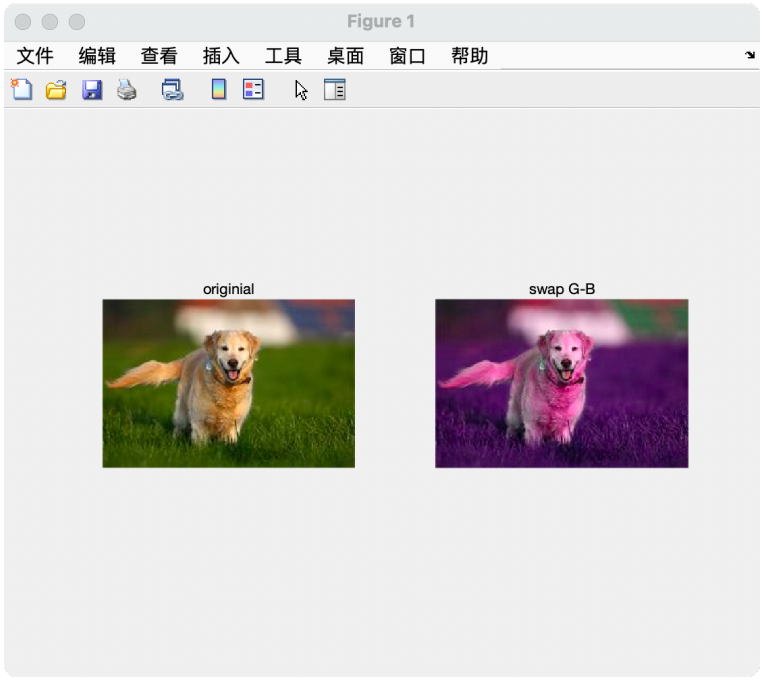
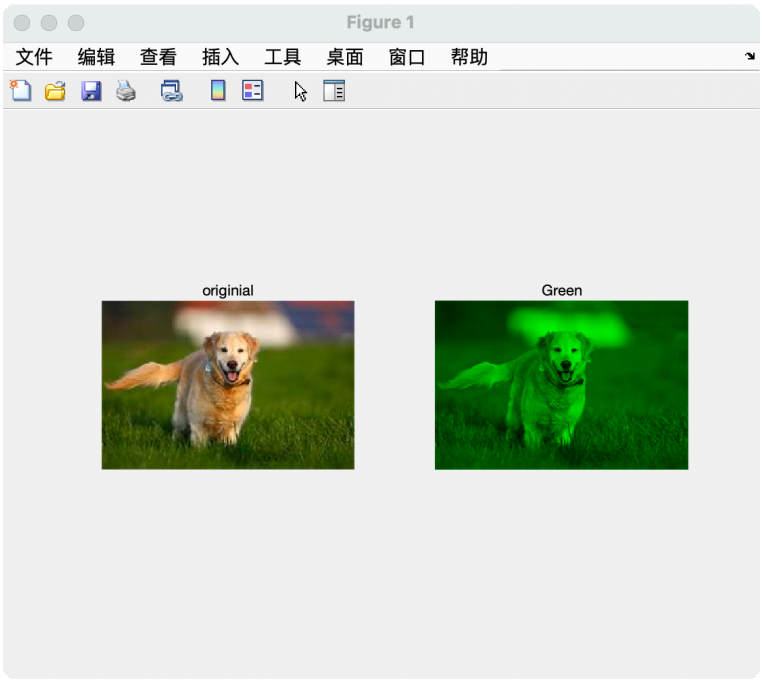
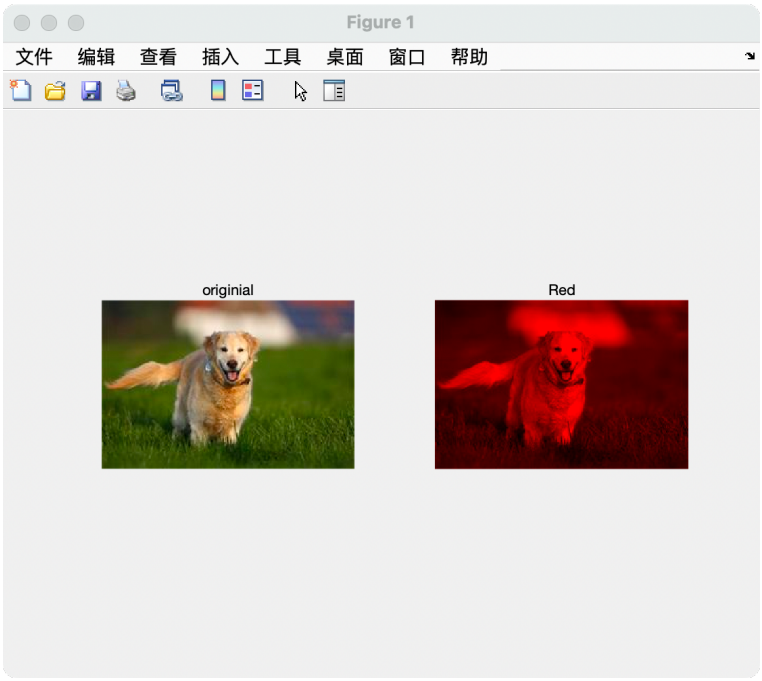


