



---

# GÖRÜNTÜ İŞLEME

---

İçerik Tabanlı Görüntü Erişimi (Content Based Image Retrieval) Uygulaması



CEREN  
YAŞAR  
14011020

# YÖNTEM

## Lbp\_out\_of\_range

```
For i in range 00000000 to 11111111
  For j in range 0 to 7
    Sum equal 0
    If j.bit different from j+1.bit then
      Sum++

  If sum <=2 then
    Add integer value of binary number to list
```

1. 8 bitle ifade edilebilen 00000000-11111111 aralığındaki tüm sayıları için bir döngü aç.
2. Döngü içinde 0.bitten 7.bite kadar 0-1 ve 1-0 dönüşümünü kontrol et ve her değişim için toplam değerini artır.
3. Eğer değişim 2 veya daha küçükse 10'luk tabandaki değerini listeye at.

## read\_png

```
For i in range 0 to length of list
  Open(file)
  Read image
  Row equal to image row
  Col equal to image column
  For i in range 0 to Row
    For j in range 0 to Col
      Read pixel's Red,Green and Blue values
      Temp equal to (2*Red+5*Green+1*Blue) divide by 8
      Add temp to list

  Add list to image2

  Call lbp_png for image2, row,col and set
```

1. Train ve Test resimlerinin listelerinin isimlerini içeren iki ayrı liste için read\_png fonksiyonunu çağır.
2. Listedeki resim sayısı kadar döngüyü döndür.
3. Resmi aç ve oku.
4. Resmin satır ve sütun sayısını oku.
5. Satır ve sütun sayısı kadar ilerleyen iki for döngüsü içinde resmin pixellerini oku ve onları griye çevir.
6. Griye çevrilmiş resim için lbp\_png fonksiyonunu çağır.

## Lbp\_png

```
For i in range 0 to 59
    hist[i] equal to 0
For i in range 0 to row-1
    For j in range 0 to col-1
        Generate Lbp of every pixel
        Call lbp_out for lbp,lbp_list,hist,row and col
```

Add hist to set

1. Her resim için bir tane 59 elemanlı histogram oluştur ve değerlerini sıfırla.
2. Resmin 1.satır ve 1. Sütunundan son satırın bir eksiği ve son sütunun bir eksiğine kadar her pixelin Local Binary Pattern oluştur.
3. Lbp\_put fonksiyonunu çağır.
4. Tüm resimlerin histogramlarını tutacak olan matrise resmin histogramını ekle.
- 5.

## Lbp\_put

```
Temp equal to 0
For i in range 0 to 8
    Temp equal to LBP's integer value

Cntrl equal to 0
i equal to 0
While i less than length of lbp_list and cntrl not equal to 1 then
    if temp equal to lbp_list[i] then
        Cntrl equal to 1
    i++

if cntrl equal to 1 then
    Hist[i] equal to hist[i] plus 1 divide by row*col
Else
    Hist[58] equal to hist[58] plus 1 divide by row*col
```

1. Lbp 10'luk tabanındaki değerini temp değişkenine atar.
2. Lbp\_list deki elemanlar ile listedeki elemanlar bitene ya da eşit bir değer bulunana kadar kontrol eder.
3. Eğer listede varsa listedeki bulunan değer indexine denk düşen indexteki histogram değerine 1'in satır\*sütun'a bölünmüş değerini ekler.
4. Listede yoksa histogramın son gözüne 1 in satır\*sütun'a bölünmüş değerini ekler.

## Find\_sim

```
For i in range 0 to length of list_test
  For j in range 0 to 56
    For k in range 0 to 59
      Temp equal to difference of train_set[j][k] and test[k][k]

      If j less than 5 then
        Add j and temp to sl

      Else
        ch equal to 0
        for l in range 1 to 5
          if sl[ch][l] less than sl[j][l] then
            ch equal to l

        if temp less than sl[ch][1] then
          Temp<->sl[ch][1]

    Sort sl ascending
    Add sl[j][0] to l
    Add l to sim_set
```

1. Test listesinin uzunluğu kadar dönecek bir for açar.
2. En benzer 5 resmi tutacak bir sl matrisi açar.
3. Train listesinin uzunluğu kadar dönecek olan bir for açar.
4. Train listesindeki her resmin histogramı ile Test listesindeki resmimizin histogramının farkını Manhattan Yöntemi ile bulur.
5. İlk 5 farkı direk matrise ekler.
6. İlk 5 farktan sonrası için matristeki en büyük farkı bulur.
7. Bu fark ile o anda bulunduğu farkı karşılaştırır eğer en son bulunduğu daha küçükse matrise onu yerleştirir.
8. Matrisi en çok benzeyenden en az benzeyene doğru sıralar.
9. Sonrasında matristeki resimlerin Train listesindeki indexlerini sim\_set matrisine ekler.

