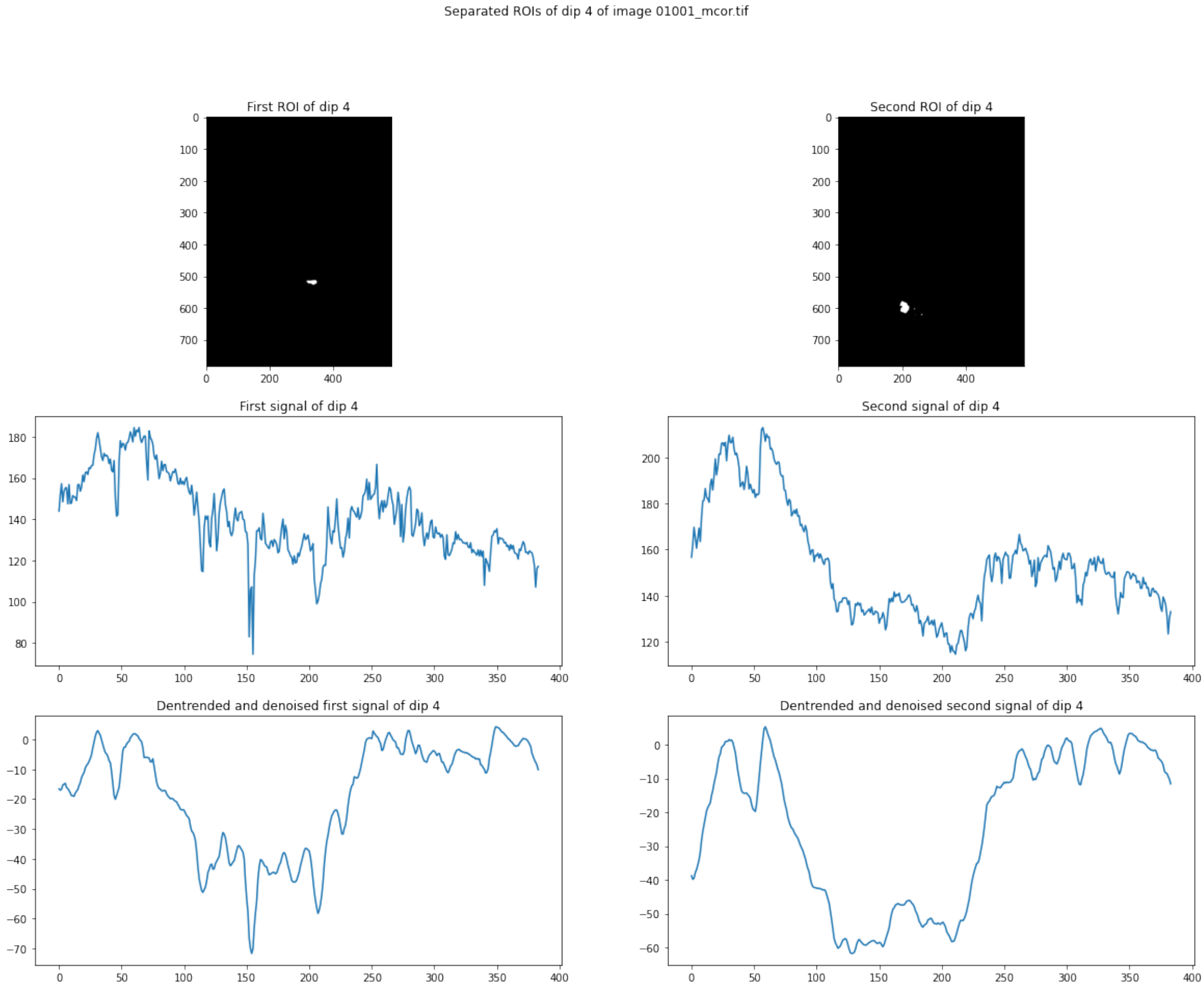


Figure 5.9: On the top are the in space averaged signals of each individual spatial region from figure 5.8 of the original video where on the bottom are the spatial averaged signals of the detrended and filtered video. Between the unprocessed signals we reach a correlation of 0.786 and between the detrended and filtered signals we have a correlation of 0.928.

We see that two spatial independent ROIs described by the matrix decomposition with a temporal vector are very similar in reality, especially after preprocessing. From this we can assume that the spatially different regions associated to the same signal vector by the matrix decomposition have a similar behavior and thus the regions mapped to them are probably actual ROIs. In case multiple predicted ROIs per area pose a problem, we suggest a clustering algorithm of choice, e.g. DBSCAN (Ester et al., 1996). However, we feel that this generally could also be of great interest to the scientists from Copenhagen.

6 Results

6.1 Quantitative Analysis

The quantitative evaluation is based on comparing manually marked ROIs vs predicted ROIs. Because our system can deal with overlapping ROIs and the manually labelled data not, we combine all ROIs together and evaluate them then. Generally, the ground truth contains ROIs that should clearly not be missed but do not include all ROIs. Thus, recall should always be high but precision can vary greatly with varying alpha. However, a low precision does not necessarily mean that our prediction failed but that the ground truth (manually labelled data) may miss those ROIs.

The results are per alpha and we start by showing the results of SNNMF on denoised and detrended videos, as mentioned in 4.2 and 4.3, and after we present the results of SNNMF on the amplified signals, as explained in subsection 3.5. First, we look at the video 2021_BLI-Baseline_KX/nm200423_A2_GFAP-GeNL/awake/01001_mcor.tiff
(shortened to 01001_mcor.tif, for which the complete factorization can be found in the appending 7.

Of course, there is no free lunch and we see in figure 6.1 that with an increasing alpha, the recall drops.

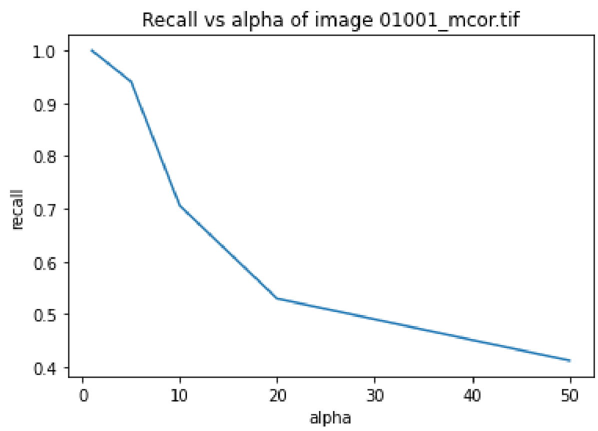


Figure 6.1: Alpha vs recall on the denoised and detrended video *01001_mcor.tiff* A higher alpha leads to a lower recall.

In table 6.1 we see the recall of the predicted ROIs vs the ground truth and in table 6.2 the pixel-wise coverage of our predicted ROIs vs the ground truth.

Table 6.1: Results ROI detection with alpha 50 on denoised and detrended 01001_mcor.tif.

recall
0.412

Table 6.2: Results ROI coverage alpha 50 on denoised and detrended 01001_mcor.tif.

precision	recall	f1-score	accuracy	IoU
0.315	0.450	0.370	0.964	0.709

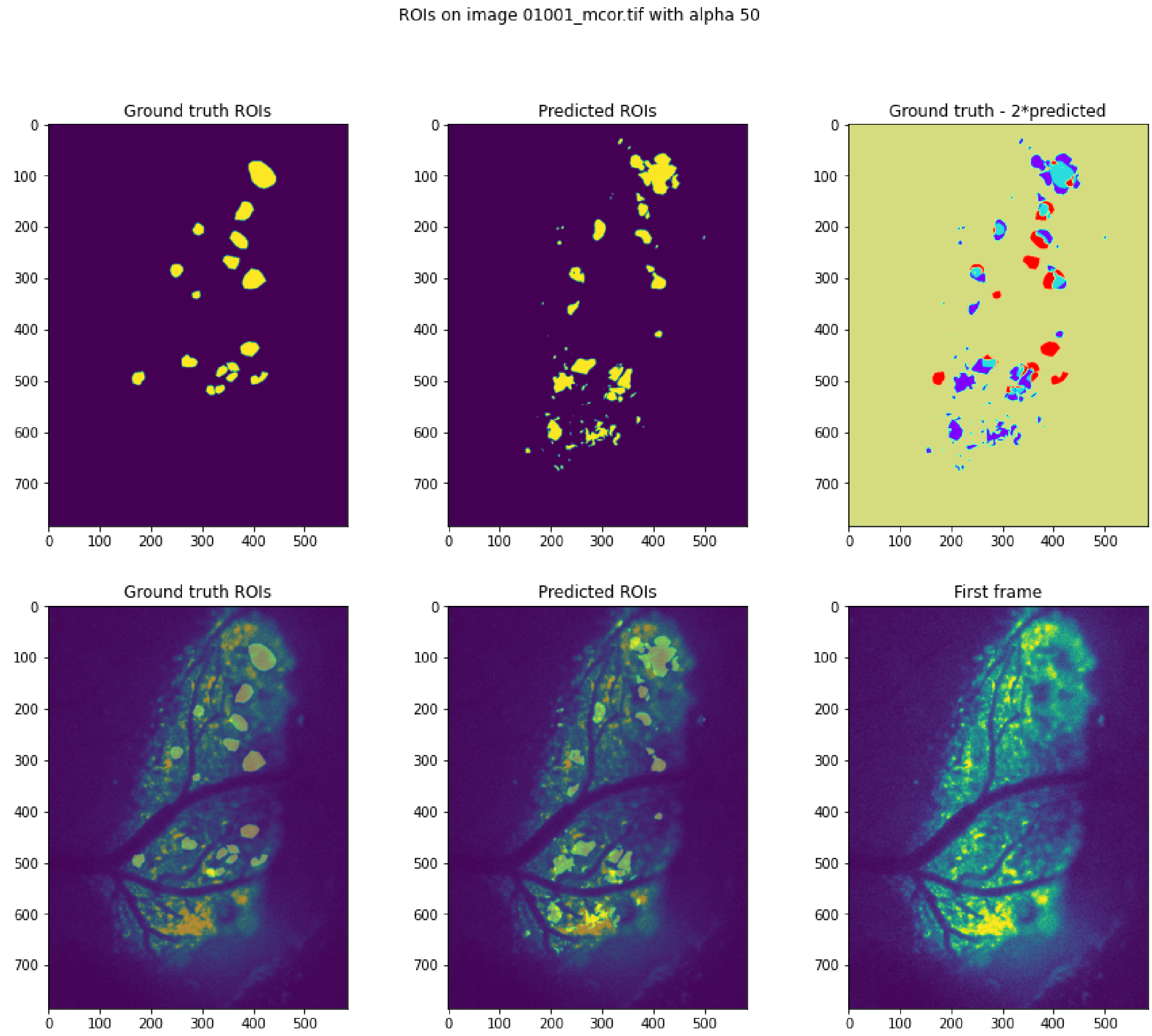


Figure 6.2: Predicted ROIs vs ground truth ROIs overlaid. So the violet regions are not recognized by the predictions and the violet regions are predicted areas with no underlying ground truth label. Thus, you can interpret the pixels in light-blue and yellow as TP and TN, the red as FN and the violet as FP.

In figure 6.3, we look closer at selected FP and FN ROIs.

