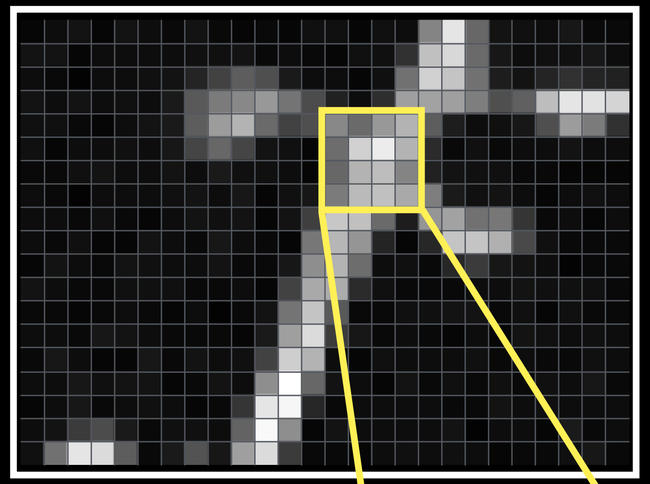
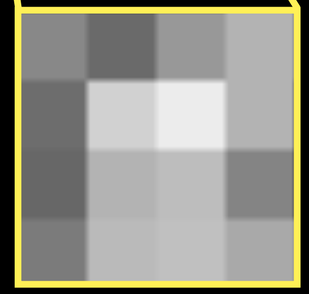
Introduction to Digital Images

Pixel = Picture Element



=

6	13	19	6	19	13	9	19	9	6	9	6	16	16	6	16	13	132	229	103	19	16	13	23	9	9
19	19	6	13	13	13	13	16	16	19	9	13	9	6	16	16	49	192	216	106	23	13	16	16	23	13
13	9	4	13	13	16	19	36	66	93	79	26	13	13	6	16	113	209	196	113	29	19	36	49	36	33
19	13	19	13	16	13	26	89	123	136	152	116	76	33	13	46	159	162	159	126	79	96	189	229	226	212
16	16	9	6	13	19	26	93	156	179	106	66	79	136	106	152	179	93	29	13	16	23	79	156	123	49
16	6	13	13	16	13	23	69	103	69	19	16	6	109	209	236	179	43	9	16	9	13	13	19	13	13
9	9	16	19	13	13	19	13	26	16	16	13	6	103	179	189	132	33	19	16	16	9	9	6	6	6
13	9	4	13	13	13	16	19	13	23	6	16	23	123	186	192	169	126	26	16	19	13	6	13	16	13
13	13	9	16	9	6	13	19	16	19	6	19	63	199	192	106	29	149	162	113	119	53	9	13	6	13
13	9	16	6	6	19	13	9	23	13	9	6	119	182	149	36	6	39	196	196	176	73	16	9	9	9
6	19	13	9	19	16	13	13	19	9	9	23	142	179	109	13	16	9	39	59	23	19	13	4	9	9
19	13	9	9	16	16	16	9	9	13	6	66	169	172	43	16	9	9	9	13	13	19	16	16	16	9
9	9	6	9	13	9	6	13	4	9	19	116	196	89	9	9	16	16	19	19	9	16	6	16	9	9
13	13	9	23	19	13	9	9	9	6	26	159	219	59	23	9	13	9	6	13	6	19	16	13	16	13
9	23	13	6	6	23	9	19	13	16	66	206	179	13	6	16	13	13	13	16	9	13	9	9	16	13
13	13	23	16	19	19	6	9	19	13	142	255	103	19	13	6	19	9	16	9	16	9	16	13	23	9
6	13	23	9	13	16	13	6	9	53	229	246	39	9	13	13	13	13	9	9	19	13	16	13	13	13
13	19	59	76	26	9	16	16	13	99	249	142	6	19	13	13	13	13	19	4	13	13	6	26	9	13
16	113	229	219	93	9	26	83	23	159	219	59	9	9	6	13	16	13	16	13	6	9	9	16	23	9



=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

A digital image is a matrix of numbers!

Introduction to Digital Images

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

+ 2 =

138	108	154	181
111	211	238	181
105	181	191	134
125	188	194	171

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

- 2 =

134	104	150	177
107	207	234	177
101	177	187	130
121	184	190	167

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

x 2 =

272	212	304	358
218	418	472	358
206	358	378	264
246	372	384	338

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

: 2 =

68	53	76	90
55	105	118	90
52	90	95	66
62	93	96	85

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

+

6	13	19	6
19	19	6	13
13	9	4	13
19	13	19	13

=

142	119	171	185
128	228	242	192
116	188	193	145
142	199	211	182

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

-

6	13	19	6
19	19	6	13
13	9	4	13
19	13	19	13

=

130	93	133	173
90	190	230	166
90	170	185	119
104	173	173	156

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

x

6	13	19	6
19	19	6	13
13	9	4	13
19	13	19	13

=

816	1378	2888	1074
2071	3971	1416	2327
1339	1611	756	1716
2337	2418	3648	2197

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

:

6	13	19	6
19	19	6	13
13	9	4	13
19	13	19	13

=

23	8	8	30
6	11	39	14
8	20	47	10
6	14	10	13

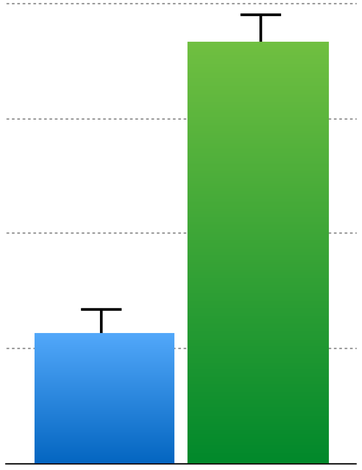
Introduction to Digital Images

Images in publications and presentations should be used to ***communicate*** a finding... not ***be*** the finding

