

Képfeldolgozás haladóknak

1. gyakorlat: A Matlab és az Image Processing Toolbox használata a képfeldolgozásban

A Matlab rendszert hatékonysága miatt számos területen használják. Segíti a gyors fejlesztést, hatékonyan vannak benne implementálva a nagy számításigényű algoritmusok. A Matlabhoz készült jónéhány olyan programcsomag, amelyben az alkalmazott informatika egyes területén közismertté vált algoritmusokat, számításokat, fejlesztési eszközöket gyűjtötték össze. Az egyik ilyen programcsomag az **Image Processing Toolbox**.

A következőkben azt tekintjük át, hogyan lehet egyszerű képműveleteket végezni Matlabban.

Képek betöltése:

A képek betöltésére Matlabban az `imread(...)` függvény szolgál. A függvény visszatérési értéke egy képmátrix. A függvényt az alábbi módokon használhatjuk:

```
img = imread('filenév');
```

Megnyitja a paraméterként megadott képfájlt. A formátumot a kiterjesztésből határozza meg. A visszatérési érték egy képmátrix, amelynek mérete a kép típusától függhet. Szürkeárnyalatos képek esetében egy $M \times N$ -es mátrixot jelent. Színes képek esetében, ahol három színcsatorna van, ott az eredmény egy $M \times N \times 3$ -as képmátrix, ha pedig 4 színcsatornája van egy képnek (pl. CMYK színtérben), akkor $M \times N \times 4$ a képmátrix mérete.

```
img = imread('filenév', 'formátumnév')
```

Megnyitja a paraméterként megadott képfájlt. A formátumot sztringként megadhatjuk második paraméterként. Ez bizonyos képeknél (pl. DICOM) nem derül mindig ki a kiterjesztésből. A képmátrix méretét illetően ugyanaz a helyzet, mint az előbbi esetben.

```
[X, map] = imread(...)
```

A függvény paramétereként megadhatjuk azokat a paramétereket, amelyeket az előző esetekben. A függvény visszatérési értéke tartalmazza a képmátrixot és a hozzá tartozó színskálát/szín térképet.

```
img = imread(URL, ...)
```

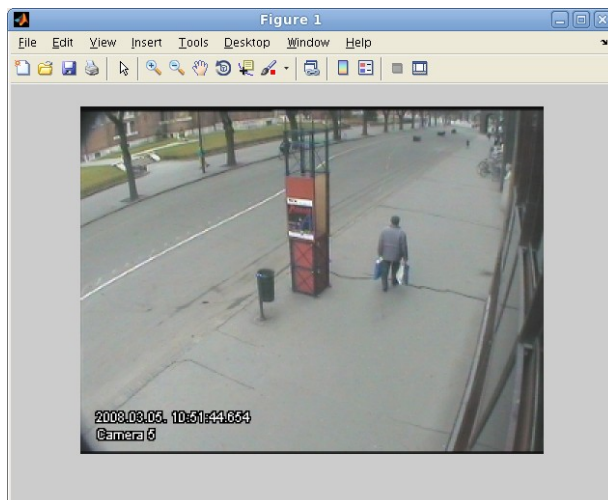
```
[X, map] = imread(URL, ...)
```

A függvény paraméterként egy URL-t kap, és az ott található képet nyitja meg. Megadható még további paraméter is, pl. a formátum.

A képek megjelenítésére az `imshow(img)` függvény szolgál. Paraméterként a képmátrixot kell neki megadni.

Egy egyszerű példaprogram a képek megnyitására és megjelenítésére:

```
>> img = imread('cam1.jpg');  
>> imshow(img)
```



Képek mentése is hasonlóképpen történik. A képek mentésére az `imwrite(...)` függvény szolgál.

```
imwrite(img, 'fajlnév')
```

Elmenti az `img` képmátrixot egy megadott nevű fájlba.

```
imwrite(img, 'fajlnév', 'formátum')
```

Elmenti az `img` képmátrixot egy megadott nevű fájlba adott formátumban.

```
imwrite(X, map, 'filenév', 'formátum')
```

A `X`-ben lévő képmátrixot a `map` szerinti színtérképpel színezi ki és az így kapott képet lementi a megadott fájlba.

