ComputerVision HW1

Domenic Mancuso

September 2025

1.1 GUI features in OpenCV

I had an enjoyable time playing with the different features in OpenCV from that small amount of documentation I read. I used the code that creates a circle upon double click and then modified it to make it more fun. I was able to event listen for left mouse clicks, releases, and any moving of the cursor on the screen to draw a circle. I also used python's random library to create random colors for the circles it draws so that every time I move my mouse on the screen it would leave a trail of rainbow circles. I also had fun changing the screen size. I next used the 'complicated' version and edited that as well. You can make some interesting designs. I also thought the control flow for the second program was interesting.

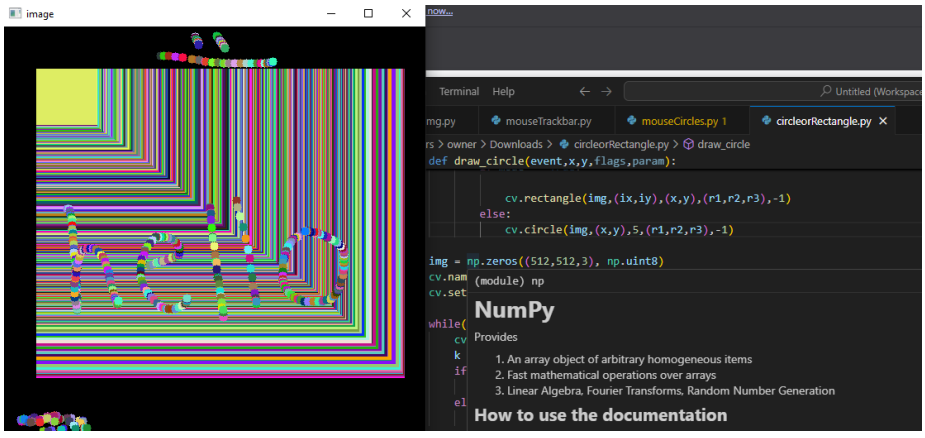

Figure 1: Circle or rectangle! Modified!

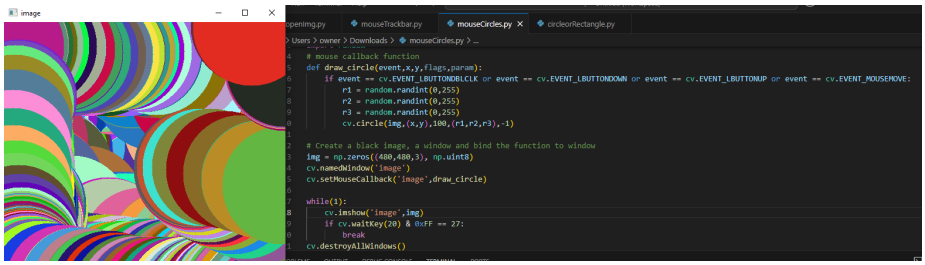

Figure 2: I love colors!

1.2: Sobel

It seems as though the way OpenCV documents its parts is by including large pages of many subsections. The cv::Sobel I viewed was near the end of a long list of methods related to Image Filtering.
The actual space for Sobel contained all of the function parameters along with their data types. This code was represented in both C++ and Python.
The reading then describes in math terms what the method is capable of; linear algebra that is a little bit above my head right now.
The section then describes what each parameter is used for and links to related methods. Update 9/19/2025 since our last meeting I understand how Sobel is used for edge detection. The math still is over my head but I now understand the significance of Sobel.

1 2

The slides were very helpful for Task 2. I was able to use them to display text as a watermark and change the video to grayscale. I used AI to help me with reverse video playback (counting frames, adding to array and reversing it) and I used the knowledge from this task to add subtitles to the video. I figured out how to add subtitles by noticing that cv.WaitKey used frame rate as an input so I set frame rate to a constant. I then used math to determine what frames are between a certain amount of seconds and I put text in during that time. I used the lecture slides information about vstack and hstack to combine the videos into a 2x2 grid.

```
# 4 Bottom-Right: Subtitled Video
bottom_right = resized_forward.copy()
if 2<= frame_index/FRAME_RATE <=7:
    cv.putText(bottom_right, "this is the first sentence!",
if 9<= frame_index/FRAME_RATE <=11:
    cv.putText(bottom_right, "this is the second!", (50, r
```

