PROJE KODLARI :

**1.KOD : TÜM RESİMLER OKUNUR İLK ÖNCE SCALING SONRA SEGMENTATION UYGULANIP KAYDEDİLMESİ**

clear all;close all;clc;

Folder = 'E:\TÜRK İŞARET DİLİ\A-B-C-D-E-F-G-H\TUĞŞAT\Z2' ;% % Resimler hangi dosyadan alınacak

filePattern = fullfile(Folder, '\*.jpg'); % İşleme alınan dosya tipi farketmeksizin son halini jpg olarak kaydet

theFiles = dir(filePattern);% dosyaları ve klasörlerı lıstelenir / Ynaı bır dosyada kac tane resım fıle var ıse o kadar ıslem yapacak

t=1;

for k = 1 : length(theFiles) % Dosyada bulunan resim sayısı kadar döngü döndürülür

baseFileName = theFiles(k).name; % işlenecek dosyanın ismi alınır

imOriginal = imread(baseFileName);% işlenecek dosyanın okunur

s= 8; % 1 resime 8 farklı scaling faktörü uygulanır

magnificationFactor = 0.1; % Piksel sayısı azaltılır veya coğaltılır

for c = 1:s

file\_name = sprintf('Image%d.png', t); % name Image with a sequence of number, ex Image1.png , Image2.png....

magnificationFactor = magnificationFactor + 0.4 ; % Uygulanan scalıng : 0.1,0.5,0.9,1.3,1.7,2.1,2.5,2.9

I = imresize(imOriginal, magnificationFactor,"bicubic"); % scaling part

% bicubic veya bilinear en iyi sonuç veren metotlardan biriydi

% SEGMENTATION PART :

final\_image = zeros(size(I,1), size(I,2)); % 200\*396 lık matrik

for i = 1:size(I,1) %200

for j = 1:size(I,2) %396

R = I(i,j,1); %115

G = I(i,j,2); % 116

B = I(i,j,3); %110

if(R > 105 && G > 50 && B > 30)

v = [R,G,B];

if((max(v) - min(v)) > 15) % 116-110 =6

if(abs(R-G) > 15 && R > G && R > B)

%it is a skin

final\_image(i,j) = 1; % BEYAZ PİKSEL

end

end

end

end

end

fullFileName = fullfile(Folder, file\_name); % dosyaya yenı dosya açıyor ve ona jpg yazdırıyor

imwrite(final\_image,file\_name,'jpg');

pause(1); % pause for one second

t=t+1;

end

end

**2.KOD : TÜM RESİMLER OKUNUR , ROTATION VE ARDINDAN SEGMENTATION UYGULANIP KAYDEDİLİR.**

clear all;close all; clc;

Folder = 'E:\TÜRK İŞARET DİLİ\NEW DATABASE\ANA RESİMLER\J\ROTATION' ;% Resimler hangi dosyadan alınacak

filePattern = fullfile(Folder, '\*.jpg'); % İşleme alınan dosya tipi farketmeksizin son halini jpg olarak kaydet

theFiles = dir(filePattern); % dosyaları ve klasörlerı lıstelenir / Ynaı bır dosyada kac tane resım fıle var ıse o kadar ıslem yapacak

t=1;

for k = 1 : length(theFiles) % Dosyada bulunan resim sayısı kadar döngü döndürülür

baseFileName = theFiles(k).name; % işlenecek dosyanın ismi alınır

imOriginal = imread(baseFileName); % işlenecek dosya okunur

s= 6; % Her bir resime 5 farklı rotation uygulaması uygulanır ( for loop is applied 5 times)

angle=120; % 120-150-180-210-240-270 derece döndürme işlemleri uygulanır

for c = 1:s

file\_name = sprintf('RotatedImage%d.png', t);

% name Image with a sequence of number, ex Image1.png , Image2.png..

angle = angle +30;

demo2 = imrotate(imOriginal,angle); % rotation part

I2=demo2;

