

فرمت تهیه گزارش

علی نصیری سروی

اطلاعات گزارش	چکیده
تاریخ:	
واژگان کلیدی:	
درون یابی	
تبدیل هندسی	
کوانتیزیشن	
registration	

1-مقدمه

1.1.2. برای حل این مسئله از شرح مسئله قبل استفاده میکنیم. بدین صورت که نقاط feature point را از منطقه overlap انتخاب میکنیم. سپس در تصویر نهایی از 0 تا شروع منطقه overlap را از تصویر اول برمیداریم. سپس باقی تصویر را از تصویر دوم به کمک نگاشت بدست آمده و درون یابی دوخطی محاسبه میکنیم.

2-شرح تکنیکال

1.1.1. برای حل این مسئله تابع نگاشت را به صورت زیر تعریف میکنیم:

اگر (x, y) مختصات یک feature point در تصویر A بوده و (v, w) متناظر آن در تصویر B باشد، آنگاه خواهیم داشت:

$$x = c1*v + c2*w + c3*v*w + c4$$

$$y = c5*v + c6*w + c7*v*w + c8$$

لازم است که ضرایب $c1, \dots, c8$ را بدست

آوریم. یک دستگاه با 8 مجهول داریم و نیازمند 8

معادله مستقل برای حل آنها میباشیم. با 4

feature point میتوان این ضرایب را بدست

آورد. حال برای تبدیل تصویر B به تصویر A

میتوان از این تابع برای تمام نقاط B استفاده

کرد. در صورتی که نگاشت از A به B خواستیم

باید دستگاه زیر را حل کنیم تا نگاشت بدست آید:

$$v = c1*x + c2*y + c3*x*y + c4$$

$$w = c5*x + c6*y + c7*x*y + c8$$

در صورتی که تعداد نقاطمان بیشتر از 4 تا بود

میتوان نگاشت پیچیده تری را برای تبدیل تعریف

کرد زیرا معادلات بیشتری داریم و میتوانیم

نگاشت پیچیده تری را بدست آوریم.

1.1.3. تبدیل های هندسی توابعی برای پردازش تصویر

بر حسب مکان میباشند. این تبدیل ها خطوط

موازی را حفظ میکنند اما لزوما زوایا را حفظ

نخواهند کرد. اگر تبدیل هندسی T را داشته

باشیم، دو نقطه (x, y) و (v, w) به صورت زیر

به هم تبدیل خواهند شد.

$$[x \ y \ 1]^T = [v \ w \ 1]^T T$$

تبدیل هندسی rotation به صورت زیر تعریف میشود.

$$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

1.2.1. در این مسئله نکته ای که مورد توجه است این است که تاثیر histeq بر روی سطوح خاکستری چگونه است.

1.2.2. در این مسئله ابتدا تصویر را به کمک دو روش averaging و remove row&col ، downsampling میکنیم و سپس به کمک درون یابی دو خطی و تکرار پیکسل ها آنرا به سائز اصلی بر میگردانیم.

در بازگرداندن به کمک تکرار پیکسل ها هر پیکسل سه بار برای پیکسل های مجهول مجاورش تکرار میشود.

در بازگرداندن به کمک درون یابی دو خطی به ازای هر 4 پیکسل 3 تای آنها مجهول است که در نتیجه یک مجموعه 16 تایی آنها را که شامل 4 معلوم میباشد و برای 12 تای مجهول میتوان از آنها استفاده کرد، استفاده کرده ام.



- نتیجه نهایی.

- نتیجه چرخش:

تصویر اصلی که چرخش بر روی آن انجام شده است.



2-شرح نتایج

- نتیجه تصویر panorama:

دو تصویری که stitching بر روی آنها انجام شده است:



-تصویر اول



-تصویر دوم



- چرخش ۴۵ درجه به کمک درون یابی نزدیک ترین
همسایه



- چرخش 30 درجه به کمک درون یابی نزدیک
ترین همسایه



- چرخش ۴۵ درجه به کمک درون یابی دو خطی

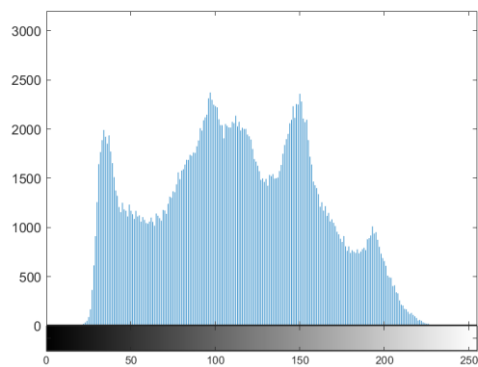


- چرخش 30 درجه به کمک درون یابی دوخطی

-نتیجه histogram:



