

Project 1: Real-time filtering

Student: Yunyi Chi

Description:

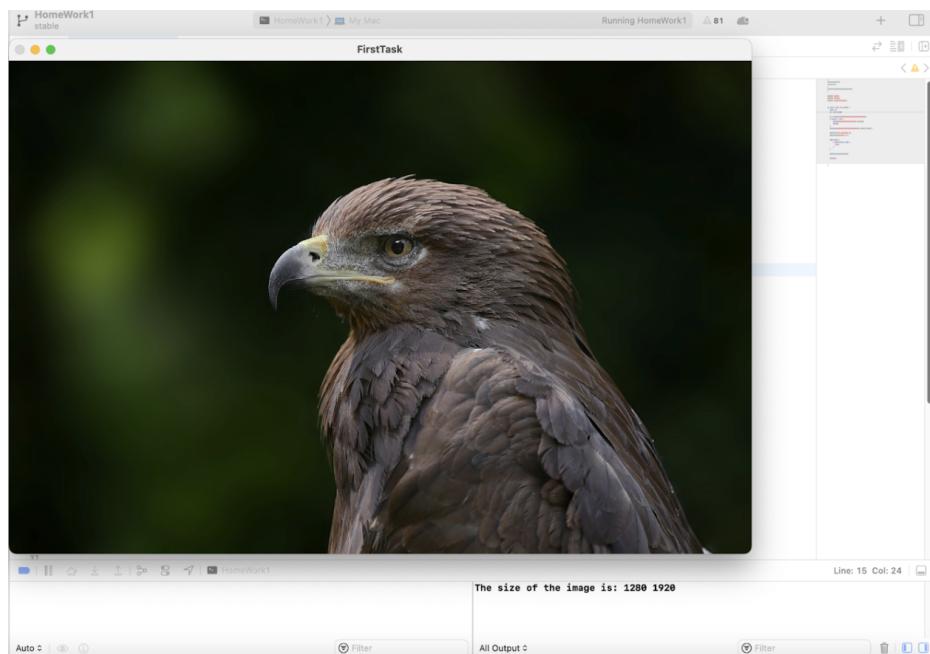
In this project, I used C++ and OpenCV to realize the opening, capturing, manipulating, and writing operations of images and real-time videos in the Xcode compilation environment.

The focus is to implement different filters for real-time video to obtain different effects and images, such as Gaussian filter (Blur filter) and Sobel filters. In order to improve the time complexity of filtering, I have taken some measures, such as dismantling some filters into separable filters.

In addition to the basic tasks, I also had 3 more tasks as extensions, which are two new filters and a video save function.

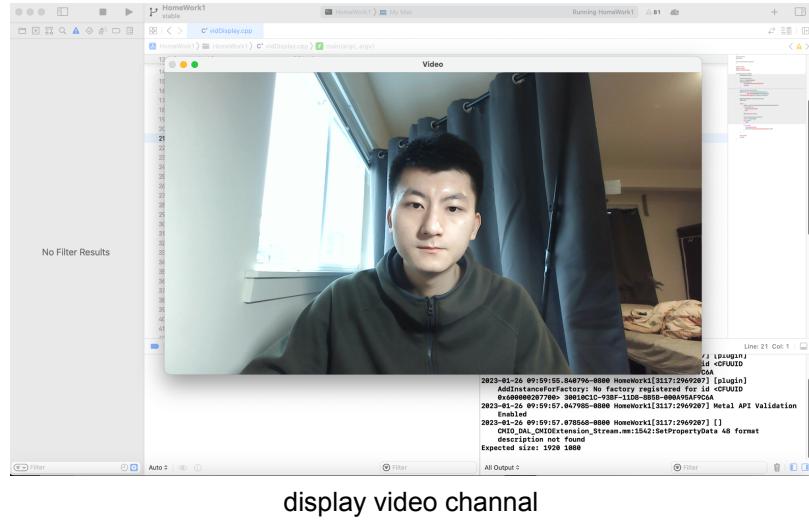
Task 1: Read an image from a file and display it

Read the 'animal.jpg' file in the same dictionary and open a window to show it. If the user types 'q', the program quits.



Task 2: Display live video

Open a video channel, create a window, and then loop, capturing a new frame and displaying it each time through the loop. Program quits if the user types 'q'. Save an frame in current dictionary if the user types 's'.



