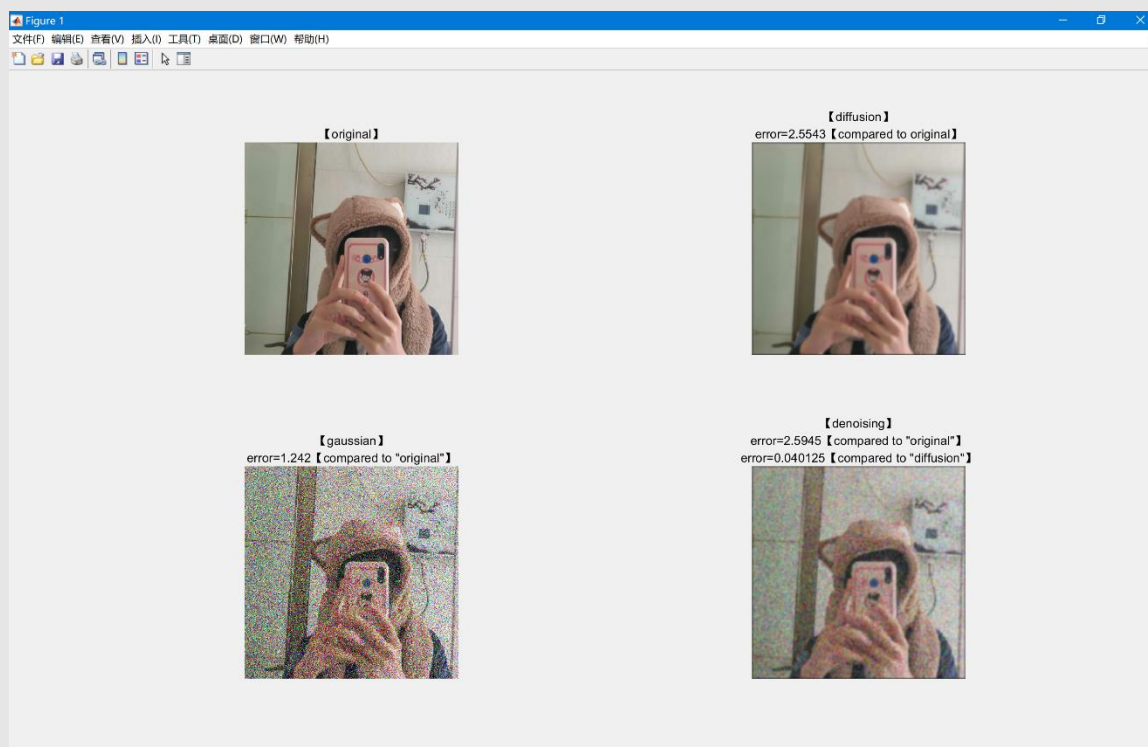


## 【实验目的】

利用热传导方程以及图像各向异性扩散方法，完成图像的denoising与edge preserving

## 【运行结果】



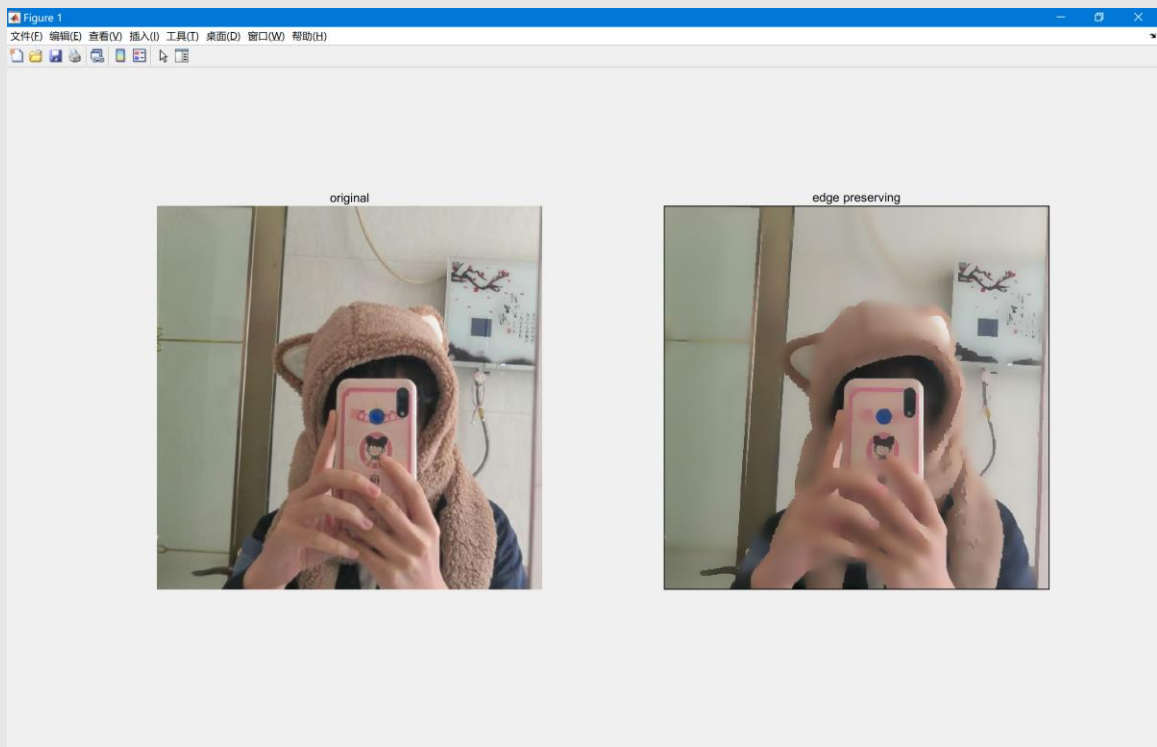
这是对于各向同性扩散的运行结果，左一是原图，左二是原图加上高斯噪声的图，下面有噪声误差

