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

استاد مربوطه : جناب دکتر سراجیان

دانشجو : مسعود مهاجری

شماره دانشجویی : 810396099

زبان برنامه نویسی : MATLAB

ITEM 1

کد مورد استفاده :

خواندن عکس ها

```
satr = 2048 ;
sotoon = 2048 ;
photo1(:,:,1) = datopen('Data1\b1.dat',satr,sotoon) ;
photo1(:,:,2) = datopen('Data1\b2.dat',satr,sotoon) ;
photo1(:,:,3) = datopen('Data1\b3.dat',satr,sotoon) ;
photo1(:,:,4) = datopen('Data1\b4.dat',satr,sotoon) ;
photo1(:,:,5) = datopen('Data1\b5.dat',satr,sotoon) ;
photo1(:,:,6) = datopen('Data1\b6.dat',satr,sotoon) ;
photo1(:,:,7) = datopen('Data1\b7.dat',satr,sotoon) ;
```

نمایش آنها

```
figure('Name','Item 1')
while true
    band_name = input('Band to show : ') ;
    if band_name==1
        clf
        imshow(uint8(photo1(:,:,1)))
    elseif band_name==2
        clf
        imshow(uint8(photo1(:,:,2)))
    elseif band_name==3
        clf
        imshow(uint8(photo1(:,:,3)))
    elseif band_name==4
        clf
        imshow(uint8(photo1(:,:,4)))
    elseif band_name==5
        clf
        imshow(uint8(photo1(:,:,5)))
    elseif band_name==6
        clf
        imshow(uint8(photo1(:,:,6)))
    elseif band_name==7
        clf
        imshow(uint8(photo1(:,:,7)))
    else
        break ;
    end
end
```

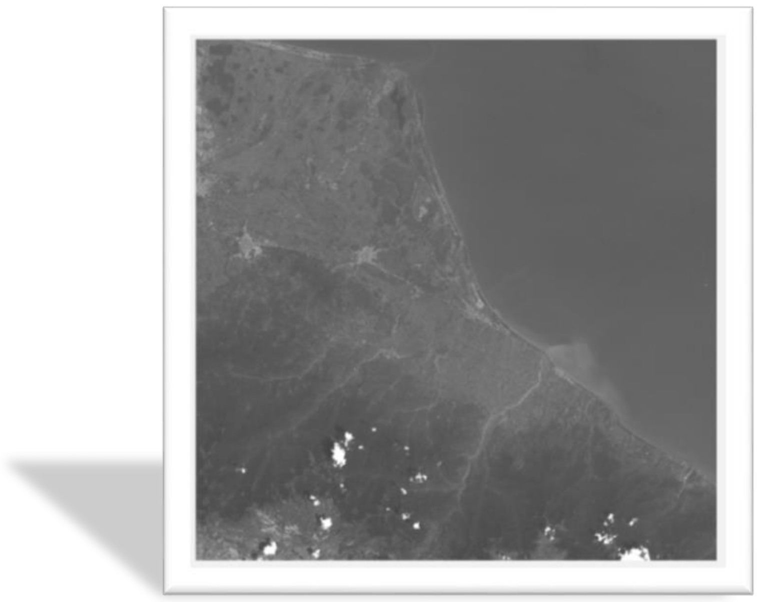
تابع های استفاده شده :

تابعی برای باز کردن عکس ها

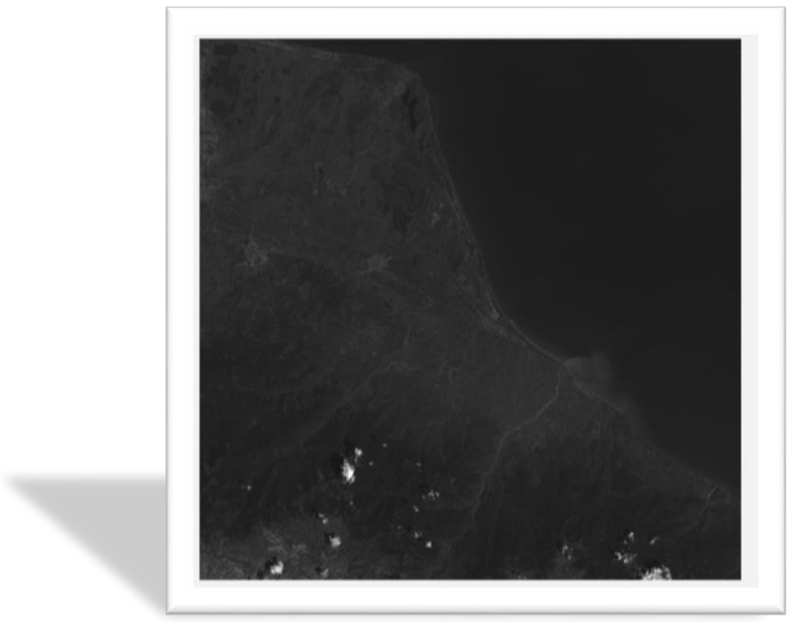
```
function [out_matrix ] =
datopen(filename,satr,sotoon)
band1 = fopen(filename);
b = zeros(satr,sotoon);
a = fread(band1);
for i=1:satr
    b(i,1:sotoon)=a((i-1)*sotoon+1:sotoon*i,1);
end
out_matrix = b ;
end
```

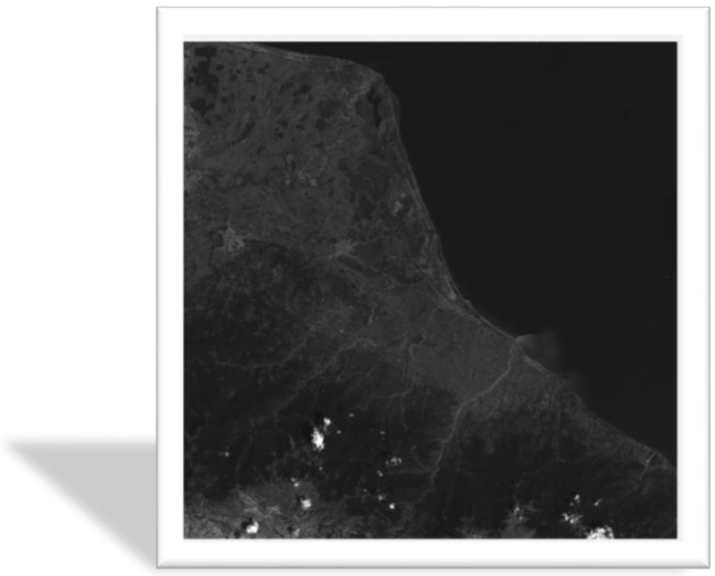
خروجی ها :