I = imresize(I2,[224 224]); % scaling part

final\_image = zeros(size(I,1),size(I,2)); % segmentatıon part

for i = 1:size(I,1) %200

for j = 1:size(I,2) %396

R = I(i,j,1); %115

G = I(i,j,2); % 116

B = I(i,j,3); %110

if(R > 105 && G > 50 && B > 30)

v = [R,G,B];

if((max(v) - min(v)) > 15) % 116-110 =6

if(abs(R-G) > 15 && R > G && R > B)

% it is a skin

final\_image(i,j) = 1; % BEYAZ PİKSEL

end

end

end

end

end

fullFileName = fullfile(Folder, file\_name); % Rotasyon ve segmentasyon uygulanmıs resımler adlandırılarak hedef dosyaya kaydedılır

imwrite(final\_image,file\_name,'jpg'); % JPG dosyası olarak yazdırılır

pause(1); % Algorıtma 1 sn aralıklarla calısır

t=t+1;

end

end

**2.KODUN FARKLISI :**

clear all;close all; clc;

Folder = 'E:\TÜRK İŞARET DİLİ\NEW DATABASE\ANA RESİMLER\J\ROTATION' ;% Resimler hangi dosyadan alınacak

filePattern = fullfile(Folder, '\*.png'); % İşleme alınan dosya tipi farketmeksizin son halini jpg olarak kaydet

theFiles = dir(filePattern); % dosyaları ve klasörlerı lıstelenir / Ynaı bır dosyada kac tane resım fıle var ıse o kadar ıslem yapacak

t=1;

for k = 1 : length(theFiles) % Dosyada bulunan resim sayısı kadar döngü döndürülür

baseFileName = theFiles(k).name; % işlenecek dosyanın ismi alınır

imOriginal = imread(baseFileName); % işlenecek dosya okunur

s= 6; % Her bir resime 5 farklı rotation uygulaması uygulanır ( for loop is applied 5 times)

angle=120; % 120-150-180-210-240-270 derece döndürme işlemleri uygulanır

for c = 1:s

file\_name = sprintf('RotatedImage%d.png', t);

% name Image with a sequence of number, ex Image1.png , Image2.png..

angle = angle +30;

demo2 = imrotate(imOriginal,angle); % rotation part

I2=demo2;

I = imresize(I2,[224 224]); % scaling part

final\_image = zeros(size(I,1),size(I,2)); % segmentatıon part

for i = 1:size(I,1) %200

for j = 1:size(I,2) %396

R = I(i,j,1); %115

G = I(i,j,2); % 116

B = I(i,j,3); %110

if(R > 105 && G > 50 && B > 30)

v = [R,G,B];

if((max(v) - min(v)) > 15) % 116-110 =6

if(abs(R-G) > 15 && R > G && R > B)

% it is a skin

final\_image(i,j) = 1; % BEYAZ PİKSEL

end

end

end

end

end

fullFileName = fullfile(Folder, file\_name); % Rotasyon ve segmentasyon uygulanmıs resımler adlandırılarak hedef dosyaya kaydedılır

imwrite(final\_image,file\_name,'jpg'); % JPG dosyası olarak yazdırılır

pause(1); % Algorıtma 1 sn aralıklarla calısır

t=t+1;

end

end

**2.KODUN FARKLISI :**

clear all;close all; clc;

Folder = 'E:\TÜRK İŞARET DİLİ\Processed Images\E' ;% Specify the folder where the files live.

% Get a list of all files in the folder with the desired file name pattern.

filePattern = fullfile(Folder, '\*.jpg'); % Change to whatever pattern you need.

theFiles = dir(filePattern);

for k = 1 : length(theFiles) % HER BIR DOSYAYI Okur sırayla

baseFileName = theFiles(k).name; % sıra sıra dosyaları alır

fullFileName = fullfile(theFiles(k).folder, baseFileName); % ve şekillendirip kaydeder

imOriginal = imread(baseFileName);

s= 10;

angle=30;

for c = 1:s

angle = angle +30;

demo2 = imrotate(imOriginal,angle);