ps0-2-a-1



ps0-2-b-1

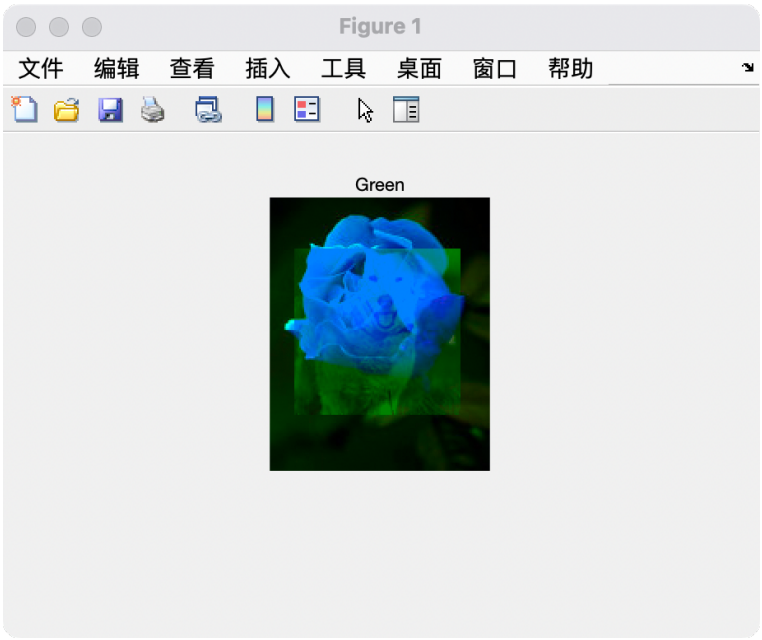


ps0-2-c-1

2-d.

I think the monochrome image of green channel is more like the one I expected because they original image has more green color.

There will always be some aspects are better than others of computer vision algorithm. What we need to do is to make the worse one get more improvements.



ps0-3-a-1

4-a.

avg	95.9184
B2	*165x133 uint8*
dev	40.2460
F1	*160x240x3 uint8*
F1_CUT	*100x100x3 uint8*
F2	*160x240x3 uint8*
G	*160x240 double*
G2	*165x133 uint8*
i	160
j	240
m	160
max	246
min	0
n	240
R2	*165x133 uint8*
sum	3683267
var	6.2198e+07
x	3

min is 0 and max is 246.

```
for i = 1:m
    for j = 1:n
        if G(i,j) > max
            max = G(i,j);
        end
        if G(i,j) < min
            min = G(i,j);
        end
```

I traverse all the pixels of the picture through two loops. And I initialize min with the possible max value 255 and max with 0. When I found a pixels value small than initial min, then the value of min equals to the value of this pixel, so finally I can get the min value of all pixels. So does the max value.

The mean is 95.9184.

