

آیتمهای تحویلی پروژه اصول سنجش از دور (2-99-1398)

نام و نام خانوادگی دانشجو: حسن رضوان

کمپایلر یا محیط برنامه نویسی: MATLAB

الف) عنوان آیتم شماره ۱ (نشان دادن هر یک از باندها در صفحه مانیتور بطور جداگانه).
ابتدا به کمک دستور های fread و fopen ماتریس تصویر را به صورت بردار درآورده و سپس یک ماتریس ۲۰۴۸*۲۰۴۸ تشکیل می دهیم. به کمک دستور switch case کاربر باند مورد نظر را انتخاب کرده و به کمک imshow آنرا نمایش می دهیم.

ب) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

while 1

claim=input('Plase Enter the band_num you prefer: ');    % The number band
that User wants
if claim~=1:7 | claim==0
    disp('ERROR! Your input is undefined. ')
    return
end

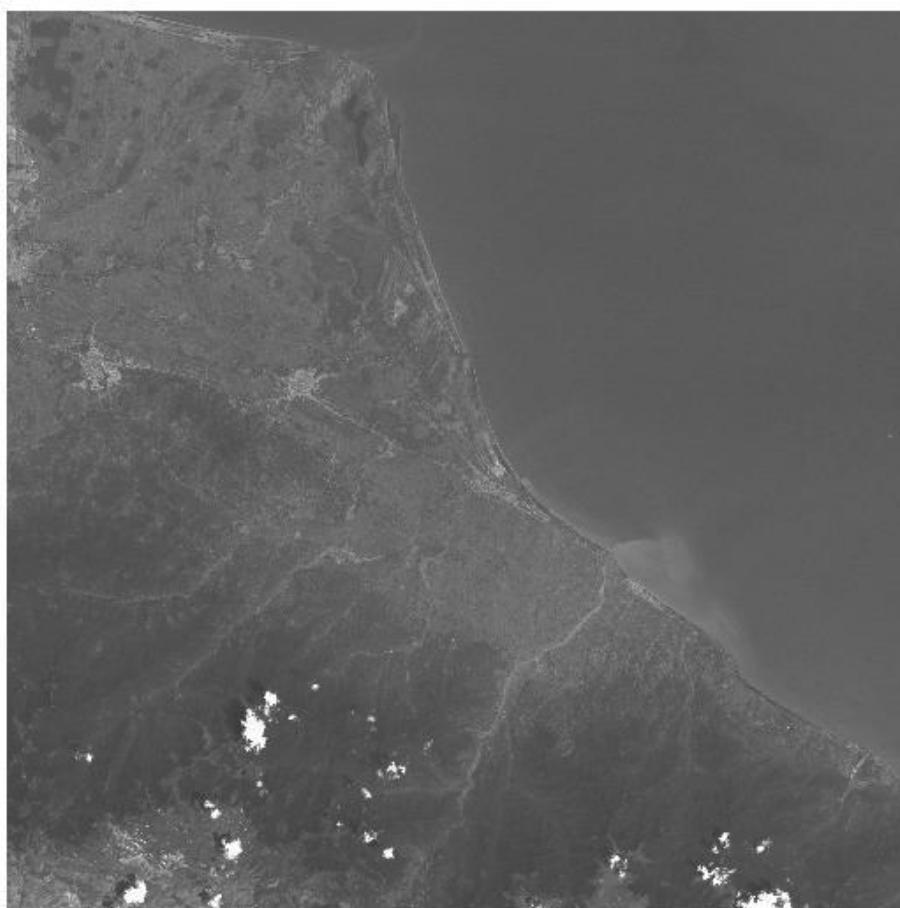
switch claim
    case 1
        fileID=fopen('b1.dat','r');
    case 2
        fileID=fopen('b2.dat','r');
    case 3
        fileID=fopen('b3.dat','r');
    case 4
        fileID=fopen('b4.dat','r');
    case 5
        fileID=fopen('b5.dat','r');
    case 6
        fileID=fopen('b6.dat','r');
    case 7
        fileID=fopen('b7.dat','r');
end

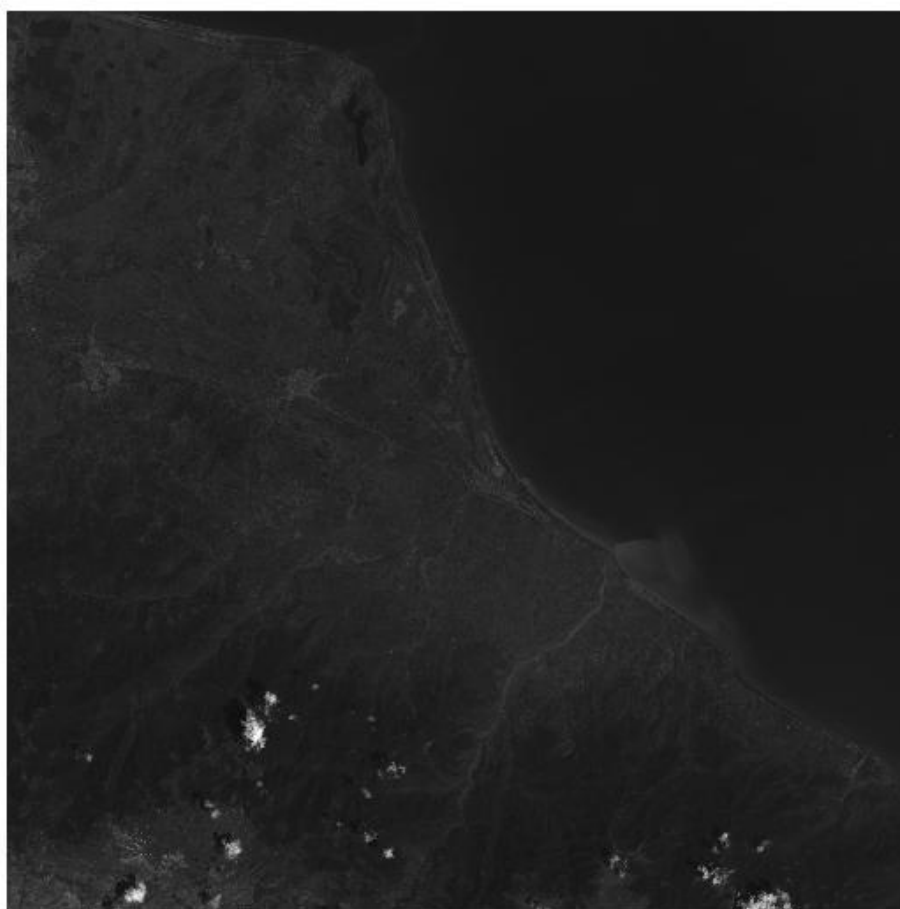
%number = fread(fileID,1,'*uint8');
numbers = fread(fileID,inf,'*uint8');    % All DNs in a vector
kernel=vec2mat(numbers,2048);    % Forming into a square matrix
imshow(kernel)

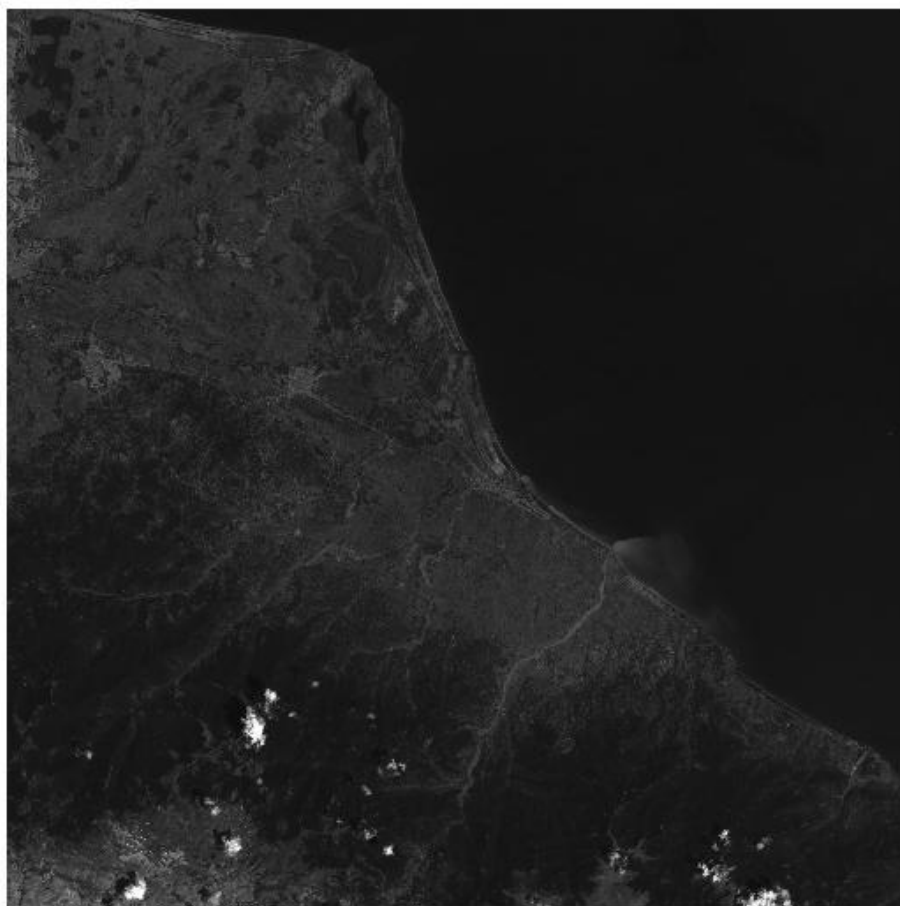
end
```

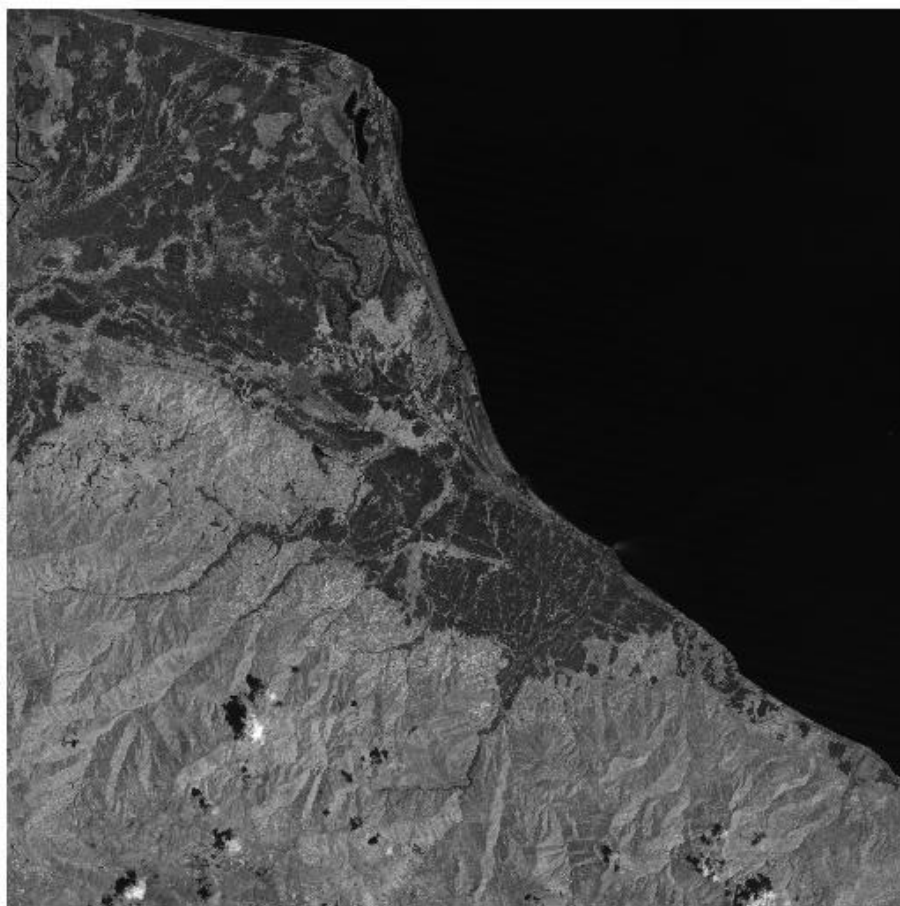
ج) خروجی یعنی تصویر باند نشان داده شده، به این محل کپی شود.

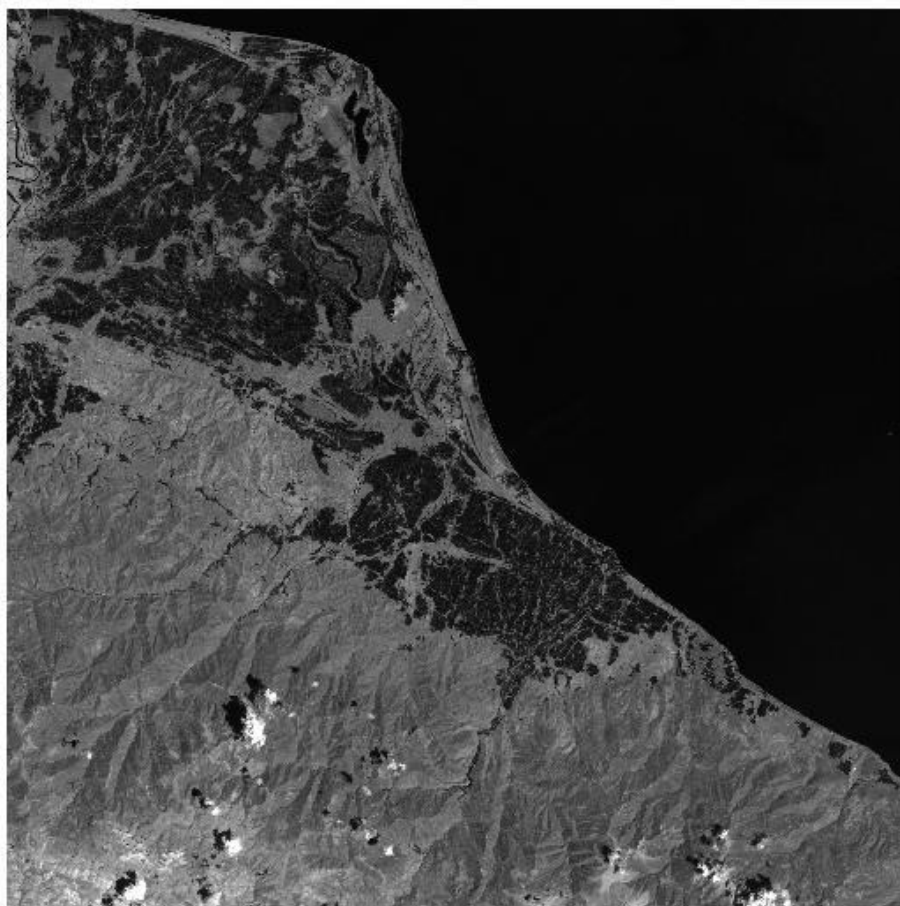
باند ۱

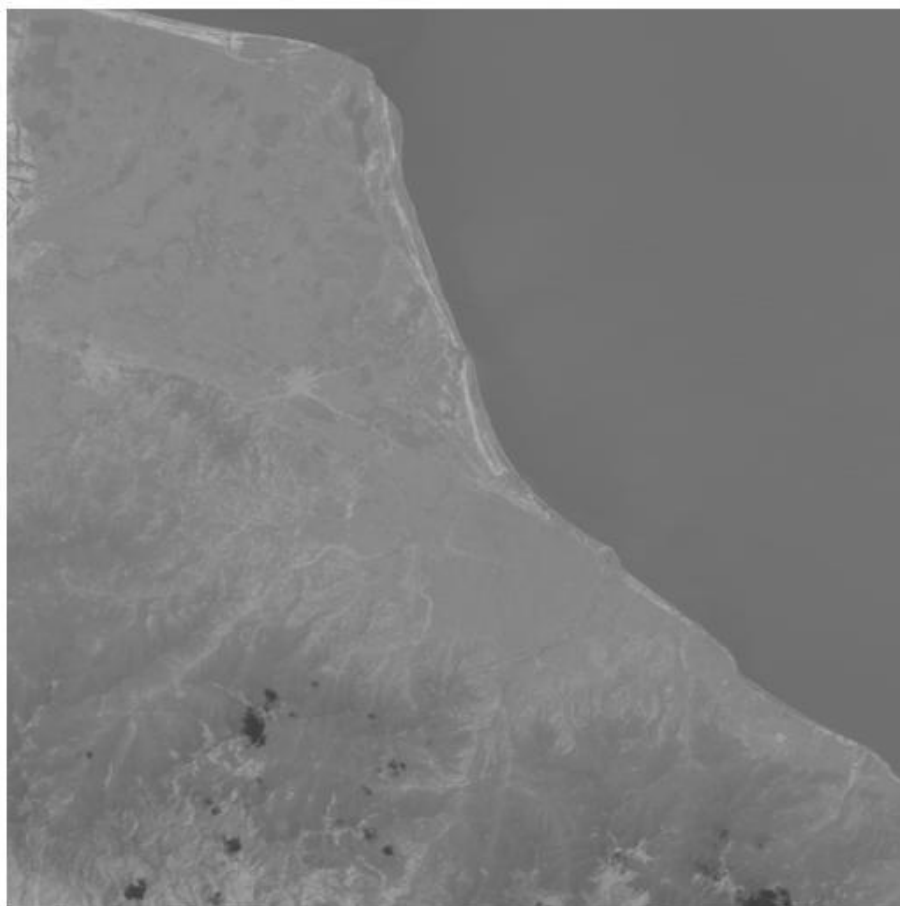










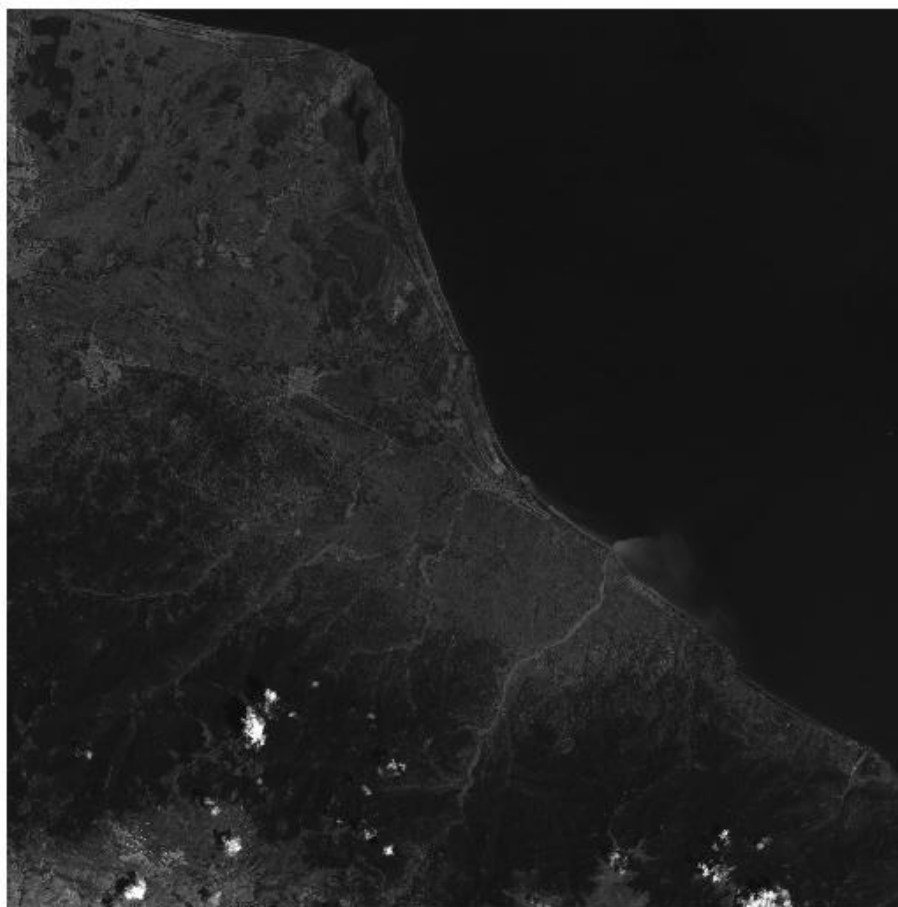




الف) عنوان آیتم شماره ۲ (محاسبه Radiance برای باندهای NIR و R تصویر سنجنده TM ماهواره Landsat و نمایش هر یک از آنها بنا به درخواست کاربر)

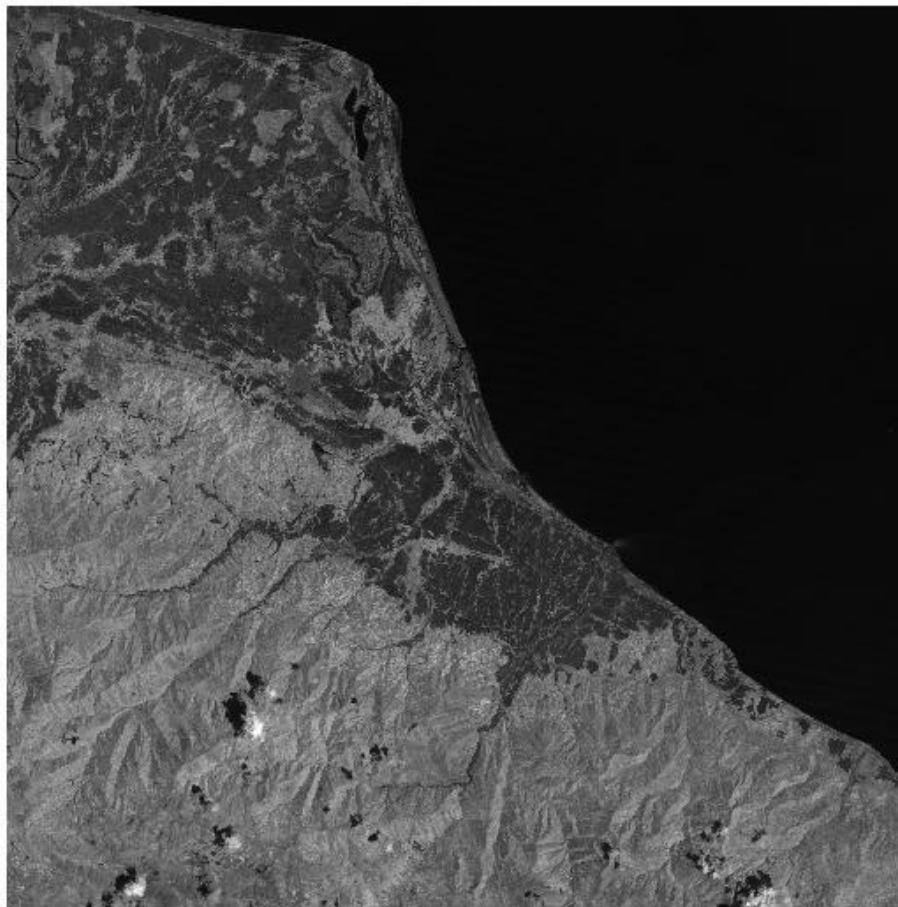
ب) دو باند ورودی یعنی تصاویر باندهای ۳ و ۴، به این محل کپی شود.

باند ۳



.
. .
. .
. .

باند ۴



ج) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

fileID_3=fopen('b3.dat','r');    %% R
fileID_4=fopen('b4.dat','r');    %% NIR

numbers_3 = fread(fileID_3,inf,'*uint8');
numbers_4 = fread(fileID_4,inf,'*uint8');

claim=input('Which Radiance you want to check? 3 or 4: ');
if claim==3                      %% R
    Lmax=264;Lmin=-1.17;
    L_3=((Lmax-Lmin)/255)*numbers_3+Lmin;
    kernel_3=vec2mat(L_3,2048);
    imshow(kernel_3)
    title('Radiance for R')
elseif claim==4                  %% NIR
    Lmax=221;Lmin=-1.51;
    L_4=((Lmax-Lmin)/255)*numbers_4+Lmin;
    kernel_4=vec2mat(L_4,2048);
    imshow(kernel_4)
    title('Radiance for NIR')
```

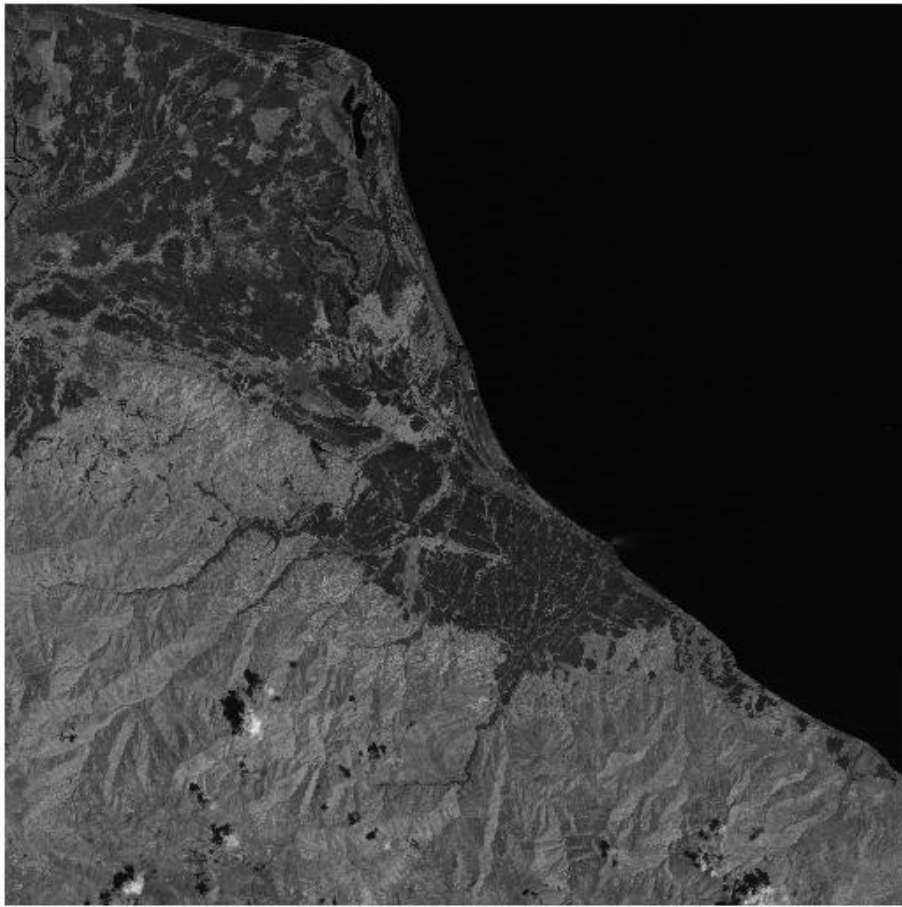
end

د) خروجیها یعنی دو تصویر Radiance برای باندهای NIR و R به این محل کپی شود.