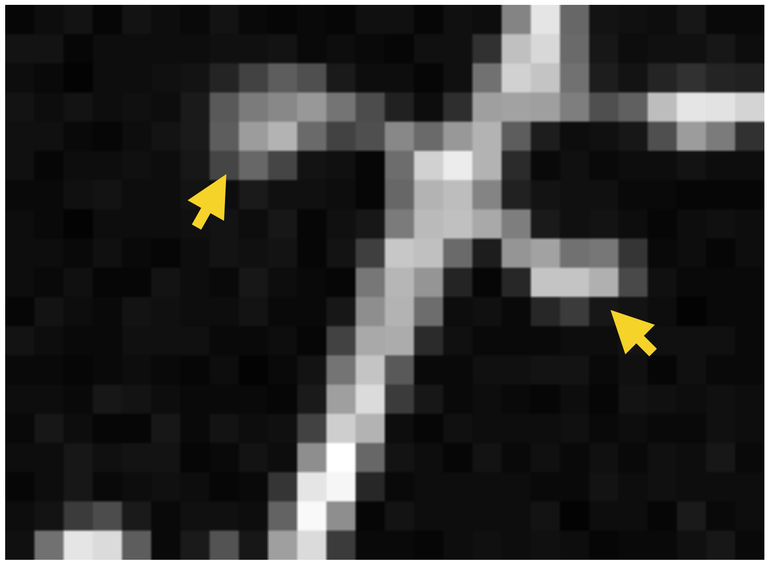
this is your **data**

6	13	19	6	19	13	9	19	9	6	9	6	16	16	6	16	13	132	229	103	19	16	13	23	9	9
19	19	6	13	13	13	13	16	16	19	9	13	9	6	16	16	49	192	216	106	23	13	16	16	23	13
13	9	4	13	13	16	19	36	66	93	79	26	13	13	6	16	113	209	196	113	29	19	36	49	36	33
19	13	19	13	16	13	26	89	123	136	152	116	76	33	13	46	159	162	159	126	79	96	189	229	226	212
16	16	9	6	13	19	26	93	156	179	106	66	79	136	106	152	179	93	29	13	16	23	79	156	123	49
16	6	13	13	16	13	23	69	103	69	19	16	6	109	209	236	179	43	9	16	9	13	13	19	13	13
9	9	16	19	13	13	19	13	26	16	16	13	6	103	179	189	132	33	19	16	16	9	9	6	6	6
13	9	4	13	13	13	16	19	13	23	6	16	23	123	186	192	169	126	26	16	19	13	6	13	16	13
13	13	9	16	9	6	13	19	16	19	6	19	63	199	192	106	29	149	162	113	119	53	9	13	6	13
13	9	16	6	6	19	13	9	23	13	9	6	119	182	149	36	6	39	196	196	176	73	16	9	9	9
6	19	13	9	19	16	13	13	19	9	9	23	142	179	109	13	16	9	39	59	23	19	13	4	9	9
19	13	9	9	16	16	16	9	9	13	6	66	169	172	43	16	9	9	9	13	13	19	16	16	16	9
9	9	6	9	13	9	6	13	4	9	19	116	196	89	9	9	16	16	19	19	9	16	6	16	9	9
13	13	9	23	19	13	9	9	9	6	26	159	219	59	23	9	13	9	6	13	6	19	16	13	16	13
9	23	13	6	6	23	9	19	13	16	66	206	179	13	6	16	13	13	13	16	9	13	9	9	16	13
13	13	23	16	19	19	6	9	19	13	142	255	103	19	13	6	19	9	16	9	16	9	16	13	23	9
6	13	23	9	13	16	13	6	9	53	229	246	39	9	13	13	13	13	9	9	19	13	16	13	13	13
13	19	59	76	26	9	16	16	13	99	249	142	6	19	13	13	13	13	19	4	13	13	6	26	9	13
16	113	229	219	93	9	26	83	23	159	219	59	9	9	6	13	16	13	16	13	6	9	9	16	23	9

this is your **result**



this just helps to ***communicate*** the result



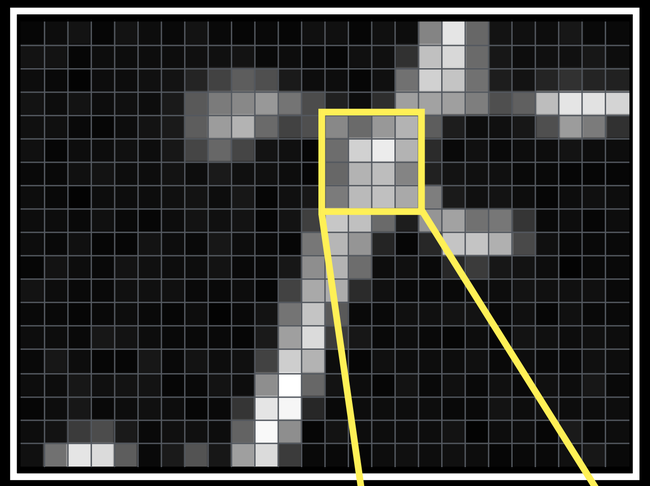
Inspect Individual Pixel Values

ndv

https://pyapp-kit.github.io/ndv

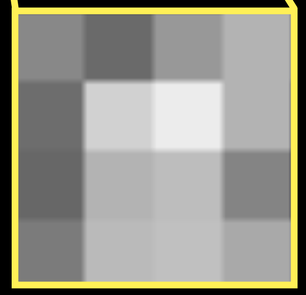
Introduction to Digital Images

Pixel = Picture Element



=

6	13	19	6	19	13	9	19	9	6	9	6	16	16	6	16	13	132	229	103	19	16	13	23	9	9
19	19	6	13	13	13	13	16	16	19	9	13	9	6	16	16	49	192	216	106	23	13	16	16	23	13
13	9	4	13	13	16	19	36	66	93	79	26	13	13	6	16	113	209	196	113	29	19	36	49	36	33
19	13	19	13	16	13	26	89	123	136	152	116	76	33	13	46	159	162	159	126	79	96	189	229	226	212
16	16	9	6	13	19	26	93	156	179	106	66	79	136	106	152	179	93	29	13	16	23	79	156	123	49
16	6	13	13	16	13	23	69	103	69	19	16	6	109	209	236	179	43	9	16	9	13	13	19	13	13
9	9	16	19	13	13	19	13	26	16	16	13	6	103	179	189	132	33	19	16	16	9	9	6	6	6
13	9	4	13	13	13	16	19	13	23	6	16	23	123	186	192	169	126	26	16	19	13	6	13	16	13
13	13	9	16	9	6	13	19	16	19	6	19	63	199	192	106	29	149	162	113	119	53	9	13	6	13
13	9	16	6	6	19	13	9	23	13	9	6	119	182	149	36	6	39	196	196	176	73	16	9	9	9
6	19	13	9	19	16	13	13	19	9	9	23	142	179	109	13	16	9	39	59	23	19	13	4	9	9
19	13	9	9	16	16	16	9	9	13	6	66	169	172	43	16	9	9	9	13	13	19	16	16	16	9
9	9	6	9	13	9	6	13	4	9	19	116	196	89	9	9	16	16	19	19	9	16	6	16	9	9
13	13	9	23	19	13	9	9	9	6	26	159	219	59	23	9	13	9	6	13	6	19	16	13	16	13
9	23	13	6	6	23	9	19	13	16	66	206	179	13	6	16	13	13	13	16	9	13	9	9	16	13
13	13	23	16	19	19	6	9	19	13	142	255	103	19	13	6	19	9	16	9	16	9	16	13	23	9
6	13	23	9	13	16	13	6	9	53	229	246	39	9	13	13	13	13	9	9	19	13	16	13	13	13
13	19	59	76	26	9	16	16	13	99	249	142	6	19	13	13	13	13	19	4	13	13	6	26	9	13
16	113	229	219	93	9	26	83	23	159	219	59	9	9	6	13	16	13	16	13	6	9	9	16	23	9



=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

A digital image is a matrix of numbers!

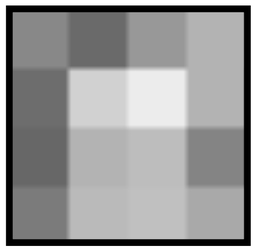
Where do these number come from?