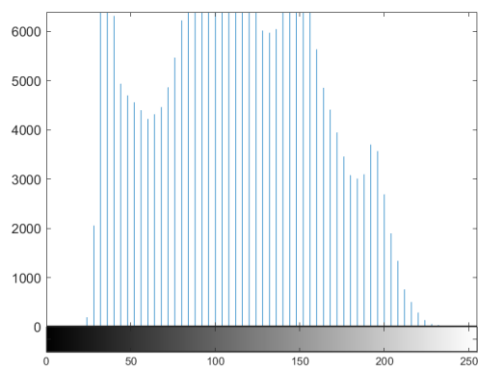
- چرخش ۸۰ درجه به کمک درون یابی
نزدیک ترین همسایه



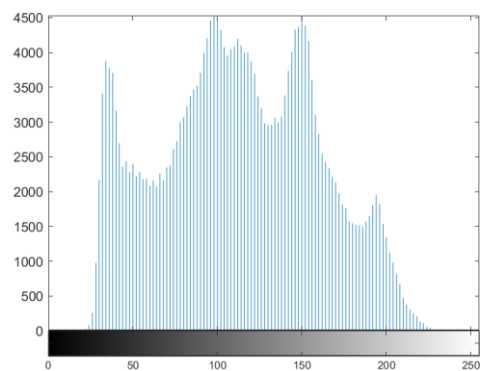
- تصویر اصلی و hist در حالت خاکستری



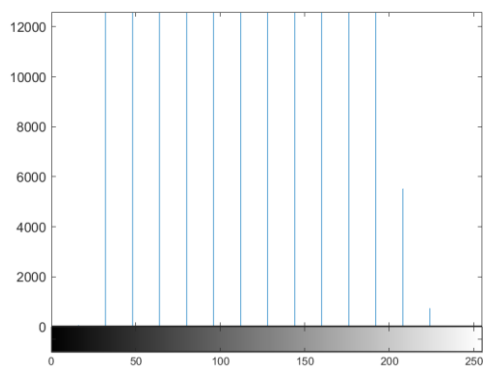
- چرخش ۸۰ درجه به کمک درون یابی دوخطی



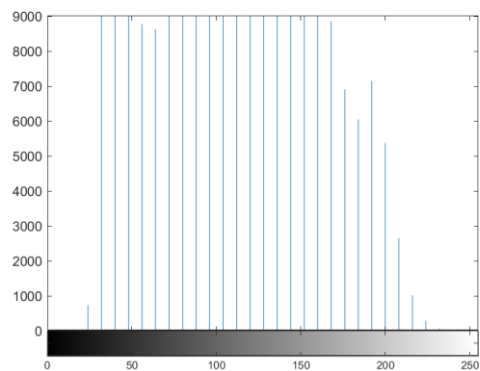
- تصویر با 64 لول خاکستری و hist آن بدون histeq