Radiance for R

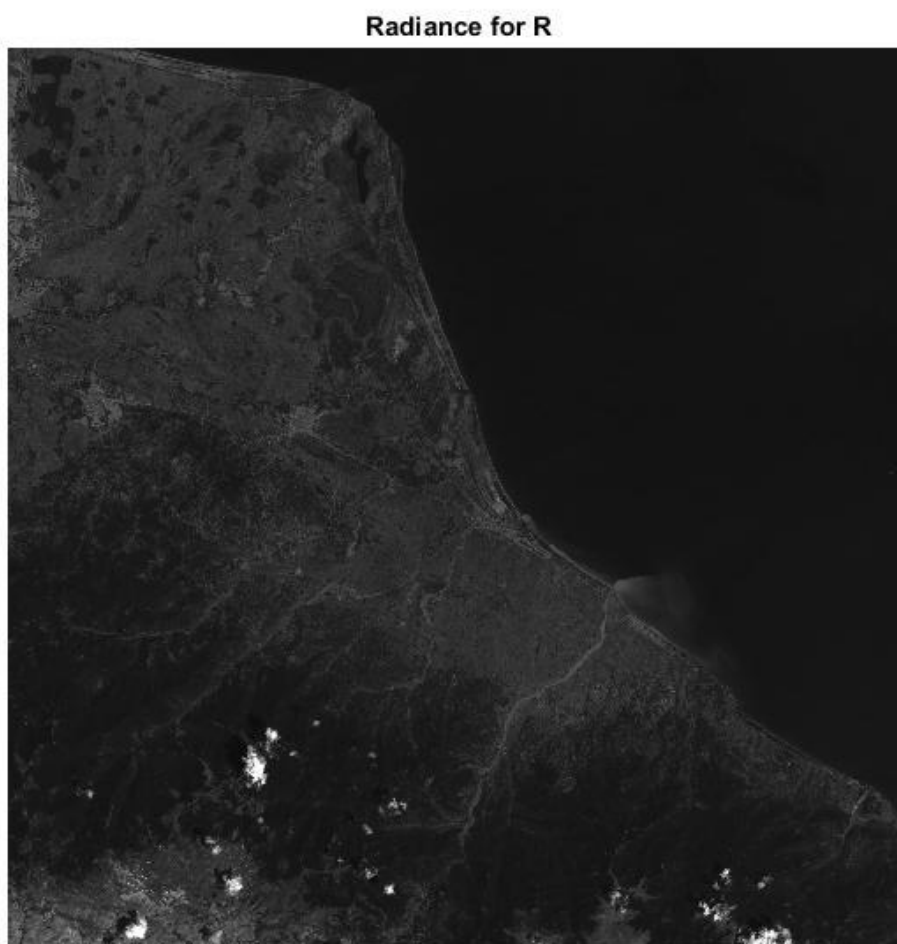


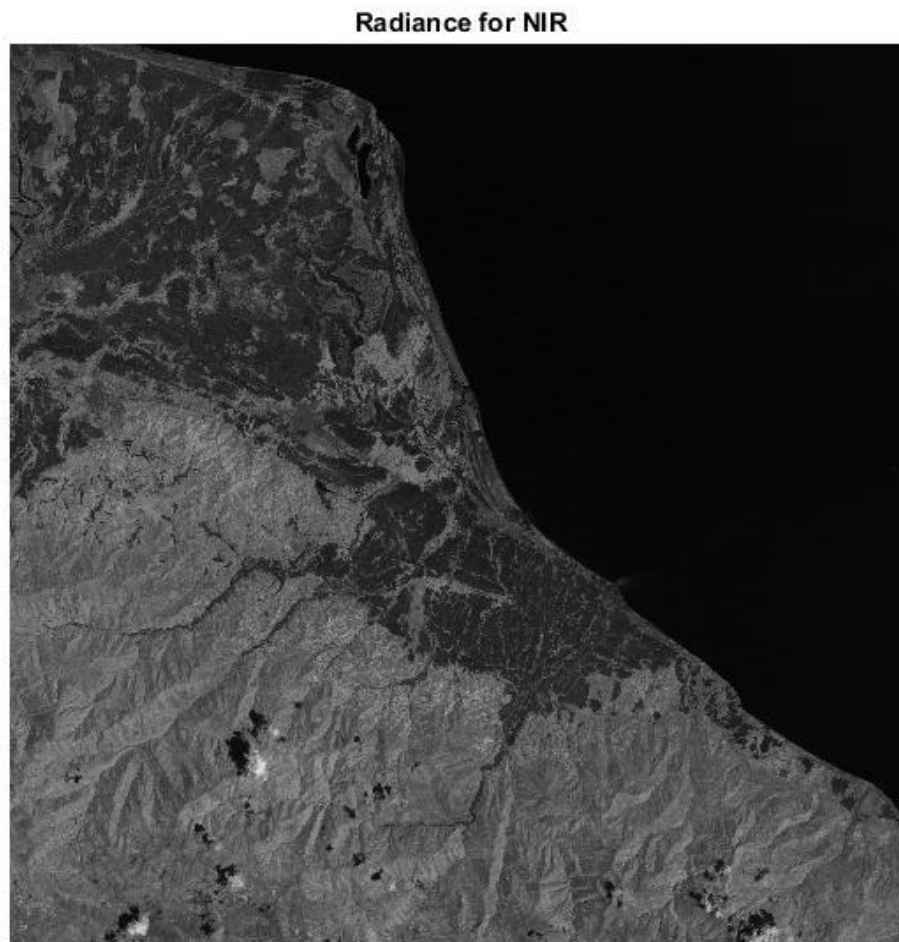
Radiance for NIR



الف) عنوان آیتم شماره ۳ (محاسبه و تهیه نقشه NDVI برای تصویر سنجنده TM ماهواره Landsat و نمایش آن). باندهای ورودی را فراخوانی کرده و به کمک فرمول، شاخص NDVI را محاسبه کرده و نمایش می دهیم.

ب) دو تصویر Radiance ورودی یعنی تصاویر Radiance باندهای ۳ و ۴، به این محل کپی شود.
رادیانس باند ۳





ج) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

format long

fileID_3=fopen('b3.dat','r');    %% R
fileID_4=fopen('b4.dat','r');    %% NIR

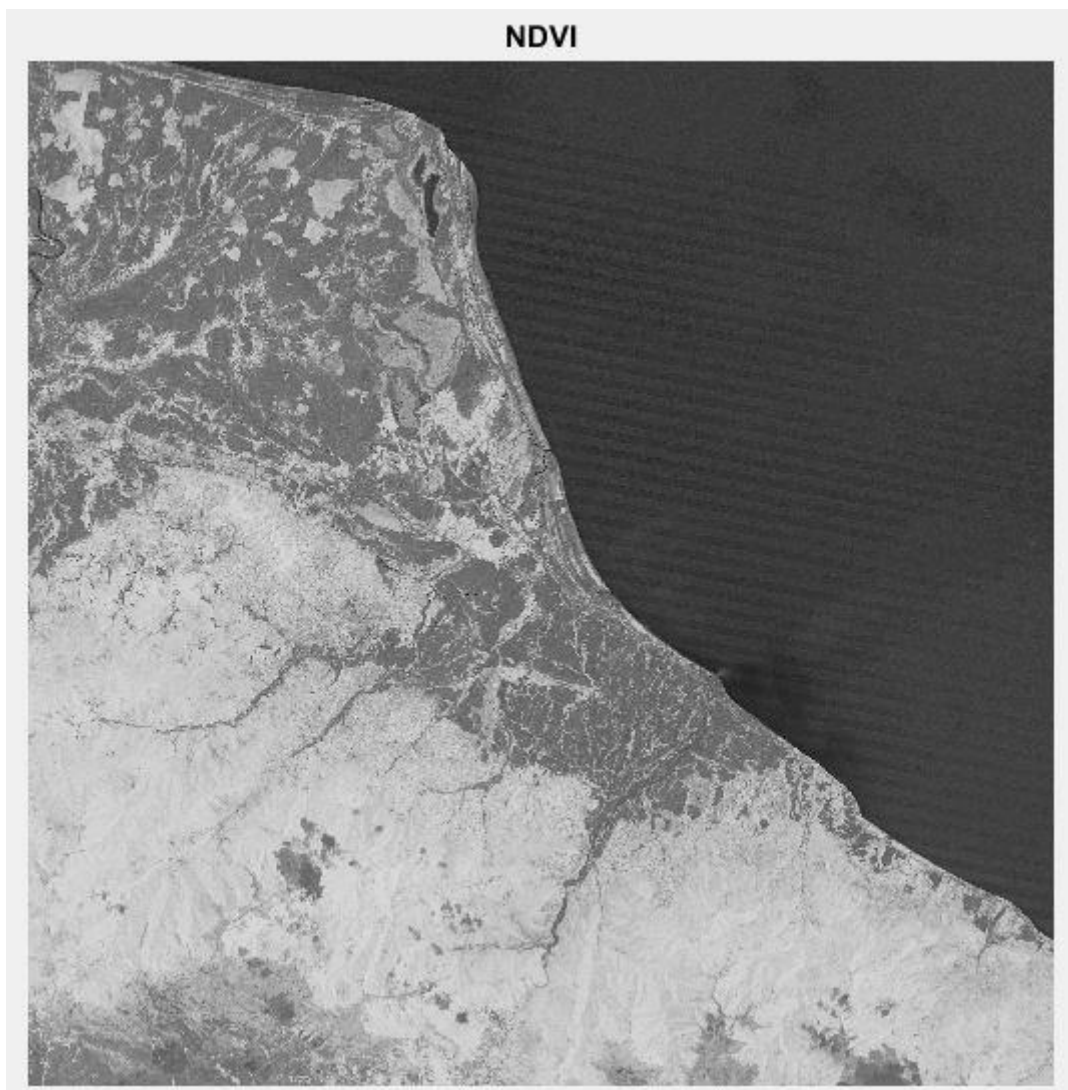
numbers_3 = fread(fileID_3,inf,'*uint8');
numbers_4 = fread(fileID_4,inf,'*uint8');

Lmax_3=264;Lmin_3=-1.17;
L_3=((Lmax_3-Lmin_3)/255)*numbers_3+Lmin_3;
kernel_3=double(vec2mat(L_3,2048));

Lmax_4=221;Lmin_4=-1.51;
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;
```

```
kernel_4=double(vec2mat(L_4,2048));  
  
NDVI=(kernel_4-kernel_3)./(kernel_4+kernel_3);  
  
NDVI_new=(NDVI+1);  
NDVI_new=NDVI_new.*127;  
  
imshow(uint8(NDVI_new))  
title('NDVI')
```

(د) خروجی یعنی تصویر NDVI به این محل کپی شود.



الف) عنوان آیتم شماره ۴ (مشخص کردن پوششهای زمینی مختلف از طریق جدول محدوده مقادیر NDVI و با استفاده از

رنگ سبز تیره برای پوشش گیاهی متراکم

رنگ سبز برای پوشش گیاهی متوسط

رنگ سبز روشن برای پوشش گیاهی تُنک

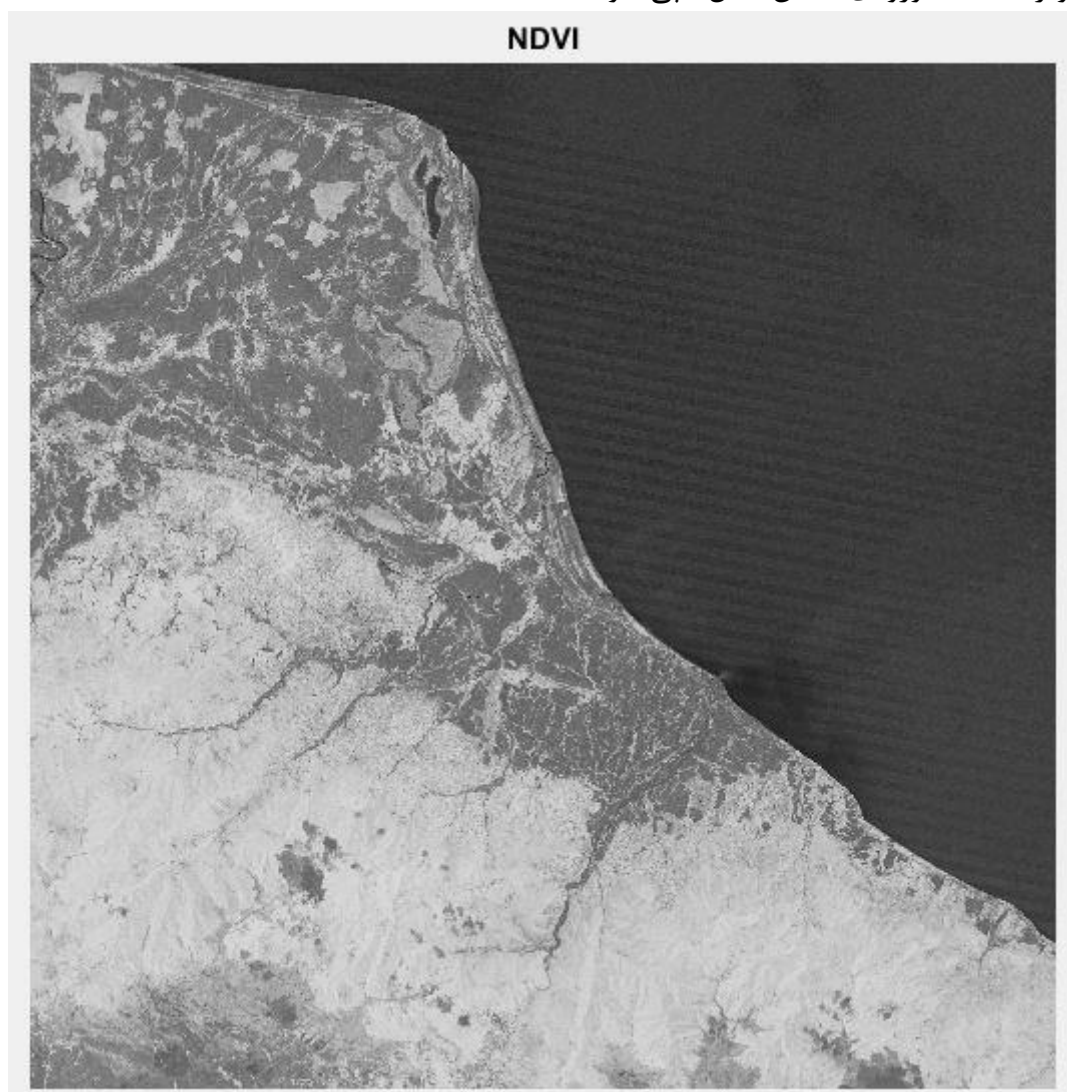
رنگ زرد برای زمین بایر

رنگ سفید برای ابر

رنگ آبی برای آب

بر روی تصویر NDVI ورودی شروط if را اعمال می کنیم. به طوریکه اگر مقادیر NDVI در بین بازه های هر کلاس که در جزوه آمده باشد، مربوط به کلاس مورد نظر است. سپس برای تعیین رنگ از مقایر RGB در سه باند تصویر استفاده می کنیم.

ب) تصویر NDVI ورودی به این محل کپی شود.



ج) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

format long

fileID_3=fopen('b3.dat','r');    %% R
fileID_4=fopen('b4.dat','r');    %% NIR

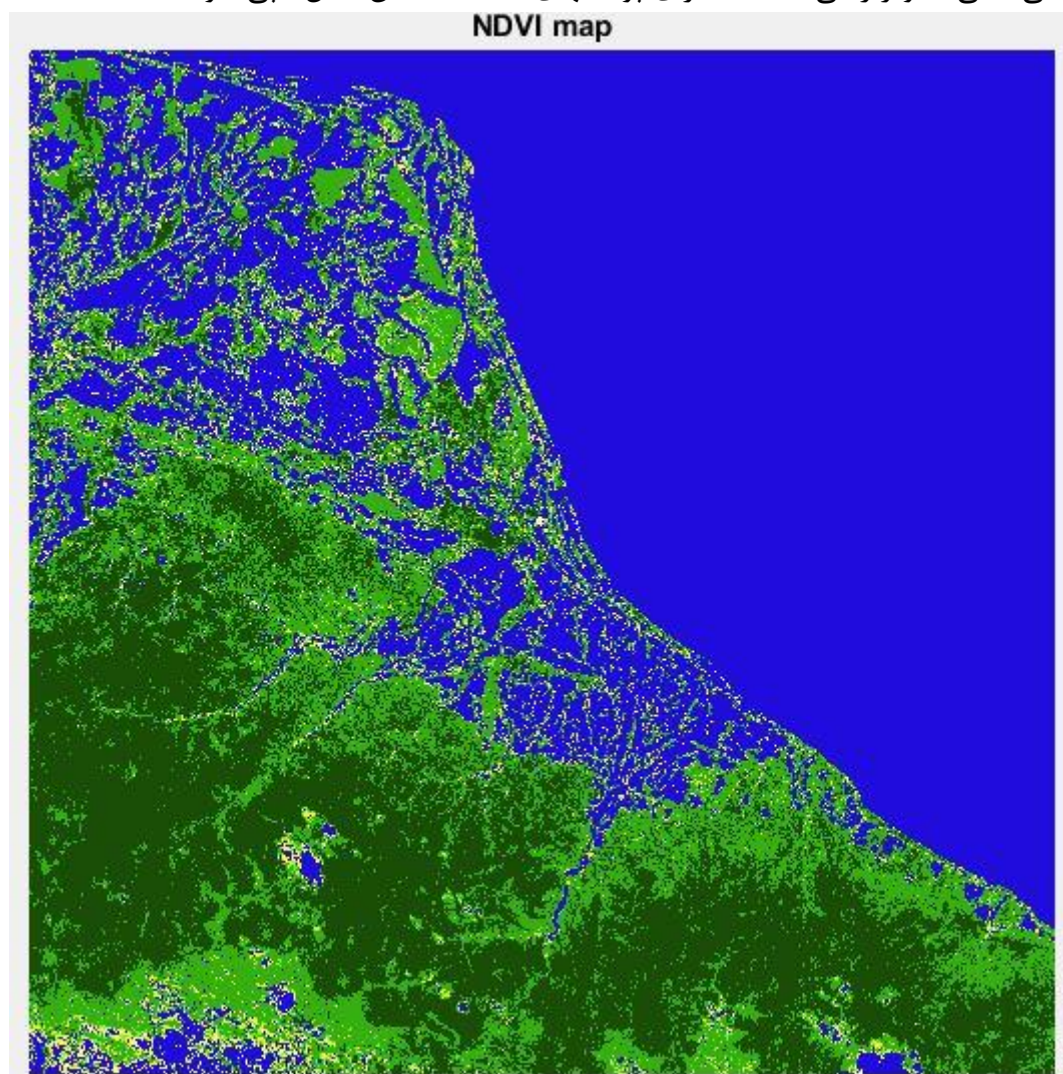
numbers_3 = fread(fileID_3,inf,'*uint8');
numbers_4 = fread(fileID_4,inf,'*uint8');

Lmax_3=264;Lmin_3=-1.17;
G=(Lmax_3-Lmin_3)/255;
L_3=((Lmax_3-Lmin_3)/255)*numbers_3+Lmin_3;

Lmax_4=221;Lmin_4=-1.51;
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;
L_4=double(L_4);
L_3=double(L_3);
NDVI=(L_4-L_3)./(L_4+L_3);
NDVI=double(vec2mat(NDVI,2048));

for i=1:2048
    for j=1:2048
        if NDVI(i,j)>=0.5
            map(i,j,1)=26;
            map(i,j,2)=78;
            map(i,j,3)=7;
        elseif NDVI(i,j)>=0.14 && NDVI(i,j)<0.5
            map(i,j,1)=56;
            map(i,j,2)=175;
            map(i,j,3)=16;
        elseif NDVI(i,j)>=0.09 && NDVI(i,j)<0.14
            map(i,j,1)=111;
            map(i,j,2)=238;
            map(i,j,3)=68;
        elseif NDVI(i,j)>=0.025 && NDVI(i,j)<0.09
            map(i,j,1)=255;
            map(i,j,2)=255;
            map(i,j,3)=0;
        elseif NDVI(i,j)>=0.002 && NDVI(i,j)<0.025
            map(i,j,1)=255;
            map(i,j,2)=255;
            map(i,j,3)=255;
        elseif NDVI(i,j)<0.002
            map(i,j,1)=33;
            map(i,j,2)=12;
            map(i,j,3)=222;
        end
    end
end
map=uint8(map);
imshow(map)
title('NDVI map')
```

د) خروجی یعنی تصویر رنگی NDVI برای پوششهای مختلف به این محل کپی شود.



الف) عنوان آیتم شماره ۵

- محاسبه OIF برای کلیه ترکیبهای سه بانده تصاویر Landsat5-TM
- تعیین مقدار ماکزیمم آن
- تعیین باندهای مربوط به آن
- نمایش تصویر رنگی حاصل از آن سه باند

ابتدا تمام باندها را به برنامه معرفی کرده و سپس سه باند سه بانده جدا کرده و دو به دو ضریب همبستگی باندها را محاسبه می کنیم. علاوه بر آن انحراف معیار هر باند را نیز بدست می آوریم. مطابق فرمول OIF، برای تمام سه باندهای جدا شده OIF را حساب می کنیم. مقدار ماکزیمم، بهترین ترکیب رنگی تصویر است. سه باندهای که بهترین ترکیب رنگی را حاصل کردند، باندهای ۱ و ۴ و ۵ بودند.

ب) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

fileID_1=fopen('b1.dat','r');numbers_1 =
fread(fileID_1,inf,'*uint8');kernel_1=vec2mat(numbers_1,2048);
fileID_2=fopen('b2.dat','r');numbers_2 =
fread(fileID_2,inf,'*uint8');kernel_2=vec2mat(numbers_2,2048);
fileID_3=fopen('b3.dat','r');numbers_3 =
fread(fileID_3,inf,'*uint8');kernel_3=vec2mat(numbers_3,2048);
fileID_4=fopen('b4.dat','r');numbers_4 =
fread(fileID_4,inf,'*uint8');kernel_4=vec2mat(numbers_4,2048);
fileID_5=fopen('b5.dat','r');numbers_5 =
fread(fileID_5,inf,'*uint8');kernel_5=vec2mat(numbers_5,2048);
fileID_7=fopen('b7.dat','r');numbers_7 =
fread(fileID_7,inf,'*uint8');kernel_7=vec2mat(numbers_7,2048);
sigma_1= std(std(double(kernel_1)));
sigma_2= std(std(double(kernel_2)));
sigma_3= std(std(double(kernel_3)));
sigma_4= std(std(double(kernel_4)));
sigma_5= std(std(double(kernel_5)));
sigma_7= std(std(double(kernel_7)));
%% 1,2,3
R12=corr2(kernel_1,kernel_2);
R23=corr2(kernel_2,kernel_3);
R13=corr2(kernel_1,kernel_3);
OIF(1)=(sigma_1+sigma_2+sigma_3)/(abs(R12)+abs(R23)+abs(R13));
%% 1,2,4
R12=corr2(kernel_1,kernel_2);
R24=corr2(kernel_2,kernel_4);
R14=corr2(kernel_1,kernel_4);
OIF(2)=(sigma_1+sigma_4+sigma_2)/(abs(R12)+abs(R24)+abs(R14));
%% 1,2,5
R12=corr2(kernel_1,kernel_2);
R25=corr2(kernel_2,kernel_5);
R15=corr2(kernel_1,kernel_5);
OIF(3)=(sigma_1+sigma_2+sigma_5)/(abs(R12)+abs(R25)+abs(R15));
%% 1,3,4
R14=corr2(kernel_1,kernel_4);
R34=corr2(kernel_3,kernel_4);
```

```

R13=corr2(kernel_1,kernel_3);
OIF(4)=(sigma_1+sigma_4+sigma_3)/(abs(R14)+abs(R34)+abs(R13));
%% 1,3,5
R15=corr2(kernel_1,kernel_5);
R35=corr2(kernel_3,kernel_5);
R13=corr2(kernel_1,kernel_3);
OIF(5)=(sigma_1+sigma_5+sigma_3)/(abs(R15)+abs(R35)+abs(R13));
%% 1,4,5
R14=corr2(kernel_1,kernel_4);
R45=corr2(kernel_4,kernel_5);
R15=corr2(kernel_1,kernel_5);
OIF(6)=(sigma_1+sigma_4+sigma_5)/(abs(R14)+abs(R45)+abs(R15));
%% 2,3,4
R23=corr2(kernel_2,kernel_3);
R24=corr2(kernel_2,kernel_4);
R34=corr2(kernel_3,kernel_4);
OIF(7)=(sigma_2+sigma_3+sigma_4)/(abs(R23)+abs(R24)+abs(R34));
%% 2,3,5
R25=corr2(kernel_2,kernel_5);
R23=corr2(kernel_2,kernel_3);
R35=corr2(kernel_3,kernel_5);
OIF(8)=(sigma_2+sigma_5+sigma_3)/(abs(R25)+abs(R23)+abs(R35));
%% 3,4,5
R34=corr2(kernel_3,kernel_4);
R45=corr2(kernel_4,kernel_5);
R35=corr2(kernel_3,kernel_5);
OIF(9)=(sigma_4+sigma_5+sigma_3)/(abs(R34)+abs(R45)+abs(R35));
%% 1,2,7
R12=corr2(kernel_1,kernel_2);
R17=corr2(kernel_1,kernel_7);
R27=corr2(kernel_2,kernel_7);
OIF(10)=(sigma_1+sigma_2+sigma_7)/(abs(R12)+abs(R17)+abs(R27));
%% 1,3,7
R13=corr2(kernel_1,kernel_3);
R17=corr2(kernel_1,kernel_7);
R37=corr2(kernel_3,kernel_7);
OIF(11)=(sigma_1+sigma_7+sigma_3)/(abs(R13)+abs(R17)+abs(R37));
%% 1,4,7
R14=corr2(kernel_1,kernel_4);
R47=corr2(kernel_4,kernel_7);
R17=corr2(kernel_1,kernel_7);
OIF(12)=(sigma_4+sigma_1+sigma_7)/(abs(R14)+abs(R47)+abs(R17));
%% 1,5,7
R15=corr2(kernel_1,kernel_5);
R17=corr2(kernel_1,kernel_7);
R57=corr2(kernel_5,kernel_7);
OIF(13)=(sigma_1+sigma_5+sigma_7)/(abs(R15)+abs(R17)+abs(R57));
%% 2,3,7
R37=corr2(kernel_3,kernel_7);
R23=corr2(kernel_2,kernel_3);
R27=corr2(kernel_2,kernel_7);
OIF(14)=(sigma_2+sigma_3+sigma_7)/(abs(R37)+abs(R23)+abs(R27));
%% 2,4,7
R24=corr2(kernel_2,kernel_4);
R47=corr2(kernel_4,kernel_7);
R27=corr2(kernel_2,kernel_7);
OIF(15)=(sigma_4+sigma_2+sigma_7)/(abs(R24)+abs(R47)+abs(R27));
%% 2,5,7
R27=corr2(kernel_2,kernel_7);
R25=corr2(kernel_2,kernel_5);
R57=corr2(kernel_5,kernel_7);

```

```

OIF(16)=(sigma_2+sigma_5+sigma_7)/(abs(R27)+abs(R25)+abs(R57));
%% 3,4,7
R34=corr2(kernel_3,kernel_4);
R47=corr2(kernel_4,kernel_7);
R37=corr2(kernel_3,kernel_7);
OIF(17)=(sigma_4+sigma_7+sigma_3)/(abs(R34)+abs(R47)+abs(R37));
%% 3,5,7
R35=corr2(kernel_3,kernel_5);
R57=corr2(kernel_5,kernel_7);
R37=corr2(kernel_3,kernel_7);
OIF(18)=(sigma_7+sigma_5+sigma_3)/(abs(R37)+abs(R57)+abs(R35));
%% 4,5,7
R47=corr2(kernel_4,kernel_7);
R45=corr2(kernel_4,kernel_5);
R57=corr2(kernel_5,kernel_7);
OIF(19)=(sigma_4+sigma_5+sigma_7)/(abs(R47)+abs(R45)+abs(R57));

maximum= max(OIF)
disp('bands num 1,4,5 provide best OIF.')
RGB_image=cat(3,kernel_1,kernel_4,kernel_5);
imshow(RGB_image); title('OIF')

```

ج) مقدار ماگزیمم OIF بدست آمده به این محل کپی شود.

```

maximum =
13.146261817088819

```

د) شماره باندهای مربوط به مقدار ماگزیمم OIF به این محل کپی شود.

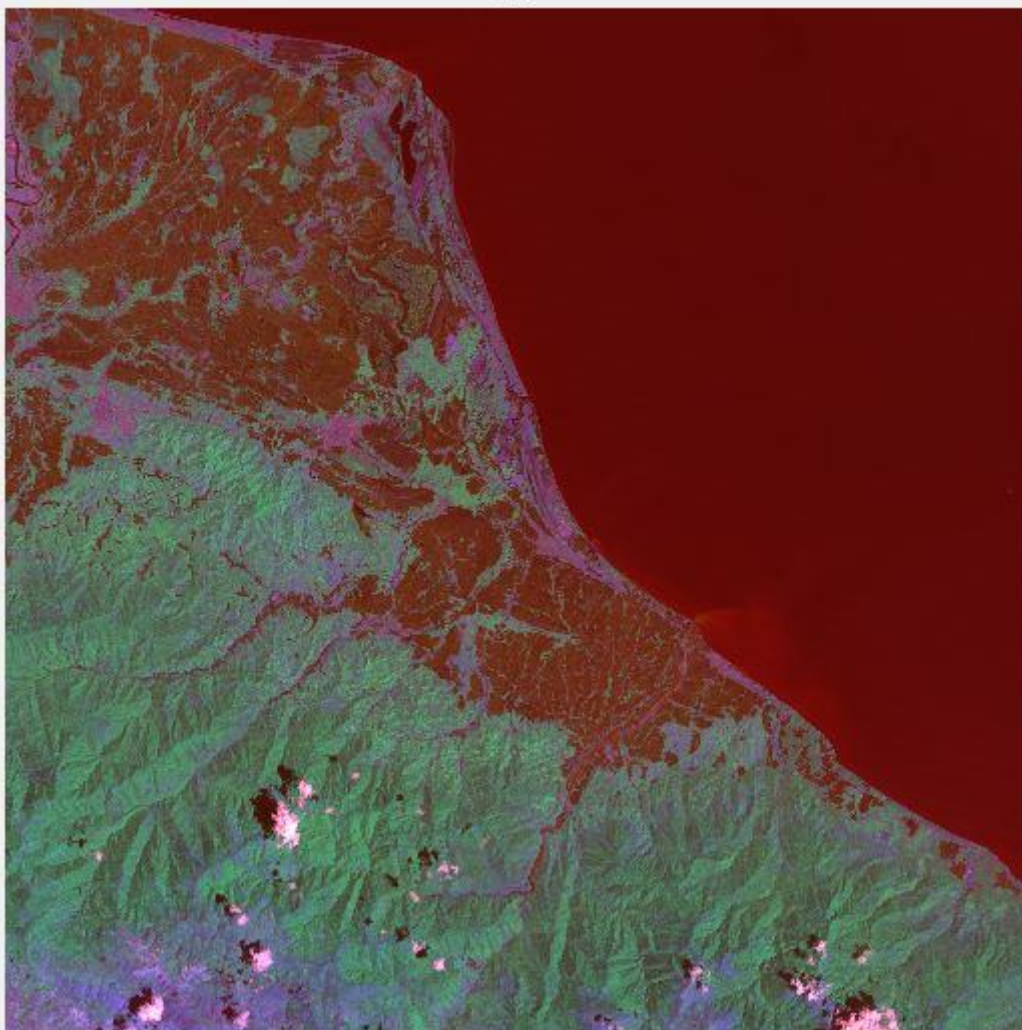
```

bands num 1,4,5 provide best OIF.

```

ه) خروجی یعنی تصویر رنگی حاصله از آن سه باند به این محل کپی شود.

OIF



الف) عنوان آیتم شماره ۶ (محاسبه و تهیه نقشه NDVI برای تصویر سنجنده TM ماهواره Landsat و نمایش آن؛ نمایش مقادیر NDVI در چند پیکسل از پوشش های مختلف).

باندهای ورودی را فراخوانی کرده و به کمک فرمول، شاخص NDVI را محاسبه کرده و نمایش می دهیم. برای نمایش مقادیر NDVI پوشش های مختلف، از ابزار data cursor استفاده می کنیم و چند پیکسل از مناطق مختلف را انتخاب کرده و محیط اسکرپ مقدار شاخص را برای آنها بدست می آوریم.

ب) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

format long

fileID_3=fopen('b3.dat','r');    %% R
fileID_4=fopen('b4.dat','r');    %% NIR

numbers_3 = fread(fileID_3,inf,'*uint8');
numbers_4 = fread(fileID_4,inf,'*uint8');

Lmax_3=264;Lmin_3=-1.17;
L_3=((Lmax_3-Lmin_3)/255)*numbers_3+Lmin_3;
kernel_3=double(vec2mat(L_3,2048));

Lmax_4=221;Lmin_4=-1.51;
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;
kernel_4=double(vec2mat(L_4,2048));

NDVI=(kernel_4-kernel_3)./(kernel_4+kernel_3);
NDVI_new=(NDVI+1).*127;

imshow(uint8(NDVI_new))
title('NDVI')

fprintf('Vegetation: %d\n',NDVI(1803,1278));
fprintf('Water: %d\n',NDVI(1227,1926));
fprintf('Bare Ground: %d\n',NDVI(637,722));
```

ج) مقادیر NDVI در چند پیکسل مختلف به این محل کپی شود.

```
Vegetation: 5.217391e-01
Water: -4.666667e-01
Bare Ground: -6.172840e-02
```

د) خروجی یعنی نقشه NDVI به این محل کپی شود.

