**Ipari képfeldolgozás házi feladat**

Kő-papír-olló játék felismerése webkamerában

**Készítették:**

Vecsei Gábor MDSZMA

Horváth Gergely LQPQX6

Mihályi Balázs LU8D88

Münczberg Tamás HLWM97

**Bevezetés**

Az általunk választott feladat, a híres kő-papír-olló nevezetű játék felismerése webkamera segítségével. A képfeldolgozási feladatokat MATLAB R2012a környezetben valósítottuk meg, az image acquistion és image processing toolboxok felhasználásával.

A program indítása után van lehetőségünk képet készíteni, vagy egy már meglévő képet felhasználni, amelyről szegmentálás és morfológiai transzformációk segítségével eldönthető, hogy melyik kézjel látható az adott képen. Az egyszerűség kedvéért egy apró módosítást eszközöltünk a játékszabályokban: a papír mutatása esetén az ujjakat nem zárjuk össze.

**A forráskód**

A MATLAB előkészítése a program számára. Letöröljük a command képernyőt, bezárjuk az esetlegesen megnyitott ablakokat, töröljük a változókat és beállítjuk a workspace-t.

*clc;*

*close all;*

*clear;*

*workspace;*

Lekérdezzük az elérhető képrögzítők adatait, majd létrehozunk egy kamera objektumot és megjelenítjük az előnézeti ablakot egy kérdéssel, hogy milyen módon szeretnénk képet bevinni a rendszerbe. Amennyiben fájlból töltjük be a képet, a megengedett formátumok: .jpg, .png és .tif.

*%Információk az elérhet? képrögzít?kr?l*

*vidInfo = imaqhwinfo;*

*%Kamera objektum létrehozása*

*vid = videoinput('winvideo', 1);*

*%El?nézeti ablak megnyílik*

*preview(vid);*

*%Kérdés - Válasz annak érdekében, hogy csinálunk egy képet, vagy pedig*

*%betöltünk egyet a gépr?l*

*message = sprintf('Would you like to take a picture?');*

*reply = questdlg(message, 'Capture Image', 'Yes', 'No', 'Yes');*

*%Választás leellen?rzése*

*if strcmpi(reply, 'Yes')*

*%Ha Igen akkor lövünk egy képet*

*imgFromCam = getsnapshot(vid);*

*I = imgFromCam;*

*else*

*%Töröljük a kamera objektumot, hogy ne foglalja a memóriát*

*delete(vid);*

*%Kép beolvasása*

*[filename, pathname] = ...*

*uigetfile({'\*.jpg';'\*.png';'\*.tif';'\*.\*'},'Select an Image file');*

*fullFileName = fullfile(pathname, filename);*

*I = imread(fullFileName);*

*end*

*%mindenképp töröljük az objektumot (így bezáródik a preview ablak is)*

*delete(vid);*

Az elkészült képet megjelenítjük egy plot-on, majd meghatározzuk a kép méretét és bejelöljük rajta a bőrfelületet. (természetesen nem kell mindig meghatározni a kép méretét, ez csak a jobb szemléltetés miatt van benne)

*figure(1)*

*subplot(2,2,1)*

*imshow(I);*

*%A méretek felvétele*

*height = size(I,1);*

*width = size(I,2);*

*%Az O-ra rajzolunk rá pirossal (azaz bejelöljük a b?rfelületet)*

*O = I;*

*BW = zeros(height,width);*

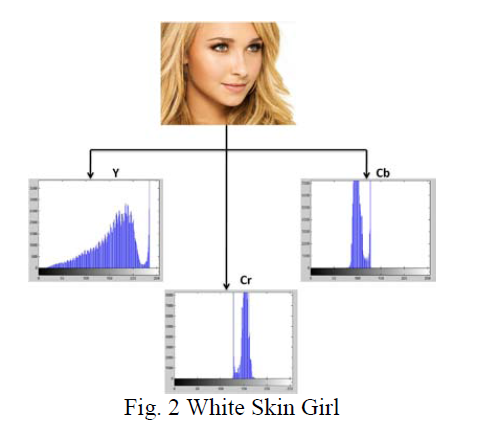
Az képet RGB-ből YCbCr-be konvertáljuk. Az YCbCr színtérben az Y a fényerőt, míg a Cb és Cr komponensek az tárolják, hogy mekkora az Y-hoz képest a kék és vörös színek fényerejének eltérése. Azért ebben a színtérben dolgozunk, mert a tapasztalatok azt mutatják így különíthetőek el egymástól hatékonyan a bőr- és a nem bőrpixelek. A bőr pixeleket megjelöljük a képen, majd egy bináris képen beállítjuk a bőrnek megfelelő pixeleket.

