SLA1 Camera Characterization

Hot Pixel Leaders in 30s Darks

On May 8, 2024 (UTC) we took various dark exposures with the QHY42 Pro camera.

As they are read in, the darks are scaled to undo zero padding (divided by 16) and the effect of gain (multiplied by 1.39).

They are then combined into a master dark. The master dark is subtracted from the individual darks.

Then we classify all pixels whose values exceed a threshold of 200 as hot pixels. From among these, only the hot pixels which are brighter than their eight nearest neighbors are declared to be “leaders” and the region around them is displayed.

In [1]:

```
# THIS COMMENT IS THE LONGEST A LINE CAN BE AND STILL RENDER COMPLETELY WHEN PRINTING IN LANDSCAPE MODE.

import os, sys
import numpy as np
from astropy import units as u
from astropy.nddata import CCDData
from astropy.io import fits
from ccdproc import ImageFileCollection, combine, subtract_dark, flat_correct # Combiner
import astroalign as aa
import matplotlib.pyplot as plt
%matplotlib inline
from math import log10, floor

home_directory = os.path.expanduser('~')

# soft link to directory containing raw images
sessions_directory = os.path.join(home_directory, '2024 SLA Sessions')

uv_project_directory = os.path.join(home_directory, 'Projects', 'uv-transients')
```

```
analysis_directory = os.path.join(uv_project_directory, 'analyses', '30s_darks')

# The path to the first dark on SLA1 is D:/Raw/2024-05-08/03_38_48/Dark30s/00001.fits
# The files to be processed need to be mirrored on the local machine
# at ~/2024 SLA Sessions/ using the same subdirectory structure.
capture_date = '2024-05-08'
capture_time = '03_38_48'
object_name = 'Dark30s'

# Amount to scale the image data (typically to undo 0 padding of 12-bit to 16-bit values)

scale_due_to_padding = 2**4  # This is division by 16

scale_due_to_gain = 1.39  # from QHYCCD manual for gain of 5

scale = scale_due_to_gain / scale_due_to_padding

# threshold for flagging hot pixels

threshold = 200

# subdirectory for the 30-second darks (following SharpCap Pro capture directory conventions)
dark_directory = os.path.join(
    sessions_directory,
    capture_date,
    capture_time,
    object_name
)

# exposure duration

dark_exposure = 30.0
dark_exposure_with_ccdproc_units = dark_exposure * u.second

# FITS header confirmation

def confirm_fits_header(image, dimensions, exposure_time, filter):
    header = image.header
    assert header['NAXIS1'] == dimensions[0]
    assert header['NAXIS2'] == dimensions[1]
    assert header['EXPTIME'] == exposure_time
    if filter:
        assert header['FILTER'].rstrip() == filter

# Reader with optional parameter to scale (divide) the ADU readings

def scaled_image_reader(file, scale=1):
    img = CCDData.read(file, unit=u.adu)
    scaled_data = img.data * scale
    img.data = scaled_data
    return img

# After all the preliminaries, we read in and combine the dark files

dark_files = ImageFileCollection(dark_directory).files_filtered(include_path='True')

darks = [scaled_image_reader(file, scale=scale) for file in dark_files]

for dark in darks:
    confirm_fits_header(dark, (2048, 2048), dark_exposure, None)

combination_method = 'median'  # alternatively, the method can be 'average'

master_dark = combine(darks, method=combination_method)
```

```
WARNING: FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.151953 from DATE-END'. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.151953 from DATE-END'.
WARNING: FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.152301 from DATE-END'. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.152301 from DATE-END'.
WARNING: FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.152648 from DATE-END'. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.152648 from DATE-END'.
WARNING: FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
```

```
Set MJD-END to 60438.152995 from DATE-END'. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.152995 from DATE-END'.
WARNING: FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.153342 from DATE-END'. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.153342 from DATE-END'.
WARNING: FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.153689 from DATE-END'. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.153689 from DATE-END'.
WARNING: FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.154037 from DATE-END'. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.154037 from DATE-END'.
WARNING: FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.154384 from DATE-END'. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.154384 from DATE-END'.
WARNING: FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.154731 from DATE-END'. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.154731 from DATE-END'.
WARNING: FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.155078 from DATE-END'. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: 'datfix' made the change 'Set MJD-OBS to -678575.000000 from DATE-OBS.
Set MJD-END to 60438.155078 from DATE-END'.
```

Inspect the Data of the Master Dark and a Representative Dark

At this point, the darks and the master_dark are observed to have values of order 1000 ADU, with some outliers far outside that range.

In [2]:

```
# np.set_printoptions(threshold=sys.maxsize) # Uncommenting this line will cause serious I/O strain

master_dark.data
```

Out[2]:

```
array([[5.56000000e+02, 5.34368125e+03, 3.04062500e-01, ...,
        1.29834687e+03, 1.30108344e+03, 1.25013125e+03],
       [4.98749375e+02, 4.98749375e+02, 1.52947781e+03, ...,
        1.25460531e+03, 1.45228937e+03, 1.23327750e+03],
       [1.49537937e+03, 1.55445437e+03, 1.55723438e+03, ...,
        1.25186875e+03, 1.30108344e+03, 1.41849500e+03],
       ...,
       [6.95955625e+02, 7.73491562e+02, 7.24885000e+02, ...,
        1.34695344e+03, 1.31633000e+03, 1.37132187e+03],
       [7.66975937e+02, 7.93081875e+02, 8.24313437e+02, ...,
        1.61109687e+03, 1.66717469e+03, 1.55719094e+03],
       [9.09190312e+02, 1.59715344e+03, 9.83555312e+02, ...,
        2.41773125e+03, 2.18690438e+03, 2.13873219e+03]])
```

In [3]:

```
darks[5].data
```

Out[3]:

```
array([[ 556.      , 5354.80125 ,    0.      , ..., 1300.17125 ,
        1300.17125 , 1241.27    ],
       [ 498.749375,  498.749375, 1518.40125 , ..., 1318.0675  ,
        1419.36375 , 1216.42375 ],
       [1516.576875, 1583.21    , 1518.40125 , ..., 1239.5325  ,
        1397.558125, 1410.32875 ],
       ...,
       [ 694.391875,  788.13    ,  712.0275  , ..., 1314.505625,
        1383.136875, 1363.155625],
       [ 763.718125,  794.7325  ,  825.920625, ..., 1629.775   ,
        1672.778125, 1522.05    ],
       [ 958.144375, 1596.24125 , 1004.014375, ..., 2470.03    ,
        2114.19    , 2175.17625 ]])
```

Subtract Master Dark from Darks

In [4]:
```
subtracted_darks = [
    subtract_dark(
        dark,
        master_dark,
        data_exposure=dark_exposure_with_ccdproc_units,
        dark_exposure=dark_exposure_with_ccdproc_units,
        scale=False
    )
    for dark in darks
]
```

Inspect the Data of a Representative Subtracted Dark

The subtracted darks are observed to have values ranging from something like -50 to +50 ADU.

In [5]:
```
# np.set_printoptions(threshold=sys.maxsize) # Uncommenting this line will cause serious I/O strain

representative_dark_data = subtracted_darks[5].data
representative_dark_data
```

Out[5]:
```
array([[  0.       ,  11.12     ,  -0.3040625, ...,   1.824375 ,
         -0.9121875,  -8.86125  ],
       [  0.       ,   0.       , -11.0765625, ...,  63.4621875,
        -32.925625 , -16.85375  ],
       [ 21.1975   ,  28.755625 , -38.833125 , ..., -12.33625  ,
         96.4746875,  -8.16625  ],
       ...,
       [ -1.56375  ,  14.6384375, -12.8575   , ..., -32.4478125,
         66.806875 ,  -8.16625  ],
       [ -3.2578125,   1.650625 ,   1.6071875, ...,  18.678125 ,
          5.6034375, -35.1409375],
       [ 48.9540625,  -0.9121875,  20.4590625, ...,  52.29875  ,
        -72.714375 ,  36.4440625]])
```

The Routines for Locating Hot Pixel Leaders

As a first cut, we will search for all pixels that exceed some threshold. These are the “hot pixels.”

Then each hot pixel is examined to see if it is the brightest relative to its eight nearest neighbors. If it is, it is added to the list of leaders. (A small bit of tie-breaking code is incorporated.)

In [6]:

```
from collections import namedtuple

HotPixel = namedtuple('HotPixel', 'x y value')
HotPixelGroup = namedtuple('HotPixelGroup', 'leader_index hot_pixels')

def is_winner(candidate_leader, i, j, data):
    if candidate_leader.value > data[j, i]:
        return True
    elif candidate_leader[2] == data[j, i]:
        # some nasty tie-breaking
        if candidate_leader[0] > i:
            return True
        elif candidate_leader[0] == i and candidate_leader[1] > j:
            return True
    else:
        return False

def is_leader(candidate_leader, data):
    data_height, data_width = data.shape
    for offset_y in [-1, 0, 1]:
        j = floor(candidate_leader.y + offset_y)
        if j < 0 or j >= data_height:
            continue
        for offset_x in [-1, 0, 1]:
            i = floor(candidate_leader.x + offset_x)
            # we don't compare the candidate wih itself
            if offset_x == 0 and offset_y == 0:
                continue
            if i < 0 or i >= data_width:
                continue
            if not is_winner(candidate_leader, i, j, data):
                return False
    return True

def find_hot_pixel_leaders(data, threshold=threshold):
    # first we simply find all the hot pixels
    data_height, data_width = data.shape
    exceedances = data > threshold  # an array of true-false values
    values_of_exceedances = data[exceedances]
    exceedance_indices = np.nonzero(exceedances)  # a crafty way of getting the indices of the exceedance
    # all of the hot pixels are candidate leaders
    candidate_leaders = np.transpose([exceedance_indices[1], exceedance_indices[0], values_of_exceedances
    leaders = []
    for i in range(candidate_leaders.shape[0]):
        row = candidate_leaders[i]
        candidate_leader = HotPixel(row[0], row[1], row[2])
        if is_leader(candidate_leader, data):
            leaders.append(candidate_leader)
    return leaders
```

Find and Display the Hot Pixel Leaders

Now we classify all the pixels whose values exceed the threshold as hot pixels. From among these, only the ones which are brighter than their eight nearest neighbors are declared to be “leaders” and the region around them is displayed.

In [7]:

```
# Find the leaders

hot_pixel_leaders = find_hot_pixel_leaders(representative_dark_data, threshold=threshold)

print(len(hot_pixel_leaders))

hot_pixel_leaders
```

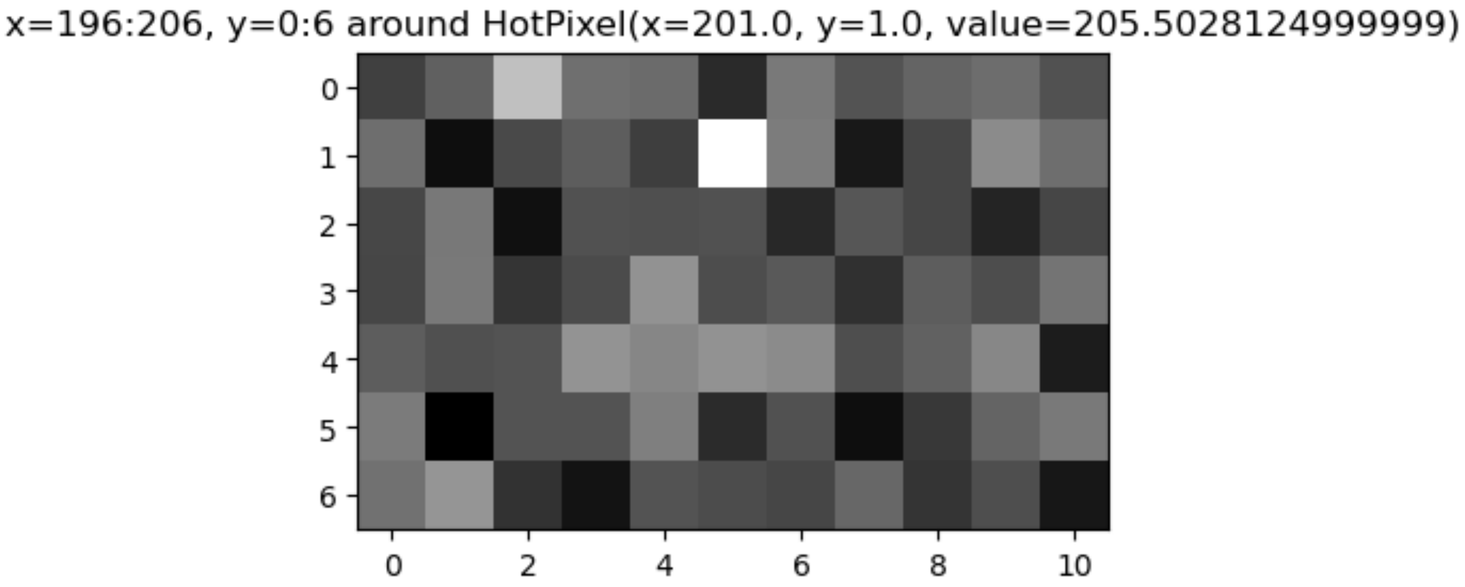
```
53
```

Out[7]:

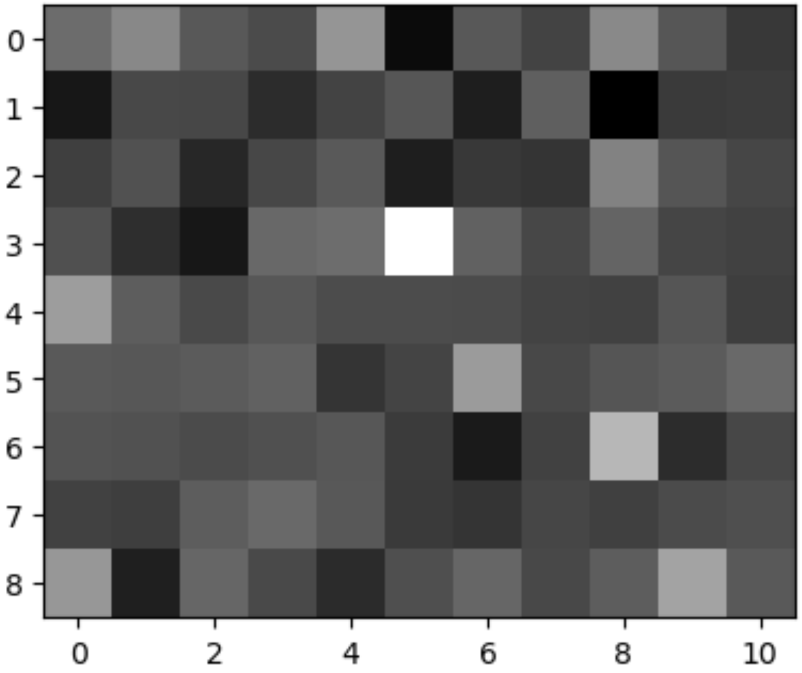
```
[HotPixel(x=201.0, y=1.0, value=205.5028124999999),
                 HotPixel(x=1225.0, y=3.0, value=225.0062499999999),
                 HotPixel(x=939.0, y=114.0, value=579.8906249999999),
                 HotPixel(x=2043.0, y=126.0, value=212.93062499999996),
                 HotPixel(x=2043.0, y=129.0, value=213.4084374999993),
                 HotPixel(x=1101.0, y=130.0, value=221.57468749999998),
                 HotPixel(x=2044.0, y=132.0, value=235.5181250000005),
                 HotPixel(x=2038.0, y=139.0, value=217.49156249999987),
                 HotPixel(x=2042.0, y=143.0, value=203.24406250000084),
                 HotPixel(x=2042.0, y=148.0, value=296.9821874999998),
                 HotPixel(x=2044.0, y=148.0, value=221.2706249999992),
                 HotPixel(x=2045.0, y=153.0, value=202.9399999999996),
                 HotPixel(x=421.0, y=169.0, value=274.4815625000001),
                 HotPixel(x=1784.0, y=257.0, value=238.34156249999978),
                 HotPixel(x=38.0, y=278.0, value=829.2653124999999),
                 HotPixel(x=482.0, y=309.0, value=211.80124999999953),
                 HotPixel(x=1484.0, y=339.0, value=204.50374999999985),
                 HotPixel(x=1435.0, y=603.0, value=344.025),
                 HotPixel(x=1696.0, y=654.0, value=214.62468750000062),
                 HotPixel(x=1461.0, y=670.0, value=201.2893750000003),
                 HotPixel(x=1225.0, y=713.0, value=206.02406249999967),
                 HotPixel(x=1013.0, y=716.0, value=204.9815625000001),
                 HotPixel(x=1771.0, y=722.0, value=243.85812499999975),
                 HotPixel(x=2.0, y=773.0, value=317.09375),
                 HotPixel(x=8.0, y=784.0, value=214.92875000000004),
                 HotPixel(x=14.0, y=788.0, value=293.7678125000002),
                 HotPixel(x=10.0, y=790.0, value=244.0753125000001),
                 HotPixel(x=9.0, y=800.0, value=258.453125),
                 HotPixel(x=10.0, y=804.0, value=210.88906250000036),
                 HotPixel(x=19.0, y=804.0, value=220.35843749999958),
                 HotPixel(x=7.0, y=809.0, value=200.7681249999996),
                 HotPixel(x=0.0, y=818.0, value=227.56906249999975),
                 HotPixel(x=0.0, y=869.0, value=233.17249999999967),
                 HotPixel(x=3.0, y=871.0, value=346.4140625),
                 HotPixel(x=3.0, y=874.0, value=256.4115624999995),
                 HotPixel(x=1.0, y=878.0, value=273.2218750000002),
                 HotPixel(x=1456.0, y=923.0, value=223.74656250000044),
                 HotPixel(x=1126.0, y=958.0, value=279.1293749999995),
                 HotPixel(x=63.0, y=1117.0, value=1240.2709375),
                 HotPixel(x=861.0, y=1179.0, value=260.625),
                 HotPixel(x=1066.0, y=1306.0, value=592.5743749999999),
                 HotPixel(x=178.0, y=1318.0, value=566.5987500000001),
                 HotPixel(x=1852.0, y=1337.0, value=202.07125000000008),
                 HotPixel(x=1571.0, y=1476.0, value=694.044375),
                 HotPixel(x=1562.0, y=1668.0, value=281.4315624999999),
                 HotPixel(x=439.0, y=1749.0, value=5163.5025),
                 HotPixel(x=351.0, y=1928.0, value=325.34687499999995),
                 HotPixel(x=836.0, y=1929.0, value=282.3003124999998),
                 HotPixel(x=51.0, y=1945.0, value=205.5896875000003),
                 HotPixel(x=1349.0, y=1961.0, value=201.94093749999956),
                 HotPixel(x=939.0, y=1984.0, value=334.64250000000015),
                 HotPixel(x=1779.0, y=2005.0, value=202.11468749999995),
                 HotPixel(x=2022.0, y=2023.0, value=223.9637499999999)]
```