Detectors in Fluorescence Microscopy

The detectors used in fluorescence microscopy are **monochromatic**.

Cameras or PMTs are not able to distinguish between different wavelengths(they just collect photons), you need fluorescence filters to separate your fluorophores.

The detector converts photons in digital numbers (linear relation).



=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

Each pixel in the digital image has **one digital value** that **depends on** the **intensity** of the signal emitted by the **sample**.

Digital Values = Pixel Intensity Value

The **range** of possible **digital values** is defined by the **bit depth**.

Detectors in Fluorescence Microscopy - Bit Depth

The **bit depth** defines the range of possible **digital values** that each pixel can have, usually **8**, **12** or **16 bit**.

The **bit depth** is expressed in **grey values**.

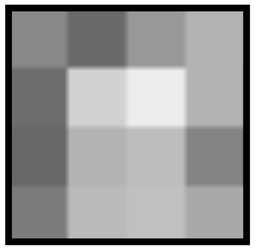
bit depth of the **image** = **bit depth** of the **detector**
(Unless you change that during acquisition)

x bit = a range of 2^x grey values

8 bit image = **each pixel** can have 2^8 **grey values** = 256 grey values = **range 0-255**

12 bit image = **each pixel** can have 2^{12} **grey values** = 4096 grey values = **range 0-4095**

16 bit image = **each pixel** can have 2^{16} **grey values** = 65536 grey values = **range 0-65535**

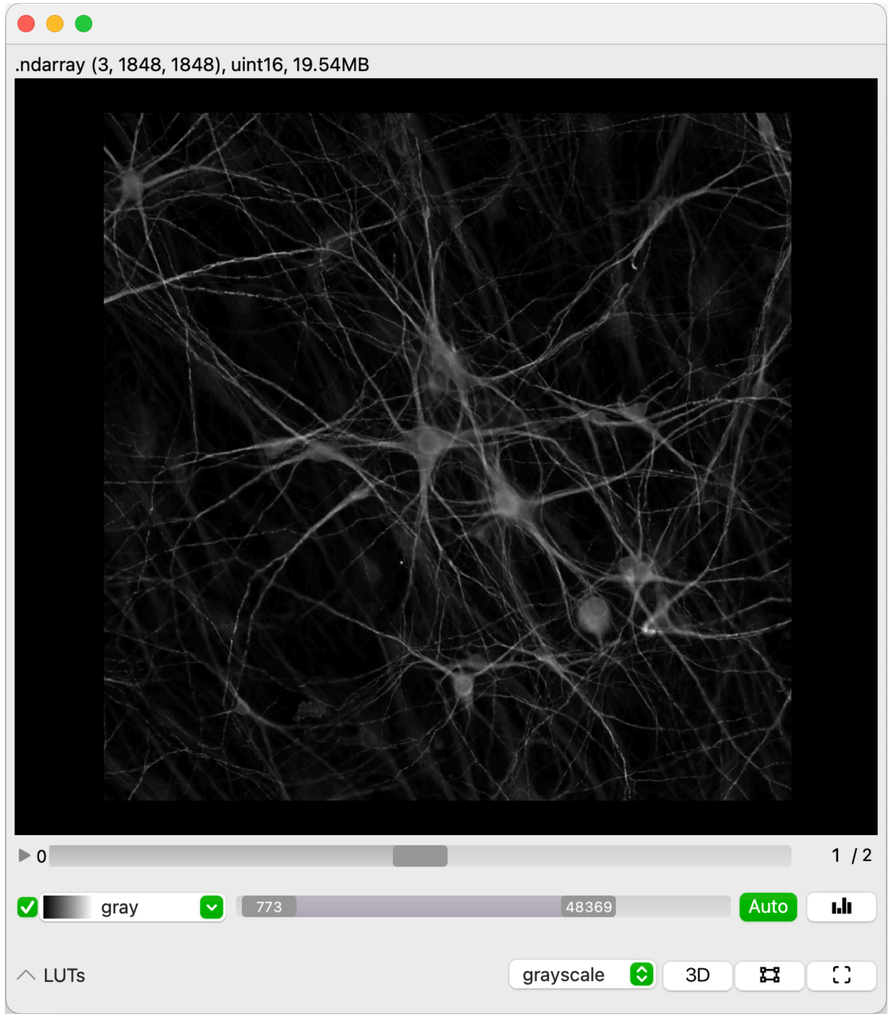


=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

Digital Value = Pixel Intensity Value = Grey Value

Bit Depth



numpy array

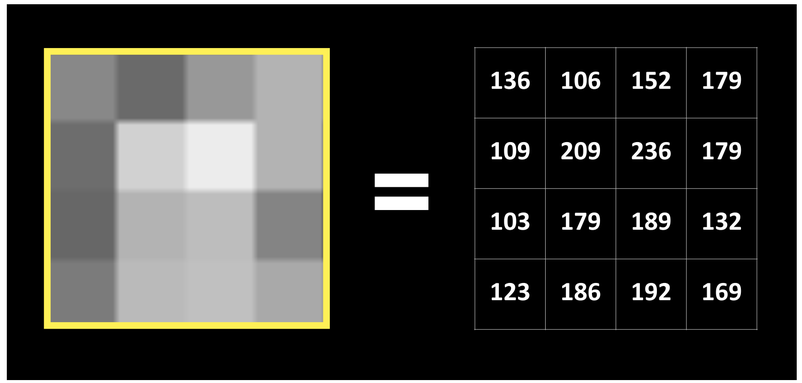
```
array([[[ 1928,  1791,  1750, ...,  1906,  1943,  1891],
        [ 1822,  1726,  1775, ...,  1795,  1921,  1862],
        [ 1843,  1749,  1759, ...,  1961,  1993,  1777],
        ...,
        [ 1737,  1696,  1800, ...,  1660,  1726,  1751],
        [ 1770,  1699,  1958, ...,  1655,  1710,  1753],
        [ 1669,  1717,  1650, ...,  1729,  1661,  1829]],

       [[ 3746,  4124,  4156, ...,  2774,  2862,  2949],
        [ 4215,  4514,  4481, ...,  2833,  2785,  2985],
        [ 4675,  4621,  4481, ...,  2937,  2893,  2841],
        ...,
        [ 4115,  4092,  3715, ...,  2891,  3136,  3226],
        [ 3938,  3949,  3895, ...,  2432,  2813,  2935],
        [ 3978,  3873,  3930, ...,  2176,  2495,  2791]],

       [[ 5819,  5702,  5581, ..., 12589, 12691, 12703],
        [ 6646,  6056,  5764, ..., 12388, 12902, 12488],
        [ 6928,  6775,  6620, ..., 12064, 12366, 12917],
        ...,
        [ 2362,  2293,  2346, ...,  4860,  5033,  5450],
        [ 2292,  2290,  2363, ...,  4285,  4389,  4335],
        [ 2161,  2237,  2395, ...,  4359,  4072,  4156]]],
      shape=(3, 1848, 1848), dtype=uint16)
```

Display your Images

Mapping Image Intensity to Monitor Intensity (LookUp Tables)

