

# Laboratory 1 Part 1: Introduction to Imaging

## Objectives

- Learn how to build a basic microscope
- Familiarize with stage and camera control (Exposure, focusing, lighting control)
- Acquire an image of a biological slide

## Introduction / Basic Rules

The first part of this laboratory is a tutorial to get familiar with your optical kit. The following is a step-by-step process describing how to assemble a basic microscope. A number of basic rules apply to the handling of this kit. These rules include:

- Any glass element in the kit should be handled carefully by the edges. Never place fingers on a surface through which light for imaging passes. This includes imaging targets and slides.
- Dust and oils are bad for most optical elements and the camera. Avoid having the interior of the objective lens and the uncovered camera exposed for any length of time. Never touch the camera sensor.
- Nothing in your kit should require a large amount of force to assemble or disassemble. If something isn't going together easily, there's probably something wrong. Never force any elements that aren't moving easily. If there are problems, ask the instructor or TA for help.
- Please be a good custodian of your optical hardware kit. The kit is your responsibility for the duration of the course. With proper care these kits should be able to be used for many years. Thank you for your help in making this possible.
- If any element of your kit appears damaged or if something doesn't appear right, contact the instructor or TA immediately.

## Microscope assembly

We will begin this lab by assembling a basic microscope. You are provided with all of the hardware elements that are required to construct this imaging device. The figure below shows a complete inventory of all of the hardware elements in your optical kit. Take a moment to ensure all of the parts are present in your kit and to learn some of the nomenclature for the various elements.

In addition to the parts in the picture below, you will have access to a computer, a power supply for the lighting module, and a number of imaging targets.



### Step 1: Construct the linear stage assembly.

First, we begin assembly of the imaging platform. This platform will hold the microscope stage and can be moved vertically to focus the imaging plane. We begin by affixing a mounting plate adapter to the linear stage using four shorter, silver 1/4-20 machine screws. These should be hand-tightened using an appropriate Allen wrench. Note the orientation of the adapter plate. Once affixed, the 1" mount plate should be attached to the silver adapter plate using four 8-32 screws. Again, note the orientation of the mount – the 1" diameter channel in the mount plate should be parallel to the direction of travel of the linear stage. Lastly, prepare a (8" length) 1" diameter post by placing a 1/4-20 set screw in one end of the post. Note that the set screw should have the Allen head *facing outward*.



### Step 2: Affix main support posts and continue stage assembly

Pick a location relatively close to the edge of your breadboard centered along the short side of the platform. Screw the 1" post into the breadboard using the previously placed set screw. Loosen the screw on the 1" mount plate and pace the linear stage assembly on the 1" post as shown below. Align the linear stage to the breadboard grid and tighten the mount clamp.

Prepare a second 16"-long post using two set screws – one to connect two posts together; and another to fix the combined 16" post to the breadboard. Affix the post 10" from the 8"-long post that will hold the microscope stage. (Note that the breadboard has tapped holes with 1" spacing.)

Using four longer 1/4-20 screws, attach the two L-brackets to the linear stage. Note that these brackets have a righthandside and lefthandside, and a top and a bottom. A notch should appear toward the center, top of the bracket as shown below. The hole patterns

are also asymmetric and can only be assembled with the top, up. Only tighten these screws enough to hold on the brackets. We will fully tighten them once the microscope stage is attached.



### Step 3: Finish microscope stage assembly

Next we attach the microscope stage to the linear stage. Handle the microscope stage with care. There is a large glass window in the stage that is fragile. Moreover, scratching this stage or getting excess dust or debris on the stage will complicate future experiments. This stage is held on with four more 1/4-20 screws. Once you've gotten these screws place and started, you may tighten all eight screws on the two L-brackets. You should have just enough room for the stage between the two upright 1" posts. If necessary, you may need to adjust the knob on the stage which controls XY position.



### Step 4: Assemble the lens-camera system components

In the next step we begin the assembly of the optical path of the imaging system. The major elements here are the objective lens, an optical tube assembly and the camera. These elements are more fragile than items in the previous steps and require additional care.

The lens is threaded differently than the optical tubes and requires an adapter. Place that adapter on the lens and hand tighten. A manual aperture (iris) may then be affixed to the back of the lens followed by an optical tube. For this lab, start with an optical tube of 1" length.

The camera is also threaded differently than the optical tubes and requires its own adapter. Hand tighten that adapter and place a 1/2" optical tube on the camera assembly. To the optical tube, the filter wheel assembly may be attached, followed by a 2" optical tube.



### Step 5: Mount lens-camera system

The lens-tube-camera assembly is held in place by two optical tube mounts. Each of these is composed of the tube mount, a 1/2" diameter post, and a right-angle mount that attaches to the taller 1" post. Assemble each tube mount and roughly position the posts

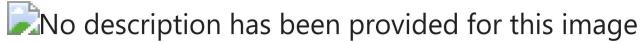
to accept the lens-tube-camera assembly.

**NOTE: Do not allow the tube mounts or the tube assembly to fall onto the glass platform of the microscope stage. You can break the glass in the platform if these drop from a significant height.**

Place the lens-tube half of the assembly in the bottom mount and the tube-camera assembly in the upper mount and screw them together. Make fine adjustments to the lens-tube-camera assembly ensuring that:

- The tube assembly is aligned with the circular hole in the bottom of the microscope stage.
- The optical tube assembly is vertical and parallel to the 1" support post.

Once positioned, hand tighten the screws that hold the right angle mounts to the 1" and 1/2" posts. And, tighten the optical tube mounts to secure the lens-tube-camera assembly.



### Step 6: Add the illumination module

Next, we will be attaching the lighting module to the microscope. This module is composed of three elements. A support arm, a coupler, and the lighting module itself. The support arm is attached to the linear stage using two 1/4-20 screws. It should fit securely between the L-brackets at the lowest available screw positions (see below). The coupler and the light module slide together as shown below.

***Do not apply significant pressure to the support arm. It is sufficient strong to hold the lighting module but will break easily if used to lift the stage, used as a brace for fastening screws, etc.***



### Step 7: Align the lighting module and add a diffuser

***Note that the lighting module is quite bright. Do not look directly into the LED light.***

The light module may be activated by plugging in the provided power supply. A brightness adjustment is provided on the side of the module for some control. Note that

there is a lower threshold where the LED will not light. The glass diffuser should be placed in the channel created by the two L-brackets. **This diffuser should slide in easily.** If it does not, contact the instructor. Also note that this diffuser can slide out easily. **Move your microscope with care to avoid letting the diffuser fall out and break.**

The position of the light module should be adjusted so that the circular illumination covers the circular hole in the bottom of the microscope stage.

Adjust the illumination at a low setting where the LED light is on and stable.



### Step 8: Start imaging – refine microscope set up and start experimenting

Connect the camera to your computer using the provided cable. Note that a USB 3.0 port is required for reliable communication. There should be a green light on the camera if everything is working properly.

You should now be ready to start imaging with your microscope. To start taking images, place an imaging target or microscope slide on the stage. (See below.)

**IMPORTANT: While stage clips are provided to hold a slide in place, the clips should NEVER be placed on the part of the slide with a sample, and NEVER used on the resolution target at all.**



## Test the Camera Module

```
In [ ]: # add BaITools to path
import os
import sys
sys.path.insert(0,r'BaITools')
```

```
In [ ]: # for basic functions
import numpy as np
from time import sleep
import matplotlib.pyplot as plt
from IPython.display import display, clear_output
import cv2
plt.rcParams['font.size'] = 16 # set the font size globally

from zelux import ZeluxCamera as Camera # for camera
```

```
In [ ]: print('Camera initialized as c')
c = Camera()
c.open()
```

Camera initialized as c

```
In [ ]: # check image properties
c.get_properties()
c.set_exposure(10)
c.get_exposure()
c.set_framerate(2)
c.get_framerate()
```

Camera Model	Thorlabs CS165MU
Pixel Size	3.45 microns
Image size	[1440, 1080]
Exposure	28.72 ms
Exposure Range	[0.06, 26843.42] ms
Framerate	18.0
Frame Range	[0.9, 18.0]

```
Out[ ]: 2.0
```

Create a script with the following commands.

You should now see a figure with a live image from the camera. The image is likely blurry since you have not yet adjusted the focus.

```
In [ ]: # try continuous capture
# to quit press live feed press "q"
while True:
    # Convert the image from 10-bit to 8-bit and then display the image.
    cv2.imshow('Camera', cv2.normalize(c.capture().copy(), None, 0, 255, cv2.NORM_
    if cv2.waitKey(30) & 0xFF == ord('s'):
        test_im = c.capture().copy()
        fig, ax = plt.subplots(1, figsize=(12, 5))
        x = ax.imshow(test_im, cmap='gray')
        fig.colorbar(x, ax=ax)
        plt.savefig('savedimage.png')
        if cv2.waitKey(30) & 0xFF == ord('q'):
            break
cv2.destroyAllWindows()
```

```
In [ ]: # acquire and plot a single image now
test_im = c.capture().copy()
fig, ax = plt.subplots(1, figsize=(12, 5))
x = ax.imshow(test_im, cmap='gray')
fig.colorbar(x, ax=ax)
plt.savefig('savedimage.png')
```

```
In [ ]: c.close()
```

## Focusing and image formatting

To change the focus, you rotate the screw adjustment on the linear stage (see below). In the remote setup, this is accomplished by moving the z-stage (variable name `stage_z` from above).

***IMPORTANT: This knob should rotate freely, if you experience resistance stop immediately. Note that you can drive the microscope stage into the lens if you are not careful (damaging the lens or microscope stage or target). Best practice is to focus "away from the lens." That is start with the stage near the lens and move the stage downward.***

Practice finding the in-focus image for your target. Best focus is most easily found by looking at the sharpness of features (e.g. edges) in your image. If you are having difficulty, ask the instructor for help.

If you cannot find a good in-focus image and your image is mostly dark or mostly white, jump ahead to "adjusting signal levels".

Your image is likely rotated and/or flipped relative to its position on the microscope stage. To fix this will require both hardware and software modifications. First, you should rotate the lens-camera assembly to be aligned with the XY-motion axes. This should be accomplished by loosening the tube clamps only (see below). ***WARNING: Do not let the tube assembly drop and fall on the microscope stage. Hand tighten clamps once the assembly is aligned.***

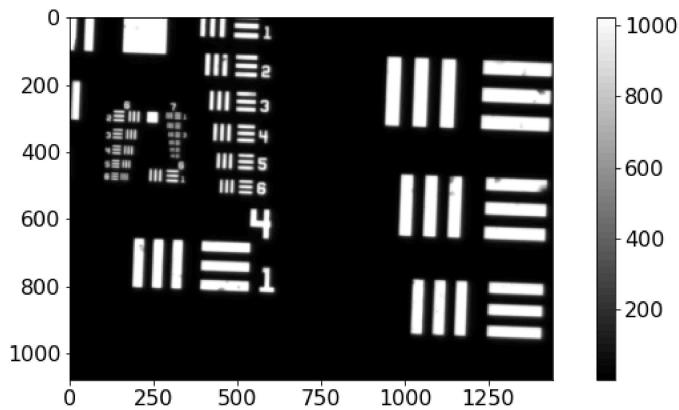
Your images are potentially rotated, flipped, or transposed. Modify your script by applying `np.rot90(img)`, `np.transpose(img)`, `np.fliplr(img)`, and/or `np.flipud(img)` commands.

 No description has been provided for this image

In [ ]:

**Data collection:** Collect a series of images at different levels of focus.

1. Which image would you consider is in focus? Show the in-focus below with x, y axis labels (in pixels) and a colorbar.

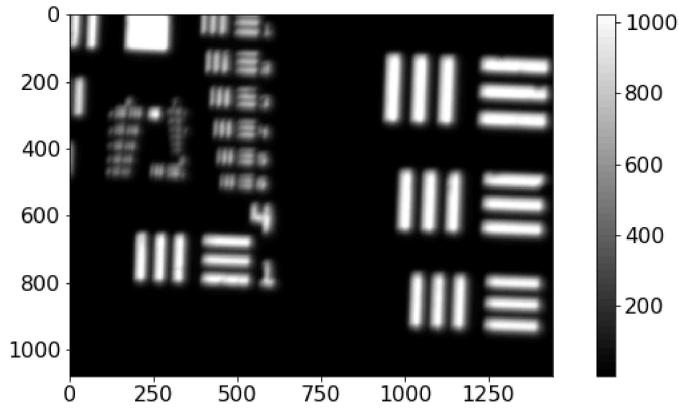


2. Why do you think this image is in focus? (Be specific and describe what features you are looking at. Provide a zoomed image to illustrate these features.)