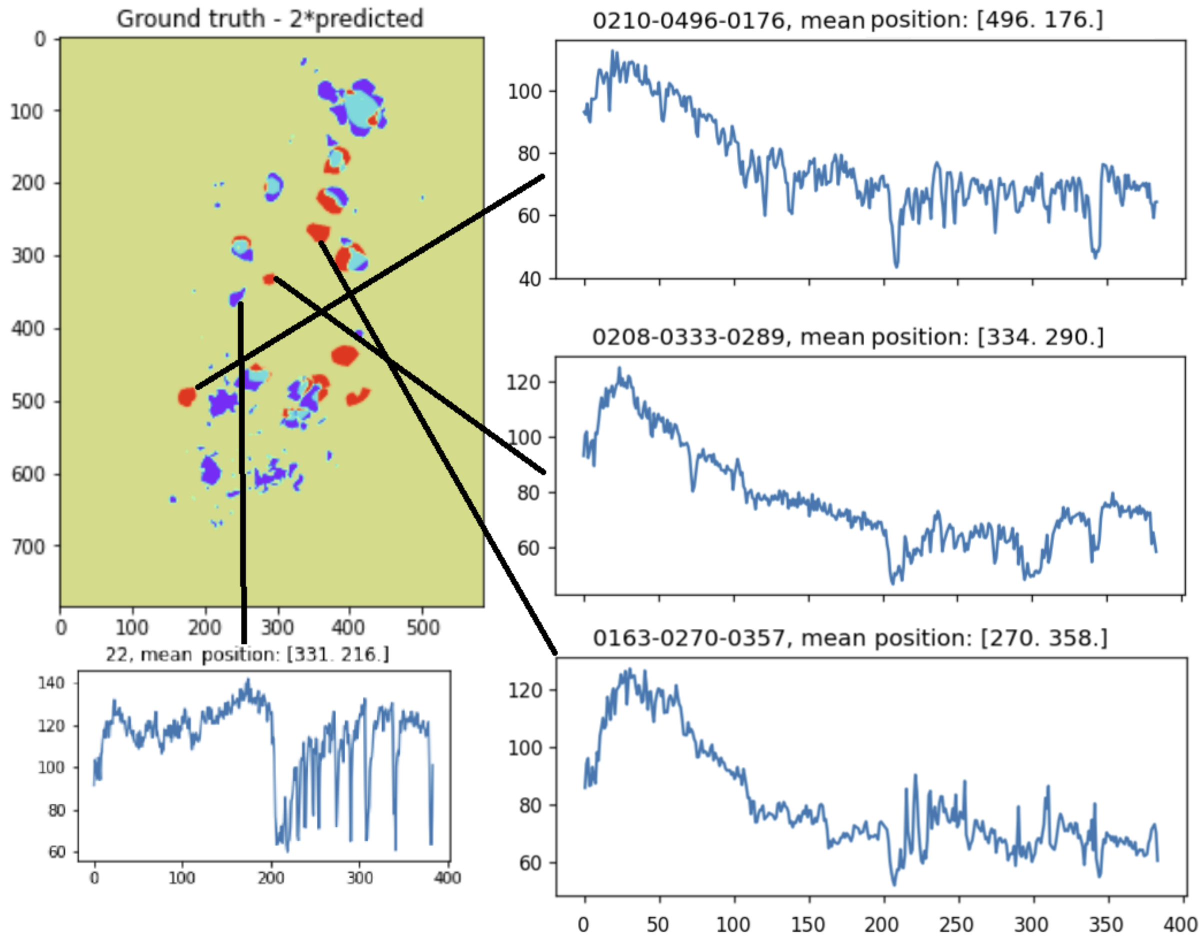


Figure 6.3: Predicted ROIs vs ground truth ROIs overlaid. So the violet regions are not recognized by the predictions and the violet regions are predicted areas with no underlying ground truth label. Thus, you can interpret the pixels in light-blue and yellow as TP and TN, the red as FN and the violet as FP.

Table 6.3: Results ROI detection with alpha 1 on amplified 01001_mcor.tif.

recall
0.118

Table 6.4: Results ROI coverage alpha 1 on amplified 01001_mcor.tif.

precision	recall	f1-score	accuracy	IoU
0.088	0.070	0.078	0.961	0.621

As we can see on the results, amplification of signals did not work perfectly. It requires careful tuning per video.

Now, let's look at *01004.tif* (shortened from 2021_BLI- Baseline_KX/nm200109_A2_GFAP-GeNL_KX/01004_mcor.tiff). In figure 6.4, we can see again our predictions vs the ground truth.

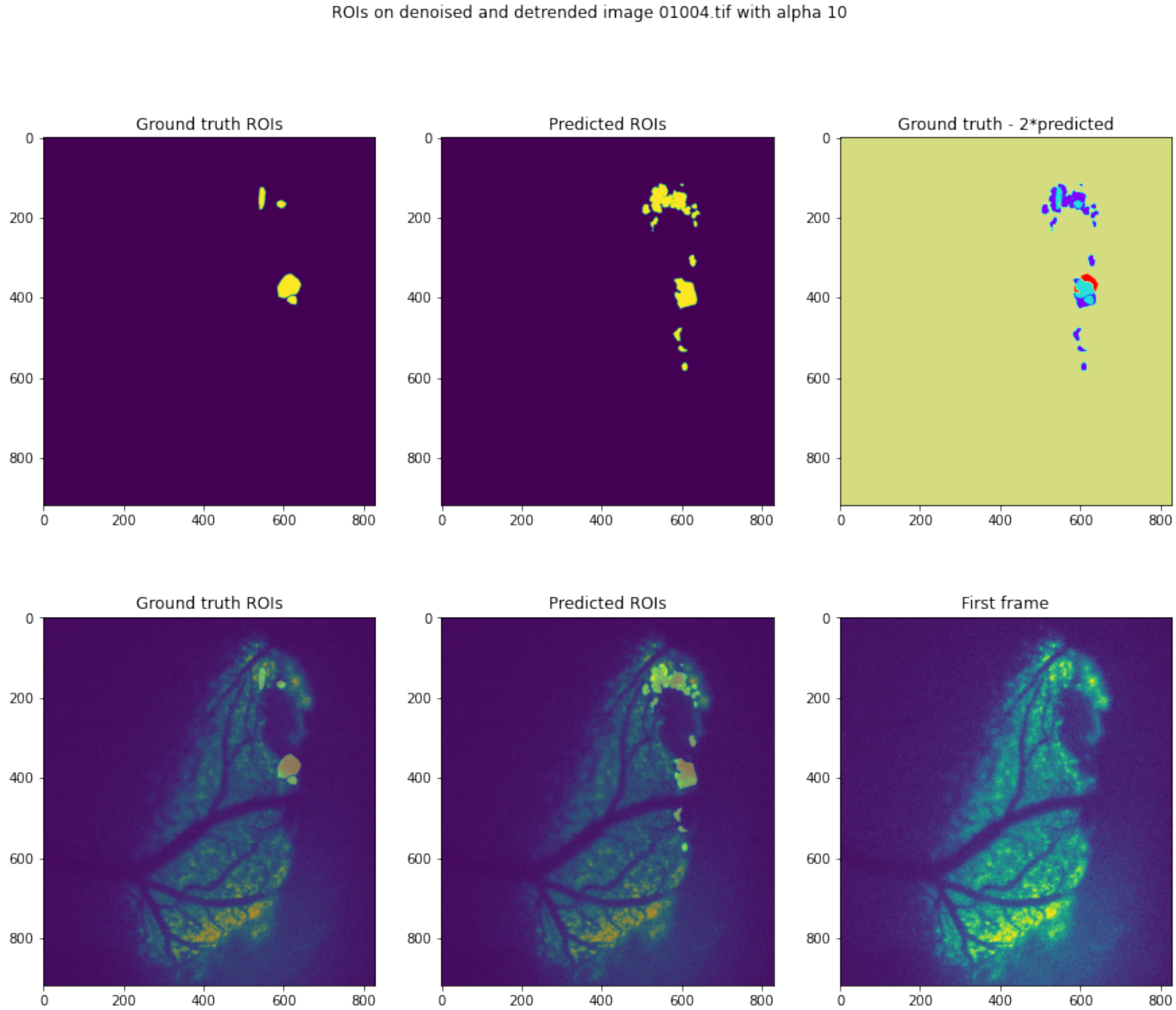


Figure 6.4: Evaluation of false positive and false negative ROIs. Red means FN and violet means FP. As we can see $SNNMF$ missed smaller and shorter dips (as seen in three pictures to the right). However, some large dips were not marked in the ground truth (seen in the bottom left picture).

Overall, it extracted the ROIs very well, at least based on the ground truth. This is also seen in the tables below in 6.5 and 6.6.

Table 6.5: Results ROI detection with alpha 10 on denoised and detrended 01004.tif.

recall
1.000

Table 6.6: Results ROI coverage alpha 10 on denoised and detrended 01004.tif.

precision	recall	f1-score	accuracy	IoU
0.314	0.772	0.446	0.990	0.494

For this video, our predictions match the ground truth very well. Now, lets look at how the result is on amplified signals.

Table 6.7: Results ROI detection with alpha 0.1 on amplified 01004.tif.

recall
1.000

Table 6.8: Results ROI coverage alpha 0.1 on amplified 01004.tif.

precision	recall	f1-score	accuracy	IoU
0.173	0.887	0.289	0.977	0.384

For this video, the amplification seems to work quiet good. When comparing the complete factorization of amplified vs non-amplified in the appendix at .11 and .12, we can see that the amplified version can find dips (for example, dip 7) which are missed by the non-amplified version.

The last video with ground truth, also called 01001_mcor.tif (2021_BLI-Baseline_KX/ nm200423_A2_GFAP-GeNL/awake/01001_mcor.tiff), will be referenced as 01001_mcor.tif (2) to avoid confusion.

Table 6.9: Results ROI detection with alpha 20 on denoised and detrended 01001_mcor.tif (2).

recall
0.875

Table 6.10: Results ROI coverage alpha 20 on denoised and detrended 01001_mcor.tif (2).

precision	recall	f1-score	accuracy	IoU
0.145	0.827	0.246	0.919	0.591

The matrix factorization approach worked well on this video.

Table 6.11: Results ROI detection with alpha 0.01 on amplified 01001_mcor.tif (2).

	recall
0	0.000

Table 6.12: Results ROI coverage alpha 0.01 on amplified 01001_mcor.tif (2).

	precision	recall	f1-score	accuracy	IoU
	0.108	0.011	0.020	0.983	0.187

For this video, amplified video failed. It shows that amplifying signals needs to be adapted to each video.

6.2 Qualitative Analysis

In this chapter, we examine the signals of the ground truth as well as those of the predictions on a descriptive level. First, we evaluate the ground truth and the predictions independently. Afterwards, we put the predictions in context with the ground truth and discuss the results. The ground truth signals and the according predictions are found in the appendix.

The ground truth dip events look mostly as expected, but there are some differences between the different ground truth videos. The ground truth signals of figure .6 are mostly as we expected a dip looks like. Typical manifestations of these are signals (0004-0461-0276, 0194-0302-0402, 0179-0205-0292, 0151-0519-0316) and a few others. These have a relatively sharp decrease in intensity followed by a relatively sharp increase in intensity. Furthermore, however, there are a few regions that we would not have expected to be labelled as ROIs in the ground truth such as (0210-0496-0176, 0208-0333-0289). The ground truths of the next video (see figure .7) are four regions, which we mainly discussed already in chapter 4.1 **ROI Analysis** and have corresponding clear dips. For ground truth ROIs, however, it seems to us that some ground truth ROIs may be missing. The ground truths of the next video (see figure .8) have very interesting characteristics. Signals like (0001-0376-0574, 1103-0443-0414, 0731-0662-0511, 1177-0616-0462) look more like peaks instead of dips, whereas these could often be characterized as a huge dip. In addition, there are characteristics here that are shared by most signals such as the steep signal rise after frame 1000 or a similar-looking signal drop from frame 0 to 100. There seems to be a lot of activity in this video. In general, the signals in this video look somehow difficult, which could lead to difficulties regarding ROI identification.

The predictions of the signal videos appear reliable for signal strong dips, whereas with the amplified videos weaker dips are detected, but this does not appear very reliable. Multiple, spatially independent predicted dip events will not be discussed in detail here, as we have already analysed and discussed this in chapter 5.3. Therefore, we simply assume that for a few (2 to 3) spatially predicted ROIs the corresponding signal is similar to the reported dip. If we look at the predictions from the video 2021_BLI-Baseline_KX/nm200109_A2_GFAP-GeNL_awake/01001_mcor.tiff with the signal video (see figure .11), we observe that most predicted ROIs look like dips, as we expected. Except for dip 13, we can somehow interpret all extracted signals as dips. It is noticeable that all signal dips appear to be relatively strong in terms of intensity. The predictions, which were performed on the same video but with amplified videos (see figure .10) show no new ROIs. This can be explained by the fact that the video simply has a lot of strong dip events, which totally overshadow the weaker ones, or simply that the method is not yet mature enough. Interesting are the predictions, made with the signal video 2021_BLI-Baseline_KX/nm200109_A2_GFAP-GeNL_KX/01004_mcor.tiff, of our development video (see figure .11). We consider most of the predictions to be significant dips. For Dip 6, we can observe that regions can also be predicted that only show a steep descent. This probably comes from detrending transforming such a signal into a dip. It remains interesting if we look at dip 2 and dip 7 of the predictions (see figure .11) with the amplified video. ROIs 1 (see figure 4.2) and 5 (see figure 4.6) are detected as hoped, even if they are only detected in combination with other dip events. The predictions with the signal video 2021_BLI-Baseline_KX/nm200423_A2_GFAP-GeNL/awake/01001_mcor.tiff look somewhat similar to the ground truth. What is special here is that often many regions were predicted at once to a dip signal. This is probably due to the high similarity of the different dip events (see ground truth figure .8). The fact that the algorithm also detects mainly spiking regions is probably because the detrending tries to get closer to the peaks, which can be recognized as a dip in the signal video. The predictions of the same video using the amplified video (see figure .14) do not seem to add additional useful information.

