q2

November 6, 2022

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import cv2
from scipy import signal
from scipy.interpolate import interp2d
from src.cp_hw2 import *
from tqdm import tqdm
```

```
/home/aramesh/anaconda3/envs/comp-photo/lib/python3.10/site-
packages/scipy/__init__.py:146: UserWarning: A NumPy version >=1.16.5 and
<1.23.0 is required for this version of SciPy (detected version 1.23.1
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

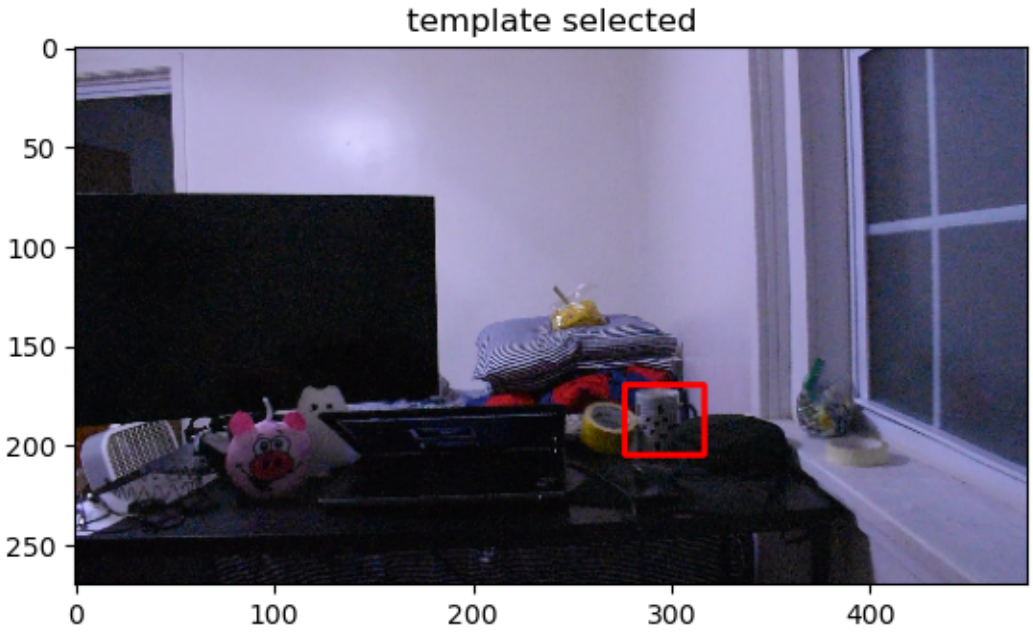
[4]:
```
# read video and select frames
subsample_skip = 4
vc = cv2.VideoCapture('data-dump/DSC_1306.MOV')
frames = []
success, frame = vc.read()
frames.append(frame[::subsample_skip,::subsample_skip,::-1])
it = 0
while True :
    success, frame = vc.read()
    if success is False : break
    if it % 3 == 0: frames.append(frame[::subsample_skip,::subsample_skip,::-1])
    it += 1
frames = np.array(frames)
frames = frames.transpose(1,2,3,0)
```

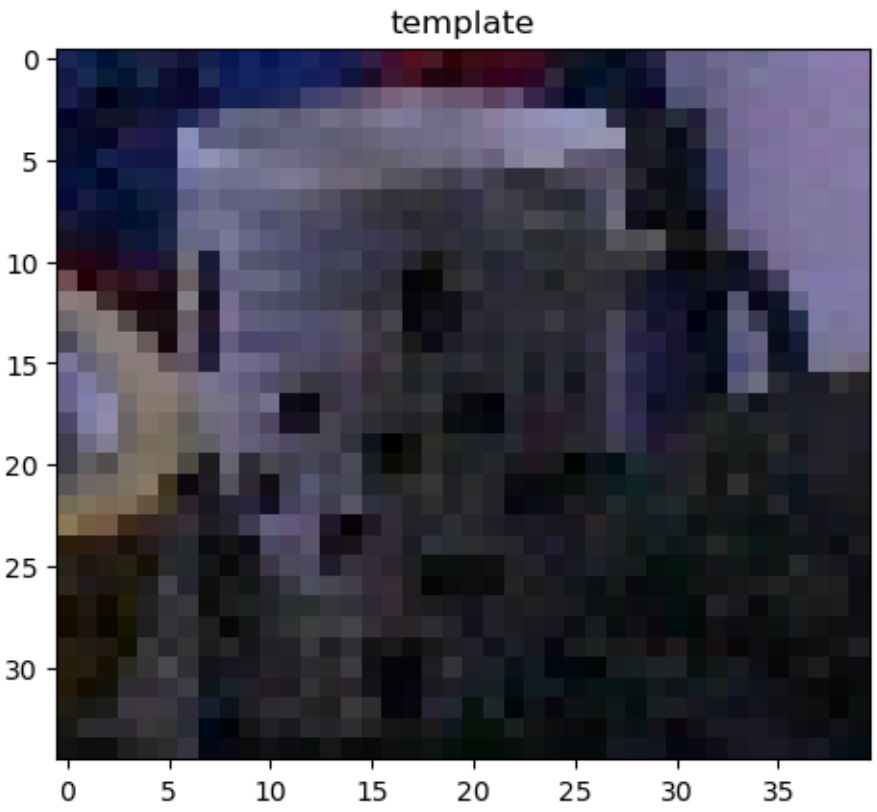
[5]:
```
# Get luminance channel for template matching
@np.vectorize
def linearize_image(C_nonlinear) :

    if C_nonlinear <= 0.0404482 :
        return C_nonlinear / 12.92
    else :
        out_num = ( C_nonlinear + 0.055 ) ** 2.4
        out_den = 1.055 ** 2.4
```

```
        return out_num / out_den

def get_luminance(I) :
    print(I.shape)
    if len(I.shape) == 3 : I = np.expand_dims(I,3)
    I_linear = linearize_image(I)
    I_XYZ = np.zeros(I_linear.shape)
    for foc_im in range(I_linear.shape[-1]) :
        I_XYZ[:,:,:,foc_im] = lRGB2XYZ(I_linear[:,:,:,foc_im])
    I_luminance = I_XYZ[:,:,1,:] # take just Y channel for relative luminance
    return I_luminance
```

[20]:
```
# template selection in the middle frame of video

# DSC 1298
# pink pig subsample frame_id % 10 == 0 (31 frames)
# bbox= (np.array([340,130,440,230])// ( subsample_skip / 2 )).astype('int') #␣
 ↪[[row1,col1,row2,col2]] sp, ep = (bbox[1],bbox[0]) , (bbox[3], bbox[2])

# pink pig subsample frame_id % 1 == 0 (297 frames) or %2 (149 frames)
# bbox= (np.array([330,135,430,235])// ( subsample_skip / 2 )).astype('int') #␣
 ↪[[row1,col1,row2,col2]] sp, ep = (bbox[1],bbox[0]) , (bbox[3], bbox[2])

# DSC 1306
# pink pig subsample frame_id % 3 == 0 (102 frames)
# bbox= (np.array([360,145,460,245])// ( subsample_skip / 2 )).astype('int') #␣
 ↪[[row1,col1,row2,col2]] sp, ep = (bbox[1],bbox[0]) , (bbox[3], bbox[2])

# bag of banana chips in the background (102 frames, subsample by % 3)
# bbox= (np.array([230,465,310,545])// ( subsample_skip / 2 )).astype('int') #␣
 ↪[[row1,col1,row2,col2]] sp, ep = (bbox[1],bbox[0]) , (bbox[3], bbox[2])

# coffee mug subsample frame_id % 3 == 0 (102 frames)
bbox= (np.array([340,555,410,635])// ( subsample_skip / 2 )).astype('int') #␣
 ↪[[row1,col1,row2,col2]] sp, ep = (bbox[1],bbox[0]) , (bbox[3], bbox[2])


sp, ep = (bbox[1],bbox[0]) , (bbox[3], bbox[2])
mid_im = frames[:,:,:,frames.shape[3]//2].copy()
mid_im_disp = mid_im.copy()
mid_im_disp = cv2.rectangle(mid_im_disp,sp,ep,(255,0,0),2)
plt.imshow(mid_im_disp)
plt.title('template selected')
plt.show()

template = mid_im[bbox[0] :  bbox[2] , bbox[1] : bbox[3]]
plt.imshow(template)
```

```
plt.title('template')
plt.show()

# template = np.mean(template,2)
# mid_im = np.mean(mid_im,2)
template = get_luminance(template)
mid_im = get_luminance(mid_im)
```

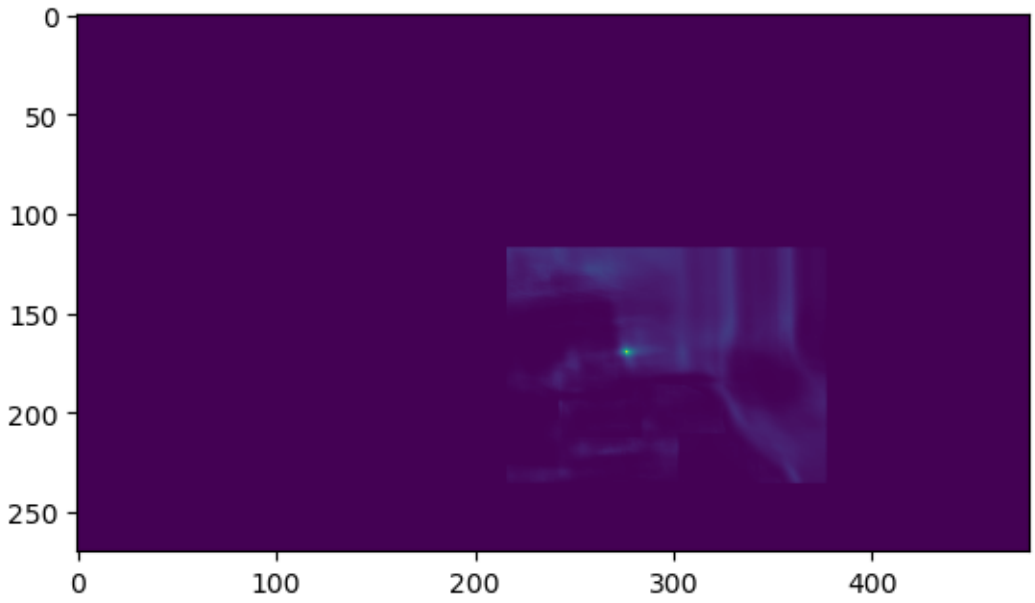


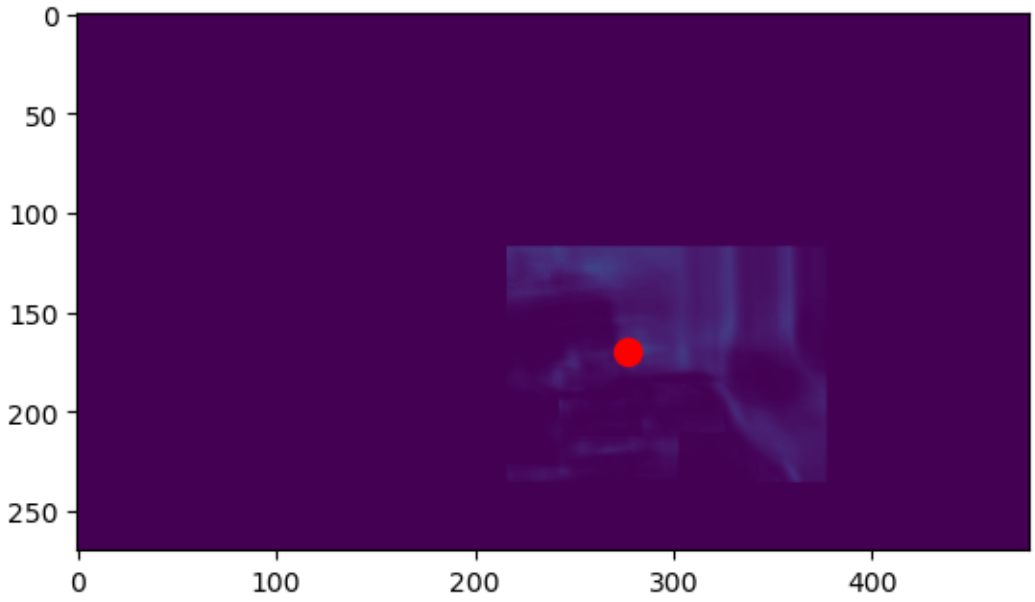



```
(35, 40, 3)
(270, 480, 3)
```

[21]:
```python
# Testing to see if our normxcorr function works

def get_normxcorr(img, template, bbox) :
    d1 = np.sum((template - template.mean())**2)
    normxcorr = np.zeros(img.shape)

    bbox_center = [(bbox[0] + bbox[2])/2, (bbox[1]+bbox[3])/2]
    bbox_size = np.array(template.shape) * 2
    count = 0

    for i in range(img.shape[0]) :

        if np.abs(i - bbox_center[0]) > bbox_size[0] : continue

        for j in range(img.shape[1]) :

            if np.abs(j - bbox_center[1]) > bbox_size[1] : continue
```

```
            if img[i:i+template.shape[0], j:j+template.shape[1]].shape !=␣
 ↪template.shape : continue

            count += 1
            num = np.sum(img[i:i+template.shape[0], j:j+template.shape[1]] *␣
 ↪template)
            I_box = img[i:i+template.shape[0], j:j+template.shape[1]] / img[i:
 ↪i+template.shape[0], j:j+template.shape[1]].sum()
            d2 = np.sum((img[i:i+template.shape[0], j:j+template.shape[1]] -␣
 ↪I_box) ** 2)
            normxcorr[i,j] = num / np.sqrt(d1*d2)

    return normxcorr

def calculate_shift(normxcorr, bbox) :

    y, x = np.unravel_index(np.argmax(normxcorr), normxcorr.shape)

    y_mid_im, x_mid_im = bbox[0], bbox[1]

    shift_x = x - x_mid_im # col shift
    shift_y = y - y_mid_im # row shift

    return shift_x, shift_y

def display_correspondance(normxcorr, frame=None) :
    y, x = np.unravel_index(np.argmax(normxcorr), normxcorr.shape)
    plt.imshow(normxcorr**4)
    plt.show()
    plt.imshow(normxcorr**4)
    plt.scatter(x,y,100,'r')
    plt.show()
    if frame is not None :
        plt.imshow(frame)
        plt.scatter(x,y,100,'r')
        plt.show()

normxcorr = get_normxcorr(mid_im[:,:,0], template[:,:,0], bbox)
display_correspondance(normxcorr)

print('giving middle image and template, we should get shift equal to (0,0)')
print('calculated shift is ', calculate_shift(normxcorr, bbox))
```





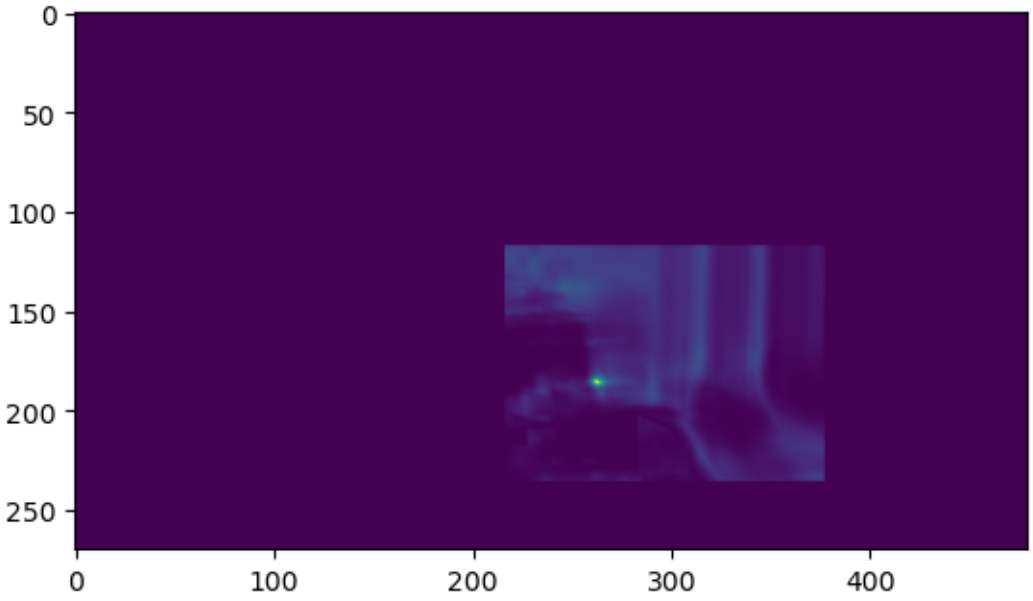

```
giving middle image and template, we should get shift equal to (0,0)
calculated shift is  (0, 0)
```

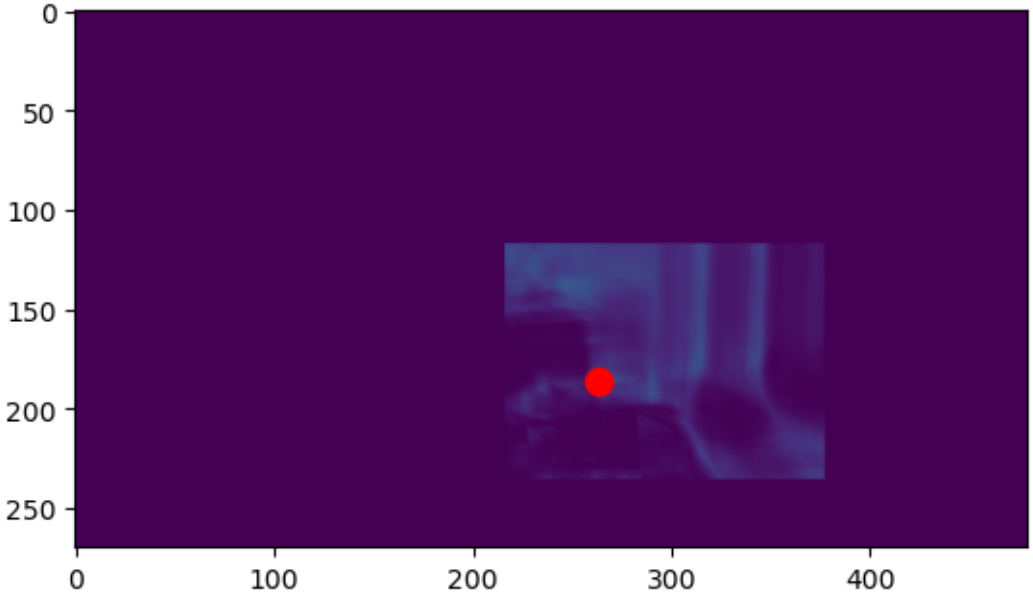
[22]:
```
# frame interpolator
L_interp = []
for i in range(frames.shape[3]) :
```

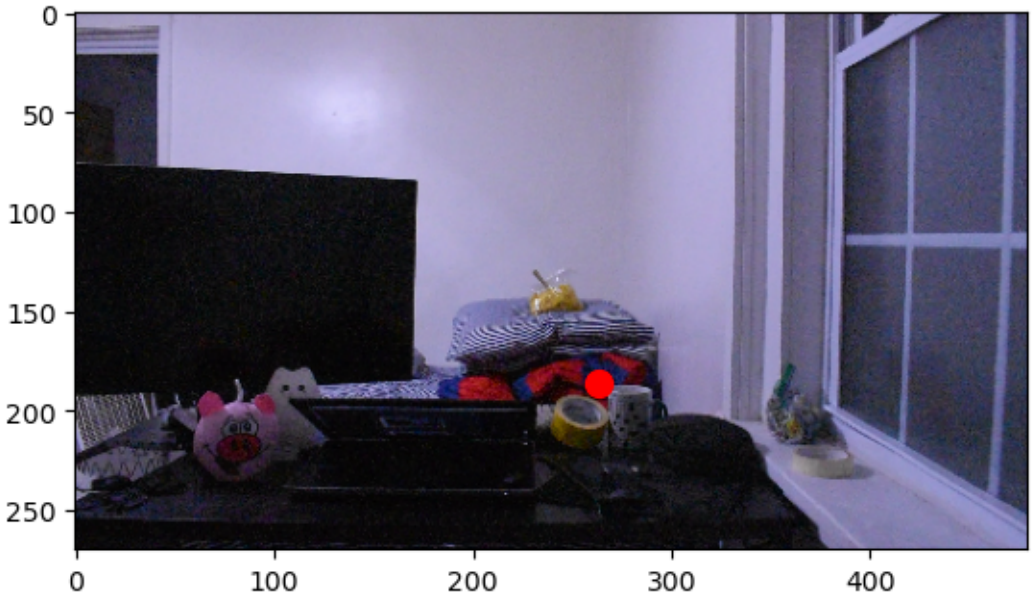
```
        temp = []
        temp.append(interp2d(np.arange(frames.shape[1]),np.arange(frames.
 ↪shape[0]),frames[:,:,0,i]))
        temp.append(interp2d(np.arange(frames.shape[1]),np.arange(frames.
 ↪shape[0]),frames[:,:,1,i]))
        temp.append(interp2d(np.arange(frames.shape[1]),np.arange(frames.
 ↪shape[0]),frames[:,:,2,i]))
        L_interp.append(temp)
```

[23]:
```
shifted_frames = np.zeros(frames.shape)
shift_array = []
frames_luminance = get_luminance(frames)
combined_frame = np.zeros(frames[:,:,:,0].shape)

# calculate shift
print('calculating shift')
for fi in tqdm(range(frames.shape[3])) :
    normxcorr = get_normxcorr(frames_luminance[:,:,fi], template[:,:,0], bbox)
    display_correspondance(normxcorr,frames[:,:,:,fi])
    shift_array.append(calculate_shift(normxcorr, bbox))

print('shifting frames')
# shift frames
for fi in tqdm(range(frames.shape[3])) :
    temp = np.zeros(frames[:,:,:,0].shape)
    cur_fun = L_interp[fi]
    temp[:,:,0] = cur_fun[0](np.arange(frames.shape[1])+shift_array[fi][0], np.
 ↪arange(frames.shape[0])+shift_array[fi][1])
    temp[:,:,1] = cur_fun[1](np.arange(frames.shape[1])+shift_array[fi][0], np.
 ↪arange(frames.shape[0])+shift_array[fi][1])
    temp[:,:,2] = cur_fun[2](np.arange(frames.shape[1])+shift_array[fi][0], np.
 ↪arange(frames.shape[0])+shift_array[fi][1])
    shifted_frames[:,:,:,fi] = temp.copy()

    combined_frame += temp.copy() / frames.shape[3]
```

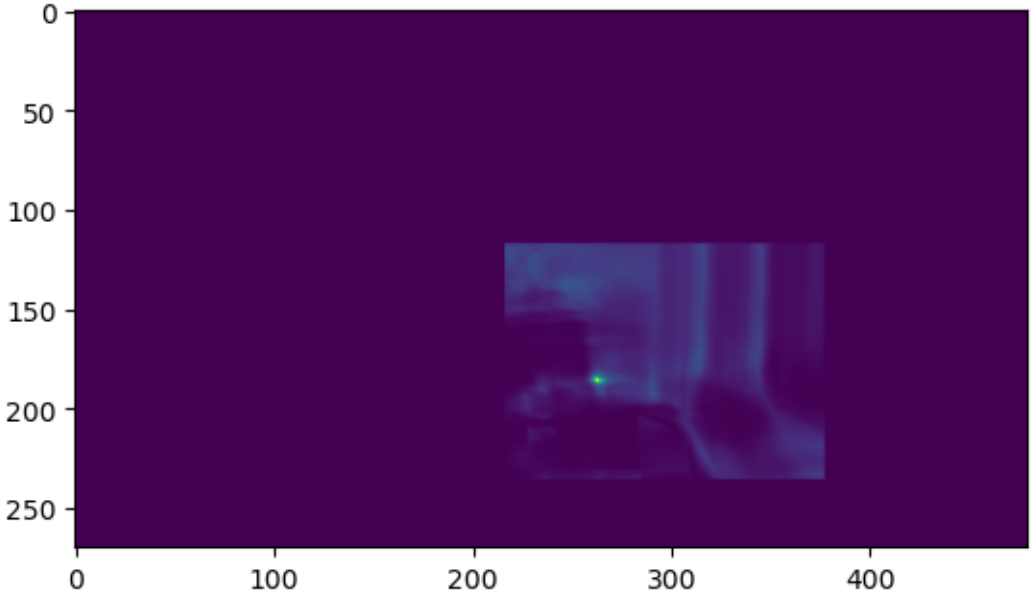
```
(270, 480, 3, 102)
calculating shift
  0%|                                                | 0/102 [00:00<?, ?it/s]
```

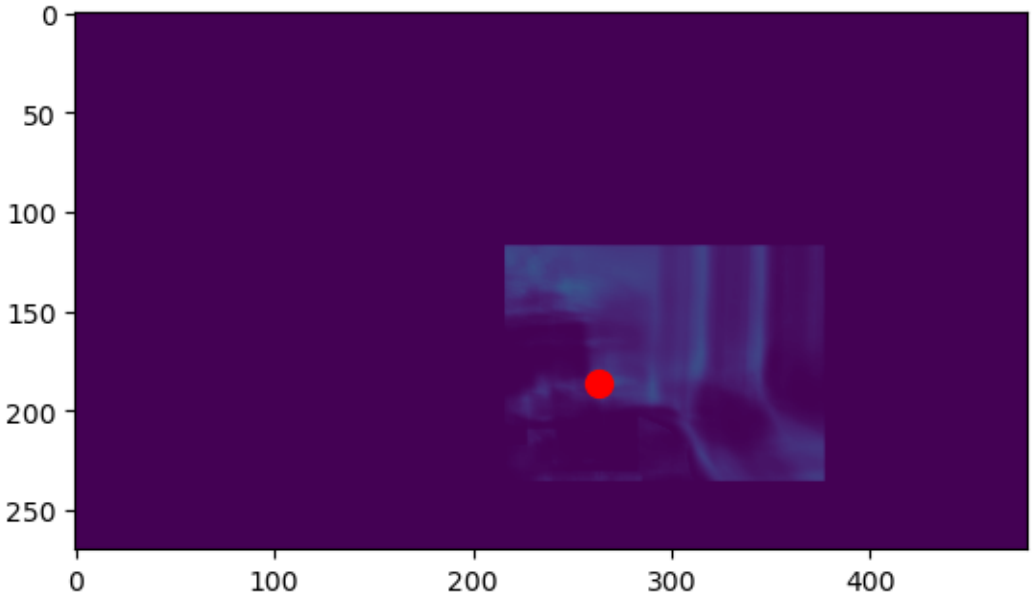

0
50
100
150
200
250
0
100
200
300
400

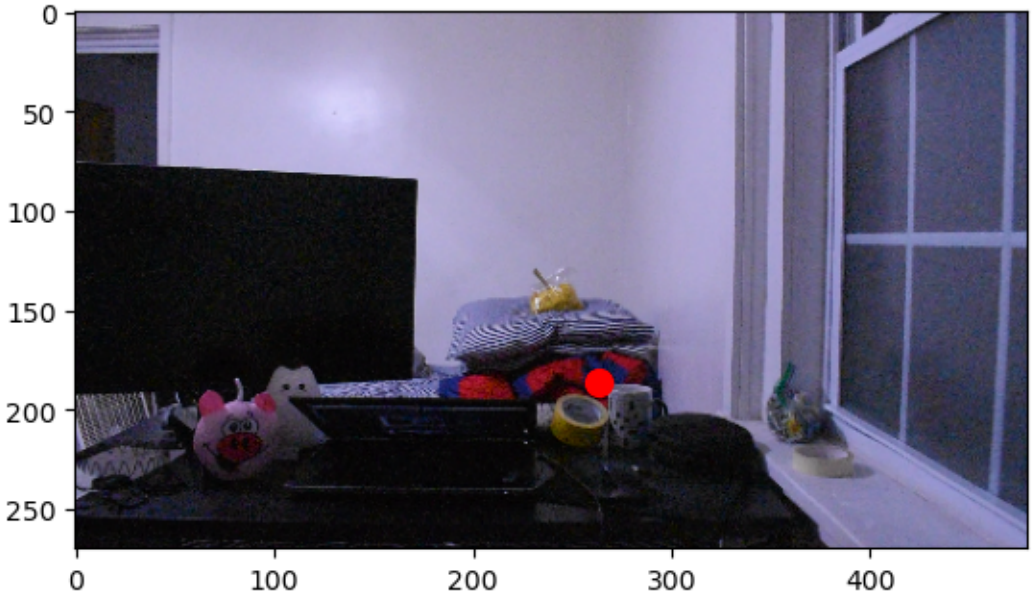

0
50
100
150
200
250
0
100
200
300
400



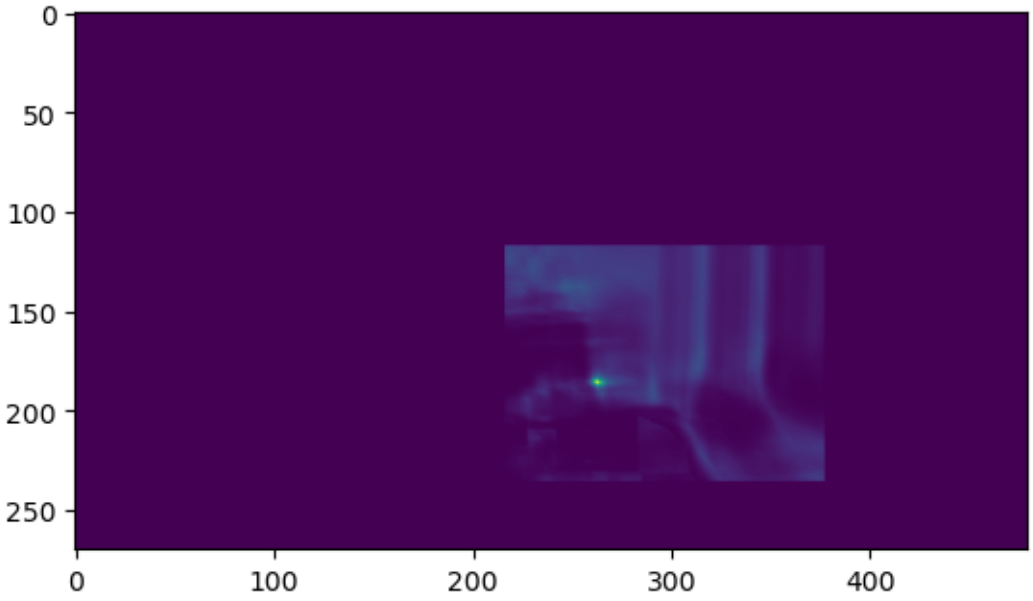
1%| | 1/102 [00:01<02:03, 1.22s/it]

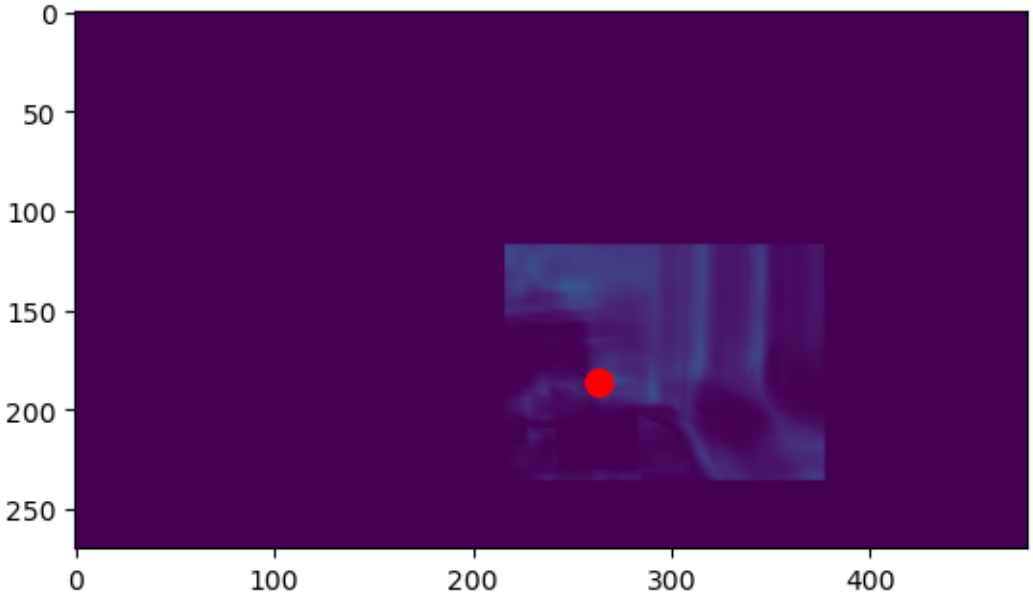


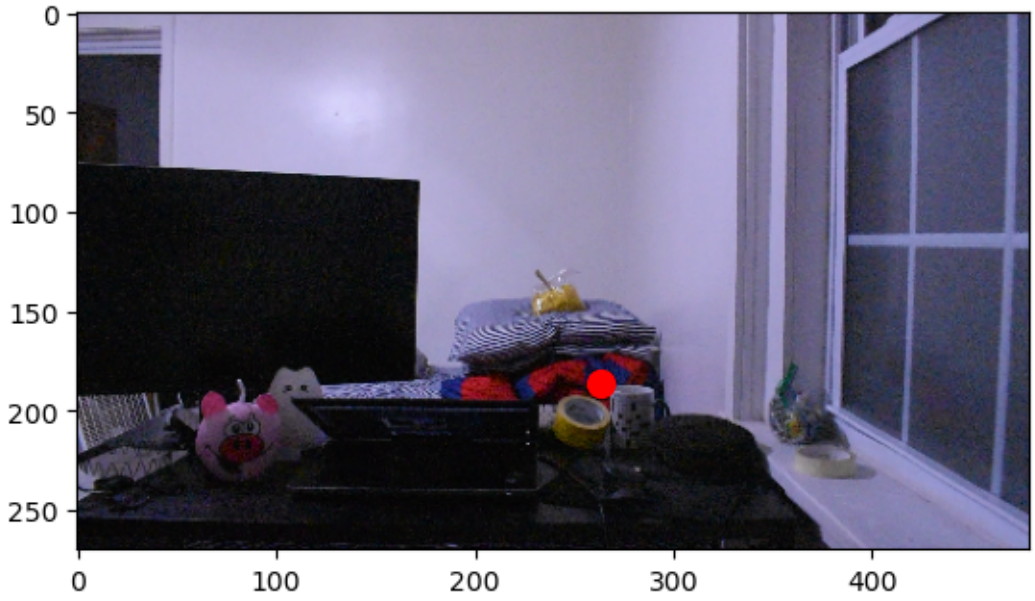




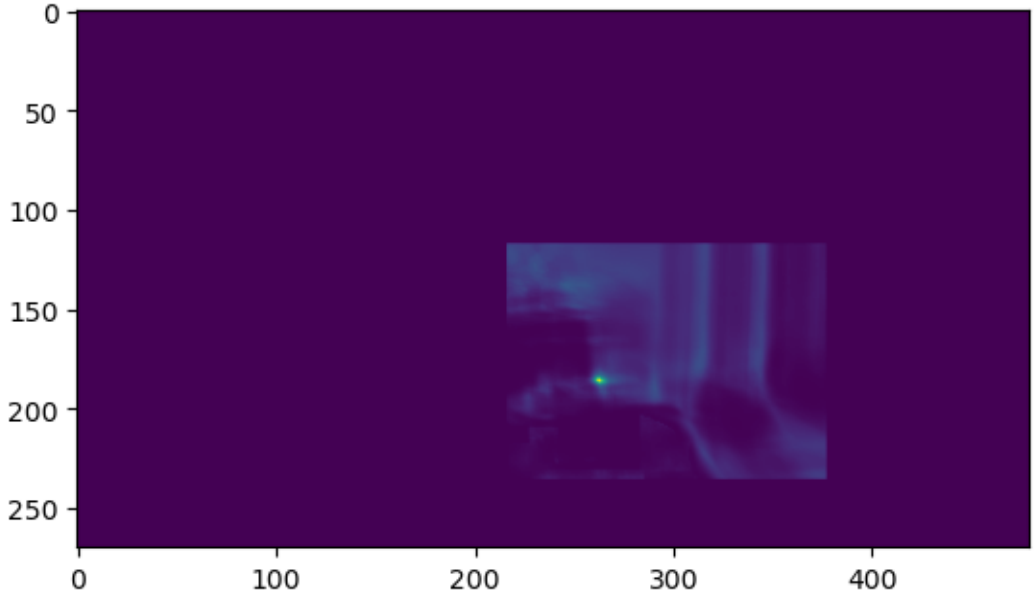
2%| | 2/102 [00:02<02:02, 1.22s/it]

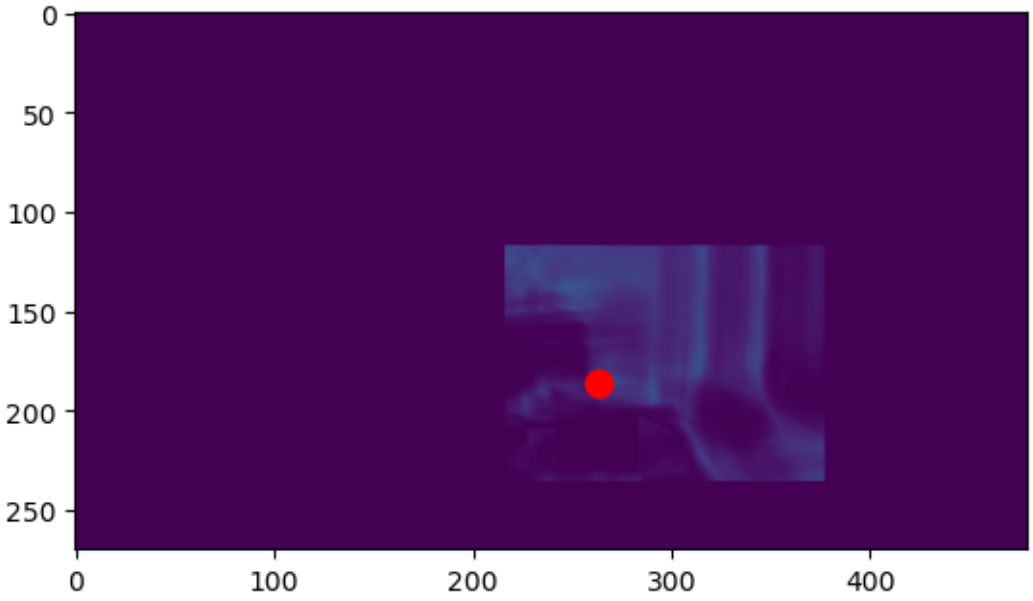

0
50
100
150
200
250
0
100
200
300
400

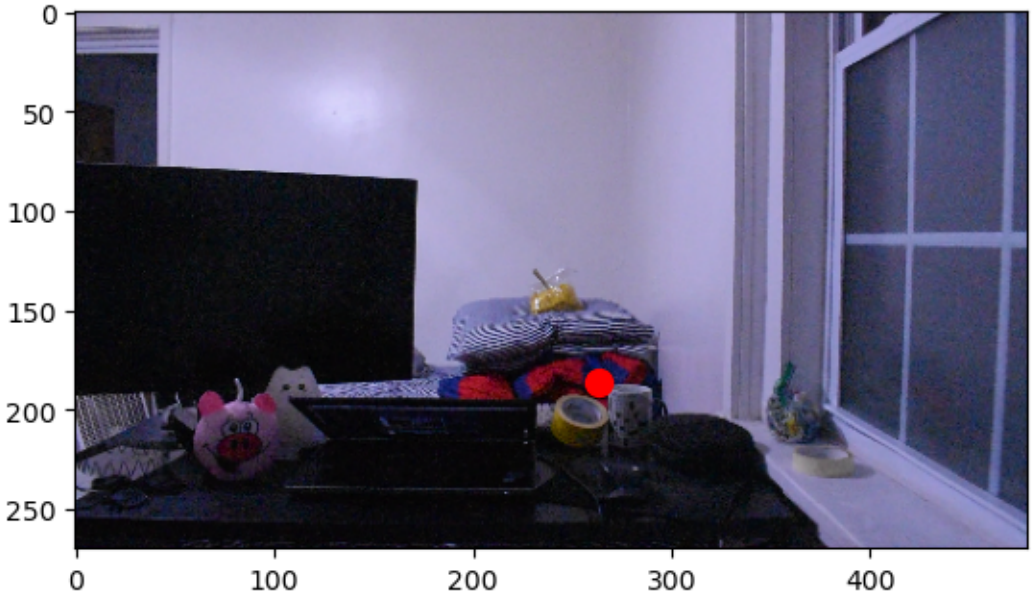

0
50
100
150
200
250
0
100
200
300
400



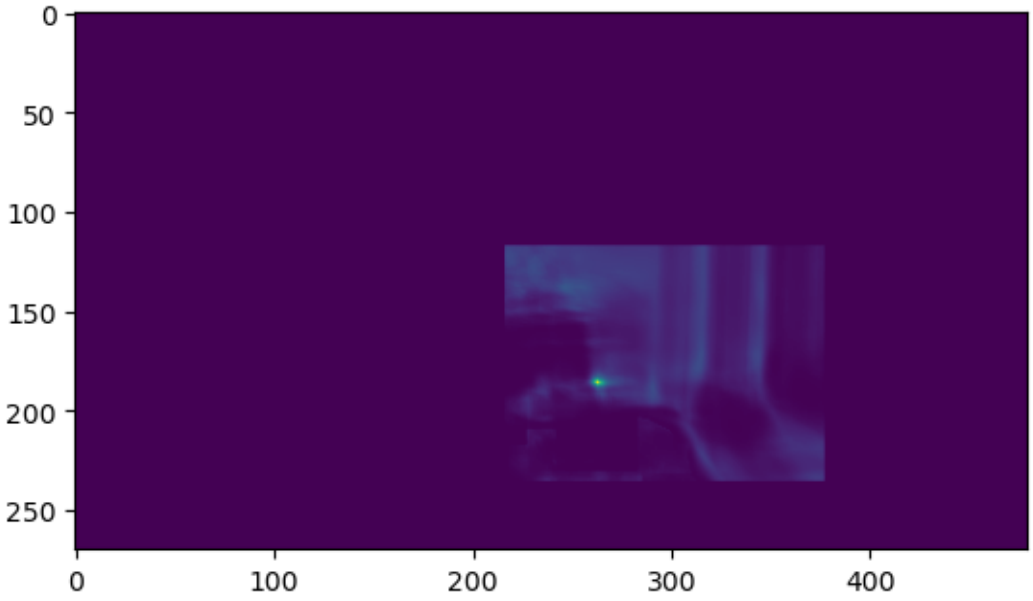
3%| | 3/102 [00:03<02:00, 1.22s/it]

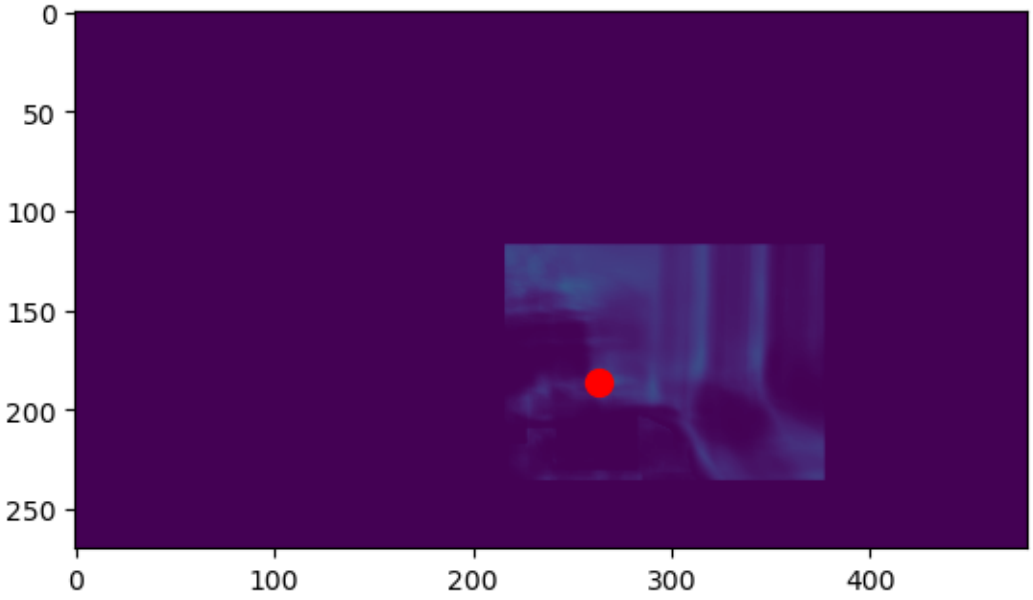


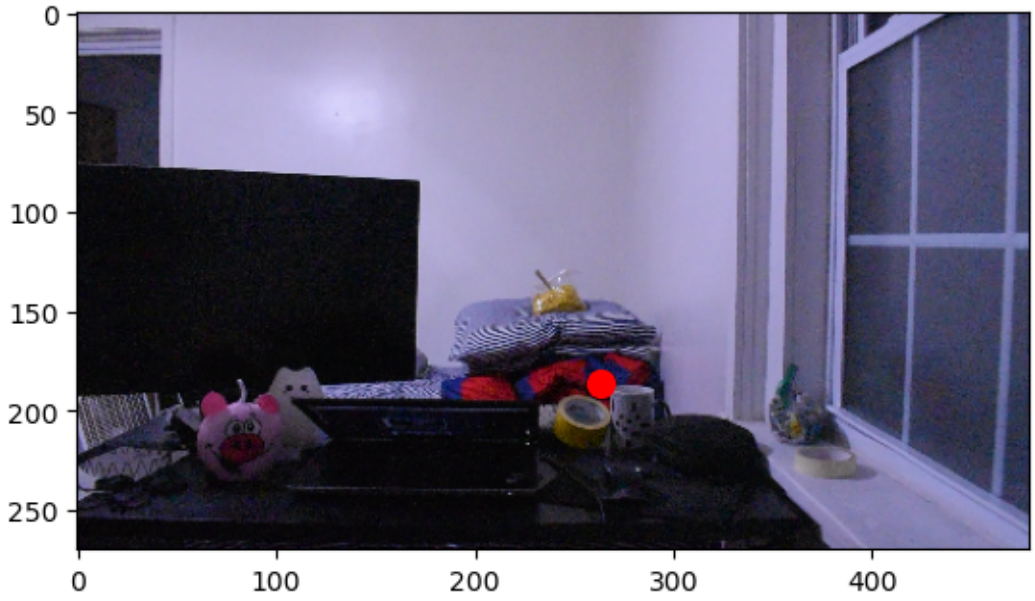




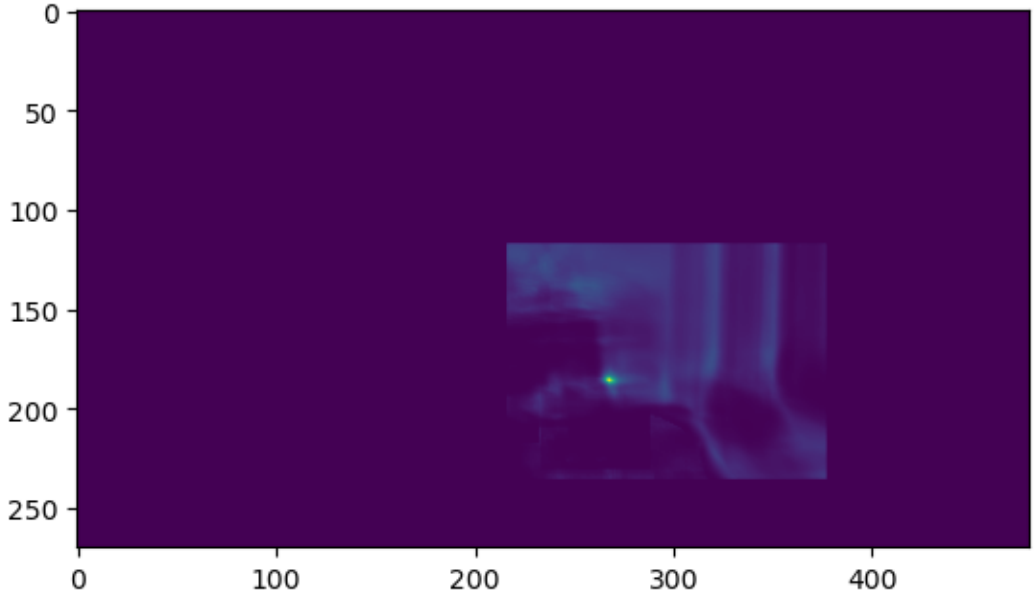
4%| | 4/102 [00:04<01:58, 1.21s/it]

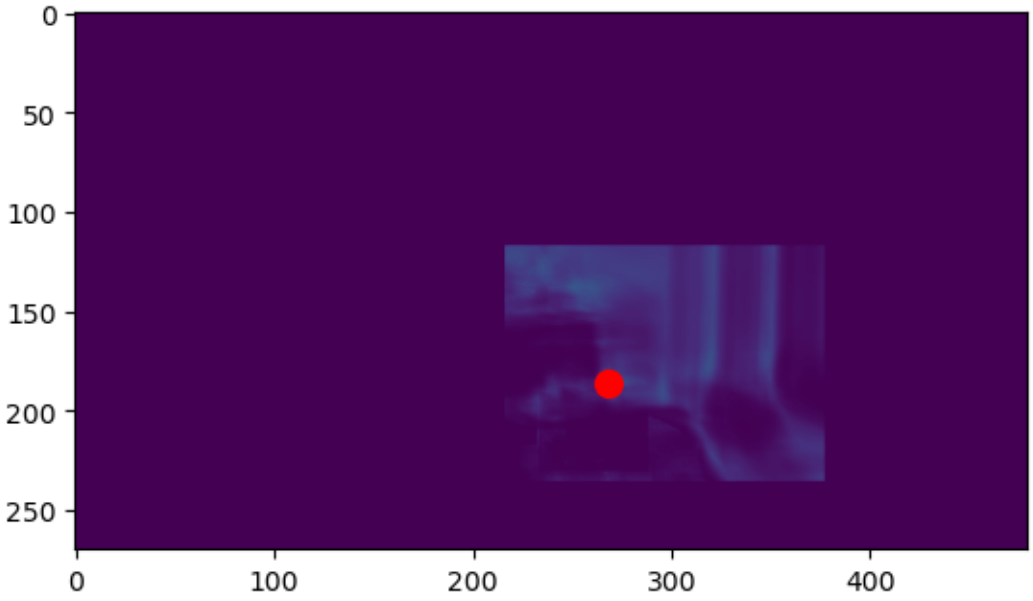

0
50
100
150
200
250
0
100
200
300
400

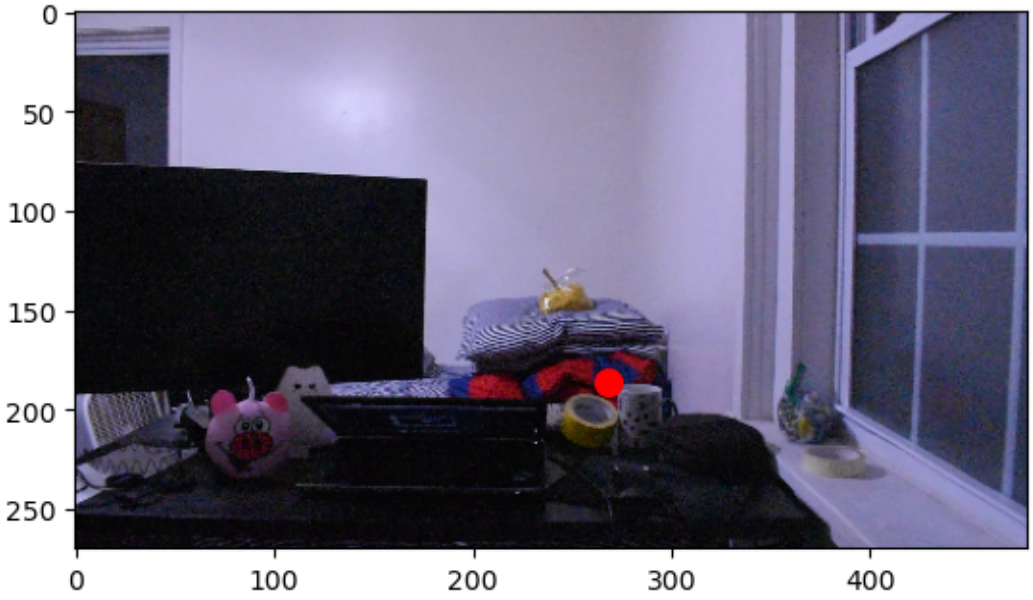

0
50
100
150
200
250
0
100
200
300
400



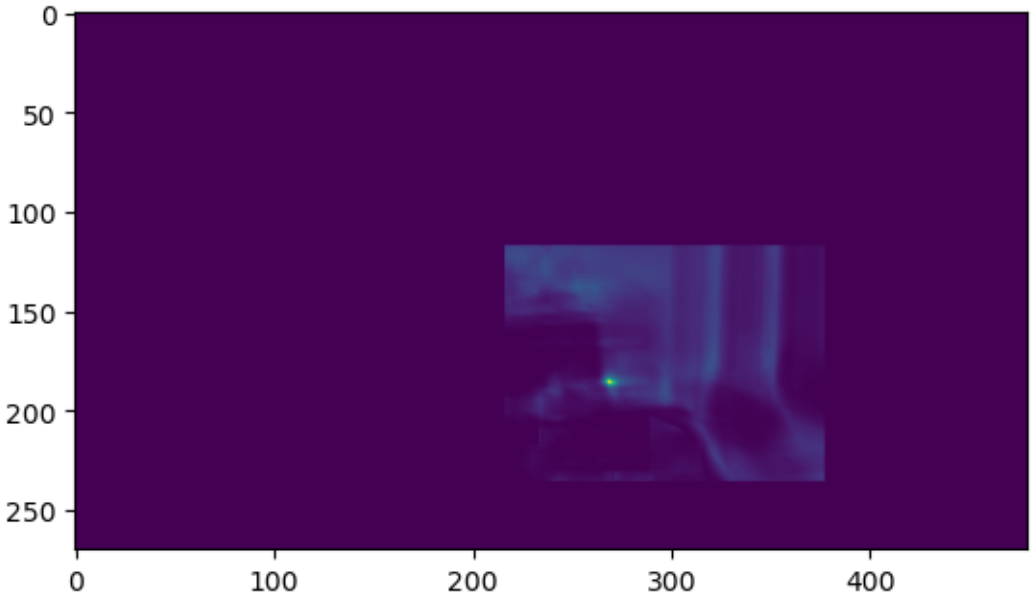
5%| | 5/102 [00:06<01:56, 1.20s/it]

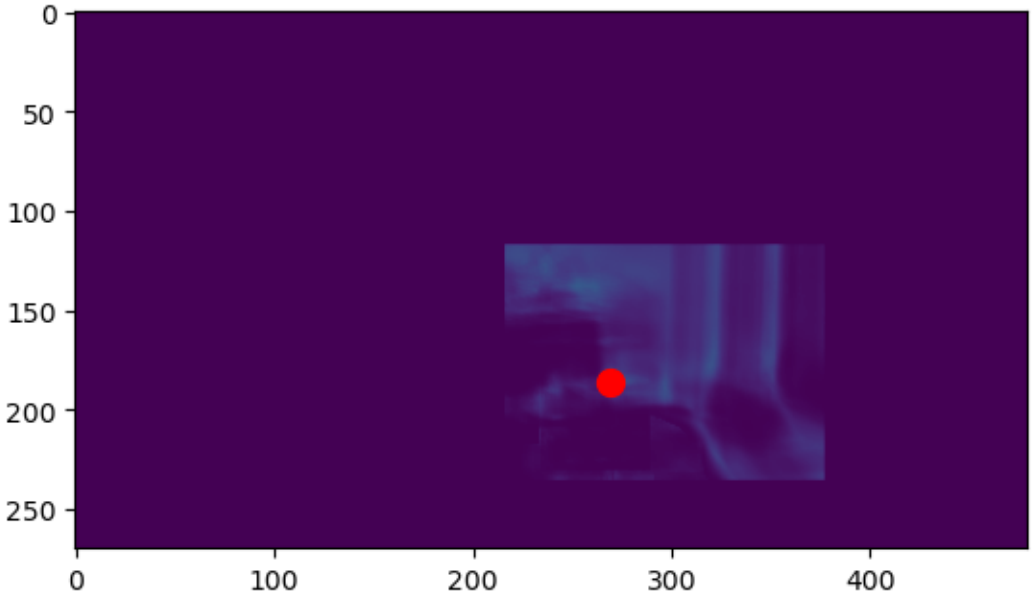


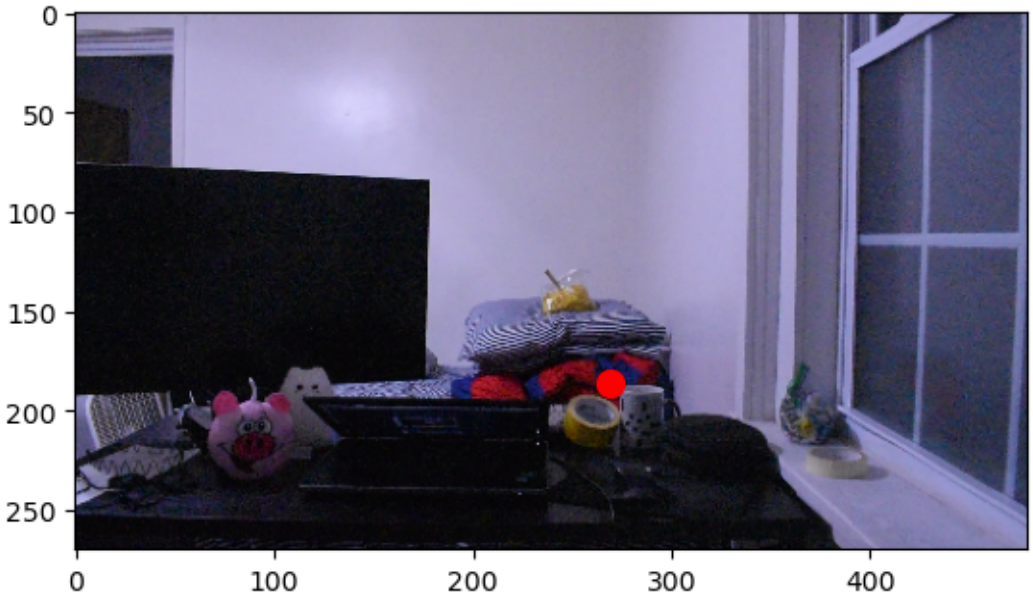




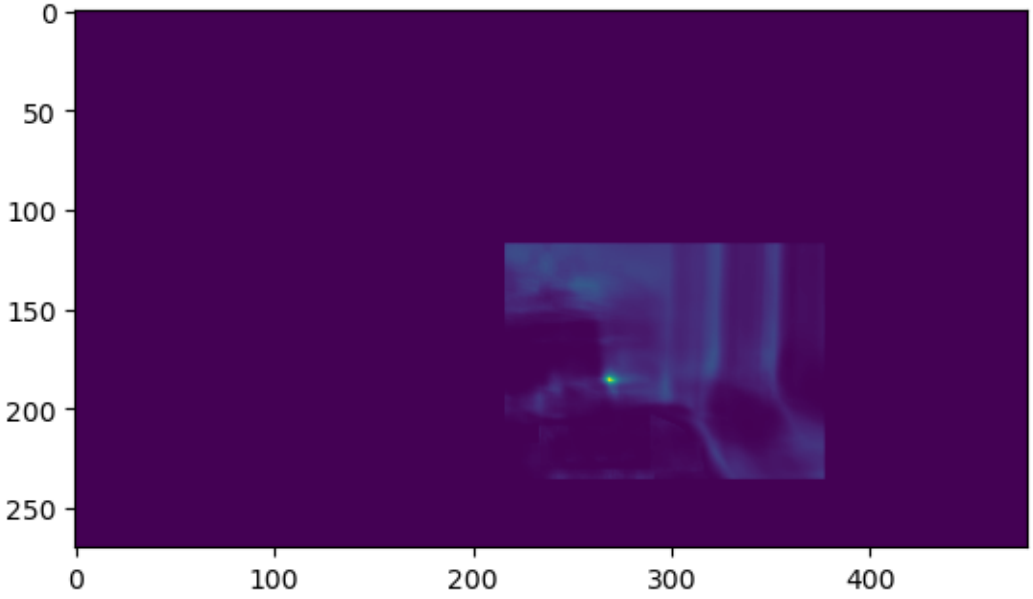
6%| | 6/102 [00:07<01:55, 1.21s/it]

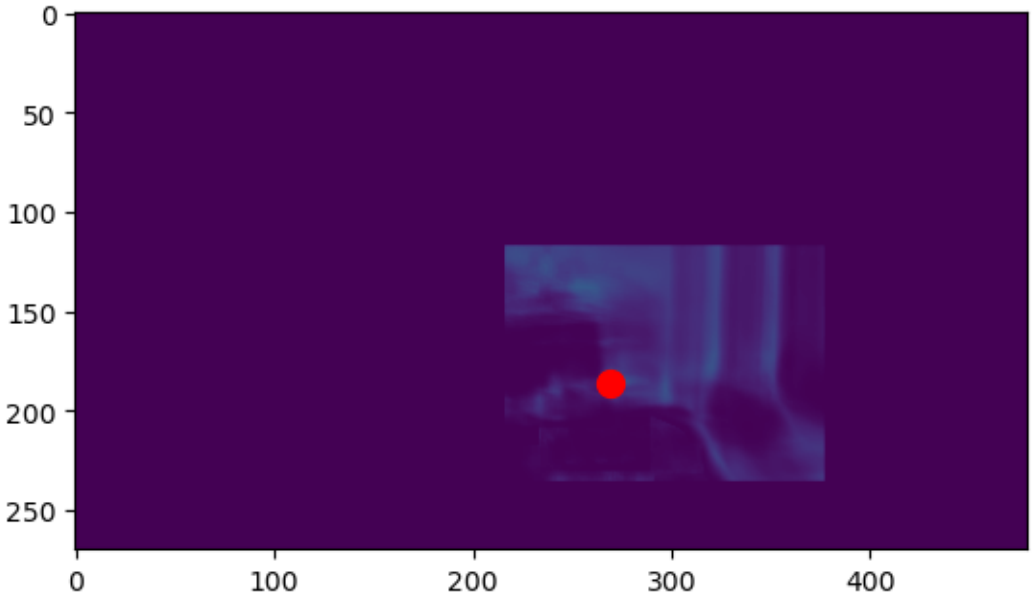

0
50
100
150
200
250
0
100
200
300
400

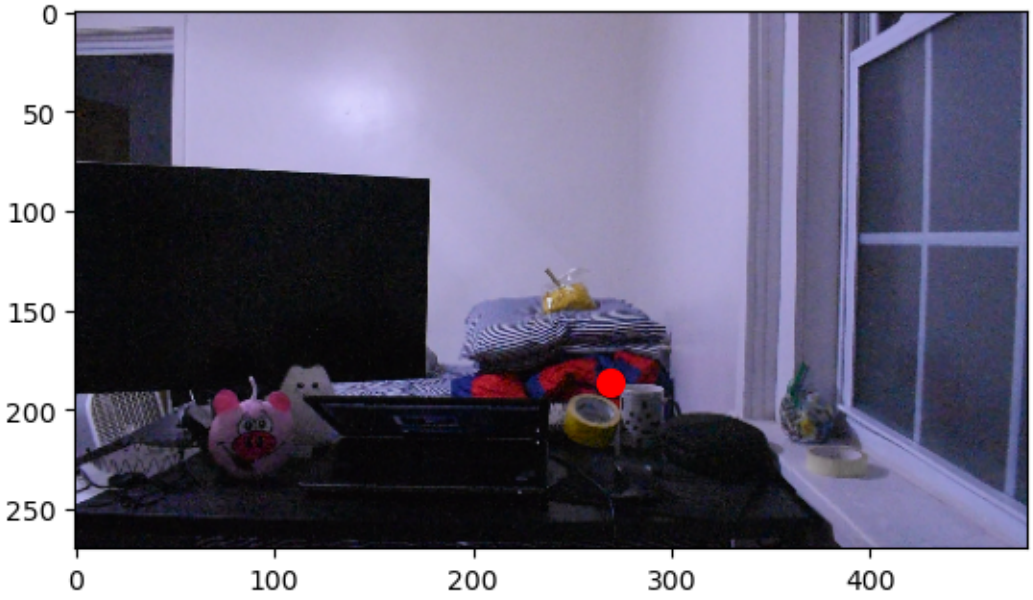

0
50
100
150
200
250
0
100
200
300
400



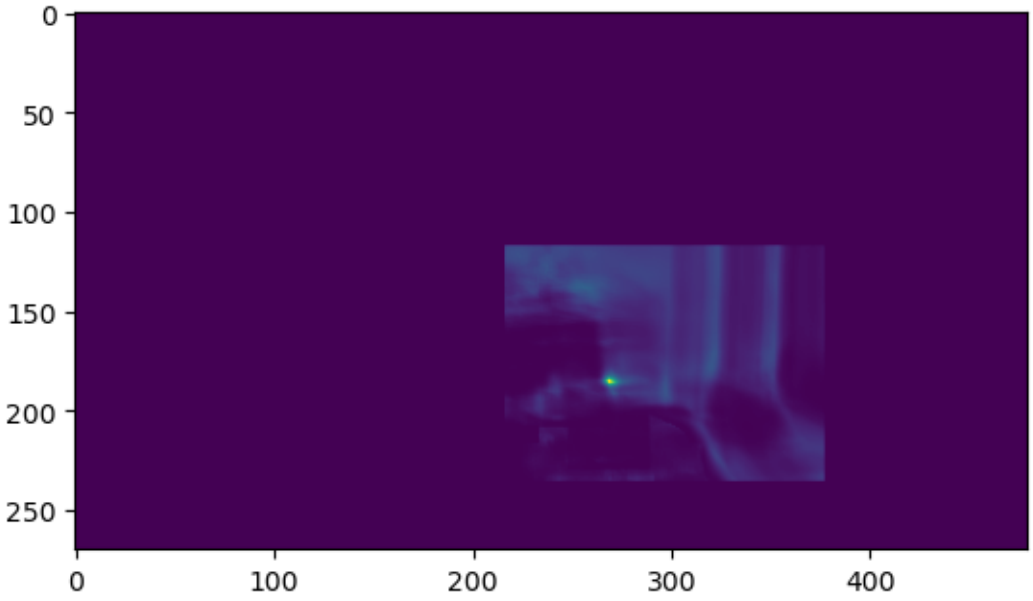
7%| | 7/102 [00:08<01:58, 1.25s/it]

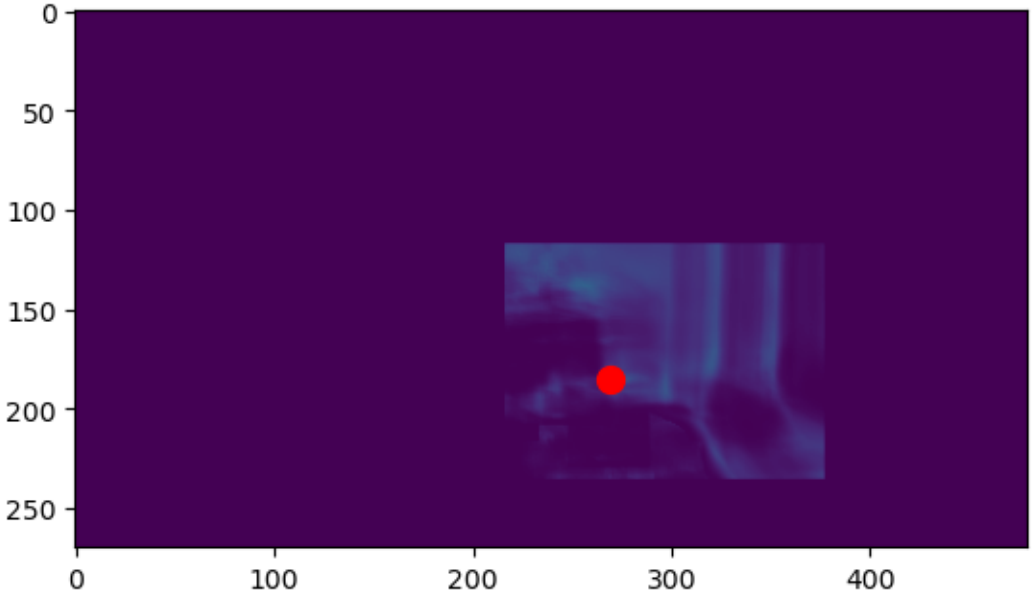


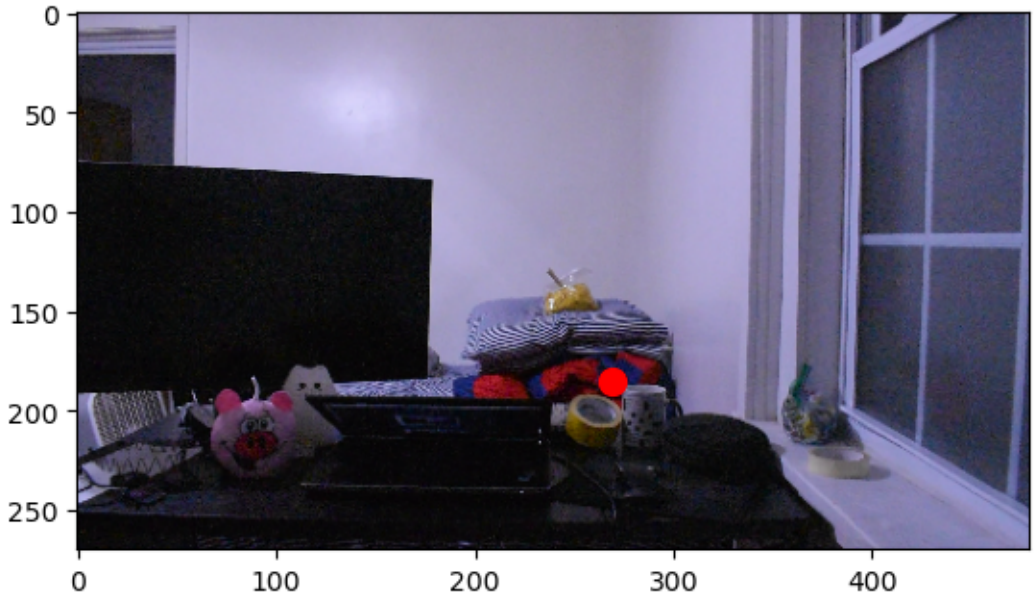




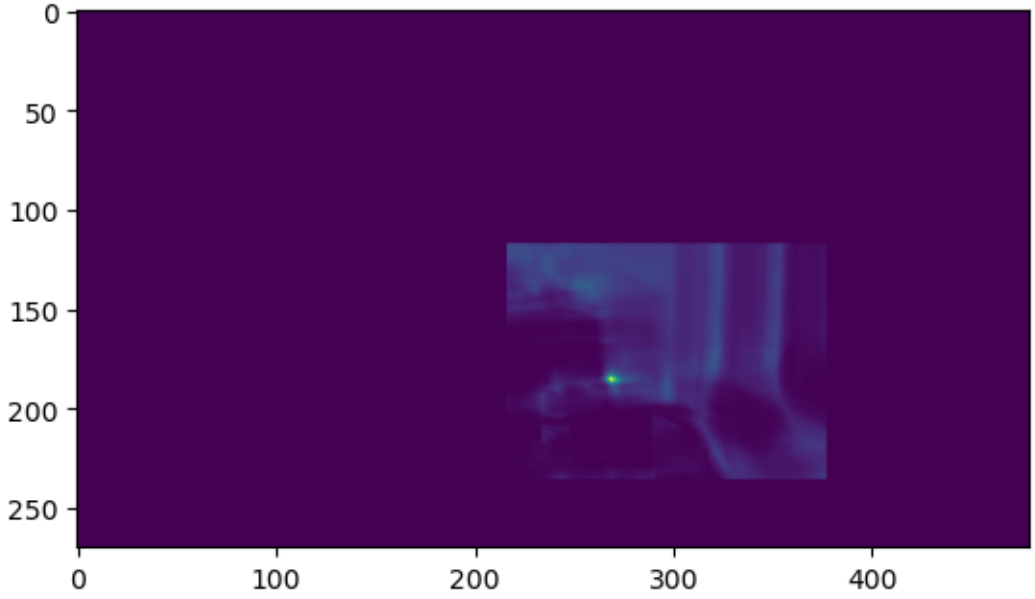
8%| | 8/102 [00:09<01:56, 1.23s/it]

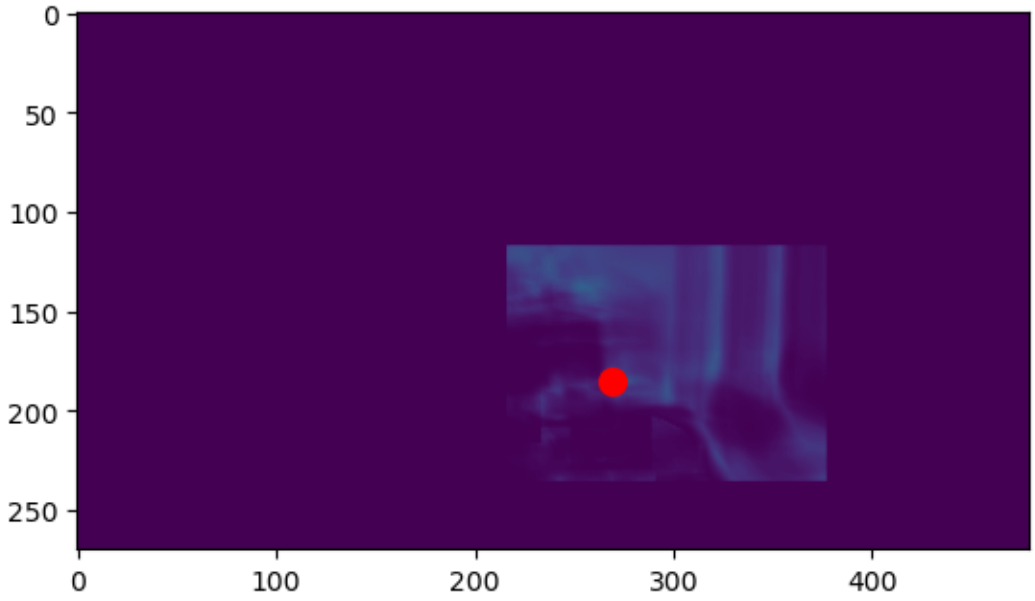

0
50
100
150
200
250
0
100
200
300
400

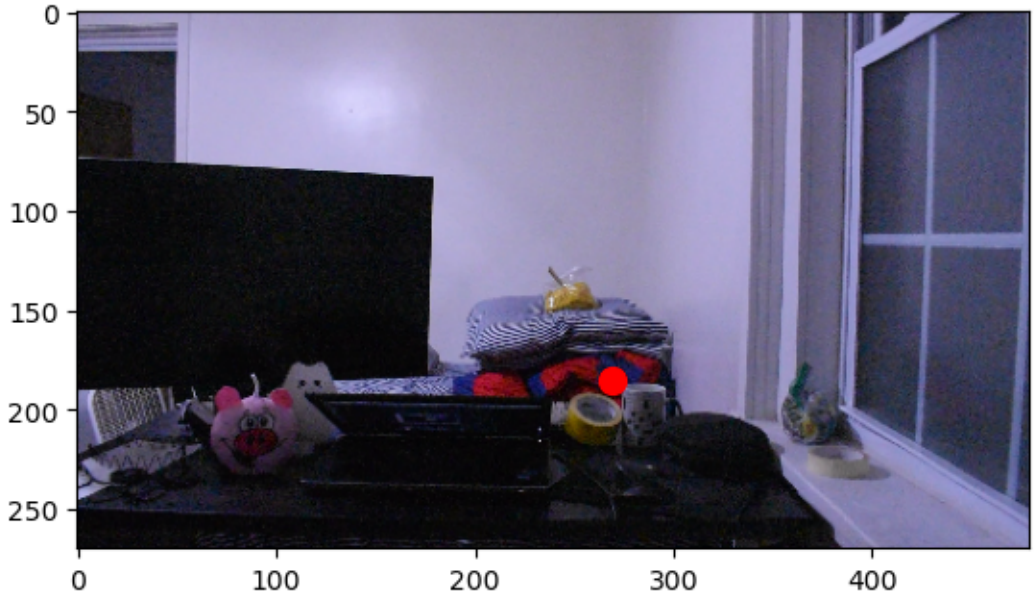

0
50
100
150
200
250
0
100
200
300
400



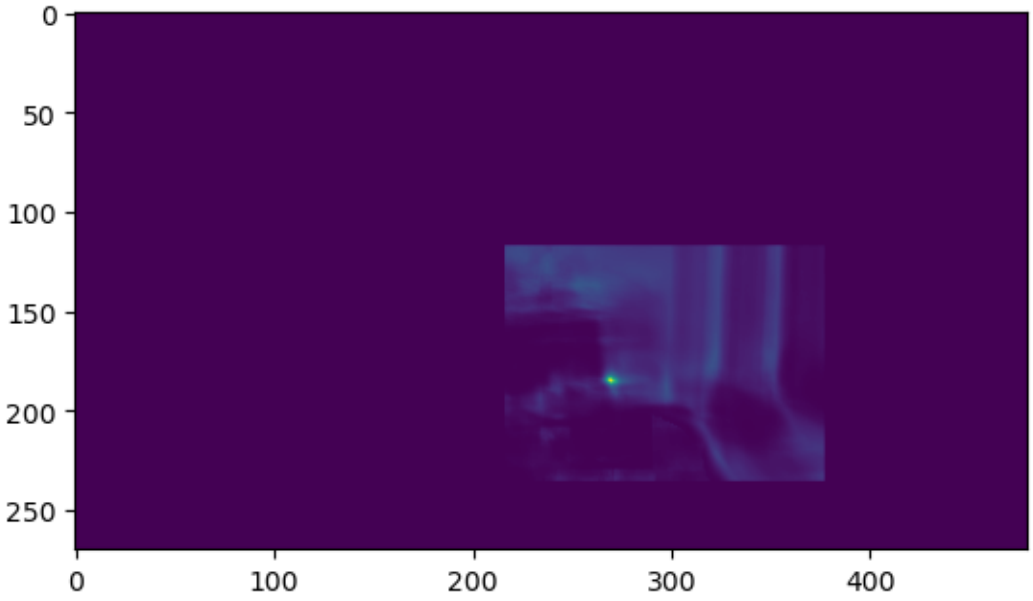
9%| | 9/102 [00:11<01:54, 1.23s/it]