The light rays at each object point convert on the imaging plane. The image is sharp and has crisp boundaries. The aperture is set at the max to allow for the highest resolution image.

3. Pick an out-of-focus image and show it below. How is this image different to the in-focus image you showed above?





The rays from a single point on the object are spread out when they reach the camera lens, creating a blurry image.

## Adjusting image signal levels

The data that is acquired by your camera is 10-bit. This means that the camera is capable of reporting values between 0 and 1023. Generally, you will want to collect data that covers most of this range. The following terms will be important to know:

- **Underexposure** – A lack of integrated signal at the detector. Most of the values will be on the low end of the 0-1023 range. Python will automatically scale the colorbar to range between the minimum and maximum values, so the image won't necessarily appear dark; however, increased noise will be apparent.
- **Saturation** - The camera cannot represent signal levels above 1023. Thus, signal levels above that range will "saturate" (or be limited) at 1023. This results in a loss of contrast for any image regions above that level.
- **Overexposure** – Too much signal is also a problem for image quality. Generally overexposed images will exhibit saturation in some part of the image.
- **"Good" exposure** – This is somewhat qualitative; however, a "good" exposure should not be underexposed (too noisy) and should not have any saturated regions. e.g. All pixel values fall in the range 0-950.

There are three basic ways to change the signal level in your imaging system:

- **Changing the illumination level** - The lighting module has an adjustment that permits changing the brightness of the LED illumination.
- **Changing the size of the aperture** - An adjustable iris has been installed right above the lens. This aperture controls the diameter of the light beam that is exiting the rear side of the lens. Smaller apertures will reduce signal levels and larger apertures will increase signal levels.
- **Changing the exposure time** – Your camera has the ability to adjust the integration time over which signal is measured for individual images. The current setting can be found by looking at the Python camera object. (The variable `c` in your python script above.) You may set this value using the statement, for example,  
`c.set_exposure(10)`. This sets the exposure to 10 milliseconds. Longer integration times permit accrual of more signal, while shorter exposure times result in less signal. The maximum exposure time is limited by the camera framerate (also adjustable). For example, changing framerate with `c.set_framerate(1)` sets the rate to 1 frame/second permitting exposures up to 1000 ms.

### Exposure and illumination experiments:

Place a biological slide on your microscope. Set your LED illumination to the lowest stable setting. Open your aperture to the widest setting. Move the filter wheel to one of

the open (no filter) settings.

Finding a good exposure setting for your image involves finding the maximum values in your image. While one could simply use Python's `np.max()` function, this is sensitive to noise and single "bright" pixels. A better solution is to plot a histogram of pixel values and then select an exposure that yields a minimal number of saturated values. In the following cell, write code that will show an image from the microscope as well as its histogram of pixel values (with 1024 intensity bins).

```
In [ ]: c.set_exposure(10)

image_10bit = c.capture().copy()

image_8bit = cv2.normalize(image_10bit, None, 0, 255, cv2.NORM_MINMAX, dtype=cv2.CV_8U)
cv2.imshow('Microscope Image', image_8bit)
cv2.waitKey(0)
cv2.destroyAllWindows()

for i in range(10):

    exposure = 5 * i
    c.set_exposure(exposure)

    sleep(5)
    real_exposure = c.get_exposure()
    print(real_exposure)

    image_10bit = c.capture().copy()
    image_8bit = cv2.normalize(image_10bit, None, 0, 255, cv2.NORM_MINMAX, dtype=cv2.CV_8U)

    fig, ax = plt.subplots(1, figsize=(12, 5))
    x = ax.imshow(image_8bit, cmap='gray')
    fig.colorbar(x, ax=ax)
    plt.savefig(f'image_exposure{exposure}.png') # Save the image

    plt.clf()

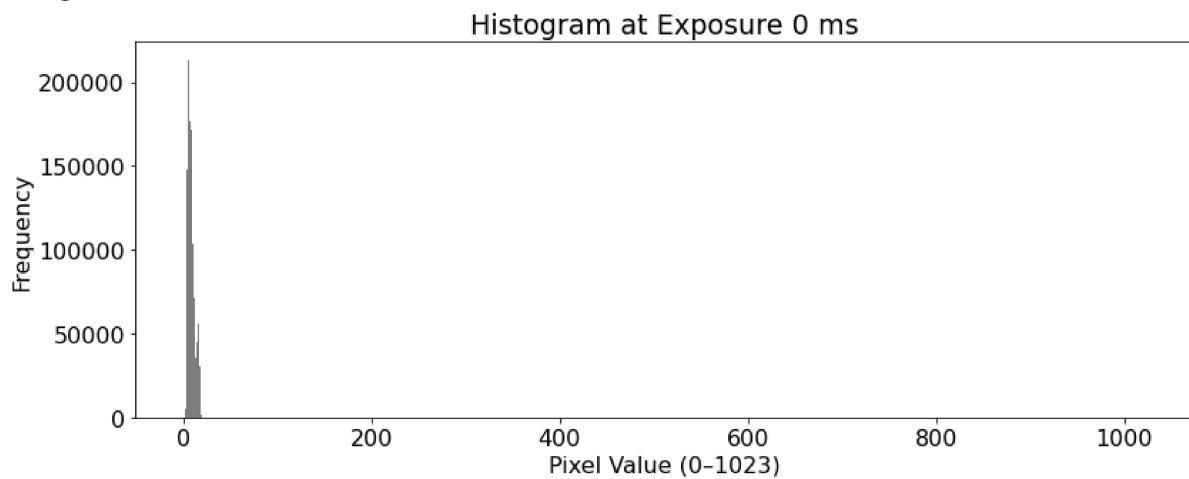
    fig, ax = plt.subplots(figsize=(12, 5))
    ax.hist(image_10bit.flatten(), bins=1024, range=(0, 1023), color='gray')
    ax.set_xlabel('Pixel Value (0-1023)')
    ax.set_ylabel('Frequency')
    ax.set_title(f'Histogram at Exposure {exposure} ms')
    plt.tight_layout()

    plt.savefig(f'histogram_exposure{exposure}.png')
    plt.show()
```

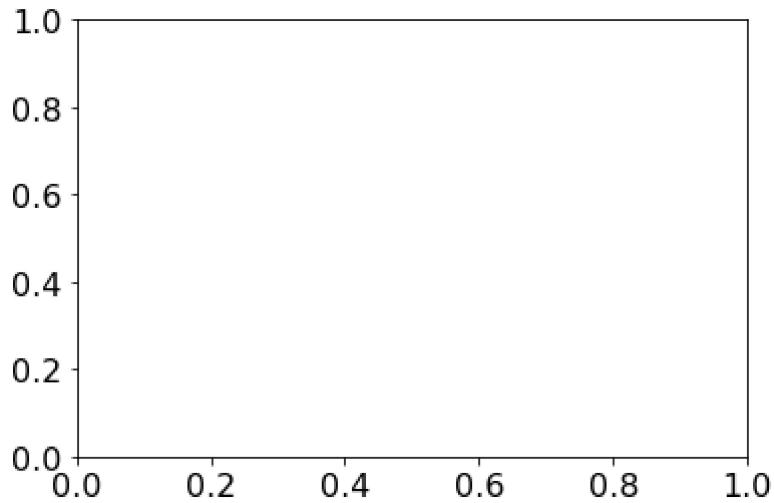
```
plt.clf()  
plt.cla()
```

0.46

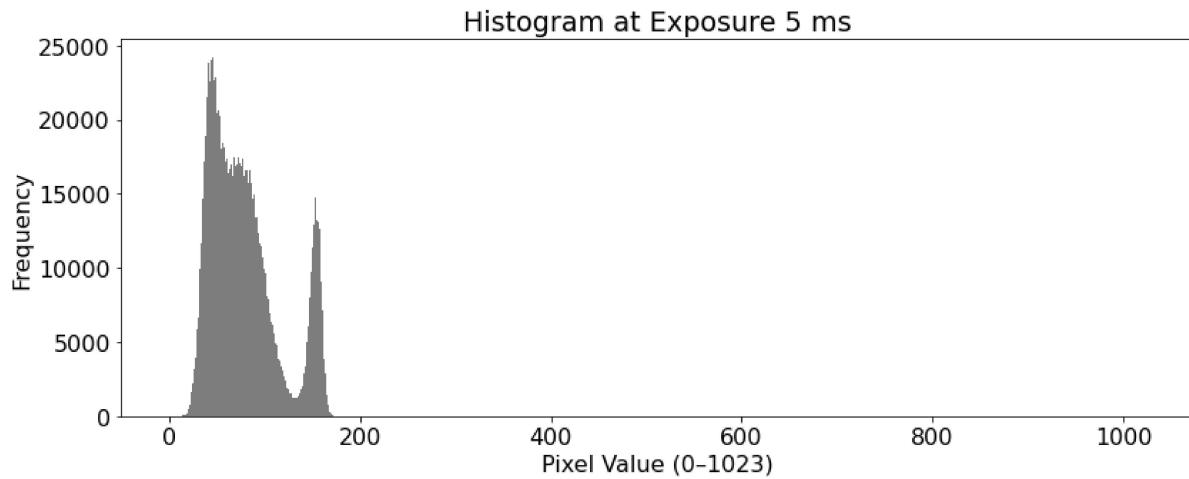
&lt;Figure size 864x360 with 0 Axes&gt;



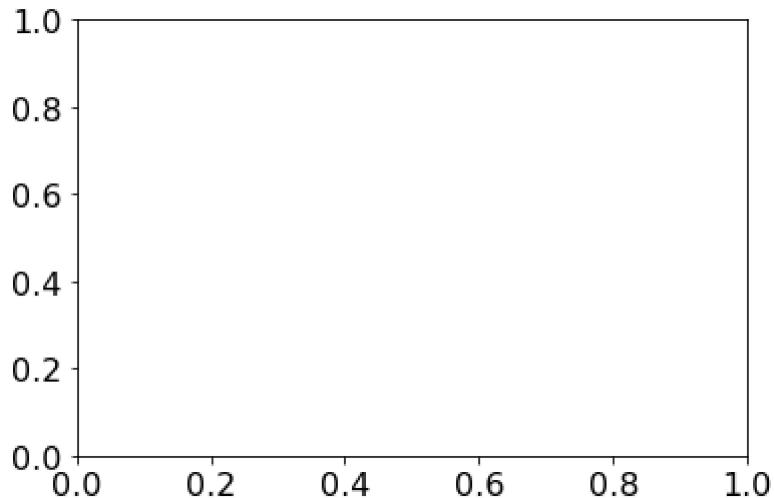
4.916



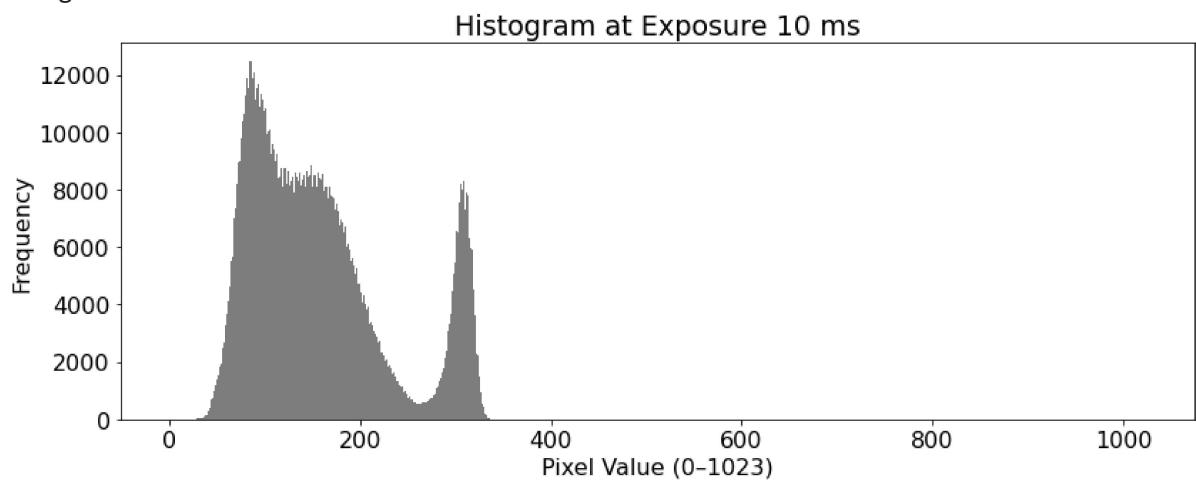
&lt;Figure size 864x360 with 0 Axes&gt;