If we compare the ROIs we predicted with the signal videos 2021_BLI-Baseline _KX/ nm200109_A2_GFAP-GeNL_awake/01001_mcor.tiff (see figure .9) and 2021_BLI-Baseline _KX/nm200109_A2_GFAP-GeNL_KX/01004_mcor.tiff (see figure .11) with the corresponding ground truth (see figures .6 and .7) we have the impression that our predicted signals correspond more to the characteristics described in chapter 3.1 (except for the coincidence signals). For the third video 2021_BLI-Baseline_KX/nm200423_A2_GFAP-GeNL/awake/ 01001_mcor.tiff we find the comparison to be difficult because it is a very lively video and the ground truth as well as the predictions are a bit similar. The predictions of the amplified videos turned out to be difficult. Only the predictions on the development video 2021_BLI-Baseline_KX /nm200109_A2_GFAP-GeNL_KX/01004_mcor. tiff (see figure .12) has shown that our signal amplification procedure (see chapter 3.5) has the potential to give less significant dips (ROI 1, see figure 4.2 and ROI 5, see figure 4.6) enough significance for segmentation. These regions were not detected by ground truth (see figure .7 or predictions with signal video (see figure .11.

7 Conclusion

The goal of our thesis was to develop a framework in python, hosted on the i4ds gitlab, to detect and segment signal dips of relatively unknown characteristics. We have shown that we can extract dips (what we think are dips) with signal processing and segment them with matrix decomposition. On the development video we achieve an F1 score of 0.446 and an IoU of 0.494 and on the test videos we achieve a mean F1 score of 0.2915 and a mean IoU of 0.6355. Qualitative analysis of the results shows that the framework is able to segment strong assumed signal dips well, but can have issues with weaker ones. Nevertheless, we cannot say exactly how well the method actually works, since the ground truth was created by hand and thus does not necessarily contain all dips of the videos. An additional difficulty was that the dips are not well characterized.

We presented multiple ways on how the videos can be preprocessed, with denoising to improve signal to noise and local detrending being the most important ones. In addition, some dips are very small that they need to be amplified that further algorithms perceive them as significant. This leads to what seems to us good results if the parameters are adapted for each video. We think that the parameter for denoising and detrending can stay the same for different videos but the amplification needs to be adapted to each video. We think that the signal amplification procedure can be improved significantly since the noise level estimation and the amplification process itself have been developed rather experimentally.

The matrix factorization algorithm has shown to reliably extract ROIs if alpha was chosen appropriately and the videos are detrended and denoised. In addition, we have shown a method for the selection of alpha. However, because of the nature of the data, the rules are more of a guideline and a "rule of thumb". Unfortunately, a dip is still not yet clearly characterized. If a dip was more clearly defined, we are confident that the matrix factorization could be more automated and the alpha could automatically be selected. Generally, the alpha should be selected more conservatively.

Currently, the factorization is somehow slow. For LARS, there is an experimental implementation in cuML (Raschka et al., 2020), which could probably be expanded to make it useable for SNNMF. This would allow for quicker iterations and more experiments. Furthermore, there are preprocessing steps, which certainly have optimization potential in terms of performance. For example, pixelwise operations are locally limited and could therefore be parallelized on several CPUs or GPUs. For the smoothing we use the convolution, which can be computed in the Fourier domain with Fast Fourier Transformation (FFT) and thus has optimization potential.

Our approach is based on how significant we process the signal dip level so that they can be segmented by the matrix decomposition. Provided more is known about the characteristics of the signal dips, other assumptions and additional or different processing steps can be made to adjust the significance level of the signals accordingly. In addition, even though signal amplification could help to detect weaker dips, this approach is highly experimental and can be improved with better noise level estimation as well as a better signal amplification function.

References

Boutsidisa, ., & Gallopoulosb, E. (2007). Svd based initialization : A head start for nonnegativematrix factorization.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, *32*(2). https://doi.org/10.1214/009053604000000067

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise, 226–231.

Holland, P. W., & Welsch, R. E. (1977). Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, *6*(9), 813–827.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pnevmatikakis, E. A., Gao, Y., Soudry, D., Pfau, D., Lacefield, C., Poskanzer, K., Bruno, R., Yuste, R., & Paninski, L. (2014). A structured matrix factorization framework for large scale calcium imaging data analysis. *arXiv preprint arXiv:1409.2903*.

Pnevmatikakis, E. A., Soudry, D., Gao, Y., Machado, T. A., Merel, J., Pfau, D., Reardon, T., Mu, Y., Lacefield, C., Yang, W., Ahrens, M., Bruno, R., Jessell, T. M., Peterka, D. S., Yuste, R., & Paninski, L. (2016). Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron*, *89*(2), 285–299. https://doi.org/https://doi.org/10.1016/j.neuron.2015.11.037

Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *arXiv preprint arXiv:2002.04803*.

Smith, S. L., & Häusser, M. (2010). Parallel processing of visual space by neighboring neurons in mouse visual cortex. *Nature neuroscience*, *13*(9), 1144–1149.

Smith, S. W., Linearity, S., Fidelity, S., Decompositions, C., Notation, P., Nuisances, P., Basics, F., High-Pass, B.-P., Convolution, F., Overshoot, S. R., et al. (n.d.). The scientist and engineer's guide to digital signal processing by steven w. smith, ph. d.

Tošić, I., & Frossard, P. (2011). Dictionary learning. *IEEE Signal Processing Magazine*, *28*(2), 27–38.

Wang, Y., DelRosso, N. V., Vaidyanathan, T. V., Cahill, M. K., Reitman, M. E., Pittolo, S., Mi, X., Yu, G., & Poskanzer, K. E. (2019). Accurate quantification of astrocyte and neurotransmitter fluorescence dynamics for single-cell and population-level physiology. *Nature neuroscience*, *22*(11), 1936–1944.

Zitnik, M., & Zupan, B. (2012). Nimfa: A python library for nonnegative matrix factorization. *Journal of Machine Learning Research*, *13*, 849–853.

List of Figures

1.1 Here is an example of predicted ROIs on the left compared to the corresponding ground truth on the right of the video 2021_BLI-Baseline_KX/ nm200109_A2_GFAP-GeNL_awake/01001_mcor.tif. The segmentation was done on amplified signals presented in 3.5. 2

3.1 Here is an example image which represents an in time averaged video 2021_BLI-Baseline_KX/nm200109_A2_GFAP-GeNL_KX/01004.tif. The colorbar on the right displays the intensity values of this image. The marked region shows an instance of an ROI. 4

3.2 This signal describes the in space averaged intensity of the ROI depicted in figure 3.1. There is a sharp dip from frame 180 to 220. After the ascension, a small peak can be observed. In addition, a steadily decreasing trend over the whole signal can be noticed. 5

3.3 Image of the values of a spatial Gaussian kernel used for signal smoothing. 8

3.4 On the left is the original signal of a spatial pixel of a video in time. A Gaussian filtering in space and time with a kernel size of (7,9,9) TxHxW with a temporal σ of 5 and spacial σ of 3 is applied. 9

3.5 The hat function (see equation 3.12) treats positive and negative outliers equally where s controls the steepness of the slopes. The higher s the higher the the data with low residuals are weighted. 11

3.6 The tangential equation 3.13 tends to weight very low negative residuals weakly and very high, positive residuals strongly. Small residuals near zero tend to be weighted similarly, although this depends very much on s. Here, s was chosen to be 5. With larger s, the function becomes steeper at the ends and flatter in the middle. 11

3.7 The equation 3.14 (here with $s = 1$) shows a linear weighting of the relative residuals. That is, the more positive the residuals, the more weight they receive. A linear behavior could also be achieved with the equations 3.15 and 3.13. However, an advantage is that s controls the slope of the linear behavior, i.e., the range of values can go from [0,s], which gives signal intensities with large positive residuals an even much stronger weight for the next fit. 12

3.8 The logistic sigmoid function 3.15 has an S-shaped form (here $s = 6$). That is, signal points with negative residuals tend to have a small weight, but this weight increases rapidly as the residuals become more positive. How fast more positive residuals gain weight depends on s. The higher s is chosen, the more the function takes on an S- or step-shaped form. In the extreme case, data points with negative residuals could be almost completely excluded from the new fit of the next iteration. 12

3.9 On the left is the smoothed signal and the assumed trend. On the right is the same signal where the trend was subtracted. For this a 5th degree polynomial was used with a sigmoid weight function (see equation 3.15) with $s = 8$. Here you can see very well how the trend tends to attach to the upper signal peaks. . 13

3.10 Here $f(x)$ is shown in the interval [0,2]. You can see the convex behavior between the interval [0,1] which leads to an attenuation of the noise. However, the closer x gets to 1, the less damping effect the function has. 14

3.11 $f(x)$ where $x > 1$ becomes concave. Due to the already very steep slope, however, rather weaker signals are thus strongly amplified. However, this amplification decreases steadily. From $x = 23$ the signal is attenuated, but we do not expect signal to noise ratios of 23 on the videos. 14

3.12 Above are four different detrended and smoothed signals from different space locations, where the orange line is the noise level estimation estimated with the mean of the block minimas with a block size of 20. The first two signals have dips, but they are quiet different regarding the signal to noise ratio. The latter two signals just look like noise, but with differences in signal intensity. Below we see that the signal intensity of the two dips have massively converged and the Noise has been attenuated. However, in all amplified signals there are also individual noise fragments introduced by this process. 15

3.13 Spatial temporal matrix created from the video *01004.tif*. One column corresponds to one frame and one row corresponds to a pixel over time in the video. . 16

3.14 The evolution of the error term described in equation 3.17 over multiple iterations of the SNNMF-algorithm described in 2 applied on the detrended and denoised video *01004.tif*. As we can see, the reconstruction error drops with each iteration of SNNMF. The best iteration is number 16 and the algorithm is stopped at iteration 21 because of early stopping. 19

4.1 Here is the video averaged in time 2021_BLI-Baseline_KX/nm200109_A2_GFAP-GeNL_KX/01004.tif which roughly highlights the regions extracted with Fiji. These are all hand-selected regions because we think they show up various signal characteristics of the video very well. 20

4.2 **Zone 1:** Around frame 300 there is a clear dip with an intensity change of about 15, which lasts a little over 10 frames. Shortly after the dip is over, a small peak appears. At frame 340 there is another dip, but much smaller in intensity and time. This zone seems to be clearly classified as an ROI based on the first dip. . 21

4.3 **Zone 2:** While there are a lot of signals dips visible, some of them can be perceived as signal peaks as the signal alternates in time. We discussed this with our scientific collaborators in Copenhagen and concluded that the dip between frames 310 and 330 is a signal dip of interest due to the steep behaviour and the relatively large intensity change of about 30. The dip from frame 390 to 450 seems to be more of a signal dip of interest rather than not since the dip and rise are rather abrupt. 21

4.4 **Zone 3:** This zone looks complex. The signal alternates with a very high intensity, which makes it difficult to classify whether it is a dip or a spike. Assuming the signal dip between frames 350 and 450 is a dip, is the signal between frames 480 and 560 also a dip or is that the trend, which has two small spikes at both ends? Even looking at the signal changes earlier in time, it is difficult to tell if they are more like dips or spikes. Also, our scientific collaborators at the University of Copenhagen were not clear about the classification. 21

4.5 **Zone 4:** Here there are a few interesting characteristics to observe. Between frames 80 and 230 we have a strong signal dip. This is compared to other observed signal dips very high in intensity (around 70) as well as very long in time (more than 100 frames). It does not seem completely clear whether the small part between frames 180 and 200 is a separate event or not. The signal increase after the dip seems to occur again here. 21

4.6 **Zone 5:** In this zone, a signal dip of interest is clearly visible between frames 190 and 215, which is followed by another short peak. The intensity change of the peak is about 10. 22