I=demo2; % rotatıon

final\_image = zeros(size(I,1), size(I,2)); % 200\*396 lık matrik

for i = 1:size(I,1) %200

for j = 1:size(I,2) %396

R = I(i,j,1); %115

G = I(i,j,2); % 116

B = I(i,j,3); %110

if(R > 105 && G > 50 && B > 30)

v = [R,G,B];

if((max(v) - min(v)) > 15) % 116-110 =6

if(abs(R-G) > 15 && R > G && R > B)

it is a skin

final\_image(i,j) = 1; % BEYAZ PİKSEL

end

end

end

end

end

figure, imshow(final\_image);

final\_image=fullFileName;

fullFileName = fullfile(theFiles(k).folder, c);

c=c+1;

end

end

%

% fprintf(1, 'Now reading %s\n', fullFileName);

% % Now do whatever you want with this file name,

% % such as reading it in as an image array with imread()

% imageArray = imread(fullFileName);

% imshow(imageArray); % Display image.

% drawnow; % Force display to update immediately.

**2.KODUN FARKLISI**

clear all;close all; clc;

Folder = 'E:\TÜRK İŞARET DİLİ\Processed Images\D' ;

imOriginal = imread("D\_0\_74.jpg");

s= 10;

angle=30;

for c = 1:s

angle = angle +30;

demo2 = imrotate(imOriginal,angle);

I=demo2; % rotatıon

final\_image = zeros(size(I,1), size(I,2)); % 200\*396 lık matrik

for i = 1:size(I,1) %200

for j = 1:size(I,2) %396

R = I(i,j,1); %115

G = I(i,j,2); % 116

B = I(i,j,3); %110

if(R > 105 && G > 50 && B > 30)

v = [R,G,B];

if((max(v) - min(v)) > 15) % 116-110 =6

if(abs(R-G) > 15 && R > G && R > B)

%it is a skin

final\_image(i,j) = 1; % BEYAZ PİKSEL

end

end

end

end

end

figure, imshow(final\_image);

saveas

end

% saveas - prınt-ımwrıte-ımsave

**TÜM FOTOĞRAFLARIN TARANIĞ SEGMENTASYONU ALINIP**

clear all;close all;clc;

Folder = 'E:\TÜRK İŞARET DİLİ\NEW DATABASE\ANA RESİMLER\A1\ANA RESİMLER' ;% % Resimler hangi dosyadan alınacak

filePattern = fullfile(Folder, '\*.png'); % İşleme alınan dosya tipi farketmeksizin son halini jpg olarak kaydet

theFiles = dir(filePattern);% dosyaları ve klasörlerı lıstelenir / Ynaı bır dosyada kac tane resım fıle var ıse o kadar ıslem yapacak

t=1;

for k = 1 : length(theFiles) % Dosyada bulunan resim sayısı kadar döngü döndürülür

baseFileName = theFiles(k).name; % işlenecek dosyanın ismi alınır

I = imread(baseFileName);% işlenecek dosyanın okunur

file\_name = sprintf('ScaledImage%d.png', t);

% SEGMENTATION PART :

final\_image = zeros(size(I,1), size(I,2)); % 200\*396 lık matrik

for i = 1:size(I,1) %200

for j = 1:size(I,2) %396

R = I(i,j,1); %115

G = I(i,j,2); % 116

B = I(i,j,3); %110

if(R > 105 && G > 50 && B > 30)

v = [R,G,B];

if((max(v) - min(v)) > 15) % 116-110 =6

if(abs(R-G) > 15 && R > G && R > B)

%it is a skin

final\_image(i,j) = 1; % BEYAZ PİKSEL

end

end

end

end

end

fullFileName = fullfile(Folder, file\_name); % dosyaya yenı dosya açıyor ve ona jpg yazdırıyor

imwrite(final\_image,file\_name,'jpg');

pause(1); % pause for one seconD

t=t+1;

end

**IMRESIZE KODU**

I = imread('Z\_0\_9956.jpg'); %Read built-in image

figure, imshow(I); %Display image

K = imresize(I,[240 240]); %Create an output image with 100 rows and 150 columns

figure, imshow(K); %Display 100 x 150 image

**IMRESIZE KODU 2:**

clear all;close all;clc;

