SLA1 Camera Characterization

Hot Pixel Leaders in 30s Darks

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

This notebook combines the darks into a master dark, and then subtracts them from each individual dark.

Then it classifies all pixels whose values exceed 200 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.

Because a large number of pixels pass these cuts, only a subset of the leaders is displayed.

In [1]:
```
# THIS COMMENT IS THE LONGEST A LINE CAN BE AND STILL RENDER COMPLETELY WHEN

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_correc
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-transien
analysis_directory = os.path.join(uv_project_directory, 'analyses', '30s_dar

# The path to the first dark on SLA1 is D:/Raw/2024-05-08/03_38_48/Dark30s/0
# 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'

# subdirectory for the 30-second darks (following SharpCap Pro capture direc
```

```
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

# Log stretch utility

def log_stretch_transform(black_point, saturation_range):

    log_saturation_range = log10(saturation_range)

    def fn(pixel_value):
        pixel_value -= black_point
        if pixel_value <= 1.0:
            return 0
        else:
            log_pixel_value = log10(pixel_value)
            if log_pixel_value >= log_saturation_range:
                return 255;
            else:
                return floor(256 * log_pixel_value / log_saturation_range)

    return fn

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

dark_files = ImageFileCollection(dark_directory).files_filtered(include_path

darks = [CCDData.read(file, unit=u.adu) 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 ranging from something like 18000 to 25000 ADU, with some outliers far outside that range.

In [2]:
```
# np.set_printoptions(threshold=sys.maxsize) # Uncommenting this line will c

master_dark.data
```

Out[2]:
```
array([[6.40000e+03, 6.15100e+04, 3.50000e+00, ..., 1.49450e+04,
        1.49765e+04, 1.43900e+04],
       [5.74100e+03, 5.74100e+03, 1.76055e+04, ..., 1.44415e+04,
        1.67170e+04, 1.41960e+04],
       [1.72130e+04, 1.78930e+04, 1.79250e+04, ..., 1.44100e+04,
        1.49765e+04, 1.63280e+04],
       ...,
       [8.01100e+03, 8.90350e+03, 8.34400e+03, ..., 1.55045e+04,
        1.51520e+04, 1.57850e+04],
       [8.82850e+03, 9.12900e+03, 9.48850e+03, ..., 1.85450e+04,
        1.91905e+04, 1.79245e+04],
       [1.04655e+04, 1.83845e+04, 1.13215e+04, ..., 2.78300e+04,
        2.51730e+04, 2.46185e+04]])
```

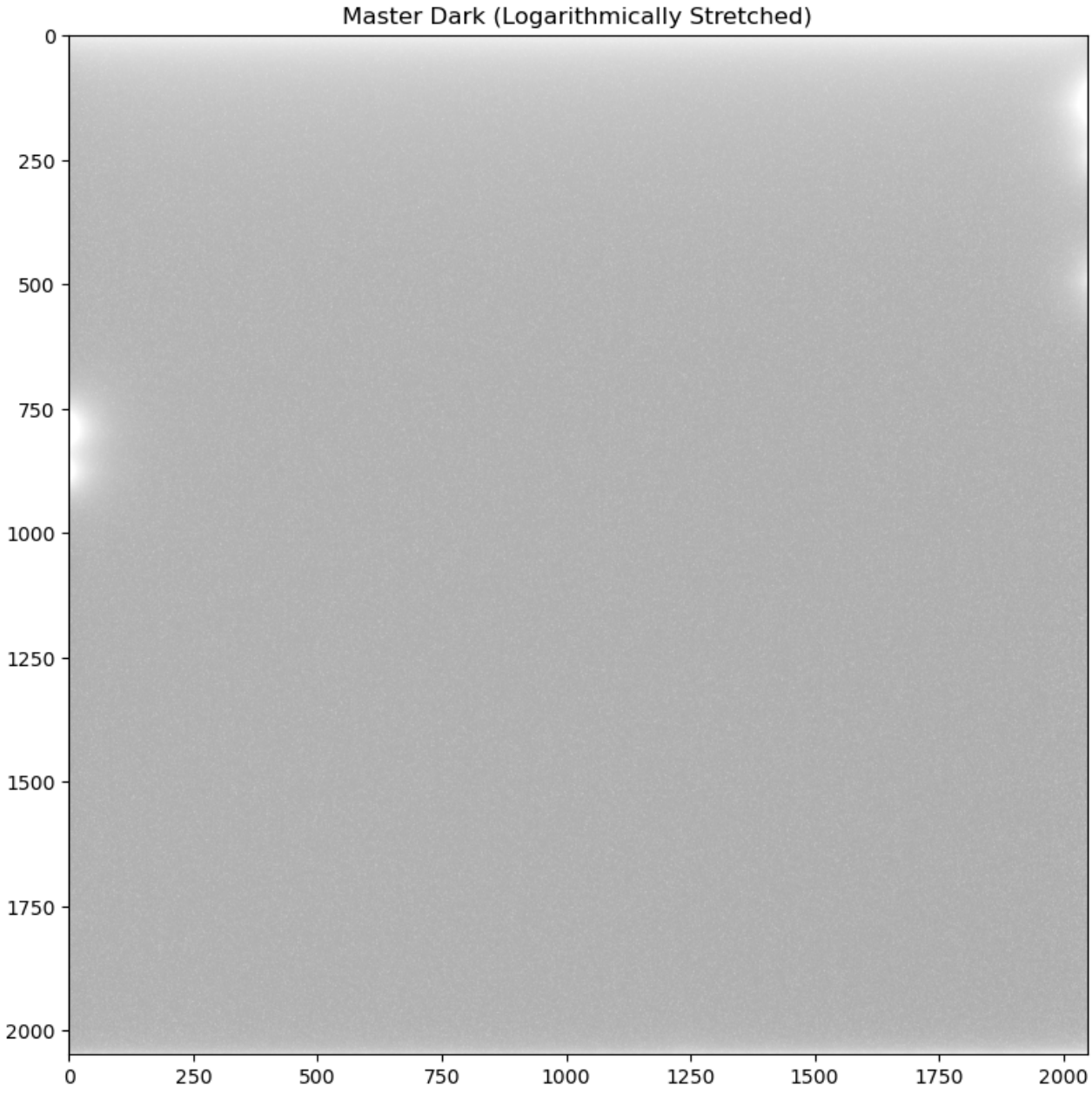
In [3]:
```
darks[5].data
```

Out[3]:
```
array([[ 6400, 61638,     0, ..., 14966, 14966, 14288],
       [ 5741,  5741, 17478, ..., 15172, 16338, 14002],
       [17457, 18224, 17478, ..., 14268, 16087, 16234],
       ...,
       [ 7993,  9072,  8196, ..., 15131, 15921, 15691],
       [ 8791,  9148,  9507, ..., 18760, 19255, 17520],
       [11029, 18374, 11557, ..., 28432, 24336, 25038]], dtype=uint16)
```

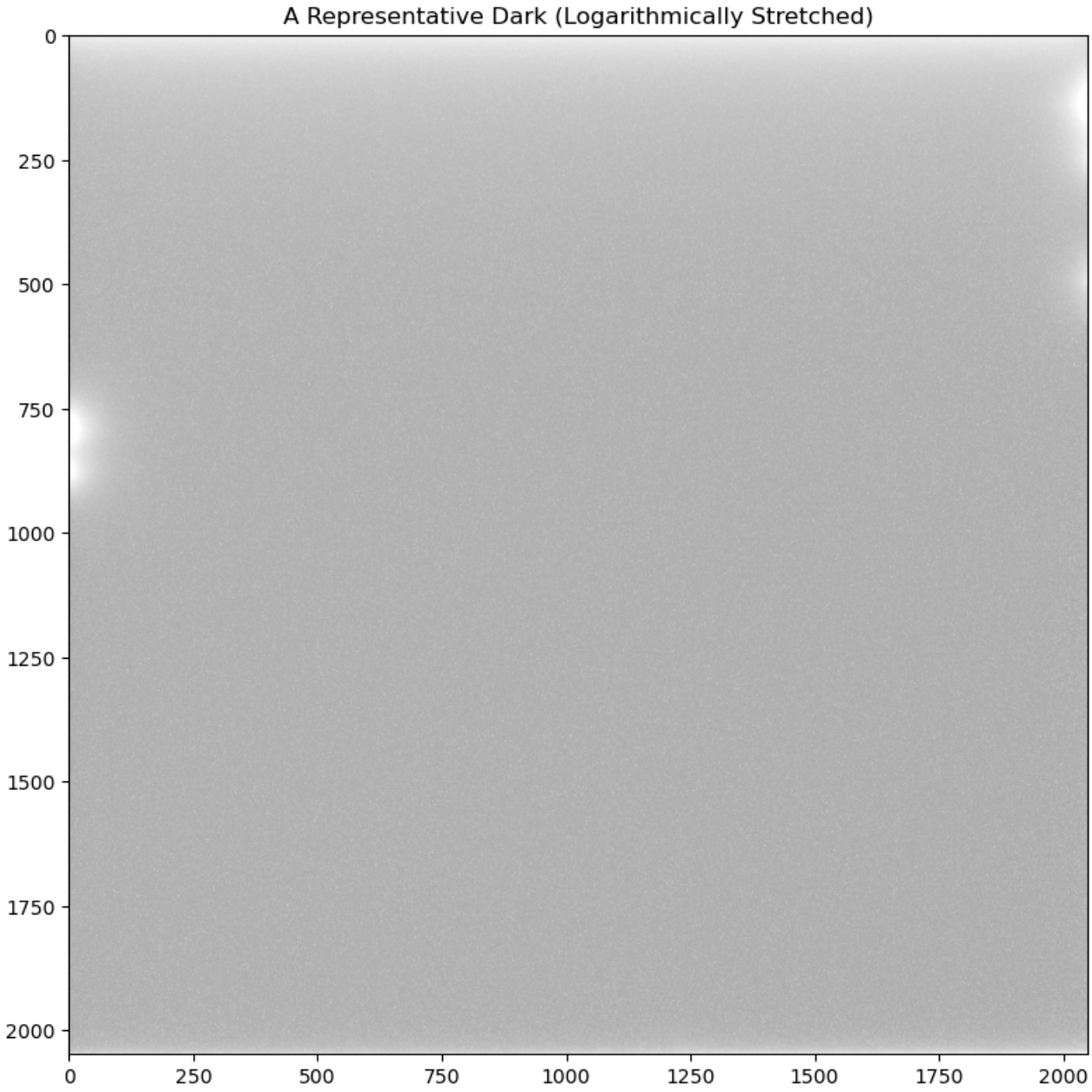
Display the Master Dark and a Representative Dark

We will display the range from 5000 to 30000, which of course cannot be accomplished with 256 gray scale values, so we will also do logarthmic stretching of that range.

In [4]:
```
# Log stretch

stretch_function = log_stretch_transform(5000, 25000)
stretch_transform = np.vectorize(stretch_function)

stretched_master_dark = stretch_transform(master_dark.data)

stretched_darks = [
    stretch_transform(dark.data) for dark in darks
]

# Display the master dark

fig, axes = plt.subplots(1, 1, figsize=(8, 8))

axes.imshow(stretched_master_dark.data, cmap='gray')
axes.set_title("Master Dark (Logarithmically Stretched)")

plt.tight_layout()
plt.show()
```



In [5]:
```
# Display a representative dark

fig, axes = plt.subplots(1, 1, figsize=(8, 8))

axes.imshow(stretched_darks[5].data, cmap='gray')
axes.set_title("A Representative Dark (Logarithmically Stretched)")

plt.tight_layout()
plt.show()
```



Subtract Master Dark from Darks

In [6]:
```
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 -700 to 1300 ADU.

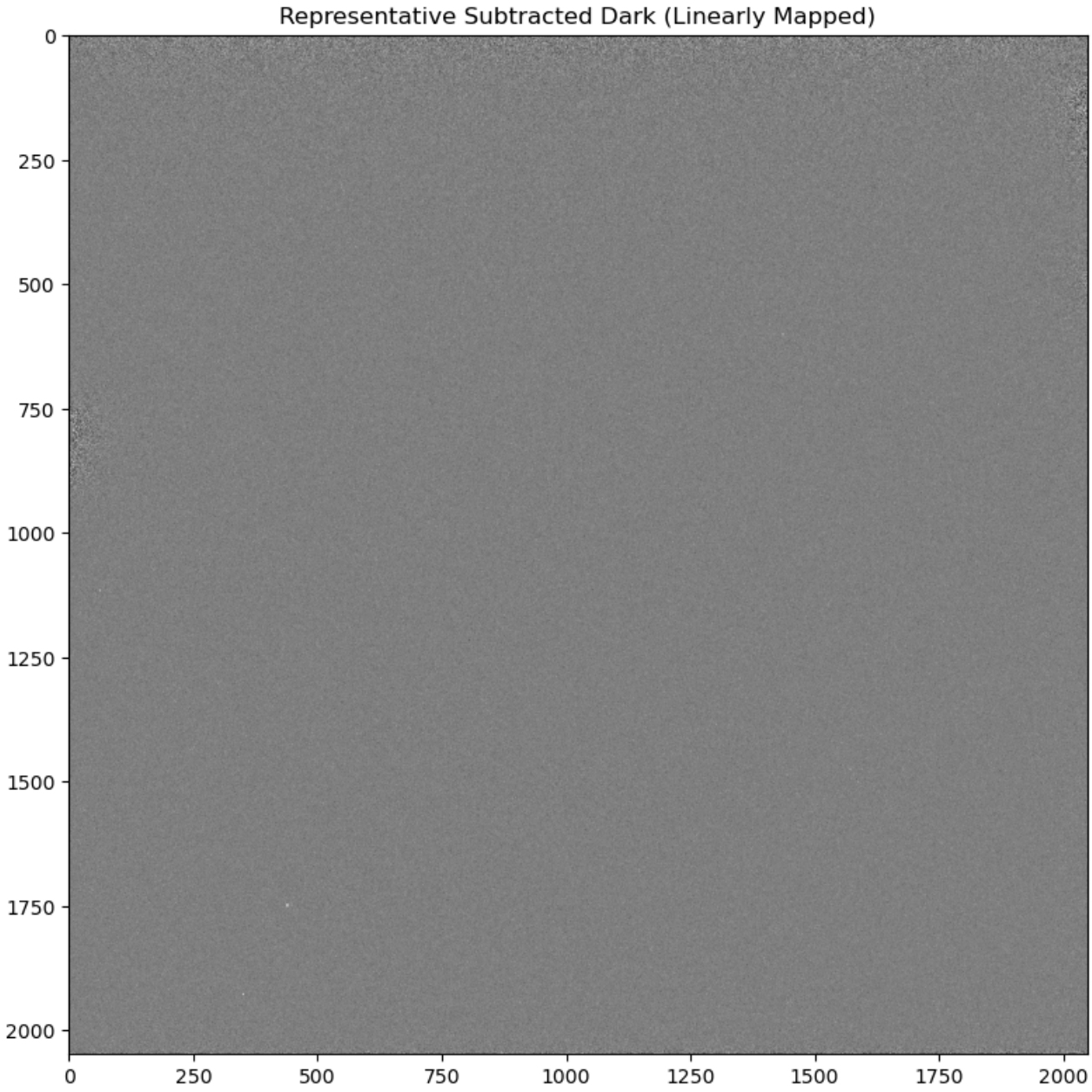
In [7]:
```
# np.set_printoptions(threshold=sys.maxsize) # Uncommenting this line will c

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

Out[7]:
```
array([[   0. ,  128. ,   -3.5, ...,   21. ,  -10.5, -102. ],
       [   0. ,    0. , -127.5, ...,  730.5, -379. , -194. ],
       [ 244. ,  331. , -447. , ..., -142. , 1110.5,  -94. ],
       ...,
       [ -18. ,  168.5, -148. , ..., -373.5,  769. ,  -94. ],
       [ -37.5,   19. ,   18.5, ...,  215. ,   64.5, -404.5],
       [ 563.5,  -10.5,  235.5, ...,  602. , -837. ,  419.5]])
```

Display a Representative Subtracted Dark

We will display the range from -1000 to +1000, which of course cannot be accomplished with 256 gray scale values, so we will also do a linear mapping of that range.

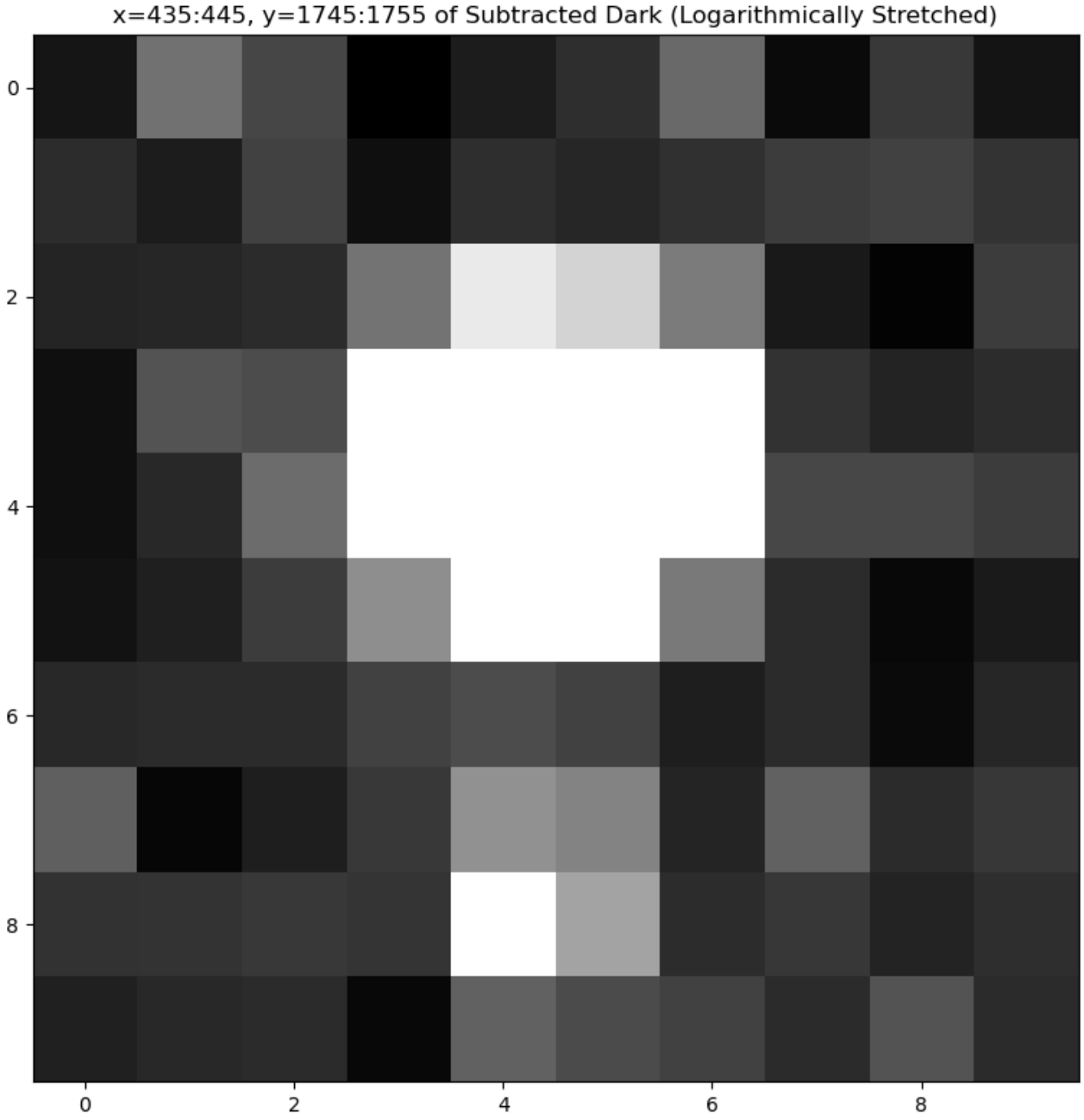
In [8]:
```
def linear_transform(black_point, saturation_range):

    def fn(pixel_value):
        pixel_value -= black_point
        if pixel_value <= 0.0:
            return 0
        else:
            if pixel_value >= saturation_range:
                return 255;
            else:
                return floor(256 * pixel_value / saturation_range)

    return fn


stretch_function2 = linear_transform(-1000, 2000)
stretch_transform2 = np.vectorize(stretch_function2)

stretched_representative_dark_data = stretch_transform2(representative_dark_

# Display the representative subtracted dark

fig, axes = plt.subplots(1, 1, figsize=(8, 8))

axes.imshow(stretched_representative_dark_data, cmap='gray')
axes.set_title("Representative Subtracted Dark (Linearly Mapped)")

plt.tight_layout()
plt.show()
```



Display Apparent Cosmic Ray in Subtracted Dark

Home in on the crud in the lower left of the above image.

In [9]:
```
# Home in on what appears to be a cosmic ray hit

stretched_subframe = stretched_representative_dark_data[1745:1755, 435:445]

# Home in on the crud in the lower left of the above image.

# subframe = stretched_representative_dark_data[1740:1939, 434:444]

# Display the representative subtracted dark

fig, axes = plt.subplots(1, 1, figsize=(8, 8))

axes.imshow(stretched_subframe, cmap='gray')
axes.set_title("x=435:445, y=1745:1755 of Subtracted Dark (Logarithmically S

plt.tight_layout()
plt.show()
```



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 [10]:
```
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=200):
    # 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
    # all of the hot pixels are candidate leaders
    candidate_leaders = np.transpose([exceedance_indices[1], exceedance_indi
    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

Due to the large number of leaders identified, we will just display the leaders in the 100x100 region, `x=400:500, y=1700:1800`.

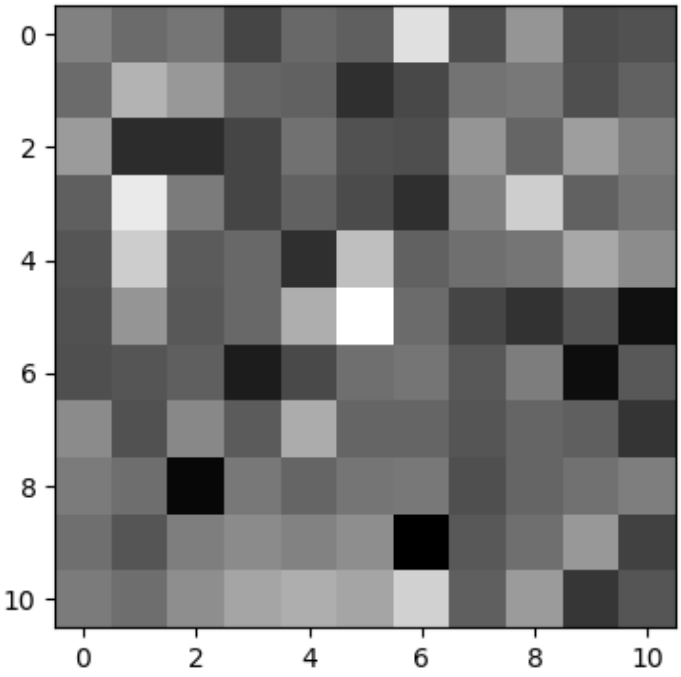
Then it classifies all pixels whose values exceed 200 as hot pixels. From among these, only the ones which are brighter than their eight nearest neighbors are declared to be "leaders" and the region around them is displayed.

Because a large number of pixels pass these cuts, only a subset of the leaders is displayed.

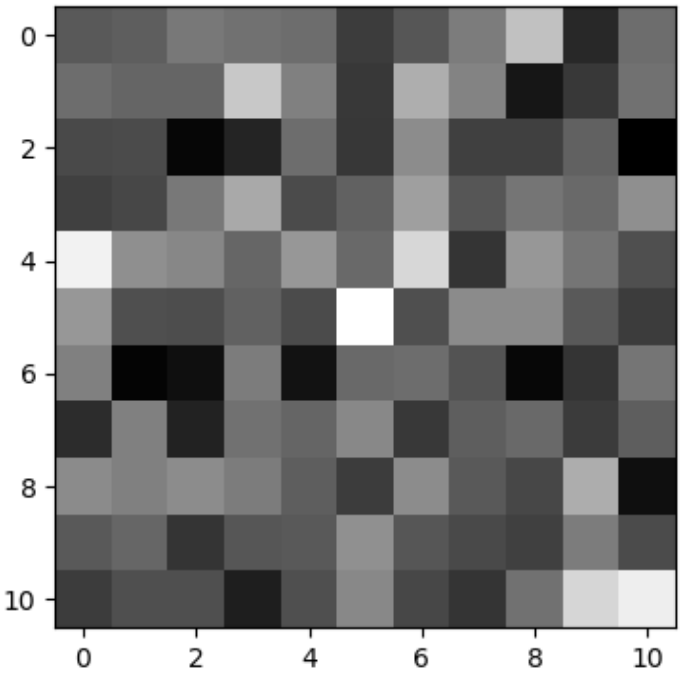
NB: In Python, a slice specification such as 400:500 includes 400 all they way to 499, but not 500.

In [ ]:
```
hot_pixel_leaders = find_hot_pixel_leaders(representative_dark_data, 200)

def display_leader(data, leader):
    fig, axes = plt.subplots(1, 1, figsize=(4, 4))

    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)

    title = "x={}:{}, y={}:{} around {}".format(lower_x, upper_x, lower_y, u

    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:
    if leader.x >= 400 and leader.x < 500 and leader.y >= 1700 and leader.y
        display_leader(representative_dark_data, leader)

hot_pixel_leaders
```

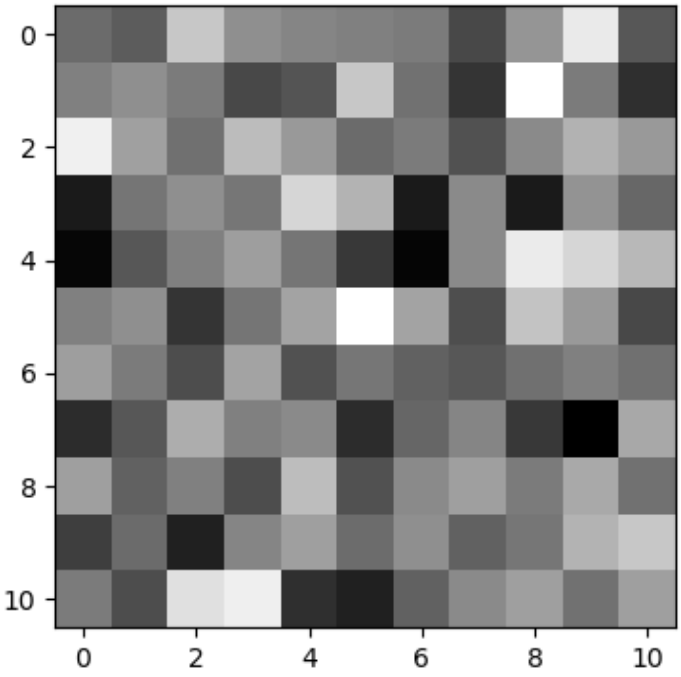
x=406:417, y=1695:1706 around HotPixel(x=411.0, y=1700.0, value=426.0)



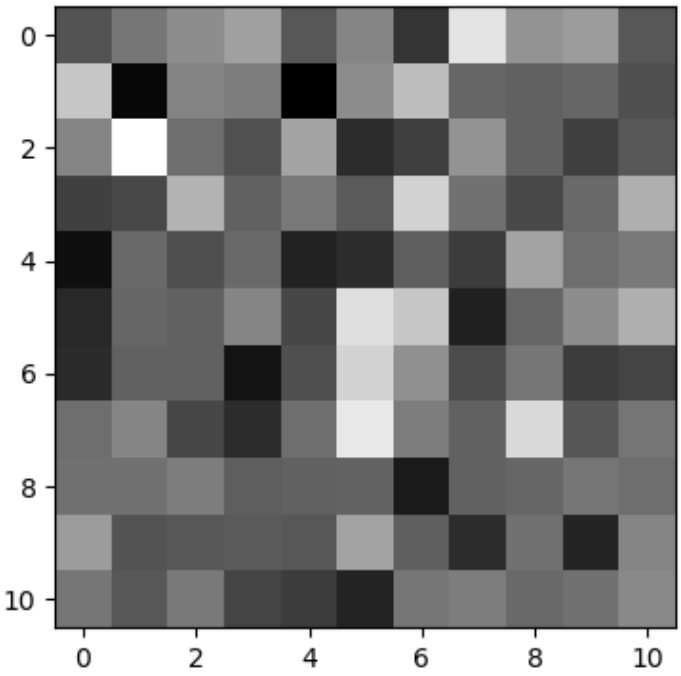
x=423:434, y=1695:1706 around HotPixel(x=428.0, y=1700.0, value=347.0)



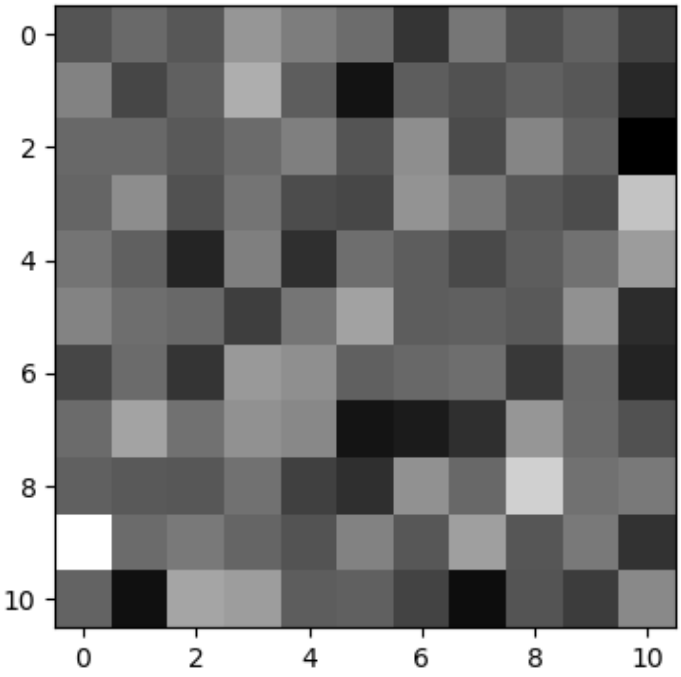
x=436:447, y=1695:1706 around HotPixel(x=441.0, y=1700.0, value=219.5)