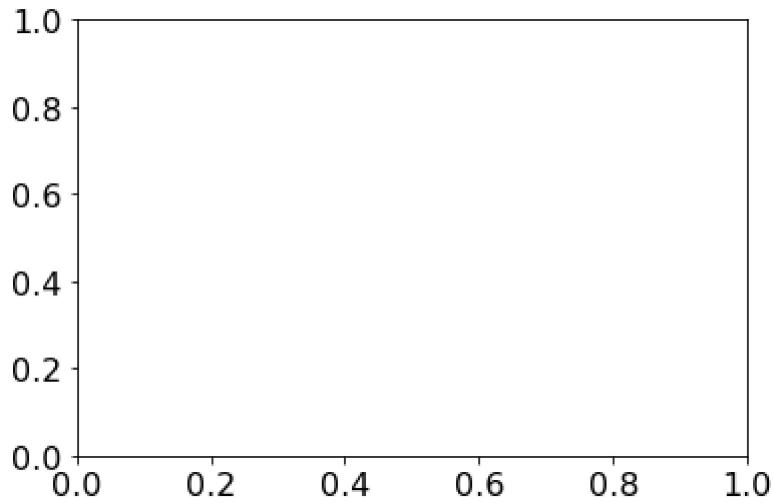
9.818



<Figure size 864x360 with 0 Axes>

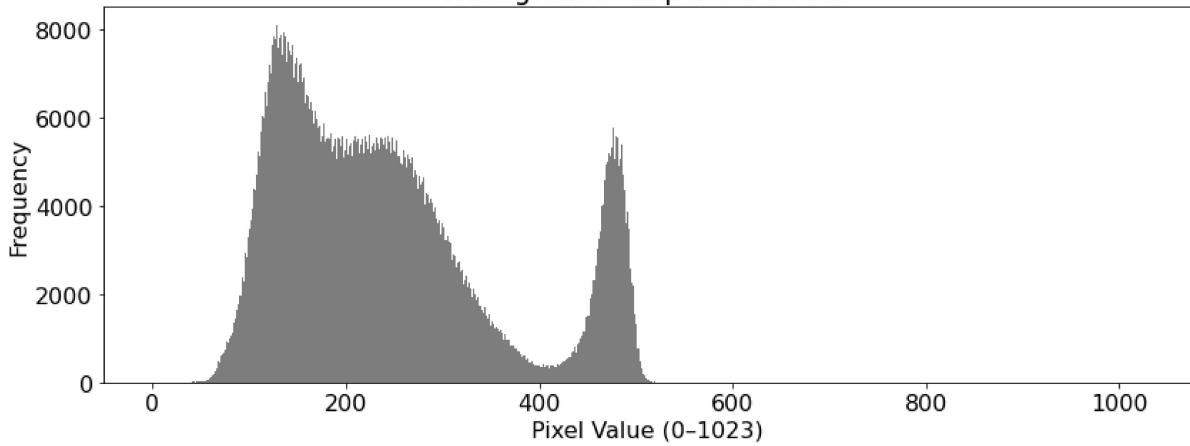


15.165

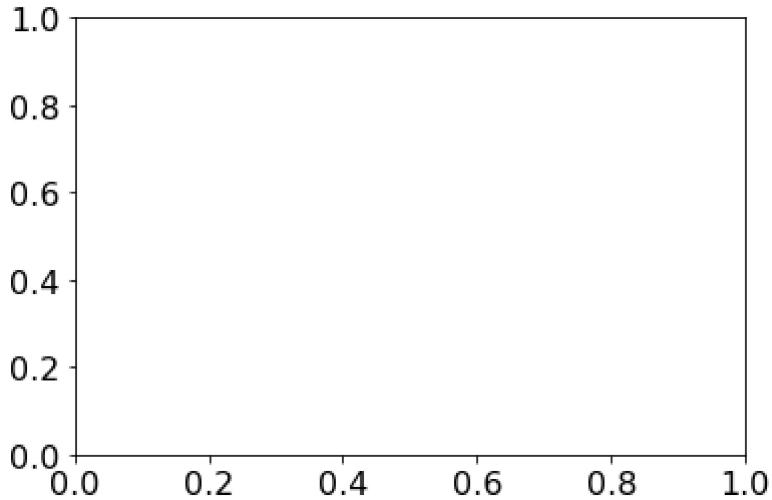


<Figure size 864x360 with 0 Axes>

Histogram at Exposure 15 ms

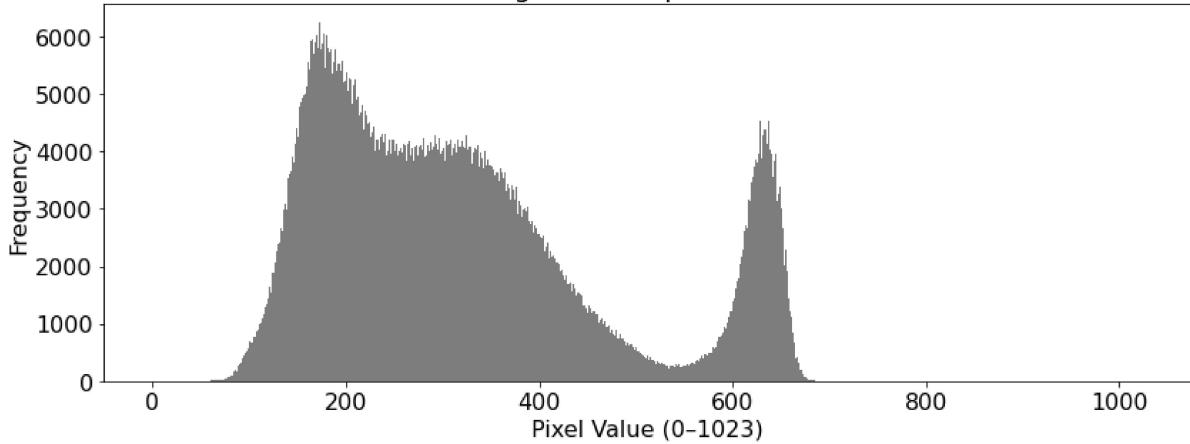


20.067

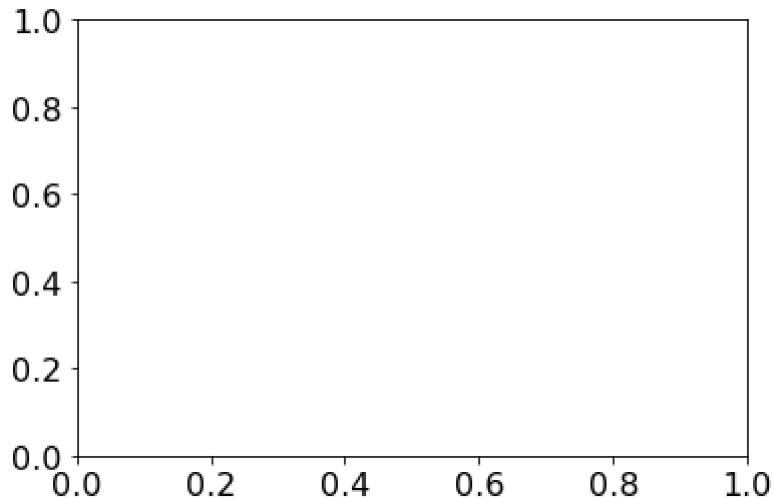


&lt;Figure size 864x360 with 0 Axes&gt;