In [8]:

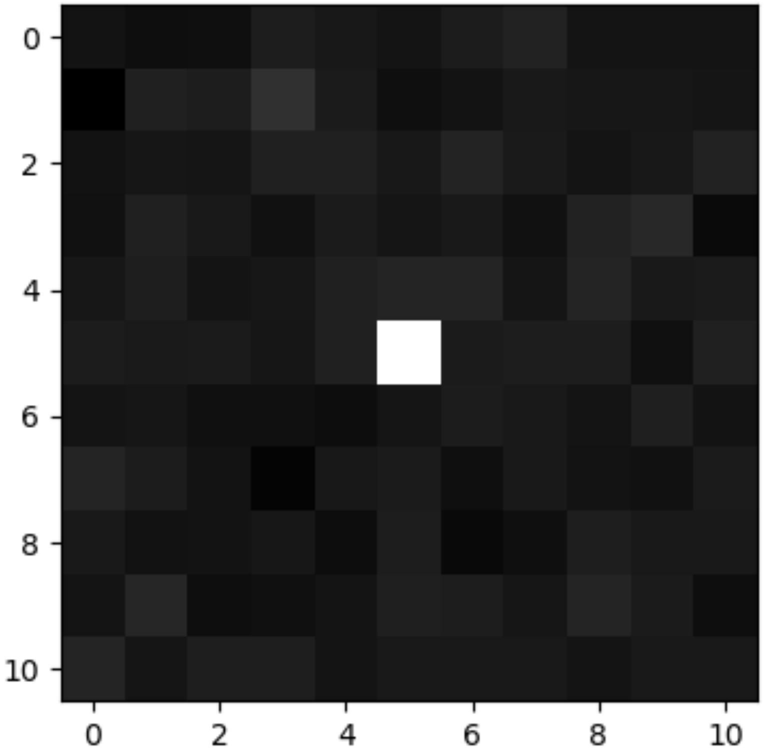
```python
# Display the leaders -- contains a bit of fussy code for dealing with leaders near the edge

def display_leader(data, leader):

    lower_x = floor(leader.x - 5)
    upper_x = floor(lower_x + 11)
    slice_x = slice(lower_x, upper_x)
    lower_y = floor(leader.y - 5)
    upper_y = floor(lower_y + 11)
    slice_y = slice(lower_y, upper_y)

    fig_size_x = 4
    fig_size_y = 4

    # a bit of fussy code for dealing with leaders near the edge

    if (lower_x < 0):
        lower_x = 0
        fig_size_x /= upper_x / 11
    elif (upper_x > 2048):
        upper_x = 2048
        fig_size_x /= (2048 - lower_x) / 11


    if (lower_y < 0):
        lower_y = 0
        fig_size_y /= upper_y / 11
    elif (upper_y > 2048):
        upper_y = 2048
        fig_size_y /= (2048 - upper_y) / 11

    fig, axes = plt.subplots(1, 1, figsize=(fig_size_x, fig_size_y))

    title = "x={}:{}, y={}:{} around {}".format(lower_x, upper_x - 1, lower_y, upper_y - 1, leader)

    subframe = representative_dark_data[lower_y:upper_y, lower_x:upper_x]

    axes.imshow(subframe, cmap='gray')
    axes.set_title(title, fontsize=12)
    plt.tight_layout()
    plt.show()

for leader in hot_pixel_leaders:
    display_leader(representative_dark_data, leader)
```


x=196:206, y=0:6 around HotPixel(x=201.0, y=1.0, value=205.5028124999999)

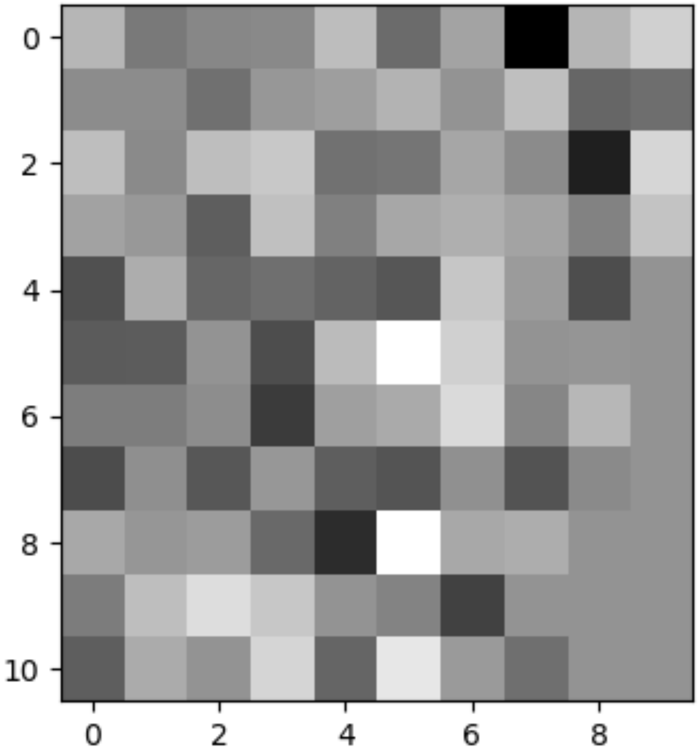
x=1220:1230, y=0:8 around HotPixel(x=1225.0, y=3.0, value=225.0062499999999)



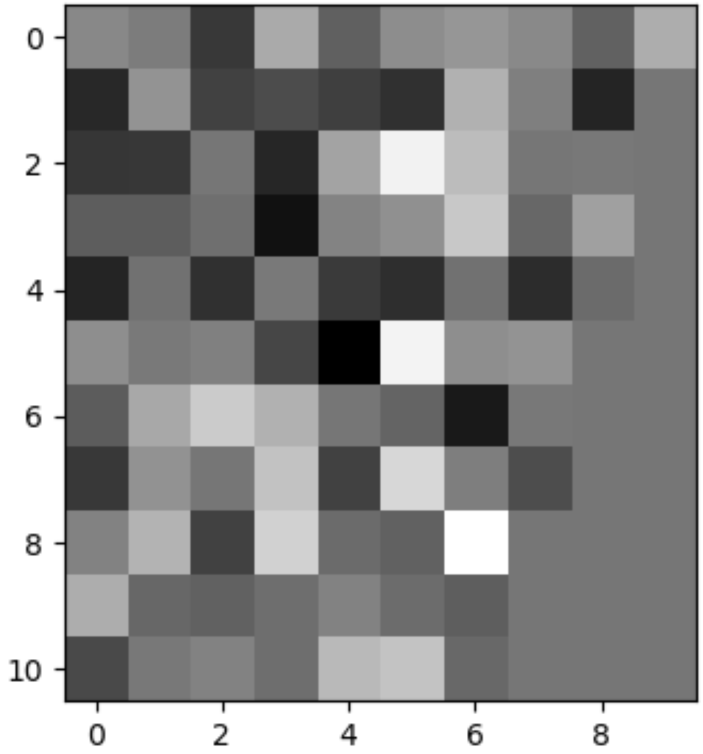
x=934:944, y=109:119 around HotPixel(x=939.0, y=114.0, value=579.8906249999999)



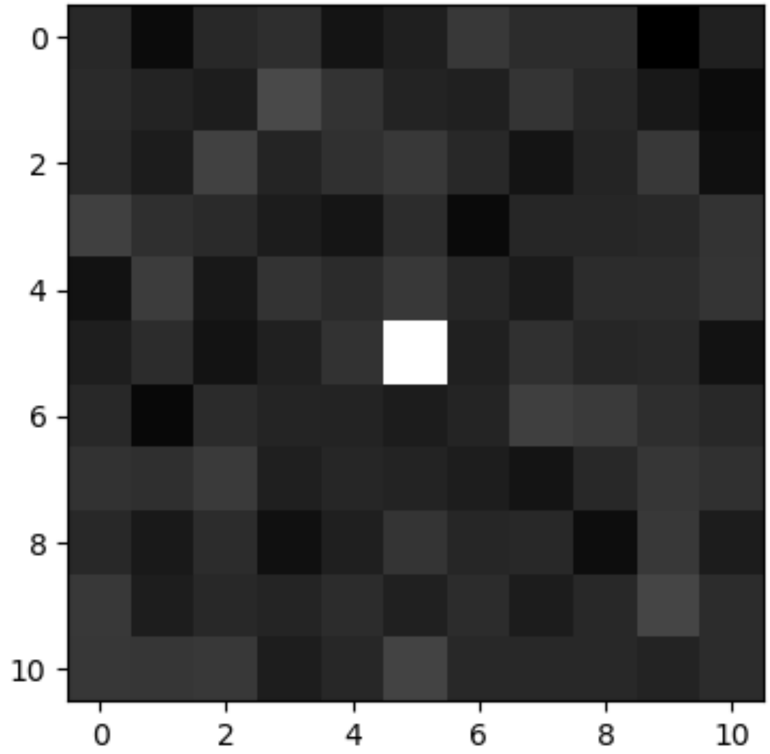
x=2038:2047, y=121:131 around HotPixel(x=2043.0, y=126.0, value=212.93062499999996)



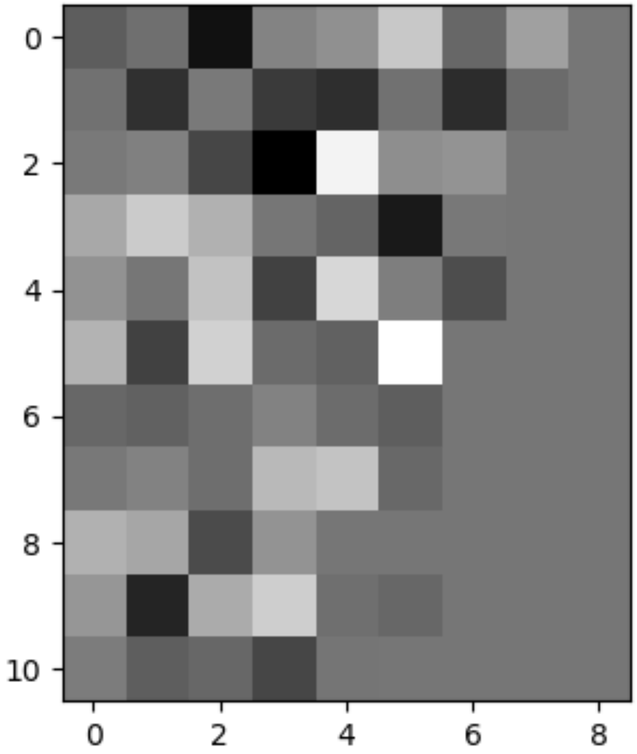
x=2038:2047, y=124:134 around HotPixel(x=2043.0, y=129.0, value=213.4084374999993)

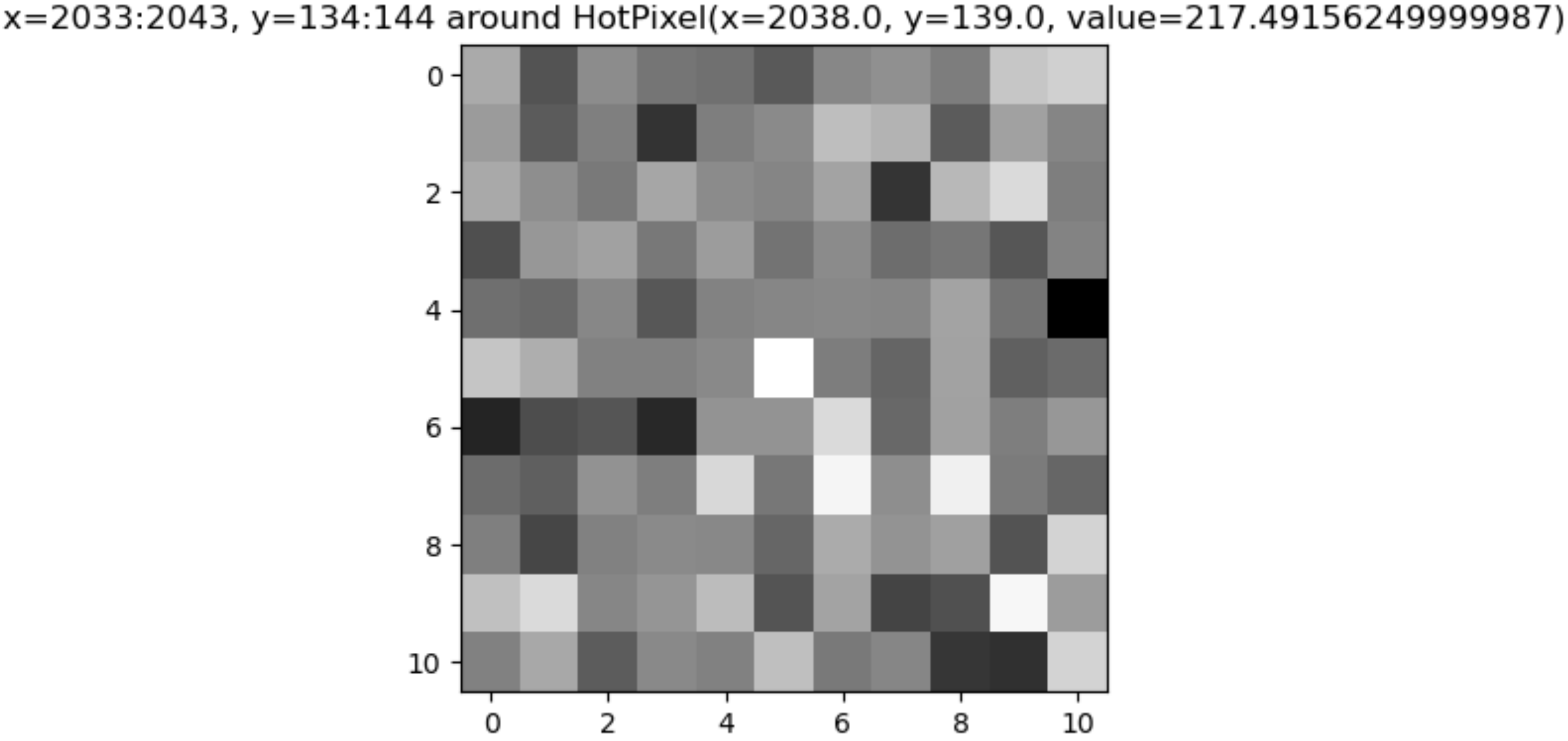


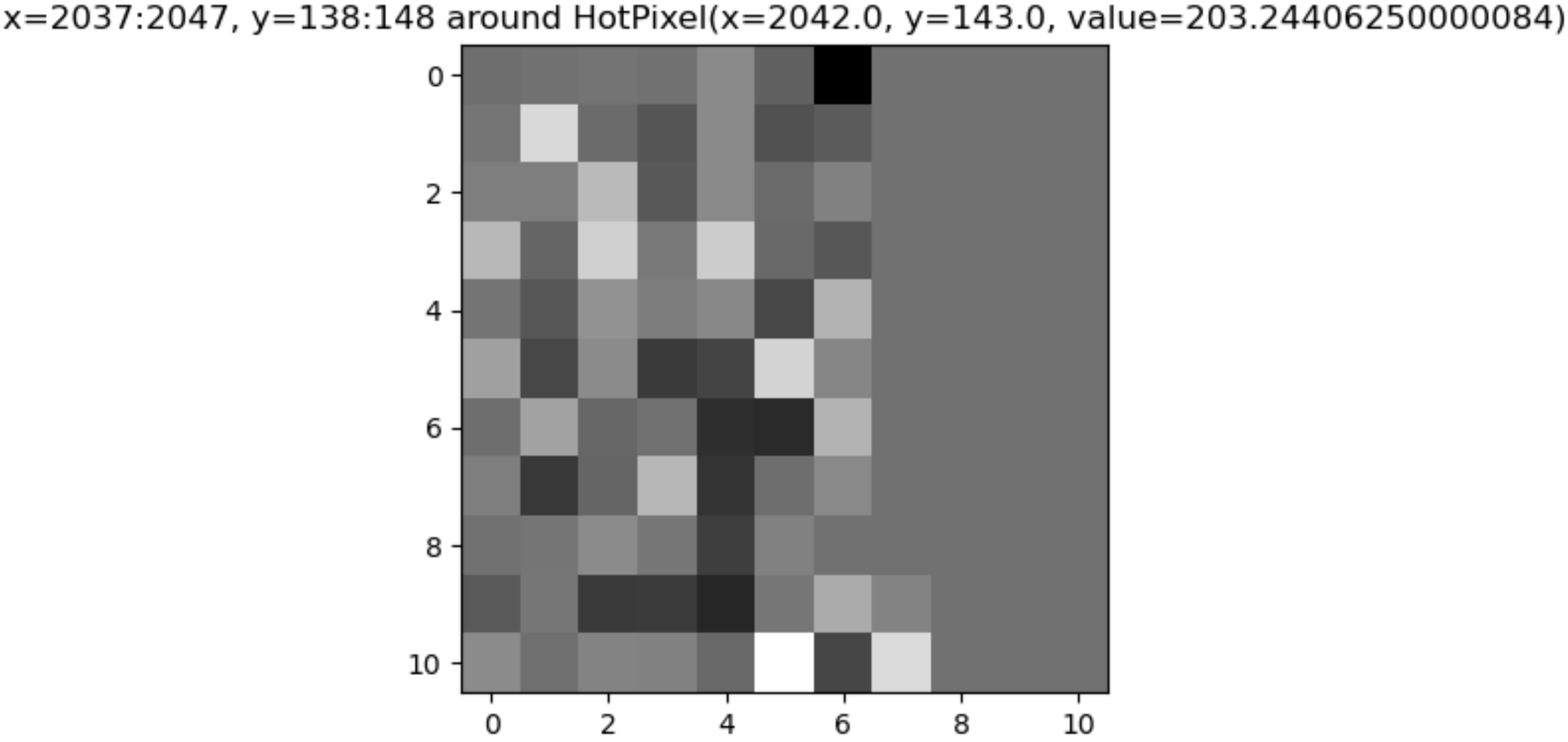
x=1096:1106, y=125:135 around HotPixel(x=1101.0, y=130.0, value=221.57468749999998)