# UYGULAMA

Başarı Yüzdesi

2/3=%67

1.Test Resmi



1.Benzer



2.Benzer



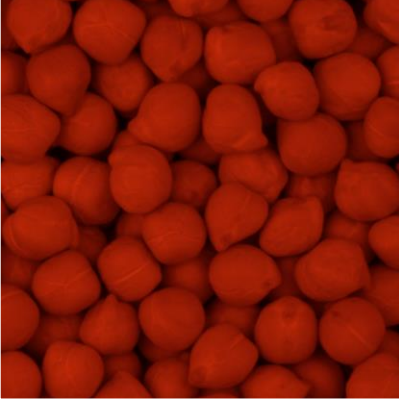
3.Benzer



Başarı Yüzdesi

0/3=%0

2.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

2/3=%67

3.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer

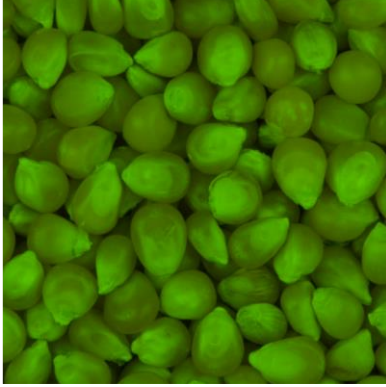




## Başarı Yüzdesi

0/3=%0

4.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer





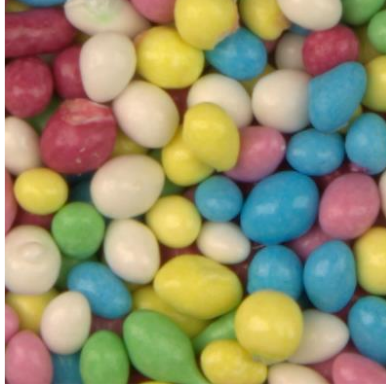
## Başarı Yüzdesi

3/3=%100

5.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



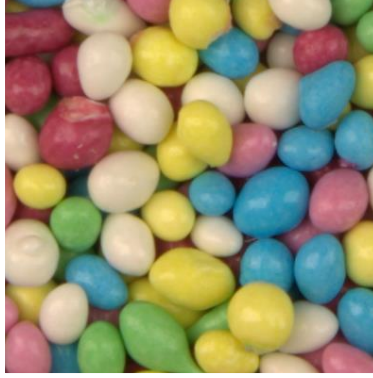
## Başarı Yüzdesi

3/3=%100

6.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

3/3=%100

7.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer





## Başarı Yüzdesi

0/3=%0

8.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

3/3=%100

9.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

0/3=%0

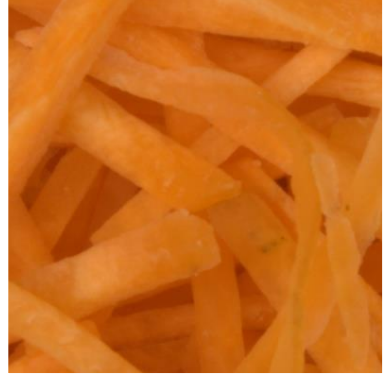
10.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