4.7 Reference pixel of **Zone 1** 23
4.8 Reference pixel of **Zone 2** 23
4.9 Reference pixel of **Zone 3** 23
4.10 Reference pixel of **Zone 4** 24
4.11 Reference pixel of **Zone 5** 24
4.12 Pixel with oxygen activity but no signal dip 24
4.13 Pixel outside of oxygen activity where just noise is expected 25
4.14 On the left is a video frame of the original signals. On the right, spatial and temporal smoothing was applied. 25
4.15 Reference pixel of **Zone 1** 26
4.16 Reference pixel of **Zone 2** 26
4.17 Reference pixel of **Zone 3** 27
4.18 Reference pixel of **Zone 4** 27
4.19 Reference pixel of **Zone 5** 27
4.20 Pixel with oxygen activity but no signal dip 28
4.21 Pixel outside of oxygen activity where just noise is expected 28
4.22 On the left is the frame 120 of the smoothed video. On the right is the same frame of the assumed background video where zone 4 (see figure 4.5) is much brighter. 29
4.23 The signal dip of zone 4 (see 4.5) can be seen very strongly at frame 120. 29
4.24 Here you can see the signal video in frame 200. Zone 3 (see figure 4.4) and zone 4 (see figure 4.5) are clearly visible. Zone 5 (see figure 4.6) also has a signal dip here. However, this is very weak relative to zones 3 and 4 in terms of intensity, which is reflected in the visibility of this image. 30
4.25 In frame 295 the dip of zone 1 (see figure 4.2) is visible. However, compared to the dip of zone 5 (see figure 4.24), this dip is relatively weak. 30
4.26 In this frame we see the signal dip of zone 2 (see figure 4.3) relatively pronounced. In addition, zones 3 (see figure 4.4) and 4 (see figure 4.5) are still visible, although they are not in focus in this frame. 31
4.27 In frame 400 we have two dips that stand out. On the one hand, we have zone 3 (see figure 4.4), where the dip reaches its low, and zone 4 (see figure 4.5), where the last significant signal dip can be seen. 31
4.28 Reference pixel of **Zone 1** 32
4.29 Reference pixel of **Zone 2** 32
4.30 Reference pixel of **Zone 3** 33
4.31 Reference pixel of **Zone 4** 33
4.32 Reference pixel of **Zone 5** 33
4.33 Pixel with oxygen activity but no signal dip 34

4.34 Pixel outside of oxygen activity where just noise is expected 34

4.35 The signal dip of zone 4 (see 4.31) can be seen very strongly at frame 120 on both, the signal and amplified videos. 35

4.36 At frame 200, at the top of the signal frame, zone 3 (see figure 4.30) and zone 4 (see figure 4.31) are clearly visible. However, zone 3 seems to vanish on the amplified frame. On the other hand, zone 5 (see figure 4.32) at [580,418] (left, bottom) gets much more visibility on the amplified frame compared to the signal frame. 35

4.37 Here at frame 295, the dip of zone 1 (see figure 4.28) of the signal frame is visible, but with low intensity compared to stronger dips. On the right figure, however, the dip of zone 1 has now a comparable signal level. 36

4.38 On the amplified frame, the dip of zone 3 (see figure 4.30) and zone 4 (see figure 4.31) vanish as expected based on the reference pixels. The spatially largest dip of zone 2 (see figure 4.29 remained. 36

4.39 Frame 400 is interesting with respect to zone 2 (see figure 4.29) and zone 3 (see figure 4.30). The second signal dip from zone 2 is visible in a small spot on [370,610]. This is probably because we declared zone 2 to be relatively large (see figure 4.1), but the second event at frame 400 is effectively not the size of zone 2 at all and therefore hurt the dip intensity of the second event. Interesting is also zone 3, which does not disappear completely, but with a value range of -4 to -5 turns out relatively small, although the signal dip has reached the low point approximately in this frame. 37

5.1 Number of non-zeros in the positional matrix per alpha of the factorization of the detrended and denoised video *01004.tif*. 38

5.2 Frobenius norm of the reconstruction per alpha of the factorization of the detrended and denoised video *01004.tif*. We can observe that the reconstruction error increases when alpha increases. 38

5.3 Overlaid predicted ROIs with different alphas on the matrix-factorized video *01004.tif*. A higher alpha leads to more regularization and thus to more zero pixels in the positional matrix. Also, brighter regions mean that there are overlapping ROIs because those figures are constructed by adding all positional, binary matrices together of the ROIs. Generally, we can observe that with a higher alpha the ROIs get more compact and less ROIs are found. 39

5.4 Different, selected ROIs with their extracted, unedited signal of the original video (*01004.tif*). Left we see the ROIs in the position matrix and on the right the spatial averaged signals of the corresponding ROI. As it can be seen, the alpha is probably too conservative because in Dip 6, we see that the ROI is spanning over the whole image and thus expected to not be an actual ROI. 40

5.5 Different, selected ROIs with their extracted, unedited signal of the video mentioned in figure 5.4. Left we see the ROIs in the position matrix and on the right the signals of the corresponding ROI. This alpha seems to be better than the alpha chosen in figure 5.4 because no ROI is expected to span over the whole spatial region. 41

5.6 Time to factorize per alpha (with warm start) of the denoised and detrended video *01004.tif*, processed on an **Intel Core i7-9700KF (3.60GHz / 12MB).** 42

5.7 Extracted signals with a different signal intensity scaling of overlapping ROIs. As we can see, the extracted signals vary per ROI. 42

5.8 Multiple ROI per area with the corresponding to dip 4 of figure .9 found in the appendix. The left figure shows the segmented spatial ROIs where the right figure depicts the temporal in space averaged signal of both ROIs. 43

5.9 On the top are the in space averaged signals of each individual spatial region from figure 5.8 of the original video where on the bottom are the spatial averaged signals of the detrended and filtered video. Between the unprocessed signals we reach a correlation of 0.786 and between the detrended and filtered signals we have a correlation of 0.928. 44

6.1 Alpha vs recall on the denoised and detrended video *01001_mcor.tiff* A higher alpha leads to a lower recall. 45

6.2 Predicted ROIs vs ground truth ROIs overlaid. So the violet regions are not recognized by the predictions and the violet regions are predicted areas with no underlying ground truth label. Thus, you can interpret the pixels in light-blue and yellow as TP and TN, the red as FN and the violet as FP. 46

6.3 Predicted ROIs vs ground truth ROIs overlaid. So the violet regions are not recognized by the predictions and the violet regions are predicted areas with no underlying ground truth label. Thus, you can interpret the pixels in light-blue and yellow as TP and TN, the red as FN and the violet as FP. 47

6.4 Evaluation of false positive and false negative ROIs. Red means FN and violet means FP. As we can see $SNNMF$ missed smaller and shorter dips (as seen in three pictures to the right). However, some large dips were not marked in the ground truth (seen in the bottom left picture). 48

.1 Evaluation of false positive and false negative ROIs of the amplified video *01001.tif*. Red means FN, violet means FP, light-blue TP and yellow TN. 60

.2 Evaluation of false positive and false negative ROIs of the video amplified video *01001.tif (2)*. Red means FN, violet means FP, light-blue TP and yellow TN. . 61

.3 Evaluation of false positive and false negative ROIs of the denoised and detrended video *01001.tif (2)*. Red means FN, violet means FP, light-blue TP and yellow TN. 62

.4 Evaluation of false positive and false negative ROIs of the detrended and denoised video *01001.tif*. Red means FN, violet means FP, light-blue TP and yellow TN. 63

.5 Evaluation of false positive and false negative ROIs of the detrended and denoised video *01001.tif*. Red means FN, violet means FP, light-blue TP and yellow TN. 64

.6 Ground truth ROI signals of the video 2021_BLI-Baseline_KX/nm200109_A2_GFAP-GeNL_awake/01001_mcor.tiff 65

.7 Ground truth ROI signals of the video 2021_BLI-Baseline_KX/nm200109_A2_GFAP-GeNL_KX/01004_mcor.tiff 66

.8 Ground truth ROI signals of the video 2021_BLI-Baseline_KX/nm200423_A2_GFAP-GeNL/awake/01001_mcor.tiff 67

.9 Predicted ROIs and corresponding signal of the signal video 2021_BLI-Baseline_KX/ nm200109_A2_GFAP-GeNL_awake/01001_mcor.tiff after applying *SNNMF*. . . 68

.10 Predicted ROIs and corresponding signal of the amplified video 2021__BLI-Baseline__KX/ nm200109__A2__GFAP-GeNL__awake/01001__mcor.tiff after applying *SNNMF*. . . 69

.11 Predicted ROIs and corresponding signal of the signal video 2021__BLI-Baseline__KX/ nm200109__A2__GFAP-GeNL__KX/01004__mcor.tiff afting *SNNMF*. 70

.12 Predicted ROIs and corresponding signal of the amplified video 2021__BLI-Baseline__KX/ nm200109__A2__GFAP-GeNL__KX/01004__mcor.tiff after applying *SNNMF*. . . . 71

.13 Predicted ROIs and corresponding signal of the signal video 2021__BLI-Baseline__KX/ nm200423__A2__GFAP-GeNL/awake/01001__mcor.tiff after applying *SNNMF*. . . 72

.14 Predicted ROIs and corresponding signal of the amplified video 2021__BLI-Baseline__KX/ nm200423__A2__GFAP-GeNL/awake/01001__mcor.tiff after applying *SNNMF*. . . 73