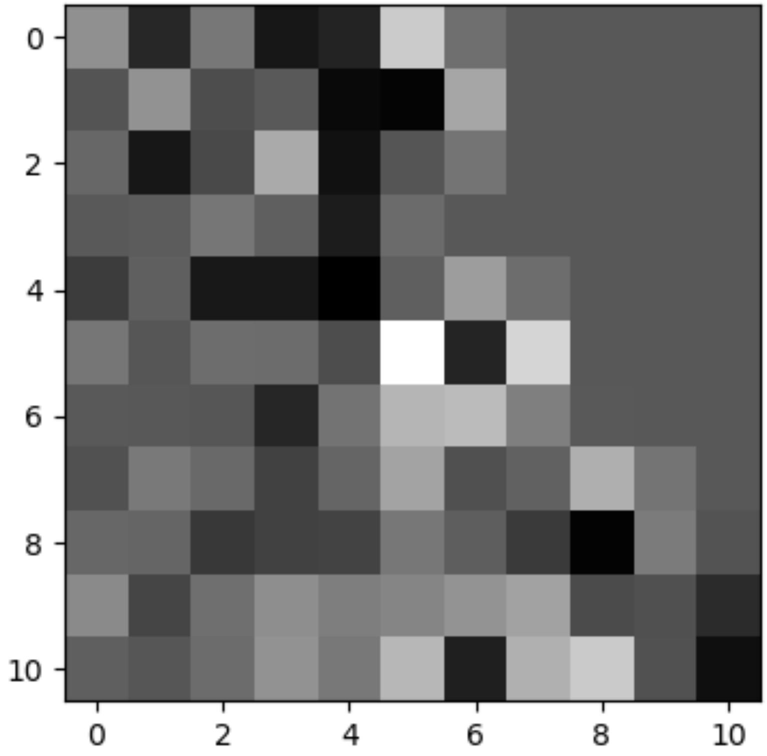
x=2039:2047, y=127:137 around HotPixel(x=2044.0, y=132.0, value=235.5181250000005)



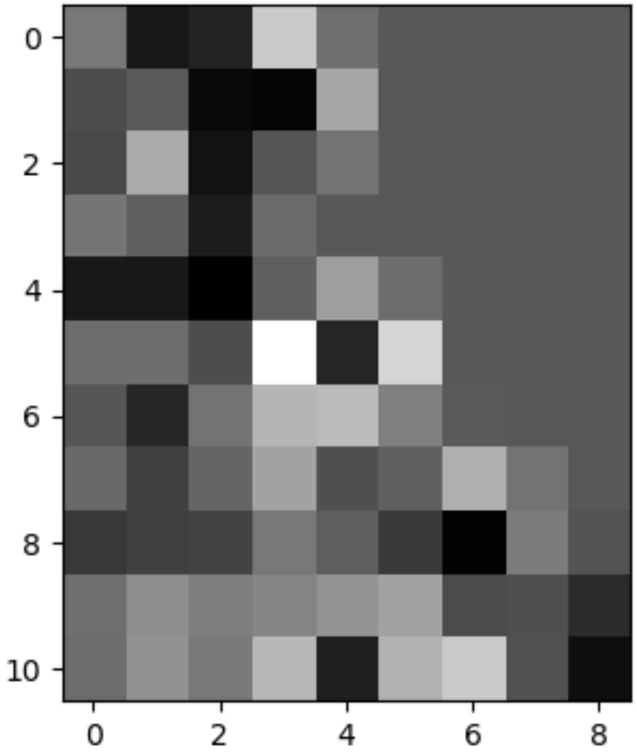

x=2033:2043, y=134:144 around HotPixel(x=2038.0, y=139.0, value=217.49156249999987)
0
2
4
6
8
10
0
2
4
6
8
10


x=2037:2047, y=138:148 around HotPixel(x=2042.0, y=143.0, value=203.24406250000084)
0
2
4
6
8
10
0
2
4
6
8
10

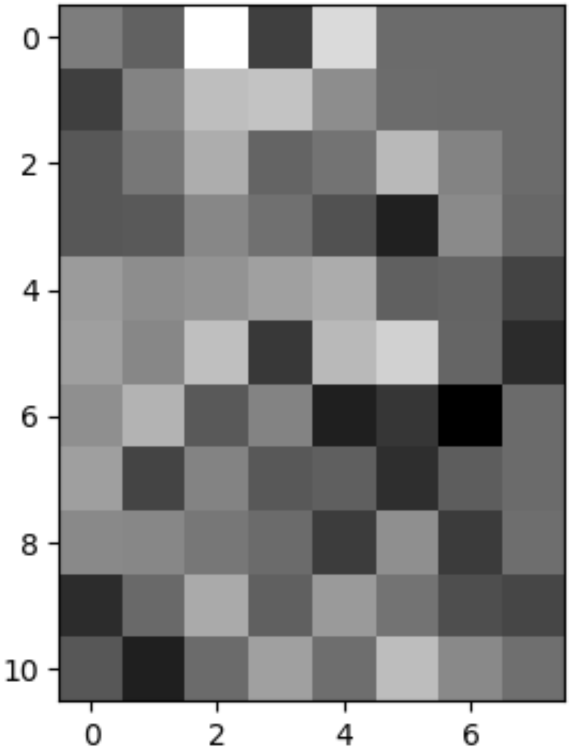
x=2037:2047, y=143:153 around HotPixel(x=2042.0, y=148.0, value=296.9821874999998)



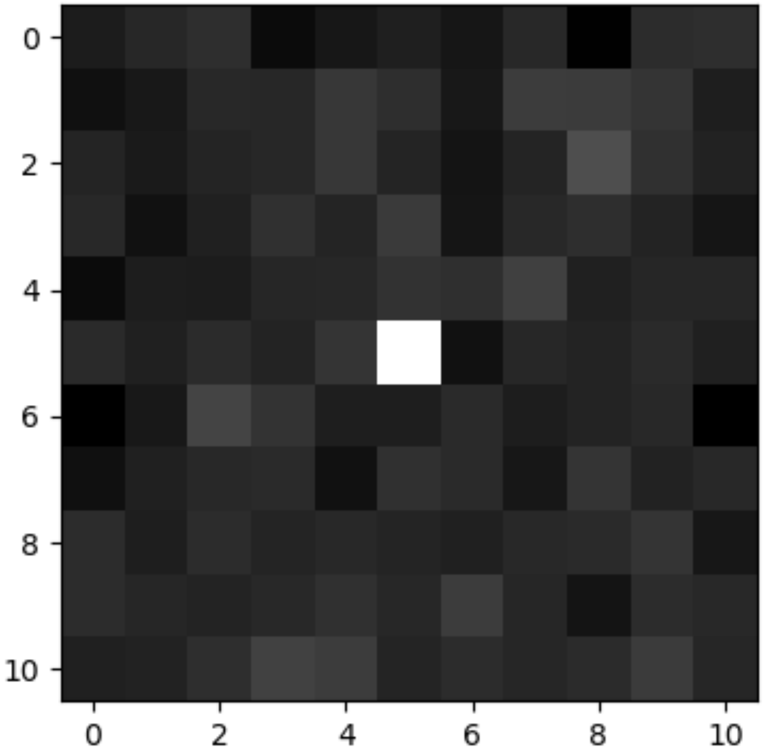
x=2039:2047, y=143:153 around HotPixel(x=2044.0, y=148.0, value=221.2706249999992)

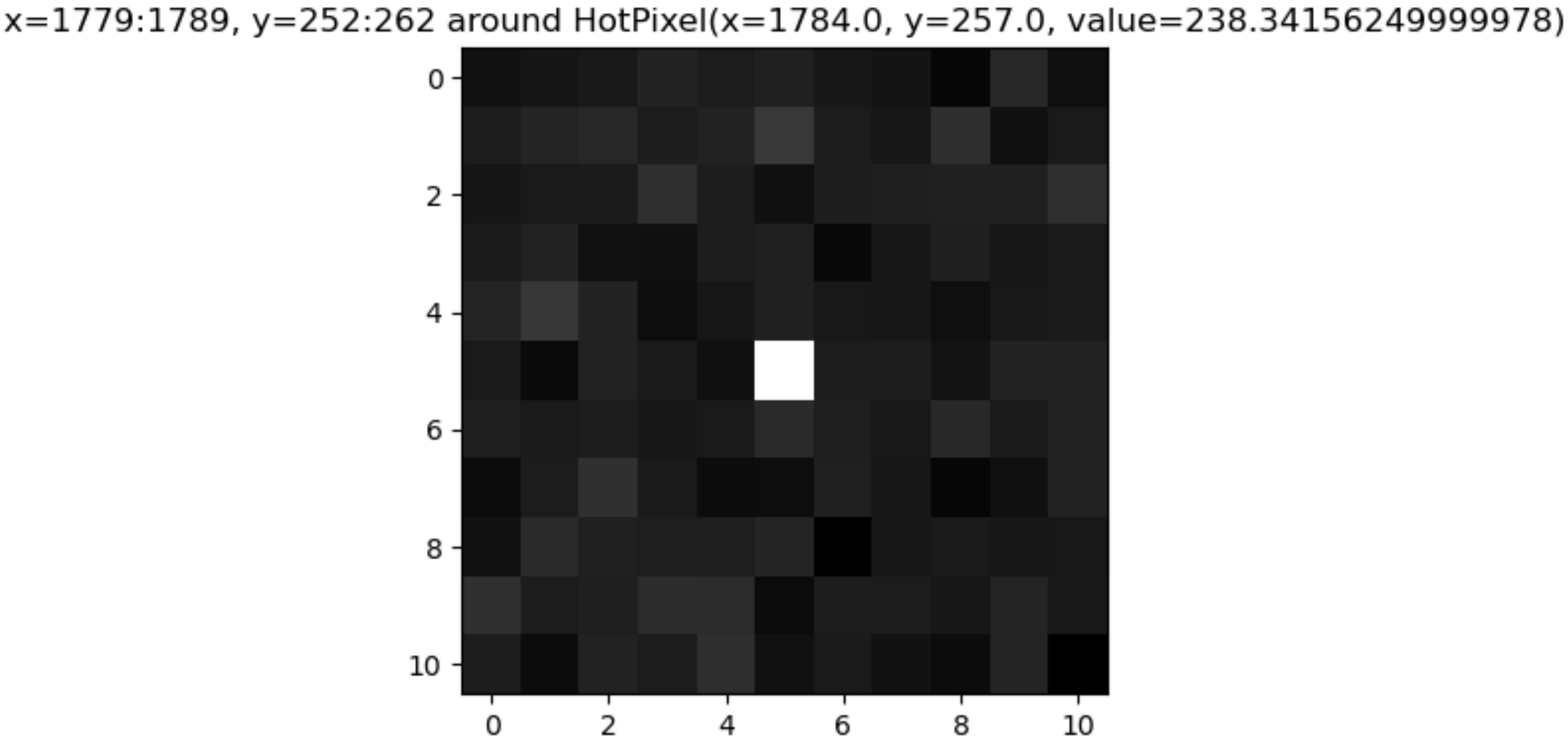


x=2040:2047, y=148:158 around HotPixel(x=2045.0, y=153.0, value=202.9399999999996)

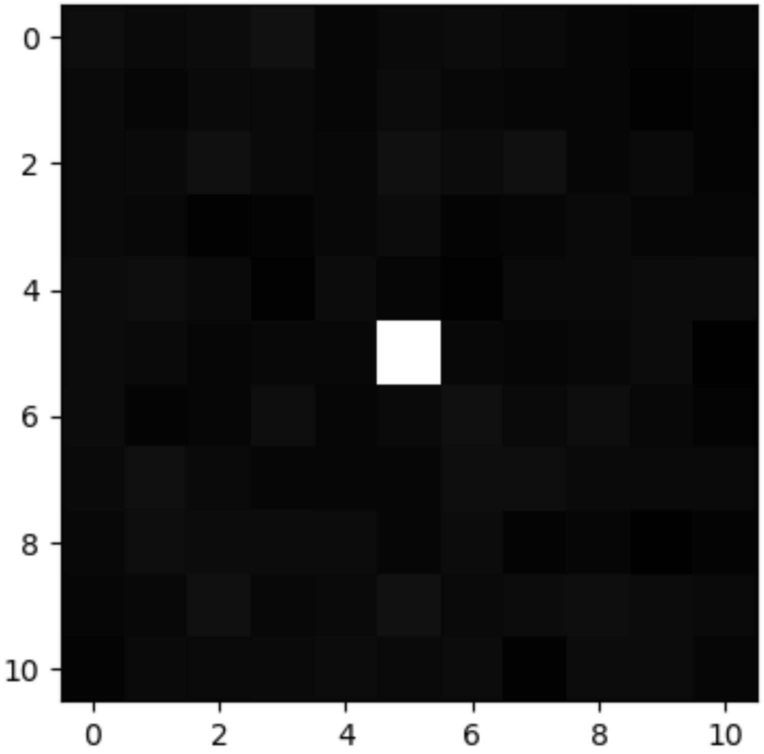


x=416:426, y=164:174 around HotPixel(x=421.0, y=169.0, value=274.4815625000001)