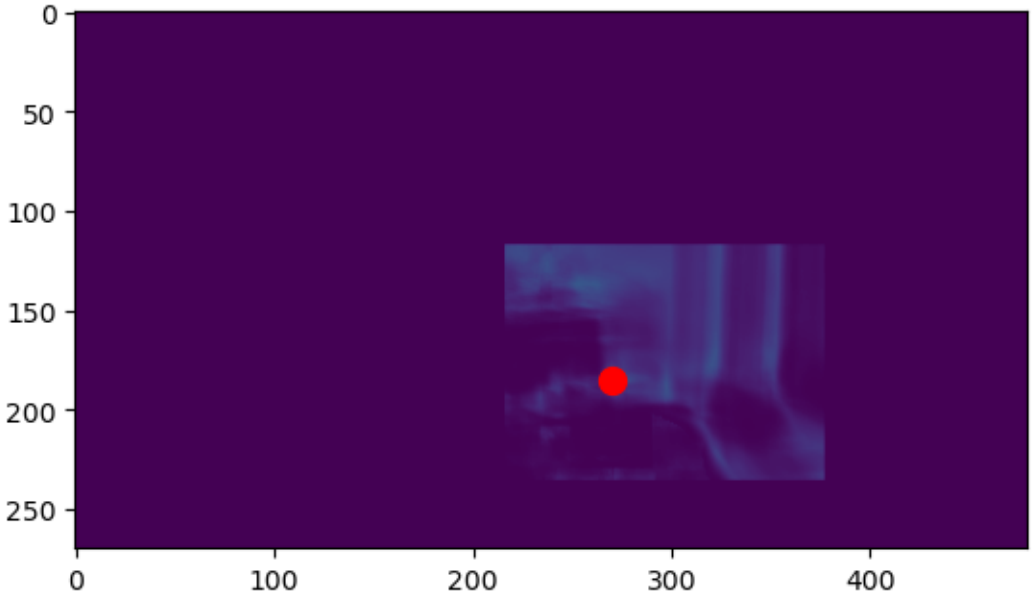


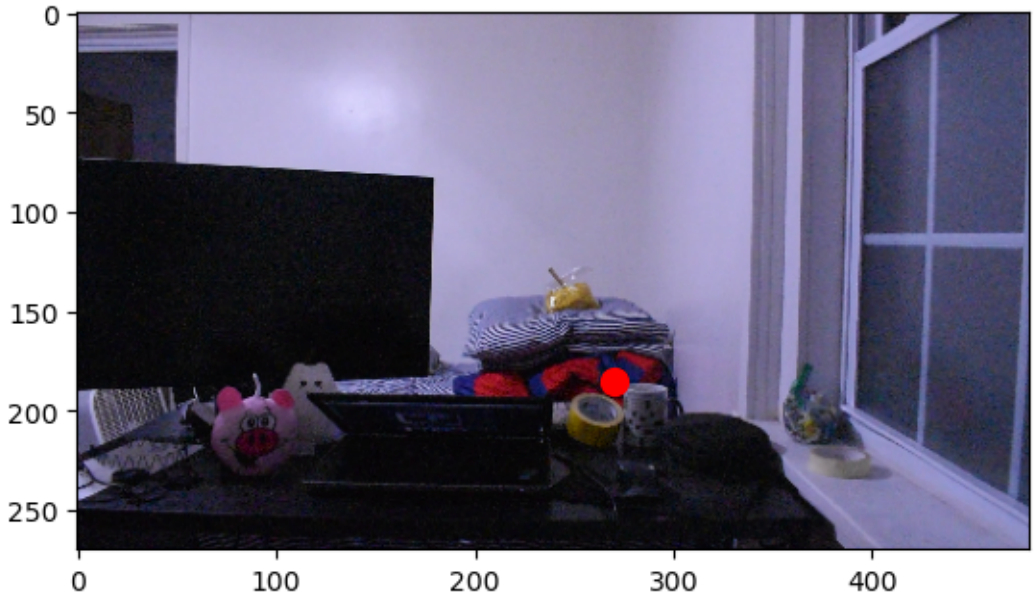




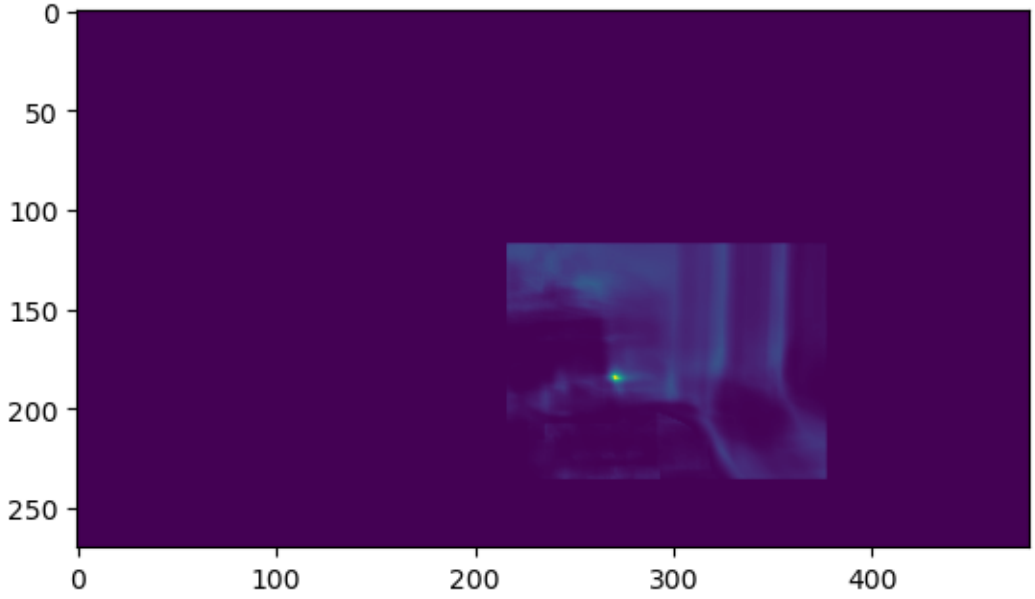
10%| | 10/102 [00:12<01:51, 1.21s/it]

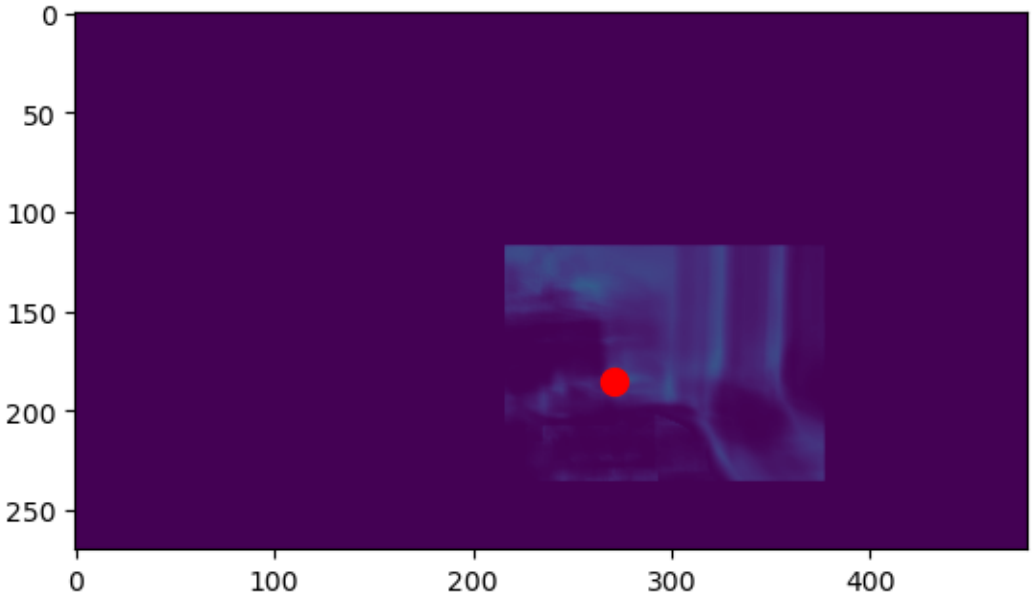

0
50
100
150
200
250
0
100
200
300
400

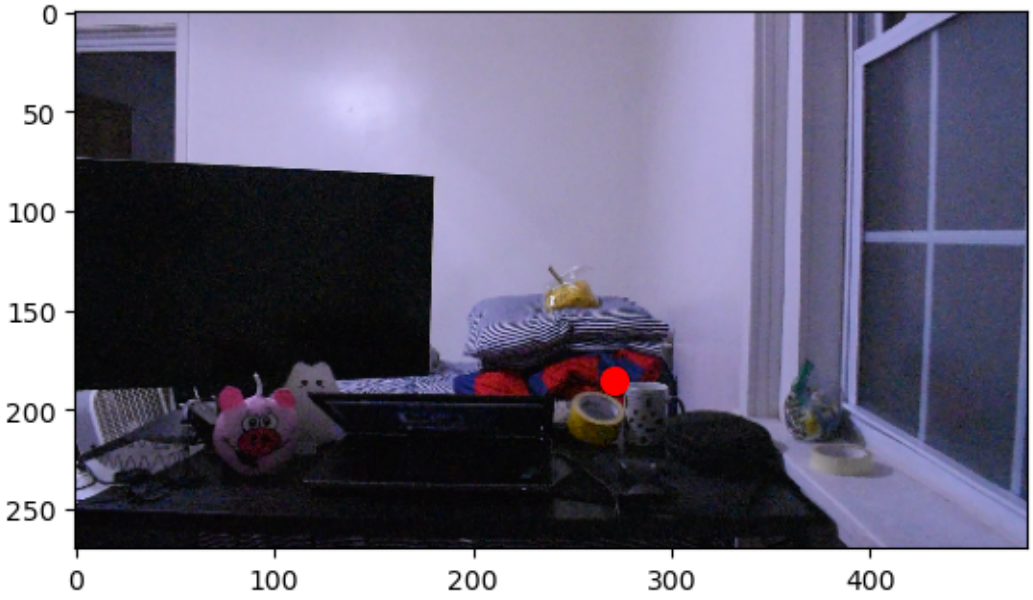

0
50
100
150
200
250
0
100
200
300
400



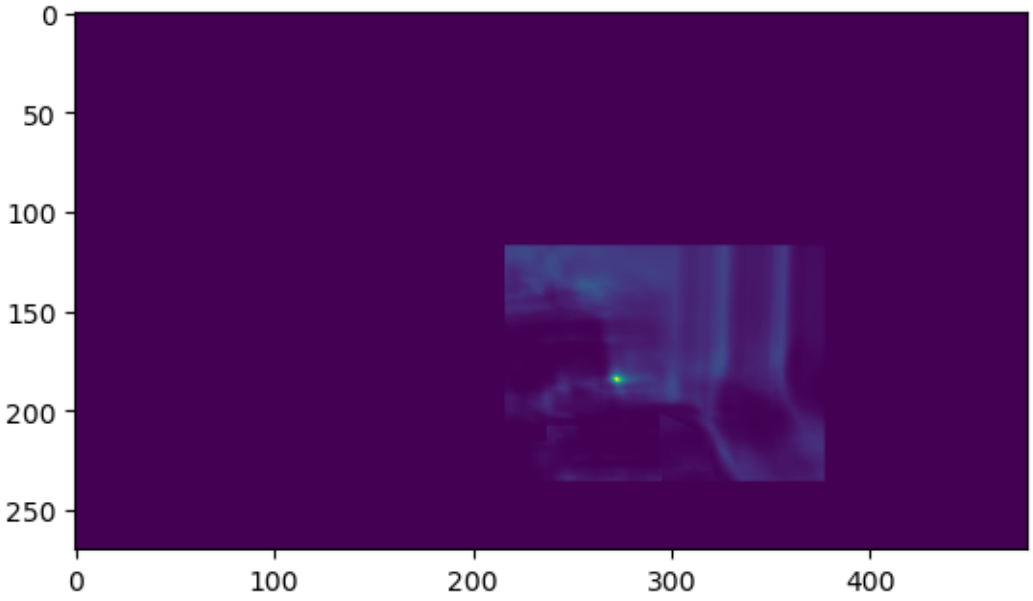
11%| | 11/102 [00:13<01:50, 1.21s/it]

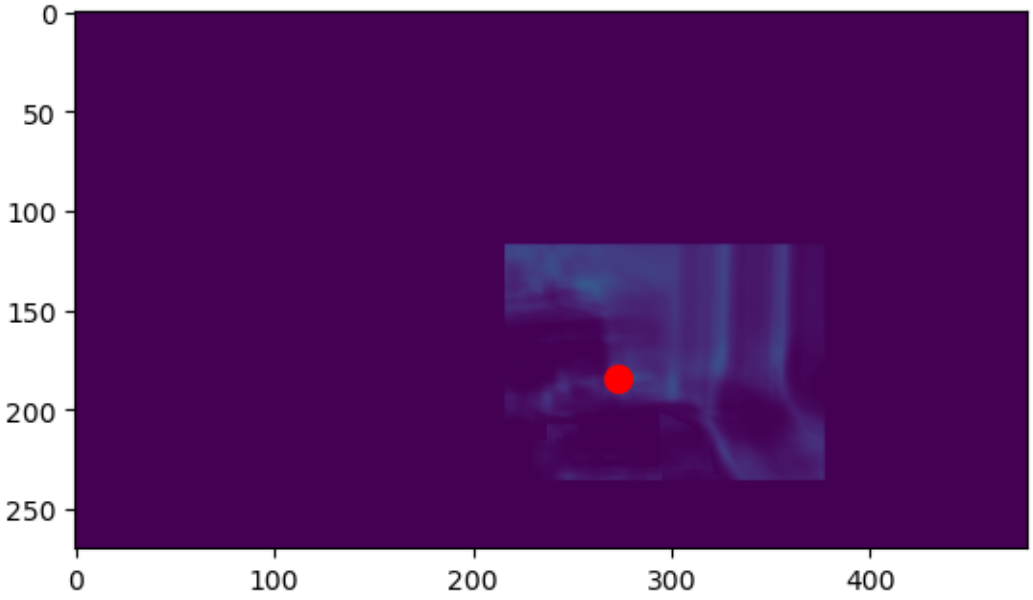


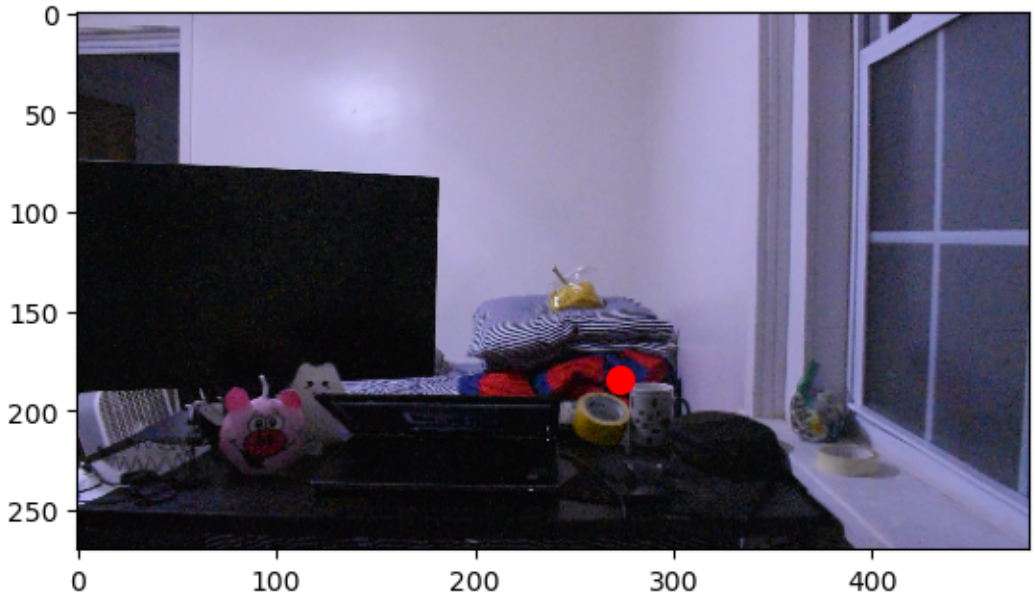




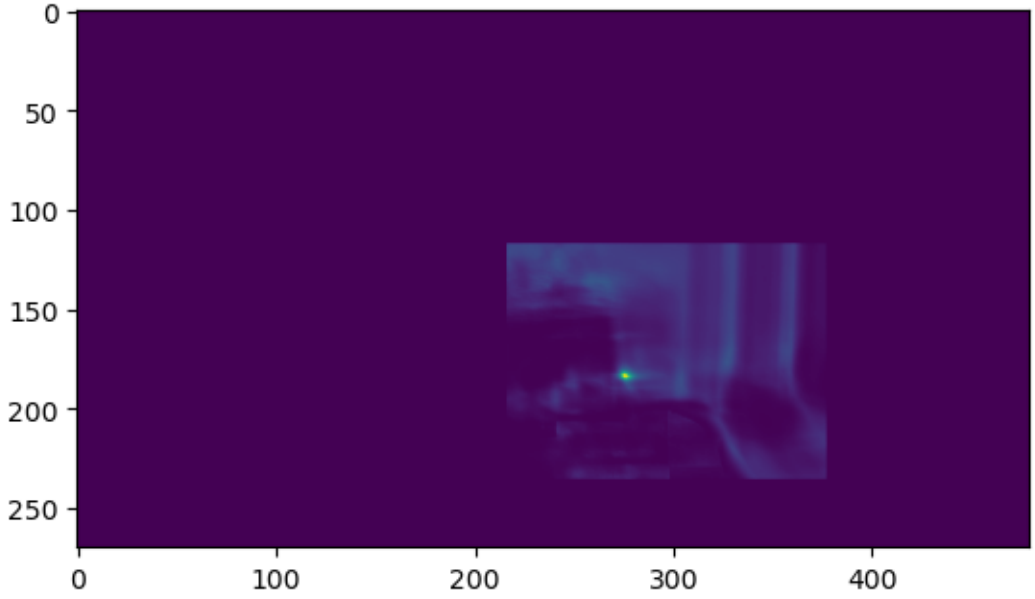
12%| | 12/102 [00:14<01:51, 1.24s/it]

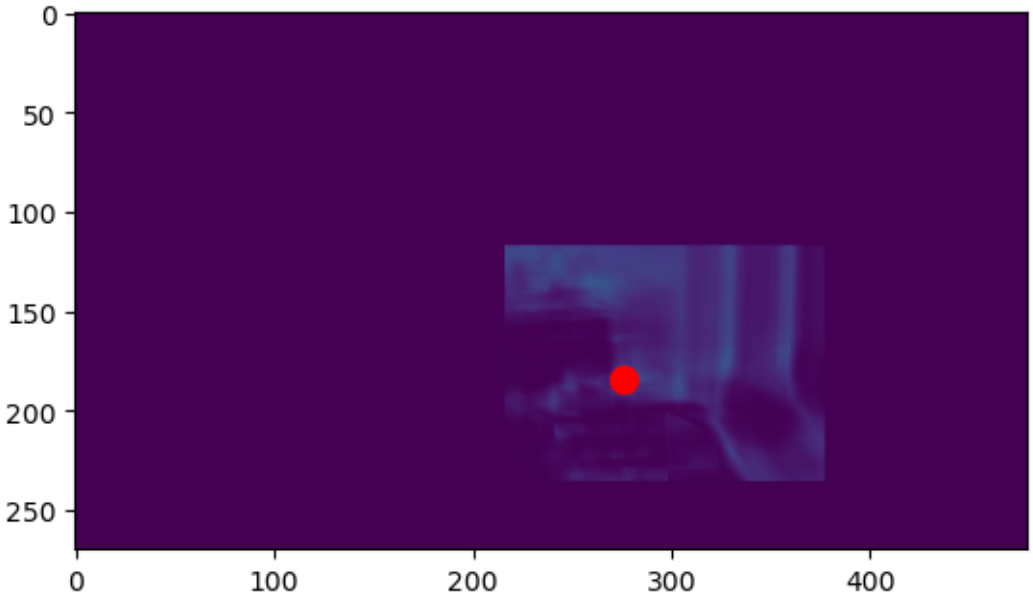

0
50
100
150
200
250
0
100
200
300
400

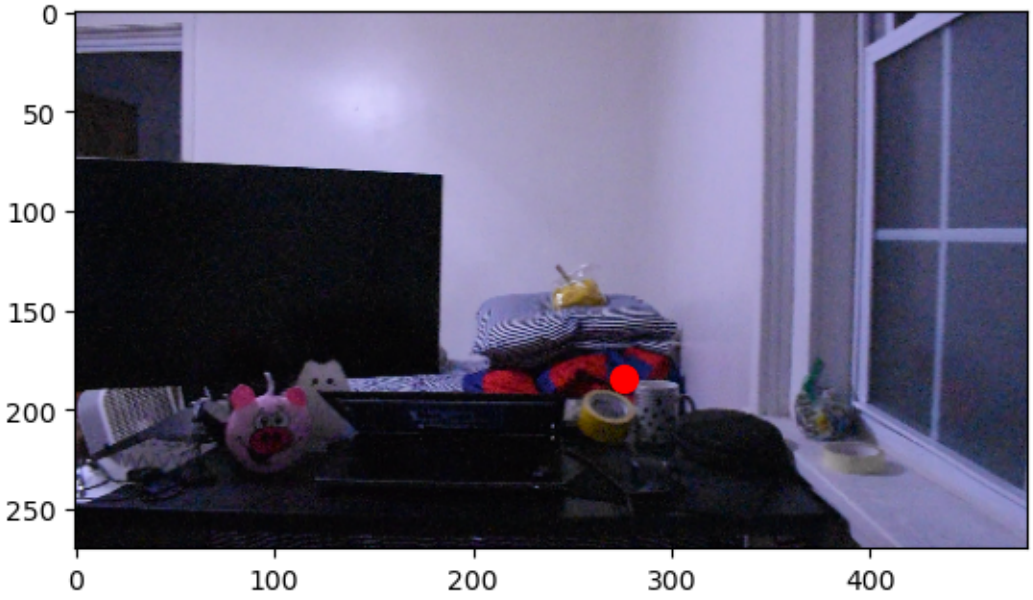

0
50
100
150
200
250
0
100
200
300
400



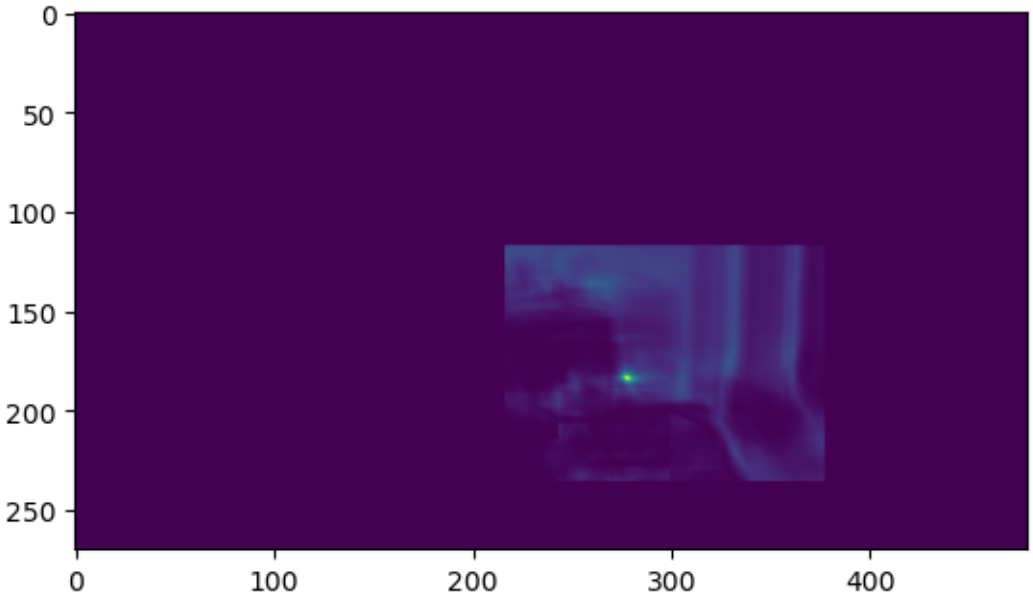
13%| | 13/102 [00:15<01:48, 1.22s/it]

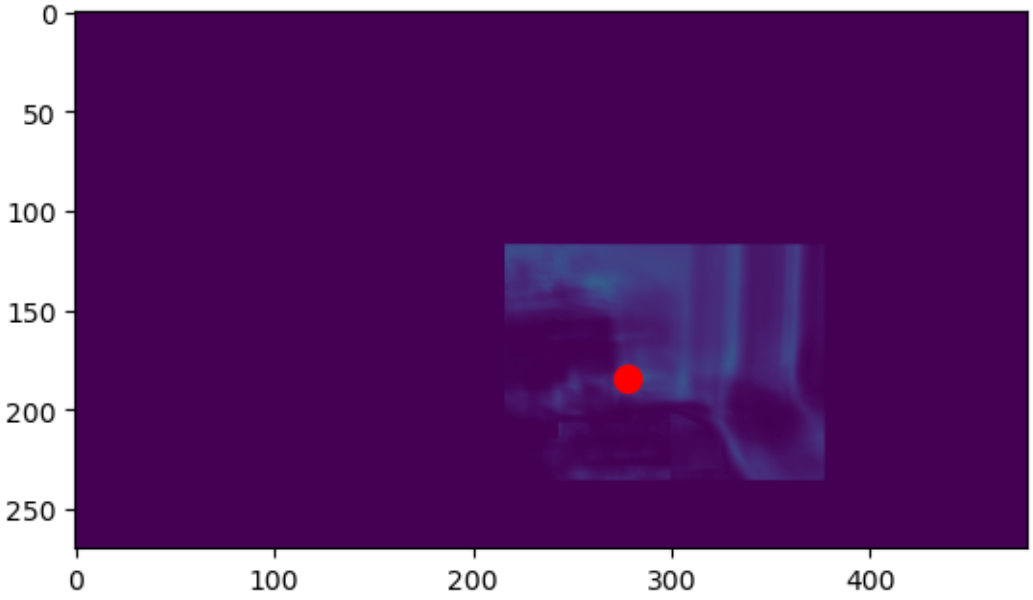


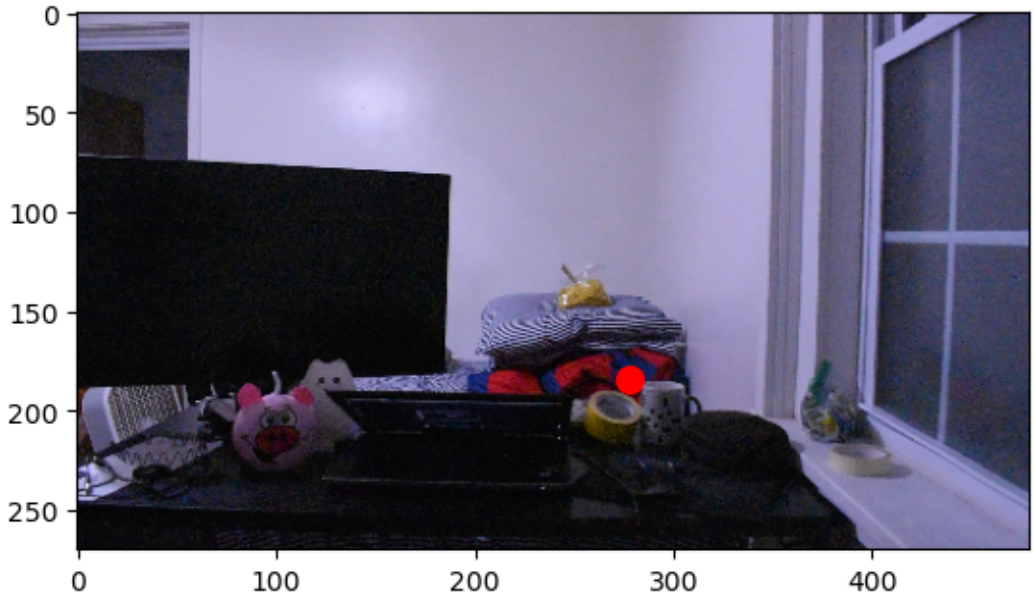




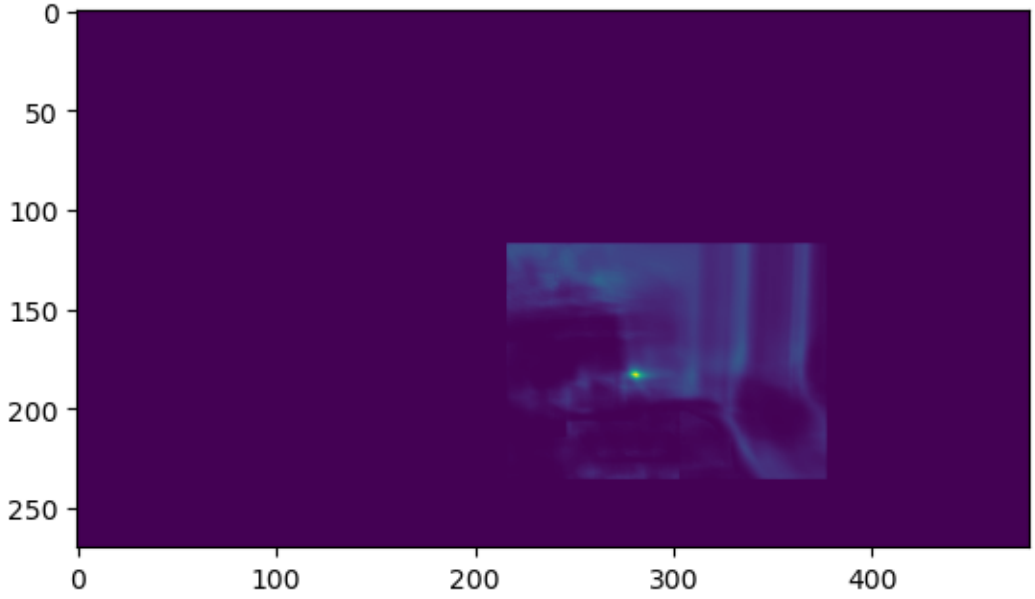
14%| | 14/102 [00:17<01:46, 1.21s/it]

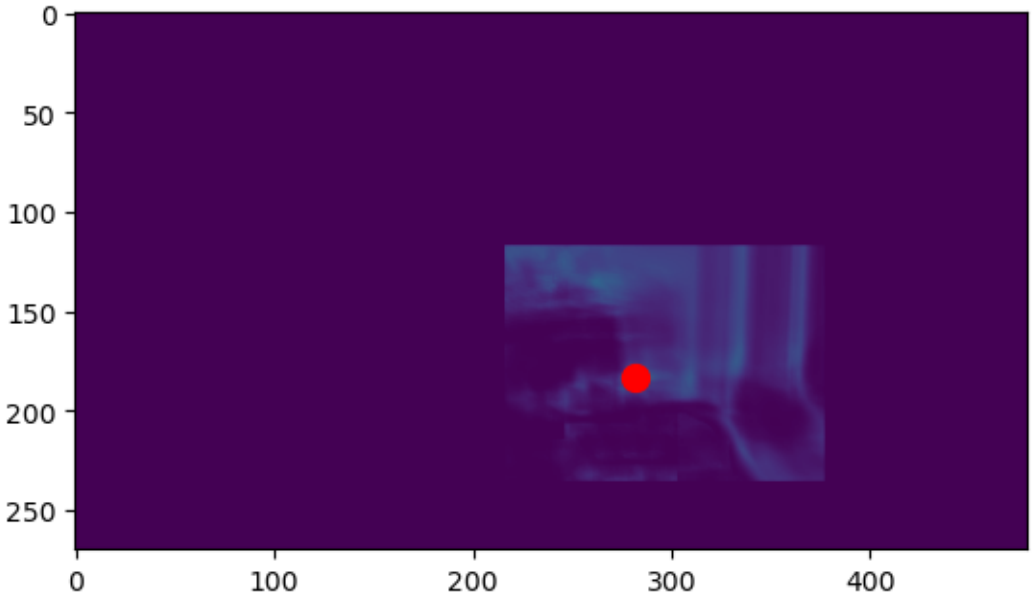

0
50
100
150
200
250
0
100
200
300
400

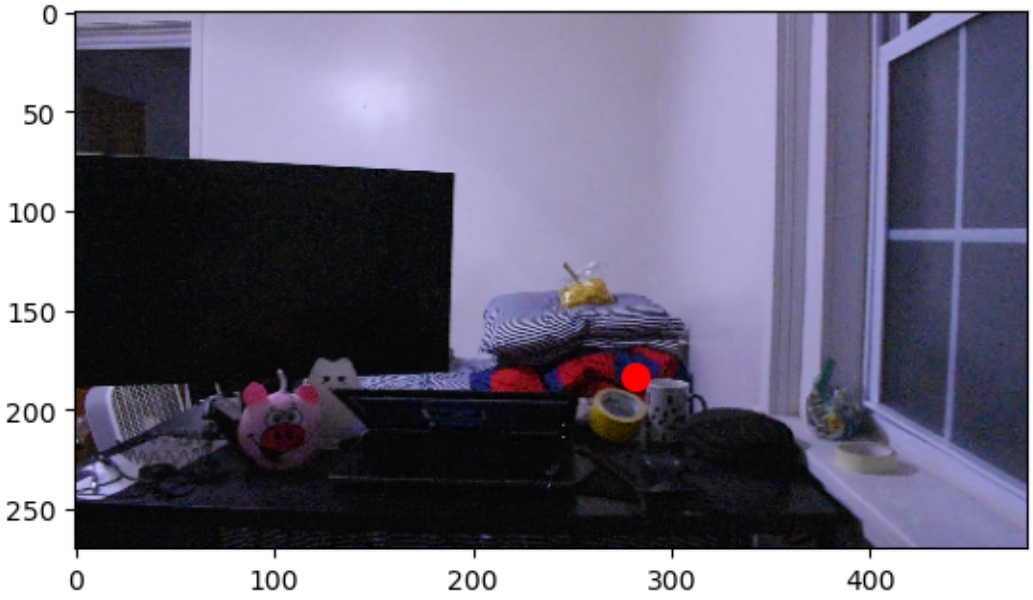

0
50
100
150
200
250
0
100
200
300
400



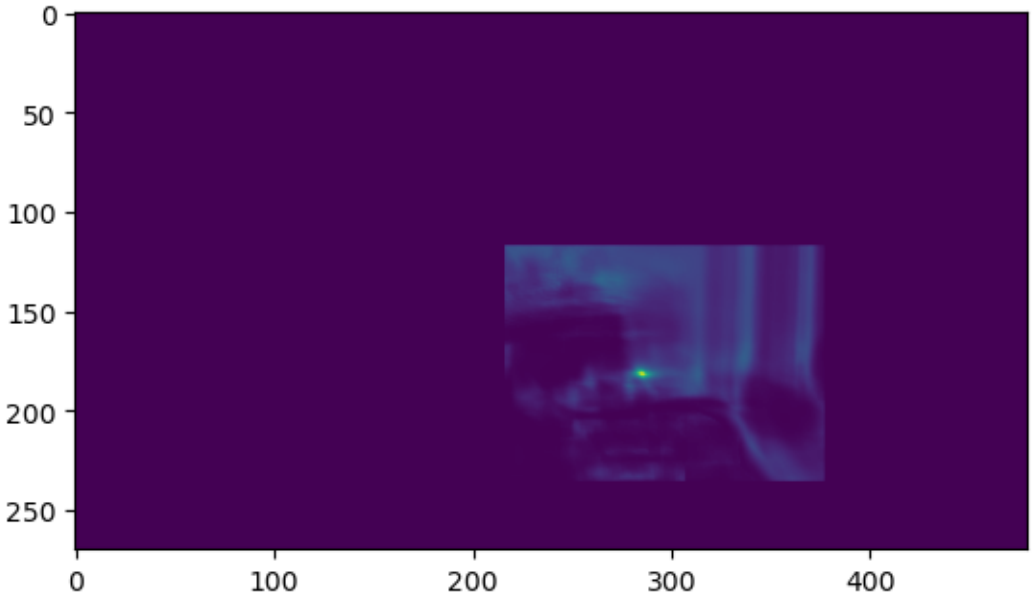
15%| | 15/102 [00:18<01:44, 1.20s/it]

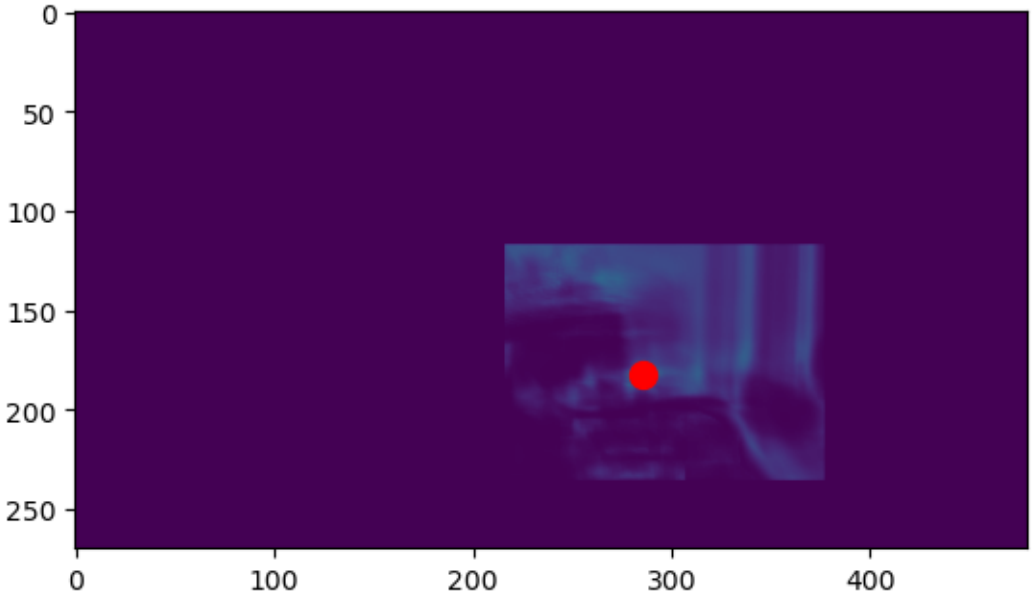


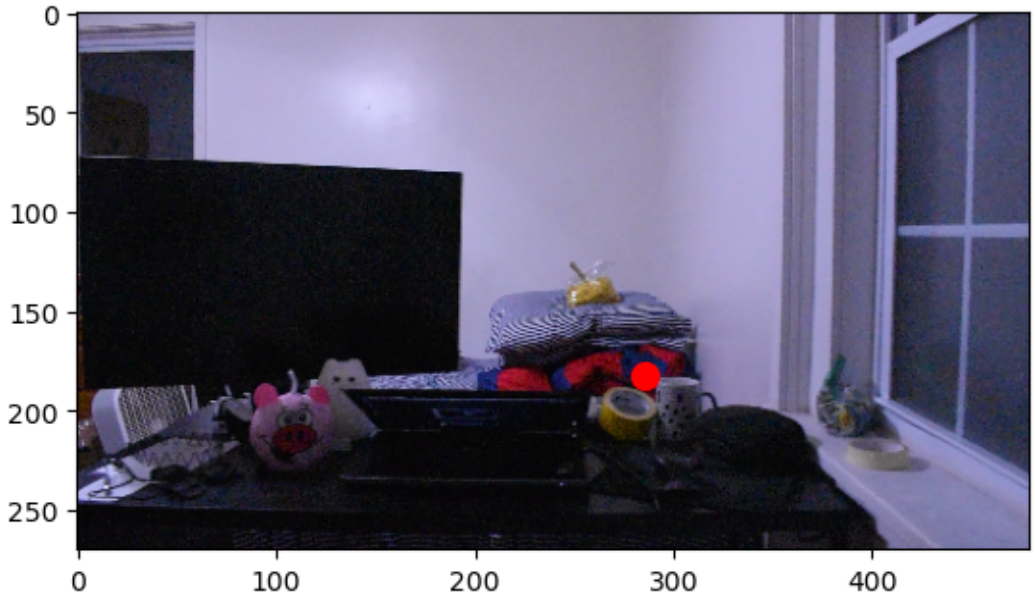




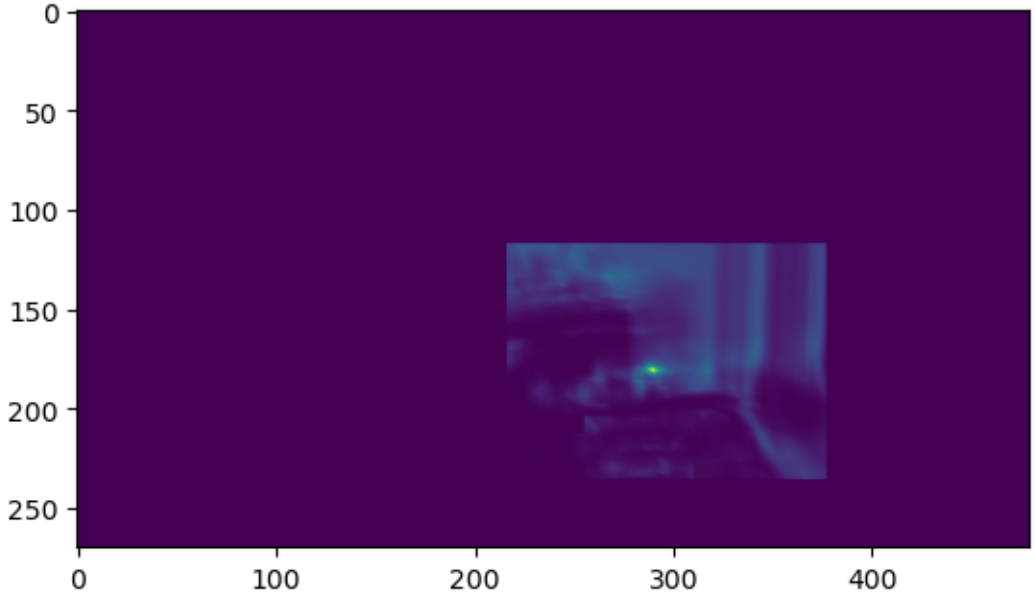
16%| | 16/102 [00:19<01:43, 1.20s/it]

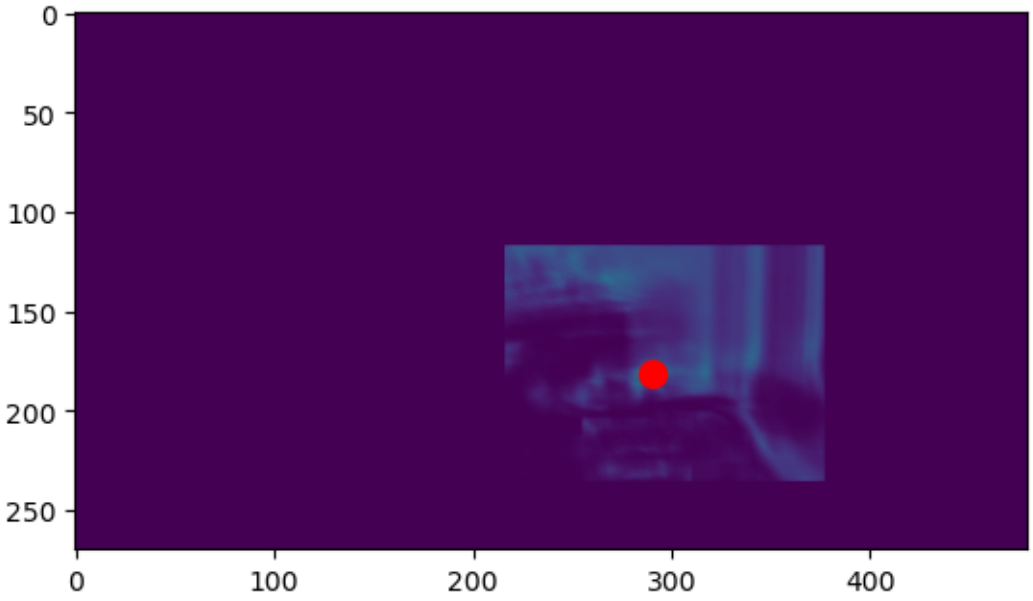

0
50
100
150
200
250
0
100
200
300
400

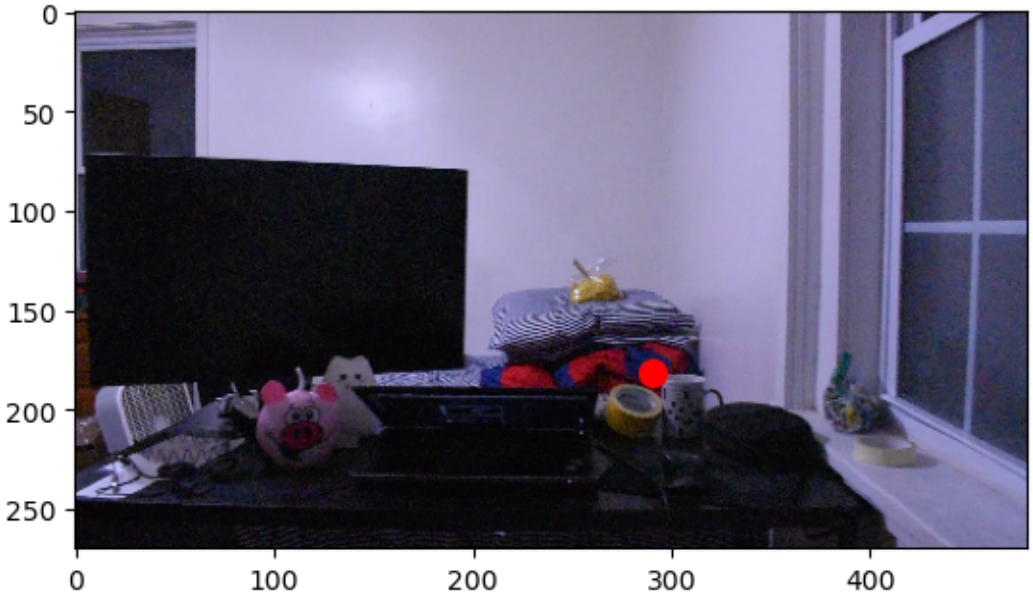

0
50
100
150
200
250
0
100
200
300
400



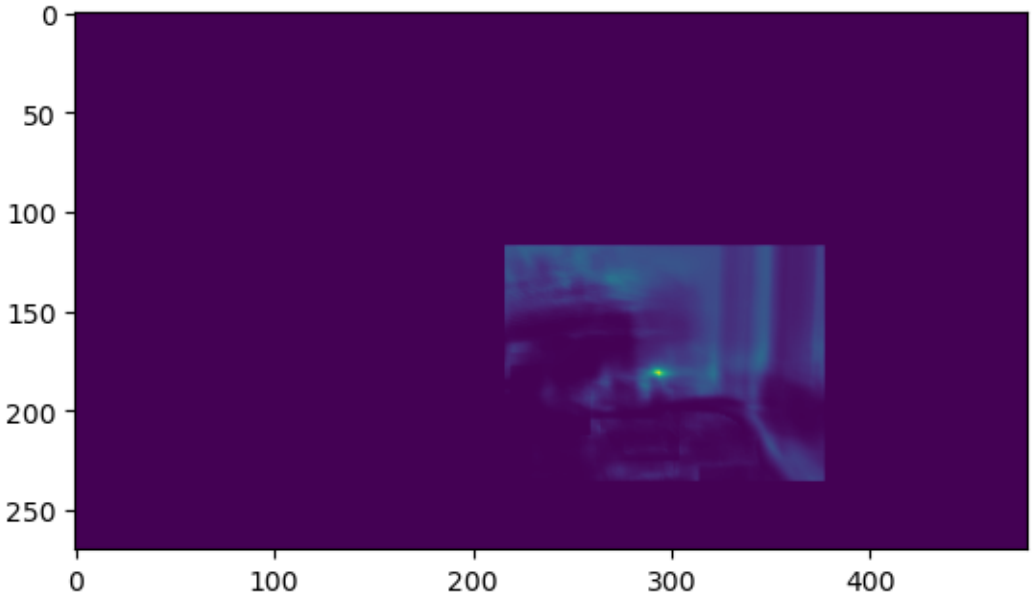
17%| | 17/102 [00:20<01:42, 1.20s/it]

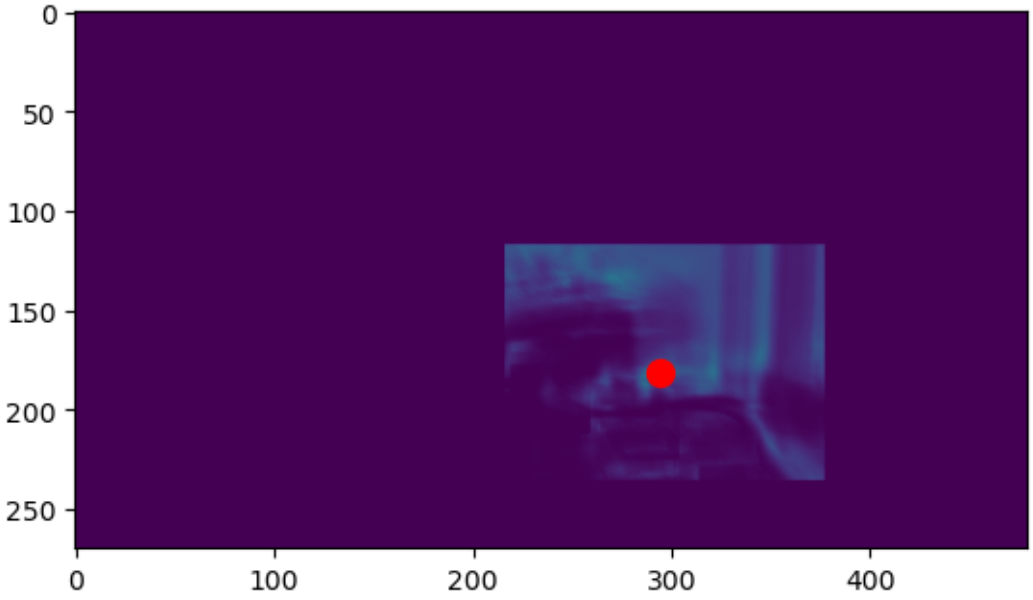


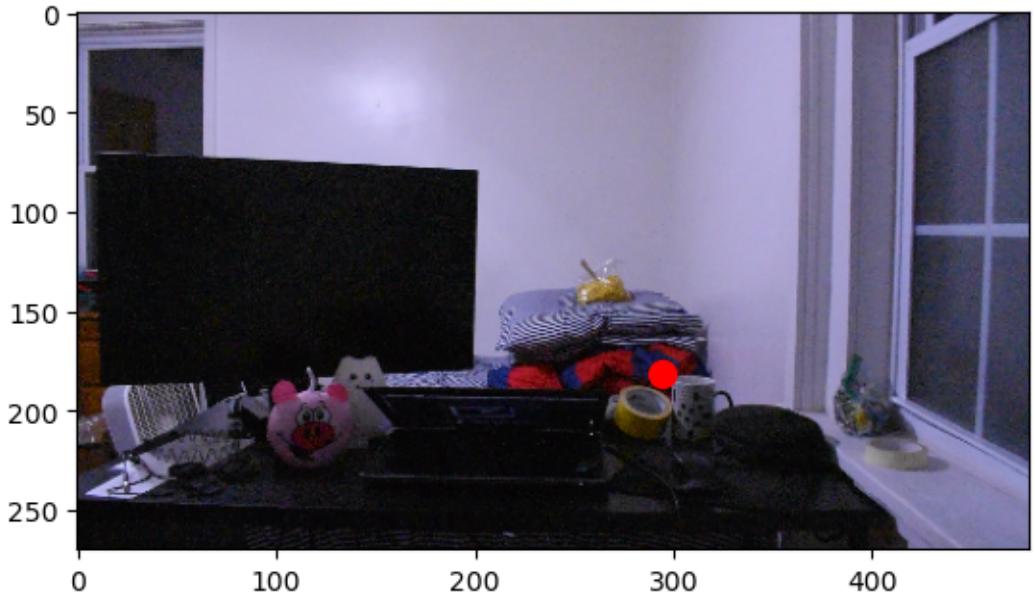




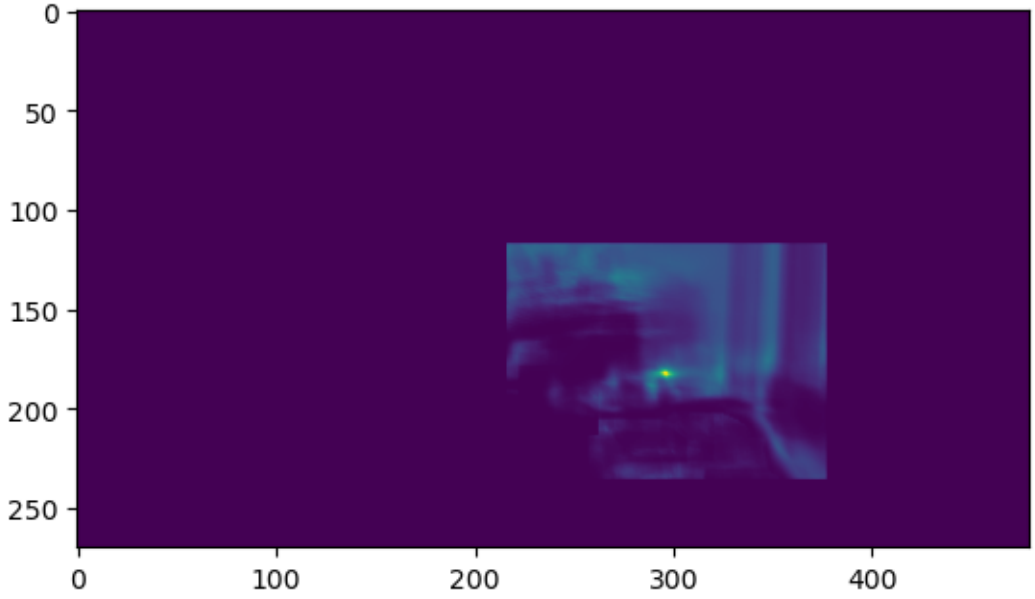
18%| | 18/102 [00:21<01:40, 1.19s/it]

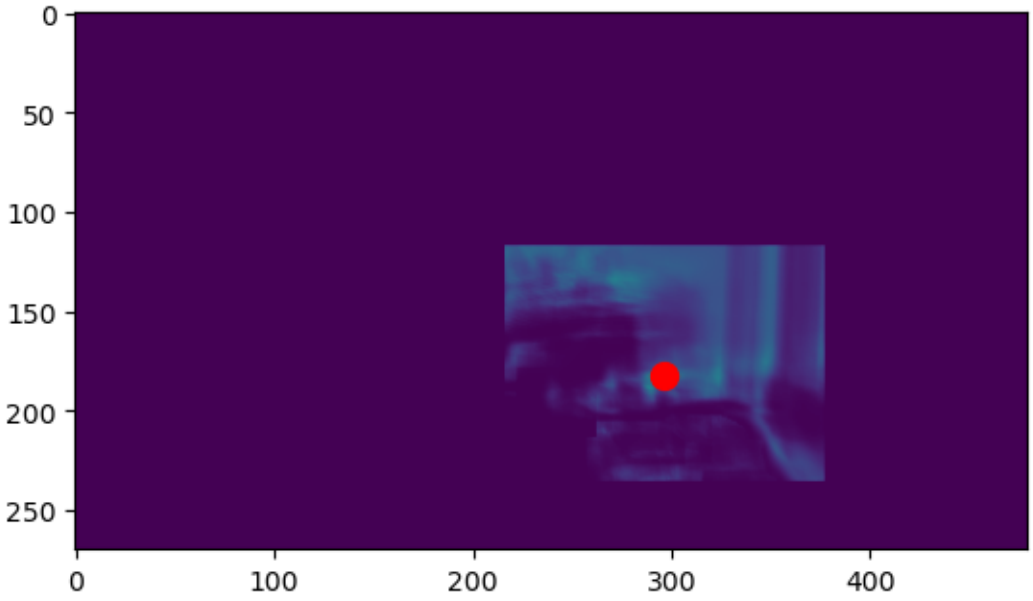

0
50
100
150
200
250
0
100
200
300
400

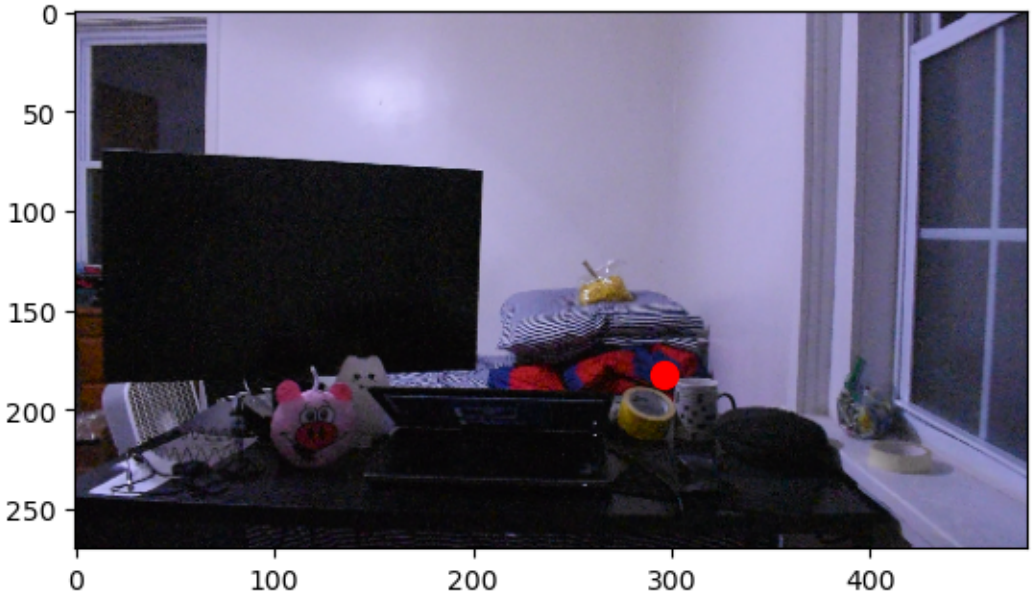

0
50
100
150
200
250
0
100
200
300
400



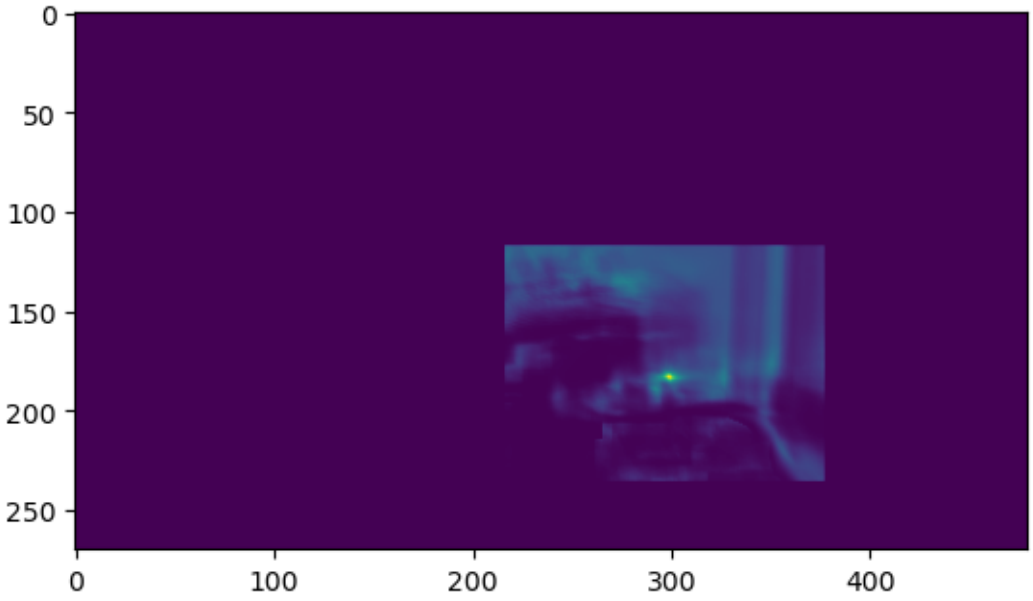
19%| | 19/102 [00:23<01:43, 1.24s/it]

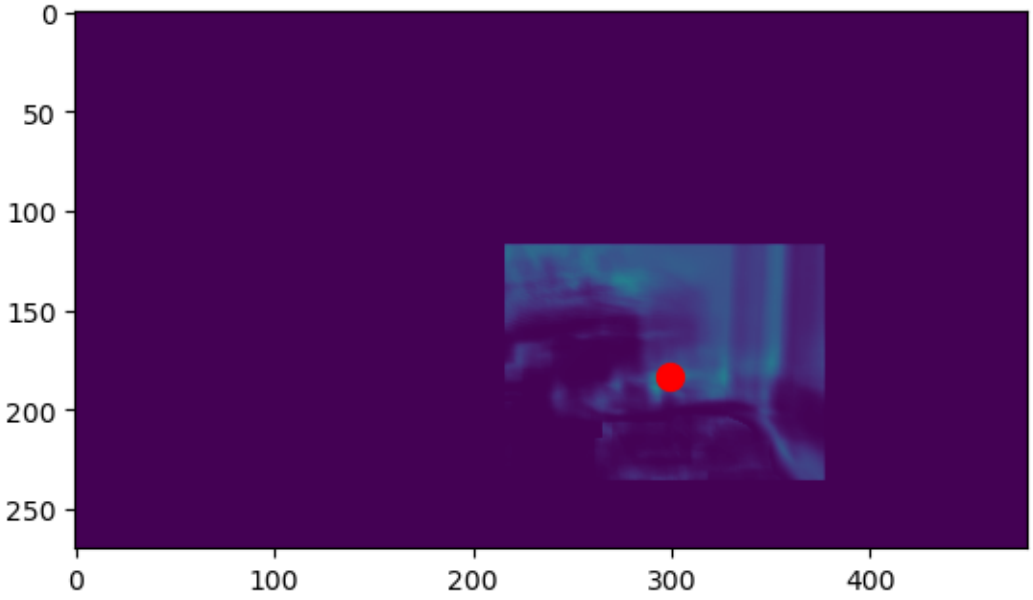


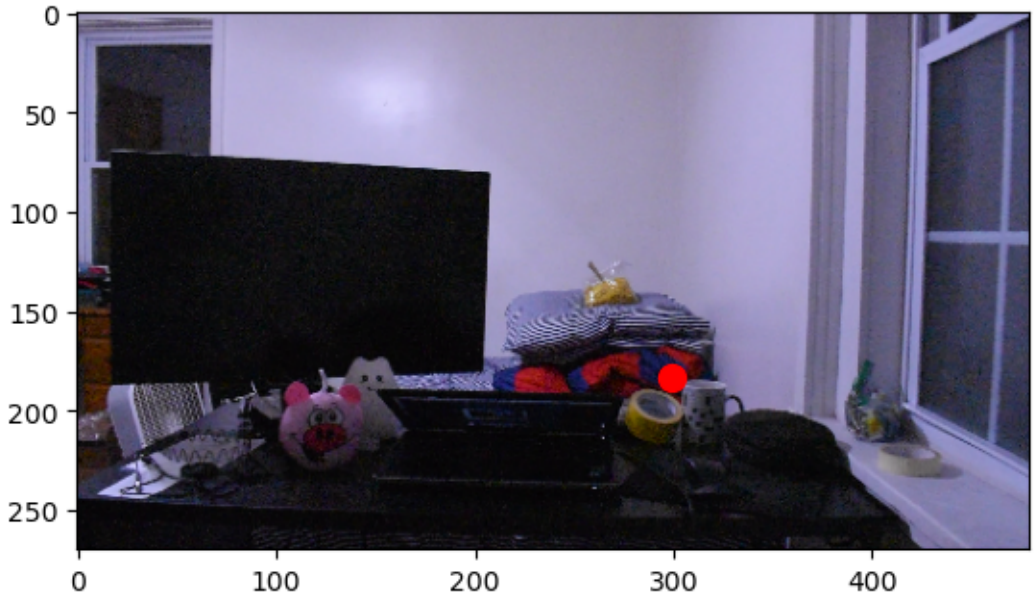




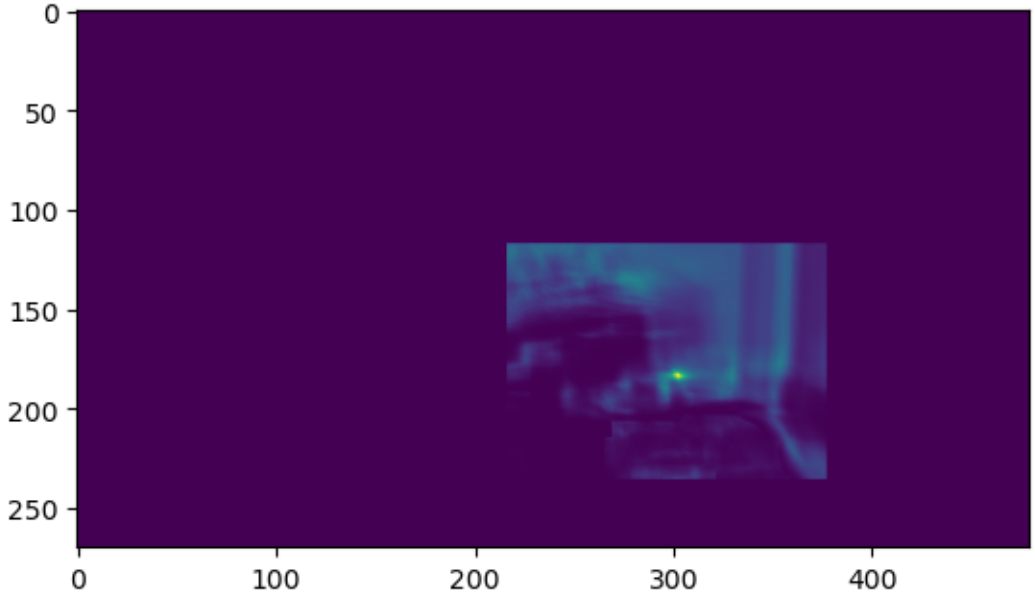
20%| | 20/102 [00:24<01:41, 1.23s/it]

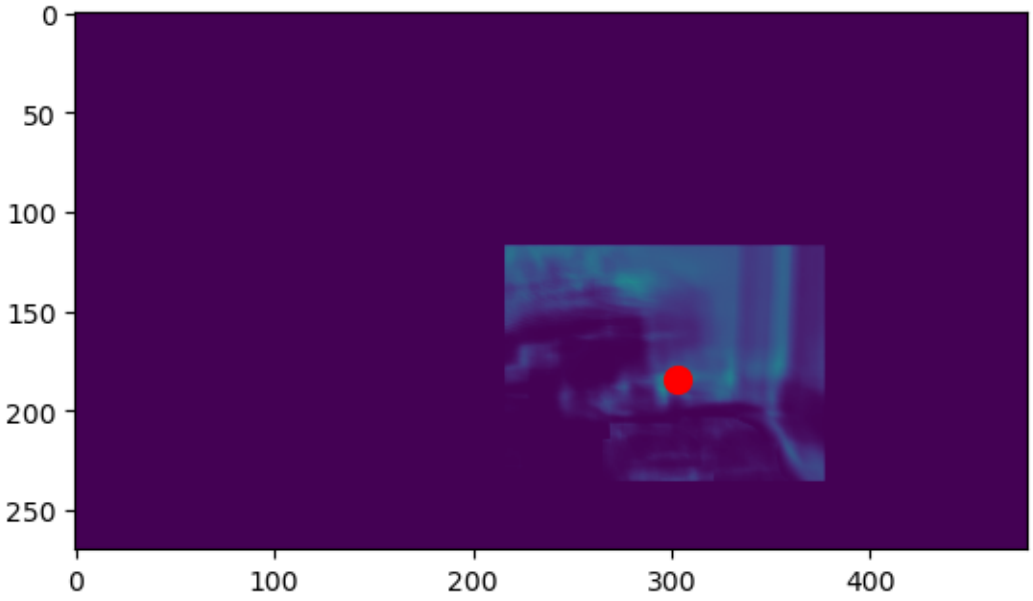

0
50
100
150
200
250
0
100
200
300
400

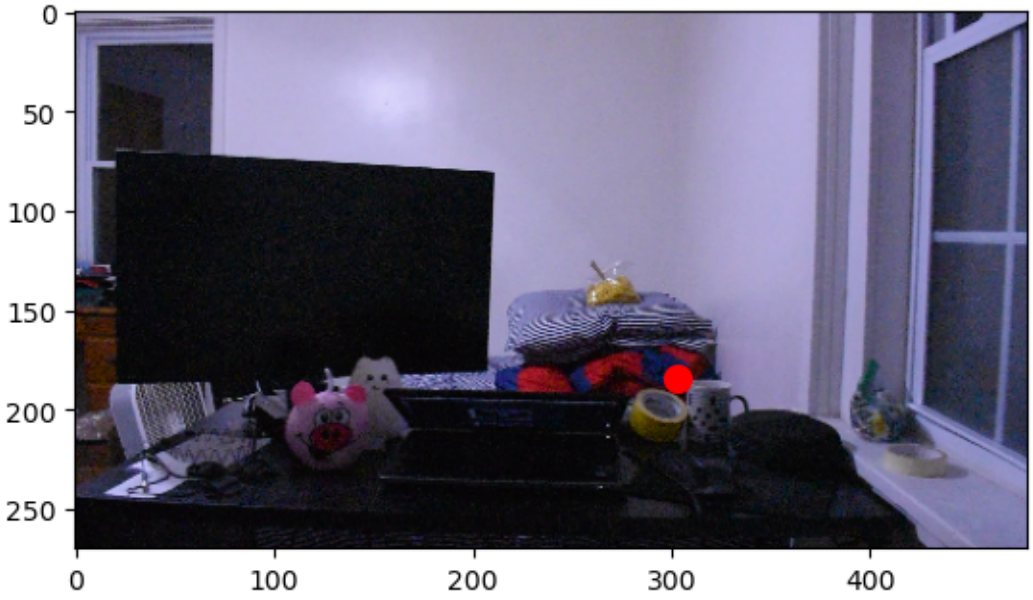

0
50
100
150
200
250
0
100
200
300
400



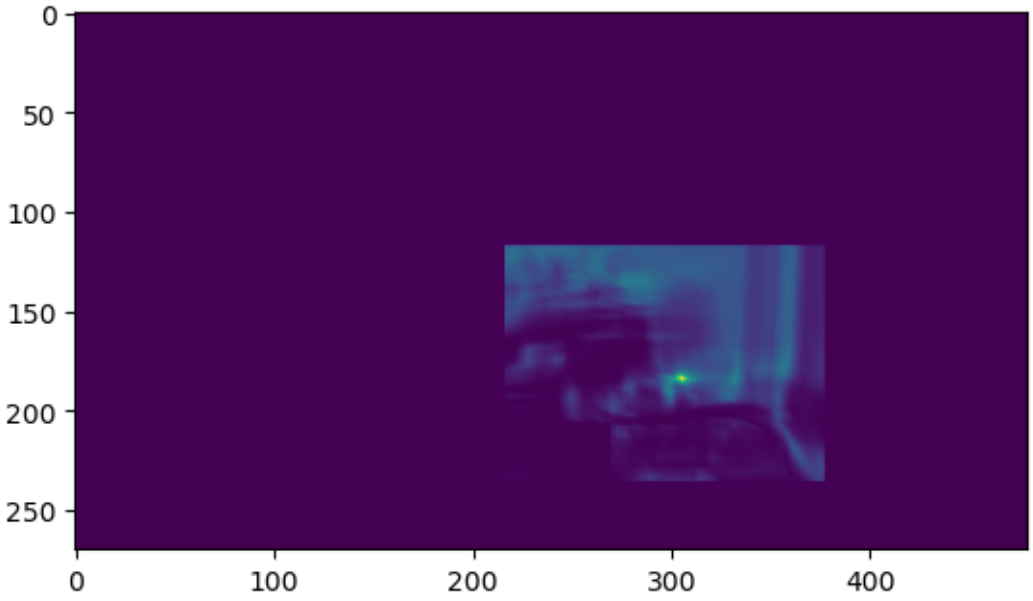
21%| | 21/102 [00:25<01:38, 1.22s/it]

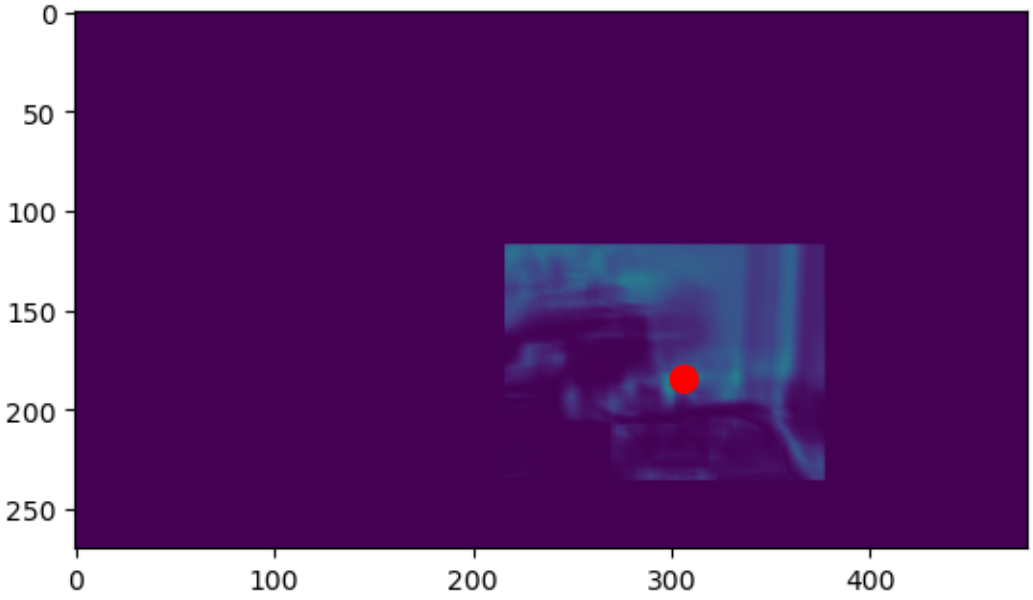


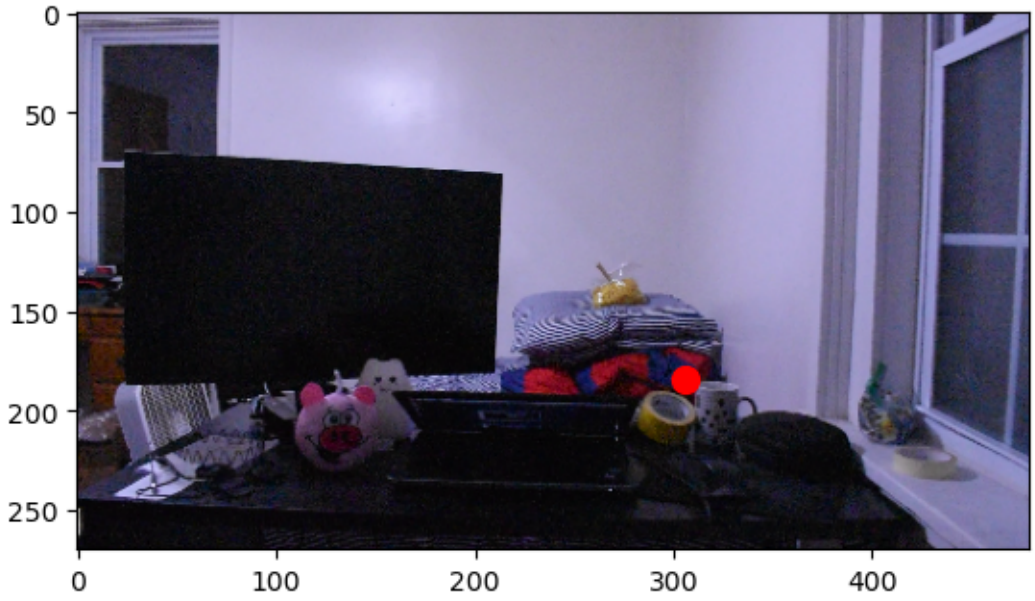




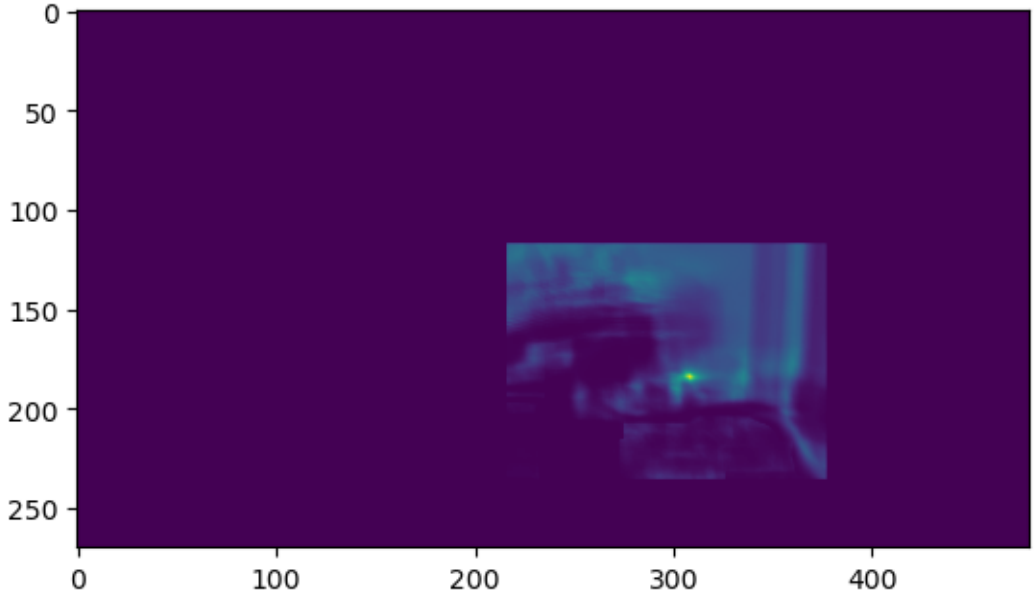
22%| | 22/102 [00:26<01:36, 1.21s/it]