List of Tables

6.1 Results ROI detection with alpha 50 on denoised and detrended 01001__mcor.tif. 45

6.2 Results ROI coverage alpha 50 on denoised and detrended 01001__mcor.tif. . . . 46

6.3 Results ROI detection with alpha 1 on amplified 01001__mcor.tif. 47

6.4 Results ROI coverage alpha 1 on amplified 01001__mcor.tif. 47

6.5 Results ROI detection with alpha 10 on denoised and detrended 01004.tif. 48

6.6 Results ROI coverage alpha 10 on denoised and detrended 01004.tif. 49

6.7 Results ROI detection with alpha 0.1 on amplified 01004.tif. 49

6.8 Results ROI coverage alpha 0.1 on amplified 01004.tif. 49

6.9 Results ROI detection with alpha 20 on denoised and detrended 01001__mcor.tif (2). 49

6.10 Results ROI coverage alpha 20 on denoised and detrended 01001__mcor.tif (2). . 49

6.11 Results ROI detection with alpha 0.01 on amplified 01001__mcor.tif (2). 50

6.12 Results ROI coverage alpha 0.01 on amplified 01001__mcor.tif (2). 50

Appendix

Ground truth

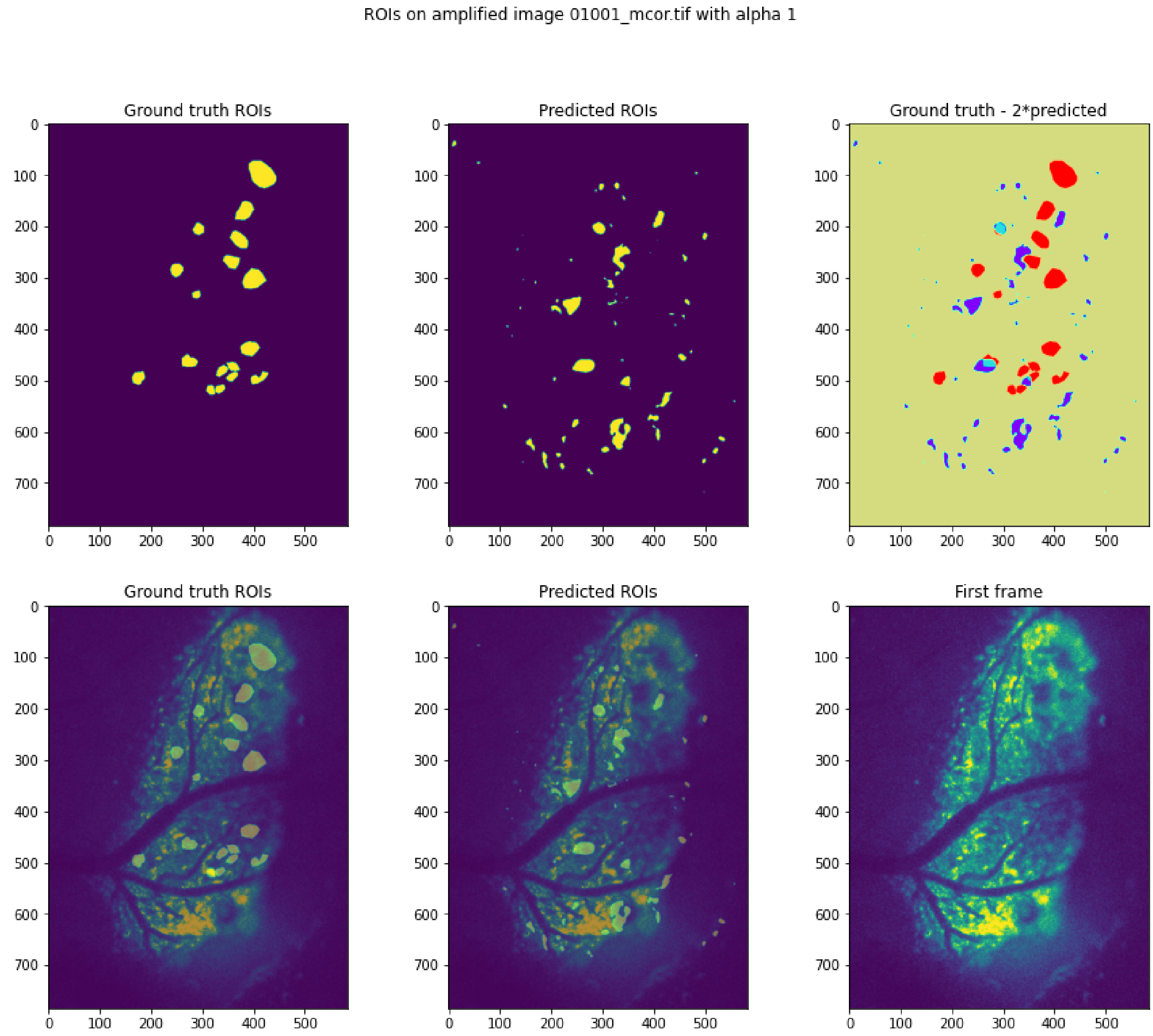


Figure .1: Evaluation of false positive and false negative ROIs of the amplified video *01001.tif*. Red means FN, violet means FP, light-blue TP and yellow TN.

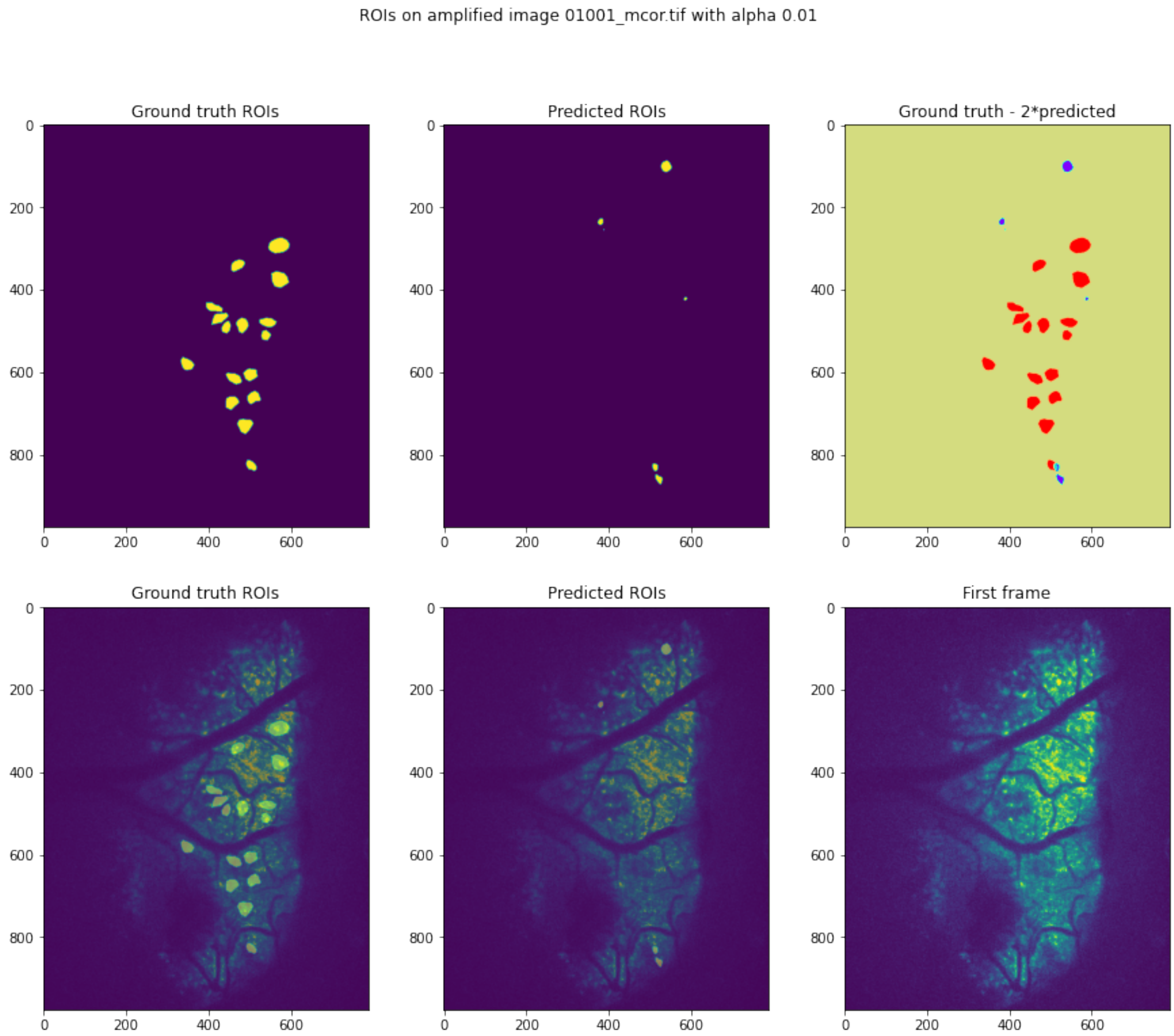


Figure .2: Evaluation of false positive and false negative ROIs of the video amplified video *01001.tif (2)*. Red means FN, violet means FP, light-blue TP and yellow TN.

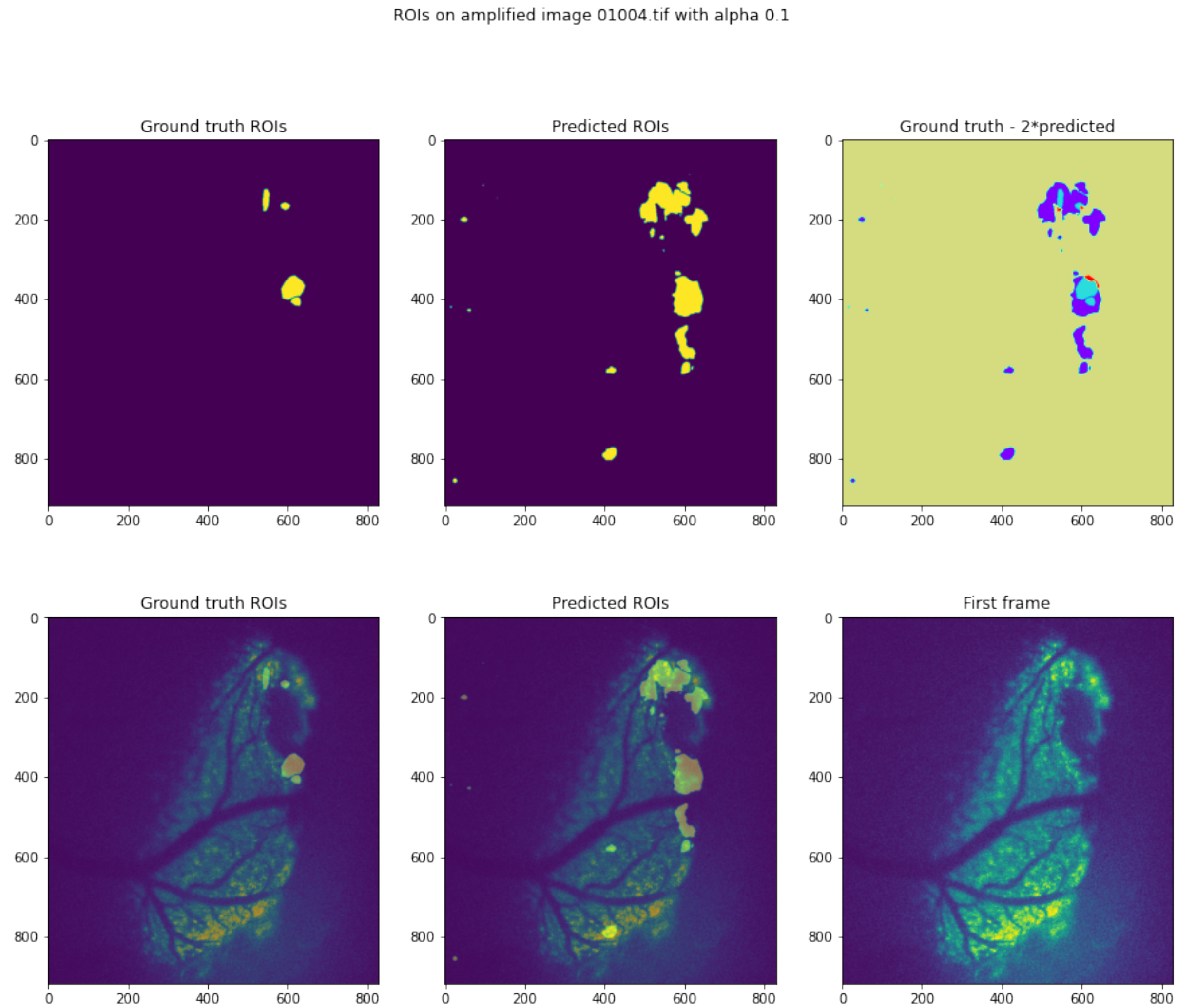


Figure .3: Evaluation of false positive and false negative ROIs of the denoised and detrended video *01001.tif (2)*. Red means FN, violet means FP, light-blue TP and yellow TN.

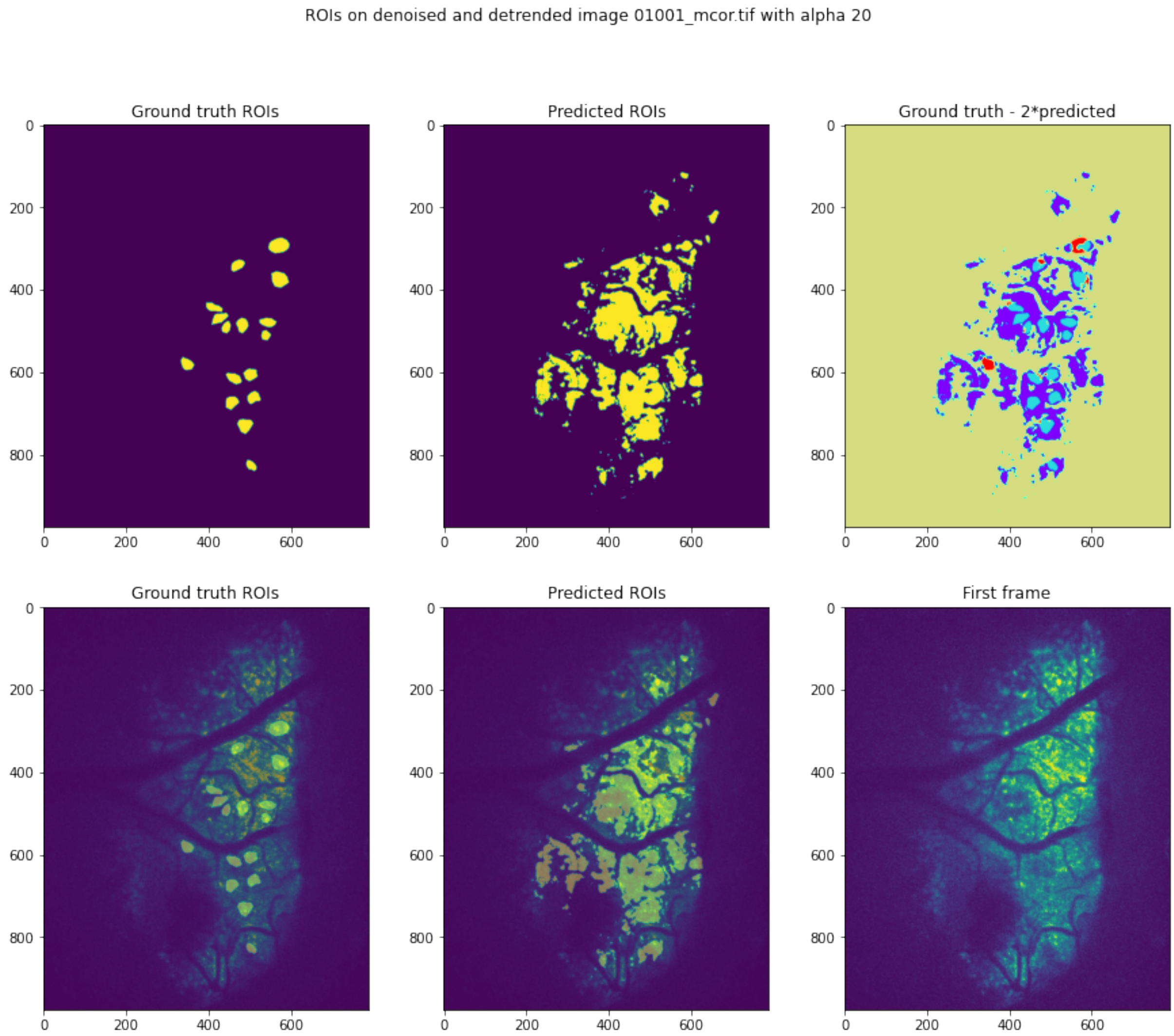


Figure .4: Evaluation of false positive and false negative ROIs of the detrended and denoised video *01001.tif*. Red means FN, violet means FP, light-blue TP and yellow TN.