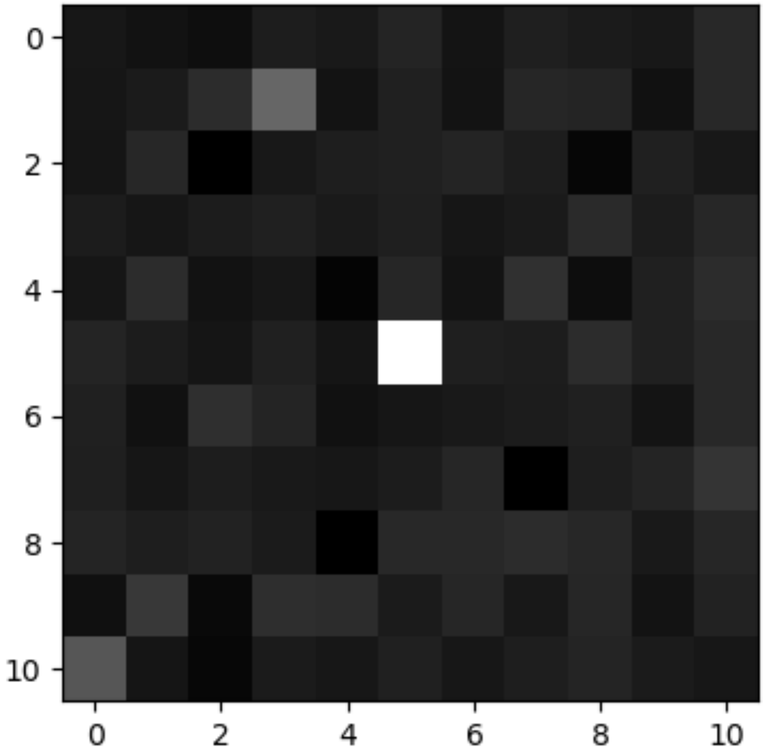


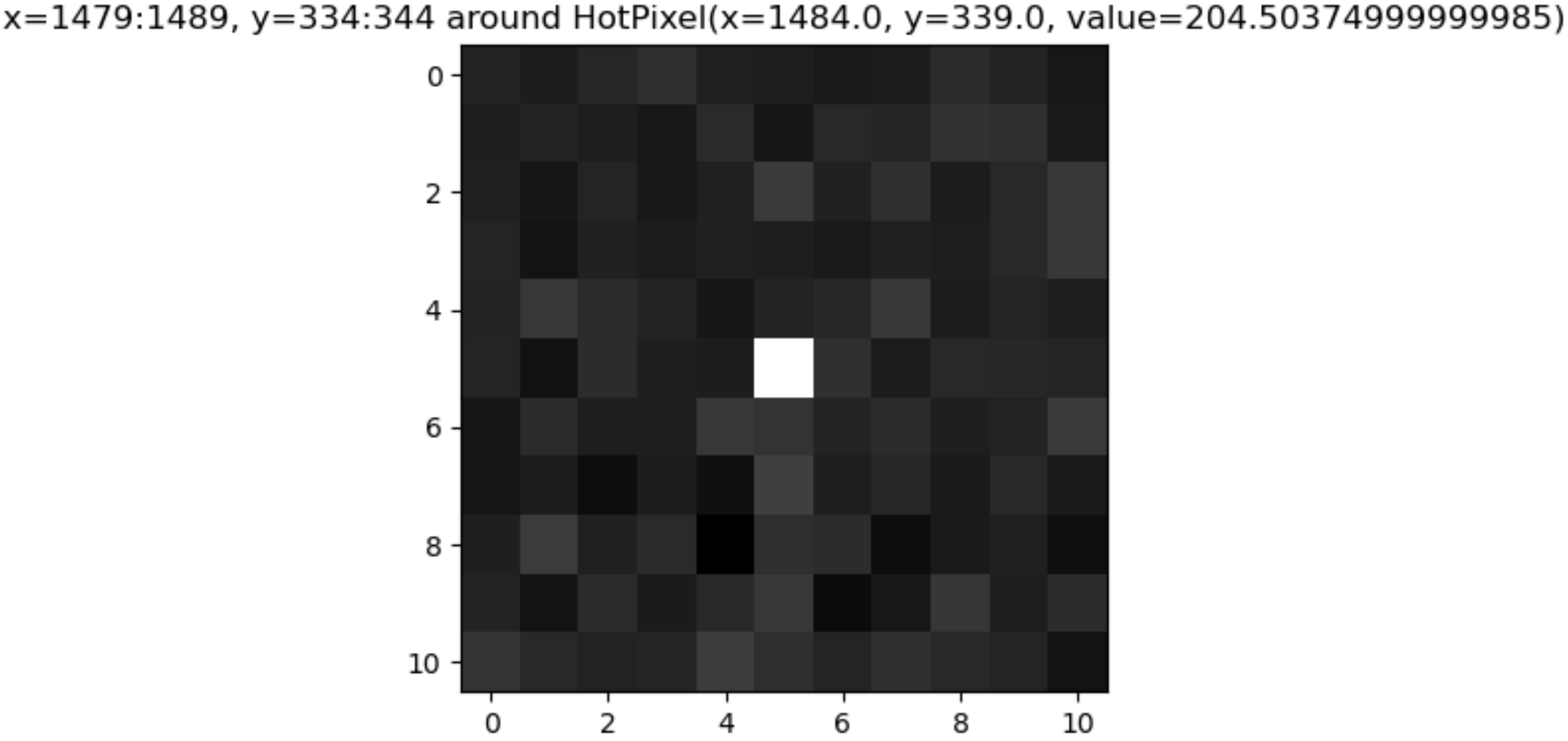

x=1779:1789, y=252:262 around HotPixel(x=1784.0, y=257.0, value=238.34156249999978)
0
2
4
6
8
10
0
2
4
6
8
10

x=33:43, y=273:283 around HotPixel(x=38.0, y=278.0, value=829.2653124999999)

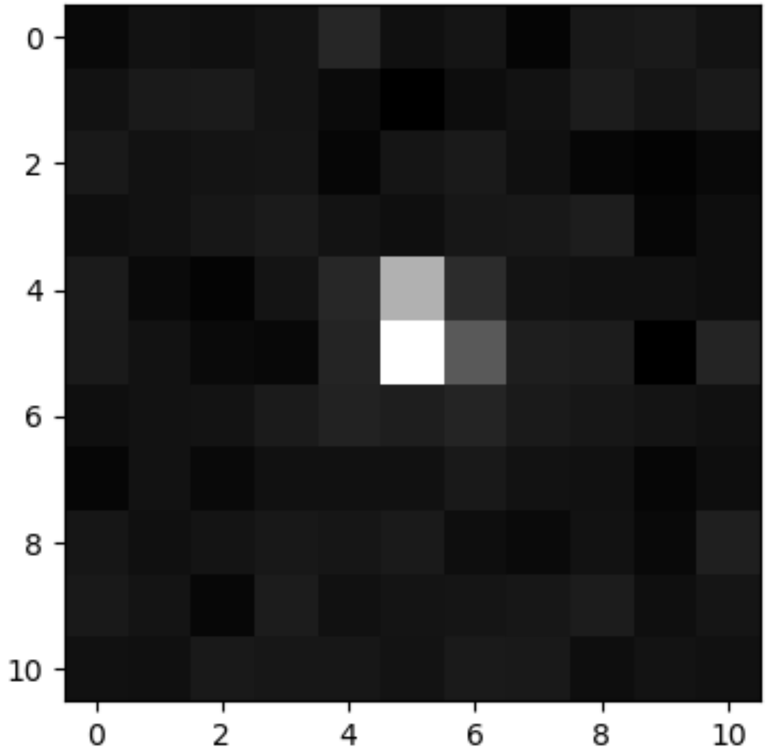


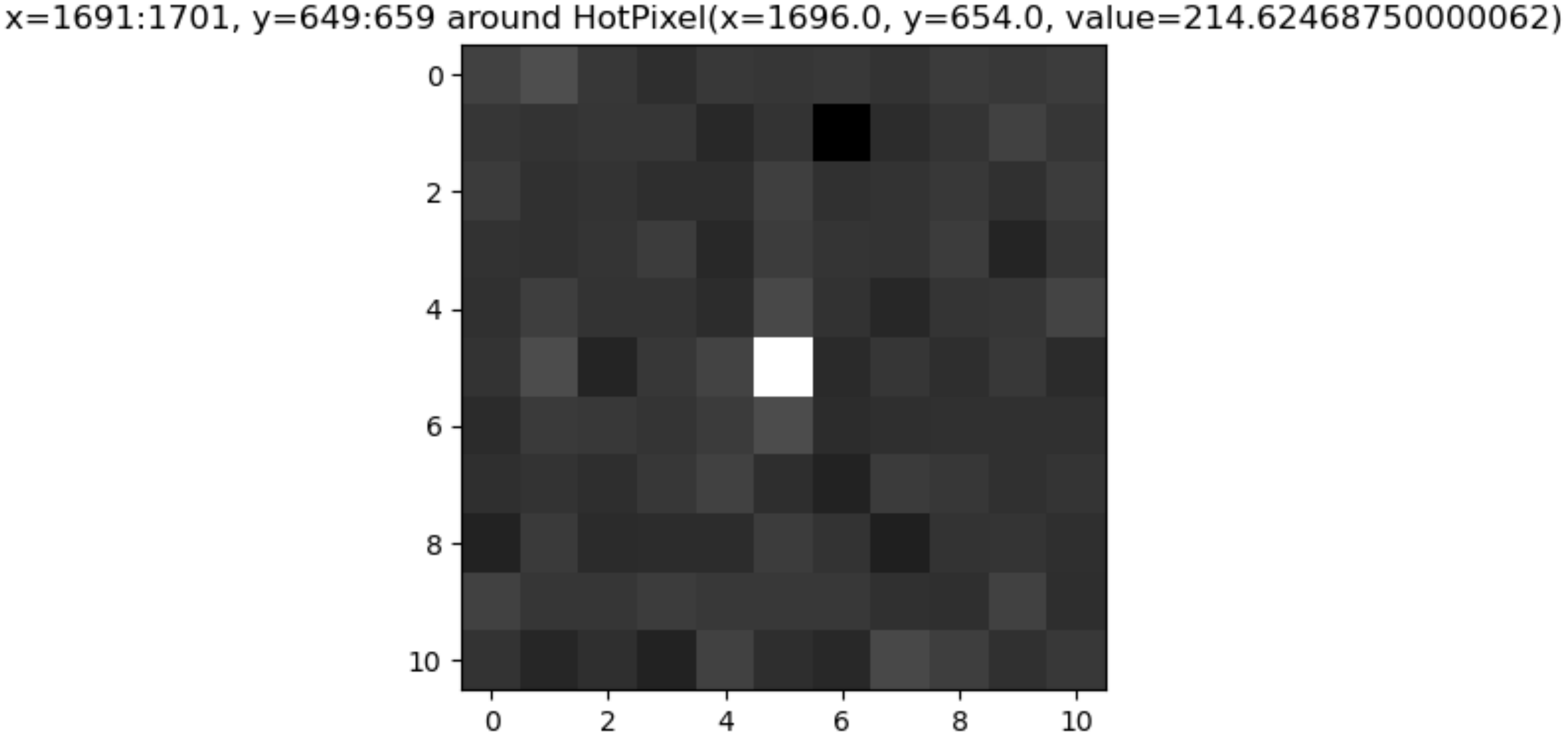
x=477:487, y=304:314 around HotPixel(x=482.0, y=309.0, value=211.80124999999953)

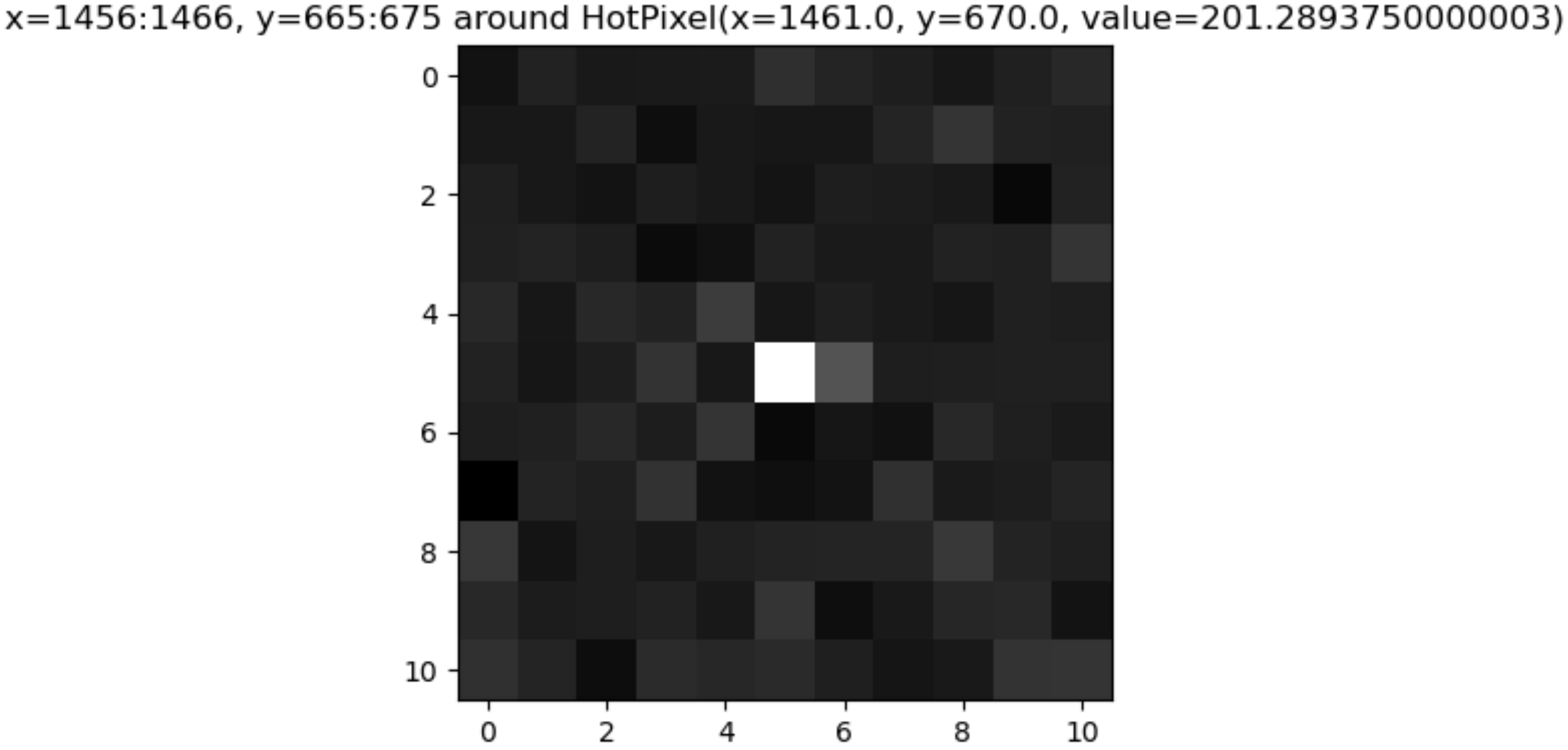



x=1479:1489, y=334:344 around HotPixel(x=1484.0, y=339.0, value=204.50374999999985)
0
2
4
6
8
10
0
2
4
6
8
10

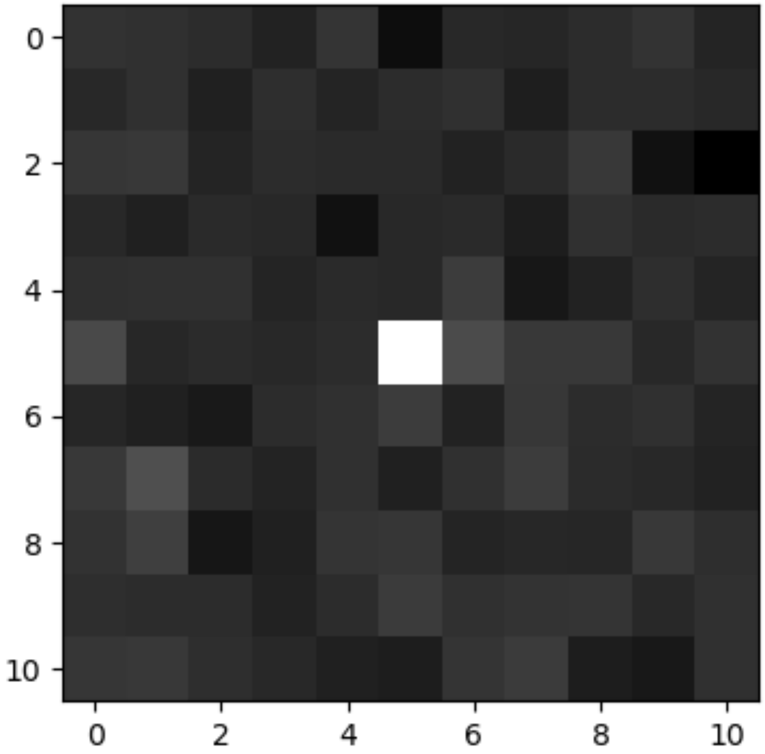
x=1430:1440, y=598:608 around HotPixel(x=1435.0, y=603.0, value=344.025)



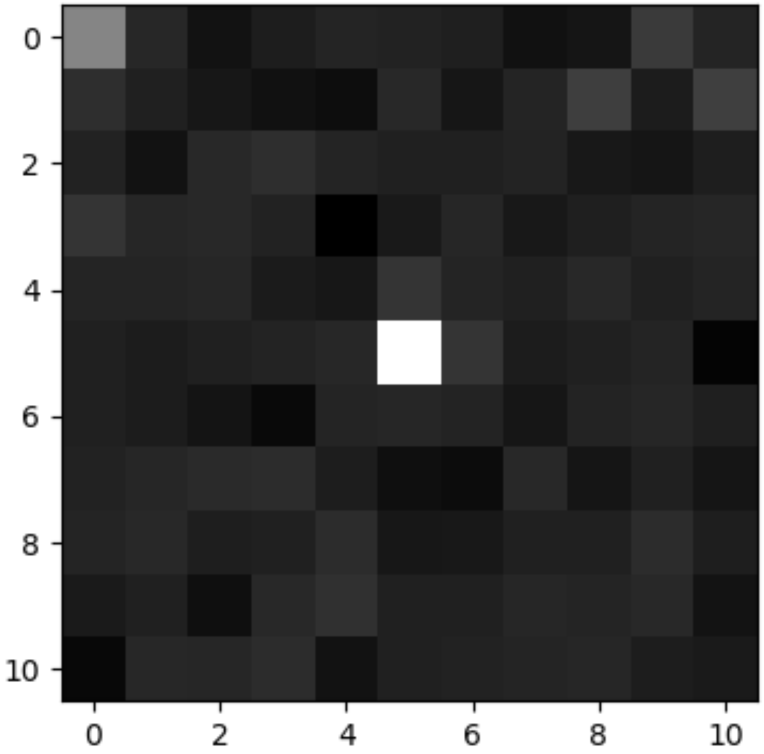

x=1691:1701, y=649:659 around HotPixel(x=1696.0, y=654.0, value=214.62468750000062)
0
2
4
6
8
10
0
2
4
6
8
10