Folder = 'E:\TÜRK İŞARET DİLİ\NEW DATABASE\ANA RESİMLER\A1' ;% % Resimler hangi dosyadan alınacak

filePattern = fullfile(Folder, '\*.jpg'); % İşleme alınan dosya tipi farketmeksizin son halini jpg olarak kaydet

theFiles = dir(filePattern);% dosyaları ve klasörlerı lıstelenir / Ynaı bır dosyada kac tane resım fıle var ıse o kadar ıslem yapacak

t=1;

for k = 1 : length(theFiles) % Dosyada bulunan resim sayısı kadar döngü döndürülür

baseFileName = theFiles(k).name; % işlenecek dosyanın ismi alınır

imOriginal = imread(baseFileName);% işlenecek dosyanın okunur

file\_name = sprintf('Image%d.png', t); % name Image with a sequence of number, ex Image1.png , Image2.png....

I = imresize(imOriginal,[224 224]); % scaling part

% bicubic veya bilinear en iyi sonuç veren metotlardan biriydi

fullFileName = fullfile(Folder, file\_name); % dosyaya yenı dosya açıyor ve ona jpg yazdırıyor

imwrite(I,file\_name,'jpg');

pause(1); % pause for one second

t=t+1;

end

**SCALİNG VE SEGMENTATION UYGULAMASI :**

% 0.5 / 0.9 / 1.3 / 1.7 / 2.1 /2.5 / 2.9

clear all;close all;clc;

Folder = 'E:\TÜRK İŞARET DİLİ\Scaling and Segmentation\R' ;% Specify the folder where the files live.

% Get a list of all files in the folder with the desired file name pattern.

filePattern = fullfile(Folder, '\*.jpg'); % Change to whatever pattern you need.

theFiles = dir(filePattern);

t=1;

for k = 1 : length(theFiles) % HER BIR DOSYAYI Okur sırayla

baseFileName = theFiles(k).name; % sıra sıra dosyaları alır

imOriginal = imread(baseFileName);

s= 8;

magnificationFactor = 0.1;

for c = 1:s

file\_name = sprintf('Image%d.png', t);

magnificationFactor = magnificationFactor + 0.4 ;

I = imresize(imOriginal, magnificationFactor,"bicubic");

%neares,cubic gibi interpolation baktık en iyisi buydu

% figure,imshow(J);title("Original Image");

% figure,imshow(I);title("Original Image");

final\_image = zeros(size(I,1), size(I,2)); % 200\*396 lık matrik

for i = 1:size(I,1) %200

for j = 1:size(I,2) %396

R = I(i,j,1); %115

G = I(i,j,2); % 116

B = I(i,j,3); %110

if(R > 105 && G > 50 && B > 30)

v = [R,G,B];

if((max(v) - min(v)) > 15) % 116-110 =6

if(abs(R-G) > 15 && R > G && R > B)

%it is a skin

final\_image(i,j) = 1; % BEYAZ PİKSEL

end

end

end

end

end

fullFileName = fullfile(Folder, file\_name); % dosyaya yenı dosya açıyor ve ona jpg yazdırıyor

imwrite(final\_image,file\_name,'jpg');

pause(1); % pause for one second

t=t+1;

end

end

clear all;close all;clc;

% Gray Level Scale Image

i=imread('A\_0\_1026.jpg');

imshow(i) ;

title('Input Image');

I1=rgb2gray(i);

figure,imshow(I1) ;

title('Gray-Scale Image');

DİLUTION + EROSION KODLARI :

clear all;close all;clc;

im = imread('A\_0\_1026.jpg');

imshow(im);

im = double(im);

% MORPHOLOGICAL MODIFICATIONS : ( AFTER SEGMENTATION, SHAPE İS REBUİLT

erosion=imerode(im);

figure;imshow(erosion);title("Erosion");

dilution=imdilate(im);

figure;imshow(dilution);title("Dilution");

opening=imdilate(erosion);

figure;imshow(erosion);

closing=imerode(dilution);

figure;imshow(erosion);