NDVI

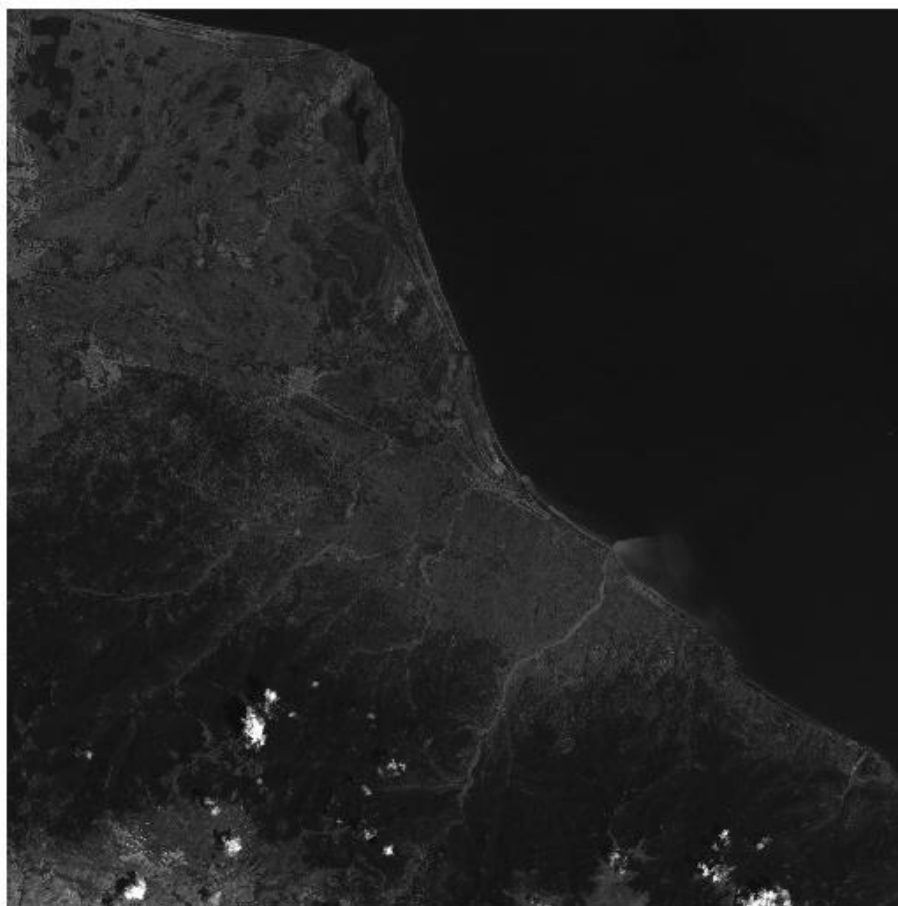


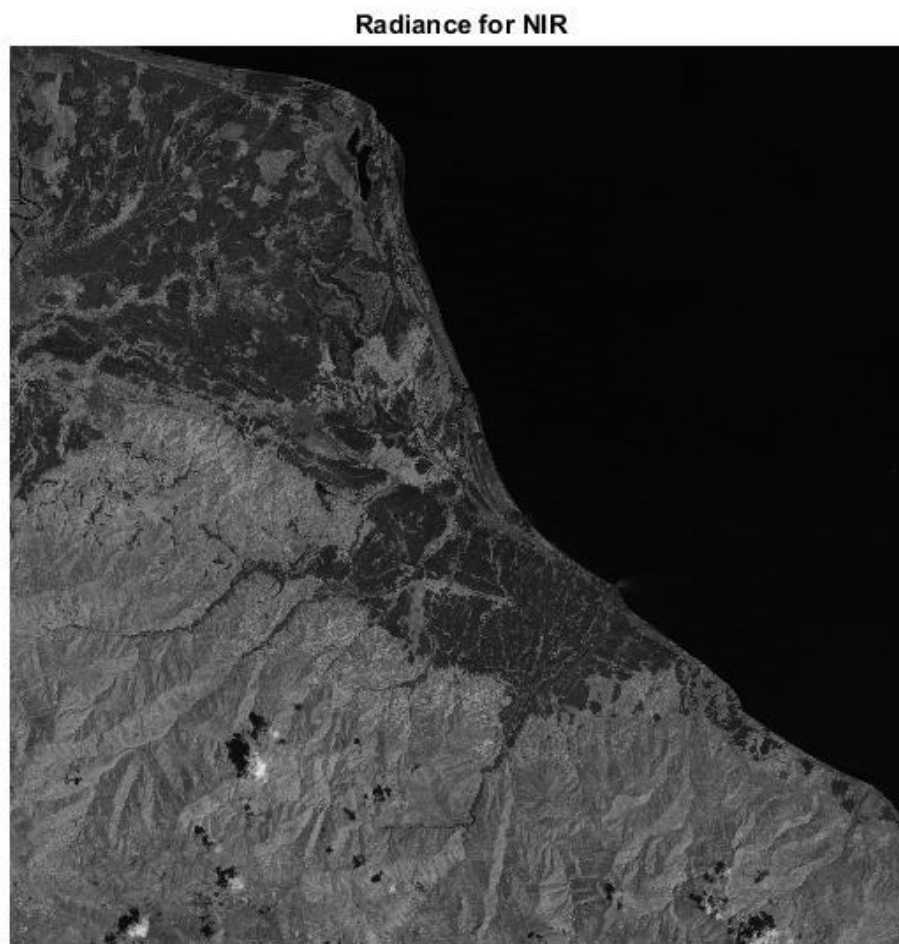
الف) عنوان آیتم شماره ۷ (محاسبه و تهیه نقشه SAVI برای تصویر سنجنده TM ماهواره Landsat) برای شناسایی گیاه به کار می رود. باند های ورودی را تعریف کرده و رادیانسانشان را بدست می آوریم. سپس با توجه به فرمول، شاخص را برای تصویر مورد نظر نمایش می دهیم. مقدار L را برای این منطقه ۰.۲ در نظر میگیریم.

ب) دو باند ورودی، به این محل کپی شود.

باند ۳

Radiance for R





(ج) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

fileID_3=fopen('b3.dat','r');    %% R
fileID_4=fopen('b4.dat','r');    %% NIR

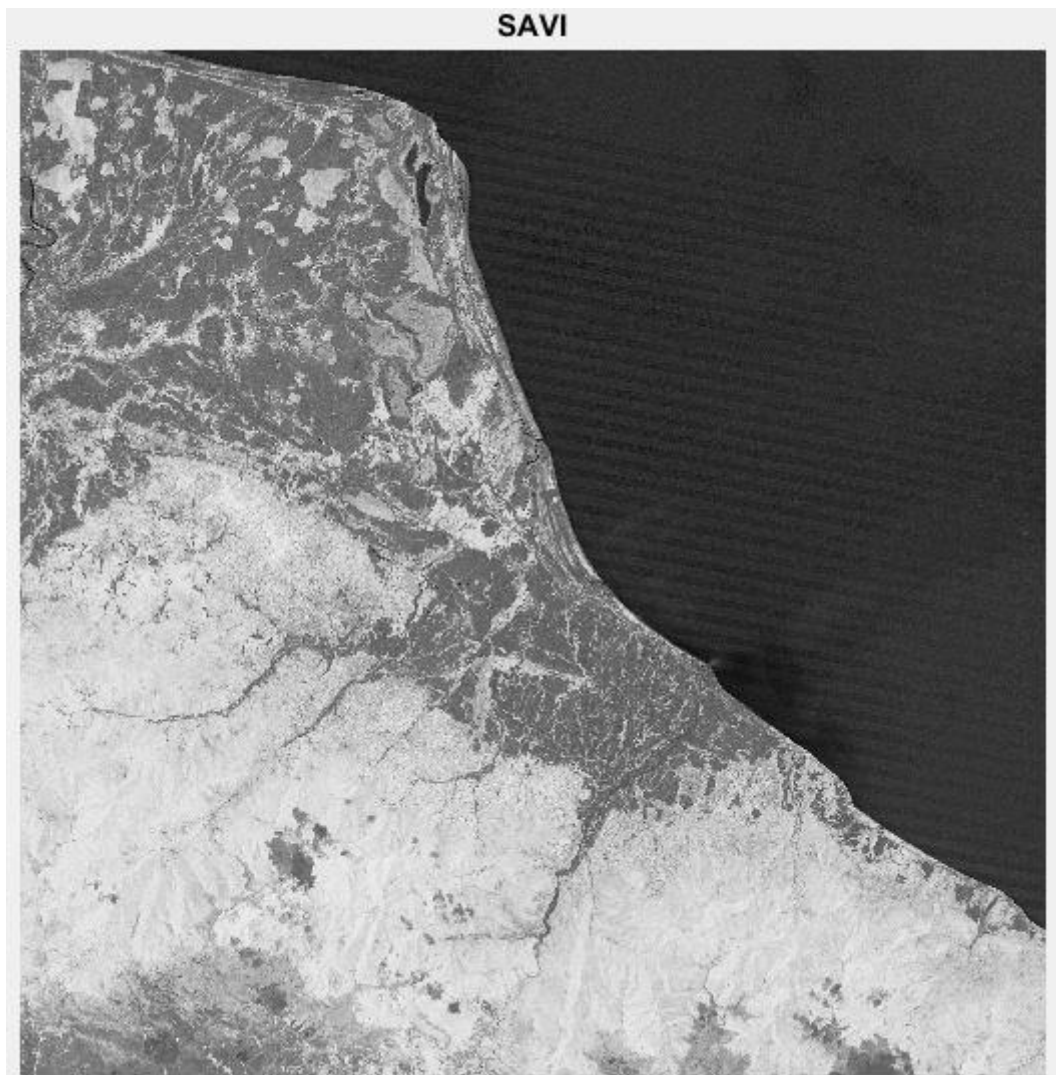
numbers_3 = fread(fileID_3,inf,'*uint8');
numbers_4 = fread(fileID_4,inf,'*uint8');

Lmax_3=264;Lmin_3=-1.17;
L_3=((Lmax_3-Lmin_3)/255)*numbers_3+Lmin_3;
kernel_3=double(vec2mat(L_3,2048));

Lmax_4=221;Lmin_4=-1.51;
```

```
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;  
kernel_4=double(vec2mat(L_4,2048));  
  
L=0.2;  
SAVI=((kernel_4-kernel_3)*(1+L))./(kernel_4+kernel_3+L);  
SAVI=(SAVI+1)*127;  
imshow(uint8(SAVI))  
title('SAVI')
```

(د) خروجی یعنی تصویر SAVI به این محل کپی شود.



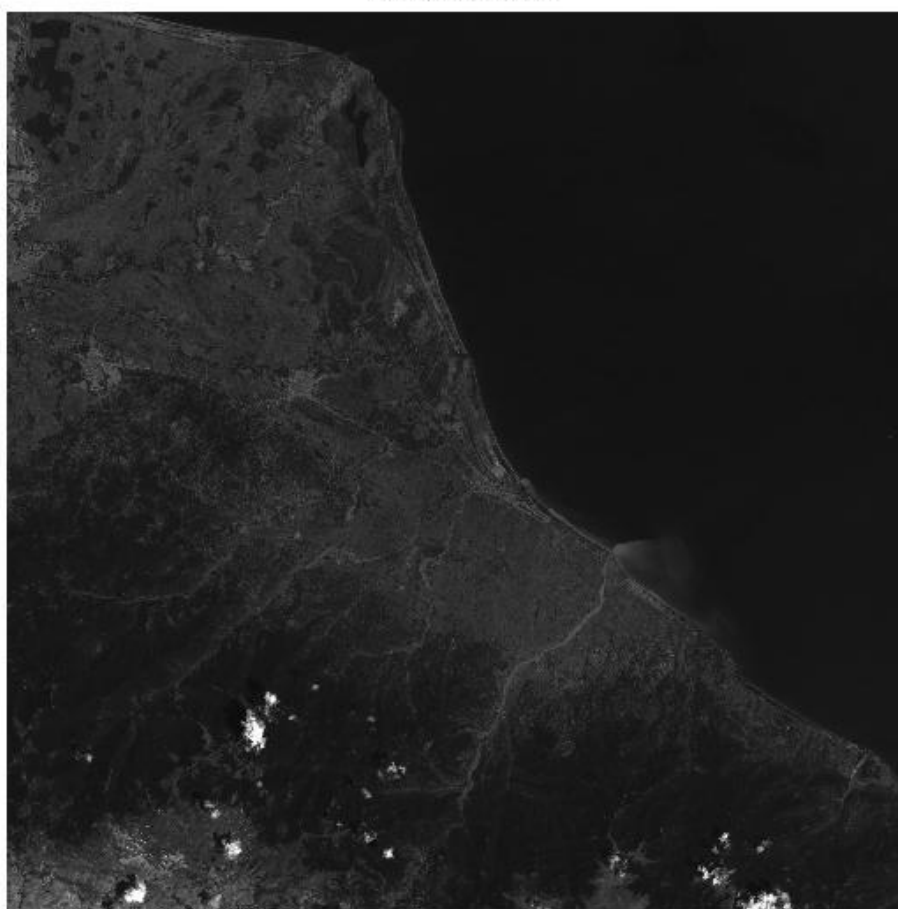
الف) عنوان آیتم شماره ۸ (محاسبه و تهیه نقشه MSAVI برای تصویر سنجنده TM ماهواره Landsat)

برای شناسایی گیاه با به حداقل رساندن اثر پس زمینه ی خاک به کار میرود. باند های ورودی را تعریف کرده و به کمک S، L را بدست می آوریم. سپس L را در فرمول شاخص جایگذاری می کنیم.

ب) دو باند ورودی، به این محل کپی شود.

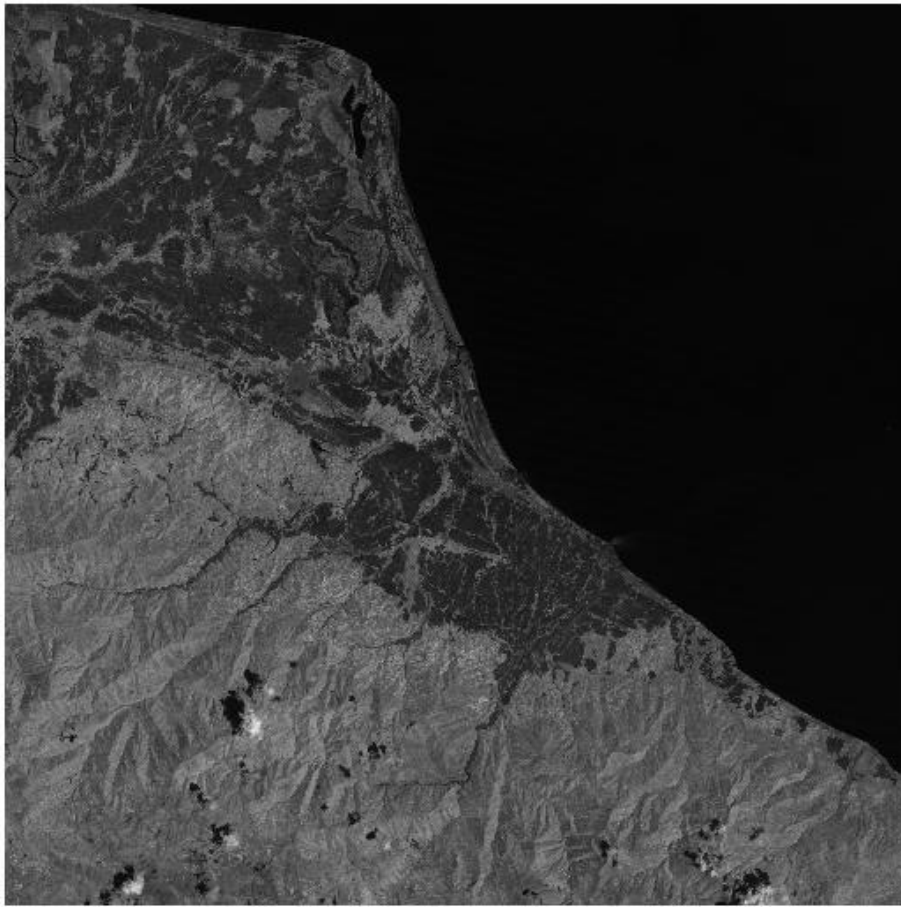
باند ۳

Radiance for R



باند ۴

Radiance for NIR



ج) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

fileID_3=fopen('b3.dat','r');    %% R
fileID_4=fopen('b4.dat','r');    %% NIR

numbers_3 = fread(fileID_3,inf,'*uint8');
numbers_4 = fread(fileID_4,inf,'*uint8');

Lmax_3=264;Lmin_3=-1.17;
L_3=((Lmax_3-Lmin_3)/255)*numbers_3+Lmin_3;
kernel_3=double(vec2mat(L_3,2048));

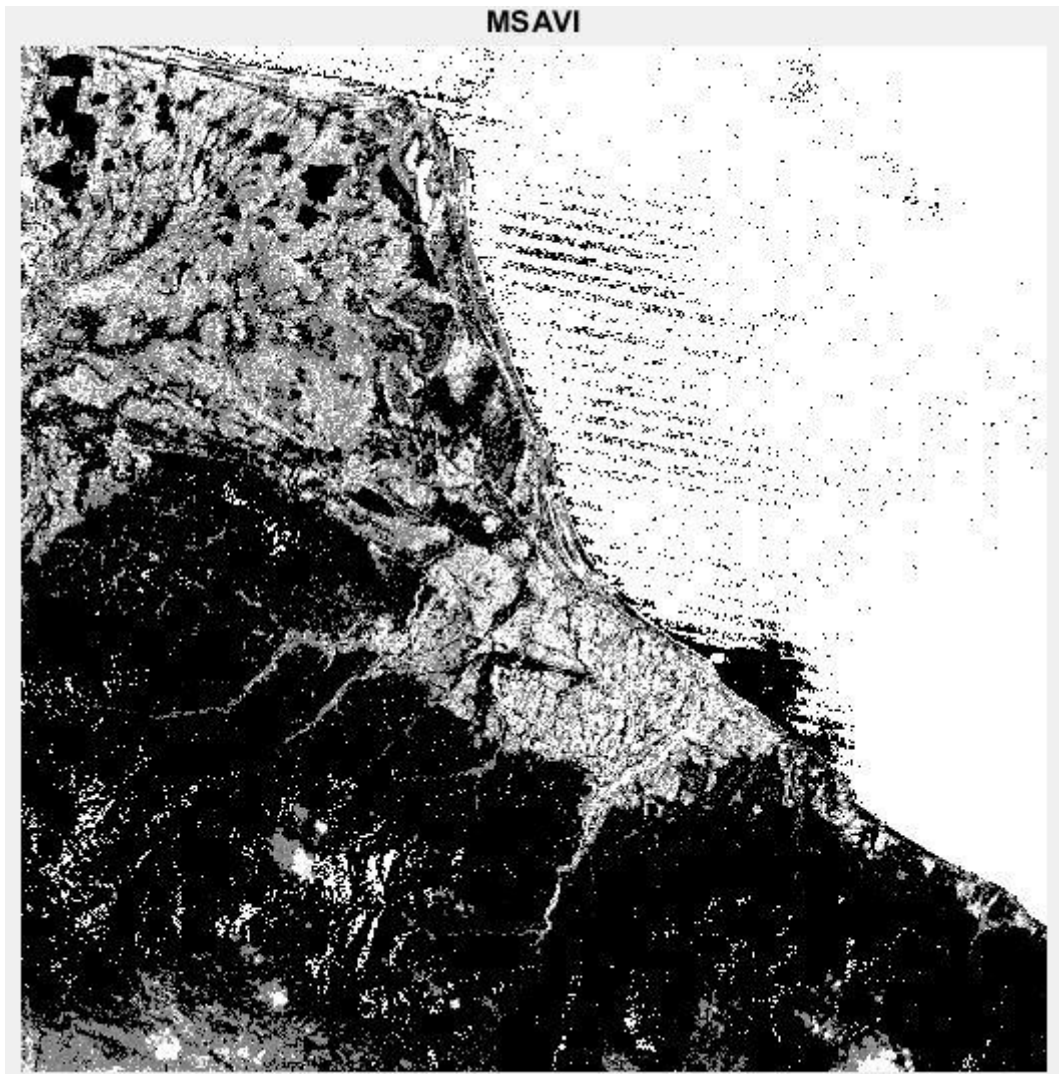
Lmax_4=221;Lmin_4=-1.51;
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;
kernel_4=double(vec2mat(L_4,2048));

s=1.17;
L=1-((2*s*(kernel_4-kernel_3).*(kernel_4-
(s*kernel_3)))./(kernel_4+kernel_3));
```



```
MSAVI=((kernel_4-kernel_3).*(1+L))./(kernel_4+kernel_3+L);  
MSAVI=(MSAVI+1)*127;  
imshow(uint8(MSAVI))  
title('MSAVI')
```

(د) خروجی یعنی تصویر MSAVI به این محل کپی شود.

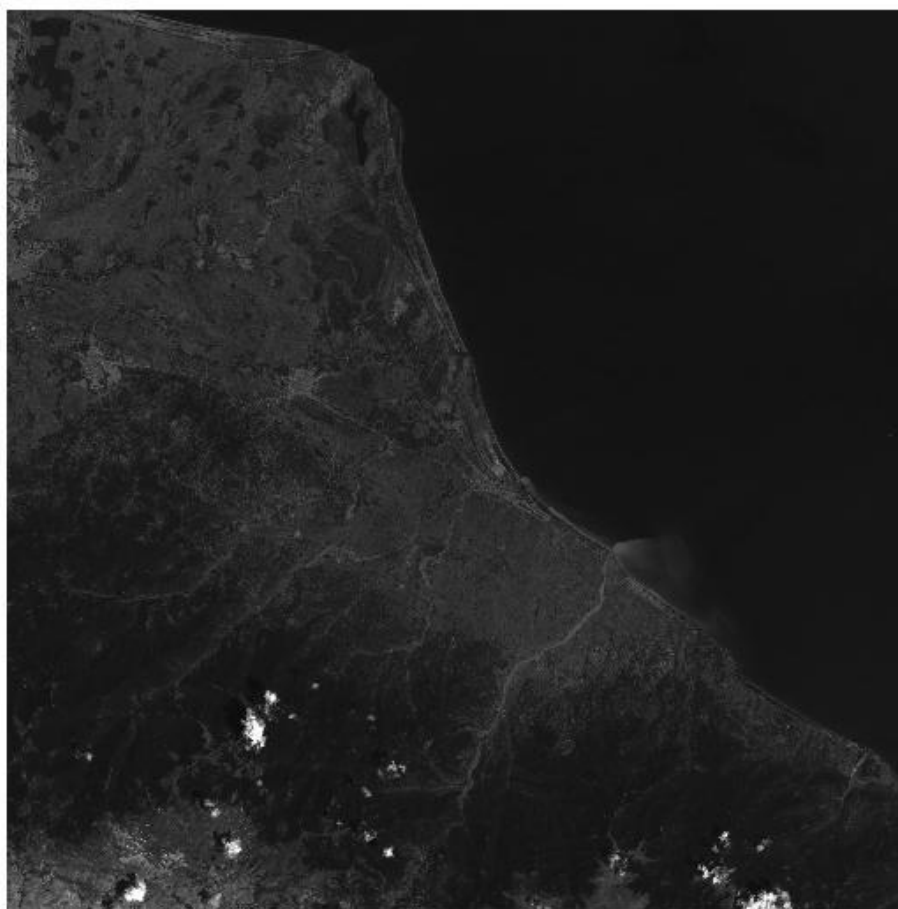


الف) عنوان آیتم شماره ۹ (محاسبه و تهیه نقشه MSAVI2 برای تصویر سنجنده TM ماهواره Landsat). به دلیل عدم قطعیت در تخمین L از این شاخص استفاده می کنیم. مقادیر رادیانس باند های ورودی را در فرمول جایگذاری می کنیم و تصویر شاخص را نمایش می دهیم.

ب) دو باند ورودی، به این محل کپی شود.