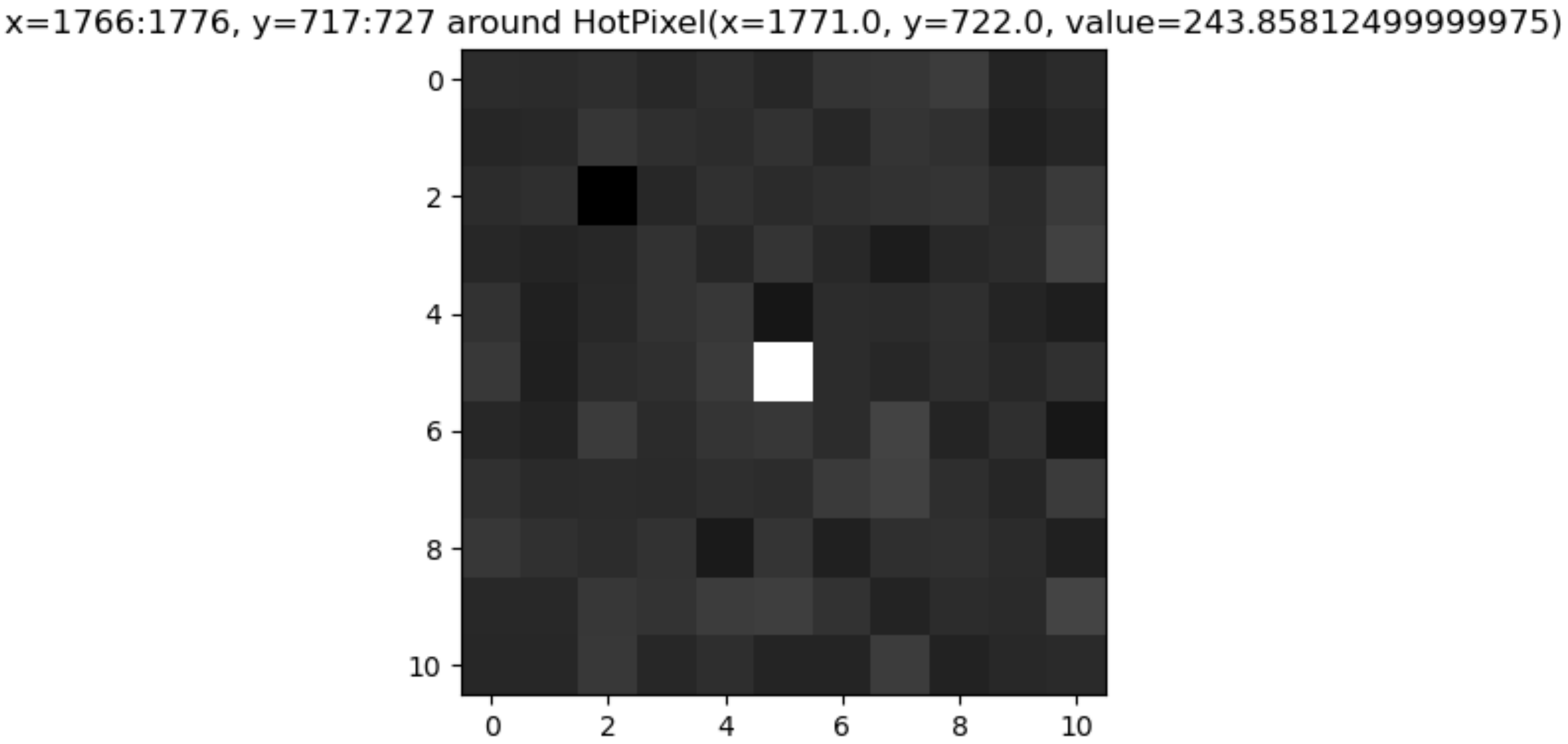

x=1456:1466, y=665:675 around HotPixel(x=1461.0, y=670.0, value=201.2893750000003)
0
2
4
6
8
10
0
2
4
6
8
10

x=1220:1230, y=708:718 around HotPixel(x=1225.0, y=713.0, value=206.02406249999967)



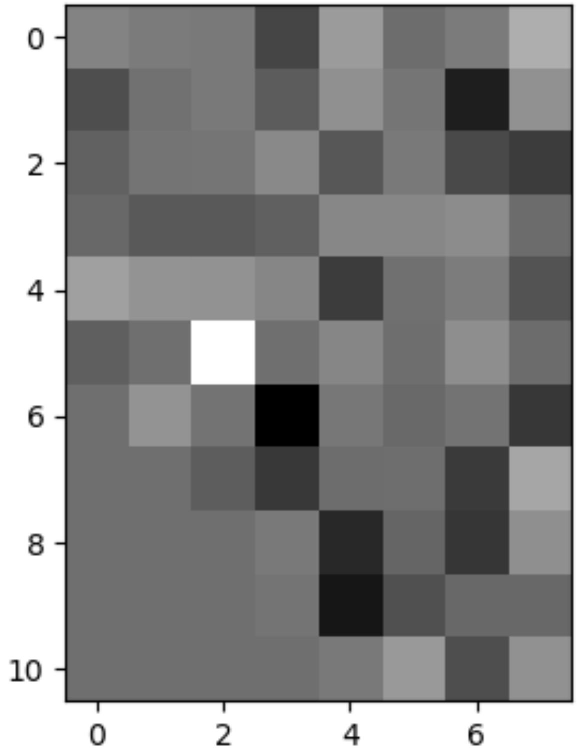
x=1008:1018, y=711:721 around HotPixel(x=1013.0, y=716.0, value=204.9815625000001)



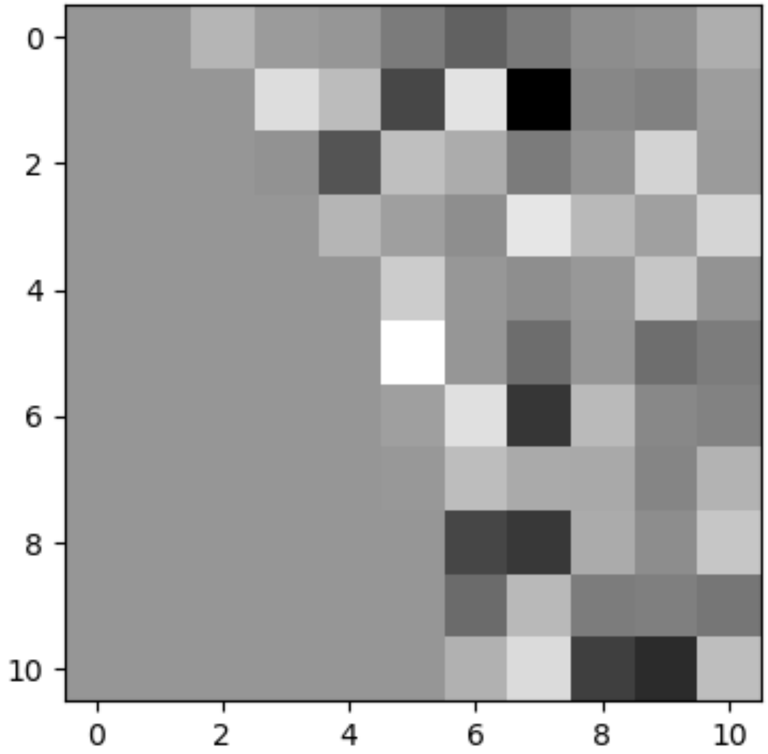


x=1766:1776, y=717:727 around HotPixel(x=1771.0, y=722.0, value=243.85812499999975)

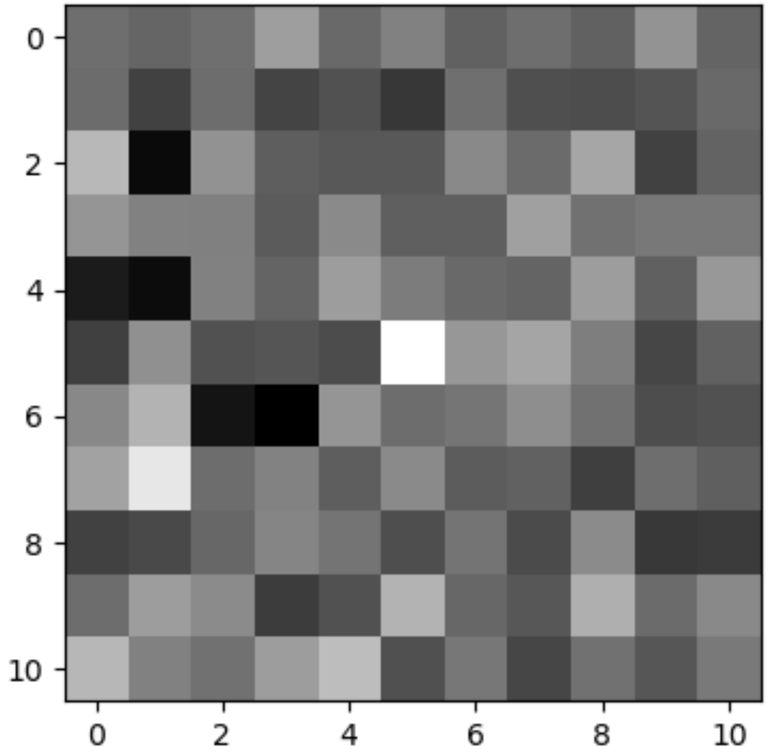
x=0:7, y=768:778 around HotPixel(x=2.0, y=773.0, value=317.09375)



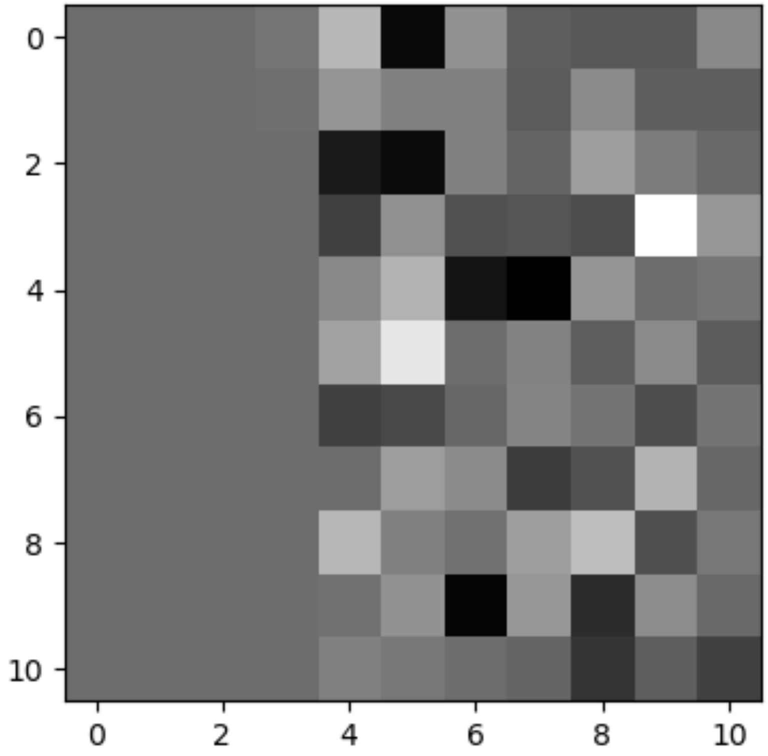
x=3:13, y=779:789 around HotPixel(x=8.0, y=784.0, value=214.92875000000004)



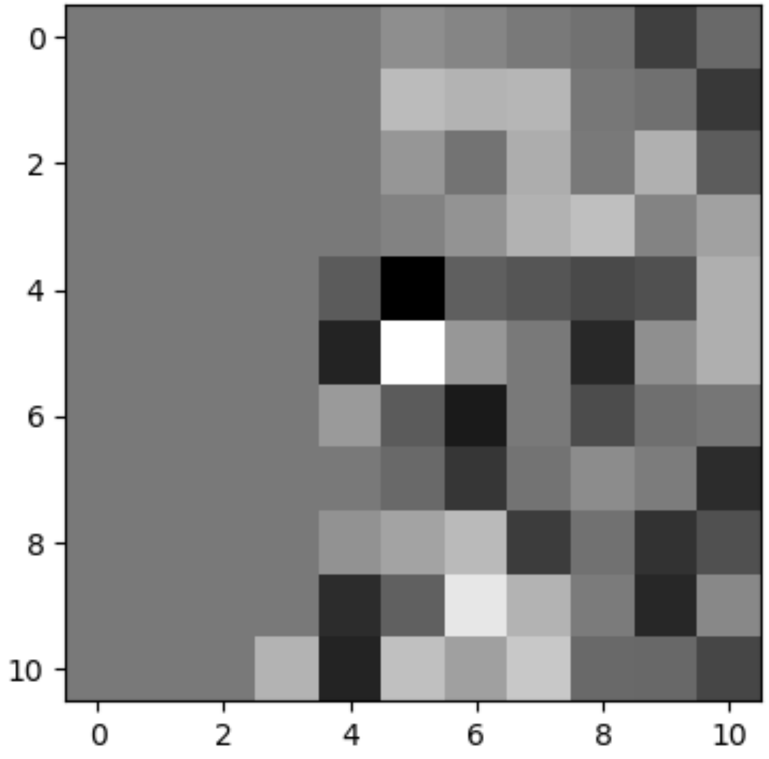
x=9:19, y=783:793 around HotPixel(x=14.0, y=788.0, value=293.7678125000002)



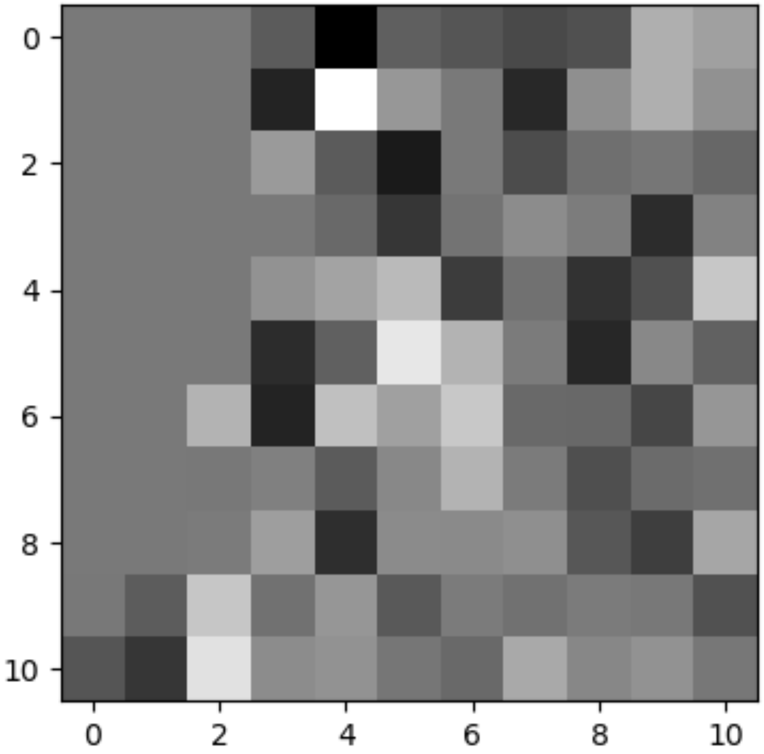
x=5:15, y=785:795 around HotPixel(x=10.0, y=790.0, value=244.0753125000001)

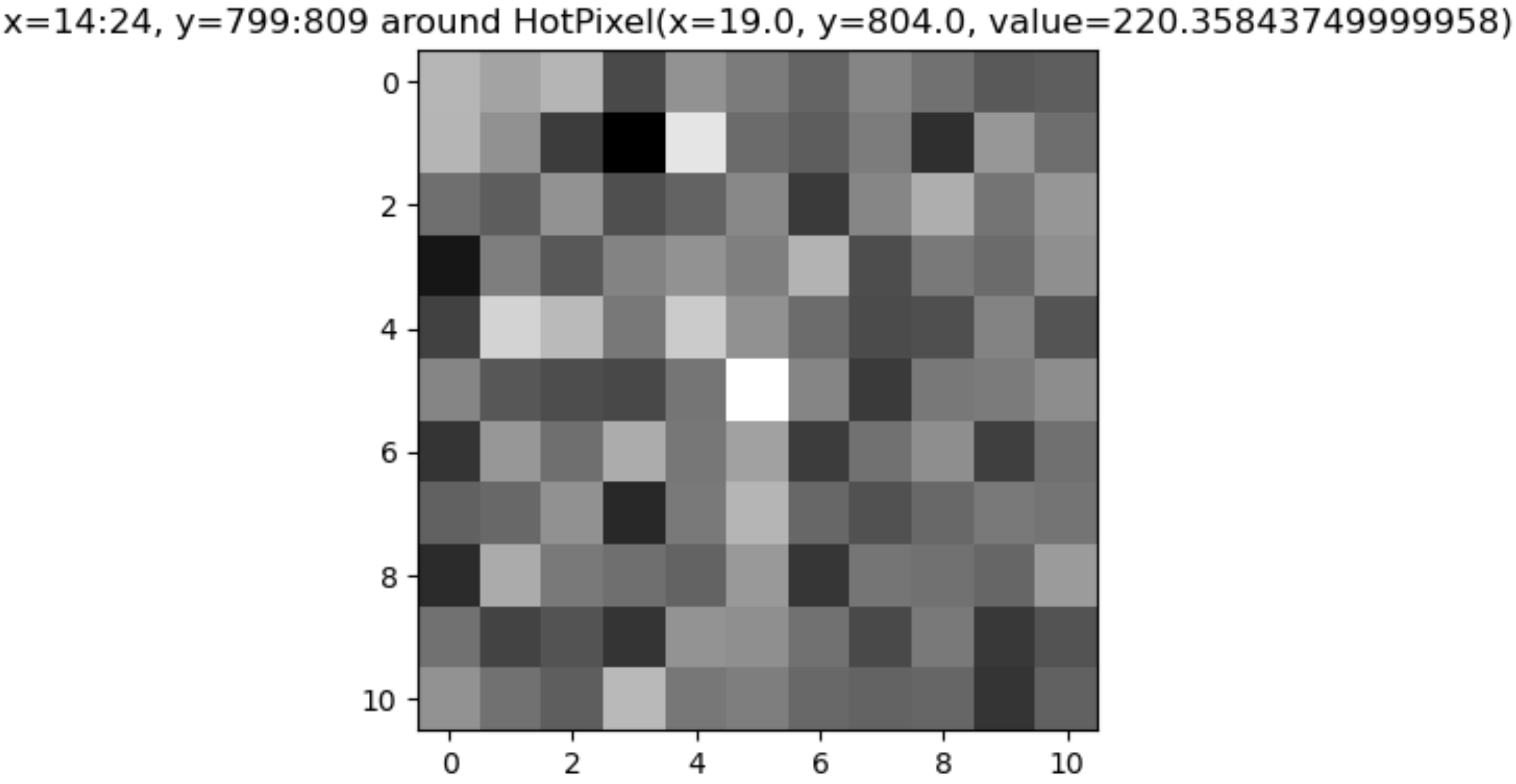


x=4:14, y=795:805 around HotPixel(x=9.0, y=800.0, value=258.453125)