IMRESIZE :

Folder = 'E:\TÜRK İŞARET DİLİ\TESTİNG\tr\_signLanguage\_dataset\ABC\C' ;% % Resimler hangi dosyadan alınacak

filePattern = fullfile(Folder, '\*.jpg'); % İşleme alınan dosya tipi farketmeksizin son halini jpg olarak kaydet

theFiles = dir(filePattern);% dosyaları ve klasörlerı lıstelenir / Ynaı bır dosyada kac tane resım fıle var ıse o kadar ıslem yapacak

t=1;

for k = 1 : length(theFiles) % Dosyada bulunan resim sayısı kadar döngü döndürülür

baseFileName = theFiles(k).name; % işlenecek dosyanın ismi alınır

imOriginal = imread(baseFileName);% işlenecek dosyanın okunur

file\_name = sprintf('Image%d.png', t); % name Image with a sequence of number, ex Image1.png , Image2.png....

final\_image = imresize(imOriginal,[227 227]); % scaling part

% bicubic veya bilinear en iyi sonuç veren metotlardan biriydi

fullFileName = fullfile(Folder, file\_name); % dosyaya yenı dosya açıyor ve ona jpg yazdırıyor

imwrite(final\_image,file\_name,'jpg');

pause(1); % pause for one second

t=t+1;

end

**2.**

% boyut değiişimi 224X224 pixels

clear all;close all;clc;

Folder = 'E:\TÜRK İŞARET DİLİ\NEW DATABASE\ANA RESİMLER\A1' ;% % Resimler hangi dosyadan alınacak

filePattern = fullfile(Folder, '\*.jpg'); % İşleme alınan dosya tipi farketmeksizin son halini jpg olarak kaydet

theFiles = dir(filePattern);% dosyaları ve klasörlerı lıstelenir / Ynaı bır dosyada kac tane resım fıle var ıse o kadar ıslem yapacak

t=1;

for k = 1 : length(theFiles) % Dosyada bulunan resim sayısı kadar döngü döndürülür

baseFileName = theFiles(k).name; % işlenecek dosyanın ismi alınır

imOriginal = imread(baseFileName);% işlenecek dosyanın okunur

file\_name = sprintf('Image%d.png', t); % name Image with a sequence of number, ex Image1.png , Image2.png....

I = imresize(imOriginal,[224 224]); % scaling part

% bicubic veya bilinear en iyi sonuç veren metotlardan biriydi

fullFileName = fullfile(Folder, file\_name); % dosyaya yenı dosya açıyor ve ona jpg yazdırıyor

imwrite(I,file\_name,'jpg');

pause(1); % pause for one second

t=t+1;

end

3 KANALLI FORMAT :

clear all;close all;clc;

Folder = 'E:\TÜRK İŞARET DİLİ\NEW DATABASE\3CHANNELEKLE\deneme' ;% % Resimler hangi dosyadan alınacak

filePattern = fullfile(Folder, '\*.png'); % İşleme alınan dosya tipi farketmeksizin son halini jpg olarak kaydet

theFiles = dir(filePattern);% dosyaları ve klasörlerı lıstelenir / Ynaı bır dosyada kac tane resım fıle var ıse o kadar ıslem yapacak

t=1;

for k = 1 : length(theFiles) % Dosyada bulunan resim sayısı kadar döngü döndürülür

baseFileName = theFiles(k).name; % işlenecek dosyanın ismi alınır

imOriginal = imread(baseFileName);% işlenecek dosyanın okunur

file\_name = sprintf('ChannelledImage%d.png', t); % name Image with a sequence of number, ex Image1.png , Image2.png....