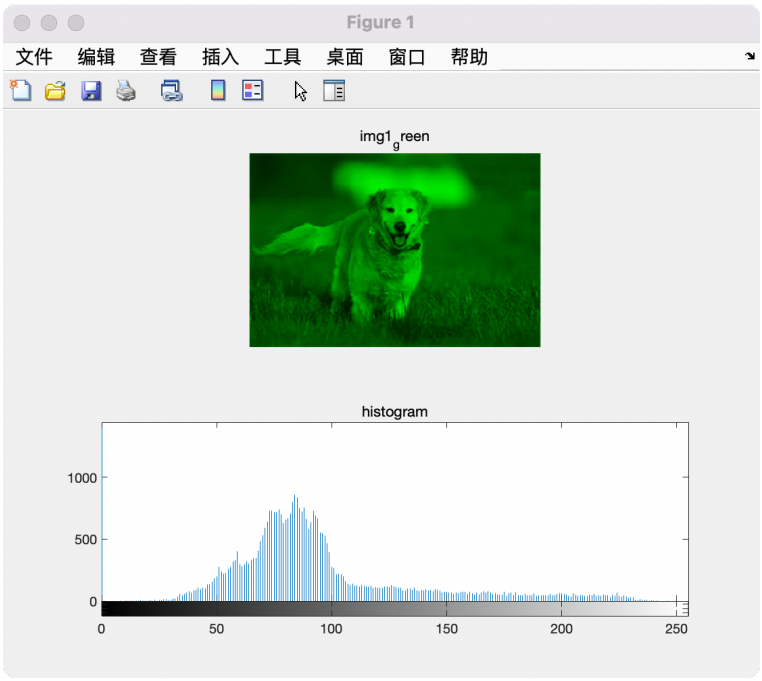
```
            sum = sum + G(i,j);
    end
end
avg = sum /(m * n);
```

As the same I used two loops to add all pixels' values together and got the sum, then divide the whole numbers of pixels, I get the mean value.

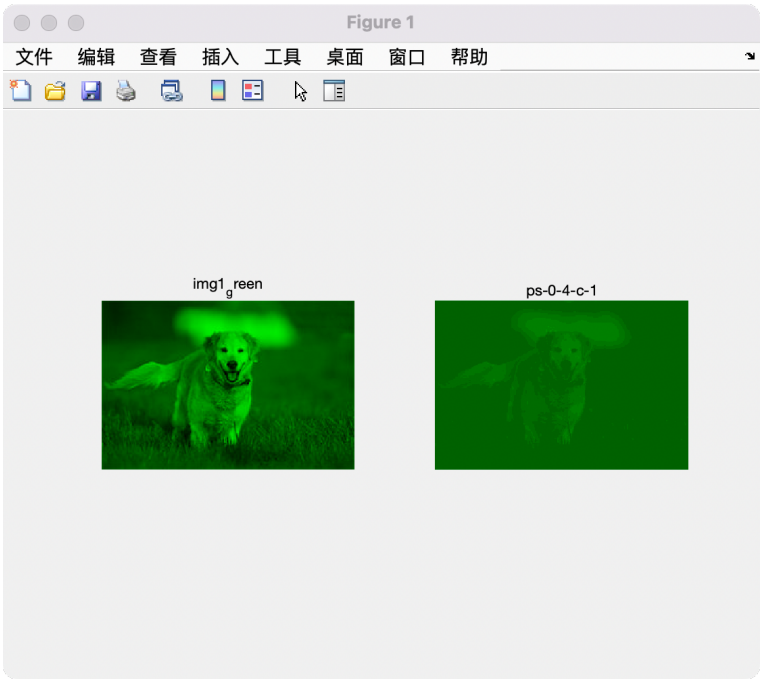
The standard deviation is 40.2460.

```
for i = 1:m
    for j = 1:n
        var = var + (G(i,j) - avg) ^ 2;
    end
end
dev = sqrt(var /(m * n));
```

