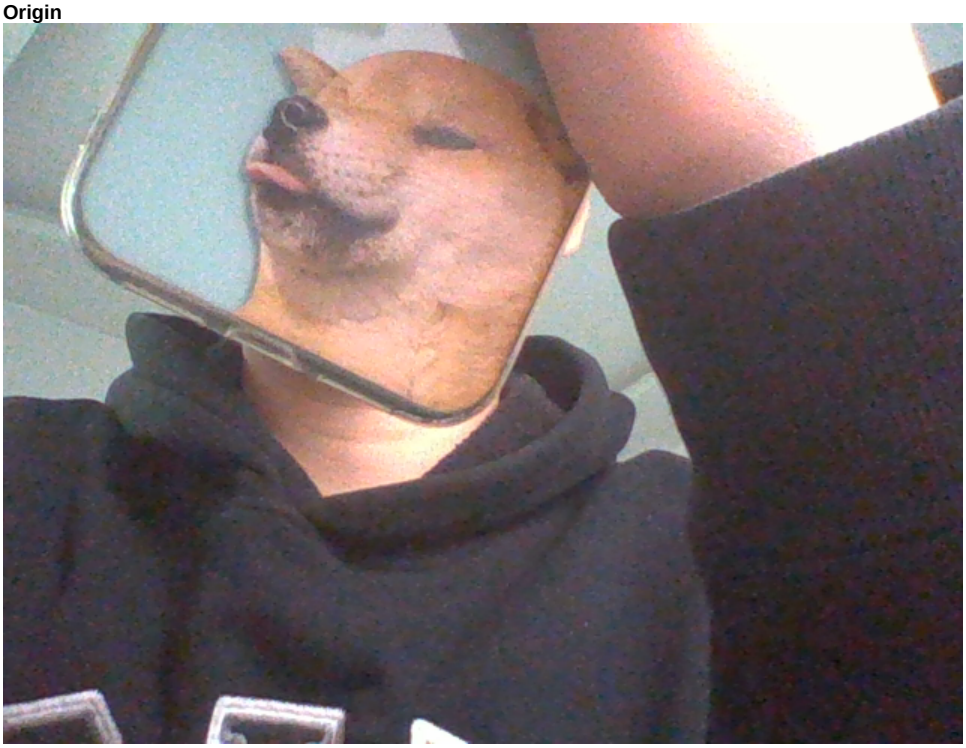
Assignment#1-Report-Jingtong

Summary

For this project, I implemented a set of filters for various effects on the image and video stream, such as blurring, edge detection, emboss, brightness /contrast adjustment, sepial transform, and cartoonization. These effects are vivid and are similar to filter effects in some camera apps. Most of them are implemented by accessing the pixels instead of calling the existing OpenCV functions, which help me have a better understanding of how the OpenCV works on the cv::Mat object.