I = imresize(imOriginal, [227 227]);

% MAIN PART :

final\_image = cat(3,I,I,I);

fullFileName = fullfile(Folder, file\_name); % dosyaya yenı dosya açıyor ve ona jpg yazdırıyor

imwrite(final\_image,file\_name,'jpg');

pause(1); % pause for one second

t=t+1;

end

**2.**

% making all gray scales as 3 channels RGB images.

grayImage=imread("ScaledImage1.png");

rgbImage = cat(3, grayImage, grayImage, grayImage);

figure,

imshow(rgbImage);

DILUTION + EROSION + IMFILL :

**1.**

clc;close all;clear all;workspace;

BW= imread("ScaledImage15.png");

SE= strel("square",3);

dil=imdilate(BW,SE);

er=imerode(dil,SE);

BW2 = imfill(er,'holes');

figure;

imshow(BW2);title('Filled Image1');

for i=71:151

BW2(i,1)=255;

i=i+1;

end

new = imfill(BW2,'holes');

% for i2=209:224

% new(1,i2)=255;

% i2=i2 +1;

% end

% new2 = imfill(new,'holes');

for i3=102:127

new(i3,224)=255;

i3=i3 +1;

end

new3 = imfill(new,'holes');

figure;

imshow(new3);title("SON");

**2.**

%erosıon-dilution

clear all;close all;clc

im = imread('ScaledImage15.png');

SE = strel('square',3); %Structuring element

di1=imdilate(im,SE);

er1=imerode(di1,SE);

di2=imdilate(er1,SE);

er2=imerode(di2,SE);

di3=imdilate(er2,SE);

er3=imerode(di3,SE);

figure;

imshow(im); title(["Main Image"]);

figure;

imshow(er3); title(["Last Image"]);

[rows, cols]=size(im); %finding the row and column numbers for forloops

X=false(rows,cols); %defining a false binary X matrix for logical calculations

X(214,119) = true; %initialize a starting point

Xk = true(rows,cols); %defining a true binary output matrix

%implementing the for loops for region filling

while 1

Xk = (imdilate(X,SE) & (~im));

if sum(X(:)-Xk(:))==0

break;

else

X=Xk;

end

%showing the filling of the region

imshow(X); pause(1e-12);

end

%showing the input and output images

figure;

imshow(X|im); title('Output')

**3.**

%erosıon-dilution

clear all;close all;clc

im = imread('ScaledImage15.png');

SE = strel('square',3); %Structuring element

%Erosion

er=imerode(im,SE);

figure;

subplot(231); imshow(im); title('Input Image');

subplot(232); imshow(er); title('After Erosion');

%Dilation

di=imdilate(er,SE);

subplot(233); imshow(di); title('After Dilation');

%Second Dilation

di2=imdilate(di,SE);

subplot(234); imshow(di2); title('After Second Dilation');

%Second Erosion

er2=imerode(di2,SE);

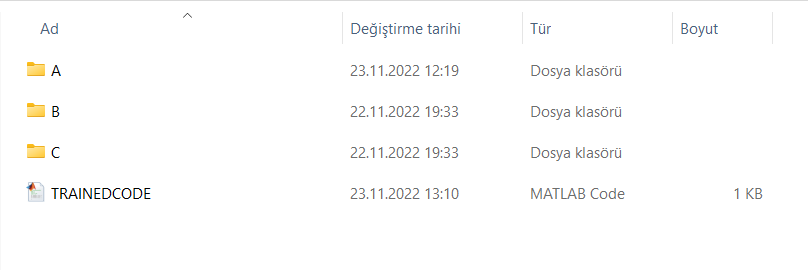
subplot(236); imshow(er2); title('After Second Erosion - Output');

**4.**

**MATLAB CODE :**

Tüm resimler ilk olarak imresize edilerek (227\*227) daha sonra cat fonksiyonu ile 3 boyutlu hale getirilmiştir.