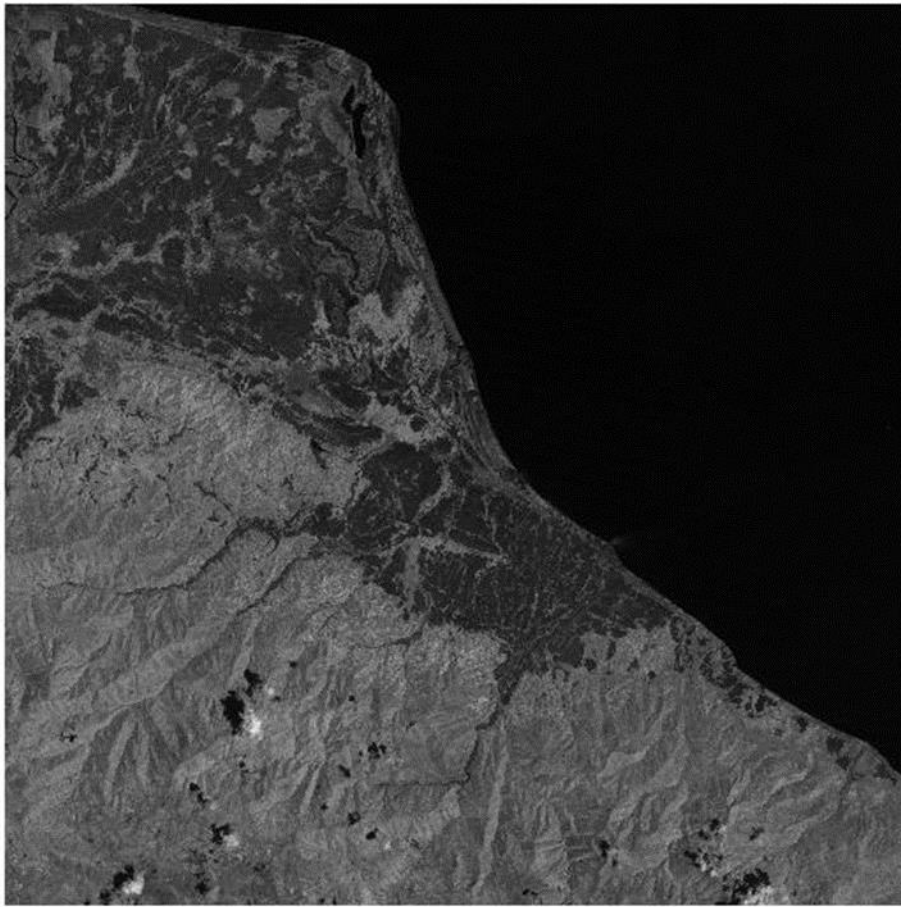
باند ۳

Radiance for R



باند ۴

Radiance for NIR



ج) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

fileID_3=fopen('b3.dat','r');    %% R
fileID_4=fopen('b4.dat','r');    %% NIR

numbers_3 = fread(fileID_3,inf,'*uint8');
numbers_4 = fread(fileID_4,inf,'*uint8');

Lmax_3=264;Lmin_3=-1.17;
L_3=((Lmax_3-Lmin_3)/255)*numbers_3+Lmin_3;
kernel_3=double(vec2mat(L_3,2048));

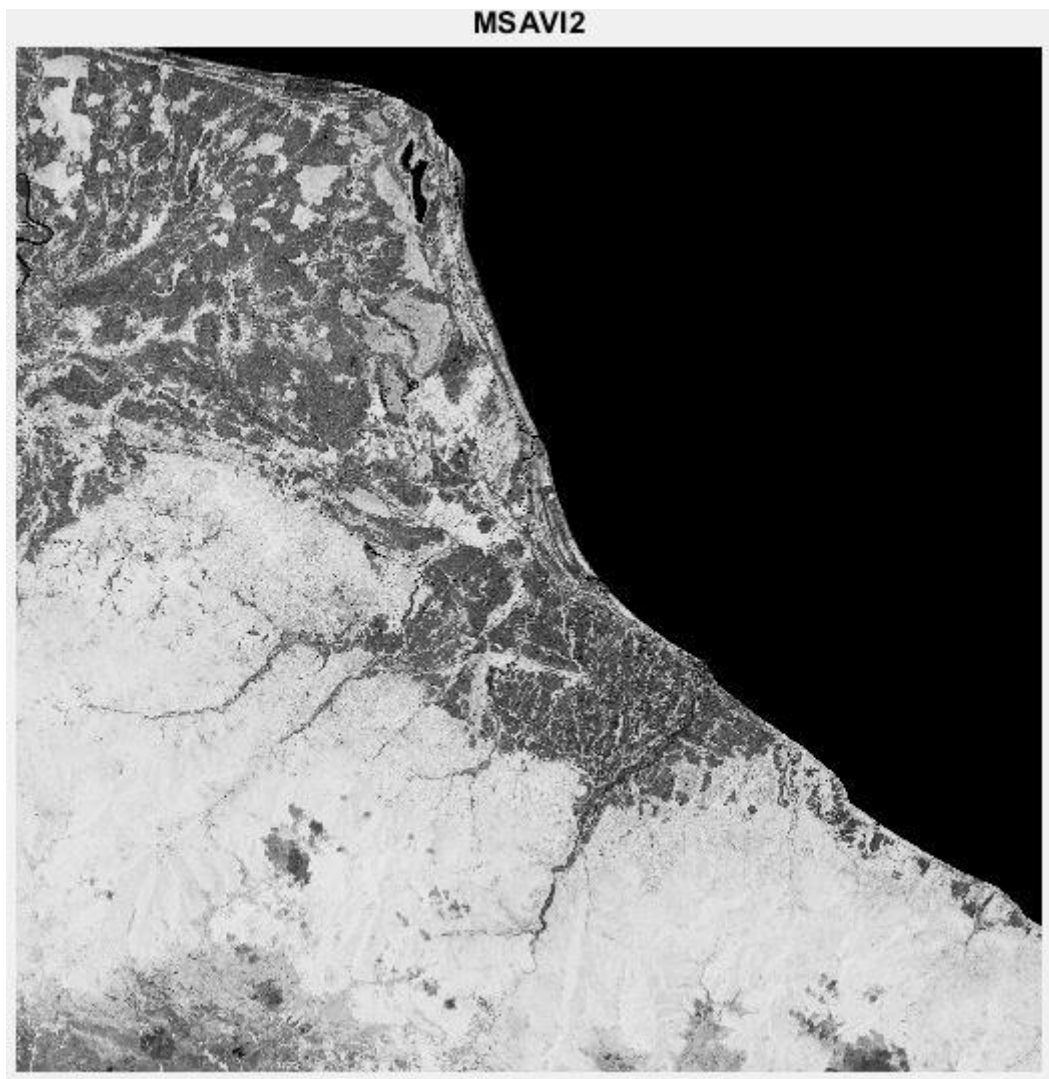
Lmax_4=221;Lmin_4=-1.51;
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;
kernel_4=double(vec2mat(L_4,2048));

MSAVI2=((2*kernel_4)+1-(sqrt((2*kernel_4+1).^2-(8*(kernel_4-
kernel_3)))))/(2);
MSAVI2=(MSAVI2+1)*127;
```



```
imshow(uint8(MSAVI2))  
title('MSAVI2')
```

(د) خروجی یعنی تصویر MSAVI2 به این محل کپی شود.

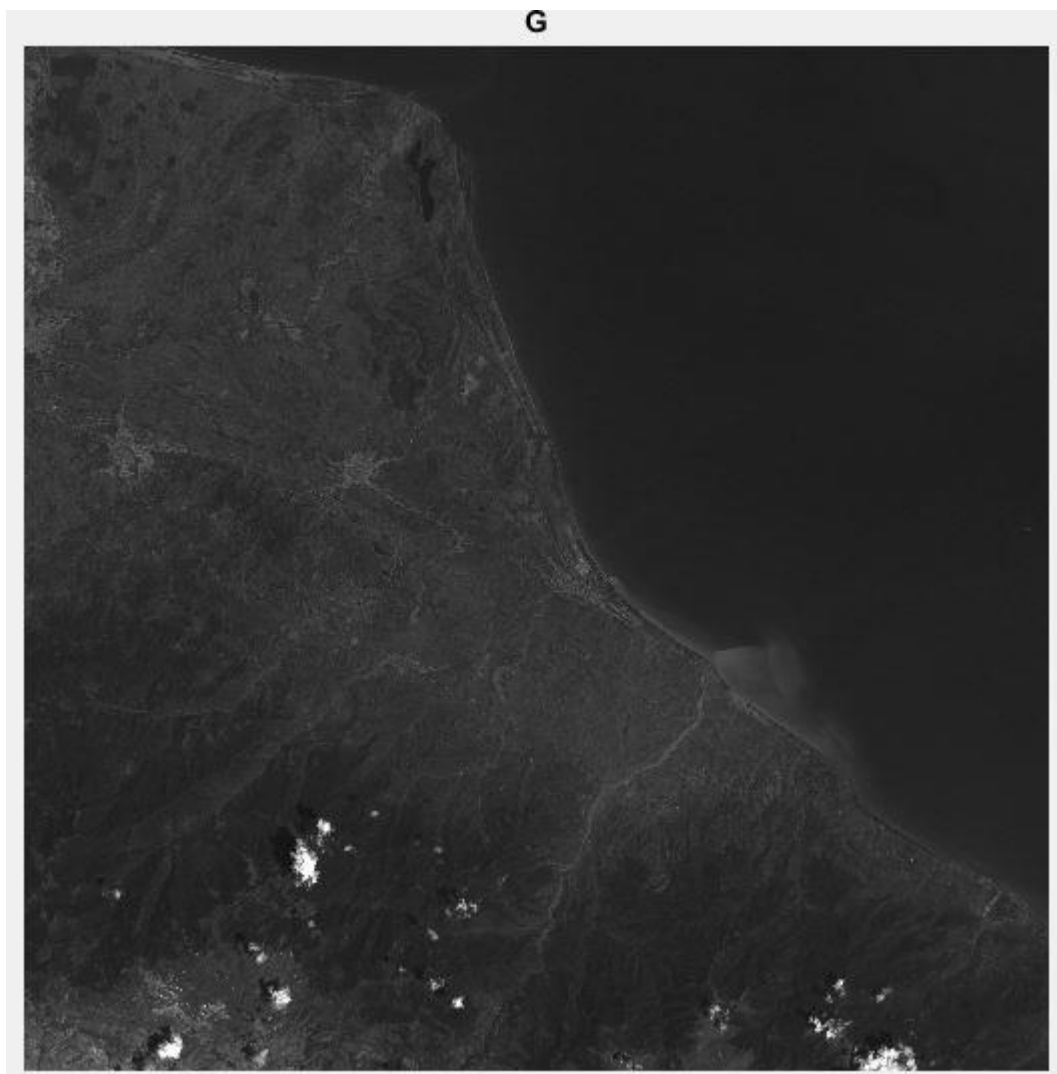


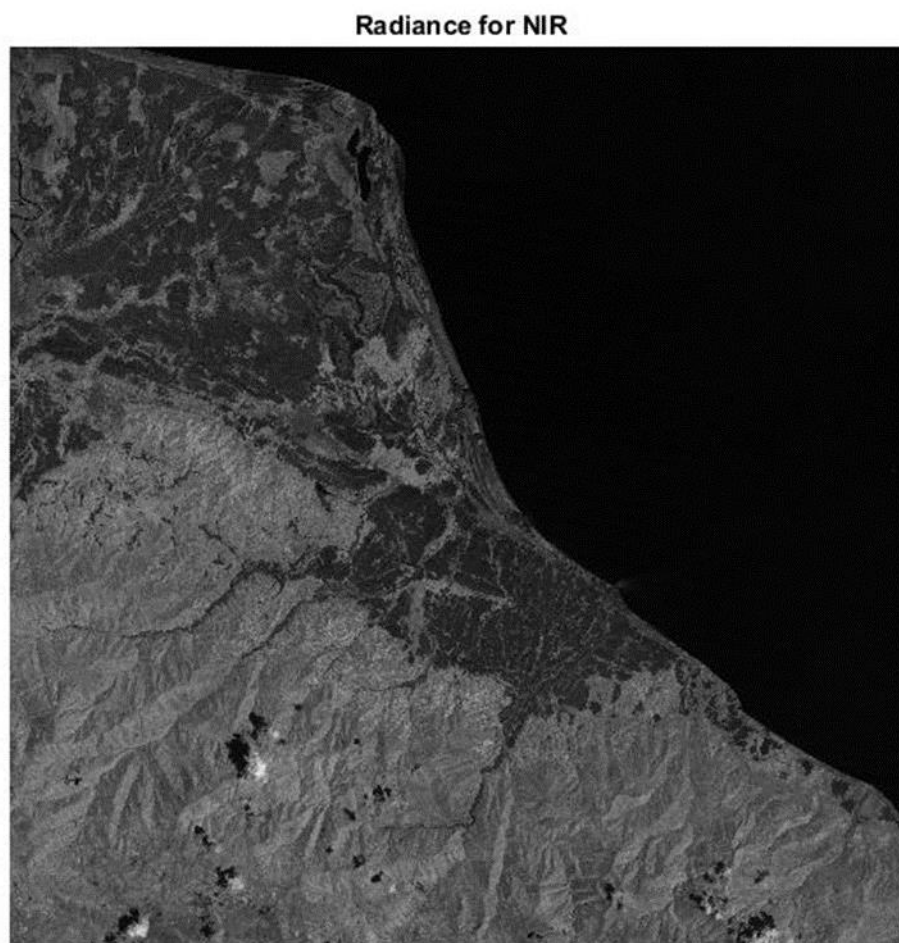
الف) عنوان آیتم شماره ۱۰ (محاسبه و تهیه نقشه NDWI برای تصویر سنجنده TM ماهواره Landsat)

برای شناسایی پهنه‌هایی آبی به کار می‌رود. ابتدا رادینانس باندهای سبز و مادون قرمز نزدیک را بدست آورده و در فرمول شاخص جایگذاری می‌کنیم.

ب) دو باند ورودی، به این محل کپی شود.

باند ۲





(ج) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

fileID_2=fopen('b2.dat','r');    %% G
fileID_4=fopen('b4.dat','r');    %% NIR

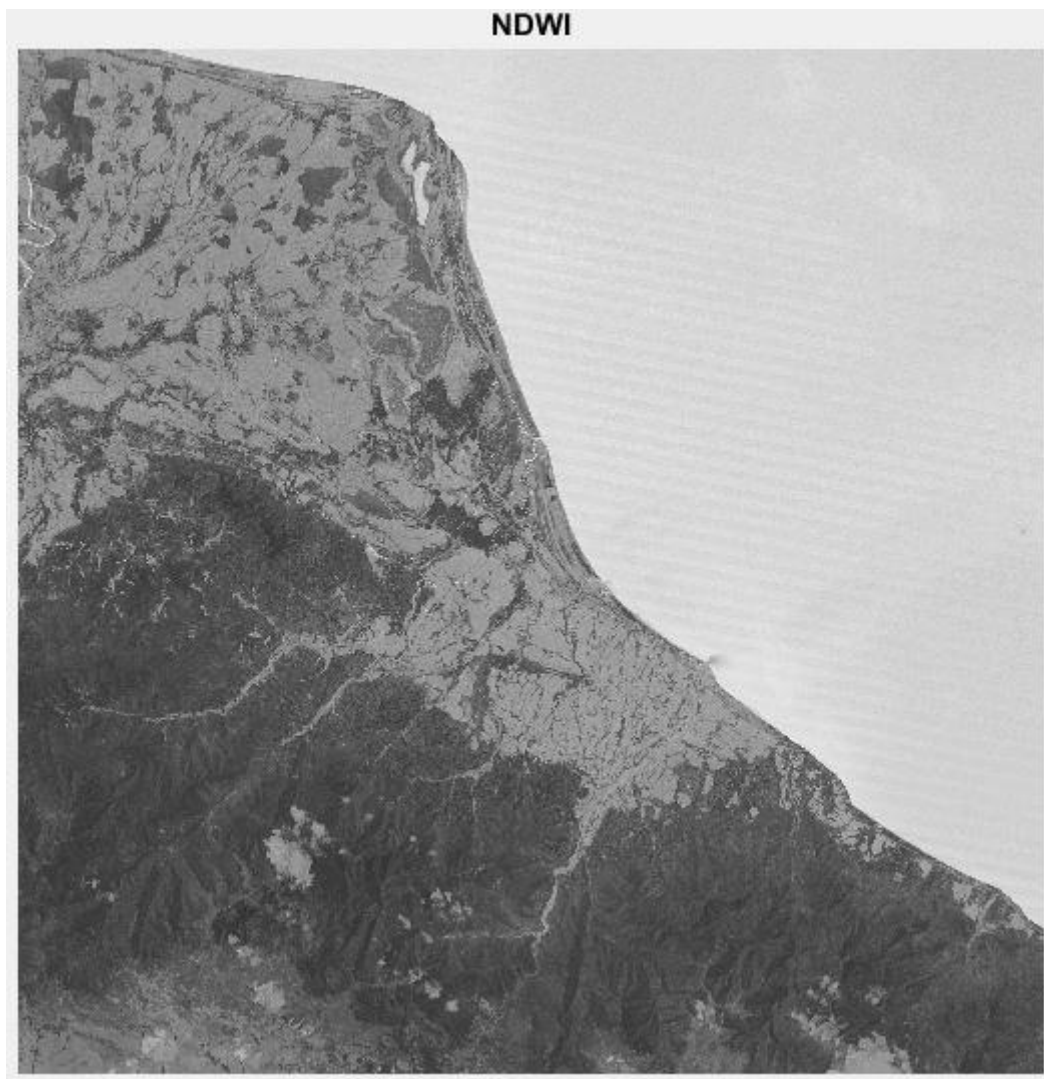
numbers_2 = fread(fileID_2,inf,'*uint8');
numbers_4 = fread(fileID_4,inf,'*uint8');

Lmax_2=365;Lmin_2=-2.84;
L_2=((Lmax_2-Lmin_2)/255)*numbers_2+Lmin_2;
kernel_2=double(vec2mat(L_2,2048));

Lmax_4=221;Lmin_4=-1.51;
```

```
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;  
kernel_4=double(vec2mat(L_4,2048));  
  
NDWI=(kernel_2-kernel_4)./(kernel_2+kernel_4);  
NDWI=(NDWI+1)*127;  
imshow(uint8(NDWI))  
title('NDWI')
```

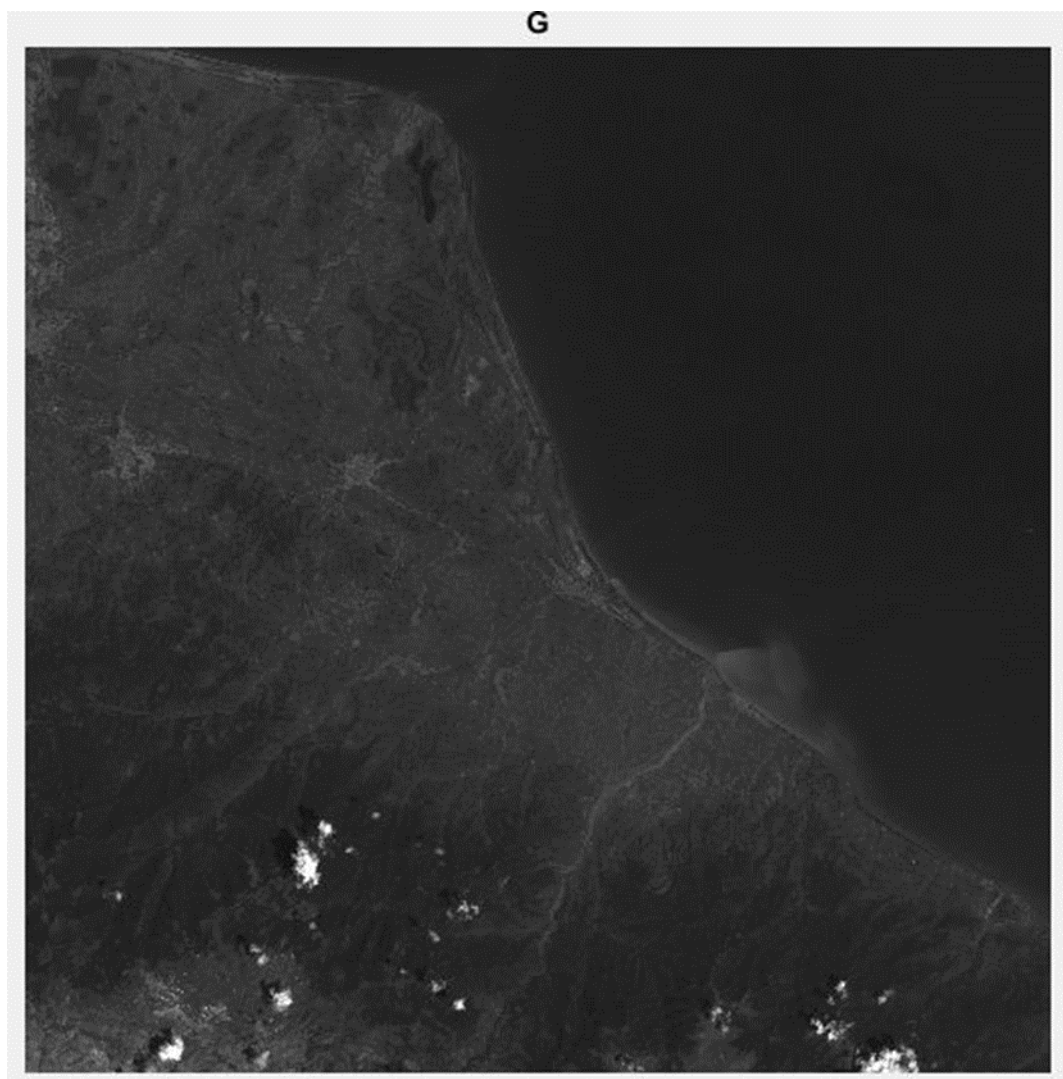
د) خروجی یعنی تصویر NDWI به این محل کپی شود.



الف) عنوان آیتم شماره ۱۱ (محاسبه و تهیه نقشه AWEI برای تصویر سنجنده TM ماهواره Landsat) رادیانس باند های ورودی را در فرمول جایگذاری کرده و شاخص AWEI که برای شناسایی دقیق تر پهنه های آبی است را بدست می آوریم.

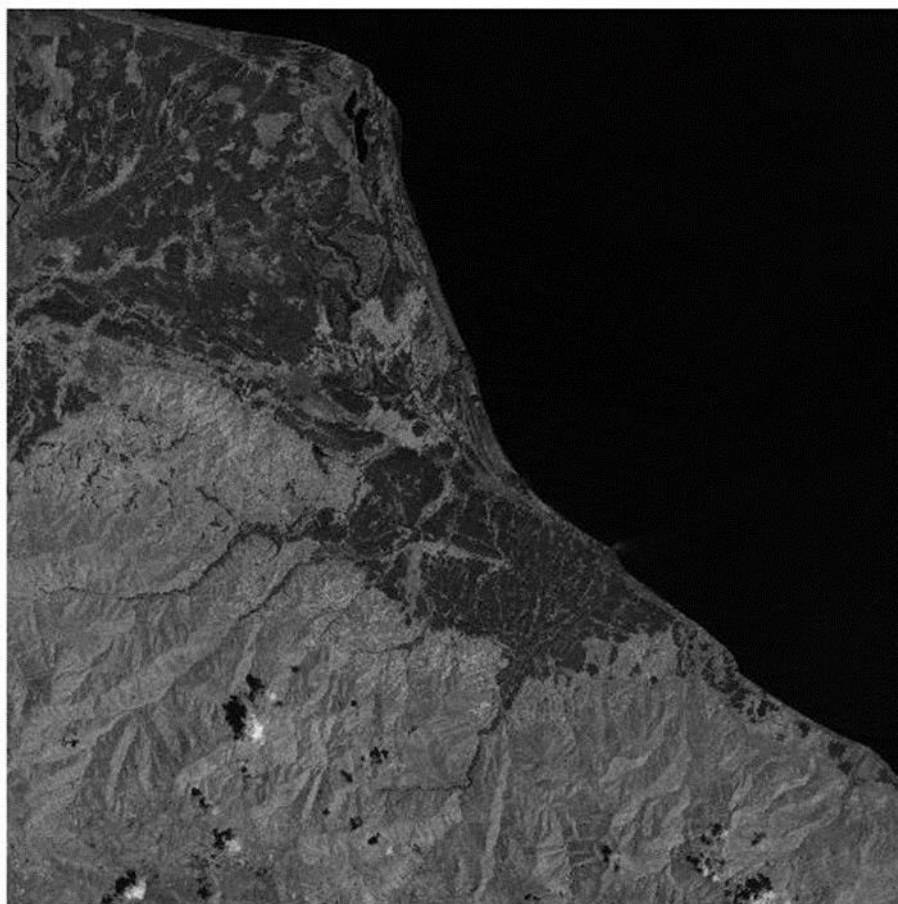
ب) چهار باند ورودی، به این محل کپی شود.

باند ۲



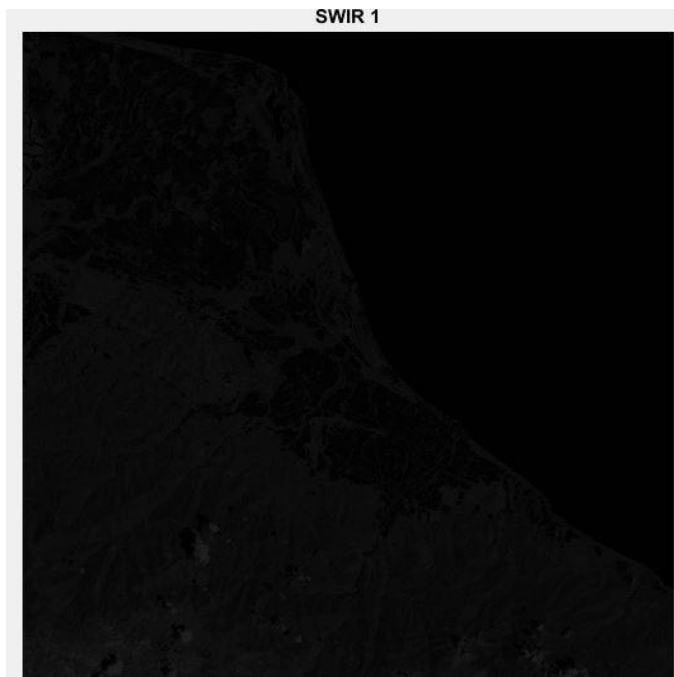
باند ۴

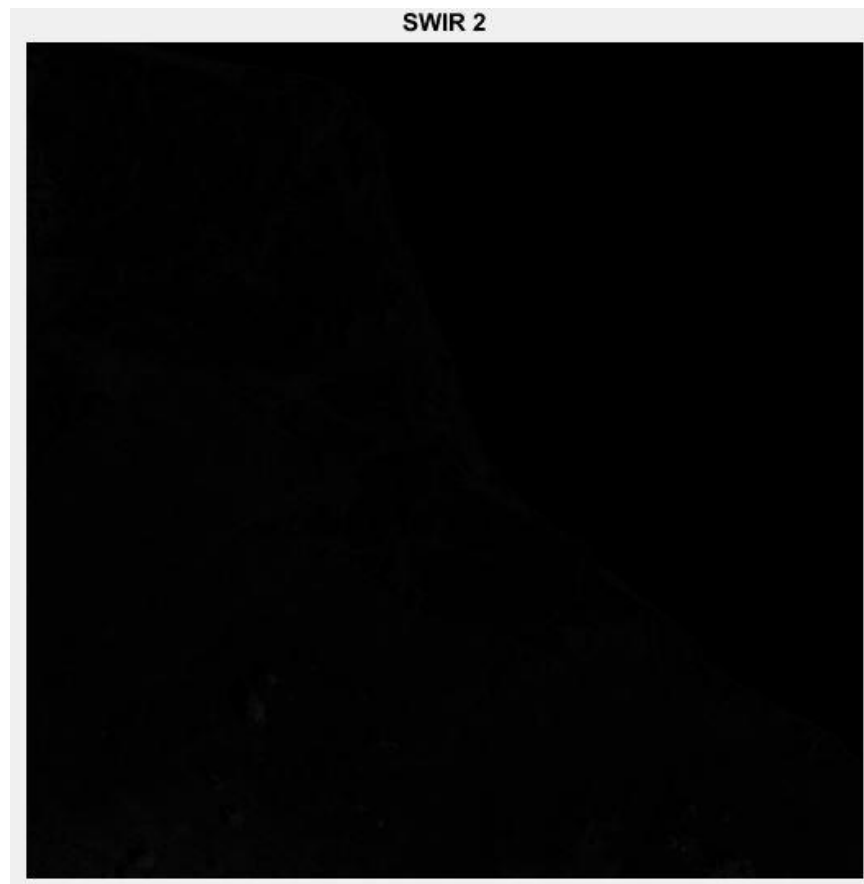
Radiance for NIR



باند ۵

SWIR 1





ج) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

fileID_2=fopen('b2.dat','r');    %% G
fileID_4=fopen('b4.dat','r');    %% NIR
fileID_5=fopen('b5.dat','r');    %% SWIR1
fileID_7=fopen('b7.dat','r');    %% SWIR2

numbers_2 = fread(fileID_2,inf,'*uint8');
numbers_4 = fread(fileID_4,inf,'*uint8');
numbers_5 = fread(fileID_5,inf,'*uint8');
numbers_7 = fread(fileID_7,inf,'*uint8');

Lmax_2=365;Lmin_2=-2.84;
L_2=((Lmax_2-Lmin_2)/255)*numbers_2+Lmin_2;
kernel_2=double(vec2mat(L_2,2048));

Lmax_4=221;Lmin_4=-1.51;
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;
```

```

kernel_4=double(vec2mat(L_4,2048));

Lmax_5=30.2;Lmin_5=-0.37;
L_5=((Lmax_5-Lmin_5)/255)*numbers_5+Lmin_5;
kernel_5=double(vec2mat(L_5,2048));

Lmax_7=16.5;Lmin_7=-0.15;
L_7=((Lmax_7-Lmin_7)/255)*numbers_7+Lmin_7;
kernel_7=double(vec2mat(L_7,2048));