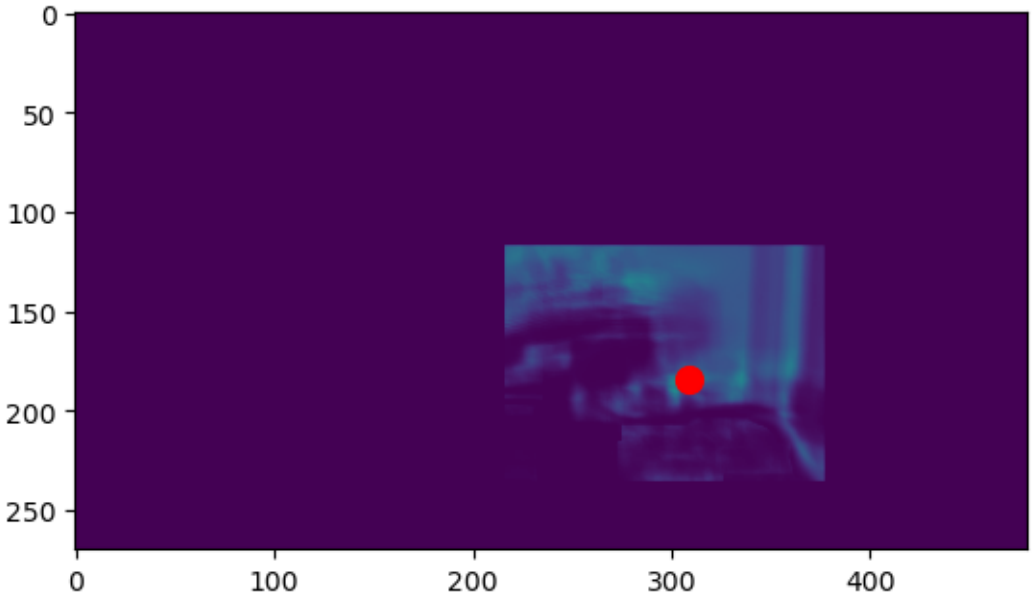

0
50
100
150
200
250
0
100
200
300
400

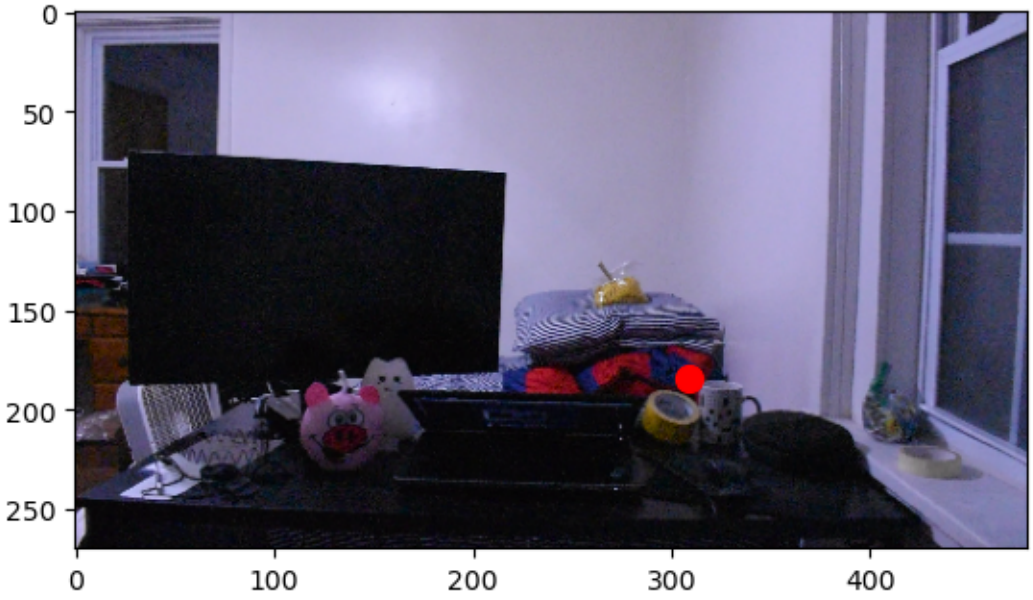

0
50
100
150
200
250
0
100
200
300
400



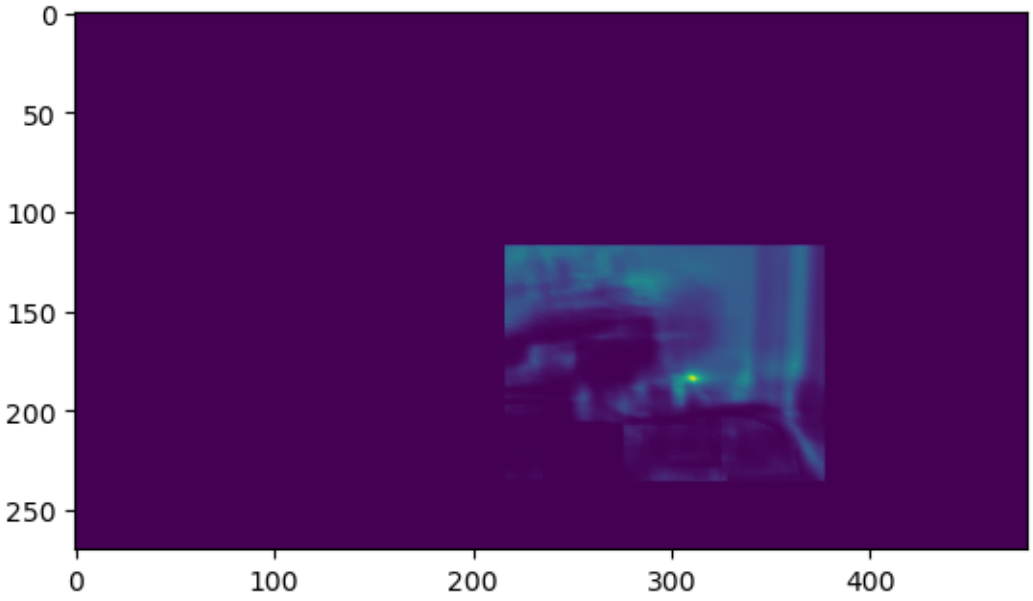
23%| | 23/102 [00:27<01:35, 1.20s/it]

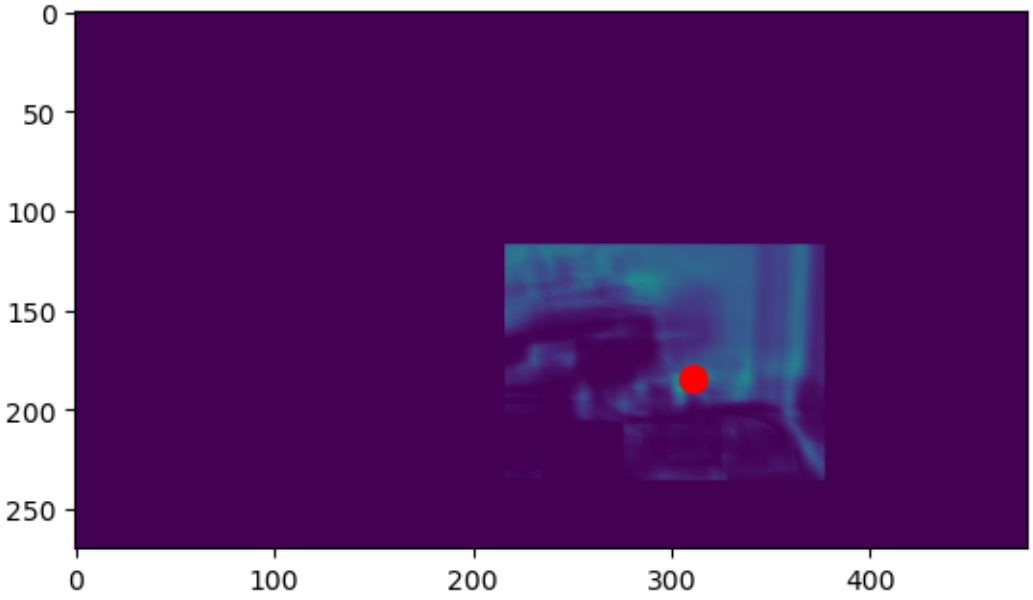


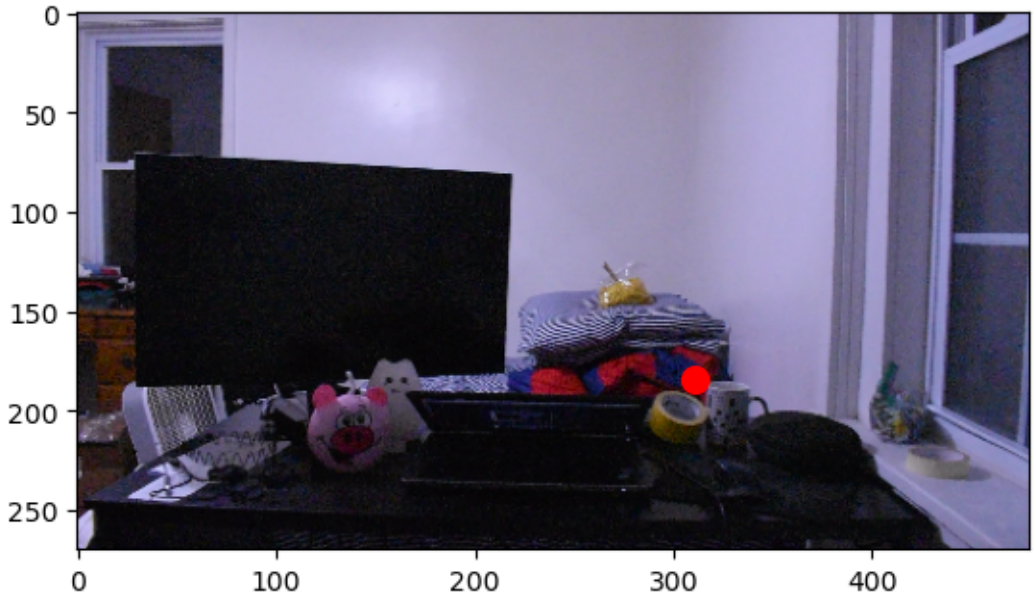




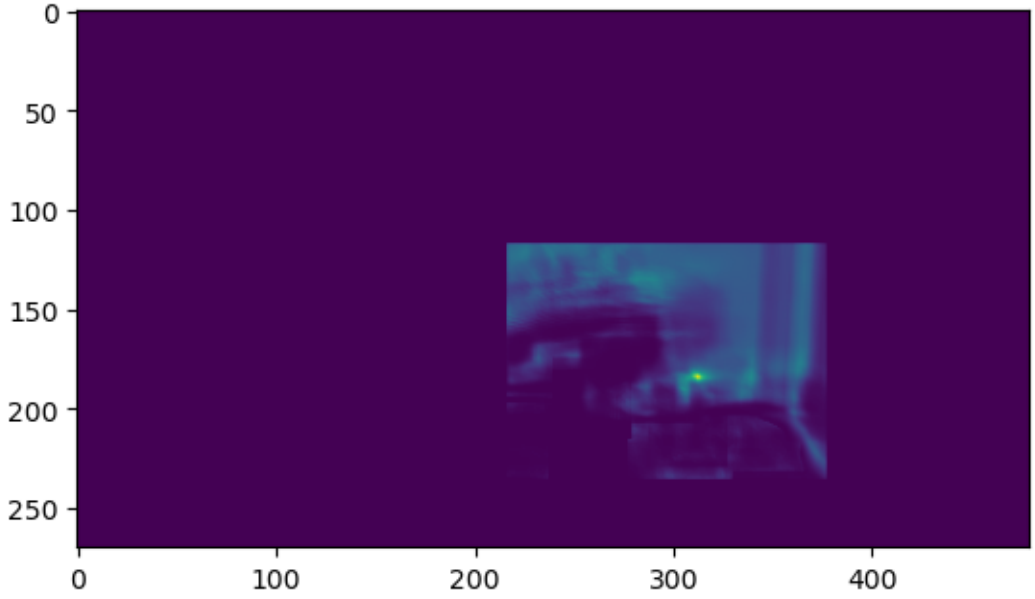
24%| | 24/102 [00:29<01:33, 1.20s/it]

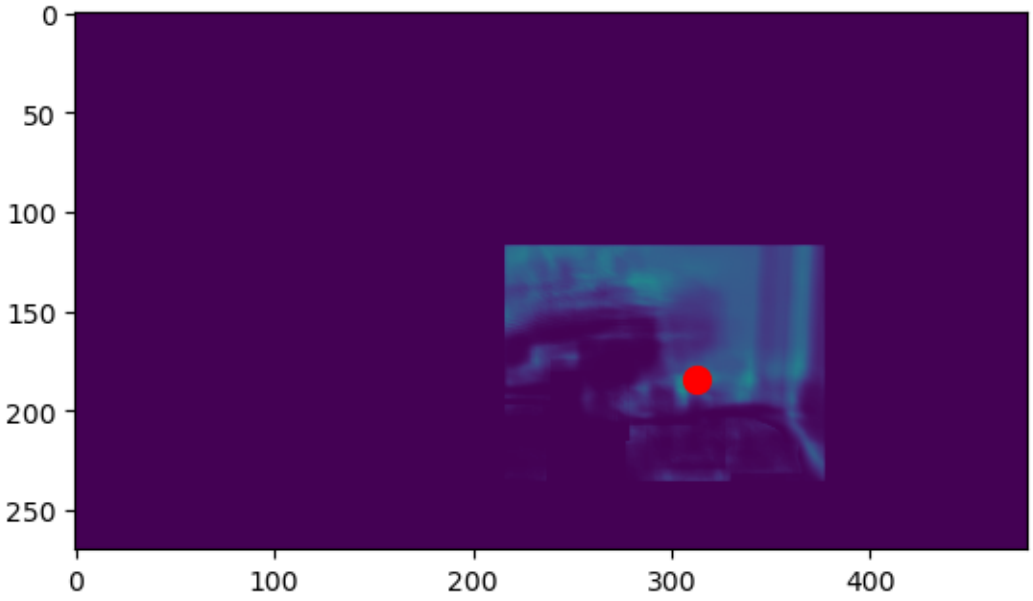

0
50
100
150
200
250
0
100
200
300
400

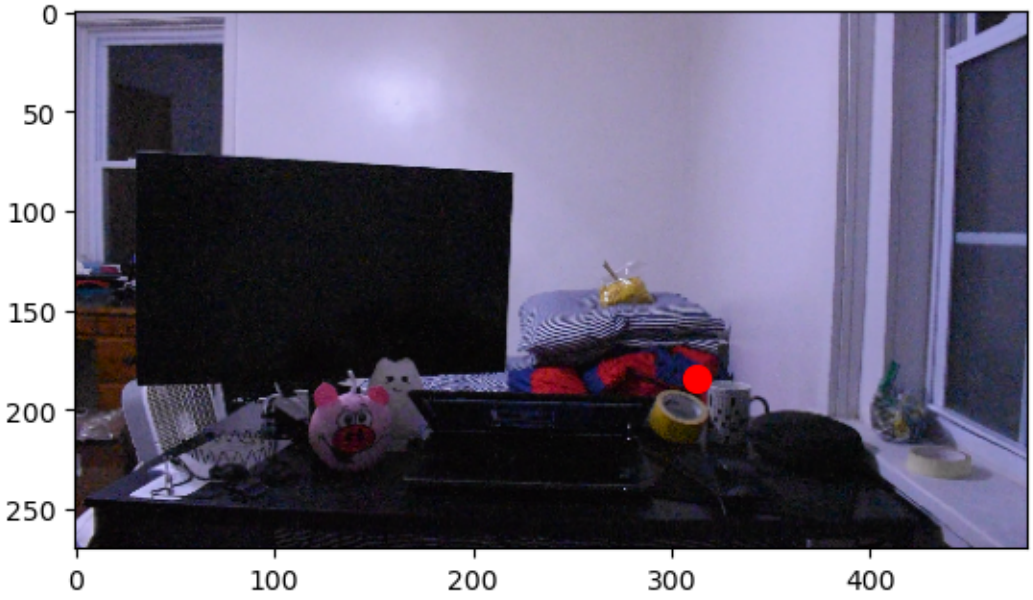

0
50
100
150
200
250
0
100
200
300
400



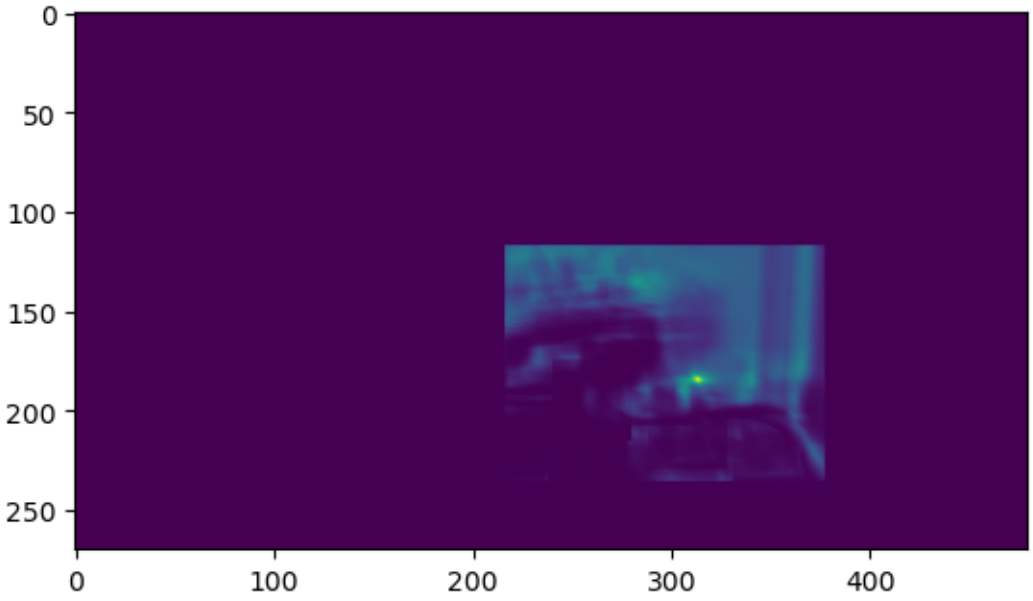
25%| | 25/102 [00:30<01:32, 1.19s/it]

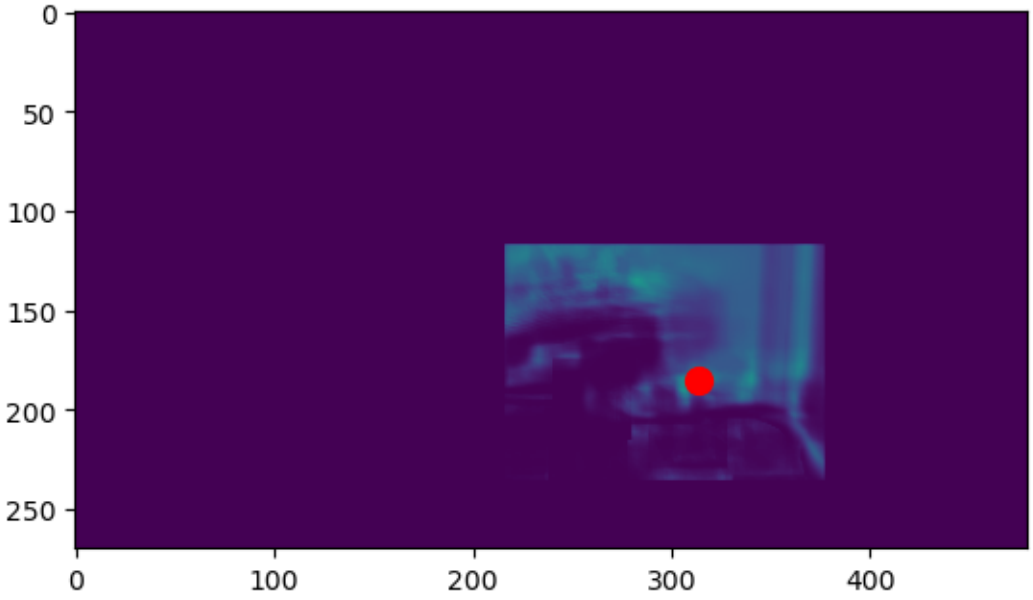


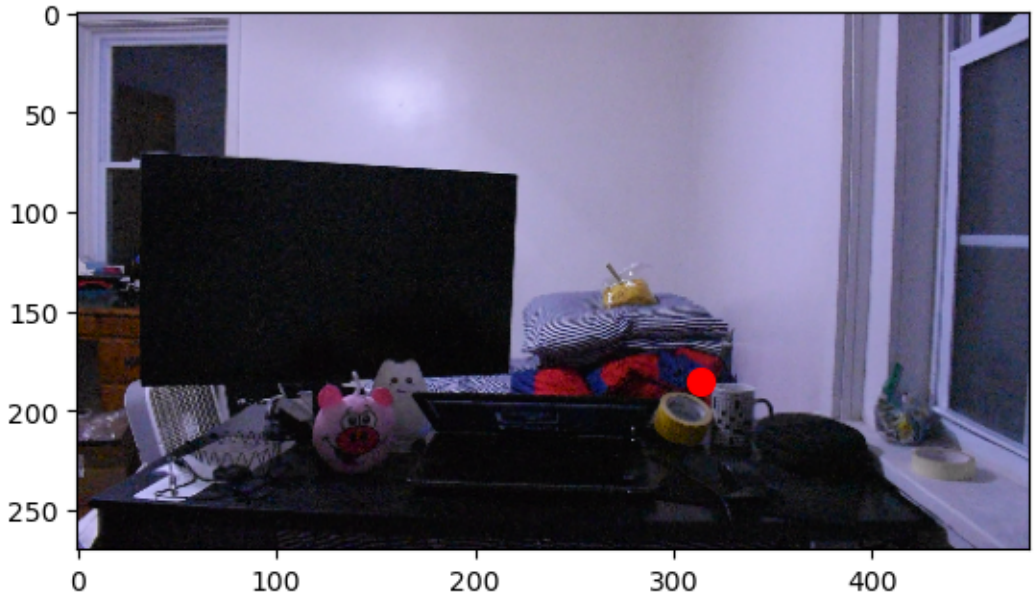




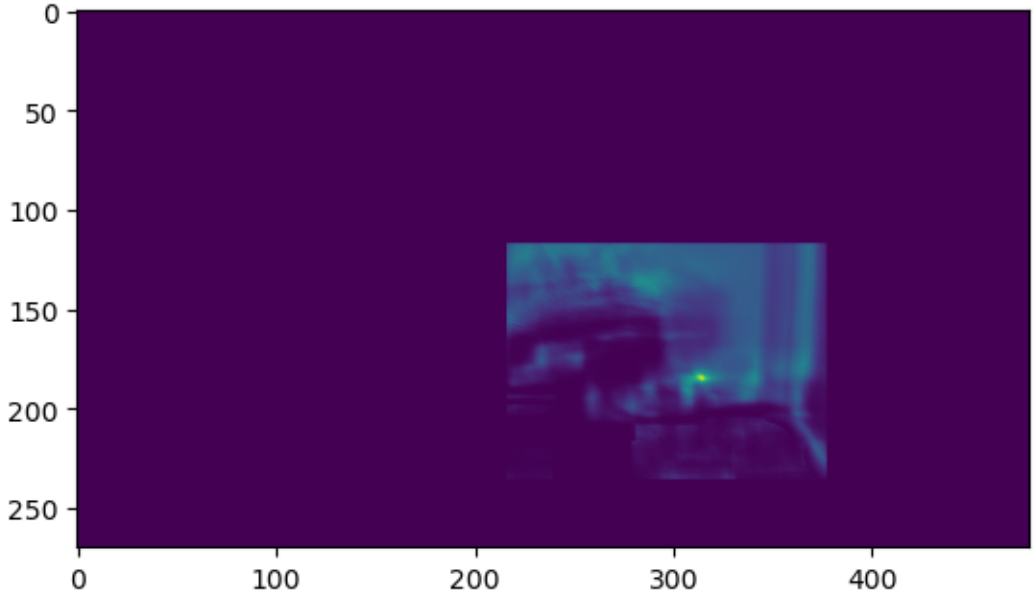
25%| | 26/102 [00:31<01:30, 1.20s/it]

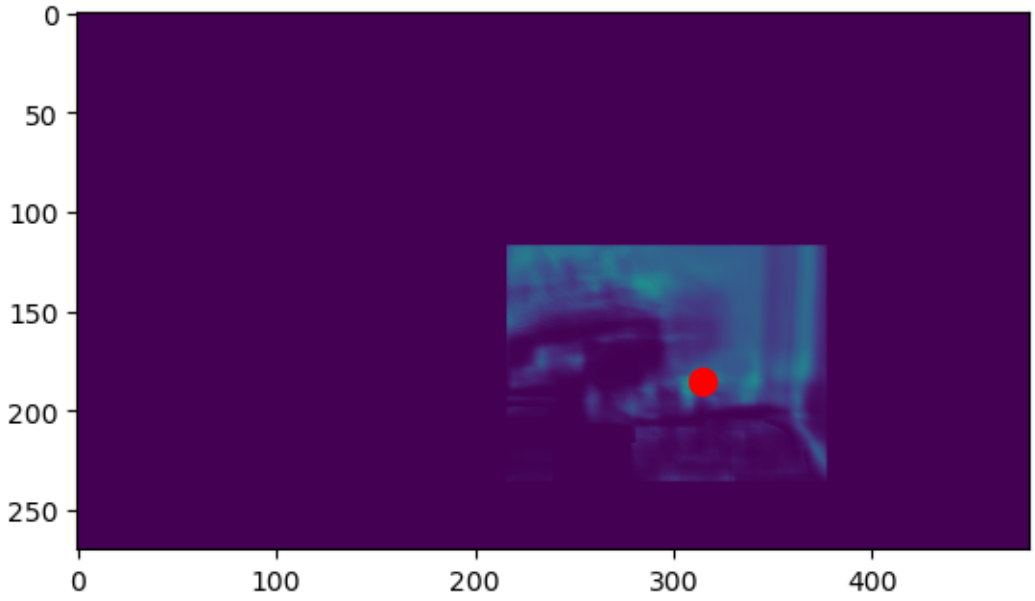

0
50
100
150
200
250
0
100
200
300
400

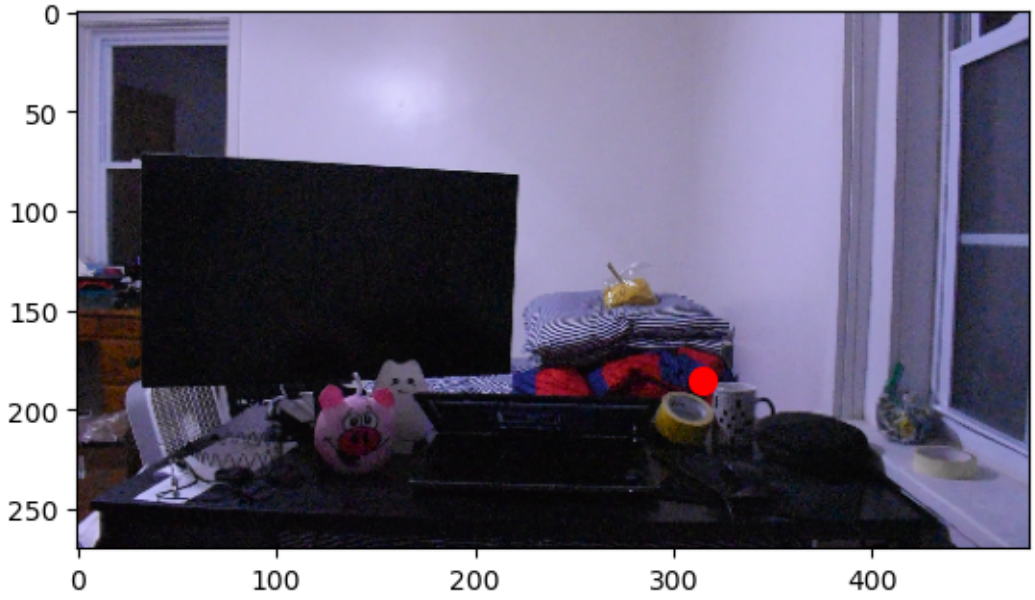

0
50
100
150
200
250
0
100
200
300
400



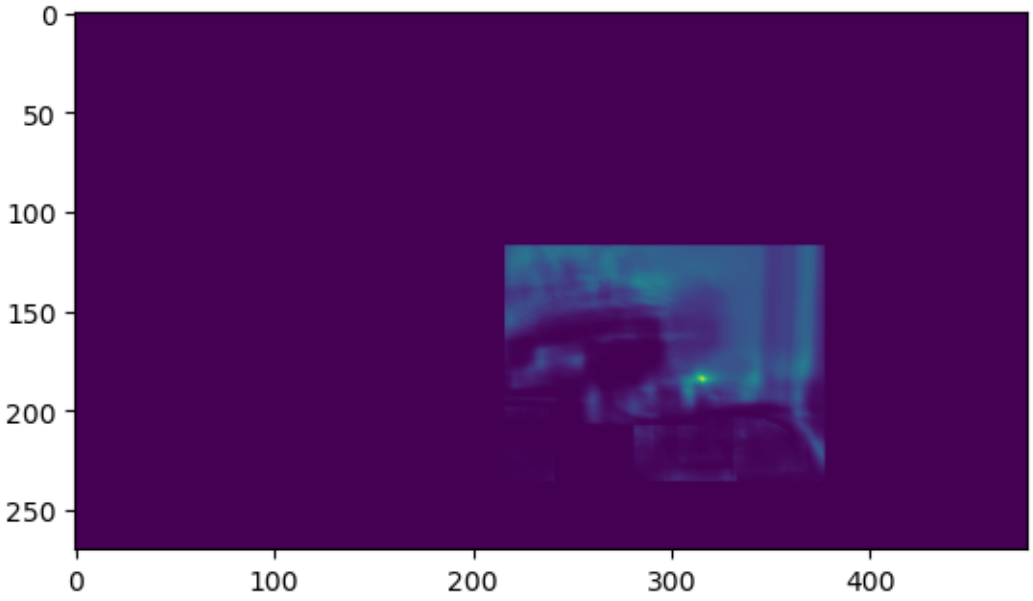
26%| | 27/102 [00:32<01:29, 1.20s/it]

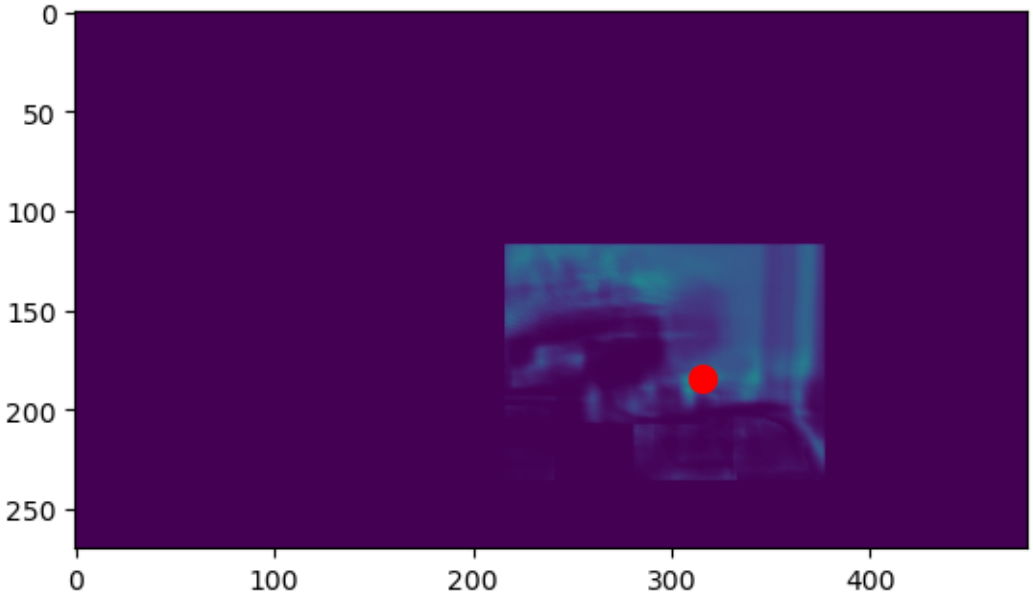


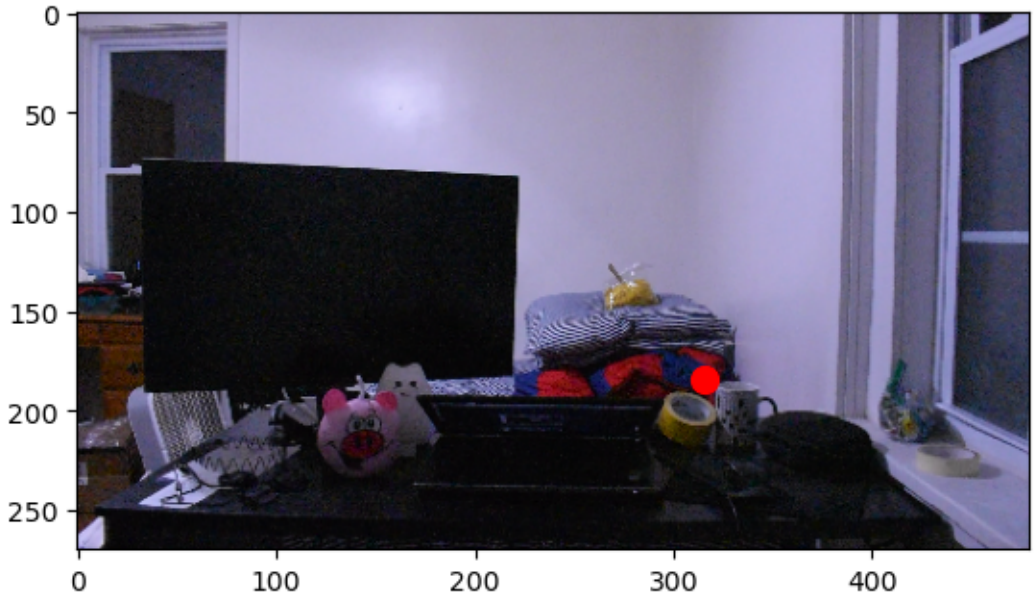




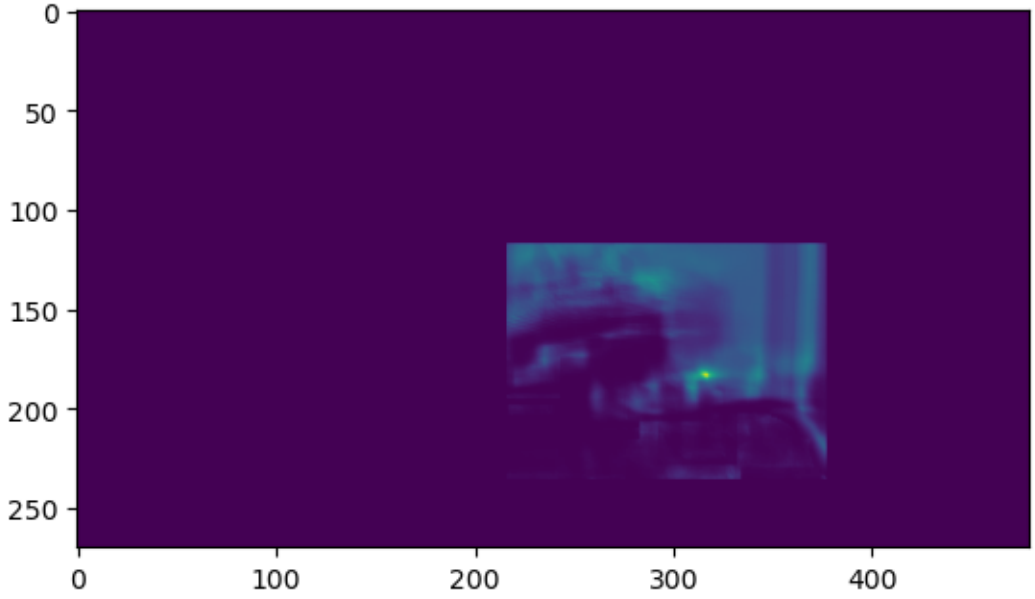
27%| | 28/102 [00:33<01:27, 1.18s/it]

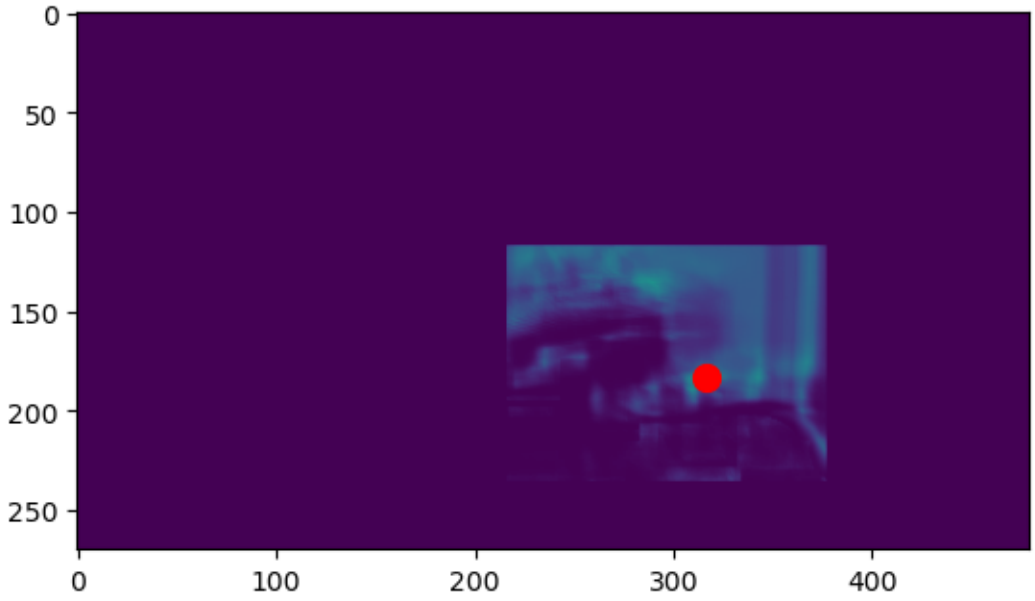

0
50
100
150
200
250
0
100
200
300
400

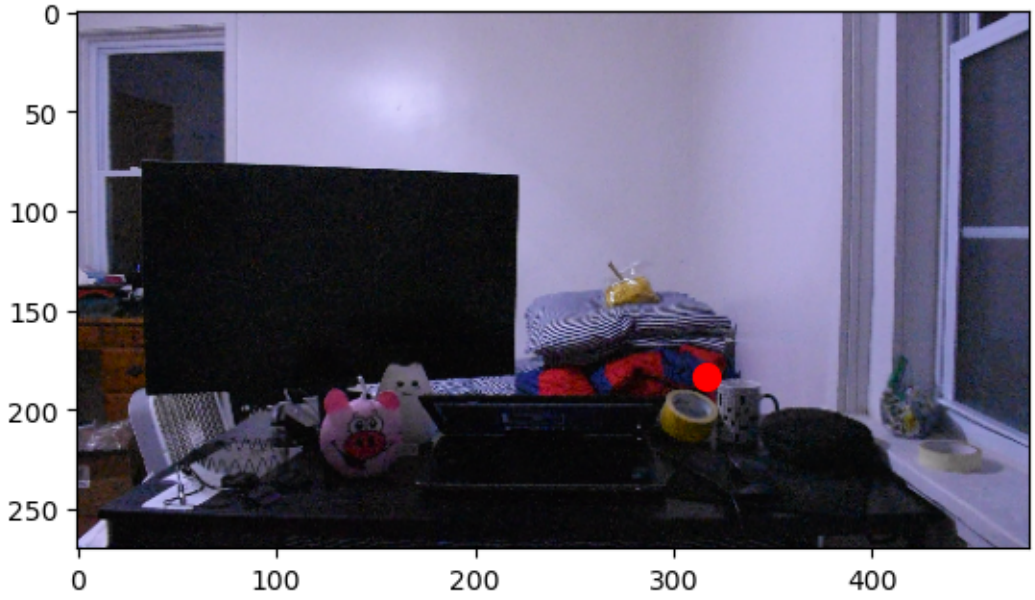

0
50
100
150
200
250
0
100
200
300
400



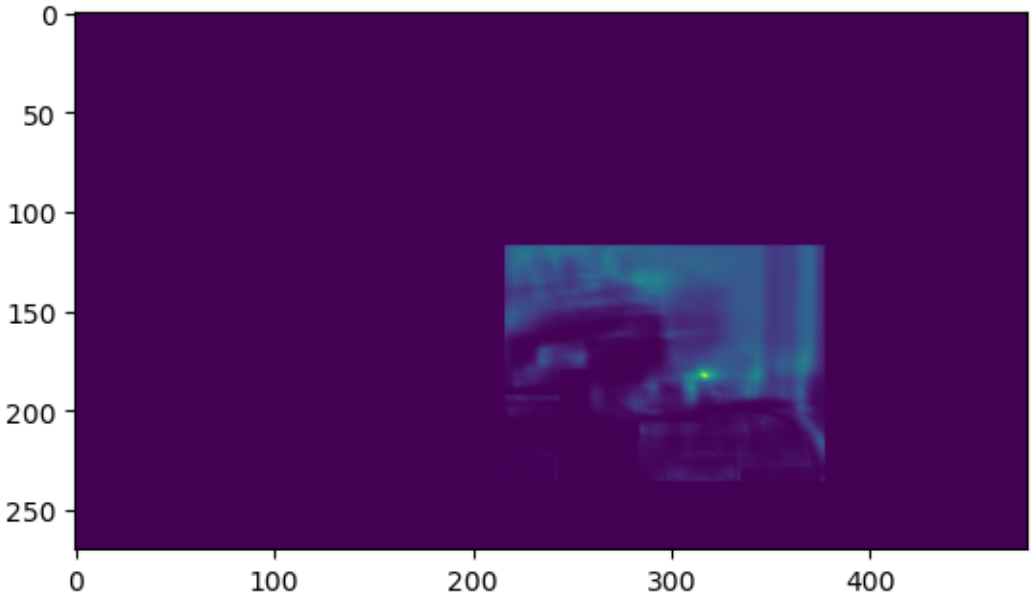
28%| | 29/102 [00:35<01:26, 1.19s/it]

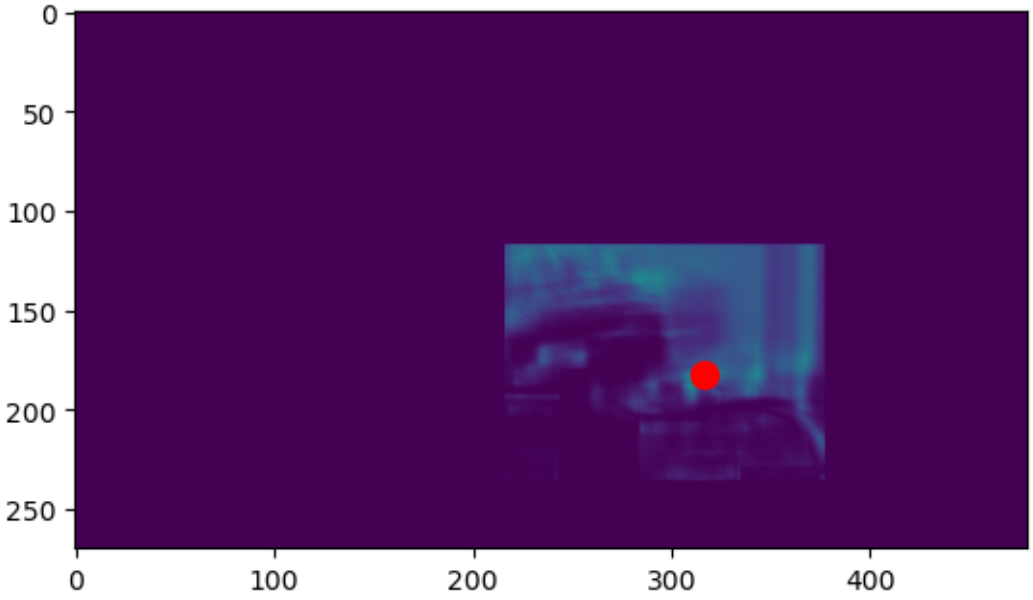


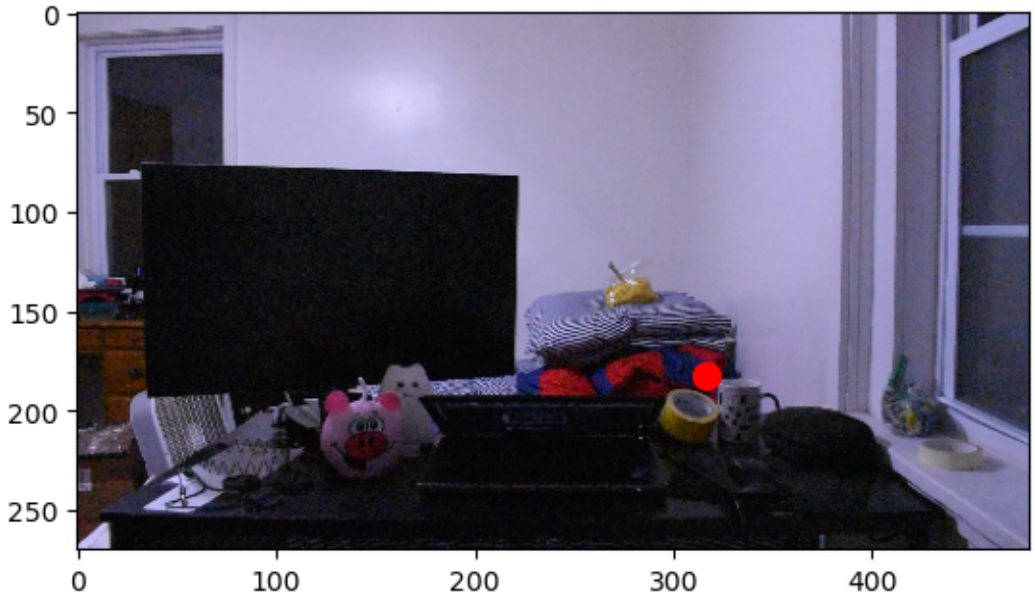




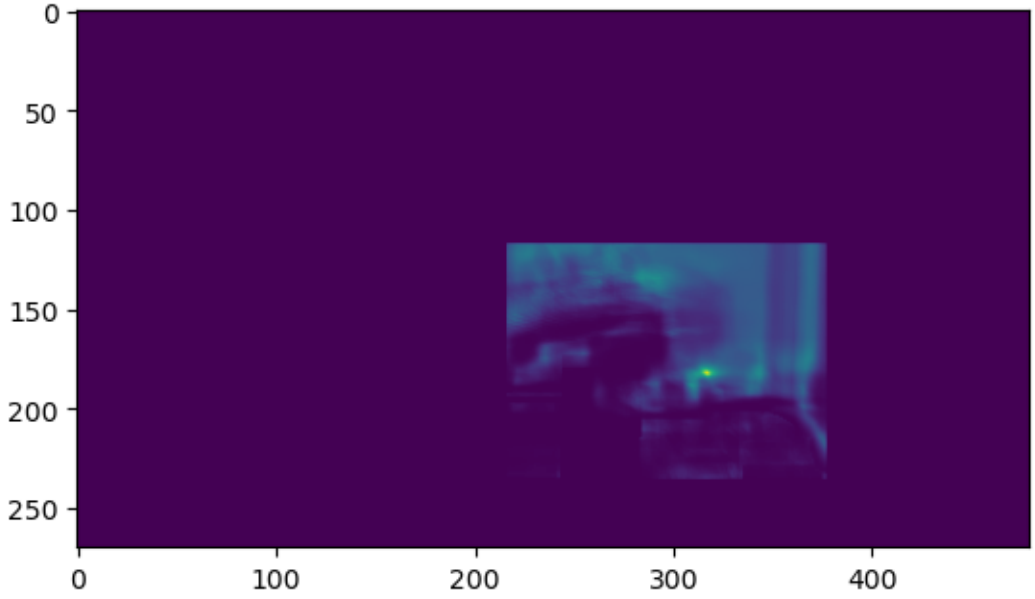
29%| | 30/102 [00:36<01:25, 1.19s/it]

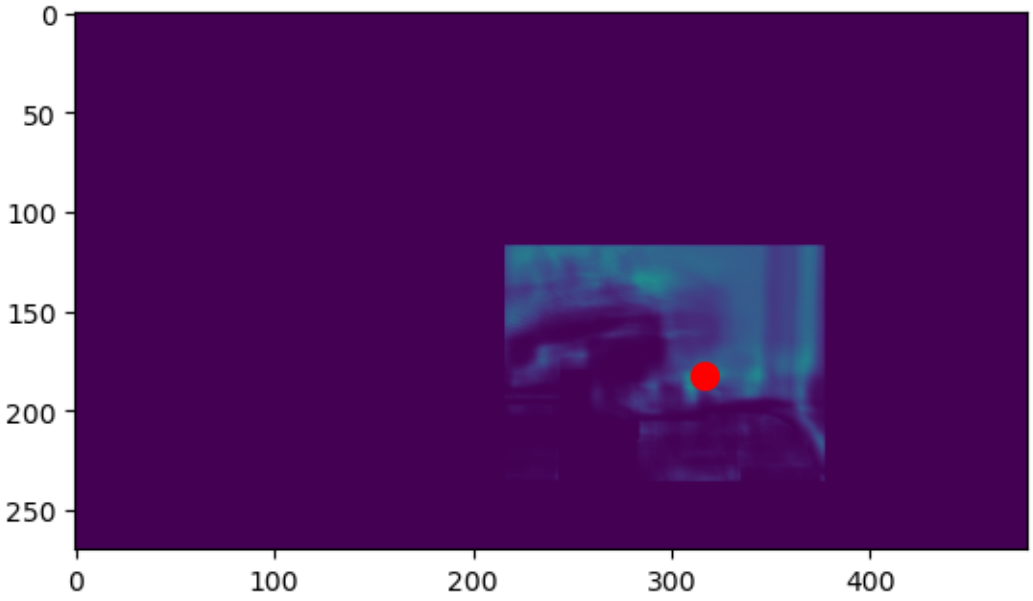

0
50
100
150
200
250
0
100
200
300
400

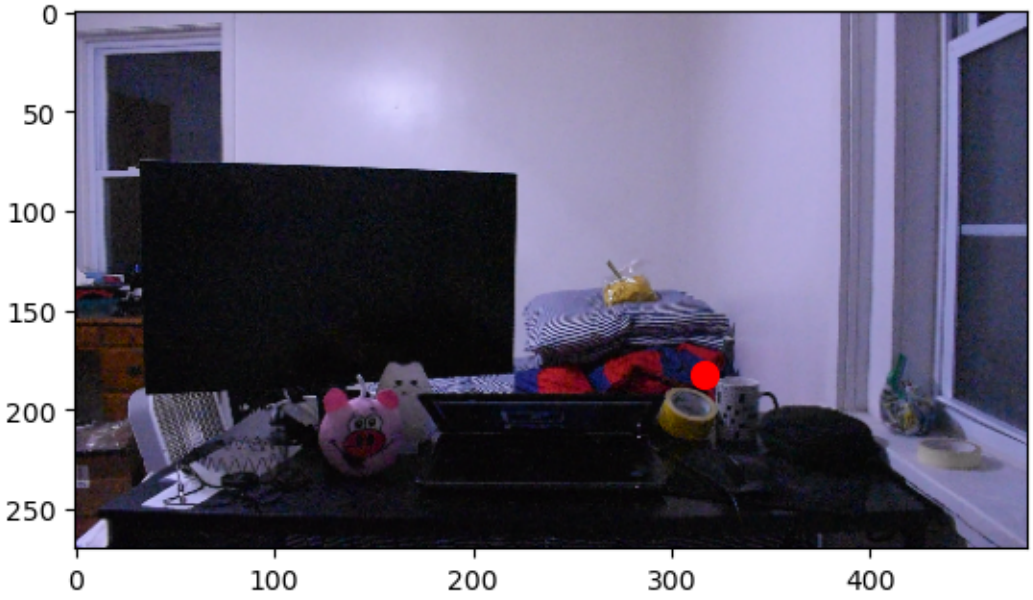

0
50
100
150
200
250
0
100
200
300
400



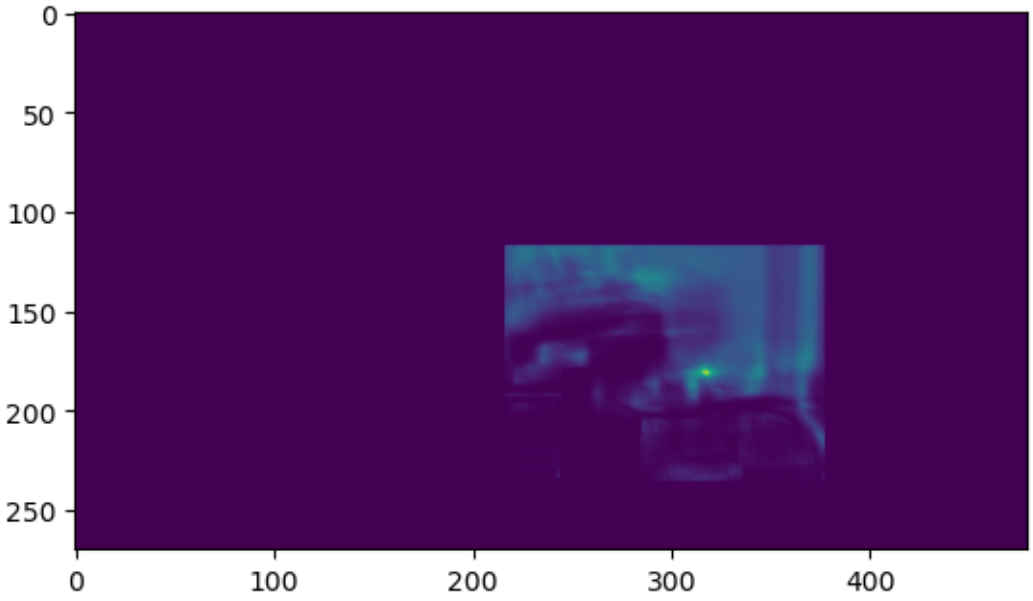
30%| | 31/102 [00:37<01:25, 1.20s/it]

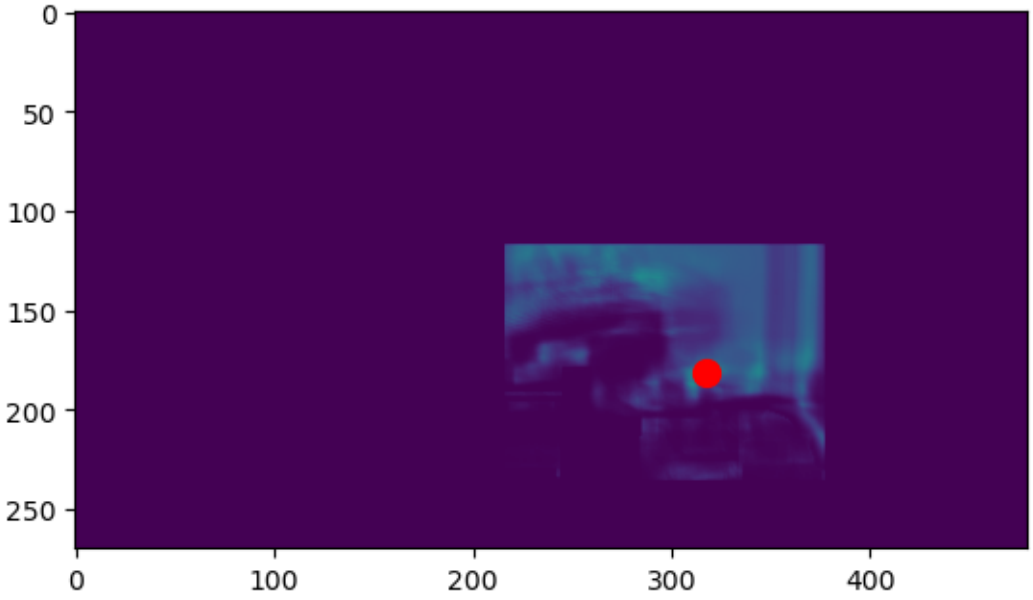


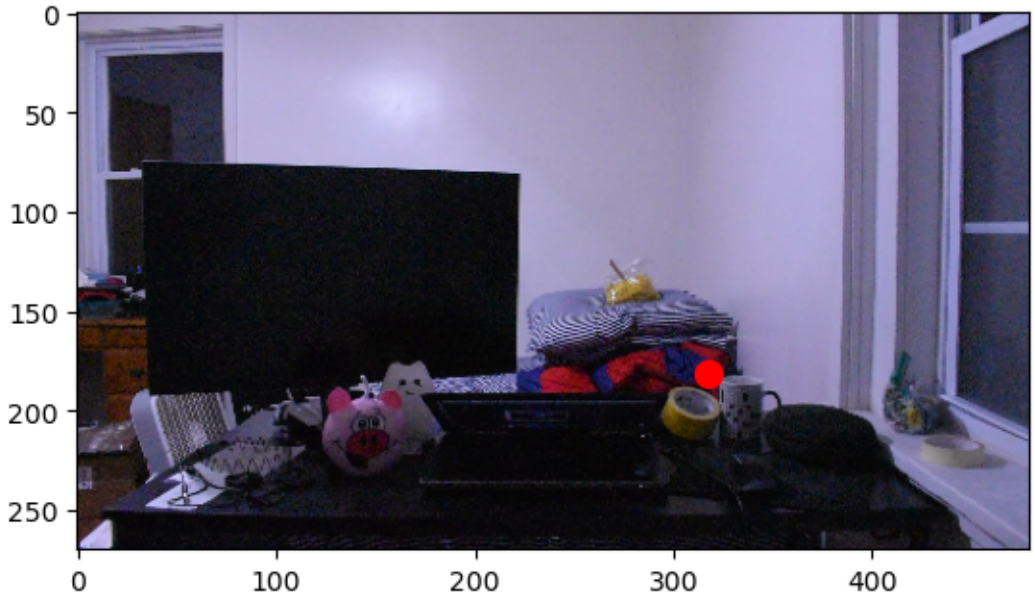




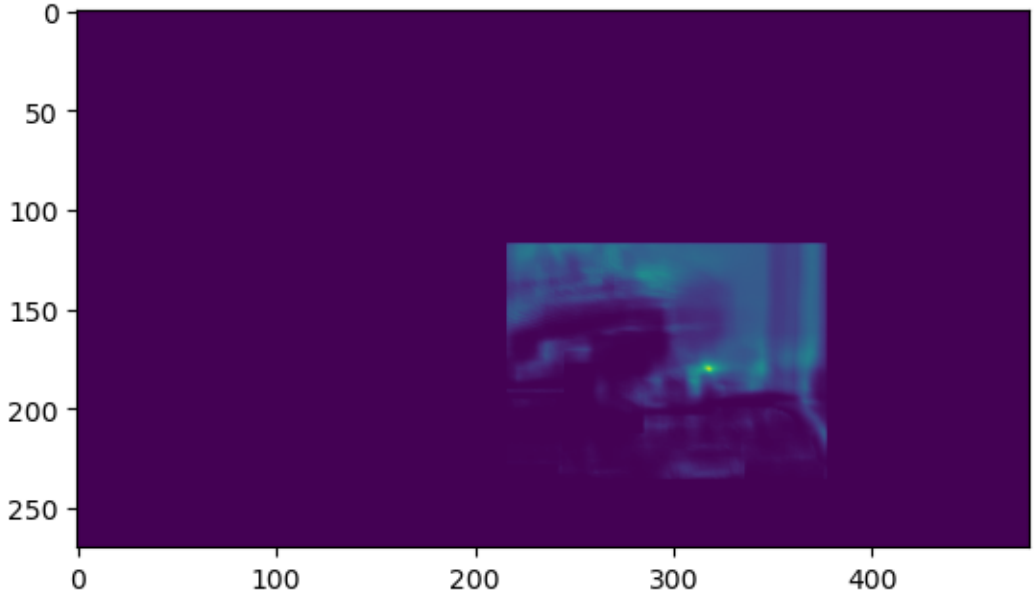
31%| | 32/102 [00:39<01:30, 1.30s/it]

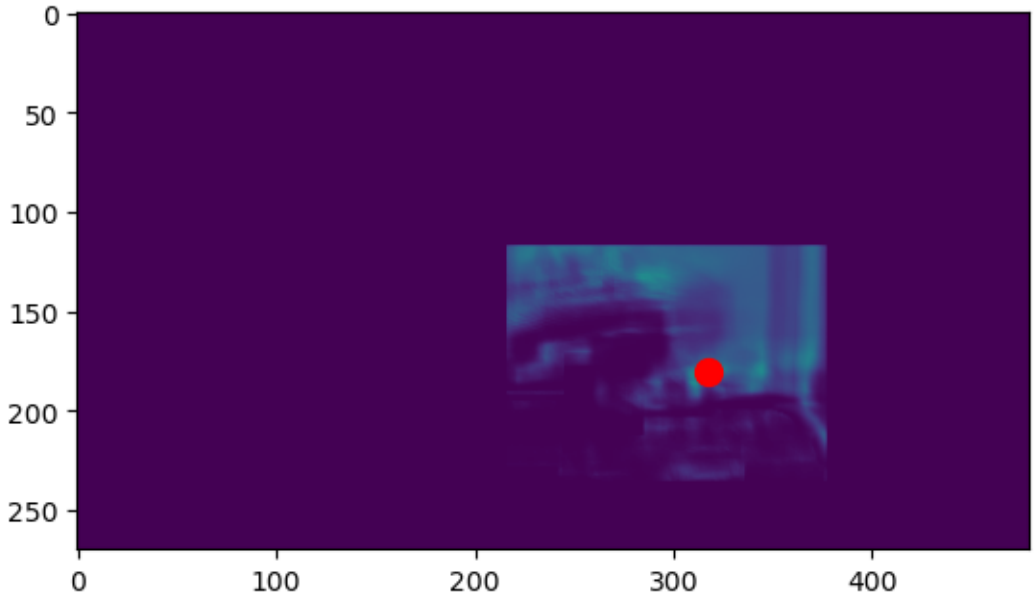

0
50
100
150
200
250
0
100
200
300
400

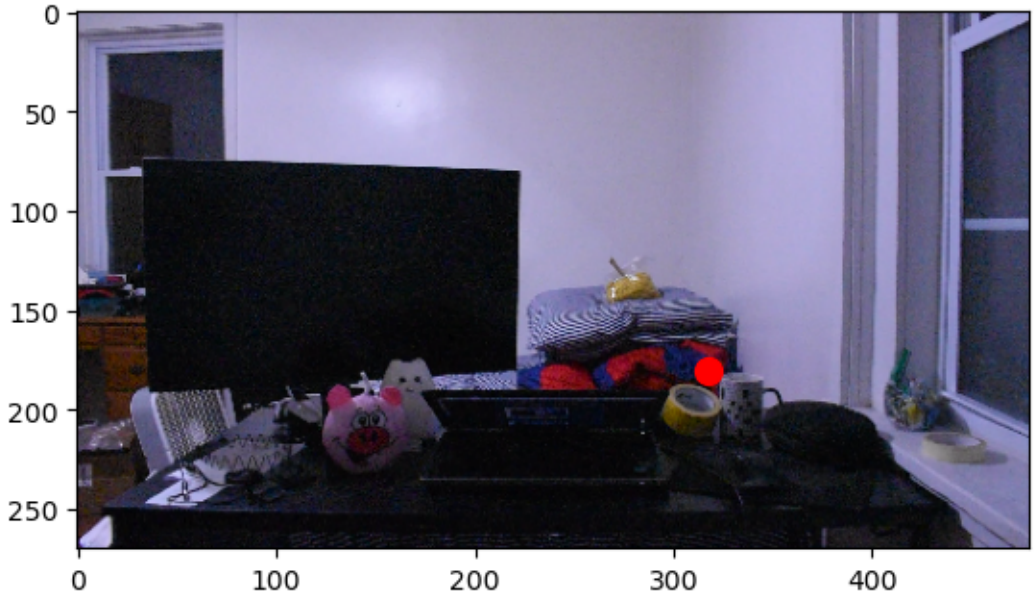

0
50
100
150
200
250
0
100
200
300
400



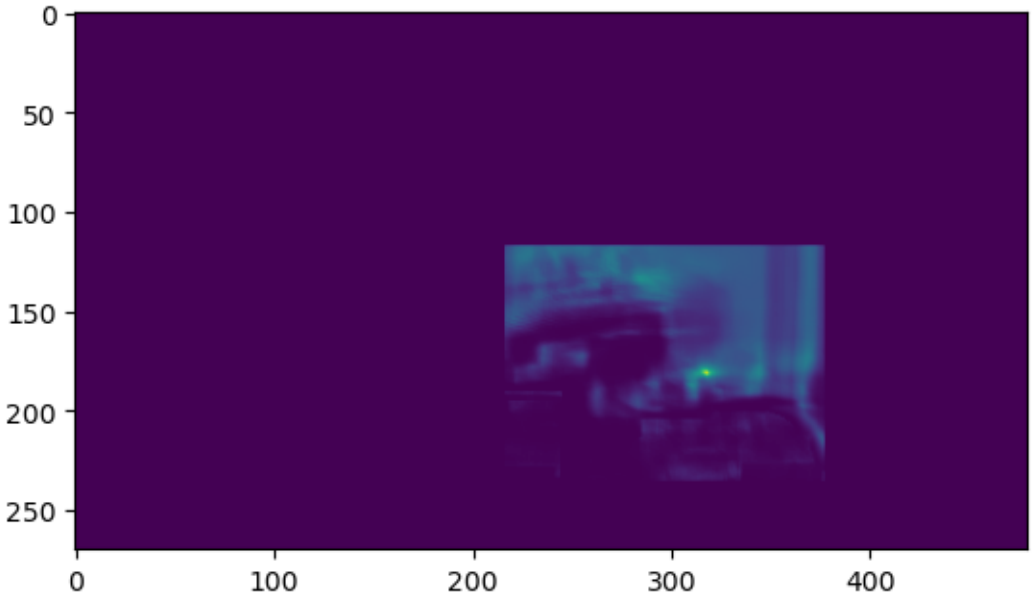
32%| | 33/102 [00:40<01:27, 1.26s/it]

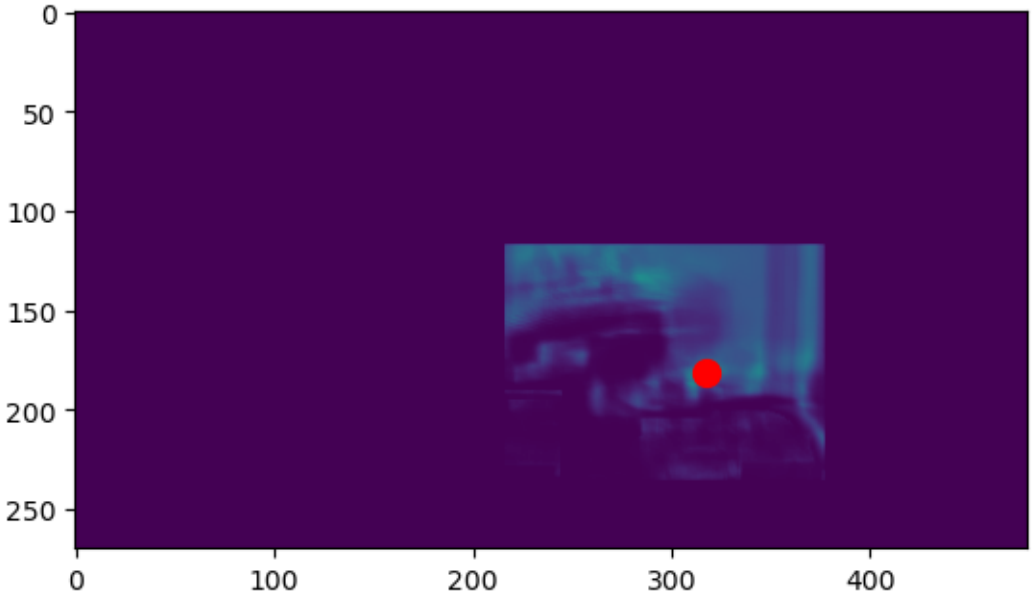


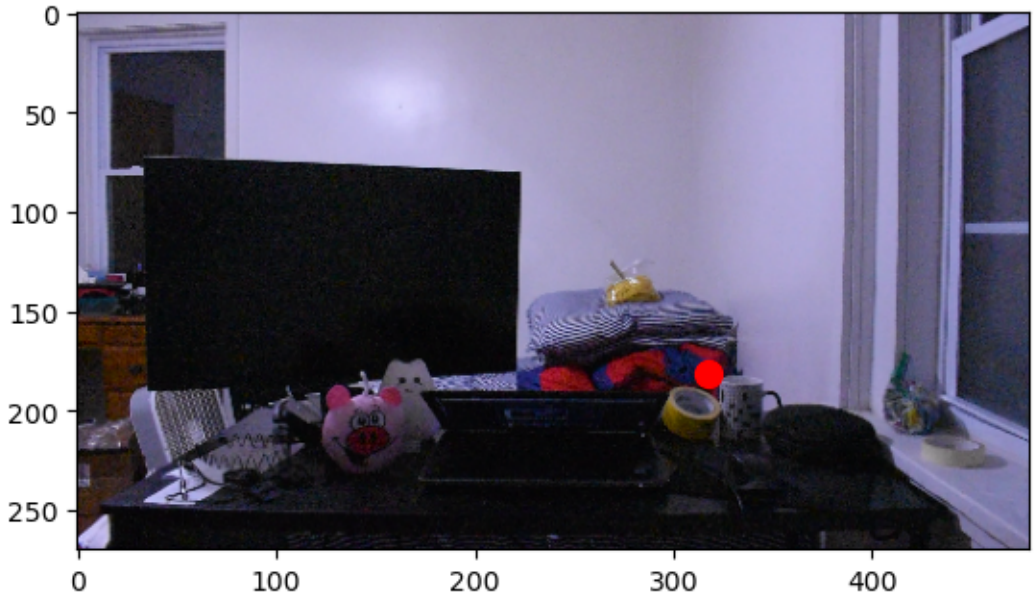




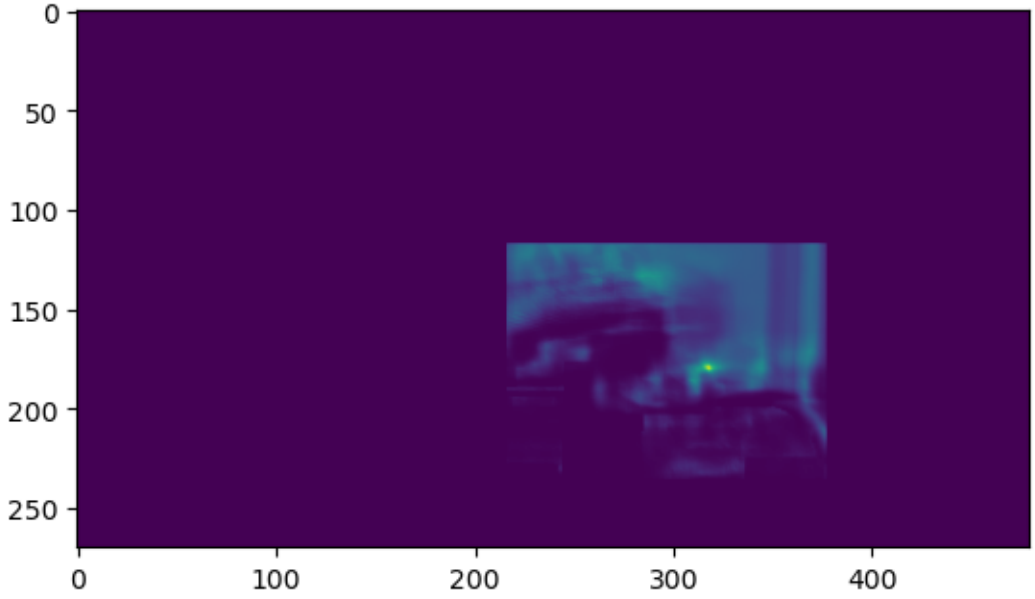
33%| | 34/102 [00:41<01:24, 1.24s/it]

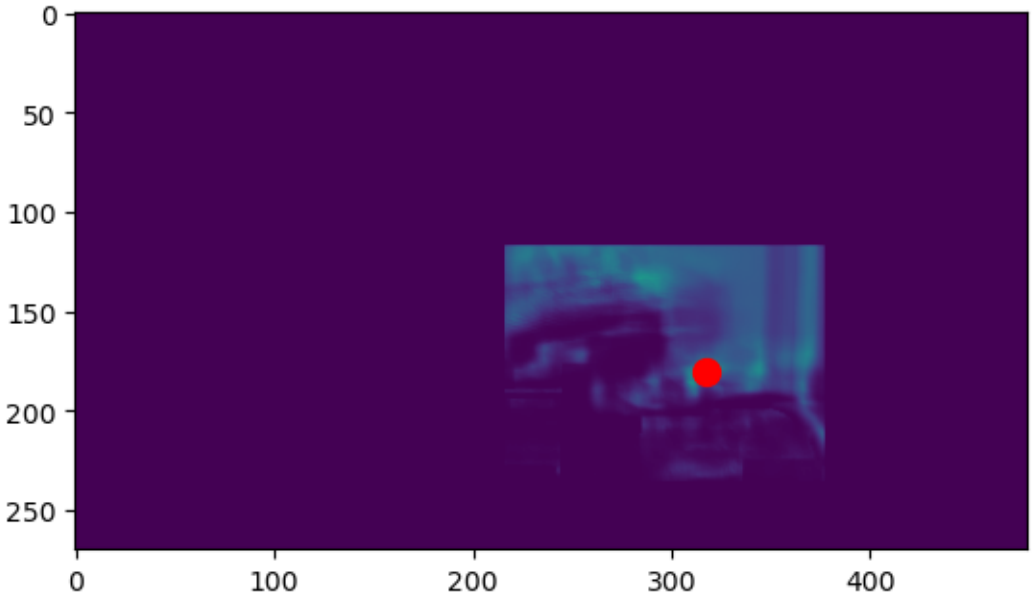

0
50
100
150
200
250
0
100
200
300
400

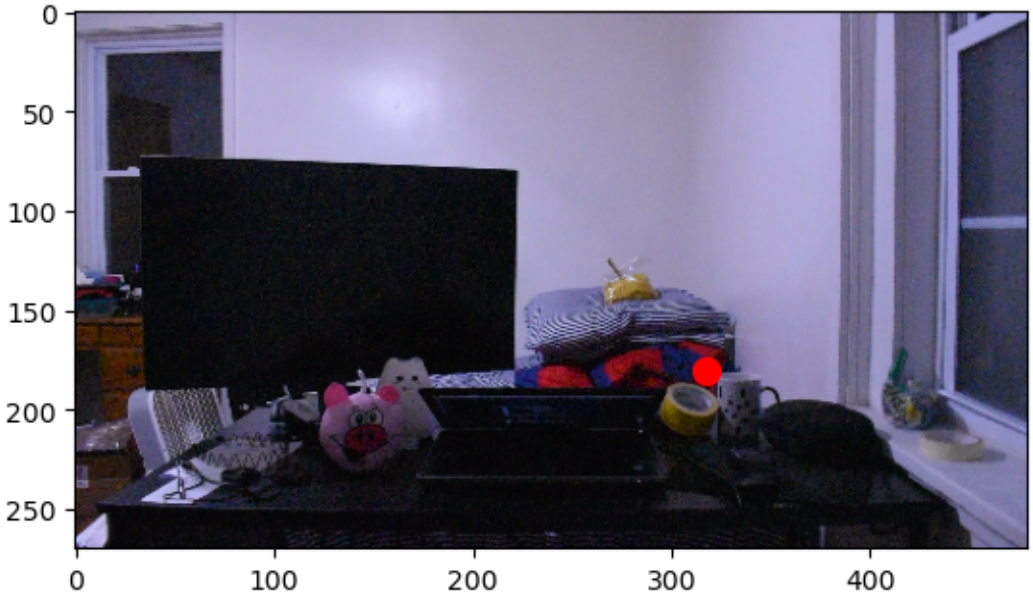

0
50
100
150
200
250
0
100
200
300
400



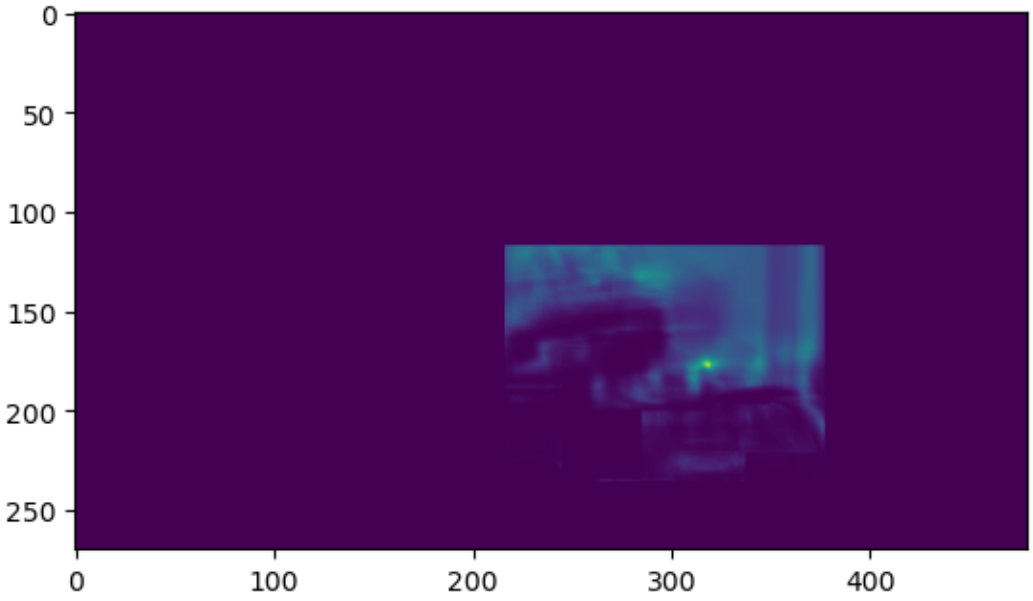
34%| | 35/102 [00:42<01:21, 1.22s/it]