باند 1

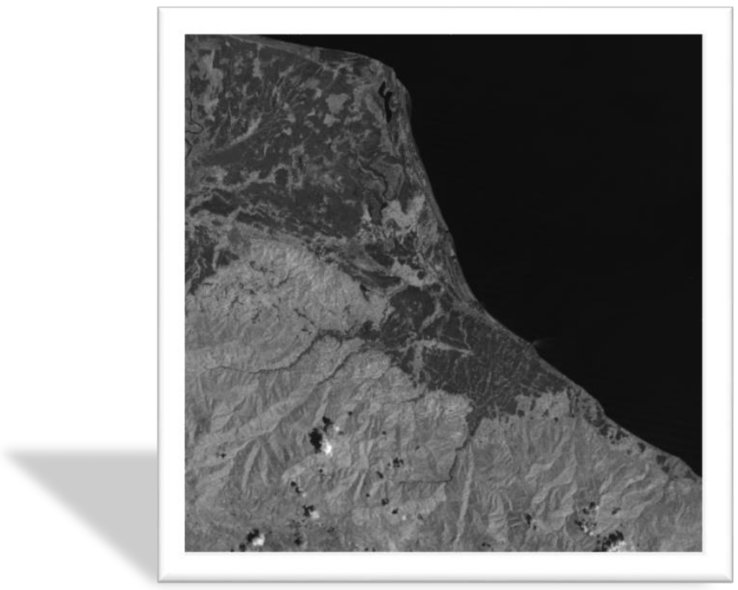


باند 2

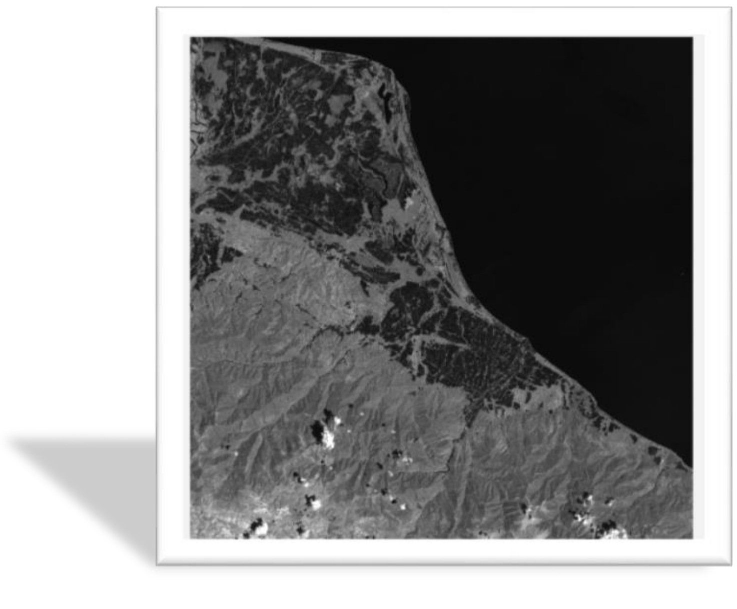




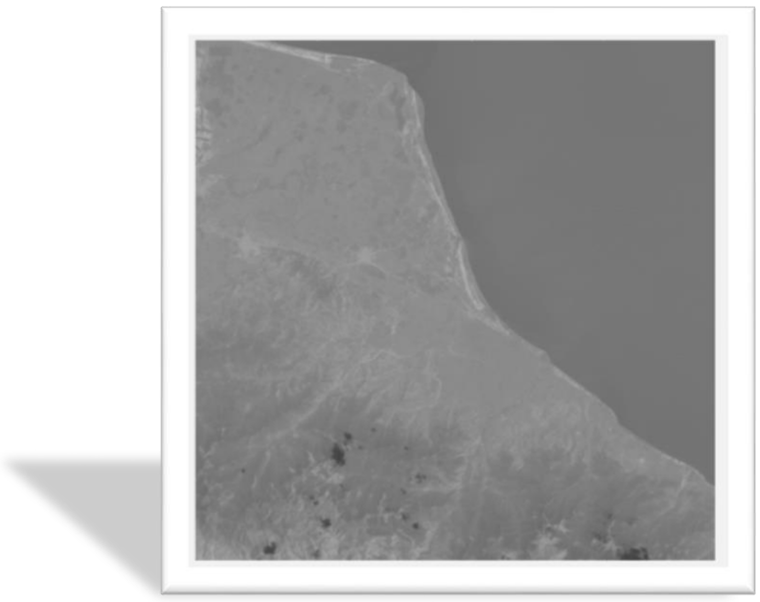
باند 3



باند 4



باند 5



باند 6

باند 7

ITEM 2

کد مورد استفاده :

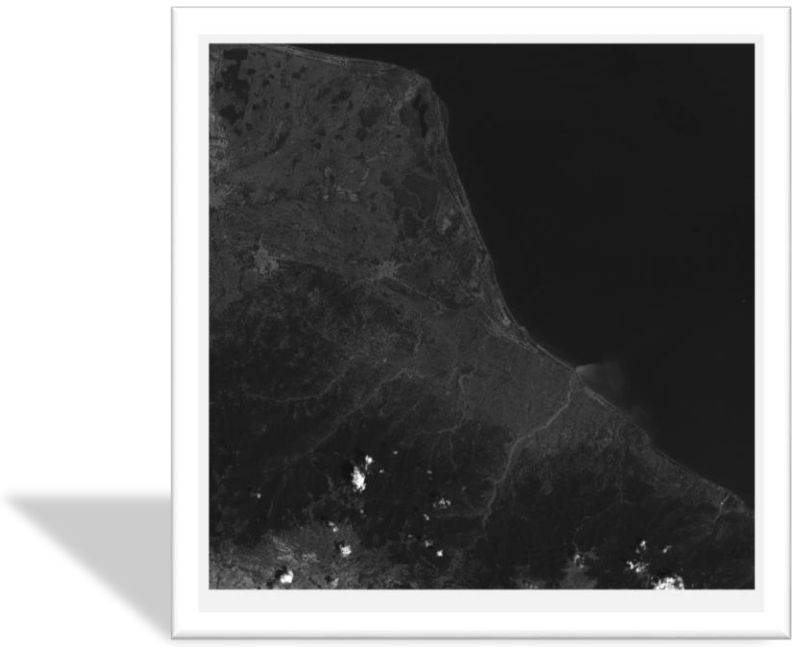
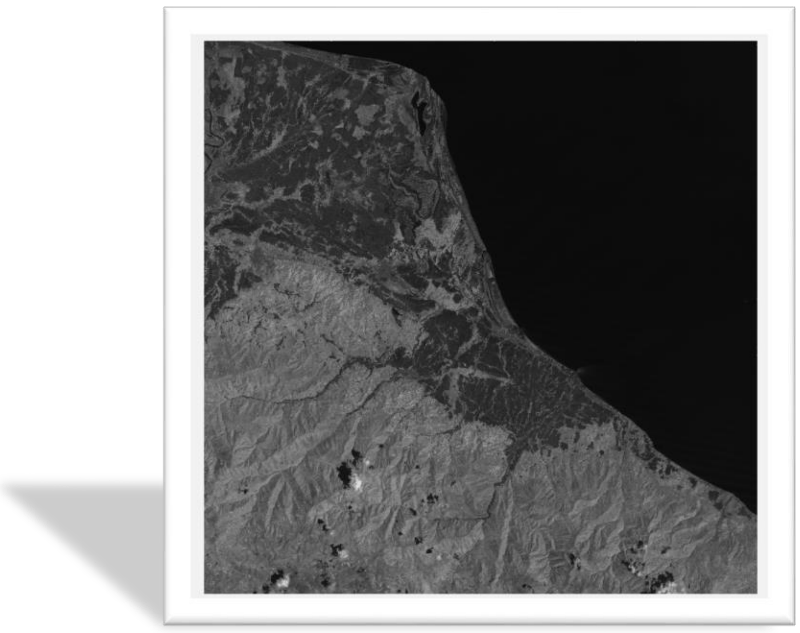
محاسبه مطلوب سوال

```
l_min_r = -1.17 ;
l_max_r = 264 ;
band_r = photo1(:,:,3) ;
radiance_r = ((l_max_r-l_min_r)/255)*band_r + l_min_r ;
l_min_nir = -1.51 ;
l_max_nir = 221 ;
band_nir = photo1(:,:,4) ;
radiance_nir = ((l_max_nir-l_min_nir)/255)*band_nir +
l_min_nir ;
```

نمایش آنها

```
figure('Name','Item 2')
while true
    band_name = input('1-R 2-NIR 3-break') ;
    switch band_name
        case 1
            clf
            imshow(uint8(radiance_r))
        case 2
            clf
            imshow(uint8(radiance_nir))
            case 3
            break ;
    end
end
```

خروجی نتایج :

رادیانس R	
رادیانس NIR	

ITEM 3

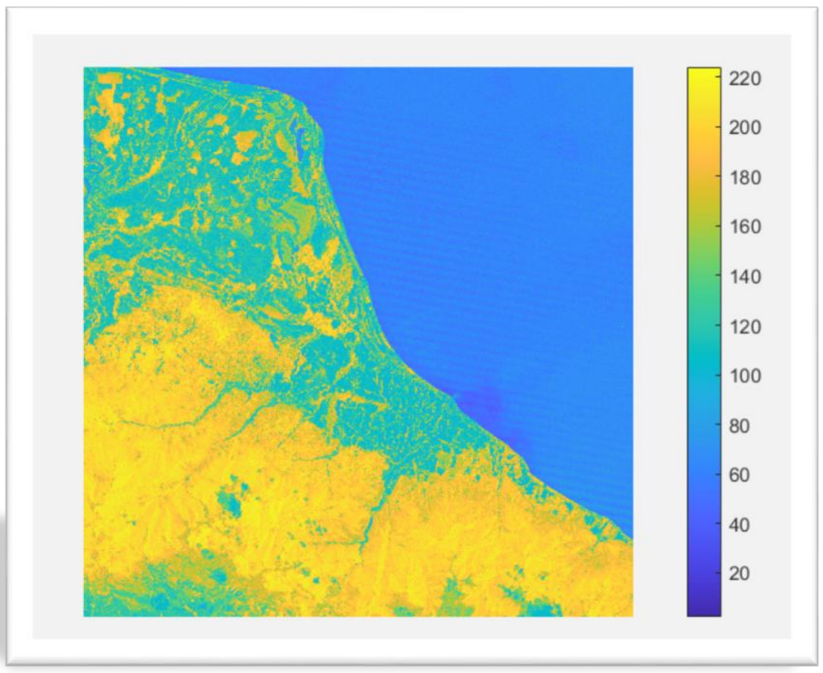
کد های مورد استفاده :

محاسبه NDVI

```
ndvi = (radiance_nir - radiance_r)./(radiance_nir + radiance_r) ;
figure('Name','Item 3')
imagesc((ndvi+1)*127)
axis equal off
colorbar
```

نمایش خروجی ها :