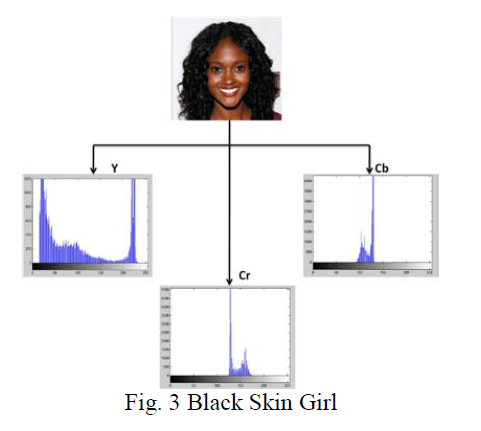
Chai és Ngan készítettek egy algoritmust ami különválasztja a bőr pixeleket:



Azaz ezek az értékek határozzák meg leginkább a bőr pixeleket.

Pár példa:





*%A képet RGB-b?l YCbCr-ba konvertáljuk*

*%https://en.wikipedia.org/wiki/YCbCr*

*I\_ycbcr = rgb2ycbcr(I);*

*Cb = I\_ycbcr(:,:,2);*

*Cr = I\_ycbcr(:,:,3);*

*figure(1)*

*subplot(2,2,2)*

*imshow(I\_ycbcr);*

*%B?r detektálása*

*[r,c,v] = find(Cb>=77 & Cb<=127 & Cr>=133 & Cr<=173);*

*numind = size(r,1);*

*%Megjelöli a "b?r pixeleket"*

*for i=1:numind*

*%Az eredeti képen beszínezzük a b?rfelületeket (szemléltetés)*

*O(r(i),c(i),:) = [255 0 0];*

*%Az eddig csak 0-ból álló tömbbe a b?r helyét feltöltjük 1-esekkel*

*BW(r(i),c(i)) = 1;*

*end*

*%A kis fekete helyeket a nagyban kitöltjük*

*BW = imfill(BW, 'holes');*

*%Eltünteti azt ami 300 pixelnél kissebb (zaj sz?rés)*

*%Egy már 1024 x 1024-es képen is kicsi az esélye, hogy 30x30 -nál kissebb*

*%legyen a kéz területe*

*BW = bwareaopen(BW,900);*

Ez a kódrészlet tartalmazza a morfológiai transzformációkat. Első lépésben eltüntetjük az ujjakat a képről, mivel azok mindig kisebbek, mint a tenyér az erózió során azok „elszakadnak” tőle. Ezek után visszaállítjuk a tenyér méretét, majd kivonjuk azt az eredeti képből, így csak az ujjakat tartalmazó képet kapunk eredményül. A műveletek után zajszűrésképp eldobjuk a 30x30 pixelnél kisebb területeket.

*%Structuring element létrehozása*

*se = strel('disk',60);*

*%El?ször eltüntetjük az ujjakat a képr?l, mivel azok mindig kissebbek mint*

*%a tenyér*

*BW2 = imerode(BW, se);*

*%Visszaállítjuk a tenyér méretét*

*BW2 = imdilate(BW2,se);*

*%Az így kapott "csak a tenyér" képet kivonjuk az eredetib?l, így megkapjuk*

*%csak az ujjakat tartalmazó képet*

*BW3 = imsubtract(BW, BW2);*

*%30x30 pixelnél kissebb területeket eldobjuk*

*BW3 = bwareaopen(BW3,900);*

*%csak azt tartjuk meg ami tényleg nagy a képen*

*BW3 = bwareaopen(BW3,9000);*

*BW = BW3;*

Ebben a részben határozzuk meg a kapott területek méretét, majd azokat méret szerinti csökkenő sorrendbe rendezzük.