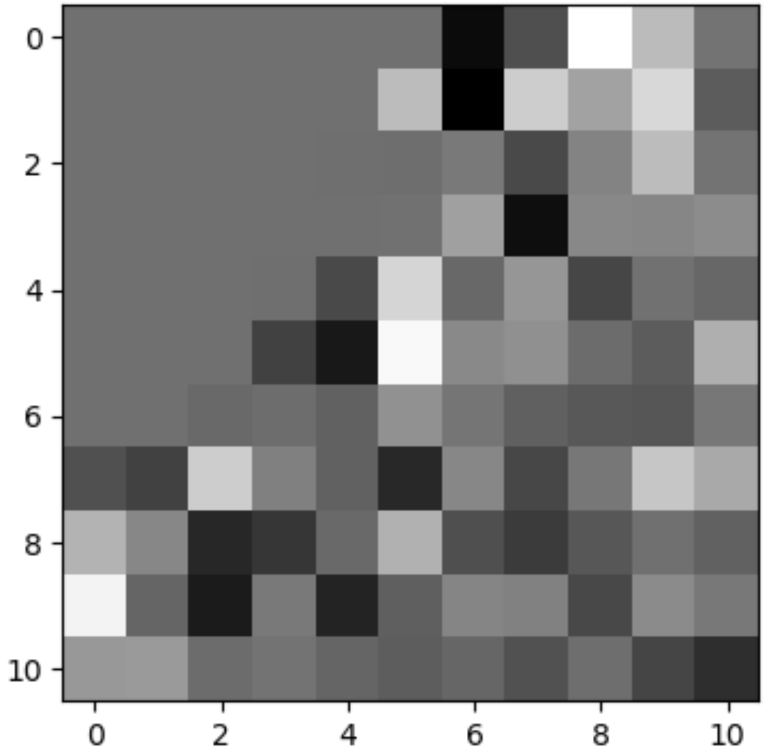


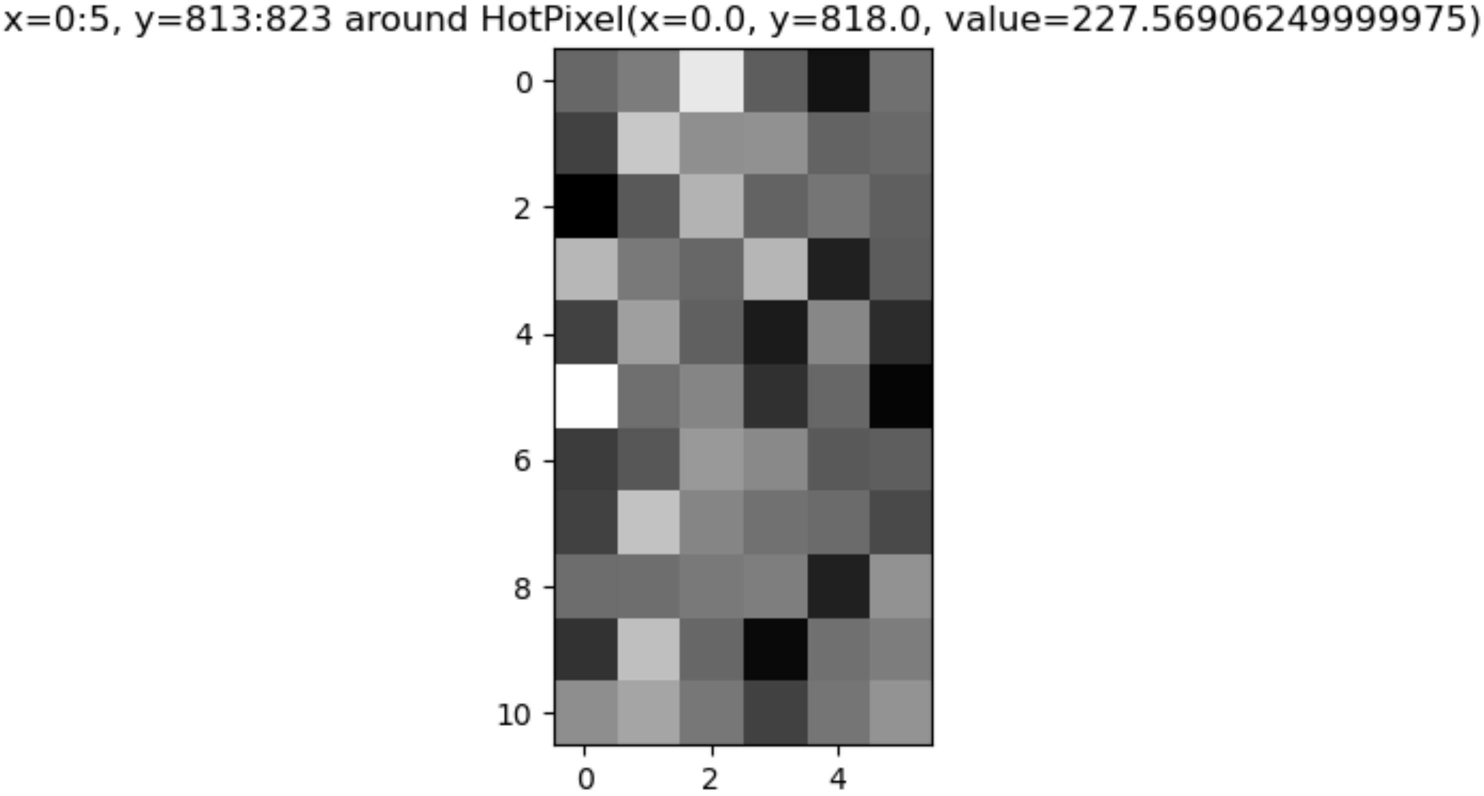
x=5:15, y=799:809 around HotPixel(x=10.0, y=804.0, value=210.88906250000036)



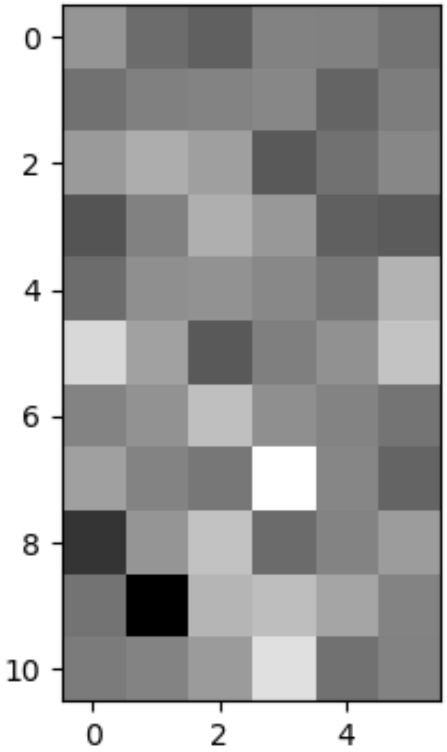

x=14:24, y=799:809 around HotPixel(x=19.0, y=804.0, value=220.35843749999958)
0
2
4
6
8
10
0
2
4
6
8
10

x=2:12, y=804:814 around HotPixel(x=7.0, y=809.0, value=200.7681249999996)

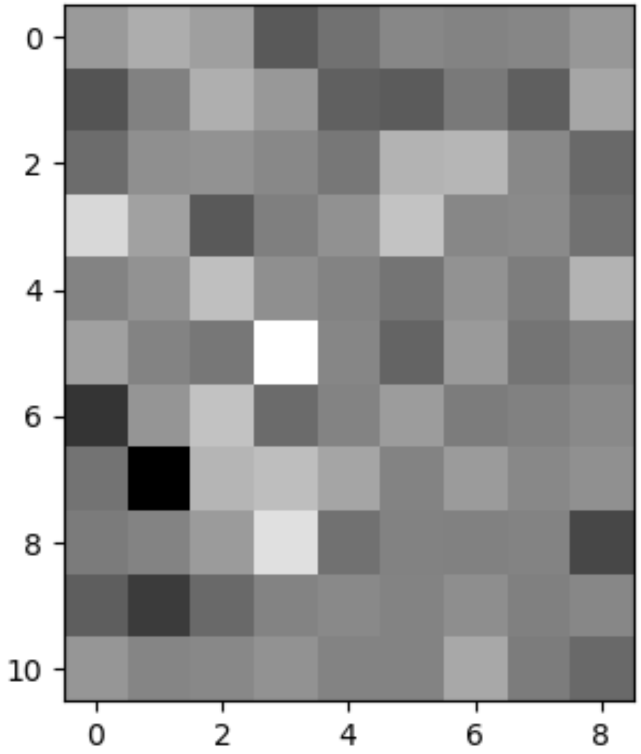



x=0:5, y=813:823 around HotPixel(x=0.0, y=818.0, value=227.56906249999975)
0
2
4
6
8
10
0
2
4

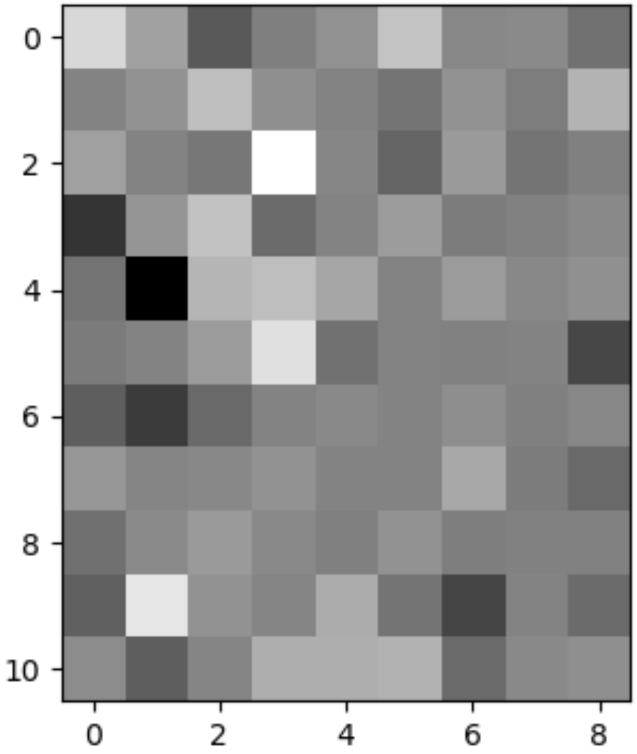
x=0:5, y=864:874 around HotPixel(x=0.0, y=869.0, value=233.17249999999967)



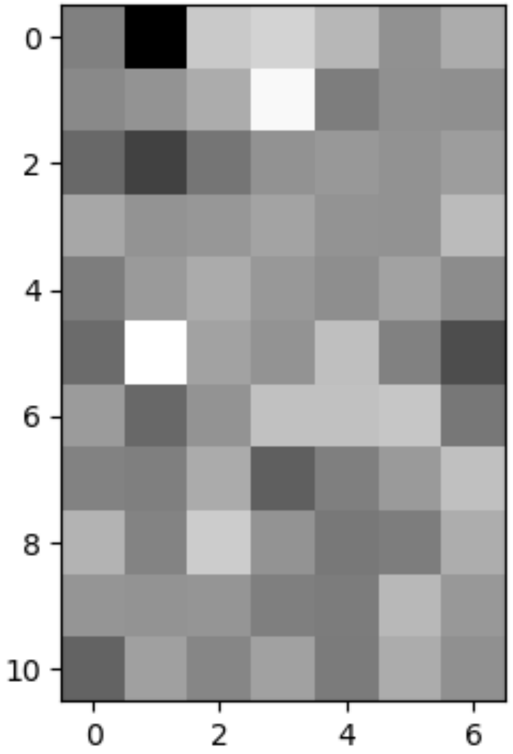
x=0:8, y=866:876 around HotPixel(x=3.0, y=871.0, value=346.4140625)