NDVI



ITEM 4

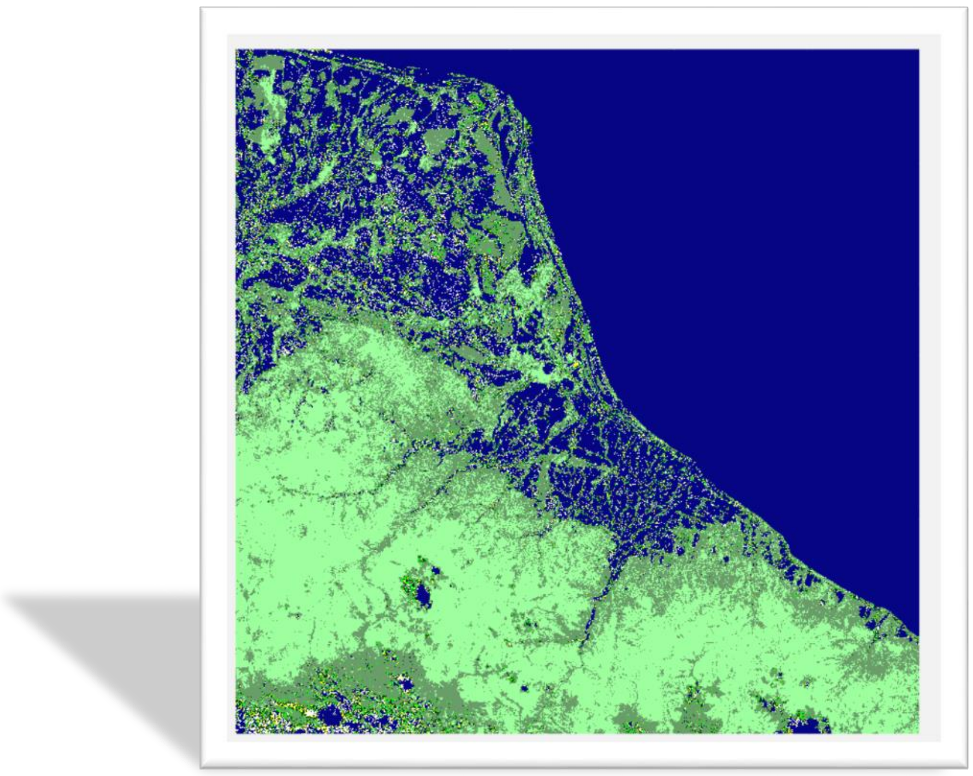
کد های مورد استفاده :

تفکیک رنگی به کمک NDVI

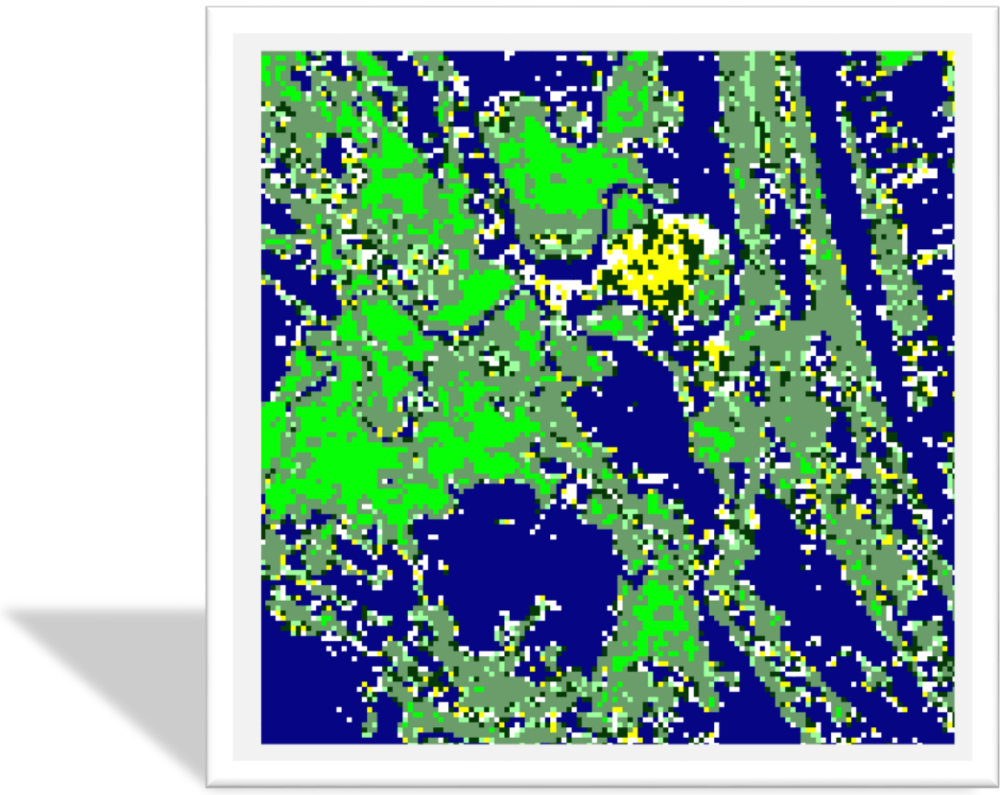
```
colori = zeros(satr,sotoon,3);
ndvi_holder=zeros(satr,sotoon,7);
for i=1:2048
    for  j=1:2048
        if (0.500 < ndvi(i,j)) && (ndvi(i,j)<1 ) % Dense vegetation
            ndvi_holder(i,j,1)=ndvi(i,j);
            colori(i,j,1) = 0.6 ;
            colori(i,j,2) = 1 ;
            colori(i,j,3) = 0.6 ;
        elseif (0.140 < ndvi(i,j)) && (ndvi(i,j)<0.5 ) % Medium vegetation
            ndvi_holder(i,j,2)=ndvi(i,j);
            colori(i,j,1) = 0.4 ;
            colori(i,j,2) = 0.6 ;
            colori(i,j,3) = 0.4 ;
        elseif (0.090 < ndvi(i,j)) && (ndvi(i,j)<0.140 ) % Sparse vegetation
            ndvi_holder(i,j,3)=ndvi(i,j);
            colori(i,j,1) = 0 ;
            colori(i,j,2) = 1 ;
            colori(i,j,3) = 0 ;
        elseif (0.025 < ndvi(i,j)) && (ndvi(i,j)<0.090) % Bare ground
            ndvi_holder(i,j,4)=ndvi(i,j);
            colori(i,j,1) = 0 ;
            colori(i,j,2) = 0.2 ;
            colori(i,j,3) = 0 ;
        elseif (0.002< ndvi(i,j)) && (ndvi(i,j)<0.025) % Cloud
            ndvi_holder(i,j,5)=ndvi(i,j);
            colori(i,j,1) = 1 ;
            colori(i,j,2) = 1 ;
            colori(i,j,3) = 0 ;
        elseif (-0.046 < ndvi(i,j)) && (ndvi(i,j)<0.002) % Ice and snow
            ndvi_holder(i,j,6)=ndvi(i,j);
            colori(i,j,1) = 1 ;
            colori(i,j,2) = 1 ;
            colori(i,j,3) = 1 ;
        elseif (-1 < ndvi(i,j)) && (ndvi(i,j)<-0.046) % Water
            ndvi_holder(i,j,7)=ndvi(i,j);
            colori(i,j,1) = 0 ;
            colori(i,j,2) = 0 ;
            colori(i,j,3) = 0.5 ;
        end
    end
end
figure()
imshow((colori))
```

نمایش خروجی ها :