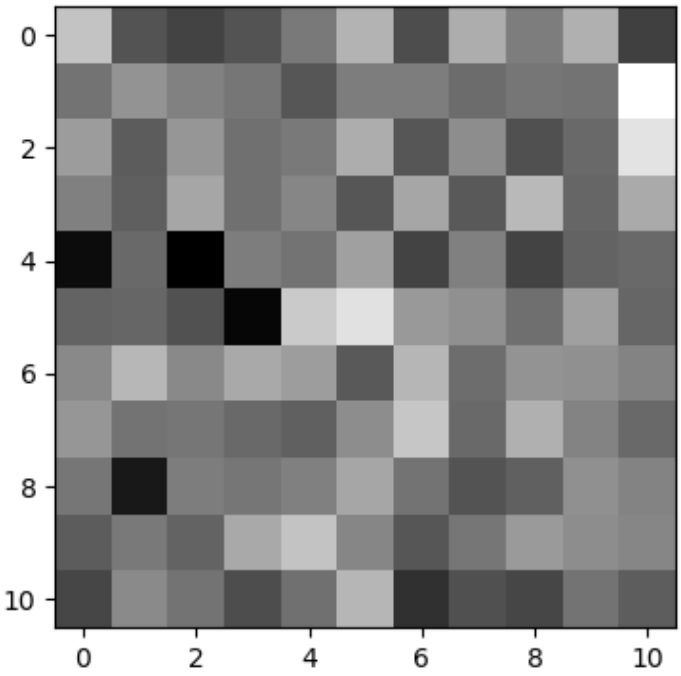
x=448:459, y=1695:1706 around HotPixel(x=453.0, y=1700.0, value=264.5)



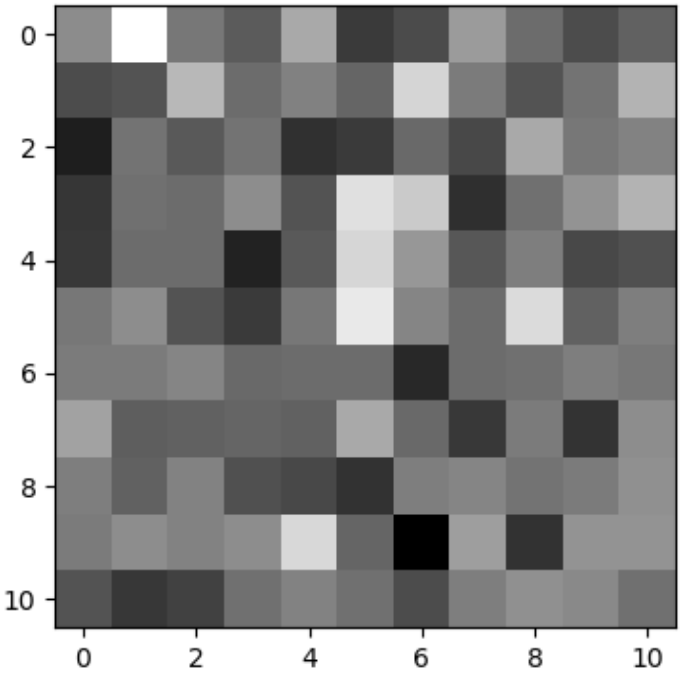
x=476:487, y=1696:1707 around HotPixel(x=481.0, y=1701.0, value=202.0)



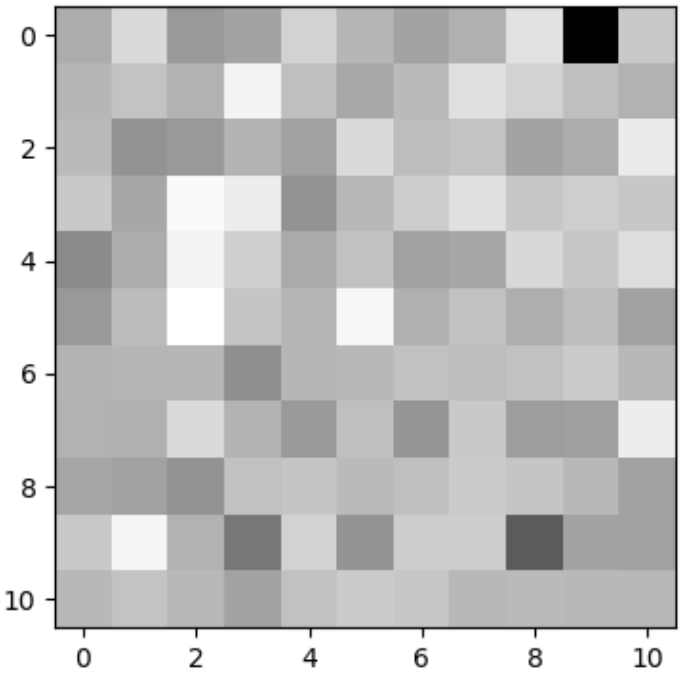
x=397:408, y=1697:1708 around HotPixel(x=402.0, y=1702.0, value=281.0)



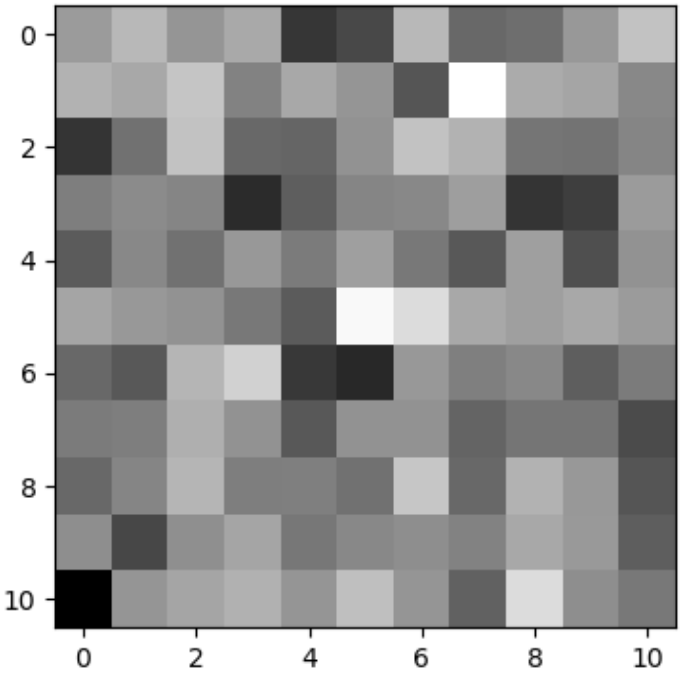
x=448:459, y=1697:1708 around HotPixel(x=453.0, y=1702.0, value=288.0)



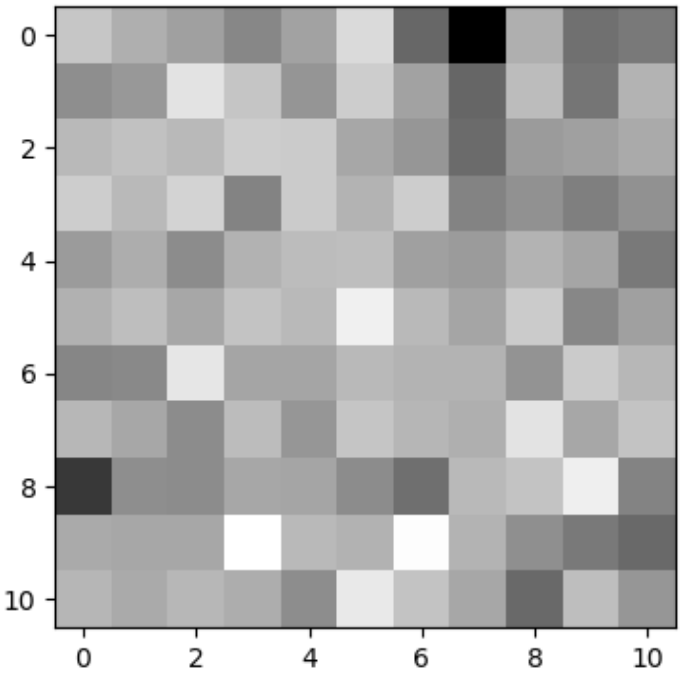
x=451:462, y=1697:1708 around HotPixel(x=456.0, y=1702.0, value=254.0)



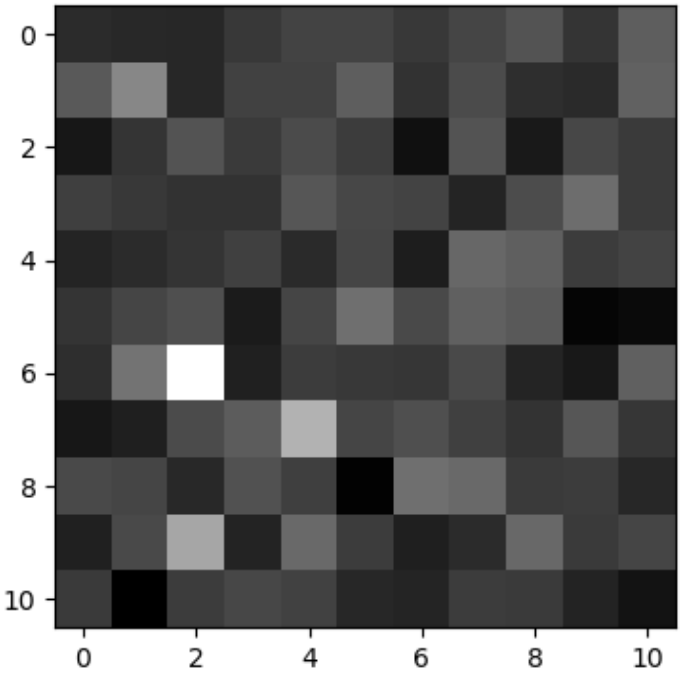
x=416:427, y=1698:1709 around HotPixel(x=421.0, y=1703.0, value=298.5)



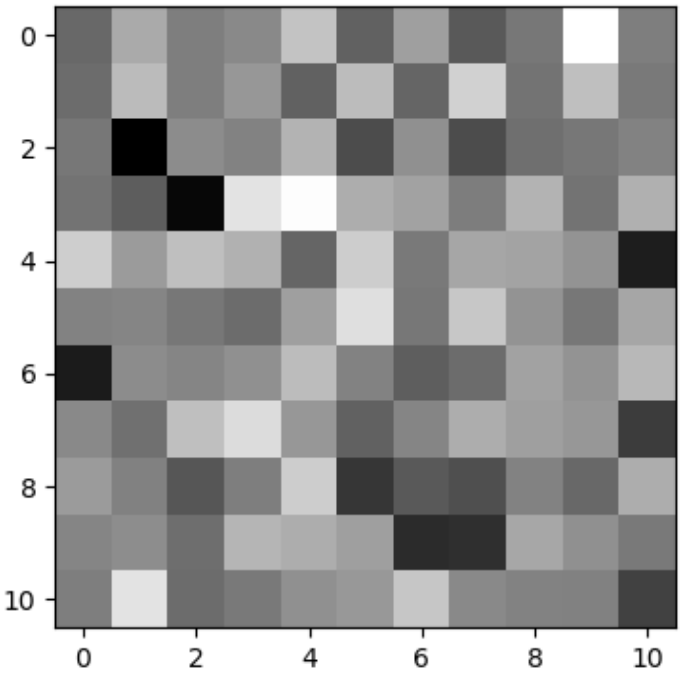
x=459:470, y=1698:1709 around HotPixel(x=464.0, y=1703.0, value=245.0)



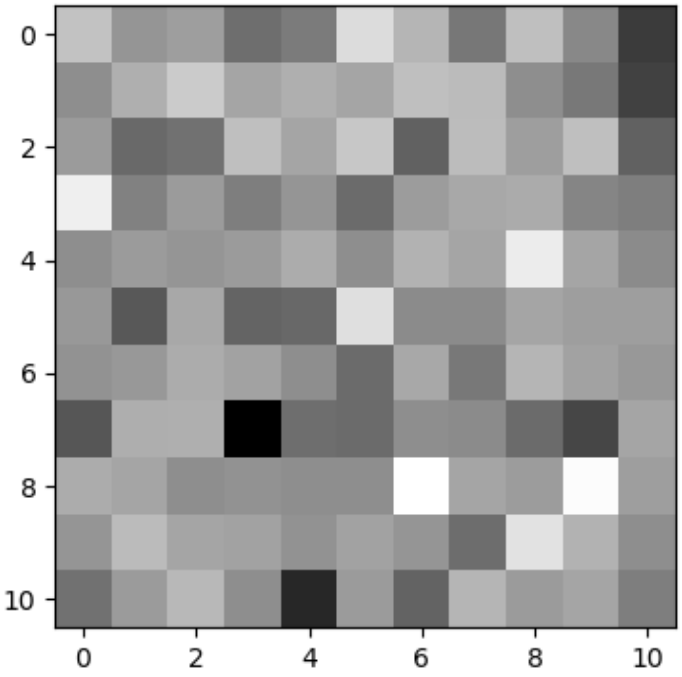
x=472:483, y=1698:1709 around HotPixel(x=477.0, y=1703.0, value=209.5)



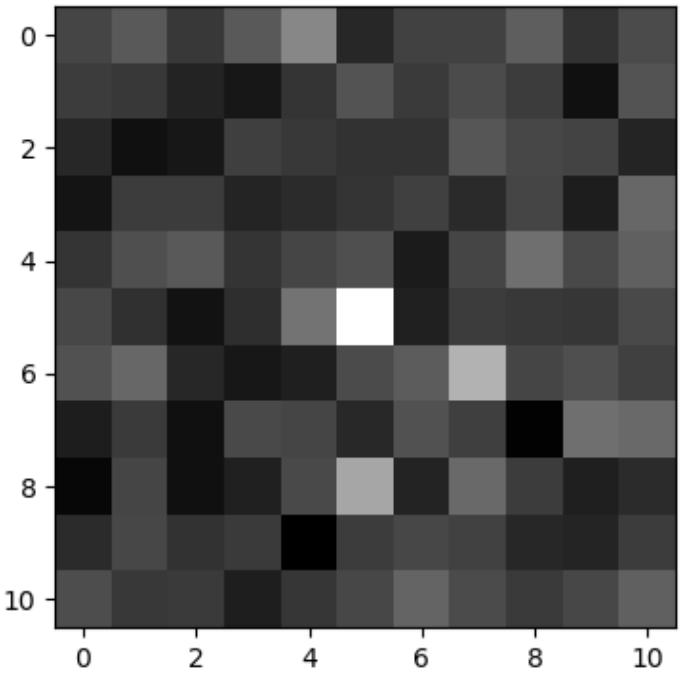
x=398:409, y=1699:1710 around HotPixel(x=403.0, y=1704.0, value=208.5)



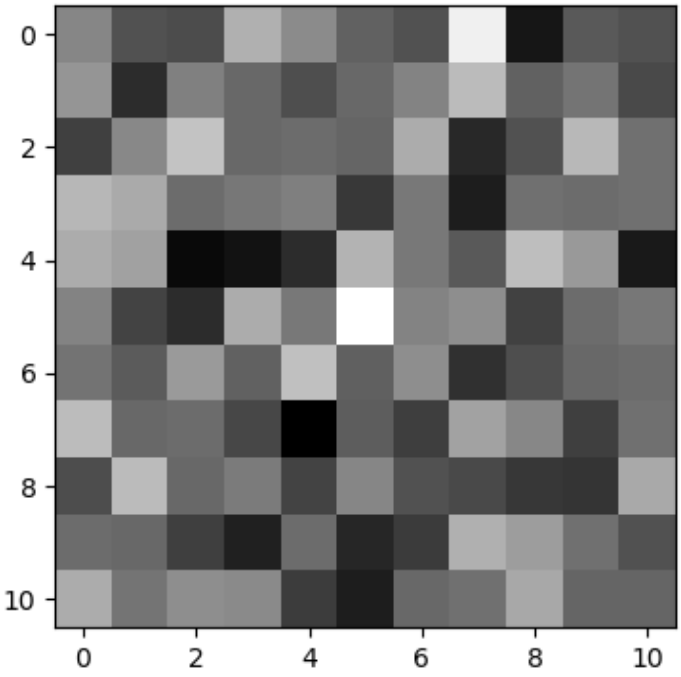
x=456:467, y=1699:1710 around HotPixel(x=461.0, y=1704.0, value=209.5)



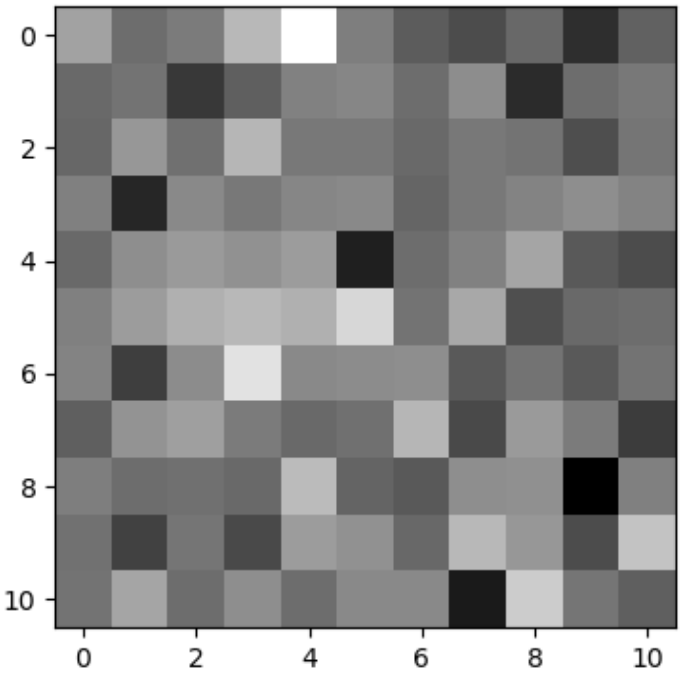
x=469:480, y=1699:1710 around HotPixel(x=474.0, y=1704.0, value=810.0)



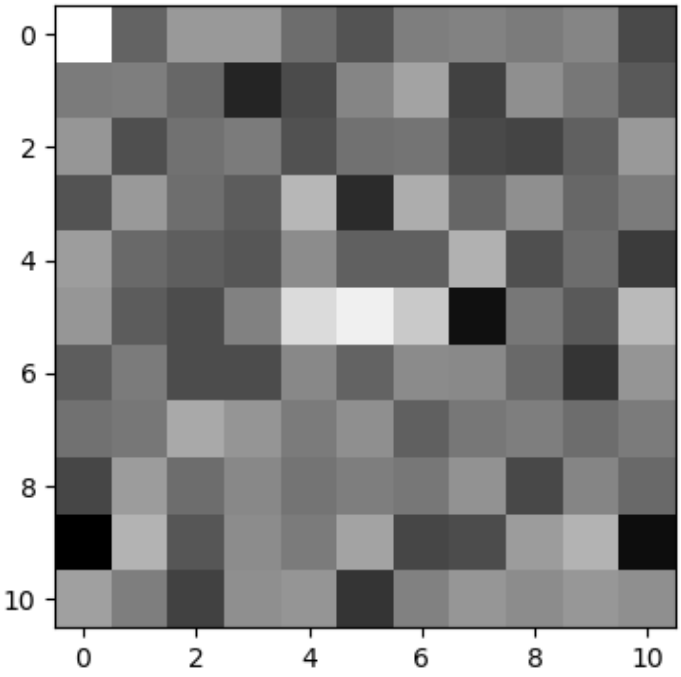
x=479:490, y=1699:1710 around HotPixel(x=484.0, y=1704.0, value=342.5)



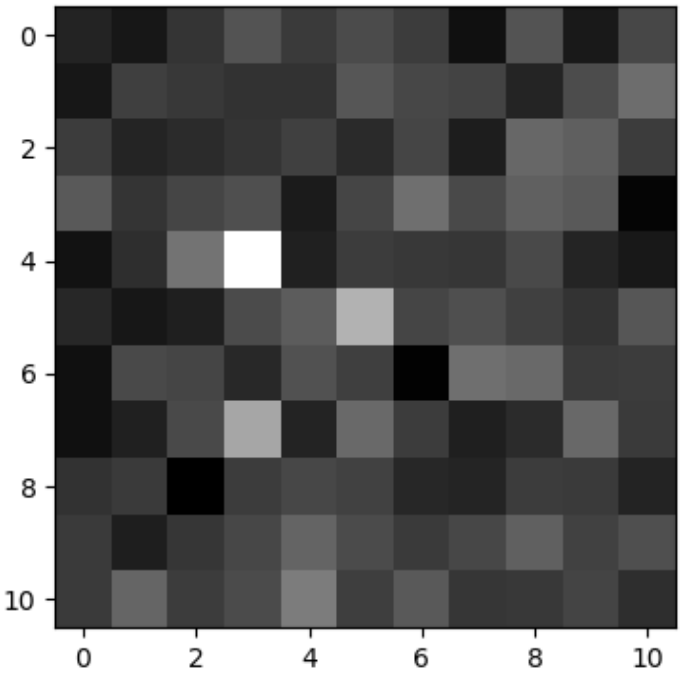
x=407:418, y=1700:1711 around HotPixel(x=412.0, y=1705.0, value=298.0)



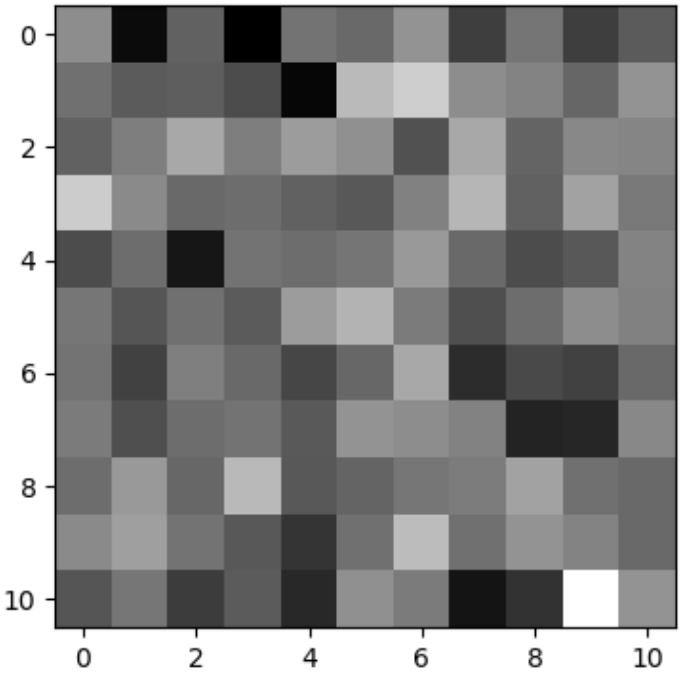
x=428:439, y=1700:1711 around HotPixel(x=433.0, y=1705.0, value=306.0)



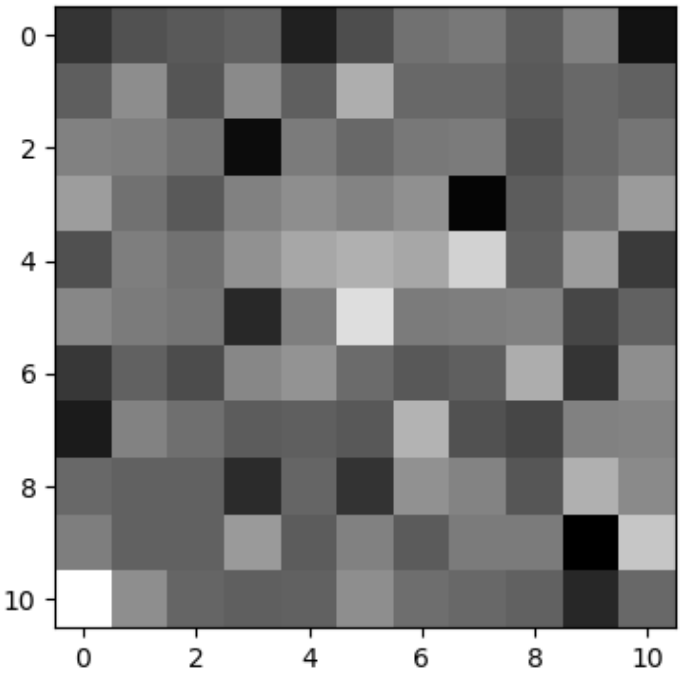
x=471:482, y=1700:1711 around HotPixel(x=476.0, y=1705.0, value=487.0)



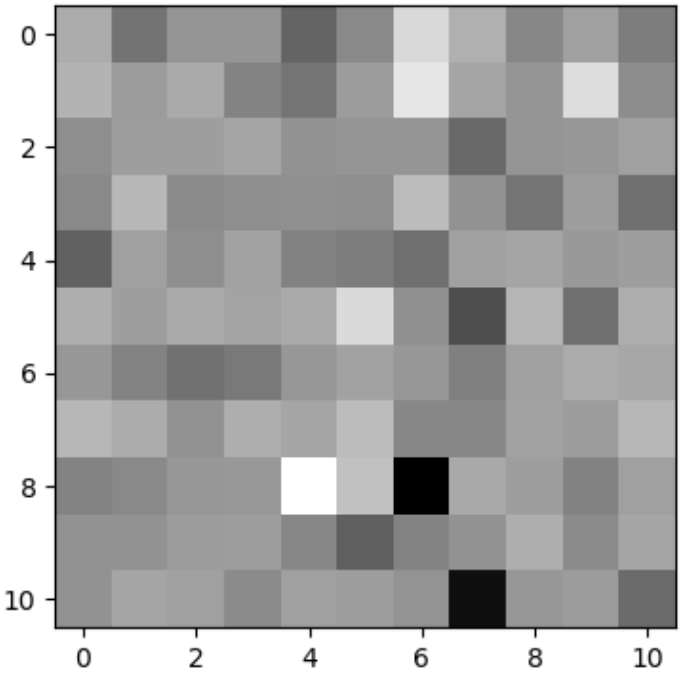
x=396:407, y=1701:1712 around HotPixel(x=401.0, y=1706.0, value=201.5)



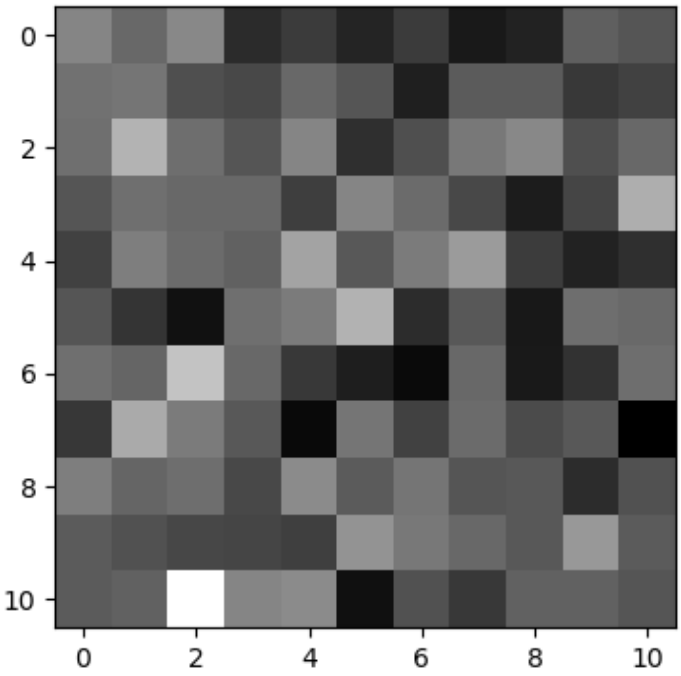
x=405:416, y=1701:1712 around HotPixel(x=410.0, y=1706.0, value=333.0)