AWEI=(4*(kernel_2-kernel_5))-0.25*kernel_4+2.75*kernel_7;

imshow(uint8(AWEI))
title('AWEI')

```

د) خروجی یعنی تصویر AWEI به این محل کپی شود.



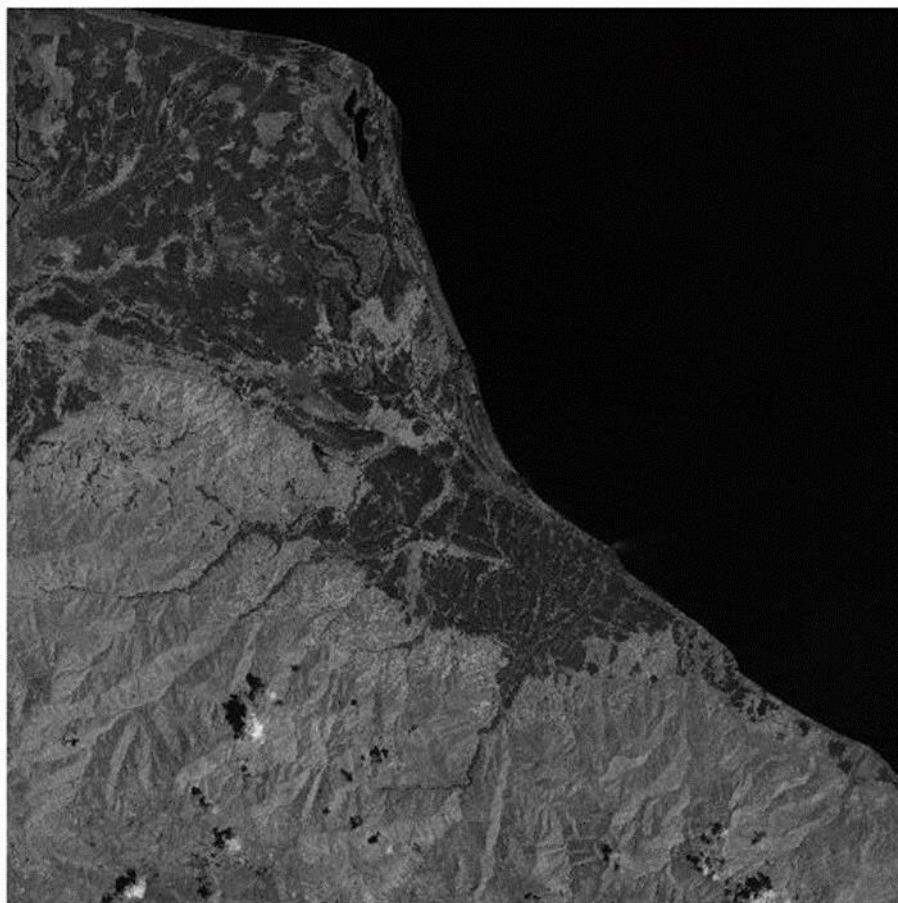
الف) عنوان آیتم شماره ۱۲ (محاسبه و تهیه نقشه NDMI برای تصویر سنجنده TM ماهواره Landsat)

ماتریس های رادیانس باند های ورودی را در داخل فرمول شاخص جایگذاری می کنیم.

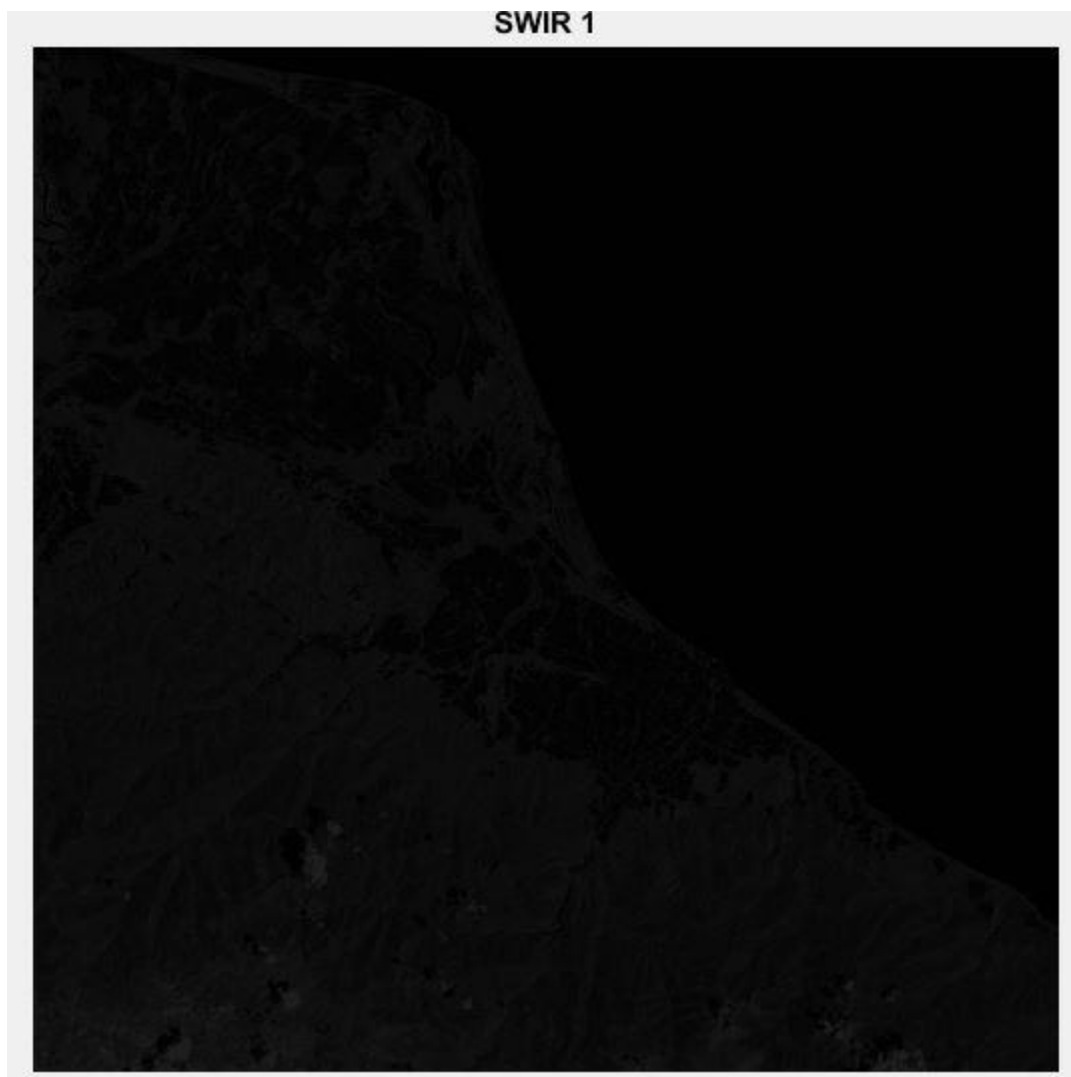
ب) دو باند ورودی، به این محل کپی شود.

باند ۴

Radiance for NIR



باند ۵



ج) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

fileID_4=fopen('b4.dat','r');    %% NIR
fileID_5=fopen('b5.dat','r');    %% SWIR

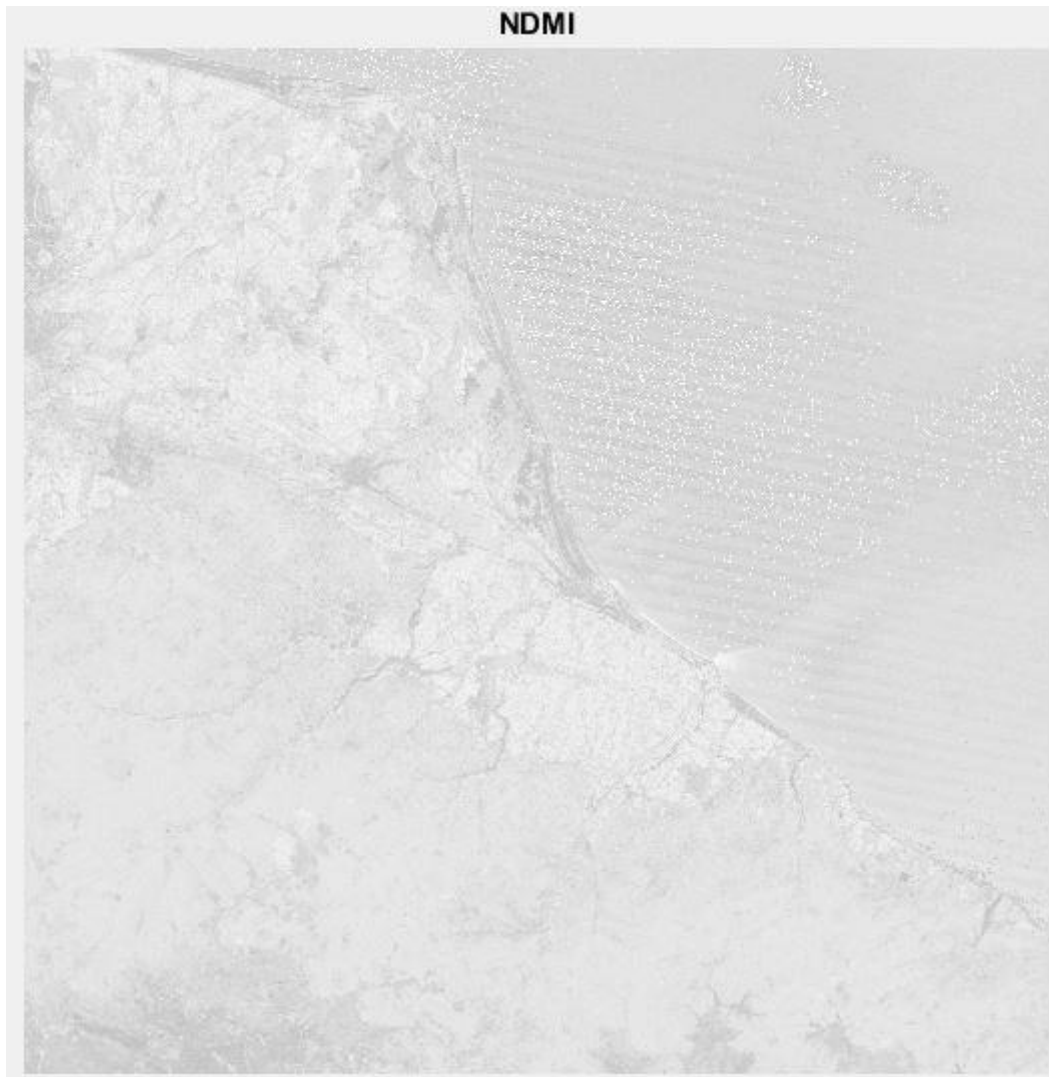
numbers_4 = fread(fileID_4,inf,'*uint8');
numbers_5 = fread(fileID_5,inf,'*uint8');

Lmax_4=221;Lmin_4=-1.51;
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;
kernel_4=double(vec2mat(L_4,2048));

Lmax_5=30.2;Lmin_5=-0.37;
L_5=((Lmax_5-Lmin_5)/255)*numbers_5+Lmin_5;
kernel_5=double(vec2mat(L_5,2048));
```

```
NDMI=(kernel_4-kernel_5)./(kernel_4+kernel_5);  
NDMI=(NDMI+1)*127;  
imshow(uint8(NDMI))  
title('NDMI')
```

د) خروجی یعنی تصویر NDMI به این محل کپی شود.

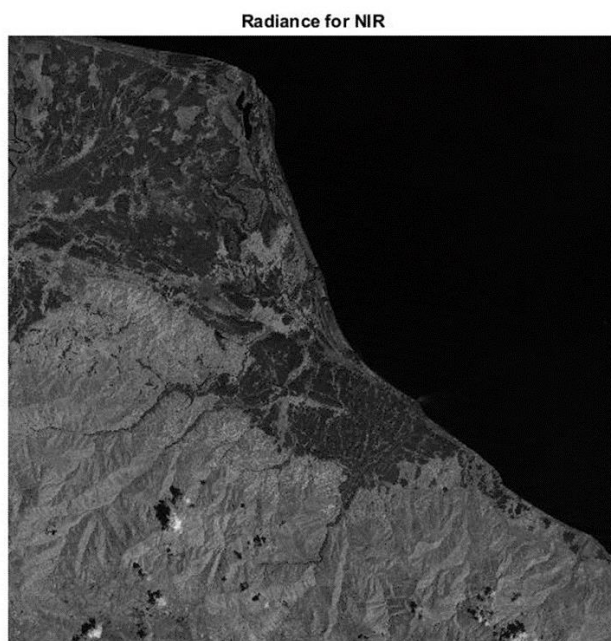


الف) عنوان آیتم شماره ۱۳ (محاسبه و تهیه نقشه NDBI برای تصویر سنجنده TM ماهواره Landsat)

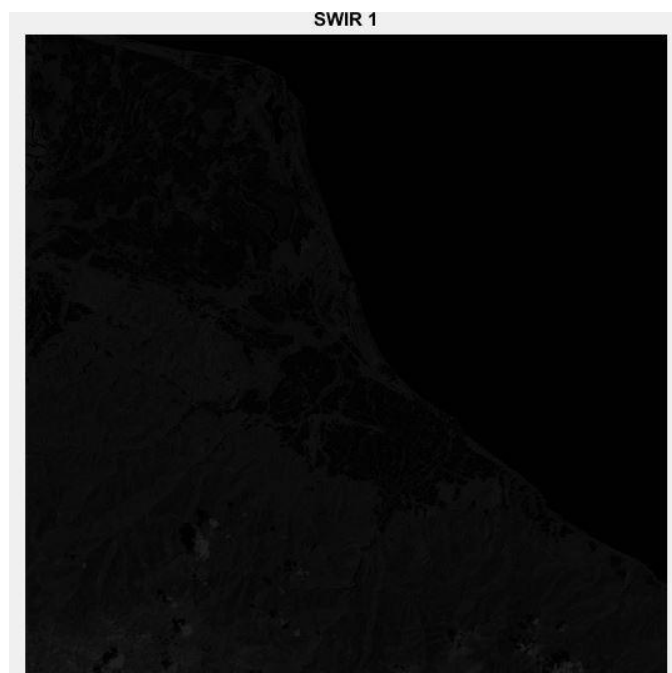
ماتریس های رادیانس باند های ورودی را در داخل فرمول شاخص جایگذاری می کنیم.

ب) دو باند ورودی، به این محل کپی شود.

باند ۴



باند ۵



ج) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

fileID_4=fopen('b4.dat','r');    %% NIR
fileID_5=fopen('b5.dat','r');    %% SWIR

numbers_4 = fread(fileID_4,inf,'*uint8');
numbers_5 = fread(fileID_5,inf,'*uint8');

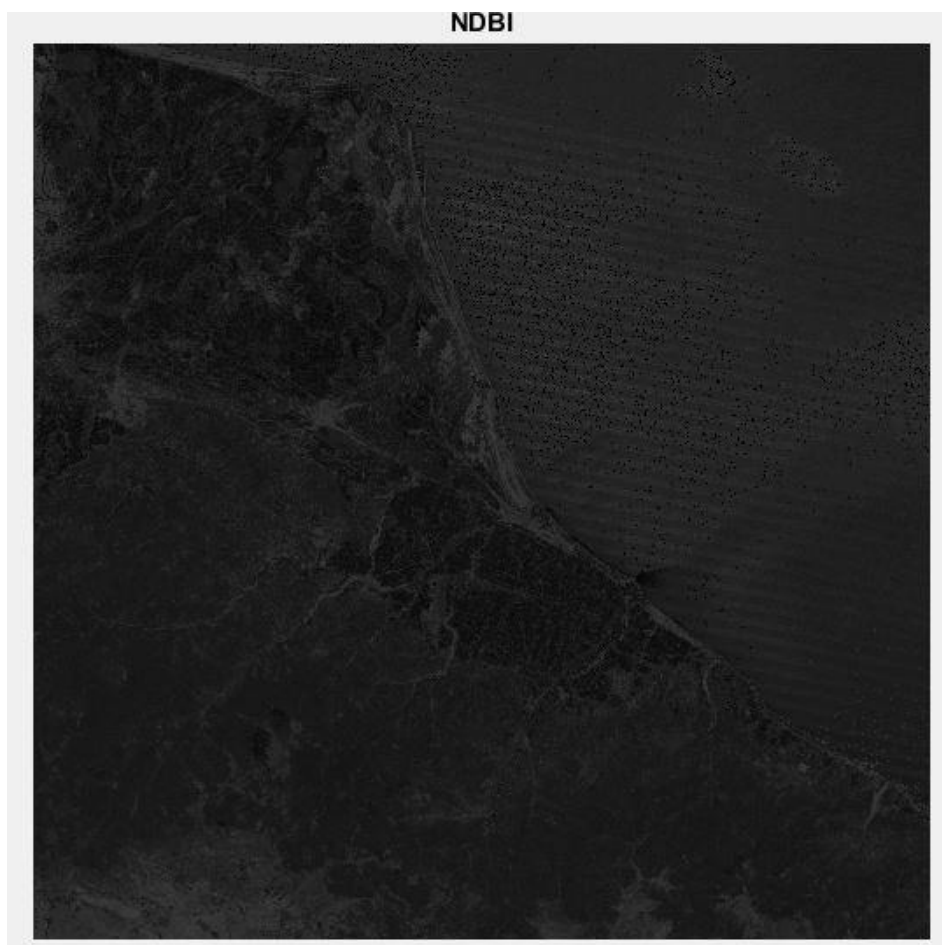
Lmax_4=221;Lmin_4=-1.51;
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;
kernel_4=double(vec2mat(L_4,2048));

Lmax_5=30.2;Lmin_5=-0.37;
L_5=((Lmax_5-Lmin_5)/255)*numbers_5+Lmin_5;
kernel_5=double(vec2mat(L_5,2048));

NDBI=(kernel_5-kernel_4)./(kernel_5+kernel_4);
NDBI=(NDBI+1)*127;

imshow(uint8(NDBI));
title('NDBI')
```

د) خروجی یعنی تصویر NDBI به این محل کپی شود.

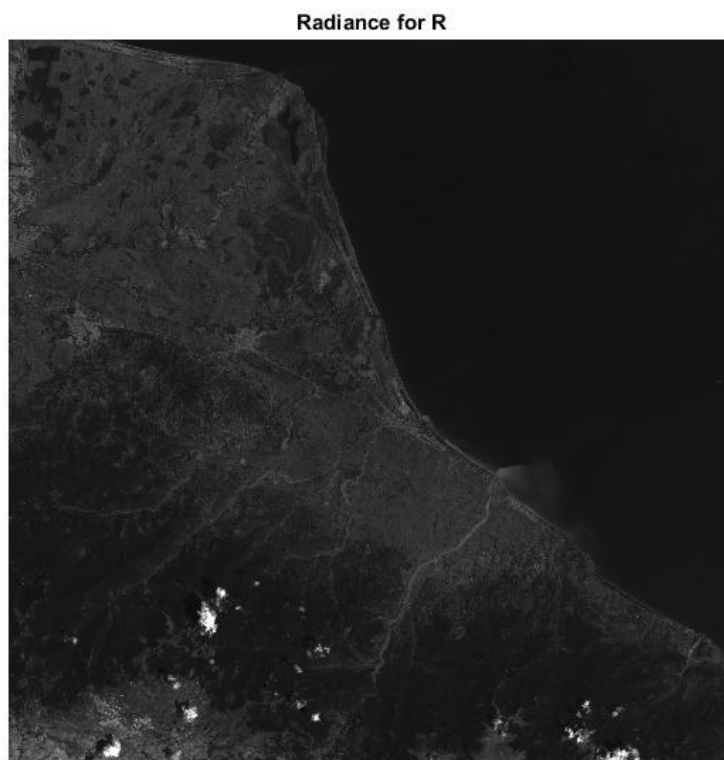


- محاسبه و تهیه ی نقشه رادیانس باند ۶ تصویر سنجنده Landsat5-TM و نمایش آن
- محاسبه و تهیه ی نقشه ی ضریب گسیل (ϵ) از طریق VIEM
- محاسبه و تهیه ی نقشه دما (LST) از تصویر سنجنده ی TM ماهواره ی LANDSAT و نمایش آن
- نمایش مقادیر دمای چند پیکسل

باند های ورودی را فراخوانی می کنیم. رادیانس تصاویر را بدست می آوریم. در روش VIEM از شاخص NDVI برای تخمین ضریب گسیل استفاده می کنیم. مطابق رابطه ی موجود در انتهای فصل پنجم جزوه، میتوانیم به ازای NDVI های مختلف، ضریب گسیل متفاوت را نسبت دهیم. سپس به کمک ضرایب گسیل، از فرمول استفاده کرده و دما را بدست می آوریم. همچنین مقدار پیکسل به ازای سه منطقه ی آبی، دارای پوشش گیاهی و زمین بایر آورده شده است.

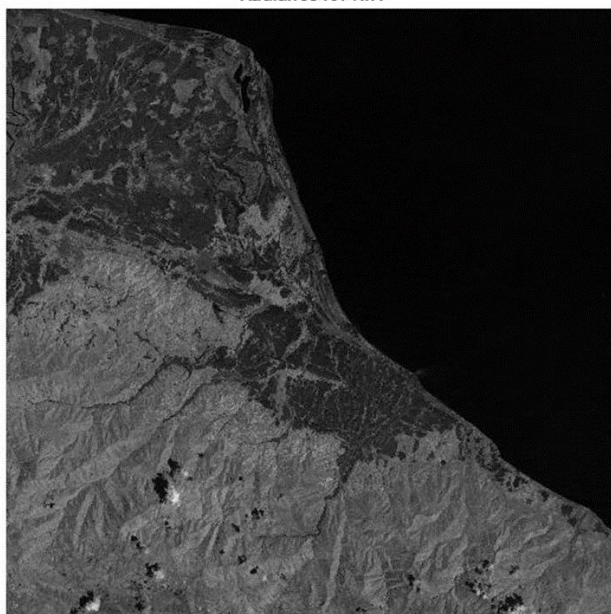
ب) باند ورودی، به این محل کپی شود.

باند ۳

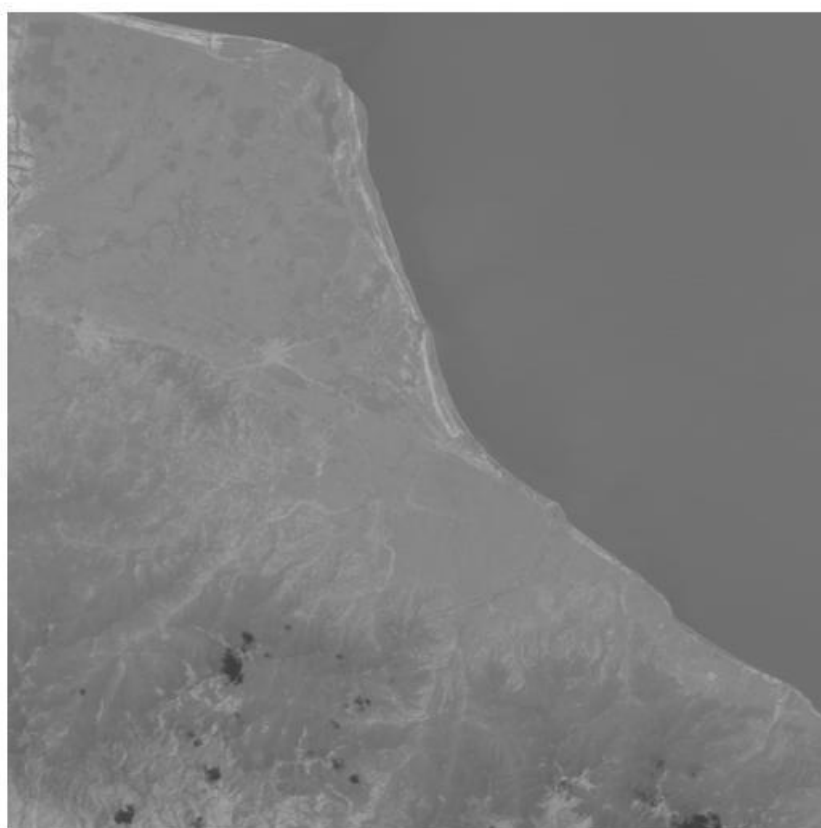


باند ۴

Radiance for NIR



باند ۶



(ج) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

fileID_6=fopen('b6.dat','r');
numbers_6= fread(fileID_6,inf,'*uint8');
Lmax=15.303;Lmin=1.2378;
L_6=((Lmax-Lmin)/255)*numbers_6+Lmin;
kernel_6=(vec2mat(L_6,2048));
imshow((kernel_6));title('Radiance of Thermal')

fileID_3=fopen('b3.dat','r');    %% R
fileID_4=fopen('b4.dat','r');    %% NIR

numbers_3 = fread(fileID_3,inf,'*uint8');
numbers_4 = fread(fileID_4,inf,'*uint8');

Lmax_3=264;Lmin_3=-1.17;
L_3=((Lmax_3-Lmin_3)/255)*numbers_3+Lmin_3;
kernel_3=double(vec2mat(L_3,2048));

Lmax_4=221;Lmin_4=-1.51;
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;
kernel_4=double(vec2mat(L_4,2048));

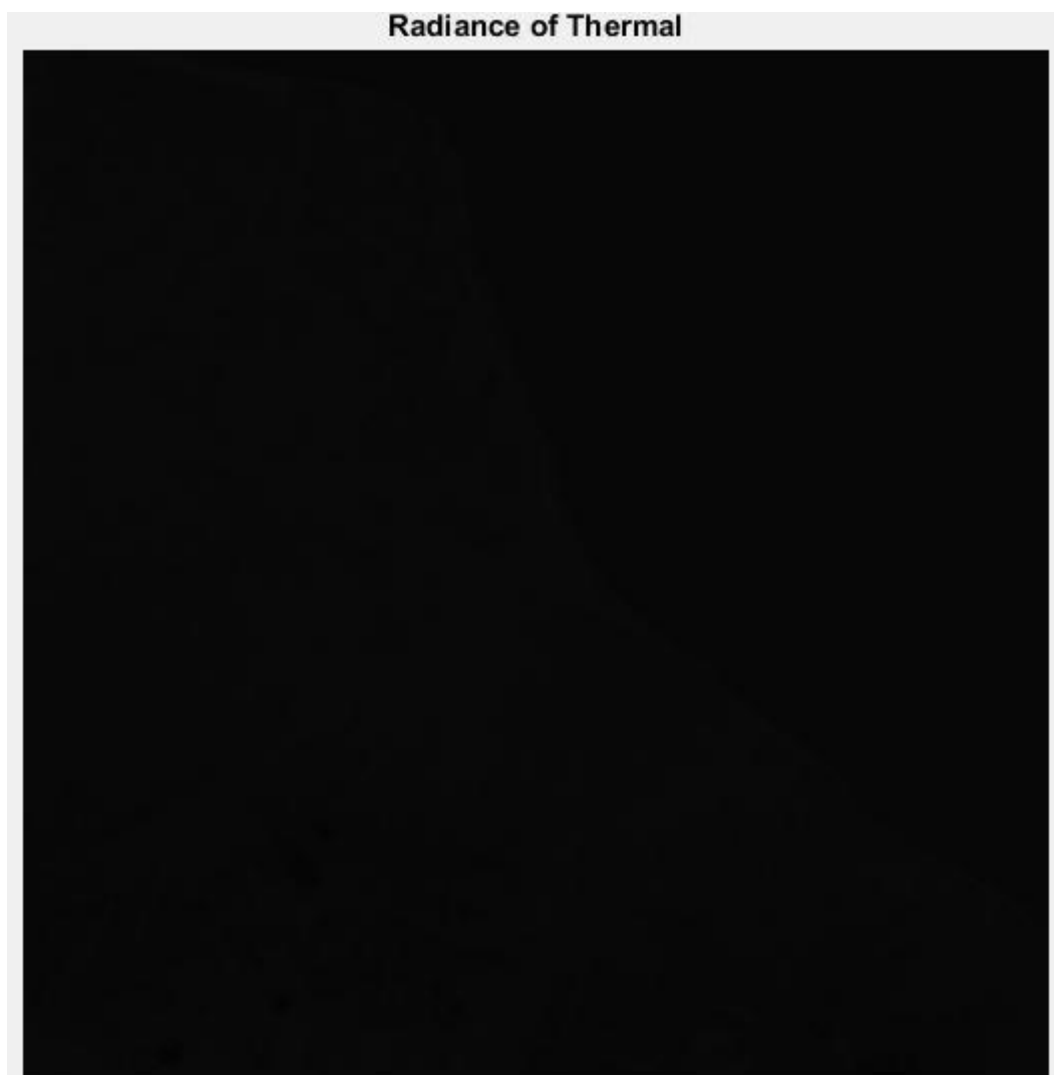
NDVI=(kernel_4-kernel_3)./(kernel_4+kernel_3);
NDVI_new=(NDVI+1).*127;

for i=1:2048
    for j=1:2048
        if NDVI(i,j) >=0.5
            eps(i,j)=0.99;
        elseif NDVI(i,j) <0.5 & NDVI(i,j) >=0.2
            eps(i,j)=0.986+(0.004*((NDVI(i,j)-0.2/0.3)^2));
        elseif NDVI(i,j) <0.2
            eps(i,j)=0.97;
        elseif NDVI(i,j) <0
            eps(i,j)=1;
        end
    end
end
figure
imshow(eps);title('Emissivity')

kernel_6= double(kernel_6);
K1=607.76 ; K2=1260.56;
LST=K2./ (log((K1*eps)./kernel_6)+1));    %% Kelvin
LST=LST-273;
max=max(max(LST)); min=min(min(LST));
LST=(254/(max-min))*(LST-min);
figure
imshow(uint8(LST));title('Temperature map')

fprintf('Vegetation: %d\n',LST(1803,1278));
fprintf('Water: %d\n',LST(1227,1926));
fprintf('Bare Ground: %d\n',LST(637,722));
```

د) خروجی تصویر رادیانس باند ۶ به این محل کپی شود.

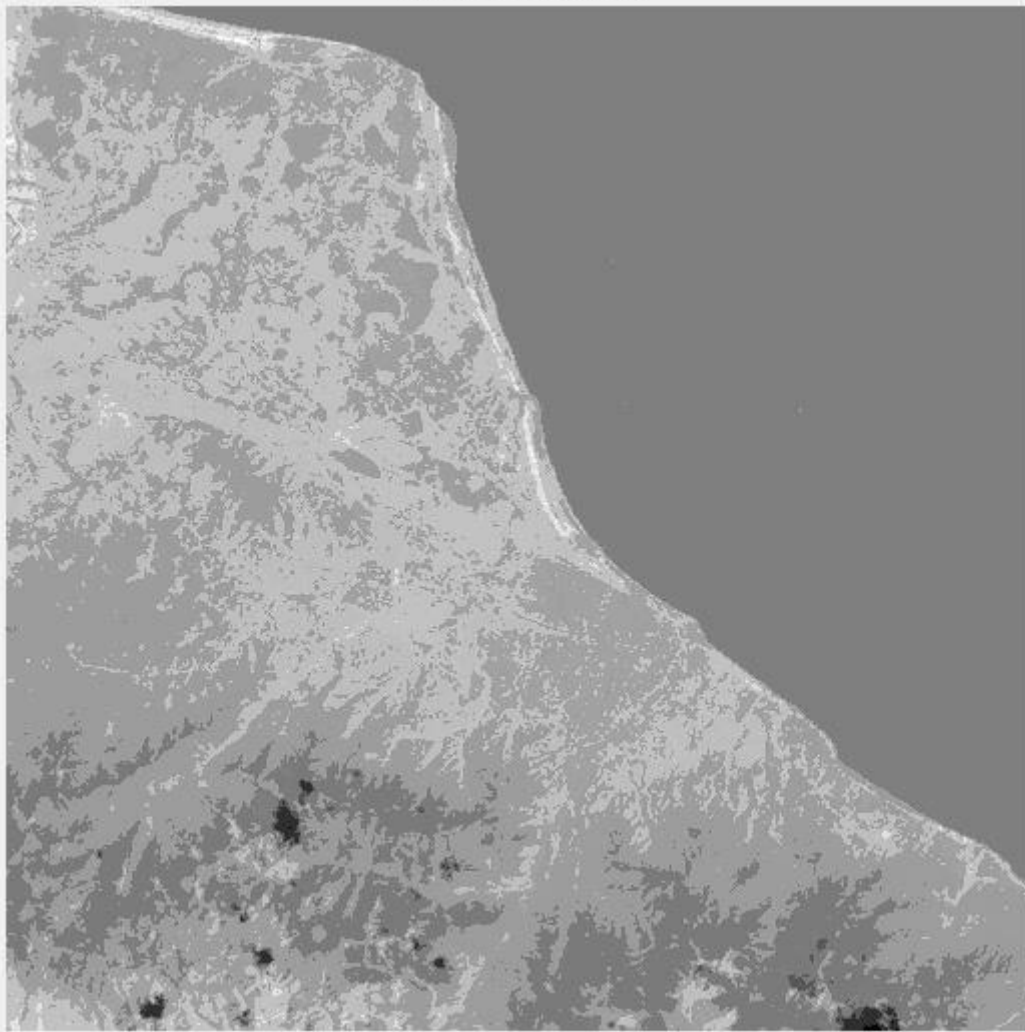


ه) خروجی تصویر نقشه ی ضریب گسیل به این محل کپی شود.