*figure(1)*

*subplot(2,2,3)*

*imshow(BW);*

*st = regionprops(BW, 'All');*

*%Kiírjuk a workspace-n a területeket számszerüsítve*

*fprintf(1,'Terulet # Area\n');*

*for k = 1 : length(st)*

*%"Bedobozolás"*

*thisBB = st(k).BoundingBox;*

*hold on*

*rectangle('Position', [thisBB(1),thisBB(2),thisBB(3),thisBB(4)],...*

*'EdgeColor','r','LineWidth',1 );*

*hold off*

*fprintf(1,'#%2d %16.1f\n',k,st(k).Area);*

*end*

*%Rendezzük az elemeket csökken?be a területük szerint*

*allAreas = [st.Area];*

*[sortedAreas, sortingIndexes] = sort(allAreas, 'descend');*

*%Megszámozza a legnagyobbtól a legkissebbig*

*for k = 1 : length(st)*

*centerX = st(sortingIndexes(k)).Centroid(1);*

*centerY = st(sortingIndexes(k)).Centroid(2);*

*text(centerX,centerY,num2str(k),'Color', 'b', 'FontSize', 14)*

*end*

Ez a kódrészlet dönti el, hogy melyik kézjel ismerhető fel a képen.

*%Itt nézzük meg, hogy mit is kaptunk:*

*if sortingIndexes > 0*