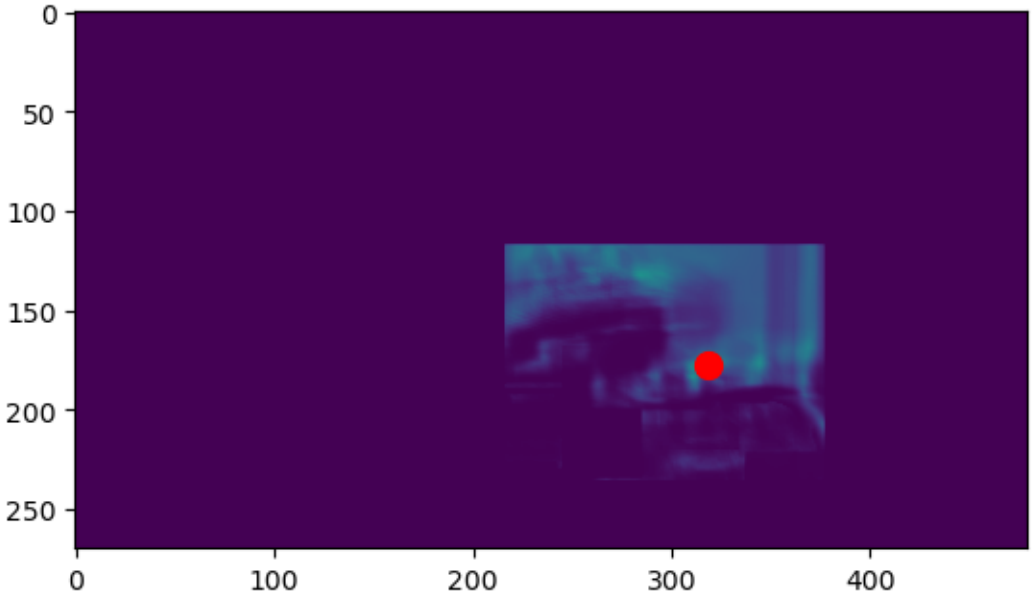


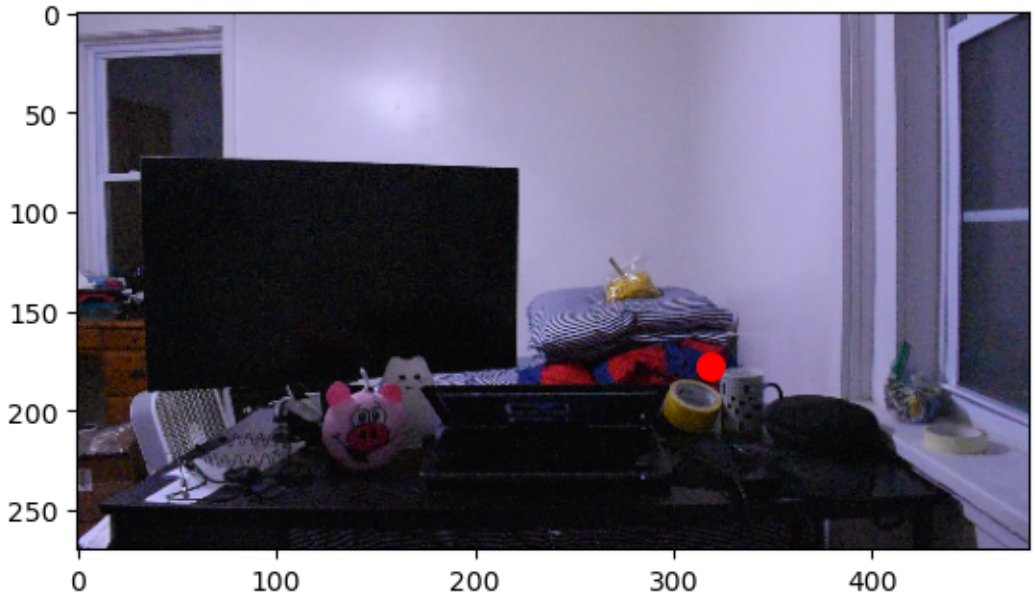




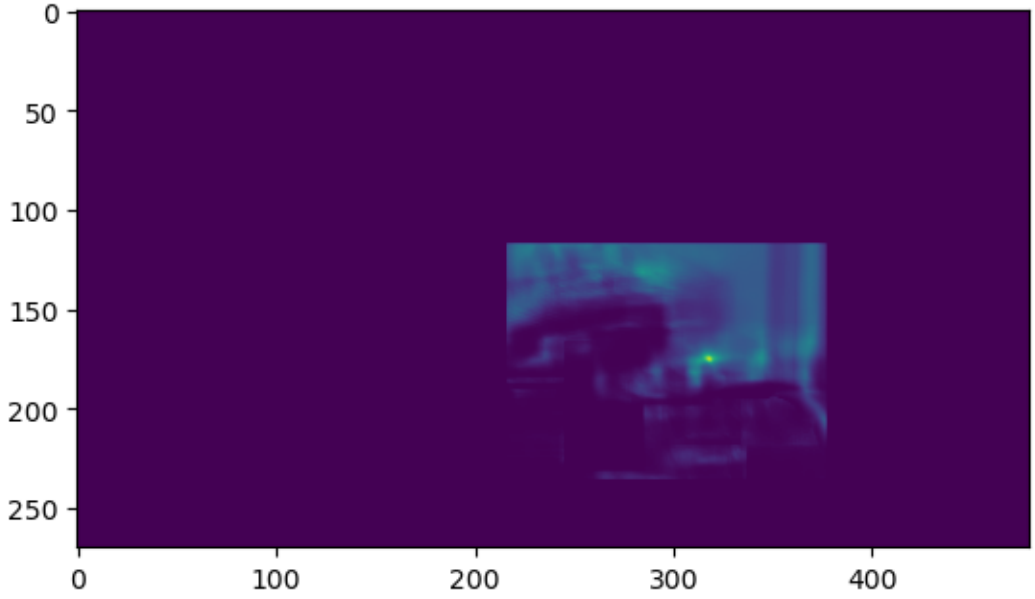
35%| | 36/102 [00:43<01:19, 1.21s/it]

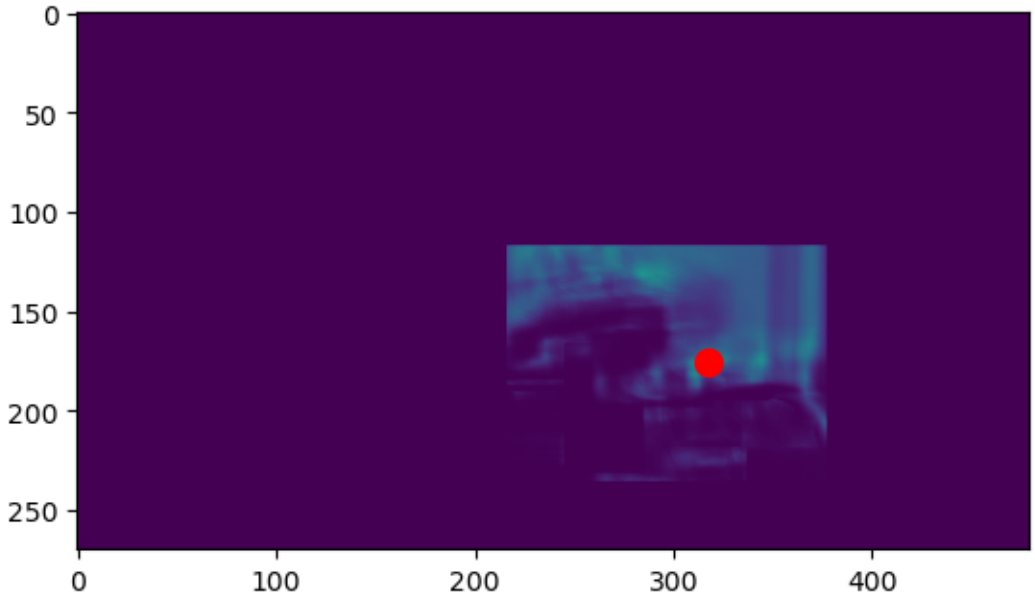

0
50
100
150
200
250
0
100
200
300
400

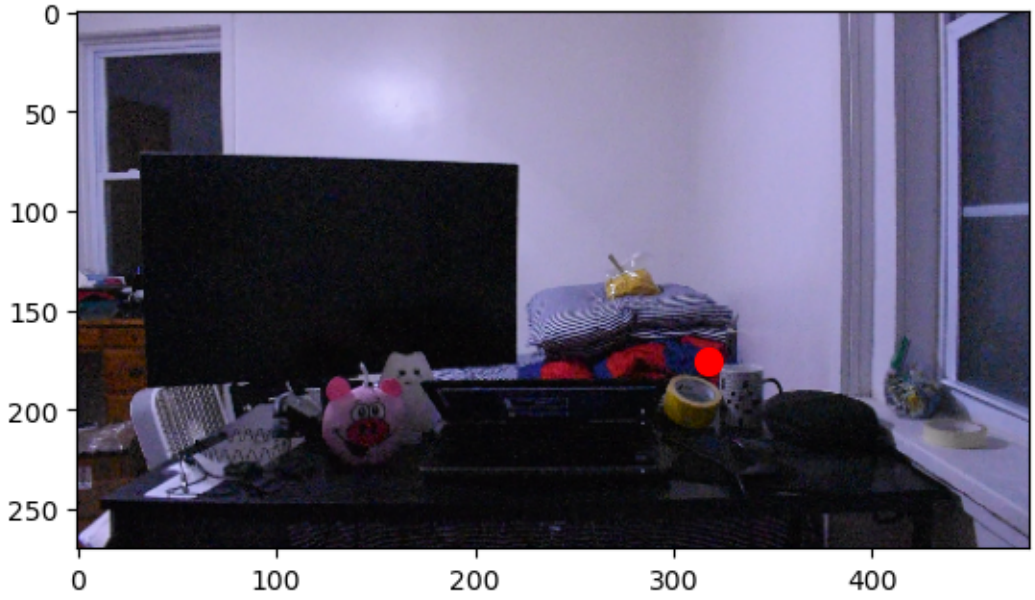

0
50
100
150
200
250
0
100
200
300
400



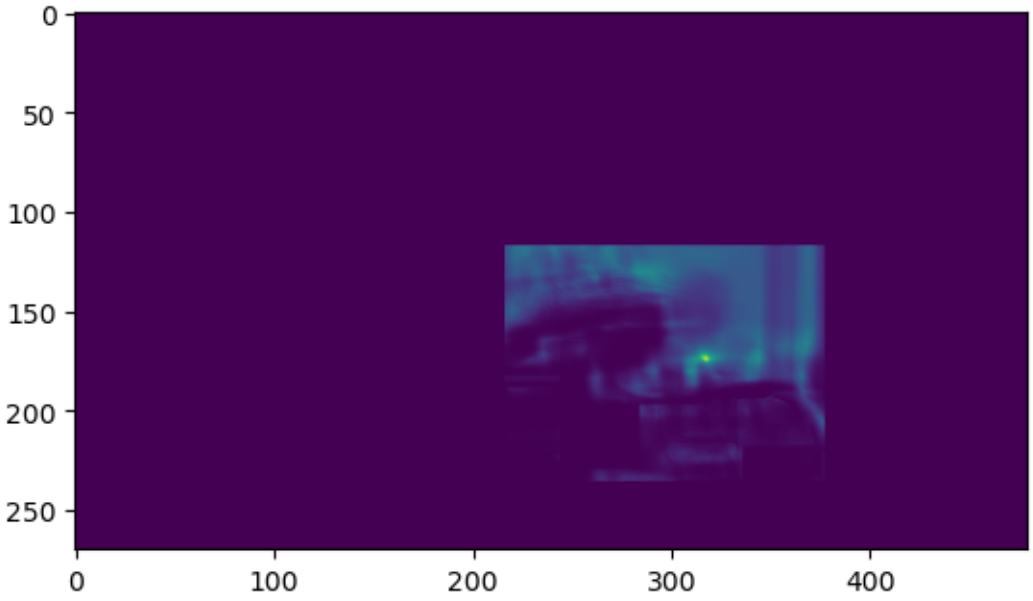
36%| | 37/102 [00:44<01:18, 1.21s/it]

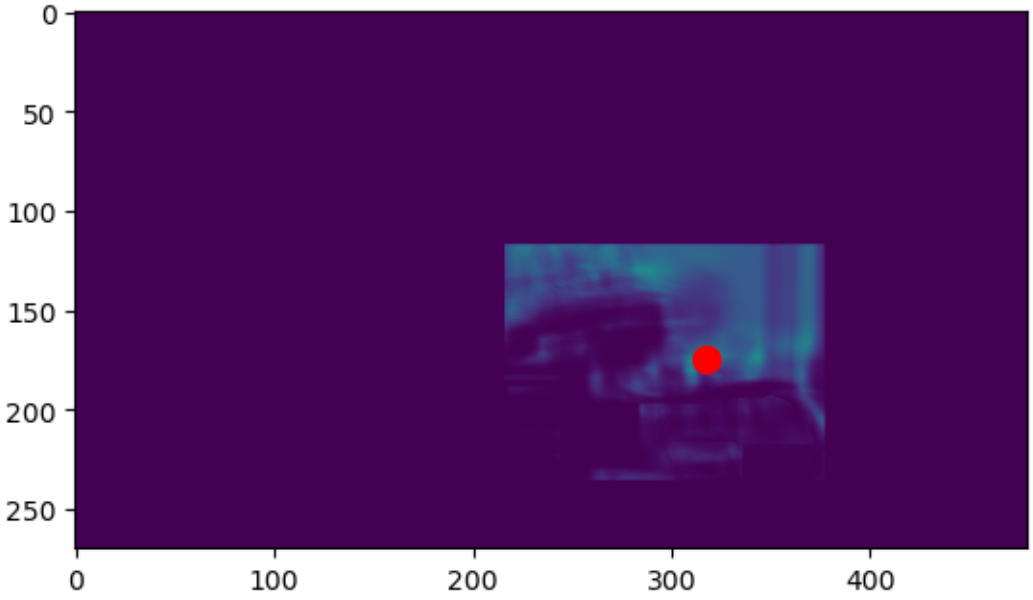


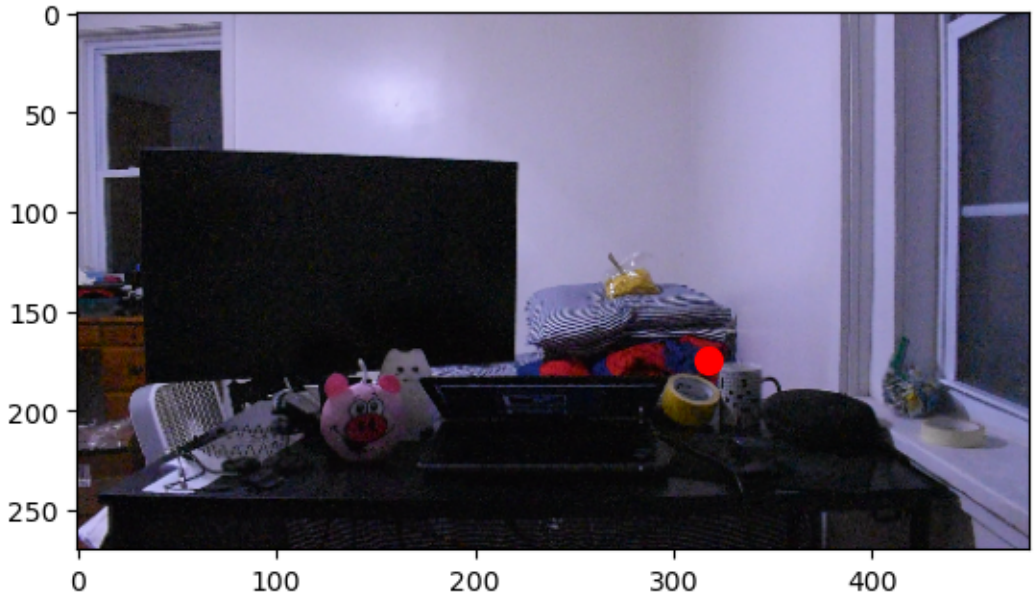




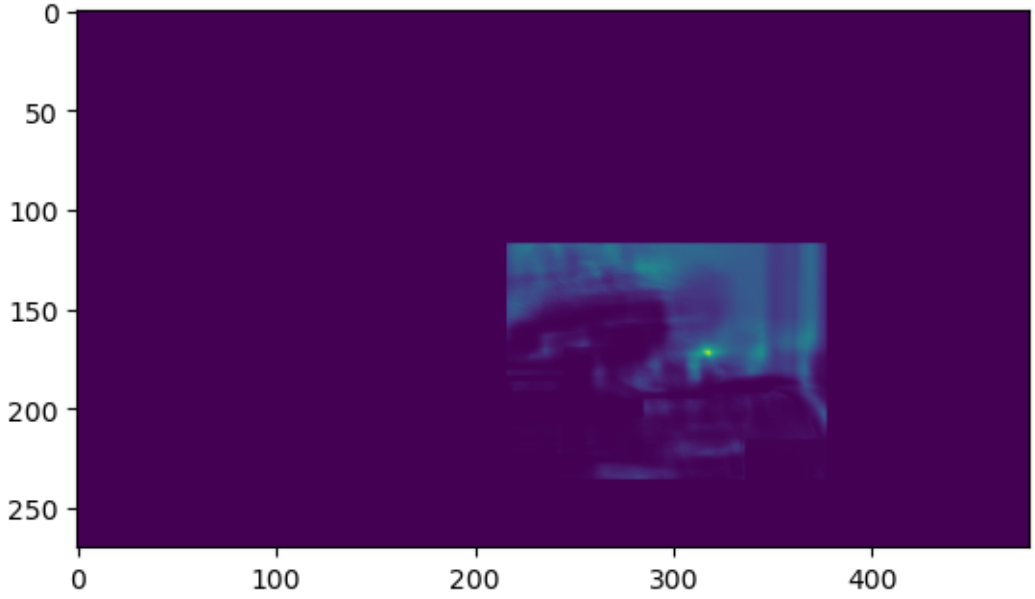
37%| | 38/102 [00:46<01:16, 1.20s/it]

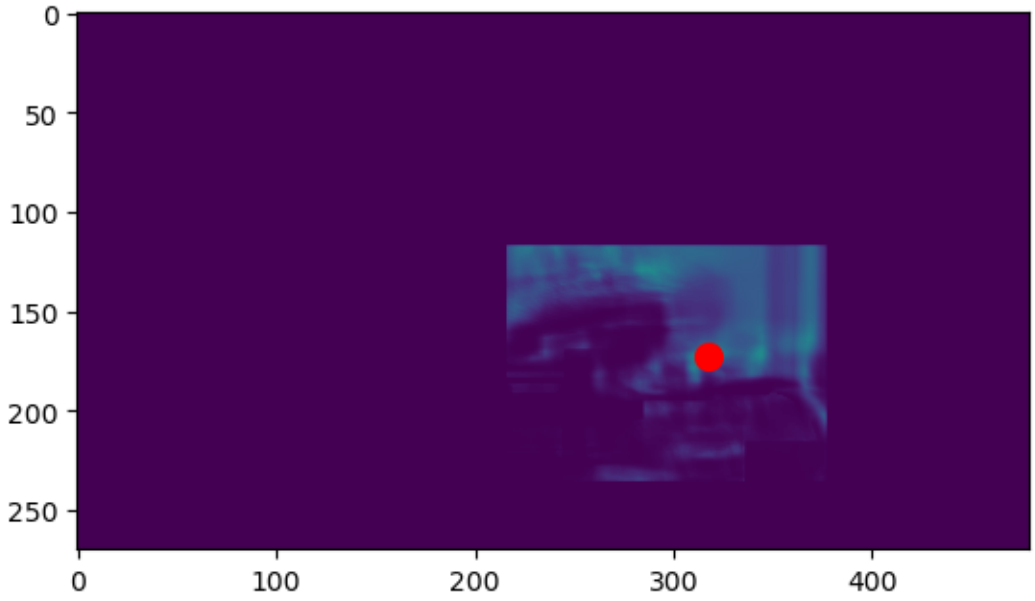

0
50
100
150
200
250
0
100
200
300
400

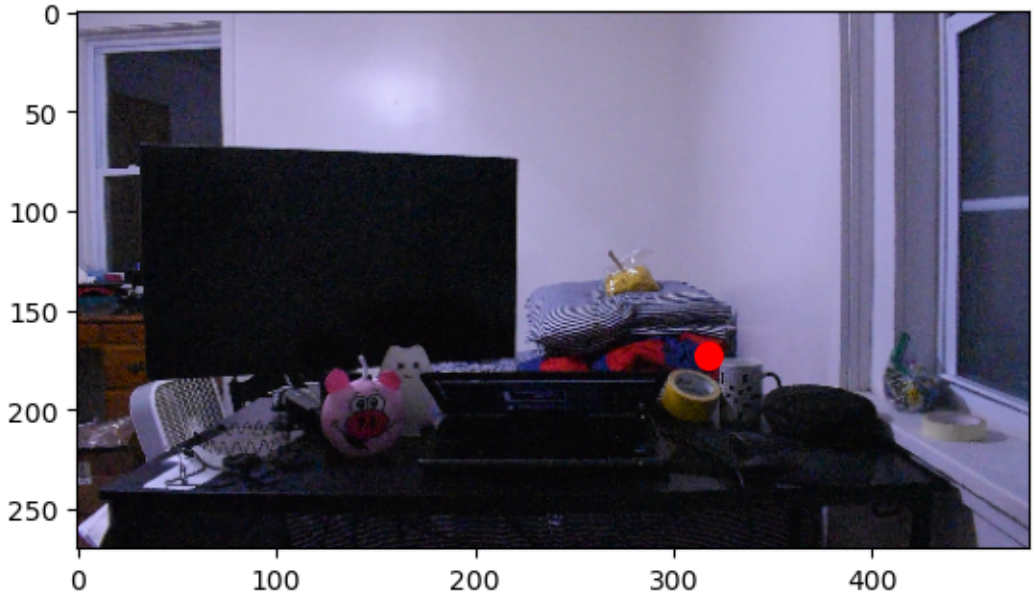

0
50
100
150
200
250
0
100
200
300
400



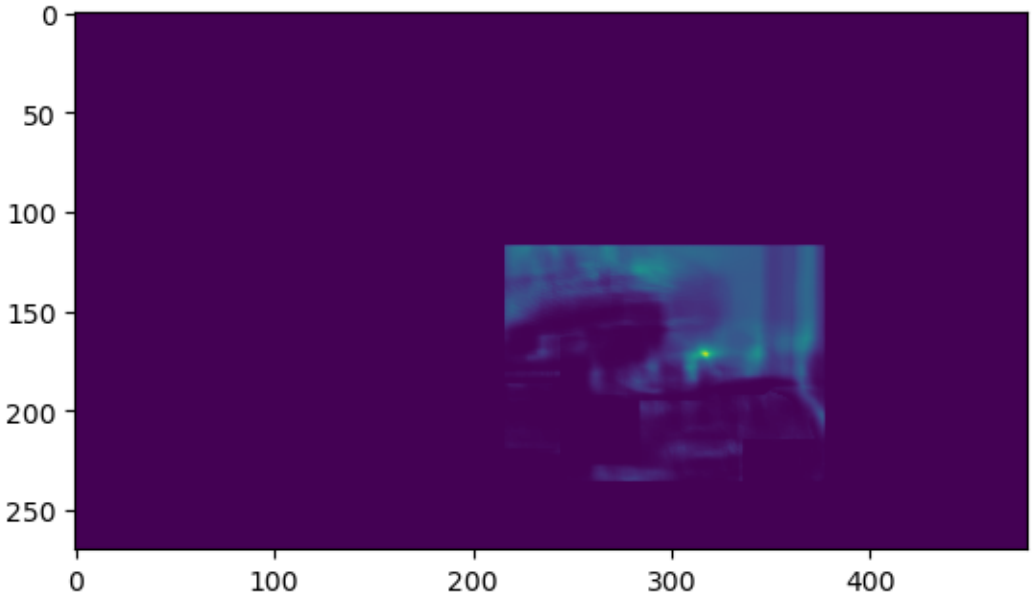
38%| | 39/102 [00:47<01:15, 1.20s/it]

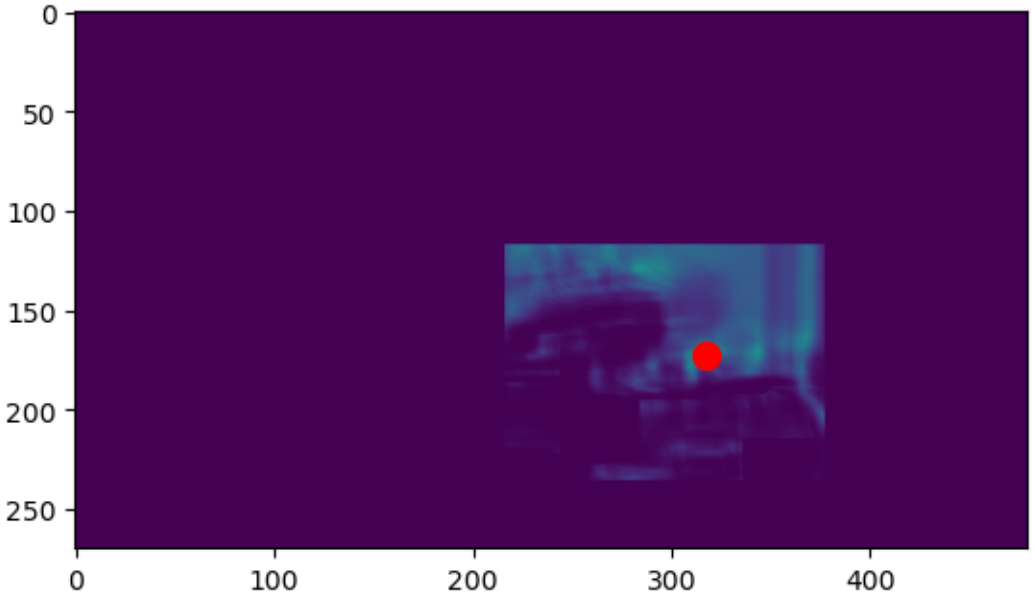


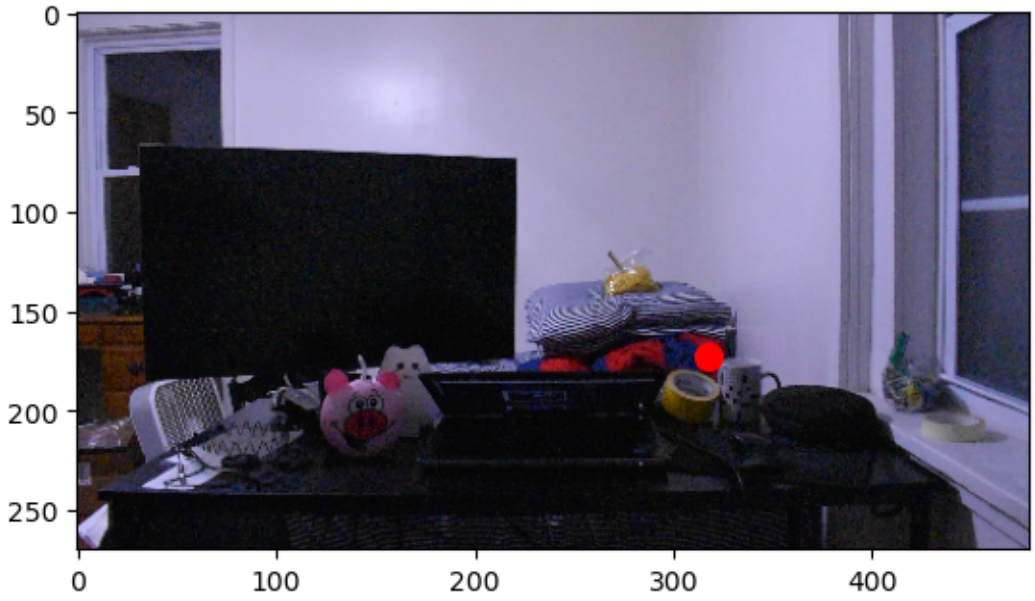




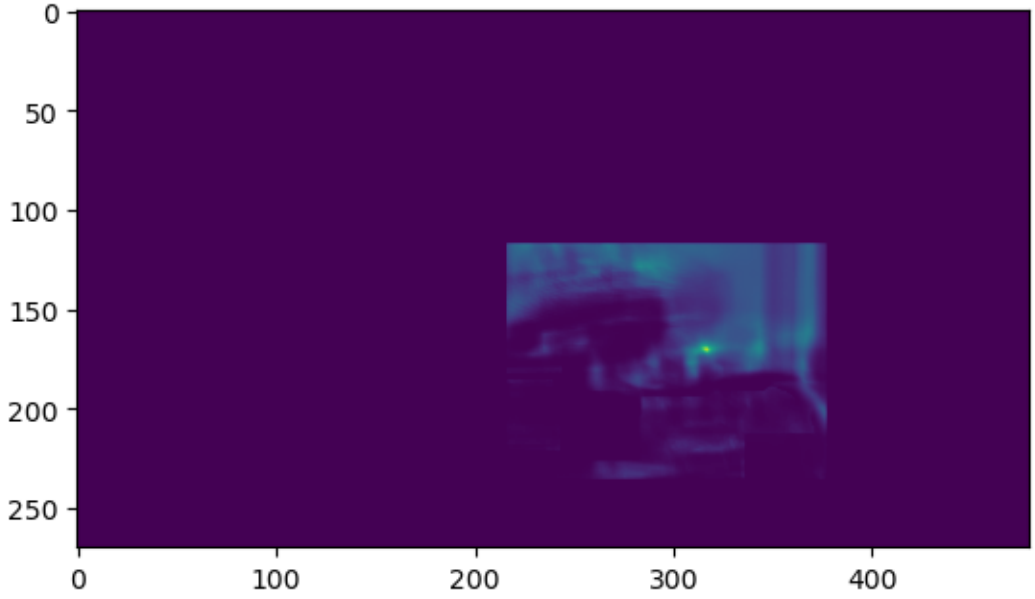
39%| | 40/102 [00:48<01:14, 1.20s/it]

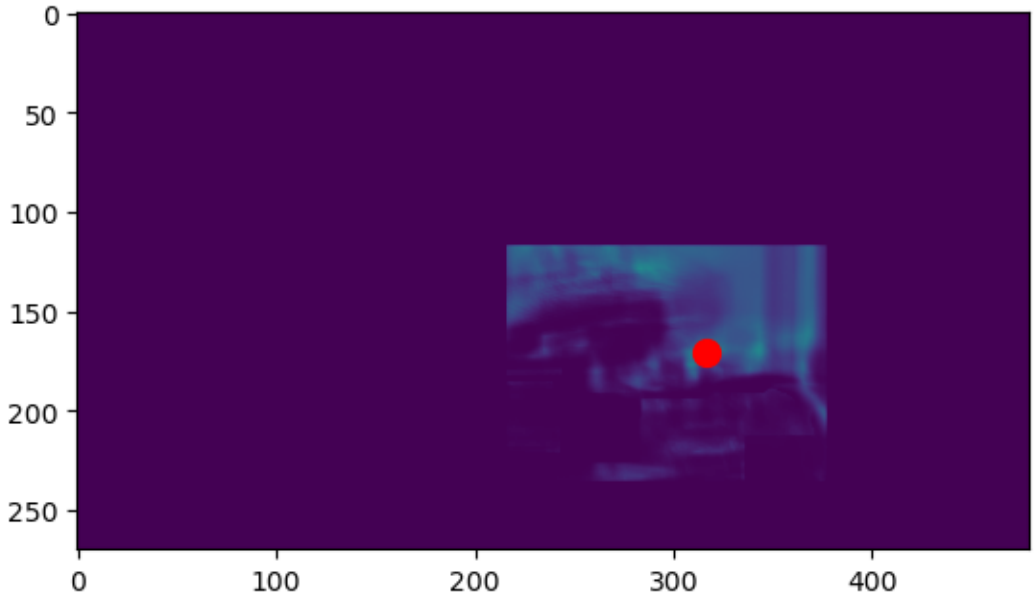

0
50
100
150
200
250
0
100
200
300
400

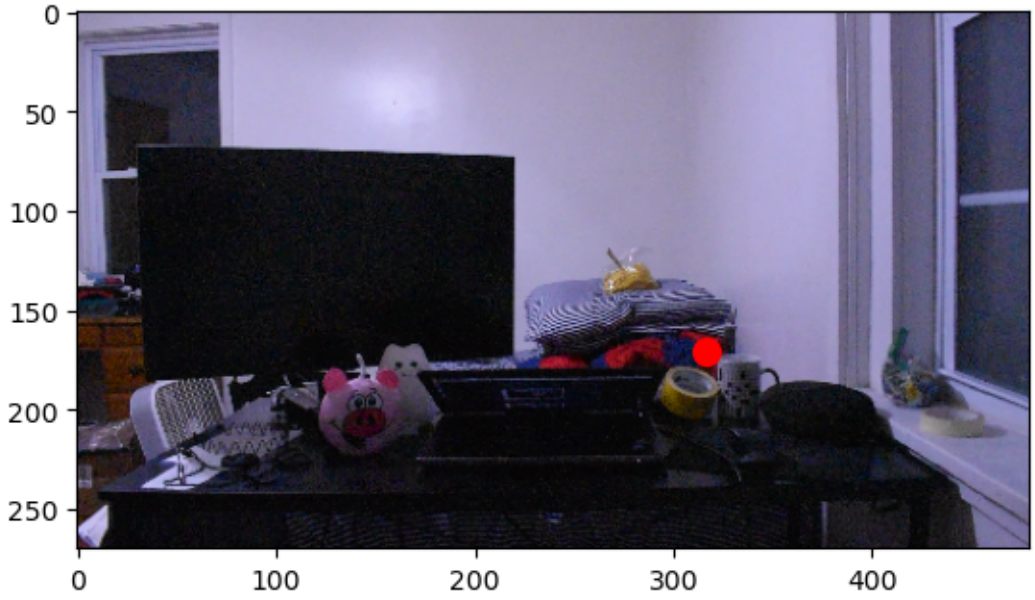

0
50
100
150
200
250
0
100
200
300
400



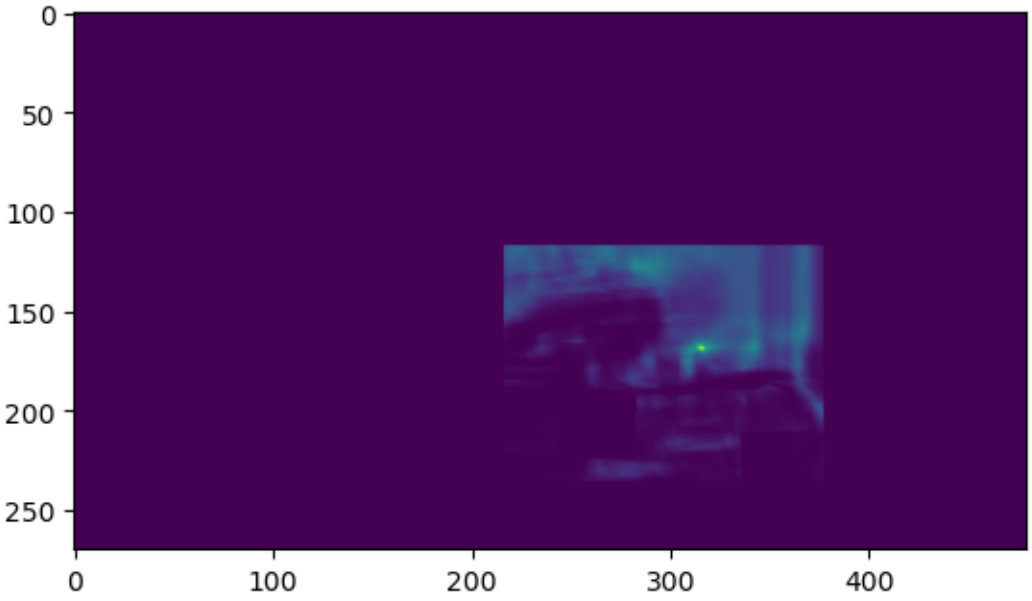
40%| | 41/102 [00:49<01:12, 1.19s/it]

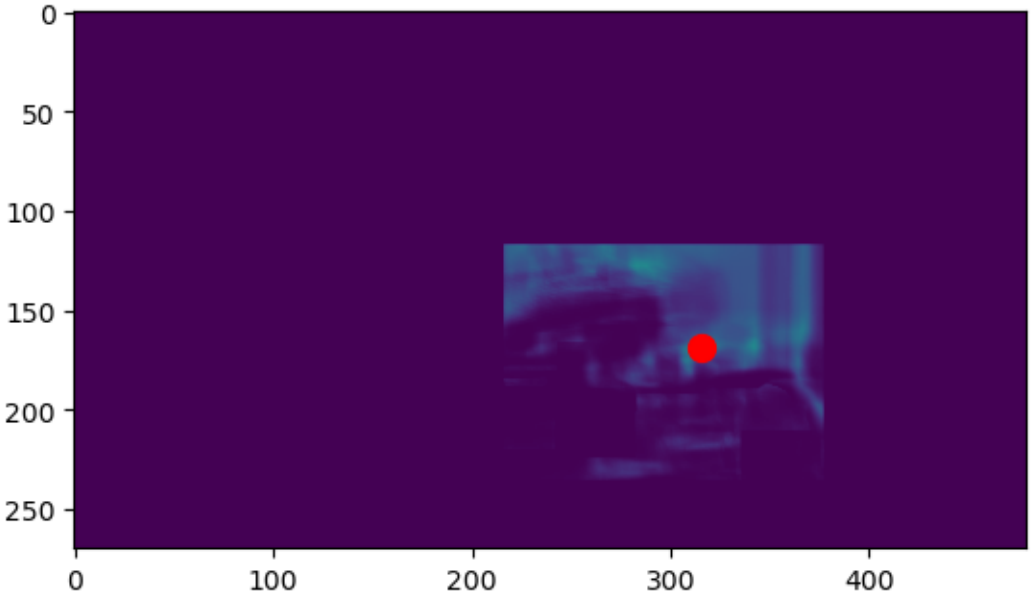


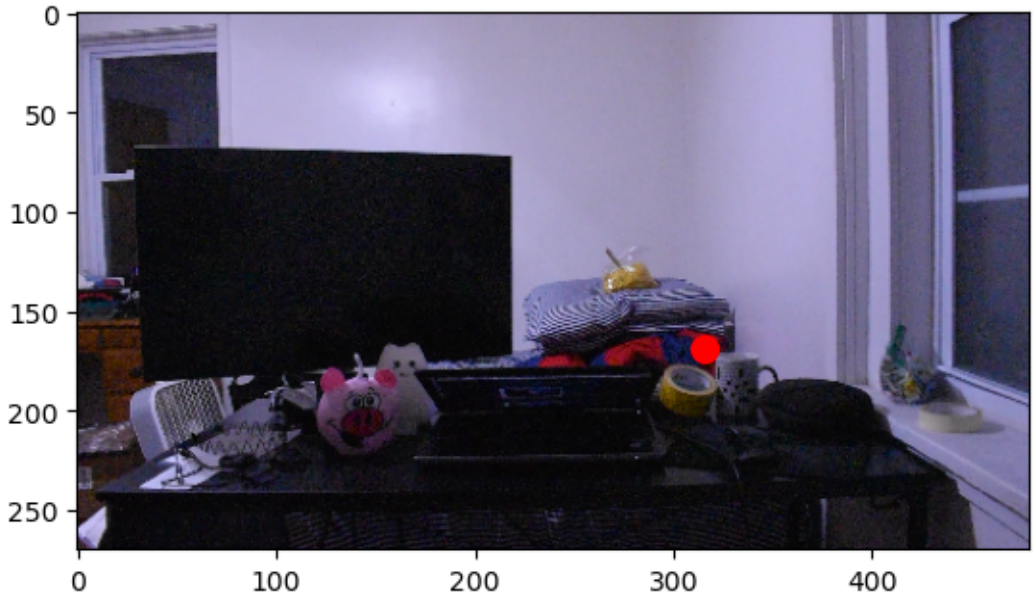




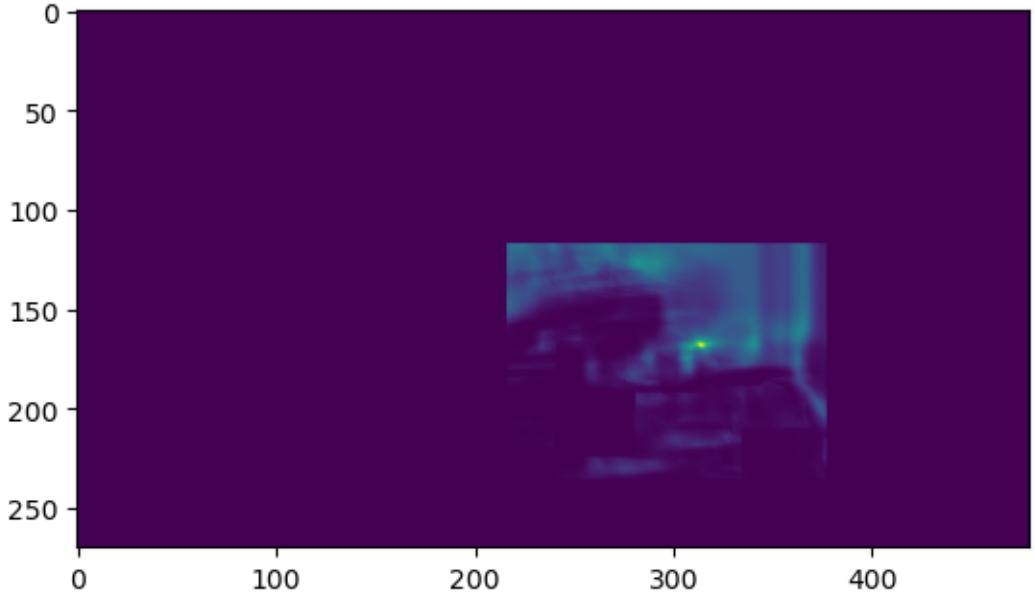
41%| | 42/102 [00:50<01:11, 1.20s/it]

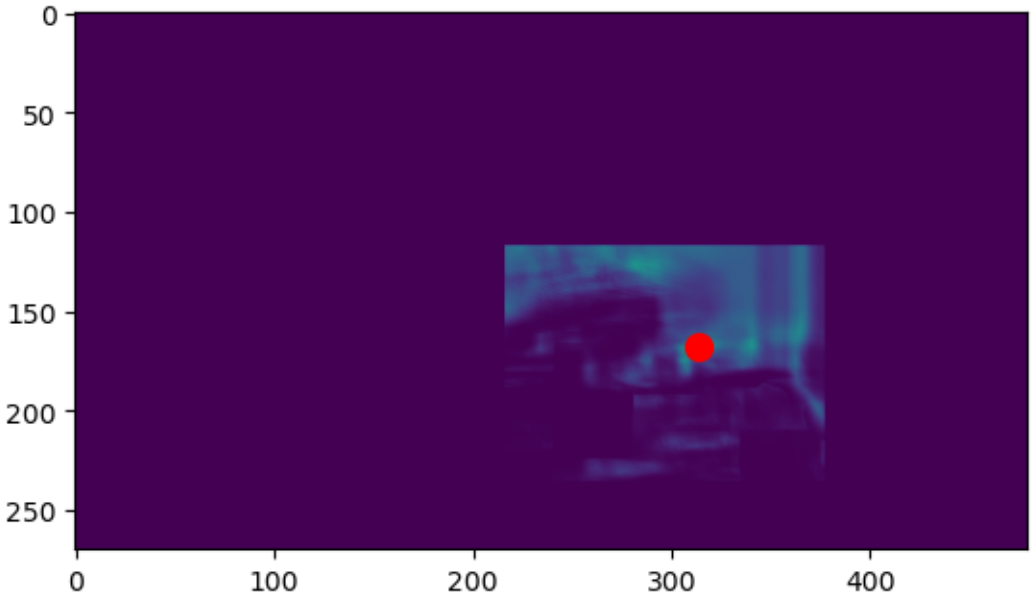

0
50
100
150
200
250
0
100
200
300
400

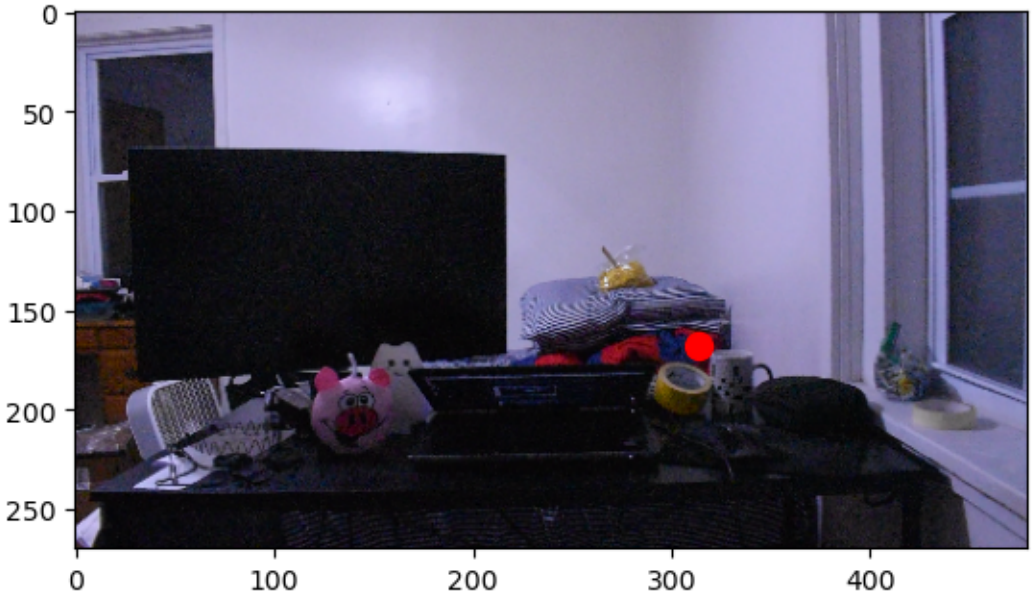

0
50
100
150
200
250
0
100
200
300
400



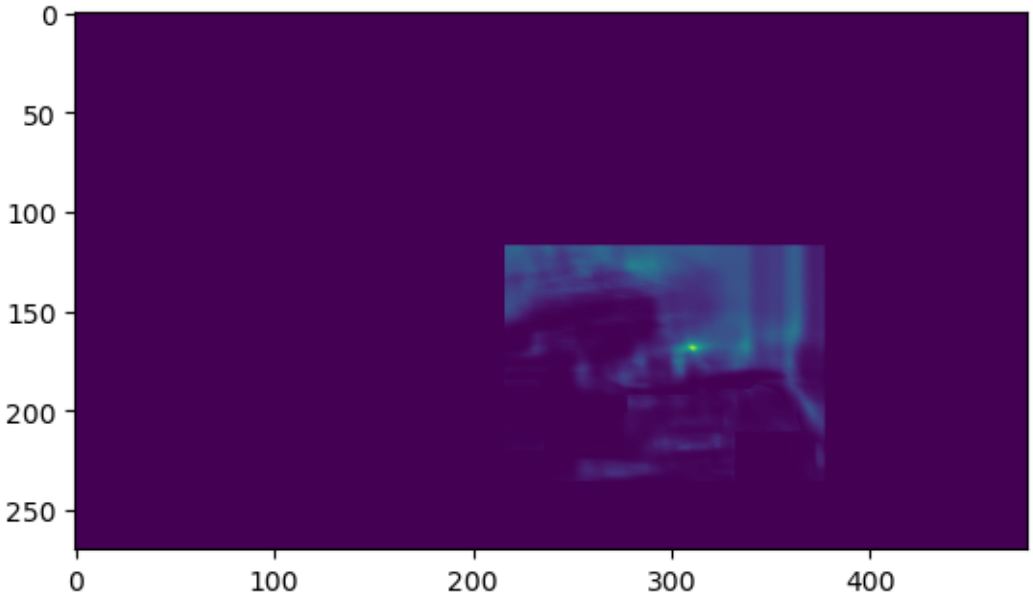
42%| | 43/102 [00:52<01:10, 1.20s/it]

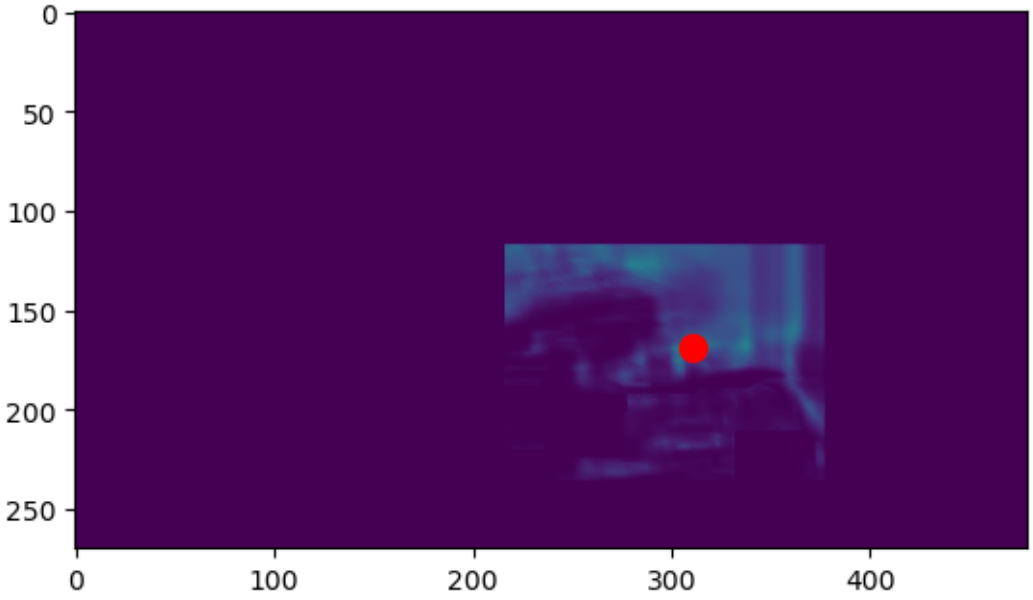


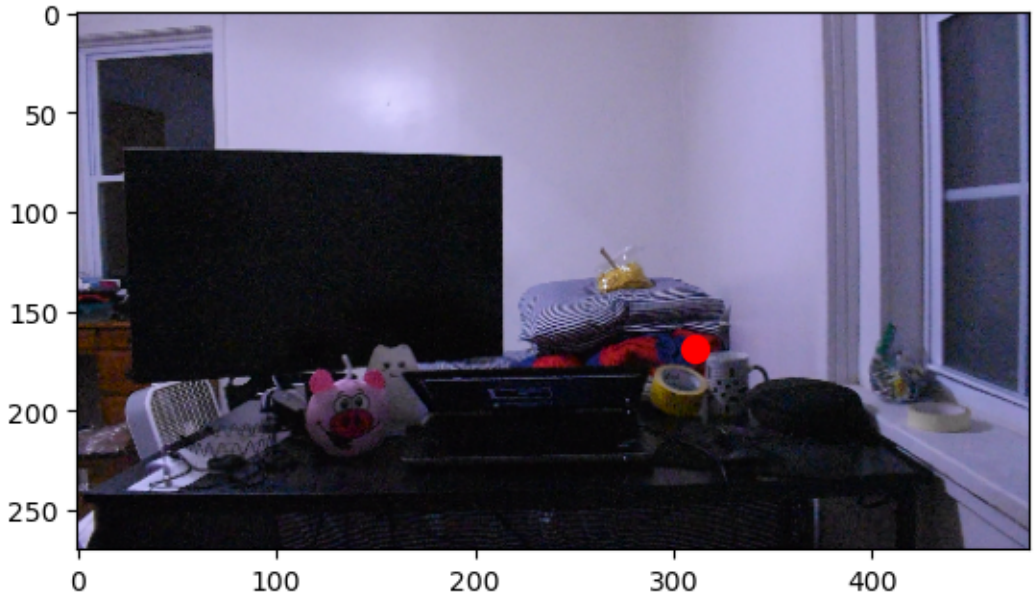




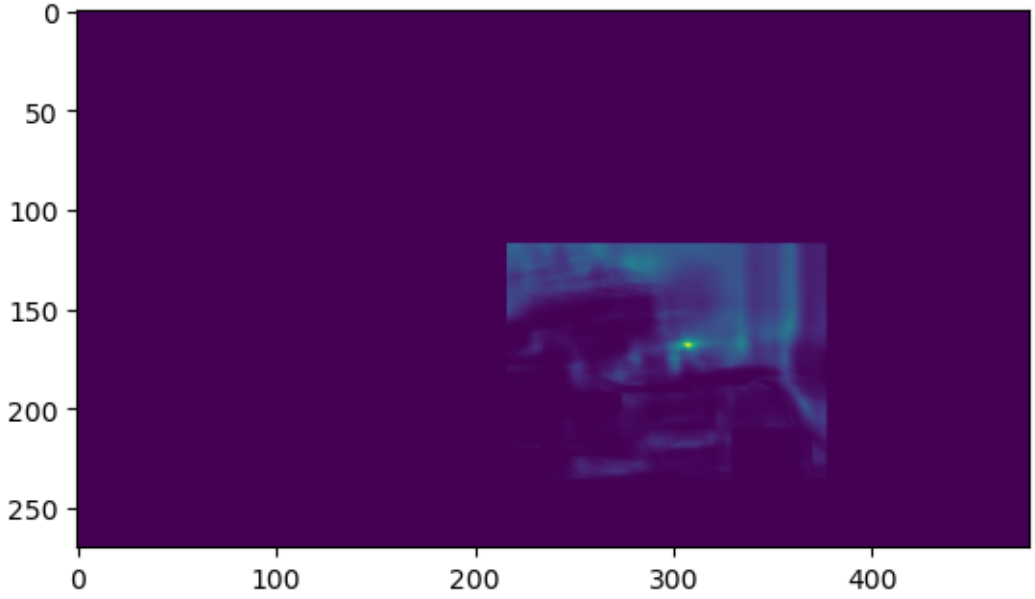
43%| | 44/102 [00:53<01:12, 1.25s/it]

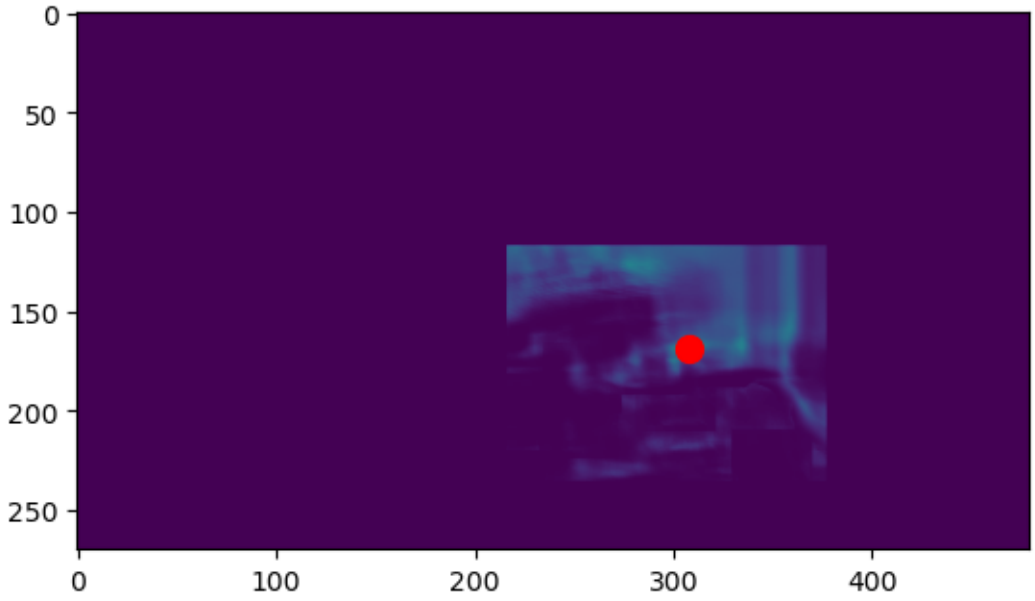

0
50
100
150
200
250
0
100
200
300
400

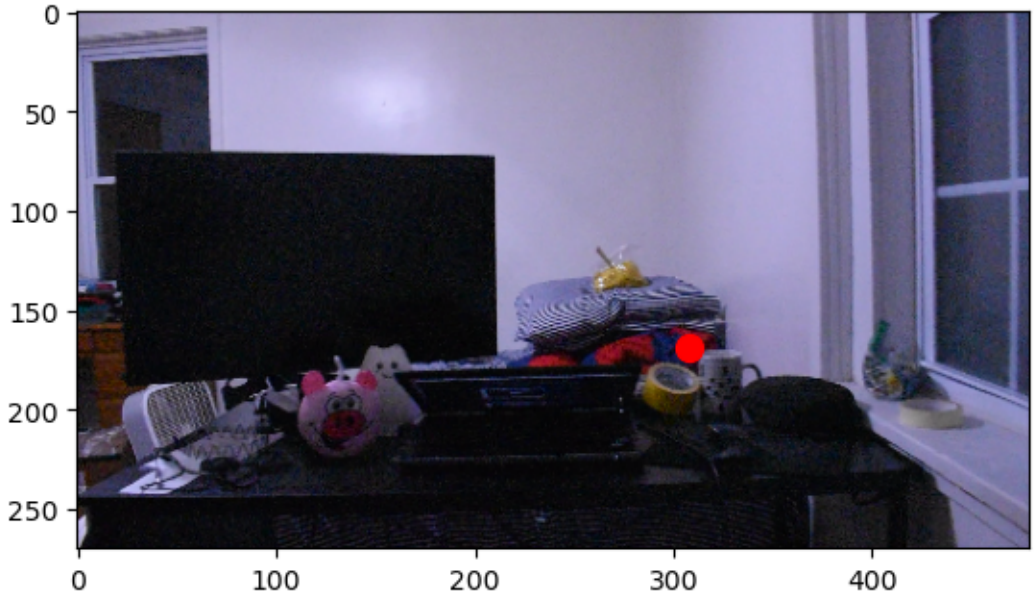

0
50
100
150
200
250
0
100
200
300
400



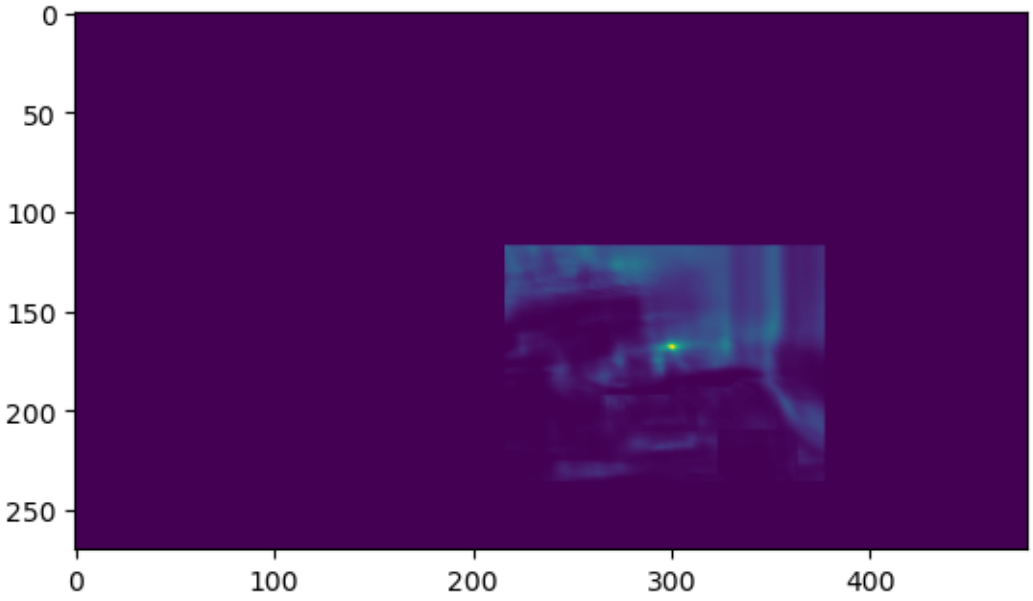
44%| | 45/102 [00:54<01:13, 1.28s/it]

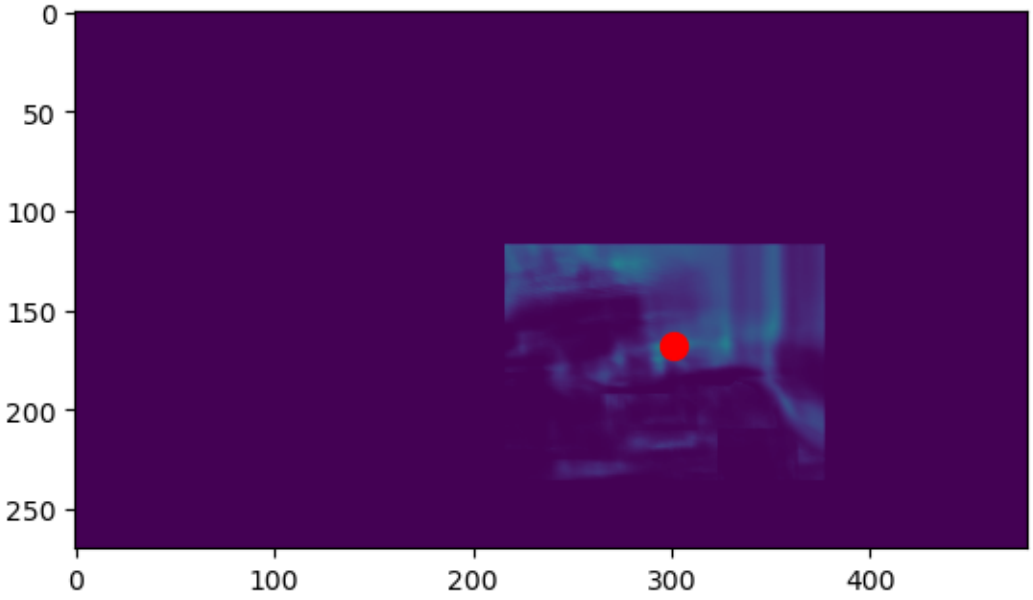


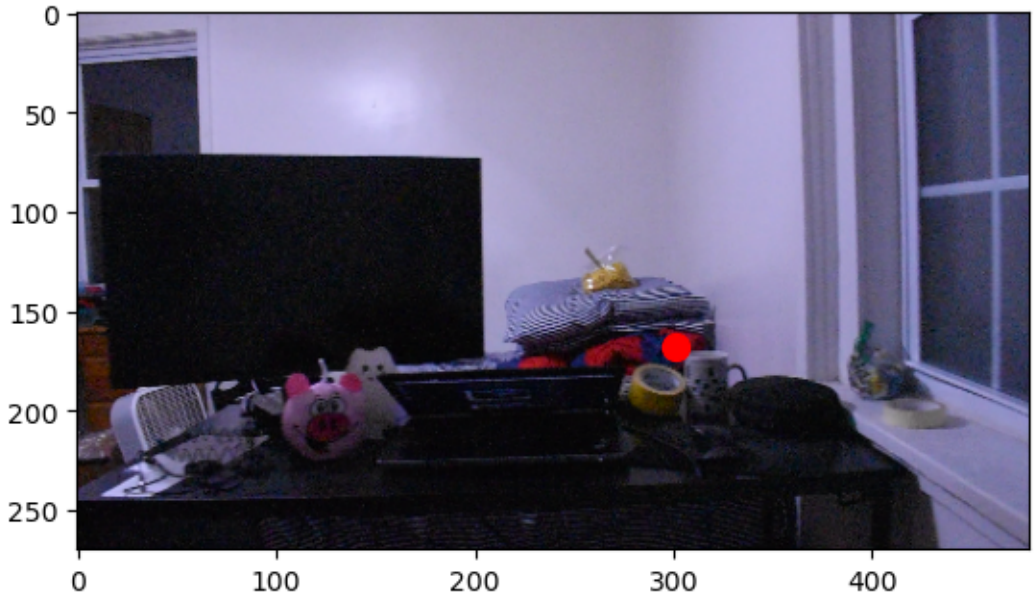




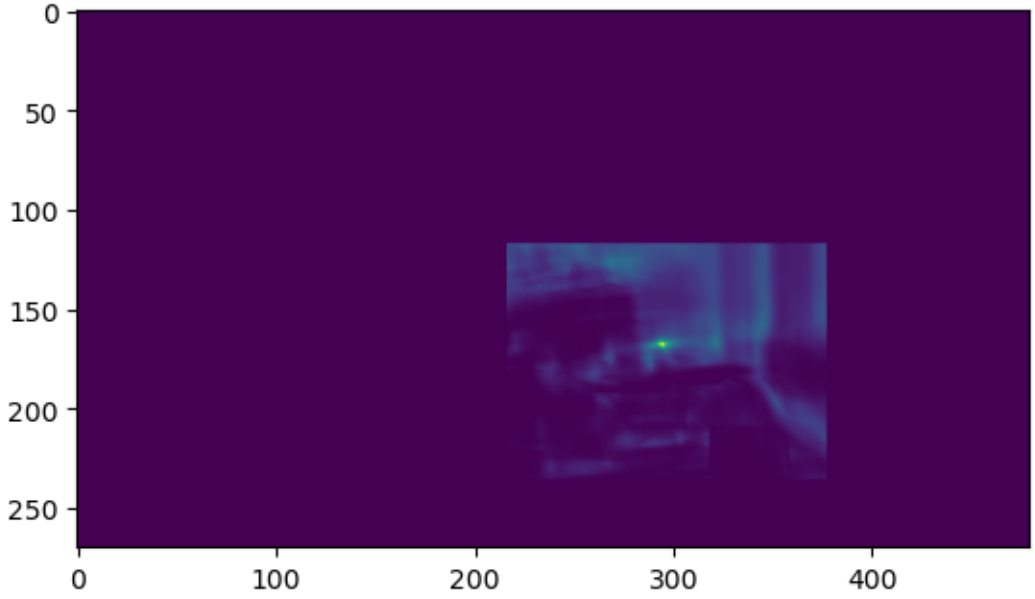
45%| | 46/102 [00:56<01:10, 1.26s/it]

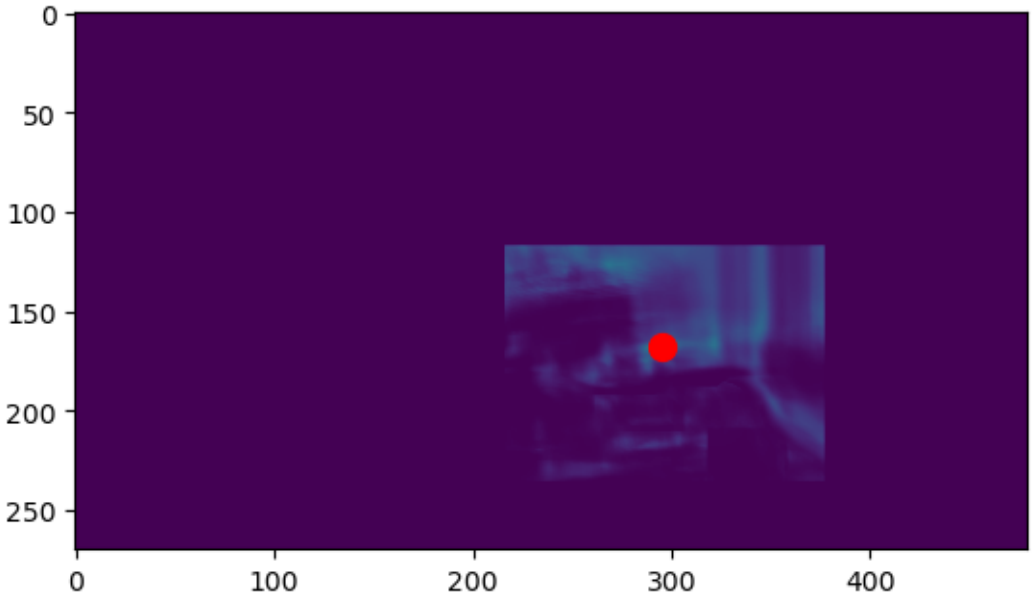

0
50
100
150
200
250
0
100
200
300
400

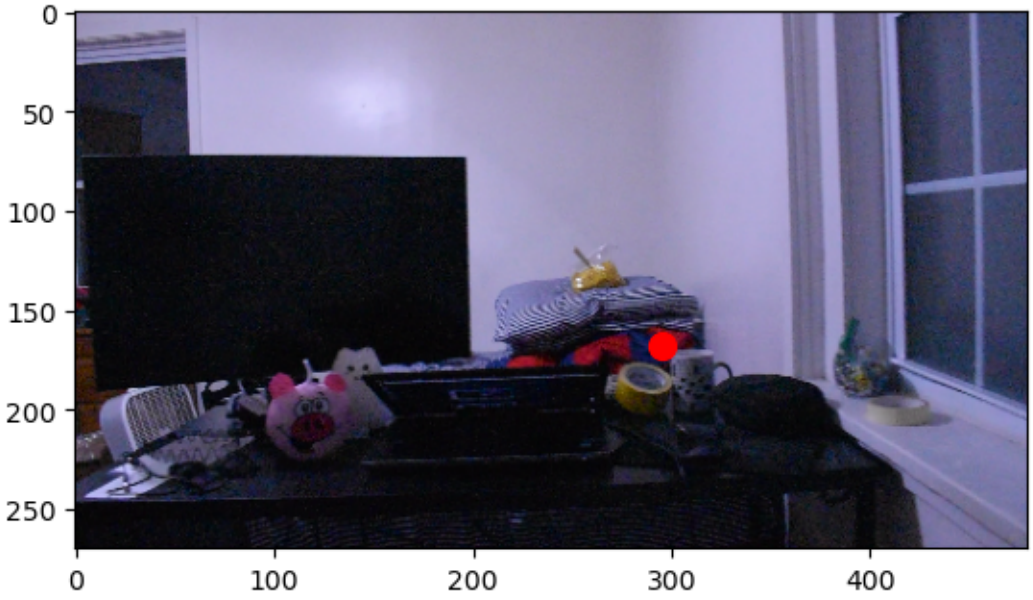

0
50
100
150
200
250
0
100
200
300
400



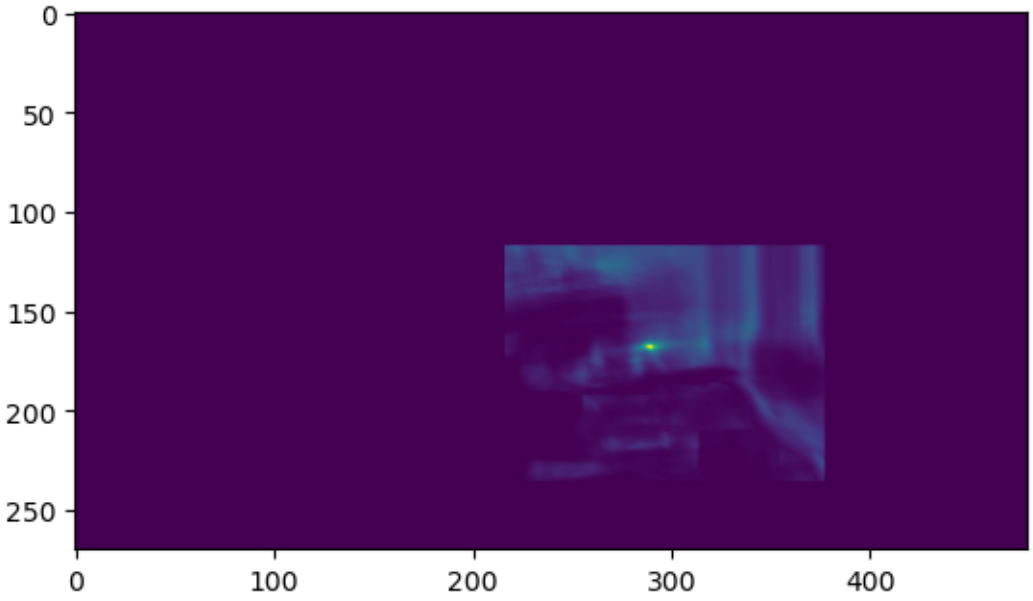
46%| | 47/102 [00:57<01:08, 1.24s/it]

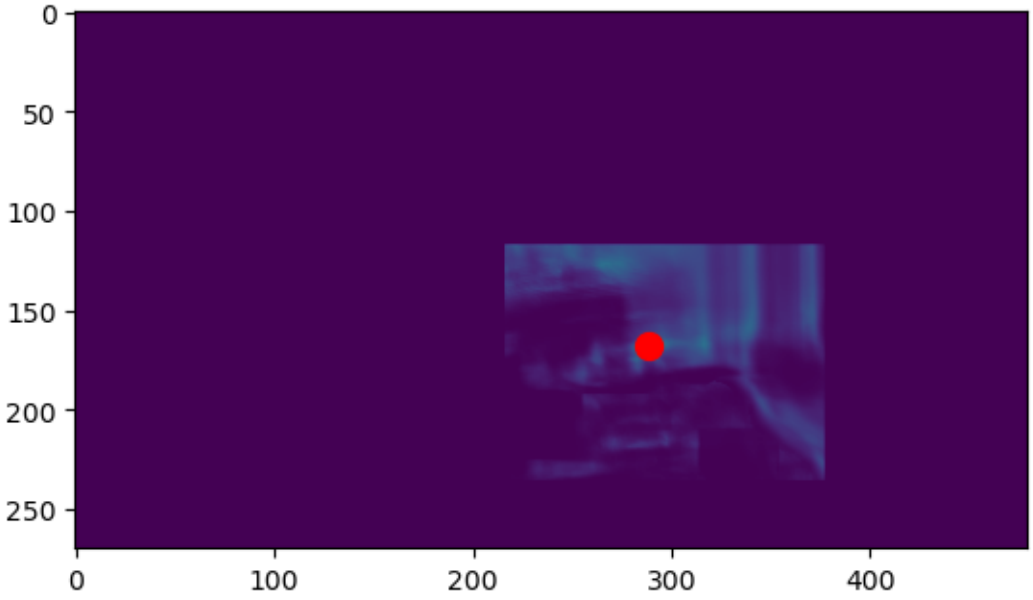


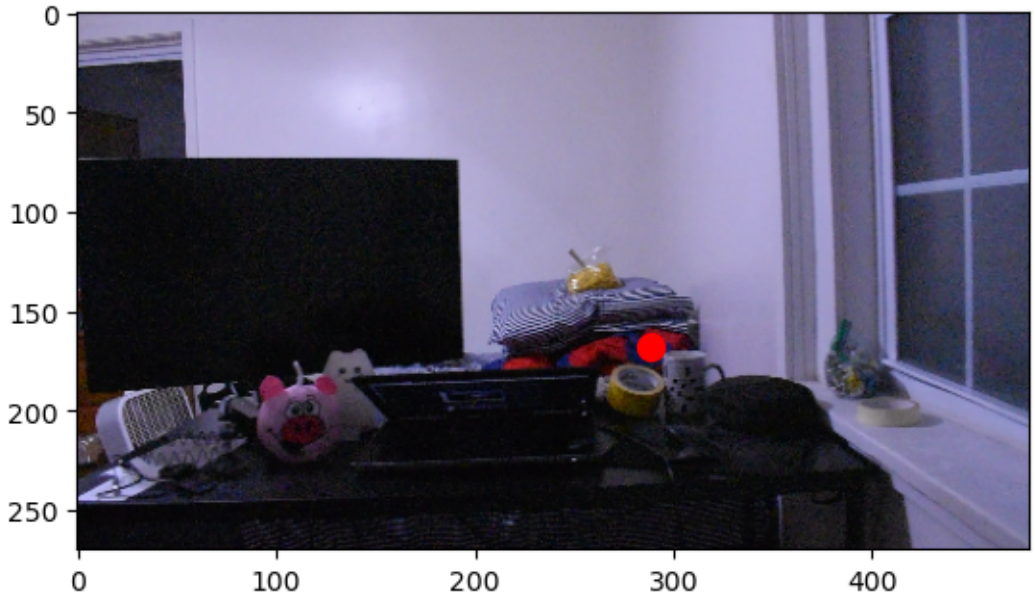




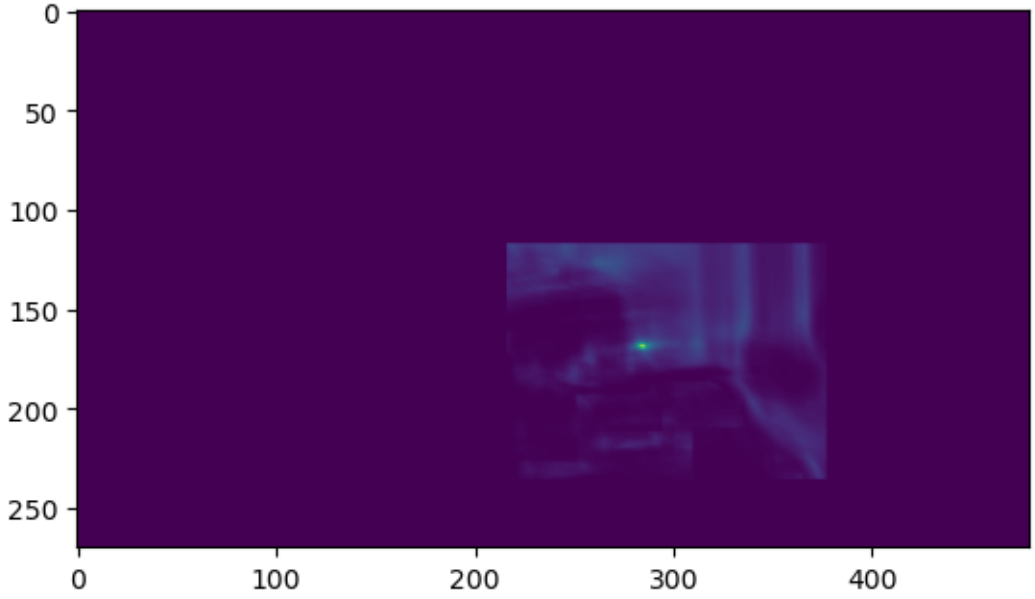
47%| | 48/102 [00:58<01:06, 1.24s/it]

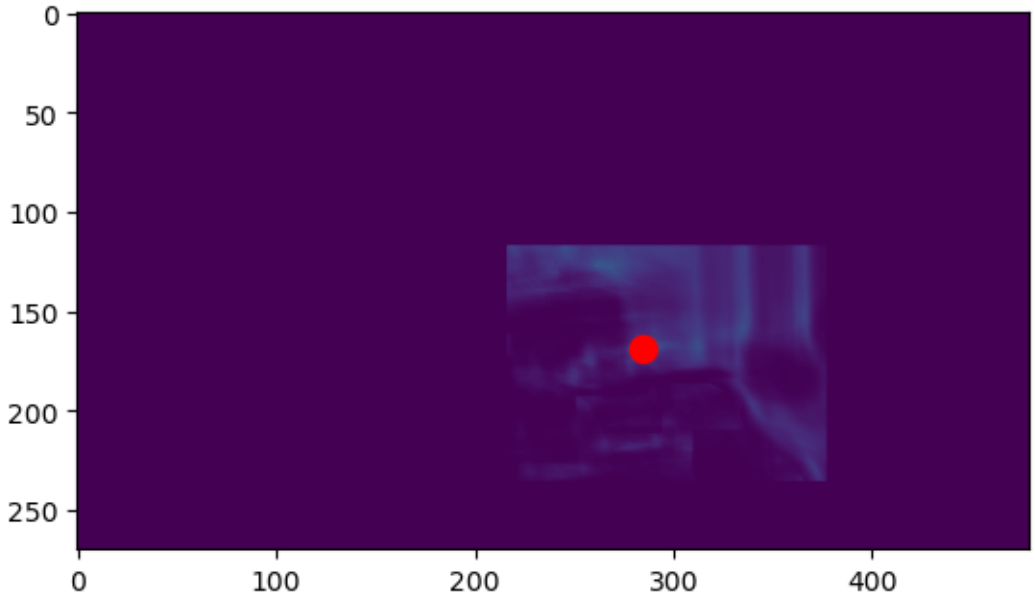

0
50
100
150
200
250
0
100
200
300
400

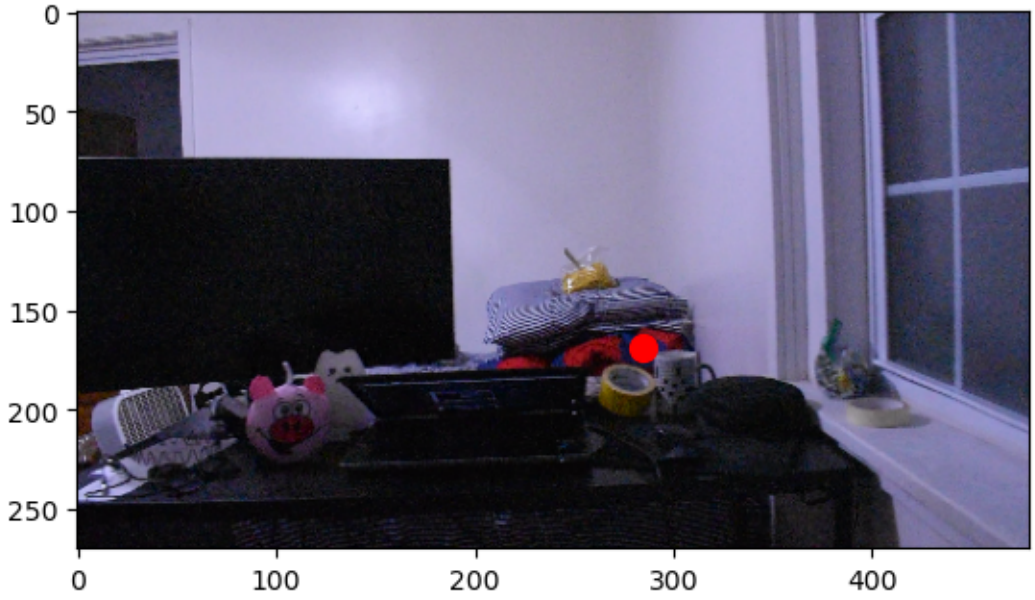

0
50
100
150
200
250
0
100
200
300
400



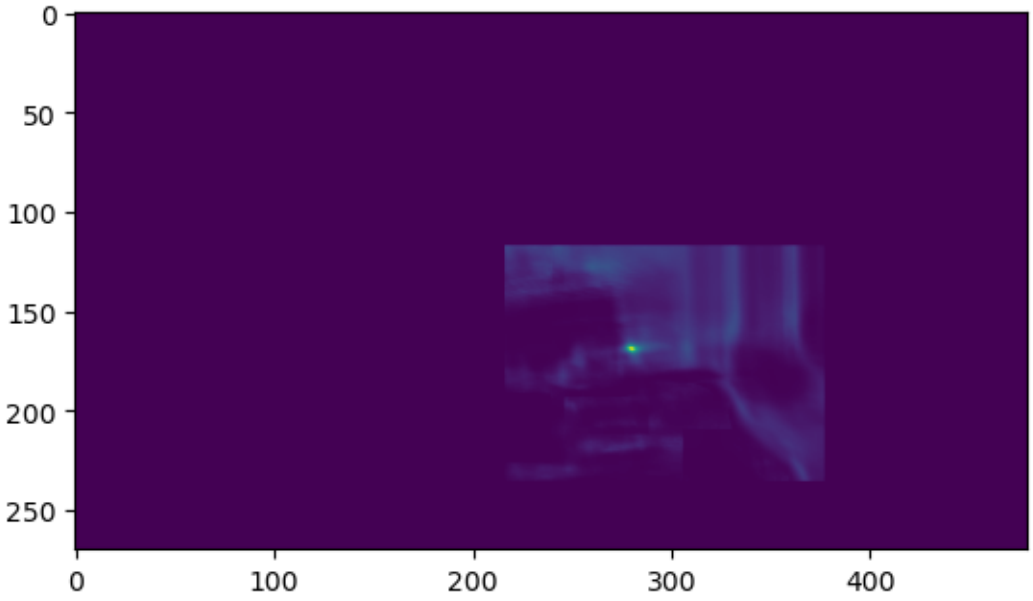
48%| | 49/102 [00:59<01:05, 1.23s/it]