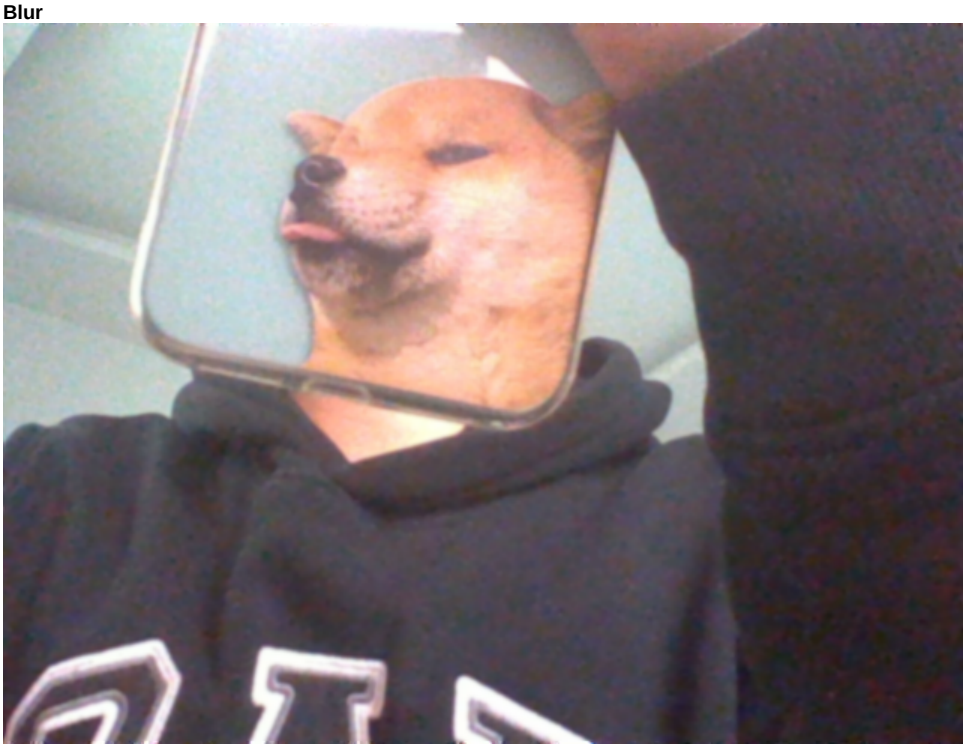
Core Project

1. **Origin**

2. **Gray**
Each channel is weighted as this:

```
value = 0.299 * RED_CHANNEL + 0.587 * GREEN_CHANNEL + 0.114 * BLUE_CHANNEL
```

3. **Alternative Gray**
   I implement the alternative gray by copying the green color channel to the other two color channels