- تصویر با 128 لول خاکستری و hist آن بدون histeq

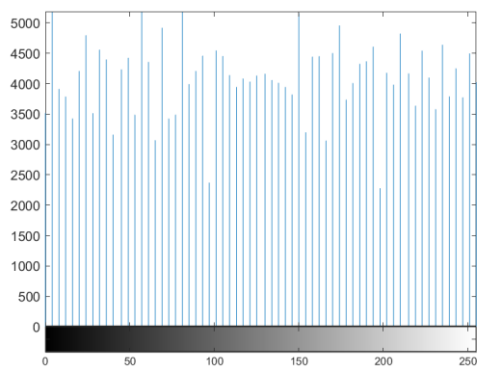


- تصویر با 16 لول خاکستری و hist آن بدون histeq

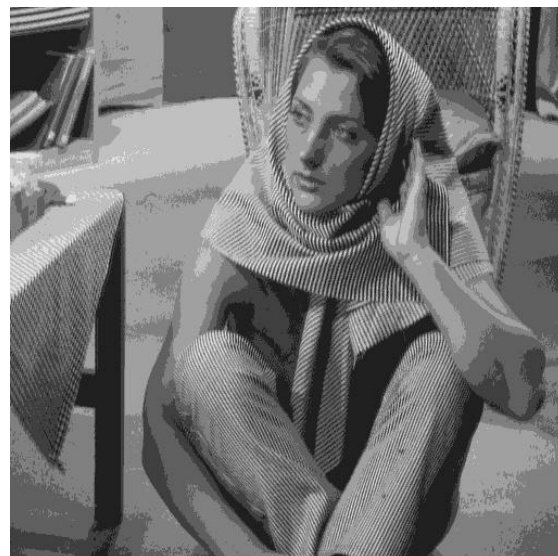


- تصویر با 32 لول خاکستری و hist آن بدون histeq

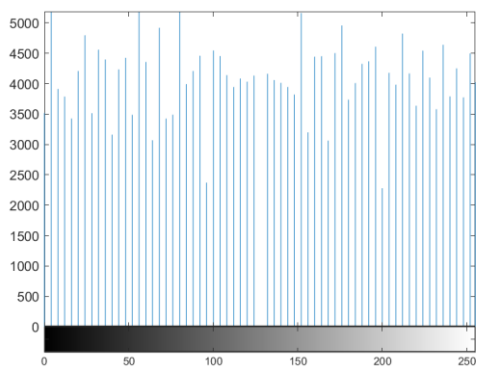
نتایج بعد از histeq:



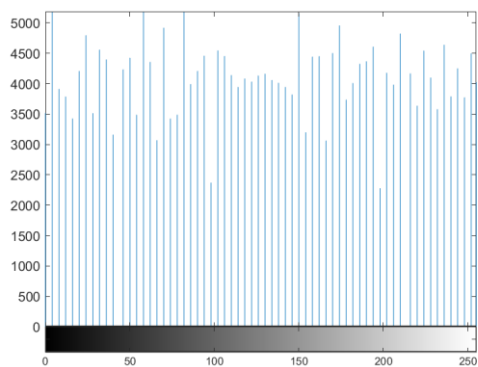
- تصویر اصلی hist آن بعد از histeq



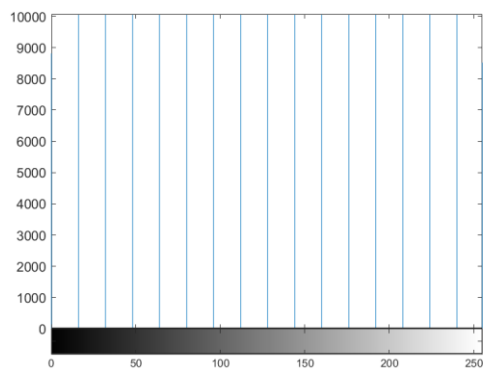
- تصویر با 8 لول خاکستری و hist آن بدون histeq



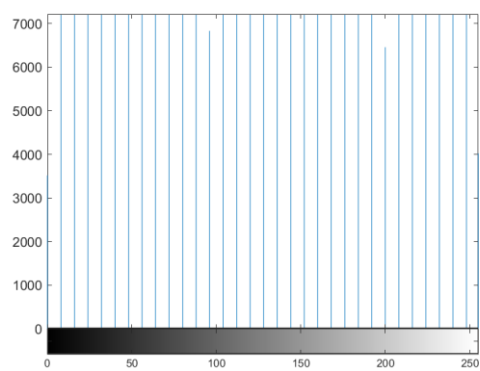
- تصویر با 64 لول خاکستری و hist آن بعد از histeq



- تصویر با 128 لول خاکستری و hist آن بعد از histeq



- تصویر با 16 لول خاکستری و hist آن بعد از histeq



- تصویر با 32 لول خاکستری و hist آن بعد از histeq

-نتایج نهایی interpolation:



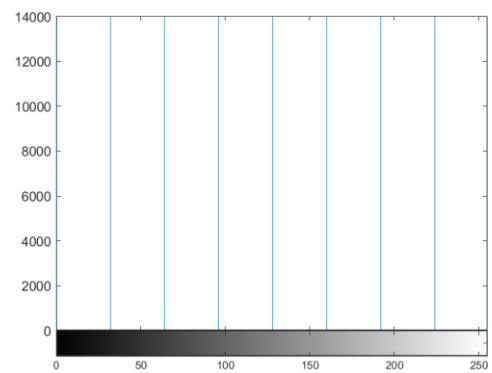
-تصویر اصلی.