display video channal



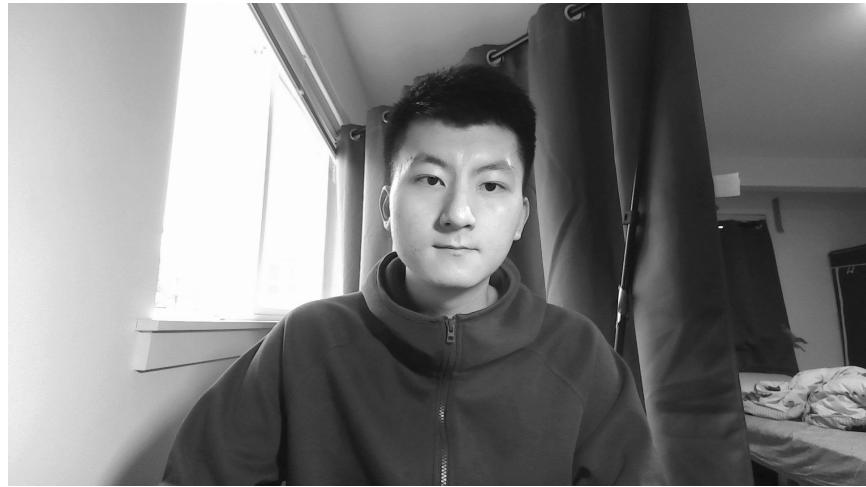
saved frame

Task 3: Display greyscale live video

If the user types 'g', displays a greyscale version of the image.



original



modified

Task 4: Display alternative greyscale live video

A different method of generating greyscale video.



original



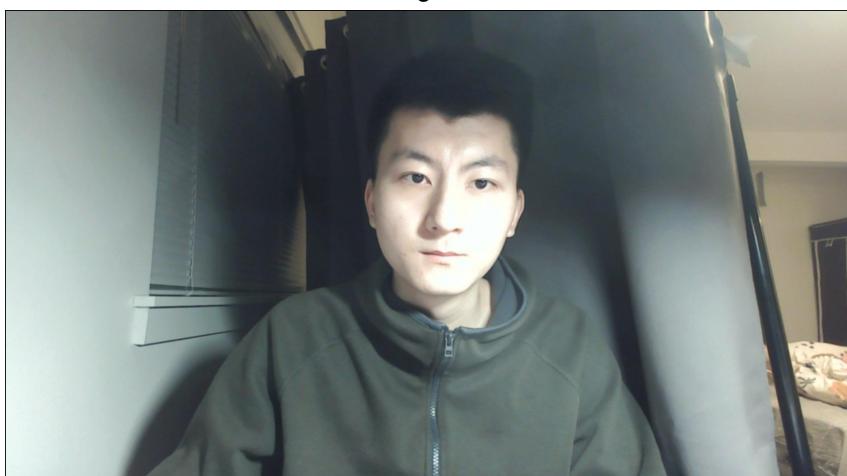
modified

Task 5: Implement a 5x5 Gaussian filter as separable 1x5 filters

To better show the blur effect, according to post <https://piazza.com/class/lcs1garq31l74i/post/33>. I also apply the blur filter more than once on the same frame to see if the effect noticeably increases. (Extra blur effect just for testing).



original



modified(1 blur)



modified(2 blur)



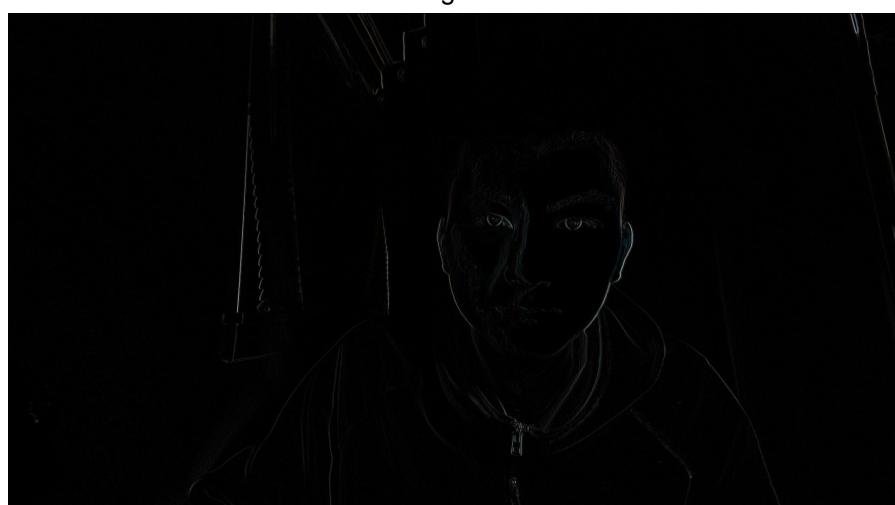
modified(3 blur)

Task 6: Implement a 3x3 Sobel X and 3x3 Sobel Y filter as separable 1x3 filters

sobelX3x3



original



modified

sobelY3x3



original



modified

Task 7: Implement a function that generates a gradient magnitude image from the X and Y Sobel images



original



modified

Task 8: Implement a function that blurs and quantizes a color image



original



modified

Task 9: Implement a live video cartoonization function using the gradient magnitude and blur/quantize filters