LUT = how the grey values are displayed

LUTs do not change the pixel values



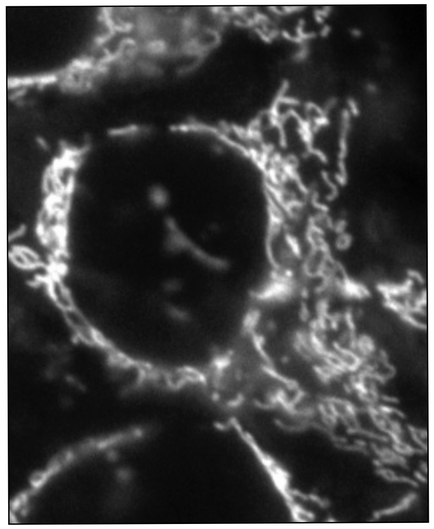
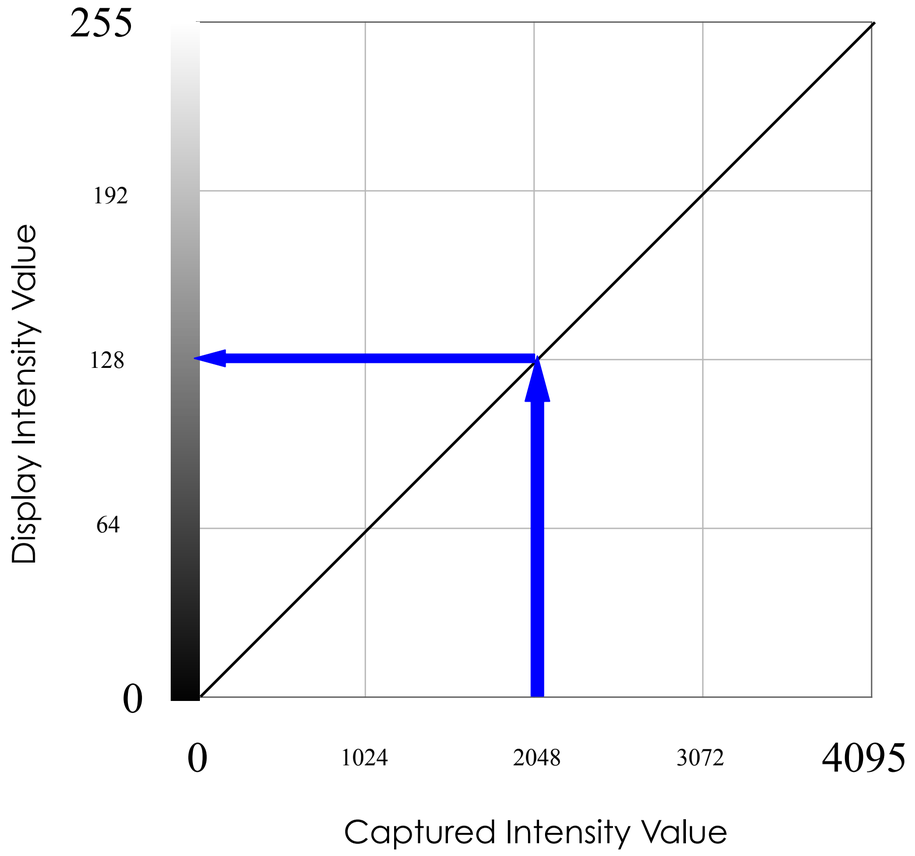


Image (12 bit)	Displayed color
0 1 … 2000 … 4095	grey LUT



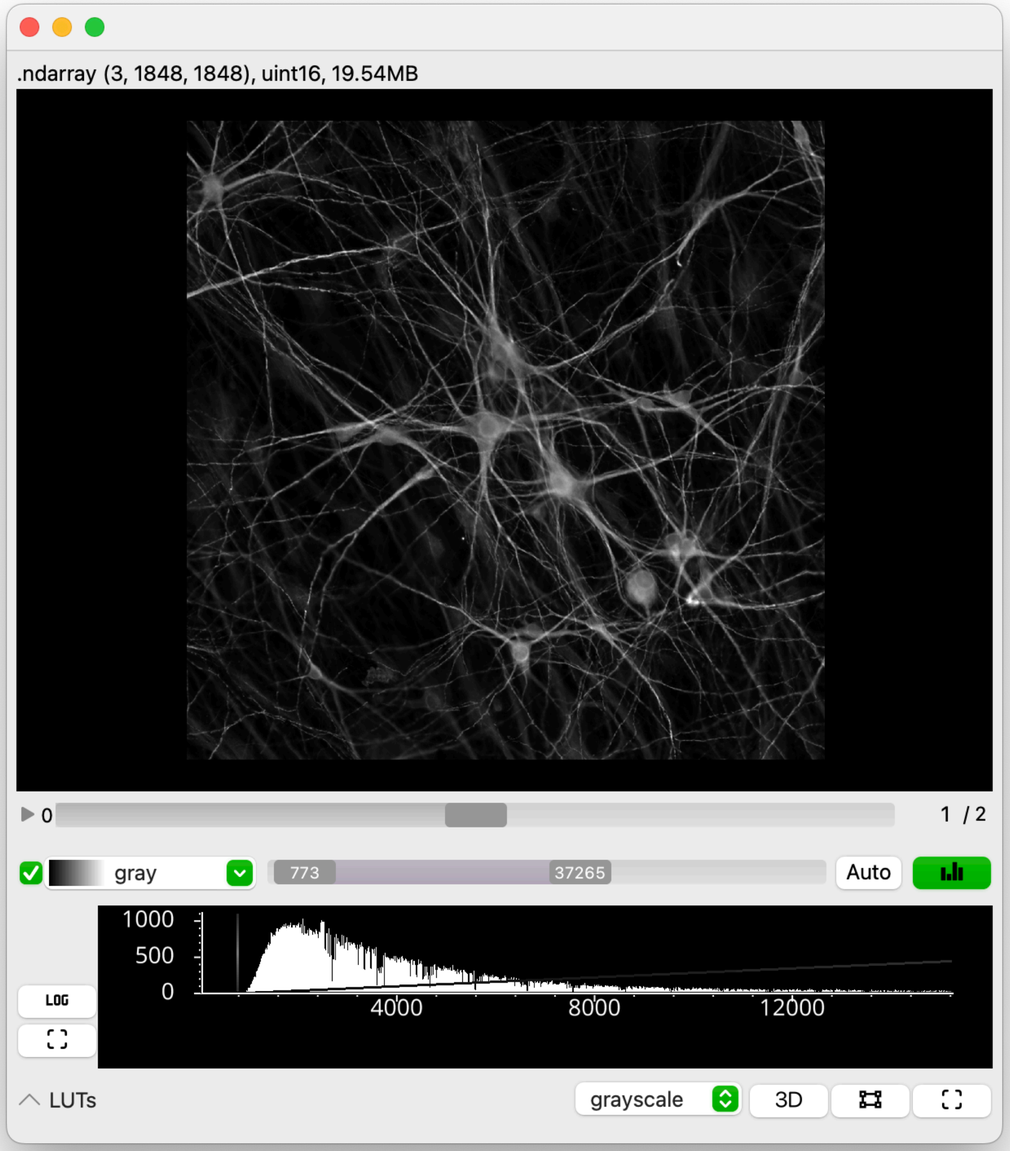
Display your Images - Brightness & Contrast