-تصویر averaging و bilinear.



-تصویر remove row_col و bilinear.



- تصویر با 8 لول خاکستری و hist آن بعد از histeq

نتایج نهایی:

Report MSE

level	8	16	32	64	128
Without histeq	83.3198	21.0897	5.4744	1.5051	0.5001
With histeq	83.9071	20.6982	5.1379	1.7999	0.4862

کدها:

1.1.2:

لود کردن ورودی:

```
car1_image =  
imread("HW/Homeworks/Images/1/car1.jpg");  
car2_image =  
imread("HW/Homeworks/Images/1/car2.jpg");  
  
width = 1430;  
output =  
zeros(750,width,3,'uint8')  
;  
overlap_start_im1 = 433;  
  
for i=1:750  
    for  
j=1:overlap_start_im1  
        output(i,j,:) =  
car1_image(i,j,:);  
        end  
end
```

تعیین نقاط و حل دستگاه:

```
car1_image =  
imread("HW/Homeworks/Images/1/car1.jpg");  
car2_image =  
imread("HW/Homeworks/Images/1/car2.jpg");  
  
width = 1430;  
output =  
zeros(750,width,3,'uint8')  
;  
overlap_start_im1 = 433;  
  
for i=1:750  
    for  
j=1:overlap_start_im1  
        output(i,j,:) =  
car1_image(i,j,:);  
        end
```



-تصویر pixel replication و averaging.



- تصویر pixel replication و remove row_col.

نتایج نهایی:

	Pixel Replication	Bilinear Interpolation
Averaging	173.3544	264.7733
Remove Row&Column	133.0759	97.7735

```

end
max_x = shape(1);
max_y = shape(2);
if(floor_x>max_x)
    floor_x = max_x;
end
if(floor_y>max_y)
    floor_y=max_y;
end
if(ceil_x>max_x)
    ceil_x=max_x;
end
if(ceil_y>max_y)
    ceil_y=max_y;
end

if(floor_x==x &&
floor_y ==y)
    val = image(x,y);
    return;
end

dx1 = x-floor_x;
dx2 = 1-dx1;
dy1 = y-floor_y;
dy2 = 1-dy1;

v =
image(floor_x,floor_y,:)*d
x2*dy2 ;
v = v +
image(floor_x,ceil_y,:)*dx
2*dy1;
v = v +
image(ceil_x,floor_y,:)*dx
1*dy2;
v = v +
image(ceil_x,ceil_y,:)*dx1
*dy1;
val = v;
end

```

end

انتقال پیکسل های ما بعد اورلپ به تصویر نهایی:

```

for i=1:750
    for
j=overlap_start_im1:width
        v = i;
        w = j-
overlap_start_im1+1;
        z = v*w;
        x = [c1 c2 c3
c4]*[v; w; z; 1];
        y = [c5 c6 c7
c8]*[v; w; z; 1];

        [t,s] =
        bilin_inter
        (x,y,car2_image);
        output(i,j,:) =
car2_image(t,s,:);

    end
end

```

تابع درون یابی دو خطی:

```

function val =
bilin_inter(x,y,image)

    shape = size(image);

    floor_x = floor(x);
    ceil_x = ceil(x);
    floor_y = floor(y);
    ceil_y = ceil(y);

    if(floor_x<=0)
        floor_x=1;
    end
    if(floor_y<=0)
        floor_y=1;
    end
    if(ceil_x<=0)
        ceil_x=1;
    end
    if(ceil_y<=0)
        ceil_y=1;
    end

```

:1.1.3

لود کردن داده و ایجاد تصویر pad شده:

```
        v =
        bilin_inter(i,j,pad_im,rotate30);

        image_bilin_inter(i,j)= v;
    end
end
```