تقسیم بندی به کمک NDVI



زوم کردن در عکس بالا



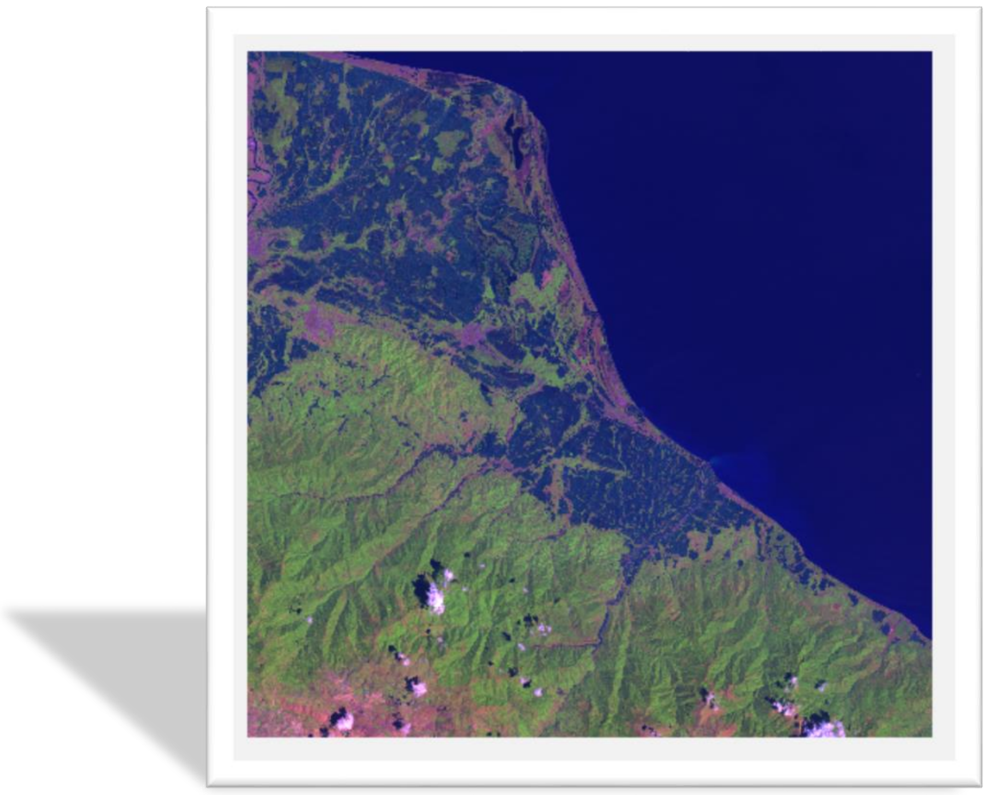
ITEM 5

کد های مورد استفاده :

```
t = 1 ;
for i = 1:7
    for j = 1+i:7
        for k = 1+j:7
            if i==6 ||j==6 ||k==6
                continue
            end
            cov_1 = cov(photo1(:,:,i),photo1(:,:,j)) ;
            cov_2 = cov(photo1(:,:,i),photo1(:,:,k)) ;
            cov_3 = cov(photo1(:,:,j),photo1(:,:,k)) ;
            R = 
corrcoef([photo1(:,:,i),photo1(:,:,j),photo1(:,:,k)]) ;
            oif(t,1:4)
=[i,j,k,((cov_1(1)+cov_2(4)+cov_3(1))/(R(2)+R(3)+R(6))) ] ;
            t = t + 1 ;
        end
    end
end
[oif_max , index_oif_max] = max(oif(:,4)) ;

photo2(:,:,3) = photo1(:,:,oif(index_oif_max,1)) ;
photo2(:,:,2) = photo1(:,:,oif(index_oif_max,2)) ;
photo2(:,:,1) = photo1(:,:,oif(index_oif_max,3)) ;

figure()
imagesc(uint8(photo2))
colormap([linspace(0,1,256)',linspace(0,1,256)',linspace(0,1,256)'])
colorbar
axis equal off
```

محاسبه OIF

نمایش خروجی ها :



نمایش خروجی که مقاد OIF برابر 113 و همچنین متشکل از 3 باند 4 و 5 و 1 است .

ITEM 6 – 13

NDVI

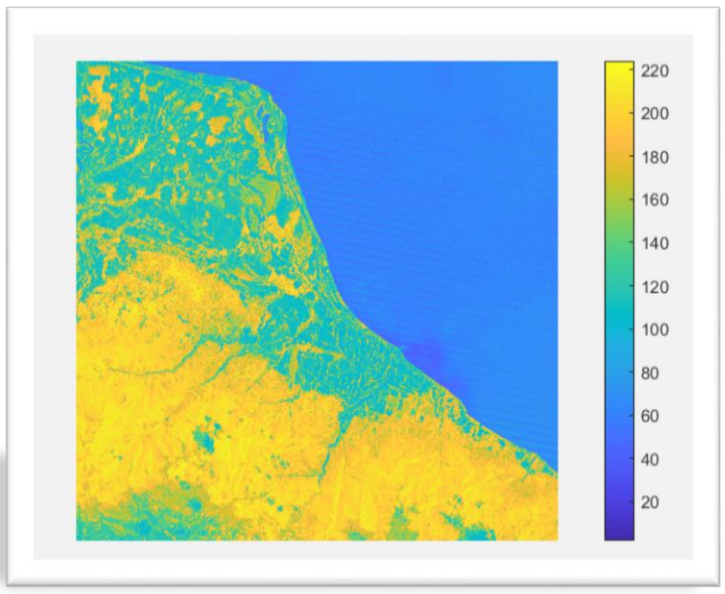
```
ndvi = (radiance_nir - radiance_r)./(radiance_nir +
radiance_r) ;
figure('Name','Item 6')
imagesc((ndvi+1)*127)
axis equal off
```

مقادیر برخی از پیکسل ها (نه مقدار NDVI) :

دریا : 55.63

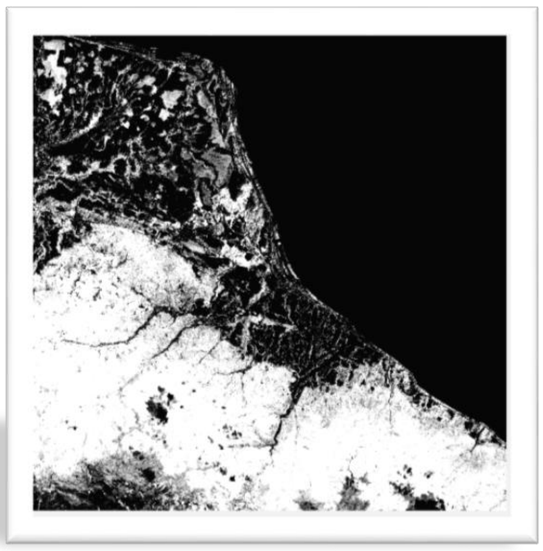
جنگل : 199.4

خشکی :192



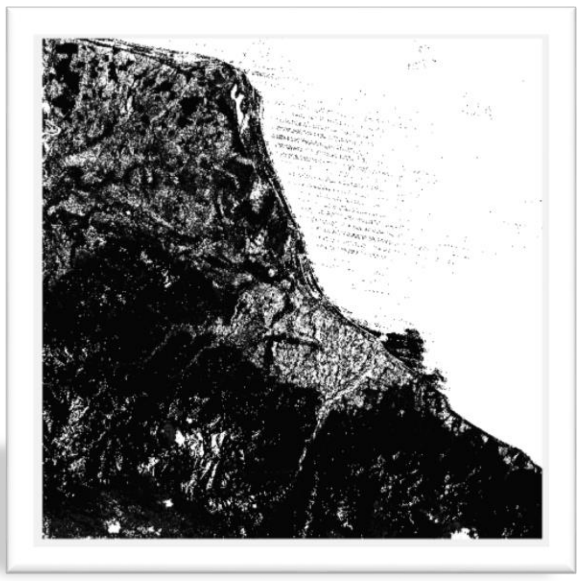
SAVI

```
l_savi = 0.5 ;
savi = (radiance_nir-
radiance_r)*(1+l_savi)./(radiance_nir+radiance_r+l_sav
i) ;
figure('Name','Item 7')
imshow(savi)
axis equal off
```



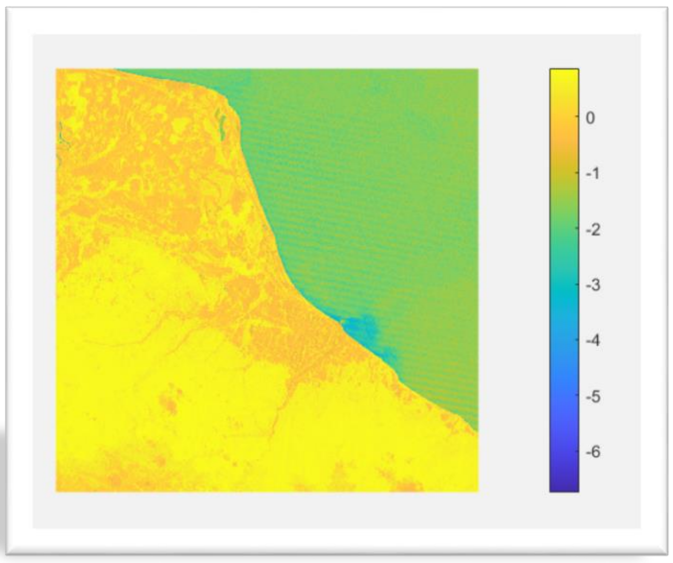
MSAVI

```
l_msavi = 1- ((2*1.17*(radiance_nir-
radiance_r).*(radiance_nir-
1.17*radiance_r))./(radiance_nir+radiance_r)) ;
msavi = (radiance_nir-
radiance_r).*(1+l_msavi)./(radiance_nir+radiance_r+l_m
savi) ;
figure('Name','Item 9')
imshow(msavi)
axis equal off
```



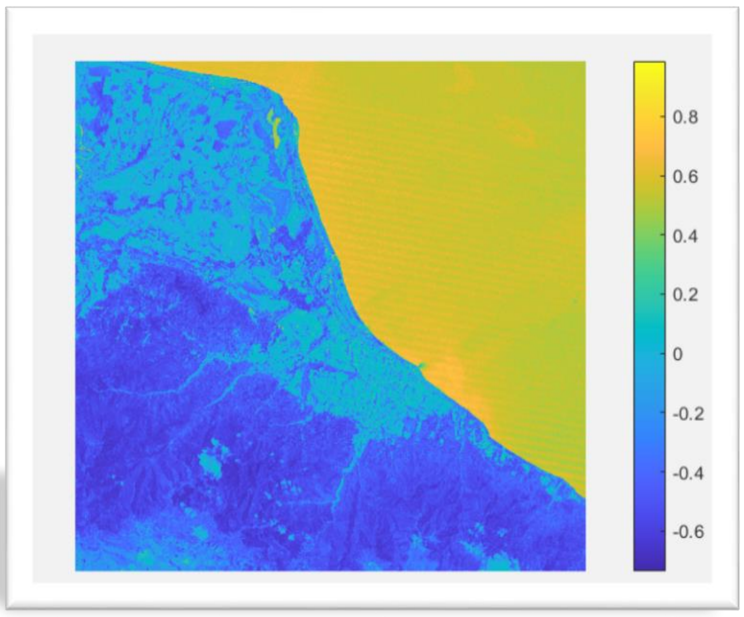
MSAVI 2

```
msavi2 = (2*radiance_nir+1-sqrt((2*radiance_nir+1).^2
-8*(radiance_nir-radiance_r)))/2 ;
figure('Name','Item 9')
imagesc(msavi2)
axis equal off
```