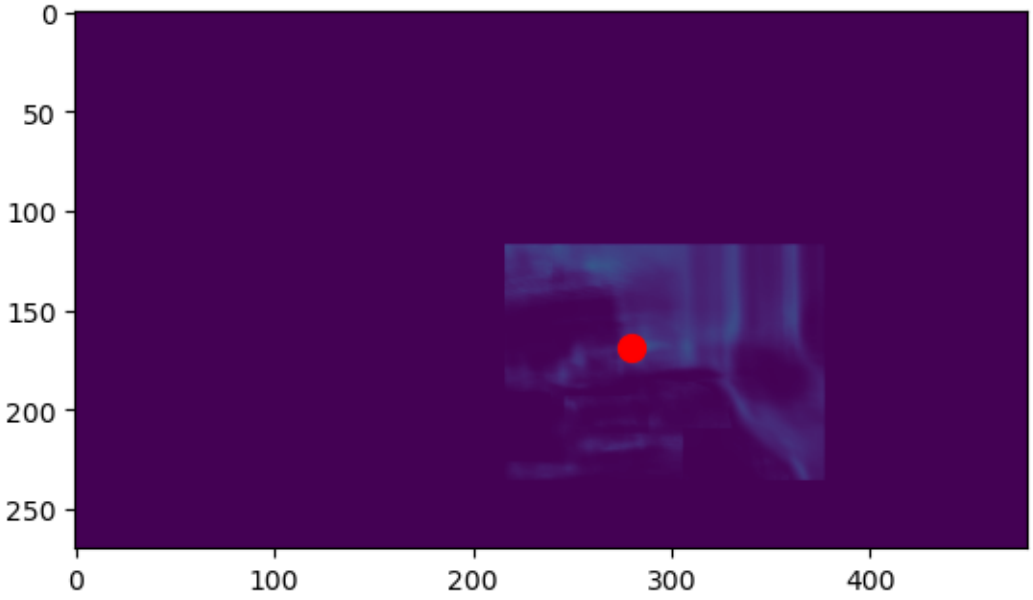


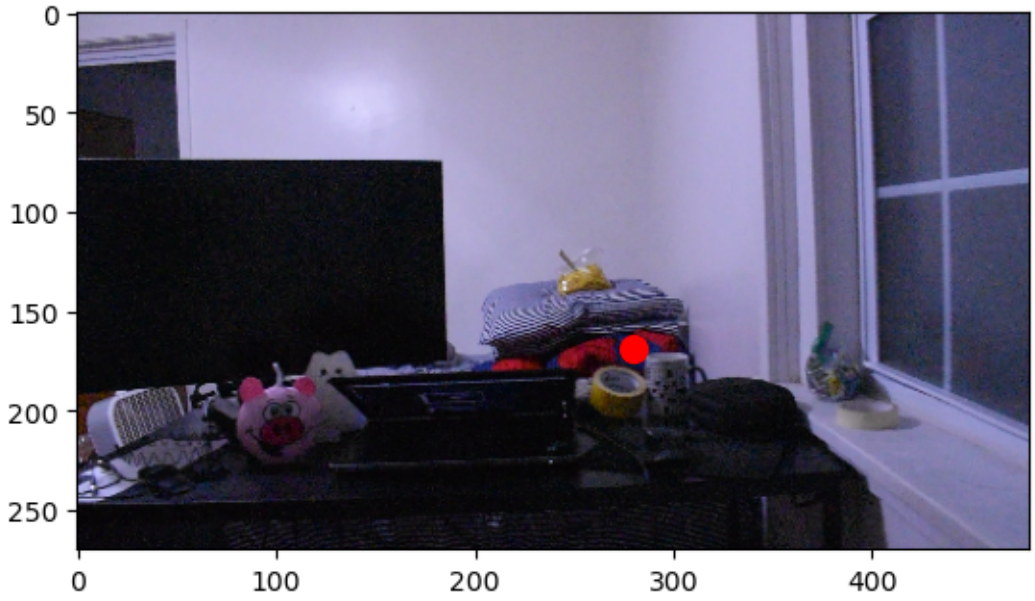




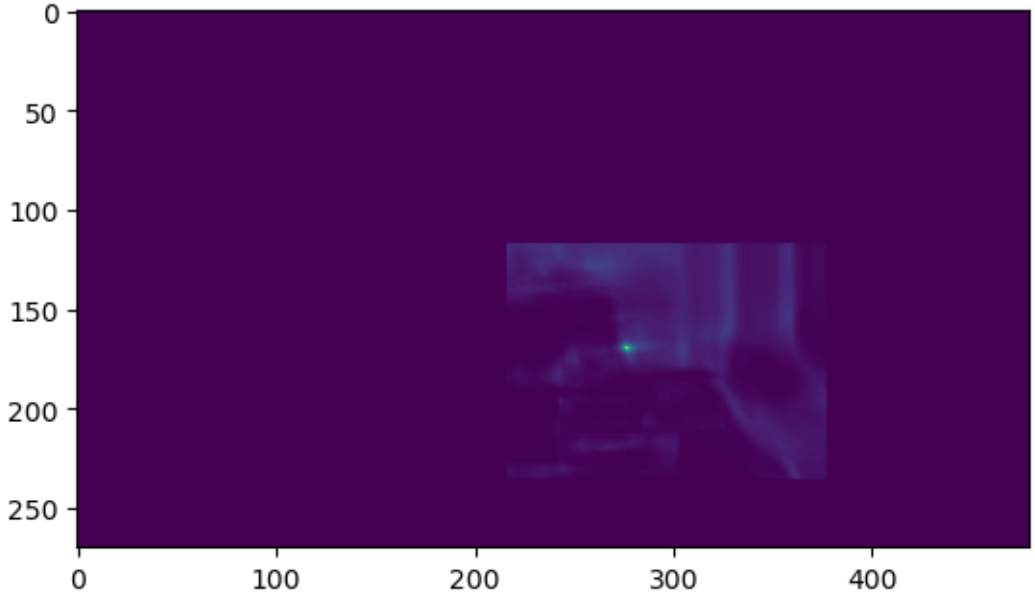
49%| | 50/102 [01:00<01:03, 1.23s/it]

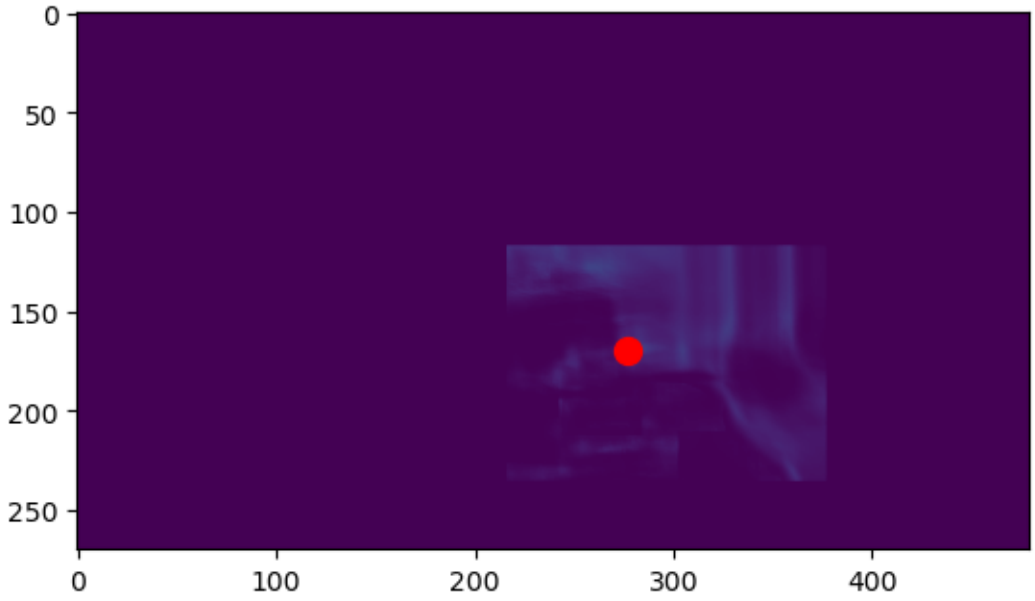

0
50
100
150
200
250
0
100
200
300
400

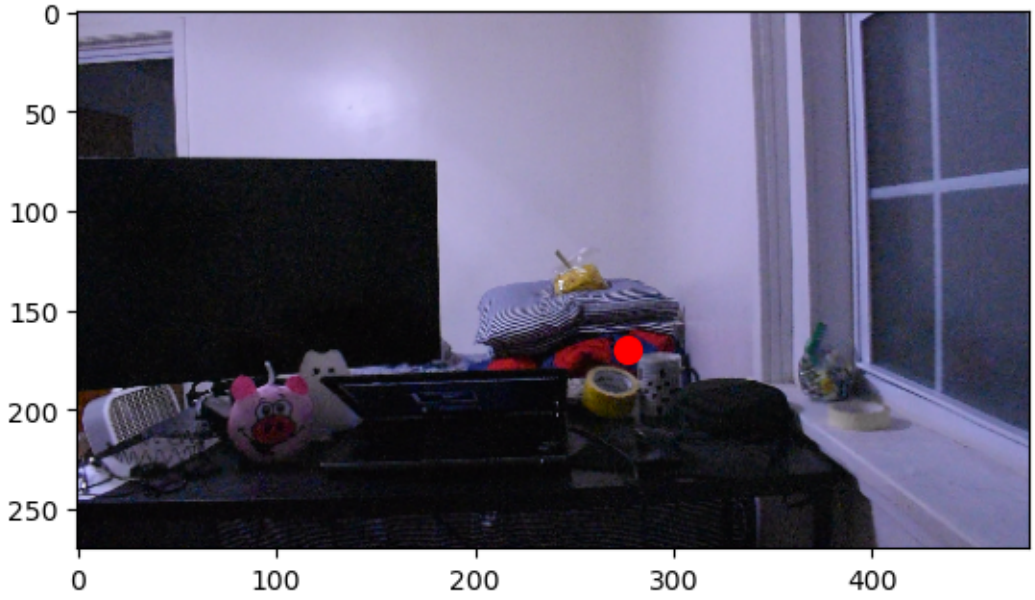

0
50
100
150
200
250
0
100
200
300
400



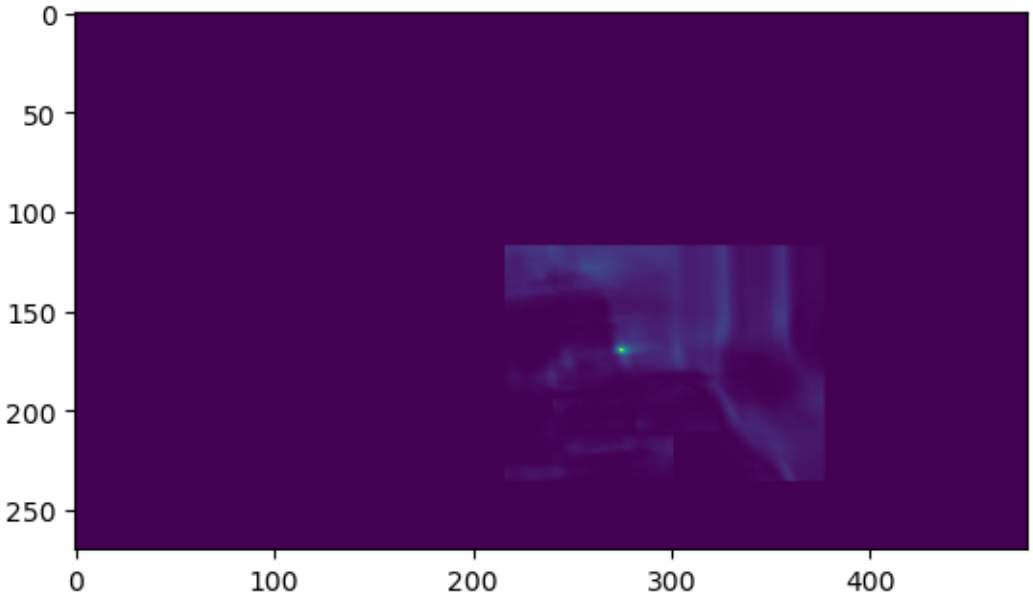
50%| | 51/102 [01:02<01:02, 1.22s/it]

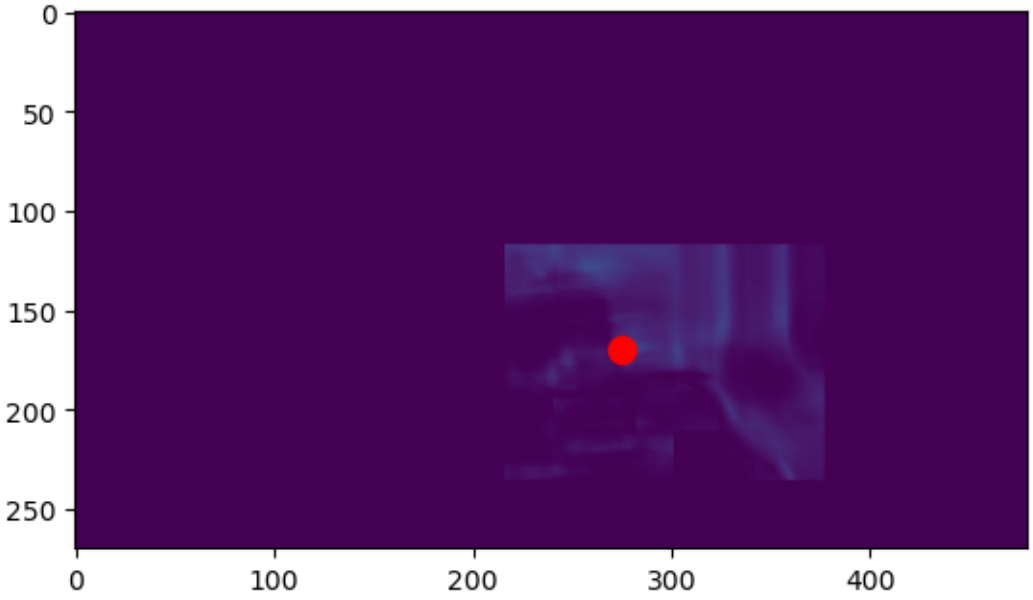


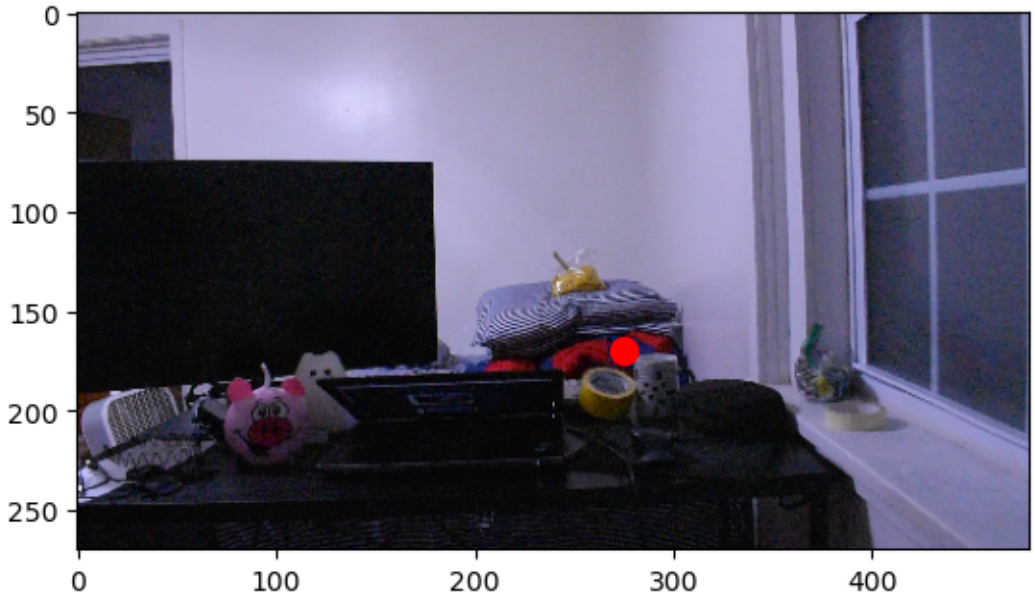




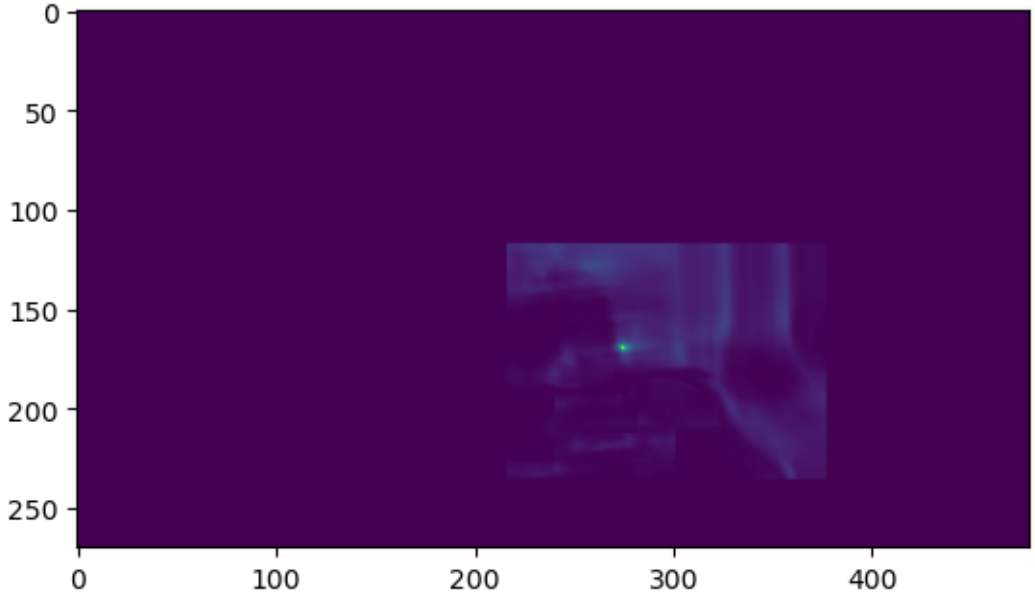
51%| | 52/102 [01:03<01:00, 1.22s/it]

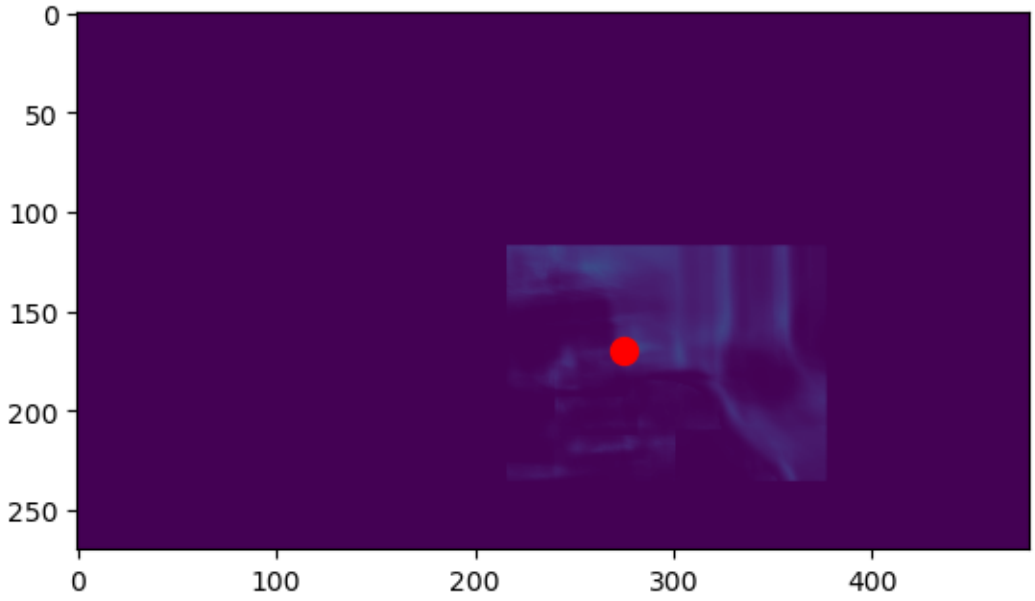

0
50
100
150
200
250
0
100
200
300
400

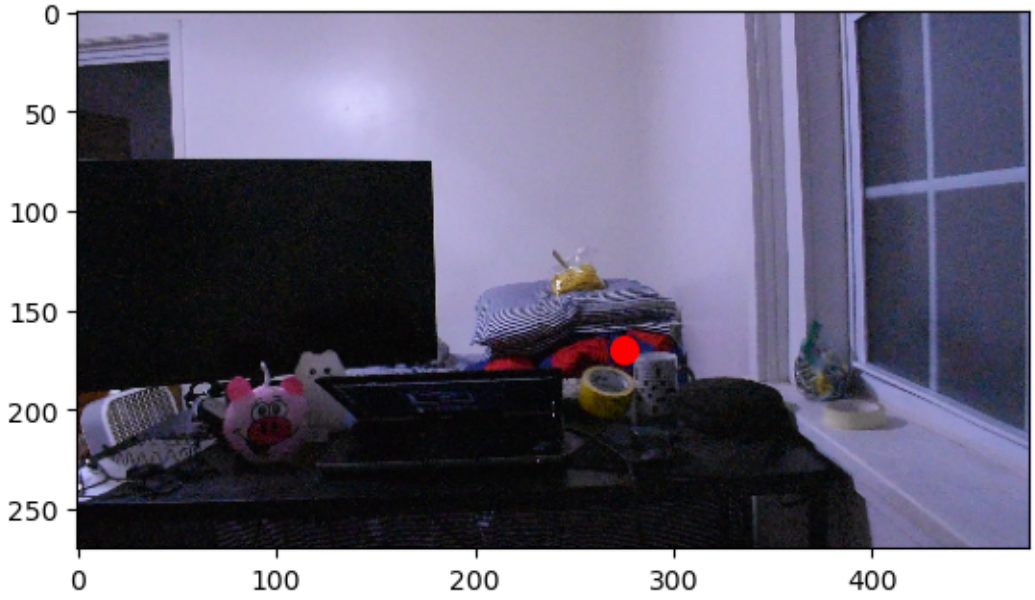

0
50
100
150
200
250
0
100
200
300
400



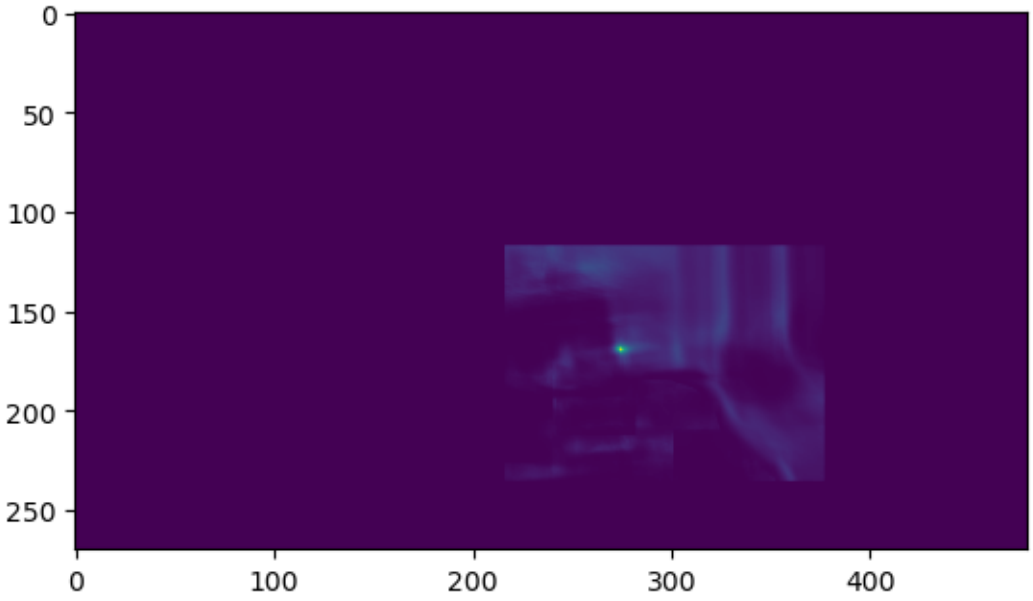
52%| | 53/102 [01:04<00:59, 1.21s/it]

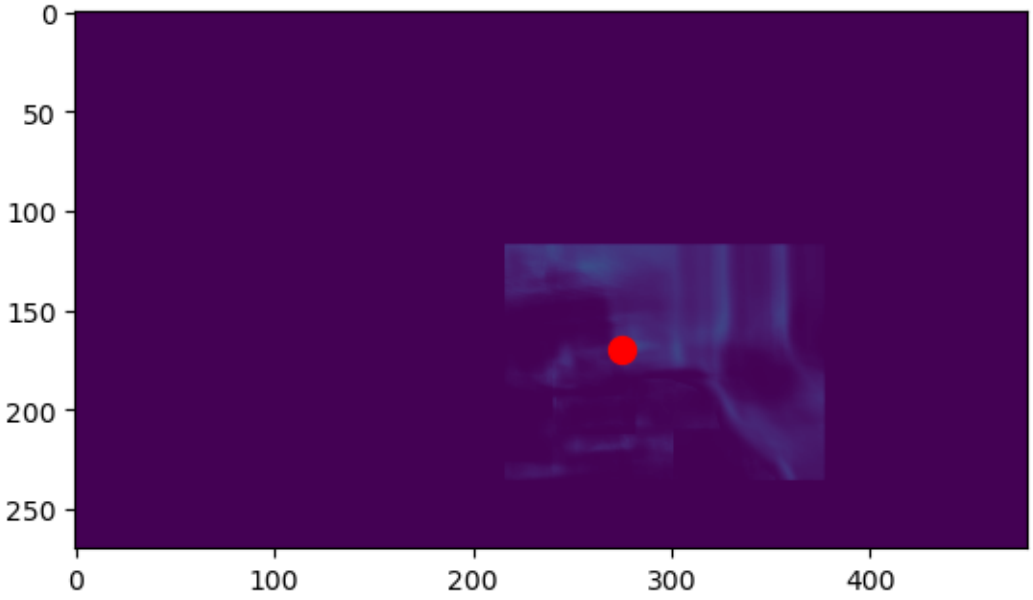


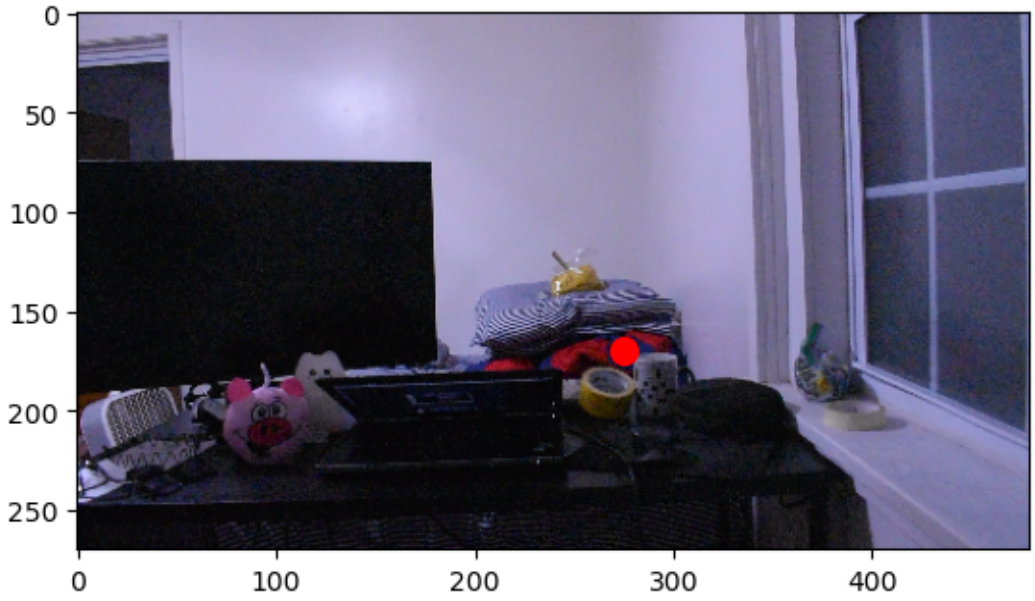




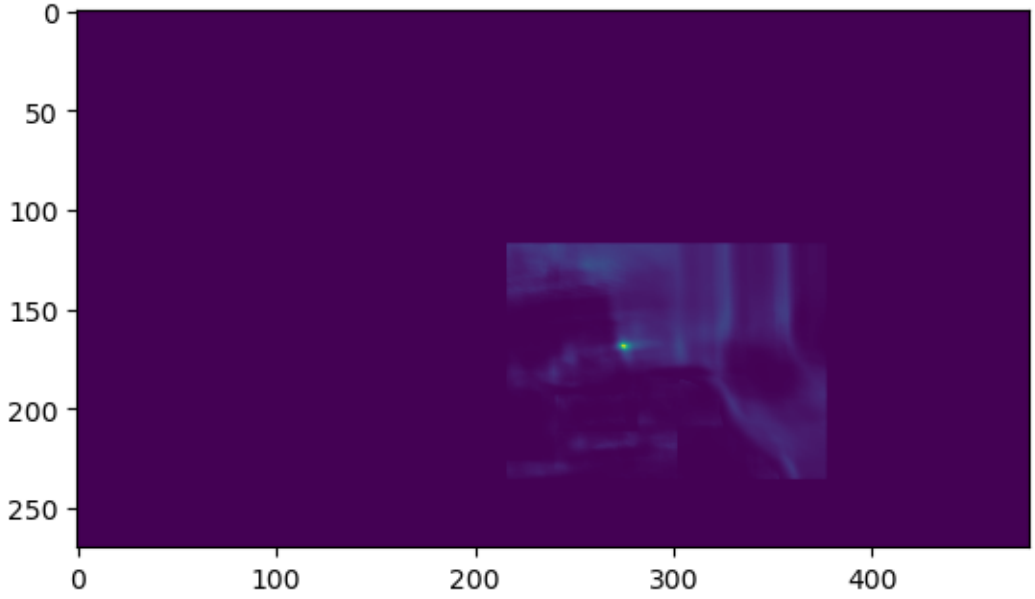
53%| | 54/102 [01:05<00:57, 1.21s/it]

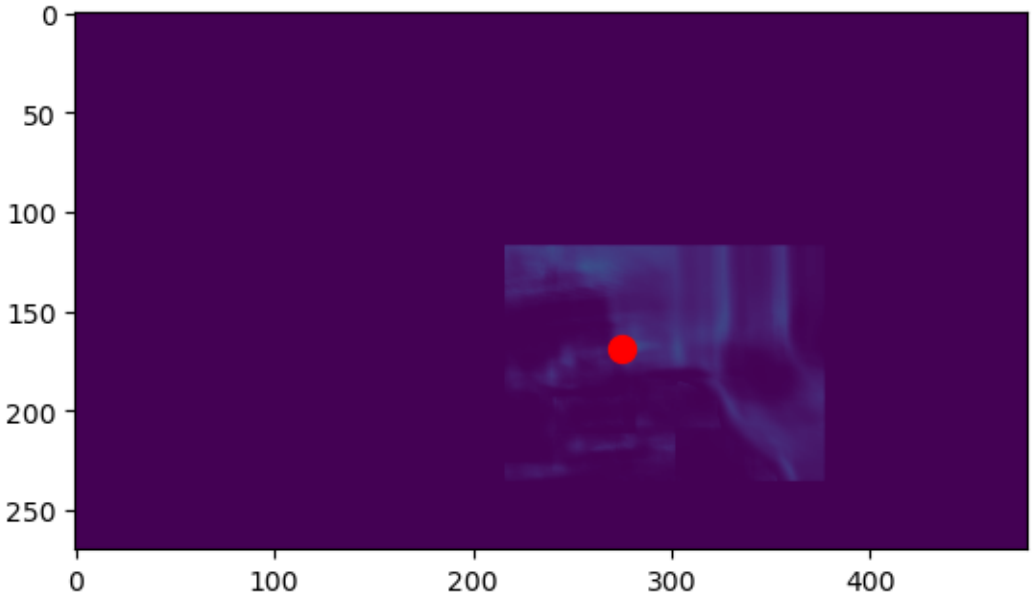

0
50
100
150
200
250
0
100
200
300
400

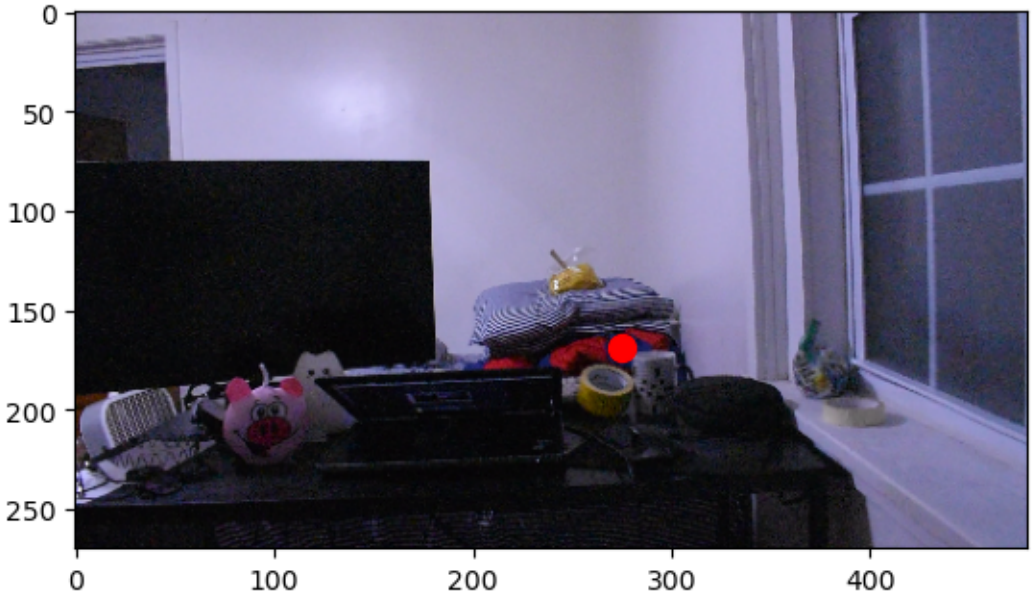

0
50
100
150
200
250
0
100
200
300
400



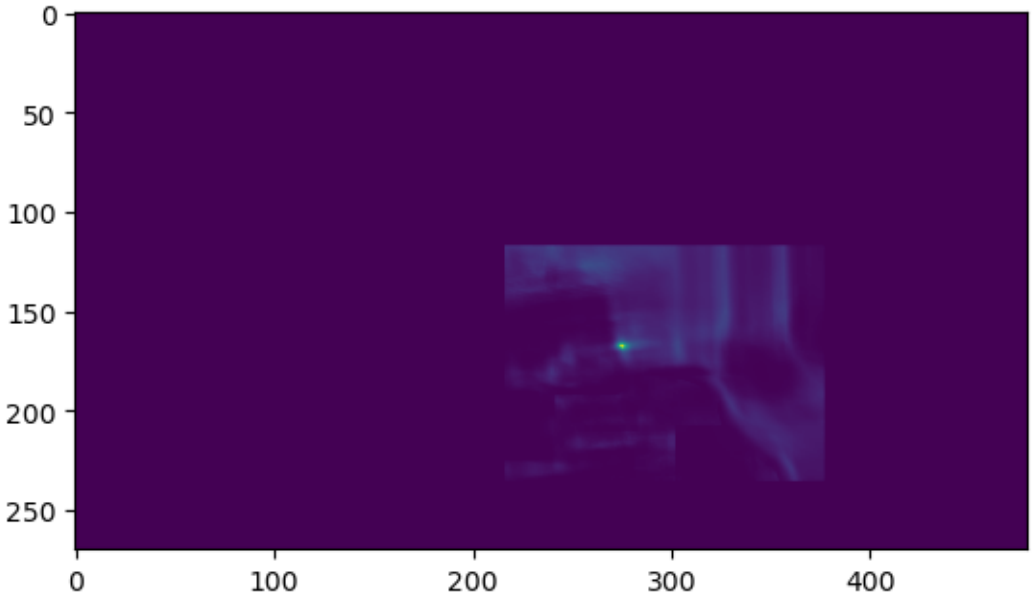
54%| | 55/102 [01:06<00:56, 1.20s/it]

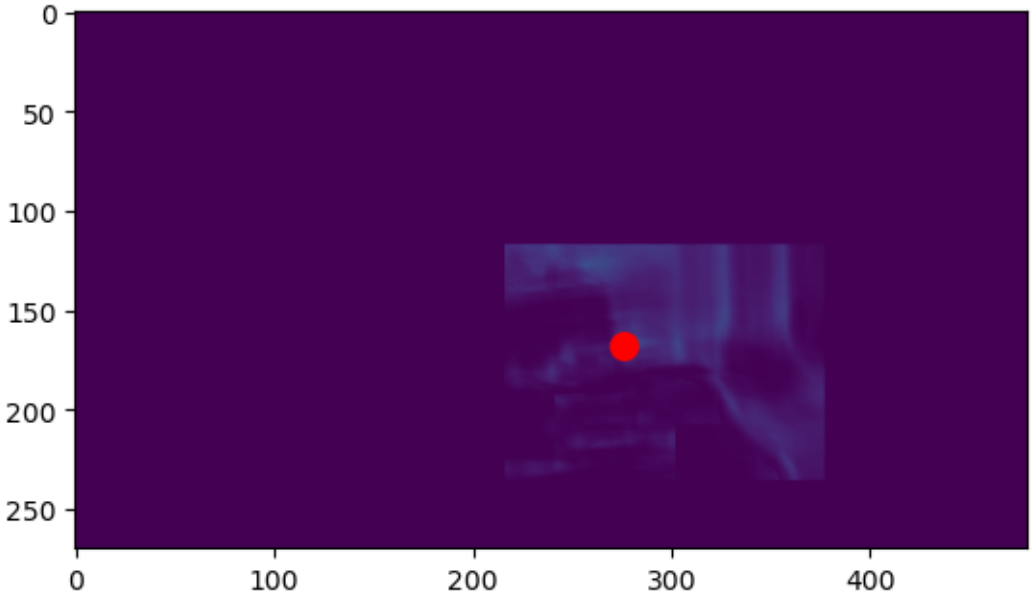


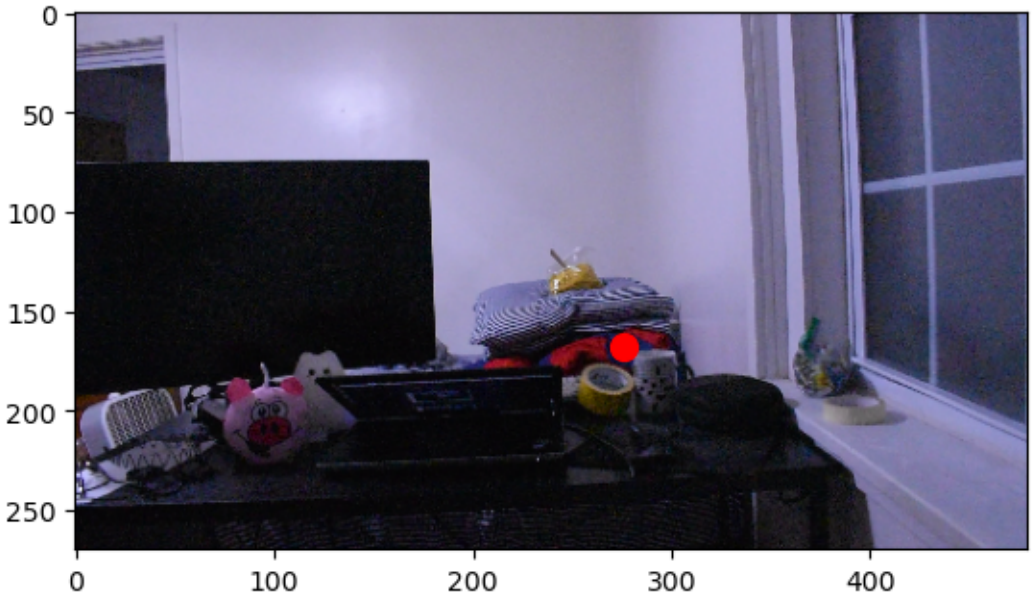




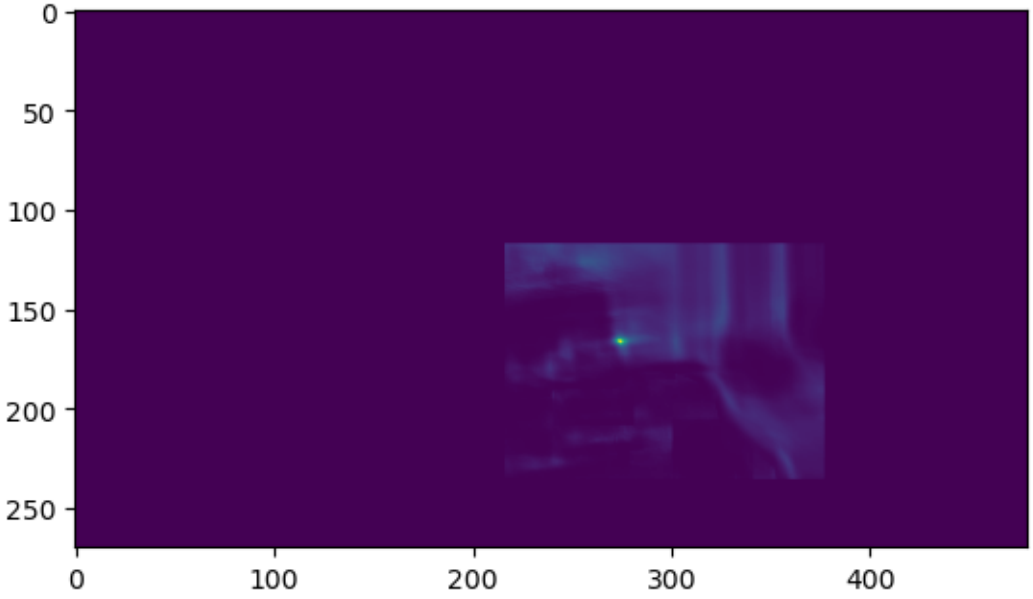
55%| | 56/102 [01:08<00:57, 1.24s/it]

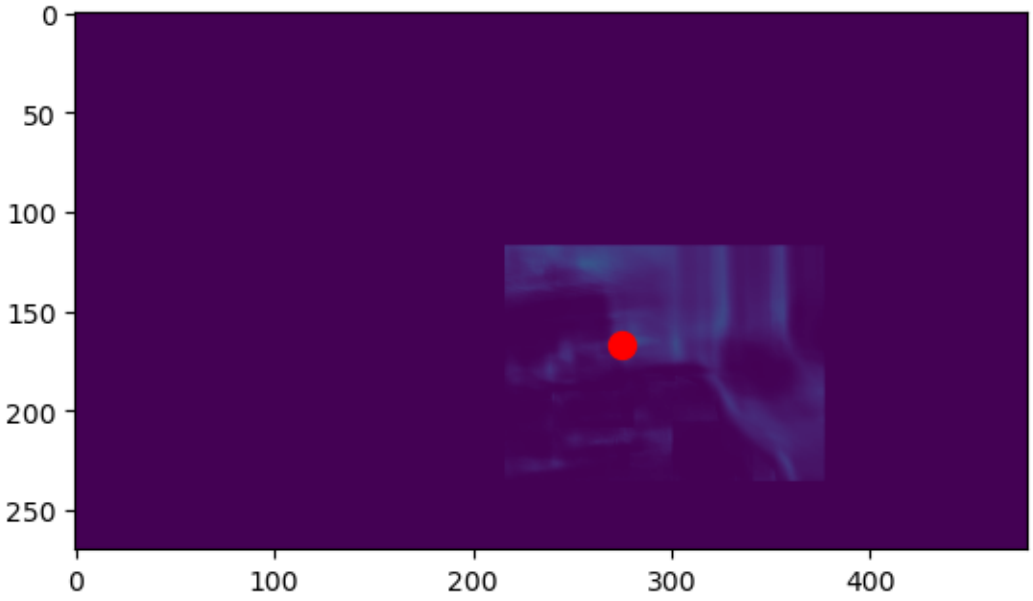

0
50
100
150
200
250
0
100
200
300
400

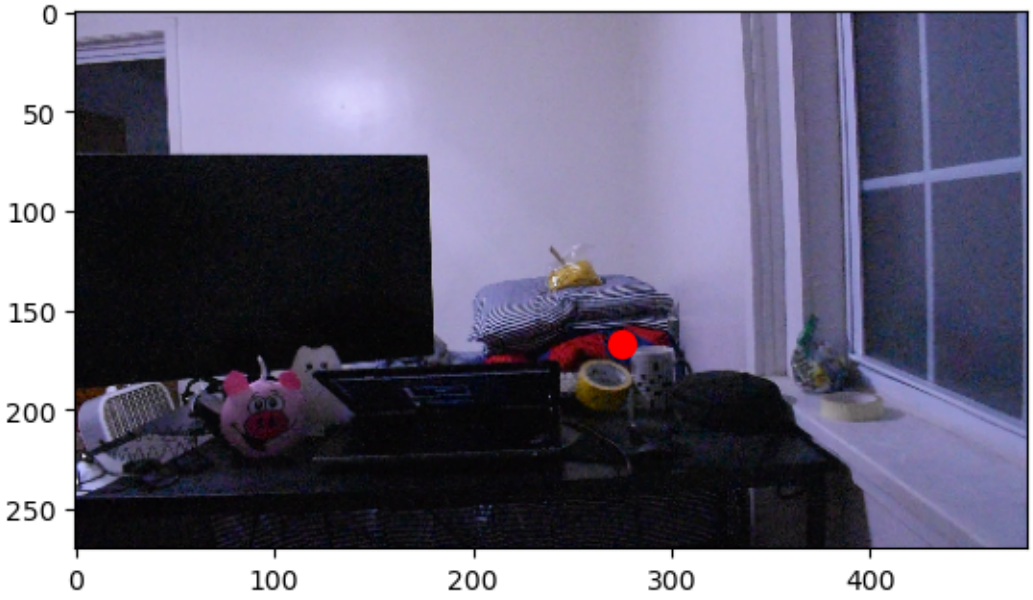

0
50
100
150
200
250
0
100
200
300
400



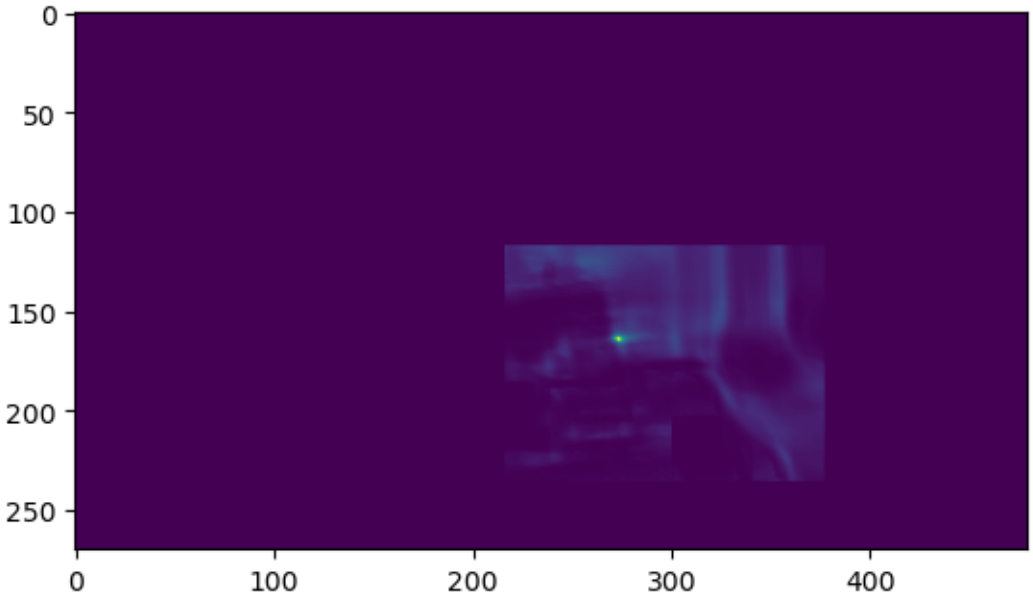
56%| | 57/102 [01:09<00:55, 1.24s/it]

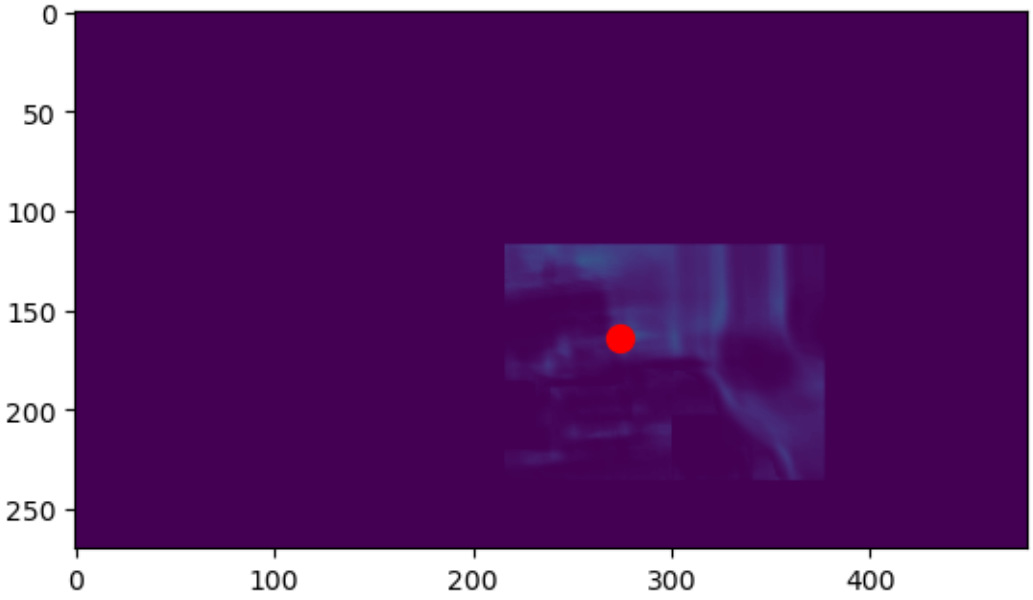


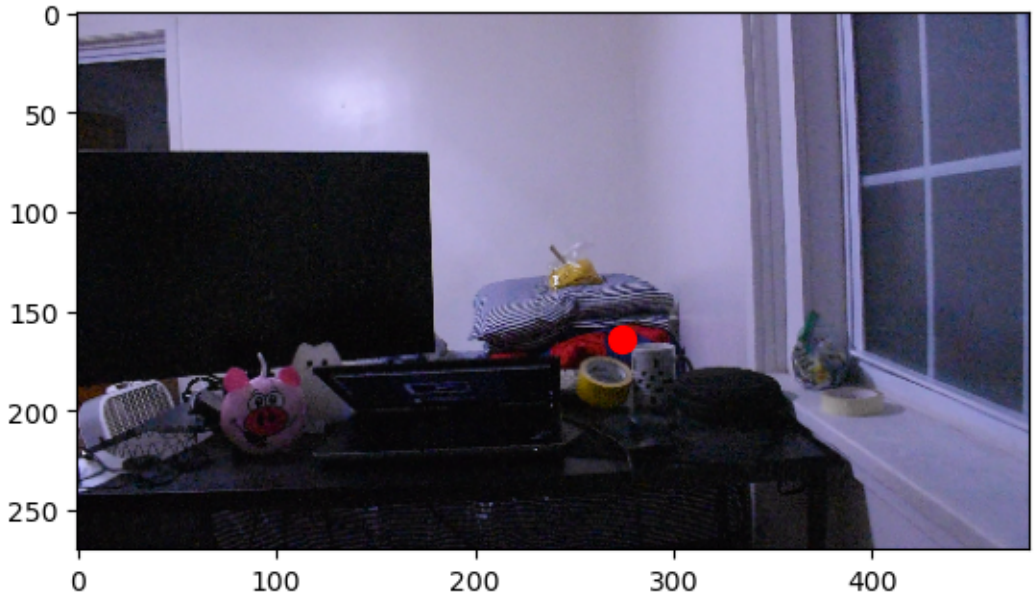




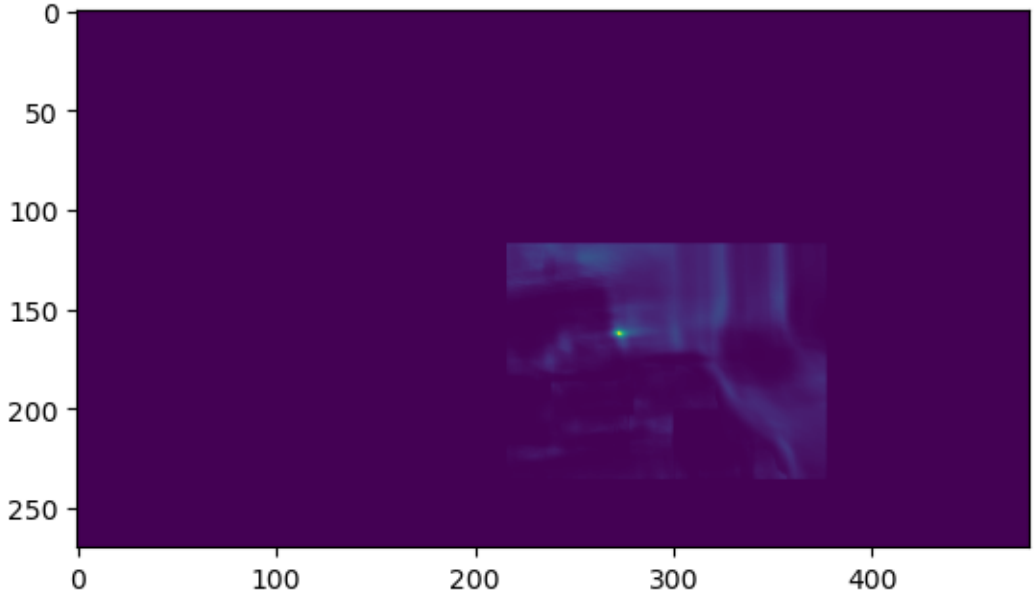
57%| | 58/102 [01:10<00:54, 1.23s/it]

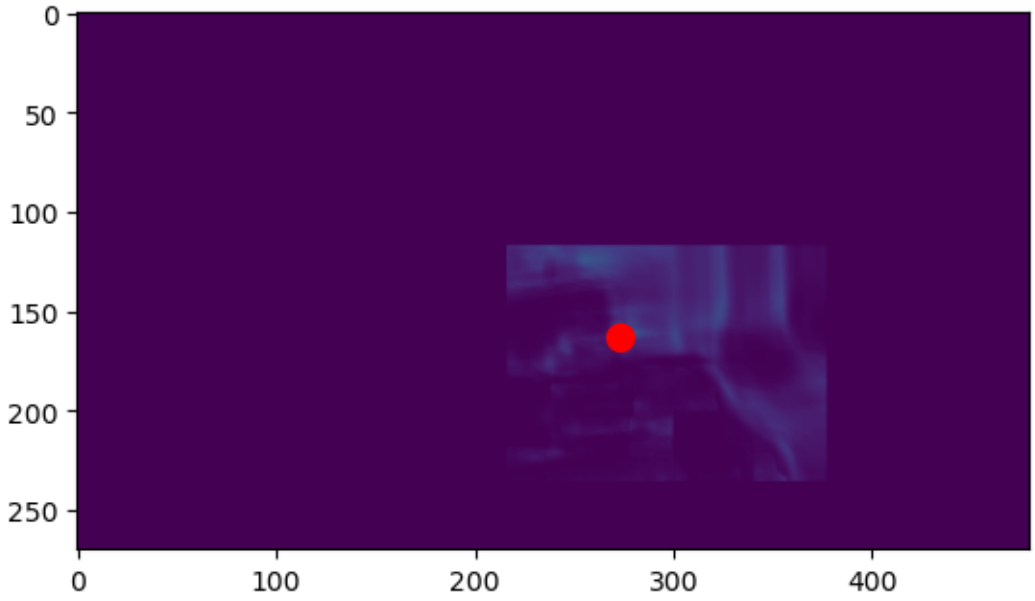

0
50
100
150
200
250
0
100
200
300
400

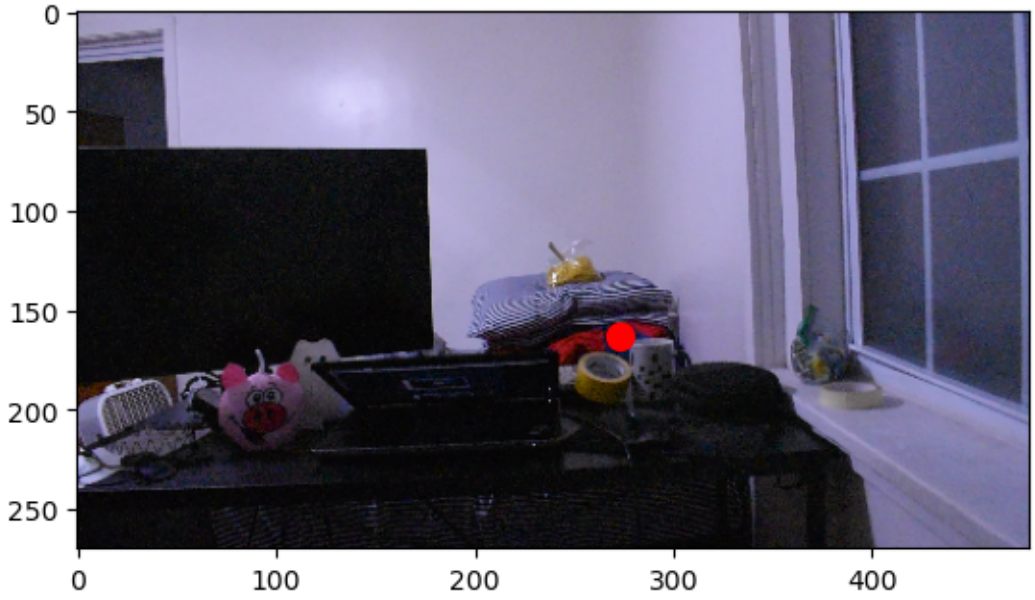

0
50
100
150
200
250
0
100
200
300
400



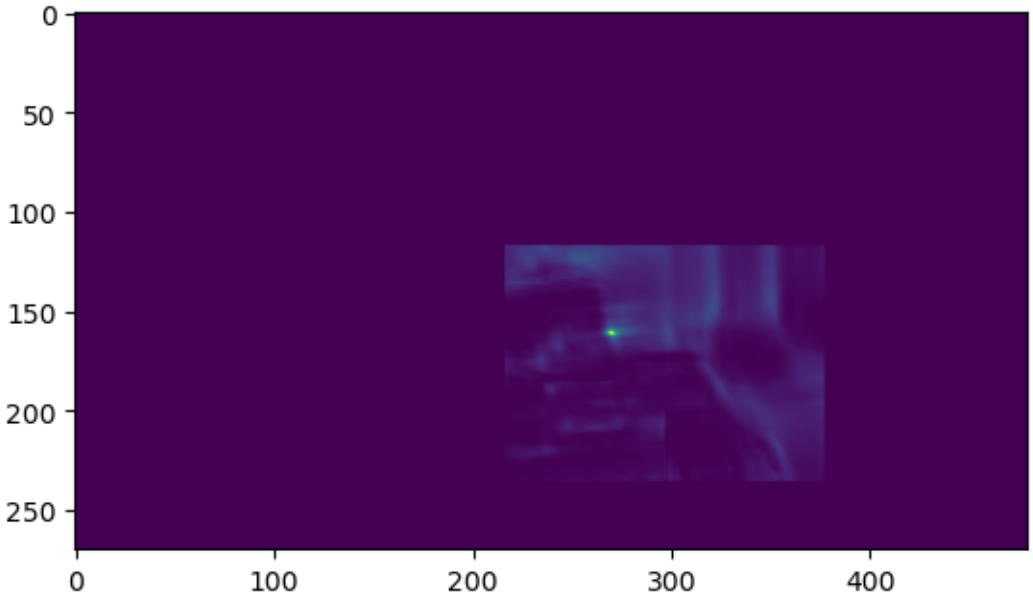
58%| | 59/102 [01:11<00:52, 1.22s/it]

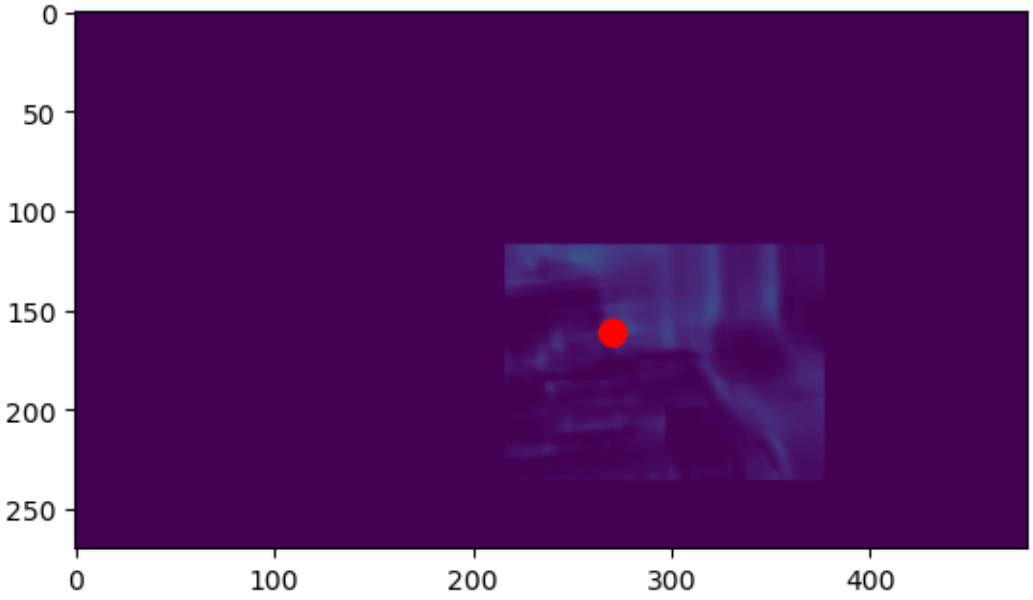


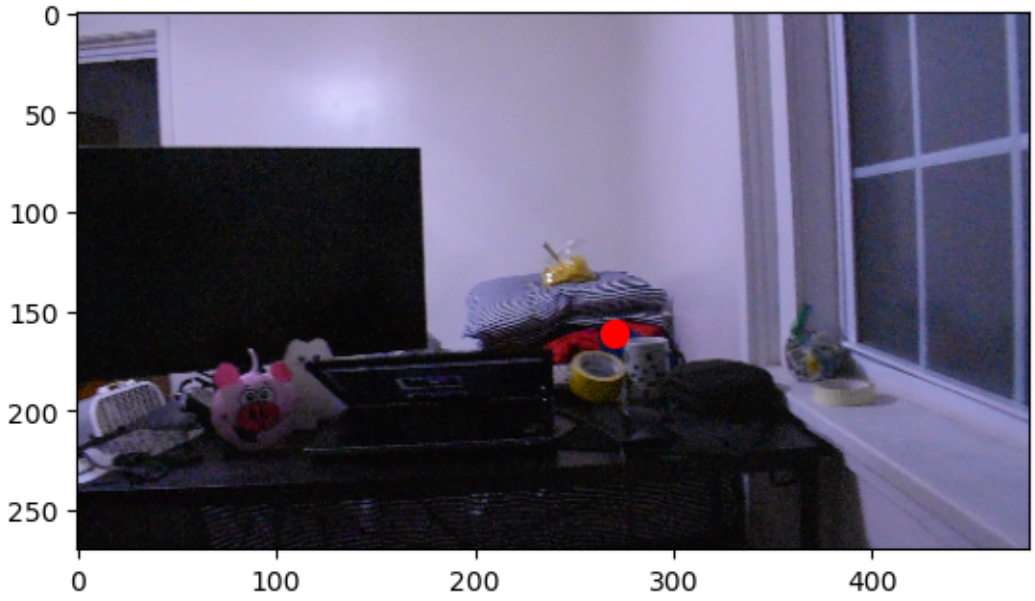




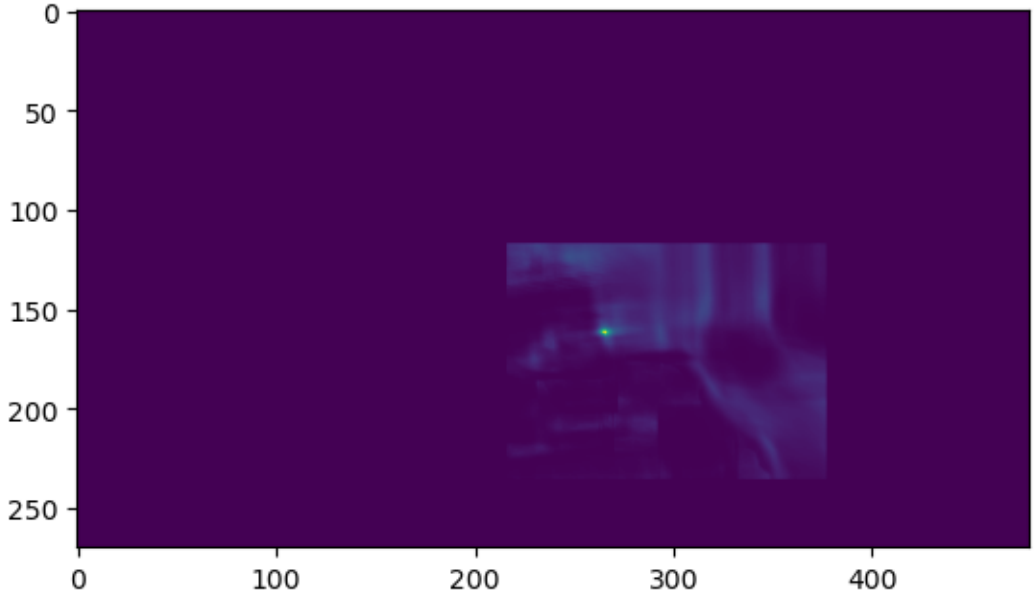
59%| | 60/102 [01:13<00:50, 1.21s/it]

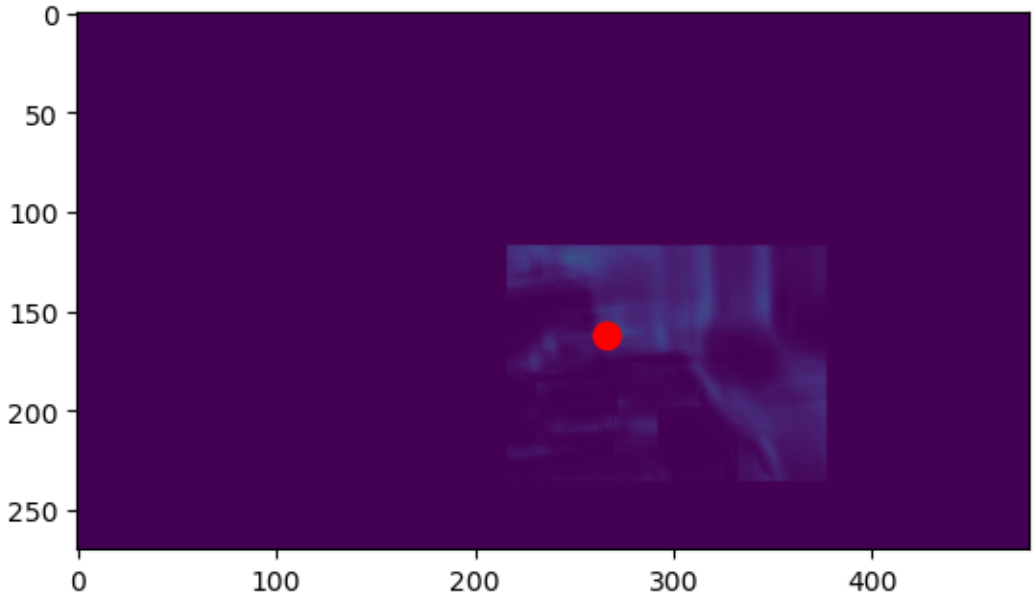

0
50
100
150
200
250
0
100
200
300
400

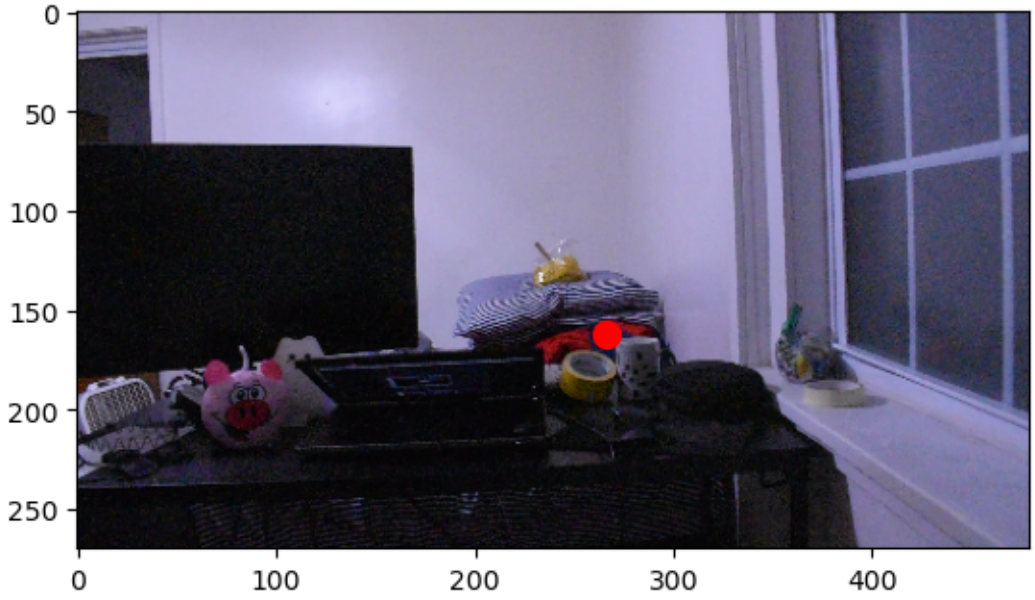

0
50
100
150
200
250
0
100
200
300
400



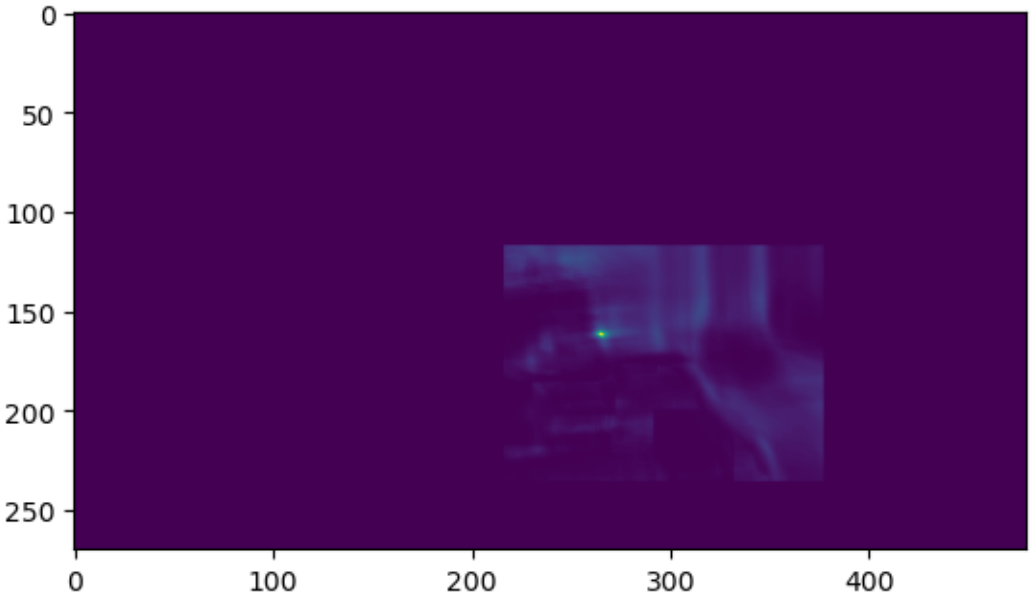
60%| | 61/102 [01:14<00:50, 1.24s/it]

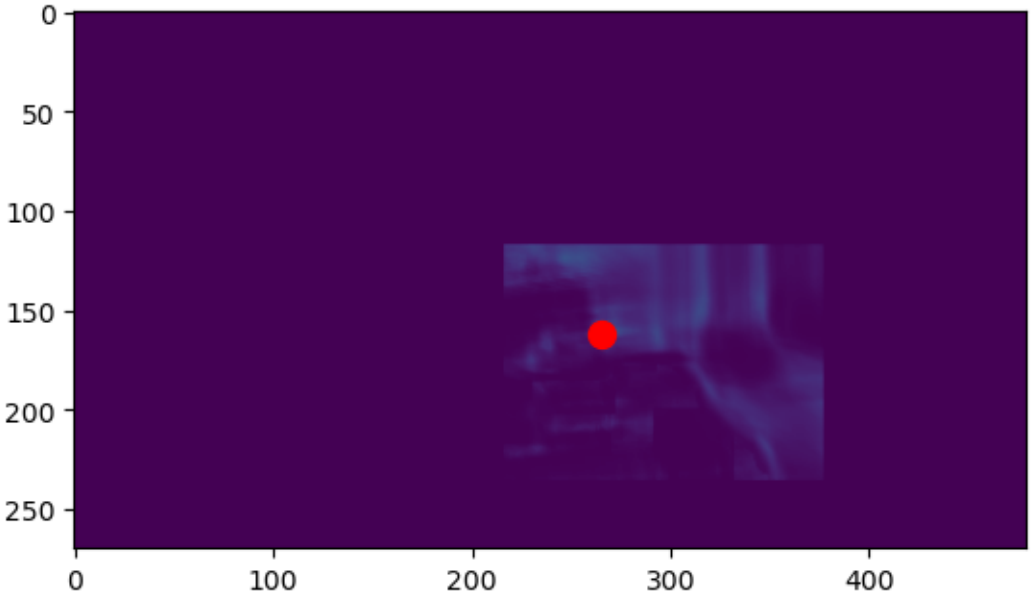


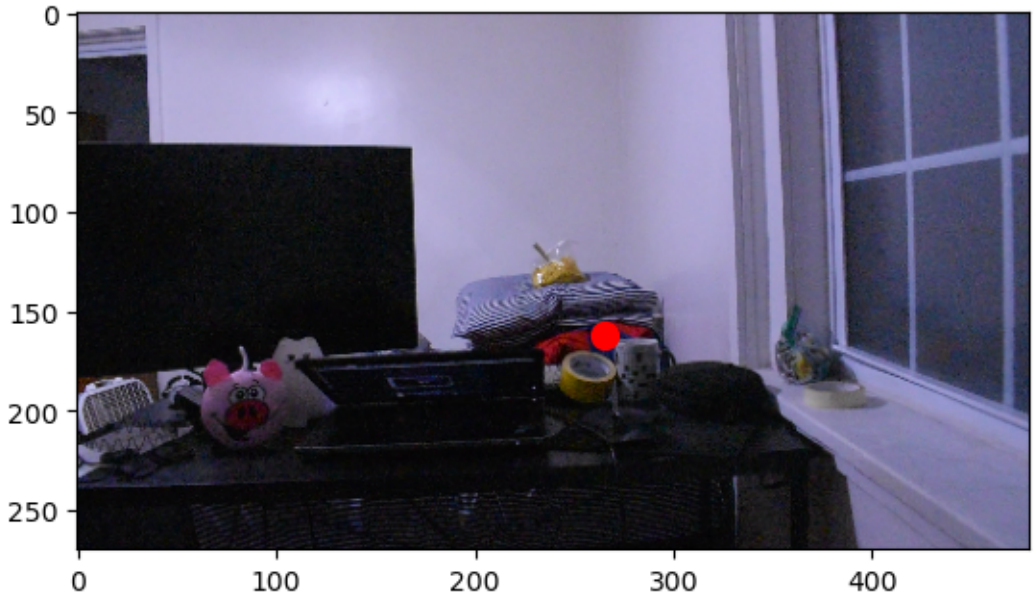




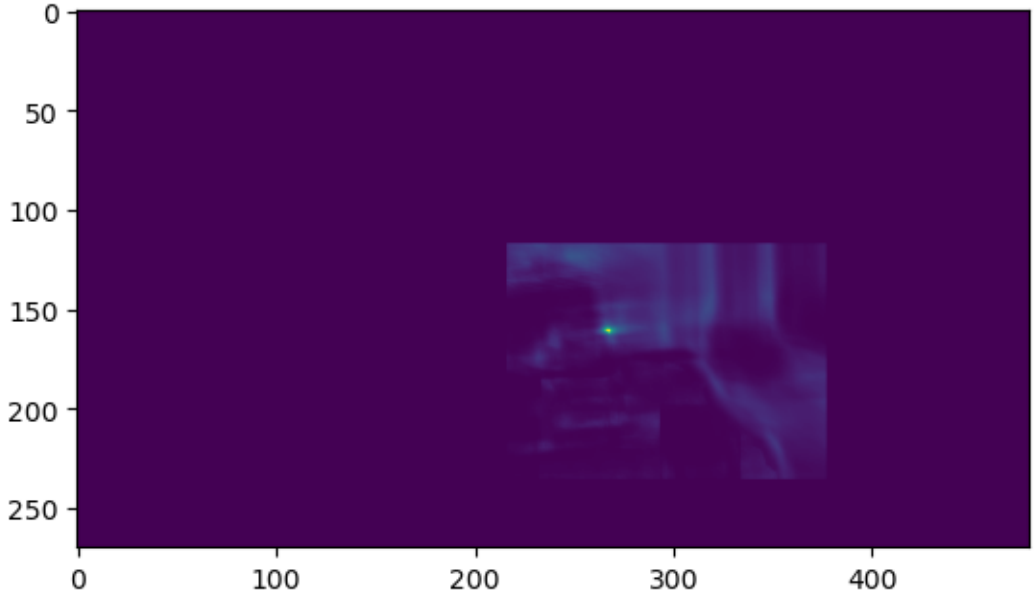
61%| | 62/102 [01:15<00:51, 1.29s/it]