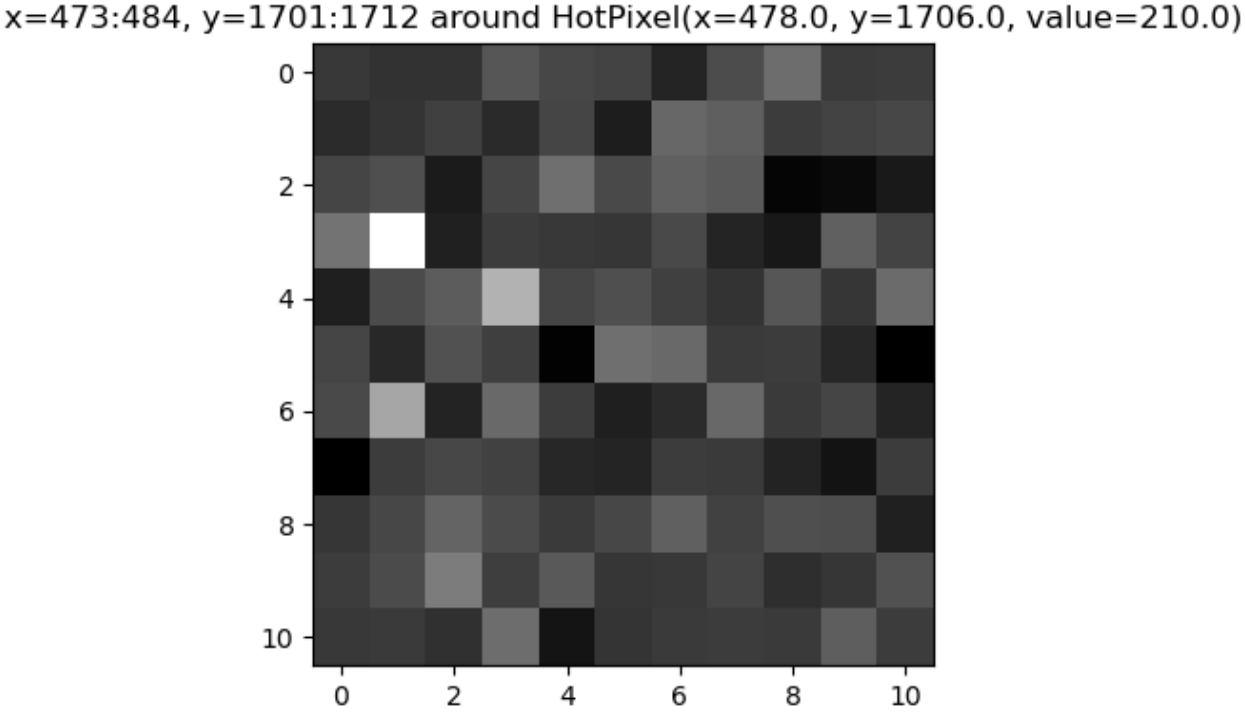


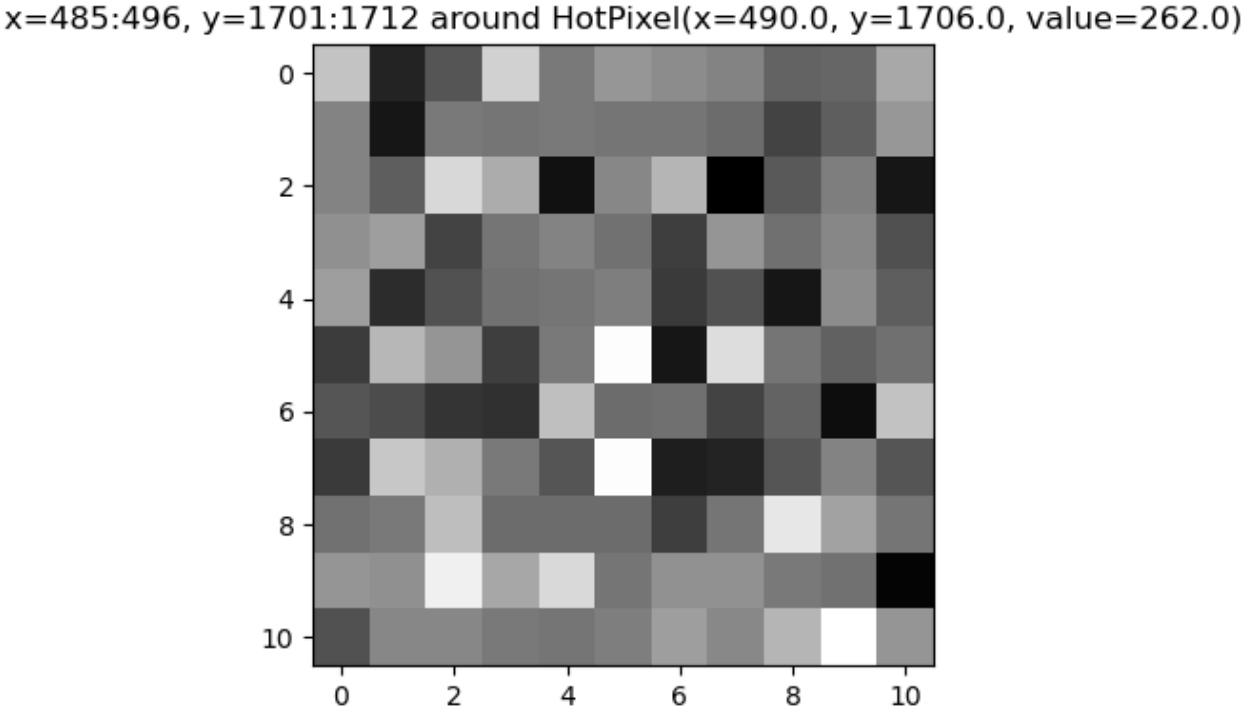
x=447:458, y=1701:1712 around HotPixel(x=452.0, y=1706.0, value=245.0)



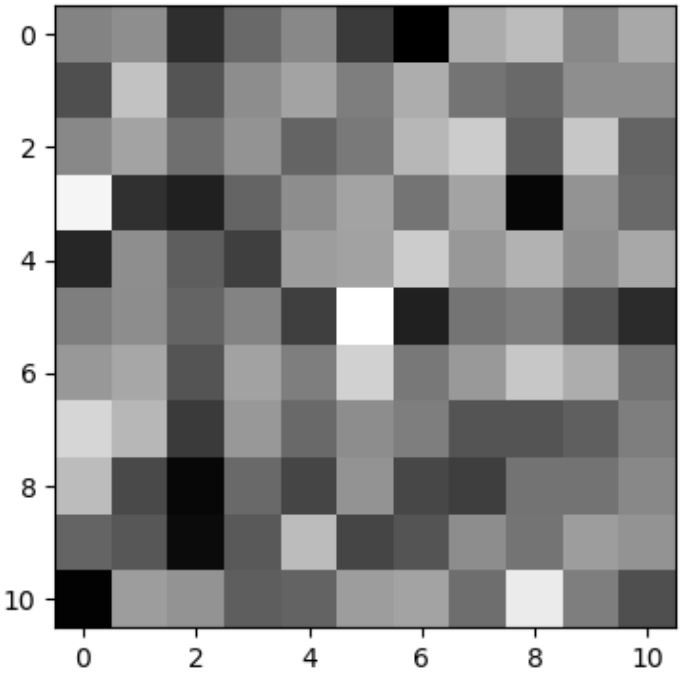
x=463:474, y=1701:1712 around HotPixel(x=468.0, y=1706.0, value=239.5)



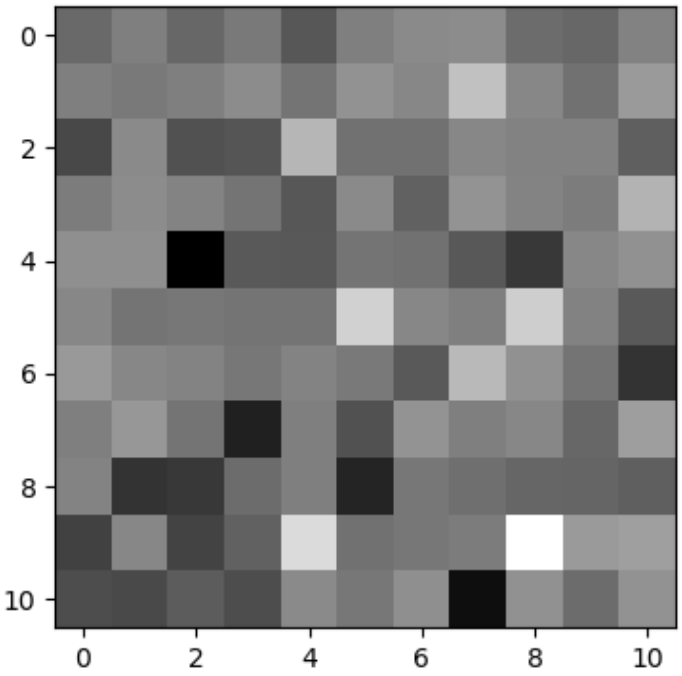

x=473:484, y=1701:1712 around HotPixel(x=478.0, y=1706.0, value=210.0)
0
2
4
6
8
10
0
2
4
6
8
10


x=485:496, y=1701:1712 around HotPixel(x=490.0, y=1706.0, value=262.0)
0
2
4
6
8
10
0
2
4
6
8
10

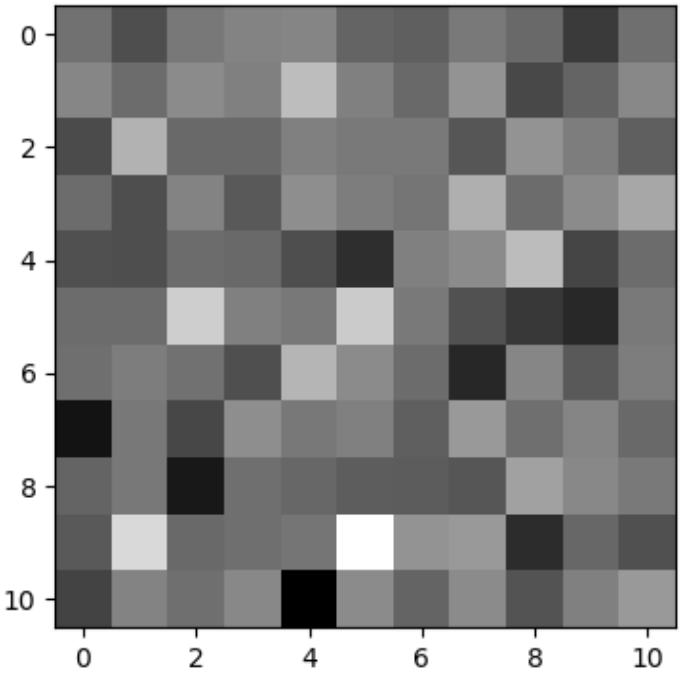
x=439:450, y=1702:1713 around HotPixel(x=444.0, y=1707.0, value=209.0)



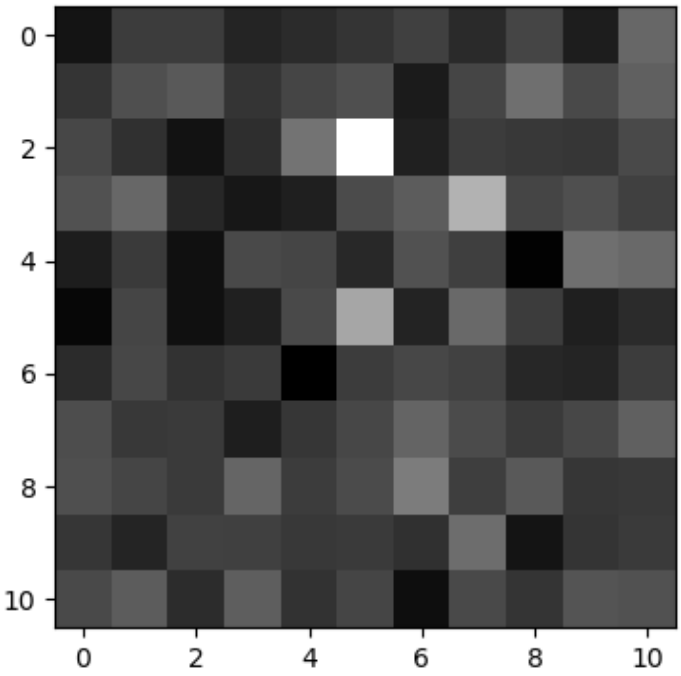
x=457:468, y=1702:1713 around HotPixel(x=462.0, y=1707.0, value=298.0)



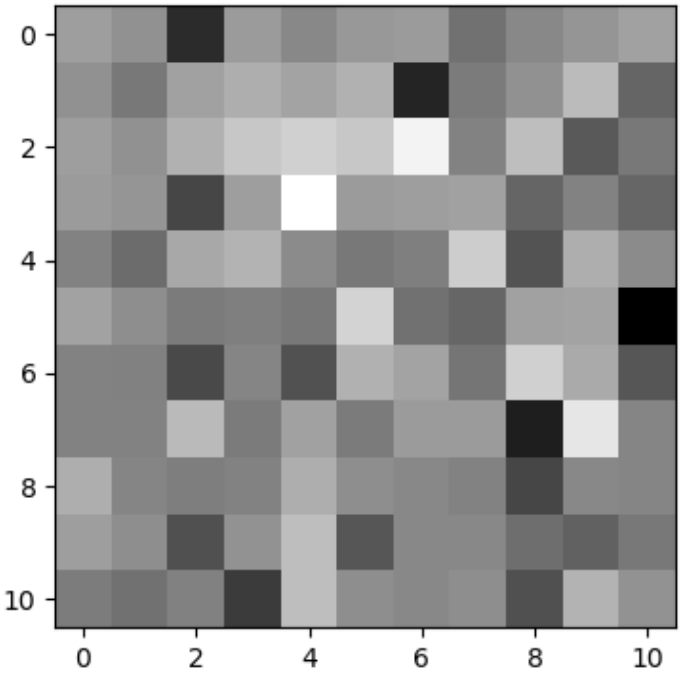
x=460:471, y=1702:1713 around HotPixel(x=465.0, y=1707.0, value=289.0)



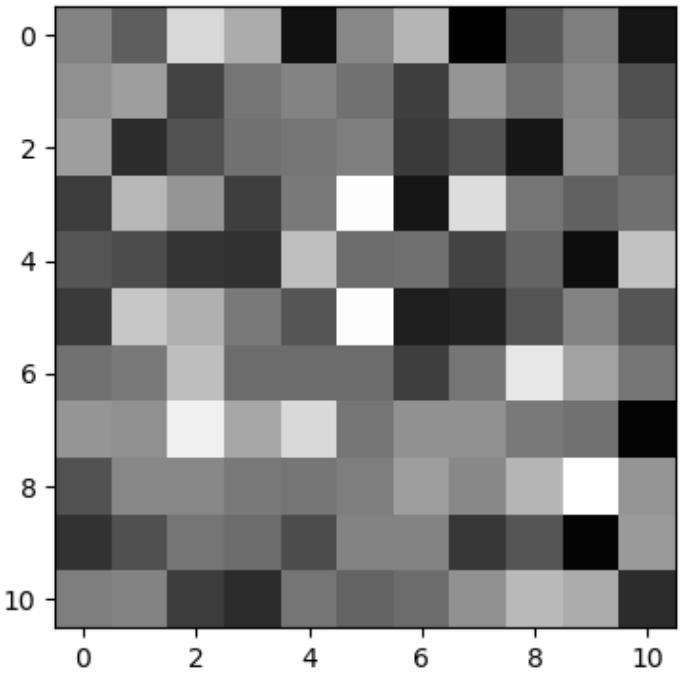
x=469:480, y=1702:1713 around HotPixel(x=474.0, y=1707.0, value=440.0)



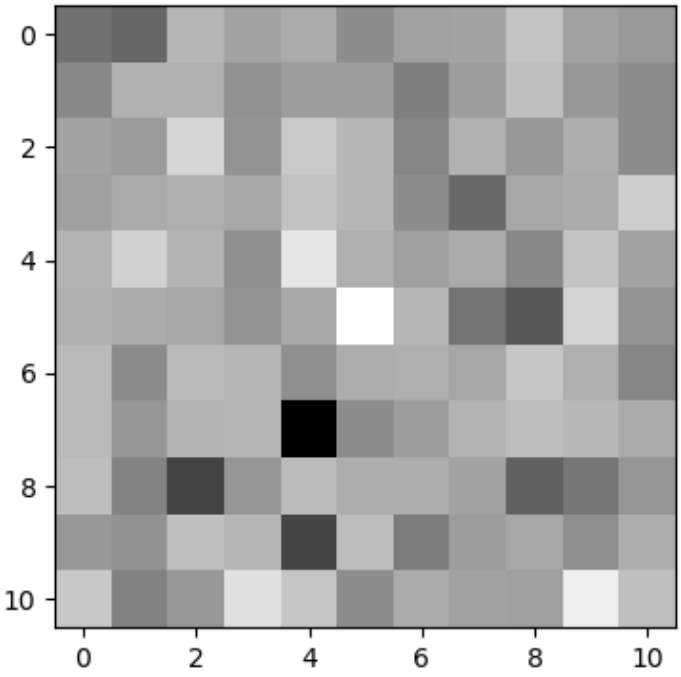
x=406:417, y=1703:1714 around HotPixel(x=411.0, y=1708.0, value=210.0)



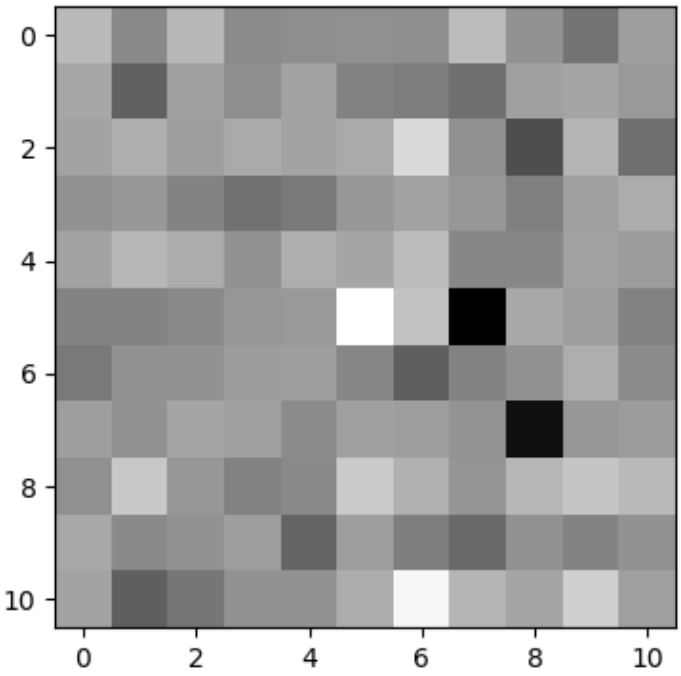
x=485:496, y=1703:1714 around HotPixel(x=490.0, y=1708.0, value=262.0)



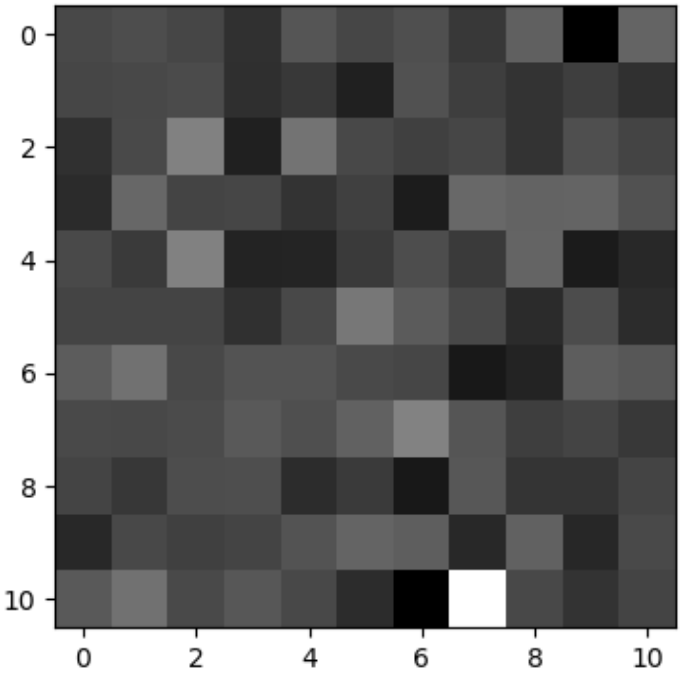
x=420:431, y=1704:1715 around HotPixel(x=425.0, y=1709.0, value=316.0)



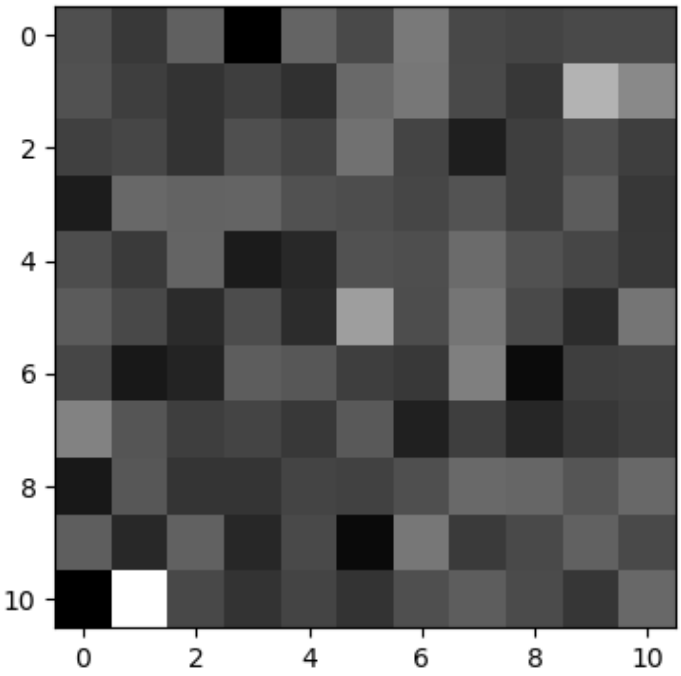
x=446:457, y=1704:1715 around HotPixel(x=451.0, y=1709.0, value=385.0)

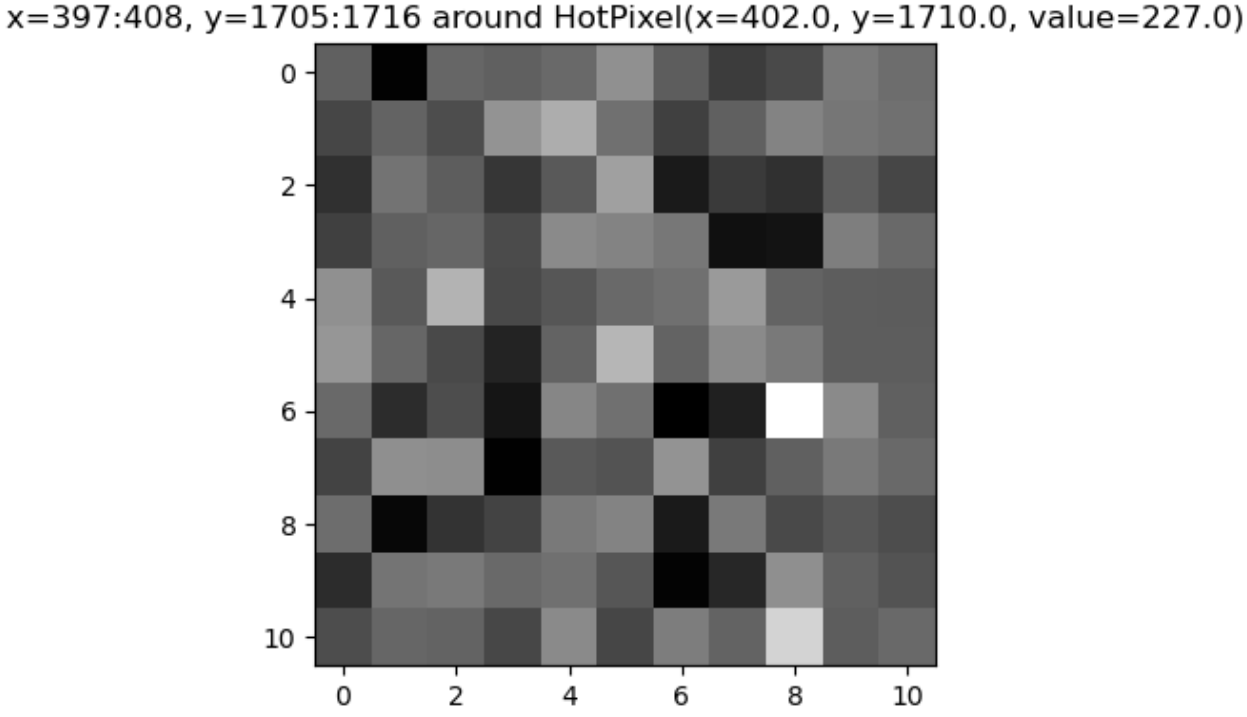


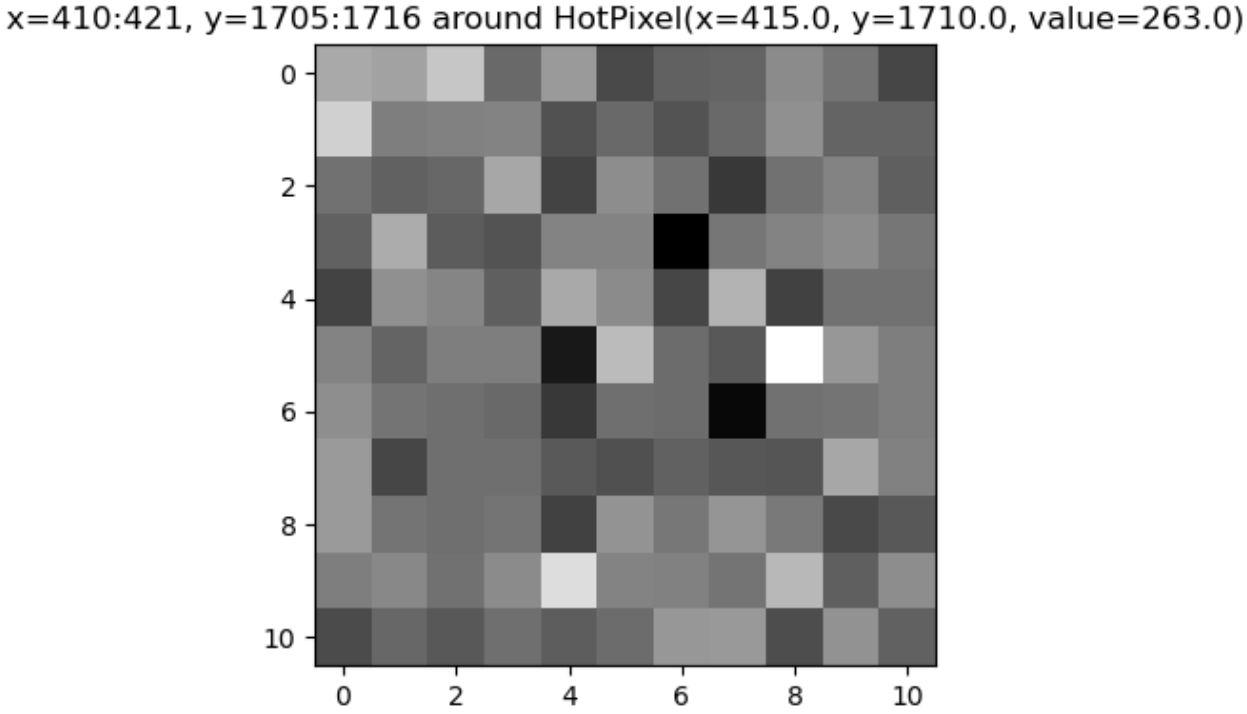
x=488:499, y=1704:1715 around HotPixel(x=493.0, y=1709.0, value=218.0)



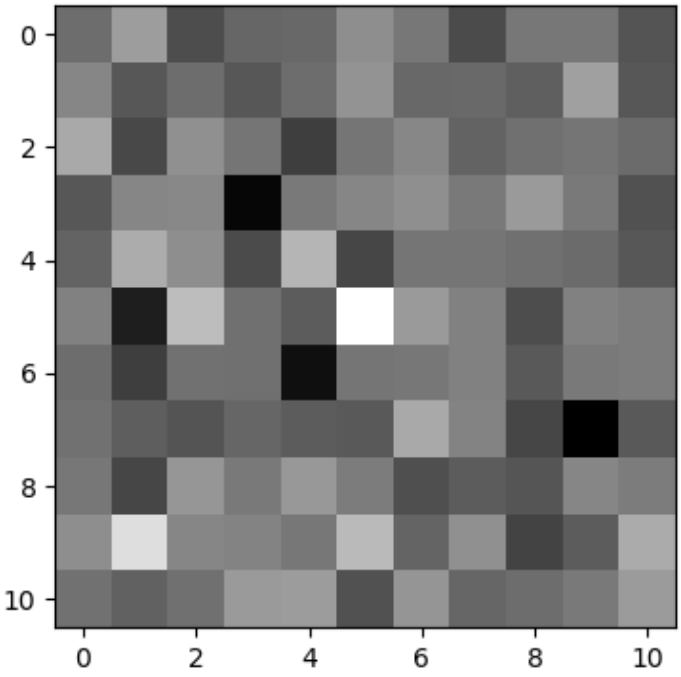
x=494:505, y=1704:1715 around HotPixel(x=499.0, y=1709.0, value=396.0)



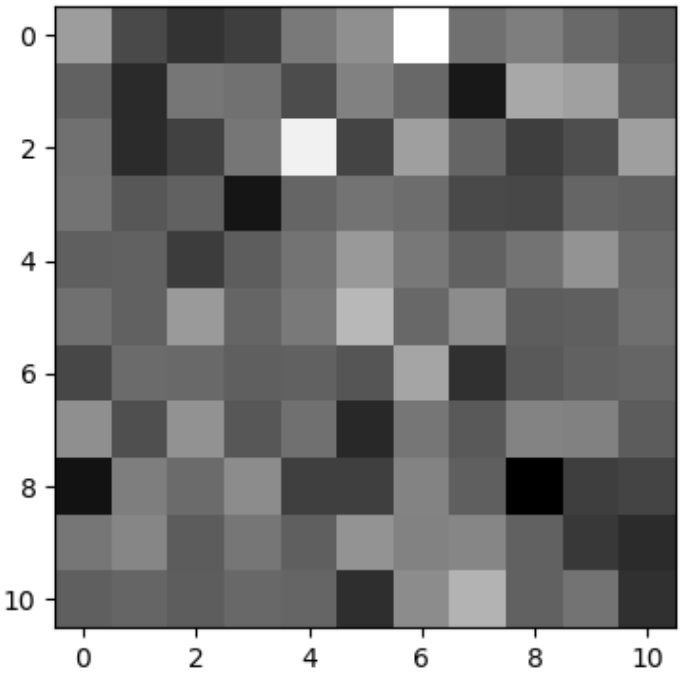

x=397:408, y=1705:1716 around HotPixel(x=402.0, y=1710.0, value=227.0)
0
2
4
6
8
10
0
2
4
6
8
10


x=410:421, y=1705:1716 around HotPixel(x=415.0, y=1710.0, value=263.0)
0
2
4
6
8
10
0
2
4
6
8
10