ndv

Histogram

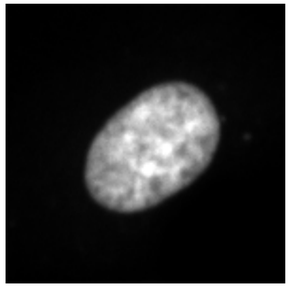
ndv

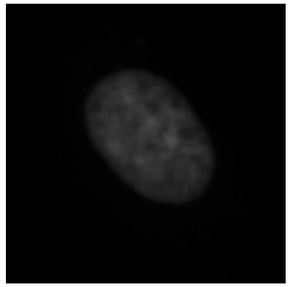
jaehyuk-lee: https://jaehyuk-lee.com/animated-image-histogram/



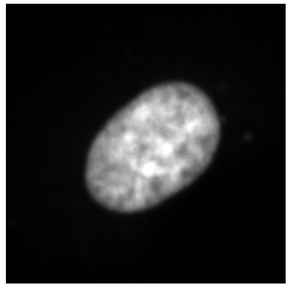
Display your Images - Brightness & Contrast

Which image has more fluorescence?





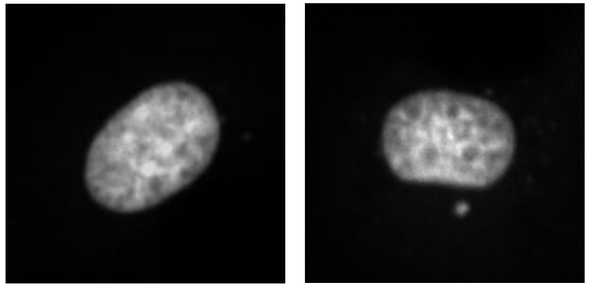
Mean:	**4803**	**4803**
Display range:	188-**16828**	188-**45514**



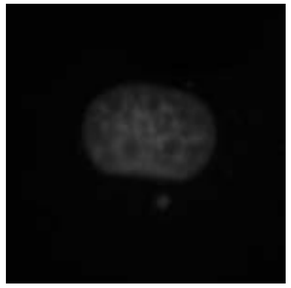
Mean:	**4803**	**4803**
Display range:	188-**16828**	188-**16828**

Display your Images - Brightness & Contrast

Which image has more fluorescence?



Mean:	**4803**	**2074**
Display range:	188-**19540**	112-**7768**



Mean:	**4803**	**2074**
Display range:	188-1**9540**	188-**19540**

Do NOT trust your eyes, rely on numbers!

Images and Colors - Lookup Tables (LUTs)

LUT = how the grey values are displayed

LUTs do not change the pixel values

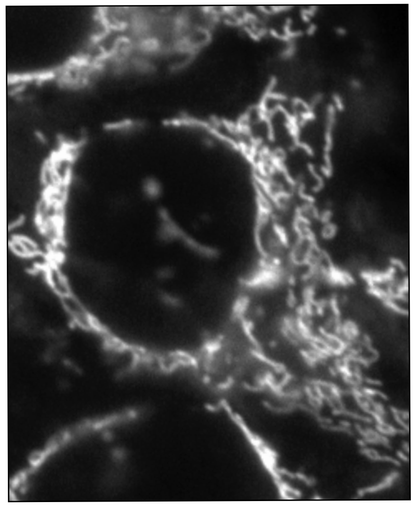
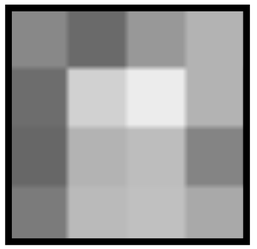


Image (8 bit)	Displayed color
0	grey LUT
1	
…	
100	
…	
255	



=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

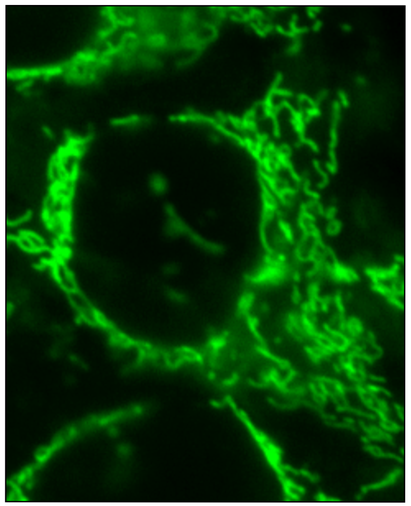
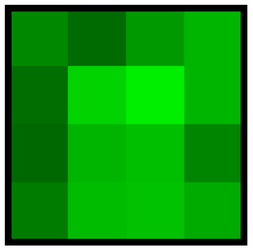