*if length(sortingIndexes) >= 3*

*%Ha kett?nél több ujj van akkor tuti hogy nem olló*

*title('PAPER');*

*elseif length(sortingIndexes) < 3 && length(sortingIndexes) >= 2*

*% Ha 2 akkor pedid tuti, hogy ollót mutatunk*

*title('SCRISSORS');*

*end*

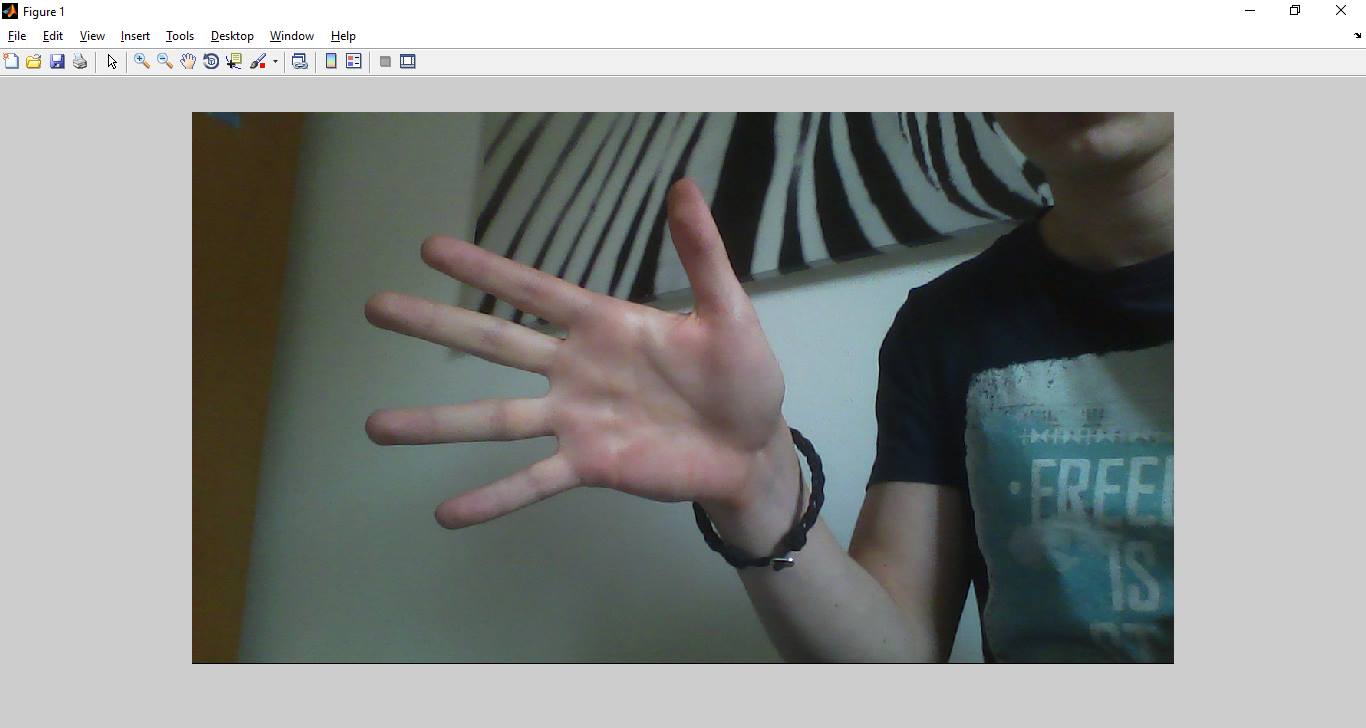
*else*

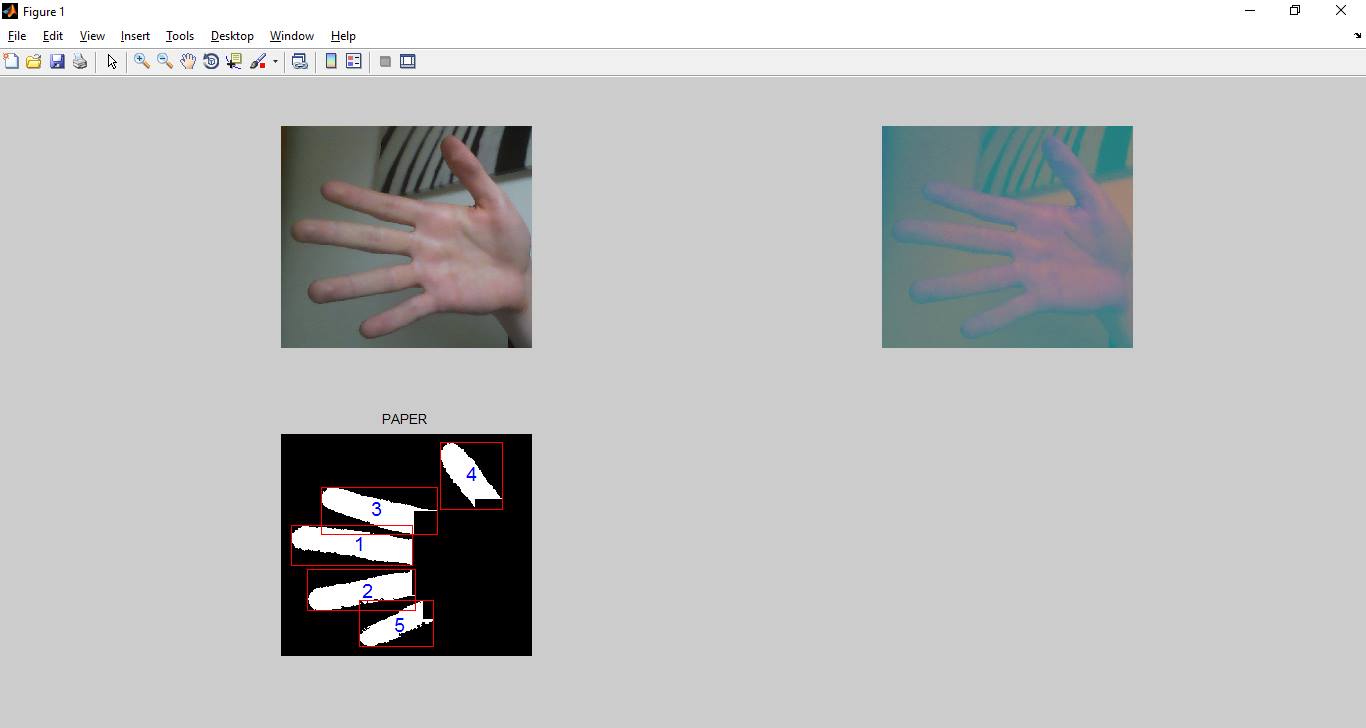
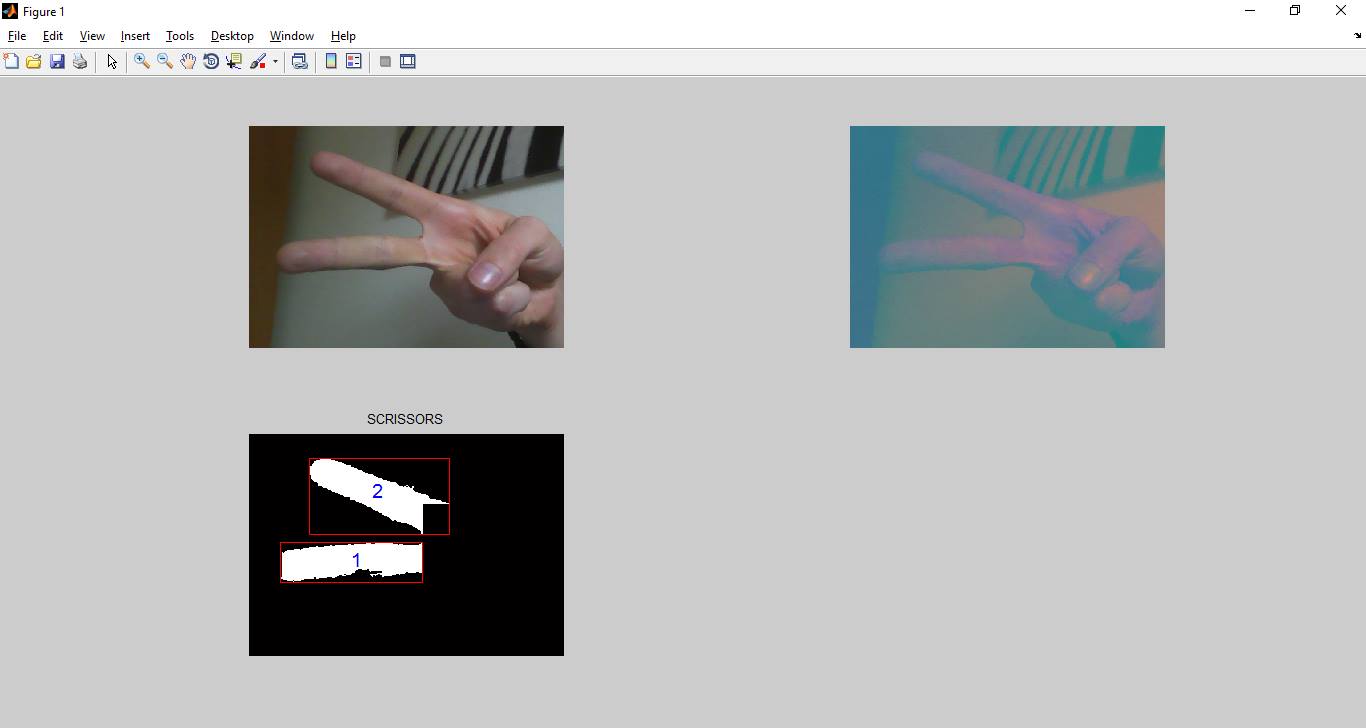
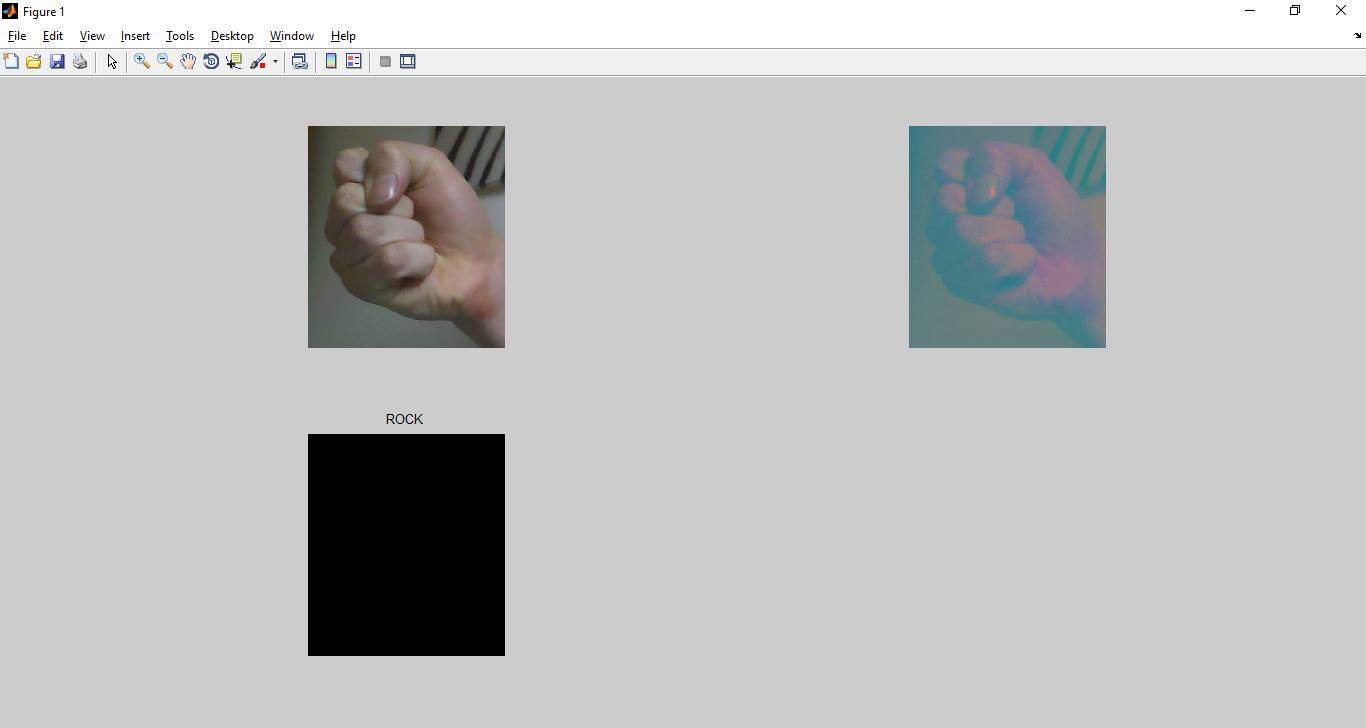
*%Minden egyéb esetben K? lesz*