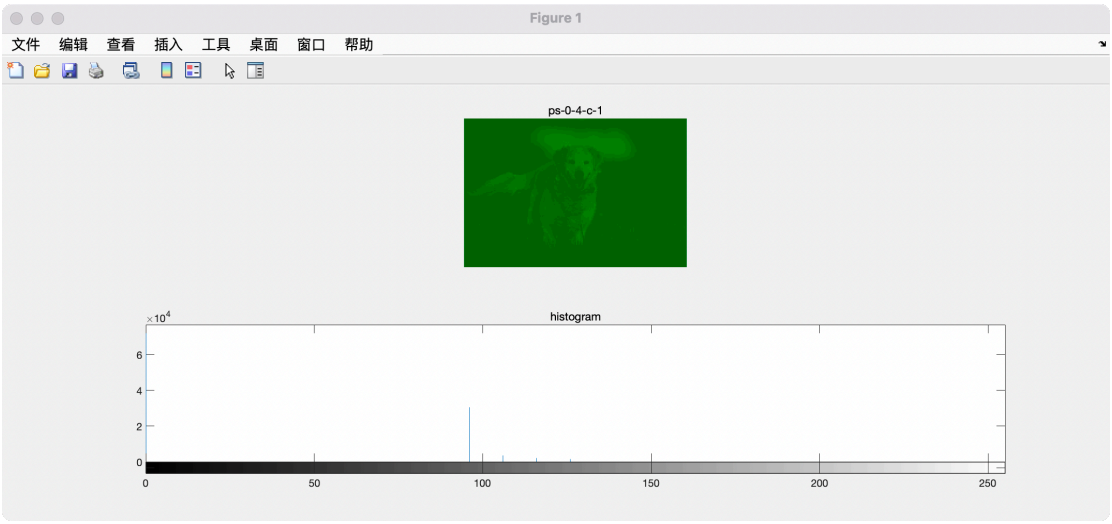
The same, I used two loops to get the deviation value.