درون یابی دو خطی:

```
function val =
    bilin_inter(i,j,image,transform_matrix)

    shape = size(image);
    X =
    transform_matrix\[i-
    shape(1)/2;j-
    shape(2)/2;1];
    x = X(1)+shape(1)/2;
    y = X(2)+shape(2)/2;

    floor_x = floor(x);
    ceil_x = ceil(x);
    floor_y = floor(y);
    ceil_y = ceil(y);

    if(floor_x<=0)
        floor_x=1;
    end
    if(floor_y<=0)
        floor_y=1;
    end
    if(ceil_x<=0)
        ceil_x=1;
    end
    if(ceil_y<=0)
        ceil_y=1;
    end
    max_x = shape(1);
    max_y = shape(2);
    if(floor_x>max_x)
        floor_x = max_x;
    end
    if(floor_y>max_y)
        floor_y=max_y;
    end
```

```
I =
imread("HW/Homeworks/Images/1/Elaine.bmp");
th=80*pi/180;
rotate30 = [cos(th)
sin(th) 0; -sin(th)
cos(th) 0;0 0 1];
imshow(I);
A = size(I);
%output_raw =
ones(A(1),A(2),'uint8');
pad_im =
zeros(A(1)*1.5,A(2)*1.5,'uint8');
for i=1:A(1)
    for j=1:A(2)

        pad_im(A(1)/4+i,A(2)/4+j)=
        I(i,j);
    end
end
```

چرخش تصویر با درون یابی ها :

```
image_near_inter = pad_im;
for i = 1:A(1)
    for j = 1:A(1)
        [v,w] =
        nearest_neighbor_inter(i,j
        ,pad_im,rotate30);

        image_near_inter(i,j)=
        pad_im(v,w);
    end
end

image_bilin_inter =
pad_im;

for i = 1:A(1)
    for j = 1:A(1)
```



```

shape(1)/2;j-
shape(2)/2;1];
    x = X(1)+shape(1)/2;
    y = X(2)+shape(2)/2;
    floor_x = floor(x);
    ceil_x = ceil(x);
    floor_y = floor(y);
    ceil_y = ceil(y);

    if(abs(floor_x-x) <
abs(ceil_x-x))
        x = floor(x);
    else
        x = ceil(x);
    end
    if(abs(floor_y-y) <
abs(ceil_y-y))
        y = floor(y);
    else
        y = ceil(y);
    end
    if(x>shape(1))
        x= shape(1);
    end
    if(x<=0)
        x=1;
    end
    if(y>shape(2))
        y= shape(2);
    end
    if(y<=0)
        y=1;
    end
end
end

```

```

    if(ceil_x>max_x)
        ceil_x=max_x;
    end
    if(ceil_y>max_y)
        ceil_y=max_y;
    end

    if(floor_x==x &&
floor_y ==y)
        val = image(x,y);
        return;
    end

    dx1 = x-floor_x;
    dx2 = 1-dx1;
    dy1 = y-floor_y;
    dy2 = 1-dy1;

    v =
image(floor_x,floor_y)*dx2
*dy2 ;
    v = v +
image(floor_x,ceil_y)*dx2*
dy1;
    v = v +
image(ceil_x,floor_y)*dx1*
dy2;
    v = v +
image(ceil_x,ceil_y)*dx1*d
y1;
    val = v;
end

```

درون یابی نزدیک ترین همسایه:

```

function [x,y] =
nearest_neighbor_inter(i,j
,image,transform_matrix)

    shape = size(image);
    X =
transform_matrix\[i-

```

1.2.1

لود کردن داده ها:

```
image =  
imread("HW/Homeworks/Images/1/Barbara.bmp");  
I = rgb2gray(image);
```

کاهش سطوح خاکستری بدون histeq:

```
x = I;  
level8 = 32*round(x/32);  
level16 = 16*round(x/16);  
level32 = 8*round(x/8);  
level64 = 4*round(x/4);  
level128 = 2*round(x/2);  
figure  
subplot(2,3,1)  
imshow(level8);  
title('immse(level8,x)');  
subplot(2,3,2)  
imshow(level16);  
title('immse(level16,x)');  
subplot(2,3,3)  
imshow(level32);  
title('immse(level32,x)');  
subplot(2,3,4)  
imshow(level64);  
title('immse(level64,x)');  
subplot(2,3,5)  
imshow(level128);  
title('immse(level128,x)');  
subplot(2,3,6)  
imshow(x);  
title('immse(x,I)');
```

کاهش سطوح خاکستری بعد از histeq:

```
x = histeq(x);  
  
level8 = 32*round(x/32);  
level16 = 16*round(x/16);  
level32 = 8*round(x/8);  
level64 = 4*round(x/4);  
level128 = 2*round(x/2);  
  
figure  
subplot(2,3,1)
```