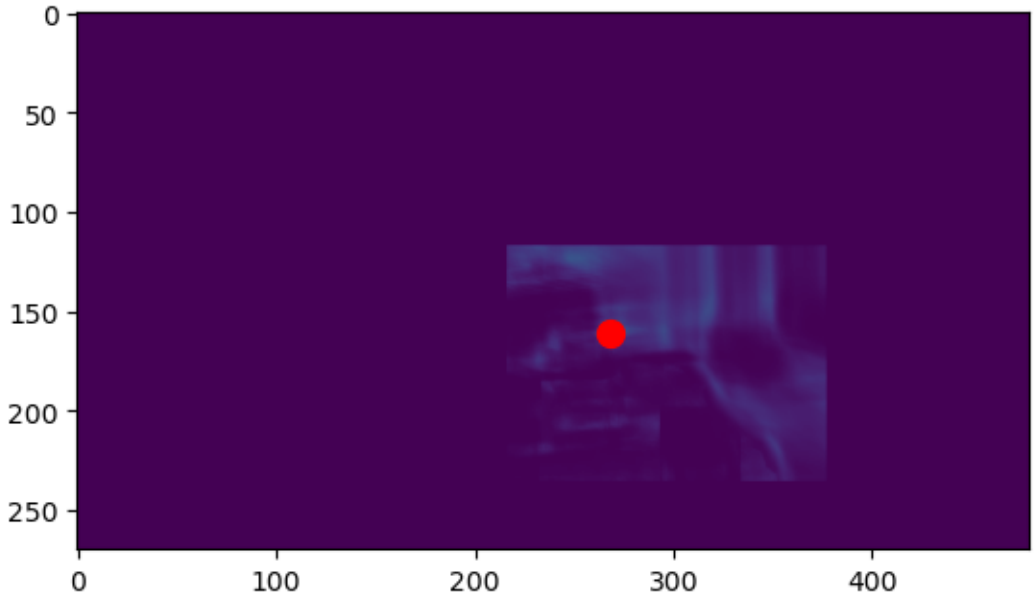

0
50
100
150
200
250
0
100
200
300
400

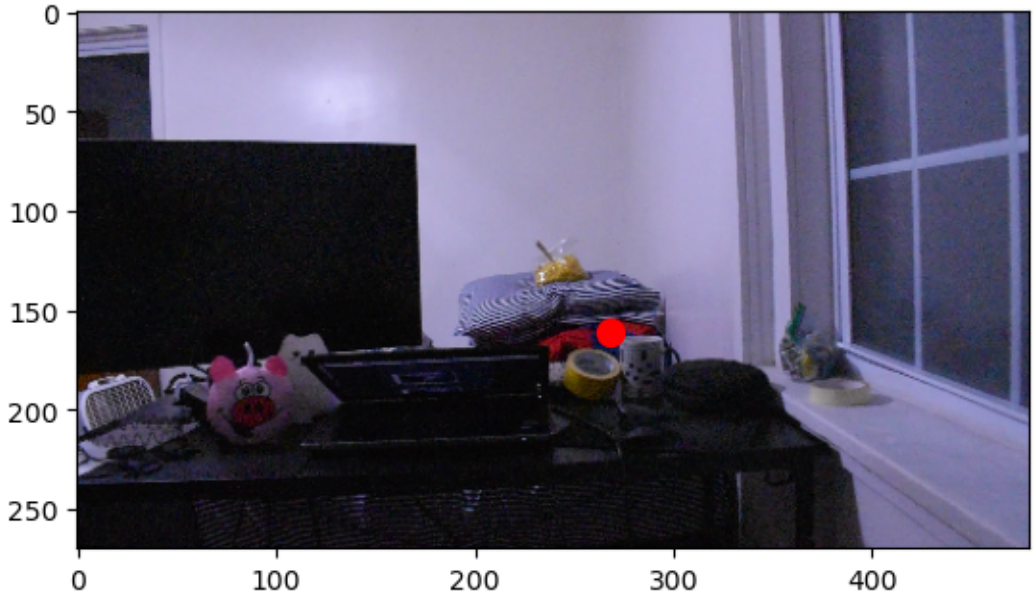

0
50
100
150
200
250
0
100
200
300
400



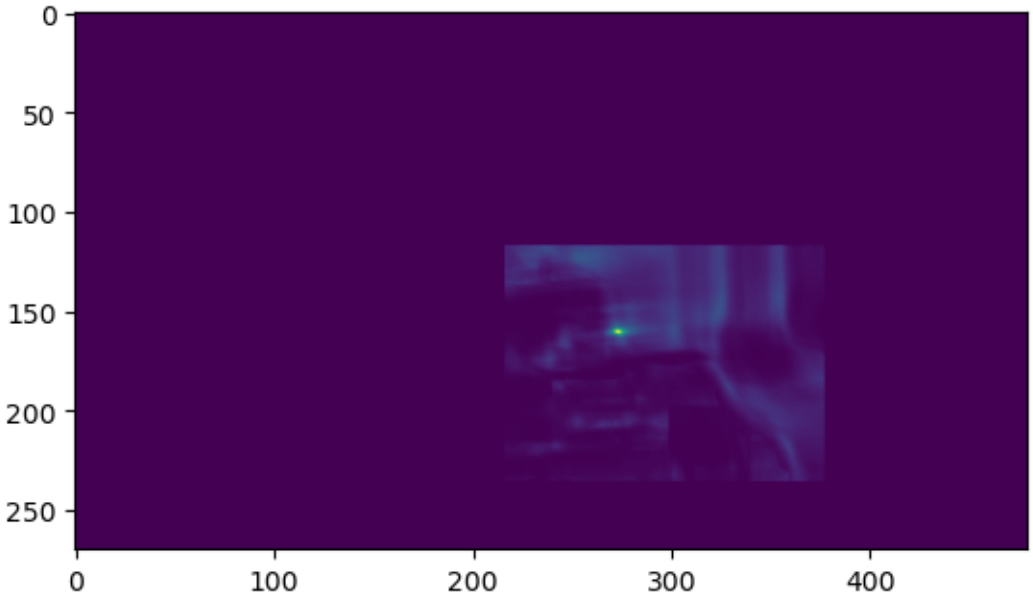
62%| | 63/102 [01:17<00:49, 1.28s/it]

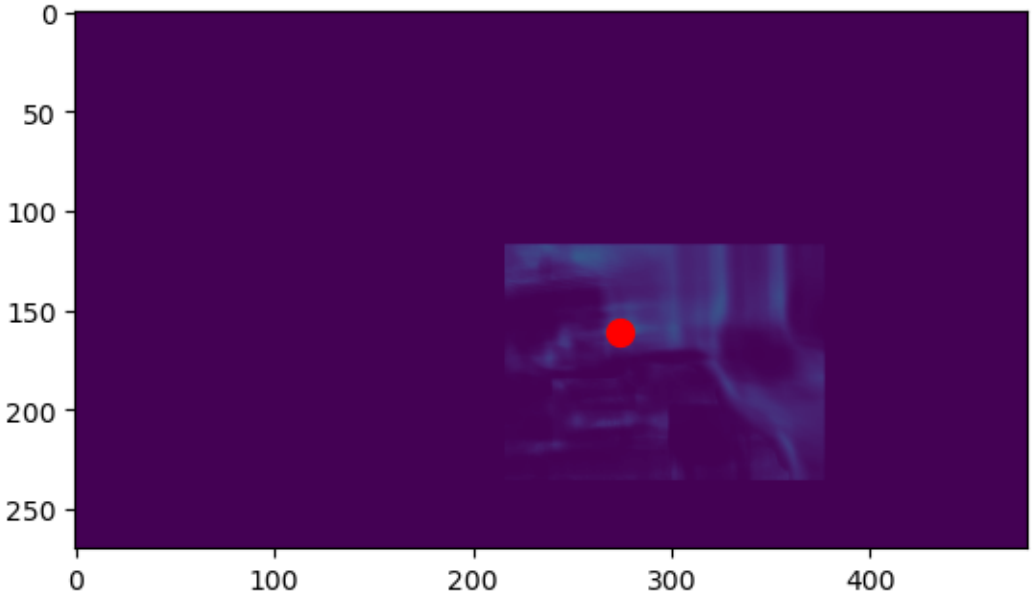


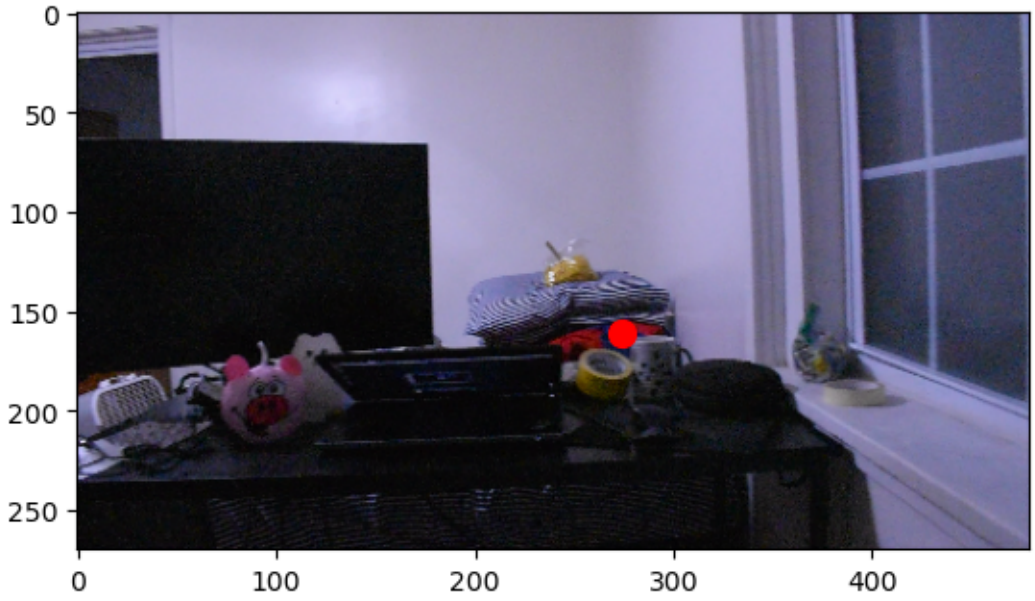




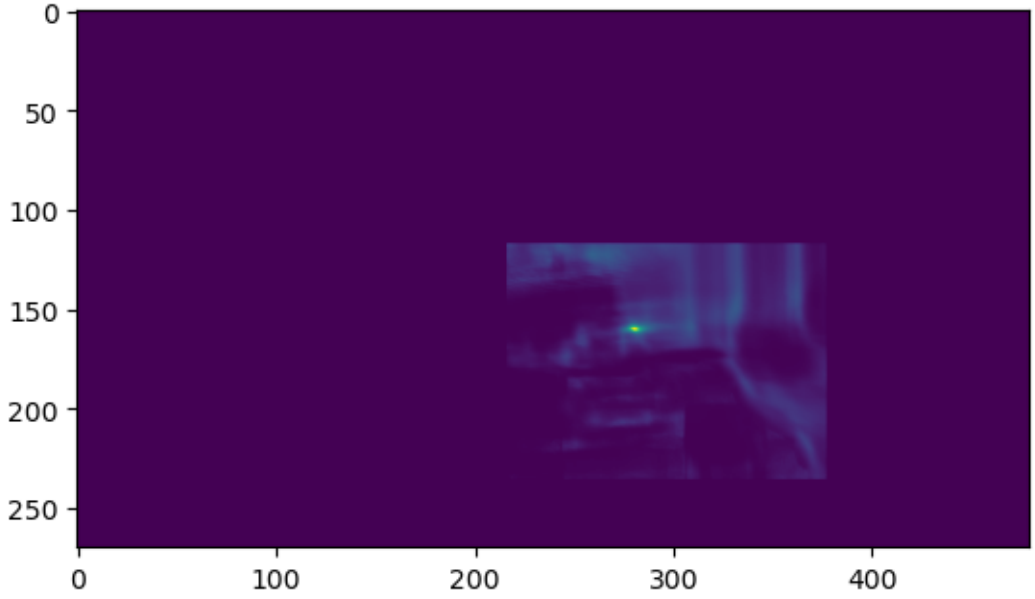
63%| | 64/102 [01:18<00:47, 1.26s/it]

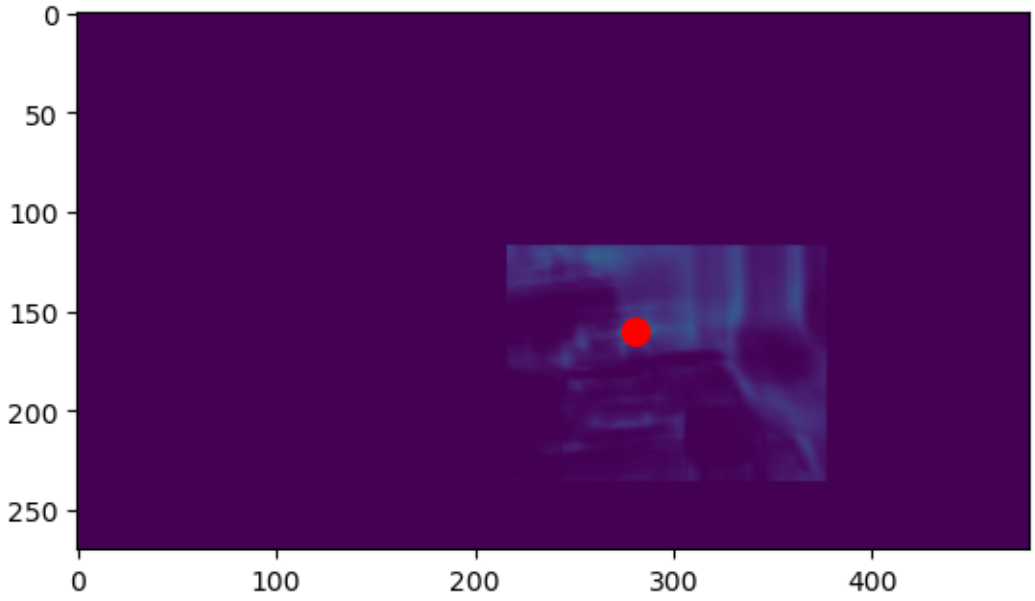

0
50
100
150
200
250
0
100
200
300
400

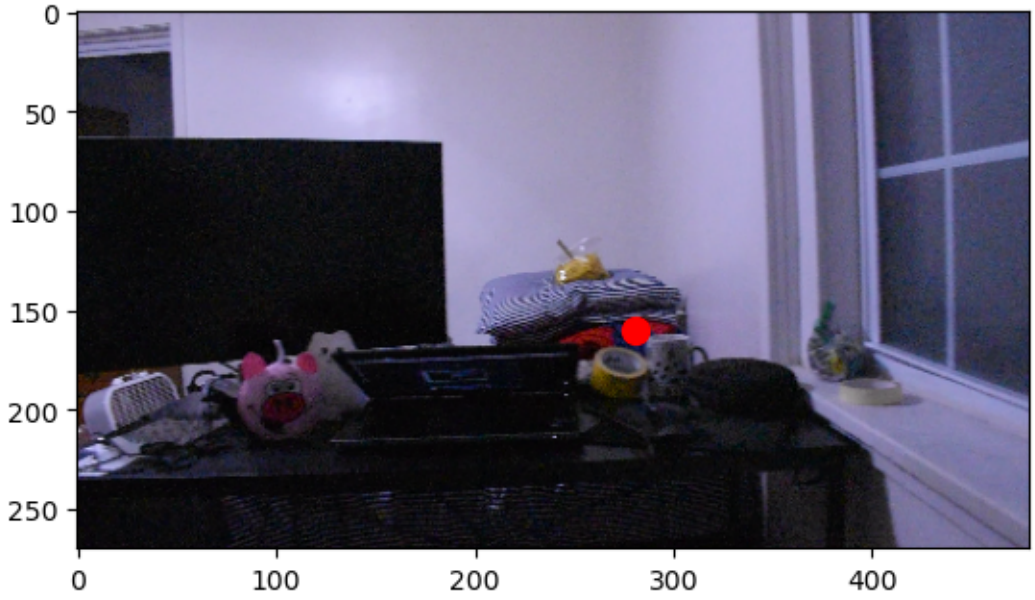

0
50
100
150
200
250
0
100
200
300
400



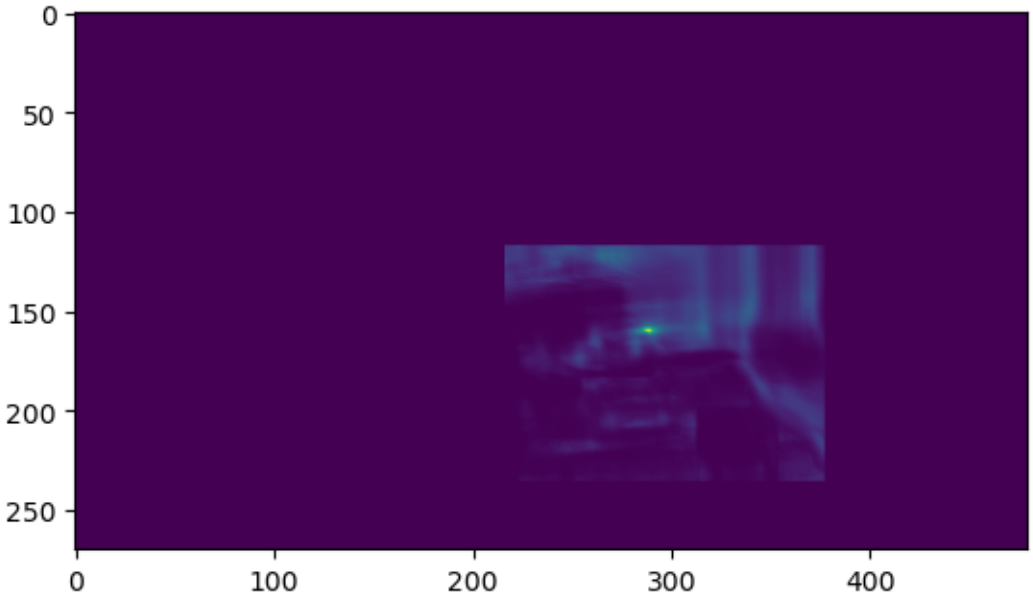
64%| | 65/102 [01:19<00:46, 1.24s/it]

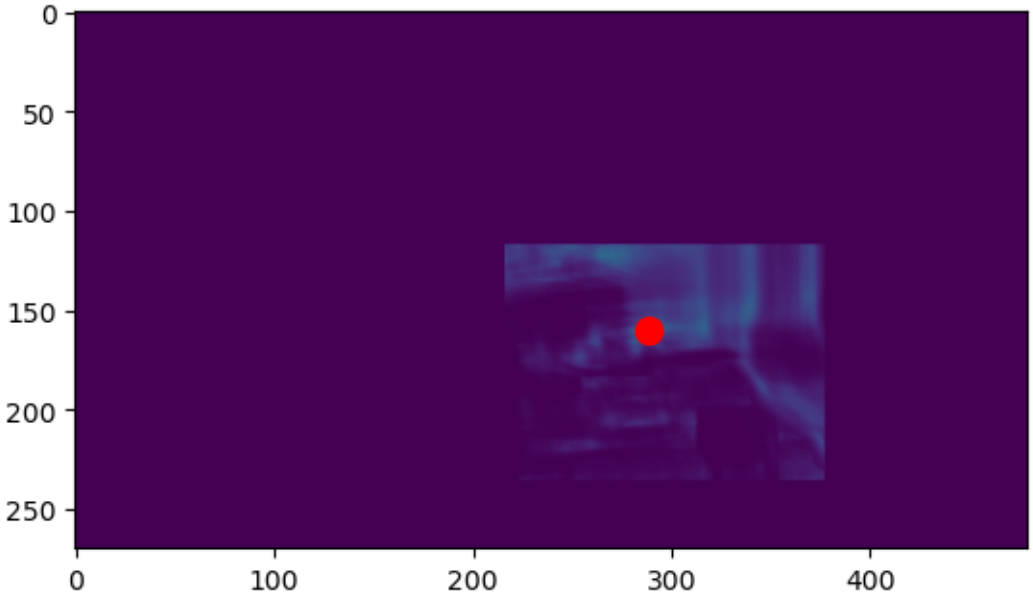


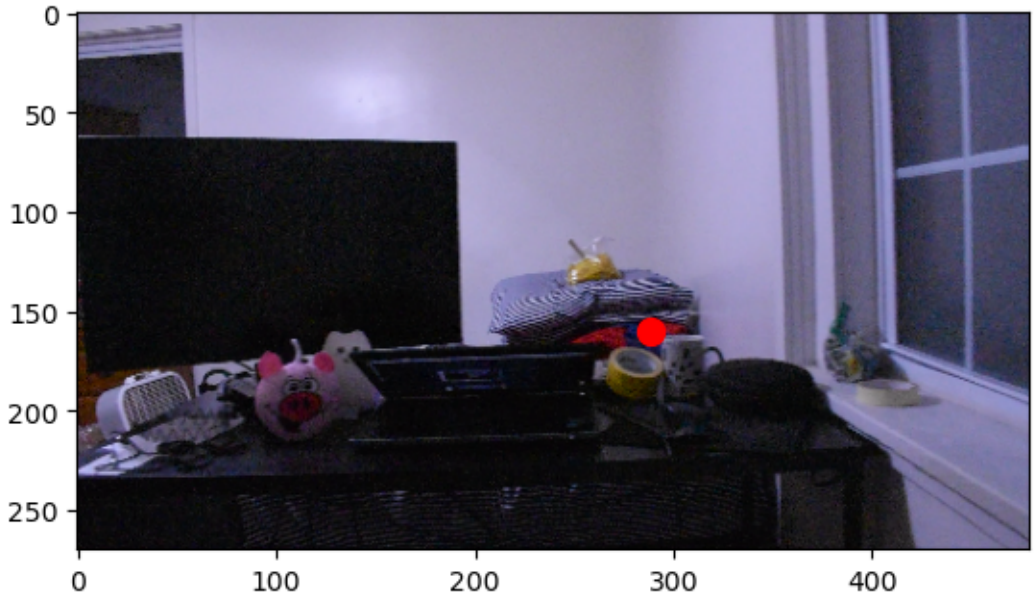




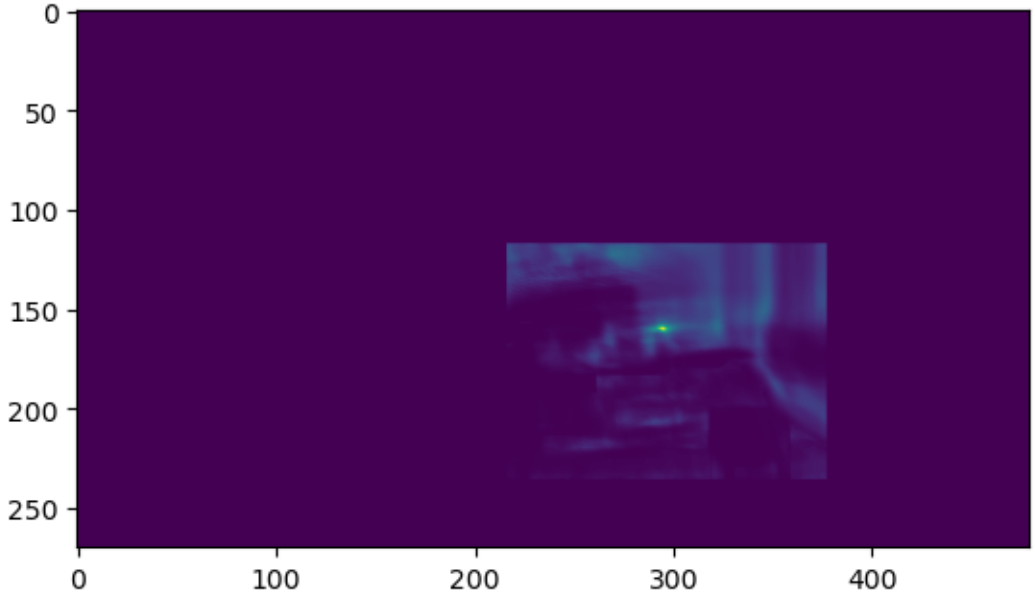
65%| | 66/102 [01:20<00:44, 1.24s/it]

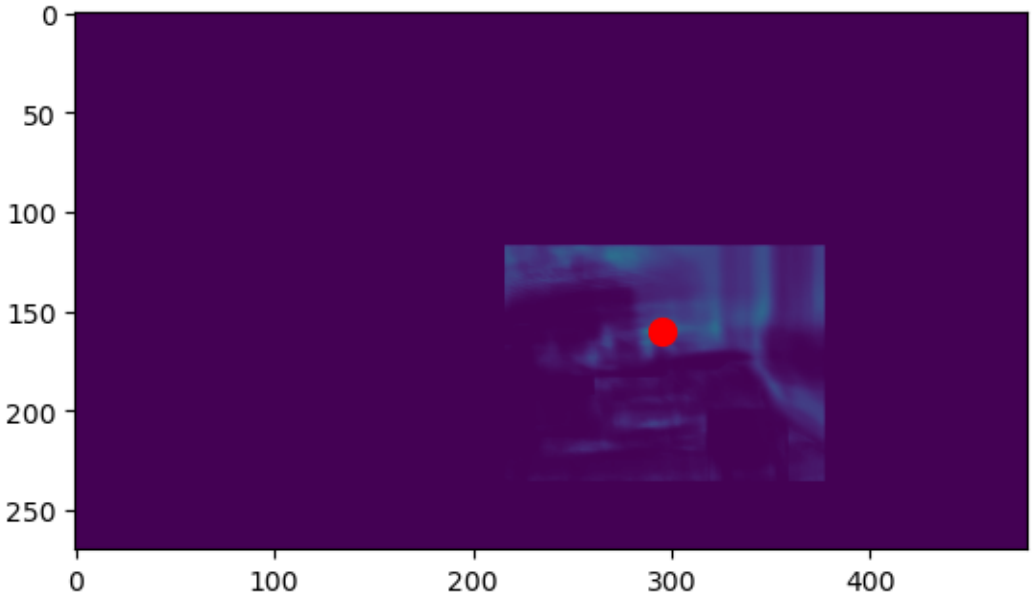

0
50
100
150
200
250
0
100
200
300
400

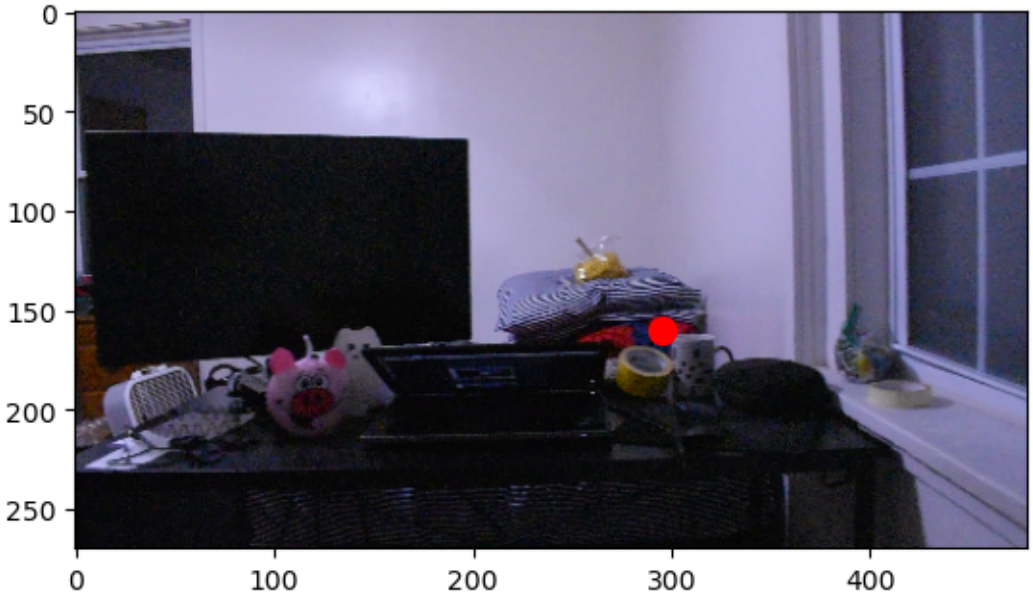

0
50
100
150
200
250
0
100
200
300
400



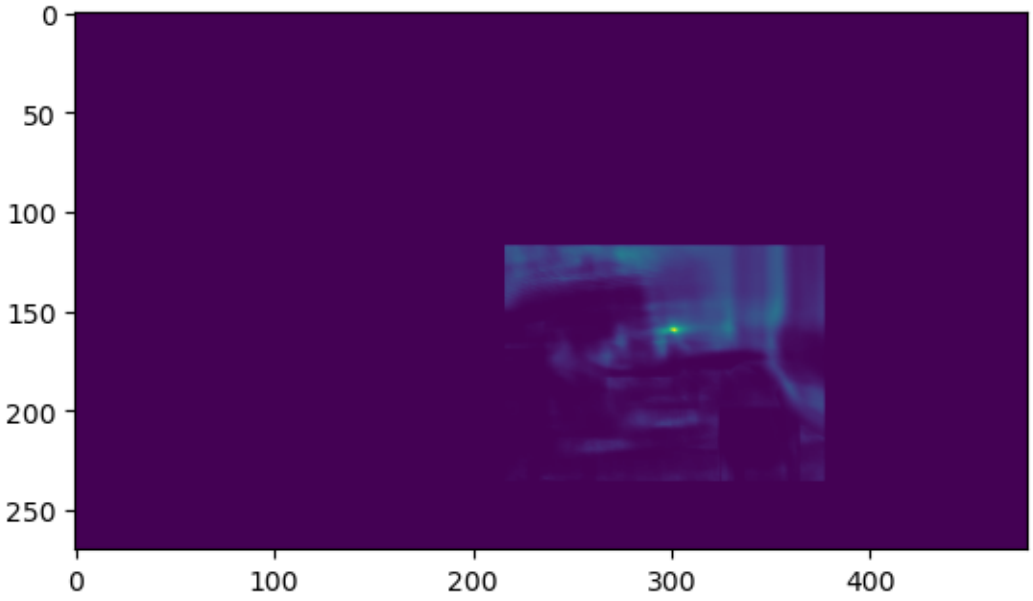
66%| | 67/102 [01:21<00:42, 1.22s/it]

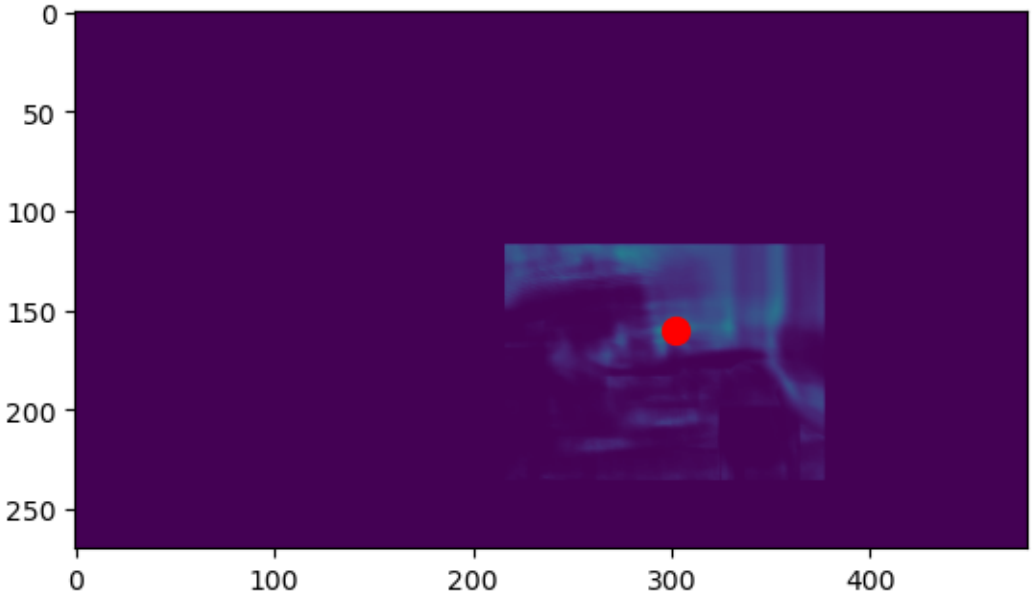


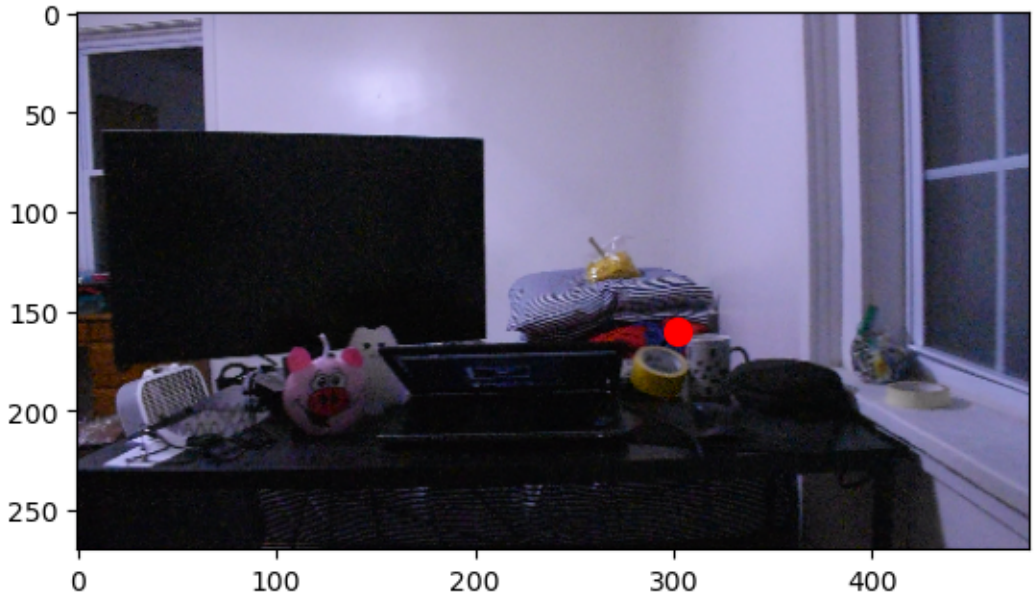




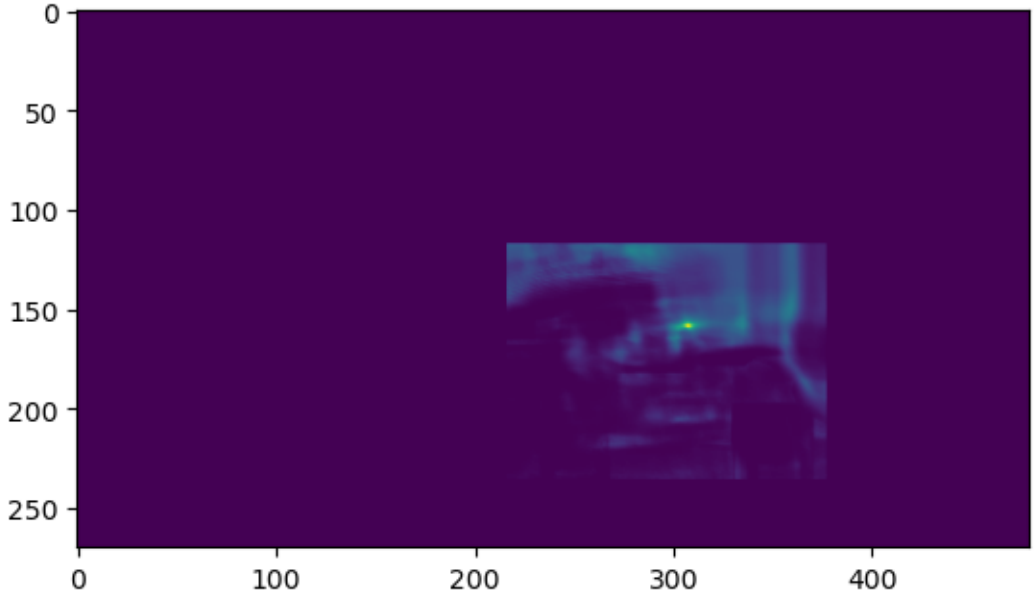
67%| | 68/102 [01:23<00:41, 1.22s/it]

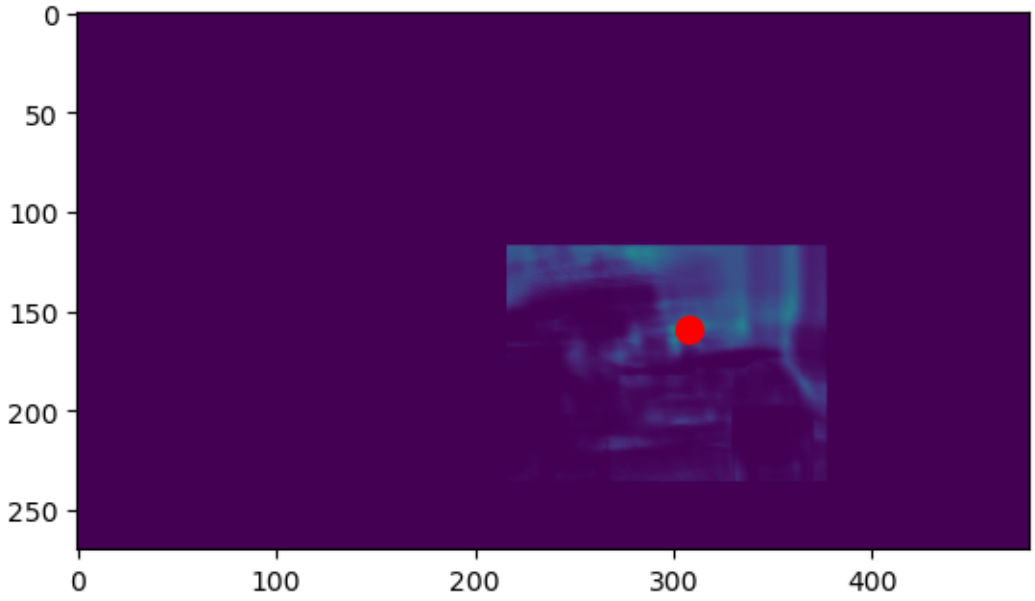

0
50
100
150
200
250
0
100
200
300
400

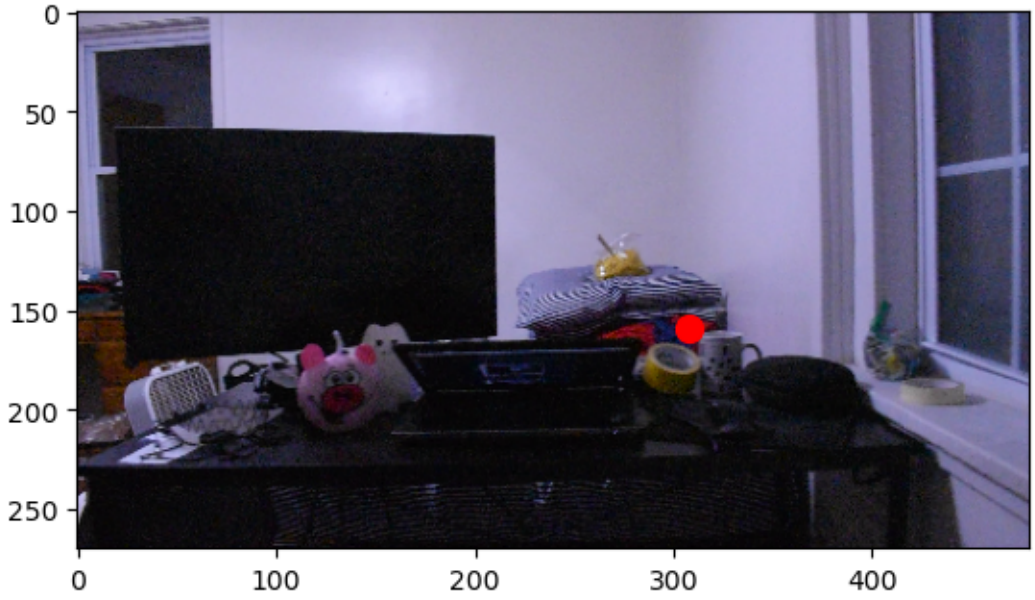

0
50
100
150
200
250
0
100
200
300
400



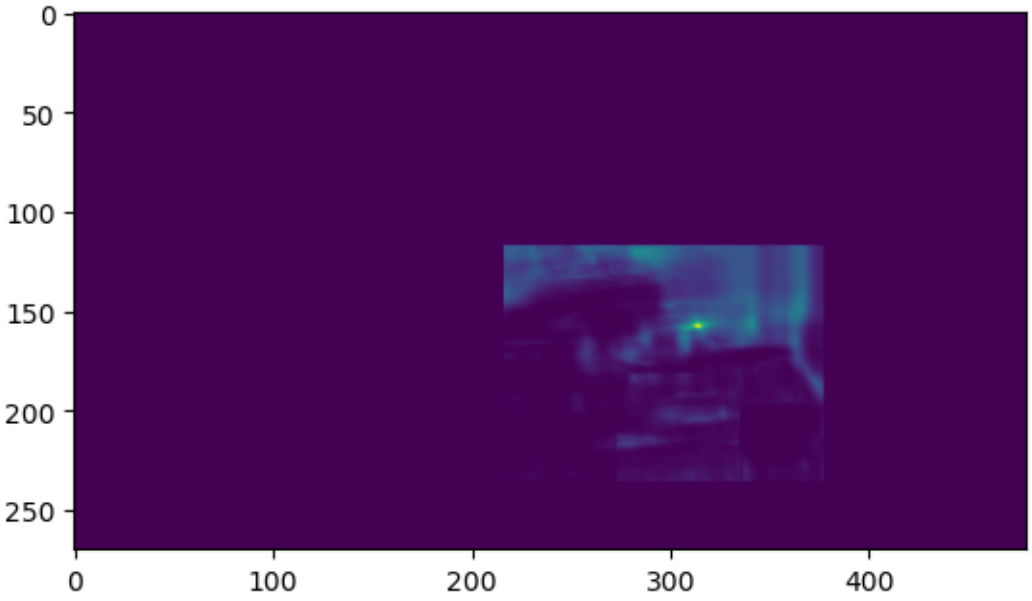
68%| | 69/102 [01:24<00:41, 1.26s/it]

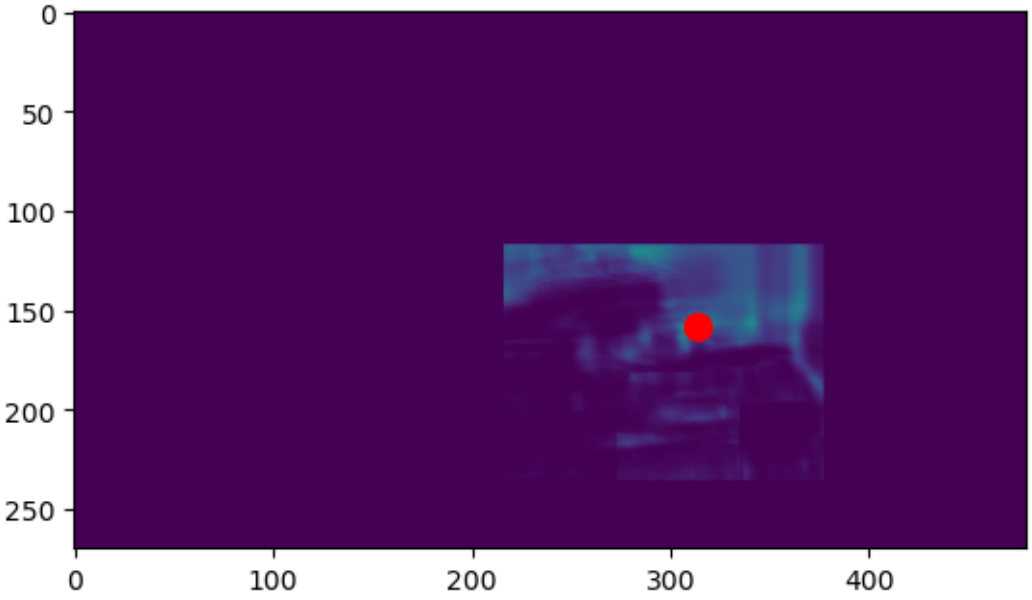


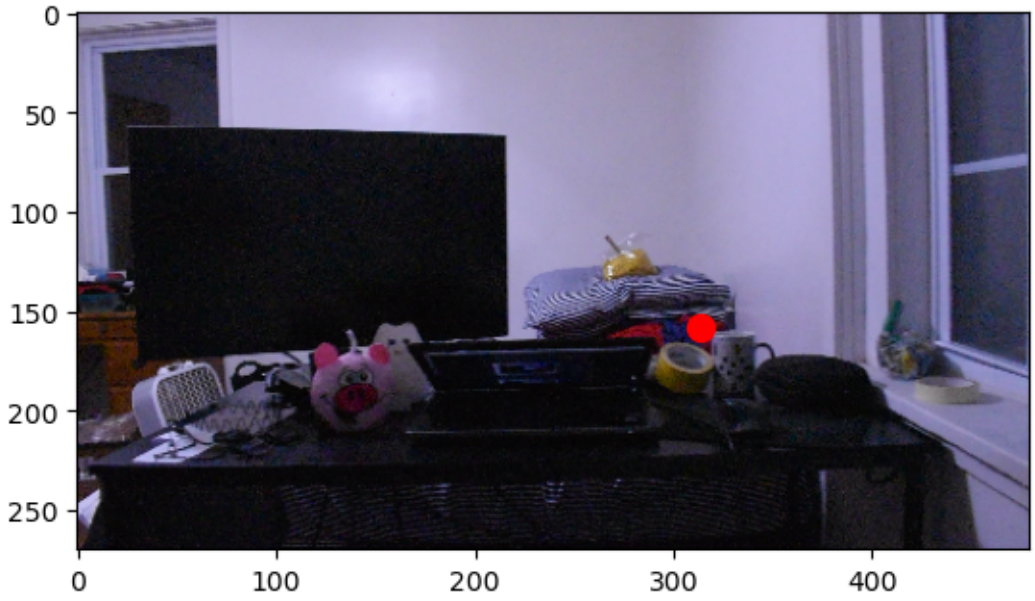




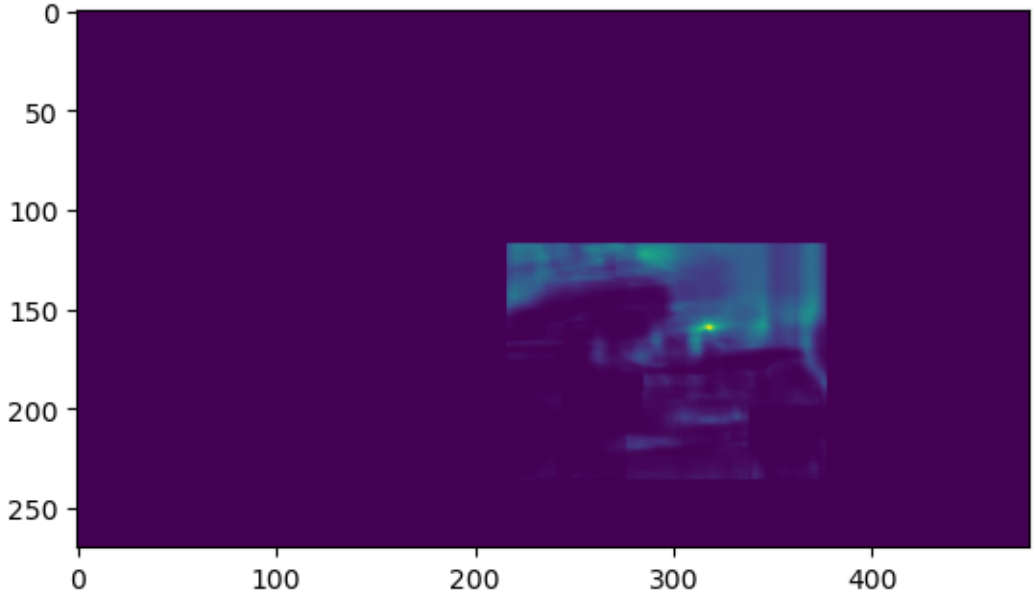
69%| | 70/102 [01:25<00:39, 1.24s/it]

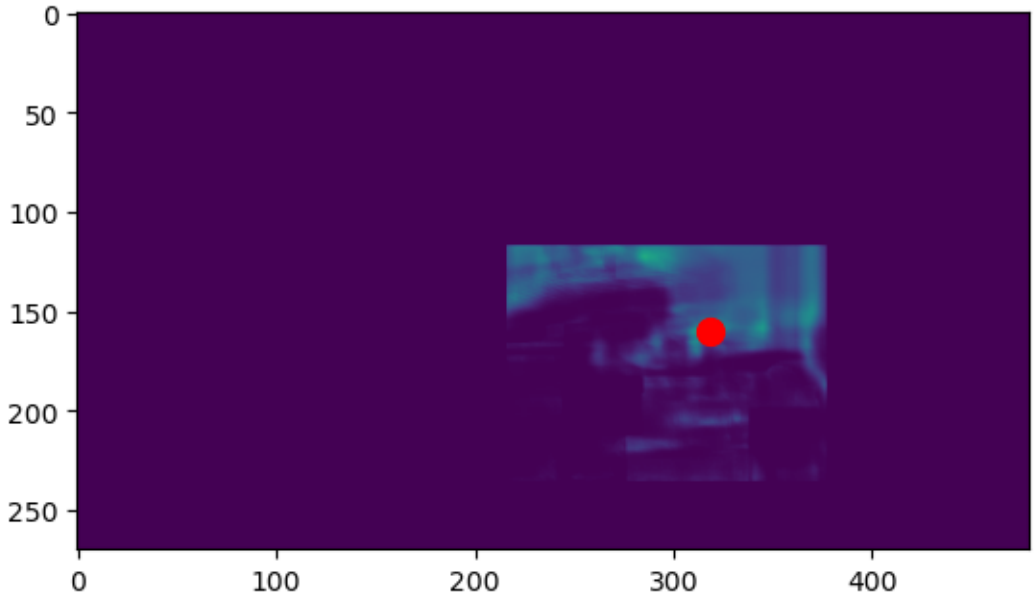

0
50
100
150
200
250
0
100
200
300
400

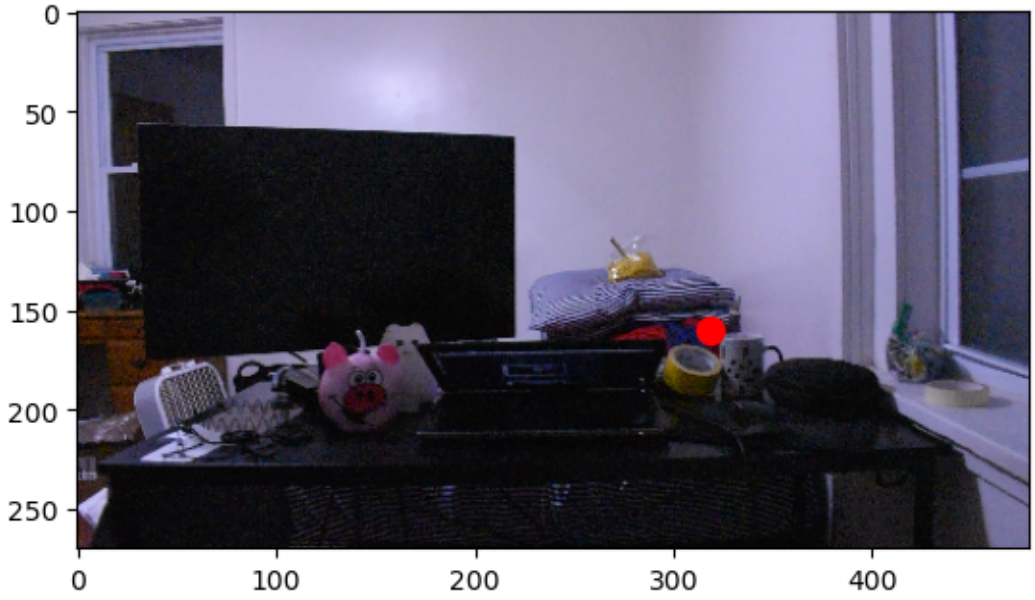

0
50
100
150
200
250
0
100
200
300
400



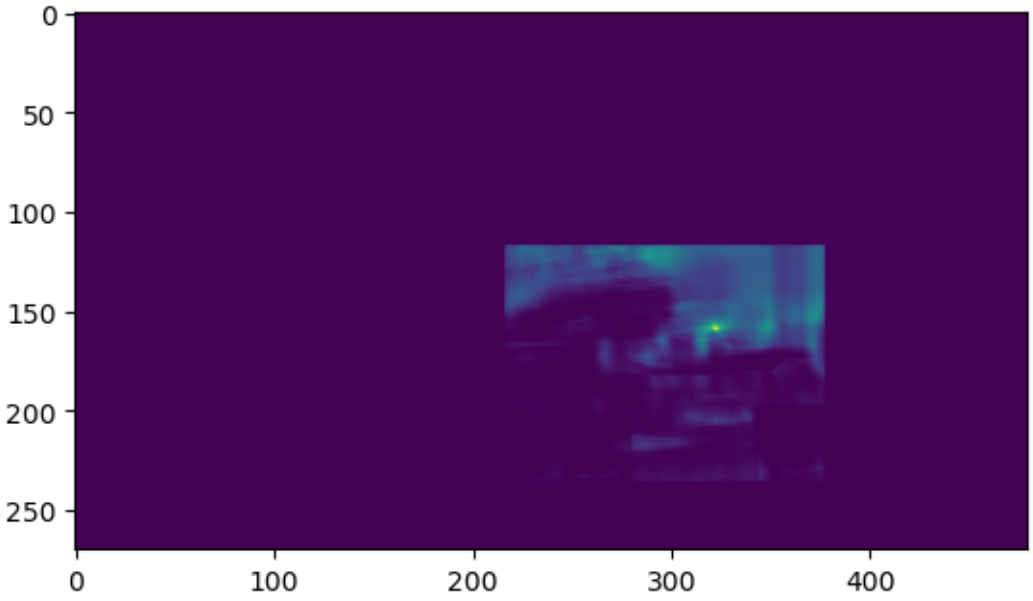
70%| | 71/102 [01:26<00:38, 1.23s/it]

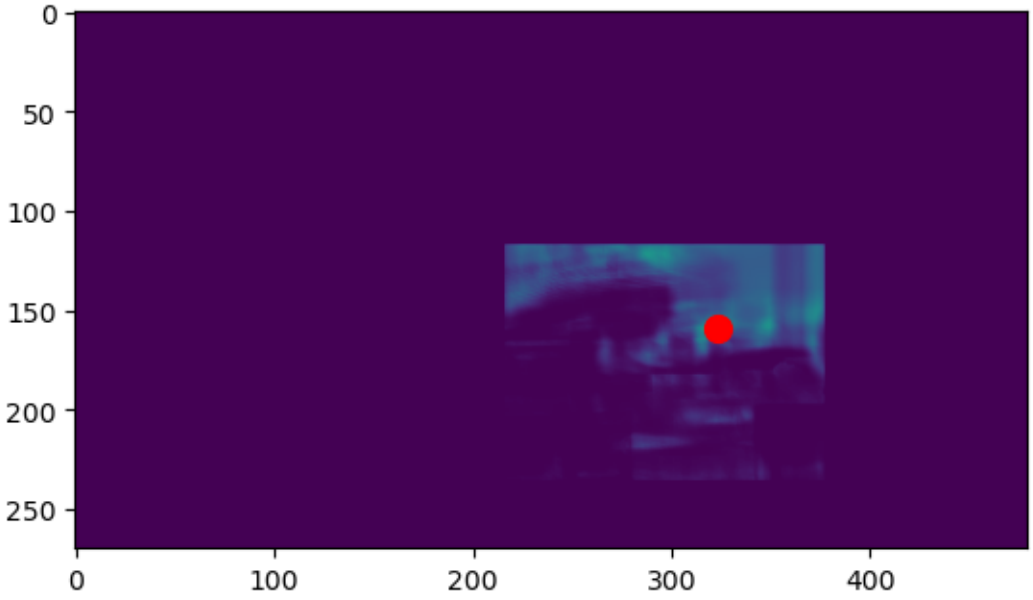


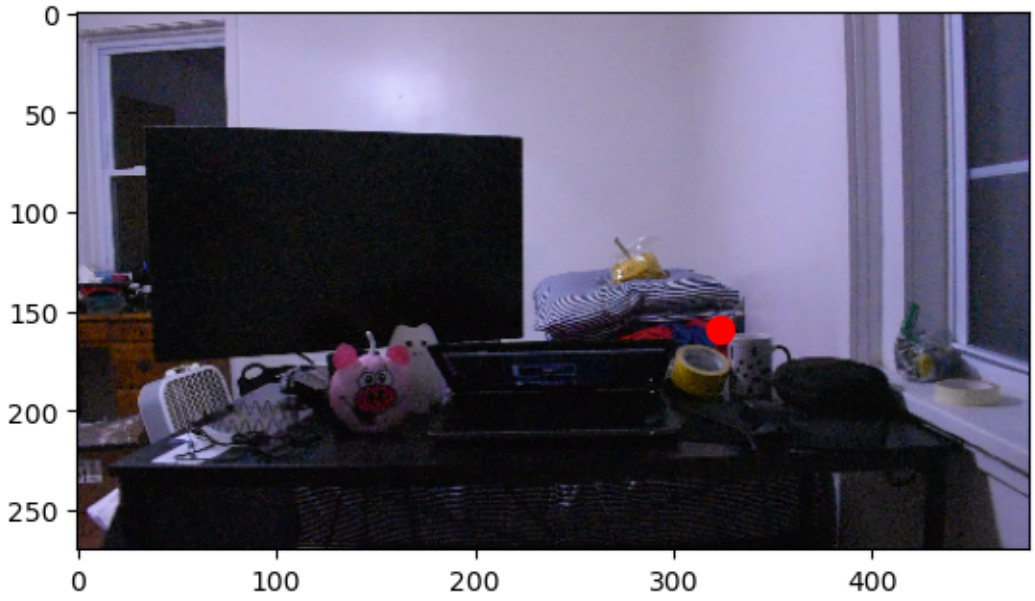




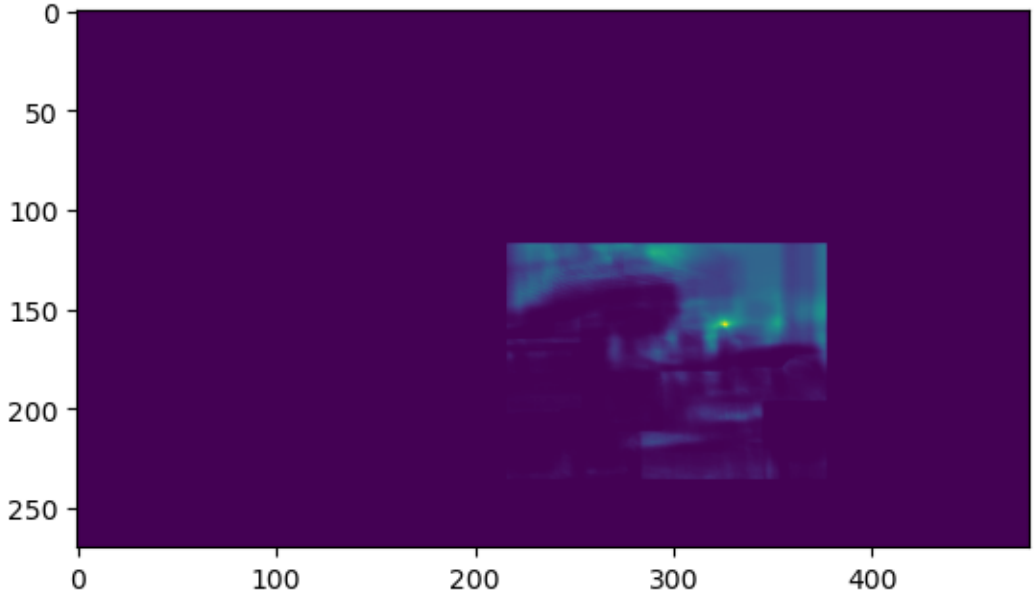
71%| | 72/102 [01:28<00:37, 1.24s/it]

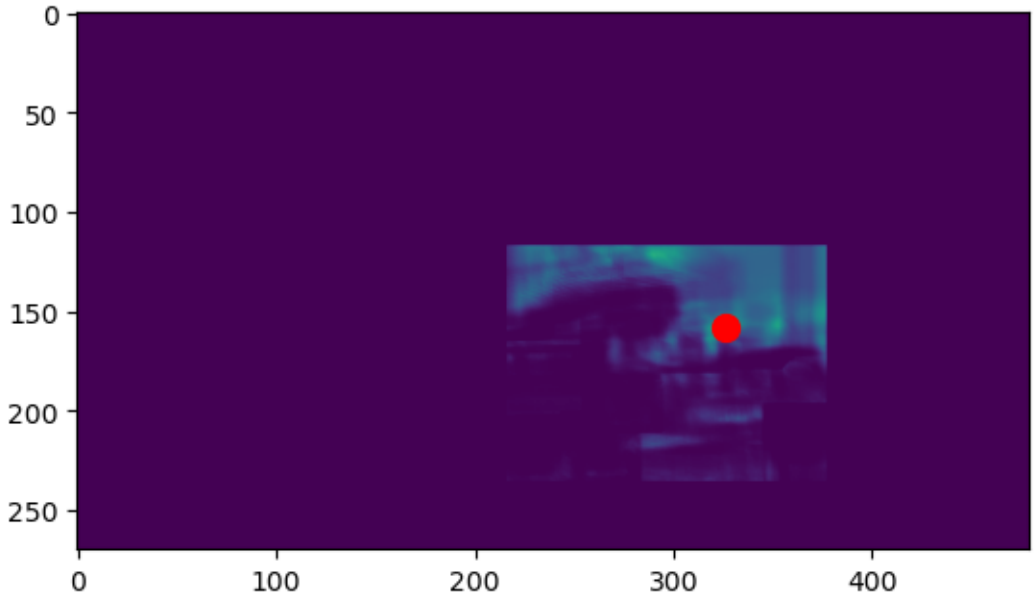

0
50
100
150
200
250
0
100
200
300
400

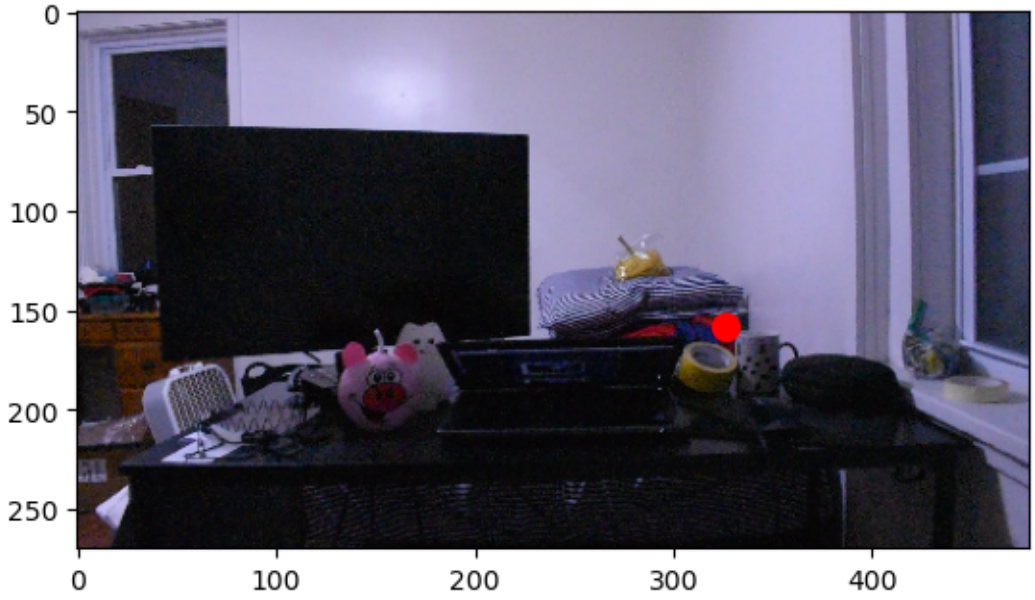

0
50
100
150
200
250
0
100
200
300
400



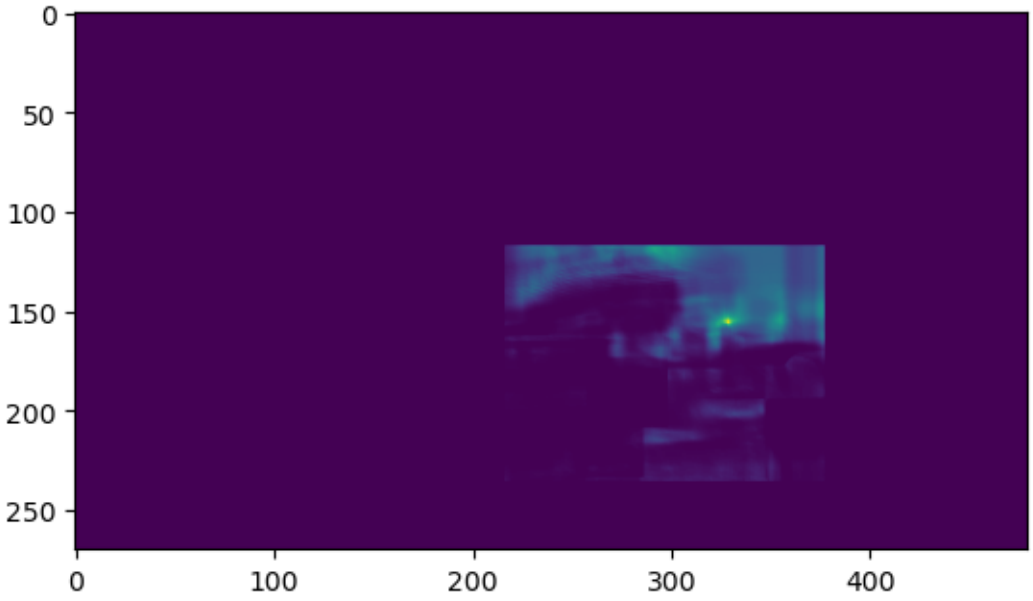
72%| | 73/102 [01:29<00:35, 1.23s/it]

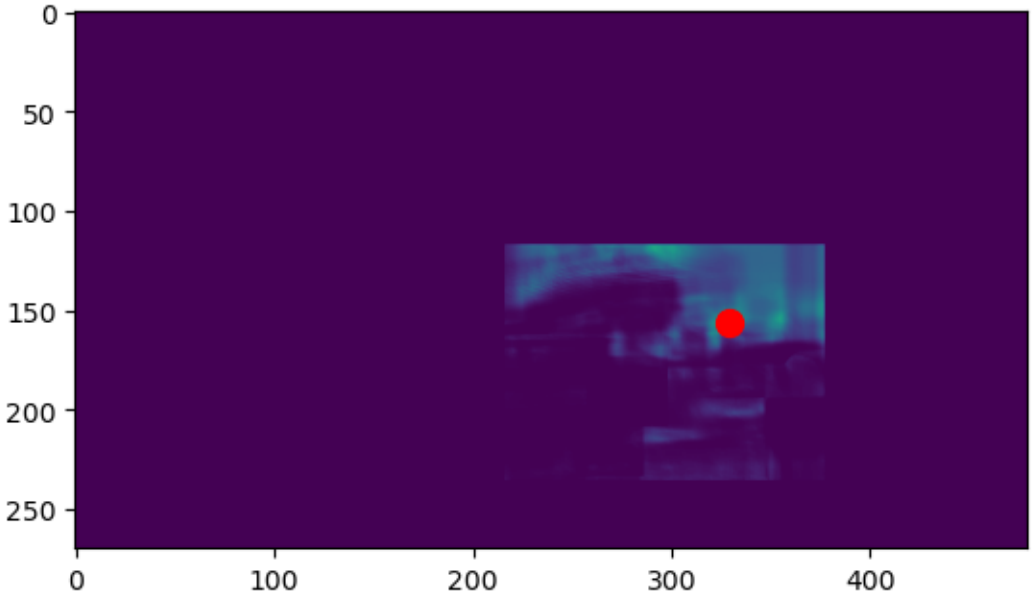


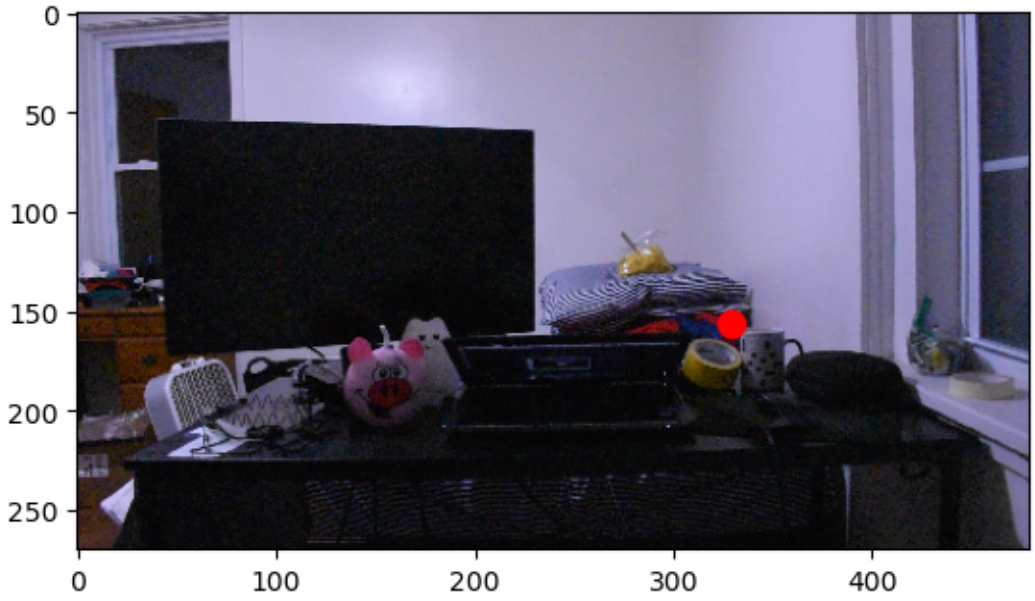




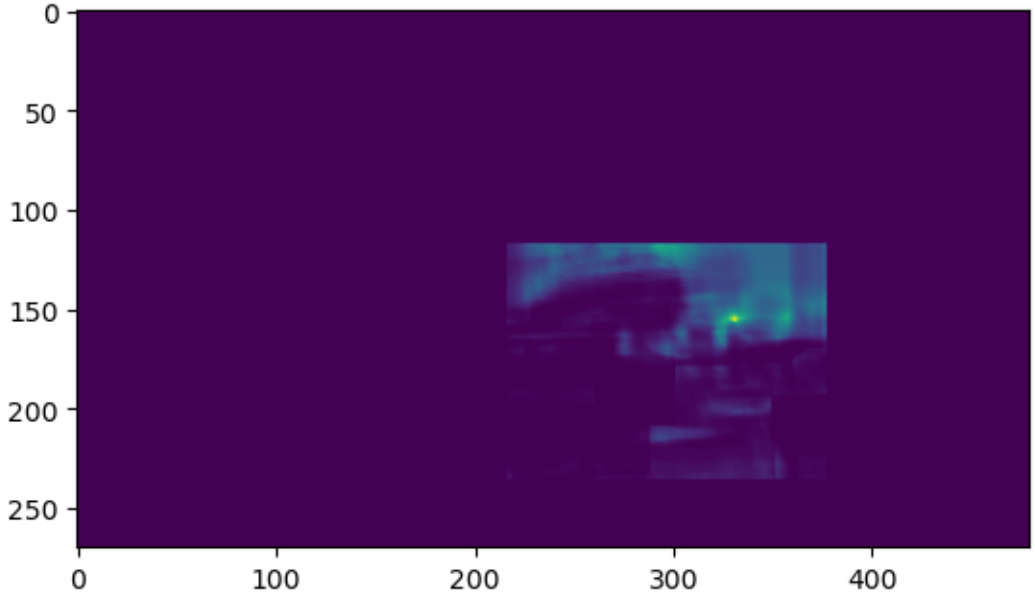
73%| | 74/102 [01:30<00:34, 1.24s/it]

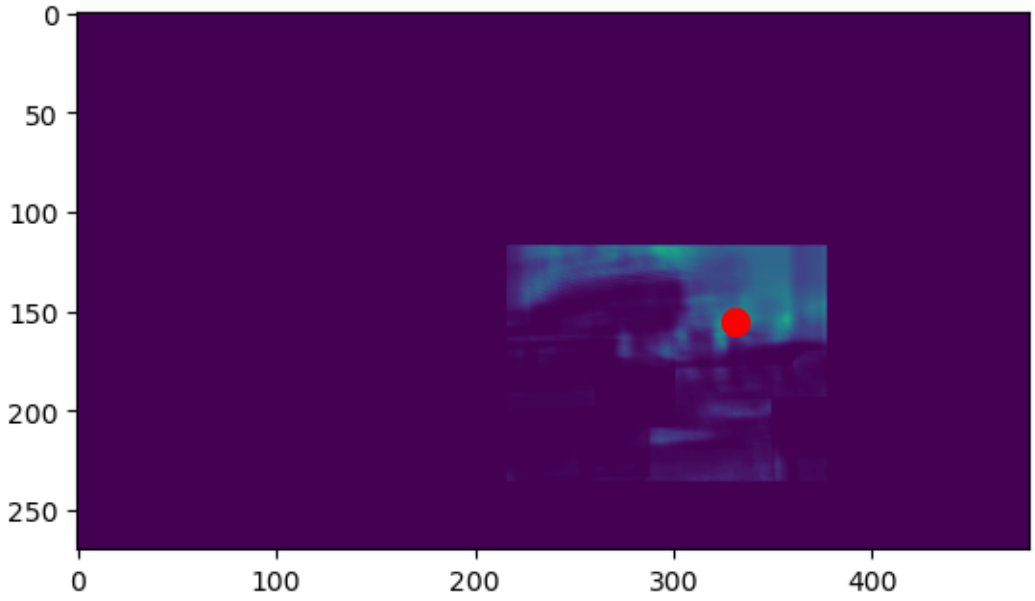

0
50
100
150
200
250
0
100
200
300
400

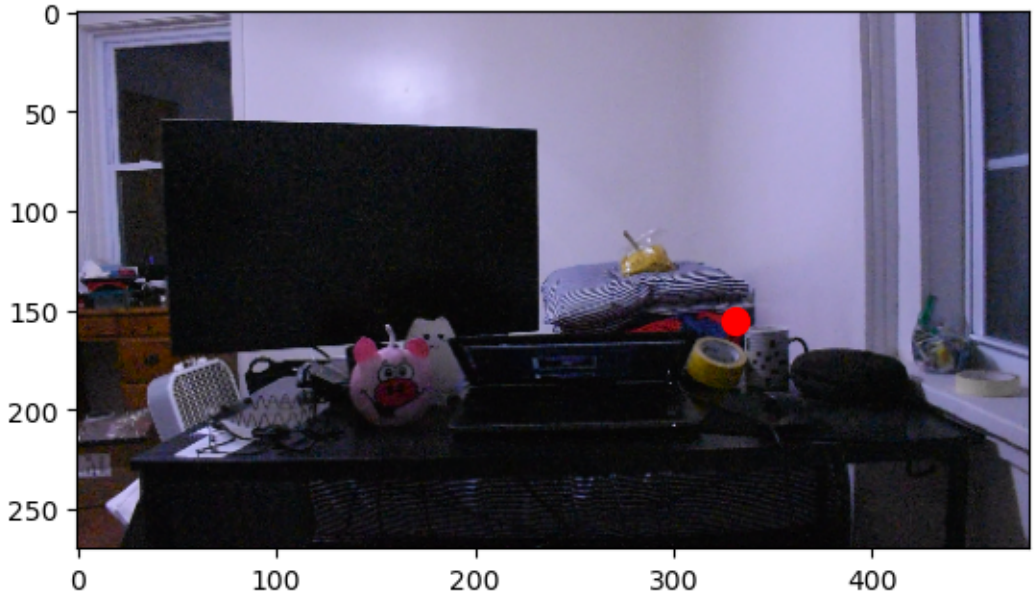

0
50
100
150
200
250
0
100
200
300
400



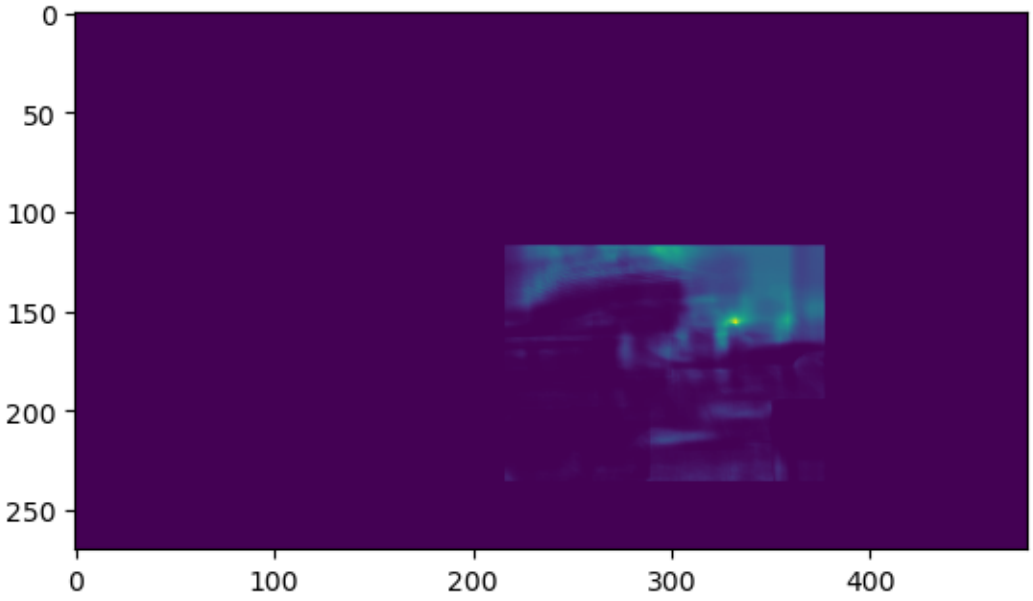
74%| | 75/102 [01:31<00:33, 1.23s/it]