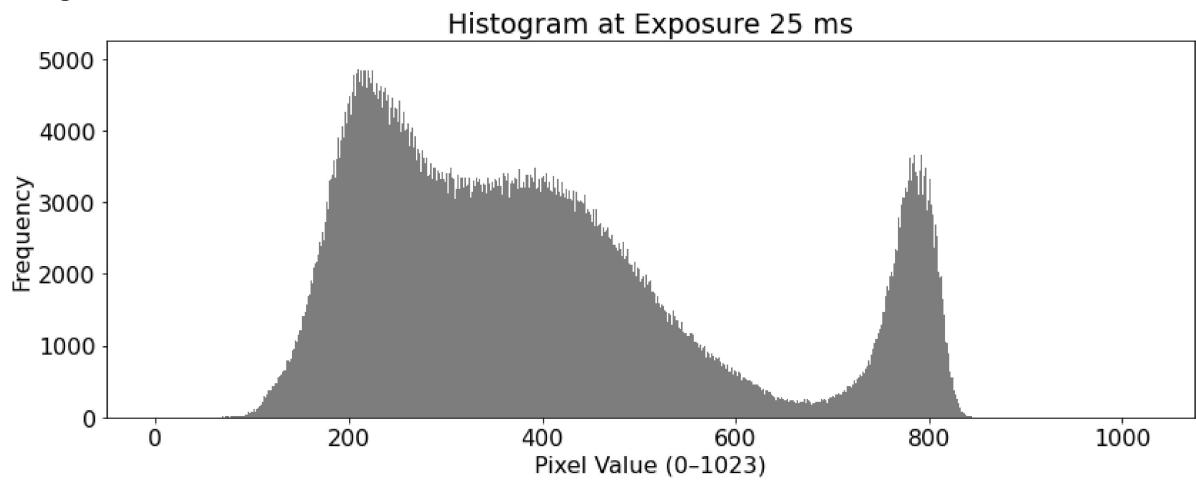
Histogram at Exposure 20 ms



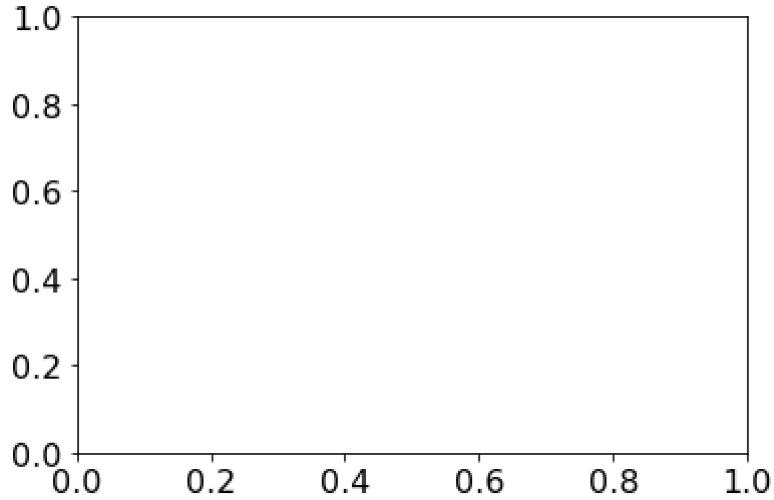
24.969



<Figure size 864x360 with 0 Axes>

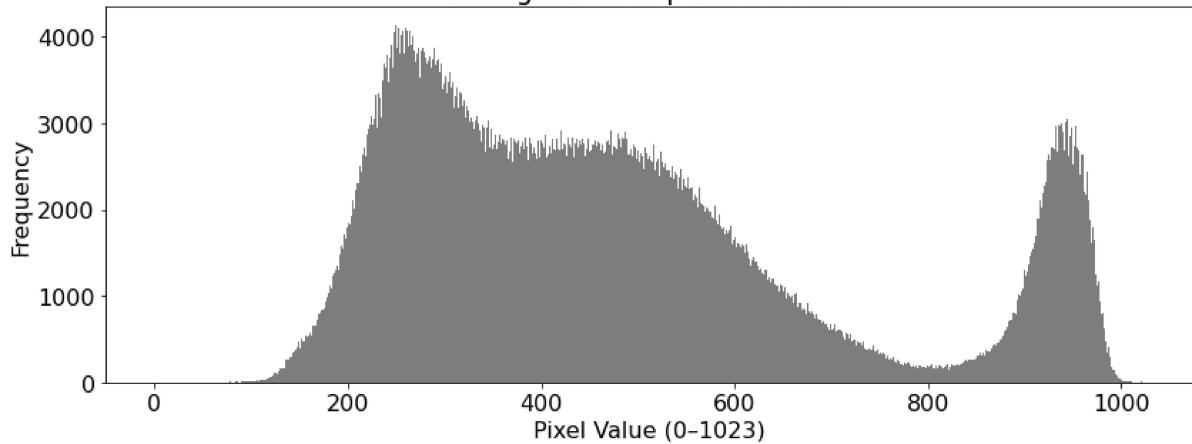


29.871

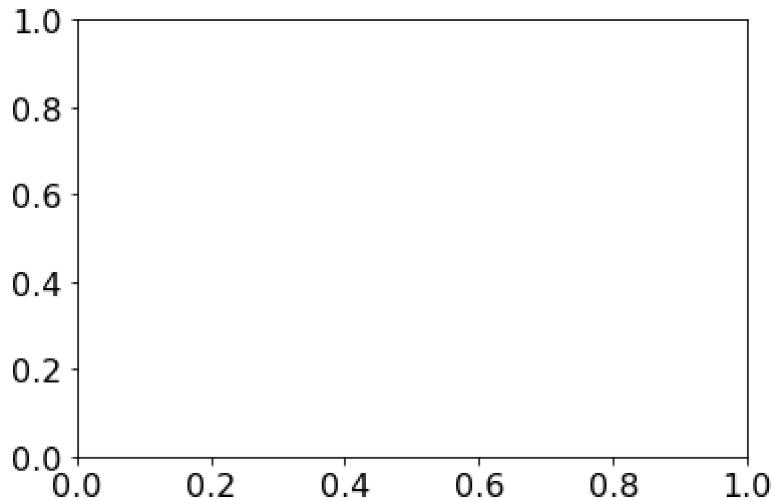


<Figure size 864x360 with 0 Axes>

Histogram at Exposure 30 ms

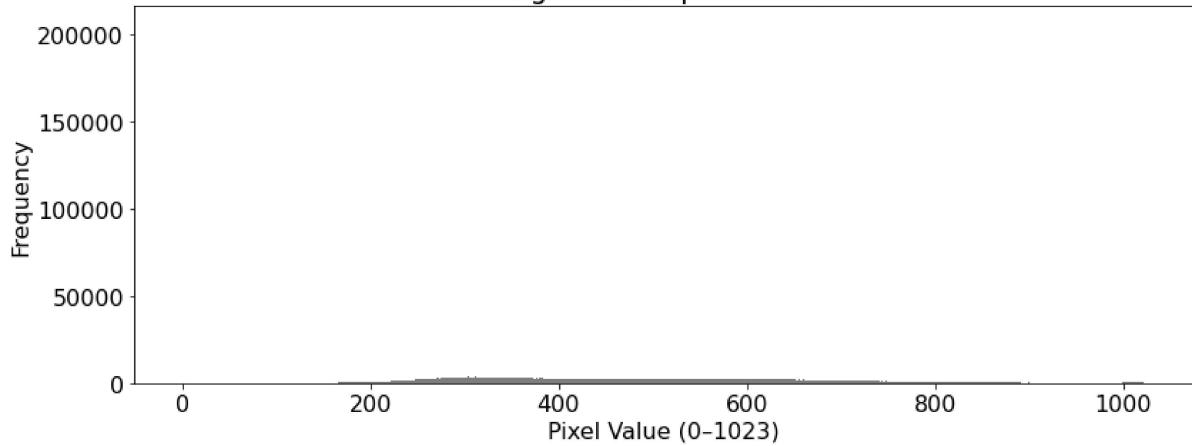


35.218

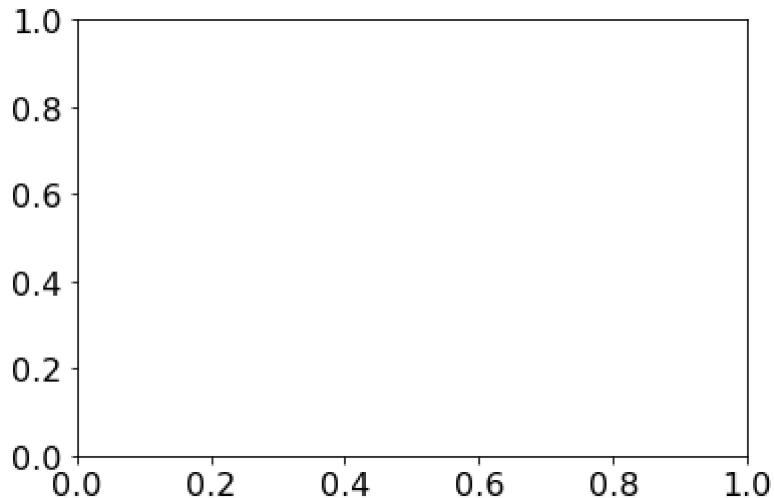


&lt;Figure size 864x360 with 0 Axes&gt;

Histogram at Exposure 35 ms

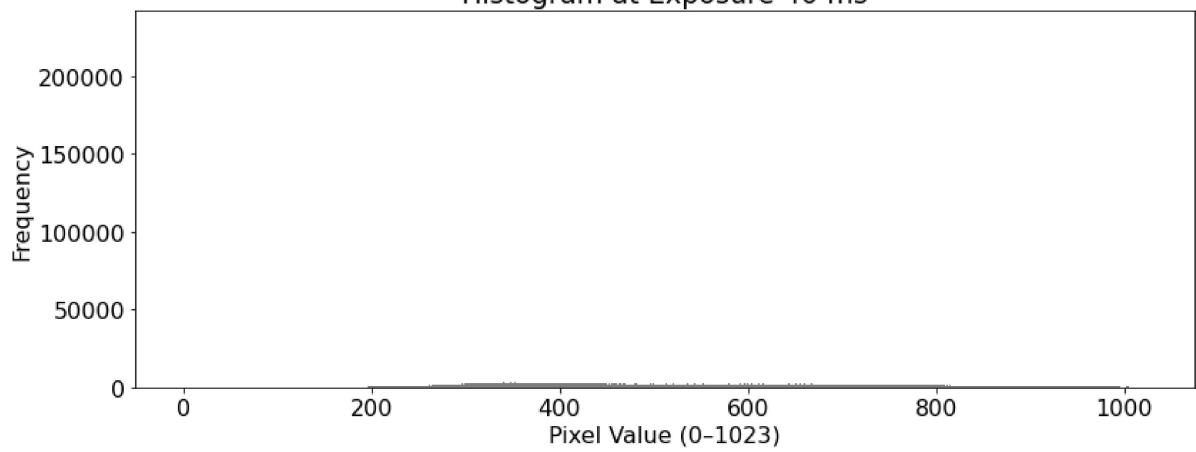


40.12

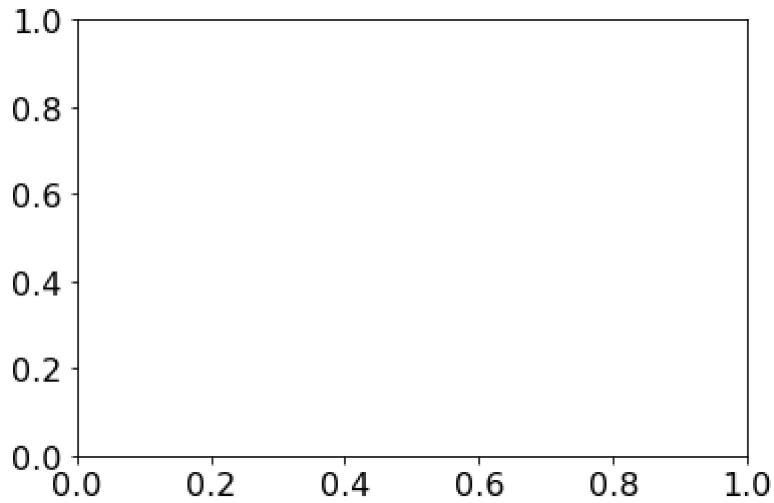


<Figure size 864x360 with 0 Axes>

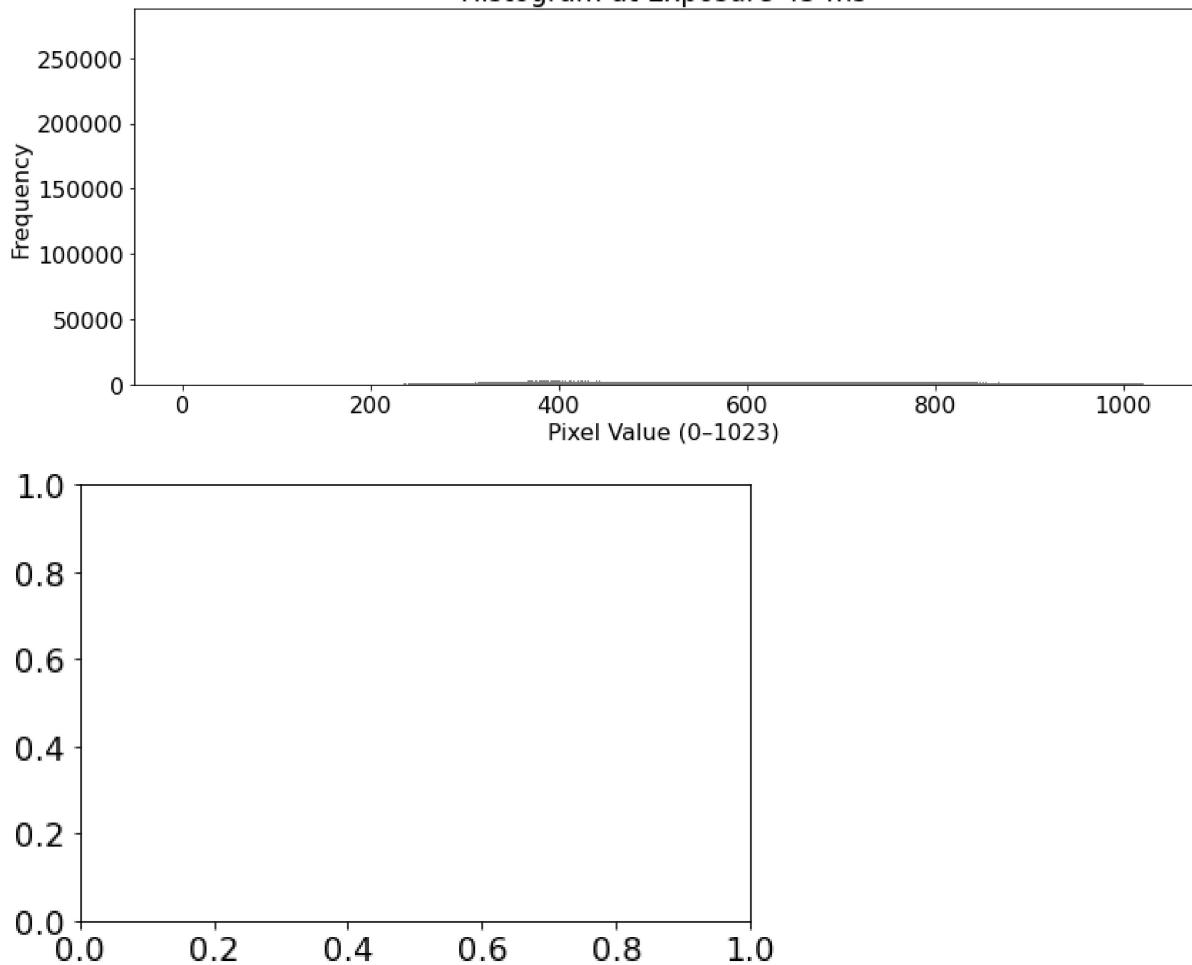
Histogram at Exposure 40 ms



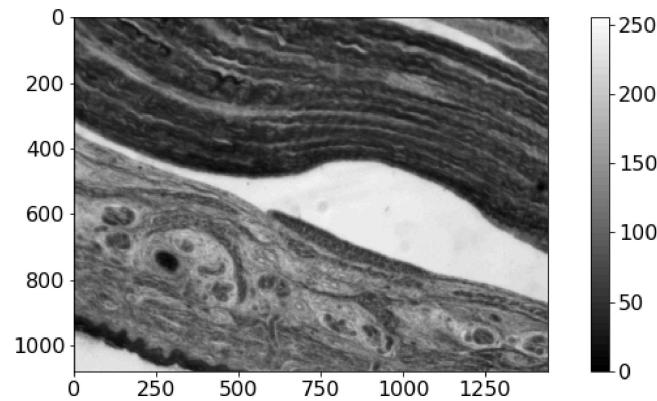
45.022



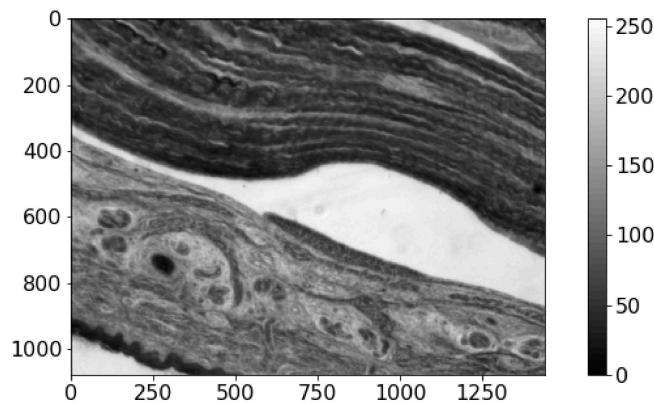
<Figure size 864x360 with 0 Axes>

**Histogram at Exposure 45 ms**

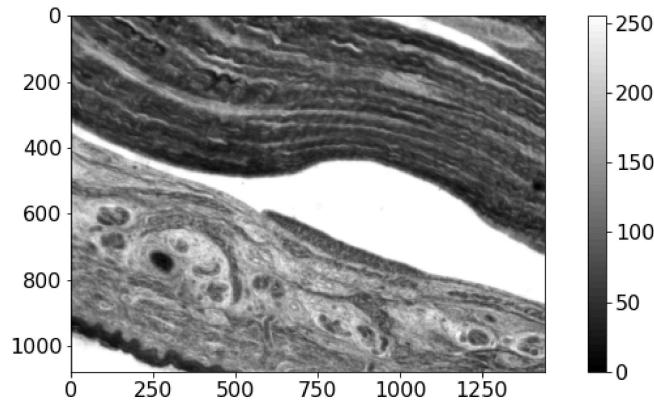
We chose exposure 30 ms based on the histogram plots (max frequency).