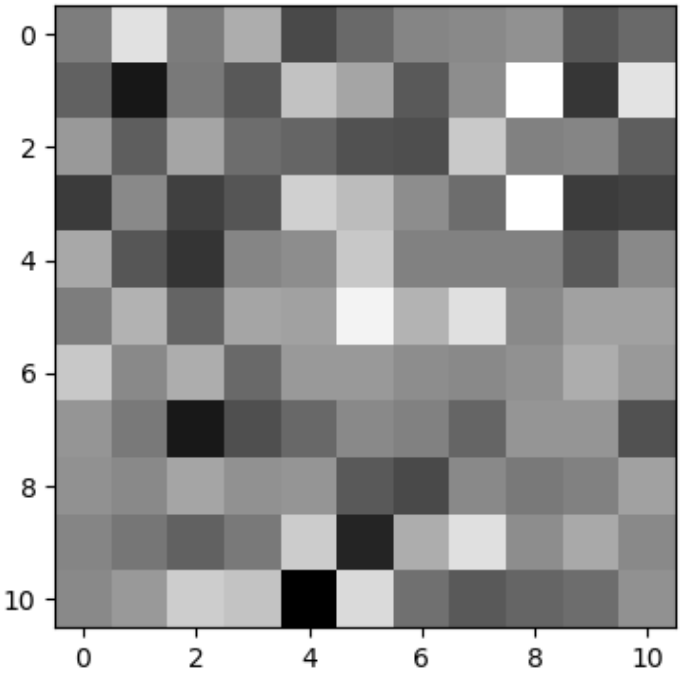
x=413:424, y=1705:1716 around HotPixel(x=418.0, y=1710.0, value=496.0)



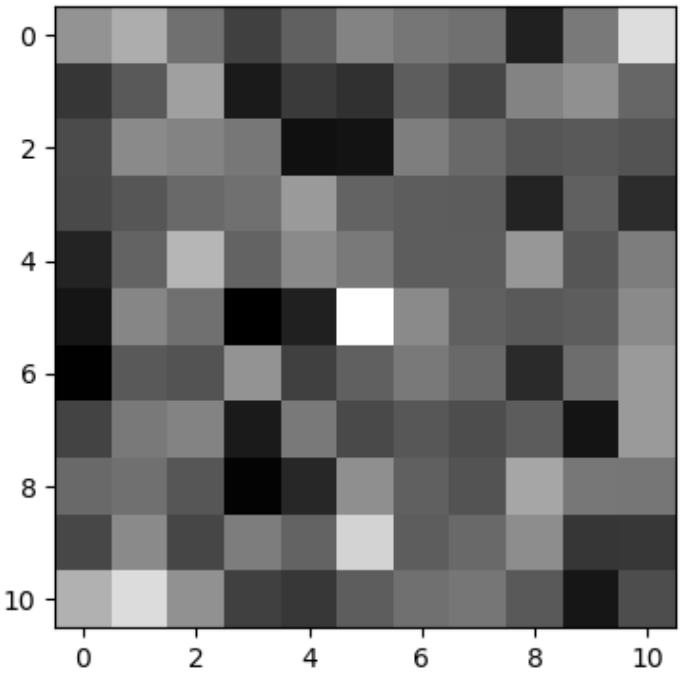
x=470:481, y=1705:1716 around HotPixel(x=475.0, y=1710.0, value=263.0)

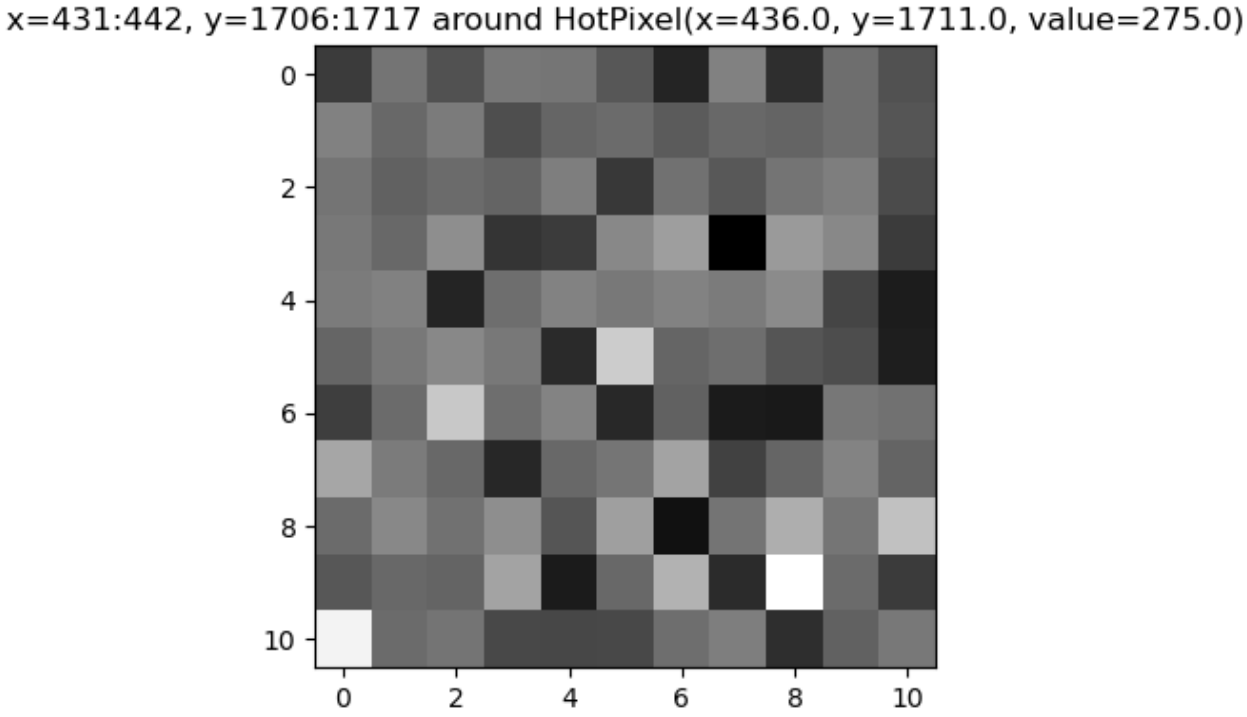


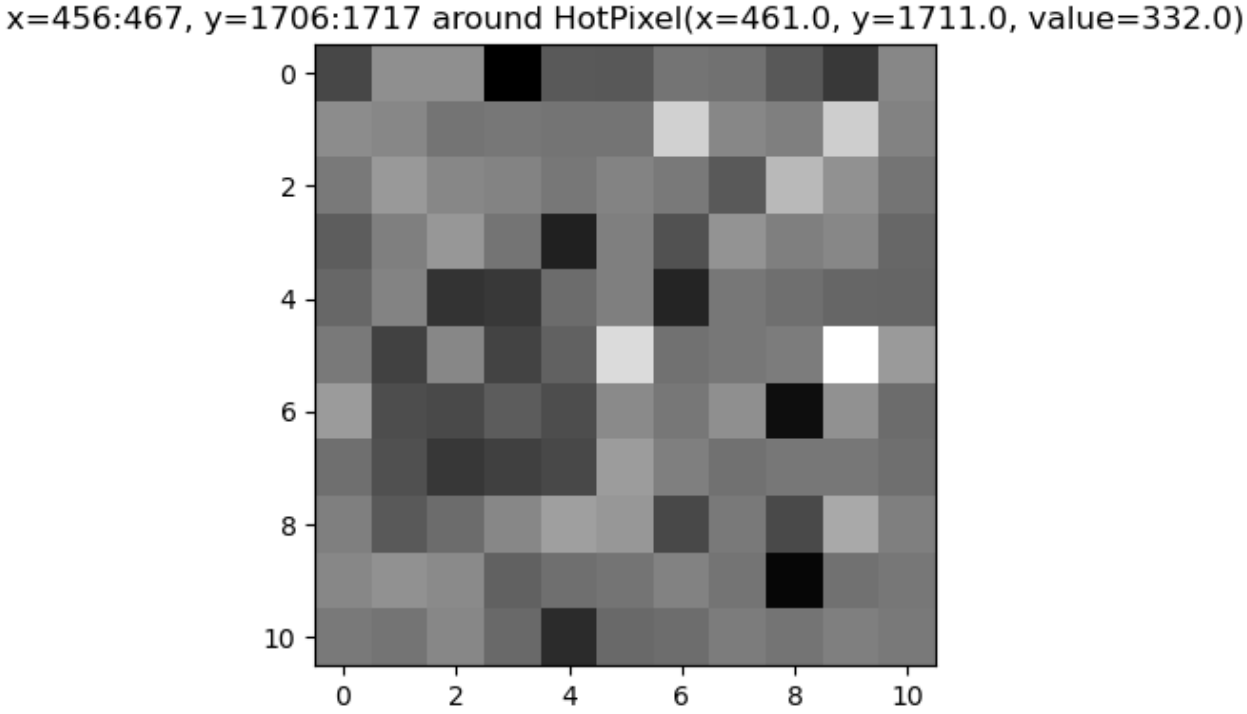
x=482:493, y=1705:1716 around HotPixel(x=487.0, y=1710.0, value=237.0)



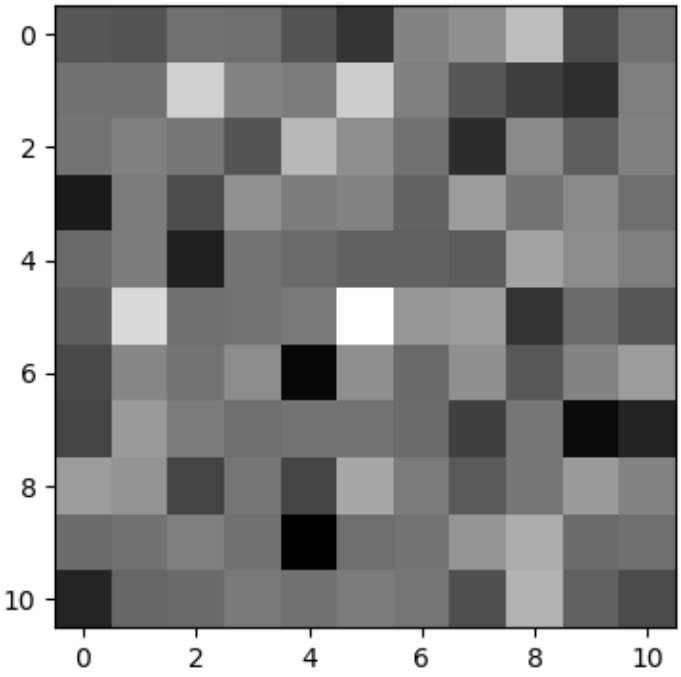
x=400:411, y=1706:1717 around HotPixel(x=405.0, y=1711.0, value=426.5)



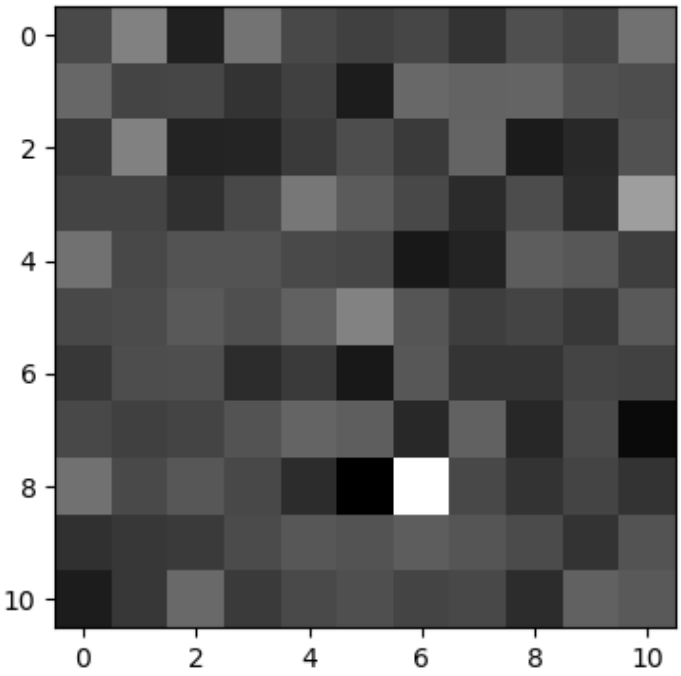

x=431:442, y=1706:1717 around HotPixel(x=436.0, y=1711.0, value=275.0)
0
2
4
6
8
10
0
2
4
6
8
10


x=456:467, y=1706:1717 around HotPixel(x=461.0, y=1711.0, value=332.0)
0
2
4
6
8
10
0
2
4
6
8
10

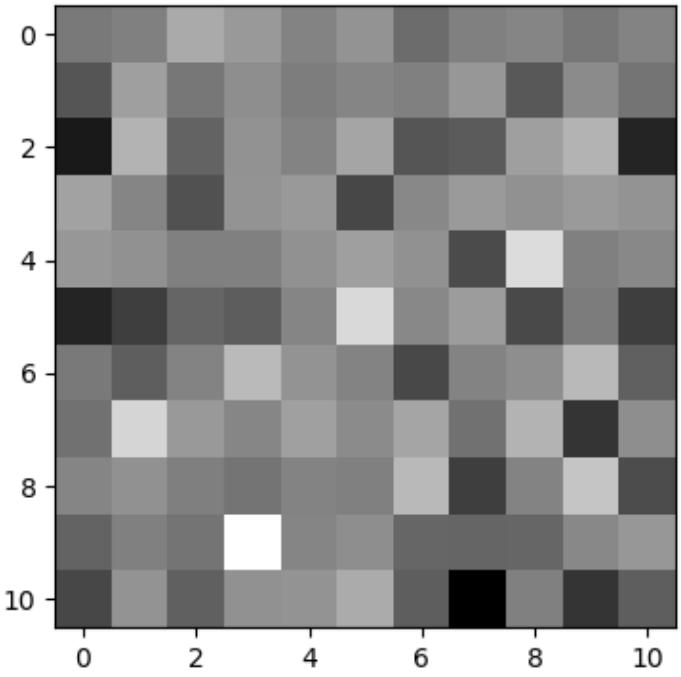
x=460:471, y=1706:1717 around HotPixel(x=465.0, y=1711.0, value=452.0)



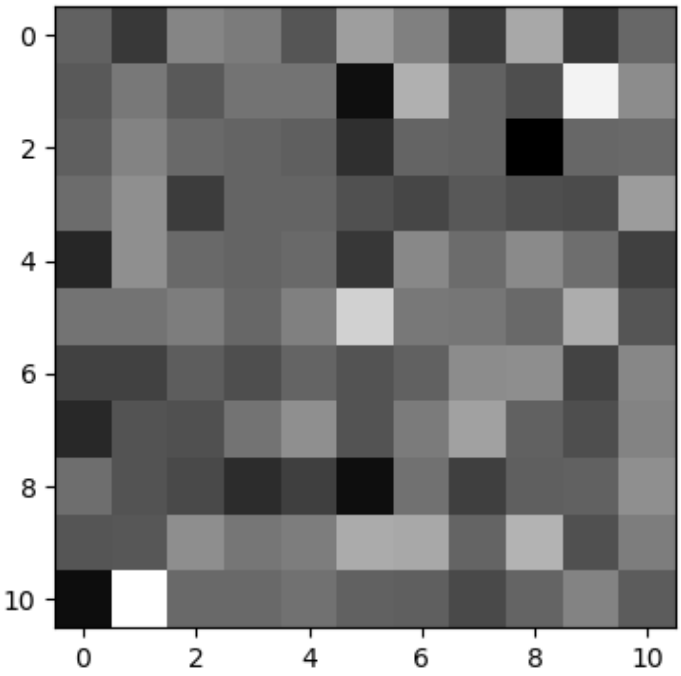
x=489:500, y=1706:1717 around HotPixel(x=494.0, y=1711.0, value=266.0)



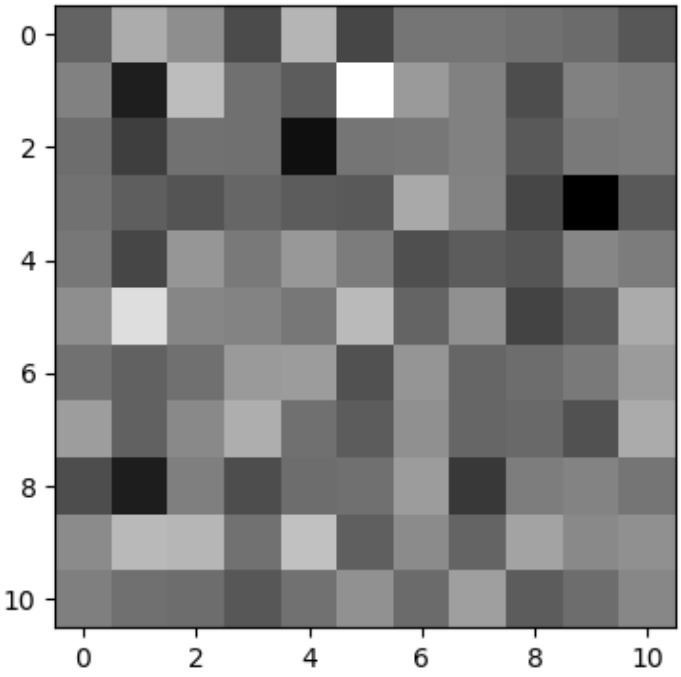
x=428:439, y=1707:1718 around HotPixel(x=433.0, y=1712.0, value=263.5)



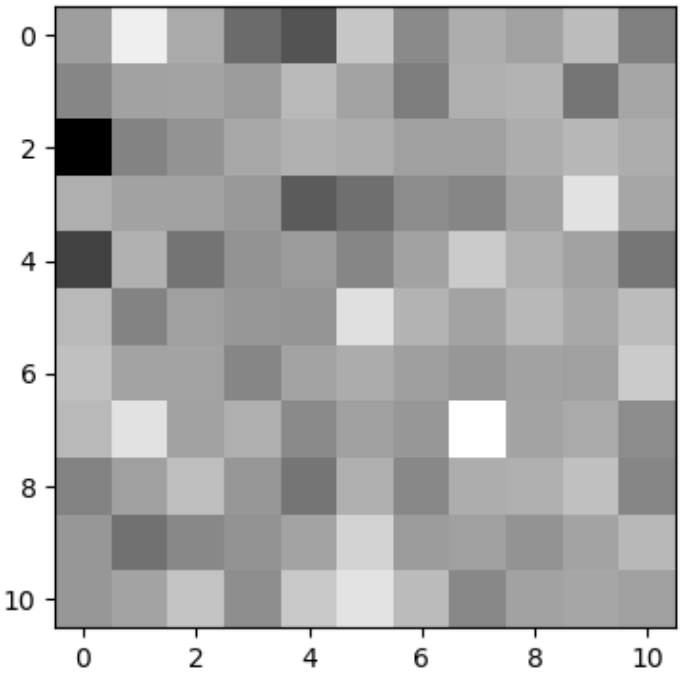
x=409:420, y=1709:1720 around HotPixel(x=414.0, y=1714.0, value=377.0)



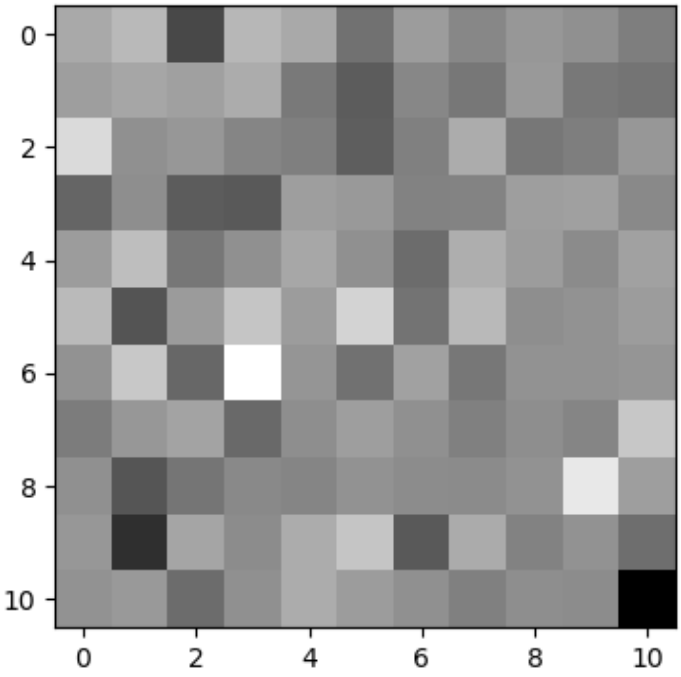
x=413:424, y=1709:1720 around HotPixel(x=418.0, y=1714.0, value=252.0)



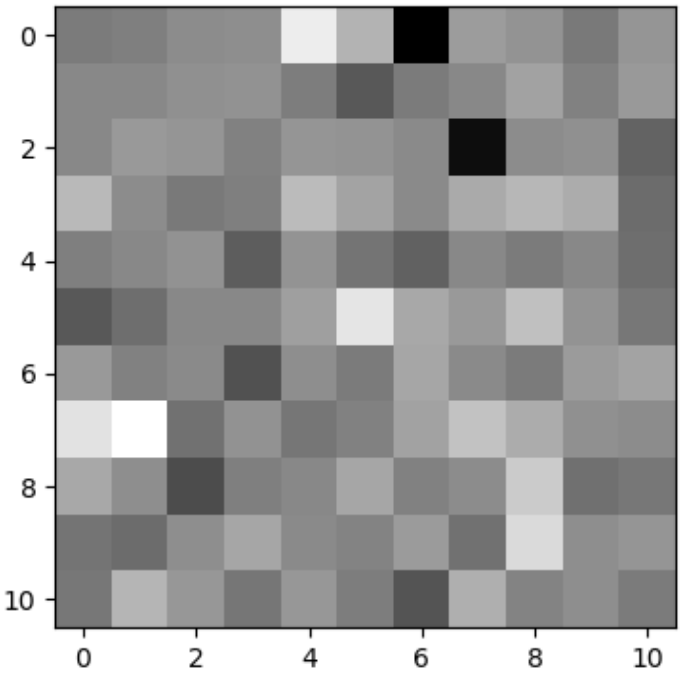
x=424:435, y=1709:1720 around HotPixel(x=429.0, y=1714.0, value=253.0)



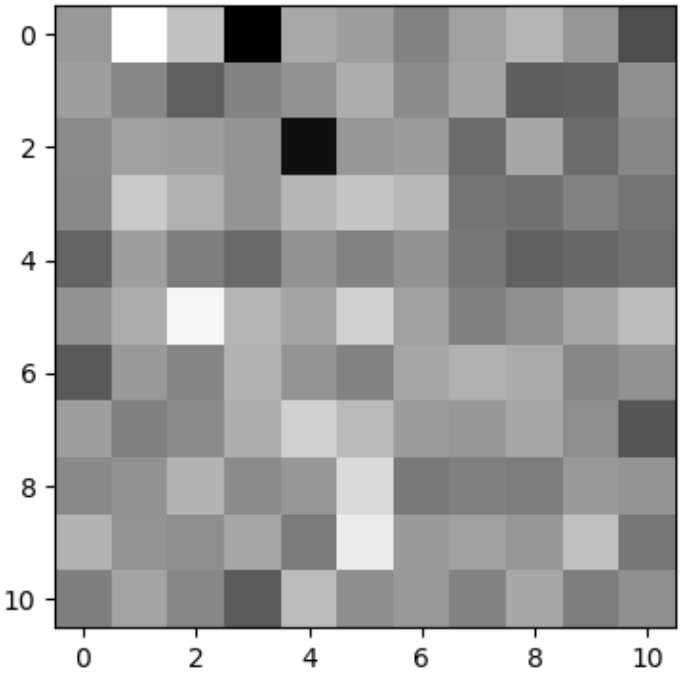
x=436:447, y=1709:1720 around HotPixel(x=441.0, y=1714.0, value=246.0)



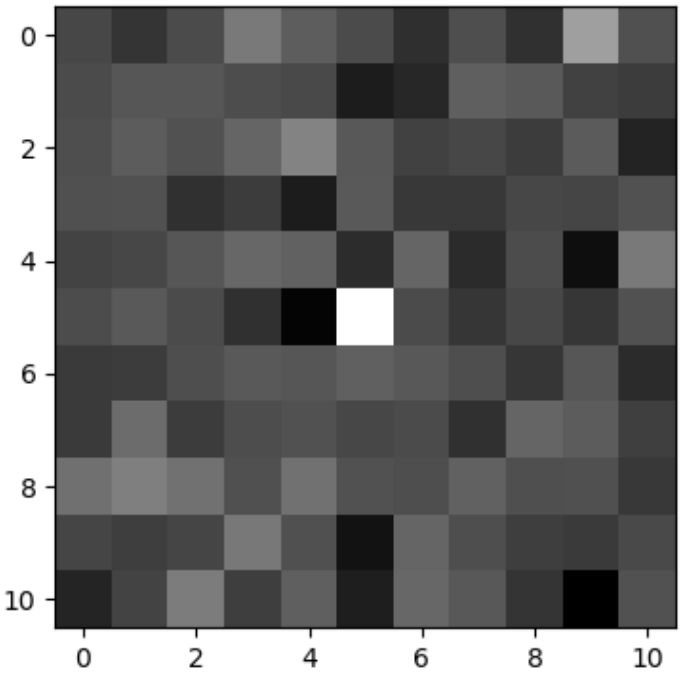
x=447:458, y=1709:1720 around HotPixel(x=452.0, y=1714.0, value=350.5)



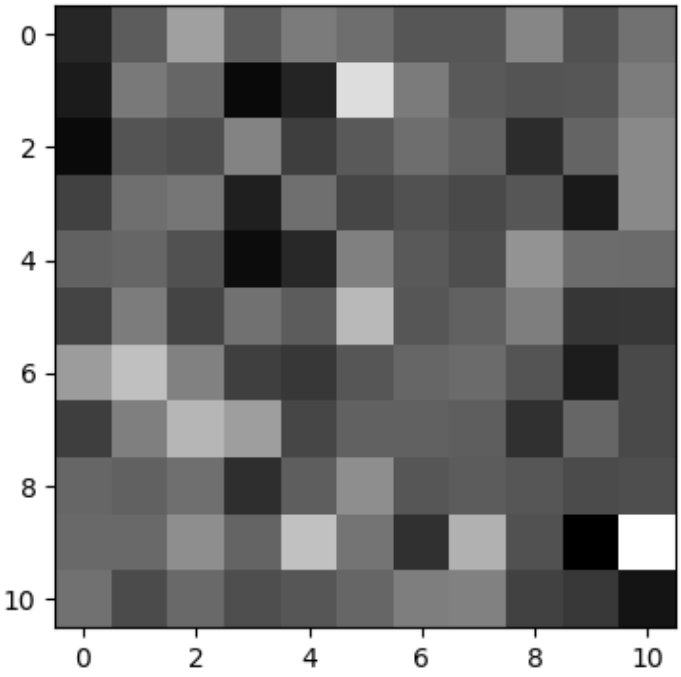
x=450:461, y=1709:1720 around HotPixel(x=455.0, y=1714.0, value=200.5)



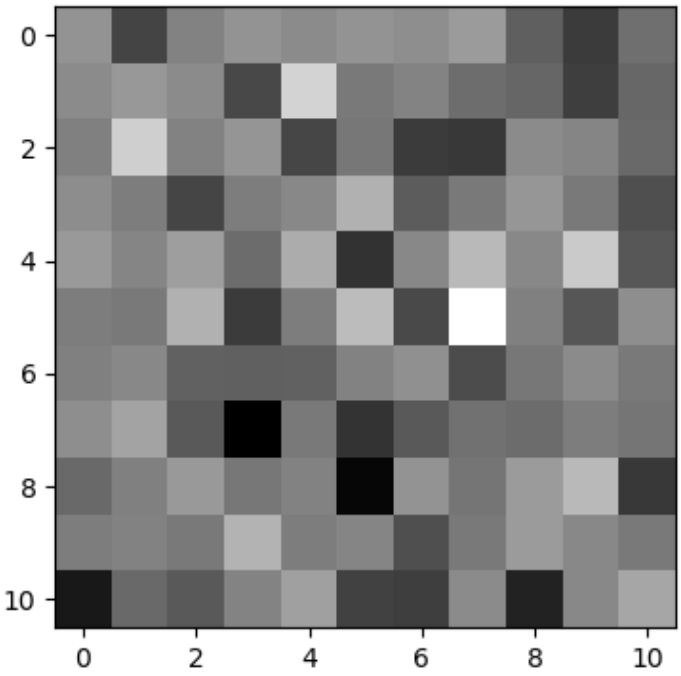
x=490:501, y=1709:1720 around HotPixel(x=495.0, y=1714.0, value=845.0)