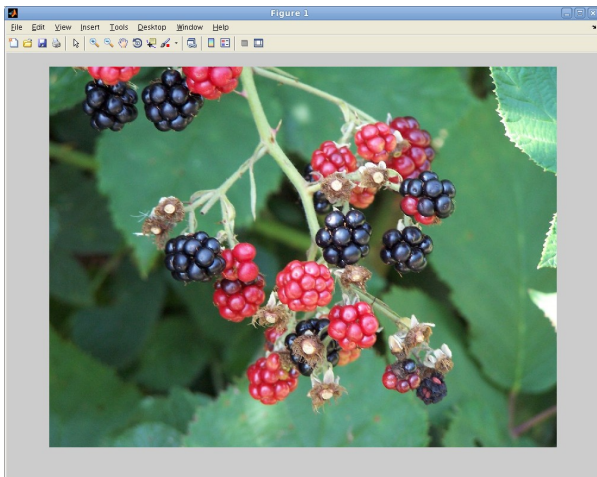
Pixelértékek lekérdezése és manipulálása:

Amint már említettük, a képfájl megnyitása után egy képmátrixot kapunk. A képmátrix méretét és típusát megtaláljuk a Matlab Workspace ablakában, ahol a változók vannak felsorolva, vagy pedig lekérdezhjük a `size(...)` függvénnyel. A `size(...)` függvény visszatérési értéke egy egydimenziós tömb, amely a kép méret attribútumait tartalmazza.

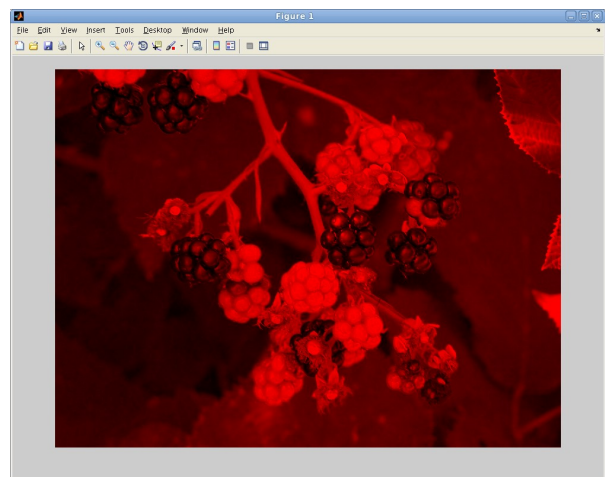
Ha végig szeretnénk menni egy képen, és egy új képbe szeretnénk kimenteni a vörös csatorna értékeit, akkor a következőképpen kell tennünk:

(parancssorban:)

```
>> img = imread('szeder.jpg');  
>> img_size = size(img);  
>> width = img_size(1);  
>> height = img_size(2);  
>> img_red = zeros(img_size, 'uint8');  
>> for y=1:height, for x=1:width, img_red(x,y,1) = img(x,y,1); end; end;  
>> imshow(img_red);
```



Eredeti kép



A kép vörös csatornája

Természetesen a Matlabban létezik ennek a műveletnek egy hatékonyabb formája is a két `for`-ciklus helyett:

```
>> img_green = zeros(img_size, 'uint8');  
>> img_green(:,:,2) = img(:,:,2);
```

Az ilyen, illetve ennél összetettebb képműveleteket is függvényben érdemes megírni. A **File / New / Function M-file** menüpontra kattintva egy új szerkesztőablak nyílik meg, amely már tartalmazza a függvény fejléc mintáját. Ezt természetesen át kell írni.

1. példa:

Készítsünk egy olyan függvényt, amely szétválasztja egy RGB kép három csatornáját! Az eredmény mátrixok mérete változni fog, most mind a három szürkeárnyalatos kép lesz és csak egy csatornával fog rendelkezni.

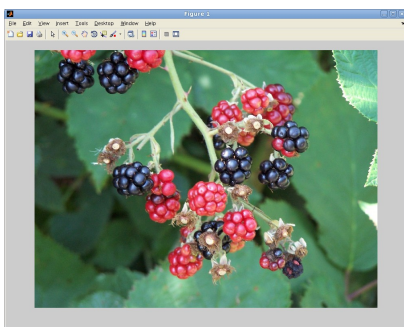
```
function [ red, green, blue ] = rgb_decompose( image )

img_size = size(image);
if img_size(3) == 3 %% ellenorizzuk a kep csatornainak szamat
    red = zeros(img_size(1),img_size(2), 'uint8');
    green = zeros(img_size(1),img_size(2), 'uint8');
    blue = zeros(img_size(1),img_size(2), 'uint8');
    red(:, :) = image(:, :,1);
    green(:, :) = image(:, :,2);
    blue(:, :) = image(:, :,3);
else %% ha nem 3 csatornas a kep, akkor a visszateresi ertekek 0-k
    red = 0;
    green = 0;
    blue = 0;
end;

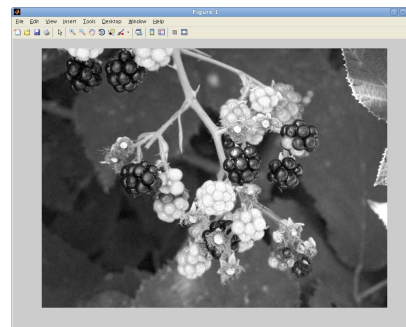
end
```