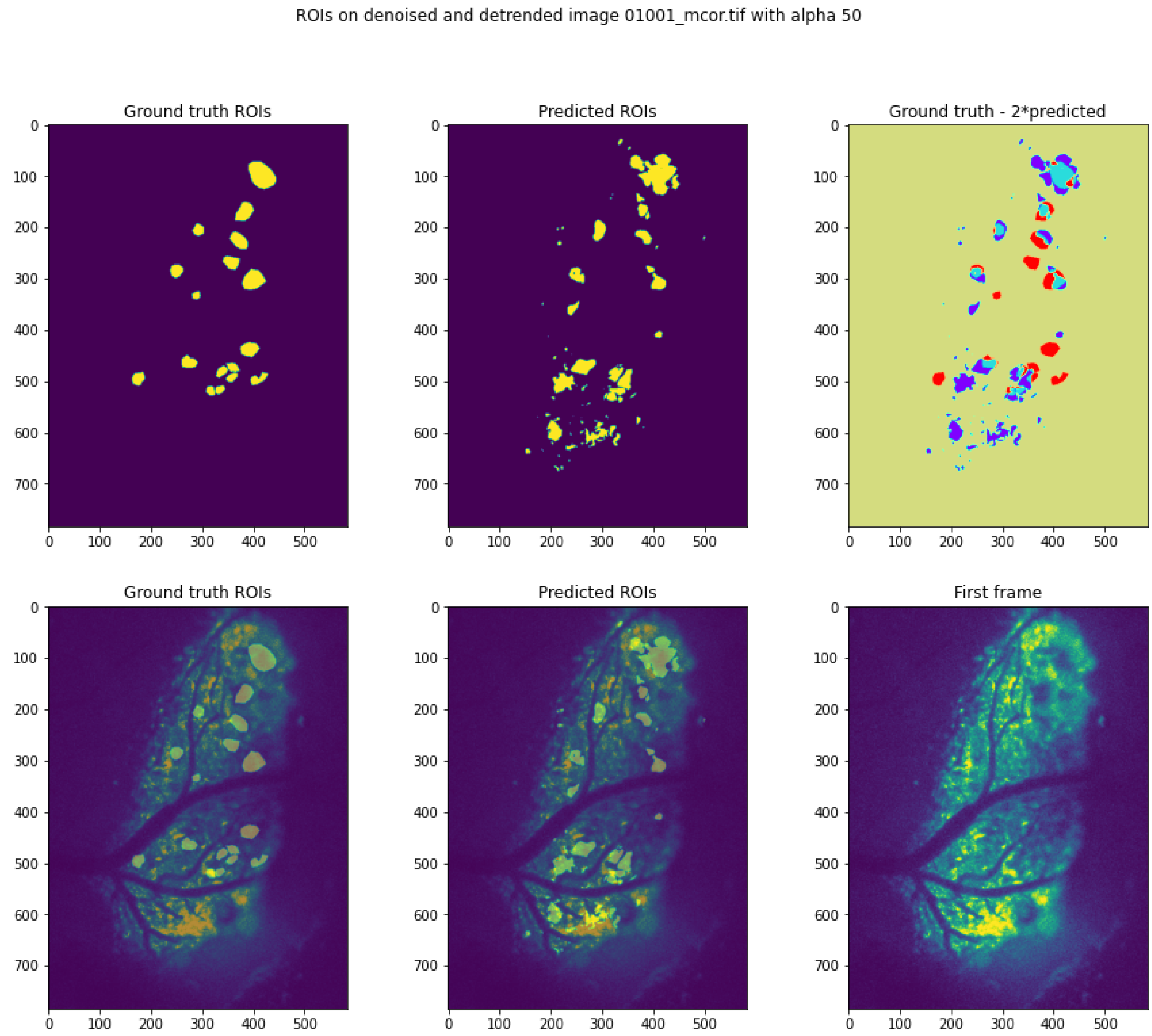


Figure .5: Evaluation of false positive and false negative ROIs of the detrended and denoised video *01001.tif*. Red means FN, violet means FP, light-blue TP and yellow TN.

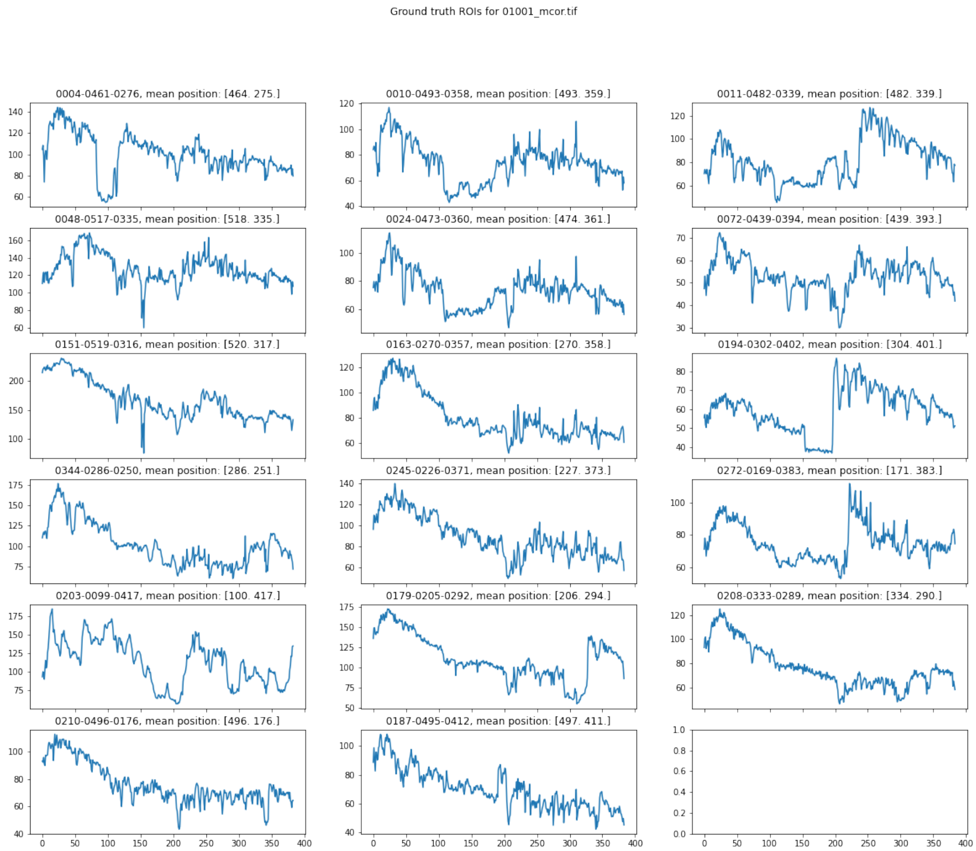


Figure .6: Ground truth ROI signals of the video 2021_BLI-Baseline_KX/nm200109_A2_GFAP-GeNL_awake/01001_mcor.tiff

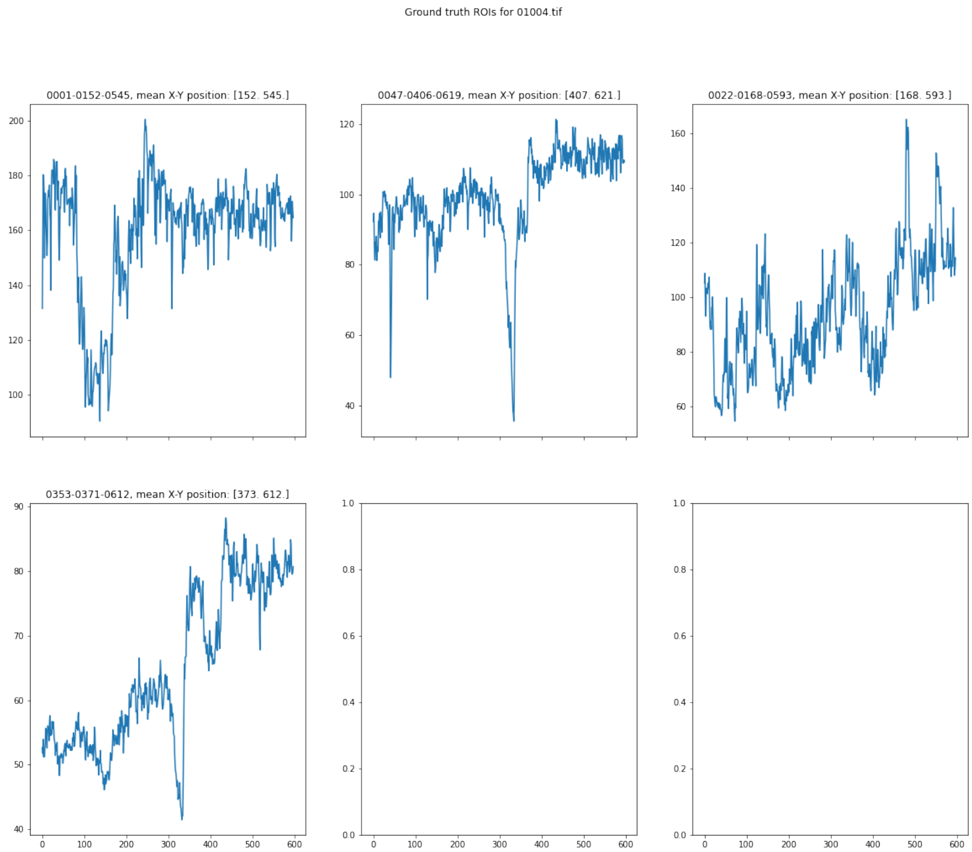


Figure .7: Ground truth ROI signals of the video 2021_BLI-Baseline_KX/nm200109_A2_GFAP-GeNL_KX/01004_mcor.tiff