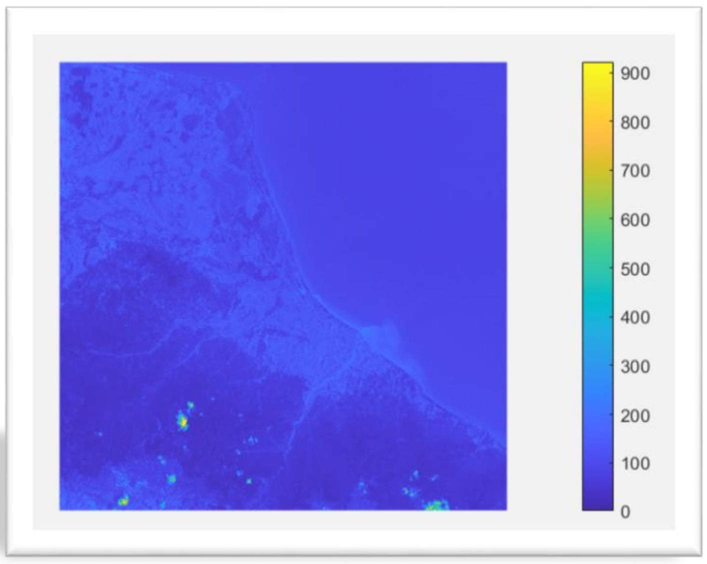
NDWI

```
l_min_g = -2.84 ;
l_max_g = 264 ;
band_g = photo1(:,:,2) ;
radiance_g = ((l_max_g-l_min_g)/255)*band_g + l_min_g
;
ndwi = (radiance_g-
radiance_nir)./(radiance_g+radiance_nir) ;
figure('Name','Item 10')
imagesc(ndwi)
axis equal off
```



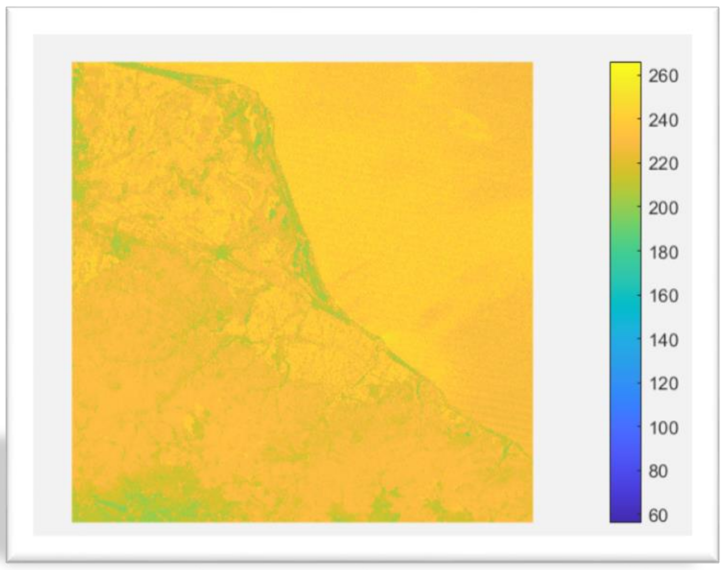
AWEI

```
l_min_swir1 = -0.37 ;
l_max_swir1 = 30.2 ;
band_swir1 = photo1(:,:,5) ;
radiance_swir1 = ((l_max_swir1-
l_min_swir1)/255)*band_swir1 + l_min_swir1 ;
l_min_swir2 = -0.15 ;
l_max_swir2 = 16.5 ;
band_swir2 = photo1(:,:,7) ;
radiance_swir2 = ((l_max_swir2-
l_min_swir2)/255)*band_swir2 + l_min_swir2 ;
awei = 4*(radiance_g-radiance_swir1)-
(0.25*radiance_nir-2.75*radiance_swir2);
figure('Name','Item 11')
imagesc(awei)
axis equal off
colorbar
```



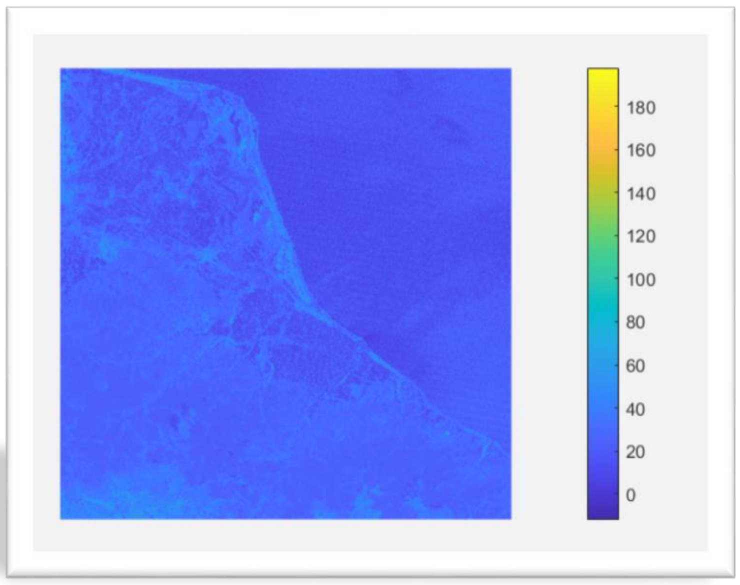
NDMI

```
ndmi = (radiance_nir-
radiance_swir1)./(radiance_swir1+radiance_nir);
figure('Name','Item 12')
imagesc((ndmi+1)*127)
axis equal off
colorbar
```



NDBI

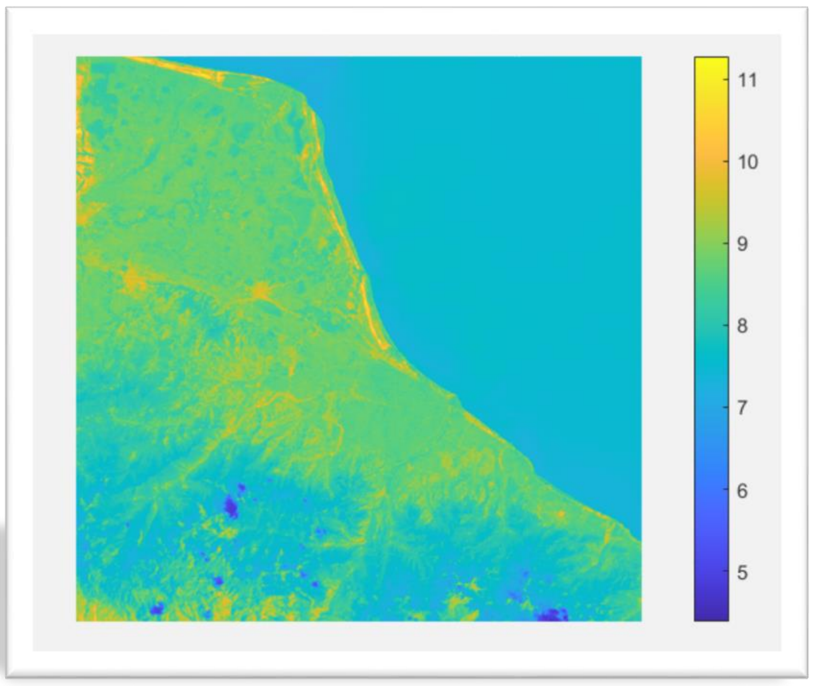
```
ndbi = (-
radiance_nir+radiance_swir1)./(radiance_swir1+radiance
_nir);
figure('Name','Item 13')
imagesc((ndbi+1)*127)
axis equal off
colorbar
```



ITEM 14

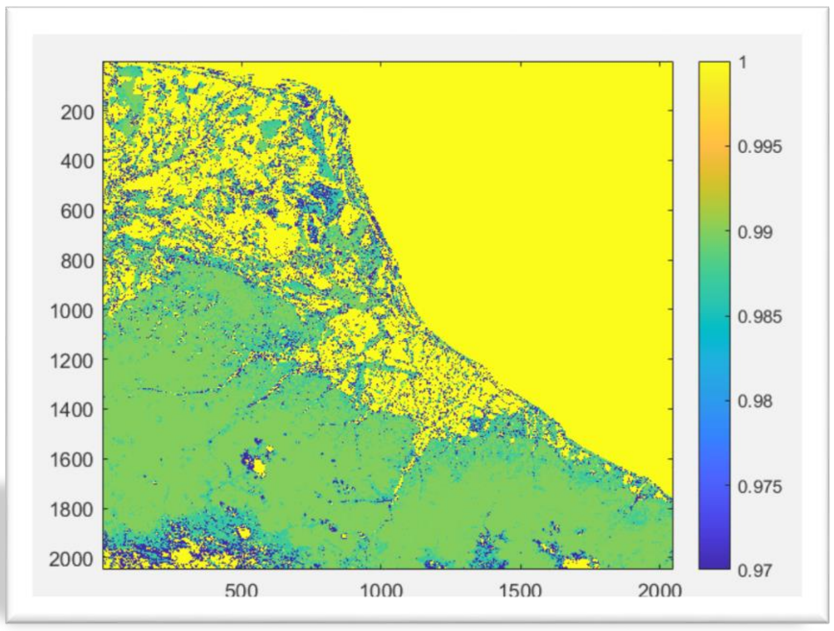
رادیان
س باند
6

```
l_min_tir = 1.2378 ;
l_max_tir = 15.303 ;
band_tir = photo1(:,:,6) ;
radiance_tir = ((l_max_tir-l_min_tir)/255)*band_tir + l_min_tir ;
figure
imagesc(radiance_tir )
```