$$error = 1.242$$

右边是进行扩散以后的图，右一是原图直接进行扩散得到的图，效果比较像averaging kernel处理以后的图像，右二是对加了高斯噪声的图进行的扩散，两个扩散的图进行比较得到

$$error = 0.040125$$

也就是说去噪还是起到了一定的效果的，但是图的话就不是那么好看了（两次的error衡量标准都是用当前图与原图差值的二范数与原图自身二范数进行的比值计算）。



对于该图，是各项异性扩散得到的图，它的主要作用是edge preserving，这里已经看见了很明显的效果。

**% 【各向异性】**

clear

clc

lambda=15;

k=0.15;% diffusivity coefficient

N=100;% iterations

img1=imread('quan.jpg');

subplot(121),imshow(uint8(img1)),title('original')

[m,n,o]=size(img1);

img2=zeros(m+2,n+2,o);

img2(2:m+1,2:n+1,1:o)=img1;

imgn=zeros(m+2,n+2,o);

for i=1:N

    for o=1:3

        for p=1:m

            for q=1:n

                NI=img2(p,q+1,o) -  
img2(p+1,q+1,o);

                SI=img2(p+2,q+1,o) -  
img2(p+1,q+1,o);

                EI=img2(p+1,q,o) -  
img2(p+1,q+1,o);

                WI=img2(p+1,q+2,o) -  
img2(p+1,q+1,o);

                cN=exp(double(-  
NI^2/(lambda\*lambda)));

```

            cS=exp(double(-
SI^2/(lambda*lambda)));
            cE=exp(double(-
EI^2/(lambda*lambda)));
            cW=exp(double(-
WI^2/(lambda*lambda)));

imgn(p+1,q+1,o)=img2(p+1,q+1,o)+k*(
cN*NI+cS*SI+cE*EI+cW*WI);
        end
    end
    end
    img2=imgn;
end
subplot(122),imshow(uint8(img2)),ti
tle('edge preserving');

```

% 【各向同性】

clear

clc

lambda=15;

k=0.15;% diffusivity coefficient

N=100;% iterations

img1=imread('quan.jpg');

subplot(121),imshow(uint8(img1)),title('original')

[m,n,o]=size(img1);

img2=zeros(m+2,n+2,o);

img2(2:m+1,2:n+1,1:o)=img1;

imgn=zeros(m+2,n+2,o);

for i=1:N

    for o=1:3

        for p=1:m

            for q=1:n

                NI=img2(p,q+1,o) -  
img2(p+1,q+1,o);

                SI=img2(p+2,q+1,o) -  
img2(p+1,q+1,o);

                EI=img2(p+1,q,o) -  
img2(p+1,q+1,o);

                WI=img2(p+1,q+2,o) -  
img2(p+1,q+1,o);

                cN=exp(double(-  
NI^2/(lambda\*lambda)));

```

            cS=exp(double(-
SI^2/(lambda*lambda)));
            cE=exp(double(-
EI^2/(lambda*lambda)));
            cW=exp(double(-
WI^2/(lambda*lambda)));

imgn(p+1,q+1,o)=img2(p+1,q+1,o)+k*(
cN*NI+cS*SI+cE*EI+cW*WI);
        end
    end
    end
    img2=imgn;
end
subplot(122),imshow(uint8(img2)),ti
tle('edge preserving');

```