3/3=%100

11.Test Resmi



1.Benzer



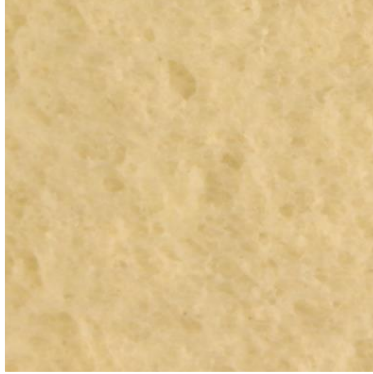
2.Benzer



3.Benzer



4.Benzer



5.Benzer





## Başarı Yüzdesi

3/3=%100

12.Test Resmi



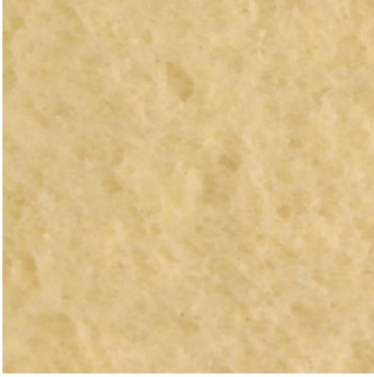
1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

3/3=%100

13.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

3/3=%100

14.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer

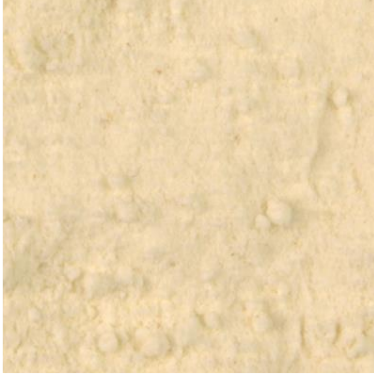




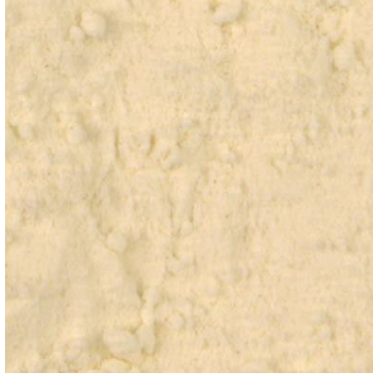
## Başarı Yüzdesi

3/3=%100

15.Test Resmi



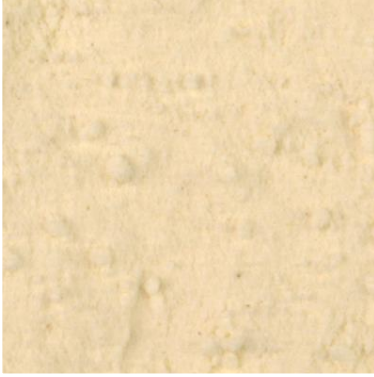
1.Benzer



2.Benzer



3.Benzer



4.Benzer



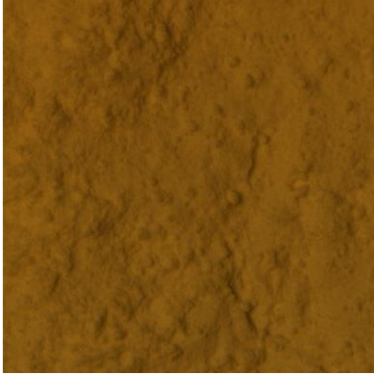
5.Benzer



Başarı Yüzdesi

0/3=%0

16.Test Resmi



1.Benzer



2.Benzer



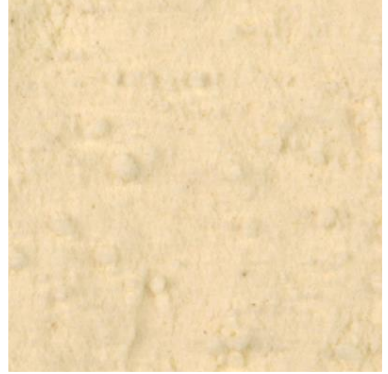
3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

3/3=%100

17.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

3/3=%100

18.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer





## Başarı Yüzdesi

3/3=%100

19.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



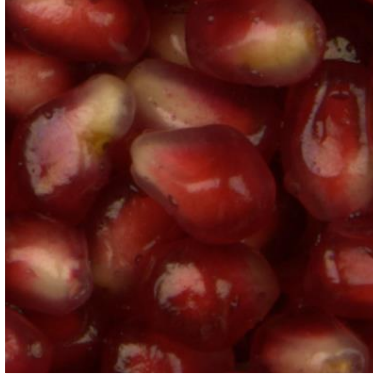
## Başarı Yüzdesi

3/3=%100

20.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

3/3=%100

21.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

3/3=%100

22.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer





## Başarı Yüzdesi

3/3=%100

23.Test Resmi



1.Benzer



2.Benzer



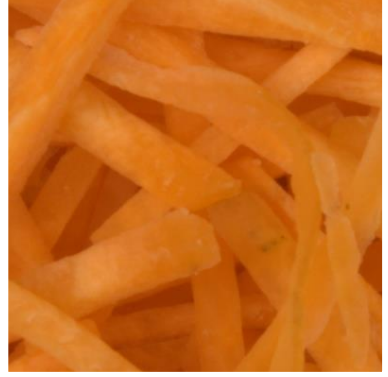
3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