ی) خروجی تصویر نقشه ی دما به این محل کپی شود.

Temperature map



و) مقادیر دمای پیکسل های مناطق مختلف به این محل کپی شود.

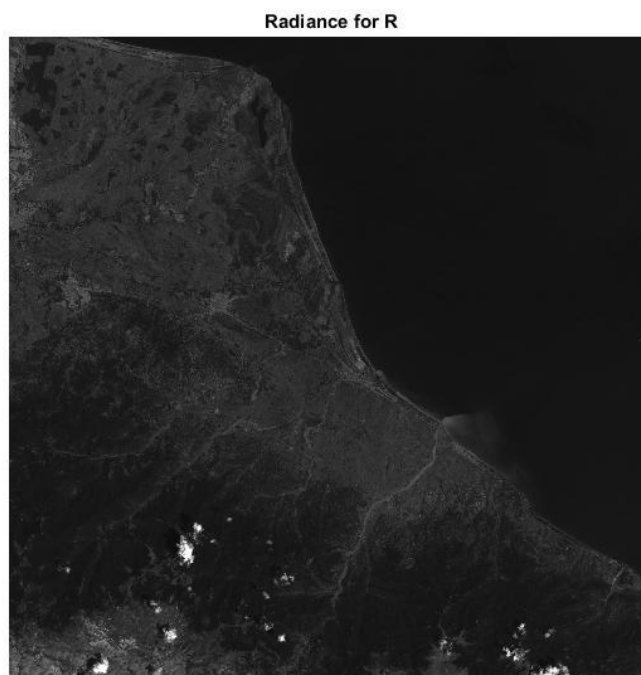
Vegetation: 1.216038e+02
Water: 1.267666e+02
Bare Ground: 1.942857e+02

الف) عنوان آیتم شماره ۱۵

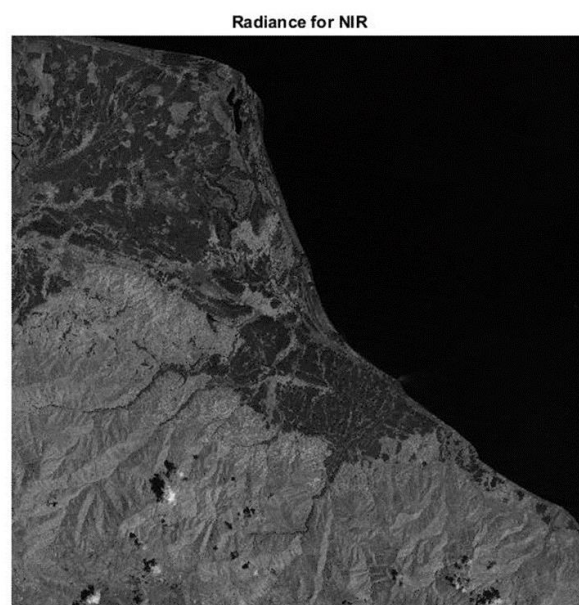
- محاسبه و تهیه ی نقشه ی ضریب گسیل (ϵ) از طریق CBEM
 - محاسبه و تهیه ی نقشه دما (LST) از تصویر سنجنده ی TM ماهواره ی LANDSAT و نمایش آن
 - نمایش مقادیر دمای چند پیکسل
- باند های ورودی را فراخوانی می کنیم. رادیانس تصاویر را بدست می آوریم. در روش CBEM نیاز داریم بدانیم، پوشش مناطق مختلف چگونه است. پس برای این کار از شاخص NDVI کمک می گیریم. مطابق جدول موجود در فصل اول جزوه، میتوانیم مناطق مختلف را شناسایی کنیم. سپس از آخرین جدول فصل ۱ برای پدیده های مختلف، ضرایب گسیل متفاوت را در نظر می گیریم.
- پدیده هایی که برای تصویر سنجنده ی TM در این آیتم در نظر گرفته شده است، عبارتند از: بوته زار (متراکم)، بوته زار (نیکه متراکم)، علفزار (خیلی کوتاه)، خاک خشک، ابر و آب.
- سپس به کمک ضرایب گسیل، از فرمول استفاده کرده و دما را بدست می آوریم. همچنین مقدار پیکسل به ازای سه منطقه ی آبی، دارای پوشش گیاهی و زمین بایر آورده شده است.

ب) باند ورودی، به این محل کپی شود.

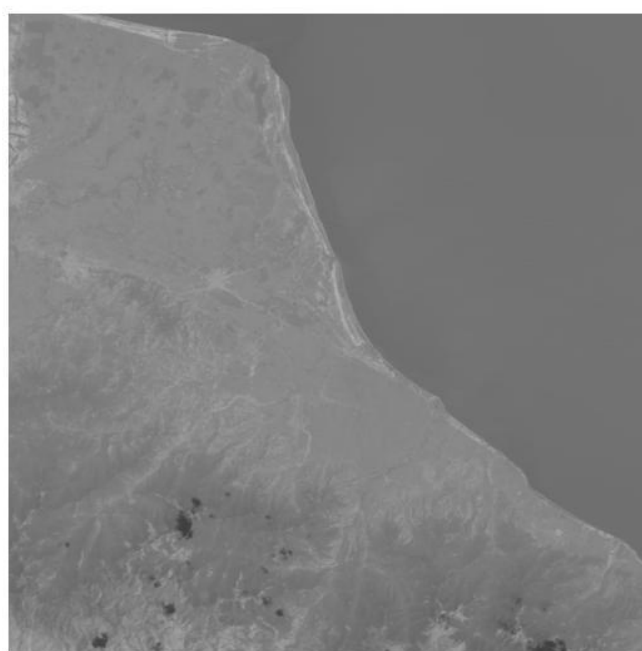
باند ۳



باند ۴



باند ۶



(ج) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

fileID_3=fopen('b3.dat','r');    %% R
fileID_4=fopen('b4.dat','r');    %% NIR

numbers_3 = fread(fileID_3,inf,'*uint8');
numbers_4 = fread(fileID_4,inf,'*uint8');

Lmax_3=264;Lmin_3=-1.17;
G=(Lmax_3-Lmin_3)/255;
L_3=((Lmax_3-Lmin_3)/255)*numbers_3+Lmin_3;

Lmax_4=221;Lmin_4=-1.51;
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;
L_4=double(L_4);
L_3=double(L_3);
NDVI=(L_4-L_3)./(L_4+L_3);
NDVI=double(vec2mat(NDVI,2048));

for i=1:2048
    for j=1:2048
        if NDVI(i,j)>=0.5
            eps(i,j)=0.986;
        elseif NDVI(i,j)>=0.14 && NDVI(i,j)<0.5
            eps(i,j)=0.976;
        elseif NDVI(i,j)>=0.09 && NDVI(i,j)<0.14
            eps(i,j)=0.979;
        elseif NDVI(i,j)>=0.025 && NDVI(i,j)<0.09
            eps(i,j,1)=0.92;
        elseif NDVI(i,j)>=0.002 && NDVI(i,j)<0.025
            eps(i,j)=0.97;
        elseif NDVI(i,j)<0.002
            eps(i,j)=0.99;
        end
    end
end
imshow(eps);title('Emissivity')

fileID_6=fopen('b6.dat','r');
numbers_6= fread(fileID_6,inf,'*uint8');
Lmax=15.303;Lmin=1.2378;
L_6=((Lmax-Lmin)/255)*numbers_6+Lmin;
kernel_6=double(vec2mat(L_6,2048));

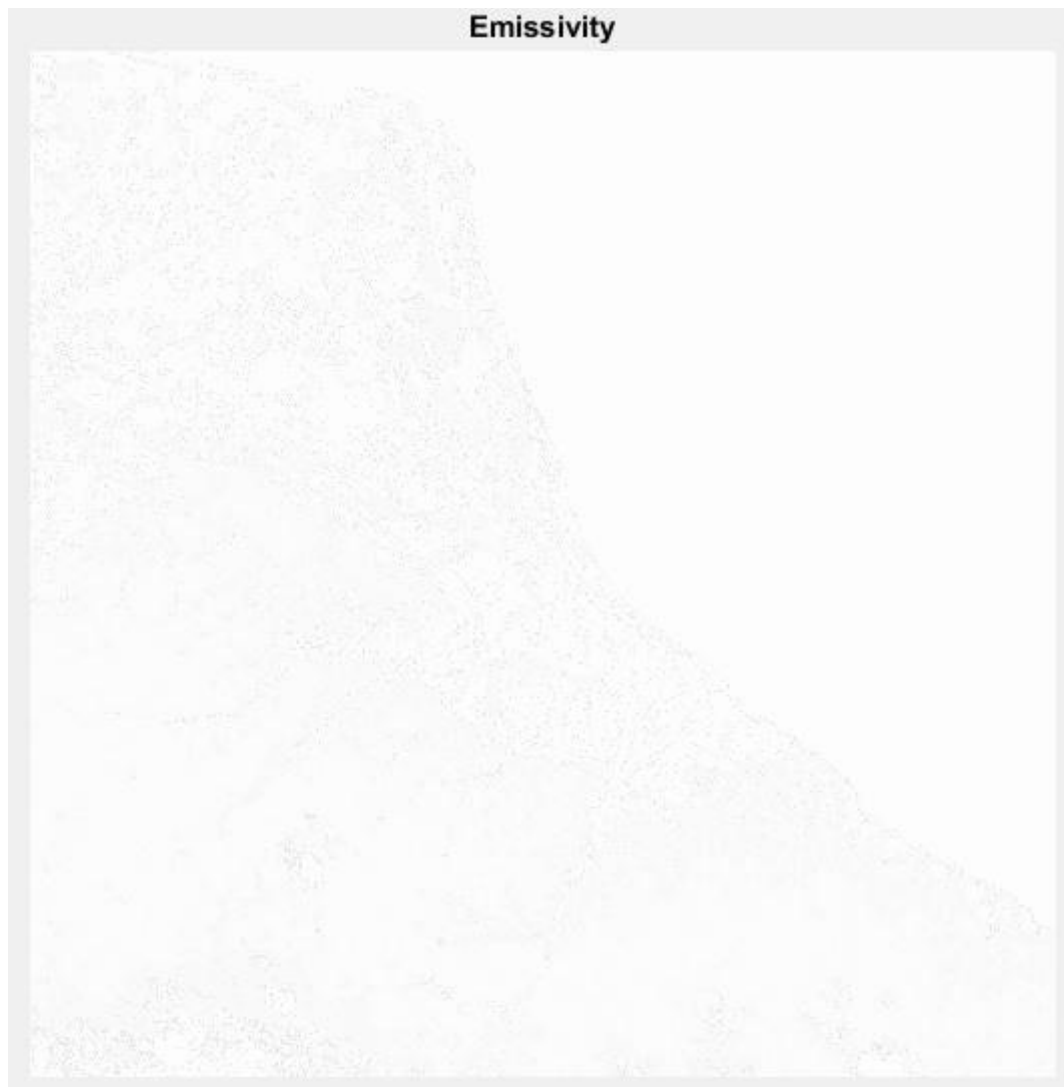
K1=607.76 ; K2=1260.56;
T=K2./(log(((K1*eps)./kernel_6)+1));
LST=K2./(log(((K1*eps)./kernel_6)+1));    %% Kelvin
LST=LST-273;
max=max(max(LST)); min=min(min(LST));
LST=(254/(max-min))*(LST-min);
figure
imshow(uint8(LST));title('Temperature map')

fprintf('Vegetation: %d\n',LST(1803,1278));
fprintf('Water: %d\n',LST(1227,1926));
```

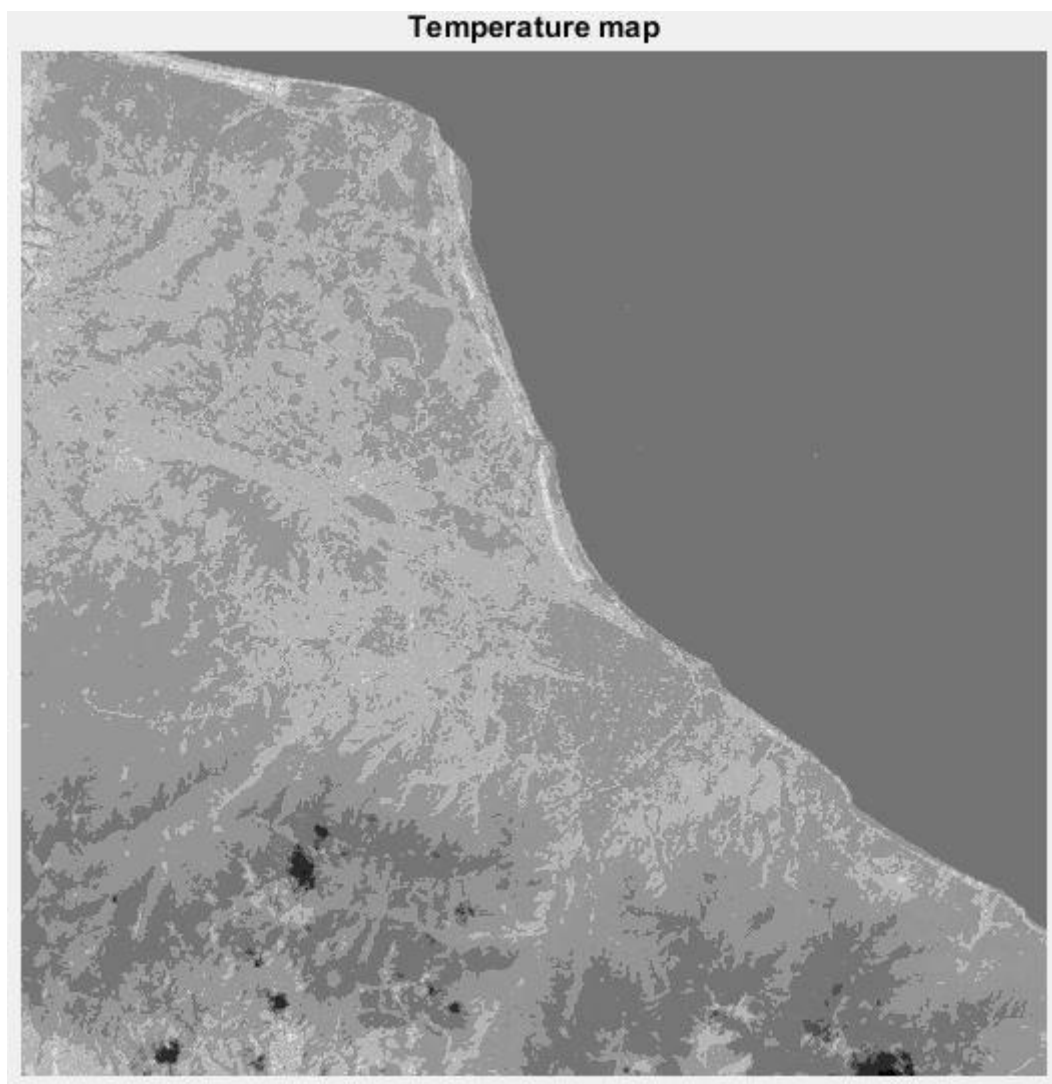


```
fprintf('Bare Ground: %d\n',LST(637,722));
```

د) خروجی تصویر نقشه ی ضریب گسیل به این محل کپی شود.



ه) خروجی تصویر نقشه ی دما به این محل کپی شود.



ی) مقادیر دمای پیکسل های مناطق مختلف به این محل کپی شود.

Vegetation: 1.171318e+02
Water: 1.161885e+02
Bare Ground: 1.780335e+02

الف) عنوان آیتم شماره ۱۶ (محاسبه مقدار ρ برای کلیه ی باند ها با فرض تاریخ 2011/06/01 برای تصویر Landsat و $\cos \theta_s$ برابر ۰.۷۷)

ابتدا کلیه ی باندها را به جز باند ۶ فراخوانی می کنیم و رادیانس هر باند را بدست می آوریم. در نهایت از

طریق رابطه ی $\rho = \frac{\pi \cdot L_{rad} \cdot d^2}{E_{sun} \cdot \cos \theta_s}$ مقدار ρ را بدست می آوریم.

ب) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;
fileID_1=fopen('b1.dat','r');
fileID_2=fopen('b2.dat','r');
fileID_3=fopen('b3.dat','r');
fileID_4=fopen('b4.dat','r');
fileID_5=fopen('b5.dat','r');
fileID_7=fopen('b7.dat','r');

numbers_1 = fread(fileID_1,inf,'*uint8');
numbers_2 = fread(fileID_2,inf,'*uint8');
numbers_3 = fread(fileID_3,inf,'*uint8');
numbers_4 = fread(fileID_4,inf,'*uint8');
numbers_5 = fread(fileID_5,inf,'*uint8');
numbers_7 = fread(fileID_7,inf,'*uint8');

Lmax_1=193;Lmin_1=-1.52;
L_1=((Lmax_1-Lmin_1)/255)*numbers_1+Lmin_1;
kernel_1=double(vec2mat(L_1,2048));

Lmax_2=365;Lmin_2=-2.84;
L_2=((Lmax_2-Lmin_2)/255)*numbers_2+Lmin_2;
kernel_2=double(vec2mat(L_2,2048));

Lmax_3=264;Lmin_3=-1.17;
L_3=((Lmax_3-Lmin_3)/255)*numbers_3+Lmin_3;
kernel_3=double(vec2mat(L_3,2048));

Lmax_4=221;Lmin_4=-1.51;
L_4=((Lmax_4-Lmin_4)/255)*numbers_4+Lmin_4;
kernel_4=double(vec2mat(L_4,2048));

Lmax_5=30.2;Lmin_5=-0.37;
L_5=((Lmax_5-Lmin_5)/255)*numbers_5+Lmin_5;
kernel_5=double(vec2mat(L_5,2048));







Lmax_7=16.5;Lmin_7=-0.15;
L_7=((Lmax_7-Lmin_7)/255)*numbers_7+Lmin_7;
kernel_7=double(vec2mat(L_7,2048));

E=[195.7;182.9;155.7;104.7;21.93;7.452];
d=1.01;
```

```
rho_1=(pi*kernel_1*d)/(E(1)*0.77);
rho_2=(pi*kernel_2*d)/(E(2)*0.77);
rho_3=(pi*kernel_3*d)/(E(3)*0.77);
rho_4=(pi*kernel_4*d)/(E(4)*0.77);
rho_5=(pi*kernel_5*d)/(E(5)*0.77);
rho_7=(pi*kernel_7*d)/(E(6)*0.77);
```

ج) خروجیها به این محل کپی شود.

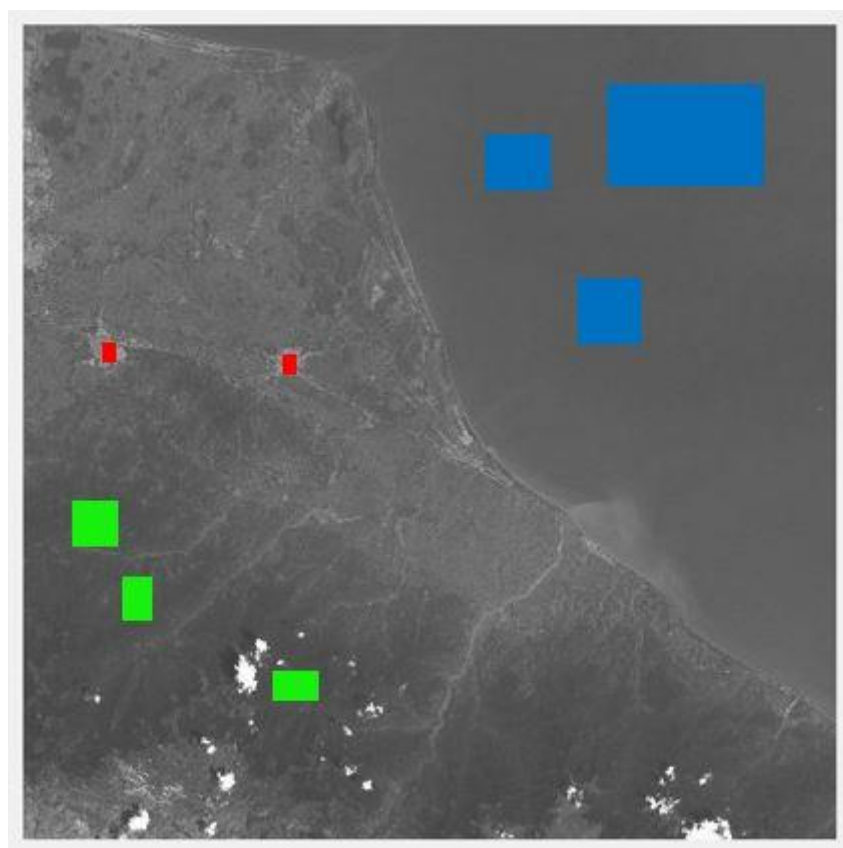
خروجی برای هر باند، یک ماتریس 2048×2048 است.

	rho_1	2048x2048 double
	rho_2	2048x2048 double
	rho_3	2048x2048 double
	rho_4	2048x2048 double
	rho_5	2048x2048 double
	rho_7	2048x2048 double

الف) عنوان آیتم شماره ۱۸ (انجام طبقه بندی متوازی السطوح بر روی تصویر سنجنده TM ماهواره Landsat)

ورودی برنامه، تمامی باندهای سنجنده TM لندست است. در طبقه بندی به روش متوازی السطوح، ابتدا داده های آموزشی را به کمک دستور rectangle تعیین می کنیم. سپس میانگین و انحراف معیار را برای هر کلاس یافته تا حدود بالا و پایین را برای هر کلاس به دست آوریم. سپس هر پیکسل تصویر را به کمک شروط منطقی (And) بررسی می کنیم تا اگر در سطوح مورد نظر را قرارگرفت مربوط به آن کلاس است. برای بررسی صحت کلی و ضریب کاپا هم یک نقطه را انتخاب می کنیم و یک مستطیل تشکیل می شود. سپس پیکسل های طبقه بندی شده ی داخل مستطیل مورد ارزیابی قرار خواهند گرفت. این طبقه بندی با ۳ کلاس منطقه شهری، پهنه ی آبی و پوشش گیاهی انجام شده است.

در طبقه بندی زیر، بسیاری از پیکسل ها، مشکی هستند؛ به این معناست که به هیچ یک از سه کلاس، تعلق ندارند. همین موضوع باعث شده تا صحت کلی و ضریب کاپا اعدادی کوچکی شوند.



ب) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```

clc;
clear;
close all;

B=zeros(2048,2048,7);
for i=1:7
    name=['b',num2str(i),'.dat'];
    band = fopen(name);
    A1 = fread(band);
    for j=1:2048
        B(j,1:2048,i)=A1((j-1)*2048+1:2048*j,1);
    end
    B(:,:,i)=mat2gray(B(:,:,i));
end
figure('Name','training data','NumberTitle','off');
imshow(B(:,:,1));
n=input('number of training test ROI:');
x1=zeros(2*n,1);
y1=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x1(2*i-1,1)=xx(1,1);
    y1(2*i-1,1)=yy(1,1);
    x1(2*i,1)=xx(2,1);
    y1(2*i,1)=yy(2,1);
    rectangle('position',[x1(2*i-1,1),y1(2*i-1,1),x1(2*i,1)-x1(2*i-1,1),y1(2*i,1)-y1(2*i-1,1)], 'facecolor',[0 0 1]);
end

x2=zeros(2*n,1);
y2=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x2(2*i-1,1)=xx(1,1);
    y2(2*i-1,1)=yy(1,1);
    x2(2*i,1)=xx(2,1);
    y2(2*i,1)=yy(2,1);
    rectangle('position',[x2(2*i-1,1),y2(2*i-1,1),x2(2*i,1)-x2(2*i-1,1),y2(2*i,1)-y2(2*i-1,1)], 'facecolor',[0 1 0]);
end
x3=zeros(2*n,1);
y3=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x3(2*i-1,1)=xx(1,1);
    y3(2*i-1,1)=yy(1,1);
    x3(2*i,1)=xx(2,1);
    y3(2*i,1)=yy(2,1);
    rectangle('position',[x3(2*i-1,1),y3(2*i-1,1),x3(2*i,1)-x3(2*i-1,1),y3(2*i,1)-y3(2*i-1,1)], 'facecolor',[1 0 0]);
end
x1=round(x1);
y1=round(y1);
x2=round(x2);
y2=round(y2);
x3=round(x3);
y3=round(y3);
%switch!
blanck=x1;
x1=y1;
y1=blanck;

```

```

blanck=x2;
x2=y2;
y2=blanck;
blanck=x3;
x3=y3;
y3=blanck;
mean1=zeros(n,1,7);
for j=1:n
    for i=1:7
        mean1(j,1,i)=mean(mean(B(x1(2*j-1):x1(2*j),y1(2*j-1):y1(2*j),i)));
    end
end
mean1=mean(mean1);
mean2=zeros(n,1,7);
for j=1:n
    for i=1:7
        mean2(j,1,i)=mean(mean(B(x2(2*j-1):x2(2*j),y2(2*j-1):y2(2*j),i)));
    end
end
mean2=mean(mean2);
mean3=zeros(n,1,7);
for j=1:n
    for i=1:7
        mean3(j,1,i)=mean(mean(B(x3(2*j-1):x3(2*j),y3(2*j-1):y3(2*j),i)));
    end
end
mean3=mean(mean3);

deviation1=zeros(n,1,7);
for j=1:n
    for i=1:7
        deviation1(j,1,i)=std(std(B(x1(2*j-1):x1(2*j),y1(2*j-1):y1(2*j),i)));
    end
end
deviation1=std(deviation1);
deviation2=zeros(n,1,7);
for j=1:n
    for i=1:7
        deviation2(j,1,i)=std(std(B(x1(2*j-1):x1(2*j),y1(2*j-1):y1(2*j),i)));
    end
end
deviation2=std(deviation2);
deviation3=zeros(n,1,7);
for j=1:n
    for i=1:7
        deviation3(j,1,i)=std(std(B(x1(2*j-1):x1(2*j),y1(2*j-1):y1(2*j),i)));
    end
end
deviation3=std(deviation3);
coef1=650;
coef2=900;
coef3=950;
% Finding Up and down:
up1=mean1+coef1*deviation1;
down1=mean1-coef1*deviation1;
up2=mean2+coef2*deviation2;
down2=mean2-coef2*deviation2;
up3=mean3+coef3*deviation3;

```