Image (8 bit)	Displayed color
0	green LUT
1	
…	
100	
…	
255	

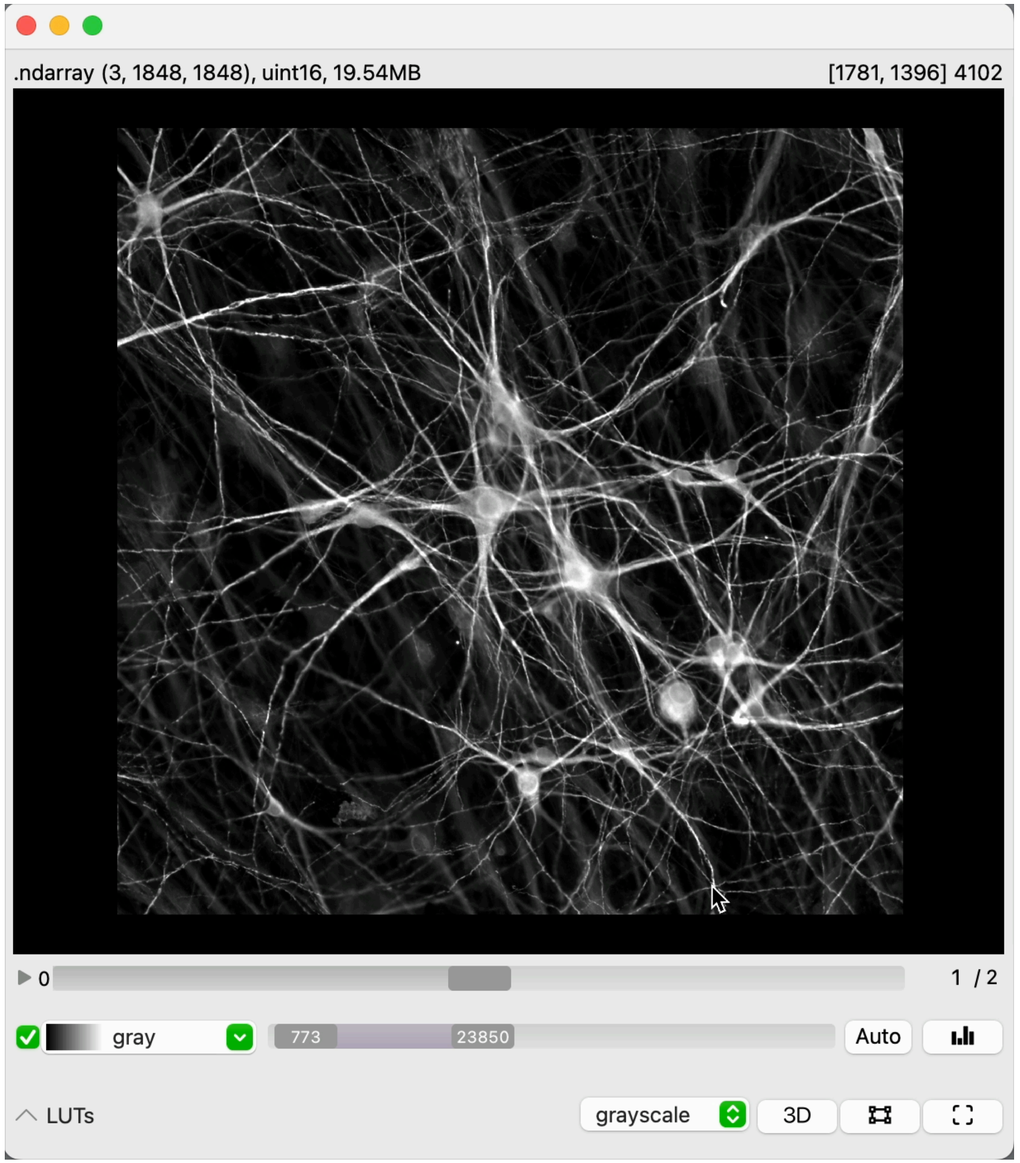


=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

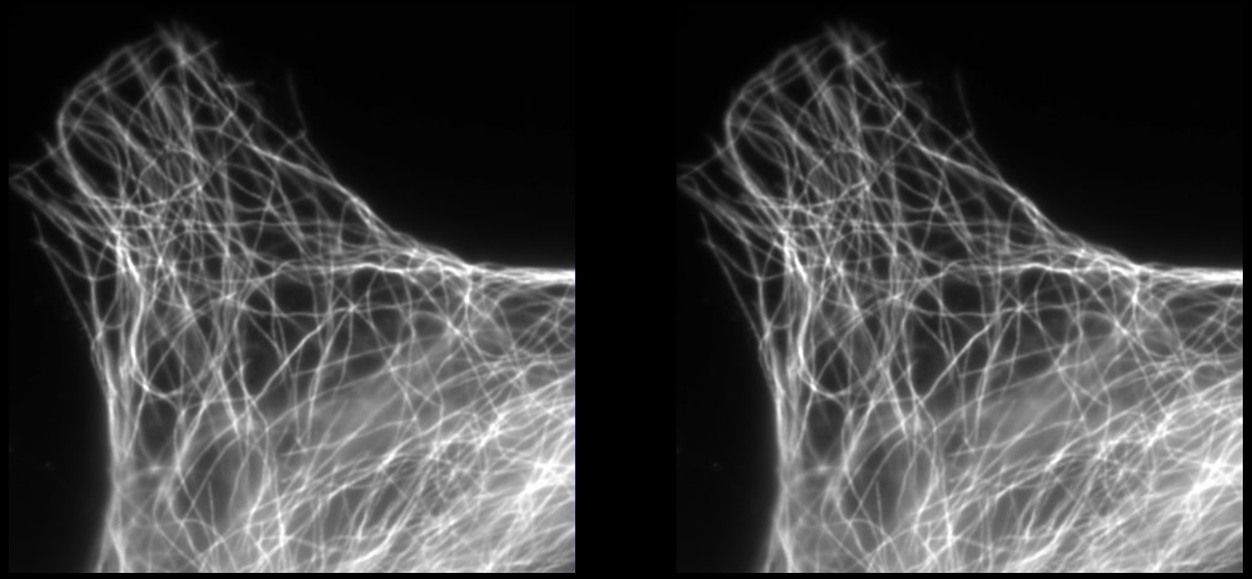
Images and Colors - Lookup Tables (LUTs)

ndv



Images and Colors - Choose the right LUT

Which is brighter?