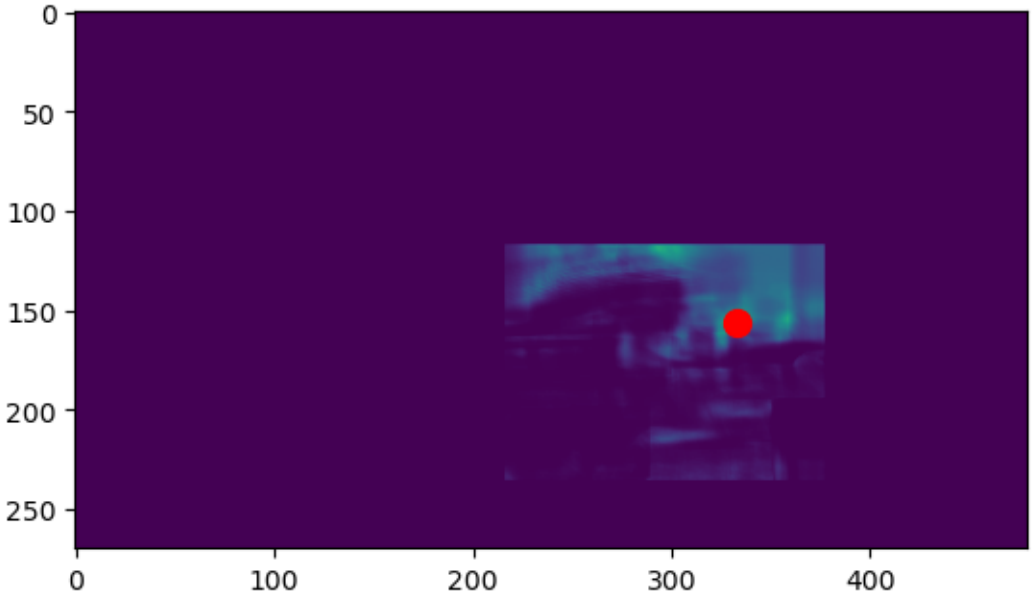


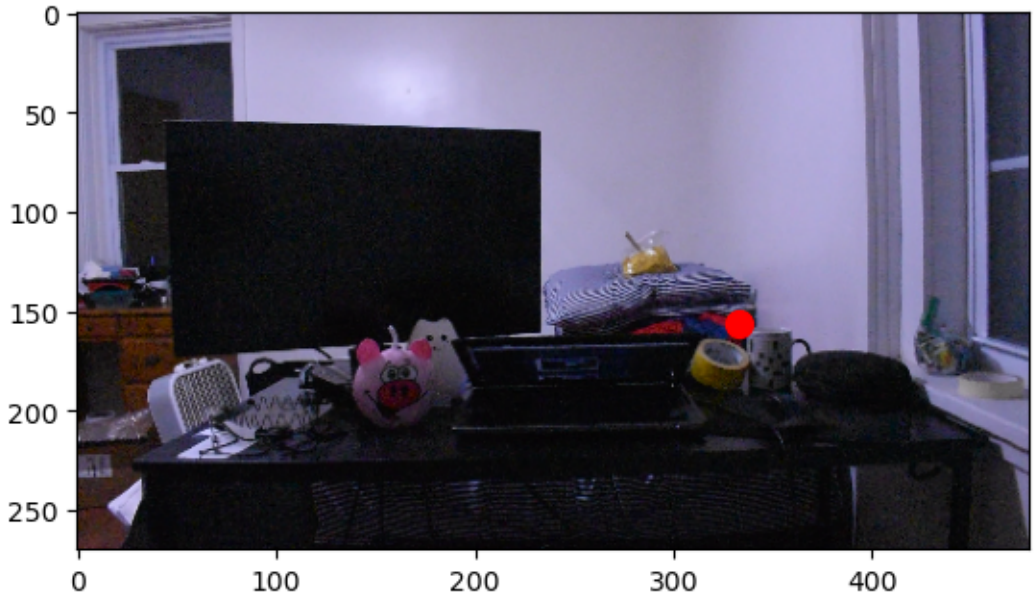




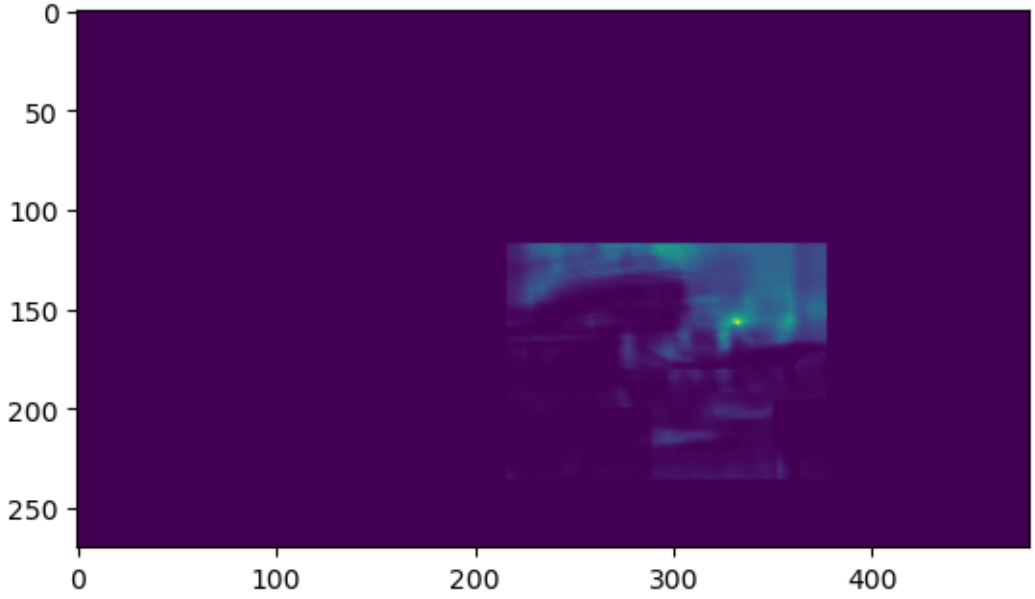
75%| | 76/102 [01:33<00:32, 1.25s/it]

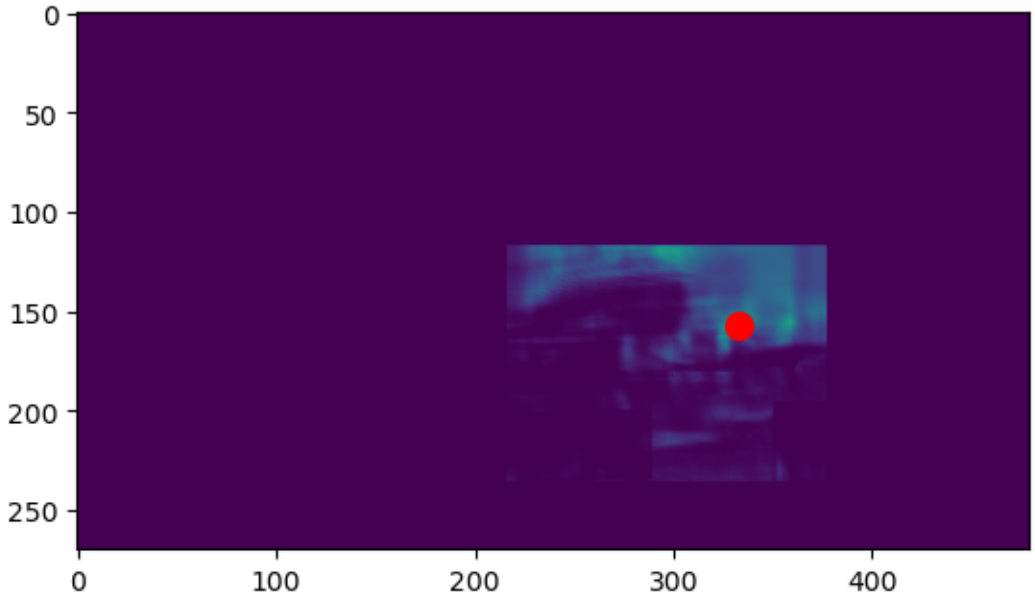

0
50
100
150
200
250
0
100
200
300
400

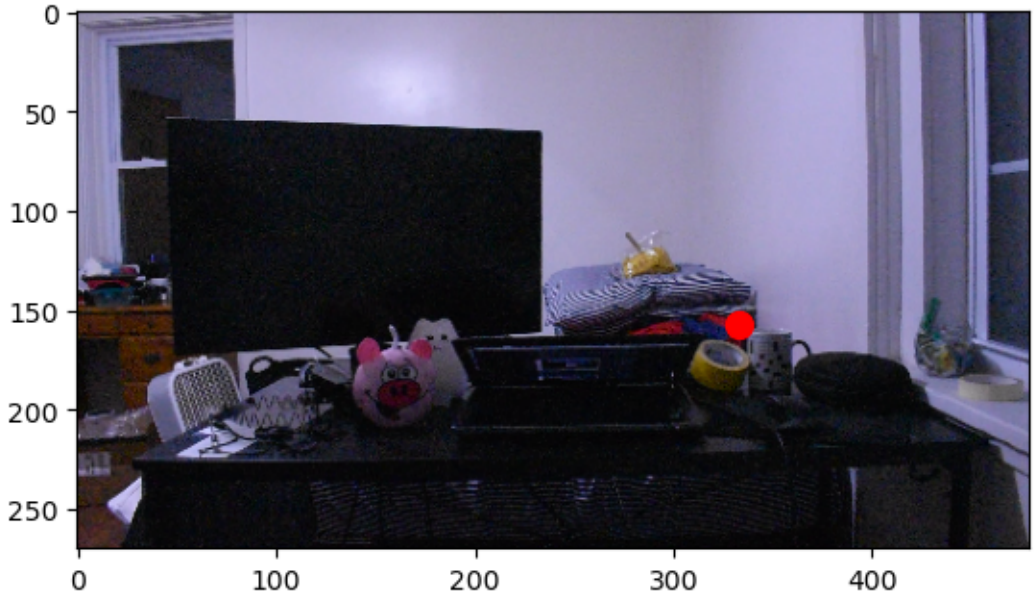

0
50
100
150
200
250
0
100
200
300
400



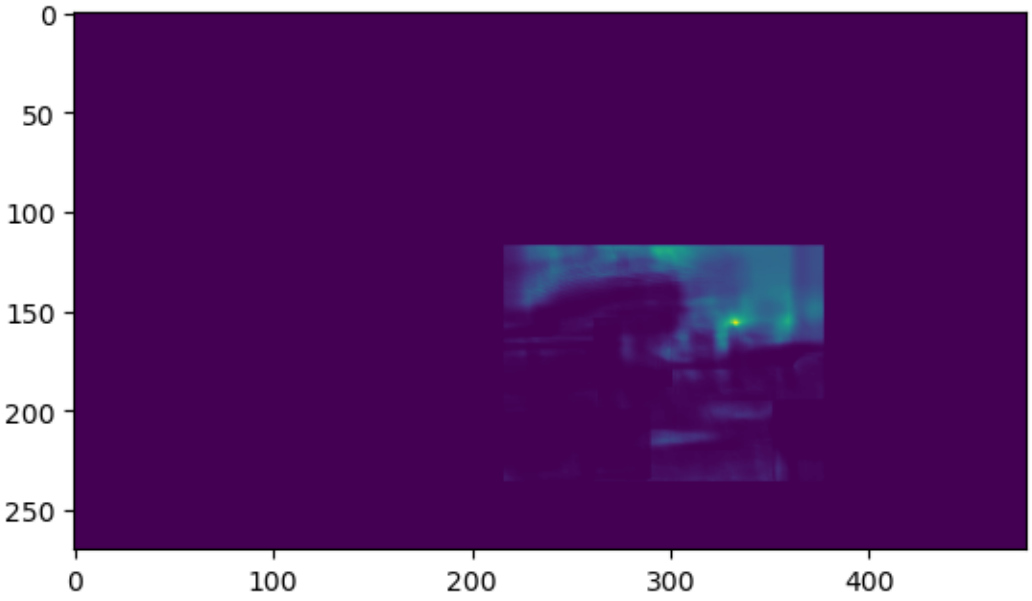
75%| | 77/102 [01:34<00:31, 1.25s/it]

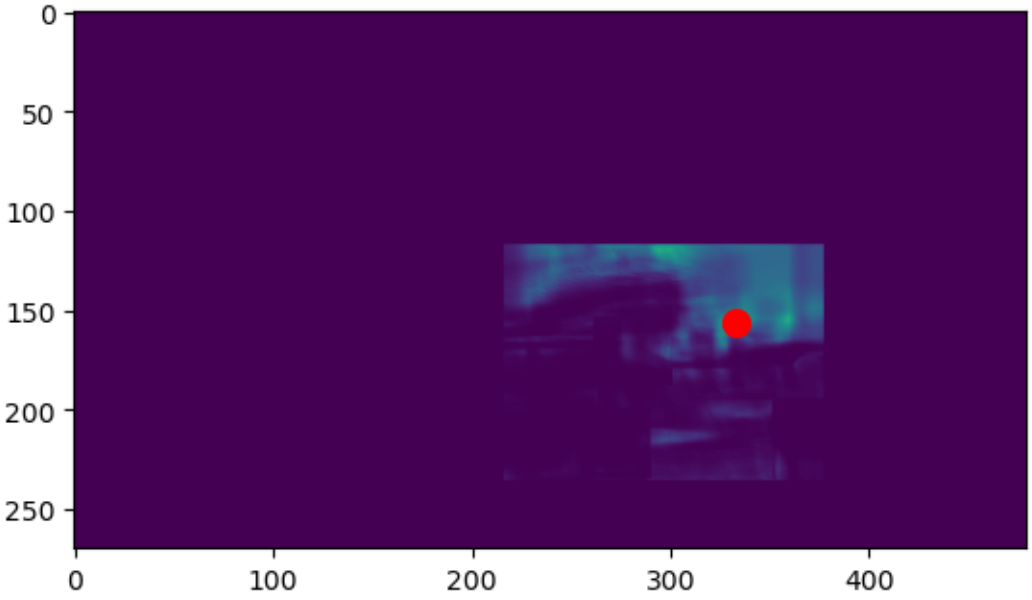


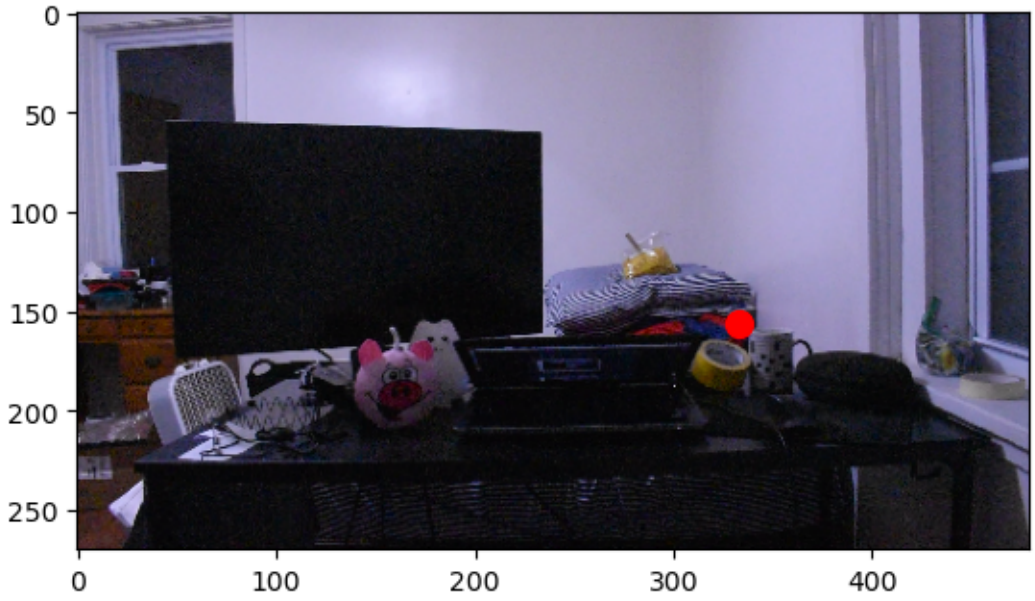




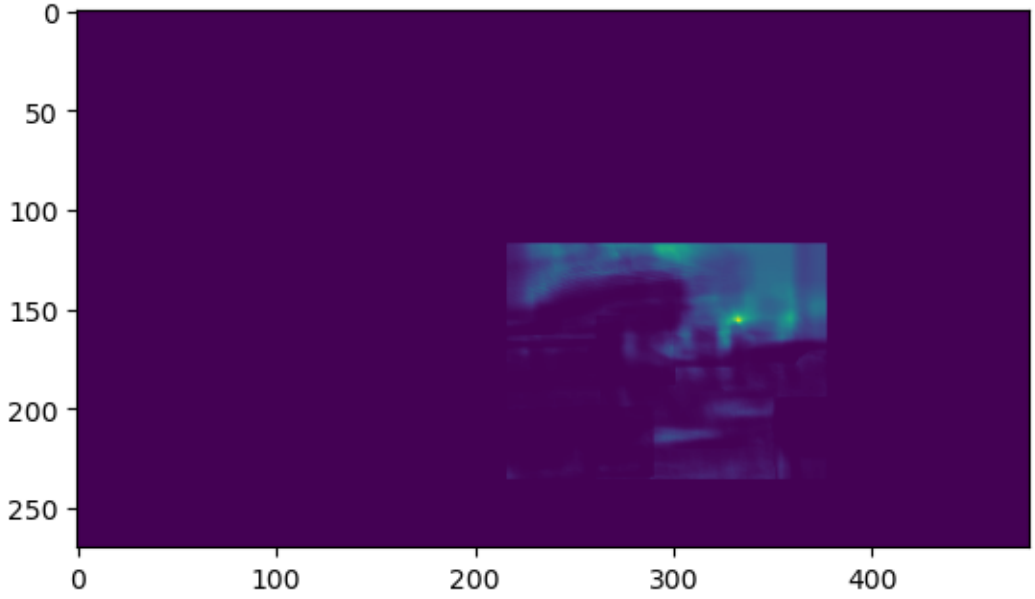
76%| | 78/102 [01:35<00:29, 1.22s/it]

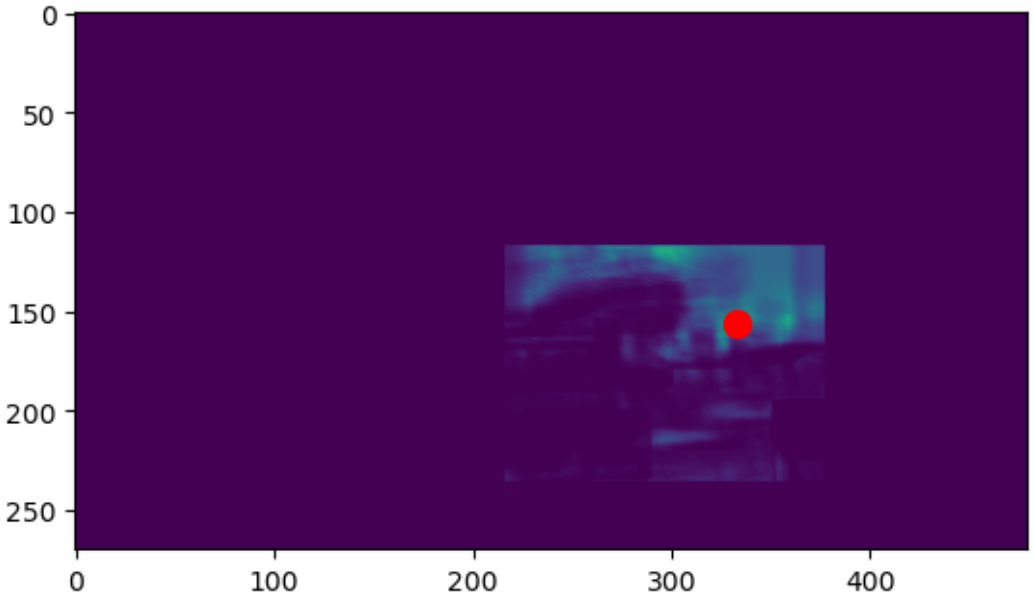

0
50
100
150
200
250
0
100
200
300
400

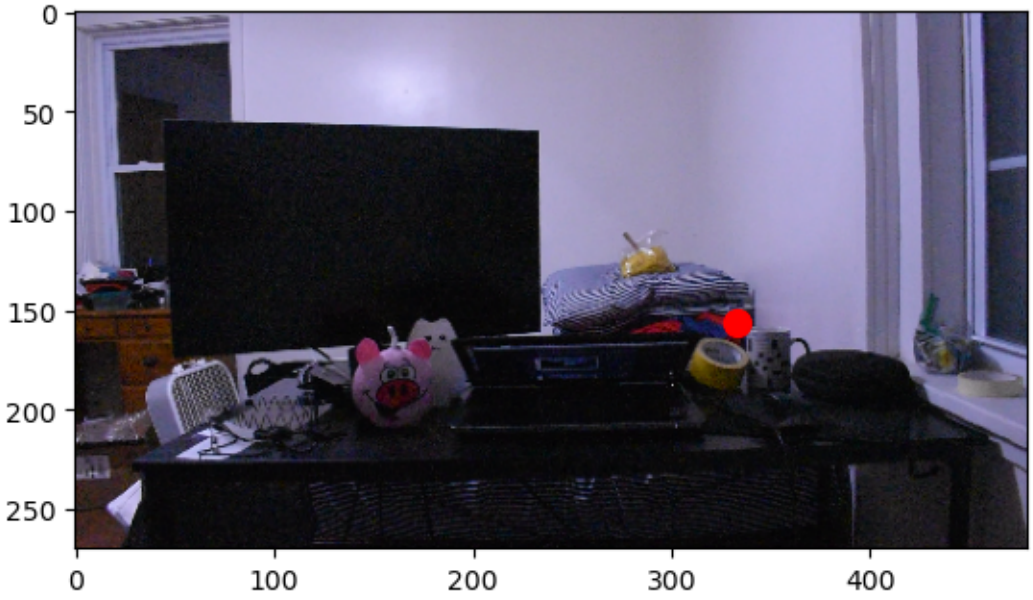

0
50
100
150
200
250
0
100
200
300
400



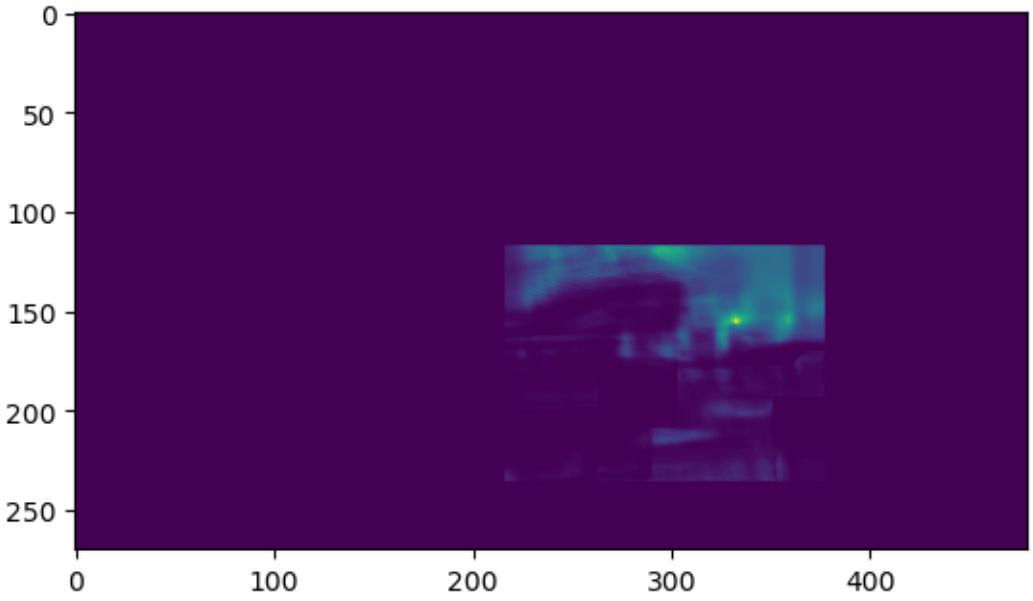
77%| | 79/102 [01:36<00:27, 1.21s/it]

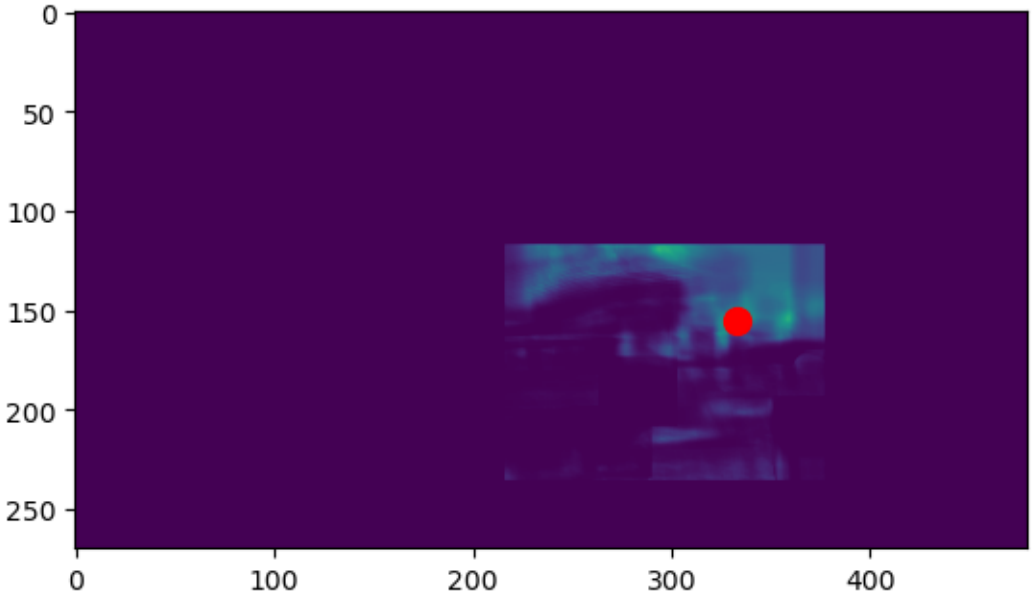


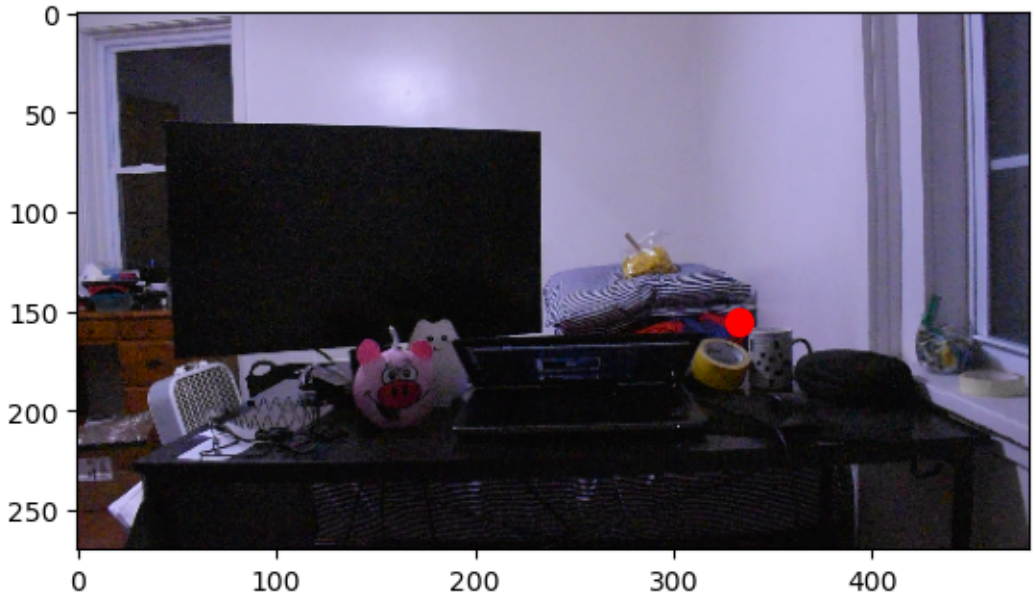




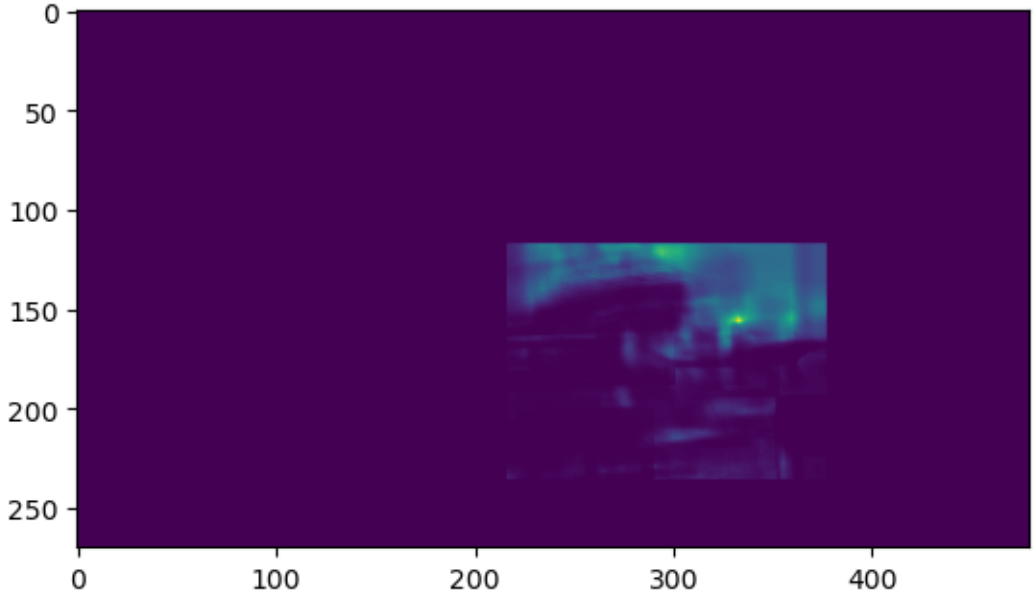
78%| | 80/102 [01:37<00:26, 1.20s/it]

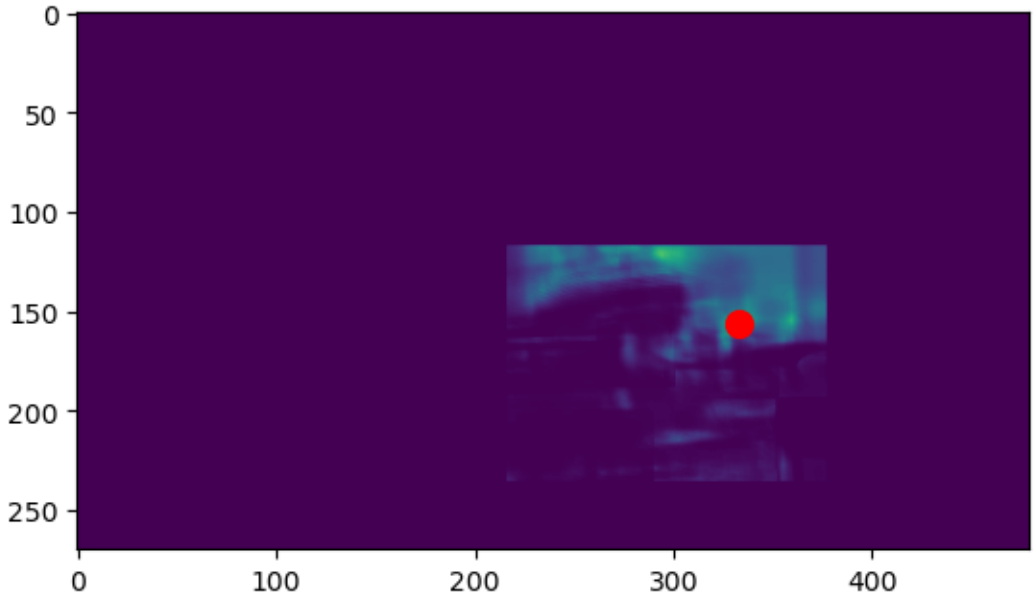

0
50
100
150
200
250
0
100
200
300
400

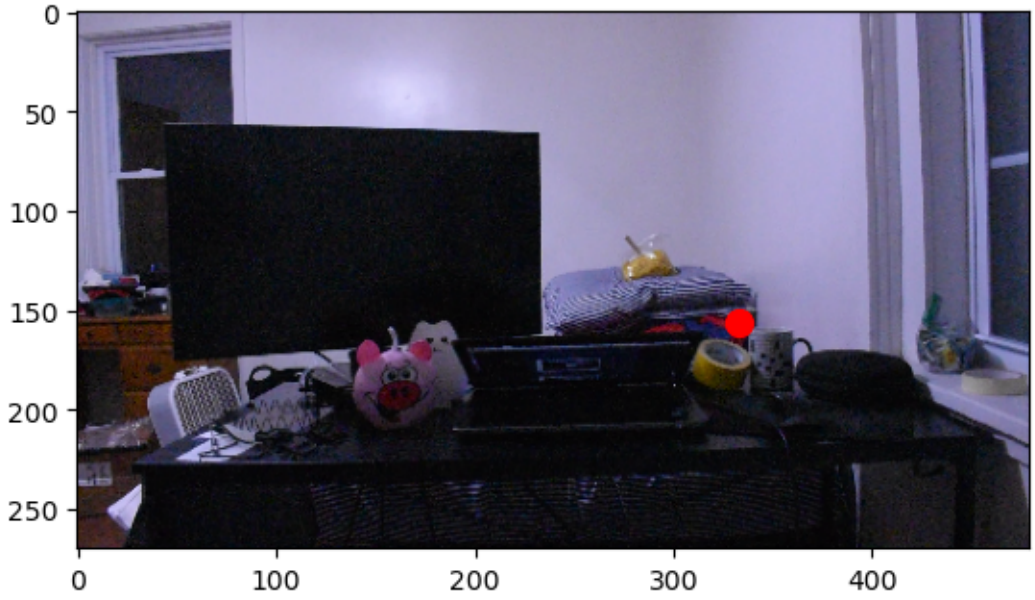

0
50
100
150
200
250
0
100
200
300
400



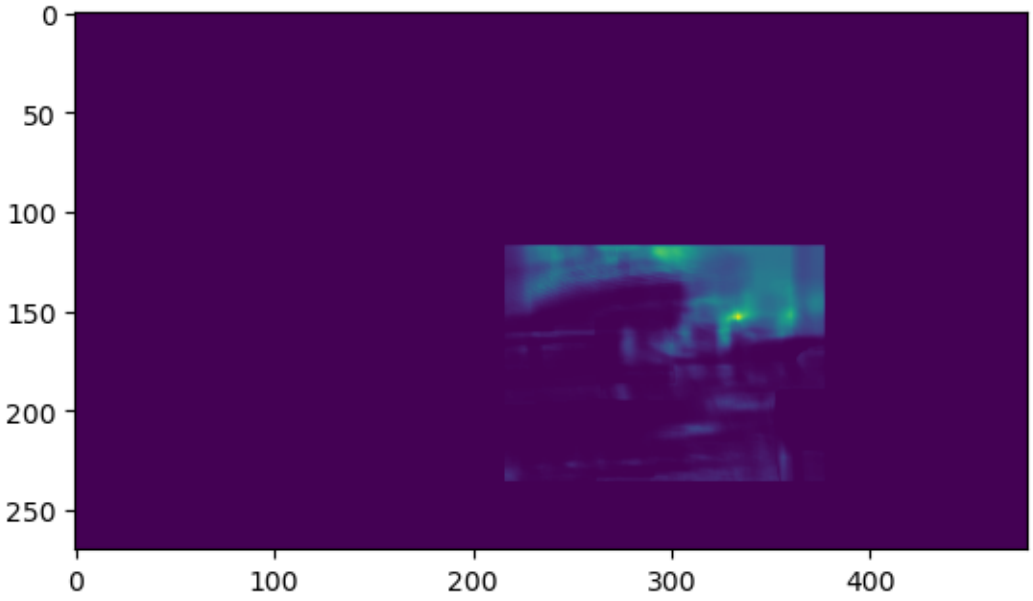
79%| | 81/102 [01:39<00:25, 1.24s/it]

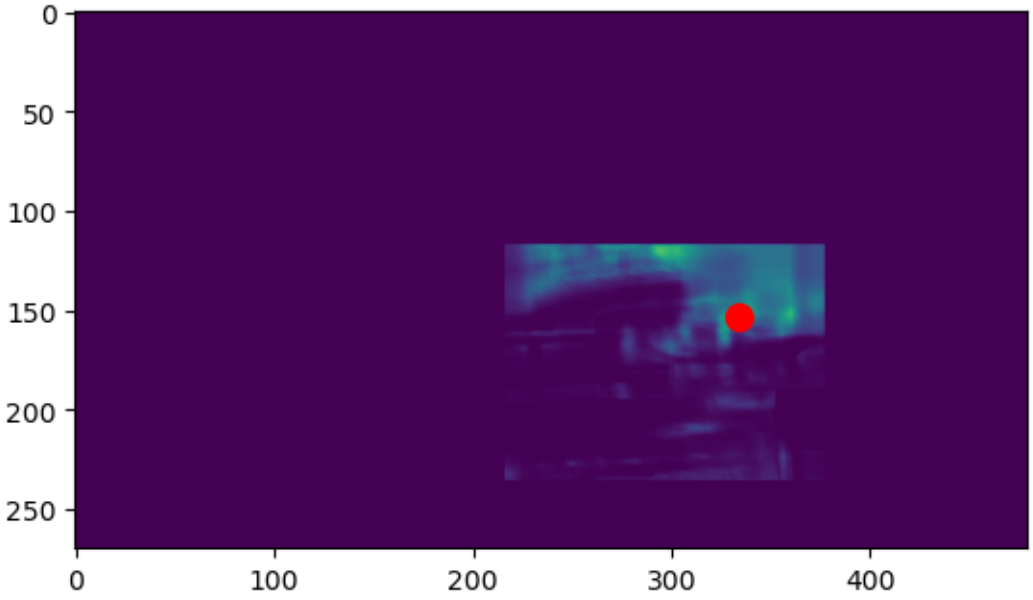


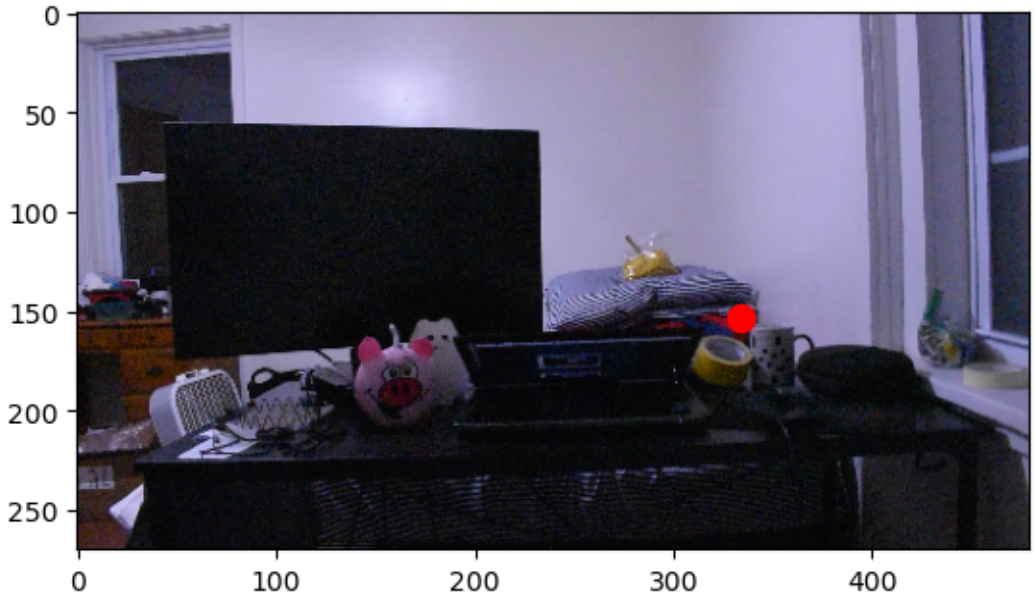




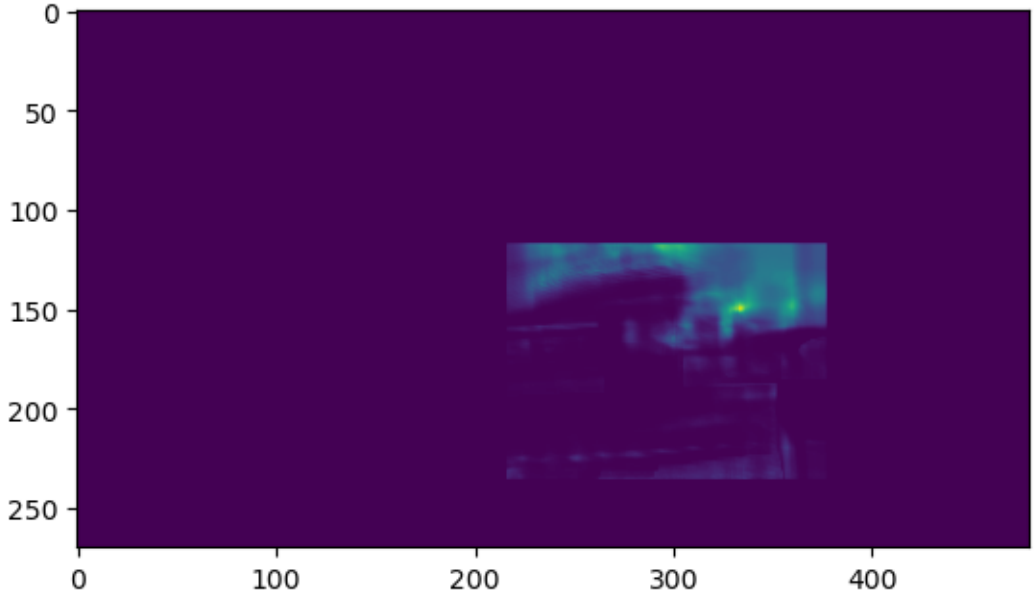
80%| | 82/102 [01:40<00:24, 1.22s/it]

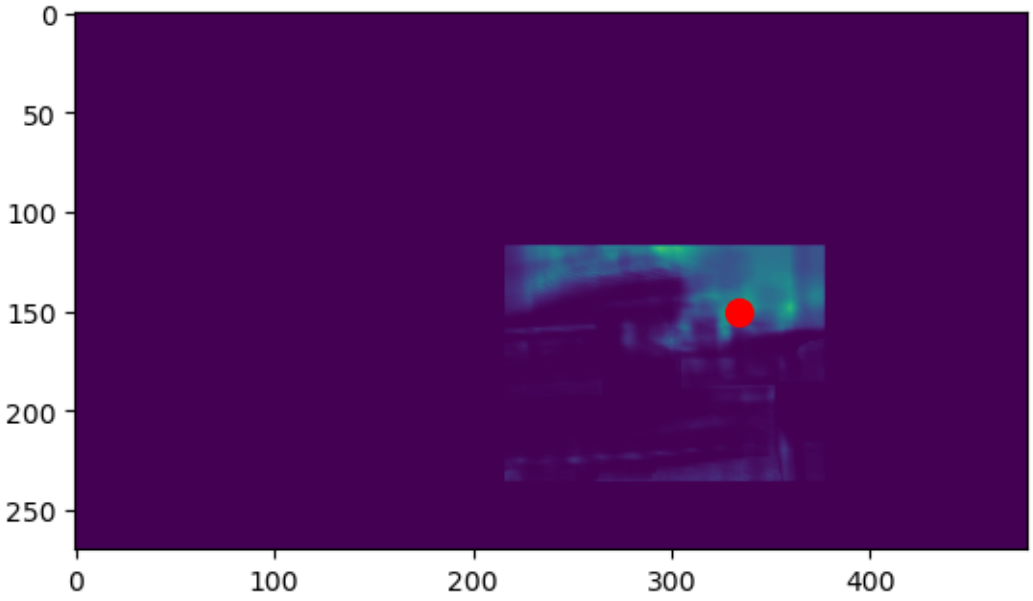

0
50
100
150
200
250
0
100
200
300
400

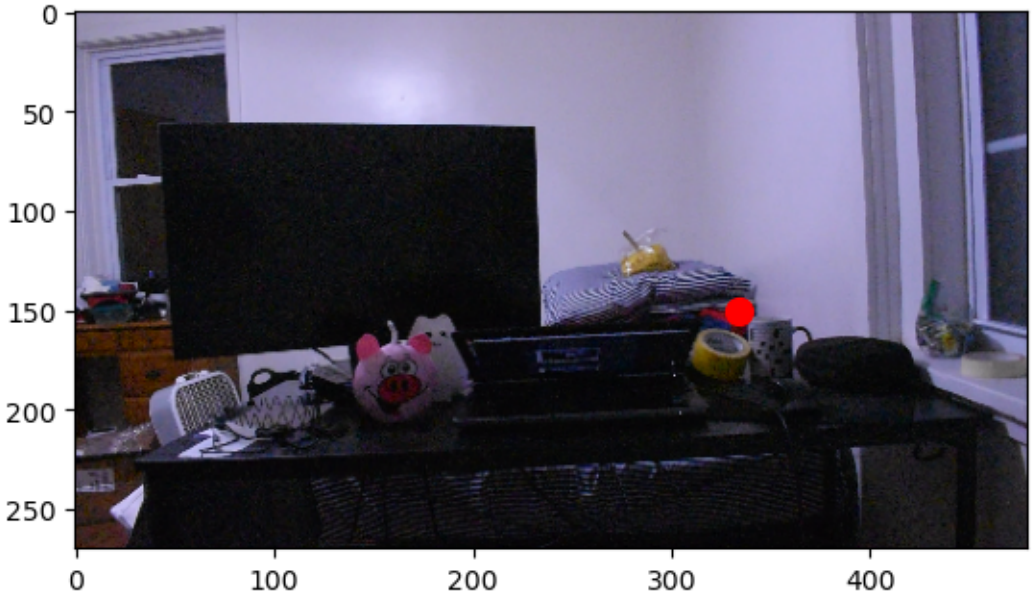

0
50
100
150
200
250
0
100
200
300
400



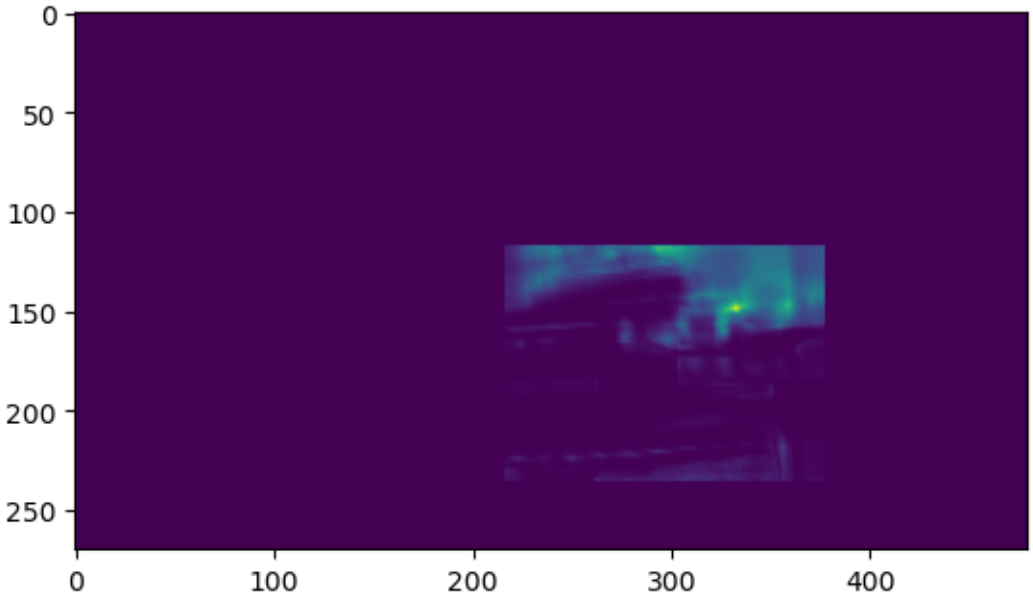
81%| | 83/102 [01:41<00:23, 1.24s/it]

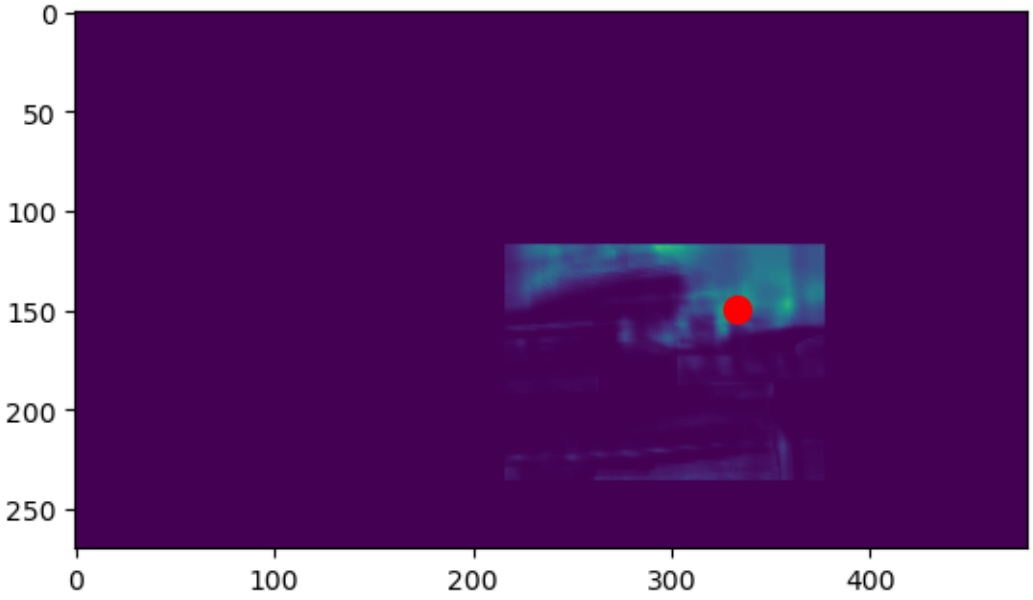


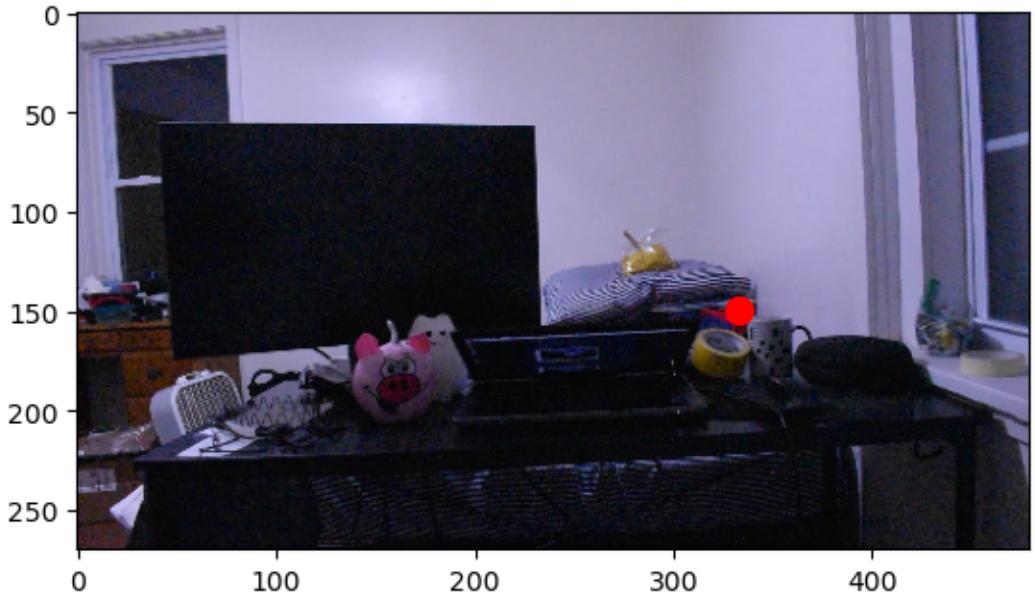




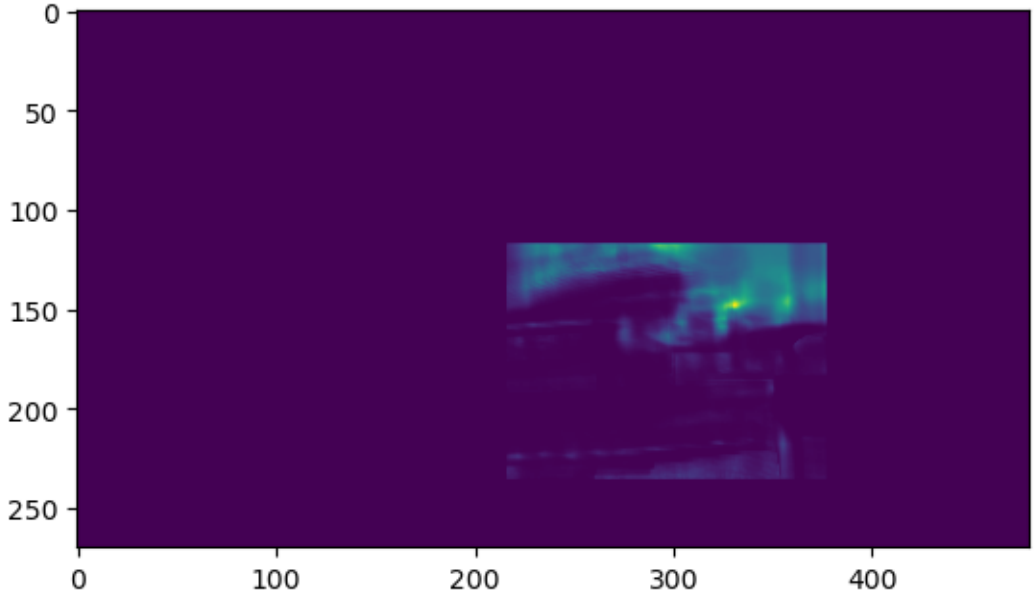
82%| | 84/102 [01:42<00:21, 1.22s/it]

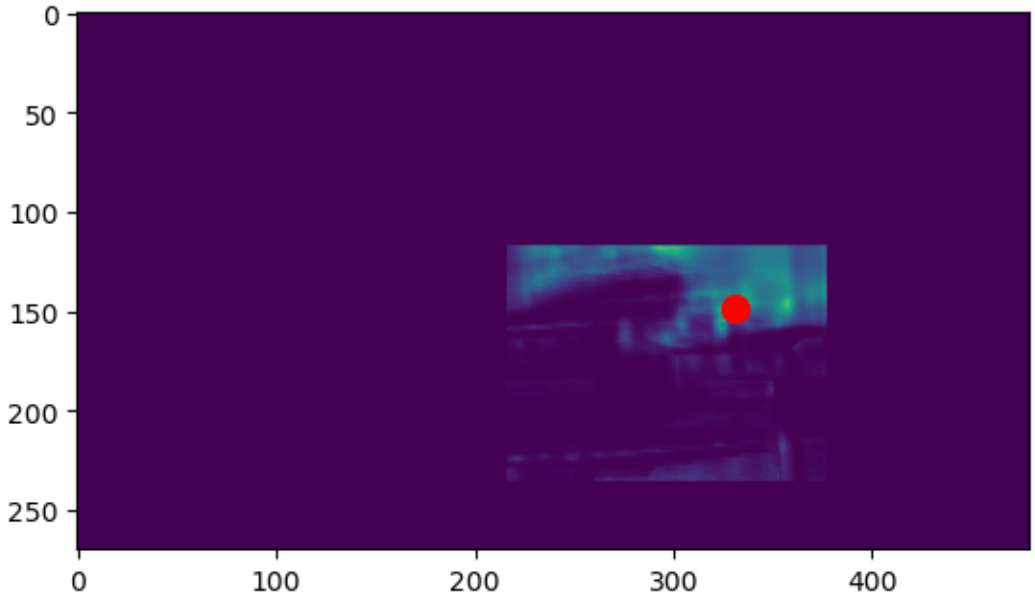

0
50
100
150
200
250
0
100
200
300
400

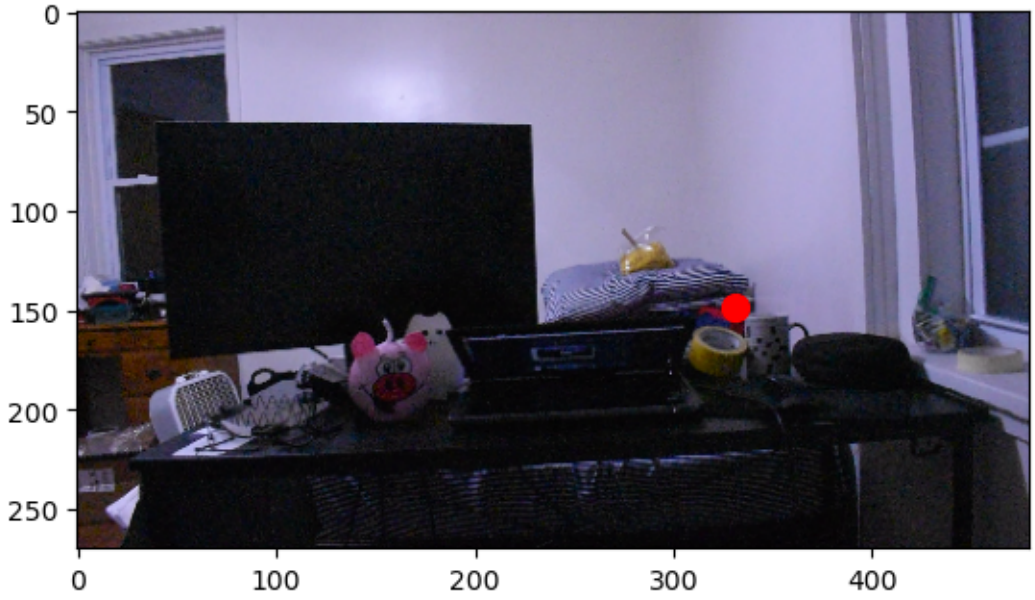

0
50
100
150
200
250
0
100
200
300
400



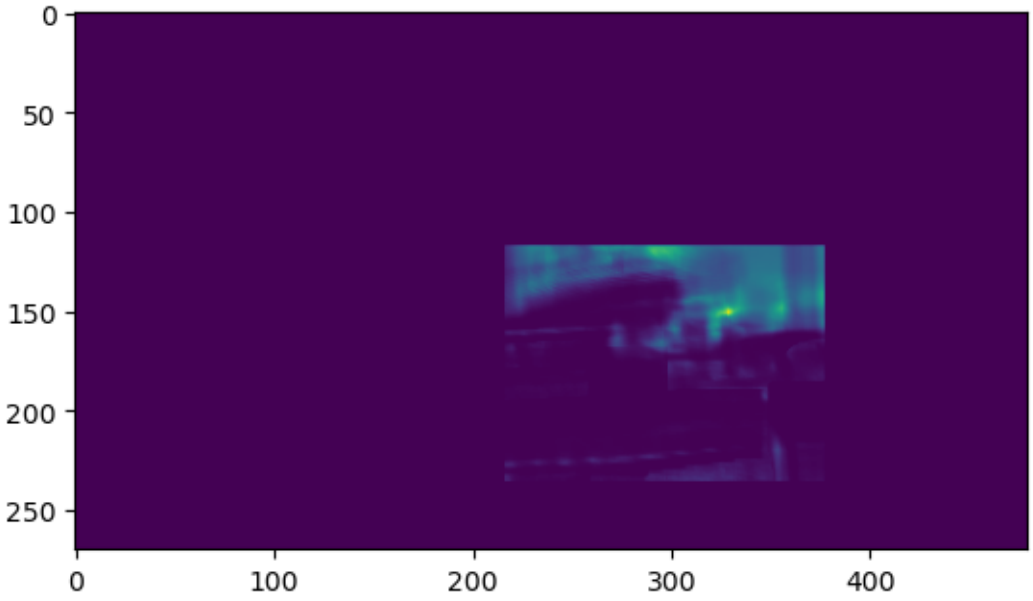
83%| | 85/102 [01:43<00:20, 1.21s/it]

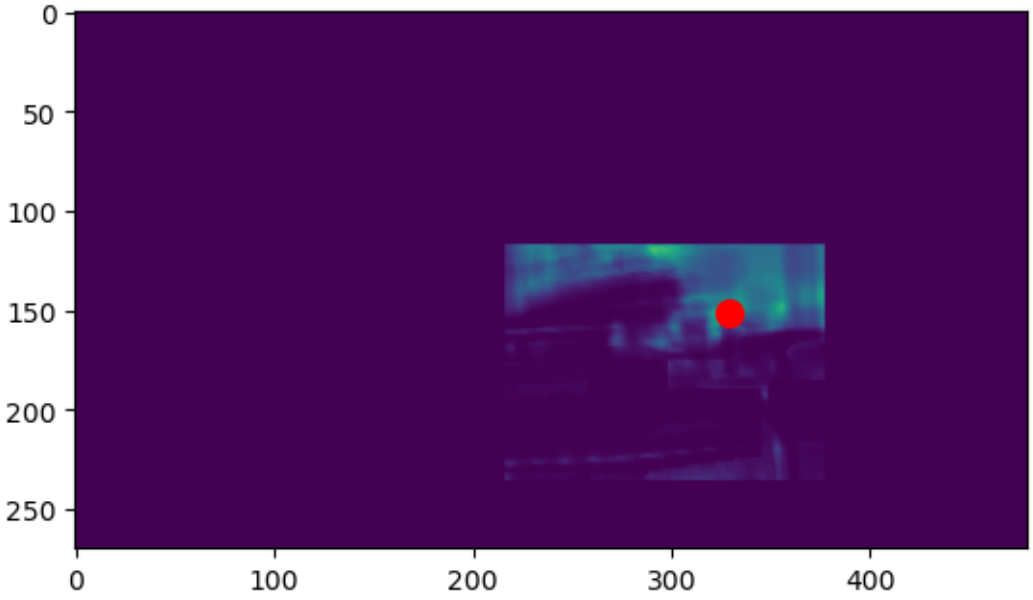


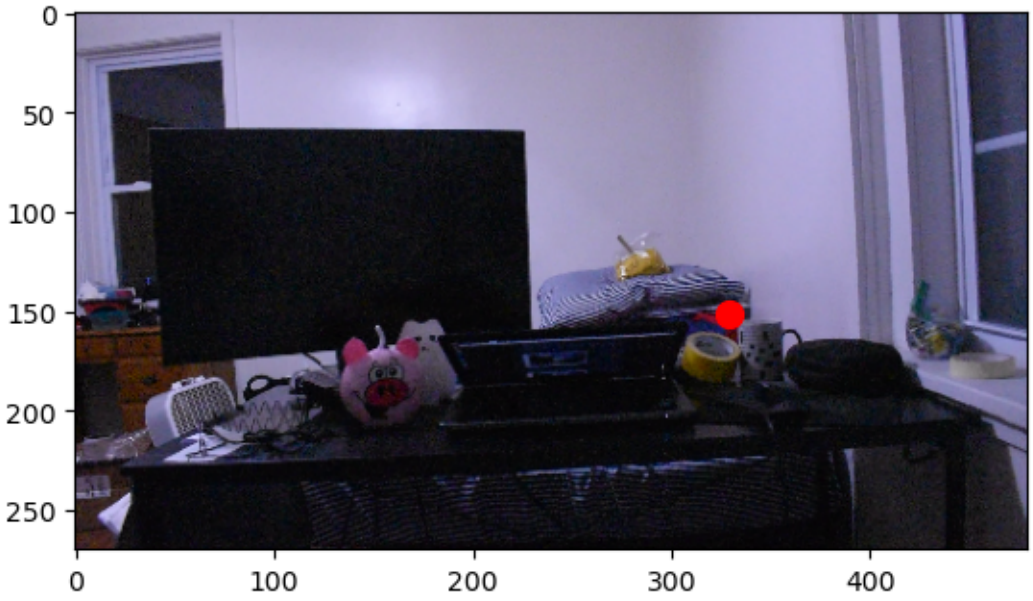




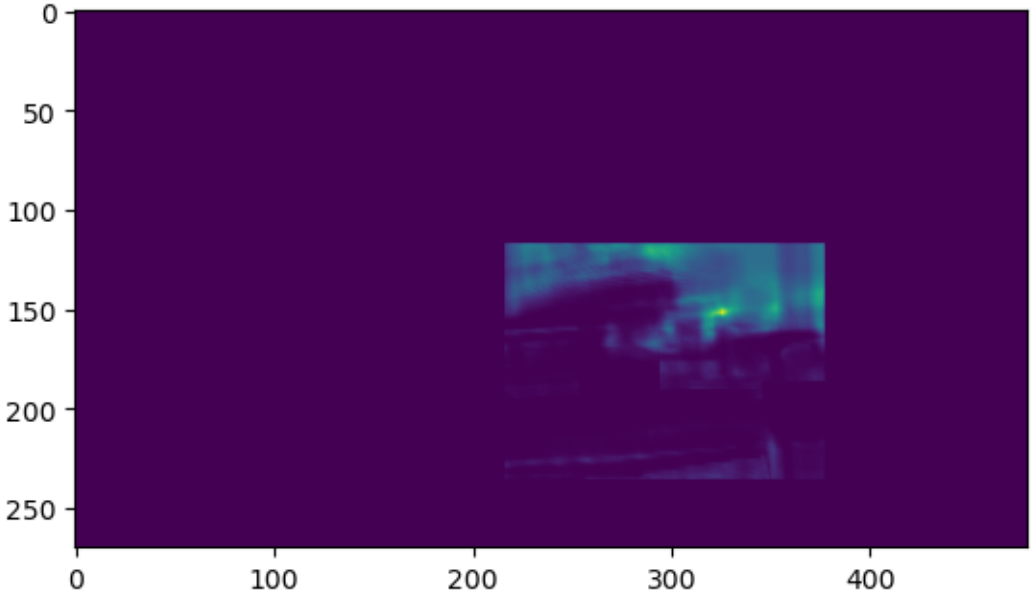
84%| | 86/102 [01:45<00:19, 1.20s/it]

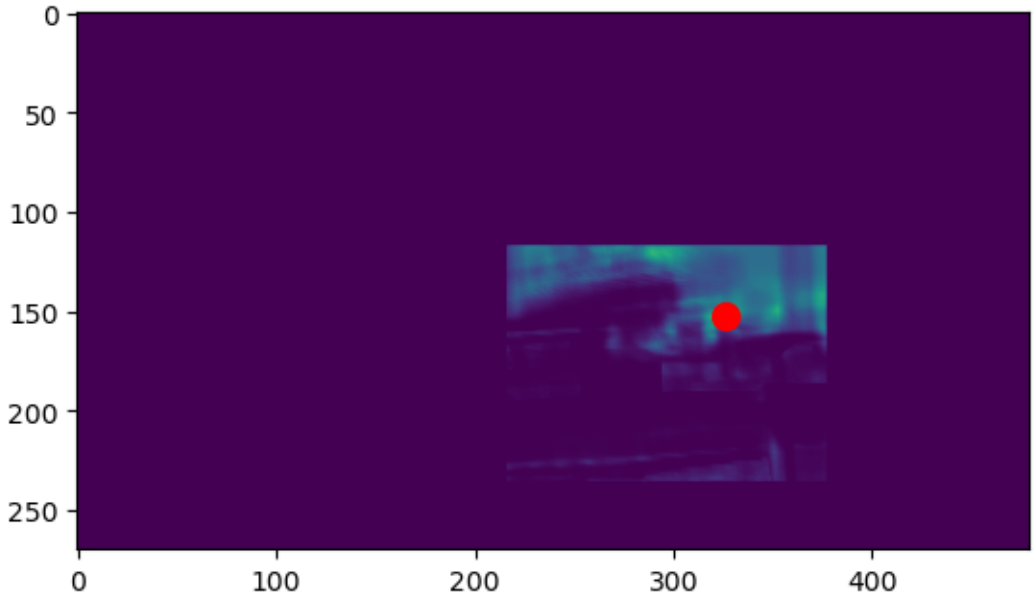

0
50
100
150
200
250
0
100
200
300
400

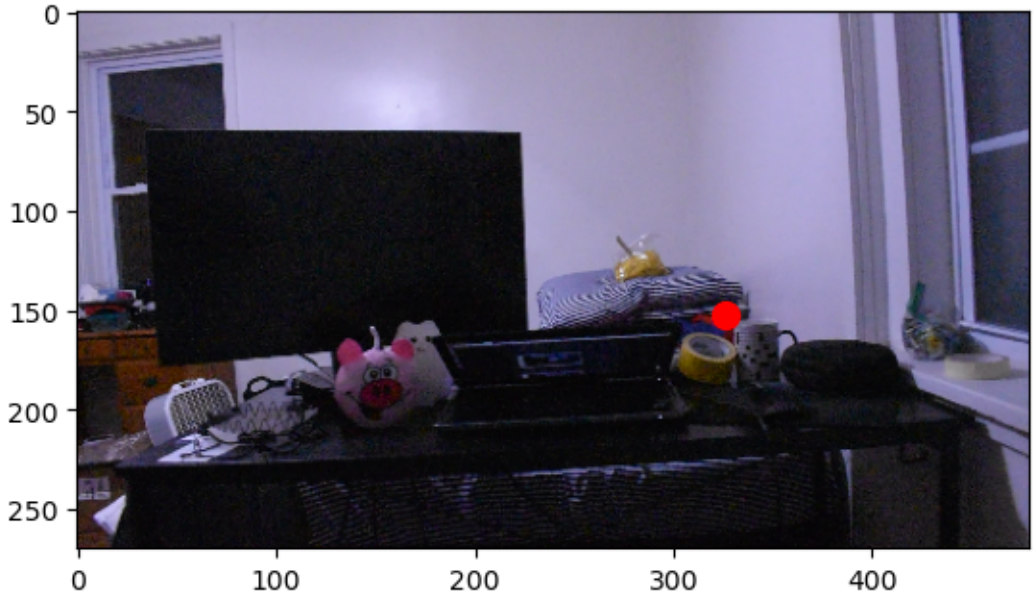

0
50
100
150
200
250
0
100
200
300
400



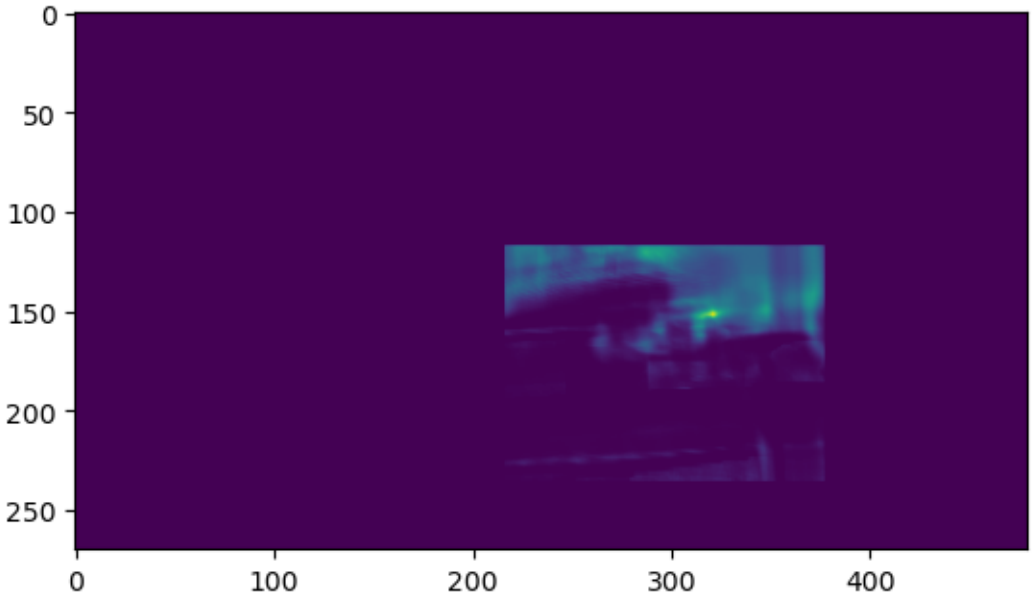
85%| | 87/102 [01:46<00:17, 1.19s/it]

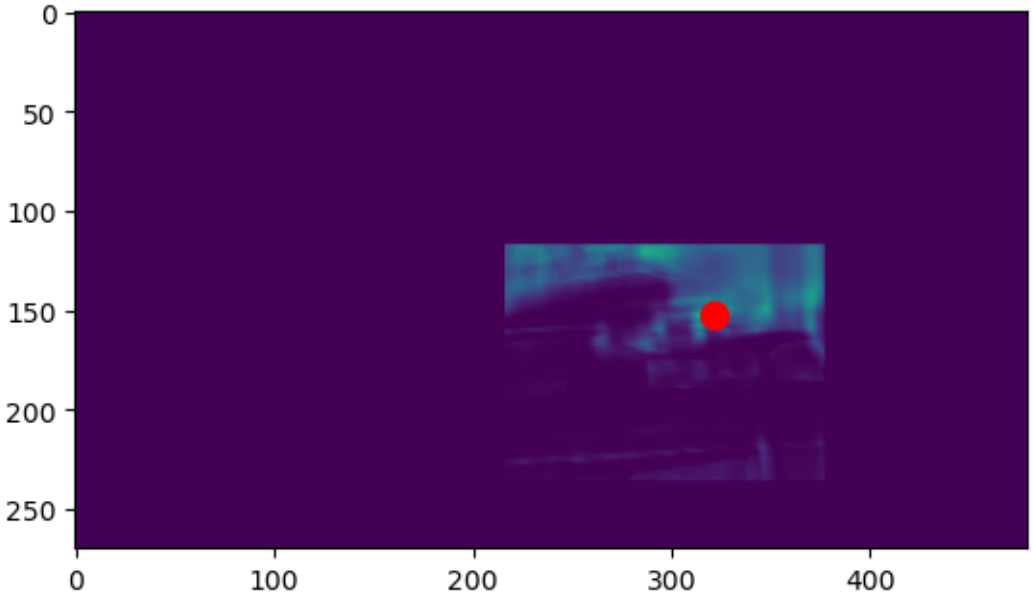


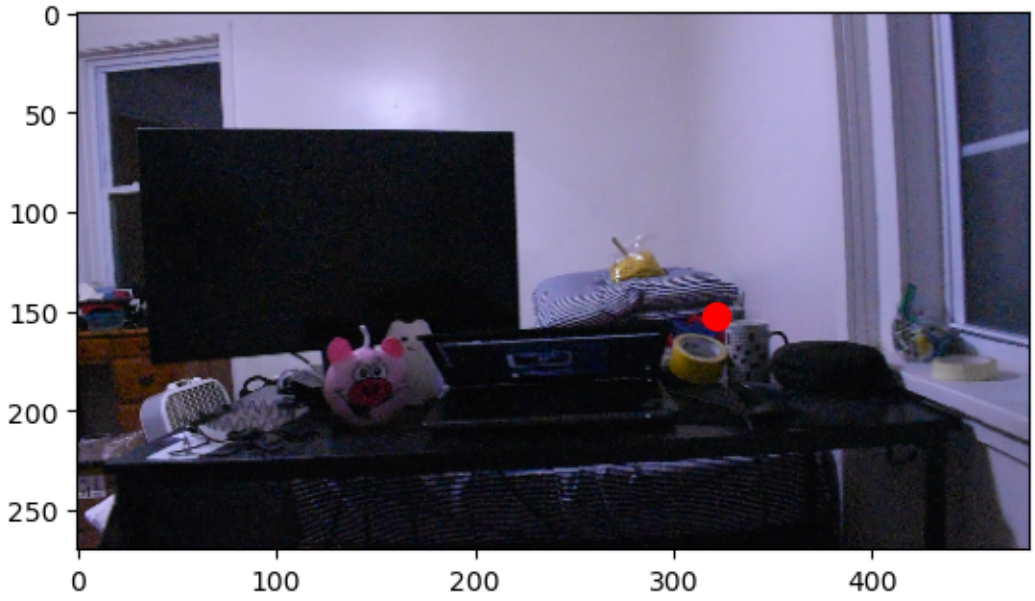




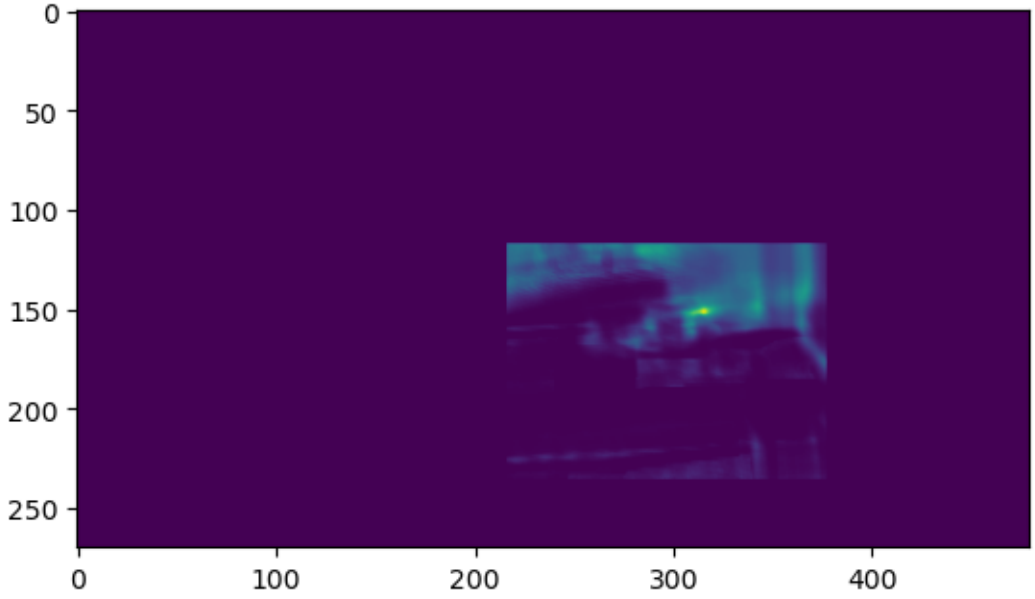
86%| | 88/102 [01:47<00:16, 1.18s/it]