original



modified

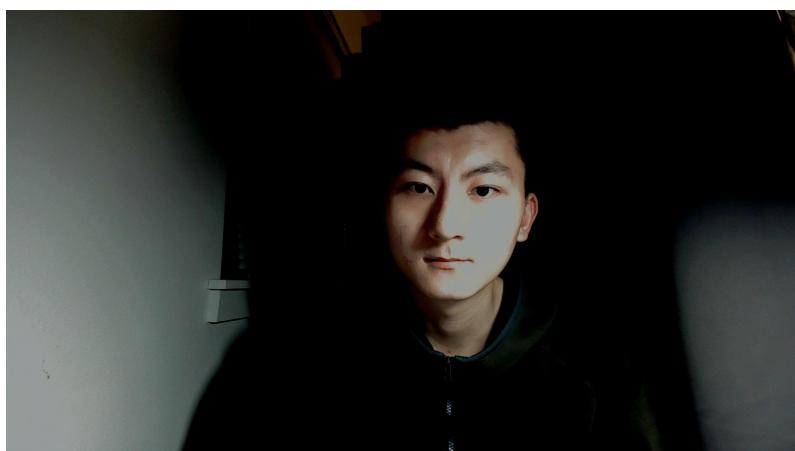
Task 10: allow user to adjust brightness, use ‘u’ to increase brightness and ‘d’ to reduce brightness



original



Modified(increase brightness)



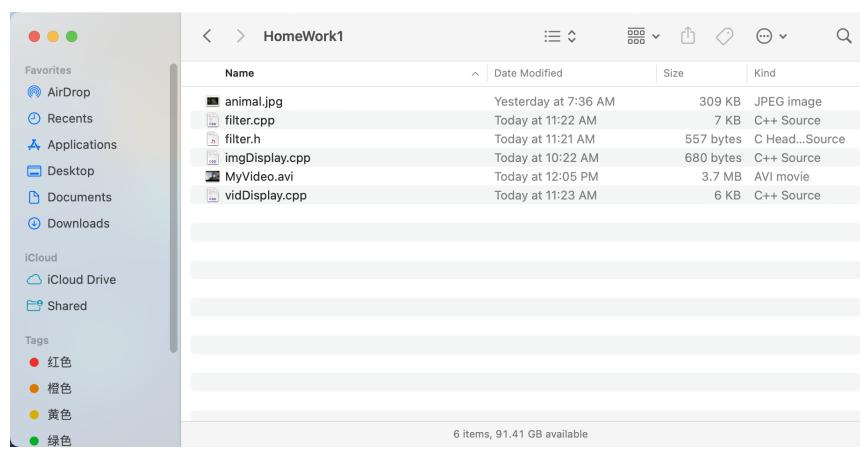
Modified(reduce brightness)

Extensions (3 extensions)

- Save a short video
- Negative Filter
- Mirror effect

1. Save a short video.

Click 'v' to start saving a video and click 'v' again to stop saving.



saved video

2. Negative Filter Effect (from scratch, written by myself).

Click 'n' to get a Negative Filter Effect image. Modified pixel of original frames. Negative all BGR values to get a Negative Filter image.



original



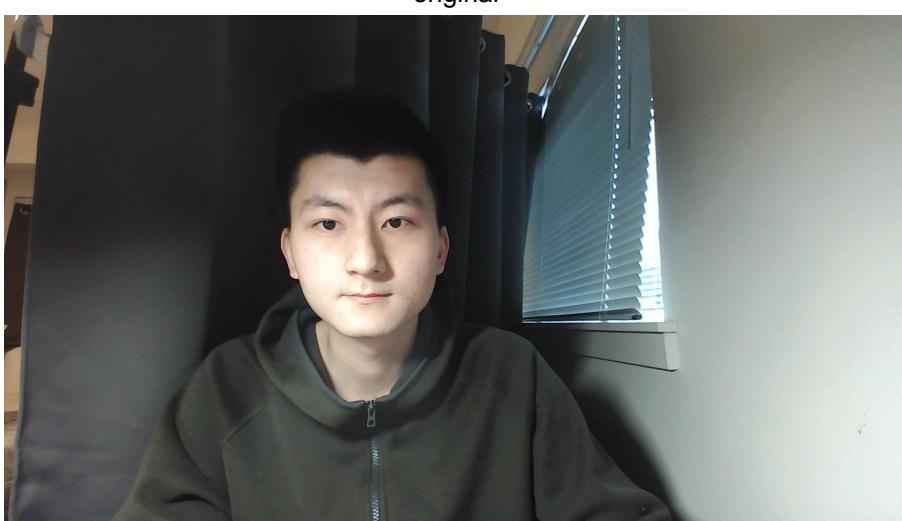
modified

3. Mirror effect.

Click 'i' to get a mirror image



original



modified

What I learned

From this project and these 3 weeks study, I learn the following things

1. Some basic C++ knowledge and syntax
2. Some opencv functions and syntax
3. Different Filters, such as Gaussian filter and Sobel filter
4. Methods to Improve the running speed of the program in the actual production environment, such as using pointer and separable filters.
5. Some linear algebra knowledge

Acknowledgement

Maxwell's lecture notes, by *Dr. Bruce A. Maxwell*

Computer Vision: Algorithms and Applications, 2nd ed, by *Richard Szeliski*

OpenCV Tutorials: https://docs.opencv.org/4.x/d9/df8/tutorial_root.html