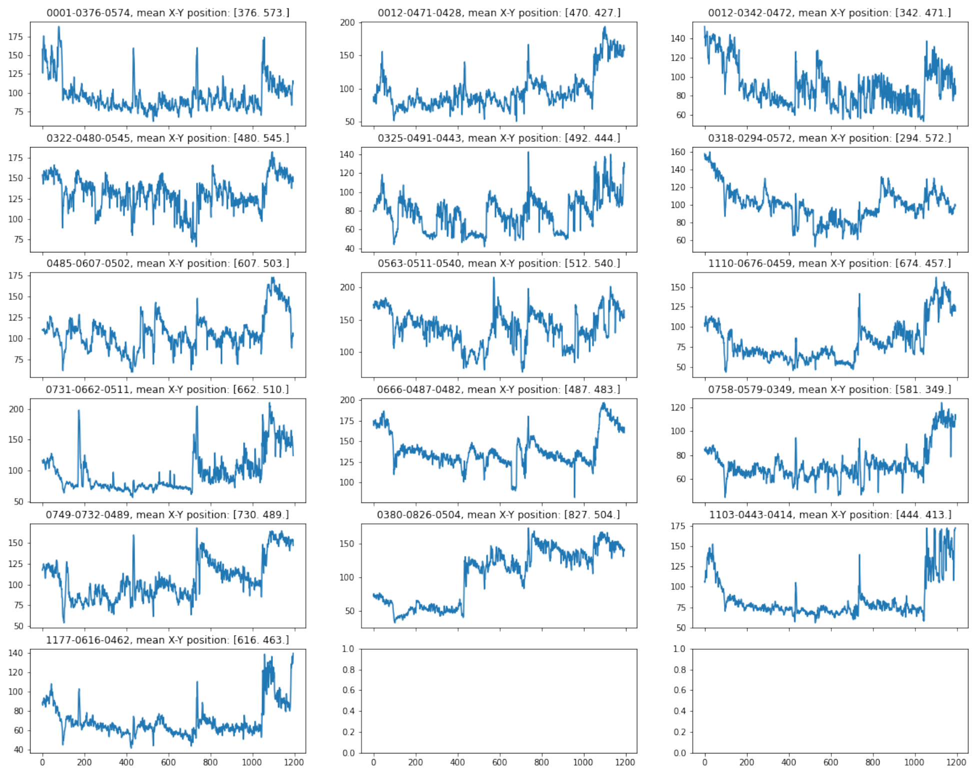


Figure .8: Ground truth ROI signals of the video 2021_BLI-Baseline_KX/nm200423_A2_GFAP-GeNL/awake/01001_mcor.tiff

Results

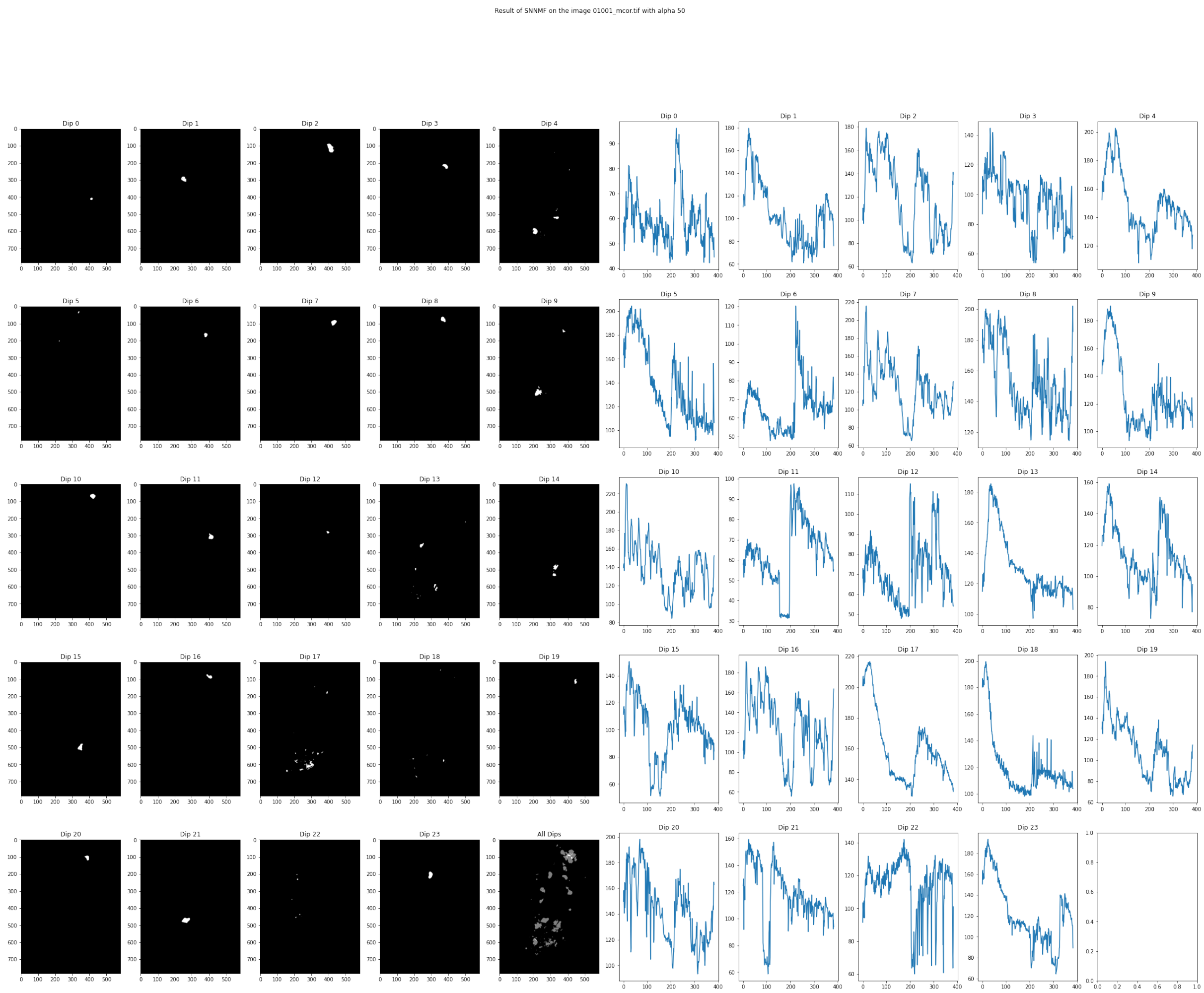


Figure .9: Predicted ROIs and corresponding signal of the signal video 2021_BLI-Baseline_KX/ nm200109_A2_GFAP-GeNL_awake/01001_mcor.tiff after applying *SNNMF*.

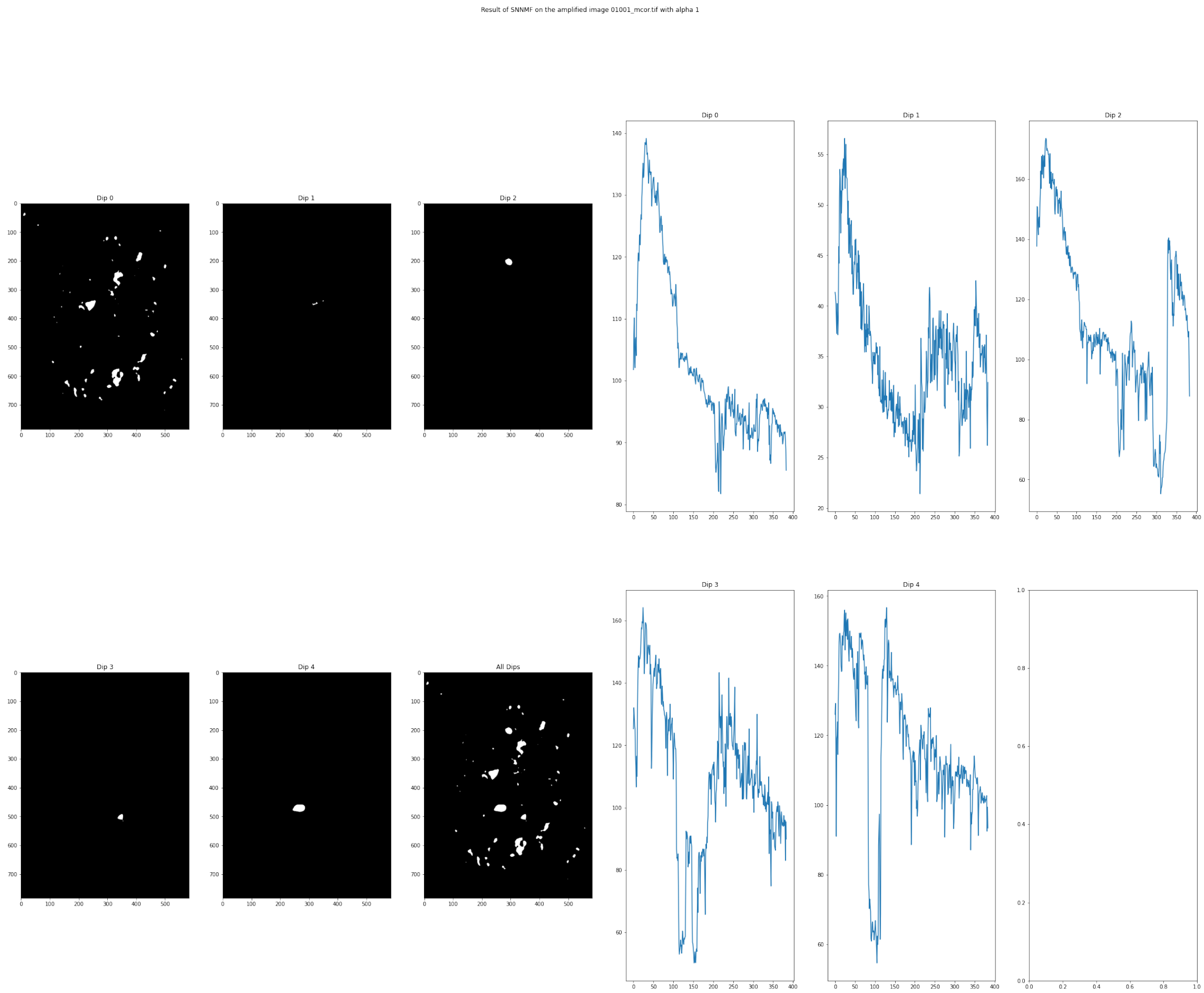


Figure .10: Predicted ROIs and corresponding signal of the amplified video 2021_BLI-Baseline_KX/
nm200109_A2_GFAP-GeNL_awake/01001_mcor.tiff after applying *SNNMF*.

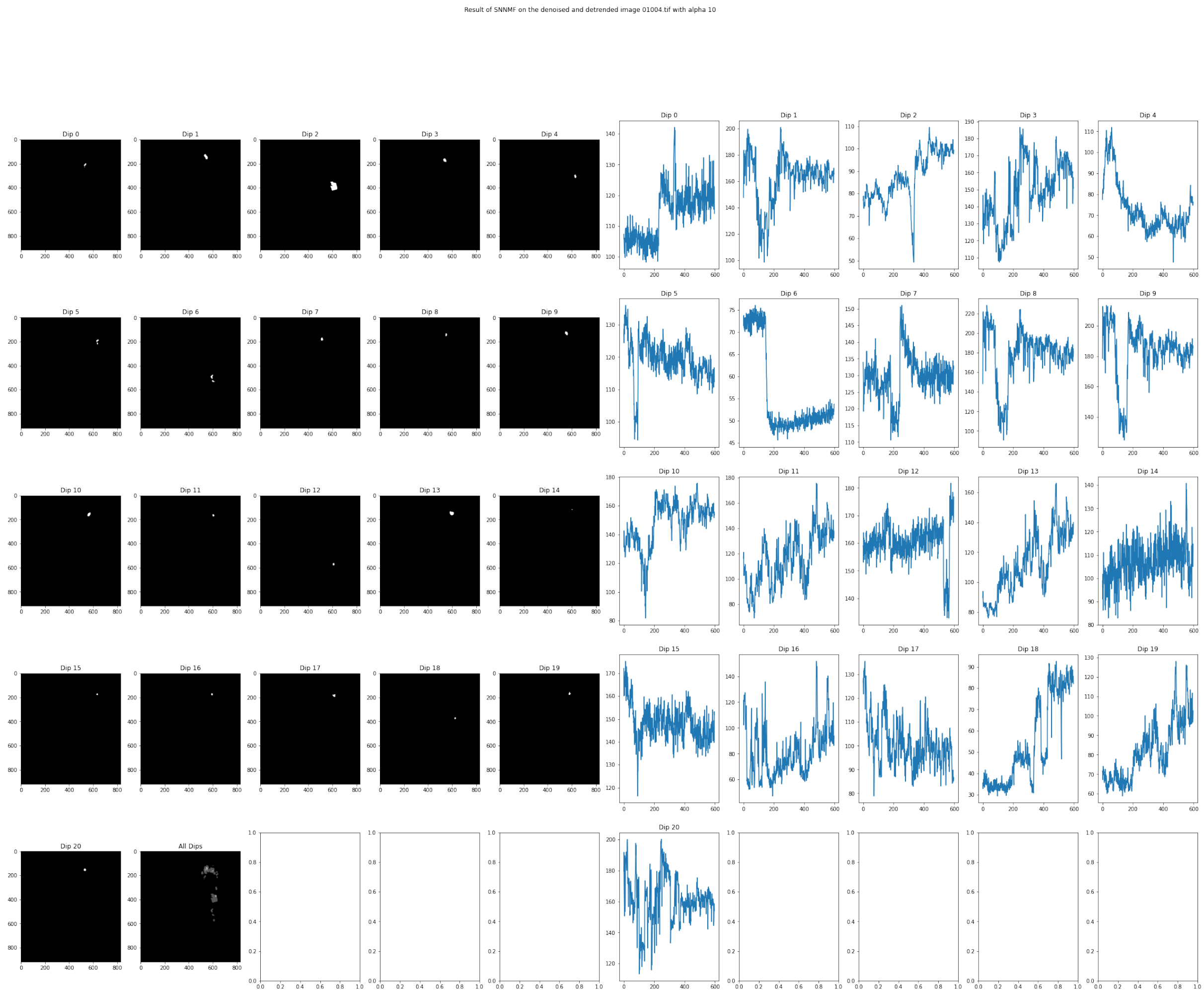


Figure .11: Predicted ROIs and corresponding signal of the signal video 2021_BLI-Baseline_KX/ nm200109_A2_GFAP-GeNL_KX/01004_mcor.tiff afting *SNNMF*.