ضریب
گسیل

```
for i = 1:satr
for j = 1:sotoon
if 0.5 <= ndvi(i,j)
eps1(i,j) = 0.99 ;
elseif  0.2<=ndvi(i,j) && ndvi(i,j)< 0.5
eps1(i,j) = 0.004*((ndvi(i,j)-0.2)/0.3)^2 +0.986 ;
elseif 0<=ndvi(i,j) && ndvi(i,j)<0.2
eps1(i,j) = 0.97 ;
elseif ndvi(i,j)<0
eps1(i,j) = 1 ;
end
end
end
figure
imagesc(eps1 )
colorbar
```



دما (به
واحد
کلوین
(

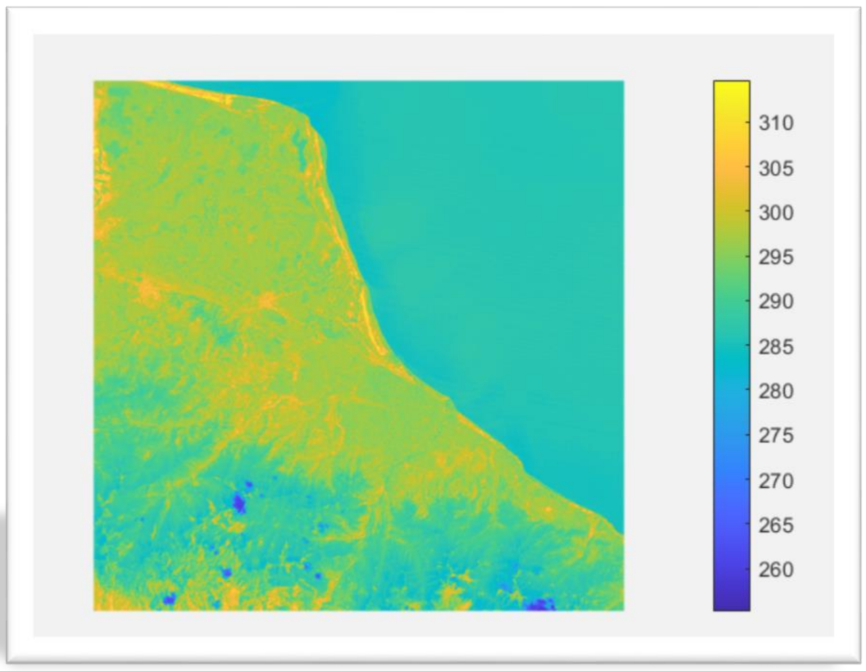
```
for i = 1:satr
for j = 1:sotoon
tt(i,j) = 1260.56/log((607.76*eps1(i,j)/radiance_tir(i,j))+1) ;
end
end
figure('Name','Item 14')
imagesc(tt)
axis equal off
colorbar
```

نمایش مقادیر چند پیکسل :

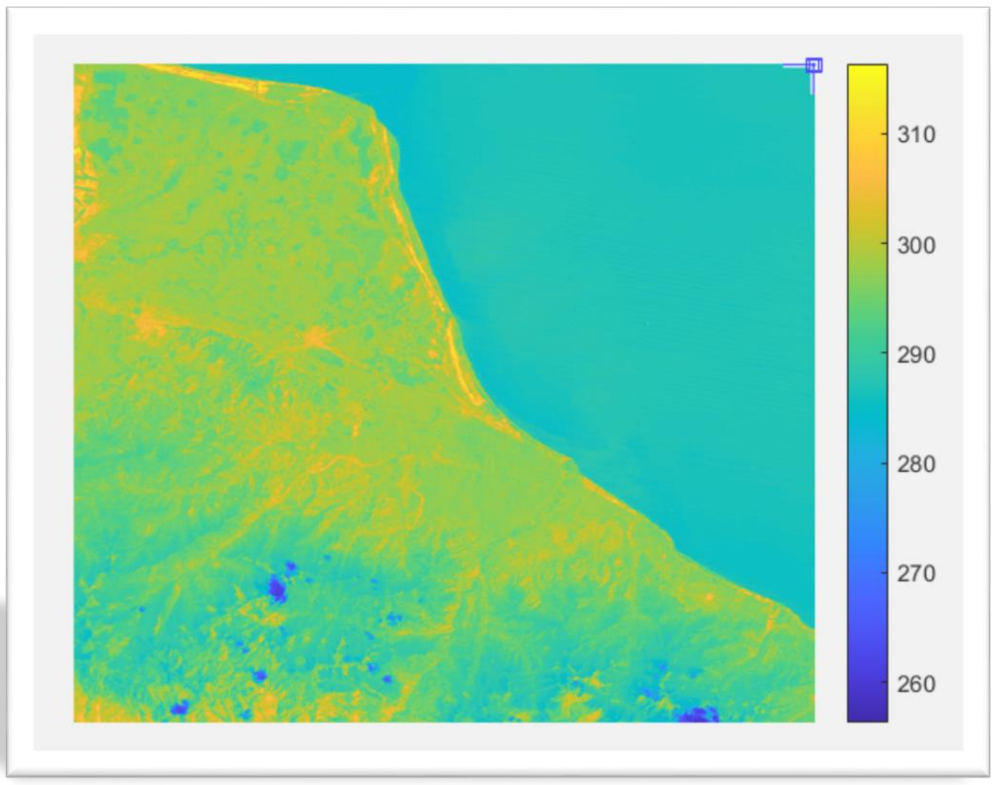
دریا : 280

جنگل : 292

خشکی : 298



ITEM 15



```
[mask,bw] = likelihood(photo1,3,'on') ;
l_min_tir = 1.2378 ;
l_max_tir = 15.303 ;
band_tir = photo1(:,:,6) ;
radiance_tir = ((l_max_tir-
l_min_tir)/255)*band_tir + l_min_tir ;
for i =1:satr
    for j =1:sotoon
        if mask(i,j,1)==1 % sea
            eps = 0.99 ;
        elseif mask(i,j,2)==0  %wet soil
            eps = 0.95 ;
        elseif mask(i,j,3)==0  %forest
            eps = 0.979 ;
        end
        t2(i,j) =
1260.56/log((607.76*eps/radiance_tir(i,j))+1)
;
    end
end
figure()
imagesc(t2)
colorbar()
```

تابع likelihood به در بخش مربوطه تفضیل توضیح داده خواهد شد .

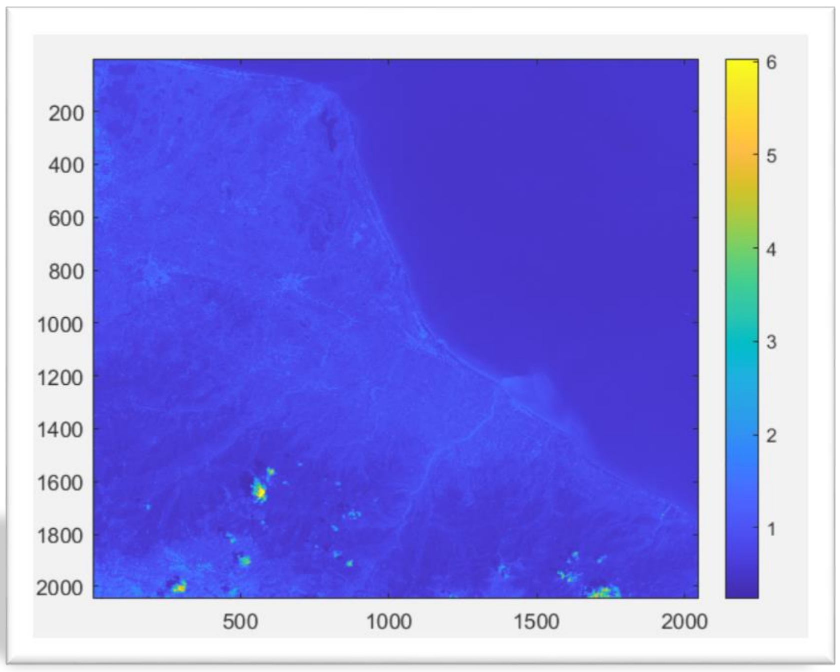
نمایش مقادیر چند پیکسل :

دریا : 280

جنگل : 292

خشکی : 298

ITEM 16



```
day = 152 ;
d_day = 1-0.01672*cosd(0.9856*(day-4)) ;
ro1 = (pi*(0.7628235*photo1(:,:,1)-
1.52)*d_day^2)/(195.7*0.77) ;
ro2 = (pi*(radiance_g)*d_day^2)/(182.9*0.77) ;
ro3 = (pi*(radiance_r)*d_day^2)/(155.7*0.77) ;
ro4 = (pi*(radiance_nir)*d_day^2)/(104.7*0.77)
;
ro5 = (pi*(radiance_swir1))/(21.93*0.77) ;
ro7 = (pi*(radiance_swir2))/(7.452*0.77) ;
imagesc(ro2)
colorbar()
```