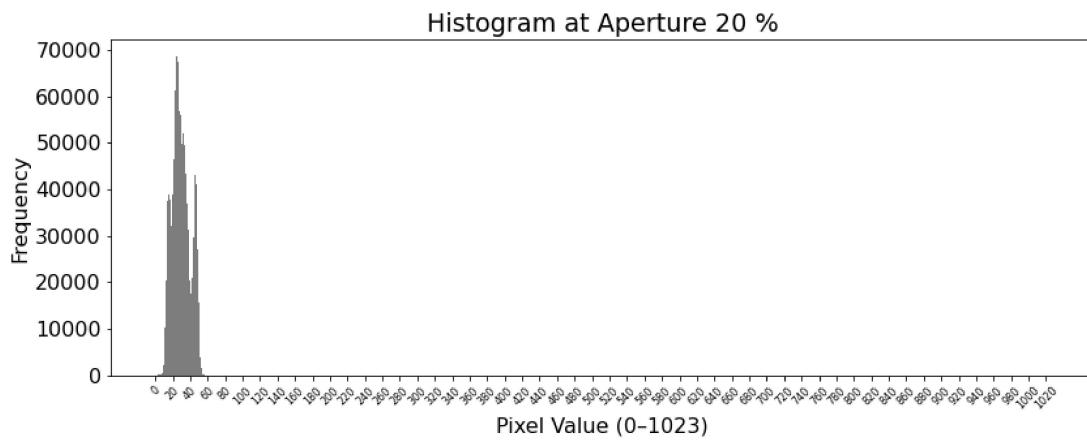
^exposure 20 ms



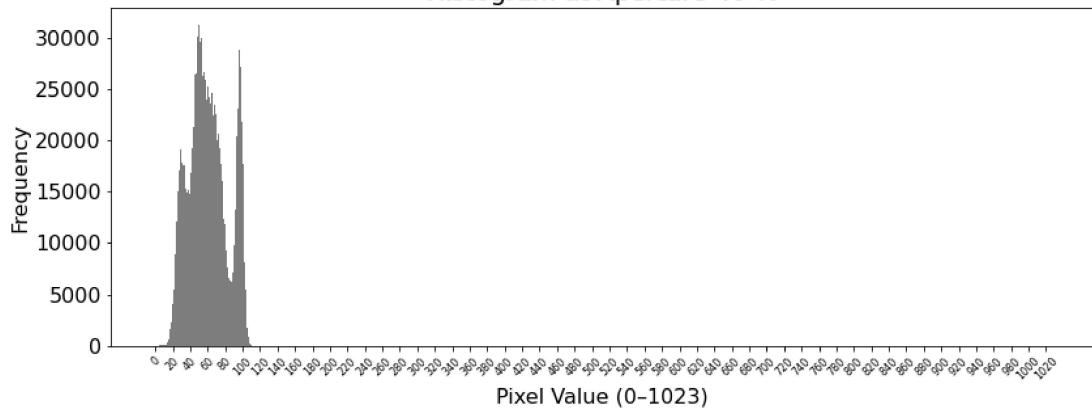
^exposure 30 ms



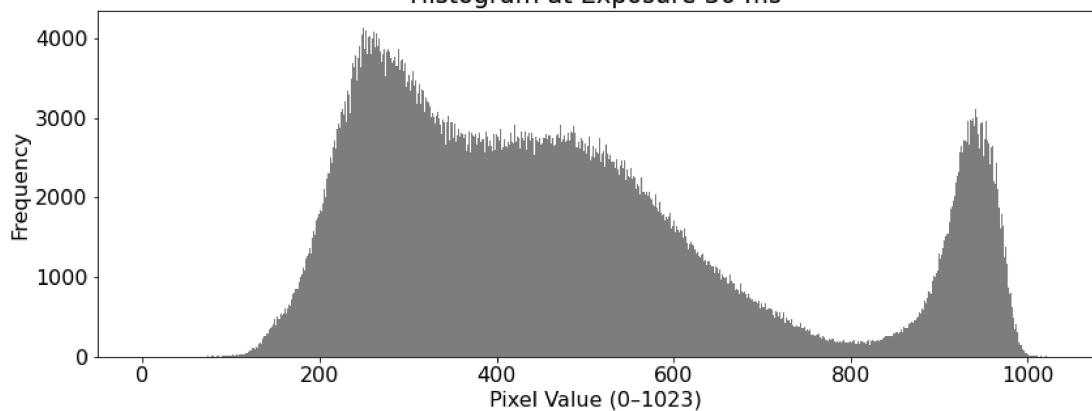
^exposure 35 ms



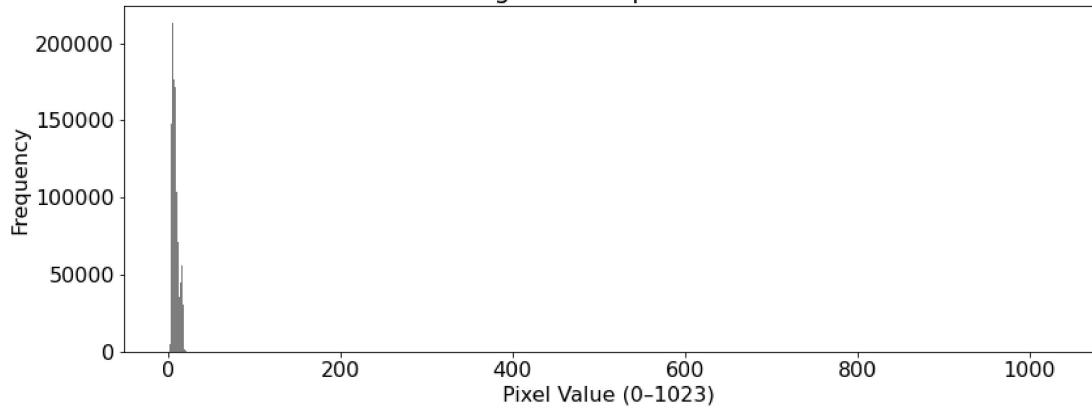
Histogram at Aperture 40 %



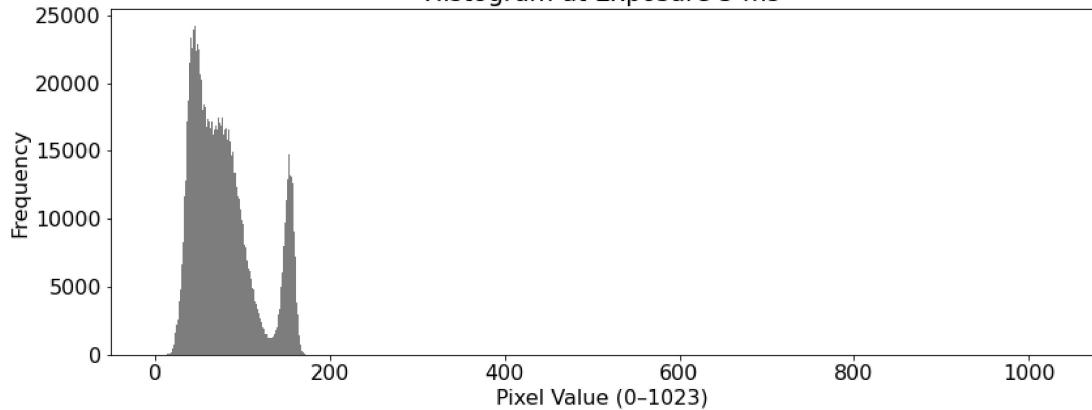
Histogram at Exposure 30 ms



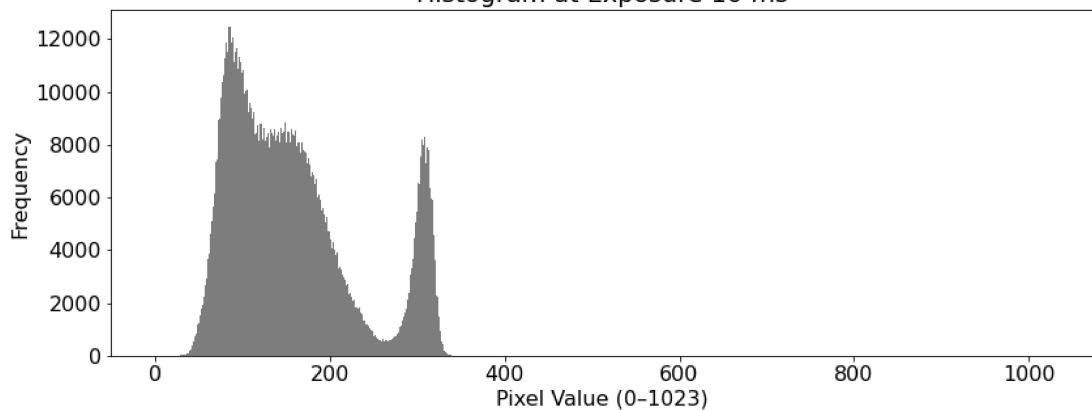
Histogram at Exposure 0 ms



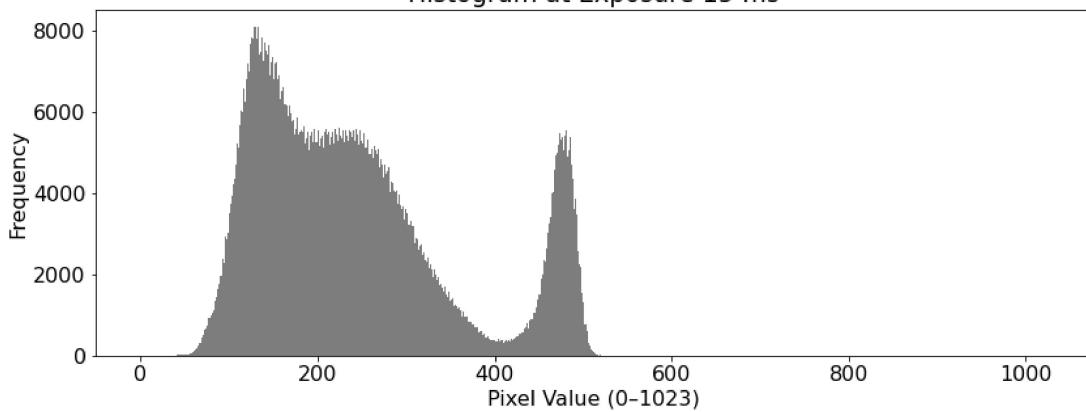
Histogram at Exposure 5 ms



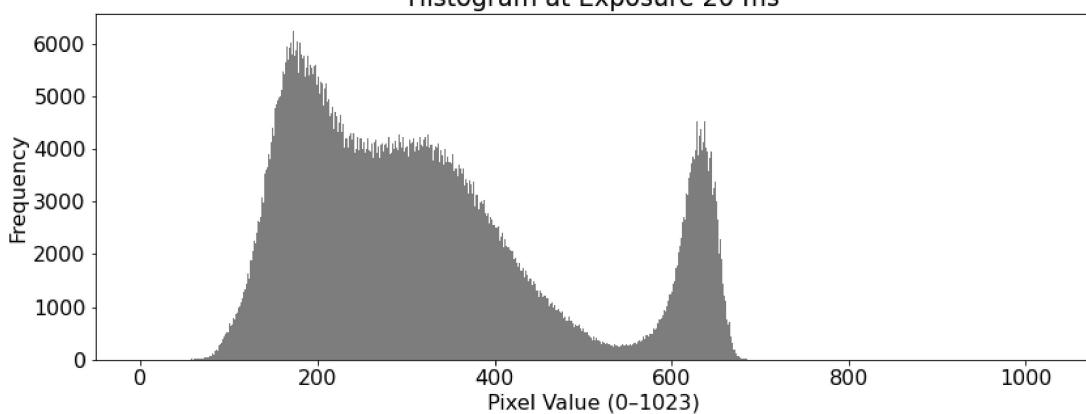
Histogram at Exposure 10 ms



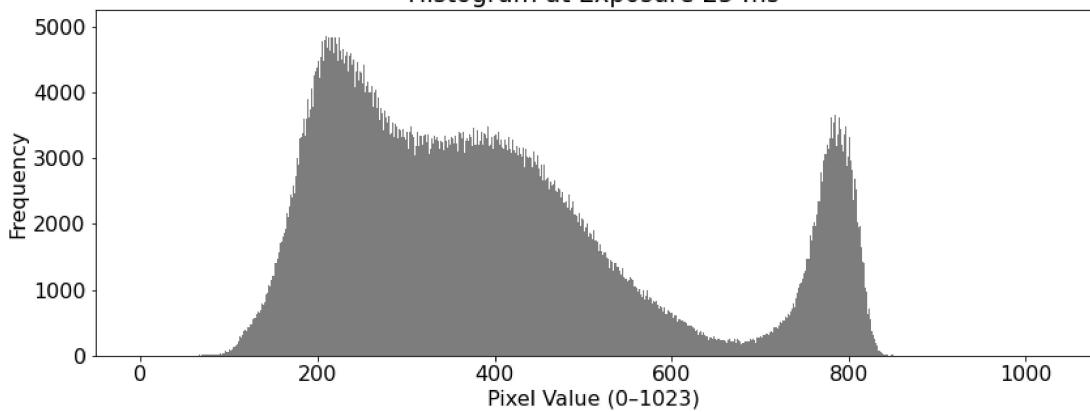
Histogram at Exposure 15 ms



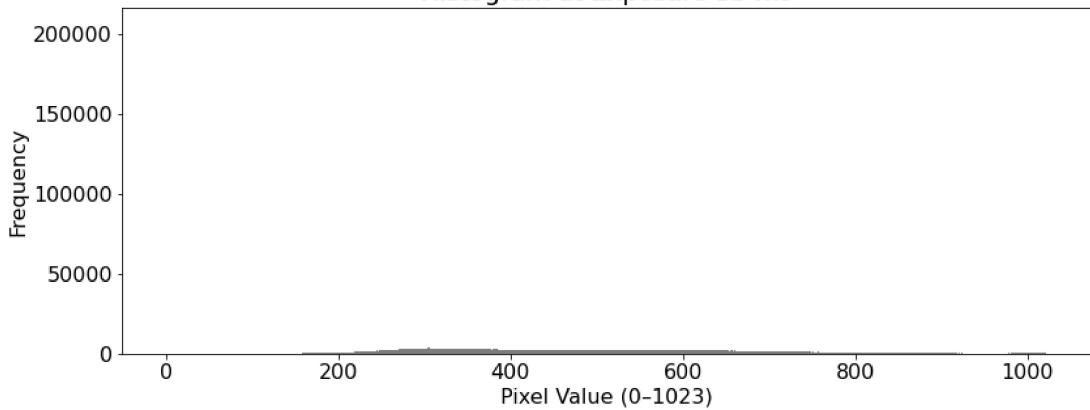
Histogram at Exposure 20 ms



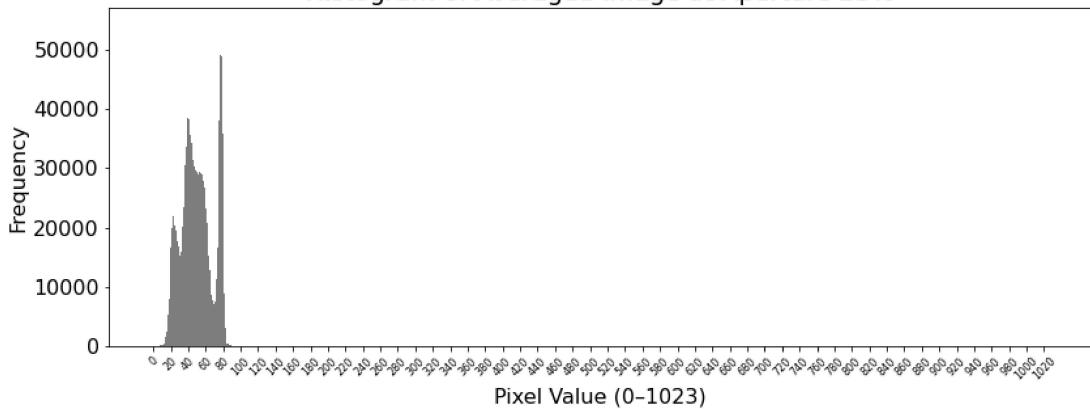
Histogram at Exposure 25 ms



Histogram at Exposure 35 ms



Histogram of Averaged Image at Aperture 23%



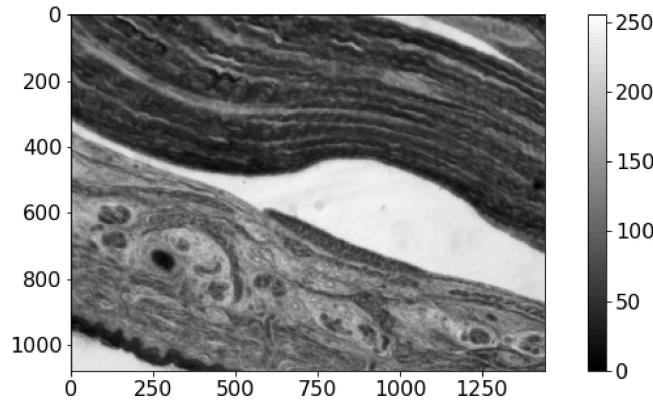
**Data collection:** Collect a series of images at different levels of exposure.

DOUBLE CHECK Use `sleep(5)` and `print(c.get_exposure())` after setting exposure to ensure the correct exposure level is reached.

refer to code under "Exposure and illumination experiments:"

1. What is a good exposure setting for this scenario?  
Show an image of the slide at these settings and paste it below:

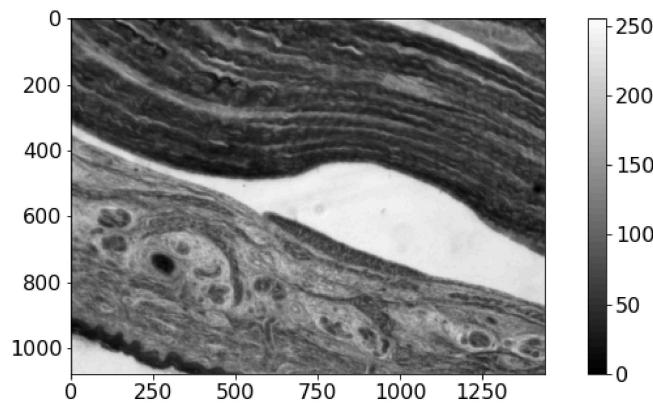
exposure 30 ms



2. Justify why this image represents a good exposure (include a gray level histogram as part of your discussion):

This image represents a good exposure because it has a minimal number of saturated values (no spikes at the high end 1023 which indicates saturation). There are also no significant values clustered near zero, which means underexposure. This balance means the image captures a wide dynamic range, which maximizes contrast and detail without clipping. The visual also supports the histogram.

3. Show an image of the slide where there is obvious saturation:



4. How can you tell that this image is saturated? Why is it saturated?

The image is saturated because the histogram shows a sharp peak at the maximum pixel value (1023). This indicates that some regions in the image received more signal than the camera could accurately represent.