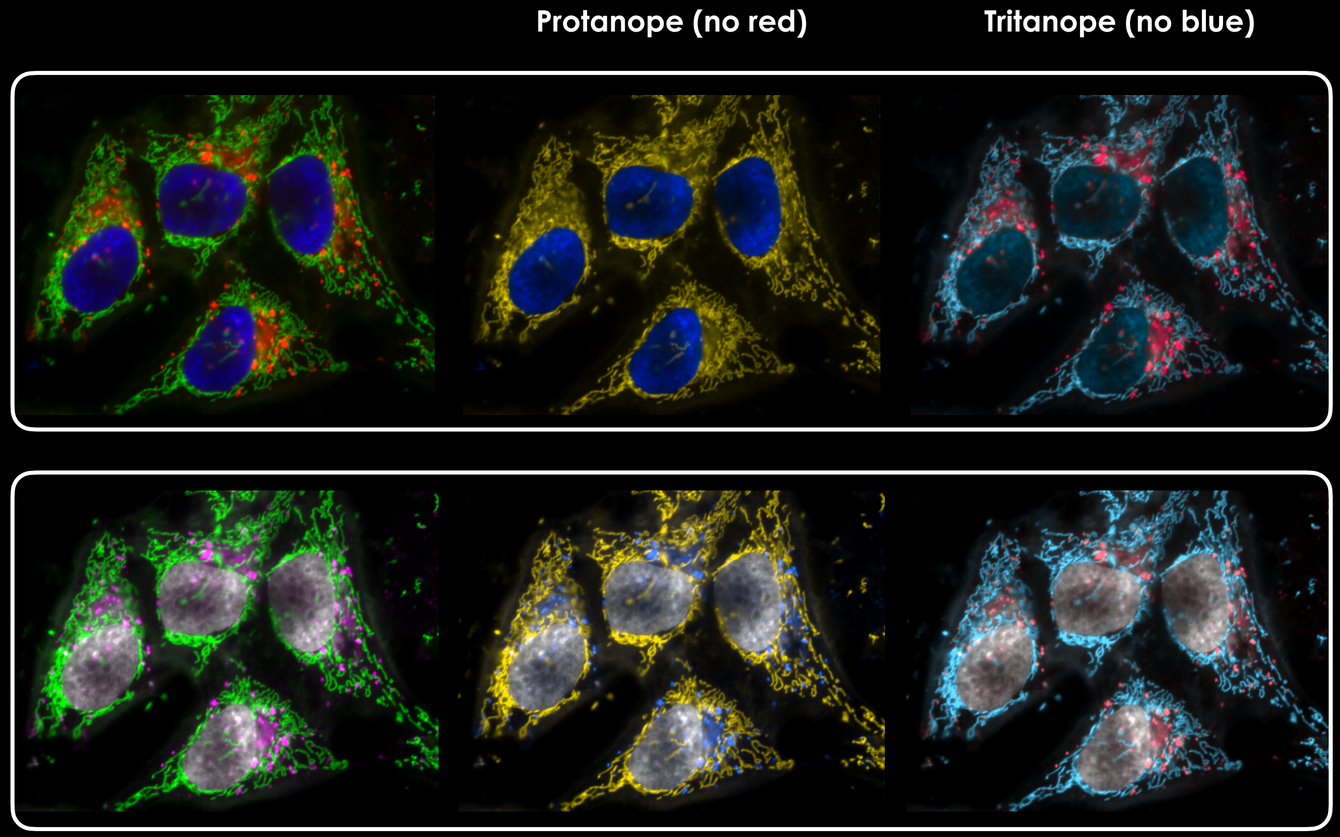


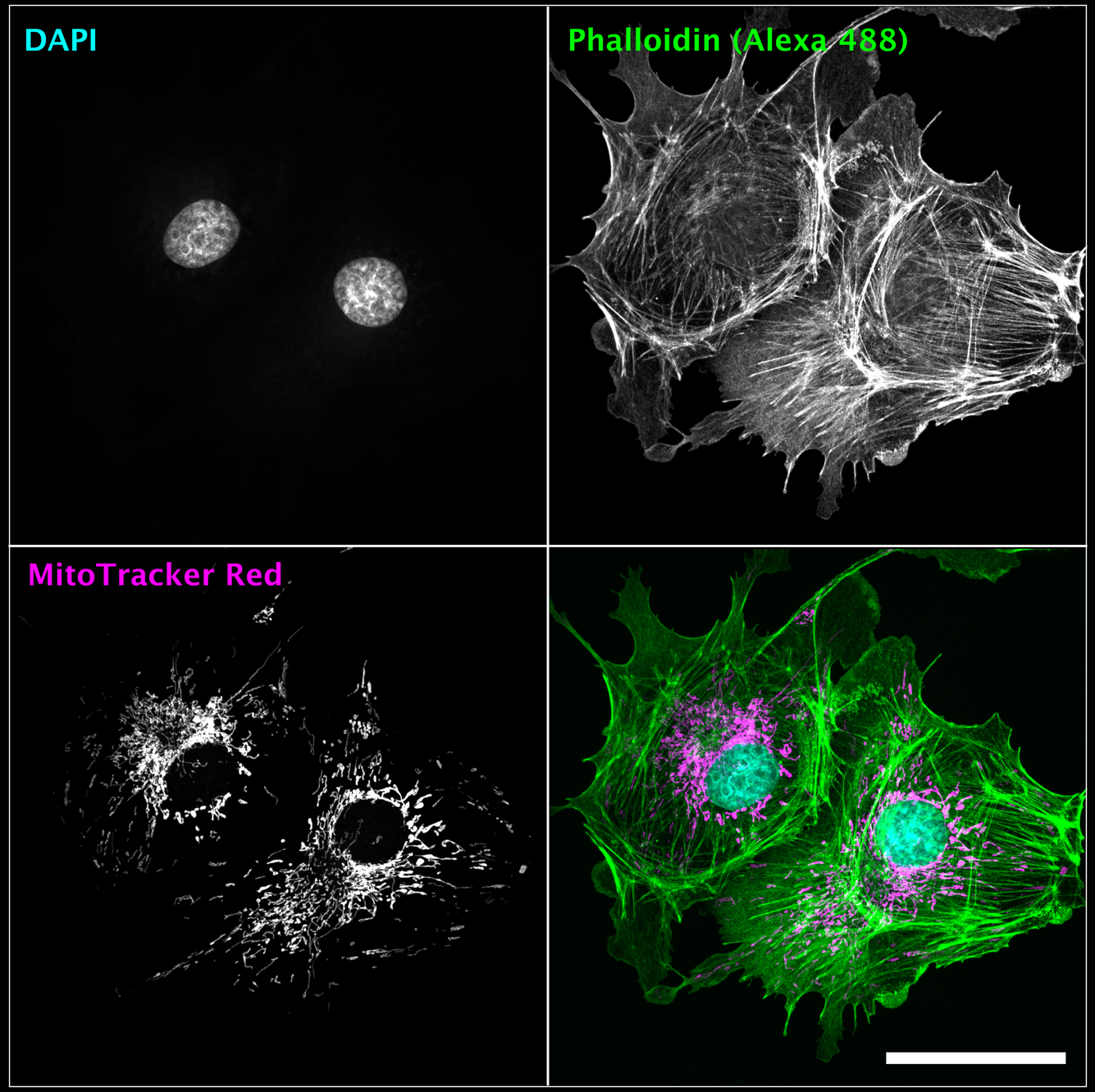
The human eye evaluates intensity best in grayscale

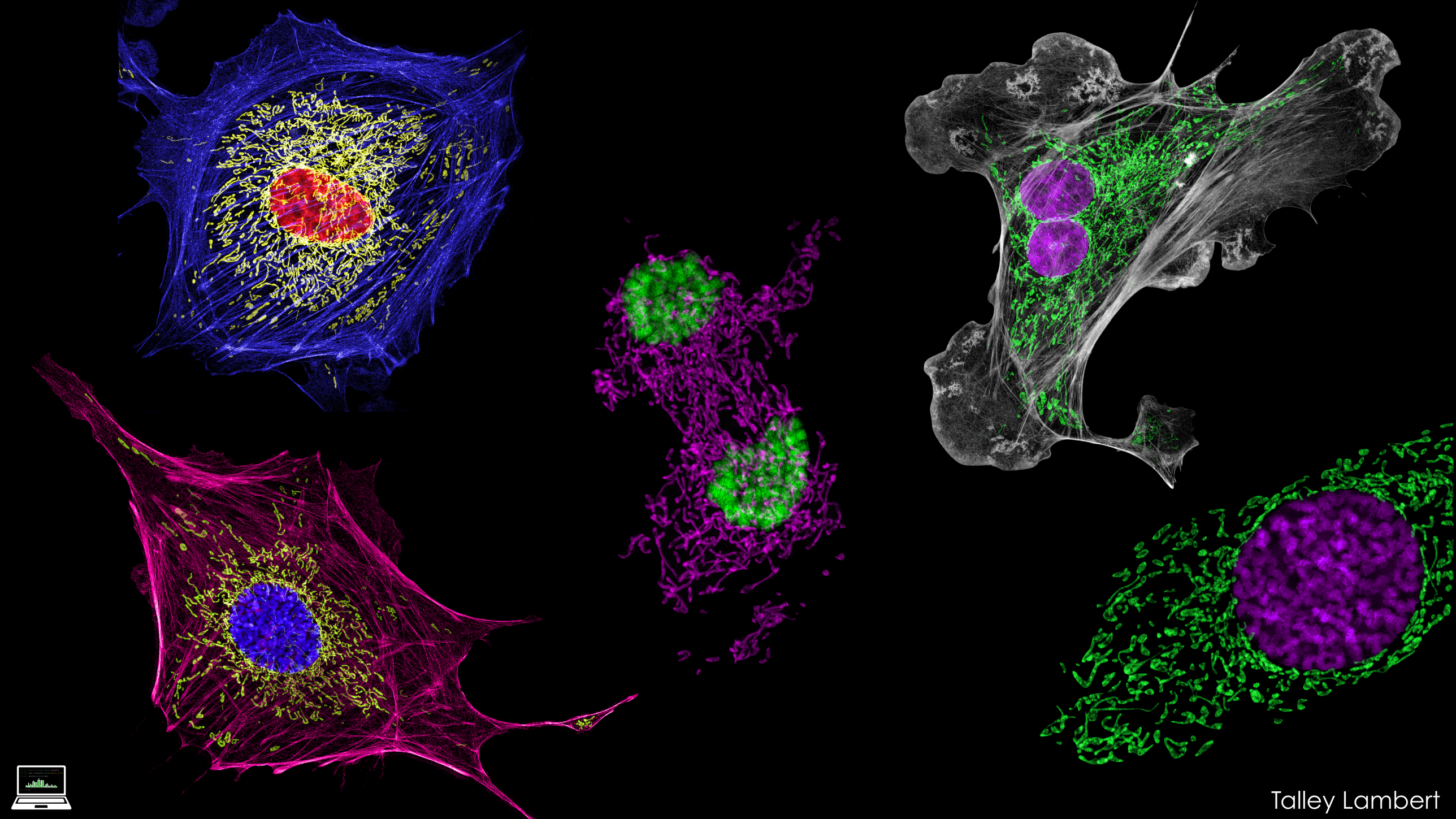
If you are imaging for example a blue fluorophore, you are NOT FORCED to display it in blue!