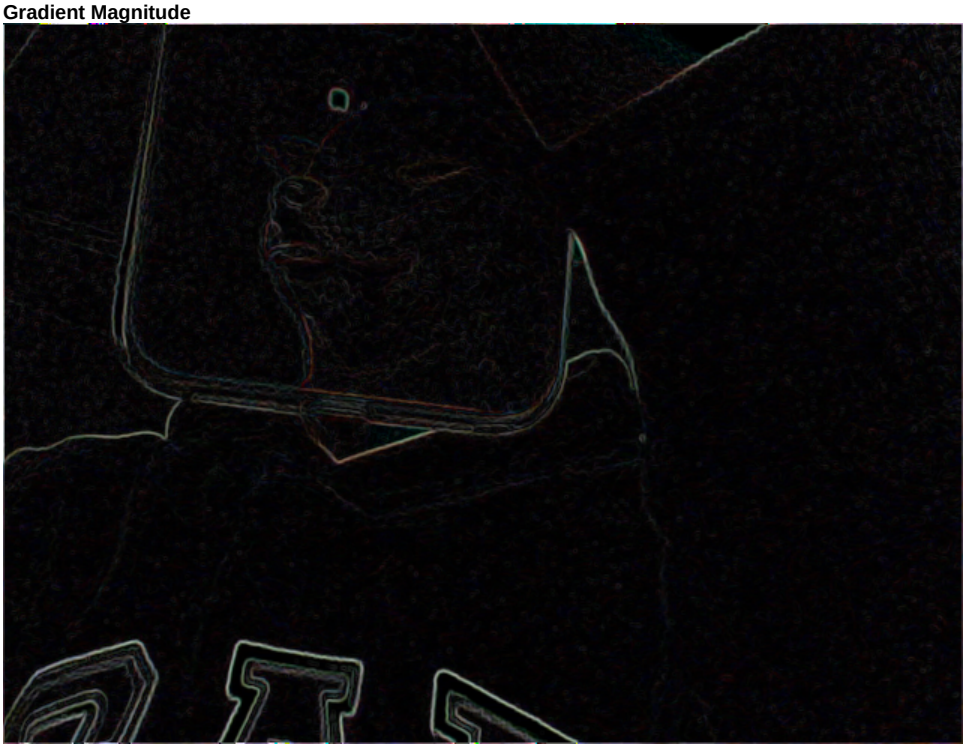
4. **Blur**



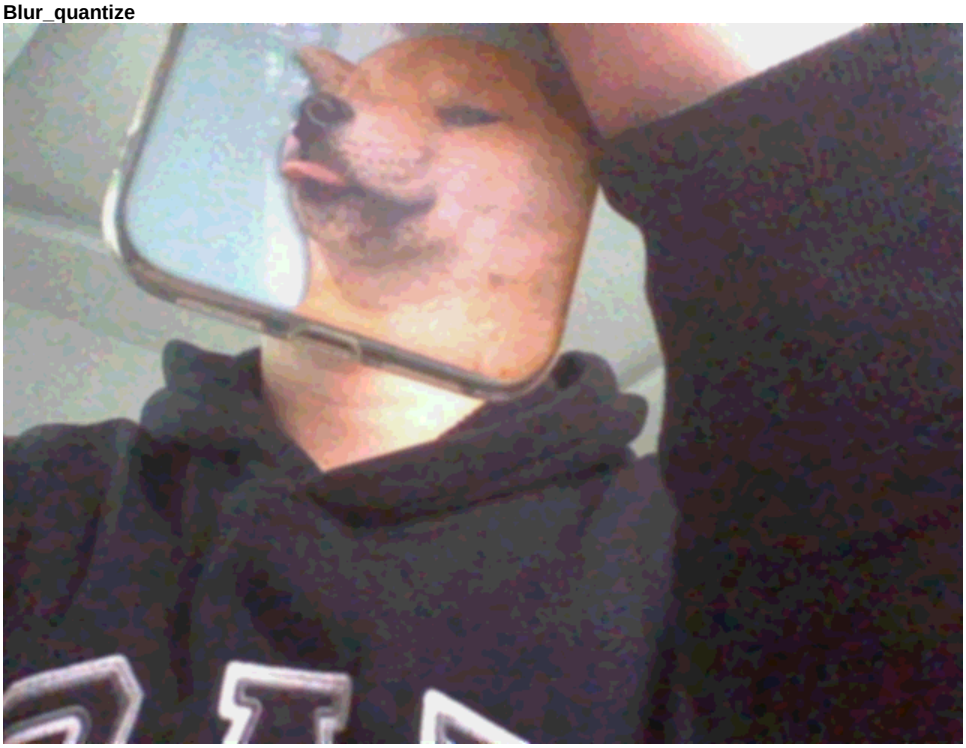
5. **X_Sobel**

6. **Y_Sobel**

7. **Gradient Magnitude**



8. **Blur_quantize**



9. **Cartoon**

10. **Negative**
Make the image a negative of itself by the equation:

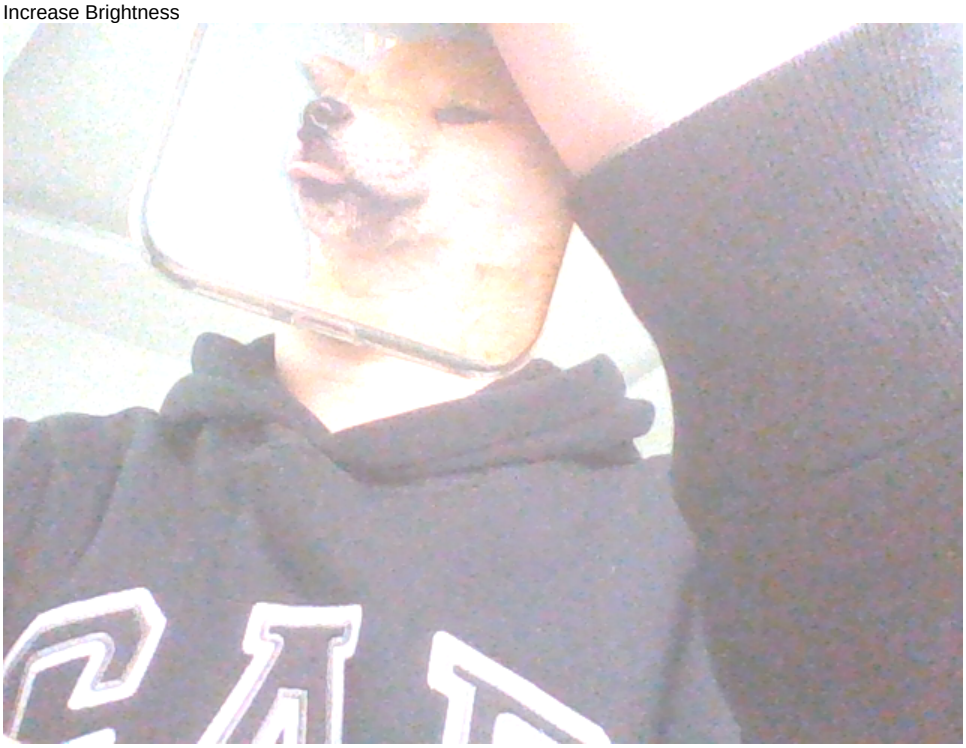
```
value = 255 - value; // it has the same effect as cv::bitwise_not function
```