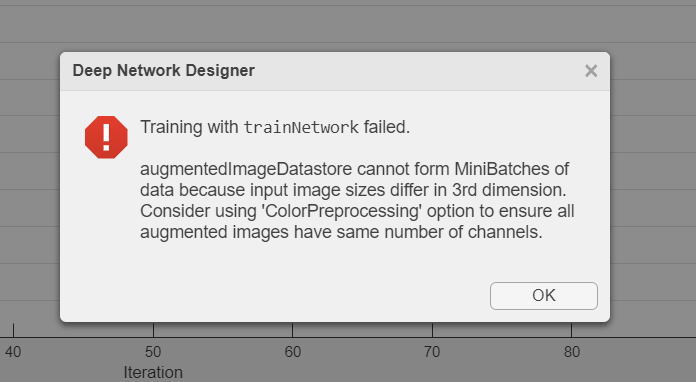
**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu** ****

****

: uınt 8 kodu idi : ana resimler ( segmentation uygulanmış ) \* maske= yenı bır resım oluşur : 3 boyutlu hale getirmiştik

MASKELER DÜZELTİLDİ + 3 KANALLI + UINT8

****

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**TEST ACCURACY HESAPLANMASI :**

testData=imageDatastore('E:\TÜRK İŞARET DİLİ\TESTİNG\tr\_signLanguage\_dataset\ABC','IncludeSubfolders' ,true, 'LabelSource' , 'foldernames');

[YPred,scores] = classify(trainedNetwork\_1,testData);

YValidation = testData.Labels;

accuracy = mean(YPred == YValidation);

clear all;close all;clc;

I= imread('E:\TÜRK İŞARET DİLİ\NEW DATABASE\ANA RESİMLER\B\ANA RESİMLER\Image7.png');

final\_image = zeros(size(I,1), size(I,2)); % 200\*396 lık matrik

for i = 1:size(I,1) %200

for j = 1:size(I,2) %396

R = I(i,j,1); %115

G = I(i,j,2); % 116

B = I(i,j,3); %110

if(R > 105 && G > 50 && B > 30)

v = [R,G,B];

if((max(v) - min(v)) > 15) % 116-110 =6

if(abs(R-G) > 15 && R > G && R > B)

%it is a skin

final\_image(i,j) = 1; % BEYAZ PİKSEL

end

end

end

end

end

image\_thresholded = final\_image;

image\_thresholded(final\_image>=0.5) = 255;

image\_thresholded(final\_image<0.5) = 0;

SE = strel('square',3); %Structuring element

RT= imclose(image\_thresholded,SE);

RC= imfill(RT,"holes");

RGB = cat(3,RC,RC,RC);

NEWIMAGE=uint8(RGB);

final2Image= bitand(NEWIMAGE,I);

figure()

subplot(1,4,1)

imshow(I,[])

subplot(1,4,2)

imshow(image\_thresholded,[])

subplot(1,4,3)

imshow(RGB,[])

subplot(1,4,4)

imshow(final2Image,[]);

**TEST VERİLERİNİN IMRESİZE LANMASI :**

Folder = 'E:\TÜRK İŞARET DİLİ\TESTİNG\tr\_signLanguage\_dataset\ABC\A' ;% % Resimler hangi dosyadan alınacak

filePattern = fullfile(Folder, '\*.jpg'); % İşleme alınan dosya tipi farketmeksizin son halini jpg olarak kaydet

theFiles = dir(filePattern);% dosyaları ve klasörlerı lıstelenir / Ynaı bır dosyada kac tane resım fıle var ıse o kadar ıslem yapacak

t=1;

for k = 1 : length(theFiles) % Dosyada bulunan resim sayısı kadar döngü döndürülür

baseFileName = theFiles(k).name; % işlenecek dosyanın ismi alınır

imOriginal = imread(baseFileName);% işlenecek dosyanın okunur

file\_name = sprintf('Image%d.png', t); % name Image with a sequence of number, ex Image1.png , Image2.png....

final\_image = imresize(imOriginal,[227 227]); % scaling part

% bicubic veya bilinear en iyi sonuç veren metotlardan biriydi

fullFileName = fullfile(Folder, file\_name); % dosyaya yenı dosya açıyor ve ona jpg yazdırıyor

imwrite(final\_image,file\_name,'jpg');

pause(1); % pause for one second

t=t+1;

end