

```

clc;clear all;close all;
im=double(imread('Image1.jpg'));
get=inputdlg(["What's the radius?", "What's the variance?"], 'Input');%input
r=str2double(get{1}); % get user input:radius
o=str2double(get{2}); % get user input:variance
% generate convolution kernel using normal distribution
nm=zeros(2*r+1,2*r+1);
for i=-r:r
    for j=-r:r
        nm(i+r+1,j+r+1)=exp(-(i*i+j*j)/(2*o*o))/(2*pi*o*o);
    end
end
nm=nm./sum(nm,'all');% remain the same brightness,let the sum of kernel be 1
res=imfilter(im,nm,'replicate');
figure,imshow(uint8(res));

```



Start



```
graph TD; Start([Start]) --> Read[/read value of r, σ from user/]; Read --> Generate[generate convolution kernel using normal distribution]; Generate --> Blur[do image blur]; Blur --> Output[/output result image/]; Output --> End([End])
```

read value of  
 $r, \sigma$  from user

generate convolution kernel  
using normal distribution

do image blur

output result  
image

End

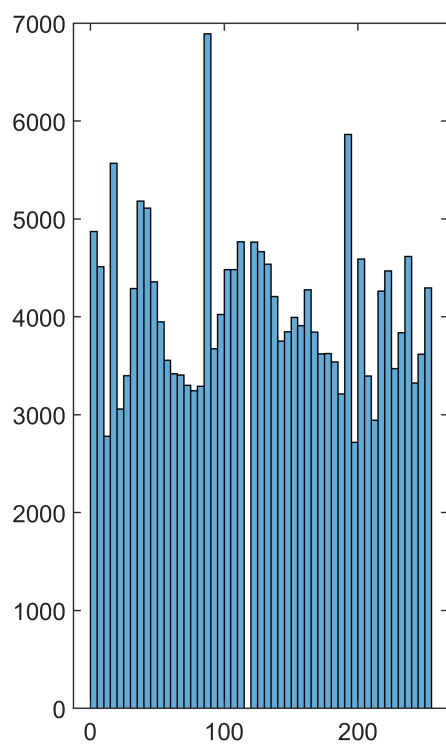
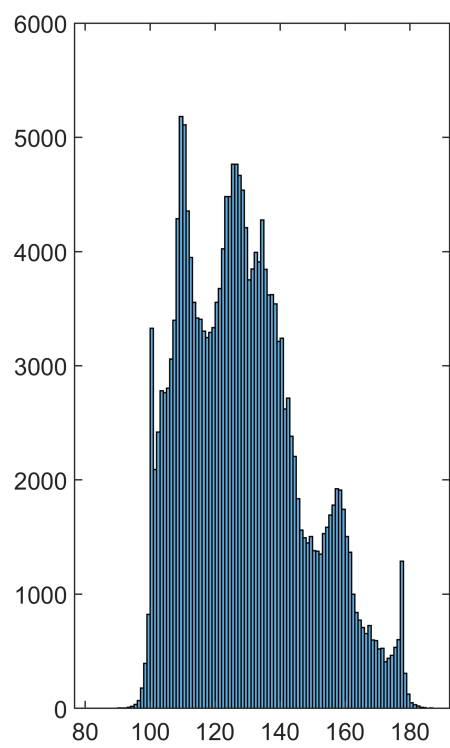
```

clc;clear all;close all;
im=double(imread('Image2.jpg'));
hist=figure;
[m,n,trash]=size(im);
R=im(:,:,1);
G=im(:,:,2);
B=im(:,:,3);
%%% RGB to YUV!
Y=0.299.*R+0.587.*G+0.114.*B;
U=-0.147.*R-0.289.*G+0.436.*B;
V=0.615.*R-0.515.*G-0.100.*B;
subplot(1,2,1);
histogram(uint8(Y));
%%% count!
count=zeros(1,256);
for i=1:m
    for j=1:n
        count(1,uint8(Y(i,j))+1)=count(1,uint8(Y(i,j))+1)+1;
    end
end
for i=1:256
    counter(1,i)=count(1,i)/(m*n);
end
%%% Equalize!
now=0;
map=zeros(1,256);
for i=1:256
    now=now+counter(1,i);
    map(1,i)=round(255*now);
end
%%% generate!
for i=1:m
    for j=1:n
        Y(i,j)=map(uint8(Y(i,j)));
    end
end
%%% YUV to RGB!
r=Y+1.14.*V;
g=Y-0.39.*U-0.58.*V;
b=Y+2.03.*U;
new(:,:,1)=r;
new(:,:,2)=g;
new(:,:,3)=b;
%%% output!
img=figure;
figure(img);
subplot(1,2,1);
imshow(uint8(im));
subplot(1,2,2);
imshow(uint8(new));

```



```
figure(hist);  
subplot(1,2,2);  
histogram(uint8(Y));
```



```

clc;clear all;close all;
I = double(imread('Image3.jpeg'));%error1:not converted to double
[M,N,L] = size(I);

R = I(:,:,1);
G = I(:,:,2);
B = I(:,:,3);

J(1:2:M,1:2:N) = R(1:2:M,1:2:N);
J(2:2:M,2:2:N) = B(2:2:M,2:2:N);
J(1:2:M,2:2:N) = G(1:2:M,2:2:N); % error 2:the value of R/B channel may also be 0
J(2:2:M,1:2:N) = G(2:2:M,1:2:N); % error 2:the value of R/B channel may also be 0

figure,imshow(uint8(J),[]);

```



```

T = zeros(M,N,3);
for i = 2:M-1%%%% error3:out of range
    for j = 2:N-1%% error3:out of range
        if mod(i,2) == 0 && mod(j,2) == 1
            T(i,j,1)=round((J(i-1,j)+J(i+1,j))/2);
            T(i,j,2)=round(J(i,j));
            T(i,j,3)=round((J(i,j-1)+J(i,j+1))/2);

```



```

elseif mod(i,2) == 1 && mod(j,2)==1 % error4:improper'elseif'condition
    T(i,j,1)=round(J(i,j));
    T(i,j,2)=round((J(i-1,j)+J(i+1,j)+J(i,j-1)+J(i,j+1))/4);
    T(i,j,3)=round((J(i-1,j-1)+J(i+1,j+1)+J(i+1,j-1)+J(i-1,j+1))/4);
    %%%%%%%%%%%%%%% error5:typo?-'shoule be'+
elseif mod(i,2) == 1 && mod(j,2) == 0 % error4:improper'elseif'condition
    T(i,j,1)=round((J(i,j-1)+J(i,j+1))/2);
    T(i,j,2)=round(J(i,j));
    T(i,j,3)=round((J(i-1,j)+J(i+1,j))/2);
else
    T(i,j,1)=round((J(i-1,j-1)+J(i+1,j+1)+J(i+1,j-1)+J(i-1,j+1))/4);
    %%%%%%%%%%%%%%% error5:typo?-'shoule be'+
    T(i,j,2)=round((J(i-1,j)+J(i+1,j)+J(i,j-1)+J(i,j+1))/4);
    T(i,j,3)=round(J(i,j));
end
end
end
figure,imshow(uint8(T),[]);

```