My Extensions

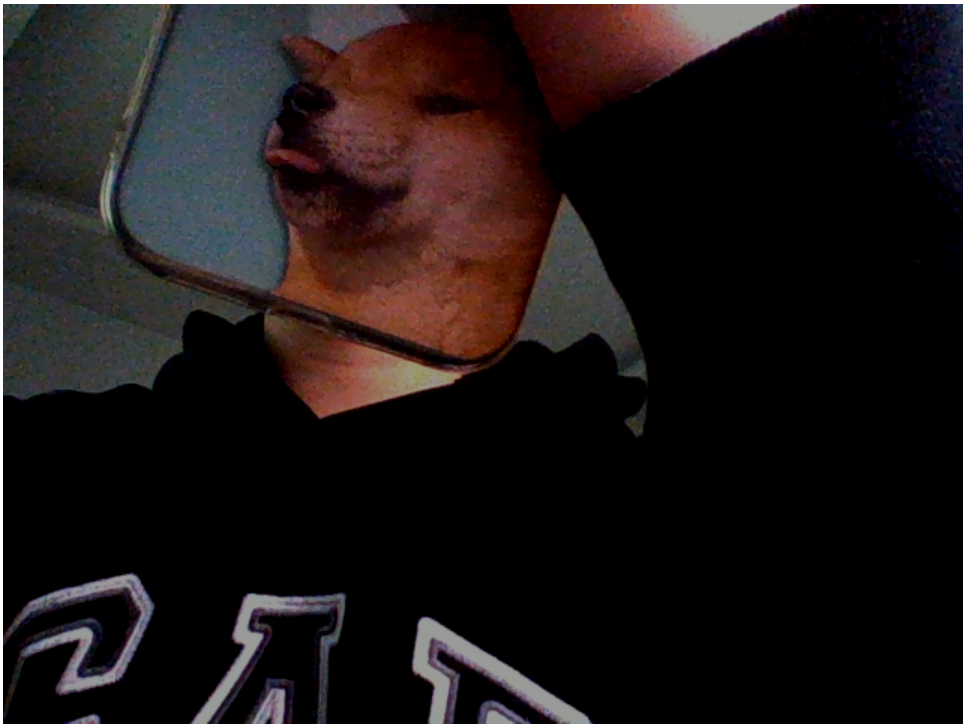
I implemented three tasks as the extensions here, which are brightness/contrast adjustment, sepia transform, and the emboss filter.

1. **Brightness and Contrast Adjustment**
   According to the OpenCV doc, the parameters and in $g(x)=f(x)+$ control the contrast and brightness, I called the cv::convertTo() functions with (,) as (2, 0) to add the contrast and (0.25, 0) to decrease the contrast, (1, 100) to add the brightness and (1, -100) to decrease the brightness.

