Final Sınavı için bilinmesi gereken Matlab fonksiyonları:

imread ( )

imwrite ( )

imshow ( )

rgb2gray ( )

im2bw ( )

imhist ( )

histeq ( )

imrotate ( )

imresize ( )

imcrop ( )

medfilt2 ( )

edge ( )

imerode( )

imdilate( )

imopen( )

imclose( )

**Edge:**özel filtreler yardımıyla gri seviyedeki görüntülerin sınırları belirlenir.

**imread:**imread ile üzerinde çalışılmak istenilen fotoğraf bir değişkene atanarak matlab workspace için tanımlanır ve ardından imshow ile pencerede gösterilebilir. Workspace’te imgenin boyutu, çözünürlüğü gibi bilgileri görebilirsiniz.( MATLAB’ da herhangi bir görüntü üzerinde işlem yapabilmek için öncelikle gerekli olan görüntünün bulunduğu dizinden alınarak bir değişkene atılması daha sonra kullanılması sırasında kolaylık sunacaktır. Bu işlemi yapabilmek için *imread* fonksiyonu kullanılır. Örneğin; aşağıda verilen kod satırlarından ilki ile “ammar.png” ismindeki resim “resim” isimli değişkene atılmıştır. Bu sayede resim değişkenine atılan bu resim MATLAB içerisinde herhangi bir komut veya fonksiyon ile kullanılabilecektir. Bu kod örneğinin ikinci satırında ise görüntünün *imshow* komutu ile ekranda görüntülenmesi sağlanmıştır ve ayrıca *imshow* komutunun yanı sıra resimleri görüntülemek için kullanılan diğer bir komut ise *imtool*’dur. Bu komut ile görüntülenen resim üzerinde daha ayrıntılı bilgi sahibi olunabilir. Gölge tespitinin ilk kısımlarında *imtool* fonksiyonu bize görüntü üzerinde yorum yapmamızı sağladı. *Title* ise isminden anlaşılacağı gibi açılan sayfaya başlık vermemizi sağlamaktadır.

herhangi bir resim yüklemek için ' imread ' komutu kullanılabilir

* ·    resim = imread('ammar.png');
* ·    imshow(resim)
* ·    title(‘deneme resim');
* ·    imtool(resim);

)

**imwrite** İmage’ler diske imwrite fonksiyonu ile yazılır. Filename’den sonra MATLAB’ın desteklediği hangi formatta yazılacaksa muhakkak belirtilmelidir. Aşağıda örnekler verilmiştir. Eğer filename isimli dosya çalışılacak dizine yazılacaksa yol tarif edilmesine gerek yoktur. İmwrite komutuyla JPG formatında kayıt yapılacaksa, ‘quality’, q yazılır. Burada q 0 ile 100 arasında bir tamsayıdır. Sıkıştırma oranını gösterir. En düşük sayı kalitesiz kayıt anlamındadır ( Görüntüyü kaydetmek için *imwrite* fonksiyonu kullanılır.)

**imresize** Matlab ile boyut değiştirme için “imresize” adındaki işlev kullanılabilmektedir. Is=imresize(I,oran,yöntem); Örn; Is=imresize(I,0.97, ‘bicubic’);

**imhist**  Bazı görüntüler düşük kontrasta sahiptir. Düşük kontrasta sahip görüntülerin yoğunluk dağılımlarını görmek için, *imhist* fonksiyonu çağrılarak bir histogram oluşturulabilir. Histogram, bir görüntü içerisindeki renk değerlerinin tekrarlanma sıklığını gösteren bir grafiktir. Bu grafik sayesinde görüntünün aydınlık ya da karanlık olduğu hakkında bilgiler elde edilebilir. Karanlık bir görüntüde sıfıra yakın renk değerlerinin sayısı fazladır. Bu yüzden histogramda küçük değerlerde bir yığılma görülür. Aydınlık bir görüntüde ise histogramda yüksek değerlerde bir yığılma görülür. Düşük karşıtlıkta histogramın belirli bölgelerinde yığılma olan görüntülerde görülür. Aydınlık veya karanlık görüntüler buna örnek verilebilir. Görüntünün histogramını göstermek için, öncelikle **rgb2gray** komutu ile görüntü gri seviyeye dönüştürülür ve *imhist* komutu ile histogramı gösterilir.

·    Image = imread(‘tez.jpg’);

·    Imshow(Image);

·    Image = rgb2gray(Image), imhist(image)

**histeq** Görüntüdeki kontrastı geliştirmek için MATLAB araç kutusu birkaç farklı yol sunar. Düşük karşıtlık olan görüntülerde histogram eşitleme yapılarak daha anlamlı görüntüler elde edilir. Histogramda belli bir bölgede yığılmış olan değerleri yayarak yeni görüntü elde edilir. Histogram eşitleme işlemi için MATLAB’ da *histeq* fonksiyonu kullanılır. Komut satırına aşağıdaki kodlar yazılarak histogram eşitleme yapılır.

·    resim = imread('C:\Users\ASLAN\Desktop\ammar.jpg');

·    resim = rgb2gray(resim);

·    imshow(resim)

·    i = histeq(resim);

·    figure, imhist(i)

·    figure, imshow(i);

**imopen** Bu örnek görüntüde, görüntünün alt tarafındaki arka plan aydınlığı daha parlaktır. Arkaplan aydınlığını tahmin etmek için morfolojik açma işlemi kullanılır. Morfolojik açma işlemini göstermek için *imopen* fonksiyonu kullanılır.