Images and Colors - Choose the right LUT

Color blind people don’t distinguish some colors




DAPI
Phalloidin (Alexa 488)
MitoTracker Red



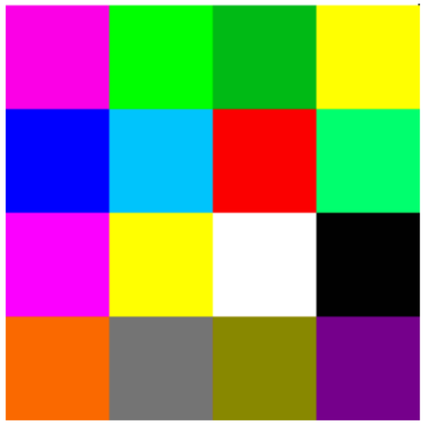
Talley Lambert

Images and Colors

RGB Images (still matrix of numbers)

LUTs cannot be applied to RGB Images

RGB image



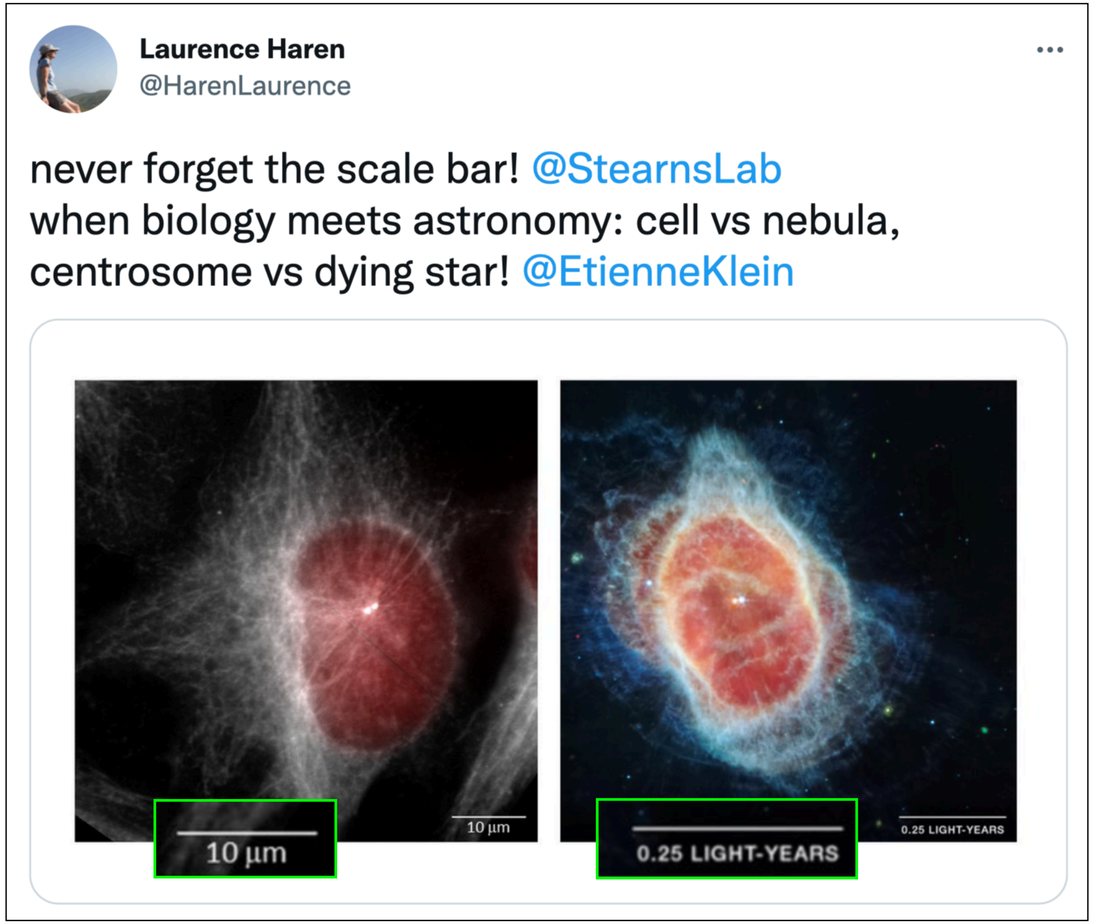
=

255 0 234	0 255 0	0 186 28	255 255 1
0 0 255	1 199 255	255 0 0	0 255 111
255 0 255	255 255 1	255 255 255	0 0 0
254 105 0	117 117 117	141 137 0	118 0 143

RGB Color image (e.g. jpeg, png) = Red + Green + Blue

RGB Color image = 8 bit Red, 8 bit Green, 8 bit Blue = R (0-255), G (0-255), B (0-255)

Pixel Size & image Scale



Pixel Size & image Scale

What if the pixel size is not stored in the metadata?

If you know the magnification and the camera you used for the acquisition, you can estimate the image pixel size.

image pixel size = camera pixel size/magnification

Example:

Magnification = 100x Objective

Camera = Hamamatsu Orca Flash 4

Product number	C13440-20CU
Imaging device	sCMOS
Cell (pixel) Size (μm²)	6.5×6.5
Pixel Array (horizontal by vertical)	2048×2048
Effective Area (horizontal by vertical in mm)	13.312×13.312

pixel width and height:
6.5 μm / 100X = 0.065 μm

Visualize Images in Python

napari matplotlib ndv