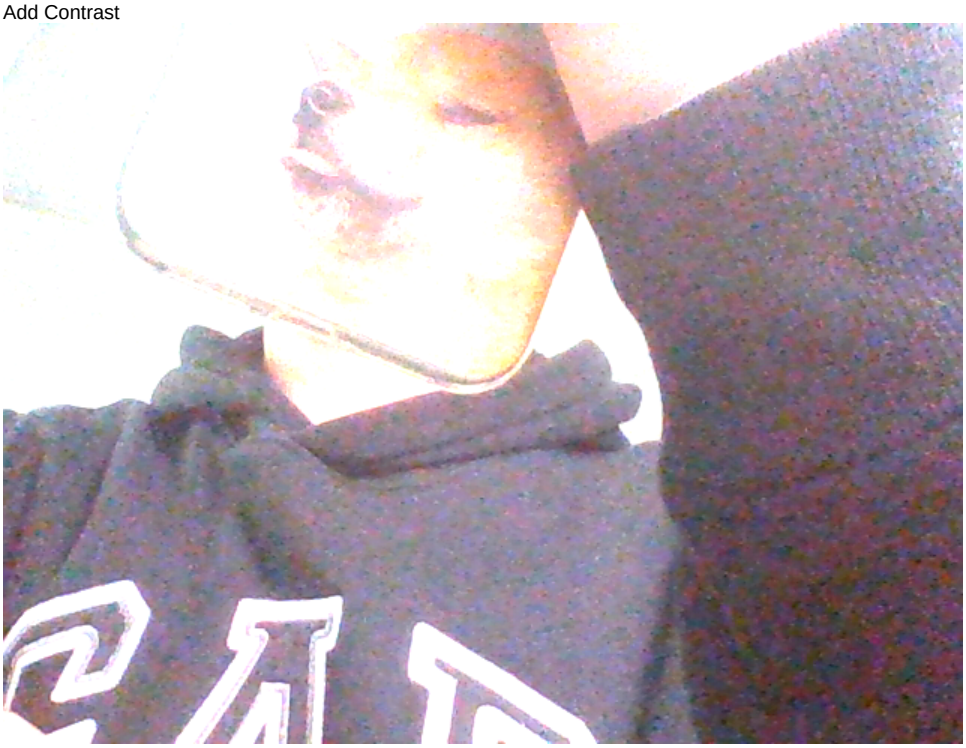
   Increase Brightness



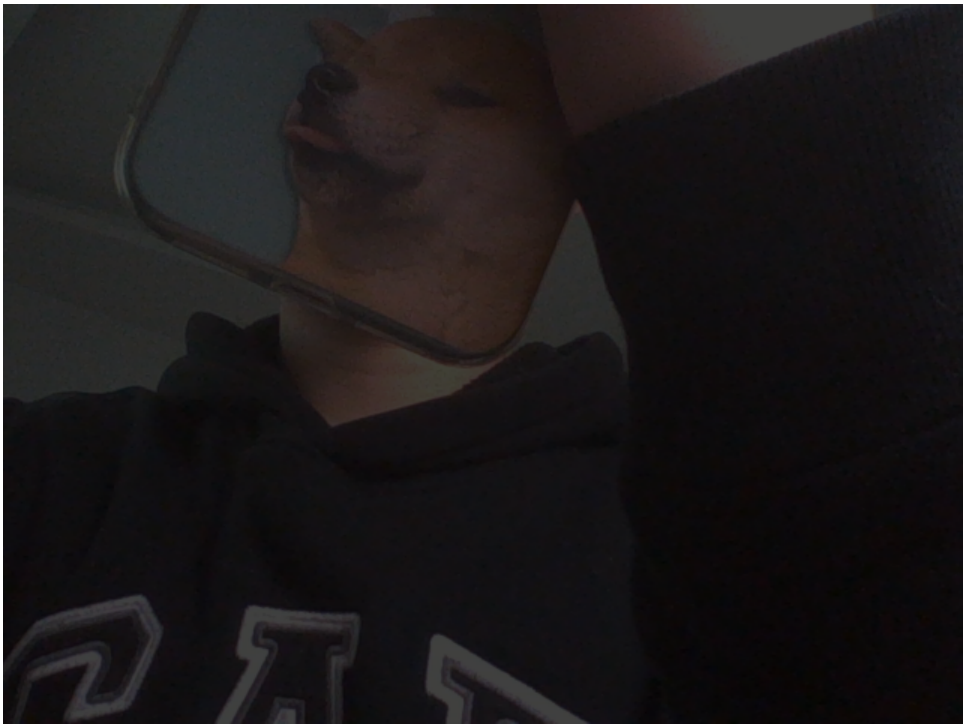
   Decrease Brightness



Add Contrast

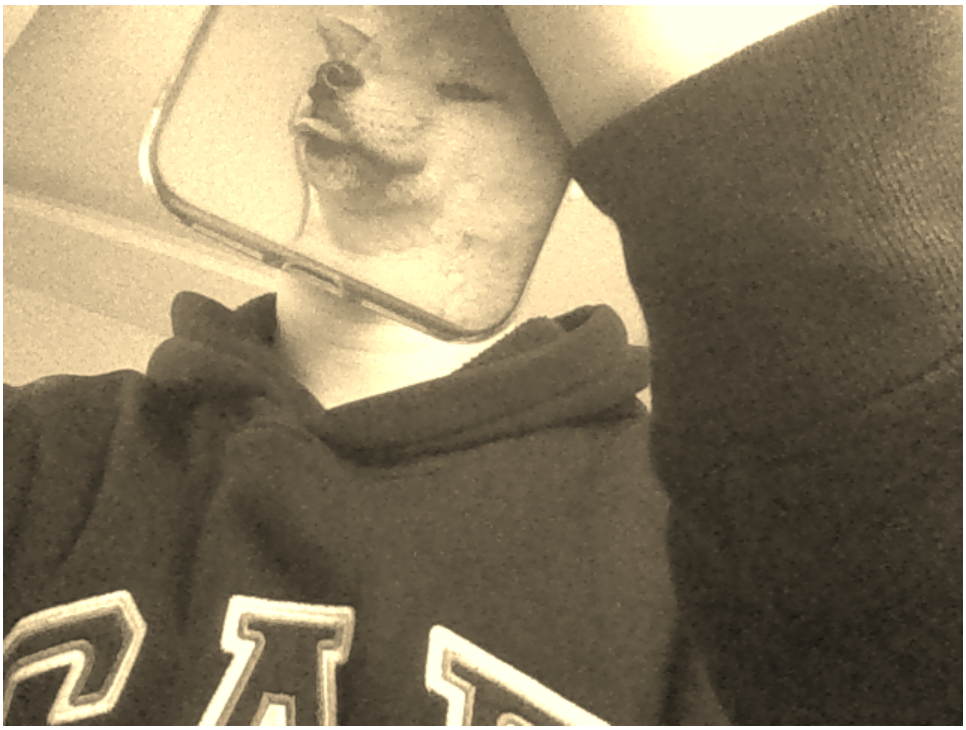


Decrease Contrast



2. **Sepia**
   Sepia is not a filter, it is a weighted combination of three channels for each new channel. The formula is as below:

```
new_r = 0.393R + 0.769G + 0.189B
new_g = 0.349R + 0.686G + 0.168B
new_b = 0.272R + 0.534G + 0.131B
```



3. **Emboss**
   I applied the 3*3 kernel to do the filter, the matrix is as below:

```
-2  -1  0
-1   1  1
 0   1  2
```

Acknowledgements

https://docs.opencv.org/4.5.1/d5/d98/tutorial_mat_operations.html

https://stackoverflow.com/questions/1061093/how-is-a-sepia-tone-created

https://setosa.io/ev/image-kernels/

https://docs.opencv.org/4.x/d3/dc1/tutorial_basic_linear_transform.html

https://en.wikipedia.org/wiki/Sepia_(color)

Reflection

From this project, I had a basic understanding of how OpenCV operates each pixel in the format of cv::Mat. Also, the filters really surprised me with different effects, especially the edge detection. Moreover, the c++ language really bothered me for a while since I haven't touched C++ for three or four years and I am very glad to be able to control the built-in data types in such an elegant way.