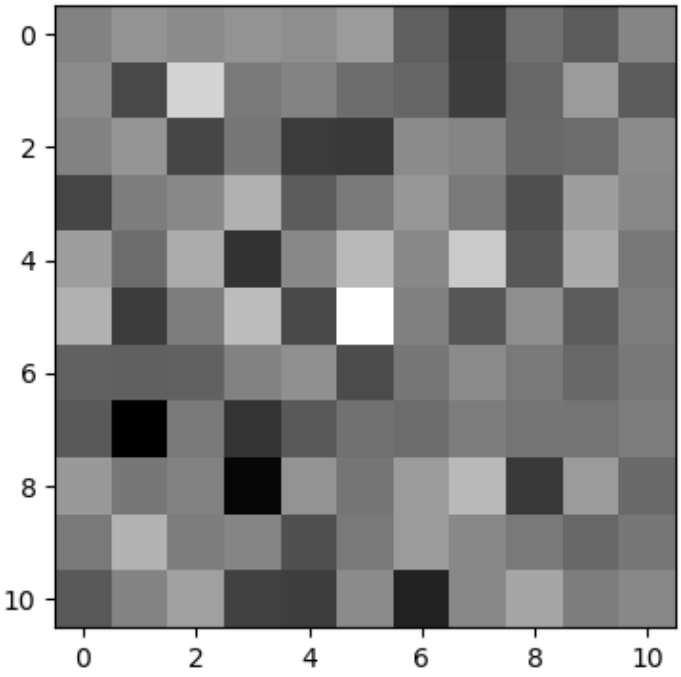
x=400:411, y=1710:1721 around HotPixel(x=405.0, y=1715.0, value=307.5)



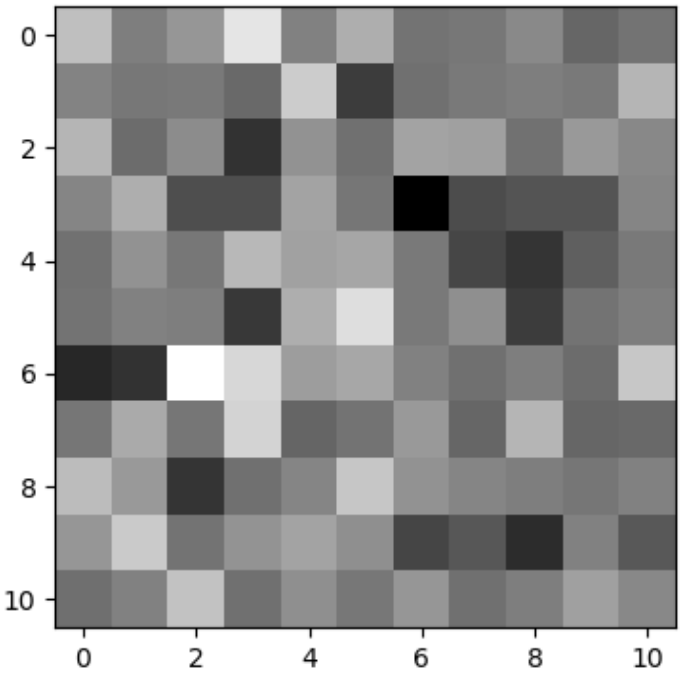
x=432:443, y=1710:1721 around HotPixel(x=437.0, y=1715.0, value=203.0)



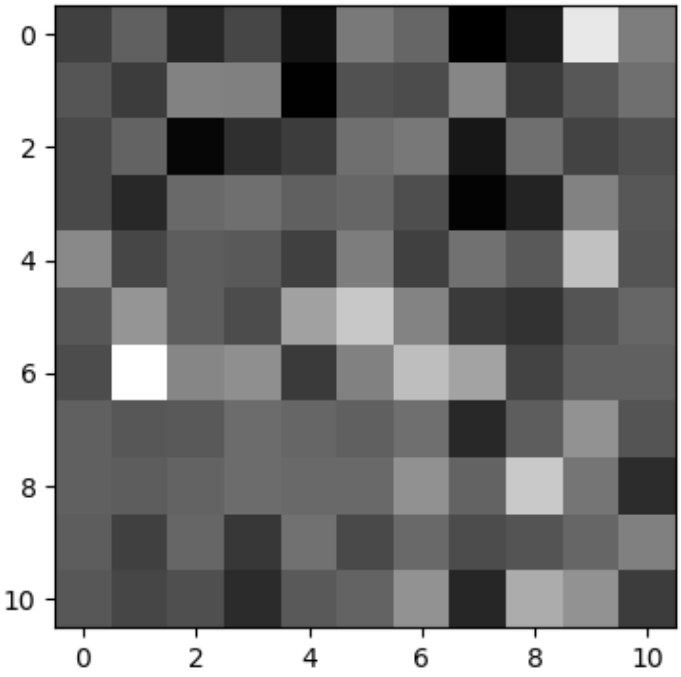
x=434:445, y=1710:1721 around HotPixel(x=439.0, y=1715.0, value=417.0)



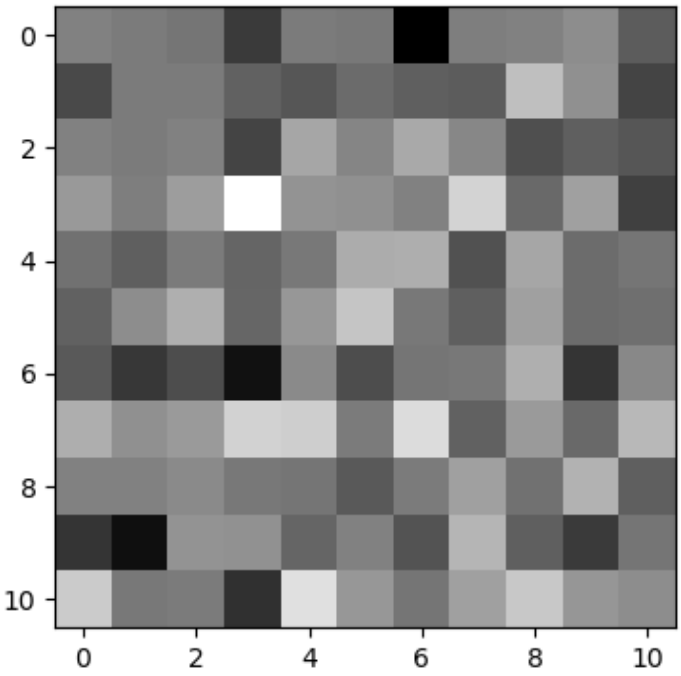
x=472:483, y=1710:1721 around HotPixel(x=477.0, y=1715.0, value=246.0)



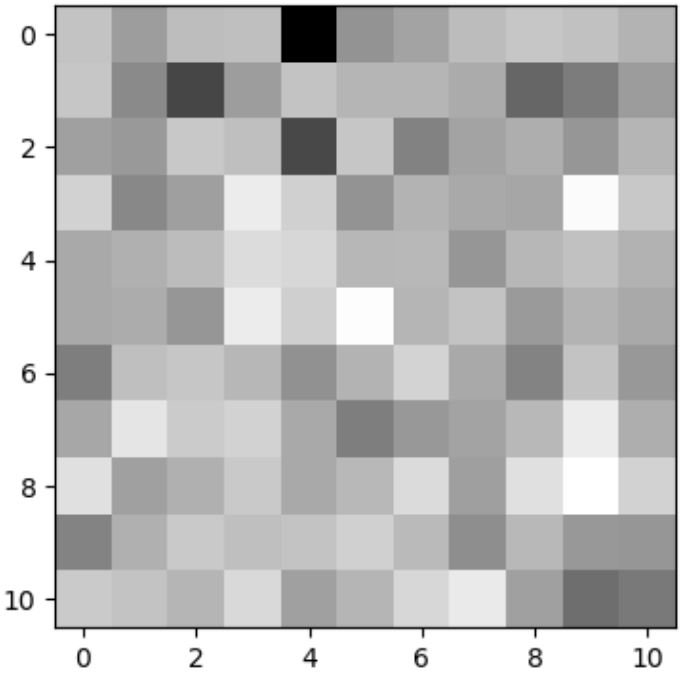
x=396:407, y=1711:1722 around HotPixel(x=401.0, y=1716.0, value=330.5)



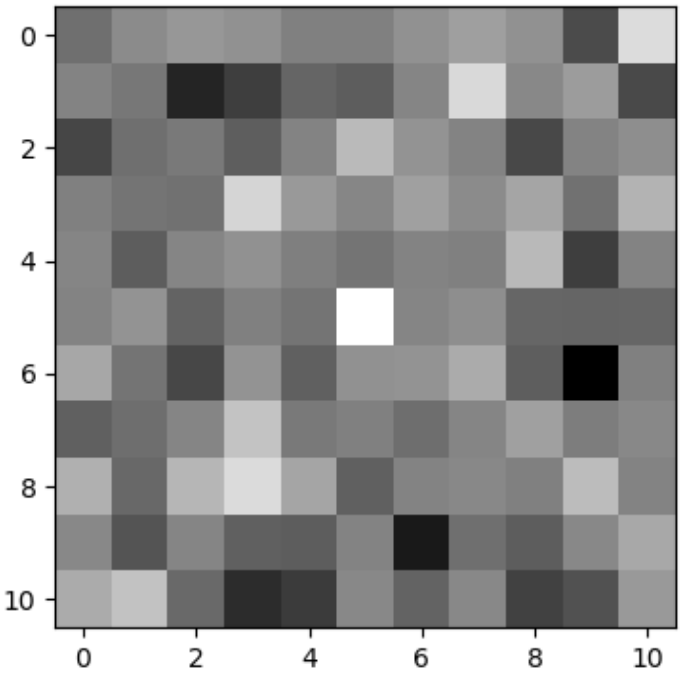
x=411:422, y=1711:1722 around HotPixel(x=416.0, y=1716.0, value=211.0)



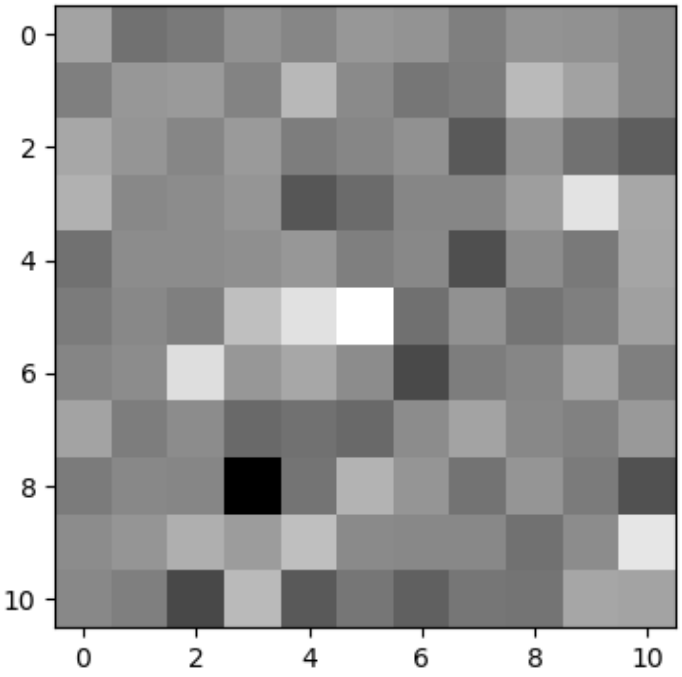
x=420:431, y=1711:1722 around HotPixel(x=425.0, y=1716.0, value=263.0)



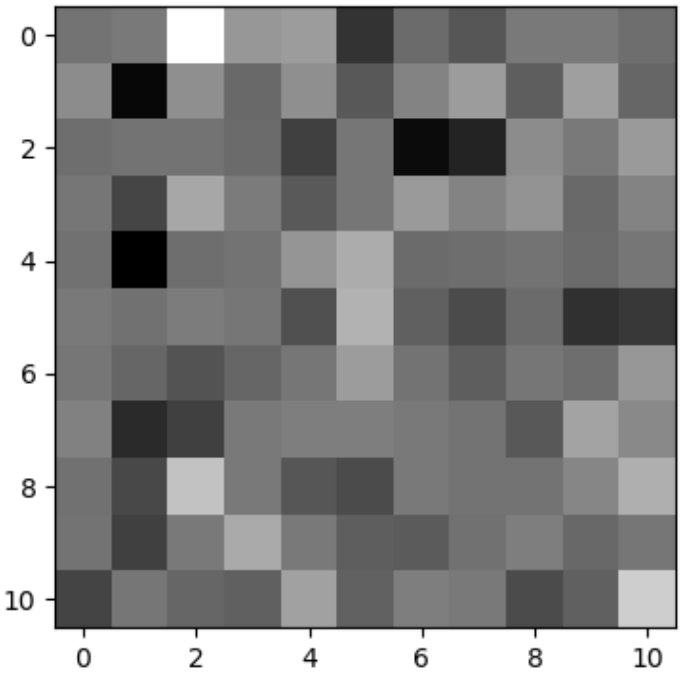
x=426:437, y=1711:1722 around HotPixel(x=431.0, y=1716.0, value=382.0)



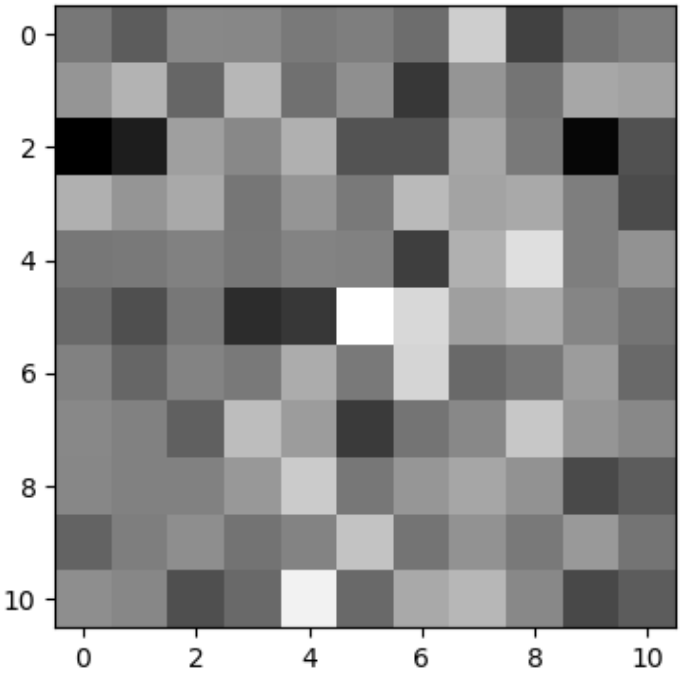
x=443:454, y=1711:1722 around HotPixel(x=448.0, y=1716.0, value=460.0)



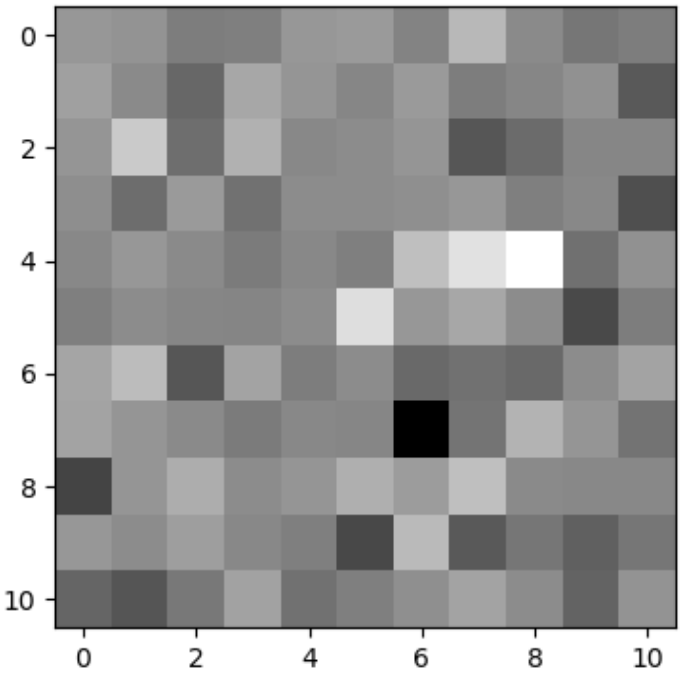
x=463:474, y=1711:1722 around HotPixel(x=468.0, y=1716.0, value=201.5)



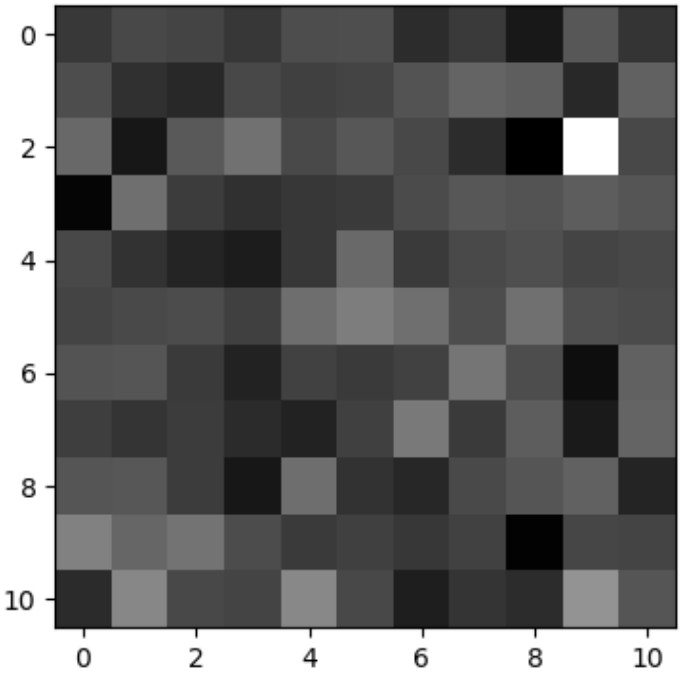
x=469:480, y=1711:1722 around HotPixel(x=474.0, y=1716.0, value=330.0)



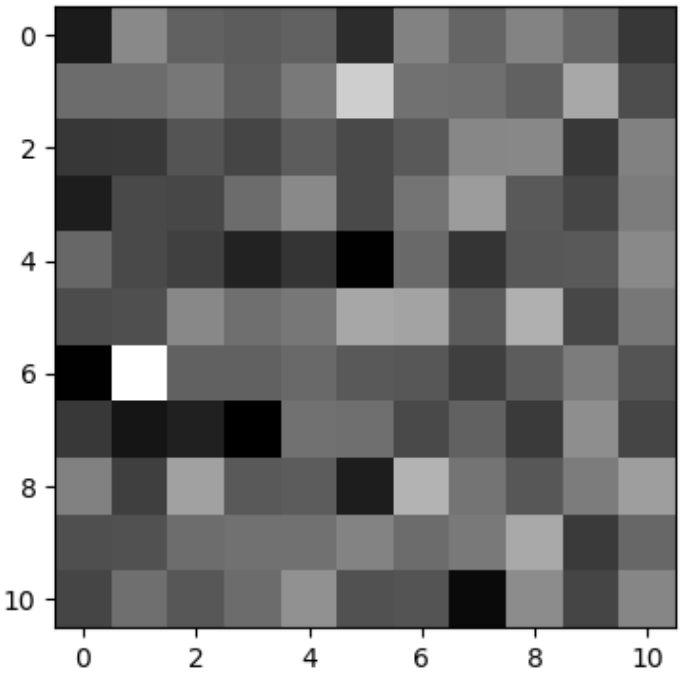
x=440:451, y=1712:1723 around HotPixel(x=445.0, y=1717.0, value=326.5)



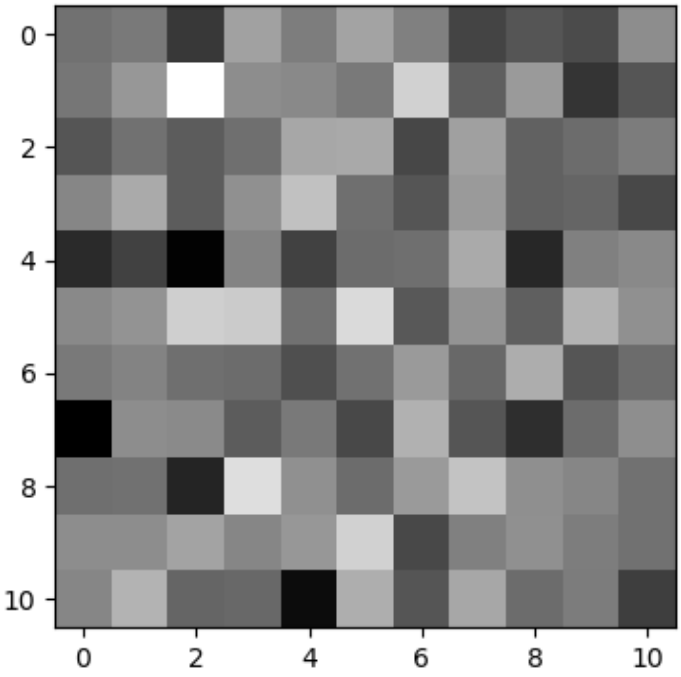
x=486:497, y=1712:1723 around HotPixel(x=491.0, y=1717.0, value=244.0)



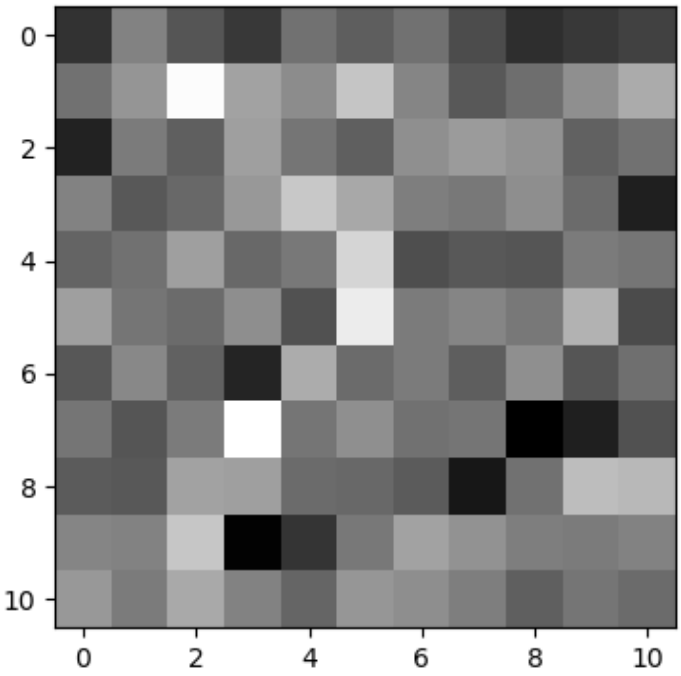
x=409:420, y=1713:1724 around HotPixel(x=414.0, y=1718.0, value=247.0)



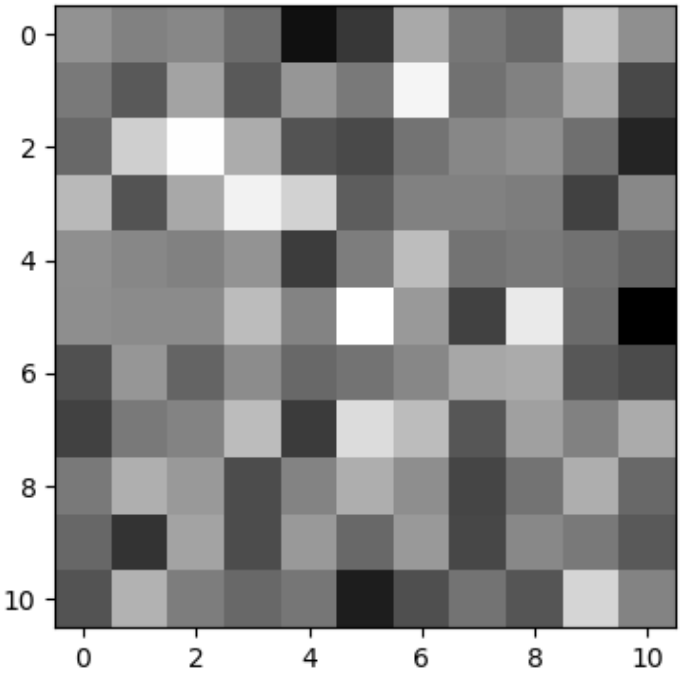
x=412:423, y=1713:1724 around HotPixel(x=417.0, y=1718.0, value=276.5)



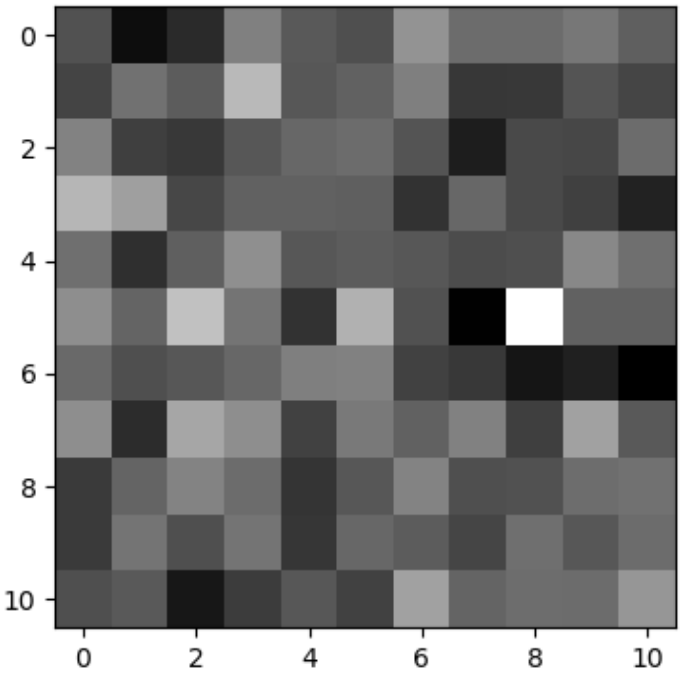
x=450:461, y=1713:1724 around HotPixel(x=455.0, y=1718.0, value=308.0)



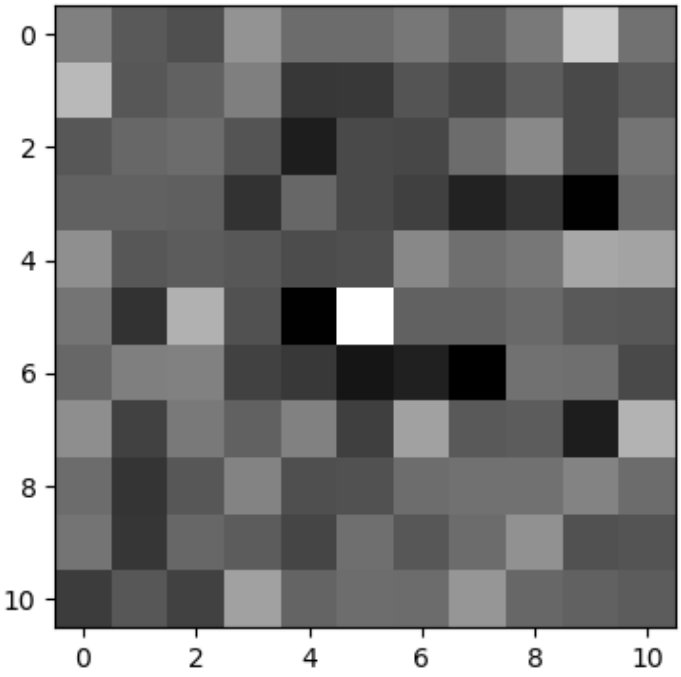
x=399:410, y=1714:1725 around HotPixel(x=404.0, y=1719.0, value=333.0)



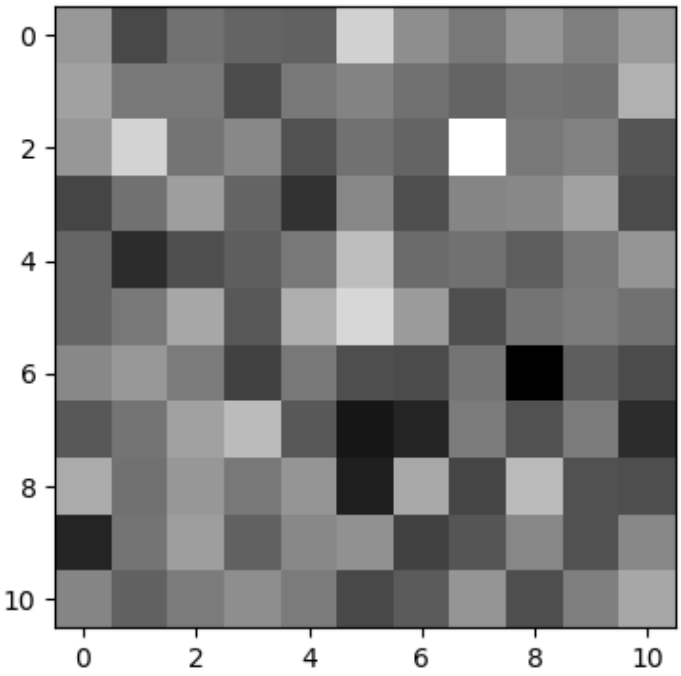
x=402:413, y=1714:1725 around HotPixel(x=407.0, y=1719.0, value=280.5)