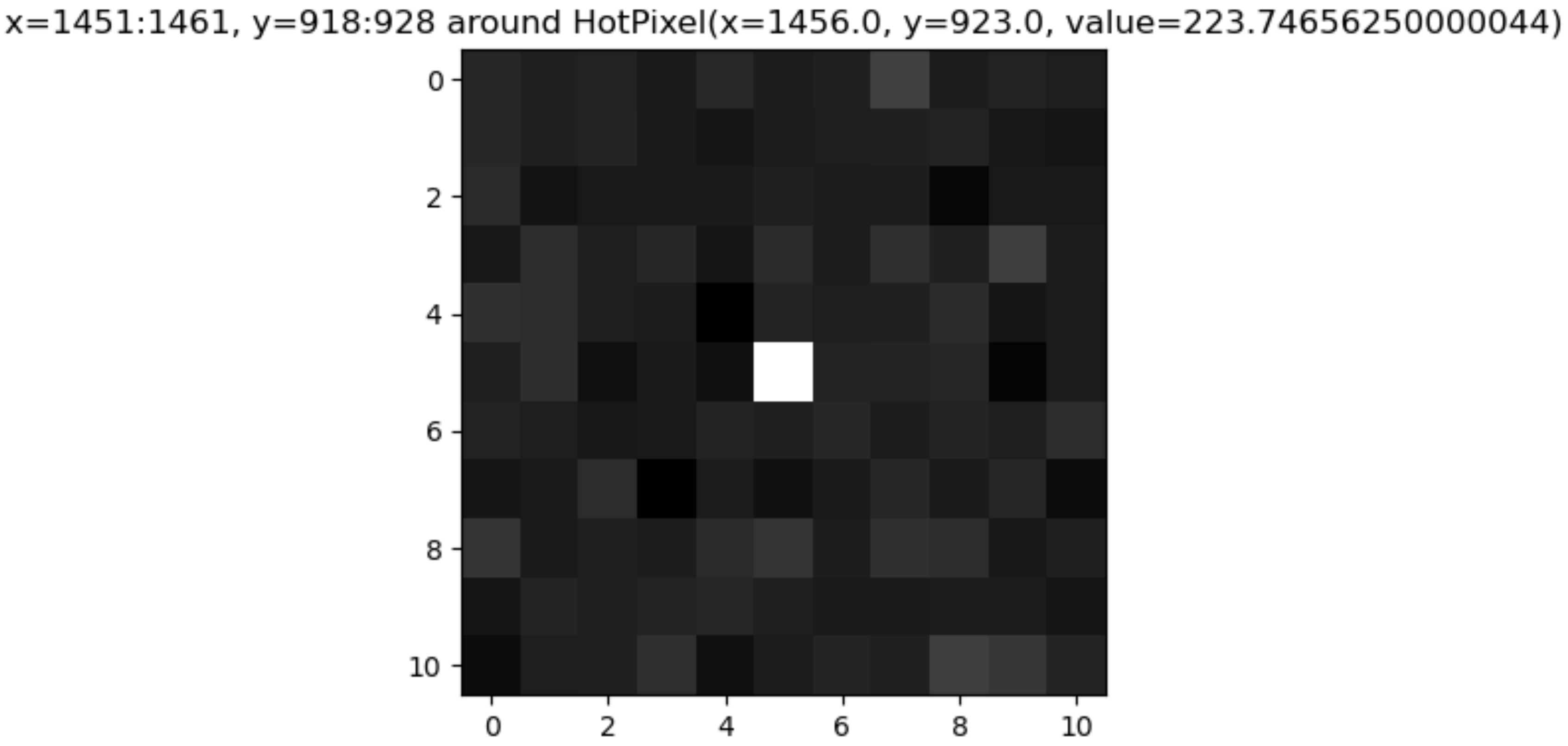


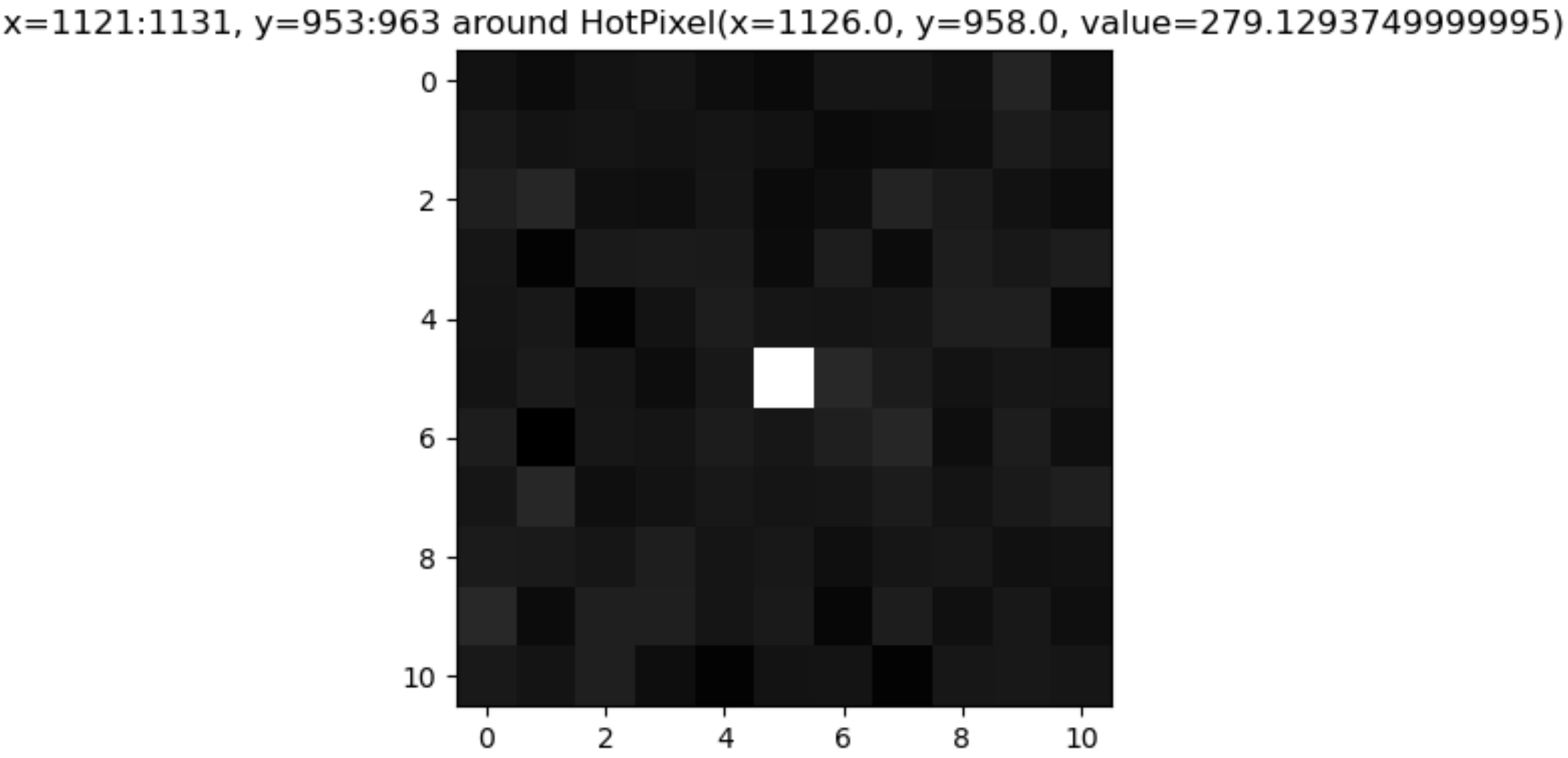
x=0:8, y=869:879 around HotPixel(x=3.0, y=874.0, value=256.4115624999995)



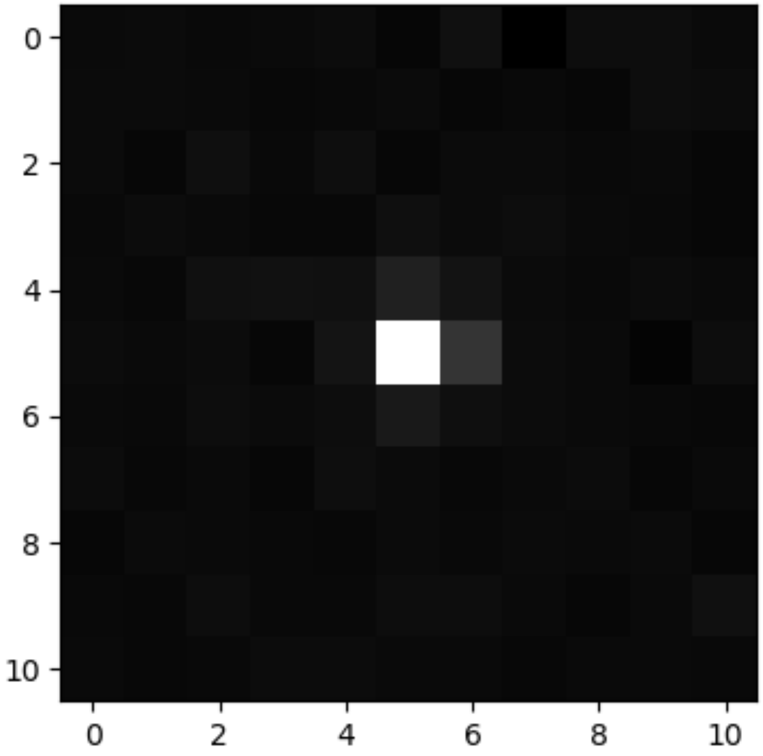
x=0:6, y=873:883 around HotPixel(x=1.0, y=878.0, value=273.2218750000002)



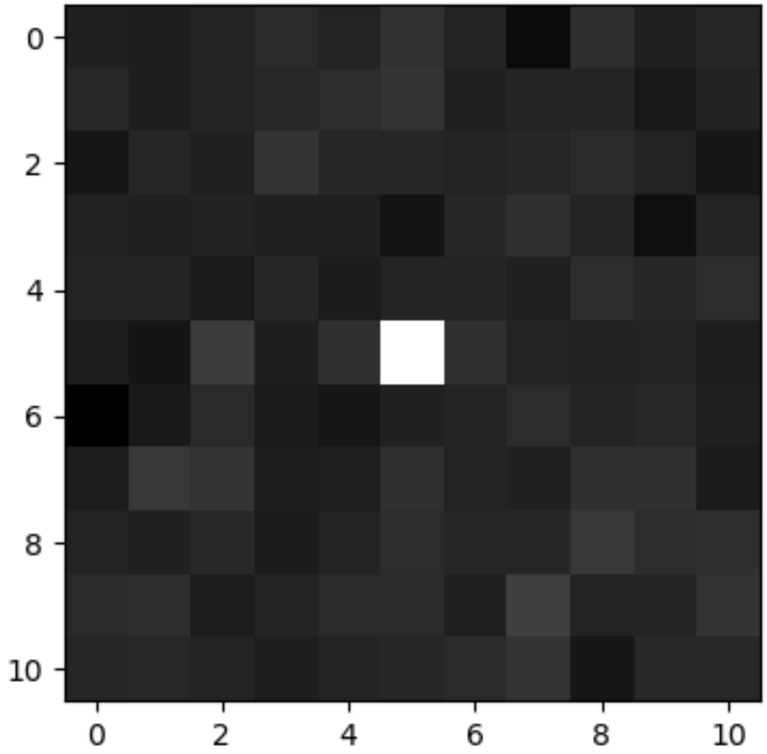

x=1451:1461, y=918:928 around HotPixel(x=1456.0, y=923.0, value=223.74656250000044)
0
2
4
6
8
10
0
2
4
6
8
10

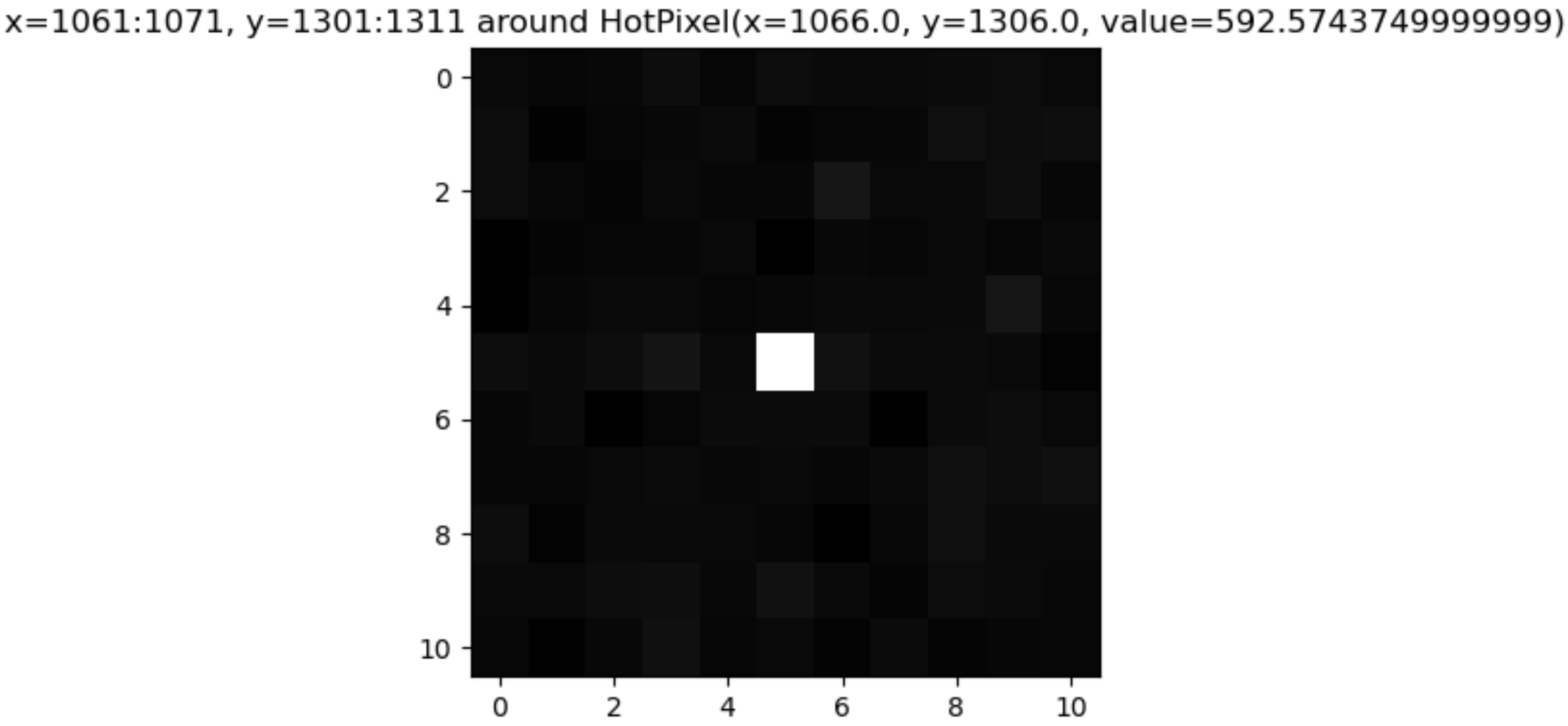

x=1121:1131, y=953:963 around HotPixel(x=1126.0, y=958.0, value=279.1293749999995)
0
2
4
6
8
10
0
2
4
6
8
10

x=58:68, y=1112:1122 around HotPixel(x=63.0, y=1117.0, value=1240.2709375)

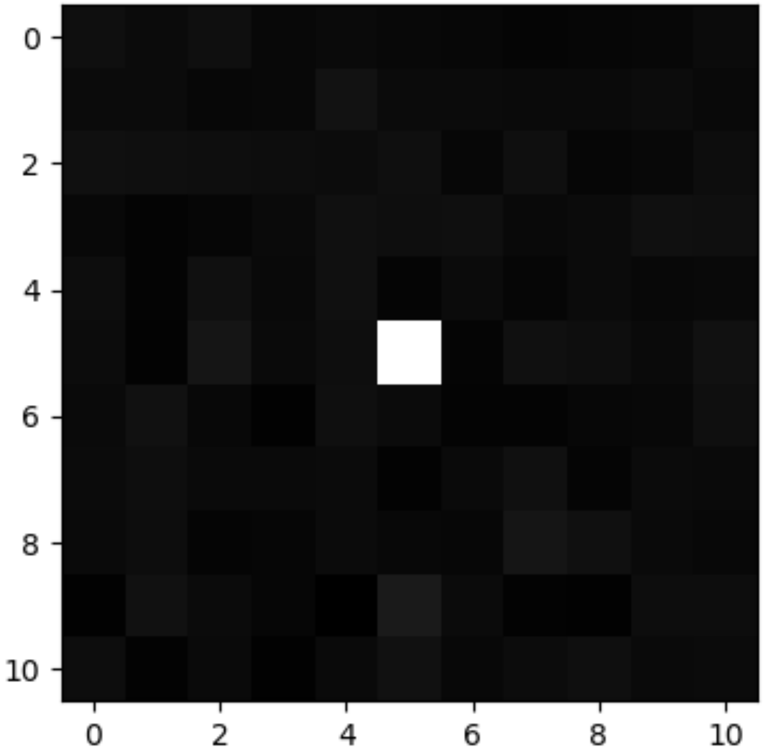


x=856:866, y=1174:1184 around HotPixel(x=861.0, y=1179.0, value=260.625)

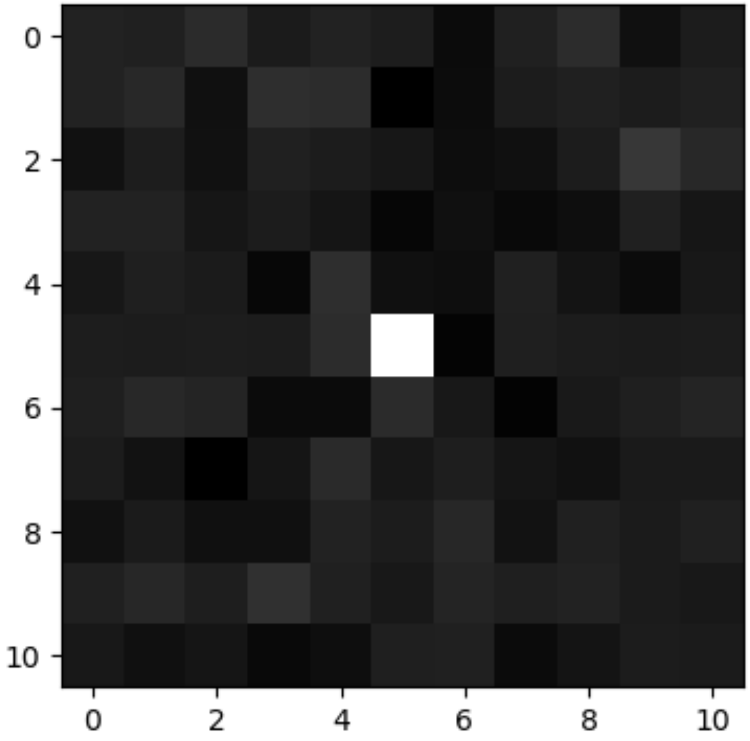



x=1061:1071, y=1301:1311 around HotPixel(x=1066.0, y=1306.0, value=592.5743749999999)
0
2
4
6
8
10
0
2
4
6
8
10

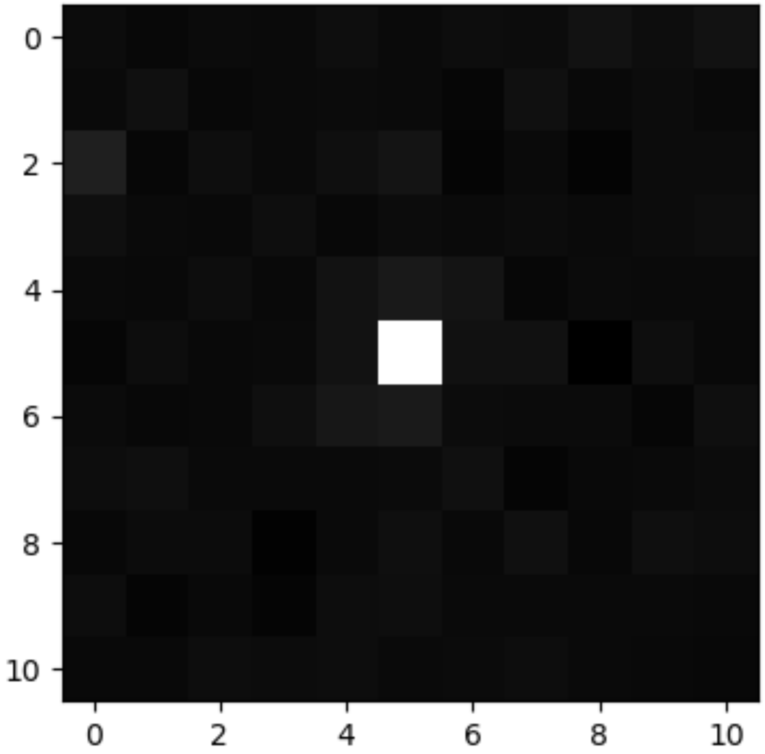
x=173:183, y=1313:1323 around HotPixel(x=178.0, y=1318.0, value=566.5987500000001)