Figure 3: Subtitled Video

```
if not cap_rev.isOpened():
    print("Error: Could not open video file.")
    exit()

all_frames = []
while True:
    ret, frame = cap_rev.read()
    if not ret:
        break
    if frame is not None and frame.ndim == 3:
        # Save to array so we can put it in reverse
        all_frames.append(frame)
        # Save videos
        out4x.write(frame)
        out2x.write(frame)
        outhalfx.write(frame)

cap_rev.release()
if not all_frames:
    print("No frames were loaded. Check your video file.")
    exit()

all_frames.reverse()
frame_count = len(all_frames)
print(f"Loaded {frame_count} frames for reverse playback.")
```

Figure 4: Reversed Video pt1

```
frame_index = 0
while True:
    ret, frame = cap.read()

    if not ret:
        print("Video stream has ended.")
        break

    if frame_index >= frame_count:
        frame_index = 0

    backwrd_img = all_frames[frame_index]
    frame_index += 1
```

Figure 5: Reversed Video pt2

```
# 1 Top-Left: Grayscale
gray = cv.cvtColor(resized_forward, cv.COLOR_BGR2GRAY)
top_left = cv.cvtColor(gray, cv.COLOR_GRAY2BGR)
```

Figure 6: Grayscale Video

```
# Stack the images to create the 2x2 grid
grid_top = np.hstack((top_left, top_right))
grid_bottom = np.hstack((bottom_left, bottom_right))
final = np.vstack((grid_top, grid_bottom))
# Save as output.avi
out.write(final)
cv.imshow('2x2 Video Grid', final)
```

Figure 7: Hstack Vstack

2 Task 3: Speed Modification

I figured out how openCV uses VideoWriter to save the output of a video. I used it to create all three video types (4x, 2x, and 1/2x) by multiplying the frame rate constant by a number. I also outputted the 2x2 grid video to submit.

```
fourcc = cv.VideoWriter_fourcc(*'XVID')
# We will be saving the 2x2 video and the original video at three different speeds
out = cv.VideoWriter('outputfinal.avi', fourcc, FRAME_RATE, (1280,  720))
out4x = cv.VideoWriter('output4x.avi', fourcc, FRAME_RATE*4, (1280,  720))
out2x = cv.VideoWriter('output2x.avi', fourcc, FRAME_RATE*2, (1280,  720))
outhalfx = cv.VideoWriter('outputhalfx.avi', fourcc, FRAME_RATE/2, (1280,  720))
```

Figure 8: Video Writer pt1

```
# Save videos
out4x.write(frame)
out2x.write(frame)
outhalfx.write(frame)
```

Figure 9: Video Writer pt2

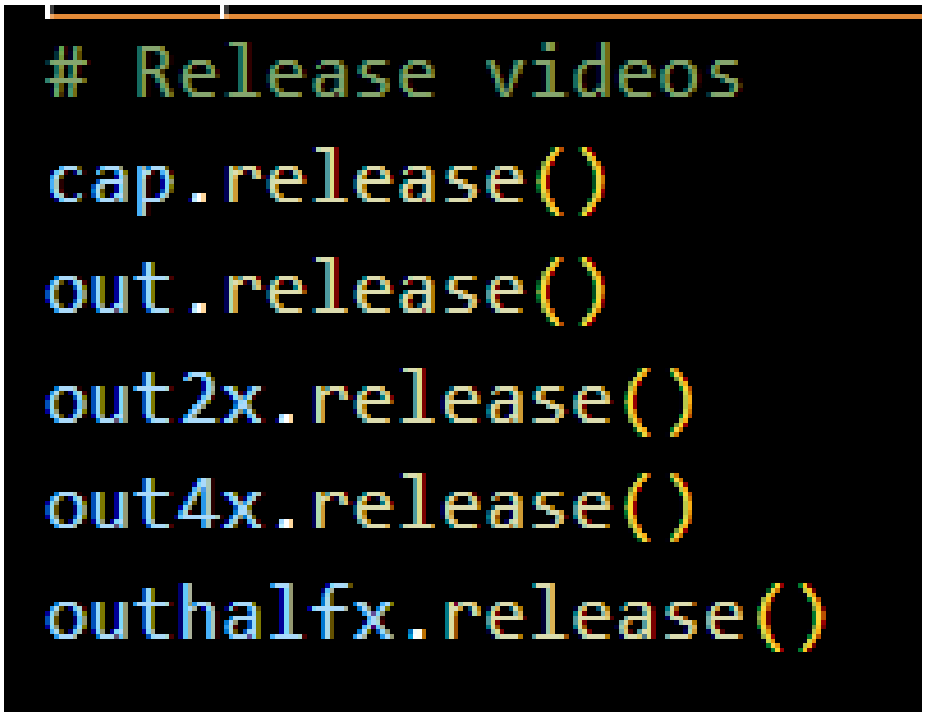

Figure 10: Video Writer pt3