1.2.2

لود کردن تصویر

```
I =  
imread("HW/Homeworks/Images/1/Goldhill.bmp");
```

تابع درون یابی دوخطی:

```
function out = bilin(I)  
    shape = size(I);  
    out =  
zeros(2*shape(1),2*shape(2)), 'uint8');  
    for i=1:shape(1)  
        for j=1:shape(2)  
            s = i*2-1;  
            t = j*2-1;  
  
out(s,t)=I(i,j);  
        end  
    end  
  
    for i=1:shape(1)*2  
        for j=1:shape(2)*2  
            if(mod(i,2)==0  
||mod(j,2)==0)
```

```

        x3 = x+2;
        y3 = y-1;

        x4 = x+2;
        y4 = y+1;
    end
    X = [x1 x2 x3 x4];
    Y = [y1 y2 y3 y4];

    for i=1:4
        if(X(i)>max_x)
            X(i) = max_x;
        end
        if(X(i)<=0)
            X(i)=1;
        end
    end

    for i=1:4
        if(Y(i)>max_y)
            Y(i) = max_y;
        end
        if(Y(i)<=0)
            Y(i)=1;
        end
    end

    x1 = X(1);
    x2 = X(2);
    x3 = X(3);
    x4 = X(4);

    y1 = Y(1);
    y2 = Y(2);
    y3 = Y(3);
    y4 = Y(4);

    v = image(x1,y1,:)/4 ;
    v = v +
image(x2,y2,:)/4;
    v = v +
image(x3,y3,:)/4;
    v = v +
image(x4,y4,:)/4;

    val = v;
end

```

```

        out(i,j) =
        bilin_inter(i,j,out);
    end

    end

end

```

تابع کمکی درون یابی دو خطی:

```

function val =
    bilin_inter(x,y,image)

    shape = size(image);
    max_x = shape(1);
    max_y = shape(2);
    if(mod(x,2)==0 &&
mod(y,2)==0)
        x1 = x-1;
        y1 = y-1;

        x2 = x-1;
        y2 = y+1;

        x3 = x+1;
        y3 = y-1;

        x4 = x+1;
        y4 = y+1;

    elseif(mod(x,2)==0)
        x1 = x-1;
        y1 = y;

        x2 = x-1;
        y2 = y+2;

        x3 = x+1;
        y3 = y;

        x4 = x+1;
        y4 = y+2;
    elseif(mod(y,2)==0)
        x1 = x;
        y1 = y-1;

        x2 = x;
        y2 = y+1;
    end

```

```

        for j
=1:shape(2)/2
            out(i,j) =
I(2*i-1,2*j-1);
        end
    end
end

```

تابع averaging:

```

function out=average(I)
shape = size(I);
out =
zeros(shape(1)/2,shape(2)/
2,'uint8');
for i=1:shape(1)/2-1
    for j=1:shape(2)/2-1
        t = i*2-1;
        s = j*2-1;
        X =
I(t:t+2,s:s+2);
        sum = X(1,1)/9;
        sum =sum +
X(1,2)/9;
        sum =sum +
X(1,3)/9;
        sum =sum +
X(2,1)/9;
        sum =sum +
X(2,2)/9;
        sum =sum +
X(2,3)/9;
        sum =sum +
X(3,1)/9;
        sum =sum +
X(3,2)/9;
        sum =sum +
X(3,3)/9;
        out(i,j) = sum;
    end
end
end

```

تابع pixel_rep:

```

function out =
pixel_rep(I)
    shape = size(I);
    out =
zeros(2*shape(1),2*shape(2)
),'uint8');
    for i=1:shape(1)
        for j=1:shape(2)
            s = i*2-1;
            t = j*2-1;

out(s,t)=I(i,j);

out(s+1,t)=I(i,j);

out(s,t+1)=I(i,j);

out(s+1,t+1)=I(i,j);

        end
    end
end
function out = row_col(I)
    shape = size(I);
    out =
zeros(shape(1)/2,shape(2)/
2,'uint8');

    for i=1:shape(1)/2
        for j
=1:shape(2)/2
            out(i,j) =
I(2*i-1,2*j-1);
        end
    end
end

```

تابع remove row&col:

```

function out = row_col(I)
    shape = size(I);
    out =
zeros(shape(1)/2,shape(2)/
2,'uint8');

    for i=1:shape(1)/2

```

ترکیب این 4 تابع اصلی 4 تصویر مورد نظر ما را
میدهند.