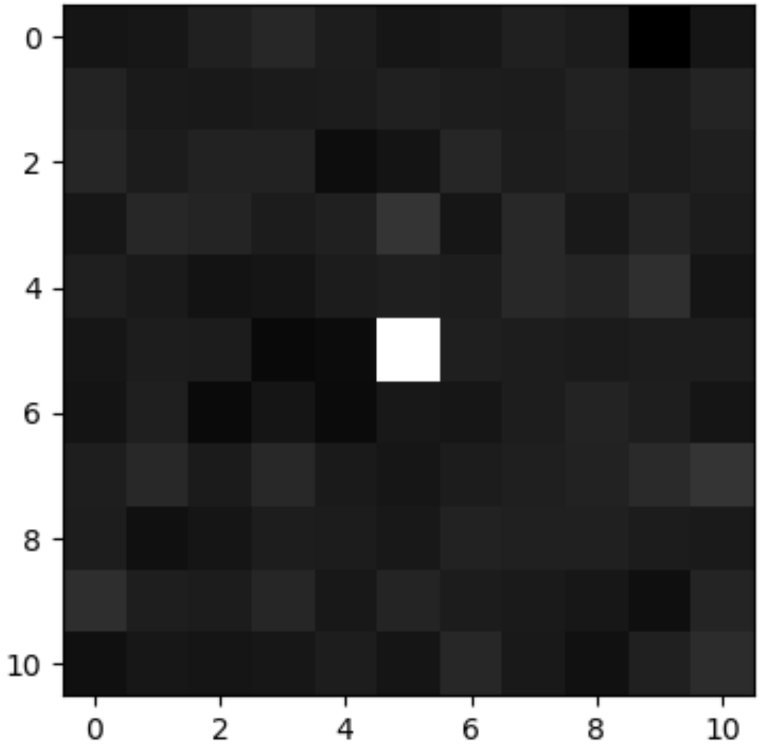
x=1847:1857, y=1332:1342 around HotPixel(x=1852.0, y=1337.0, value=202.07125000000008)



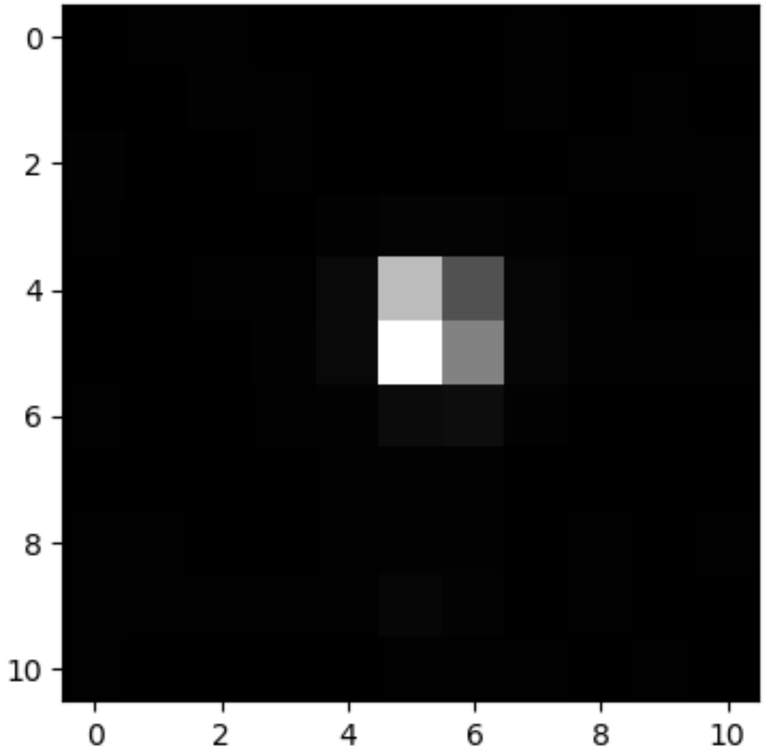
x=1566:1576, y=1471:1481 around HotPixel(x=1571.0, y=1476.0, value=694.044375)



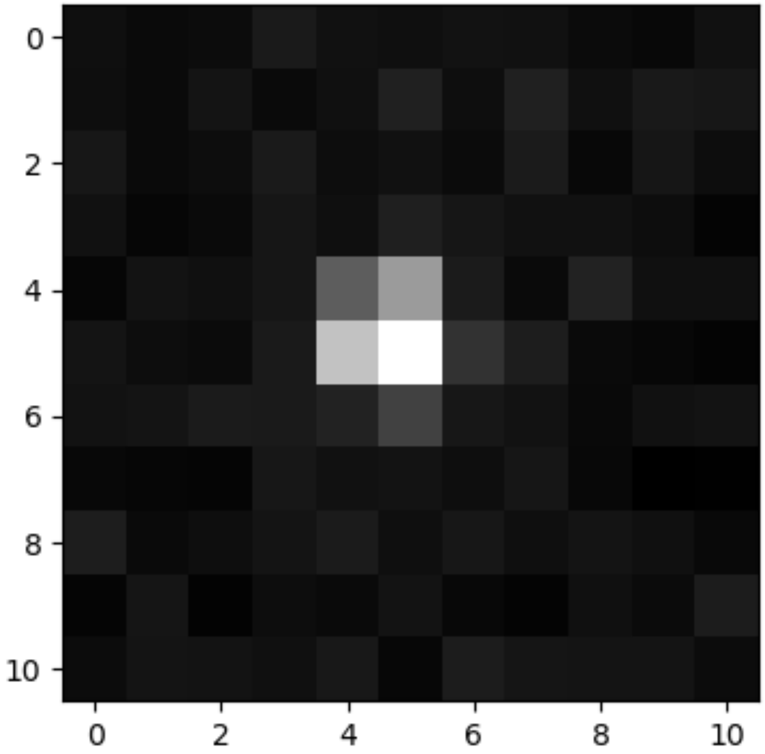
x=1557:1567, y=1663:1673 around HotPixel(x=1562.0, y=1668.0, value=281.4315624999999)



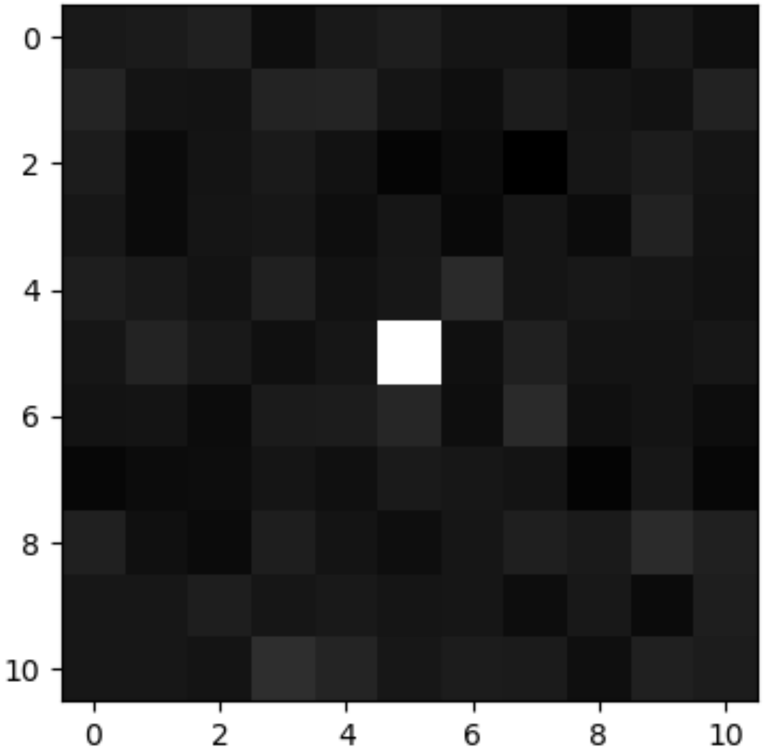
x=434:444, y=1744:1754 around HotPixel(x=439.0, y=1749.0, value=5163.5025)



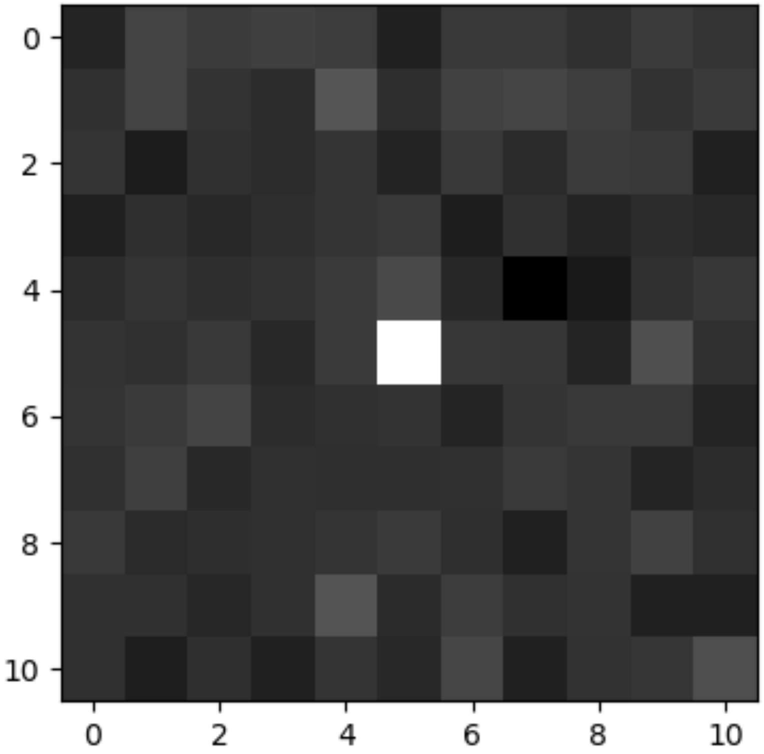
x=346:356, y=1923:1933 around HotPixel(x=351.0, y=1928.0, value=325.34687499999995)



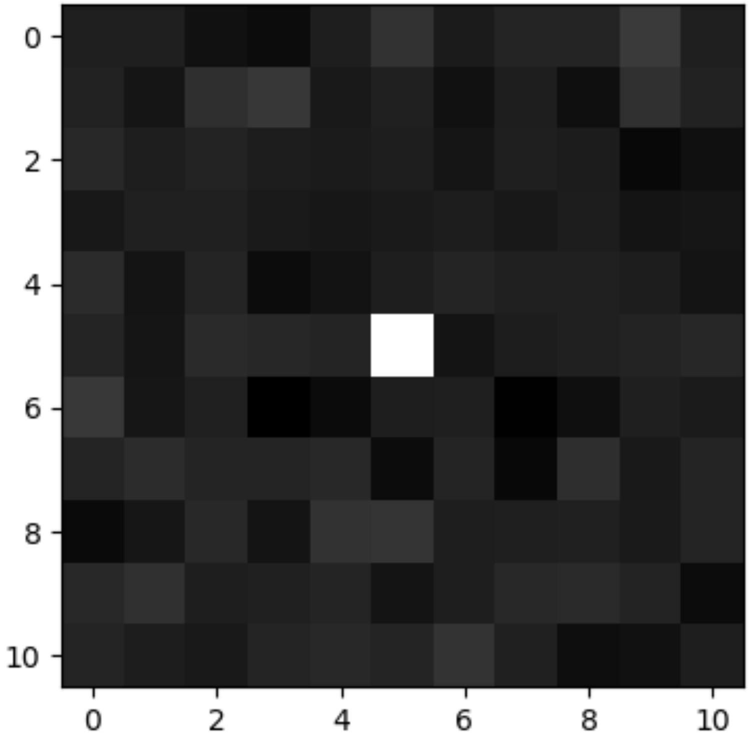
x=831:841, y=1924:1934 around HotPixel(x=836.0, y=1929.0, value=282.3003124999998)



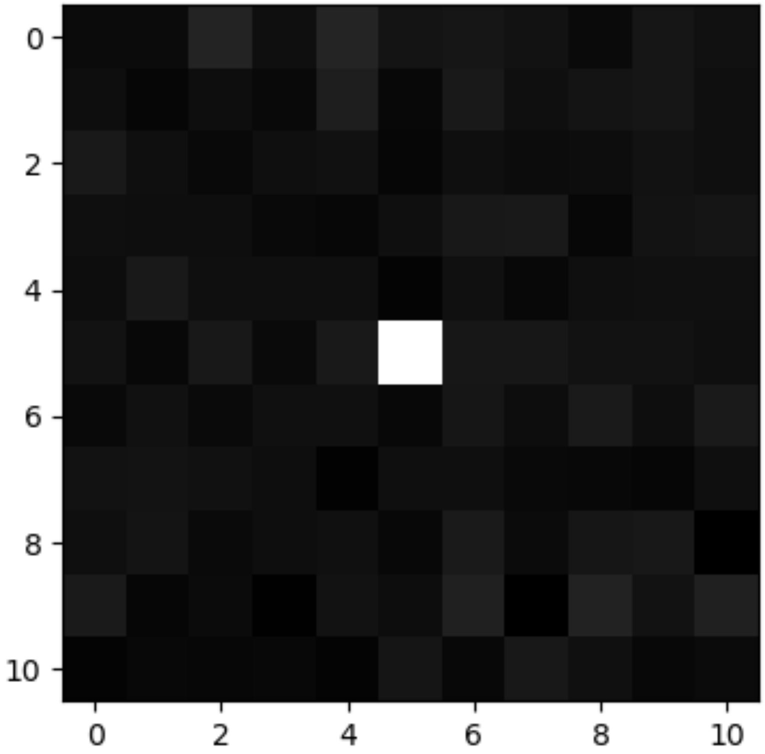
x=46:56, y=1940:1950 around HotPixel(x=51.0, y=1945.0, value=205.5896875000003)



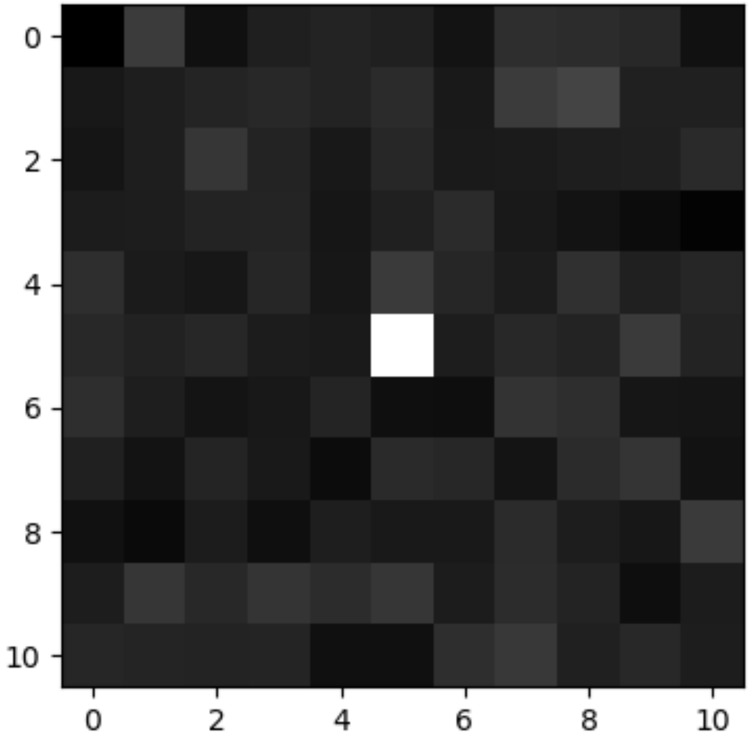
x=1344:1354, y=1956:1966 around HotPixel(x=1349.0, y=1961.0, value=201.94093749999956)



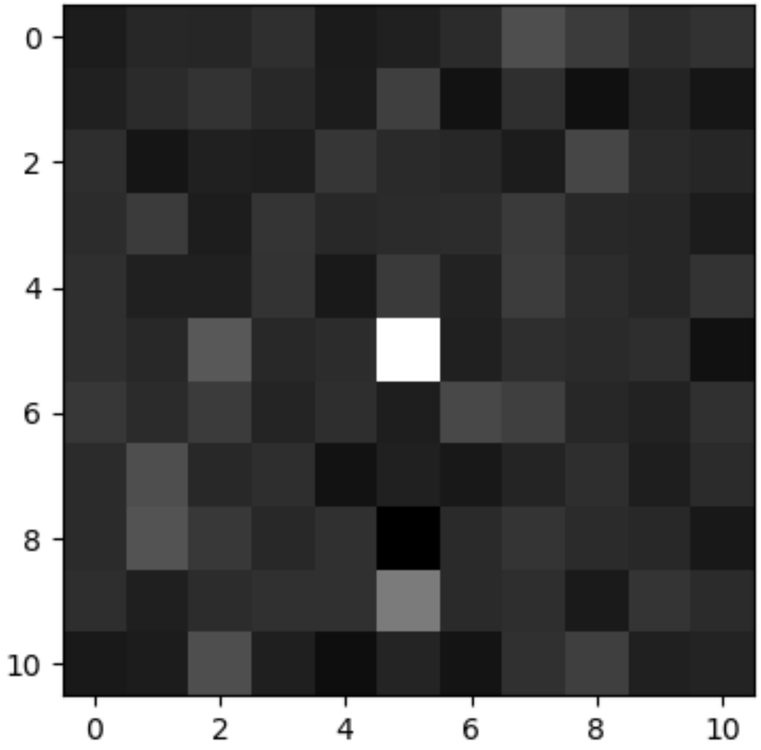
x=934:944, y=1979:1989 around HotPixel(x=939.0, y=1984.0, value=334.64250000000015)



x=1774:1784, y=2000:2010 around HotPixel(x=1779.0, y=2005.0, value=202.11468749999995)



x=2017:2027, y=2018:2028 around HotPixel(x=2022.0, y=2023.0, value=223.96374999999999)


0
2
4
6
8
10
0
2
4
6
8
10