ps0-4-b-1



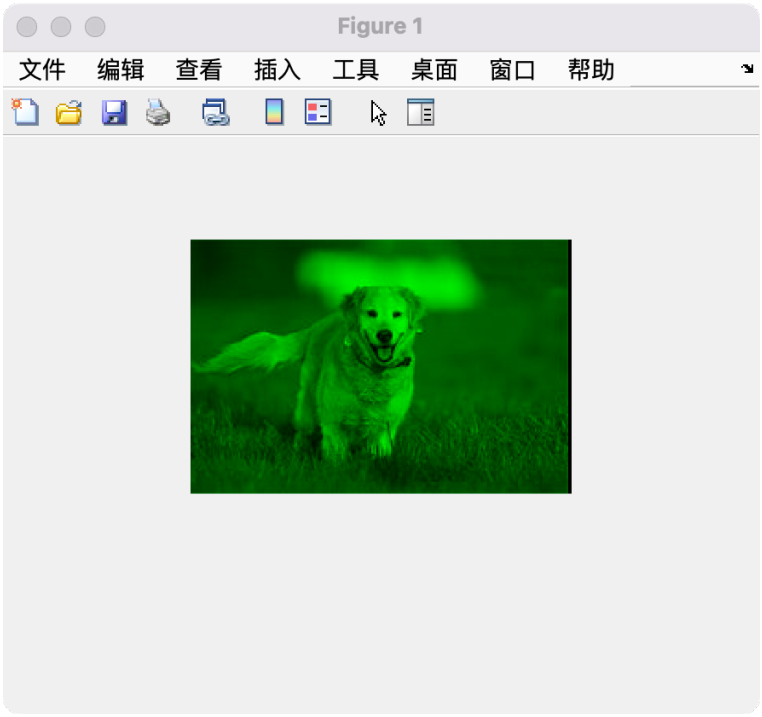
ps0-4-c-1



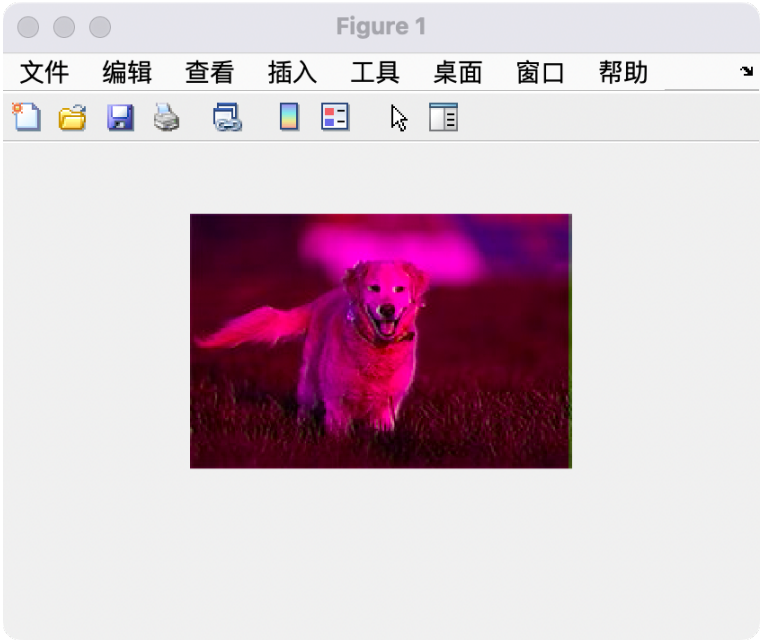
ps0-4-d-1

4-d.

Comparing with the histogram of 4-b, I found that in 4-d, only several value of pixels exist and the object looks less obvious.