5. What is the relationship between exposure and maximum pixel value? Plot this for the data we collected and describe.

As exposure time increases, the maximum pixel value also increases. This is roughly represented by a non-linear curve that flattens once saturation begins. Initially, there should be an almost linear relationship where more exposure results in more light collected and higher pixel values. However, once the brightest regions reach the 1023 limit, increases in exposure will not raise the max value, and the image becomes overexposed.

## Small aperture and image averaging

**Data collection:** Make multiple small aperture image acquisitions

1. Close the aperture until your maximum image value is 80-100.

Show a single image of the slide at these settings and paste it below:

```
In [ ]: c.set_exposure(30)

i=0
while True:
    cv2.imshow('Camera', cv2.normalize(c.capture().copy(), None, 0, 255, cv2.NORM_
    if cv2.waitKey(30) & 0xFF == ord('s'):
        aperture = 20*i
        image_10bit = c.capture().copy()
        image_8bit = cv2.normalize(image_10bit, None, 0, 255, cv2.NORM_MINMAX, dtype=cv2.CV_8U)

        fig, ax = plt.subplots(1, figsize=(12, 5))
        x = ax.imshow(image_8bit, cmap='gray')
        fig.colorbar(x, ax=ax)
        plt.savefig(f'image_aperture{aperture}.png')

    plt.clf()

    image_10bit = c.capture().copy()
    image_8bit = cv2.normalize(image_10bit, None, 0, 255, cv2.NORM_MINMAX, dtype=cv2.CV_8U)

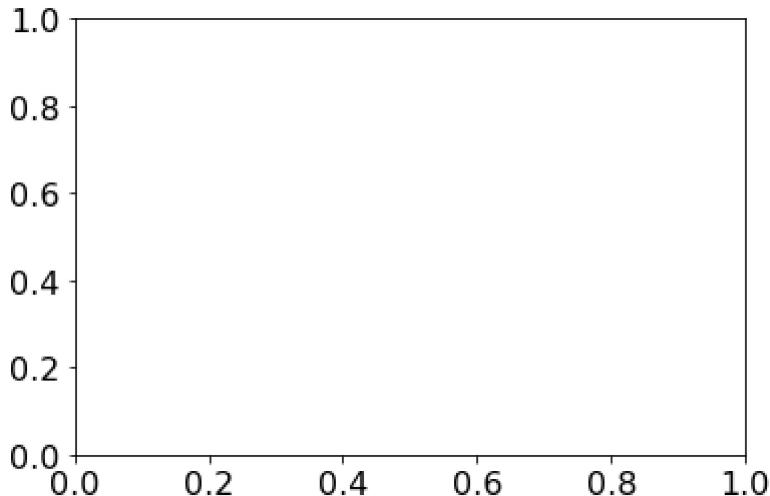
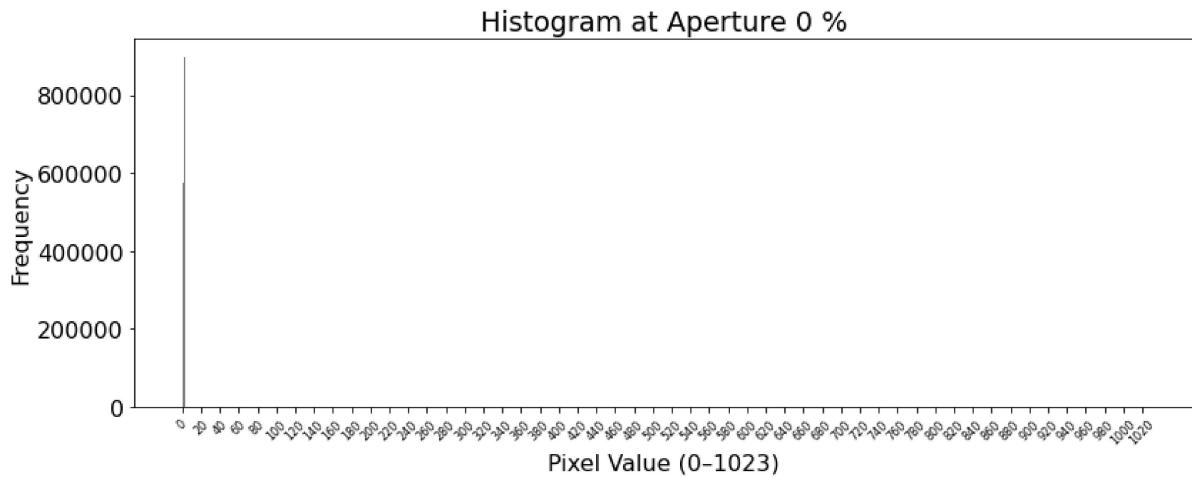
    fig, ax = plt.subplots(figsize=(12, 5))
    ax.hist(image_10bit.flatten(), bins=1024, range=(0, 1023), color='gray')
```

```
ax.set_xticks(np.arange(0, 1023, 20))
plt.xticks(rotation=45, fontsize=8)
ax.set_xlabel('Pixel Value (0-1023)')
ax.set_ylabel('Frequency')
ax.set_title(f'Histogram at Aperture {aperture} %')
plt.tight_layout()

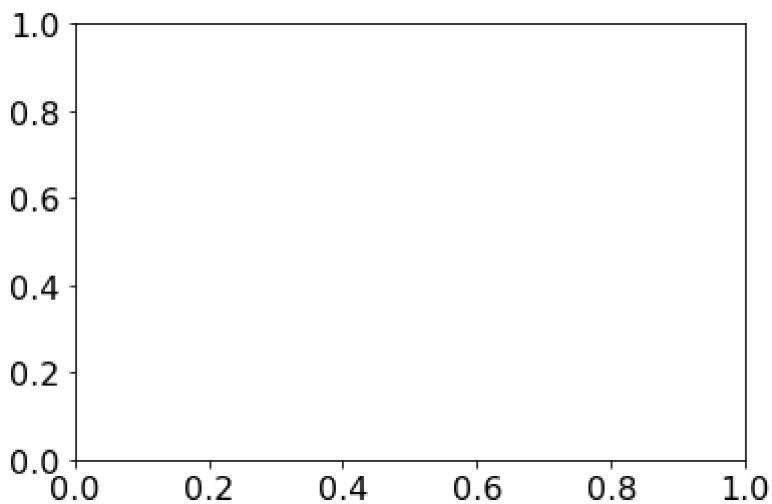
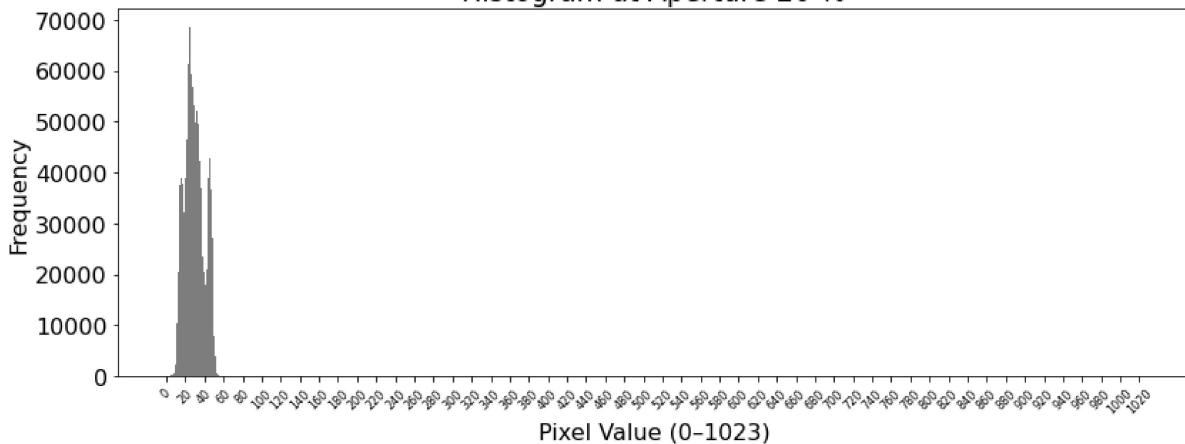
plt.savefig(f'histogram_aperture{aperture}.png')
plt.show()

plt.clf()
plt.cla()
i += 1
if cv2.waitKey(30) & 0xFF == ord('q'):
    break
cv2.destroyAllWindows()
```