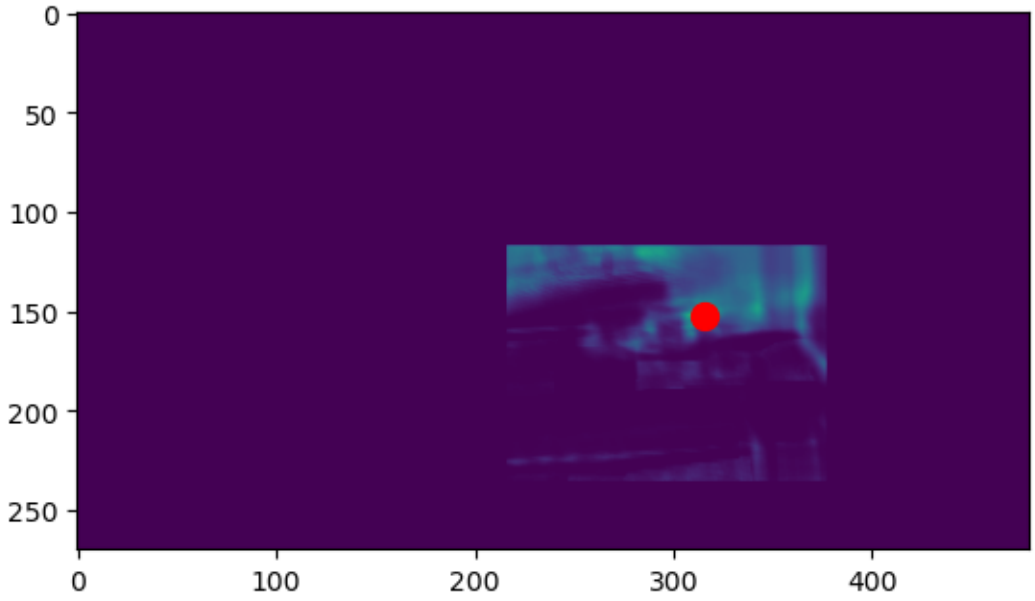

0
50
100
150
200
250
0
100
200
300
400

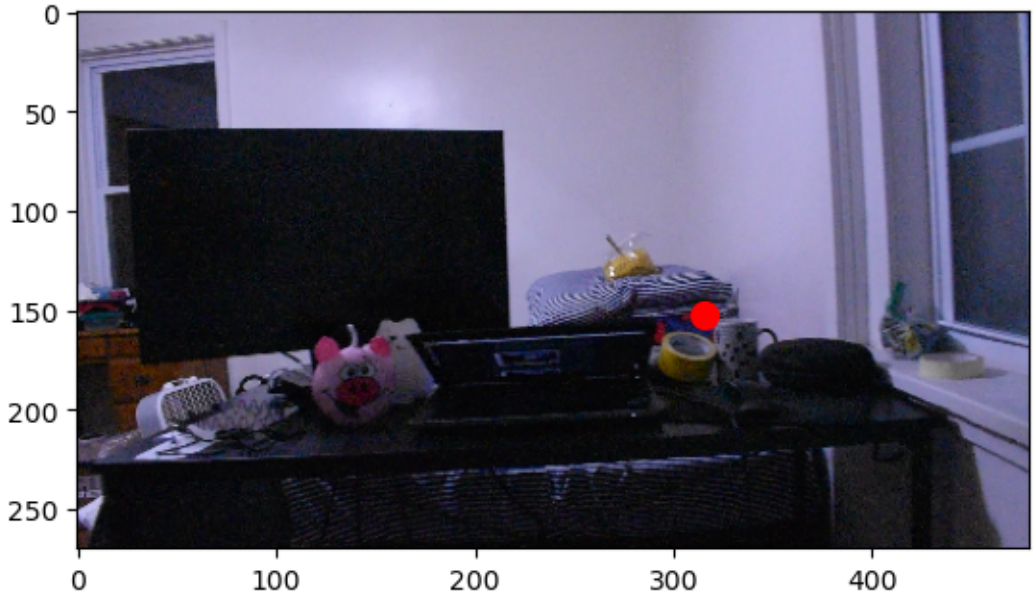

0
50
100
150
200
250
0
100
200
300
400



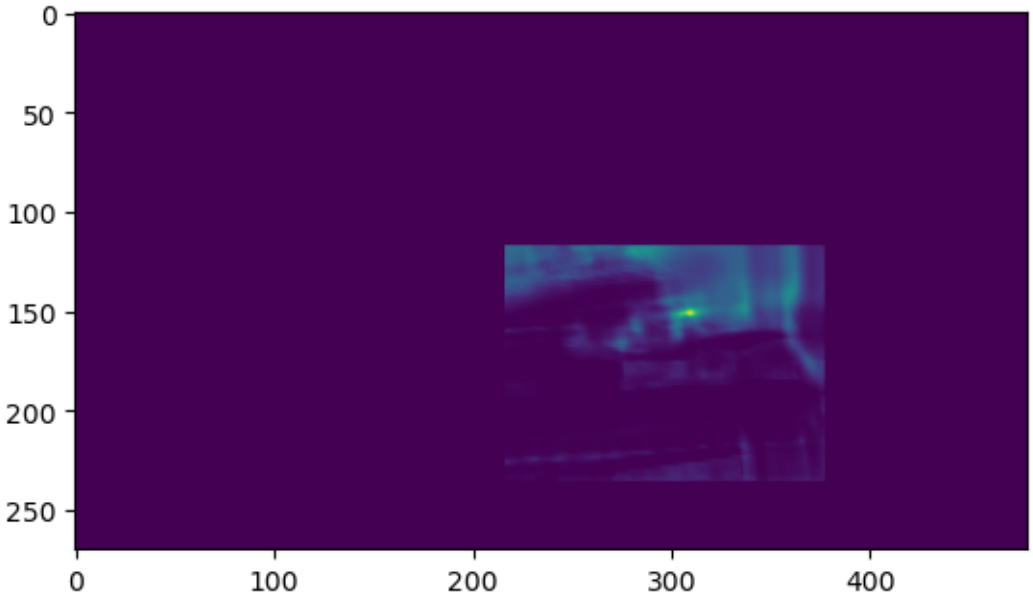
87%| | 89/102 [01:48<00:16, 1.24s/it]

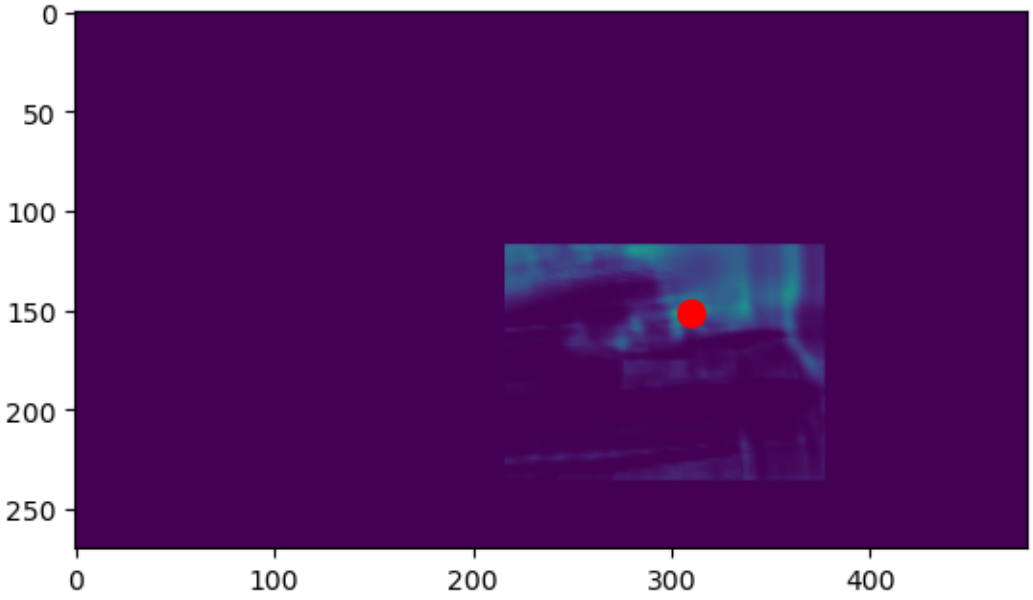


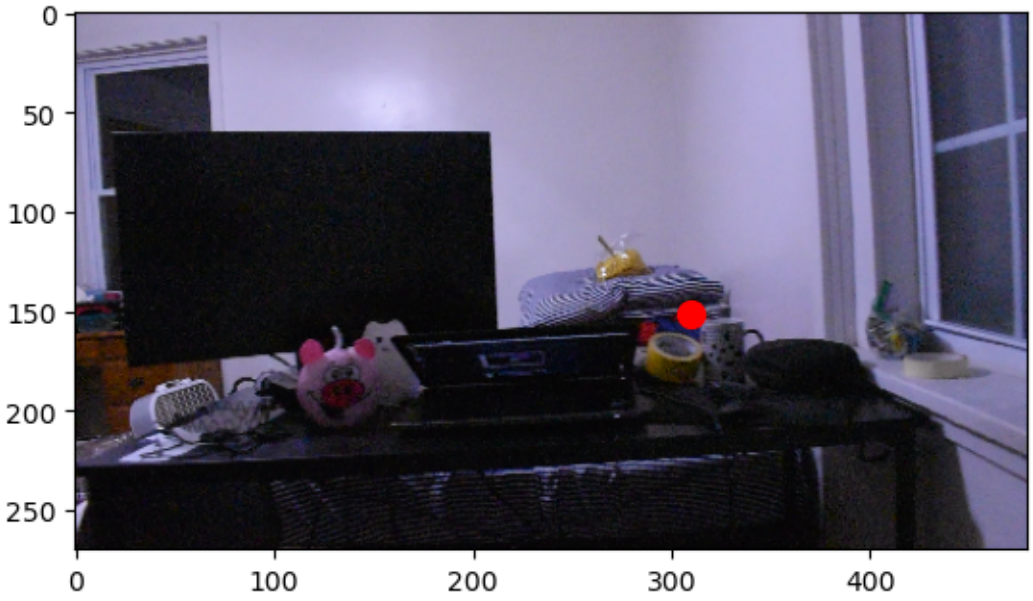




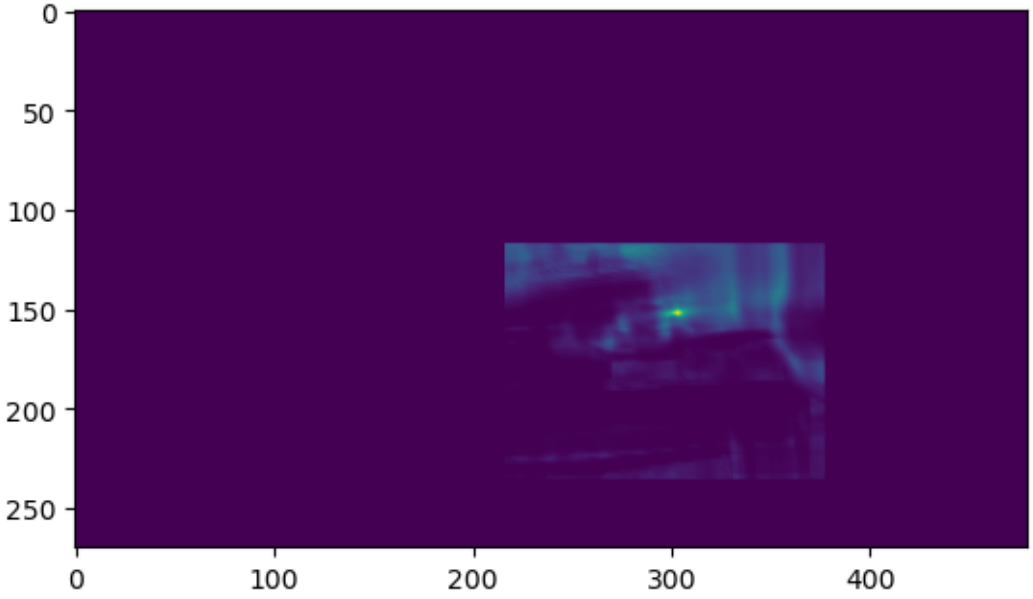
88%| | 90/102 [01:50<00:15, 1.25s/it]

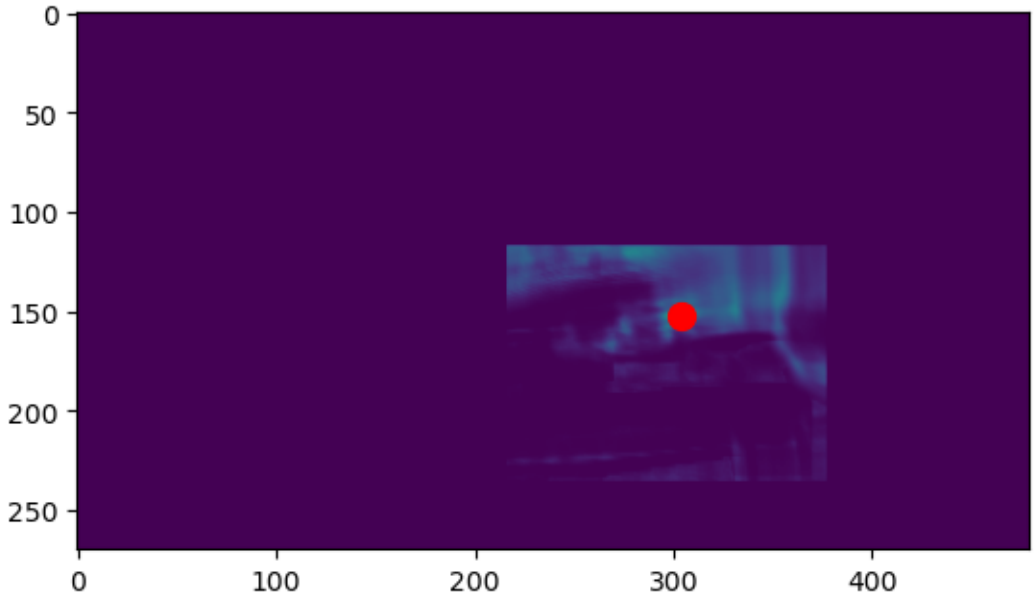

0
50
100
150
200
250
0
100
200
300
400

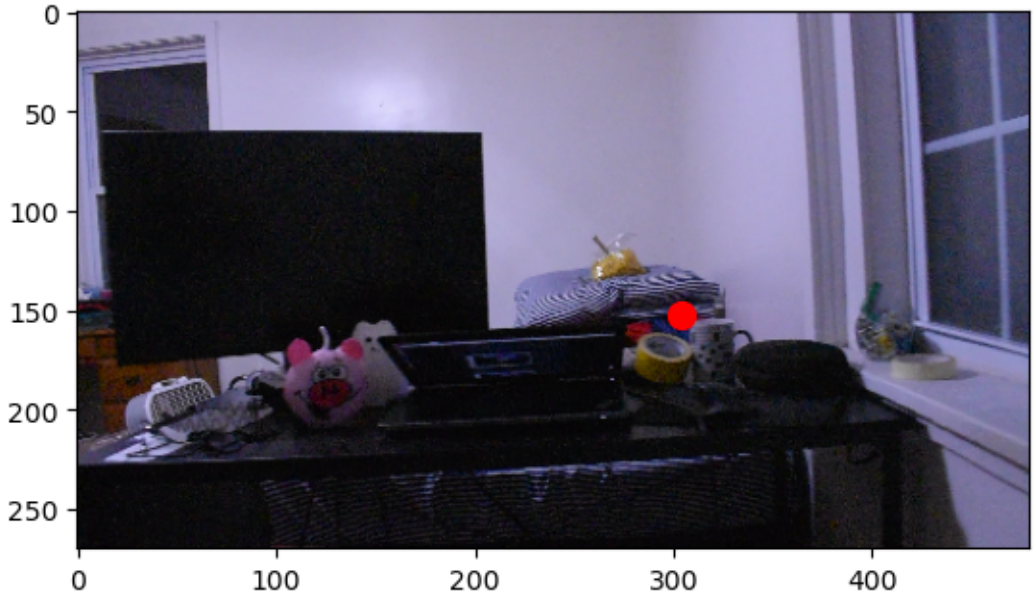

0
50
100
150
200
250
0
100
200
300
400



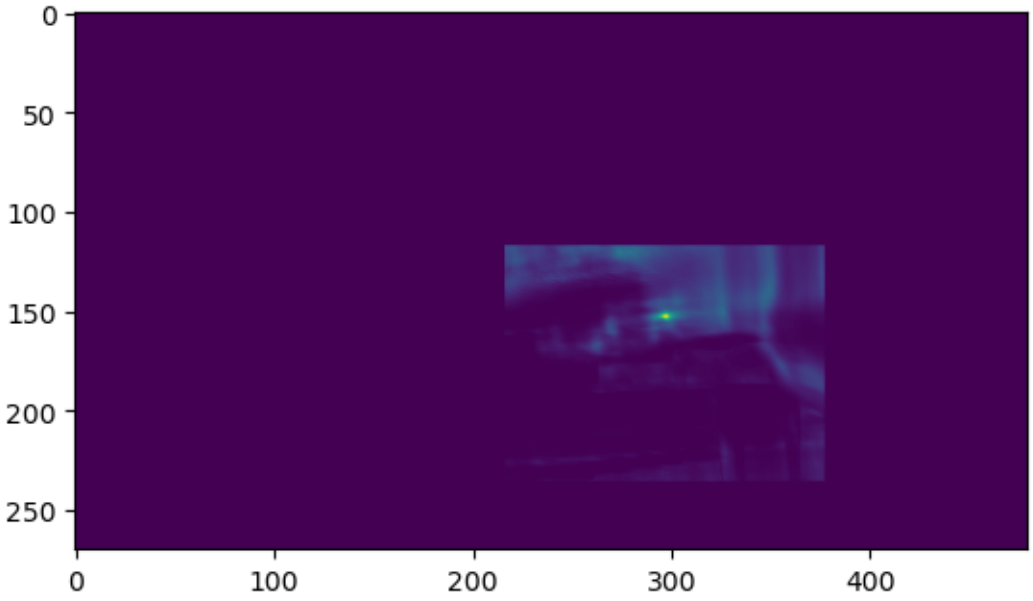
89%| | 91/102 [01:51<00:13, 1.25s/it]

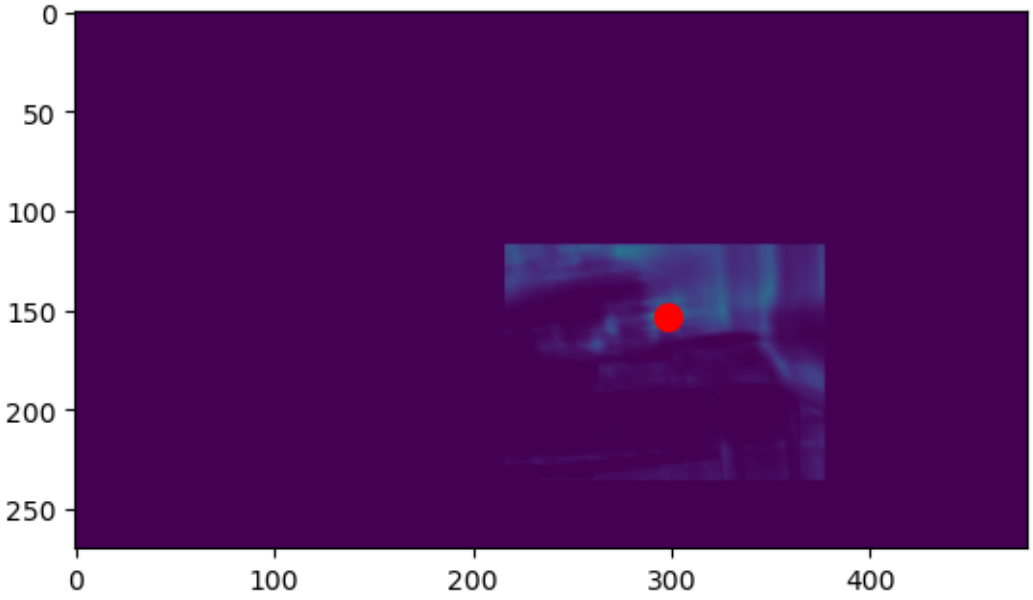


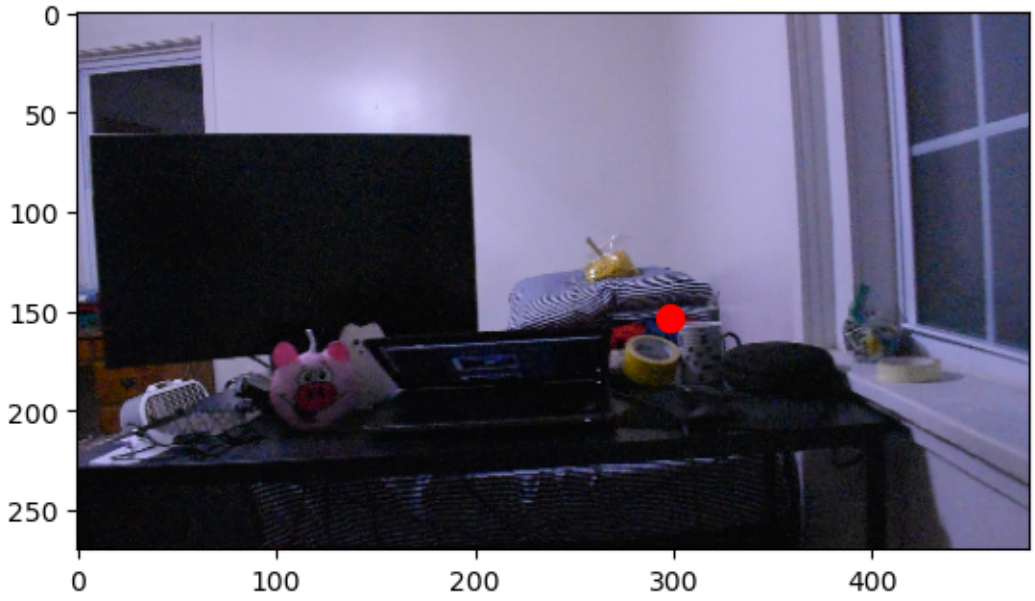




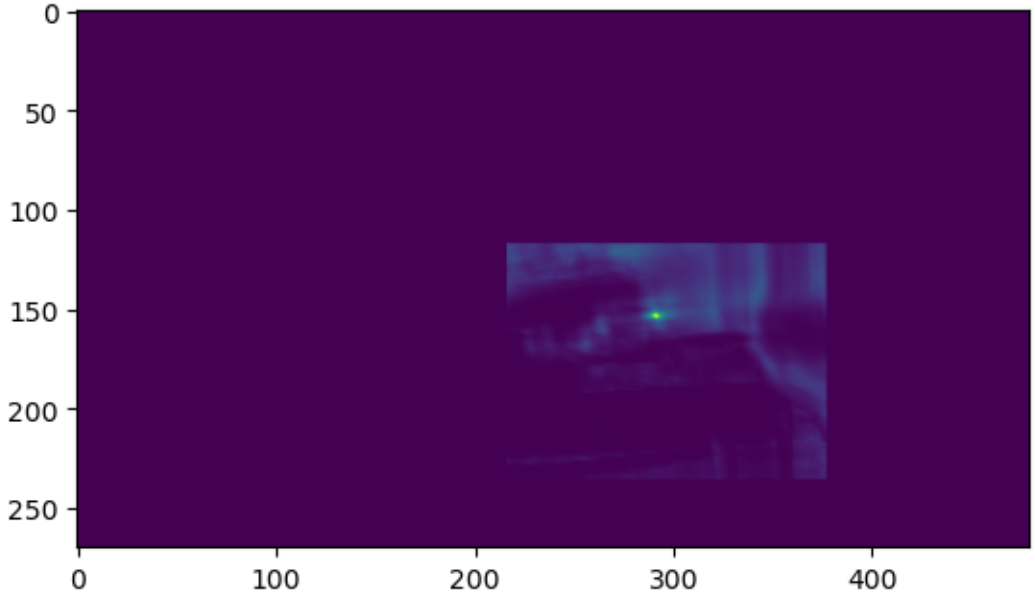
90%| | 92/102 [01:52<00:12, 1.29s/it]

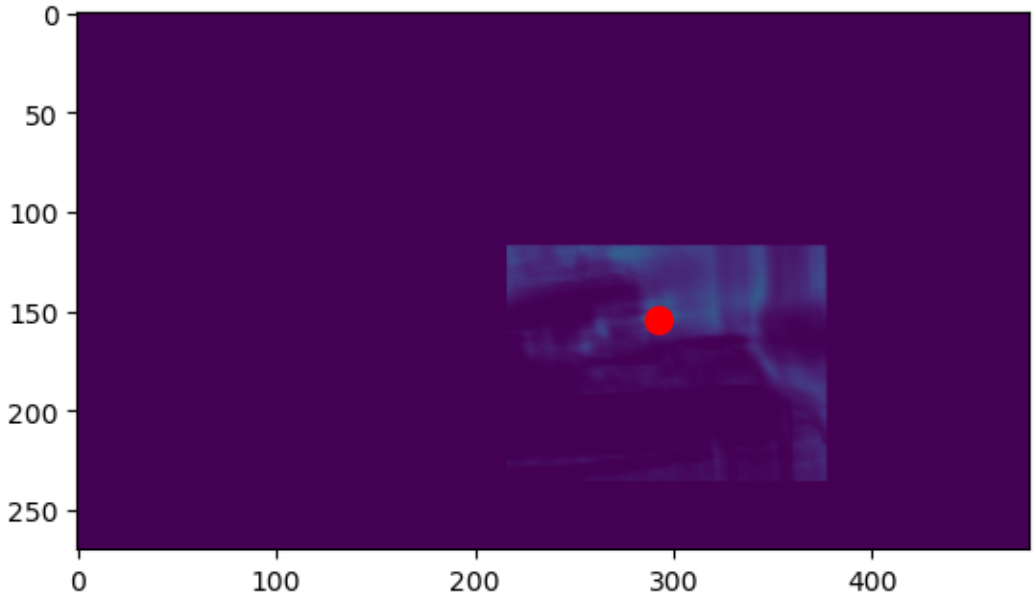

0
50
100
150
200
250
0
100
200
300
400

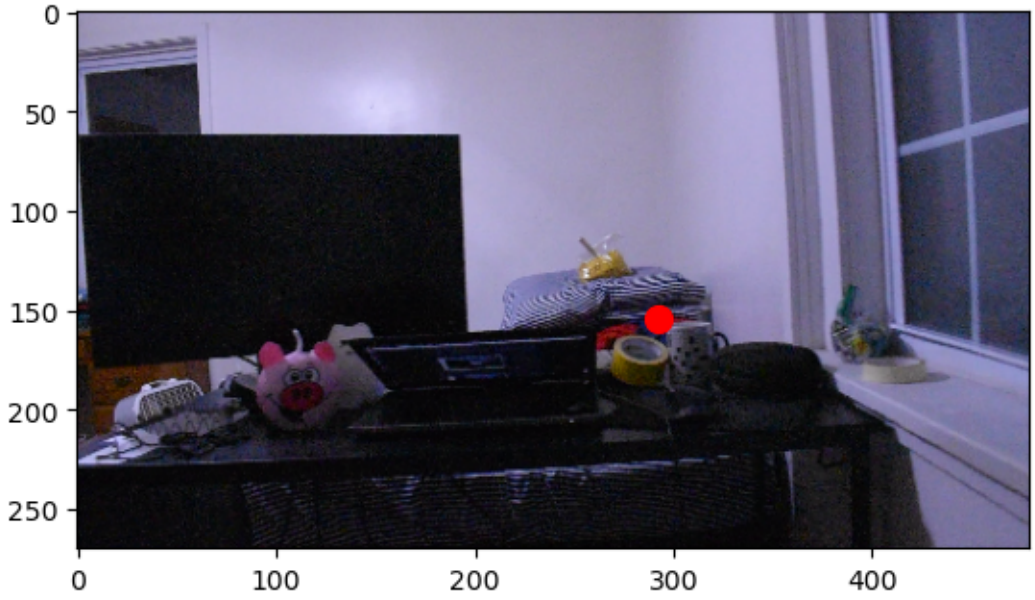

0
50
100
150
200
250
0
100
200
300
400



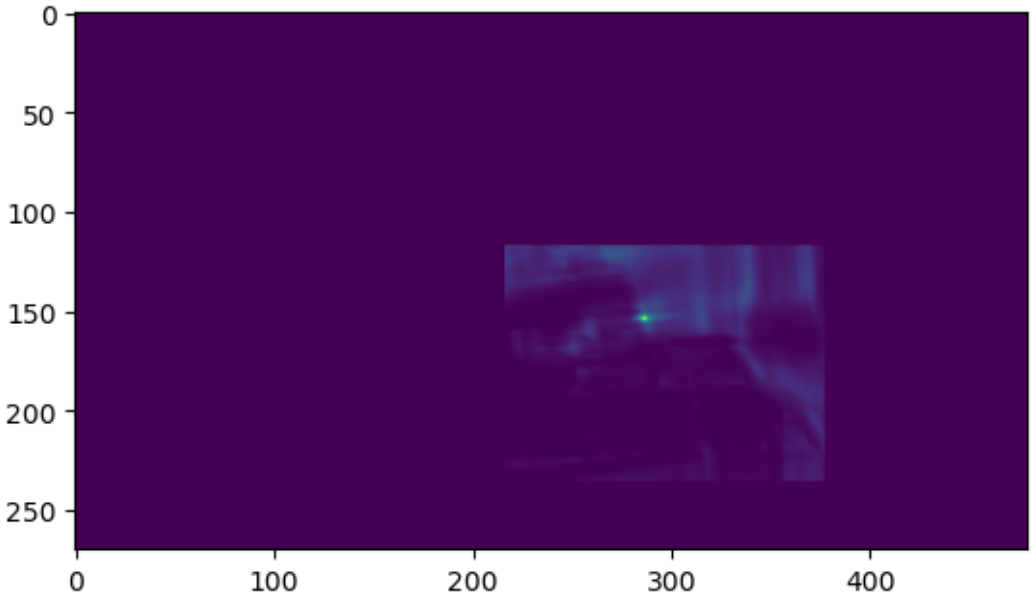
91%| | 93/102 [01:54<00:11, 1.31s/it]

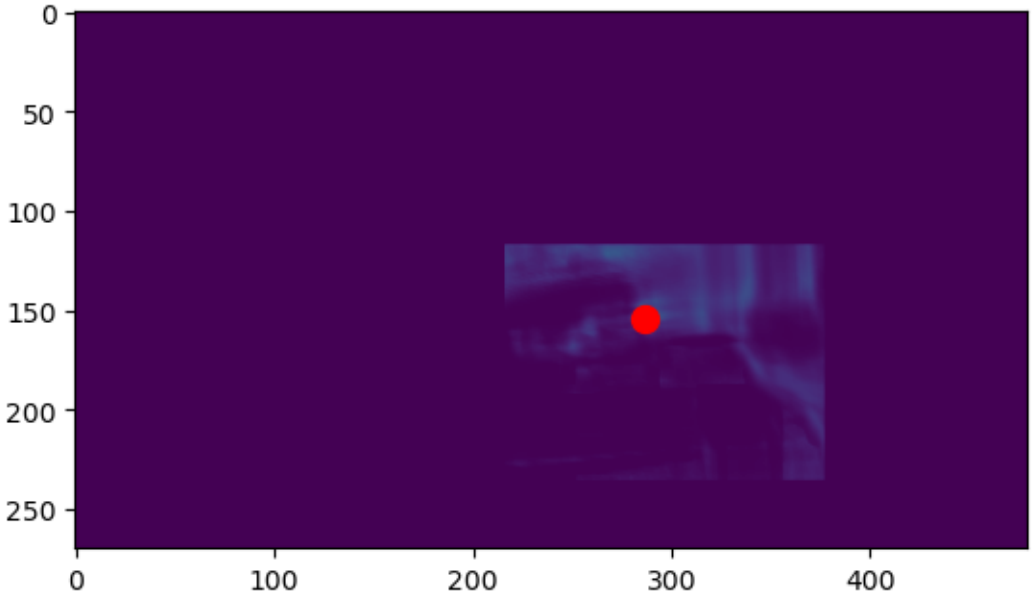


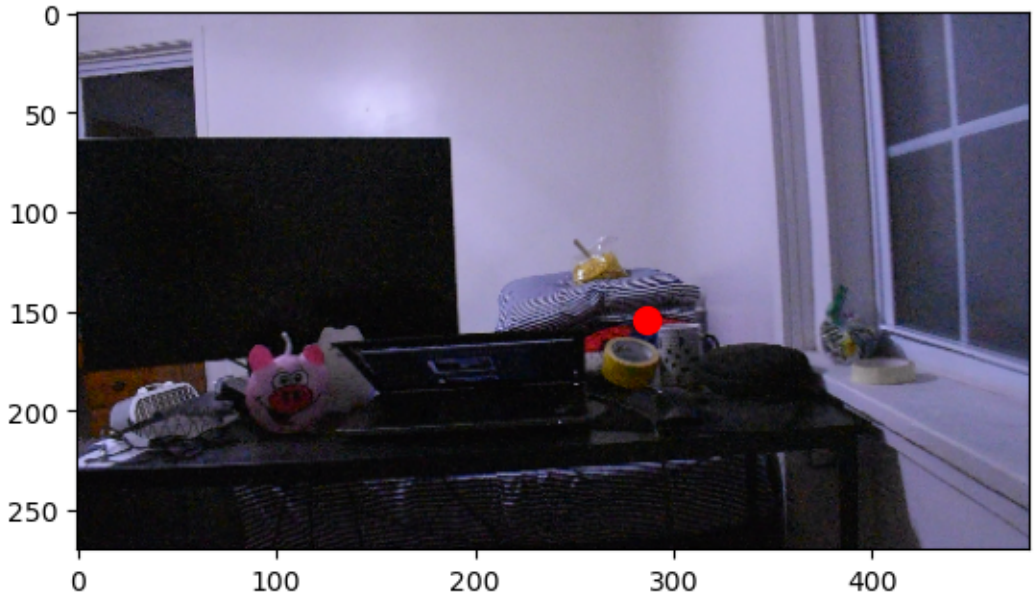




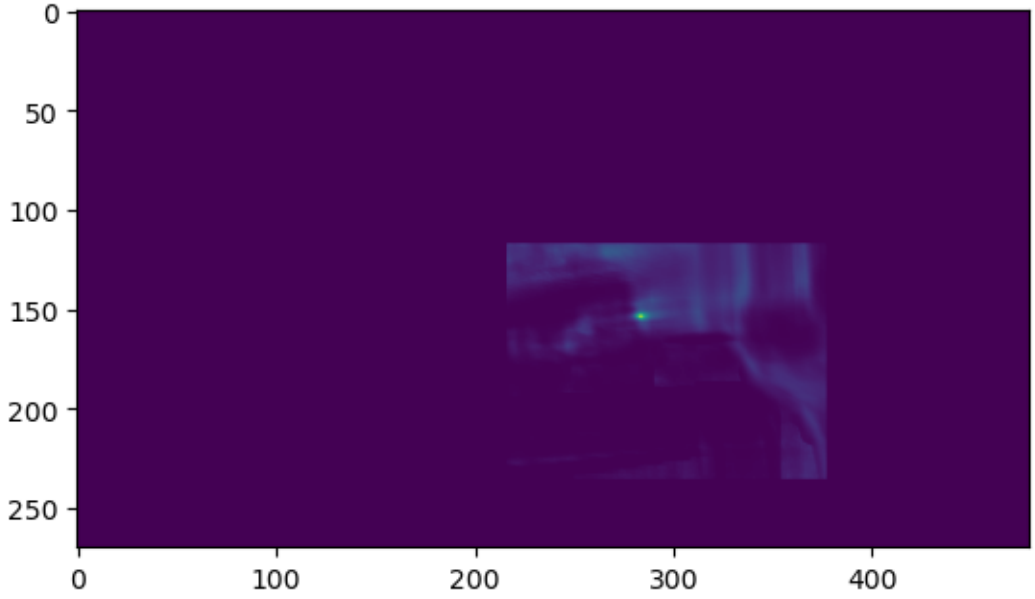
92%| | 94/102 [01:55<00:10, 1.27s/it]

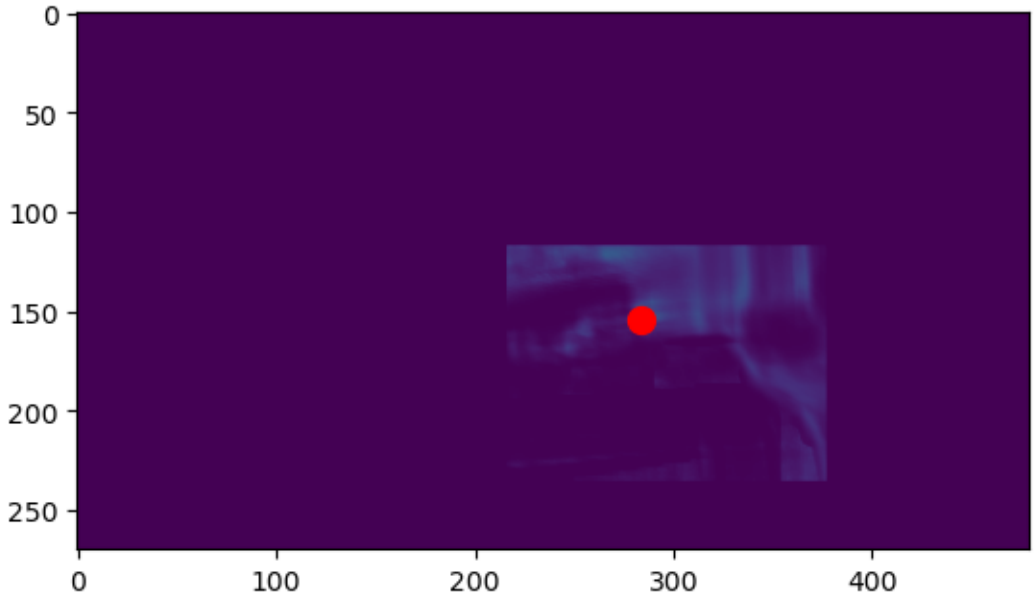

0
50
100
150
200
250
0
100
200
300
400

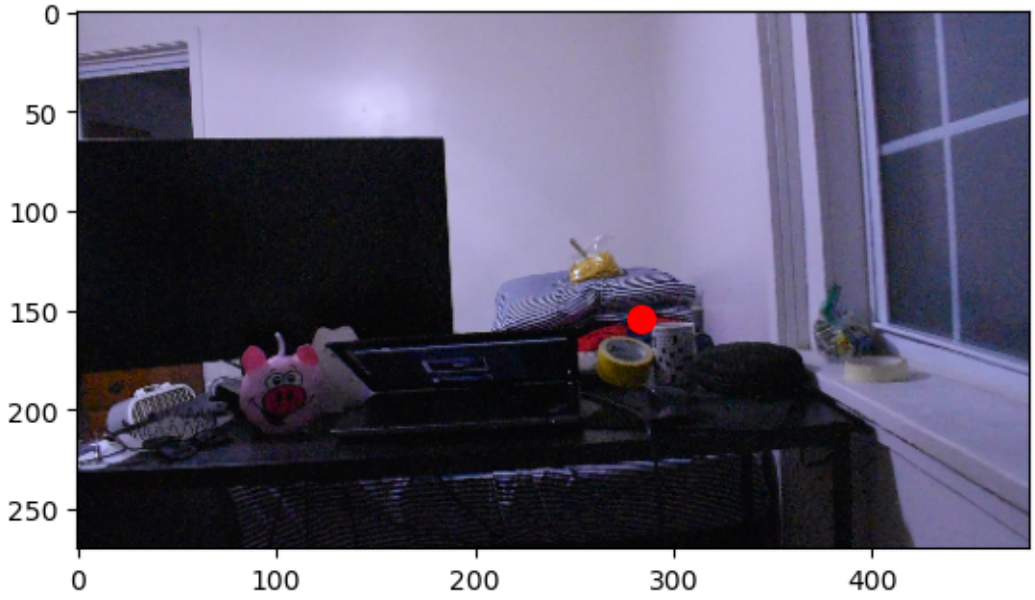

0
50
100
150
200
250
0
100
200
300
400



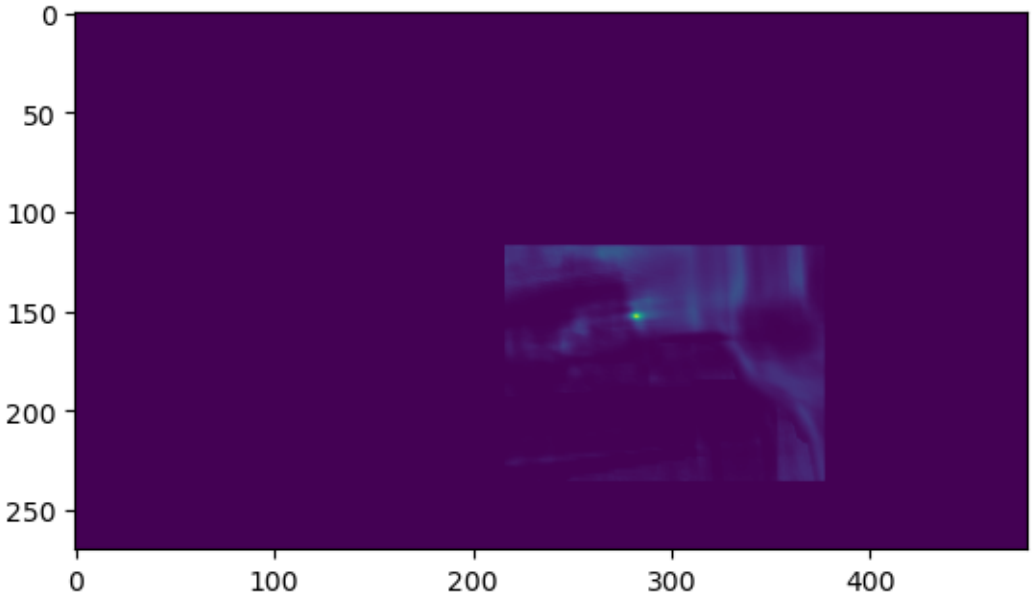
93%| | 95/102 [01:56<00:08, 1.25s/it]

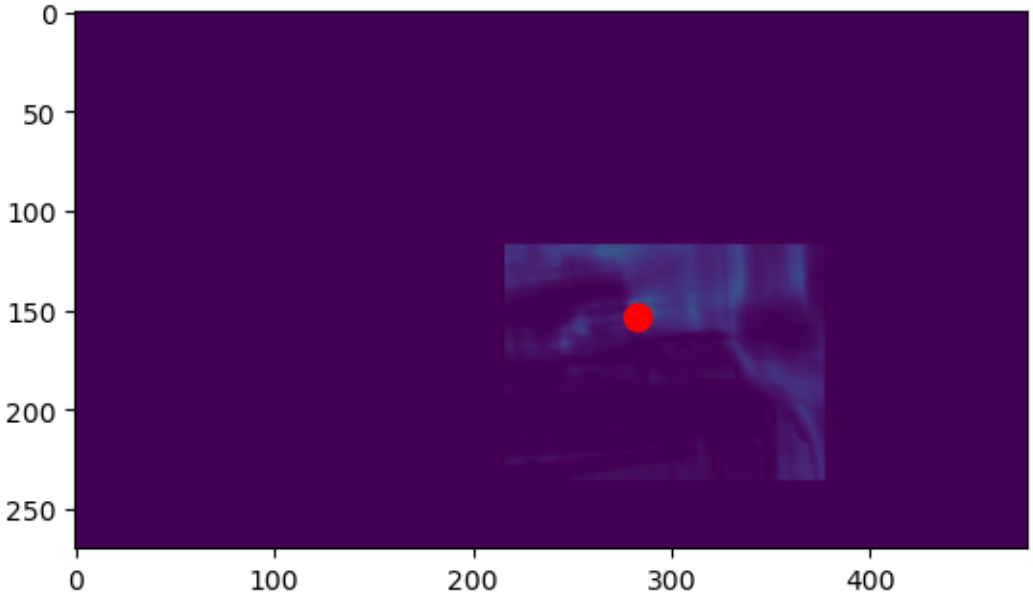


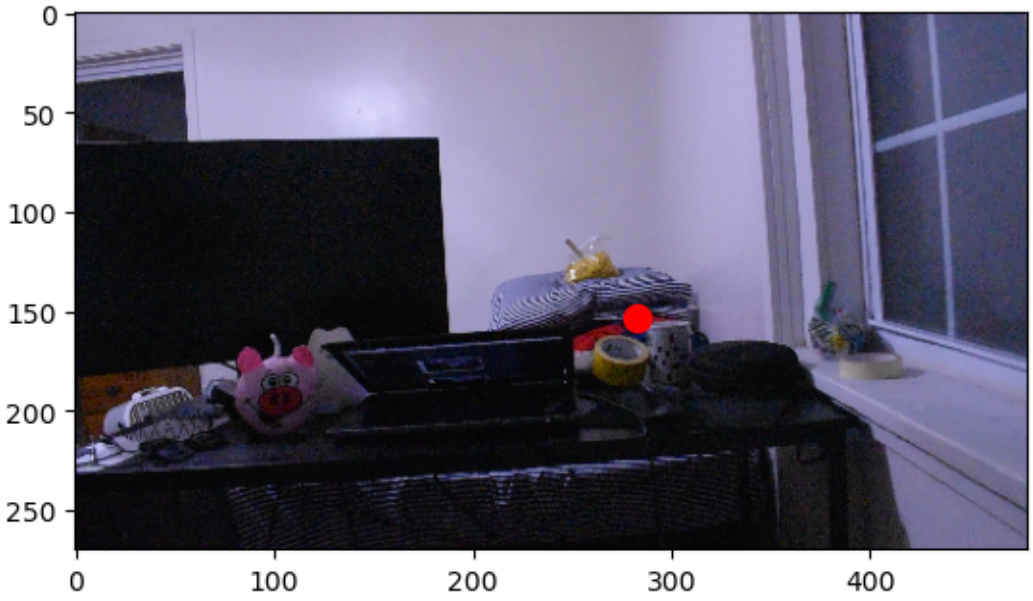




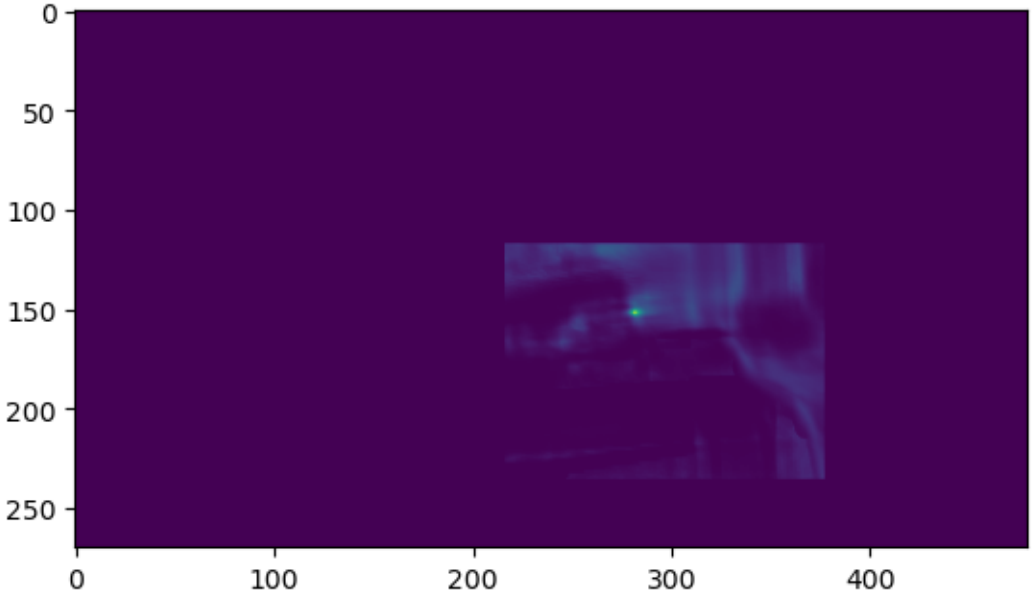
94%| | 96/102 [01:57<00:07, 1.24s/it]

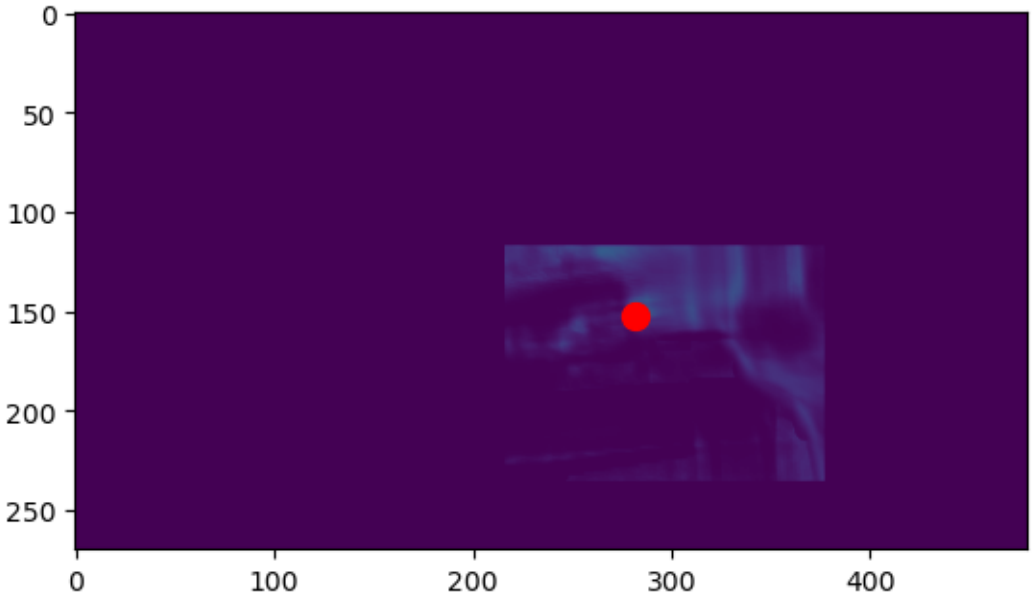

0
50
100
150
200
250
0
100
200
300
400

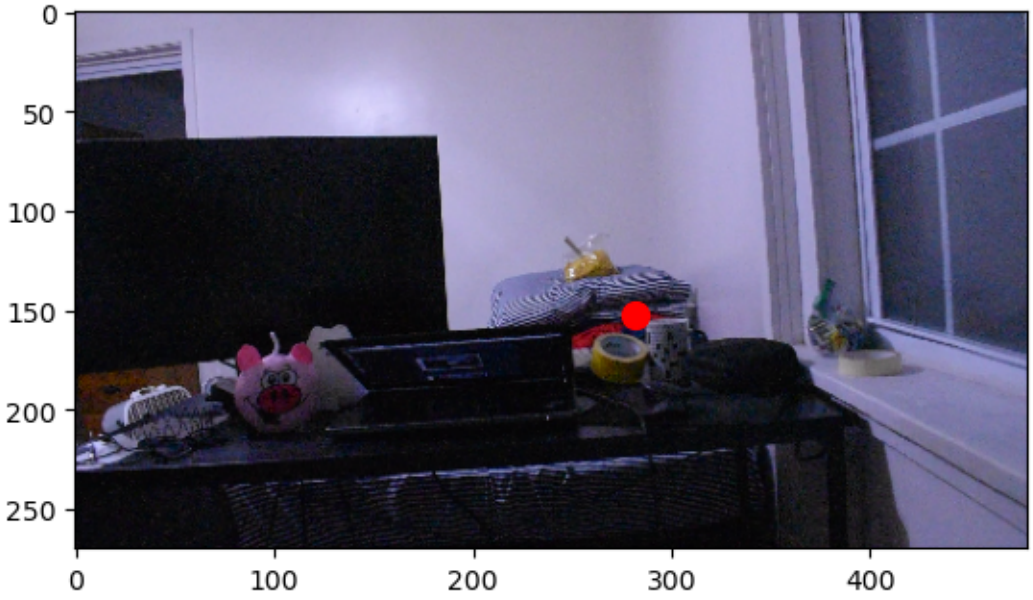

0
50
100
150
200
250
0
100
200
300
400



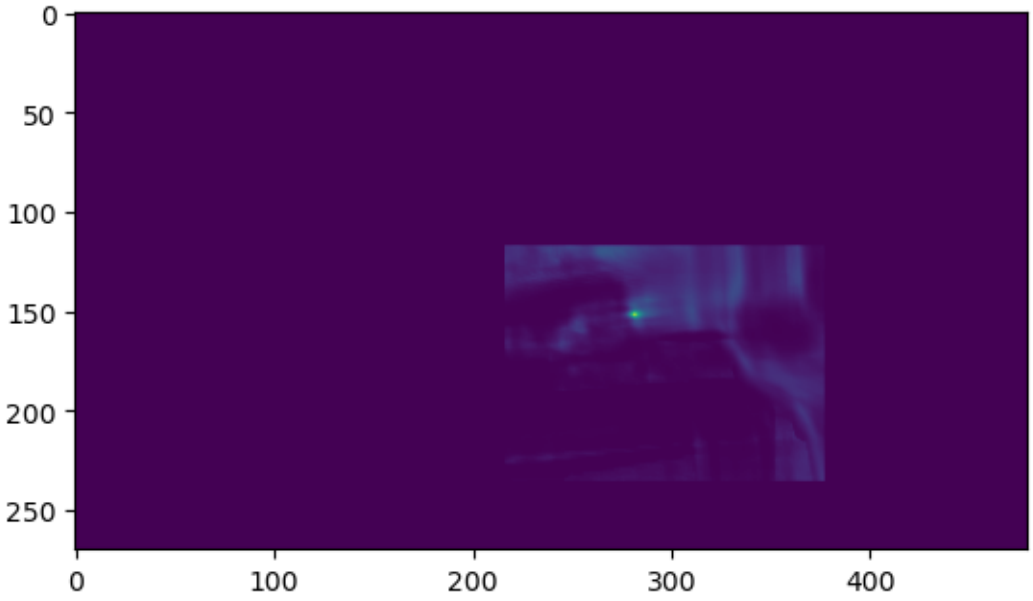
95%| | 97/102 [01:58<00:06, 1.23s/it]

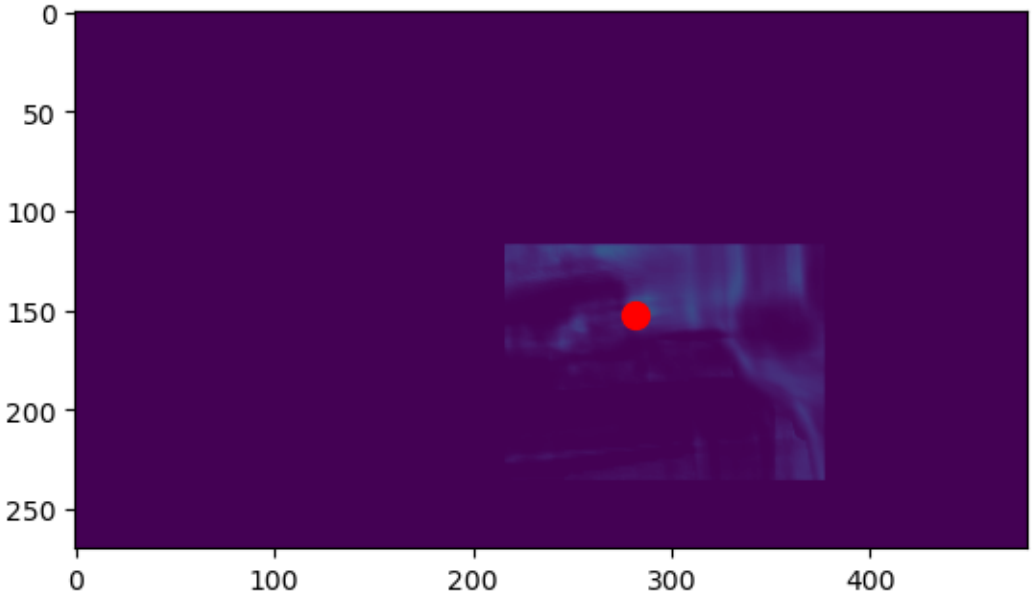


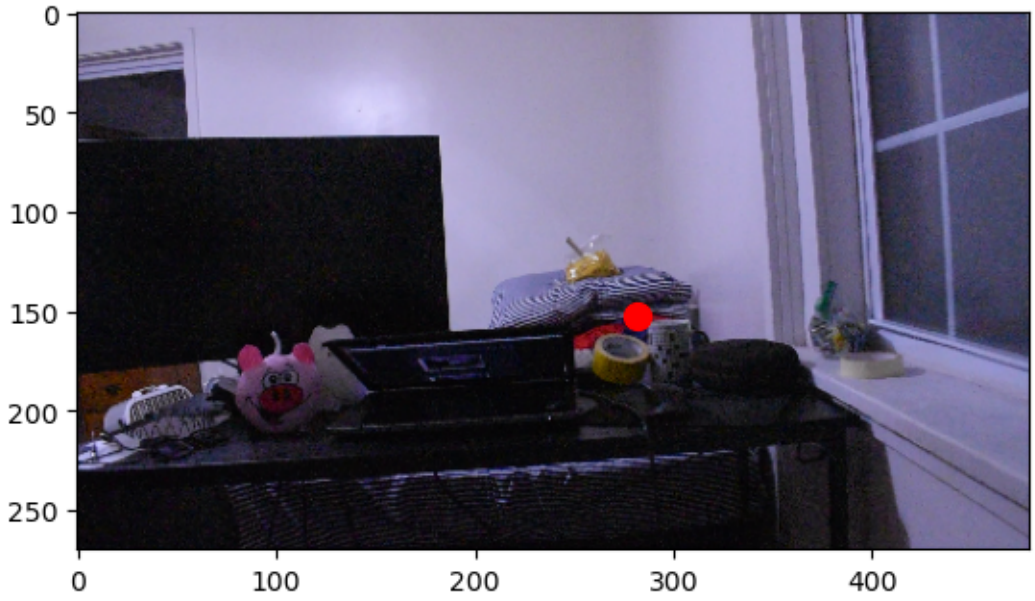




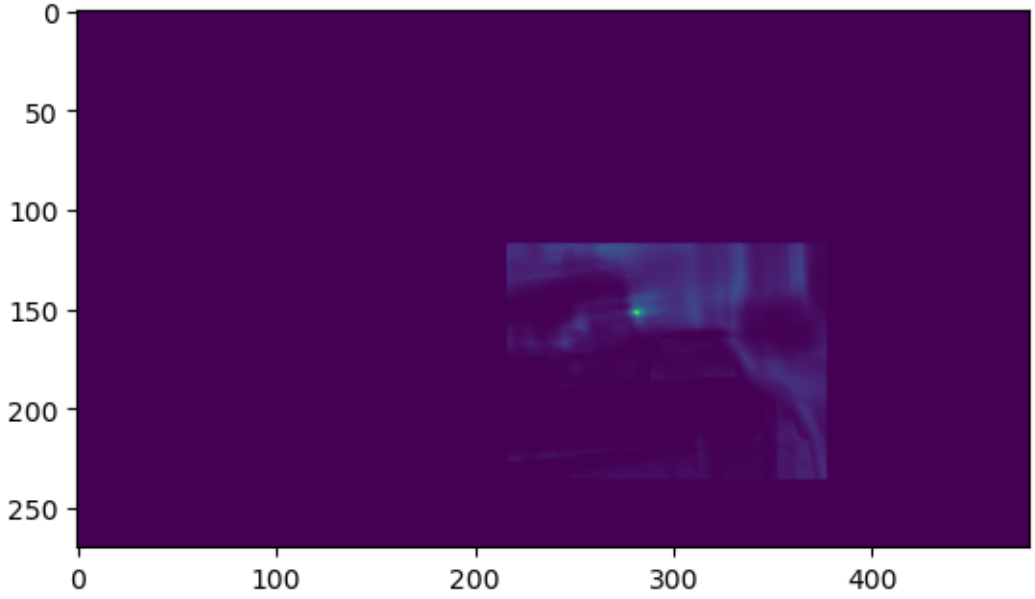
96%| | 98/102 [02:00<00:04, 1.22s/it]

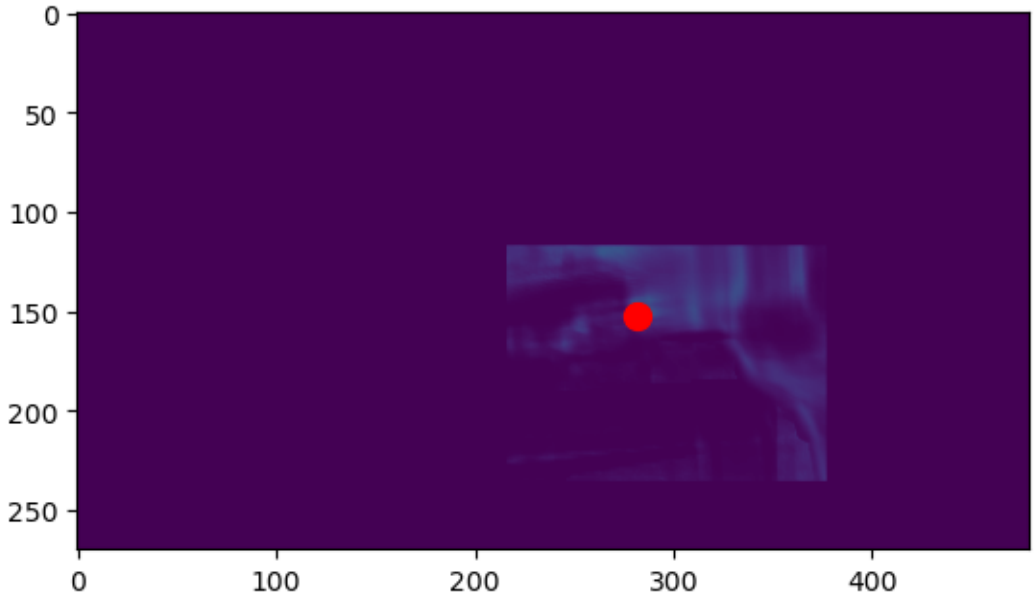

0
50
100
150
200
250
0
100
200
300
400

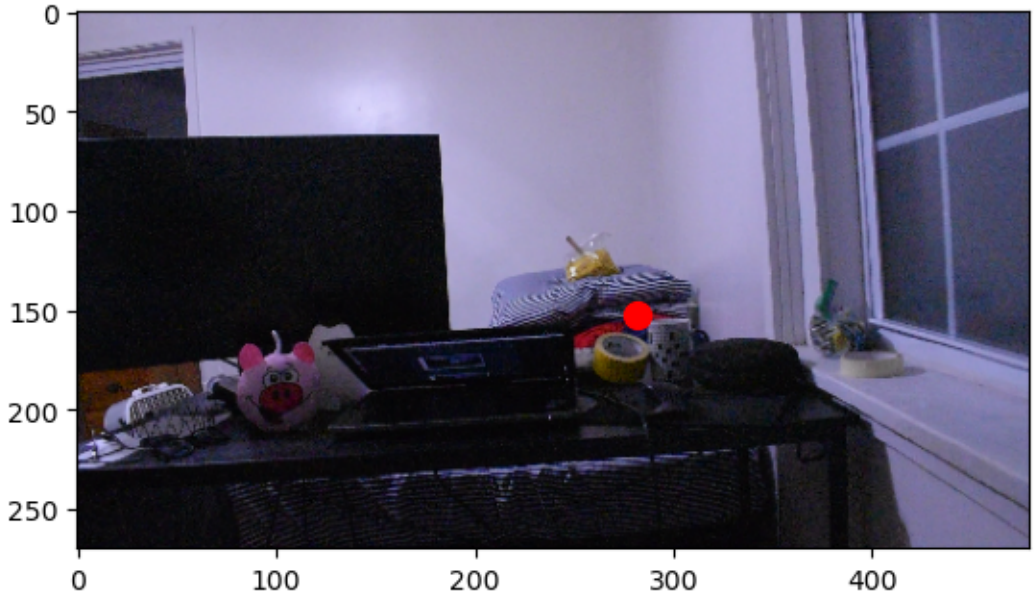

0
50
100
150
200
250
0
100
200
300
400



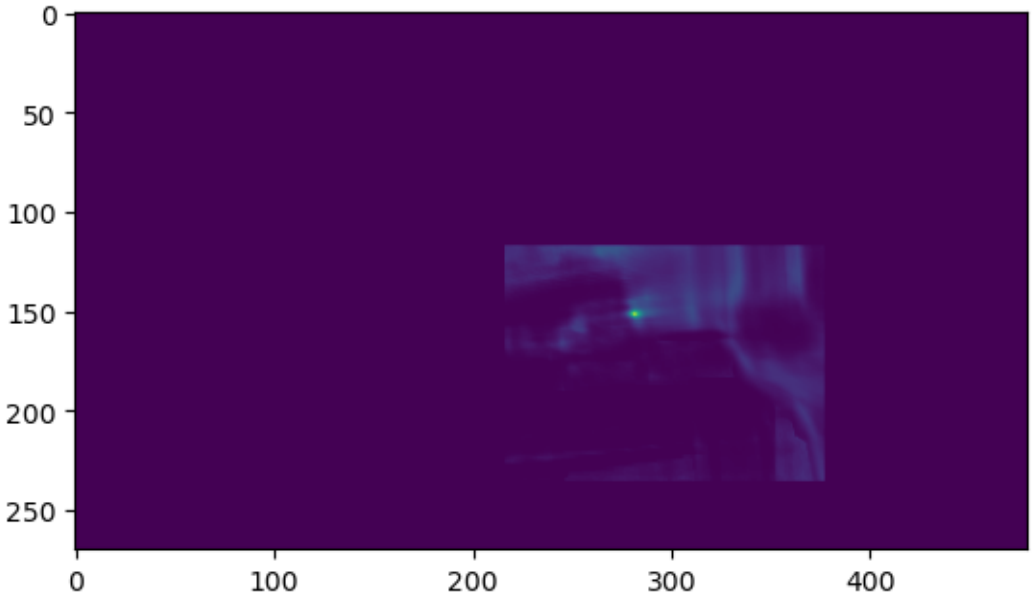
97%| | 99/102 [02:01<00:03, 1.22s/it]

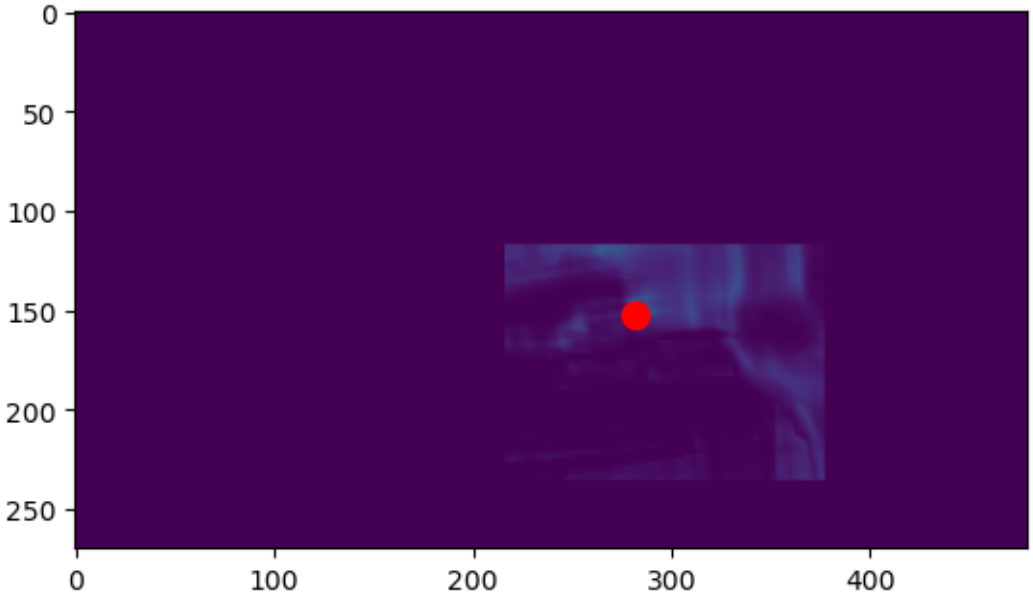


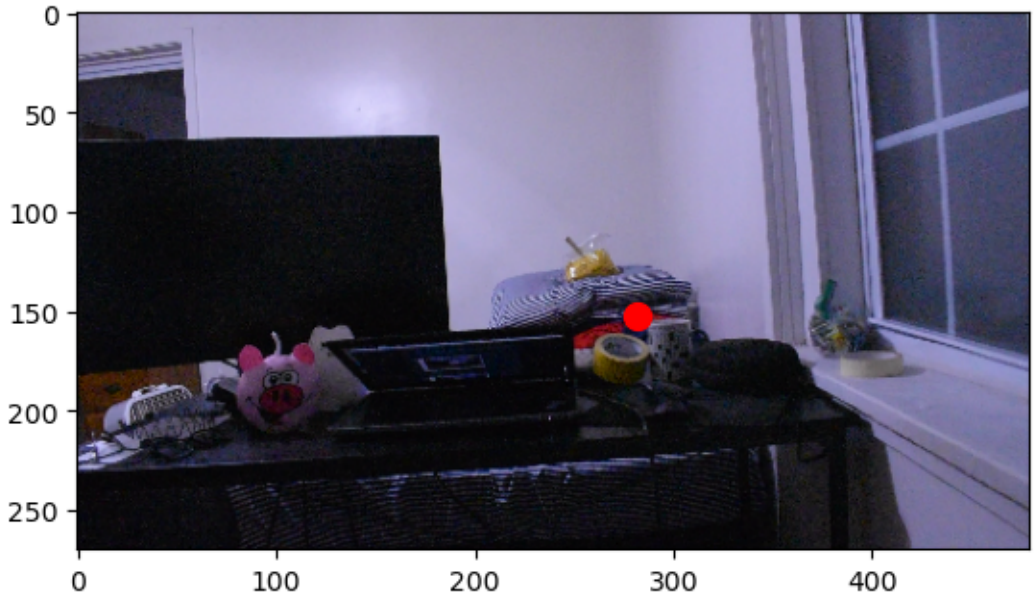




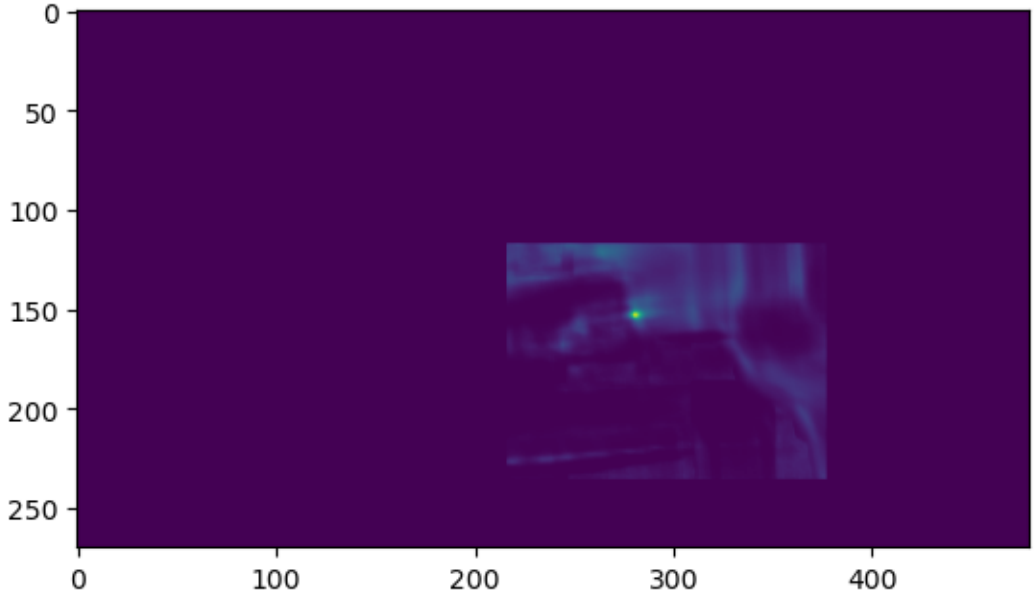
98%| | 100/102 [02:02<00:02, 1.21s/it]

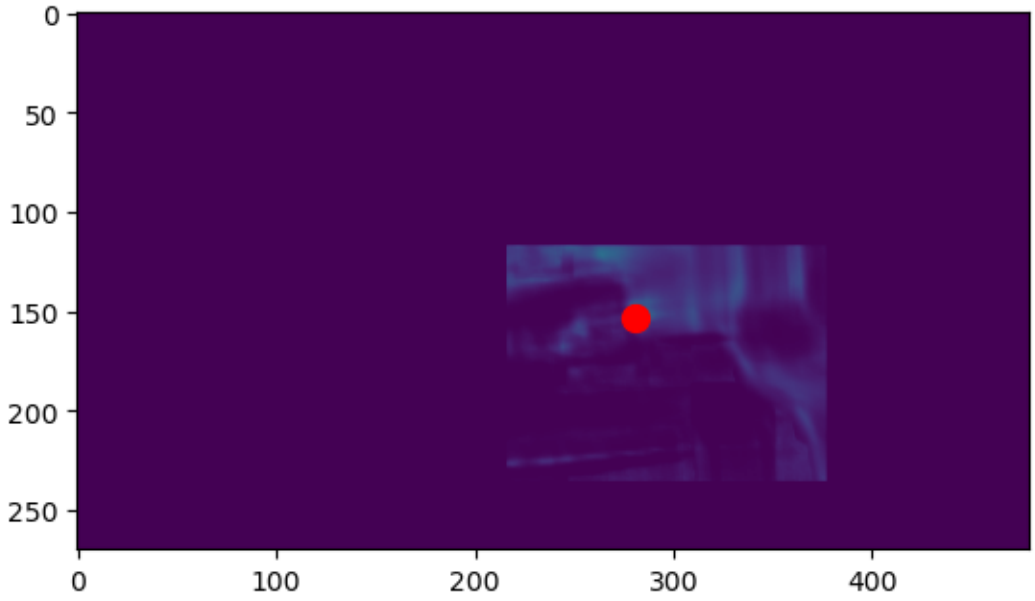

0
50
100
150
200
250
0
100
200
300
400

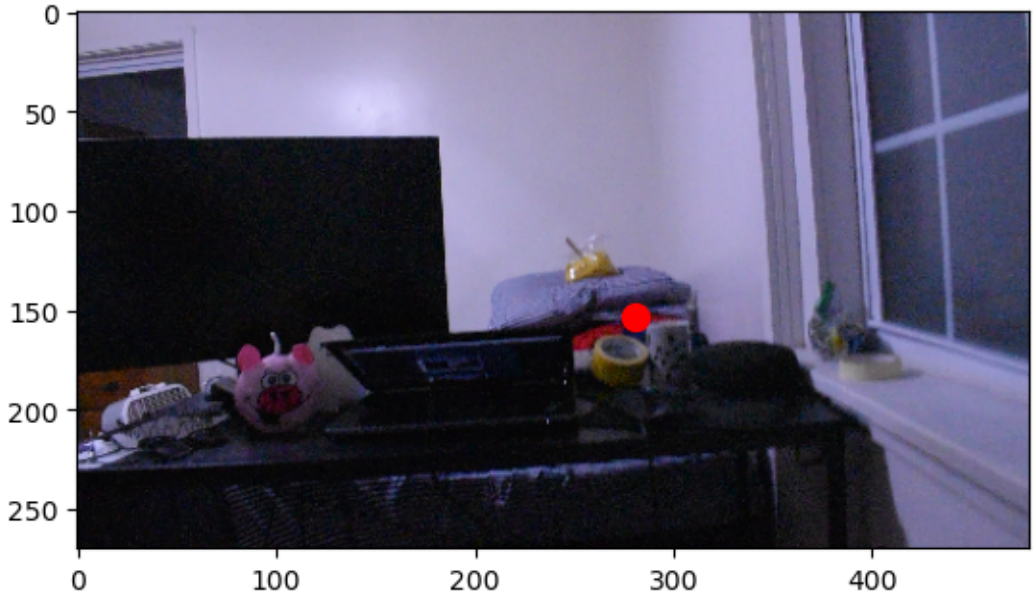

0
50
100
150
200
250
0
100
200
300
400



99%| | 101/102 [02:03<00:01, 1.21s/it]



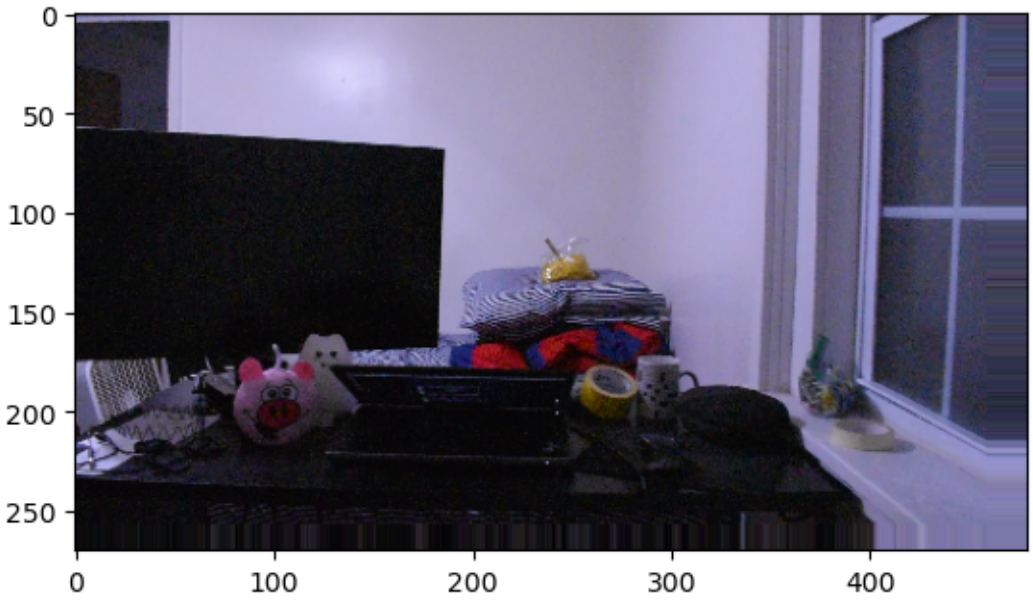





```
100%|                          | 102/102 [02:05<00:00,  1.23s/it]
shifting frames
100%|                         | 102/102 [00:00<00:00, 109.88it/s]
```

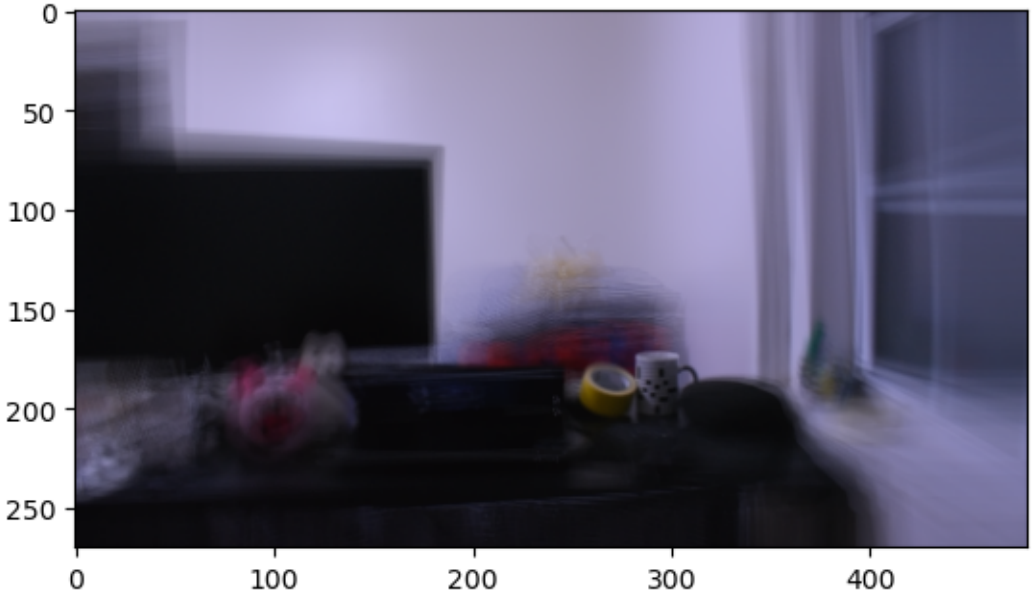
[24]:
```
plt.imshow(shifted_frames[...,19].astype('int'))
```

[24]: <matplotlib.image.AxesImage at 0x7fe165f982b0>



[25]:
```
plt.imshow(combined_frame.astype('int'))
cv2.imwrite('output-dump/rename-me.png',combined_frame[...,::-1])
```

[25]: True



[ ]:
```
shift_array
```

[16]:
```
frames.shape
```

[16]:
```
(270, 480, 3, 102)
```

[ ]:
```
bbox
```

[ ]:
```

```