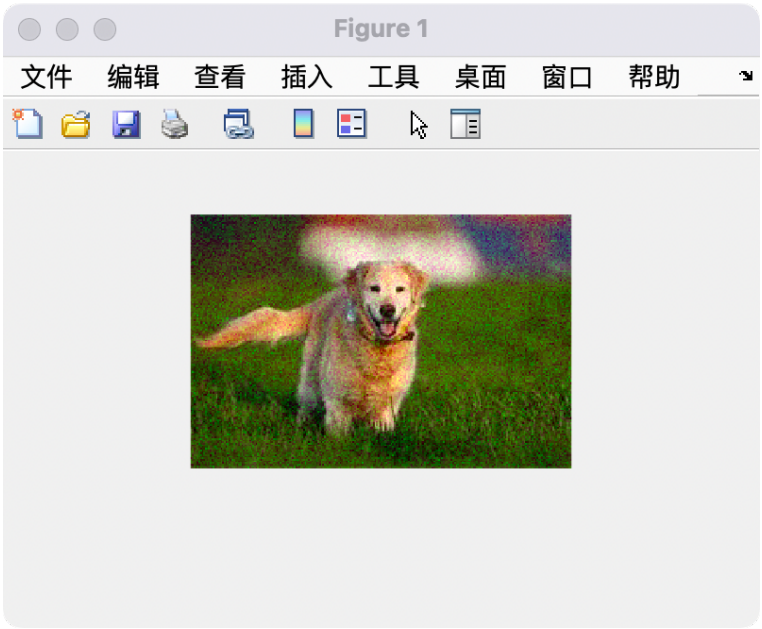
ps0-4-e-1



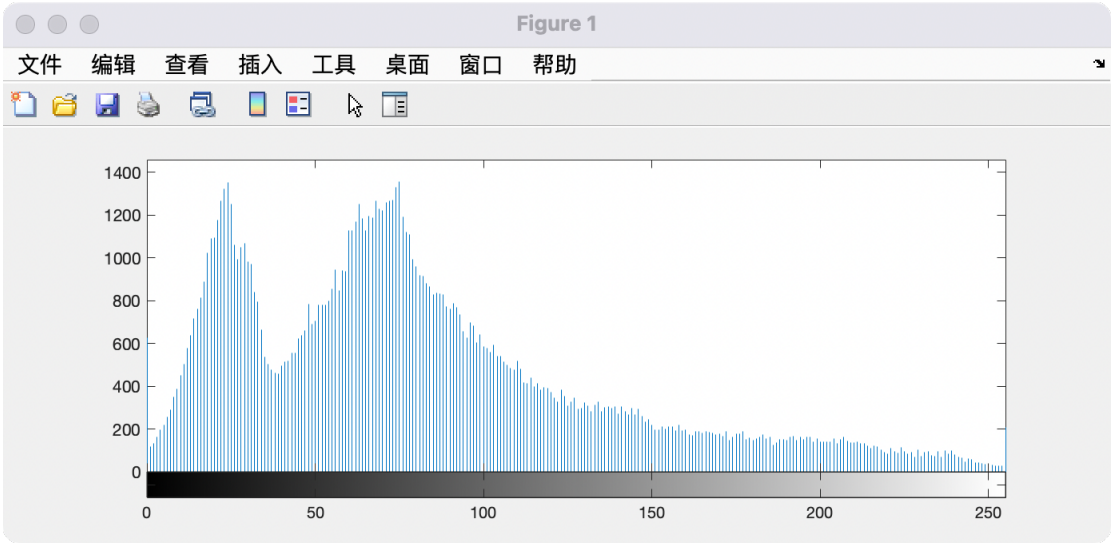
ps0-4-f-1

4-f

I think the negative pixel values can not be detected by human eyes, it looks still like the black.



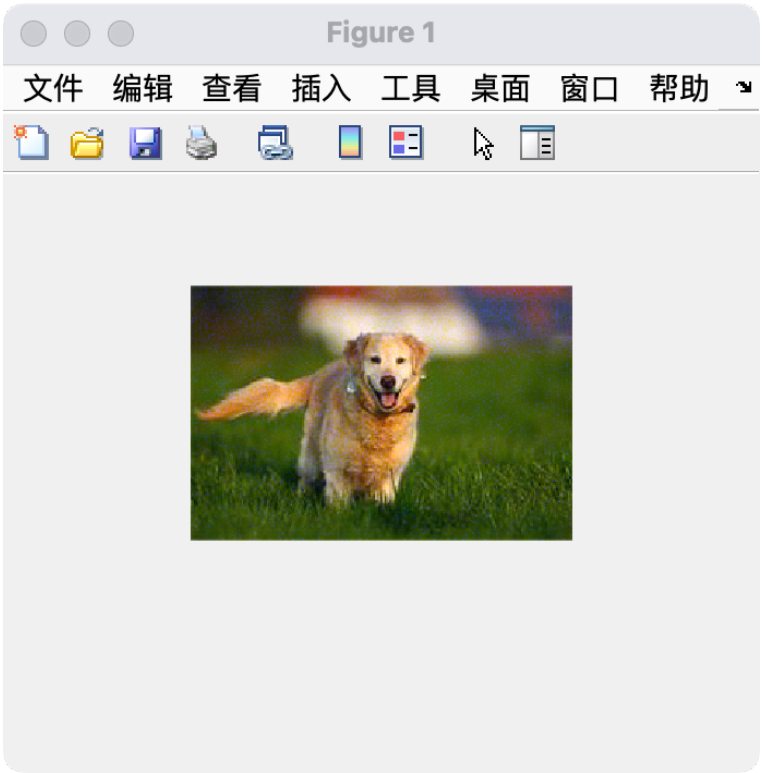
ps0-5-a-1(sigma = 0.01)



ps0-5-b-1

5-b.