```

down3=mean3-coef3*deviation3;
A1=zeros(2048,2048,7);
for i=1:2048
    for j=1:2048
        for k=1:7
            if up1(1,1,k)>=B(i,j,k) && down1(1,1,k)<=B(i,j,k)
                A1(i,j,k)=1;
            end
        end
    end
end
A2=zeros(2048,2048,7);
for i=1:2048
    for j=1:2048
        for k=1:7
            if up2(1,1,k)>=B(i,j,k) && down2(1,1,k)<=B(i,j,k)
                A2(i,j,k)=1;
            end
        end
    end
end
A3=zeros(2048,2048,7);
for i=1:2048
    for j=1:2048
        for k=1:7
            if up3(1,1,k)>=B(i,j,k) && down3(1,1,k)<=B(i,j,k)
                A3(i,j,k)=1;
            end
        end
    end
end

classification=zeros(2048,2048,3);
for i=1:2048
    for j=1:2048
        if
            A1(i,j,1)+A1(i,j,2)+A1(i,j,3)+A1(i,j,4)+A1(i,j,5)+A1(i,j,6)+A1(i,j,7)==7
                classification(i,j,3)=255;
            end
        end
    end
end

for i=1:2048
    for j=1:2048
        if
            A2(i,j,1)+A2(i,j,2)+A2(i,j,3)+A2(i,j,4)+A2(i,j,5)+A2(i,j,6)+A2(i,j,7)==7
                classification(i,j,2)=255;
            end
        end
    end
end

for i=1:2048
    for j=1:2048
        if
            A3(i,j,1)+A3(i,j,2)+A3(i,j,3)+A3(i,j,4)+A3(i,j,5)+A3(i,j,6)+A3(i,j,7)==7
                classification(i,j,1)=255;
            end
        end
    end
end
imshow(classification);
[x,y]=ginput(2);

```

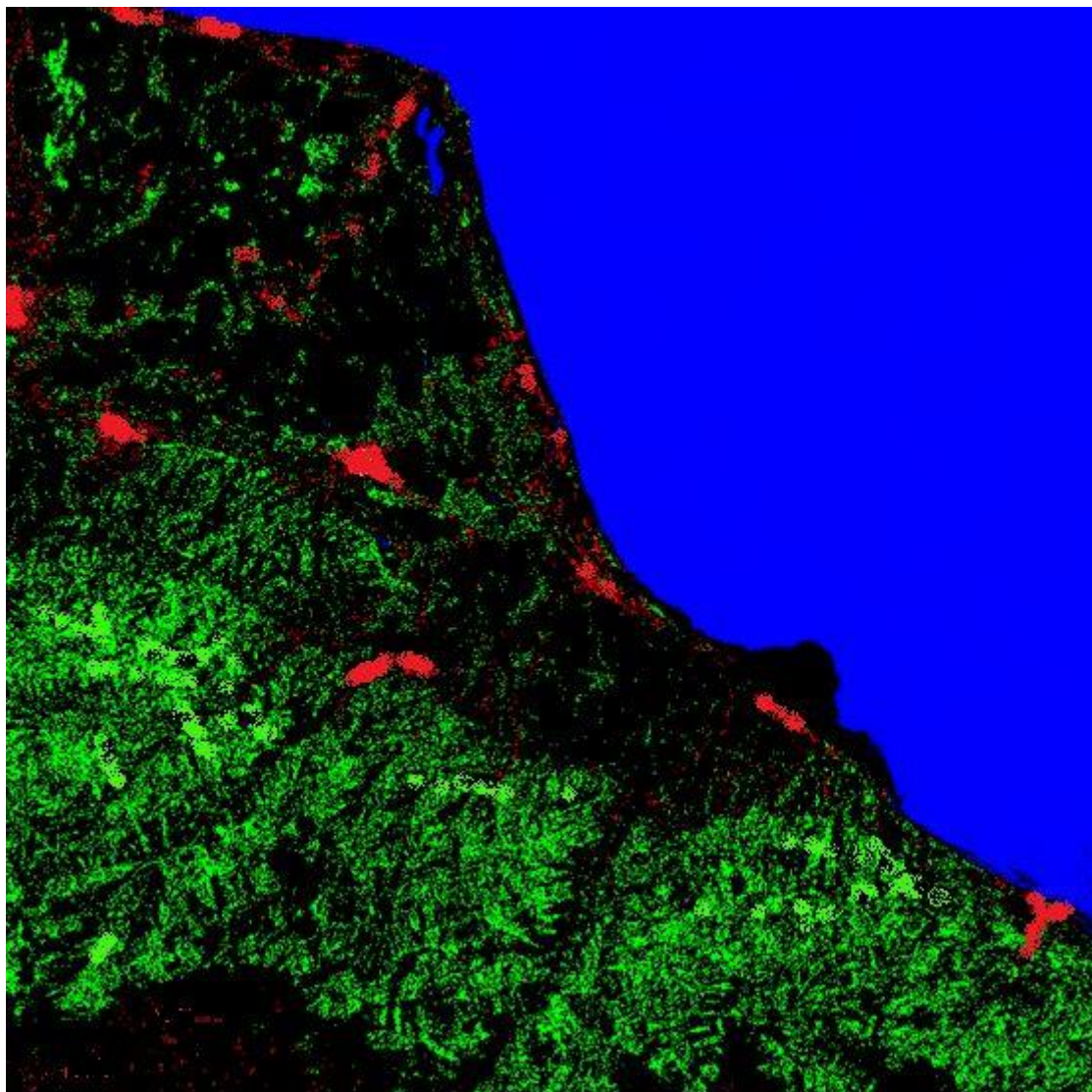
```

rectangle('position',[x(1,1),y(1,1),x(2,1)-x(1,1),y(2,1)-
y(1,1)], 'facecolor',[0 0 1]);
blanck=y;
y=x;
x=blanck;
test=classification(x(1,1):x(2,1),y(1,1):y(2,1),:);
imshow(test)
[x,y,Z]=size(test);

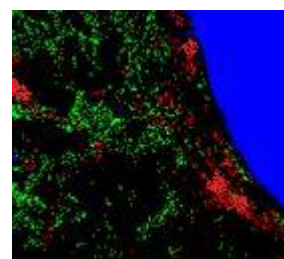
a=zeros(3,3);
n=x*y;
for i=1:x
    for j=1:y
        if test(i,j,3)==255
            a(1,1)=a(1,1)+1;
        end
    end
end
for i=1:x
    for j=1:y
        if test(i,j,2)==255
            a(2,2)=a(2,2)+1;
        end
    end
end
for i=1:x
    for j=1:y
        if test(i,j,1)==255
            a(3,3)=a(3,3)+1;
        end
    end
end
end
er(1,1)=input('error: Water Region :');
er(2,1)=input('error: Vegetation Region :');
er(3,1)=input('error: Urban Area :');
a(1,1)=a(1,1)-er(1,1);
a(2,2)=a(2,2)-er(2,1);
a(3,3)=a(3,3)-er(3,1);
sumd=a(1,1)+a(2,2)+a(3,3);
a(1,2)=input('error: Water Region/Vegetation Region:');
a(1,3)=input('error: Water Region/Urban Area:');
a(2,1)=input('error: Vegetation Region/Water Region:');
a(2,3)=input('error: Vegetation Region/Urban Area:');
a(3,1)=input('error: Urban Area/Water Region:');
a(3,2)=input('error: Urban Area/Vegetation Region:');
sum_row=sum(a,2);
sum_column=sum(a,1);
OA=sumd/n;
sigma=sum_row(1,1)*sum_column(1,1)+sum_row(2,1)*sum_column(1,2)+sum_row(3,1)
)*sum_column(1,3);
kapa=(n*sumd)-sigma)/(n^2-sigma);
disp('Overall Accuracy');
disp(OA*100);
disp('Kapa Coeficient');
disp(kapa*100);

```

ج) خروجیها به این محل کپی شود.



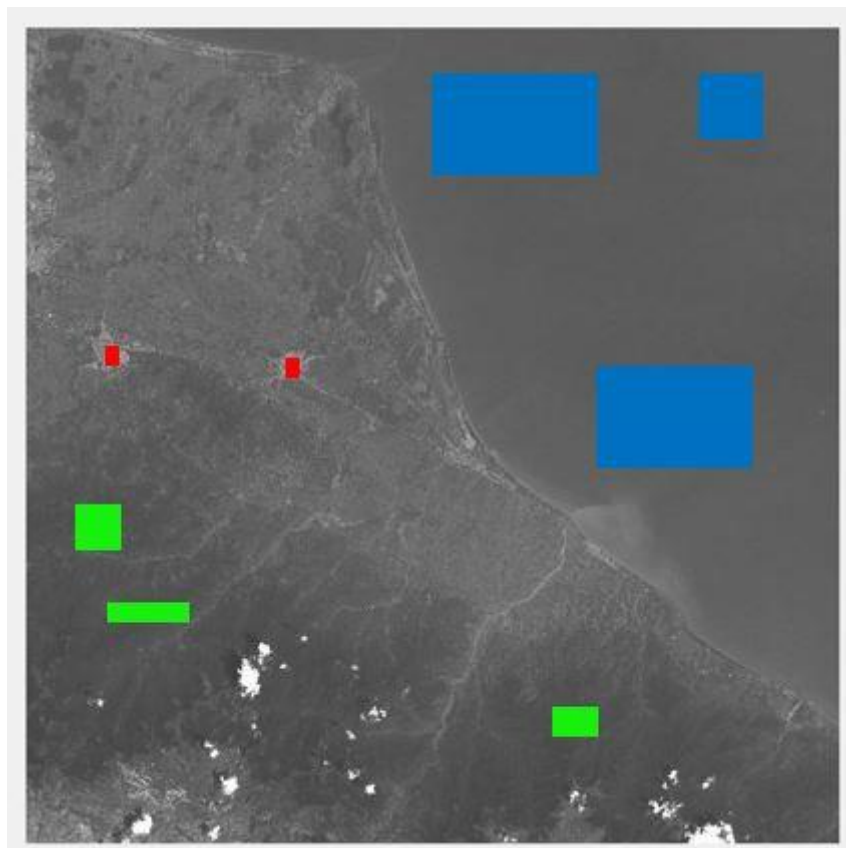
محاسبه ی دقت و ضریب کلی:



صحت کلی: ۴۳.۱۳۴۴ درصد ؛ ضریب کاپا: ۳۹.۷۱۸۰ درصد

الف) عنوان آیتم شماره ۱۹ (انجام طبقه بندی حداقل فاصله بر روی تصویر سنجنده TM ماهواره Landsat)

در این روش همانند روش قبل، داده های آموزشی را تعیین می کنیم. مراکز کلاس ها را بدست می آوریم. سپس به کمک حلقه ی for برای هر پیکسل تصویر، فاصله را از مرکز کلاس بدست می آوریم. پیکسل به کلاسی تعلق میگیرد که کمترین فاصله را از مرکز کلاس داشته باشد.



ب) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

B=zeros(2048,2048,7);
for i=1:7
    name=['b',num2str(i),'.dat'];
    band = fopen(name);
    A1 = fread(band);
    for j=1:2048
        B(j,1:2048,i)=A1((j-1)*2048+1:2048*j,1);
    end
    B(:,:,i)=mat2gray(B(:,:,i));
end
```

```

figure('Name','traning data','NumberTitle','off');
imshow(B(:,:,1));
n=input('number of training test ROI:');
x1=zeros(2*n,1);
y1=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x1(2*i-1,1)=xx(1,1);
    y1(2*i-1,1)=yy(1,1);
    x1(2*i,1)=xx(2,1);
    y1(2*i,1)=yy(2,1);
    rectangle('position',[x1(2*i-1,1),y1(2*i-1,1),x1(2*i,1)-x1(2*i-1,1),y1(2*i,1)-y1(2*i-1,1)], 'facecolor',[0 0 1]);
end

x2=zeros(2*n,1);
y2=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x2(2*i-1,1)=xx(1,1);
    y2(2*i-1,1)=yy(1,1);
    x2(2*i,1)=xx(2,1);
    y2(2*i,1)=yy(2,1);
    rectangle('position',[x2(2*i-1,1),y2(2*i-1,1),x2(2*i,1)-x2(2*i-1,1),y2(2*i,1)-y2(2*i-1,1)], 'facecolor',[0 1 0]);
end

x3=zeros(2*n,1);
y3=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x3(2*i-1,1)=xx(1,1);
    y3(2*i-1,1)=yy(1,1);
    x3(2*i,1)=xx(2,1);
    y3(2*i,1)=yy(2,1);
    rectangle('position',[x3(2*i-1,1),y3(2*i-1,1),x3(2*i,1)-x3(2*i-1,1),y3(2*i,1)-y3(2*i-1,1)], 'facecolor',[1 0 0]);
end

x1=round(x1);
y1=round(y1);
x2=round(x2);
y2=round(y2);
x3=round(x3);
y3=round(y3);

blanck=x1;
x1=y1;
y1=blanck;
blanck=x2;
x2=y2;
y2=blanck;
blanck=x3;
x3=y3;
y3=blanck;

mean1=zeros(n,1,7);
for j=1:n
    for i=1:7
        mean1(j,1,i)=mean(mean(B(x1(2*j-1):x1(2*j),y1(2*j-1):y1(2*j),i)));
    end
end
mean1=mean(mean1);

```

```

mean2=zeros(n,1,7);
for j=1:n
    for i=1:7
        mean2(j,1,i)=mean(mean(B(x2(2*j-1):x2(2*j),y2(2*j-1):y2(2*j),i)));
    end
end
mean2=mean(mean2);
mean3=zeros(n,1,7);
for j=1:n
    for i=1:7
        mean3(j,1,i)=mean(mean(B(x3(2*j-1):x3(2*j),y3(2*j-1):y3(2*j),i)));
    end
end
mean3=mean(mean3);
%minimum distance classification:
dist1=zeros(7,1);
dist2=zeros(7,1);
dist3=zeros(7,1);
classification=zeros(2048,2048,3);
for i=1:2048
    for j=1:2048
        for k=1:7
            dist1(k,1)=(B(i,j,k)-mean1(1,1,k))^2;
            dist2(k,1)=(B(i,j,k)-mean2(1,1,k))^2;
            dist3(k,1)=(B(i,j,k)-mean3(1,1,k))^2;
        end
        dist1=sqrt(sum(dist1));
        dist2=sqrt(sum(dist2));
        dist3=sqrt(sum(dist3));
        if dist2>dist1 && dist3>dist1
            classification(i,j,3)=255;
        elseif dist2<dist1 && dist2<dist3
            classification(i,j,2)=255;
        else
            classification(i,j,1)=255;
        end
    end
end
imshow(classification);

[x,y]=ginput(2);
rectangle('position',[x(1,1),y(1,1),x(2,1)-x(1,1),y(2,1)-y(1,1)], 'facecolor',[0 0 1]);
blanck=y;
y=x;
x=blanck;
test=classification(x(1,1):x(2,1),y(1,1):y(2,1),:);
imshow(test)
[x,y,Z]=size(test);

a=zeros(3,3);
n=x*y;
for i=1:x
    for j=1:y
        if test(i,j,3)==255
            a(1,1)=a(1,1)+1;
        end
    end
end
for i=1:x
    for j=1:y

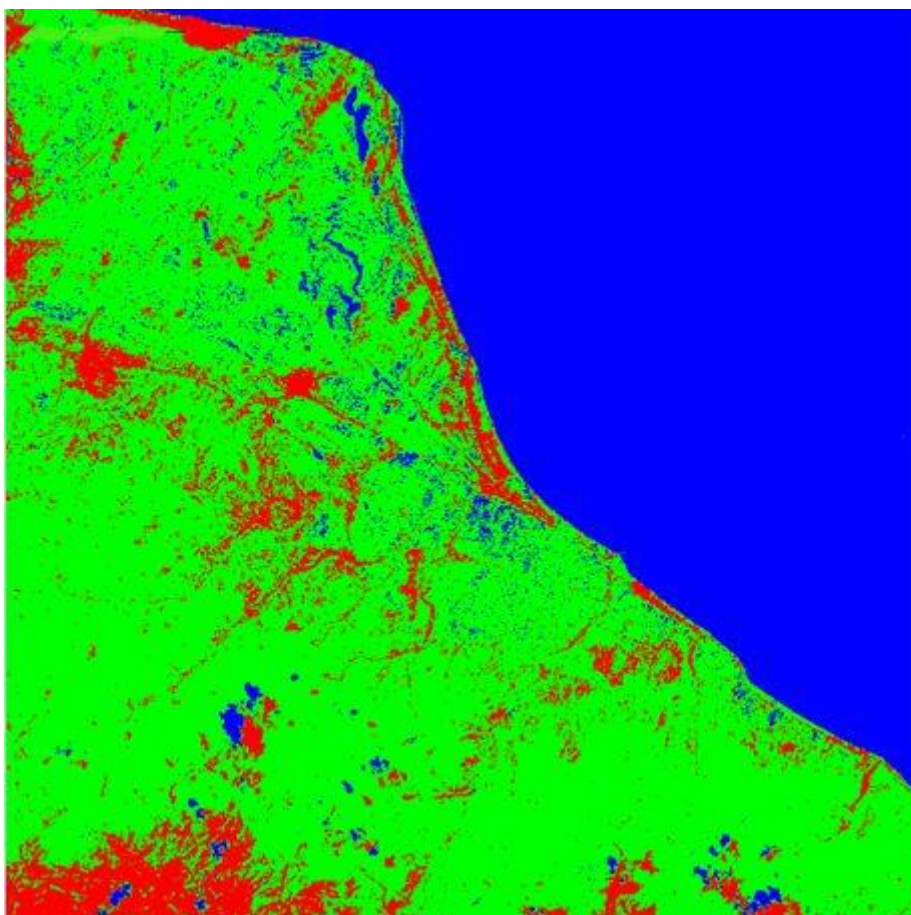
```

```

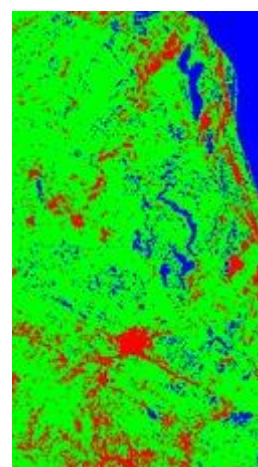
        if test(i,j,2)==255
            a(2,2)=a(2,2)+1;
        end
    end
end
for i=1:x
    for j=1:y
        if test(i,j,1)==255
            a(3,3)=a(3,3)+1;
        end
    end
end
error(1,1)=input('error: Water Region :');
error(2,1)=input('error: Vegetation Region :');
error(3,1)=input('error: Urban Area :');
a(1,1)=a(1,1)-error(1,1);
a(2,2)=a(2,2)-error(2,1);
a(3,3)=a(3,3)-error(3,1);
sumd=a(1,1)+a(2,2)+a(3,3);
a(1,2)=input('error: Water Region/Vegetation Region:');
a(1,3)=input('error: Water Region/Urban Area:');
a(2,1)=input('error: Vegetation Region/Water Region:');
a(2,3)=input('error: Vegetation Region/Urban Area:');
a(3,1)=input('error: Urban Area/Water Region:');
a(3,2)=input('error: Urban Area/Vegetation Region:');
sum_row=sum(a,2);
sum_column=sum(a,1);
OA=sumd/n;
sigma=sum_row(1,1)*sum_column(1,1)+sum_row(2,1)*sum_column(1,2)+sum_row(3,1)
)*sum_column(1,3);
kapa=((n*sumd)-sigma)/(n^2-sigma);
disp('Overall Accuracy');
disp(OA*100);
disp('Kapa Coeficient');
disp(kapa*100);

```

ج) خروجیها به این محل کپی شود.



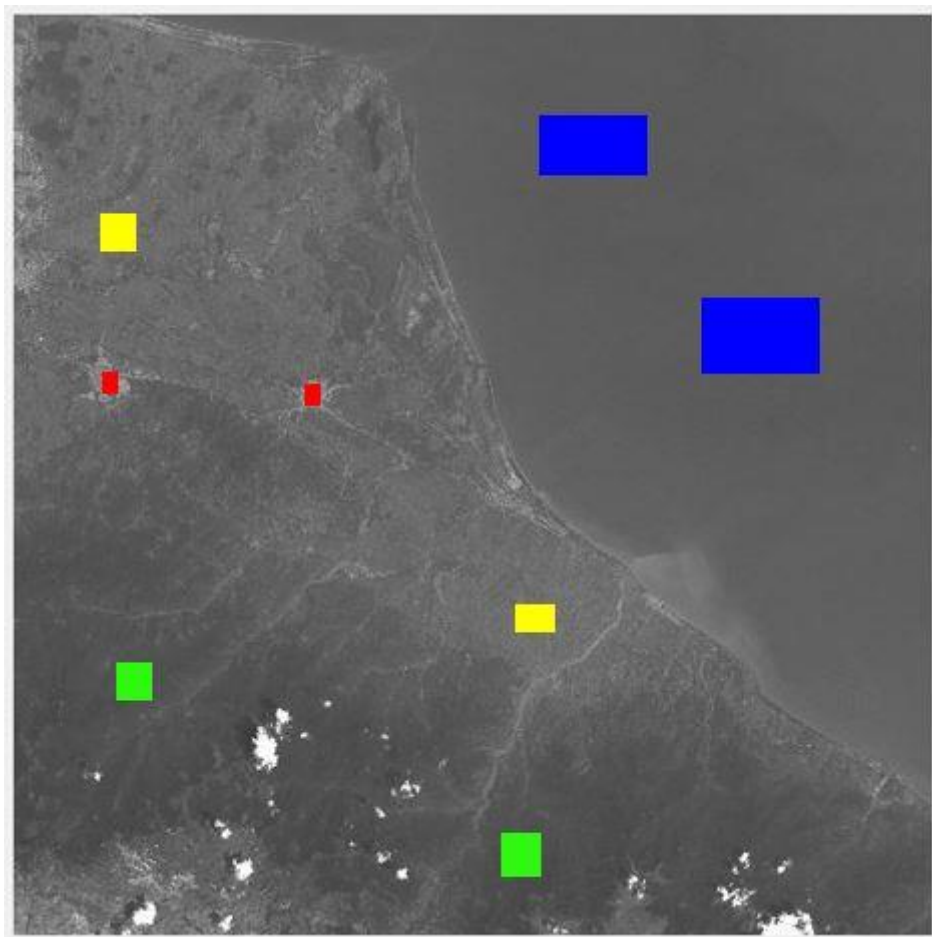
محاسبه ی دقت و ضریب کلی:



صحت کلی: ۹۳.۵۵۰۱ درصد ؛ ضریب کاپا: ۸۹.۰۶۳۷ درصد

الف) عنوان آیتم شماره ۲۰ (انجام طبقه بندی حداکثر احتمال بر روی تصویر سنجنده TM ماهواره Landsat)

برای طبقه بندی به روش حداکثر احتمال، از توابع توزیع احتمال کمک می گیریم. مقدار تابع احتمال را برای هر پیکسل بدست آورده و پیکسل به کلاسی تعلق میگیرد که دارای بیشترین مقدار تابع احتمال برای کلاس نام باشد. قبل از آن نیاز به داده های آموزشی داریم که به کمک دستور rectangle، این مناطق را تعیین می کنیم. برای این طبقه بندی، چهار کلاس در نظر گرفته شده است: پهنه ی آبی به رنگ آبی، منطقه ی شهری به رنگ قرمز، پوشش گیاهی متراکم به رنگ سبز و پوشش گیاهی کم تراکم به رنگ زرد. سپس بردار میانگین و کوواریانس های مربوط به هر کلاس را یافته و توابع توزیع احتمال را برای هر پیکسل تصویر $(g_i(x) = -\ln|\Sigma_i| - (x - m_i)^T \Sigma_i^{-1} (x - m_i))$ تشکیل می دهیم. هرچه برای هرپیکسل مقدار تابع توزیع احتمال به ازای کلاس i بیشتر باشد، به آن کلاس تعلق میگیرد.



ب) برنامه (Source code) که در مطلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

B=zeros(2048,2048,7);
for i=1:7
    name=['b',num2str(i),'.dat'];
    band = fopen(name);
    A1 = fread(band);
    for j=1:2048
        B(j,1:2048,i)=A1((j-1)*2048+1:2048*j,1);
    end
    B(:,:,i)=mat2gray(B(:,:,i));
end
figure('Name','traning data','NumberTitle','off');
imshow(B(:,:,1));
n=3;
x1=zeros(2*n,1);
y1=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x1(2*i-1,1)=xx(1,1);
    y1(2*i-1,1)=yy(1,1);
    x1(2*i,1)=xx(2,1);
    y1(2*i,1)=yy(2,1);
    rectangle('position',[x1(2*i-1,1),y1(2*i-1,1),x1(2*i,1)-x1(2*i-1,1),y1(2*i,1)-y1(2*i-1,1)], 'facecolor',[0 0 1]);
end

x2=zeros(2*n,1);
y2=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x2(2*i-1,1)=xx(1,1);
    y2(2*i-1,1)=yy(1,1);
    x2(2*i,1)=xx(2,1);
    y2(2*i,1)=yy(2,1);
    rectangle('position',[x2(2*i-1,1),y2(2*i-1,1),x2(2*i,1)-x2(2*i-1,1),y2(2*i,1)-y2(2*i-1,1)], 'facecolor',[0 1 0]);
end

x3=zeros(2*n,1);
y3=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x3(2*i-1,1)=xx(1,1);
    y3(2*i-1,1)=yy(1,1);
    x3(2*i,1)=xx(2,1);
    y3(2*i,1)=yy(2,1);
    rectangle('position',[x3(2*i-1,1),y3(2*i-1,1),x3(2*i,1)-x3(2*i-1,1),y3(2*i,1)-y3(2*i-1,1)], 'facecolor',[1 0 0]);
end

x4=zeros(2*n,1);
y4=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x4(2*i-1,1)=xx(1,1);
    y4(2*i-1,1)=yy(1,1);
    x4(2*i,1)=xx(2,1);
```

```

        y4(2*i,1)=yy(2,1);
        rectangle('position',[x4(2*i-1,1),y4(2*i-1,1),x4(2*i,1)-x4(2*i-1,1),y4(2*i,1)-y4(2*i-1,1)], 'facecolor',[1 1 0]);
    end
    x1=round(x1);
    y1=round(y1);
    x2=round(x2);
    y2=round(y2);
    x3=round(x3);
    y3=round(y3);
    x4=round(x4);
    y4=round(y4);

    blanck=x1;
    x1=y1;
    y1=blanck;
    blanck=x2;
    x2=y2;
    y2=blanck;
    blanck=x3;
    x3=y3;
    y3=blanck;
    blanck=x4;
    x4=y4;
    y4=blanck;

    mean1=zeros(n,1,7);
    for j=1:n
        for i=1:7
            mean1(j,1,i)=mean(mean(B(x1(2*j-1):x1(2*j),y1(2*j-1):y1(2*j),i)));
        end
    end
    mean1=mean(mean1);
    mean2=zeros(n,1,7);
    for j=1:n
        for i=1:7
            mean2(j,1,i)=mean(mean(B(x2(2*j-1):x2(2*j),y2(2*j-1):y2(2*j),i)));
        end
    end
    mean2=mean(mean2);
    mean3=zeros(n,1,7);
    for j=1:n
        for i=1:7
            mean3(j,1,i)=mean(mean(B(x3(2*j-1):x3(2*j),y3(2*j-1):y3(2*j),i)));
        end
    end
    mean3=mean(mean3);
    mean4=zeros(n,1,7);
    for j=1:n
        for i=1:7
            mean4(j,1,i)=mean(mean(B(x4(2*j-1):x4(2*j),y4(2*j-1):y4(2*j),i)));
        end
    end
    mean4=mean(mean4);

    mbar1=zeros(7,1);
    mbar2=zeros(7,1);
    mbar3=zeros(7,1);
    mbar4=zeros(7,1);
    for i=1:7
        mbar1(i,1)=mean1(1,1,i);
    end

```

```

        mbar2(i,1)=mean2(1,1,i);
        mbar3(i,1)=mean3(1,1,i);
        mbar4(i,1)=mean4(1,1,i);
end

covariance1=zeros(7,7);
covariance2=zeros(7,7);
covariance3=zeros(7,7);
covariance4=zeros(7,7);

sub_pic11=B(x1(2*1-1):x1(2*1),y1(2*1-1):y1(2*1),:);
sub_pic12=B(x1(2*2-1):x1(2*2),y1(2*2-1):y1(2*2),:);
sub_pic13=B(x1(2*3-1):x1(2*3),y1(2*3-1):y1(2*3),:);
sub_pic21=B(x2(2*1-1):x2(2*1),y2(2*1-1):y2(2*1),:);
sub_pic22=B(x2(2*2-1):x2(2*2),y2(2*2-1):y2(2*2),:);
sub_pic23=B(x2(2*3-1):x2(2*3),y2(2*3-1):y2(2*3),:);
sub_pic31=B(x3(2*1-1):x3(2*1),y3(2*1-1):y3(2*1),:);
sub_pic32=B(x3(2*2-1):x3(2*2),y3(2*2-1):y3(2*2),:);
sub_pic33=B(x3(2*3-1):x3(2*3),y3(2*3-1):y3(2*3),:);

sub_pic41=B(x4(2*1-1):x4(2*1),y4(2*1-1):y4(2*1),:);
sub_pic42=B(x4(2*2-1):x4(2*2),y4(2*2-1):y4(2*2),:);
sub_pic43=B(x4(2*3-1):x4(2*3),y4(2*3-1):y4(2*3),:);

[a11,b11,c11]=size(sub_pic11);
[a12,b12,c12]=size(sub_pic12);
[a13,b13,c13]=size(sub_pic13);
[a21,b21,c21]=size(sub_pic21);
[a22,b22,c22]=size(sub_pic22);
[a23,b23,c23]=size(sub_pic23);
[a31,b31,c31]=size(sub_pic31);
[a32,b32,c32]=size(sub_pic32);
[a33,b33,c33]=size(sub_pic33);

[a41,b41,c41]=size(sub_pic41);
[a42,b42,c42]=size(sub_pic42);
[a43,b43,c43]=size(sub_pic43);
cov11=zeros(a11*b11,1);
for n=1:7
    for i=1:b11
        cov11((i-1)*a11+1:i*a11,n)=sub_pic11(:,i,n);
    end
end
cov12=zeros(a12*b12,1);
for n=1:7
    for i=1:b12
        cov12((i-1)*a12+1:i*a12,n)=sub_pic12(:,i,n);
    end
end
cov13=zeros(a13*b13,1);
for n=1:7
    for i=1:b13
        cov13((i-1)*a13+1:i*a13,n)=sub_pic13(:,i,n);
    end
end
cov21=zeros(a21*b21,1);
for n=1:7
    for i=1:b21
        cov21((i-1)*a21+1:i*a21,n)=sub_pic21(:,i,n);
    end
end

```