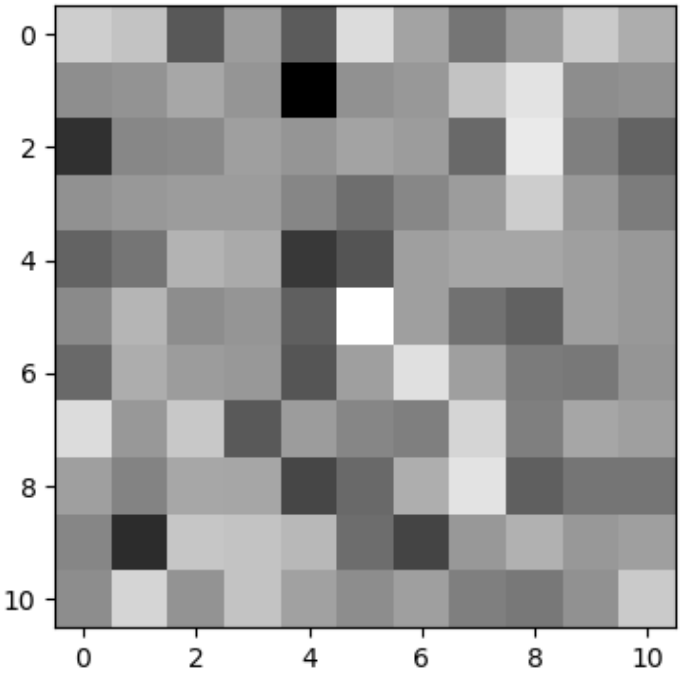
x=405:416, y=1714:1725 around HotPixel(x=410.0, y=1719.0, value=540.5)



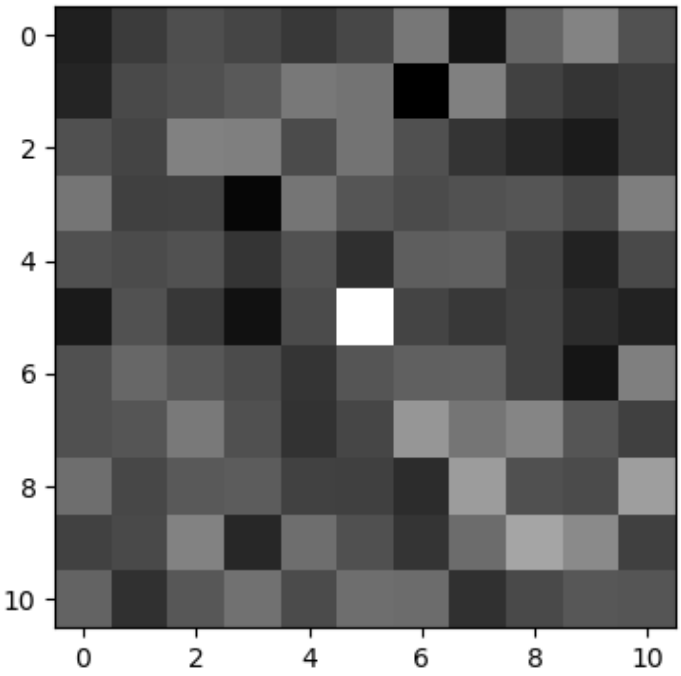
x=424:435, y=1714:1725 around HotPixel(x=429.0, y=1719.0, value=271.0)



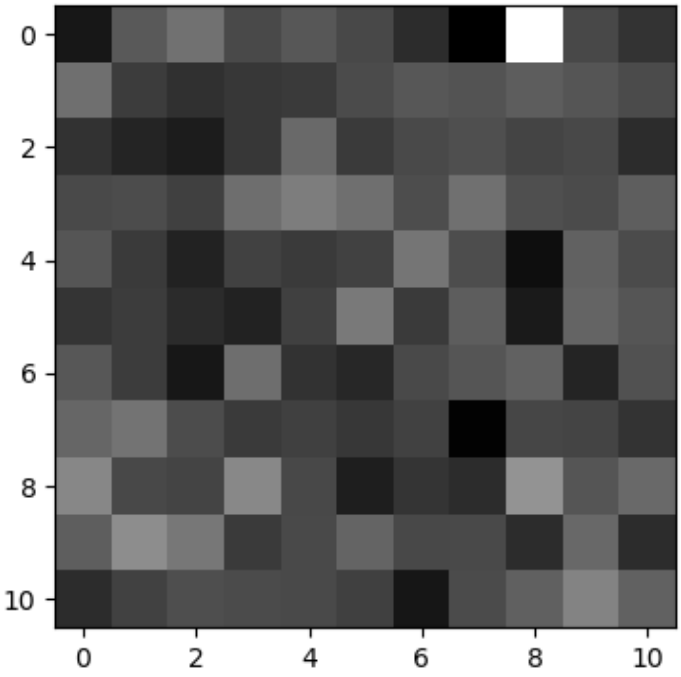
x=460:471, y=1714:1725 around HotPixel(x=465.0, y=1719.0, value=253.5)



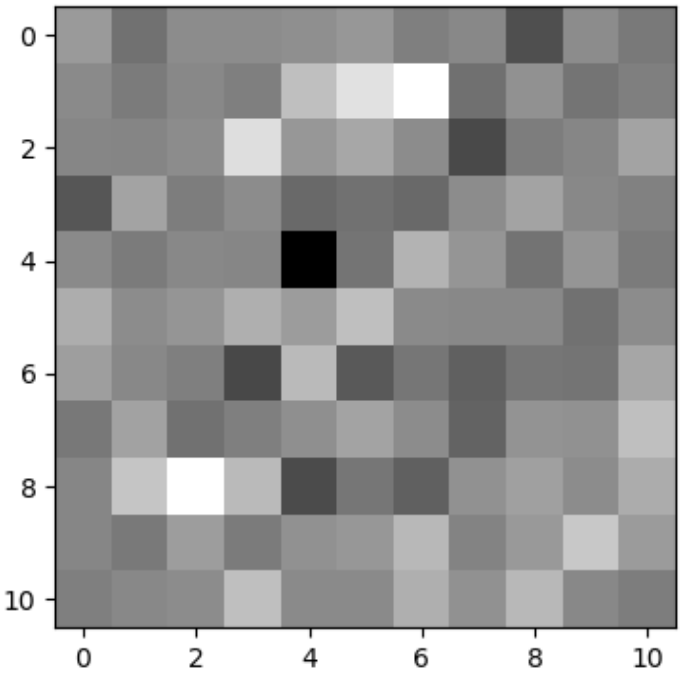
x=480:491, y=1714:1725 around HotPixel(x=485.0, y=1719.0, value=664.0)



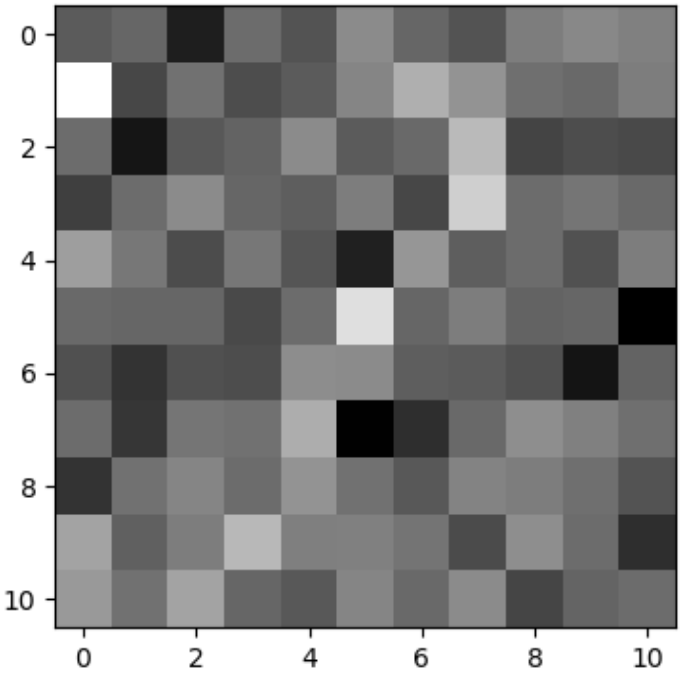
x=487:498, y=1714:1725 around HotPixel(x=492.0, y=1719.0, value=231.0)



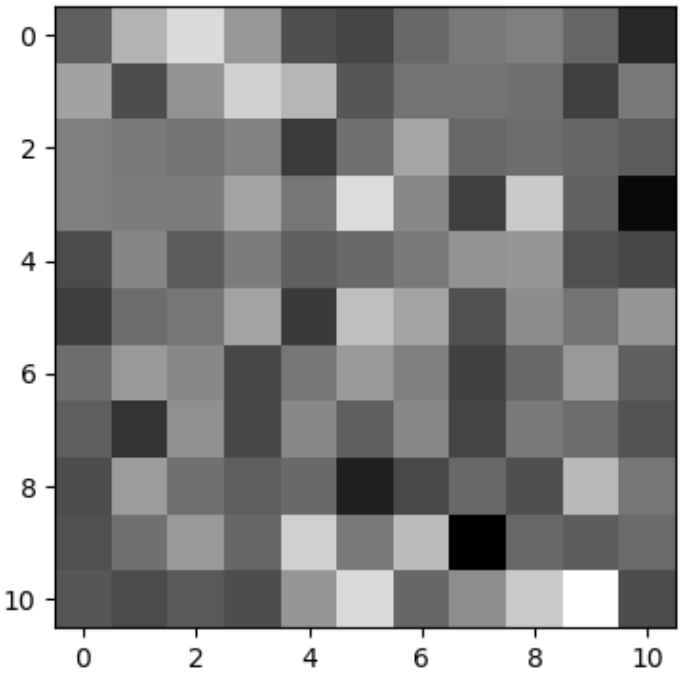
x=442:453, y=1715:1726 around HotPixel(x=447.0, y=1720.0, value=202.5)



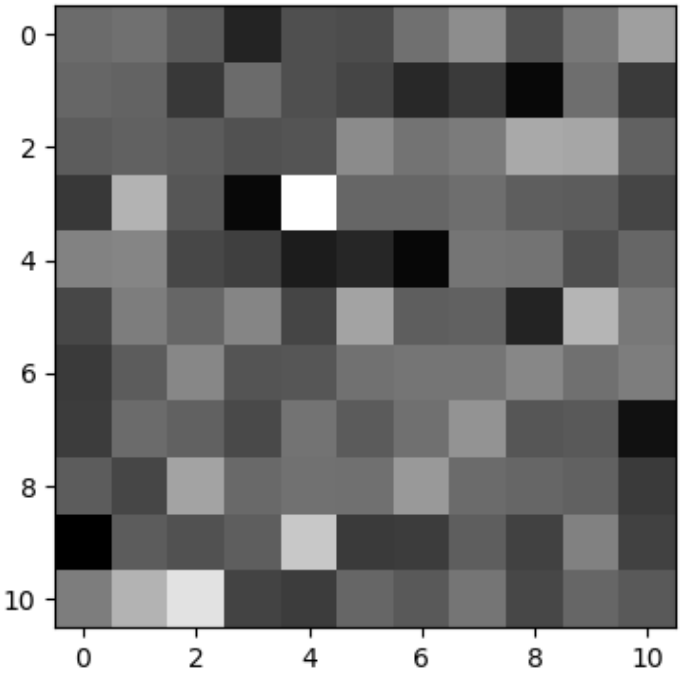
x=448:459, y=1715:1726 around HotPixel(x=453.0, y=1720.0, value=360.0)



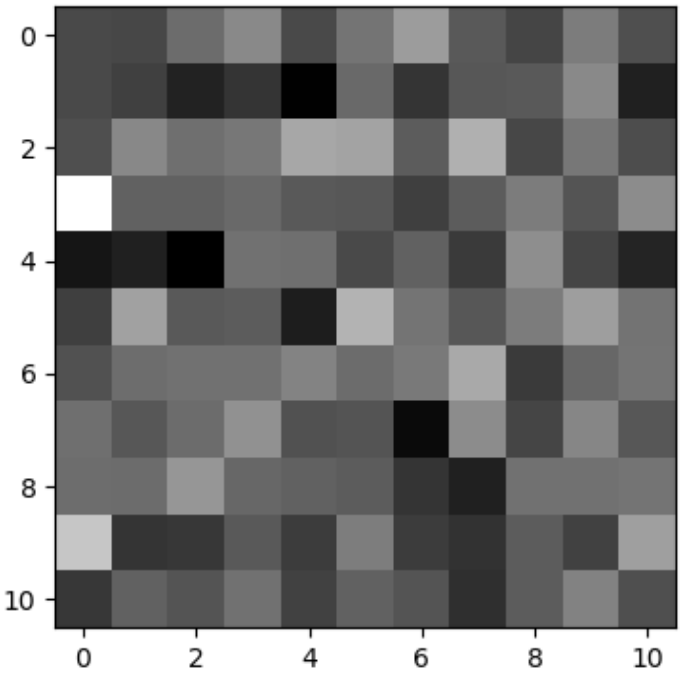
x=399:410, y=1716:1727 around HotPixel(x=404.0, y=1721.0, value=245.0)



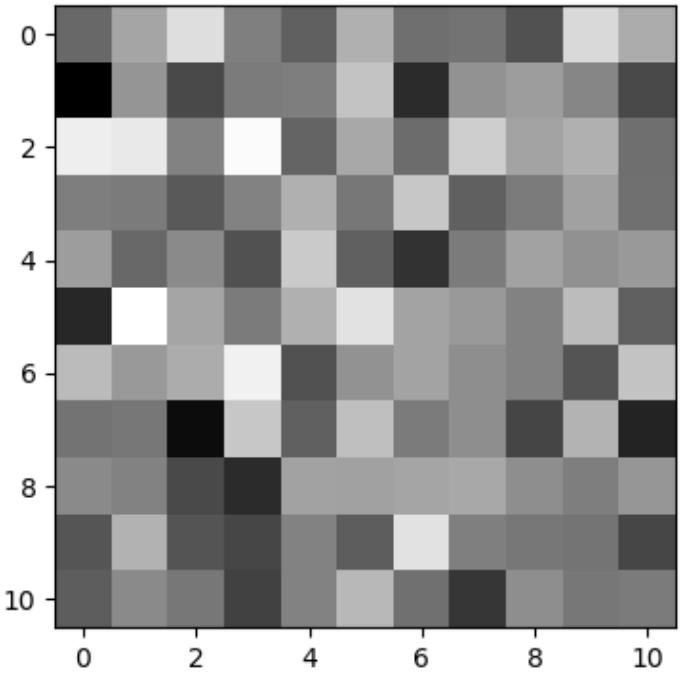
x=406:417, y=1716:1727 around HotPixel(x=411.0, y=1721.0, value=229.0)



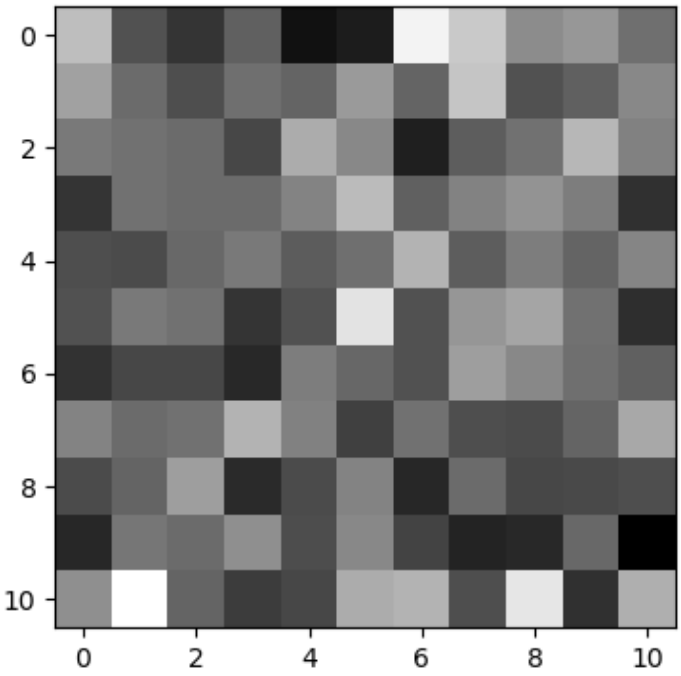
x=410:421, y=1716:1727 around HotPixel(x=415.0, y=1721.0, value=288.0)



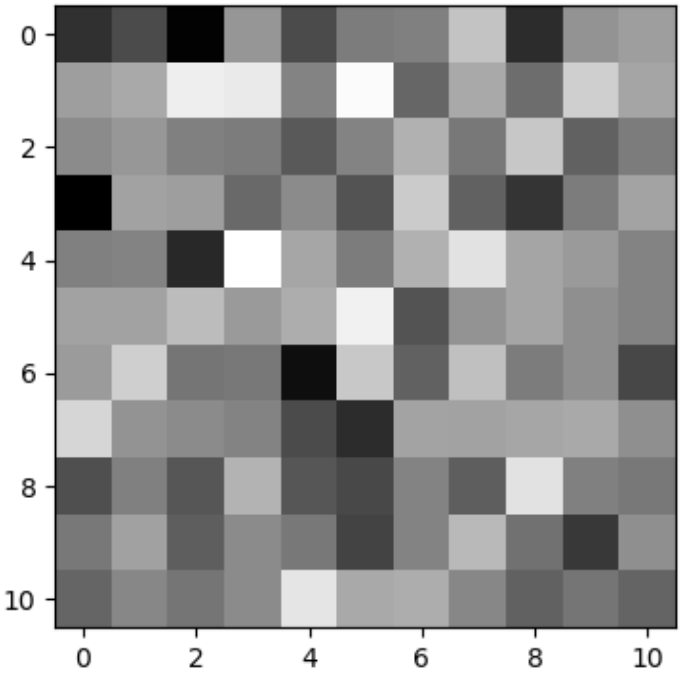
x=414:425, y=1716:1727 around HotPixel(x=419.0, y=1721.0, value=219.0)



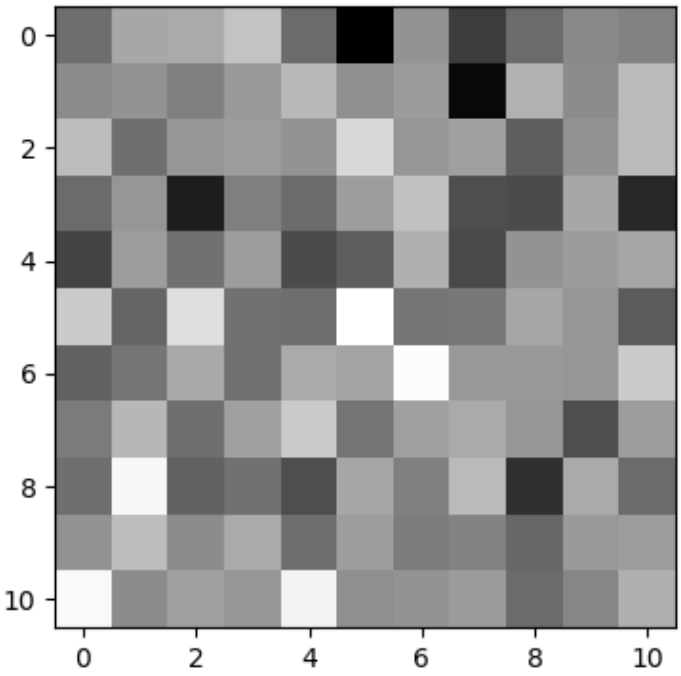
x=468:479, y=1716:1727 around HotPixel(x=473.0, y=1721.0, value=293.5)



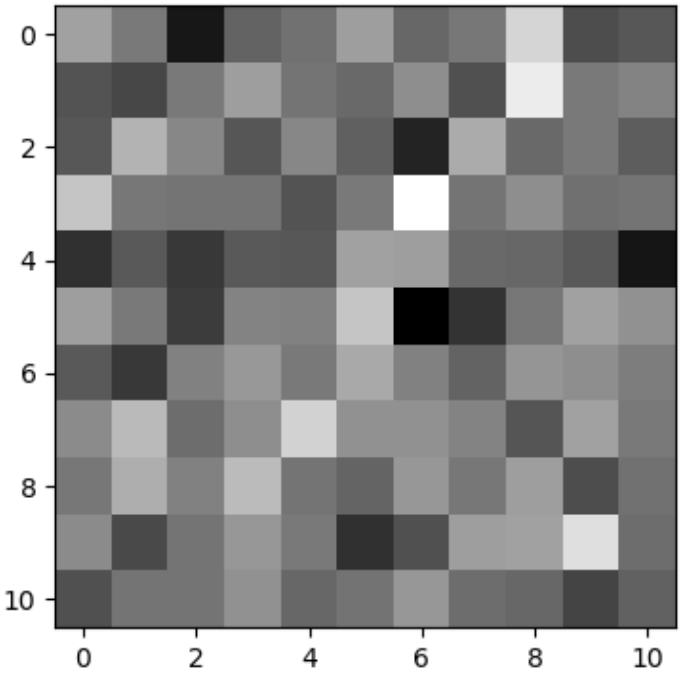
x=412:423, y=1717:1728 around HotPixel(x=417.0, y=1722.0, value=254.0)



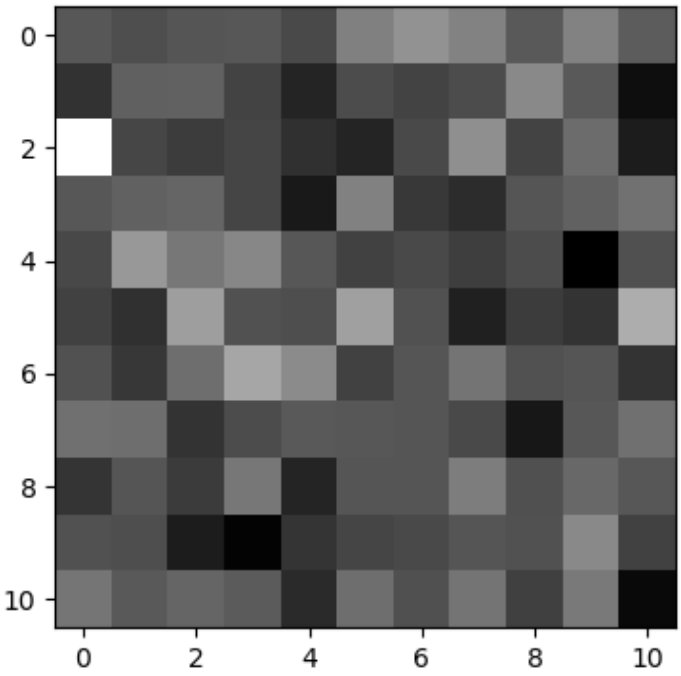
x=430:441, y=1717:1728 around HotPixel(x=435.0, y=1722.0, value=281.0)



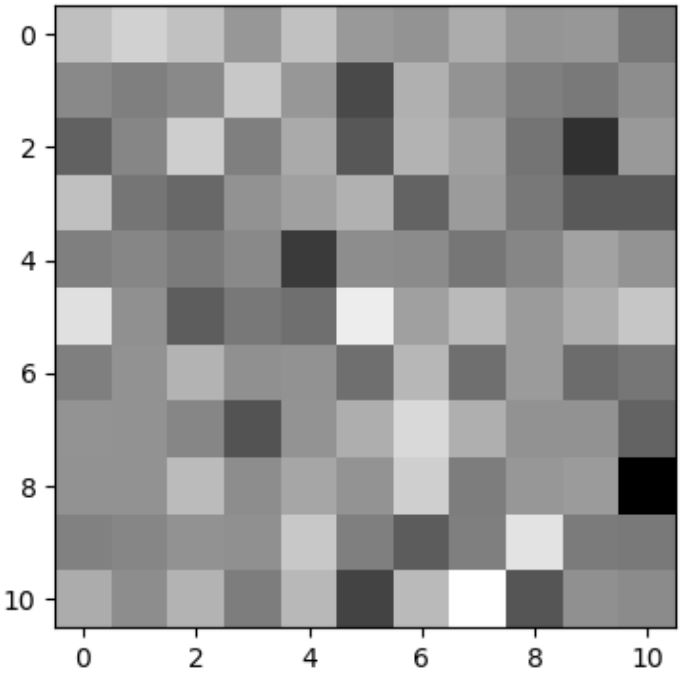
x=447:458, y=1717:1728 around HotPixel(x=452.0, y=1722.0, value=202.5)



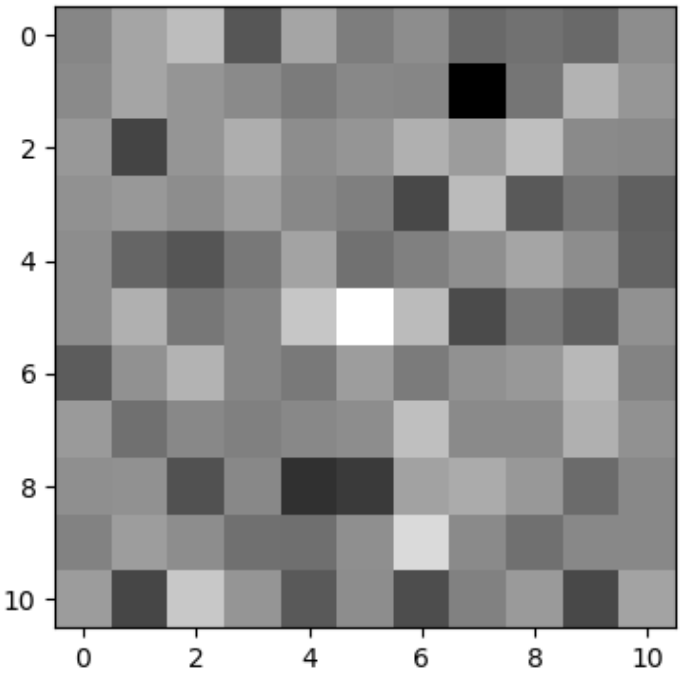
x=485:496, y=1717:1728 around HotPixel(x=490.0, y=1722.0, value=297.0)



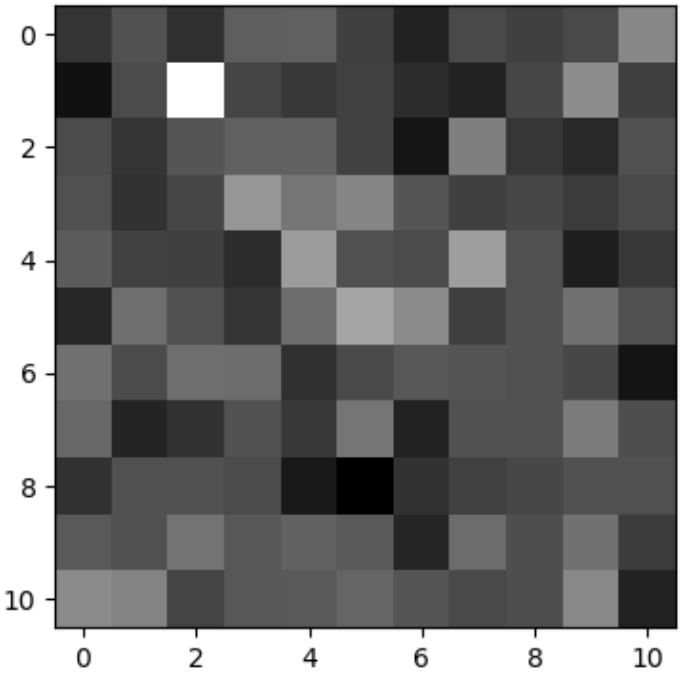
x=490:501, y=1717:1728 around HotPixel(x=495.0, y=1722.0, value=347.0)



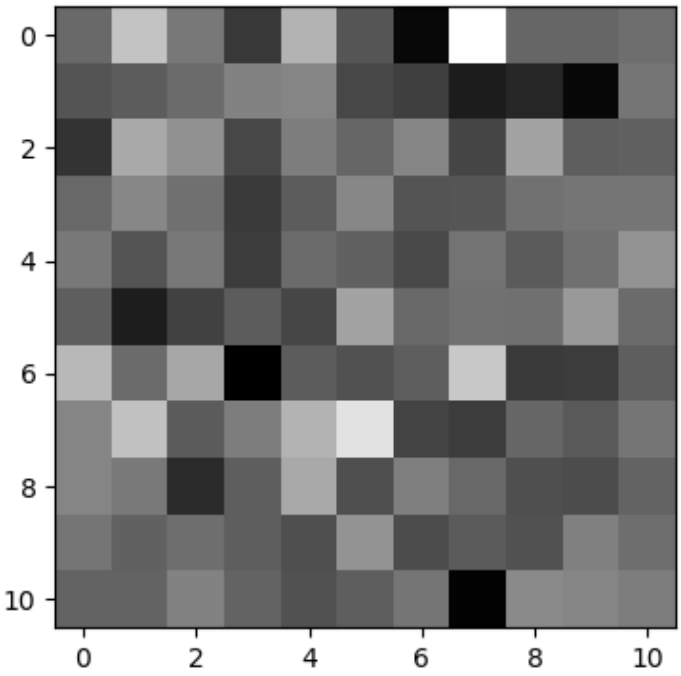
x=439:450, y=1718:1729 around HotPixel(x=444.0, y=1723.0, value=457.5)



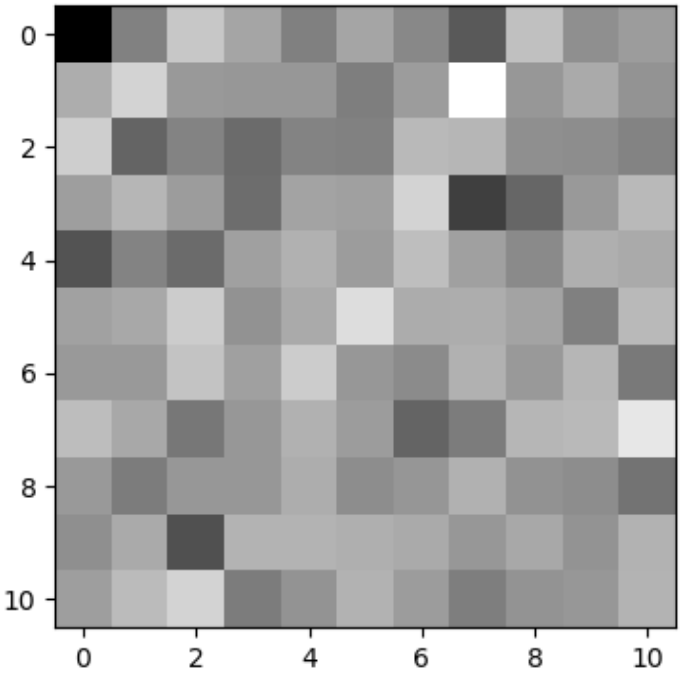
x=483:494, y=1718:1729 around HotPixel(x=488.0, y=1723.0, value=321.0)