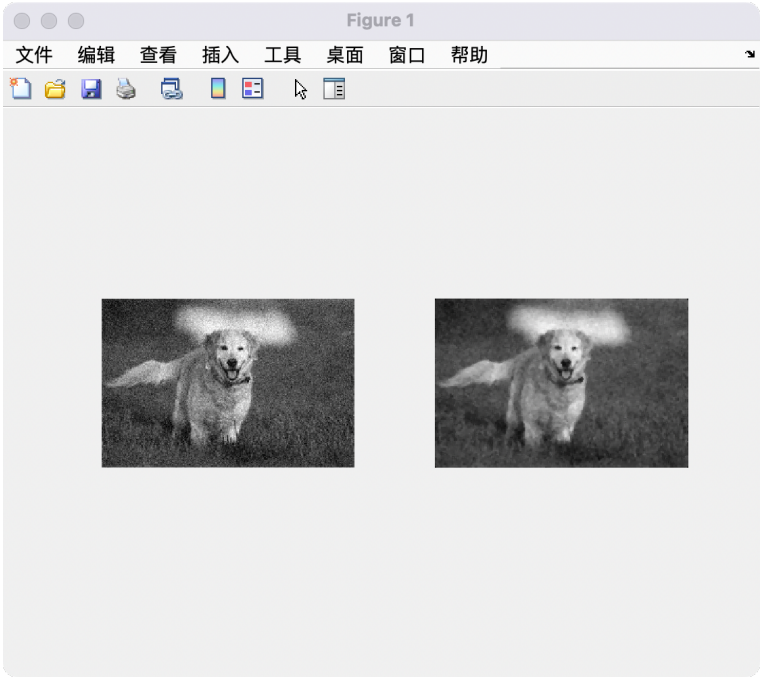
Comparing with 4-b, I think the histogram of 5-b looks denser, almost every value have many pixels from 0 to 255.



ps0-5-c-1

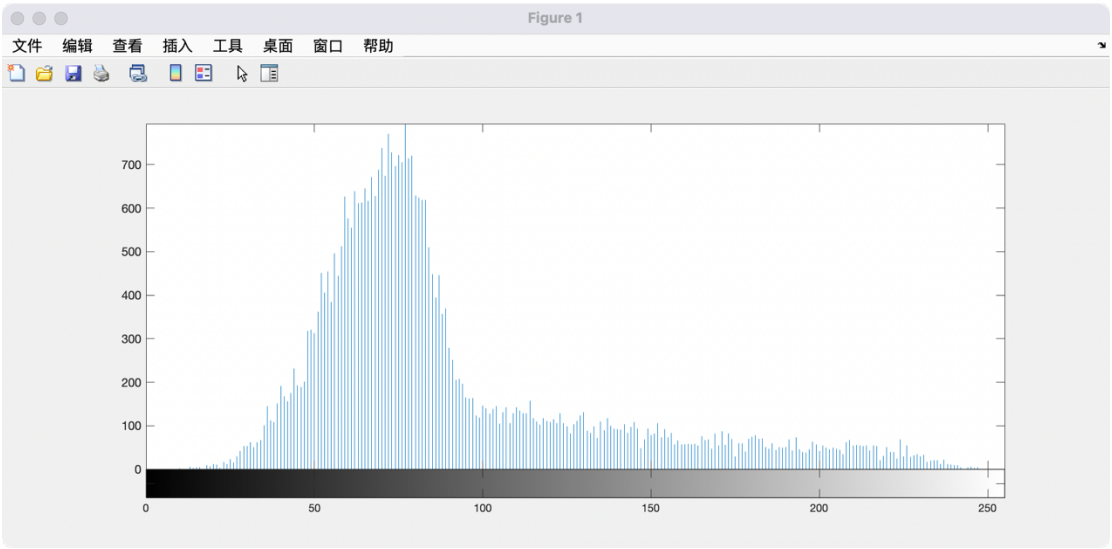
5-d.

I think Gaussian noise add in blue channel looks better, maybe because green is more sensitive for human eyes than blue.