```

        end
    end
    cov22=zeros(a22*b22,1);
    for n=1:7
        for i=1:b22
            cov22((i-1)*a22+1:i*a22,n)=sub_pic22(:,i,n);
        end
    end
    cov23=zeros(a23*b23,1);
    for n=1:7
        for i=1:b23
            cov23((i-1)*a23+1:i*a23,n)=sub_pic23(:,i,n);
        end
    end
    cov31=zeros(a31*b31,1);
    for n=1:7
        for i=1:b31
            cov31((i-1)*a31+1:i*a31,n)=sub_pic31(:,i,n);
        end
    end
    cov32=zeros(a32*b32,1);
    for n=1:7
        for i=1:b32
            cov32((i-1)*a32+1:i*a32,n)=sub_pic32(:,i,n);
        end
    end
    cov33=zeros(a33*b33,1);
    for n=1:7
        for i=1:b33
            cov33((i-1)*a33+1:i*a33,n)=sub_pic33(:,i,n);
        end
    end

    cov41=zeros(a41*b41,1);
    for n=1:7
        for i=1:b41
            cov41((i-1)*a41+1:i*a41,n)=sub_pic41(:,i,n);
        end
    end
    cov42=zeros(a42*b42,1);
    for n=1:7
        for i=1:b42
            cov42((i-1)*a42+1:i*a42,n)=sub_pic42(:,i,n);
        end
    end
    cov43=zeros(a43*b43,1);
    for n=1:7
        for i=1:b43
            cov43((i-1)*a43+1:i*a43,n)=sub_pic43(:,i,n);
        end
    end

    cov1=cov11;
    cov1(a11*b11+1:a11*b11+a12*b12,:)=cov12;
    cov1(a11*b11+a12*b12+1:a11*b11+a12*b12+a13*b13,:)=cov13;
    cov1=cov(cov1);
    cov2=cov21;
    cov2(a21*b21+1:a21*b21+a22*b22,:)=cov22;
    cov2(a21*b21+a22*b22+1:a21*b21+a22*b22+a23*b23,:)=cov23;
    cov2=cov(cov2);
    cov3=cov31;

```

```

cov3(a31*b31+1:a31*b31+a32*b32,:)=cov32;
cov3(a31*b31+a32*b32+1:a31*b31+a32*b32+a33*b33,:)=cov33;
cov3=cov(cov3);

cov4=cov41;
cov4(a41*b41+1:a41*b41+a42*b42,:)=cov42;
cov4(a41*b41+a42*b42+1:a41*b41+a42*b42+a43*b43,:)=cov43;
cov4=cov(cov4);

classification=zeros(2048,2048,3);
b=zeros(7,1);
for i=1:2048
    for j=1:2048
        for n=1:7
            b(n,1)=B(i,j,n);
        end
        g1=-log(det(cov1)-(b-mbar1)'*(inv(cov1))*(b-mbar1)));
        g2=-log(det(cov2)-(b-mbar2)'*(inv(cov2))*(b-mbar2)));
        g3=-log(det(cov3)-(b-mbar3)'*(inv(cov3))*(b-mbar3)));
        g4=-log(det(cov4)-(b-mbar4)'*(inv(cov4))*(b-mbar4)));
        if g1>g2 && g1>g3 && g1>g4
            classification(i,j,3)=256;
        elseif g2>g1 && g2>g3 && g2>g4
            classification(i,j,2)=256;
        elseif g3>g1 && g3>g2 && g3>g4
            classification(i,j,1)=256;
        else
            classification(i,j,1:2)=250;
        end
    end
end

imshow(classification);
%degat:
[x,y]=ginput(2);
rectangle('position',[x(1,1),y(1,1),x(2,1)-x(1,1),y(2,1)-y(1,1)], 'facecolor',[0 0 1]);
blanck=y;
y=x;
x=blanck;
teest=classification(x(1,1):x(2,1),y(1,1):y(2,1),:);
imshow(teest)
[x,y,Z]=size(teest);
%matrix of errors: - satr va soton aval pahne abi , dovom poshesh giahhi va
%sevom shari
a=zeros(4,4);
n=x*y;
for i=1:x
    for j=1:y
        if teest(i,j,3)==256
            a(1,1)=a(1,1)+1;
        end
    end
end
for i=1:x
    for j=1:y
        if teest(i,j,2)==256
            a(2,2)=a(2,2)+1;
        end
    end
end
end
end

```

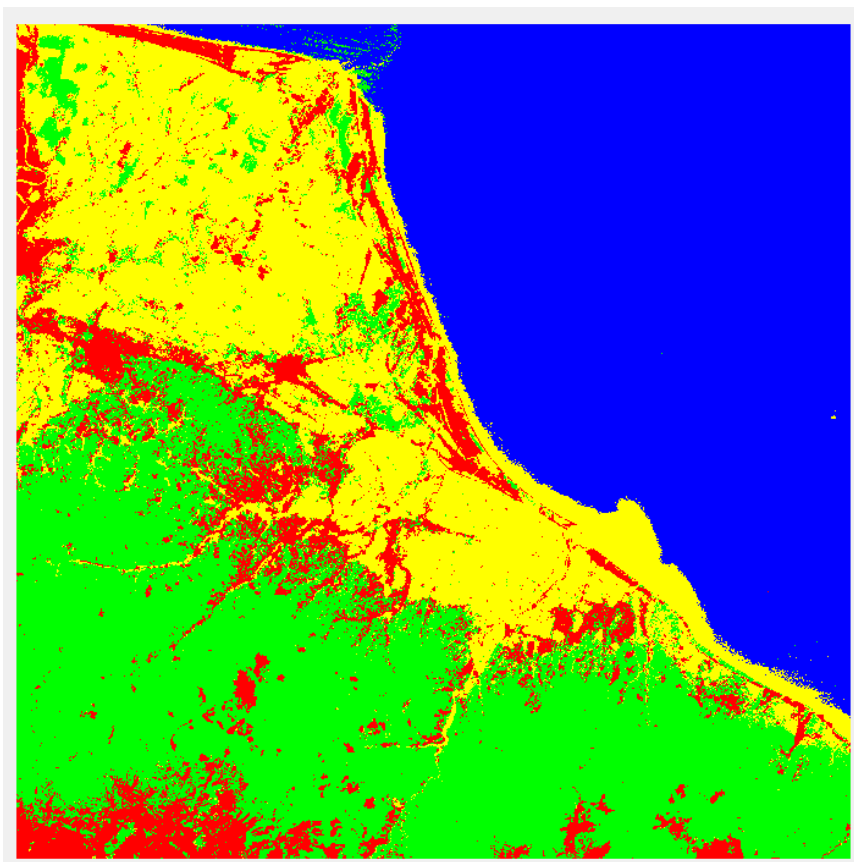


```

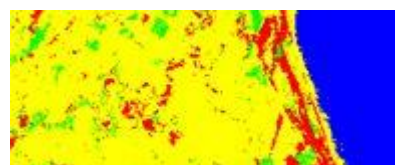
for i=1:x
    for j=1:y
        if teest(i,j,1)==256
            a(3,3)=a(3,3)+1;
        end
    end
end
for i=1:x
    for j=1:y
        if teest(i,j,2)==250 && teest(i,j,1)==250
            a(4,4)=a(4,4)+1;
        end
    end
end
error(1,1)=input('error: Water Region :');
error(2,1)=input('error: Vegetation Region :');
error(3,1)=input('error: Urban Area :');
error(4,1)=input('error: Low-Vegetation Region :');
a(1,1)=a(1,1)-error(1,1);
a(2,2)=a(2,2)-error(2,1);
a(3,3)=a(3,3)-error(3,1);
a(4,4)=a(4,4)-error(4,1);
sumd=a(1,1)+a(2,2)+a(3,3)+a(4,4);
a(1,2)=input('error: Water Region/Vegetation Region:');
a(1,3)=input('error: Water Region/Urban Area:');
a(1,4)=input('error: Water Region/Low-Vegetation Region:');
a(2,1)=input('error: Vegetation Region/Water Region:');
a(2,3)=input('error: Vegetation Region/Urban Area:');
a(2,4)=input('error: Vegetation Region/Low-Vegetation Region:');
a(3,1)=input('error: Urban Area/Water Region:');
a(3,2)=input('error: Urban Area/Vegetation Region:');
a(3,4)=input('error: Urban Area/Low-Vegetation Region:');
a(4,1)=input('error: Urban Area/Water Region:');
a(4,2)=input('error: Low Vegetation Region/Vegetation Region:');
a(4,3)=input('error: Low Vegetation Region/Urban Area:');
sum_row=sum(a,2);
sum_column=sum(a,1);
OA=sumd/n;
sigma=sum_row(1,1)*sum_column(1,1)+sum_row(2,1)*sum_column(1,2)+sum_row(3,1)
)*sum_column(1,3)+sum_row(4,1)*sum_column(1,4);
kapa=((n*sumd)-sigma)/(n^2-sigma);
disp('Overall Accuracy');
disp(OA*100);
disp('Kapa Coefficient');
disp(kapa*100);

```

ج) خروجیها به این محل کپی شود.



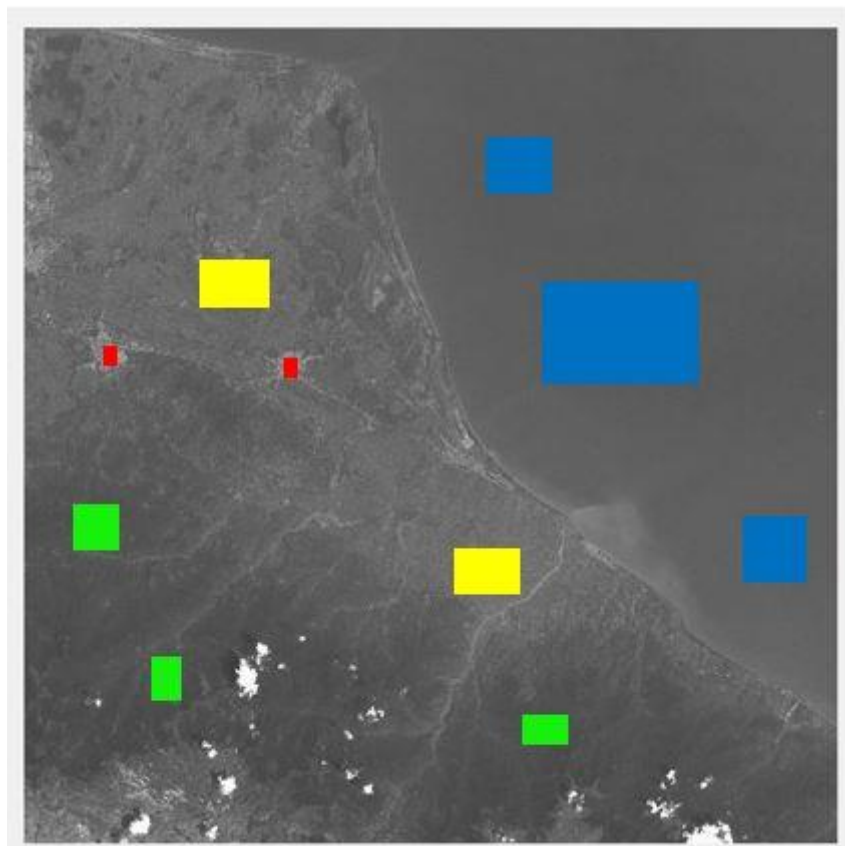
محاسبه ی دقت و ضریب کلی:



صحت کلی: ۸۸.۴۱۶۷ درصد ؛ ضریب کاپا: ۷۹.۵۹۰۲ درصد

الف) عنوان آیتم شماره ۲۱ (انجام طبقه بندی ماهالانوبیس بر روی تصویر سنجنده TM ماهواره Landsat)

روش کدنویسی این طبقه بندی همانند روش طبقه بندی حداکثر احتمال است، با دوتفاوت. اول اینکه مقدار تابع تمایز هر پیکسل به ازای هر کلاس کمتر شد، مربوط به آن کلاس است و دوم آنکه به دلیل برابر فرض کردن ماتریس های کووریانس هر کلاس، $\ln|\Sigma_i|$ دیگر نقشی در تابع تمایز ندارد.



ب) برنامه (Source code) که در متلب یا هر کمپایلر دیگر نوشته اید به این محل کپی شود.

```
clc;
clear;
close all;

B=zeros(2048,2048,7);
for i=1:7
    name=['b',num2str(i),'.dat'];
    band = fopen(name);
    A1 = fread(band);
    for j=1:2048
        B(j,1:2048,i)=A1((j-1)*2048+1:2048*j,1);
    end
    B(:, :, i)=mat2gray(B(:, :, i));
end
```

```

figure('Name','traning data','NumberTitle','off');
imshow(B(:,:,1));
% n=input('number of training test ROI:');
n=3;
x1=zeros(2*n,1);
y1=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x1(2*i-1,1)=xx(1,1);
    y1(2*i-1,1)=yy(1,1);
    x1(2*i,1)=xx(2,1);
    y1(2*i,1)=yy(2,1);
    rectangle('position',[x1(2*i-1,1),y1(2*i-1,1),x1(2*i,1)-x1(2*i-1,1),y1(2*i,1)-y1(2*i-1,1)], 'facecolor',[0 0 1]);
end

x2=zeros(2*n,1);
y2=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x2(2*i-1,1)=xx(1,1);
    y2(2*i-1,1)=yy(1,1);
    x2(2*i,1)=xx(2,1);
    y2(2*i,1)=yy(2,1);
    rectangle('position',[x2(2*i-1,1),y2(2*i-1,1),x2(2*i,1)-x2(2*i-1,1),y2(2*i,1)-y2(2*i-1,1)], 'facecolor',[0 1 0]);
end

x3=zeros(2*n,1);
y3=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x3(2*i-1,1)=xx(1,1);
    y3(2*i-1,1)=yy(1,1);
    x3(2*i,1)=xx(2,1);
    y3(2*i,1)=yy(2,1);
    rectangle('position',[x3(2*i-1,1),y3(2*i-1,1),x3(2*i,1)-x3(2*i-1,1),y3(2*i,1)-y3(2*i-1,1)], 'facecolor',[1 0 0]);
end

x4=zeros(2*n,1);
y4=zeros(2*n,1);
for i=1:n
    [xx,yy]=ginput(2);
    x4(2*i-1,1)=xx(1,1);
    y4(2*i-1,1)=yy(1,1);
    x4(2*i,1)=xx(2,1);
    y4(2*i,1)=yy(2,1);
    rectangle('position',[x4(2*i-1,1),y4(2*i-1,1),x4(2*i,1)-x4(2*i-1,1),y4(2*i,1)-y4(2*i-1,1)], 'facecolor',[1 1 0]);
end

x1=round(x1);
y1=round(y1);
x2=round(x2);
y2=round(y2);
x3=round(x3);
y3=round(y3);
x4=round(x4);
y4=round(y4);

blanck=x1;
x1=y1;
y1=blanck;

```

```

blanck=x2;
x2=y2;
y2=blanck;
blanck=x3;
x3=y3;
y3=blanck;
blanck=x4;
x4=y4;
y4=blanck;

mean1=zeros(n,1,7);
for j=1:n
    for i=1:7
        mean1(j,1,i)=mean(mean(B(x1(2*j-1):x1(2*j),y1(2*j-1):y1(2*j),i)));
    end
end
mean1=mean(mean1);
mean2=zeros(n,1,7);
for j=1:n
    for i=1:7
        mean2(j,1,i)=mean(mean(B(x2(2*j-1):x2(2*j),y2(2*j-1):y2(2*j),i)));
    end
end
mean2=mean(mean2);
mean3=zeros(n,1,7);
for j=1:n
    for i=1:7
        mean3(j,1,i)=mean(mean(B(x3(2*j-1):x3(2*j),y3(2*j-1):y3(2*j),i)));
    end
end
mean3=mean(mean3);
mean4=zeros(n,1,7);
for j=1:n
    for i=1:7
        mean4(j,1,i)=mean(mean(B(x4(2*j-1):x4(2*j),y4(2*j-1):y4(2*j),i)));
    end
end
mean4=mean(mean4);

miangin1=zeros(7,1);
miangin2=zeros(7,1);
miangin3=zeros(7,1);
miangin4=zeros(7,1);
for i=1:7
    miangin1(i,1)=mean1(1,1,i);
    miangin2(i,1)=mean2(1,1,i);
    miangin3(i,1)=mean3(1,1,i);
    miangin4(i,1)=mean4(1,1,i);
end
%matrix covariance:
covariancel=zeros(7,7);
covariance2=zeros(7,7);
covariance3=zeros(7,7);
covariance4=zeros(7,7);

sub_pic11=B(x1(2*1-1):x1(2*1),y1(2*1-1):y1(2*1),:);
sub_pic12=B(x1(2*2-1):x1(2*2),y1(2*2-1):y1(2*2),:);
sub_pic13=B(x1(2*3-1):x1(2*3),y1(2*3-1):y1(2*3),:);
sub_pic21=B(x2(2*1-1):x2(2*1),y2(2*1-1):y2(2*1),:);
sub_pic22=B(x2(2*2-1):x2(2*2),y2(2*2-1):y2(2*2),:);
sub_pic23=B(x2(2*3-1):x2(2*3),y2(2*3-1):y2(2*3),:);

```

```

sub_pic31=B(x3(2*1-1):x3(2*1),y3(2*1-1):y3(2*1),:);
sub_pic32=B(x3(2*2-1):x3(2*2),y3(2*2-1):y3(2*2),:);
sub_pic33=B(x3(2*3-1):x3(2*3),y3(2*3-1):y3(2*3),:);

sub_pic41=B(x4(2*1-1):x4(2*1),y4(2*1-1):y4(2*1),:);
sub_pic42=B(x4(2*2-1):x4(2*2),y4(2*2-1):y4(2*2),:);
sub_pic43=B(x4(2*3-1):x4(2*3),y4(2*3-1):y4(2*3),:);

[a11,b11,c11]=size(sub_pic11);
[a12,b12,c12]=size(sub_pic12);
[a13,b13,c13]=size(sub_pic13);
[a21,b21,c21]=size(sub_pic21);
[a22,b22,c22]=size(sub_pic22);
[a23,b23,c23]=size(sub_pic23);
[a31,b31,c31]=size(sub_pic31);
[a32,b32,c32]=size(sub_pic32);
[a33,b33,c33]=size(sub_pic33);

[a41,b41,c41]=size(sub_pic41);
[a42,b42,c42]=size(sub_pic42);
[a43,b43,c43]=size(sub_pic43);
cov11=zeros(a11*b11,1);
for n=1:7
    for i=1:b11
        cov11((i-1)*a11+1:i*a11,n)=sub_pic11(:,i,n);
    end
end
cov12=zeros(a12*b12,1);
for n=1:7
    for i=1:b12
        cov12((i-1)*a12+1:i*a12,n)=sub_pic12(:,i,n);
    end
end
cov13=zeros(a13*b13,1);
for n=1:7
    for i=1:b13
        cov13((i-1)*a13+1:i*a13,n)=sub_pic13(:,i,n);
    end
end
cov21=zeros(a21*b21,1);
for n=1:7
    for i=1:b21
        cov21((i-1)*a21+1:i*a21,n)=sub_pic21(:,i,n);
    end
end
cov22=zeros(a22*b22,1);
for n=1:7
    for i=1:b22
        cov22((i-1)*a22+1:i*a22,n)=sub_pic22(:,i,n);
    end
end
cov23=zeros(a23*b23,1);
for n=1:7
    for i=1:b23
        cov23((i-1)*a23+1:i*a23,n)=sub_pic23(:,i,n);
    end
end
cov31=zeros(a31*b31,1);
for n=1:7

```

```

        for i=1:b31
            cov31((i-1)*a31+1:i*a31,n)=sub_pic31(:,i,n);
        end
    end
cov32=zeros(a32*b32,1);
for n=1:7
    for i=1:b32
        cov32((i-1)*a32+1:i*a32,n)=sub_pic32(:,i,n);
    end
end
cov33=zeros(a33*b33,1);
for n=1:7
    for i=1:b33
        cov33((i-1)*a33+1:i*a33,n)=sub_pic33(:,i,n);
    end
end

cov41=zeros(a41*b41,1);
for n=1:7
    for i=1:b41
        cov41((i-1)*a41+1:i*a41,n)=sub_pic41(:,i,n);
    end
end
cov42=zeros(a42*b42,1);
for n=1:7
    for i=1:b42
        cov42((i-1)*a42+1:i*a42,n)=sub_pic42(:,i,n);
    end
end
cov43=zeros(a43*b43,1);
for n=1:7
    for i=1:b43
        cov43((i-1)*a43+1:i*a43,n)=sub_pic43(:,i,n);
    end
end

cov1=cov11;
cov1(a11*b11+1:a11*b11+a12*b12,:)=cov12;
cov1(a11*b11+a12*b12+1:a11*b11+a12*b12+a13*b13,:)=cov13;
cov1=cov(cov1);
cov2=cov21;
cov2(a21*b21+1:a21*b21+a22*b22,:)=cov22;
cov2(a21*b21+a22*b22+1:a21*b21+a22*b22+a23*b23,:)=cov23;
cov2=cov(cov2);
cov3=cov31;
cov3(a31*b31+1:a31*b31+a32*b32,:)=cov32;
cov3(a31*b31+a32*b32+1:a31*b31+a32*b32+a33*b33,:)=cov33;
cov3=cov(cov3);

cov4=cov41;
cov4(a41*b41+1:a41*b41+a42*b42,:)=cov42;
cov4(a41*b41+a42*b42+1:a41*b41+a42*b42+a43*b43,:)=cov43;
cov4=cov(cov4);

classification=zeros(2048,2048,3);
b=zeros(7,1);
for i=1:2048
    for j=1:2048
        for n=1:7
            b(n,1)=B(i,j,n);
        end
    end
end

```



```

        end
        g1=((b-mianguin1) *(inv(cov1)) *(b-mianguin1));
        g2=((b-mianguin2) *(inv(cov2)) *(b-mianguin2));
        g3=((b-mianguin3) *(inv(cov3)) *(b-mianguin3));
        g4=((b-mianguin4) *(inv(cov4)) *(b-mianguin4));
        if g1<g2 && g1<g3 && g1<g4
            classification(i,j,3)=256;
        elseif g2<g1 && g2<g3 && g2<g4
            classification(i,j,2)=256;
        elseif g3<g1 && g3<g2 && g3<g4
            classification(i,j,1)=256;
        else
            classification(i,j,1:2)=250;
        end
    end
end

imshow(classification);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[x,y]=ginput(2);
rectangle('position',[x(1,1),y(1,1),x(2,1)-x(1,1),y(2,1)-
y(1,1)], 'facecolor',[0 0 1]);
blanck=y;
y=x;
x=blanck;
teest=classification(x(1,1):x(2,1),y(1,1):y(2,1),:);
imshow(teest)
[x,y,Z]=size(teest);

a=zeros(4,4);
n=x*y;
for i=1:x
    for j=1:y
        if teest(i,j,3)==256
            a(1,1)=a(1,1)+1;
        end
    end
end
for i=1:x
    for j=1:y
        if teest(i,j,2)==256
            a(2,2)=a(2,2)+1;
        end
    end
end
for i=1:x
    for j=1:y
        if teest(i,j,1)==256
            a(3,3)=a(3,3)+1;
        end
    end
end
for i=1:x
    for j=1:y
        if teest(i,j,2)==250 && teest(i,j,1)==250
            a(4,4)=a(4,4)+1;
        end
    end
end
er(1,1)=input('error: Water Region :');
er(2,1)=input('error: Vegetation Region :');

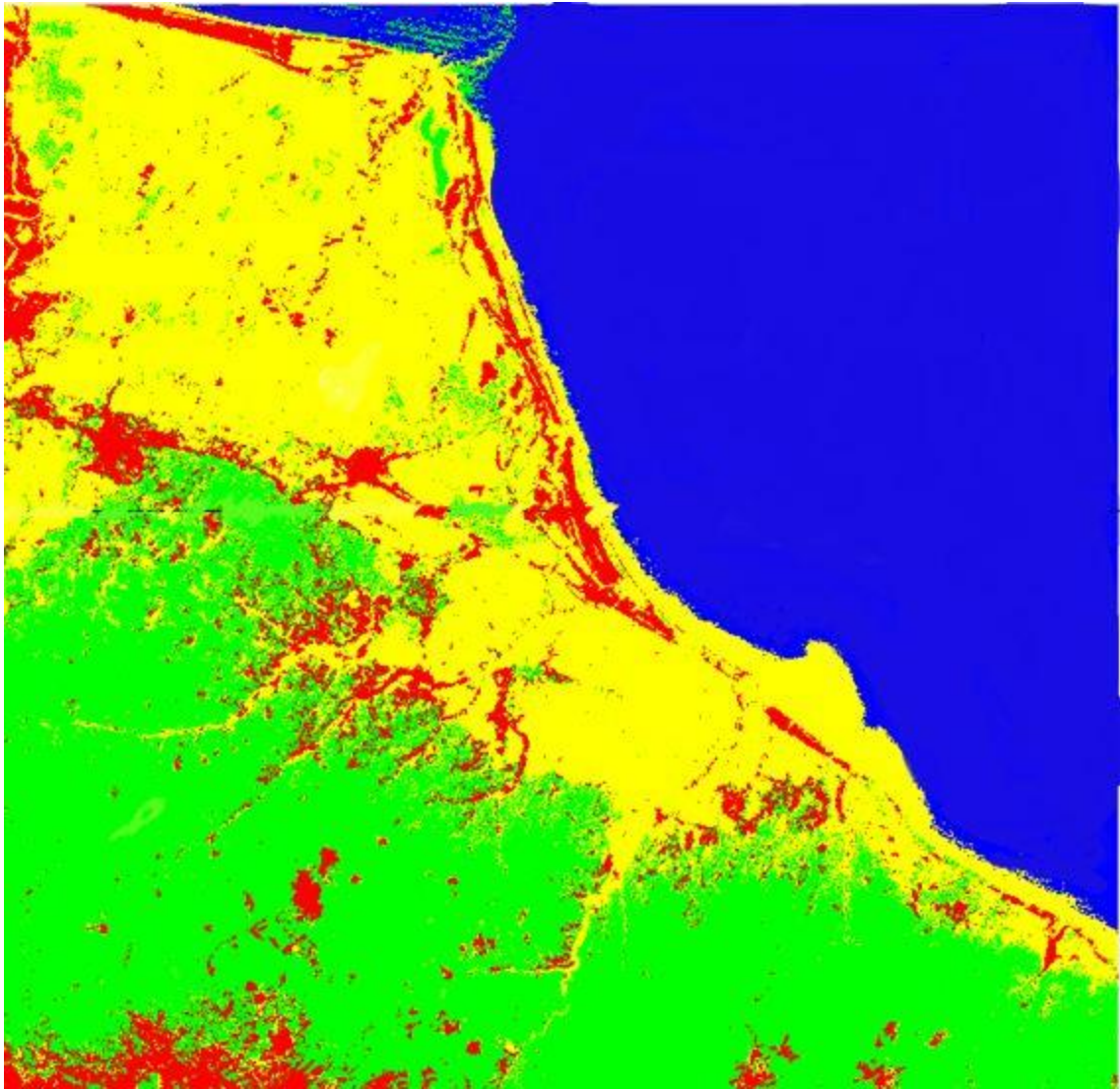
```

```

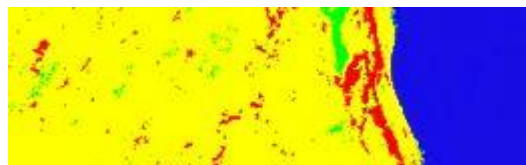
er(3,1)=input('error: Urban Area :');
er(4,1)=input('error: Low-Vegetation Region :');
a(1,1)=a(1,1)-er(1,1);
a(2,2)=a(2,2)-er(2,1);
a(3,3)=a(3,3)-er(3,1);
a(4,4)=a(4,4)-er(4,1);
sumd=a(1,1)+a(2,2)+a(3,3)+a(4,4);
a(1,2)=input('error: Water Region/Vegetation Region:');
a(1,3)=input('error: Water Region/Urban Area :');
a(1,4)=input('error: Water Region/Low-Vegetation Region:');
a(2,1)=input('error: Vegetation Region/Water Region:');
a(2,3)=input('error: Vegetation Region/Urban Area:');
a(2,4)=input('error: Vegetation Region/Low-Vegetation Region:');
a(3,1)=input('error: baraye nesbat Urban Area/Water Region:');
a(3,2)=input('error: Urban Area/Vegetation Region:');
a(3,4)=input('error: Urban Area/Low-Vegetation Region:');
a(4,1)=input('error: Urban Area/Water Region:');
a(4,2)=input('error: Low Vegetation Region/Vegetation Region:');
a(4,3)=input('error: Low Vegetation Region/Urban Area:');
sum_row=sum(a,2);
sum_column=sum(a,1);
OA=sumd/n;
sigma=sum_row(1,1)*sum_column(1,1)+sum_row(2,1)*sum_column(1,2)+sum_row(3,1)
)*sum_column(1,3)+sum_row(4,1)*sum_column(1,4);
kapa=((n*sumd)-sigma)/(n^2-sigma);
disp('Overall Accuracy');
disp(OA*100);
disp('Kapa Coeficient');
disp(kapa*100);

```

ج) خروجیها به این محل کپی شود.



محاسبه ی دقت و ضریب کلی:



صحت کلی: ۹۶.۳۳۴۱ درصد ؛ ضریب کاپا: ۸۹.۹۳۲۸ درصد