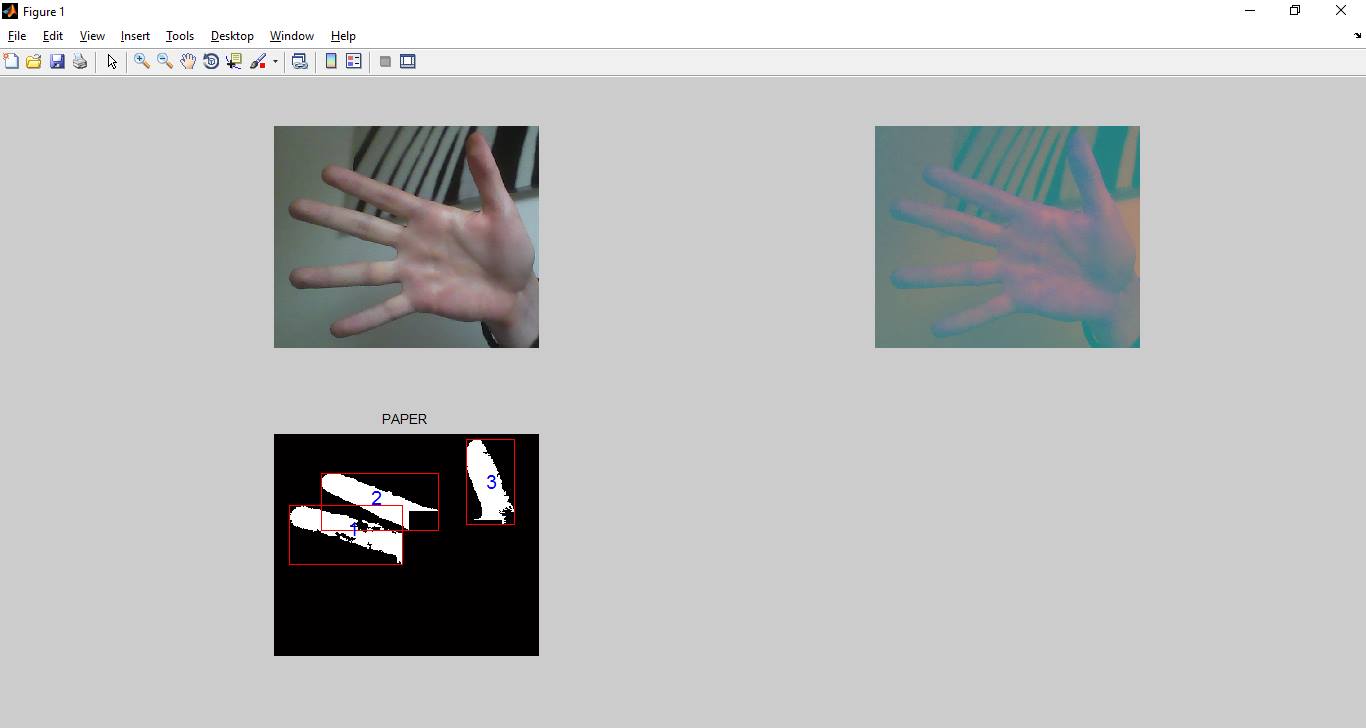
*title('ROCK')*

*end*

**Tesztelés**  
A kéznek nagyjából ilyen távolságban kell lennie a kamerától, hogy a felismerés hiba nélkül megtörténjen. A többi képen jól látszik, hogy a program felismerte a mutatott jelzéseket.







Ezek a képek azt ábrázolják, hogy hol detektáltunk bőrfelületet a képeken.