ps0-5-e-1

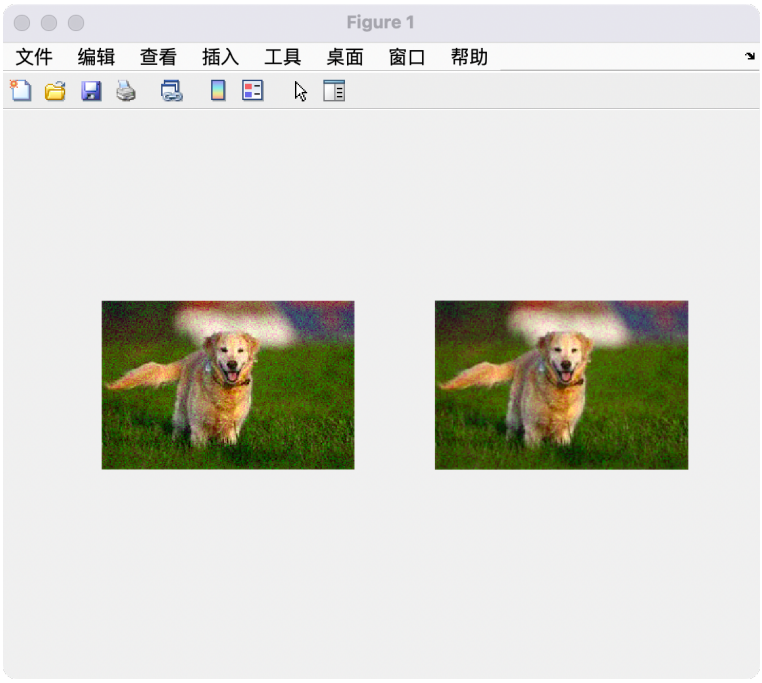
I think the prerequisite of using median filter is that the image should be in 2's matrix, so I turn this image into grey-level matrix.



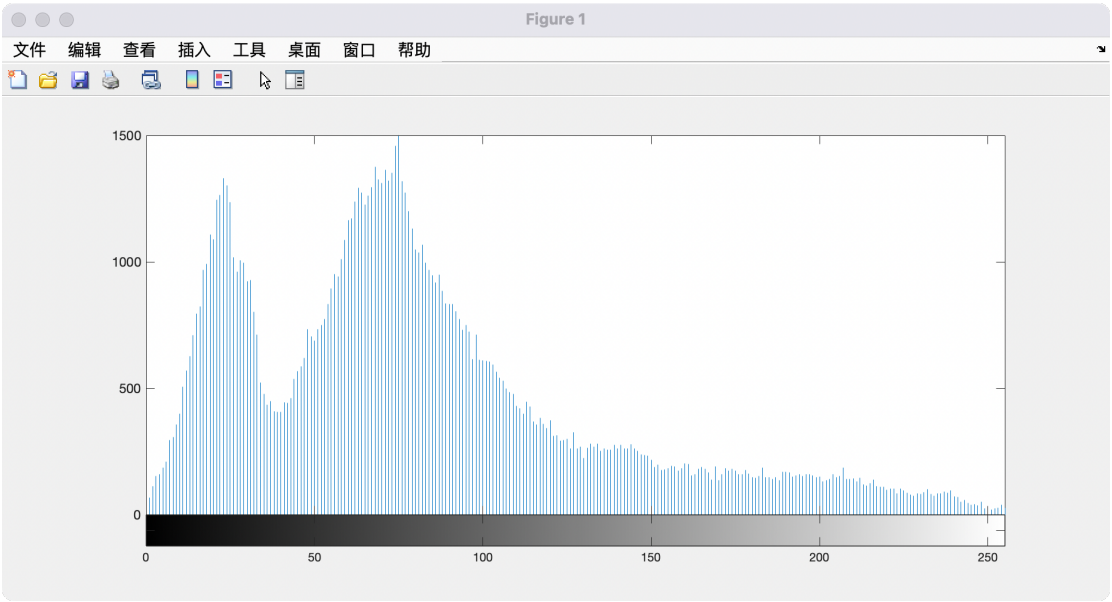
ps0-5-e-2

5-e

Comparing with the noisy image, the image looks more smooth and there are not so many messy points. And from the histogram I can find that the value of pixels seems closer than noisy image.



ps0-5-f-1



ps0-5-f-2

5-f.

Comparing with the noisy image, the image optimize by Gaussian filter seems more smooth, many messy points disappeared, it seems more clearly and I think the histogram looks similar to the histogram of noisy image, actually, I didn't find the big difference and principle of them.

Comparing with the output of median filter. I think the Gaussian filter is better, at least of this image, because I found that the function medfilt2() only can use in the 2's matrix, so I turn the RGB image into grey-level image. However, Gaussian can be the filter of RGB image. Actually, those two ways both will make the image dimmer, but for this image, I think Gaussian filter has a better effect.