A függvényt a következőképpen hívhatjuk meg:

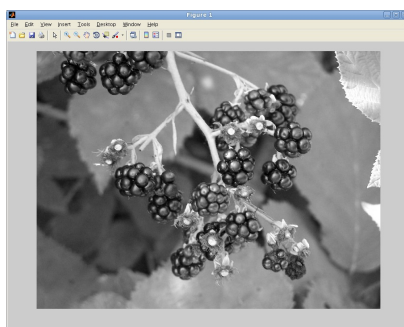
```
>> img = imread('szeder.jpg');
>> [red, green, blue] = rgb_decompose( img);
>> imshow(red);
>> imshow(green);
>> imshow(blue);
```



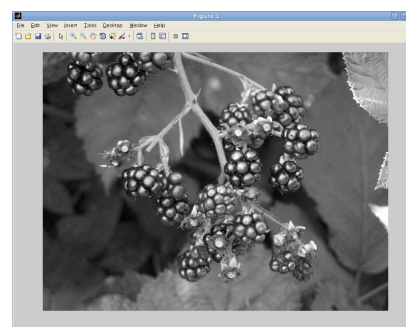
Eredeti kép



A vörös csatorna intenzitásai



A zöld csatorna intenzitásai



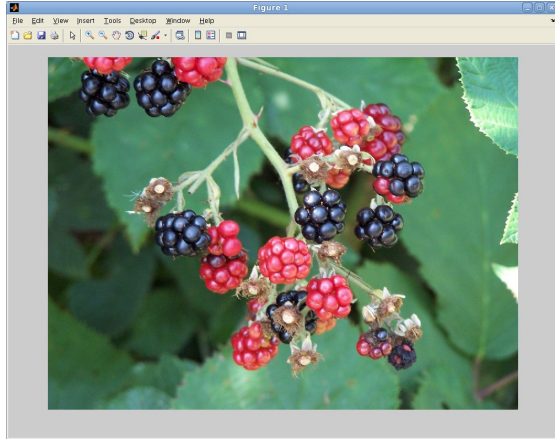
A kék csatorna intenzitásai

2. példa:

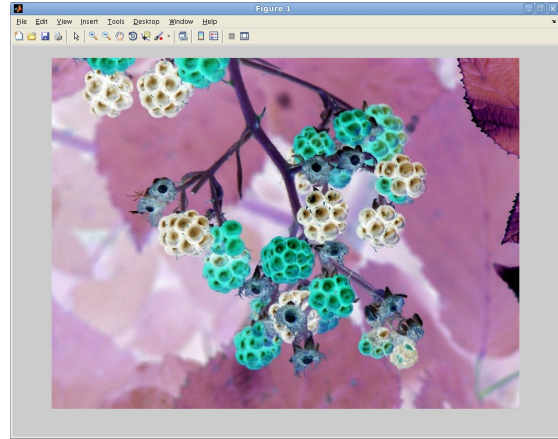
Írjunk olyan függvényt, amely egy paraméterként kapott 3-csatornás képmátrix inverzét állítja elő. Az inverz értékeket csatornánként kell kiszámolni.

```
function inverse_image = myinverse( image )
    dims = size(image);
    inverse_image = zeros(dims, 'uint8');
    inverse_image(:, :, :) = 255 - image(:, :, :);

end
```



Eredeti kép



Inverz kép

Feladat:

1. Készíts olyan függvényt, amely egy RGB kép csatornáit felcseréli BRG sorrendbe!
2. Készíts olyan függvényt, amely egy RGB képből szürkeárnyalatos képet készít! Az eredmény kép csak egy csatornával rendelkezik. A színcsatornák súlyozása a konverzió során: $0.33 * R$, $0.56 * G$ és $0.11 * B$