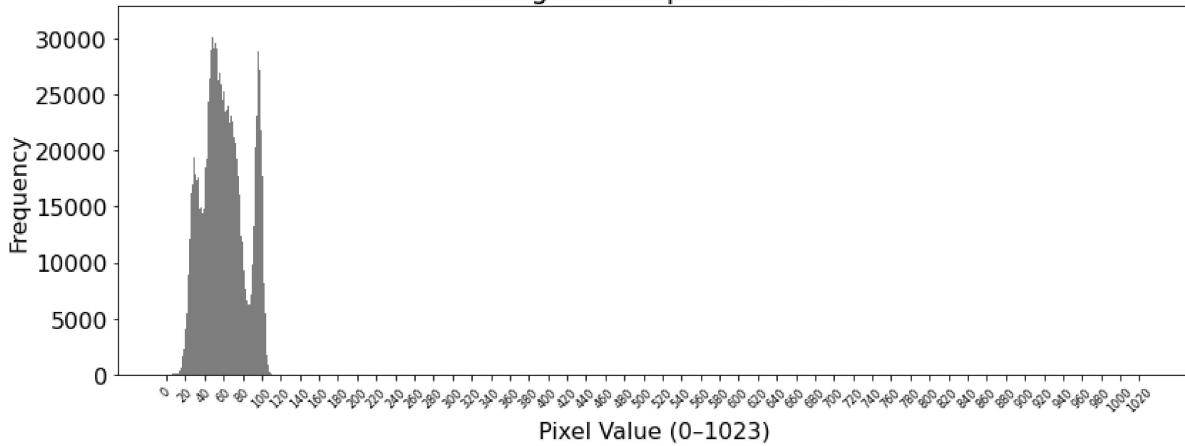
&lt;Figure size 864x360 with 0 Axes&gt;

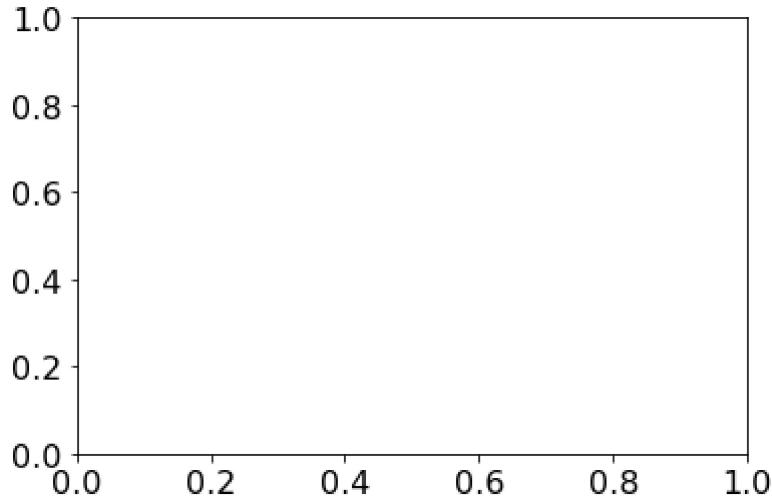


&lt;Figure size 864x360 with 0 Axes&gt;

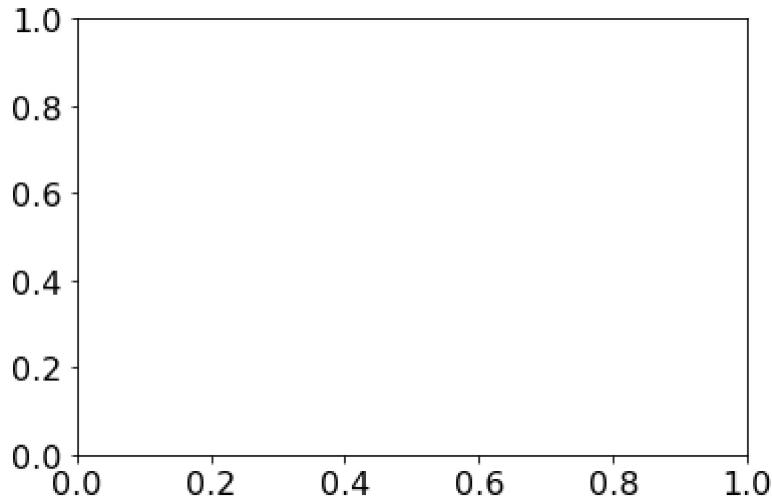
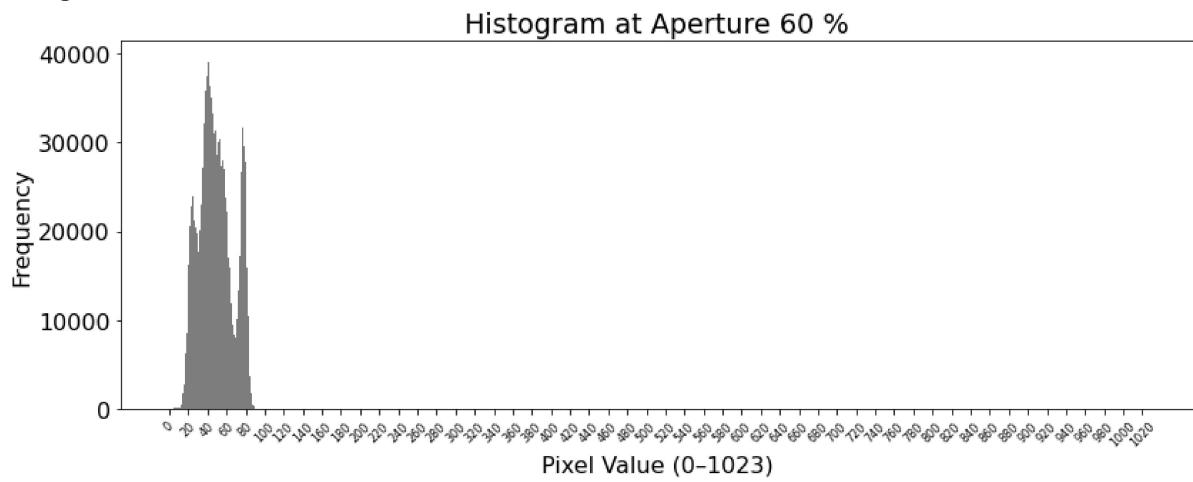
**Histogram at Aperture 20 %**

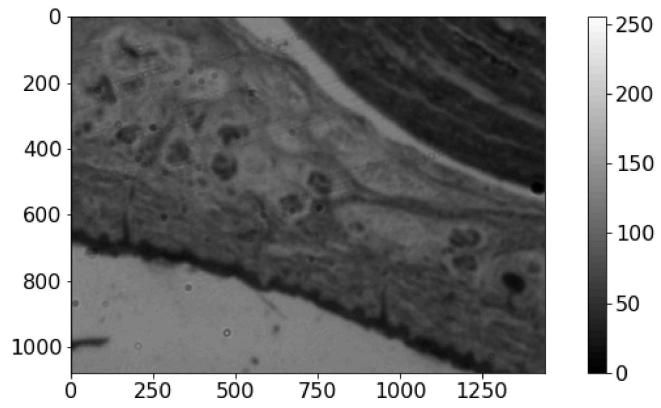
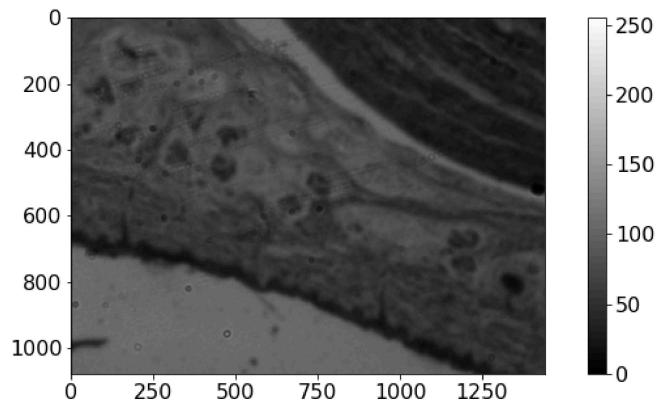
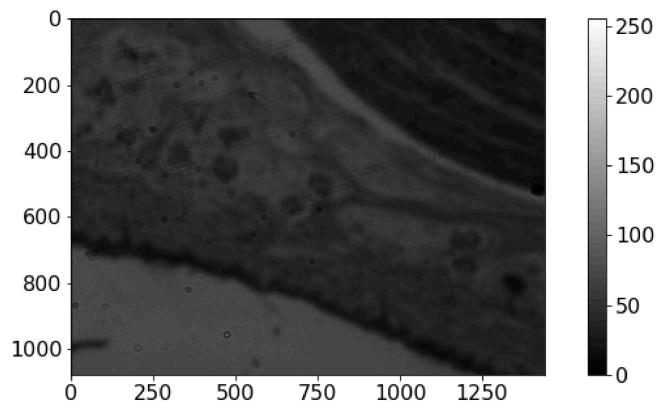
&lt;Figure size 864x360 with 0 Axes&gt;

**Histogram at Aperture 40 %**

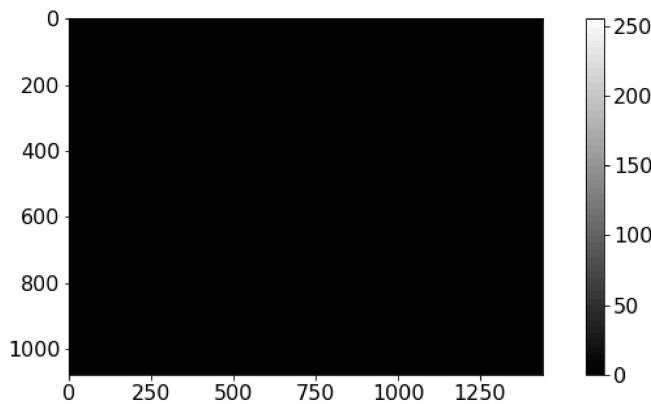


<Figure size 864x360 with 0 Axes>





above image order: aperture level 20% -> 23% -> 25%



aperture level: 0%

2. How is this image different than the good image you showed above? Why?

This image is different than the well exposed good image above because this image is darker and noisier. It's darker because the small aperture reduces the amount of light that reaches the camera. Also, the signal is low so the random sensor noise becomes more visible, especially in darker regions.