2/3=%67

24.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



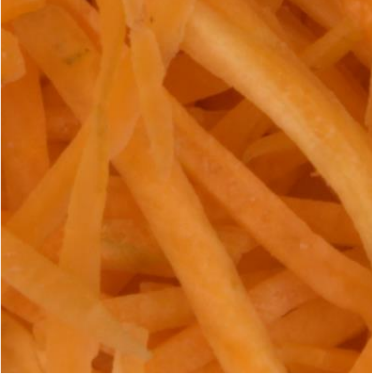
5.Benzer



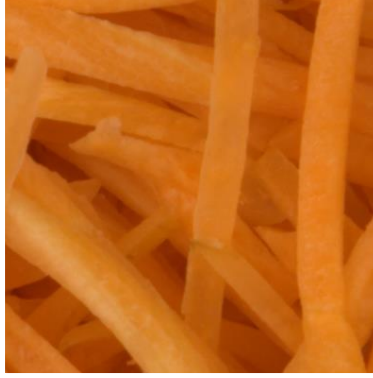
## Başarı Yüzdesi

3/3=%100

25.Test Resmi



1.Benzer



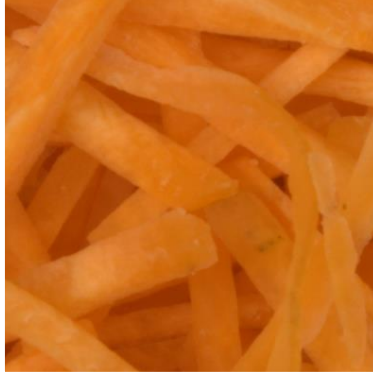
2.Benzer



3.Benzer



4.Benzer



5.Benzer





## Başarı Yüzdesi

1/3=%33

26.Test Resmi



1.Benzer



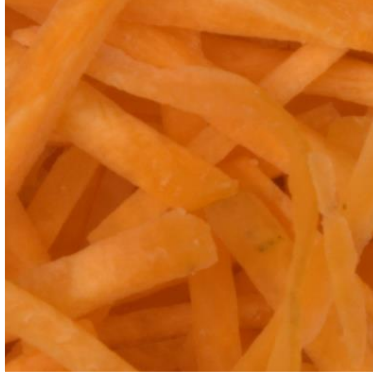
2.Benzer



3.Benzer



4.Benzer



5.Benzer



## Başarı Yüzdesi

3/3=%100

27.Test Resmi



1.Benzer



2.Benzer



3.Benzer



4.Benzer



5.Benzer



Başarı Yüzdesi

0/3=%0

28.Test Resmi



1.Benzer



2.Benzer



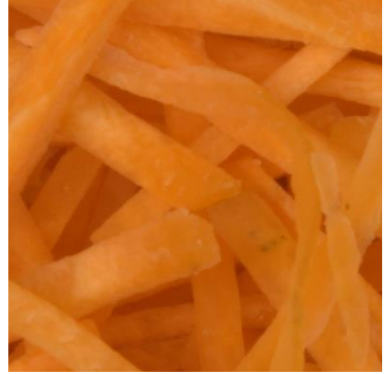
3.Benzer



4.Benzer



5.Benzer



## SONUÇ

Gözlemlediğim kadarıyla elimizdeki resim dokuya yakın ya da uzak baksın genellikle buluyor. Ancak hem daha uzaktan bakıp hem de renk değişimi olduğundan genellikle bulamıyor. Doku yakından ve renk değişimi olduğunda renk değişiminin oranı ve dokunun daha belirgin şekilli olmasına bağlı olarak bulma oranı ters yönlü olarak düşüyor. Daha az belirginlik içeren dokularda renk değişimi de olsa bulmayı başardı.