کد استفاده شده :

Parallelepiped Classification

```
function [mask2,bw] = Parallelepiped( photo1,ploting)
[var_mat,mean_mat,x_c,y_c,bw] = mean_var(photo1) ;
[satr,sotoon , band] = size(photo1);
mask = zeros(satr,sotoon,band) ;

for i=1:satr
    for j=1:sotoon
        for k = 1:band
            if mean_mat(k) - var_mat(k) <=photo1(i,j,k) && photo1(i,j,k)
<=mean_mat(k) + var_mat(k)
                mask(i,j,k) = 1 ;
            end
        end
    end
end
mask2 = zeros(satr,sotoon) ;
for i=1:satr
    for j=1:sotoon
        if sum(mask(i,j,:)) == band
        mask2(i,j) =1 ;
        end
    end
end
if ploting =='on'
    figure()
    imagesc(mask2)
    hold on
    plot(x_c,y_c,'r*')
    colorbar()
    axis equal
    axis off
end




end
```

Minimum Distance Classification

```
function [mask,bw] = Minimum_Distance(photo1,class_num,ploting )

[satr,sotoon , band] = size(photo1);
mean_mat = zeros(band,1,class_num) ;
x_c = zeros(band,1,class_num) ;
y_c = zeros(band,1,class_num) ;
for i = 1:class_num
    [~,mean_mat(1:band,1,i),x_c(i),y_c(i),bw(:,:,i)] = mean_var(photo1)
;
    var_cov_mask(:,:,i) = inv( mask_cov_mat(photo1,bw(:,:,i))) ;
end
mask = zeros(satr,sotoon,class_num) ;
for i=1:satr
    for j=1:sotoon
        for k = 1:class_num
            x_mat = [ photo1(i,j,1) ;
                      photo1(i,j,2) ;
                      photo1(i,j,3) ;
                      photo1(i,j,4) ;
                      photo1(i,j,5) ;
                      photo1(i,j,6) ;
                      photo1(i,j,7) ] ;
            norm_mat(k) = (x_mat -
mean_mat(:,1,k))'*var_cov_mask(:,:,k)*(x_mat - mean_mat(:,1,k) );


        end
        [~,index] = min(norm_mat) ;
        mask(i,j,index) = 1 ;

    end
end
if ploting=='on'
    figure()
    for i=1:class_num
        subplot(1,class_num,i)
        imagesc(mask(:,:,i))
        hold on
        plot(x_c(i),y_c(i),'r*')
        axis off
        axis equal
    end

    colorbar()
end
end
```

```
function [mask,bw] = likelihood(photo1,class_num,ploting)
[satr,sotoon , ~] = size(photo1);

bw = zeros(satr,sotoon,class_num) ;
for i =1:class_num
    [bw(:,:,i),xi2,yi2] = roipoly(uint8(photo1(:,:,1:3)));
    x_c(i,1) = sum(xi2)/length(xi2) ;
    y_c(i,1) = sum(yi2)/length(yi2) ;
    close all
    mean_vec = sum(sum(bw(:,:,i).*photo1 ))./sum(sum(bw(:,:,i))) ;
    for j=1:length(mean_vec)
        mean_mat(j,1,i) = mean_vec(j) ;
    end
    var_cov_mask(:,:,i) = mask_cov_mat(photo1,bw(:,:,i)) ;
end

mask = zeros(satr,sotoon,class_num) ;
for k = 1:class_num
   ldvcm(k,1) = -log(det(var_cov_mask(:,:,k)));
   inv_vcm(:,:,k) = inv(var_cov_mask(:,:,k));
end
for i=1:satr
    for j=1:sotoon
        for k = 1:class_num
            normal_mat = [photo1(i,j,1) - mean_mat(1,1,k),
                          photo1(i,j,2) - mean_mat(2,1,k),
                          photo1(i,j,3) - mean_mat(3,1,k),
                          photo1(i,j,4) - mean_mat(4,1,k),
                          photo1(i,j,5) - mean_mat(5,1,k),
                          photo1(i,j,6) - mean_mat(6,1,k),
                          photo1(i,j,7) - mean_mat(7,1,k)];
            g_mat(k) = -ldvcm(k,1) -
normal_mat'*inv_vcm(:,:,k)*normal_mat;
        end
        [~,index] = max(g_mat) ;
        mask(i,j,index) = 1 ;
    end
end
if ploting=='on'
    figure()
    for i=1:class_num
        subplot(1,class_num,i)
        imagesc(mask(:,:,i))
        hold on
        plot(x_c(i),y_c(i),'r*')
        axis off
        axis equal
    end
    colorbar()
end
end
```

Maximum Likelihood Classification

```
function [mask,bw] = Mahalanobis(photo1,class_num,ploting)
 [satr,sotoon , ~] = size(photo1);
bw = zeros(satr,sotoon,class_num) ;
for i =1:class_num
    [bw(:,:,i),xi2,yi2] = roipoly(uint8(photo1(:,:,1:3)));
    x_c(i,1) = sum(xi2)/length(xi2) ;
    y_c(i,1) = sum(yi2)/length(yi2) ;
    close all
    mean_vec = sum(sum(bw(:,:,i).*photo1 ))./sum(sum(bw(:,:,i))) ;
    for j=1:length(mean_vec)
        mean_mat(j,1,i) = mean_vec(j) ;
    end
    var_cov_mask(:,:,i) = mask_cov_mat(photo1,bw(:,:,i)) ;
end
var_cov_mat = photo_cov_mat(photo1) ;

mask = zeros(satr,sotoon,class_num) ;
   inv_vcm = inv(var_cov_mat);
for i=1:satr
    for j=1:sotoon
        for k = 1:class_num
            normal_mat = [photo1(i,j,1) - mean_mat(1,1,k),
                          photo1(i,j,2) - mean_mat(2,1,k),
```

Mahalanobis Classification

```
                          photo1(i,j,3) - mean_mat(3,1,k),
                          photo1(i,j,4) - mean_mat(4,1,k),
                          photo1(i,j,5) - mean_mat(5,1,k),
                          photo1(i,j,6) - mean_mat(6,1,k),
                          photo1(i,j,7) - mean_mat(7,1,k)];
            g_mat(k) = normal_mat'*inv_vcm*normal_mat ;
        end
        [~,index] = min(g_mat) ;
        mask(i,j,index) = 1 ;

    end
end

if ploting=='on'
    figure()
    for i=1:class_num
        subplot(1,class_num,i)
        imagesc(mask(:,:,i))
        hold on
        plot(x_c(i),y_c(i),'r*')
        axis off
    end
    colorbar()
end
end
```

دقت کلی

```
function [overall_acuracy] = overall_accuracy(photo1,mask)
[~,~,mband] = size(mask) ;
for i = 1:mband
    mask2 = mask(:,:,i) ;
    figure()
    [mask_test,~,~] = roipoly(uint8(photo1(:,:,1:3)));
    mask_test = logical(mask_test) ;
    mask_train = logical(mask2(mask_test)) ;
    aa(i) = sum(mask_train)*100 ;
    bb(i) = sum(sum( mask_test)) ;
    close all
end
overall_acuracy = sum(aa)/sum(bb) ;
end
```

ضریب کاپا

```
function [kappa] = kappa_accuracy(mask,bw)
[~,~,class_num ] = size(bw) ;
error_mat = zeros(class_num) ;
for k = 1:class_num
    for t = 1:class_num
        mask2 = mask(:,:,k);
        bw2 = bw(:,:,t) ;
        error_mat(k,t) = sum(sum(mask2(logical(bw2))));
    end
end
ii_mat = diag(error_mat) ;
i_plus = sum(error_mat) ;
plus_i = sum(error_mat') ;
n_num = sum(sum(sum(bw))) ;

kappa = (n_num*sum(ii_mat)-sum(i_plus.*plus_i) )/(n_num^2-
sum(i_plus.*plus_i) ) ;

end
```

تابع های استفاده شده :

محاسبه کوورریانس ماسک ورودی

```
function [var_cov_mask] =
mask_cov_mat(photo1,mask)

X = photo1;
X_size = size(X);

mask = repmat(mask, [1 1 X_size(3)]);
mask = logical(mask);
X_data = X(mask);
X_data = reshape(X_data,
[length(X_data)/X_size(3) 1 X_size(3)]);
var_cov_mask = photo_cov_mat(X_data);

end
```

عکس را به عنوان ورودی می گیرد و کاربر یک منطقه را انتخاب می کند و تابع به عنوان خروجی مرکز پولی گون ، میانگین و واریانس پیکسل های موجود در ماسک را میدهد .

```
function [var_mat,mean_mat,x_c,y_c,bw] =
mean_var(img1)

[bw,xi2,yi2] = roipoly(uint8(img1(:,:,1:3)));
x_c = round(sum(xi2)/length(xi2)) ;
y_c = round(sum(yi2)/length(yi2)) ;
img1 = im2double(img1);
mean_vec = sum(sum(bw.*img1 ))/sum(sum(bw)) ;
for i=1:length(mean_vec)
    mean_mat(i,1) = mean_vec(i) ;
end
[satr,sotoon , band] = size(img1) ;
for i=1:band
    r = 0 ;
    for j=1:satr
        for k=1:sotoon
            if bw(j,k) == 1
                r = r + 1 ;
                holder(r,1) = img1(j,k,i) ;
            end
        end
    end
    var_mat(i,1) = var(holder) ;
end
var_mat = sqrt(var_mat);
close all
end
```