3. Show an average of all the images that were acquired with the small aperture:

4. What happens when you average many of the images with small image values? Why?

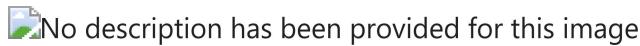
When you average a bunch of low-signal images, the random noise kind of cancels itself out because it's different in each frame. But the actual image — the part that stays the same — gets clearer. So the final image ends up looking smoother and less noisy, even though each individual one was really dim. Basically, averaging helps boost the signal and reduce the noise.

## Moving the filter wheel

Move the filter wheel to filter 1, 2, and 3 respectively. There are three color filters in the filter wheel: 460 nm (blue), 540 nm (green), and 600 nm (orange). You will obtain one of the following four slides for imaging.

1. What do we need to do to get a good image?

To get a good image, we need to adjust the camera settings like exposure time or increase the light intensity because the filters block out a lot of light. We can also adjust the image after by changing the intensity bar afterwards to brighten the image.



**Data collection:** Image the biological slide with different color filters

2. Show the image data below. What effects do the filters have on the images of each slide? Explain why.

```
In [ ]: i=0
while True:
    cv2.imshow('Camera', cv2.normalize(c.capture().copy(), None, 0, 255, cv2.NORM_
    if cv2.waitKey(30) & 0xFF == ord('s'):
        image_10bit = c.capture().copy()
        image_8bit = cv2.normalize(image_10bit, None, 0, 255, cv2.NORM_MINMAX, dty

        fig, ax = plt.subplots(1, figsize=(12, 5))
        x = ax.imshow(image_8bit, cmap='gray')
        fig.colorbar(x, ax=ax)
        plt.savefig(f'image_filter{i}')

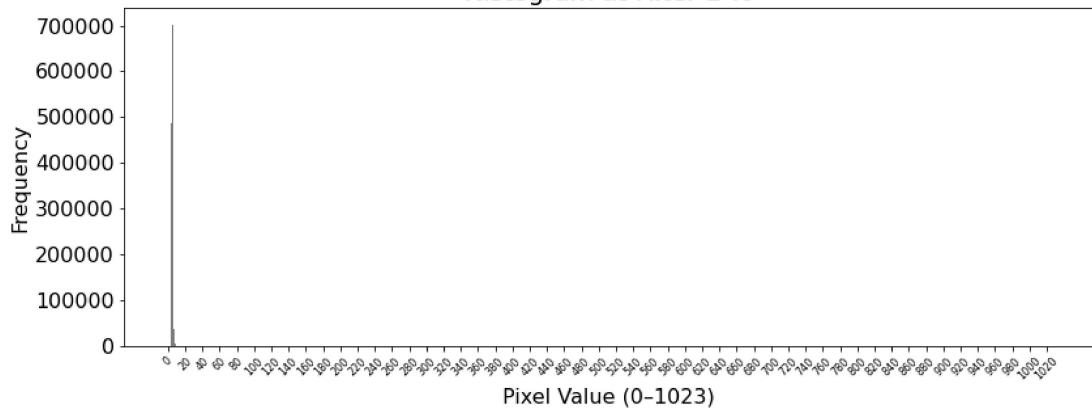
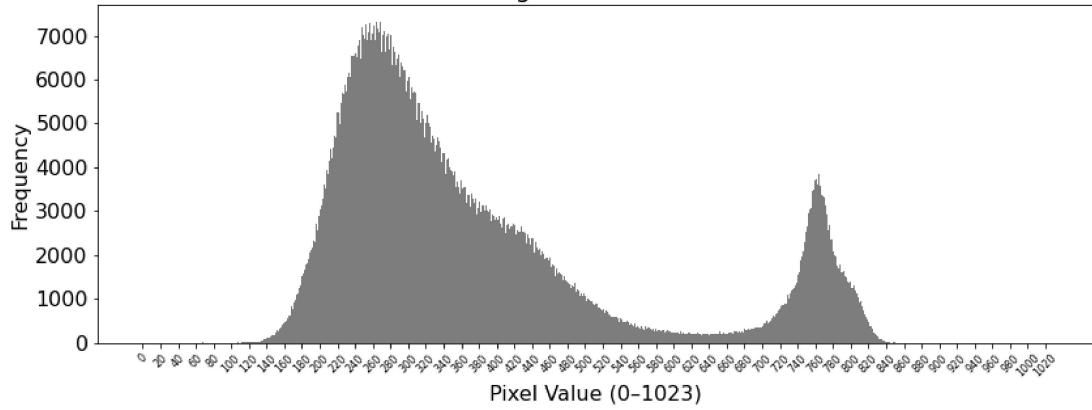
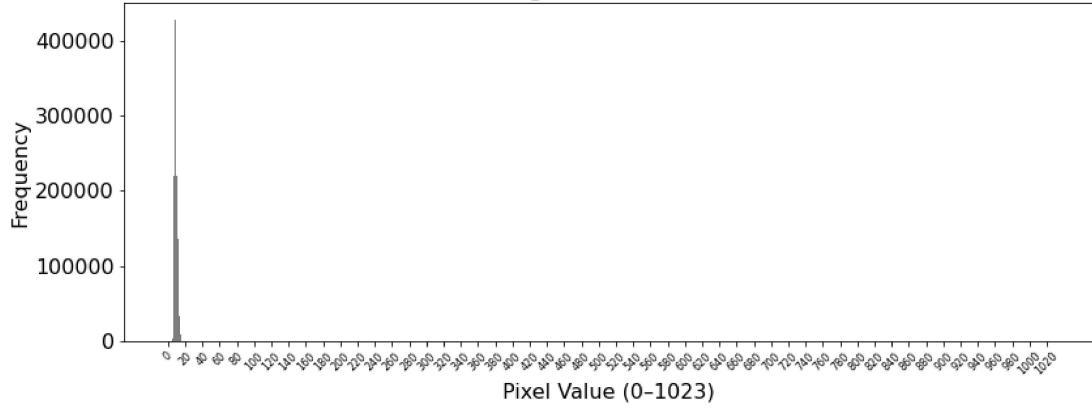
    plt.clf()

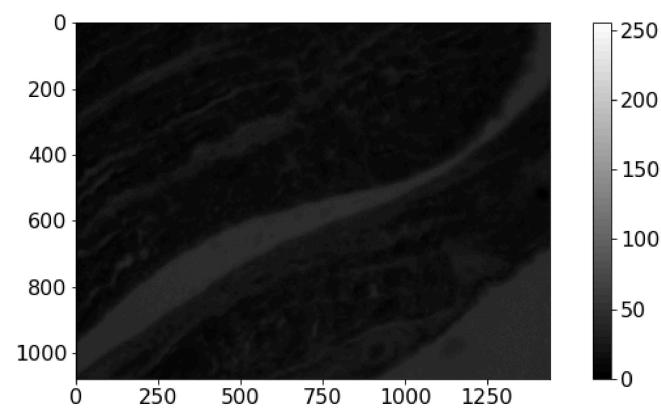
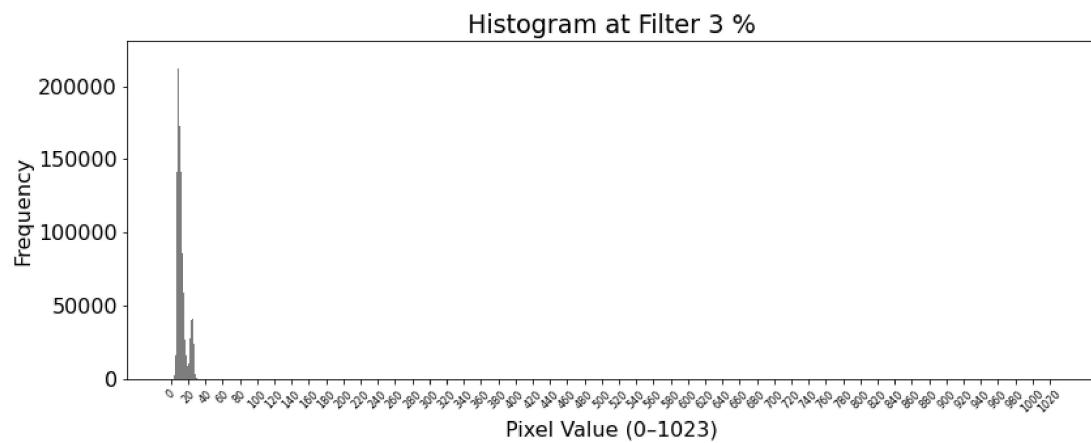
    image_10bit = c.capture().copy()
    image_8bit = cv2.normalize(image_10bit, None, 0, 255, cv2.NORM_MINMAX, dty

    fig, ax = plt.subplots(figsize=(12, 5))
    ax.hist(image_10bit.flatten(), bins=1024, range=(0, 1023), color='gray')
    ax.set_xticks(np.arange(0, 1023, 20))
    plt.xticks(rotation=45, fontsize=8)
    ax.set_xlabel('Pixel Value (0-1023)')
    ax.set_ylabel('Frequency')
    ax.set_title(f'Histogram at Filter {i} %')
    plt.tight_layout()

    plt.savefig(f'histogram_filter{i}.png')
    plt.show()

    plt.clf()
    plt.cla()
    i += 1
if cv2.waitKey(30) & 0xFF == ord('q'):
    break
cv2.destroyAllWindows()
```

**Histogram at Filter 1 %****Histogram at Filter 0 %****Histogram at Filter 2 %**



^image w/ filter 3

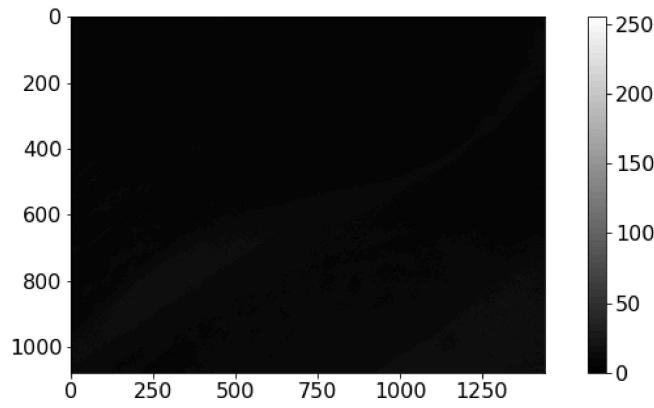
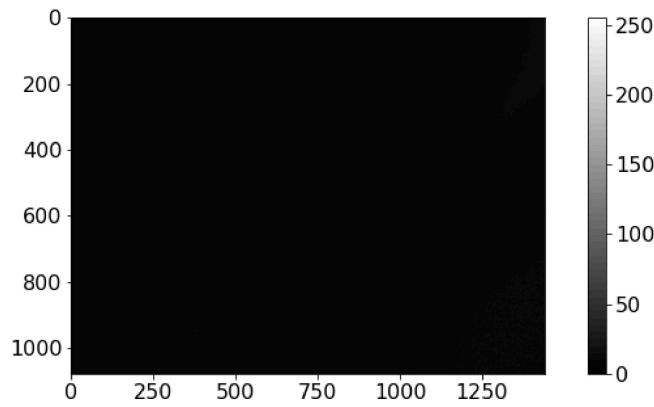
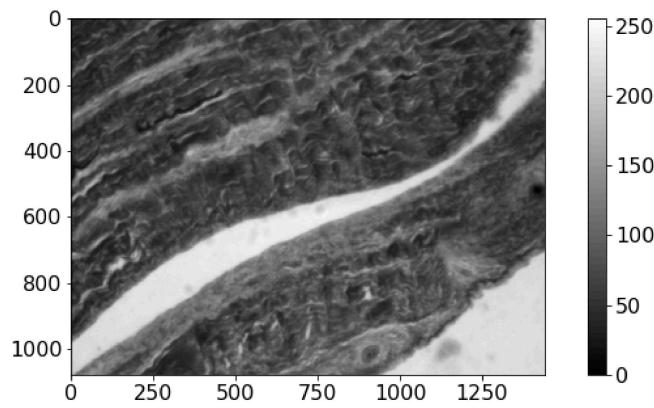


image order: image w/ filter 0, 1, 2, respectively