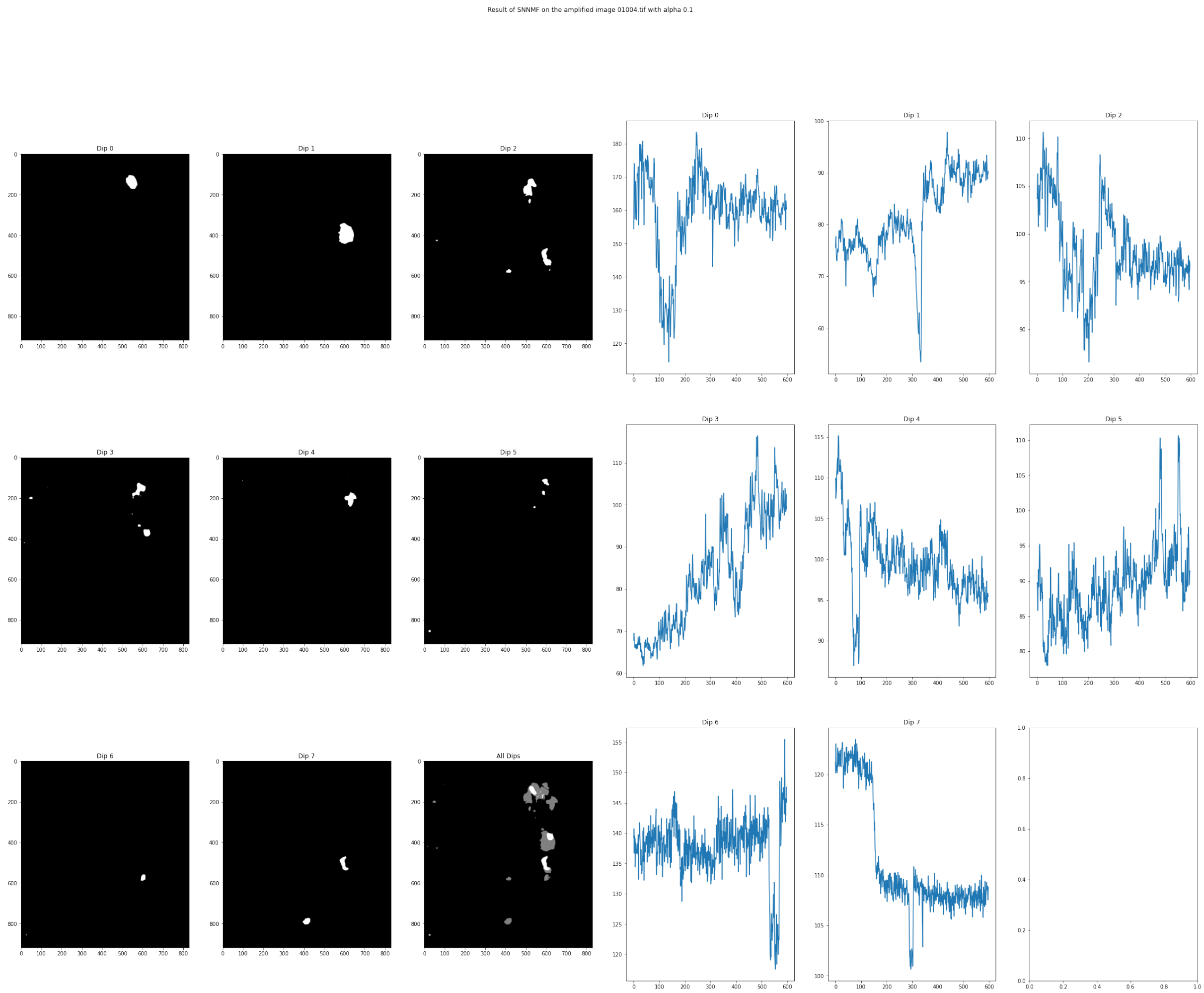


Figure .12: Predicted ROIs and corresponding signal of the amplified video 2021_BLI-Baseline_KX/
nm200109_A2_GFAP-GeNL_KX/01004_mcor.tiff after applying *SNNMF*.

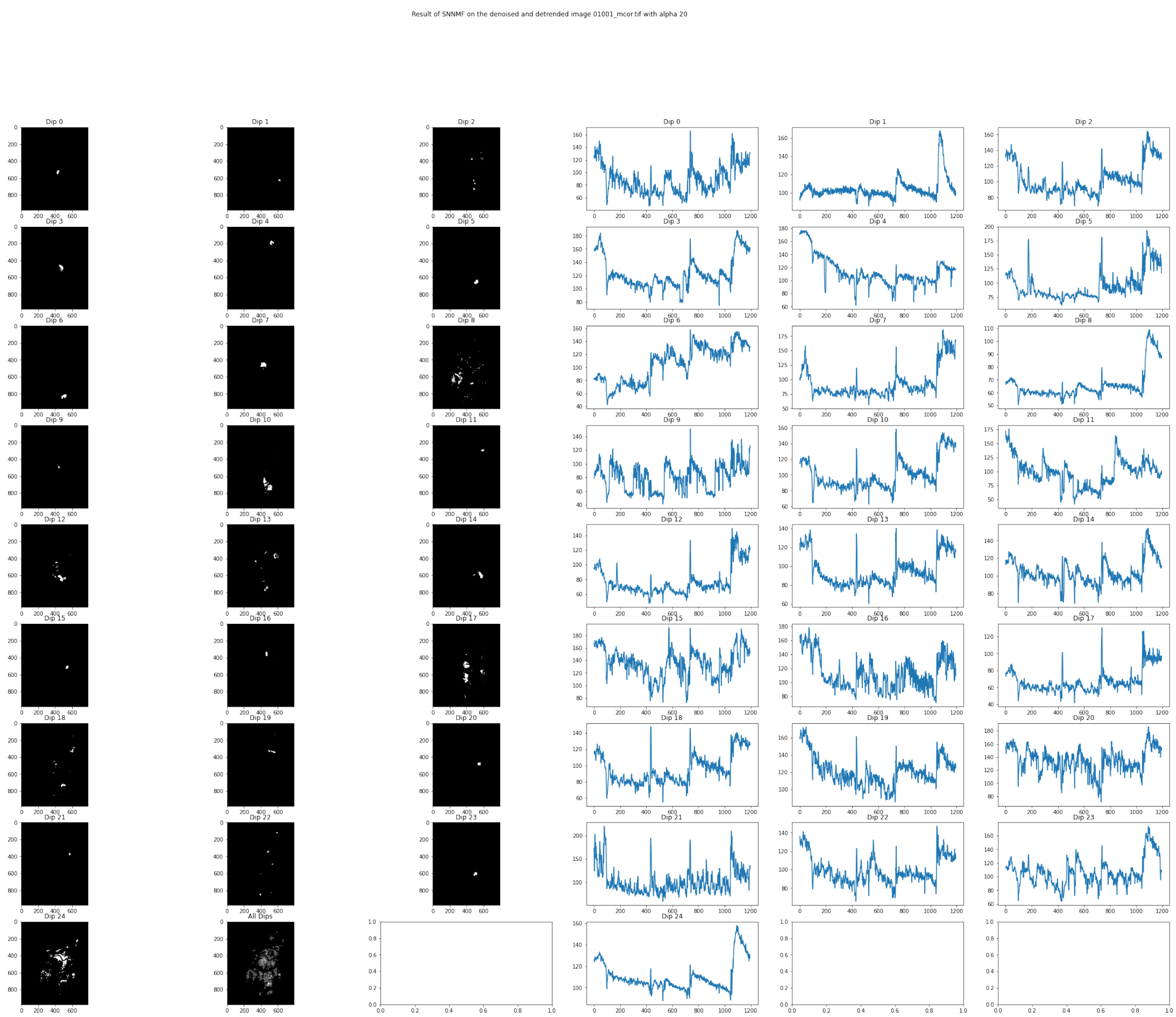


Figure .13: Predicted ROIs and corresponding signal of the signal video 2021_BLI-Baseline_KX/ nm200423_A2_GFAP-GeNL/awake/01001_mcor.tiff after applying *SNNMF*.

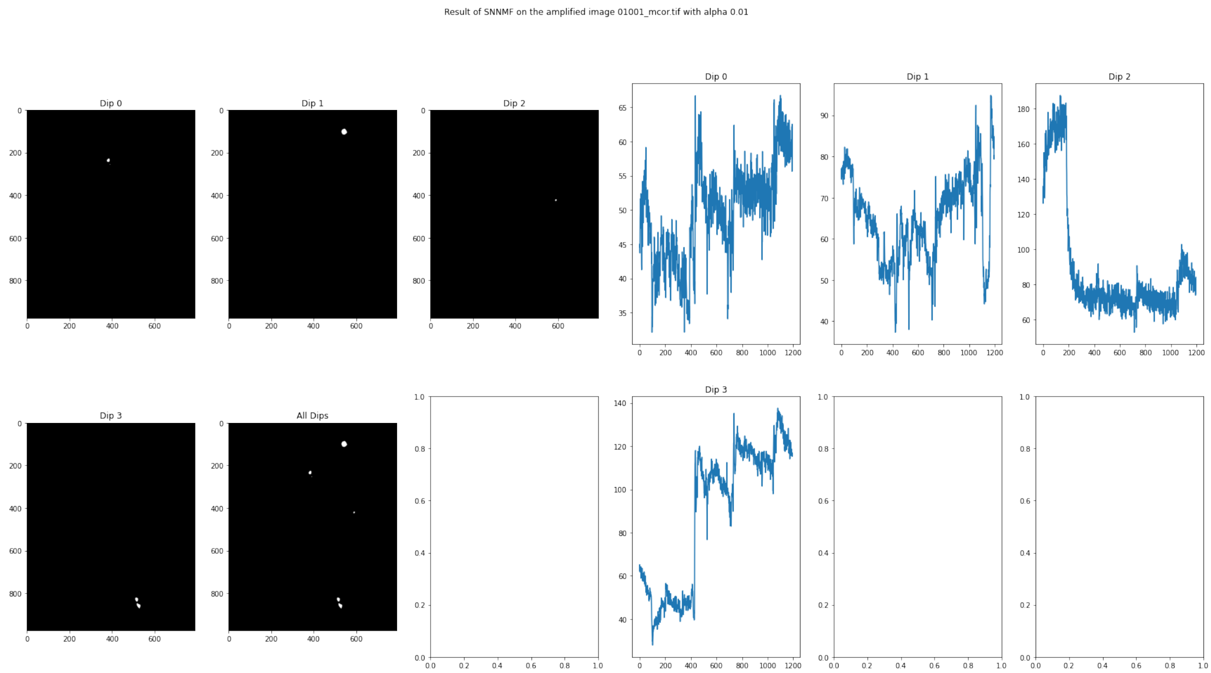


Figure .14: Predicted ROIs and corresponding signal of the amplified video 2021_BLI-Baseline_KX/
nm200423_A2_GFAP-GeNL/awake/01001_mcor.tiff after applying *SNNMF*.