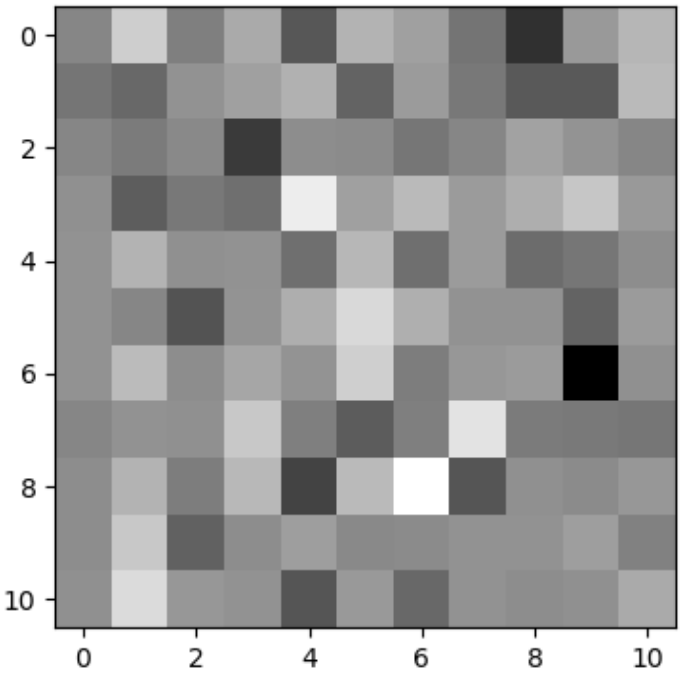
x=403:414, y=1719:1730 around HotPixel(x=408.0, y=1724.0, value=227.0)



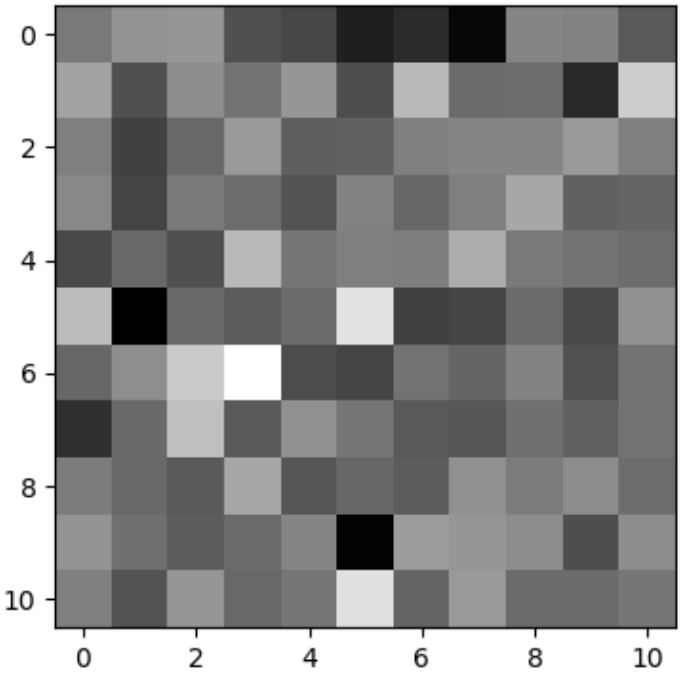
x=446:457, y=1719:1730 around HotPixel(x=451.0, y=1724.0, value=237.0)



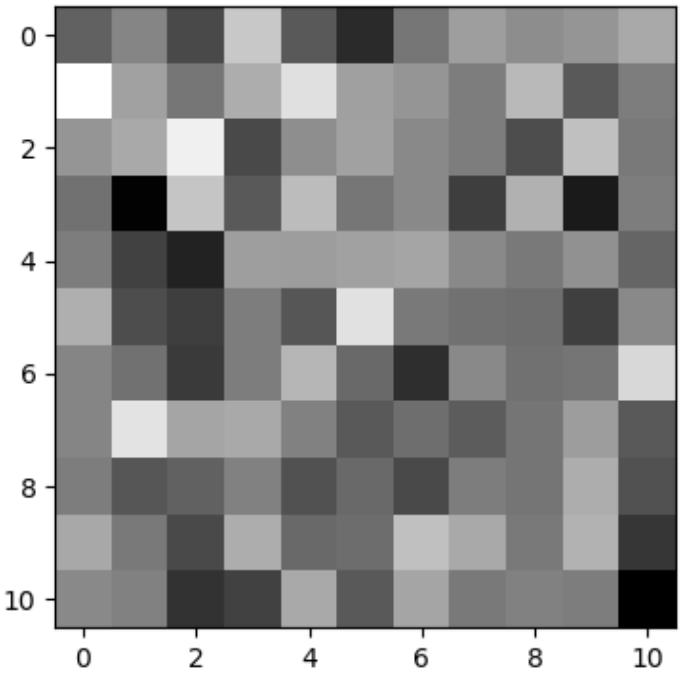
x=491:502, y=1719:1730 around HotPixel(x=496.0, y=1724.0, value=272.0)



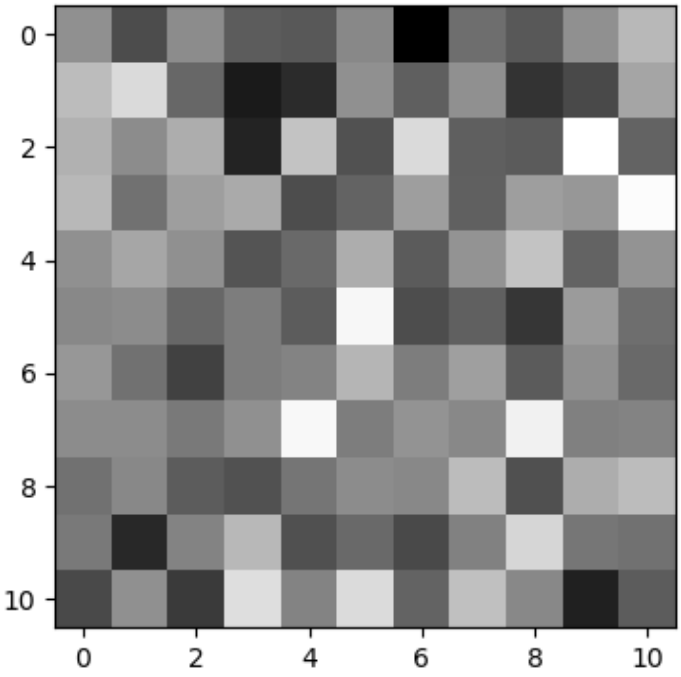
x=405:416, y=1720:1731 around HotPixel(x=410.0, y=1725.0, value=350.5)



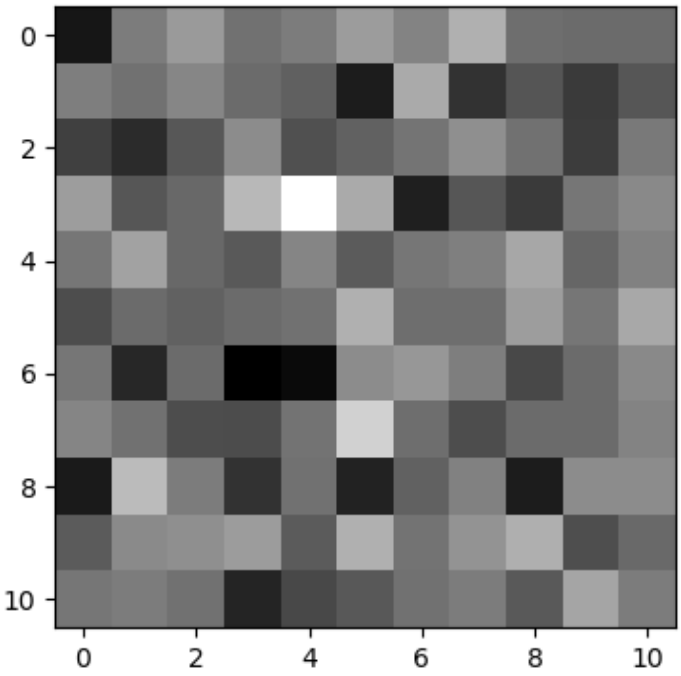
x=415:426, y=1720:1731 around HotPixel(x=420.0, y=1725.0, value=220.0)



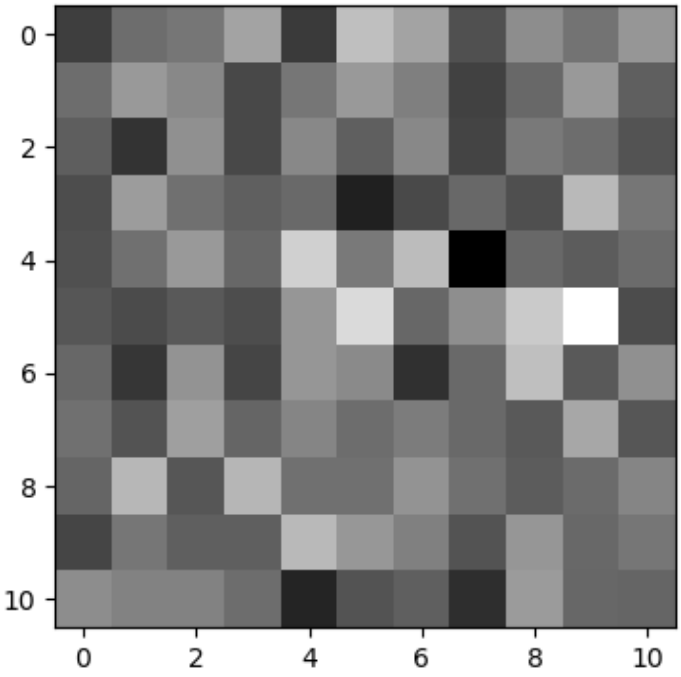
x=426:437, y=1720:1731 around HotPixel(x=431.0, y=1725.0, value=260.0)



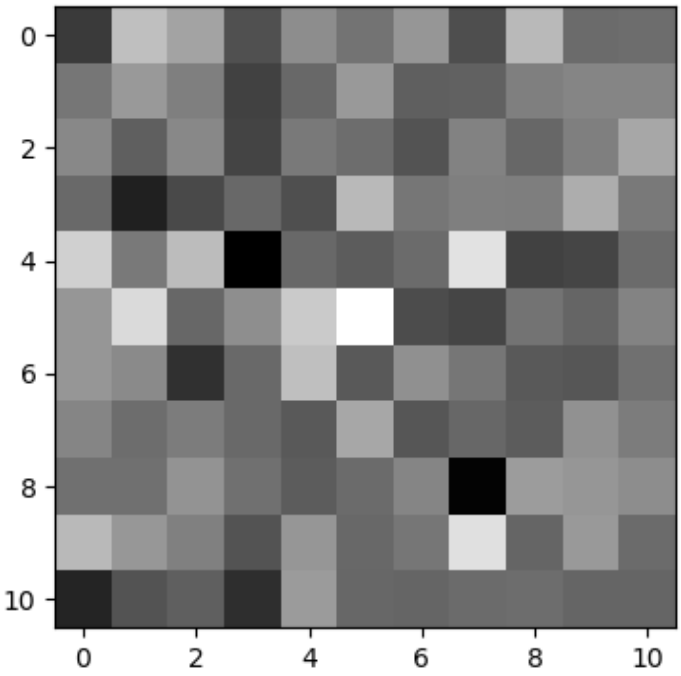
x=440:451, y=1720:1731 around HotPixel(x=445.0, y=1725.0, value=202.5)



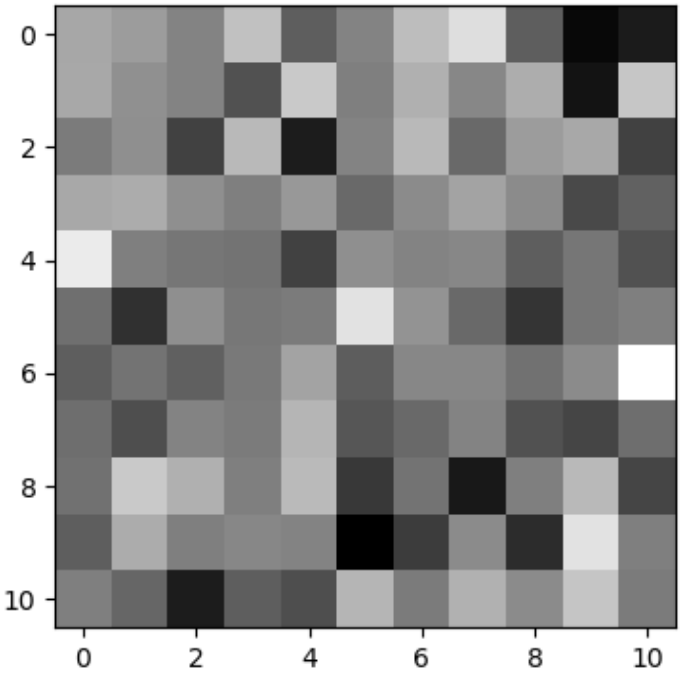
x=399:410, y=1721:1732 around HotPixel(x=404.0, y=1726.0, value=326.5)



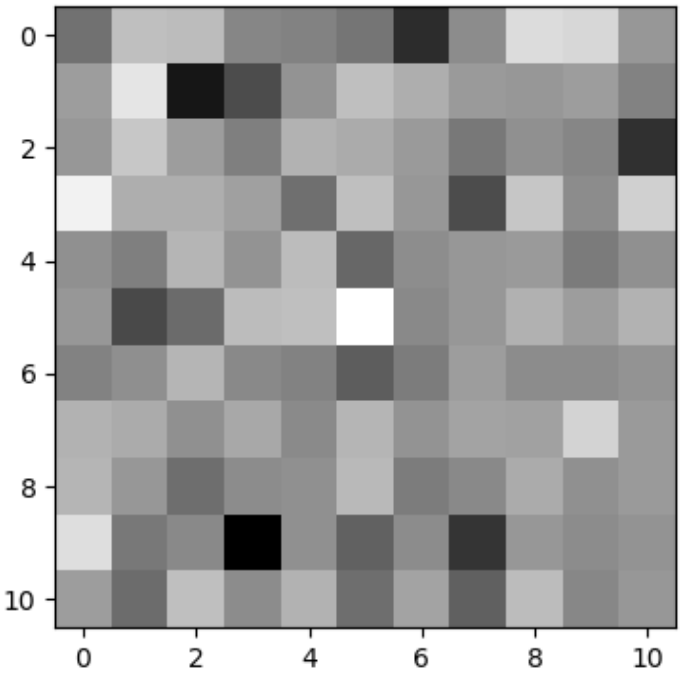
x=403:414, y=1721:1732 around HotPixel(x=408.0, y=1726.0, value=440.5)



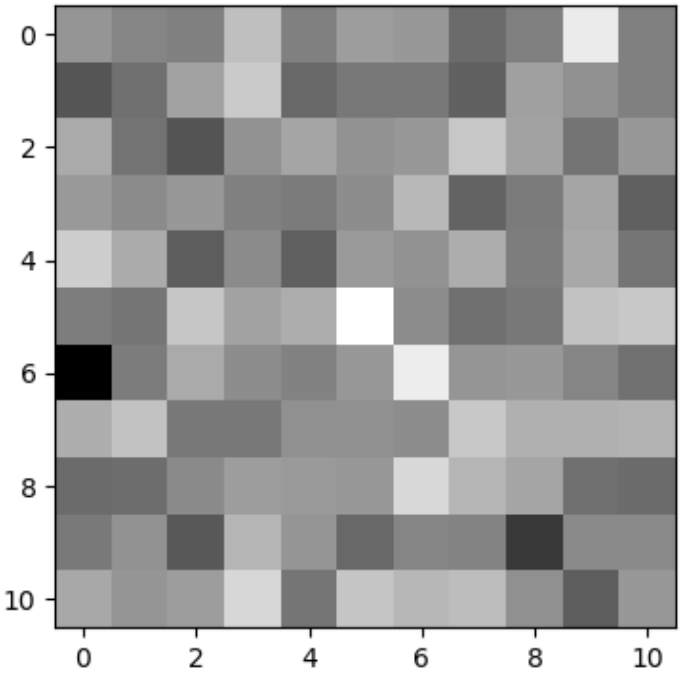
x=420:431, y=1721:1732 around HotPixel(x=425.0, y=1726.0, value=201.0)



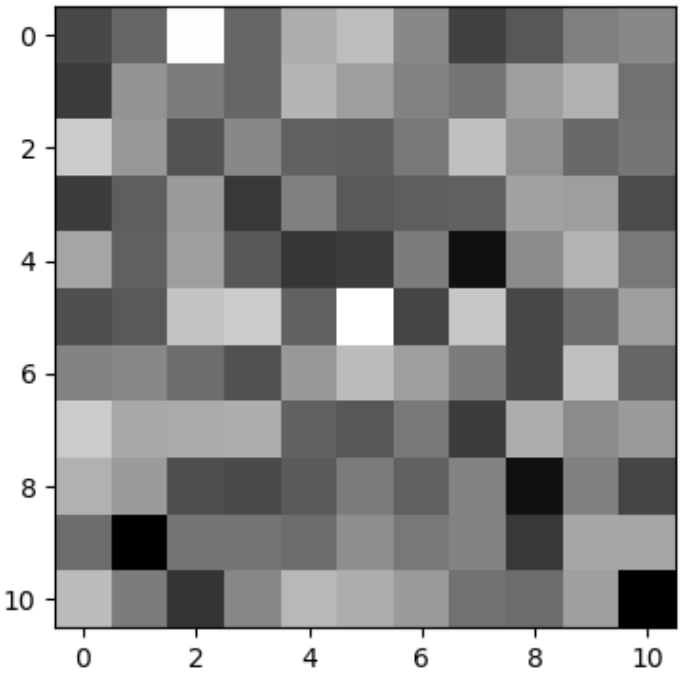
x=451:462, y=1721:1732 around HotPixel(x=456.0, y=1726.0, value=271.5)

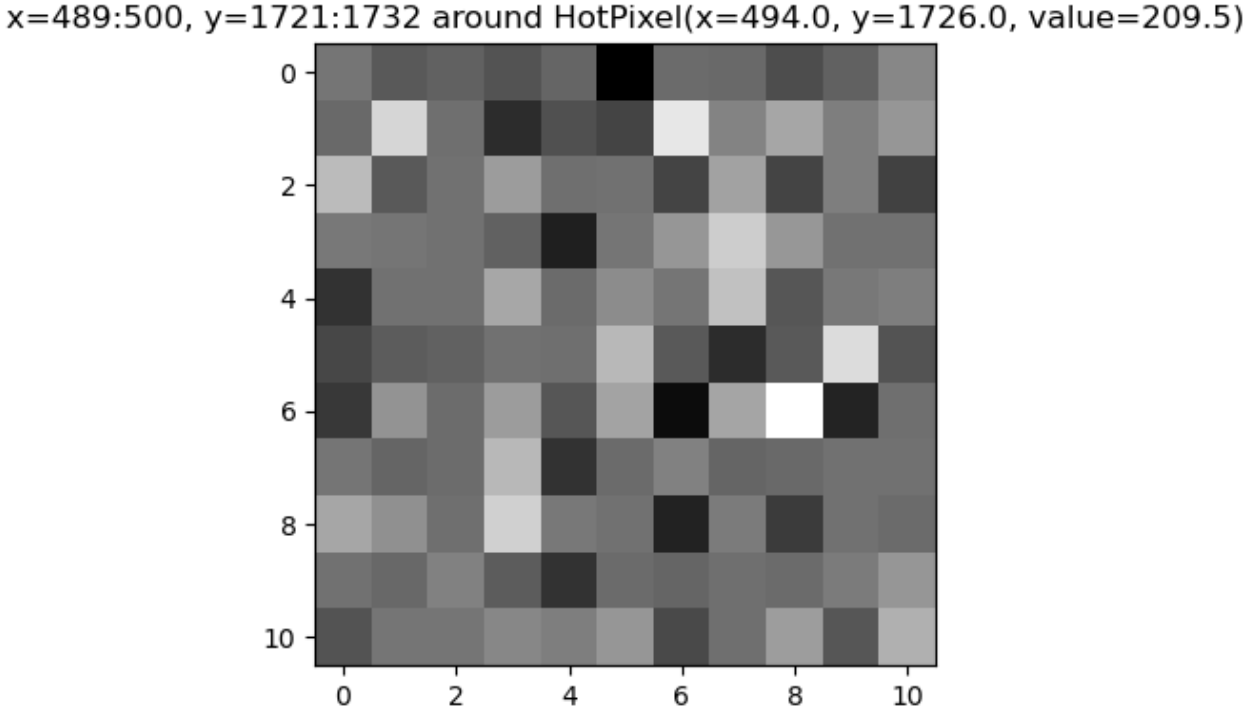


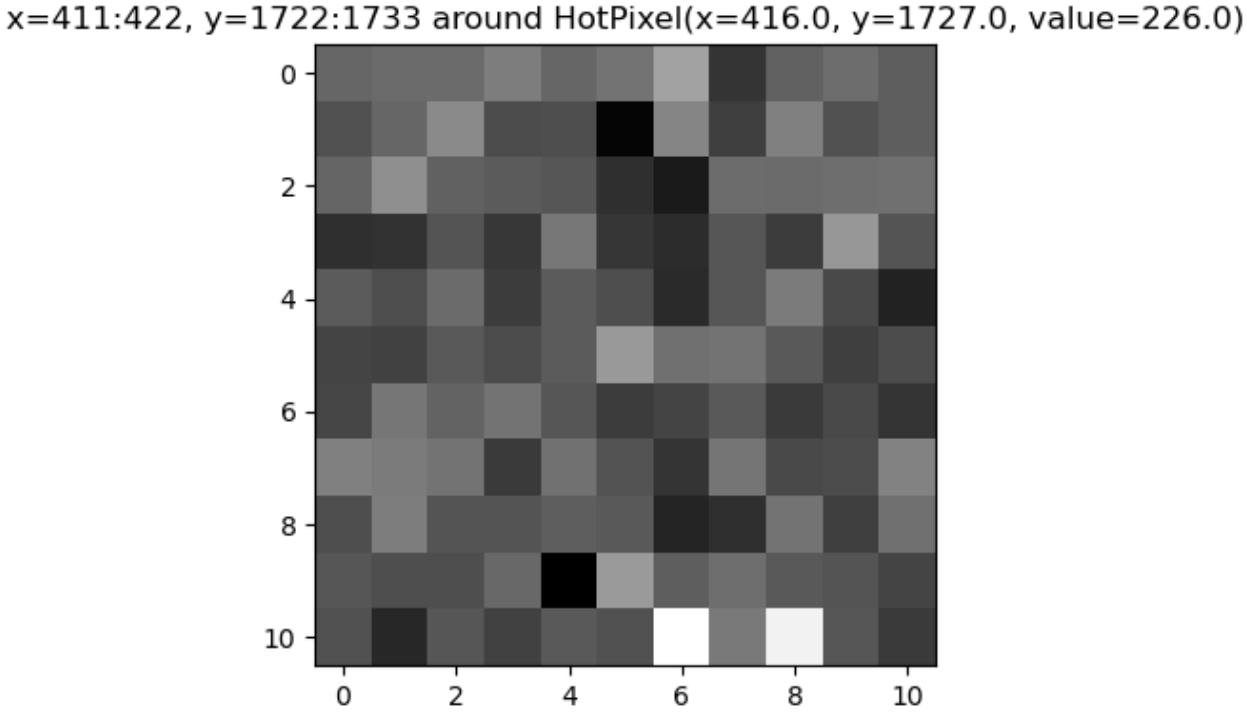
x=464:475, y=1721:1732 around HotPixel(x=469.0, y=1726.0, value=359.5)



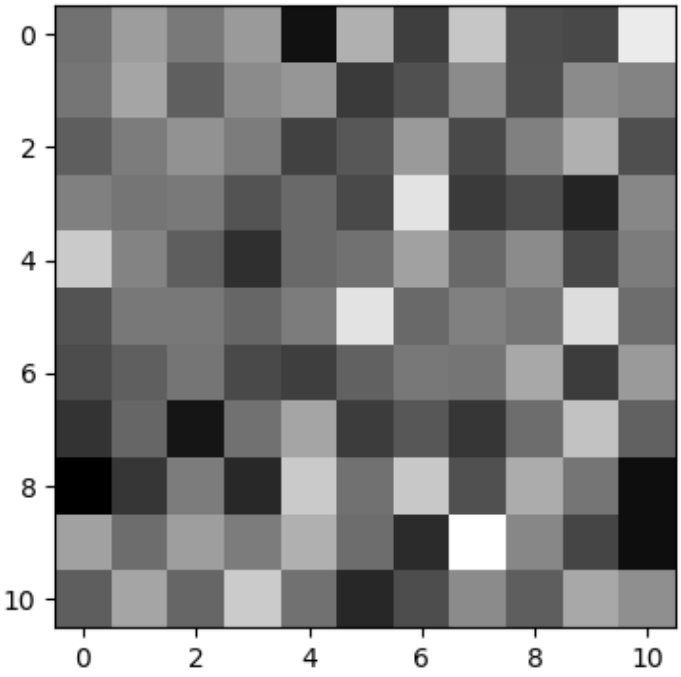
x=471:482, y=1721:1732 around HotPixel(x=476.0, y=1726.0, value=297.0)



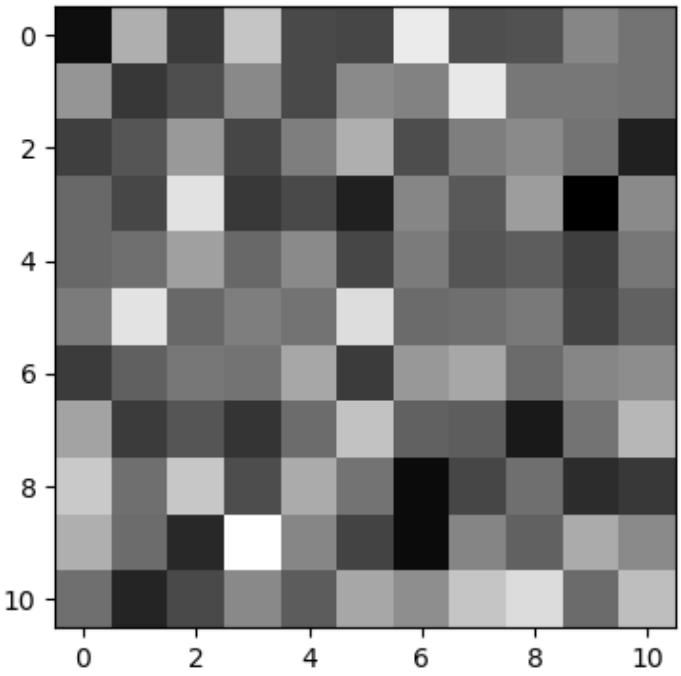

x=489:500, y=1721:1732 around HotPixel(x=494.0, y=1726.0, value=209.5)
0
2
4
6
8
10
0
2
4
6
8
10


x=411:422, y=1722:1733 around HotPixel(x=416.0, y=1727.0, value=226.0)
0
2
4
6
8
10
0
2
4
6
8
10

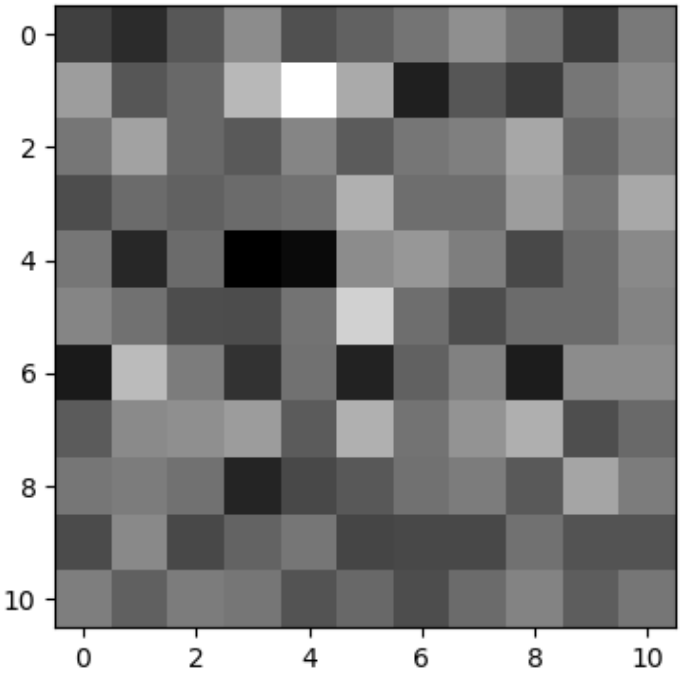
x=425:436, y=1722:1733 around HotPixel(x=430.0, y=1727.0, value=263.5)