محاسبه ماتریس واریانس-کوورریانس بین باند های مختلف یک عکس

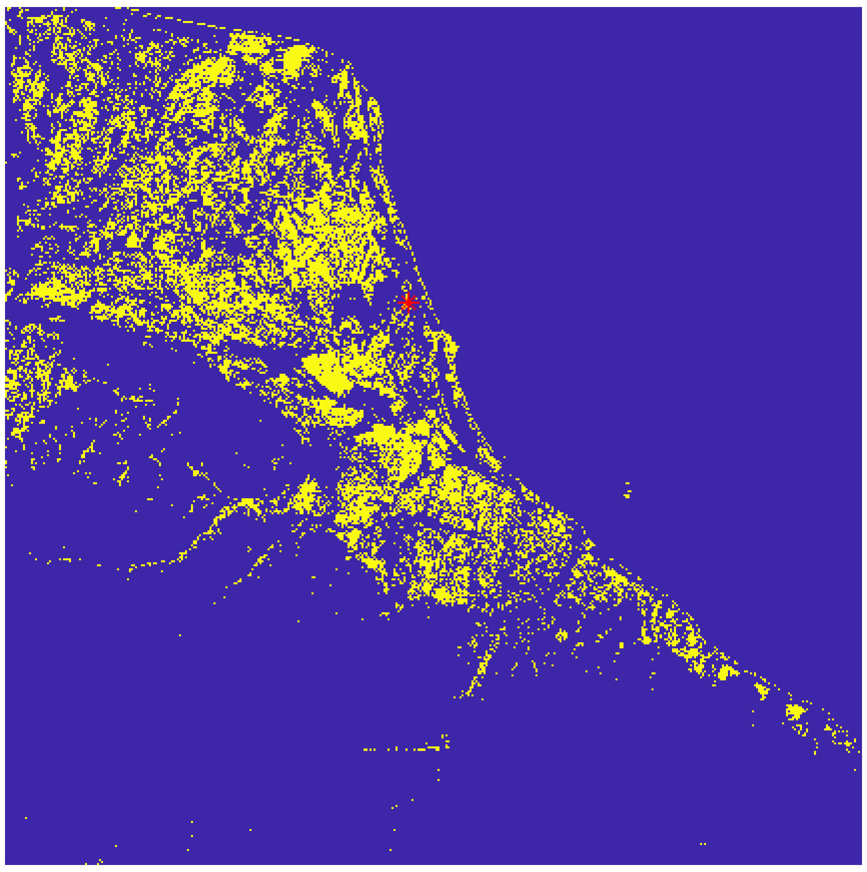
```
function [ var_cov_mat  ] = photo_cov_mat(
photo1 )

[satr,sotoo,band] = size(photo1) ;
var_cov_mat = zeros(band,band);
for i = 1 :band
    for j = i : band
        covar =
cov(photo1(:,:,i),photo1(:,:,j)) ;
        var_cov_mat(i,i) = covar(1,1) ;
        var_cov_mat(i,j) = covar(1,2) ;
        var_cov_mat(j,i) = covar(2,1) ;
        var_cov_mat(j,j) = covar(2,2) ;
    end
end
end
```

خروجی ها : (دقت شود که مناطق زرد مناطق مطلوب هر کلاس هستند و مرکز پولی گون هر داده آموزشی با ستاره قرمز مشخص شده است .)

Parallelepiped Classification

*(توجه شود که کلاس بندی های دیگر به
صورت مقایسه ای عمل می کنند ولی متوازی
السطوح به صورت مطلق کلاسبندی می کند پس
کد را به شکلی نوشتم که با هر بار اجرا کردن
آن بتوان فقط یک کلاس را جدا کرد .)*

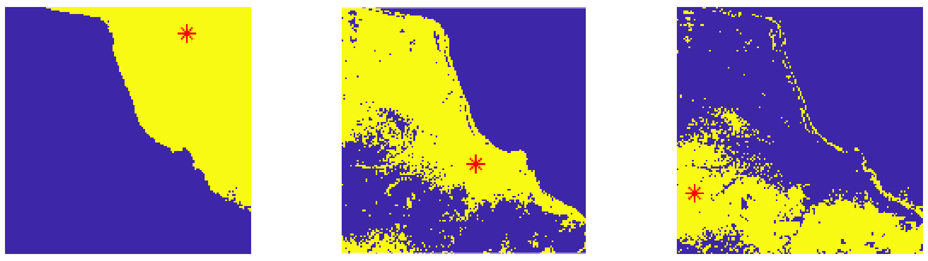
ضریب کاپا = 0.42
دقت کلی = 56 درصد
**(این نتایج برای تشخیص پهنه خشکی
ست ، برای دریا دقت کلی 44 درصد و کاپا
0.38 و برای پهنه جنگل دقت کلی 49
درصد و کاپا 0.53 است)**



Minimum Distance Classification

(محاسبه به صورت وزندار)
دقت کلی = 98 %

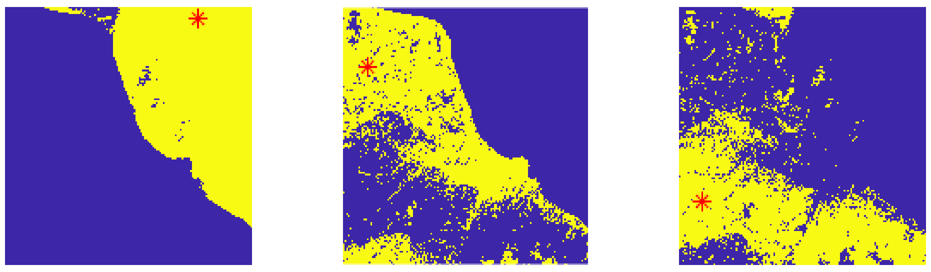
ضریب کاپا = 0..97



Maximum Likelihood Classification

دقت کلی = 95 %

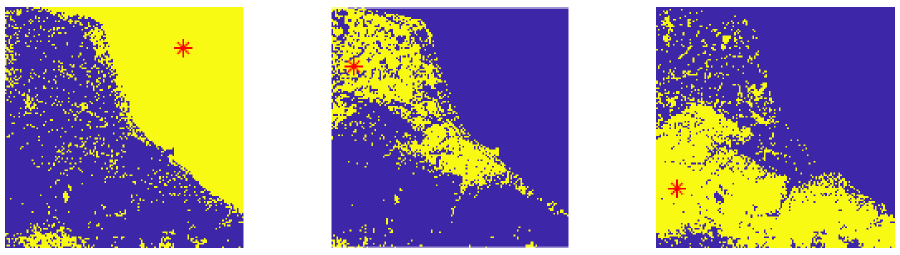
ضریب کاپا = 0.94



Mahalanobis Classification

دقت کلی = 88 ٪

ضریب کاپا = 0.93



دقت شود که مناطق زرد مناطق مطلوب هر کلاس هستند و مرکز پولی گون هر داده آموزشی با ستاره قرمز مشخص شده است .

کد کلی 4 بخش آخر :

```
clc
clear
close all
satr   = 2048 ;
sotoon = 2048 ;
photo1(:,:,1) = datopen('Data1\b1.dat',satr,sotoon) ;
photo1(:,:,2) = datopen('Data1\b2.dat',satr,sotoon) ;
photo1(:,:,3) = datopen('Data1\b3.dat',satr,sotoon) ;
photo1(:,:,4) = datopen('Data1\b4.dat',satr,sotoon) ;
photo1(:,:,5) = datopen('Data1\b5.dat',satr,sotoon) ;
photo1(:,:,6) = datopen('Data1\b6.dat',satr,sotoon) ;
photo1(:,:,7) = datopen('Data1\b7.dat',satr,sotoon) ;
%% Parallelepiped
[mask,bw] = Parallelepiped( photo1,'on');
overall_acuracy_Parallelepiped = overall_accuracy(photo1,mask)
kapa_Parallelepiped = kappa(mask,photo1)
%% Minimum Distance
[mask,bw] = Minimum_Distance(photo1,3,'on' );
overall_acuracy_Minimum_Distance = overall_accuracy(photo1,mask)
kapa_Minimum_Distance = kappa(mask,photo1)
% %% likelihood
[mask,bw] = likelihood(photo1,3,'on') ;
overall_acuracy_likelihood = overall_accuracy(photo1,mask)
kapa_likelihood = kappa(mask,photo1)
%% Mahalanobis
[mask,bw] = Mahalanobis(photo1,3,'on') ;
overall_acuracy_Mahalanobis = overall_accuracy(photo1,mask)
kapa_Mahalanobis = kappa(mask,photo1)
```

ورودی تابع Parallelepiped عکس مورد نظر و همچنین فعال یا غیر فعال بودن نمایش کلاسبندی است ، و برای سه تابع دیگر عکس ، تعداد کلاس و وضعیت نمایش کلاسبندی است .