·    Background = imopen(I,strel(‘disk’,15));

**rgb2gray** Renkli bir görüntüyü gri seviyeli bir resme dönüştürmek için *rgb2gray* komutu kullanılır.

·    resim = imread('C:\Users\ASLAN\Desktop\deneme.jpg');

·    resim = rgb2gray(resim);

·    imshow(resim)

* '**im2bw** ' fonksiyonu eşik değerine göre resmi beyaz(1) ile siyah(0) olarak yeniden çizer:

BWlevel = im2bw(RGB,level); imshow(BW) % resimde bulmamız gereken noktalar dışında beyazlıklar var.

BWem = im2bw(RGB,EM); imshow(BW) % resimde bulmamız gereken noktalar daha belirgindir.

* ' **medfilt2** ' resimde bulunan parazit görüntüleri yok etmek için kullanılır, matris değeri büyüdükçe hassaslık azalır:

BW = medfilt2(BWem, [3 3]);

* ' **bwlabel** ' fonksiyonu kullanarak kaç tane cisim olduğunu ve bu cizimlerin bir dizisini döndürür:

[etiketli\_resimler, sayisi] = bwlabel(BW);

**imshow**(etiketli\_resimler == 2) % etiket matrisinde değeri 2 olan elemanları göster. resmi göstermek için ' imshow ' kullanılabilir:

figure,imshow(I) komutu her çalıştırıldığında yeni bir figür penceresi açılır.

figure(1),imshow(I)

**imnoise**(); % görüntü bulanıklaştırma işlemi yapar.

**imrotate**(); % resmi döndürme işlemi

**fspecial**(); % resme uygulanacak filtre için katsayılar matrisi oluşturur

**imdilate:** açma işlemini yapar.  
**imerode:** aşındırma işlemi yapar. **imclose, imopen:** imdilate ve imerode işlemlerinin birlikte kullanılması ile oluşan filtrelerdir. Uygulama mantıkları benzerdir.

**imcrop** Kırpma işlemi için “imcrop” komutu da kullanılabilmektedir; >>e=imcrop(m, [126 125 376 375]);

**J= imabsdiff(I, K); (çıkarma)**

**J= imadd(I, K); (toplama)**

**J= imcomplement(I); (tersini alma)**

**J= imdivide(I, background); (bölme)**

**J= immultiply(I, 1.2); (çarpma)**

**J= imsubtract(I, K); (çıkarma)**

**J= imresize(I, 1.25); (boyut değiştirme)**

**J= imrotate(I, 35, ‘bilinear’); (döndürme)**

**I= imcrop; (kesme)**

Python dilinde yazılmış OpenCV uygulamaları ile ilgili sorumlu olunan konular:

1. **Resim okuma ve ekrana çıkarma**

import cv2

import cv2 numpy as np

image=cv2.imread( ‘C:\\Users\\USER\\Desktop\\images\\elephant.jpg’) #döndürmek istediğim görüntünün bilgisayarımda ki konumunu belirtiyorum

height, width = image.shape[:2]

start\_row, start\_col=int(height \* .05), int(width \* .30) #kırpmak istediğiniz boyuta göre değerler verebilirsiniz

end\_row, end\_col=int(height \* .75), int(width \* .75)

cropped=image[start\_row:end\_row , start\_col:end\_col]

cv2.imshow(“Original Image”, image) #açılan pencerede ilk orjinal görüntümüz açılır

cv2.waitKey(0)

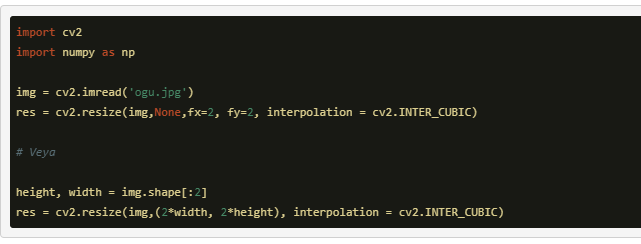
cv2.imshow(“Cropped Image”, cropped) #çarpıya tıkladığımız da ise kırpılmış görüntü ile karşılaşırız

cv2.waitKey(0)

cv2.destroyAllWindows()

1. **Piksel manipülasyonu ve resim kırpma**
2. **Resimi kanallara ayırma ve kanalları birleştirme**
3. **Resim ölçekleme**

Ölçekleme(Scaling)

Ölçekleme(scaling) resmin yeniden boyutlandırılması işlemidir. OpenCV, bu amaçla cv2.resize() işleviyle birlikte gelir. Bunun dışında görüntünün boyutu manuel olarak belirtilebilir veya ölçeklendirme(scaling) faktörü belirtilebilir.Bu işlemlerde; küçültme için cv2.INTER\_AREA ve yakınlaştırma için cv2.INTER\_CUBIC ve cv2.INTER\_LINEAR fonksiyonları kullanılabilir. Varsayılan olarak, kullanılan yeniden boyutlandırma amaçları için cv2.INTER\_LINEAR yöntemi kullanılır. Aşağıdaki kodlardaki değerler ile oynayıp sonuçları inceleyebilirsiniz. 

1. Resim döndürme ve öteleme
2. Resim filtreleme