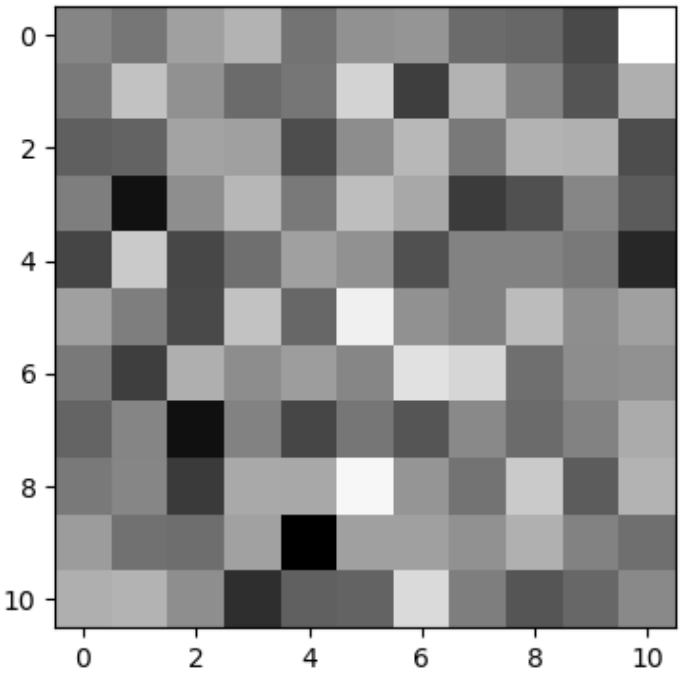
x=429:440, y=1722:1733 around HotPixel(x=434.0, y=1727.0, value=247.0)



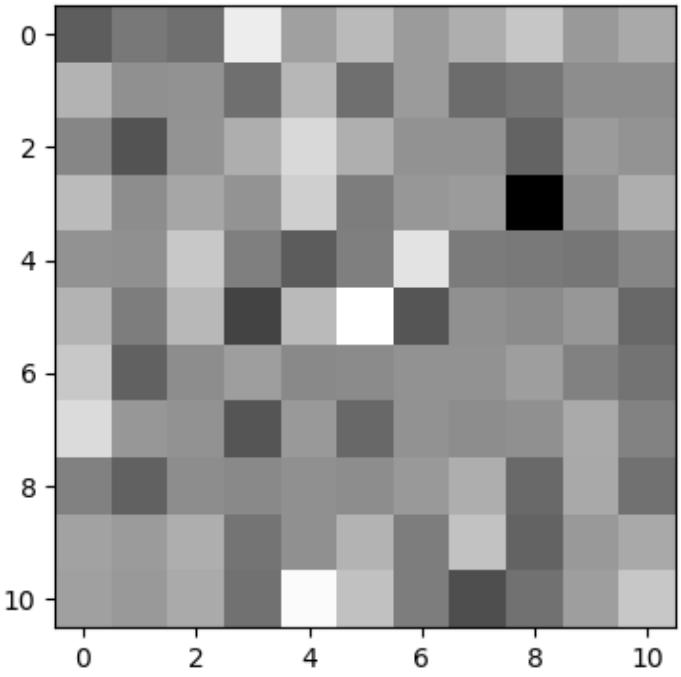
x=440:451, y=1722:1733 around HotPixel(x=445.0, y=1727.0, value=308.0)



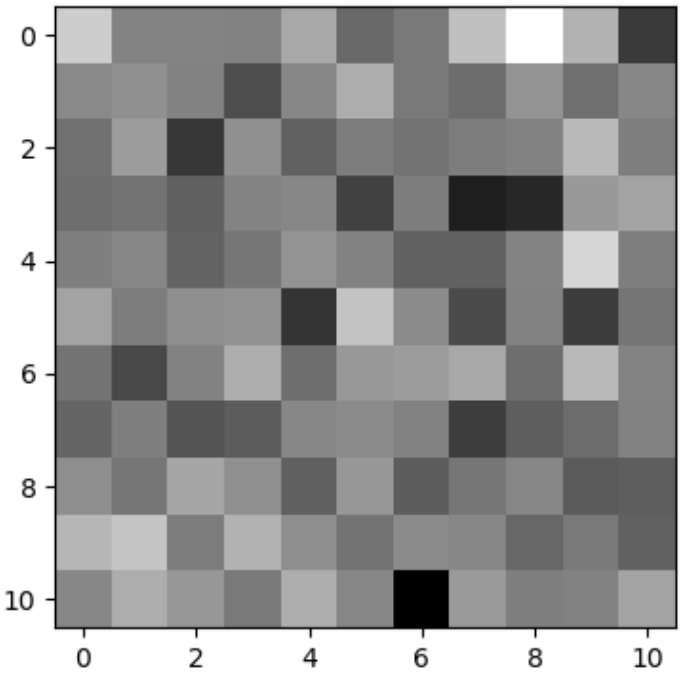
x=477:488, y=1722:1733 around HotPixel(x=482.0, y=1727.0, value=253.0)



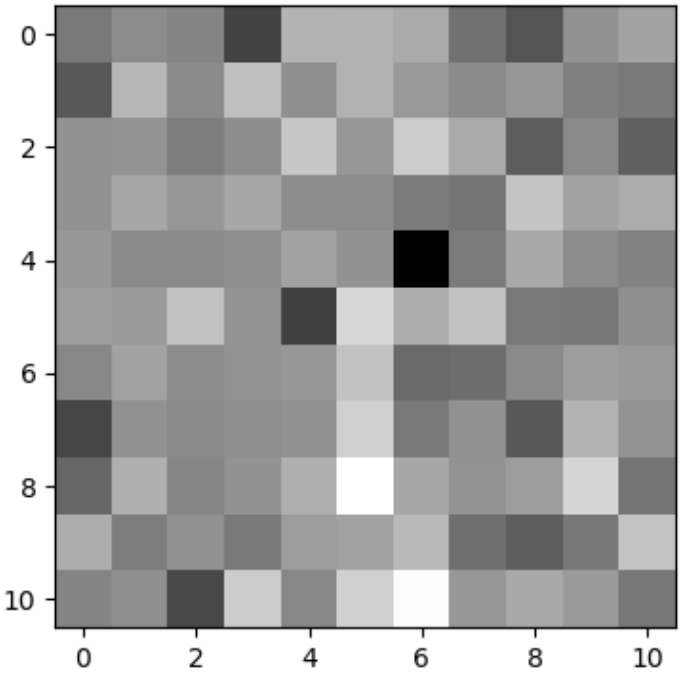
x=492:503, y=1722:1733 around HotPixel(x=497.0, y=1727.0, value=416.0)



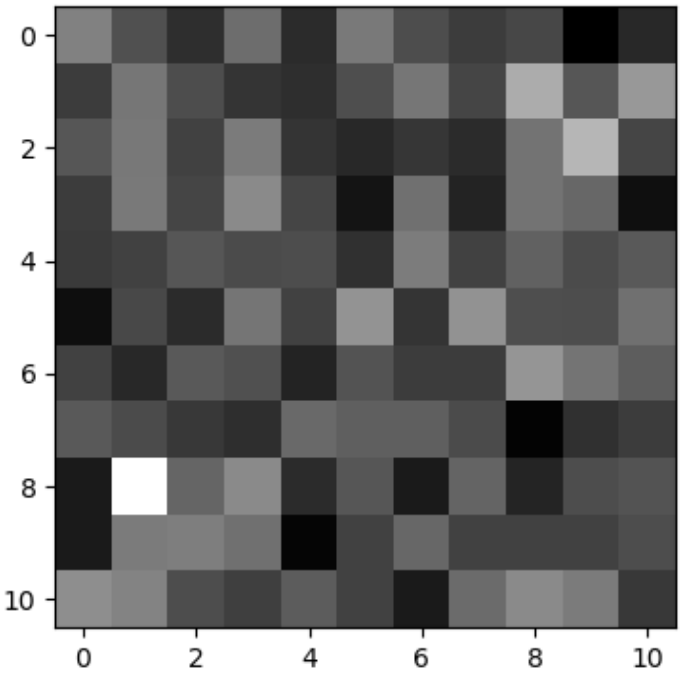
x=436:447, y=1723:1734 around HotPixel(x=441.0, y=1728.0, value=237.0)

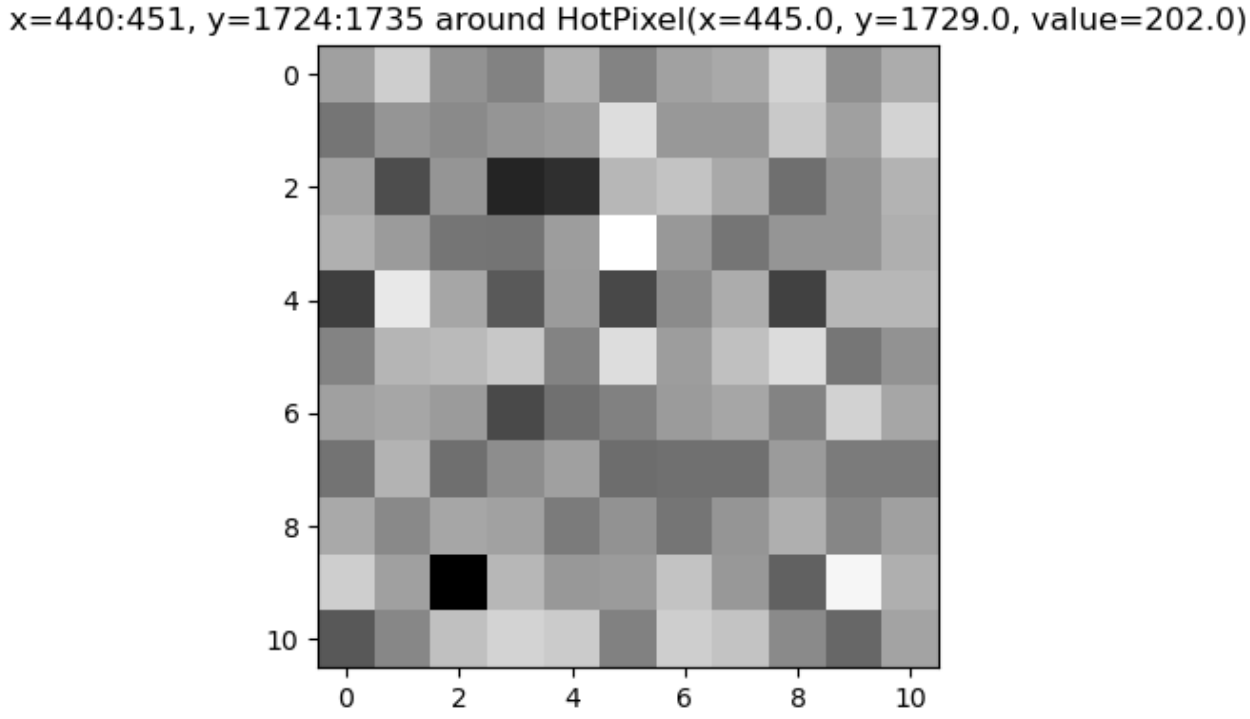


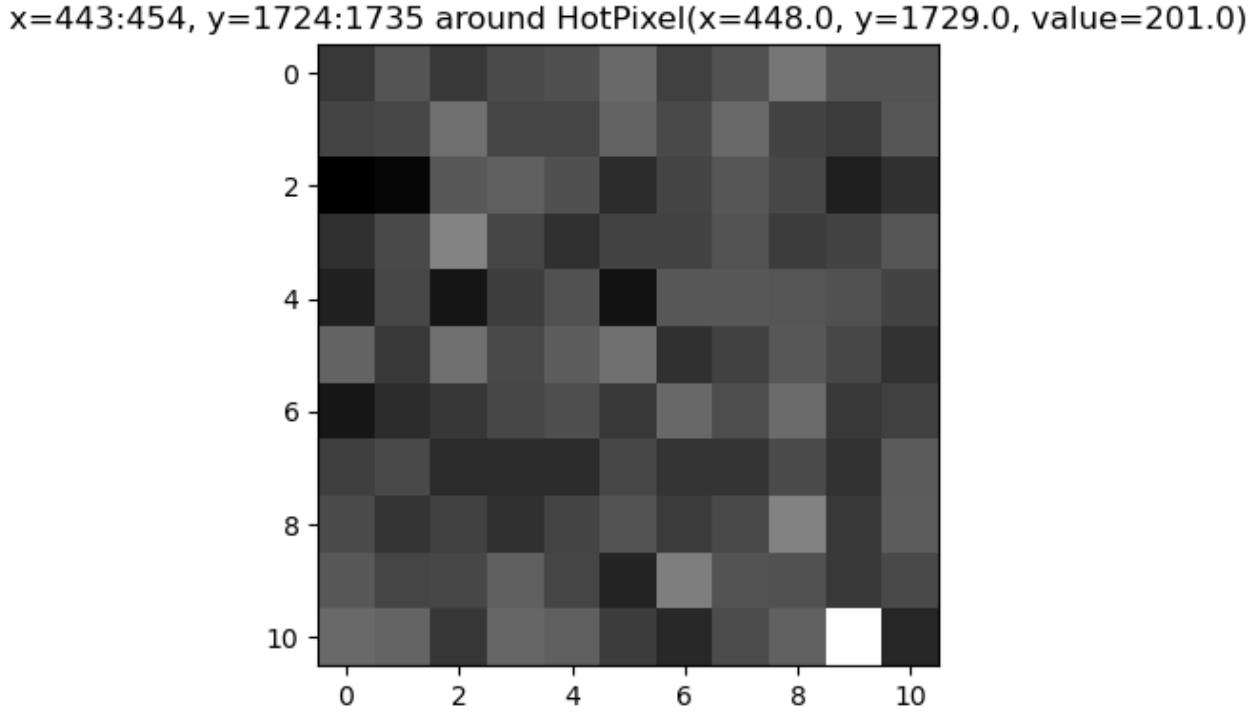
x=458:469, y=1723:1734 around HotPixel(x=463.0, y=1728.0, value=228.0)

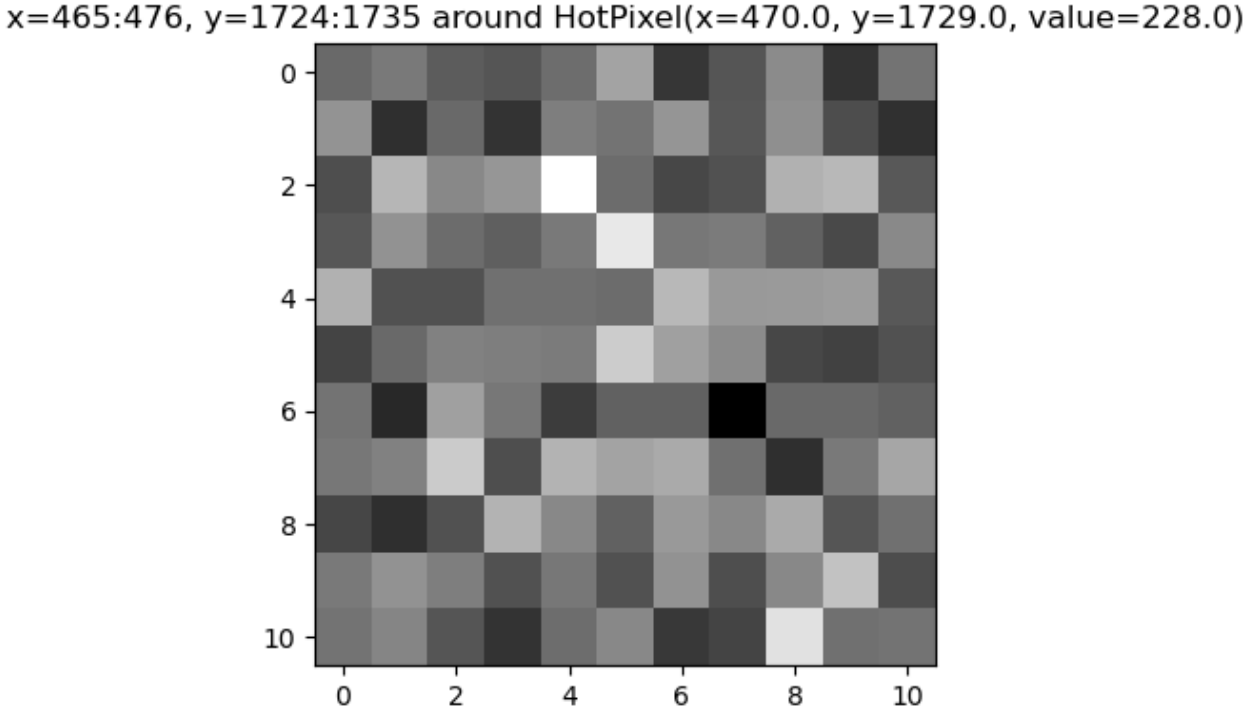


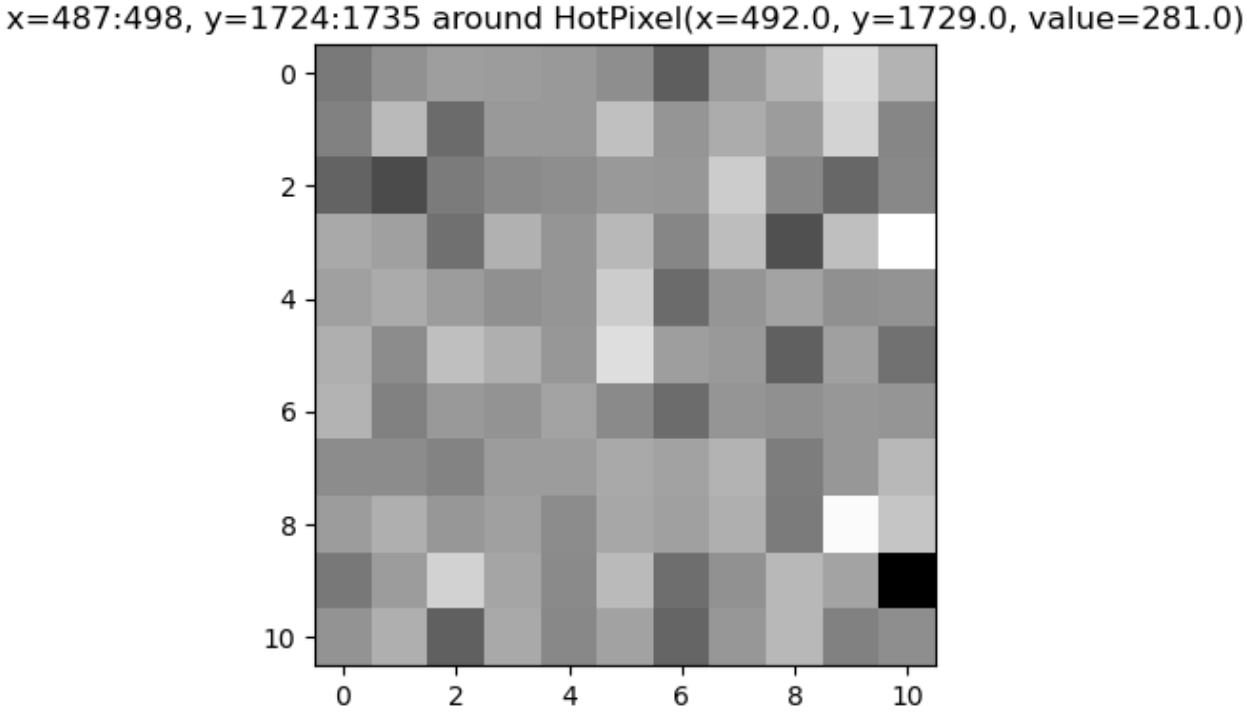
x=395:406, y=1724:1735 around HotPixel(x=400.0, y=1729.0, value=221.0)



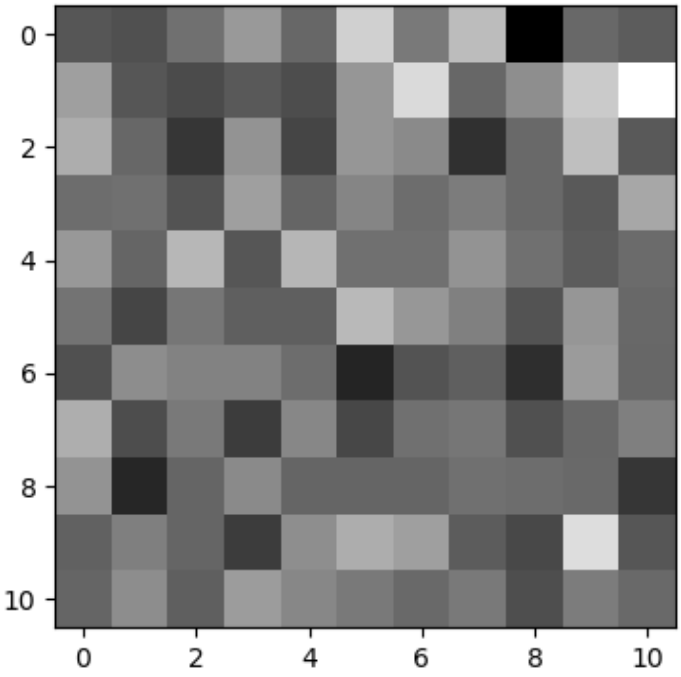

x=440:451, y=1724:1735 around HotPixel(x=445.0, y=1729.0, value=202.0)
0
2
4
6
8
10
0
2
4
6
8
10


x=443:454, y=1724:1735 around HotPixel(x=448.0, y=1729.0, value=201.0)
0
2
4
6
8
10
0
2
4
6
8
10

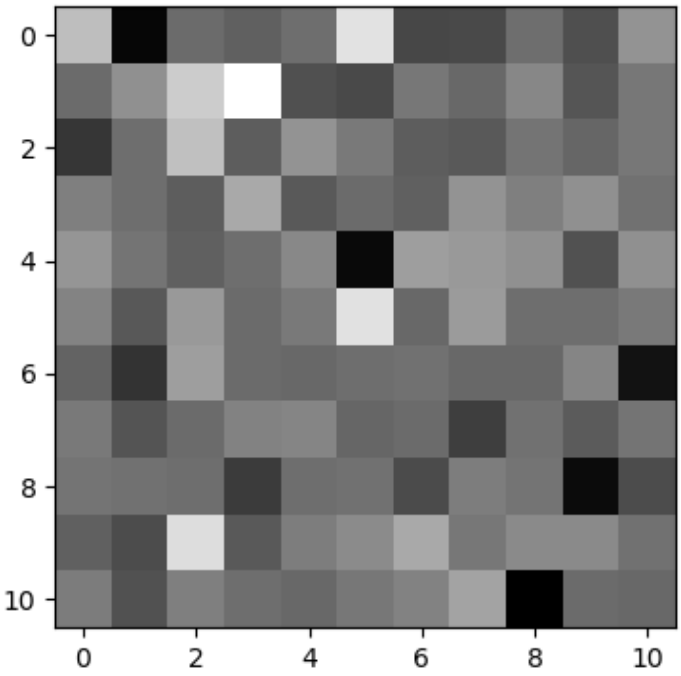

x=465:476, y=1724:1735 around HotPixel(x=470.0, y=1729.0, value=228.0)
0
2
4
6
8
10
0
2
4
6
8
10


x=487:498, y=1724:1735 around HotPixel(x=492.0, y=1729.0, value=281.0)
0
2
4
6
8
10
0
2
4
6
8
10

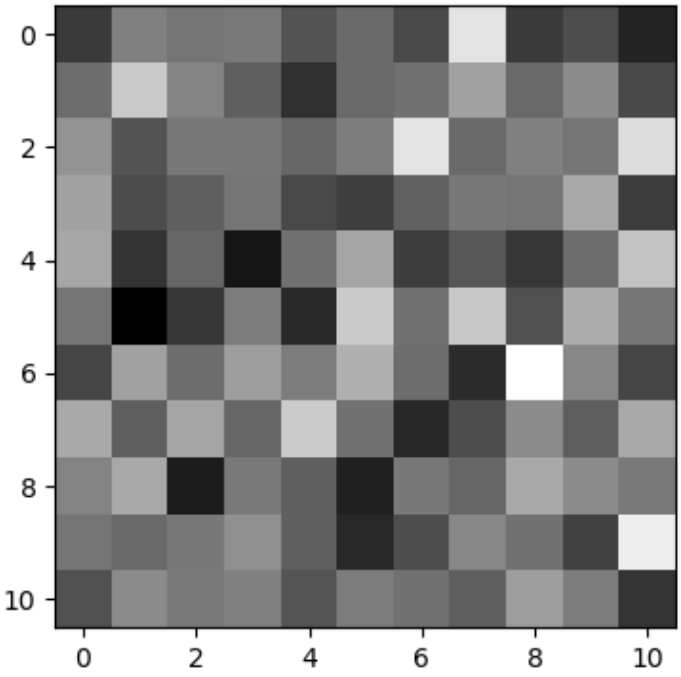
x=398:409, y=1725:1736 around HotPixel(x=403.0, y=1730.0, value=227.5)



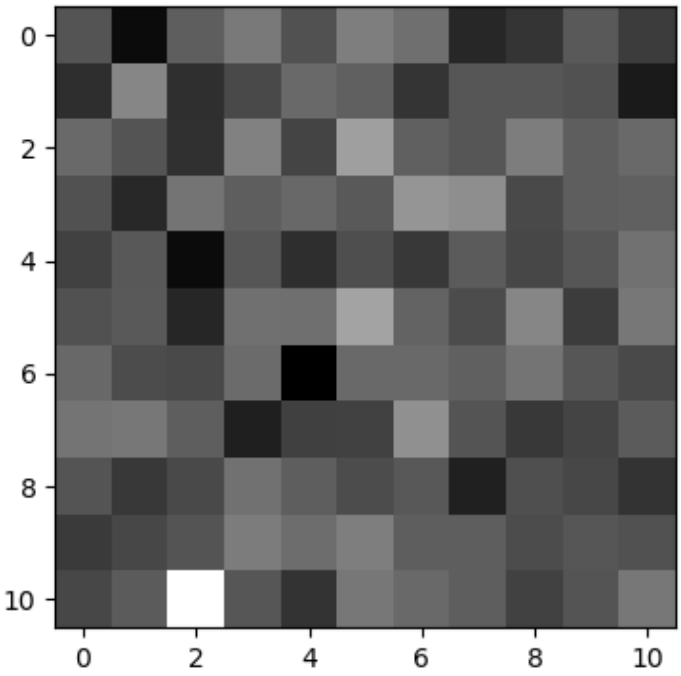
x=405:416, y=1725:1736 around HotPixel(x=410.0, y=1730.0, value=346.0)

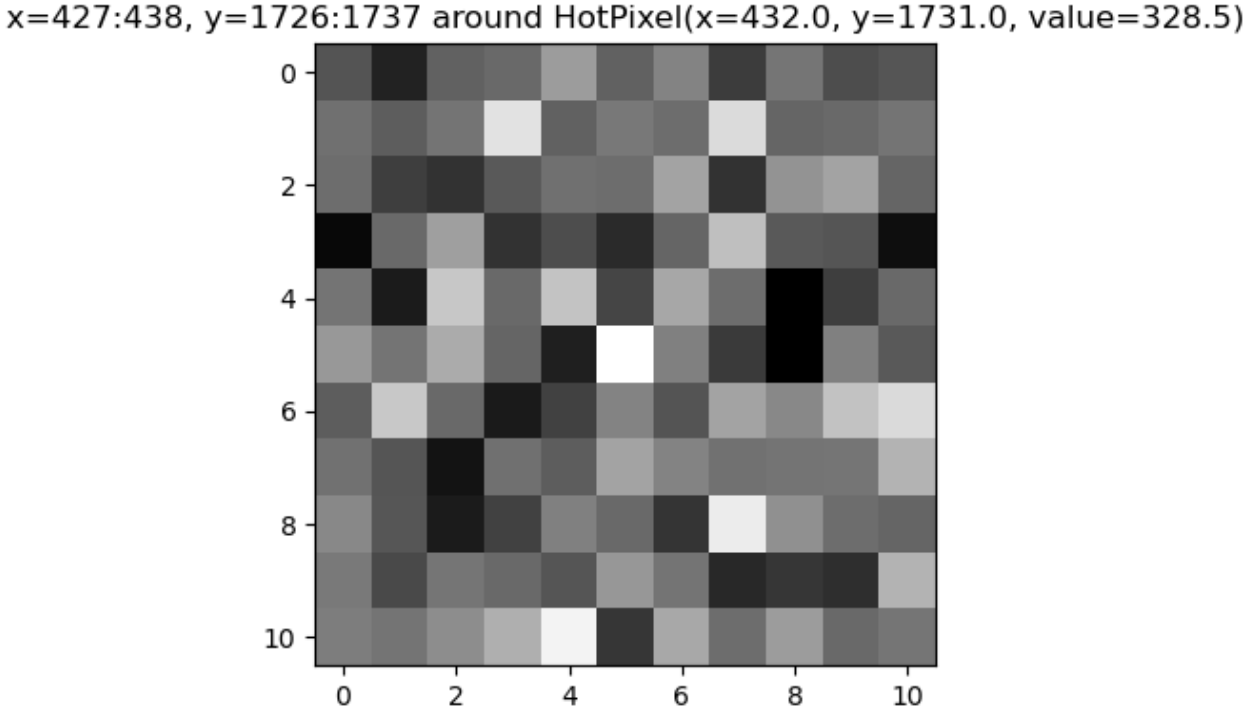


x=424:435, y=1725:1736 around HotPixel(x=429.0, y=1730.0, value=201.5)



x=477:488, y=1725:1736 around HotPixel(x=482.0, y=1730.0, value=270.5)




x=427:438, y=1726:1737 around HotPixel(x=432.0, y=1731.0, value=328.5)
0
2
4
6
8
10
0
2
4
6
8
10

In [ ]: