

Görüntü İşleme Giriş

Tubitak proje lisesi

- Digital Image Processing (6th Edition) Rafael C. Gonzalez
- <http://www.imageprocessingplace.com>

2 döv 0/20

1 proje 0/15

yatılım 0/20

final 0/25

Sayısal Görüntü

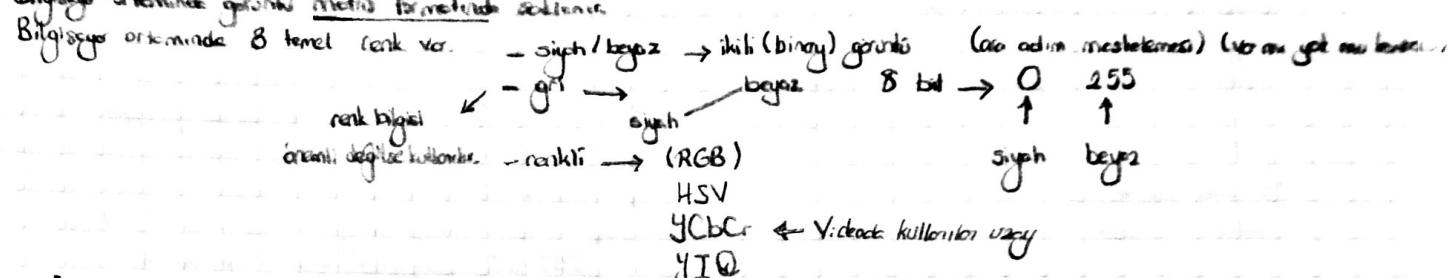
Sayısal görüntü, analog görüntüden pixel olarak itde edilen sayısal değerlerle gösterilebilir, 2 boyutlu hellerdir.

Görünür istek: 400-700 nm arasıdır.

mor kırmızı

pixel: picture element

Bilgisayar ortamında görüntü metris formatta saklanır.



Önlemleme-Kuantolama

Koordinat değerlerinin sayılaşılması: Önlemleme (Sampling)

Rengin eşitsizliklerini: Kuantolama (quantizing)

Üçlüşel yoğunluk

Üçlüşel yoğunluk, görüntüdeki aynı entiteyi en iyi şekilde tanımlamak.

Partikül yoğunluk

Partikül yoğunluk, görüntüdeki aynı entiteyi en iyi şekilde tanımlamak.

2 boyutlu resimlerde kullanılanlar

Low level process

• Gris ölçük verilen görüntüyü iyileştirme, düzeltme

Middle level process

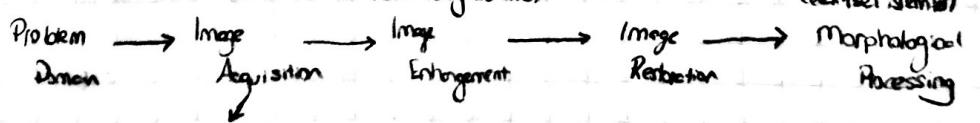
• Verilen görüntüden özelliği elde etmek

- Matris işleme
- Doku bilgisi
- Nette toplamak bilgisi

High level process

• Gris ölçük yarın yapmak
zone aturma
- Görüntünün entitelendirilmesi
- Farklı hallerde bulgular

(Görüntü iyileştirme)



(Sekizel işlem)

• Video iyileştirme

• Nette toplamak işlemi

(Ober etc.)

↓
Representation
Description

Görüntü döngü verilerinin

sayısalleştirilmesi bilgisi

ortasına eklenmesi

Yapılacak işleme uygun senar

• X-Ray imaging bulutlu hallerdir.

• Ultraviyole ray imaging

• Mikrodalga band imaging (Radar)

• İmaging in the radio band

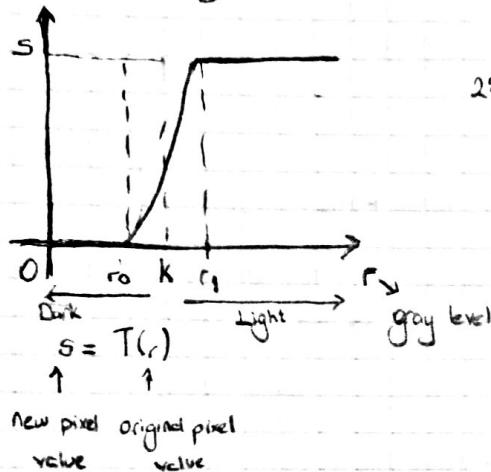
Image Compression (Görüntü sıkıştırma)

* VQ + Huffman sıkıştırma

Colour Image Processing:

Görüntü İşleme Point Process İşlemleri

- Görüntüdeki bilgilere göre işlemi gerçekleştir.



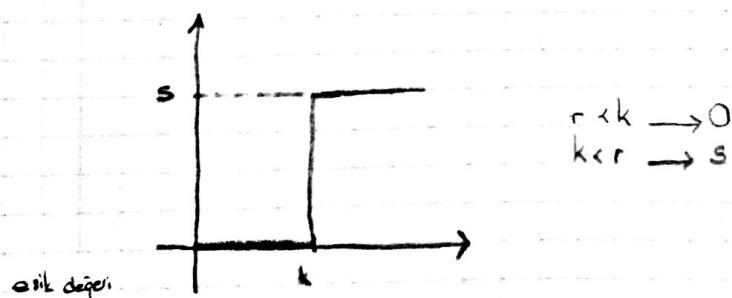
0 → siyah
255 → beyaz

$$\begin{aligned} 0 < r_0 &: 0 \\ r_0 < r &: \text{artı} \\ r_1 < r &: s \end{aligned}$$

$200-255 \rightarrow 200$ pixel koyulacaktır.
 $0-40-220 \rightarrow 5$

- r_0 değerine kader siyah kalsın
- r_0 değerinden sonra k ’a kadar artınsın
- r_1 değerinden itibaren belirli degerde sabitlensin.

* Bilgi kaybı olacaktır.
(pixel)



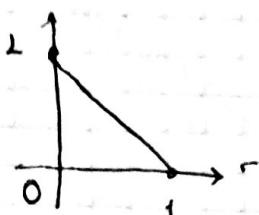
çok değer

- k değerinden önce beyaz k değerinden sonra siyah olsun.
- Eski değerden kararlılığı diğerlerinden ayırmayı sağla. (siyah beyazdan ayırtla.)

* NOT: Bir pixel içerisindeki tonları paralelden etkileşir.

Görüntü Negatif.

Negatifin etkisi her birin boyutunu tersine çevirmektedir.



$$s = L - 1 - r$$

Yanlışlardan biri:

- Ortalama degerlerin tersi

genetik histogramma göre bir histogram

olur. Bir histogramda her bir degerin frekansi birer bir degerin frekansı olur.

key frame olarak

bir resimdeki renk degerinin
sayisini gösteren grifdir.

maximum gray level (255)

Görüntü Histogram

görüntükteki pixelin gri degerine
istekli degit sıklıkta.

r_0, r_1, \dots, r_L

$$p(r_k) = \frac{n_k}{n} \rightarrow k^{\text{th}} \text{ level pixel number}$$

[0, 1] arsında degişti.

$n \rightarrow \text{total number}$

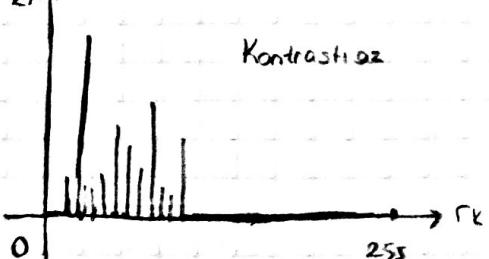
k^{th} gray level

probability of k^{th} gray level

Kontраст: Görüntüye ayırması

• Koyu goruntu

$$p(r_k)$$



Okçuluk obekt o renkle o pikelle

kaç tane oldug

- Yüksek kontрастlı goruntude pixelin bulunma olasılıkları data dengele doğrudır.

iyi sonuçlar veren goruntuelidir.

→ Görüntü kontраст: içe kontrest ettilerken goruntü iyileştirilir.

İstekli degit
sıklıkta
değişimi

Histogram Equalization → histogram esittene bir resimdeki renk degerinin belli bir yerde kumelenmesi amaciyla histogram, her degitin degitligini giderken birbirini genetir.

Yaz tanıma

• Renk degerinin data iyi yesilmesi için kullanılır.

• Histogram esittene yesildeki isitten kaynaklanan (kirk vege yesil) lapiseleri engeller.

• Üstten bir degitin ortaya çıkartır.

birbirini genetir.

Yazın belli bilgi kilitinde.
Bütün gri bilgi kilitinde. Histogramlar
almak 50% ekstra zaman alır.

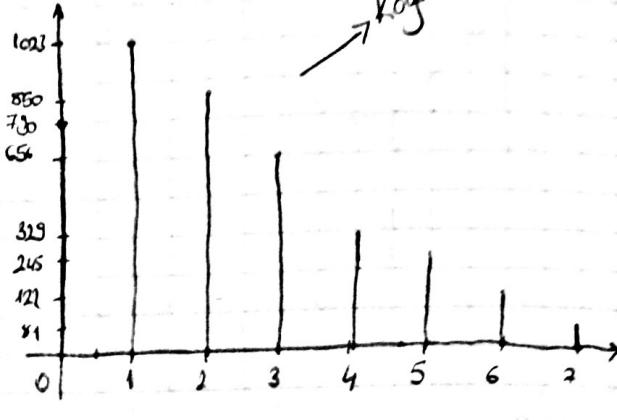
$$\textcircled{1} \quad p(r_k) = \frac{n_k}{n} \quad 0 \leq r_k < 1 \quad 0 \leq k \leq L-1$$

\downarrow
of gray levels

$$\textcircled{2} \quad D(W) = T(r_k) = \sum_{j=0}^k p(j) \quad 0 \leq k \leq L-1$$

k	r_k	n_k	p(r_k)
0	0	790	0.19
1	1/7	1023	0.25
2	2/7	850	0.21
3	3/7	656	0.16
4	4/7	329	0.08
5	5/7	245	0.06
6	6/7	122	0.03
7	1	81	0.02

4.096



$$S_0 = T(r_0) = p_0(r_0) = 0.19 \rightarrow 1/7$$

$$S_1 = T(r_1) = p_0(r_0) + p_1(r_1) = 0.44 \rightarrow 3/7$$

$$S_2 = T(r_2) = p_0(r_0) + p_1(r_1) + p_2(r_2) = 0.65 \rightarrow 5/7$$

$$S_3 = 6/7 \quad S_4 = 6/7 \quad S_5 = 1 \quad S_6 = 1 \quad S_7 = 1$$

• Bulutlu olsakta da yeni renk分配される。

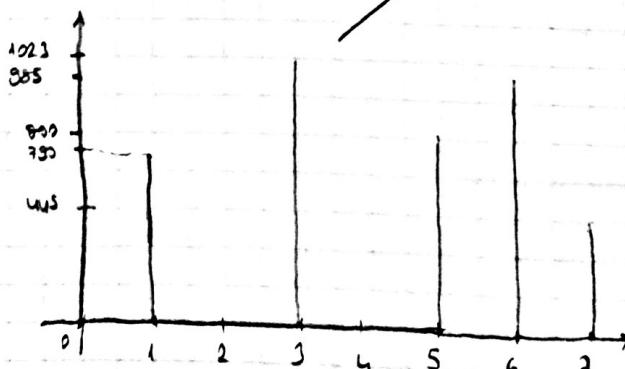
• Bulutlu sıkılıktır göre分配される。

yapılır.

Xeni renkler	k	know	Outcome
	0	1	790
	1	3	1023
	2	5	850
	3	6	656 + 329 = 985
	4		
	5		
	6	7	245 + 122 + 81 = 445
	7		

Renkler分配される。

know	p(know)
1	0.19
3	0.25
5	0.21
6	0.24
7	0.11



4 renge göre etglim istenirse

0	Renk olsakta bulutlu olacaktır.
1/3	
2/3	

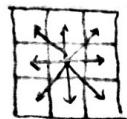
Bölgeel okul histogram isteminin yapılması
Local histogram →

Üzeyit Korelasyon (Nokteler konumları ile doğrudan ilişkili)

Yüksek korelasyon = Konsantre birleşen noktalar

Dürtük korelasyon = Konsantre toplu noktalar

En çok konsantre konvolut:



→ 3x3

5x5

7x7

Image Filtering
Spatial Filters

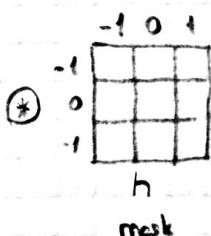
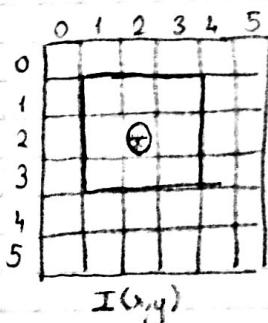
① Linear Filters

$$I(x,y) \rightarrow f(x,y) \quad \text{convolution} \quad (\text{her pixel convolution işlemi yapılır})$$

Konsantreyle konsantre denimine konsantre.

$$g(x,y) = f(x,y) \otimes h(x,y) = \sum_{i=-\frac{m}{2}}^{\frac{m}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} f(x+i, y+j) * h(i, j)$$

output pixel convolution mask



$m=3$

$$\sum_{i=-1}^1 \sum_{j=-1}^1 f(x+i, y+j) * h(i, j)$$

$$h(2,2) = f(1,1) * h(-1,-1) + f(1,2) * h(-1,0) + f(1,3) * h(-1,1) + \\ + f(2,1) * h(0,-1) + f(2,2) * h(0,0) + f(2,3) * h(0,1) + \\ + f(3,1) * h(1,-1) + f(3,2) * h(1,0) + f(3,3) * h(1,1)$$

• Etki eden sonuc normalize edilmeli.

Ön

$0 \rightarrow 255$

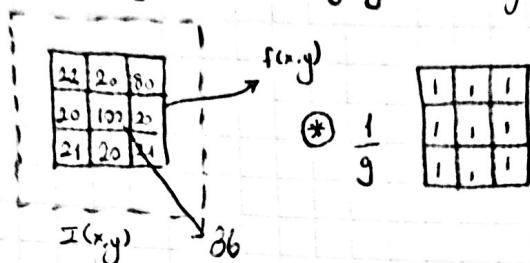
$mn \rightarrow mn$

-5 → 400

Smoothing Filters (Low pass filters) merkezi sıkılıkla sınırlı etkisi yok.
 Gauss: Gaussian: ortalamaya uygunluk. Blurbanan ve gürültüyü kırınma için kullanılır.
 Pencere: Pencere: çevreleme.

Averaging Filter

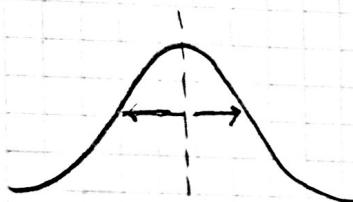
Ortalama algoritması: doğrudan ortalamayı yapmakta. Geleneksel hale gelir.



$$f(x,y) = \frac{1}{9} [22+20+80+20+100+20+21+20+21] = \frac{324}{9} = 36$$

Gauss Filter

Görüntüdeki detayları kırınma için kullanılır.

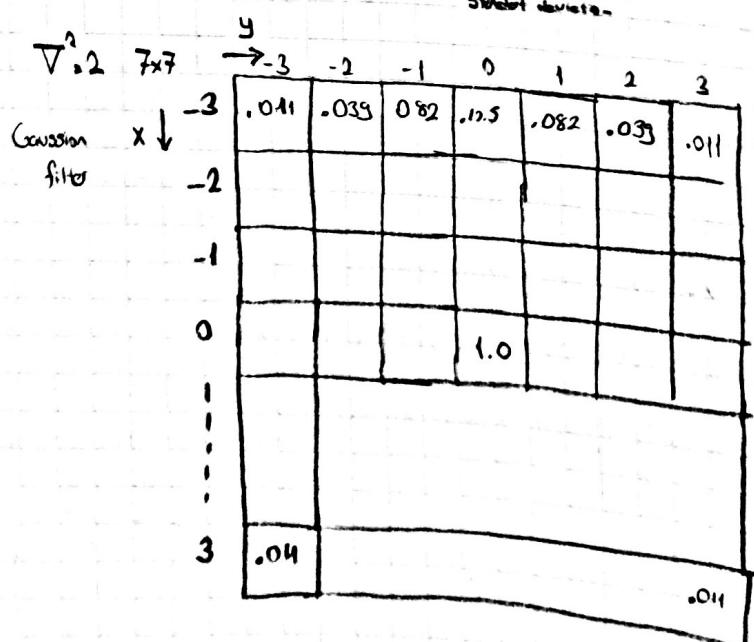


Merkazdeki değerlerin olasılıkları daha etkilidir.

$$1\text{-D Gaussian: } G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

$$2\text{-D Gaussian: } G(x,y) = \frac{1}{2\pi\sigma_x^2\sigma_y^2} e^{-\frac{x^2+y^2}{2(\sigma_x^2+\sigma_y^2)}}$$

Cen: genetikini etkiliyor. Gen boyutu ve tane sayıda da etkiliyor. Gen: heterotípik.



tam sayılara çevirme

1 4 7 10 7 4 1

$$x = -3 \dots 3$$

$$y = -3 \dots 3$$

$$\text{① Fitting curve et } G(-3, -3) = \frac{1}{2x,2} e^{-\frac{(-3)^2 + (-3)^2}{2+2}} = 0.011$$

o Ver sera gente del stat
deportivo Olivos.

$$G(0,0) = \frac{1}{2\pi \cdot 2} e^{-0} = 1$$

- Are smaller clusters more numerous?
- Major genetic clusters are found at the extremes between

② Filtrejeli kesikli dökümler kurul
→ Taramayı nasıl getir.

O Otl en kwaik dager

$0.01 \rightarrow 1$ yapanıza, ($0.1e$ losunu istem yapanıza) $0.01 \rightarrow \frac{h(x,y)}{0.01}$

$$h(0,0) : \frac{1.0}{0.011} \rightarrow \underline{\underline{g_1}}$$

(3) Tamsoyde istem yapılıcak sonucu bulacaktır.

$$\sum_{x=-3}^{2} \sum_{y=-3}^{2} h(x,y) = 1445 \quad \text{→ filtersiz regulen topom}$$

$$g(x,y) = \frac{1}{m^2} \cdot f(x,y) \odot h(x,y)$$

Median Filter \rightarrow 3x3 kuvan lisensointi.

Gevreinde uygun olmaya pişiklik caressiyle uygunluğa getirmeye çalışırıysa

Low pass (averaging filter)

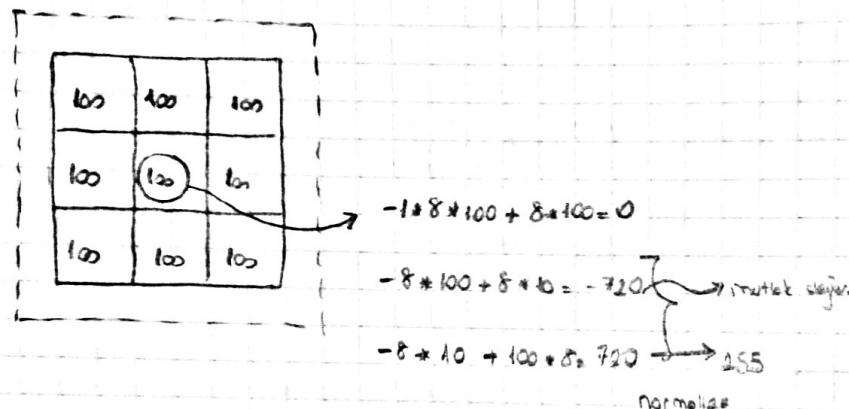
$$\frac{1}{9} \left(15 + 18 + 19 + 4 + 20 + 22 + 103 \right) = 28$$

Medior
Fitter

High pass filter. \rightarrow Renk genetiksel algoritmasında kullanılır. Amaç, gizli faktörleri ortaya koymaktır.

\rightarrow mutlak değer olmaz istenilen gibi

-1	-1	-1
-1	8	-1
-1	-1	-1



Edge Detection Etrafında farklı olan pikselleri bulmak edge tespiti
Prewitt Filter

1-D



$$S = S[i] - (S[i-1], S[i+1])$$

$$S' = S[i] - S[i-1], S[i+1] - S[i]$$

Örneklenme Mesafesi $\Delta x=1$

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{\Delta x}$$

$$\Delta x=1$$



$$50/100$$

$$|100 - 50| = 50$$

2-D

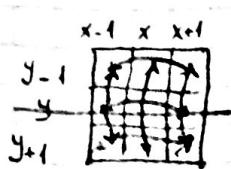
x genetiksel

$$\frac{\partial I}{\partial x} = \frac{1}{3} \left[\frac{I(x+1, y) - I(x-1, y)}{2\Delta x} + \frac{I(x+1, y-1) - I(x-1, y-1)}{2\Delta y} + \frac{I(x+1, y+1) - I(x-1, y+1)}{2\Delta y} \right]$$

bilim

Hesap yapın son uygulama için

$\frac{1}{6}$ degeri uygulayın ve yatsı



$$\frac{\partial f}{\partial y} = \frac{1}{3} \left[I(x,y+1) - I(x,y-1) + \frac{I(x+1,y+1) - I(x-1,y-1)}{2} + \frac{I(x-1,y+1) - I(x+1,y-1)}{2} \right]$$

$$M_y =$$

(matriks)

	$x-1$	x	$x+1$
$y-1$	-1	0	1
y	-1	0	1
$y+1$	-1	0	1

$$M_y =$$

(matriks)

	$x-1$	x	$x+1$
$y-1$	-1	-1	-1
y	0	0	0
$y+1$	1	1	1

İki yonelik degrəni garnet isteqmək

• Filtre her tərəfənən əsaslıdır.

• Bu iki filtre degrəni təpki

→ Gradient of $I(x,y)$

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \text{mag}(\nabla f) = \sqrt{\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2} = \sqrt{G_x^2 + G_y^2}$$

100	10	10
100	10	10
100	10	10

$$G_x = (10 - 100) * 3 = (-270)$$

$$G_y = (10 - 10) = 0$$

$$G = \sqrt{G_x^2 + G_y^2} = 270 \text{ min} \Rightarrow 0 \text{ max} \Rightarrow 255$$

$$\min \Rightarrow 0$$

$$\max \Rightarrow 255$$

10	10	10
10	10	10
10	10	10

$$\left. \begin{array}{l} G_x = 0 \\ G_y = 0 \end{array} \right\} G \rightarrow 0$$

Gərçək dərinlik yox sayılır
Gərçək olub yox giri

$$② m_{xy}(v_i) = 10x_i + 10y_i$$

Byk. 2. mukde
Sıra ve düzle
ekleme

(1-D)

Subel filter Pixelin (sağ-sol, yukarı, aşağı, daire etkili)



$M_x =$

-1	0	1
-2	0	2
-1	0	1

$M_y =$

-1	-2	-1
0	0	0
1	2	1

* Resim parçası varsa pozitifde geçiş olurken bir patrone obaktır.
daire etkili obaktır.

• Edge Orientation (Edge yönü bulma) sağa

$$\theta = \arctan \frac{G_y}{G_x}$$

→ Paritten külentenin Gauss filtresi. külenten.

daire genis obak bekti.

Amaç genis obak tekrarlı yapmak.
Daire tekrarlı

Yük. Tavsiye için külenten.
Ancak Gauss Matematik.

• Gauss ve Edge Detection birlikte yepilen bir yöntem geliştirmek

2. özerken külenten

2nd order derivatives

1.1.

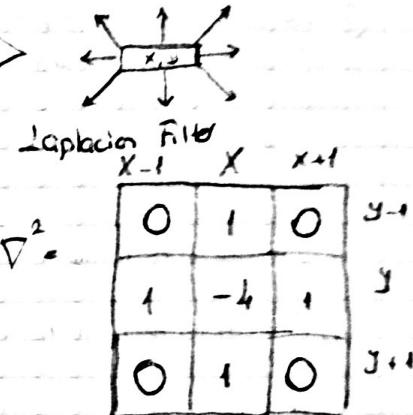
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial [f(x,y+1) - f(x,y)]}{\partial x} = \frac{\partial f[x,y+1]}{\partial x} - \frac{\partial f[x,y]}{\partial x}$$

$$= f[x,y+2] - 2f[x,y+1] + f[x,y]$$

$$= f[x,y+1] - 2f[x,y] + f[x,y-1]$$

Nitekz (x,y) olusa



$$\frac{\partial^2 f}{\partial y^2} = f[x+1,y] - 2f[x,y] + f[x-1,y]$$

Q3

100	100	100
100	100	100
100	100	100

0

100	100	0
100	100	0
100	100	0

$300 - 400 = 0$

Left detector
start for

Laplacian of Gaussian (LoG)

$$h(x,y) = \nabla^2 [g(x,y) * f(x,y)] = [\nabla^2 g(x,y)] * f(x,y)$$

LoG

mask convolution

$$-\frac{x^2 + y^2}{2\nabla^2}$$

$$\nabla^2 g(x,y) = \left(\frac{x^2 + y^2 - 2\nabla^2}{\nabla^4} \right) a$$

$\nabla^2 2$

0	0	-1	0	0
0	-1	-1	-1	0
-1	-2	-4	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

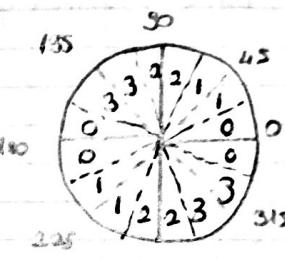
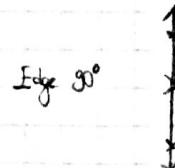
Edge \rightarrow resit degerlerdeki okular resme
stabildegilere iyi edge etrafında.

① Gauss filtre ile peronelde temizlenir.

② Sobel filtre ile antifer bir hale gelir.

$$\text{ve ya Robert's filtre ile } M_x = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad M_y = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

③ Edge açısı hesaplanır. $\theta = \arctan \frac{G_y}{G_x}$ * Görüntükte açılar 45°'er olmaktadır.



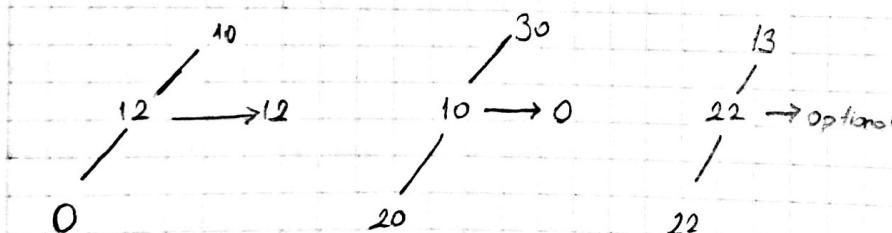
④ Yan değerini disk üzerindeki [0-3] aralığında değiştirebilirsiniz.

⑤ Non maximum suppression ile zayıf eğimler silinir.

Non maximum suppression

Eğer bir edge pixelin edge yarımdağındaki komşularının değerleri kendisinden büyükse o silinir.
Birimde 0 yapılır.

Strong Edge Detector



(geliş)

⑥ Hysteresis thresholding yöntemi ile edge pixeller bayrağı düşülebilir.

Ölçüm convolution yöntemi uygulanır.

Pozitif very edge detection

pgm \rightarrow portable gray map

\uparrow genel bilgiler bulunur. (Kesici satır - konum)

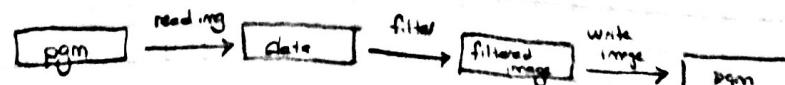
grayscale formats : Bir header, bir data bloğu bulunmaktadır.

\downarrow matris okutulur

format: primitive

header 2Byte

2 byte nrow ncol width



\rightarrow C de yerleştir.

fopen

freac

fwrc

Sobel = Prewitt'e
başlıyor.

Segmentasyon Nedir?

"Görüntünün farklılık ve birbirini tek kullanımayan bilgilerine" uygunlaşır.

- Görüntü tiplarına göre
- Forma ve renk resimlerine göre
- Uzaklık

- Tek bir nesneyi veya nesnenin farklı parçalarını gösterir.
- İnkıllı pixeller birbirine bağlıdır.
- Pixellerin ortak özellikler vardır (renk, dokular, eğriler).
- Uniform genel

→ Derinlik bilgisine göre segmente edilebilir

Renk
Kenar
Doku

- İki renk birbirine benzeyen iki okulda okulları tutar.

$$\sqrt{\sum_{i=0}^m (p_i - q_i)^2}$$

Genelde derinlik tutar.

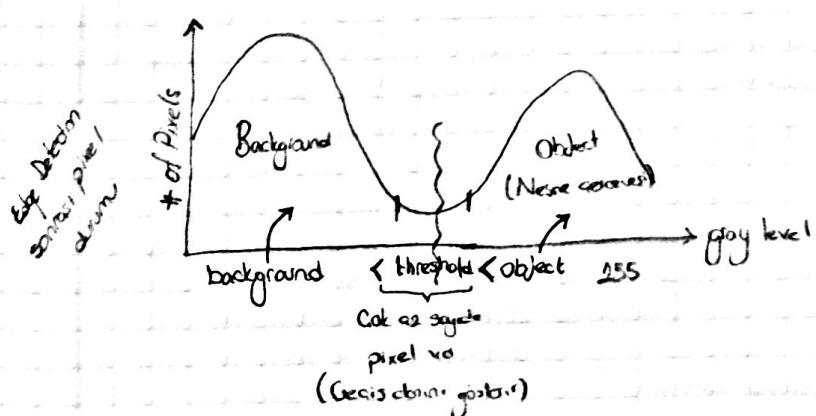
Şimdiği: Benzer olan bilgiler aynı segmente topluyolarlar. (Eskişehir kentinden skin detection)

Discontinuity;

Segmentasyon Yöntemleri

- * Edge Based
- * Region Based : Renk bilgilerine dayanır. (Kenar Tuttanlı) (Sıfır Pekanlı)

Görüntü durumuna göre bayra ve sıfır dağılımları farklıdır.



→ Edge olur ve aralığı şekilde sona erde edilir.

Histogram after edge detection

Kural:

If $f(x,y) > T$ then $f(x,y) = 1 \xrightarrow{255} \text{object}$

else $f(x,y) = 0 \Rightarrow \text{background}$

- Farklı gri seviyeleri tek eşit değer ile概括 etmek.
- (sayı - bayan)

Global Eşik
Local Eşik

GüçlüLOGYEVSİ eşit değer
olmak istenir.

- Geçerlilik nesil belirleme?

Global eşit seviyesi belirleme

① Adaptive Thresholding (iyi sonucu sağlamak için. Hızlı olası.)

- T eşit seviyesi degeri için basitçe değer belirler.
- T 'den küçük olan değerlerin ρ_1 ρ_2 olacak şekilde resimdeki gibi gösterilmesi.
- ρ_1 ve ρ_2 nin N_1 ve N_2 ortaklık renk değerleri hesaplanır.
- Yani T değeri $T = \frac{1}{2}(N_1 + N_2)$ hesaplanır.

$$5. \Delta T = |T_t - T_{t+1}|$$

$\Delta T < \epsilon$ olana kadar 2. adımdan devam edilir.
hata sınırları.

ardından iki eşit seviyesi

$T=10$

$$\begin{cases} f(x,y) < T \rightarrow G_1 \\ f(x,y) \geq T \rightarrow G_2 \end{cases}$$

Background ve object
2 sınıf elde edilir.

$$N_1 = \frac{1}{T+1} \sum_{i=0}^{T-1} p(i)$$

$$N_2 = \frac{1}{(L-T)} \sum_{i=T}^{L-1} p(i) \rightarrow 255$$

Pixel sayısı

② Otsu Yöntemi

2 sınıf gibi düşündür. Bu iki sınıfı birbirinden farklı ayırtmeye çalışın.

Eşit seviyesi belirleme
Within-class variance min
between-class variance max

$$\nabla_w^2(t) = q_1(t) \nabla_1^2(t) + q_2(t) \cdot \nabla_2^2(t)$$

$T = t_c$ threshold

Sınıf ~~okunaklı~~ okunaklı.

$$q_1(t) = \sum_{i=0}^{t-1} p(i)$$

$$q_2(t) = \sum_{i=t}^{L-1} p(i)$$

Sınıf ortalamaları

$$\mu_1(t) = \frac{\sum_{i=0}^{t-1} i * p(i)}{q_1(t)}$$

objekt n. pixel nomb
total number

$$\mu_2(t) = \frac{\sum_{i=t}^{L-1} i * p(i)}{q_2(t)}$$

standardize before feed it. It is needed to make the features have the same scale.

Sınıf varyansları

$$\nabla_1^2(t) = \sum_{i=0}^{L-1} [i - \mu_1(t)]^2 \frac{p_i(t)}{q_1(t)}$$

Normalizasyon

$$\nabla_2^2(t) = \sum_{i=t}^{L-1} [i - \mu_2(t)]^2 \frac{p_i(t)}{q_2(t)}$$

Toplam varyans:

$$\nabla^2 = \underbrace{\nabla_w^2(t)}_{\text{Within class variance}} + \underbrace{q_1(t) [1 - q_1(t)] \cdot [\mu_1(t) - \mu_2(t)]^2}_{\text{Between class variance}} \rightarrow \text{MAXIMIZE!!!}$$

① Initialization

$$q_1(0) = p(0)$$

$$\mu_1(0) = 0$$

② Recursion

$$q_1(t+1) = q_1(t) + p(t+1)$$

$$\mu_1(t+1) = \frac{q_1(t) \mu_1(t) + (t+1) \cdot p(t+1)}{q_1(t+1)}$$

$$\mu_2(t+1) = \frac{N - q_1(t+1) \cdot \mu_1(t+1)}{1 - q_1(t+1)}$$

toplan

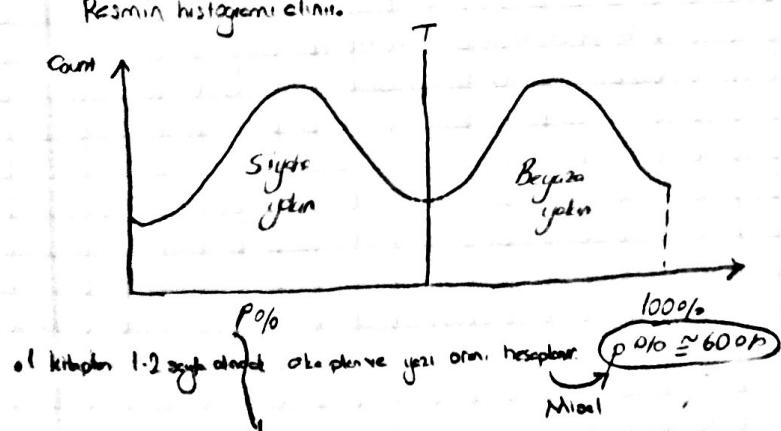
ortalaması

$$N = N_1 + N_2$$

* P-tile Eşik Yöntemi (Histogram, basit bir yöntemdir)

Nesne sınırlarını bilinen pixel oranında yes varyansı histogram ile eşit seviyesi bulmak

Resmin histogramı elde et.



Toplu toplu pixelin sayısı
birinci

* 1000 piksel 6000 piksel sayısının tarihanca osit varyansı diper. Diferans

* Arka planın okunan(pixel) yes oranı
Histogram ile eşit seviyesi bulmak

Local Estl (Genetikle g parçalı bulur)

- 1. Genetik mem'lik potansiyelini ölçer.
- 2. Heselenen en yüksek potansiyeli bulur.

Canny or kucuk olursa doğru edge
bulur.
Canny or büyük olursa diğer edge
edge #sayt bulur.

Edge Thresh Segmentation

- Based on sets Bir resim belirliye göre sınırları tespit etmek için.
- Each image R is a set of regions R_i .
- Every pixel belongs to one region

$$R = \bigcup_{i=1}^S R_i \quad R_i \cap R_j = \emptyset \text{ if } i \neq j$$

... Resim içinde birbirin segmantleri bulur.



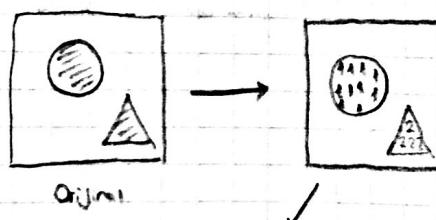
Filling Algorithms

① Region Growing

Bir noktadan başlar
8 komşusuna bakılır.
Benzetile bülgeye dahil edilir.

Original resim boyalıca. Kopya sevmek
üzerine getirilen 1. yapılmış
olmazsa sevmek ve islene devam
edilir.

Bir pixel sevmek ve 8 komşusu
yönde recursive olarak istenilen
Bütün resim için bu yapılmış. Bölgeler elde edilebilir.



etiketlenmiş birlikler
ayırma işlemi yapabilmek

quantization ile rank
sayısı azaltılabilir.

256 → 16
rank → rank

Color ve texture ayırmaları yapılabılır.
(status)

② Region Merging (↑Bottom-up)

- Divide image into an initial set of regions
- One region per pixel

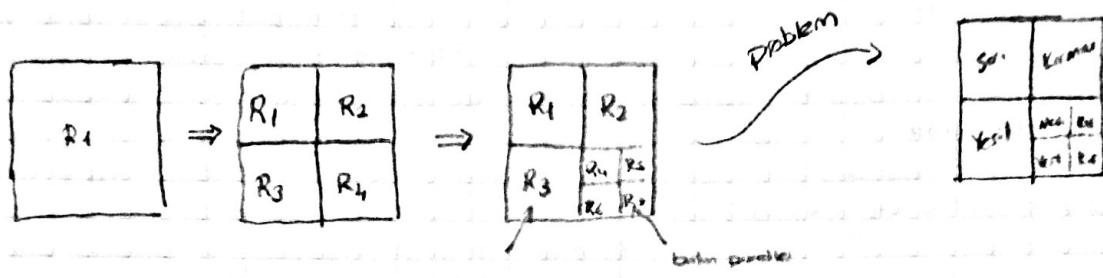
- Bir pixel bir region maneviye
bulur.
- Geçersizki pixelin en yüksek region
coğrafi

Region Growing teknigi:

Region Splitting

Horng tekniklerde
algoritma

- Birer bir bölge olarak algıyalır.
- Birer bir bölge halinde analiz eder.
- Hangen etrafına pek çok girdiye de pek çok bilgiyi alır.
- Hangen etrafına tek bir girdiye de pek çok bilgiyi alır.



- R_3 ile R_4 aynı renkte olabilir. Bu durumda ne yapılmalıdır?

✓ Gözüm

Split & Merge

- Combination of both algorithms

- Can handle a larger variety of shapes

Simply apply previous algorithm consecutively.

Renk Nedir?

Trafik ışıkları

- * Renk, ışığın değişik dalga boylarının gözün retina sine ulaşması ile ortaya çıkan bir algılamadır. \rightarrow Yüz tanıma (renk)
- * Farklı renkler farklı dalga boylarında (λ) ışınım yapar.
- * Tüm orta boyalar sinyali orta gözümüze ulaşırsa bunu beyaz, hiç ışık ulaşmazsa siyah olarak algılar.
- * insa gözü 380 nm ile 780 nm arası görür. (Görsel ışık)
 \rightarrow Bu kısma görürsü wit denir.

İlk kriterleriniği

- * Işık akromatik (renksiz) ise tek özelligi parlaklık (intensity) algıları. Sıyah beyaz televizyonlarda genetik bir tipdir.
- * Kromatik ışık : 400-700 nm arasıdır.
Yözelikler:
 - Radiance = Işınılık. Işık kaynağından gelen ışık miktarı.
 - Luminance = Işıklılık. Bir ışık kaynağı yararlaşırken parlaklık ölçüsü. Gözle görülen ışık miktarı.
 - Brightness = Renk parlaklığı

Cores ve rods görme için gerekliye katınlara hücreler.

İnsan gözde renk tercih etmede sayılır.

RGB:

CRT, Plasma ve LCD monitörlerde kullanılır.

Verimli renkler CMYK: (Cyan, Magenta, Yellow, Black)

Kullanılır. Yazıcılerde kullanılır.

HSL: (Hue, Saturation, Intensity)

Pek bilgisayarlarla istenilen kulanır.

YIQ-YUV (Luminance - Chrominance)

YCbCr Videoformat kulanır.

RGB: Belirli oranda red, green, blue toplamı

1 pixel :	R	G	B
	↓	↓	↓
Kirmizi	0 255	0 255	0 255
Yeşil	255	0	0
Grı	0	255	0
	$R+G+B$		

* Renk utangında dijital renklerdir.

$$\text{Toplam} : (2^8)^3 = 16777216 \text{ renk gösterilebilir.}$$

RGB Normalizasyon :

$$r = R / (R+G+B)$$

$$g = G / (R+G+B)$$

$$b = B / (R+G+B)$$

$$r+g+b=1$$

$$0-255 \rightarrow 8 \text{ bit} \quad \rightarrow 220 \quad 0 \quad 0 \rightarrow 0-255 \xrightarrow[0-1 \text{ aralığındakı}]{} 220/255$$

$$0-127 \rightarrow 7 \text{ bit}$$

$$\rightarrow 100 \quad 0 \quad 0 \rightarrow 0-127$$

$$100/100$$

CMY Uzayı

Düzenleme

$$\text{benzer } C = 1 - (B+G)$$

Renkler birbirinden ayırt edilebilir.

Luminance ve Chromaticity

(hue-Saturation)

Intensity-Brightness \rightarrow Lightness \rightarrow Luma-Value

Renk dizi, renk tonu
Düzenleme

* Kirmızı düşmən bir renk
* Dayanıklılığı 0 olur renkleri

geçer

\rightarrow Renk renklerin arasındaki mesafə
düşmən, səfli məskənlərdən

Hue, saturation

Color (Renk) \rightarrow Chromaticity \rightarrow Brightness

(Renk eynimədə)

RGB 24 bit külənləndirən
ayırılmış şəkildə.)

Potenslik

İstiklaliyət

Aşağı sağda külənləndirən

Üstən sağda

Üstən

1:100 orantı

RGB-HSI
Düzenleme

H: 0-360 Boşluq ölçüsü

S: 0-1 (0.grı, 1.beyaz) Nötrden yaşlı

I: 0-1 (Renk nötrinin yaşlılığı) (0.en yaşlı, 1.en şartlı)

HSI-RGB
Düzenleme: Yaptırılmış deşifrelerin gizlilik istenir.

Bu yoxlanır külənləndir.

RGB uzayına gidecək işi 11 adımlıdır.
HSV Modeli

50-600

↑
Yeşil + Kirmizi = Sarı

Yeşil + Mavi = Yeşil

Mavi + Kirmizi = Magenta

↓

600-700 nm

Yeşil + Mavi + Kirmizi = Beyaz

Luminance-Chrominance ($YUV \& YIQ$) \rightarrow Küçük Amerika NTSC teknolojisi.

Güvenlik video
kullanımı

Potansiyel ile renk bilgisini ayırmak emsali.

TV kırıcı buttonları

Renkler de de ezi biliği ile gizlendirilmeli.

- YÜZER, YPbPr, YDbDr, YIQ \rightarrow Renkli TV'lerde buttonları.

- RGB şablonları 4 parçaya bölünür:

1 luminance (grayscale) bileşeni = Y

3 chrominance (color) bileşeni = Cr, Cg, Cb (Mavi ve Kırmızı renk buttonları)

- Cr, Cb yetenli, Cg buttonlu receptor

chroma bilgisi

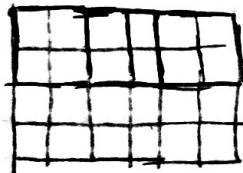
Chroma Subsampling $(4:4:4)$, $(4:2:0)$, $(4:2:2)$

öncek

sayısal sıkıştırma obi

\downarrow

enine yeri öncelenme, boyutu tara öncelenme
3deler ilk sayısında renk bilgisi buttonları

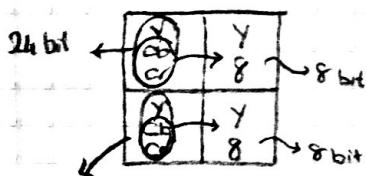


Sıkıştırma için 4 tane değerlendirilir.

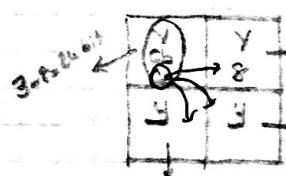
YCbCr

8 8 8

4:2:2



Yanındaki Cr/Cb deşifreni buttonlar.



8bit

Cr deşifre buttonlar.

1 pixel $\rightarrow 24 \times 1 = 24$ bit

4x24 = 96 bit

$$4:2:2 \rightarrow 24 \times 2 + 8 \times 2 = 48 + 16 = 64 \text{ bit}$$

$$24 + 8 \times 3 = 48 \text{ bit}$$

\rightarrow Gözde gri renk ekle edildik isteniyorsa YUV buttonları.

\rightarrow Gri resim buttonlarında buu buttonları.

\rightarrow Ortalama olursa gri renk ekle edilmesi

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.486 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

YUV

\downarrow \downarrow

luminance Blue Chroma Red Chroma

\rightarrow Renk varyasyonu koruyanı yapılış.

Paslı Renklerinleme

Gri renklerin range

denetirilmesi

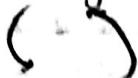
1. Yagınlık Dilekme

- Gri renklerin genetik $f(x,y) = I$ boyutlu
keşfet.

2. Gri renklerin denetim

- R, G, B kanallarının her biri 256 ayrı değerden
takip eden taktirde.

RGB \rightarrow HSI Image \rightarrow RGB



Histogram

\rightarrow (80-Hue Resmin negatifini west)

Renkli Görüntü Histogram

- RGB bilgileri için 3 histogram oluştur. 24 bit sayı
- 3 tonal. (R, G, B) ayrı histogramını alır. (Pekâl, her renkler birbirini ile başkası karışılı olsun.)
- Sadece Intensity değerinin histogramı alır.

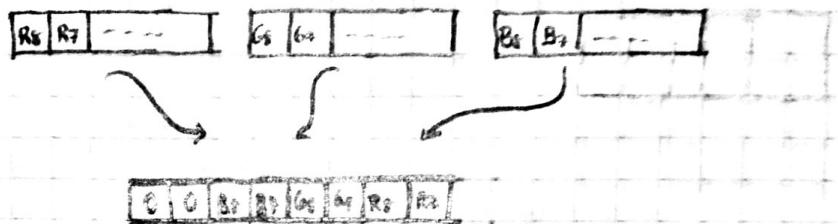
RGB \rightarrow HSI Hue histogram

RGB \rightarrow 64 bin histogram

abu birka bin bilgisini

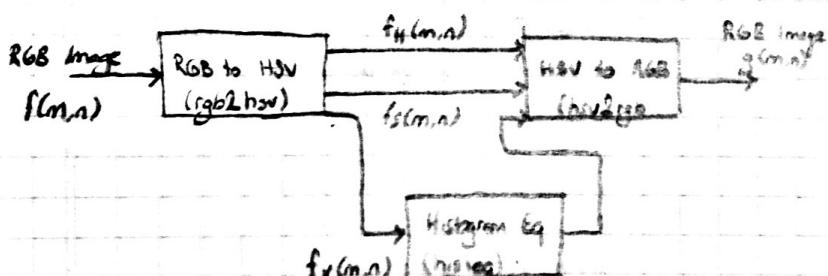
yüklük etmek. better renk bilgisini west.

• Intensity histogram



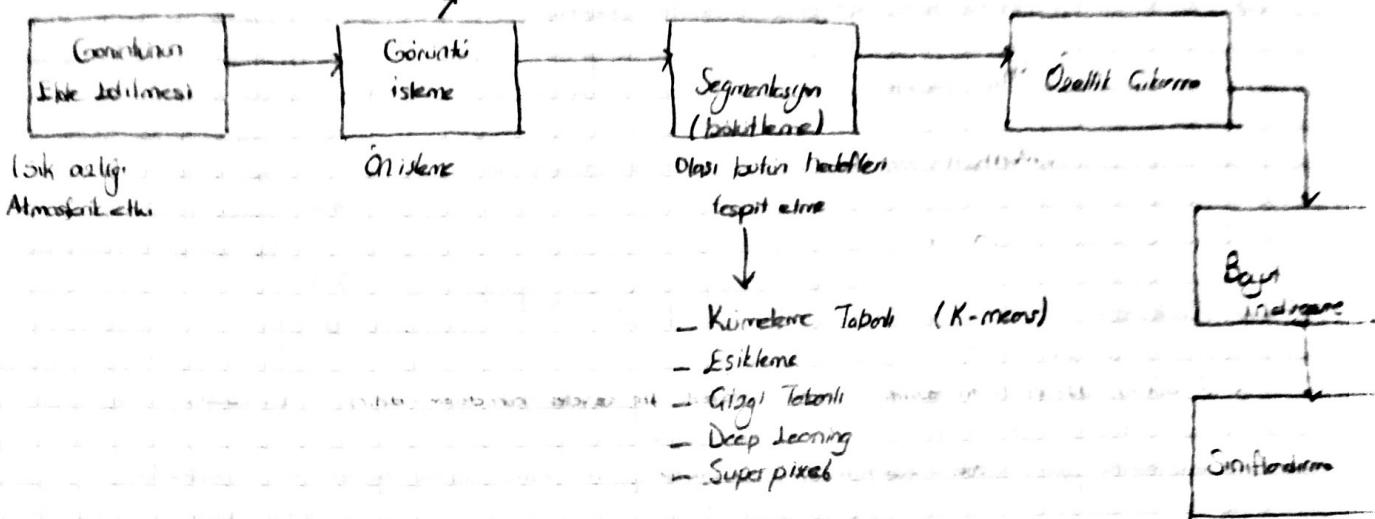
Renkli Histogram Esteme

Rengin kontrastını artırmak.



Clustering

Görüntü işleme adımları



Pixellering (upyeşdirilmişler)

- Unsupervised segmentation. (Denetimsiz sənədliyi)
- Required data, but no labels.
- Detects pattern
- Group email and search result.
- Customer shopping pattern.
- Regions and images.
- Useful when don't know what you're looking for

* Slide from James Hayes

Lecture 16, p10

Basic idea: group together similar instances.

$$\text{dist}(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\|^2$$

- Partition algorithms

- K means
- Mixture of Gaussian

- Hierarchical algorithms

- Bottom-up
- Top down

K-means

An iterative clustering algorithm.

Initialize: Pick K random points as cluster center

Alternates:

1. Assign data points to closest cluster center
2. Change the cluster center to the average of its assigned points

Stop when no points assignments change

→ Rootdeki Ci adet hizmetin merkezi olur.

→ Her bir örneğin yerinden hizmet merkezine olan mesafeleri hesaplarız.

• Öklid Mesafesi : $\sqrt{(x_1 - x_{c_i})^2 + (y_1 - y_{c_i})^2}$ distansı

• Manhattan Mesafesi : $|x_1 - x_{c_i}| + |y_1 - y_{c_i}|$

X	Y	Küme Merkezi	
x_1	y_1	c_1	c_1
x_2	y_2	c_2	c_2

→ Merkezlerin değiştiri, güncellendir. → Değişenler ortaklarını yeni Ci deşitlerdir.

→ 2. adım istenilen teknik adıdır. (Küme merkezi değişimeye kadar devam eder)

Iteration sayısı

• Birbirine yakin hizmet merkezi seçilmesi durumunda aynı zamanda herhangi bir sayıda farklılık olabilir.

$O(KN)$ time.

Öm

İlet	Weight	FH index
A	1	1
B	2	1
C	4	3
D	5	4

$$C_1 = (1,1)$$

$$C_2 = (2,1)$$

$$AC_1 = \sqrt{(1-1)^2 + (1-1)^2} = 0$$

A $\rightarrow C_1$

$$AC_2 = \sqrt{(2-1)^2 + (1-1)^2} = 1$$

$$BC_1 = \sqrt{(2-1)^2 + (4-1)^2} = 1$$

B $\rightarrow C_2$

$$BC_2 = \sqrt{(2-2)^2 + (4-1)^2} = 0$$

$$CC_1 = \sqrt{(4-1)^2 + (3-1)^2} =$$

C $\rightarrow C_2$

$$CC_2 = \sqrt{(4-2)^2 + (3-1)^2} =$$

D $\rightarrow C_2$

A	1	1	C_1
B	2	1	C_2
C	4	3	C_1
D	5	4	C_2

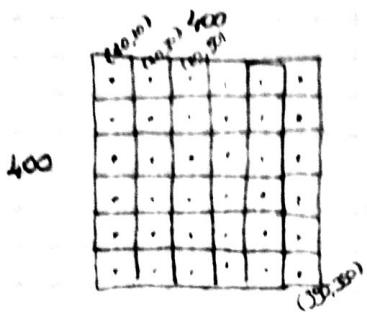
$$C_1_{\text{new}} = 1,1$$

$$C_2_{\text{new}} = \left(\frac{11}{3}, \frac{2}{3} \right)$$

While ($iter < 100$ || $C_{\text{new}} - C_{\text{old}} < 0.001$) 88 ($C_{\text{new}} - C_{\text{old}} < 0.001$)

(superpixel size)

Superpixel



- Erit ettiğimiz bölgelerin $\rightarrow N = \sqrt{\frac{W * H}{K}}$ $= \sqrt{\frac{400 * 400}{400}} = 20 * 20$
- Metrelerine noktalar kayıtları (bölgeye)
- $2N * 2N$ lik parçaları kayıtları.

for $i=0 : i < 400 ; i++$

for $j=0 : j < 400 ; j++$

for $2N \times 2N$ (Area)

L18

$$dc = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}$$

$$ds = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

$$D = \sqrt{dc^2 + \left(\frac{ds}{S}\right)^2} \cdot m^2$$

sabit (kullanıcı seçimi)
Grid(N)

$$S = \sqrt{N/K}$$

$$da = \sqrt{(r_i - r_{i'})^2 + (g_i - g_{i'})^2 + (b_i - b_{i'})^2}$$

$$ds = \sqrt{(x_i - x_{i'})^2 + (y_i - y_{i'})^2}$$

Local pixel değer
dönemi: herkeş geri

Araç: Portaklık değerleri

yakın olan pikselleri $D = dc + m \cdot ds$
bulmak.

distance ne kadar önemli olursa belirtilebilir. (40'a kadar değer verilebilir.)

$m = 0-1$ arası (koordinat çok örenmiş değil)

en yüksek verilisi: karesel pikseller ebe赤etili.

• Diferansiyel şekil benzerliği

• Herşer hatalı ve kullanımı kolay

sekol

Morphological Operators

(H)istorically

- Matematiksel morfolojik��作ler kümeler teorisiyle yapılmıştır.
- Görüntü bilgisayar teknolojisi kullanılmıştır. (Açıklama yapan bulge aranmış). boundary extraction
2D boyutlu image: $I(x,y)$

Binary ↗

Graycode ↗

$$A = \{ (x,y) \mid I_A(x,y) = 1 \}$$

$$\text{Disjoint} \quad A^C = \{ w \mid w \notin A \} : \text{complement}$$

haber ↗

$$\text{difference} \quad A - B = \{ w \mid w \in A, w \notin B \} : \text{difference}$$

$$ACB = \text{subset}$$

$$A \cap B = \text{intersection}$$

kopuklukları gidermek ↗

$$A \cup B = \text{union}$$

D ① Dilation: $A \oplus B = \{ z \mid (B)_z \cap A = \emptyset \}$

1. Reflect. B (yansıtma, uygunluk)

2. Shift by z

3. If it overlaps with A output a 1 at the center of B

0	0	0
0	1	0
0	1	0
1	1	0
1	1	0



0	1	0
1	1	1
0	1	0

Dilation symbol
(OR)

A

B ← structuring element

Sadece bir pixeldeki 1'lerin, kernel uygulanır. (1'den pixel yararlanırsın sen)

Merkezi 1 olan pixel olmaz.

1	0	0
0	0	0
0	0	0



1	0	0
0	0	0
0	0	0

=

1	1	1
1	1	1
1	1	1

Cikis resmine istem yepitildi.

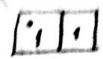
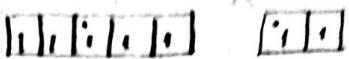
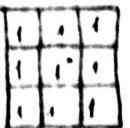
1	0	0
0	0	0
0	0	0
0	0	0
0	0	0

- Kenotoda yelpaze işlem

- Kenotara bolumu

- Sadece ligili kisimlar yelpaze

Yapısal elementler

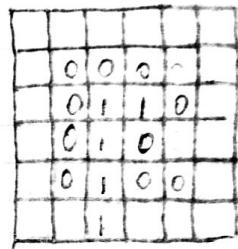
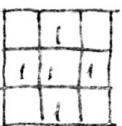
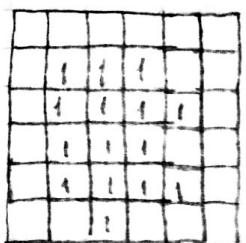


Get fixed storage
needed later.

Problems gone soon, good!

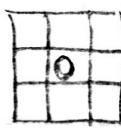
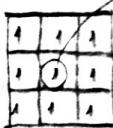
Erosion

$$A \ominus B = \{z | (B)_z \subseteq A\} : \text{shift } B \text{ by } z, \text{ if it is completely inside } A, \text{ output a } 1$$



A

B



1 olasılık sonuc
1 kontrollü

Sonuçta sadece matris 2 bölge.

Opening Erosion özellikleri ve uygulamalar

$$A \circ B = (A \ominus B) \oplus B$$

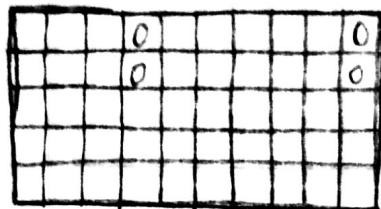
Closing

$$A \bullet B = (A \oplus B) \ominus B$$

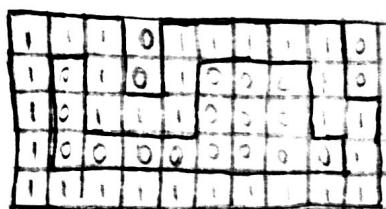
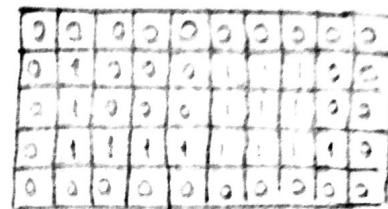
① Boundary Extraction \rightarrow Salin Conservative ad bolgi istenye



$$A - (A \oplus B)$$



1

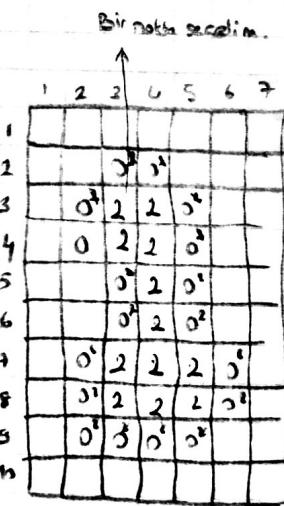
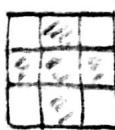
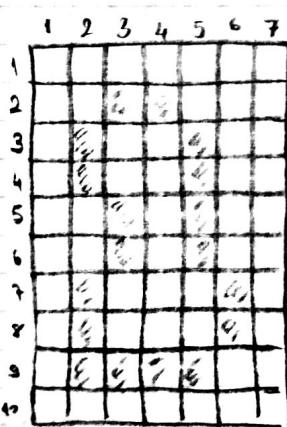


② Region Filling

$$X_0 = P \quad (\text{bit rate } \infty)$$

$$X_k = (X_{k-1} \oplus B) \cap A^c$$

Until $x_k = x_{k-1}$



Beslaga notissia. but not strictly so because

Connected Component Labeling



$$X_0 = P$$

$$X_t = (X_{t-1} \oplus B) \cap A$$

Üstüne $X_{t+1} = X_t$

→ Yarım otomatik tane tek istem yapsılabilece.

Texture Analysis (Doku analizi)

"Texture is a repeating pattern"

Tekstö eden bilgi. Texture cannot be defined for a point

* Doku sıfatı özellikti.

* Bir gecitlilikli dokuların olması segmentasyon yapsılabileceğini göster.

* Aynı boyutlu dokuların benzerliklerini göster.

① Structural → artificial texture

② Statistical → natural texture

Structural Texture

→ Yarım dokularda tane tek istem yapsılabilece. (texel: texture element)

is group of pixels that have similar texel properties - (color, gray level, average)

* Voronoi dögümeleri ile entegrasyon çalışılabilece.

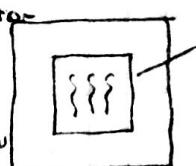
Statistical Texture

* Tekstö bir istatistikçi almış.

* Segmenting art texels is difficult. (Gerçek görüntülerde texelleke aynı olmamaktadır.)

① Edge Density and Director

Bugnus: Edginess per unit area.
(Birim alan başına kırıltı)



testimde herhangi bir bölge

$$\text{Bugnus} = \frac{\text{Topl. Mıknatıslar}}{\text{Topl. pixel}} \times 100$$

$\text{N} \leftarrow \text{topl. bulgu} / \text{pixel sayısı}$

* Doku onlarda ill. obuk konu keşfetmeye yapsılabilece.

- o Sabit bir bulgedeki kırıltıların sayısı.
- o Bulgeye ne kırıltı yapan obukları göster.
- o Kırıltı genetik kontrastı en yüksek olanı olur.

magnitude = boyutlu

* Edge yoğunluğu aynı olabili. Bu yüzden edge yoğunluğu detektörlüde.

$$Fr_{grad,dir} = \frac{1}{2} H_{mag}(R), H_{dir}(R) F$$

histograms of
gradient magnitude
gradient orientation

$$\theta = \arctan \frac{dy}{dx}$$

* Compare normalized histogram of gradient magnitudes and gradient directions, respectively.

A setinde: 6 boyu, 19 aks
12 horizontal, 13 vertical

(pixel sayisi: koltukluk)
boyu aks

B setinde: 6 aks, 6 diagonal

$$H_{mag}(A) = \left(\frac{6}{25}, \frac{19}{25} \right) = (0.24, 0.76)$$

$$H_{mag}(B) = (0.0, 1.0)$$

- Bir resme edge detection uygulayın.
- Onun usulune threshold uygulayın. (Edge olan alanları ayırt edin.) 2 threshold ile aks olan ve boyu obs aksları tercüme edilebilir.
- Aks ve boyu pixelinin histogramını elde edin.

$$H_{dir}(A) = \left(\frac{12}{25}, \frac{13}{25}, 0 \right)$$

$$H_{dir}(B) = (0, 0, 1)$$

$$L_1(H_A, H_B) = \sum_{dr}^n |H_{A,dr}(i) - H_{B,dr}(i)|$$

↓
manzıla distance

$$L_{mag}(H_A, H_B) = \sum_{i=1}^n |H_{A,mag}(i) - H_{B,mag}(i)|$$

bitiş sayısı

Gray Level Co-Currence Matrices

Düzenli gradient istatistikleri

Renklerin birlikte

bulundum概率.

1	1	0	0
1	1	0	0
0	0	2	2
0	0	2	2

1 in boyunda aks düzleme	0	1	2
0	4	0	2
1	2	2	0
2	0	0	2

d(0,1)

Normalized Gray
Level Co-Occurrence
matrix

$$NGLCM = \frac{1}{12}$$

4	0	2
2	2	0
0	0	2

Image $I(x,y) \rightarrow C[i,j]$ $d = (dr, dc)$ $d(0,1)$
 ↓ ↓ ↓ ↓
 gray value i gray value j row column $d(1,0)$



$d(1,1)$



$$\text{Normalized } N[i,j] = \frac{C[i,j]}{\sum_{i,j} C[i,j]}$$

otkere metrix olmeli

\rightarrow 256x256 matris olmali. Ancak quention edildiginda 16x16 matris de isteniyor. Hepsini birlikte kullanmamak isteniyor.

* Dokuya gore sacim istemi yopitmelis.

Oni-space

Fotli doktora bilgileri icin deneyip, hogni deger maximiza ediyorsa o degerdir.

$$x^2(d) = \left(\sum_{i,j} \frac{N_d(i,j)}{N_d(i), N_d(j)} - 1 \right)$$

Entropy Dagiliminin ne kadar random oldugunu gösterir.

$C[i,j]$ 'i birbirine benzette yubek olur.
dusuk olur.

Contrast Doku degismesi gösterice $cdf(2,2)$

\rightarrow displacement matrix nasil secilecegi mukemmel

\rightarrow Fotli yonetimde fotli yonetimler deneyerek hangisinin basit oldugunu bulunucu.

Laws Texture Energy Measures

\rightarrow Doku denilen oey filtrelerin karsilik vermek icin sagdir.

• Gabor, LBP, Gradient, Gassian

Gauss
filter
denilen
olarak

* Doku filtreleri ile resmi filtreleme.

* Bir doku doeliginin her bir pixelde risiklenmesi

* Her bir pixelin gerisindeki (yakindaki)

filtrelerin sonucunun mutlak farkları ile topikal doku enerjisini algımlama.

\rightarrow Birden fazla filtrenin cevapları kullanimi. (3 tane ve)

• lark fotabilitiklerinin problem olmasini engellemek icin

bir window boyutunun genelligi 100px'linin ortalamasina eklin, ve bu window boyutundan 100px'lik bir window boyutu (100px)

\rightarrow 9 tane 3x3 matris, bu 9 tane 3x3 matrisin ortalamasi elde ediliyor.

\rightarrow 9 matrisden sonra 9 gruntu elde ediliyor.

\rightarrow 9 tane 9'lu doğrusal usul her gruntu ortalamaya alınarak (7x7). Sadece öncelikli bilgilerin kalmasi icin yapiliyor.

\rightarrow Konsoloslama istemi icin close edilen feature imgeleri mormatten distince de okular.

Local Binary Pattern (LBP) (Benzetik olamak icin kullanılır.)

Göruntudeki her pixel'in 8 bitlik bir sayı oluşturur. Doku bu dusun 8 bit ile ifade edilir.

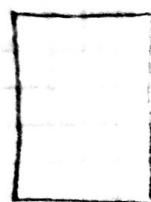
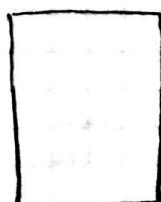
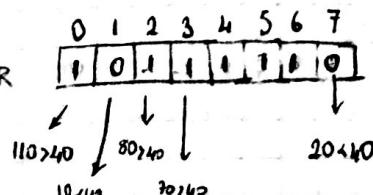
* Yüz tanma

* Yüz itedisi tanma

* Cinsiyet tanma

* Hedef nesne tanma

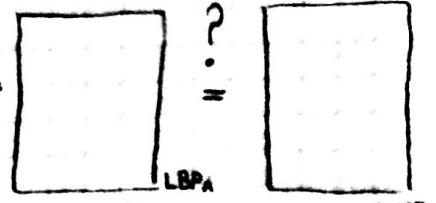
I(x,y)	BR
110	0
10	0
80	1
7	0
40	1
100	1
10	1
20	0



0	1	2
3	P	4
5	6	7

$$b_i = \begin{cases} 1 & I_i(x,y) > I_p(x,y) \\ 0 & I_i(x,y) \text{ otherwise} \end{cases}$$

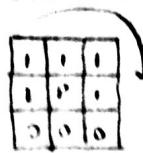
$$dist(A,B) = \sum_{i=0}^{n-1} |Hist(i) - Hist(i)|$$



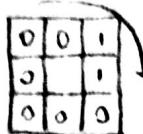
Dimensionality Reduction (Birbirine yakın pixellerin yeri değiştiği durumlarda kullanılır.)

- Tekrar etmeye gelme durumuna göre boyut azaltılığı. Arditik bloklar 5x5 tane şeklinde gösteriliyor. } 5x5 tane boyut azaltılığı ediliyor.
Geri laterlerde de just value ekleyelim.

25x25 tane boyut ile 5x5 tane kullanılır. (8 bit)



0110000
1111000
1111000



001
01
000

Bu şekilde histogram oluşturulur.

• Sıgnalın basırıdır.

- Bu da uygunlukla iyi sonuçlar elde edilebilir.
- Fast : $O(n)$
- Cheap
- Good

* Face Recognition.

- Divide face into blocks, construct LBP histograms.

Eigenfaces

Face Recognition Feature Based
Holistic

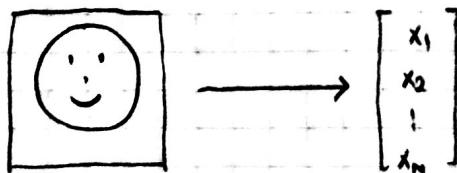
Principal Component Analysis (PCA)

- * Ana bileşenler korelasyon istemini kullanır.
- * PCA orijinalde sonuc yine haneli dönerdir.

Dimensionality reduction

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \xrightarrow{\text{PCA}} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

(feature extraction)



N = vektör
pixel = element.

① Feature selection: Özelliği seçme. Özelliği boyutunuza kısaltın.

② Feature extraction: Eliminatörlü vektörler. Karşıya yarıştırır ve farklı değerler elde edilir.

xxx	xxx	xxx	
xxx	xxx	xxx	
xxx	xxx	xxx	
xxx	xxx	xxx	
A	B	A	B

bad feature
α

good feature
✓

Özellikler ayırt edici olmaz.

Özellik sayısı çok teknik metodlar. → Sınıflar içi benzerlik var, sınıf içi benzerlik çok olmaz.

Özellik sayısı çok ise örnek sayısı da çok olmaz.

• mean: $\bar{x} = \mu = \frac{1}{n} \sum_{i=1}^n x_i$ Gök ayırt edici bir özellik değil. $\rightarrow 3 5 2 6 \xrightarrow{\mu=4}$
 $\rightarrow 4 4 5 3 \xrightarrow{\mu=3}$

• variance: $\text{var}[x] = \overline{x^2} - E[(x-\mu_x)^2] = \frac{1}{n-1} \sum_{i=1}^{n-1} (x_i - \mu_x)^2$ \rightarrow Bir tane data için
 Standard deviation \rightarrow Bir grup için

2 bilgilerin birbirine göre deviation

değilimi

• Co-variance $\text{cov}(x, y) = \frac{1}{n-1} \sum_{i=1}^{n-1} (x_i - \mu_x)(y_i - \mu_y)$ veya $n-1$

$\text{cov}(x, y) > 0$: positively correlated

Gök olsaydı çok yüksek not alırdım. ↑↑

< 0 : negatively correlated

Gök olsaydı çok düşük not alırdım. ↓↓

0: uncorrelated

figizde

$$\text{Co-variance matrix: } \text{cov}(x, y, z) = \begin{bmatrix} \text{cov}(xx) & \text{cov}(xy) & \text{cov}(xz) \\ \text{cov}(yx) & \text{cov}(yy) & \text{cov}(yz) \\ \text{cov}(zx) & \text{cov}(zy) & \text{cov}(zz) \end{bmatrix}$$

Eigenvalue-Eigenvector

A: $n \times n$ matrix

$$A * x = \lambda * x$$

↑↑

eigenvalue eigenvector (vektör)

$$\frac{I}{2x^2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Ax - \lambda x = 0$$

$$x, (A - \lambda I) = 0 \rightarrow \det(A - \lambda I) = 0$$

birim matris

$$\begin{bmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{bmatrix}$$

$$(a_{11} - \lambda) \cdot (a_{22} - \lambda) - a_{12} \cdot a_{21} = 0$$

$$\lambda_1 \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda_1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

↑
eigenvektor

$$\begin{bmatrix} 1-\lambda_1 & 2 \\ 2 & 1-\lambda_1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$\det \begin{bmatrix} 1-\lambda_1 & 2 \\ 2 & 1-\lambda_1 \end{bmatrix} \rightarrow (1-\lambda_1)^2 - 4 = 0$$

$$\lambda_1 = -1 \quad \lambda_2 = 3$$

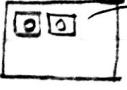
$$\lambda_1 = -1 \rightarrow \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\lambda_2 = 3 \rightarrow \begin{bmatrix} -2 & 2 \\ 2 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

N pixel matrix $\rightarrow M$ adet
(image)

$$\begin{bmatrix} \quad \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

N pixel

① Training:

 forklı. somontrib çekilmiş
 M tane resim vektör.

② Testing:

Training

④ Mean. image: $m = \frac{1}{P} \sum_{i=1}^P x_i$
 P: # of image

② $\bar{x}_i = x_i - m$ $i = 1 \dots n$

mean centered
image

③ $\bar{x} = [\bar{x}_1 \quad \bar{x}_2 \quad \dots \quad \bar{x}_n]_{N \times P}$

↓
1st column is image 1

④ Cov: \bar{x}, \bar{x}^T

$$\textcircled{5} \quad \text{Cov} * V = \lambda V$$

set of eigenvectors set of eigenvalues

Up to P eigenvectors!

\textcircled{6} λ büyüklerde sıralanır. ilk k adet + eigenvalue'ya kosulluk getir eigenvektör'lerdir.

$$V = [V_1 \dots V_k]$$

\uparrow en büyük λ ya kosulluk
getir $\rightarrow k$ maximum ? olabilir.

\textcircled{7} Centered training images are projected into eigenspace

$$\tilde{X}^i = V^T \tilde{x}$$

\downarrow mean centered

\downarrow centered image

Tanımın

bu resimler kütüphanedeler.

Test

mean centered

$$\textcircled{1} \quad \tilde{y} = y - m$$

query
image

$$\textcircled{2} \quad \tilde{y} = V^T * \tilde{y}$$

Eigenface

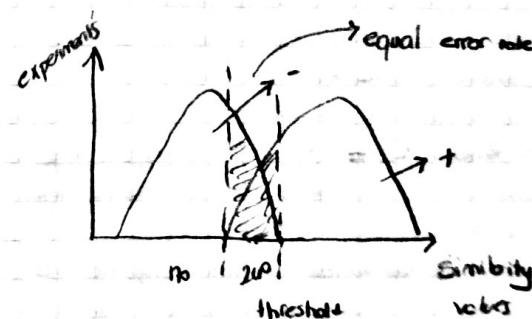
\textcircled{3} Calculate distance to all training images:

Benzetik bir eşit seviyesinden en ise tanrımyorumuz demeliyiz.

$$\text{dist}(\tilde{y}, \tilde{X}^i) = \sqrt{\sum_{j=1}^n (\tilde{y}_j - \tilde{X}^i_j)^2}$$

Euclidean distance

Threshold belirleme



* Yüz tanıma sistemi: yüzlerin istenmesi

- Face detection (kisi yüzleri gibi tespit etmek)

Viola & Jones

- Histogram feature

- Resizing

- Recognition

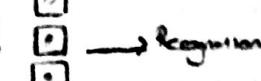
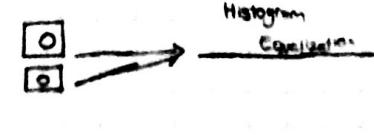
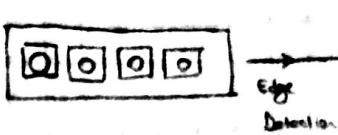
Threshold 240 \Rightarrow dörtün hepsi doğruyadı

çünkü bir hatalı tanımayı yok

Threshold 110 \Rightarrow dörtün hepsi doğuyadı

bir hatalı yanlış etrafında yoğunlaşmaktadır

• Probleme göre değiştirebiliriz



Görseldeki Rank, Dokuz, Sekiz tane bilgileri verdir.

↓

Low level



SHIF

Histogram of oriented gradient

→ Edge histograms olusturulur.

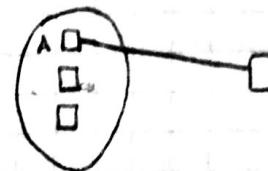
Content base image retrieval

Ortalama benzerligi gecen resim benzerligiyle uyeedir.

↓

Rank histogram

- Python ile yaratılabilir.



A
↓
Rank histogram
↓ LBP histogram

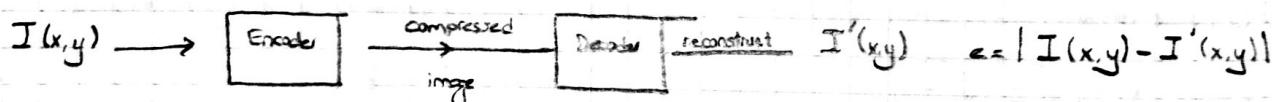
Sigurbank Testim

Mesafe ölçümü: Okul distansı

Görüntü
skorlama

Basic Data Redundancies

- Interpixel Redundancy: Arka olaylardan bağımlılık.
- Psychovisual Redundancy: İnsan gözünün reaksiyonları. İhtiyaçları gözlemlenip, 100 30 82
100 30 83
- Coding Redundancy:
 - Az kullanılan renkler fazla bit.
 - Çok kullanılan renkler çok az bitle ifade ediliyor.Huffmann



- Lossless Compression: ε=0
- Lossy Compression: Quantization

$$\text{Compression Ratio: } CR = \frac{\# \text{ of bits in original image}}{\# \text{ of bits in encoded image}}$$

$$CR = \frac{n_1}{n_2} \quad n_2 = n_1 \rightarrow R_D = 0$$

↓

Redundant bilgi
yat

$n_2 \ll n_1 \rightarrow$ Çok redundan bilgi →

$$R_D = 1 - \frac{1}{CR}$$

Relative
redundancy

1-Intepixel redundancy \rightarrow Kayıporsuz

Run-length encoding (runlength in pixels, color code)

$$C_R = \frac{\# \text{of pixels} * \# \text{of bits}}{(\# \text{of pixels} + \# \text{of color}) \text{ in bits} * \# \text{of runlength codes}}$$

$$(40,3), (20,5) \dots (10,1)$$

40 pixellik 20 pixellik

3 rank 5 rank

6

Bilgi dene or bilgi ile tutulacaktır.

* 2-Coding redundancy \rightarrow Kayıporsuz

Huffman coding

- Rank histogram atları. Hangi renklerin kaç tane tükendiği bilinir.

1. Sort the colors by decreasing probability

2. Add the two smallest probabilities

3. Insert the value into the list

4. Repeat until only two probabilities remain

Aşağıda sol tane 0
Eski tane 1

3-Psythovized redundancy \rightarrow Kayıporsuz

We notice errors in homogeneous regions

Low frequencies (Parazit)

We notice errors in edges

High frequency (Büyük genetikin olduğu yer)

We don't notice noise in textured areas.

Medium frequency.

Lossy image compression

Compression ration 10~50

11k olabilir.

Lossless image compression

Compression ration 3

Uniform Color Quantization

↳ Native Color Quantization

Uniform : Native Color Quantization, Uniform Quantization
Non-uniform

R : 3 bit

G : 3 bit

B : 2 bit

Cok basit bir yapı. Cok büyük okuma yapan renklerde başarısızdır.

Uniform Quantization

1) Farklı boyutlu tek bir blok okunur.

Cok karanlık ve gizemli renklerde başarısızdır.

Düzenli cok olası yatanlar için eklendi.

Nor-uniform
Geliş hatalarını redüksiyon.

Run-length Algoritma

Rengin matrisini oluştur.

- 256 renk \rightarrow 16 renge
 elementlerin
 distribütörlerini
 sayıları.
- Geniş hataları kısaltır ve yine renge换成.
 \rightarrow Az hataları en fazla bir bilgi kaybabilir.

Consept: Bilgi sayısını azaltır.
Aksinca: Renk çeşitliliğinin azaltılır.
Tüm renklerin (Renk genel)
Renklerin dağılımı biraz da şartsız.
Belli seviyelerde de olur.

Median Cut

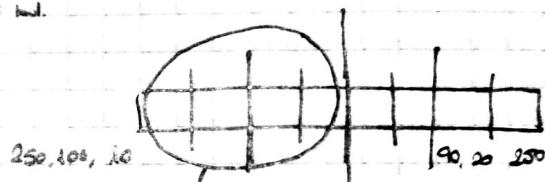
Hiz加快 yeşil rengin iyi sezik sonucunu verir.

RGB koordinatları herhangi bir şekilde sıralanır.

$$R: 10-200 : 190$$

$$G: 40-80 : 60$$

$$B: 10-250 : 210$$



Aynı blokta renk
sayısı azdır.

İşte böyle.

Bu da B' den

İşte bir diğer gibi

yapılabilir.

- * Hergün her blokta renk sayısını en fazla 3'e azaltır.
- * Yerel hataları kısaltır ve herhangi bir blokta sezik sonucunu verir.
- 8x8 blok \rightarrow
 (her blok 64 pikselde 3 renk)
- * Renkler arasında kitle变换.
- * Aşağıdaki recursive stock sayısının her blokta 3'tür.
- * Aşağıda istenilen renk sayıları her pikselde 3'tür.
- * Her pikselin içindeki renklerin oraneleri nesne ile ilgili nerede - dir.

(enkapsülasyon)

(kötüye girmek, ... hafiflem)

Image \Rightarrow Transform \Rightarrow Quantization \Rightarrow Symbol Encoding \Rightarrow Compressed Image

General Steps in Image Compression

Discrete

Image \Rightarrow DCT \Rightarrow Quantization \Rightarrow Huffman \Rightarrow JPEG
Transform Encoding Image

Steps in JPEG Image Compression

Transform RGB to YIQ/UVVuv.

Discrete Cosine Transform (DCT)

JPEG Image Compression

- Renk wayı değiştirilir.

- RGB luminans bilgisi ve renk bilgisi birlikte dir. Luminans ve renk bilgisi ayırt edilen yarımnesi olmaz.

Transformation: Farklı ortamda redundan bilgi birleştirmeye catatır.

- ① The image is broken into 8×8 blocks of pixels.
- ② From left to right, top to bottom \downarrow apply DCT to each block.

64 katayı elde edilir. \downarrow deysimin obugu yor lade yuzek deysiler elde edilir.

- Original resimden 128 olur.

-128×127

- ③ Quantization 1 ile 100 arasında bir sayi belirler.

Q: Quantization matrix

$$C_{ij} = \frac{D_{ij}}{Q_{ij}} \rightarrow \text{DCT}$$

Q_{ij} → Quantization matrix

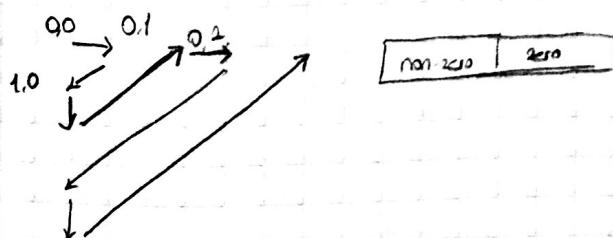
Q_{80}

Degisim olan yeter sayi koltur. Yuksek frekanslı yeterdir koltur.

Degisim olmayan yeterdir.

* Yuksek genisleme edildi.

- ④ zig-zag : Encode quantized coefficient with sequence

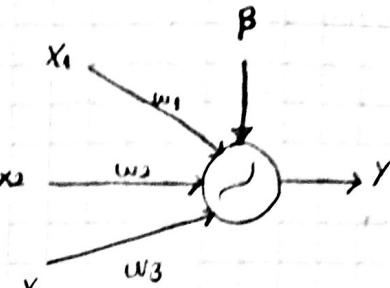


- ⑤ Run-length encoding uygulanır. (0'lar at adegide ligi iken kontroller)

- ⑥ Apply Huffman coding

Keywords

- Convolutional Neural Network
- GPU-Cuda Programming → Geoffrey Hinton
- Python OpenCV → Alex K.
- Dersler → Yann LeCun
- CS231n → Stanford 82x82 girdi



- ImageNet
- TensorFlow : Matris doğrulama ortak olduğu faktura bir araya getirilen tensor teknisi.
- pdf : A. Efros

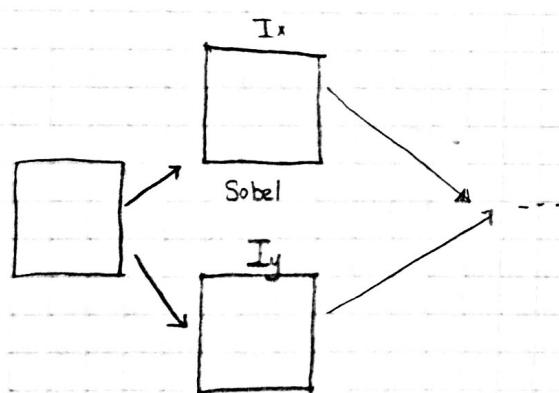
Operations

- * Rotation
- * Translation
- * Cropping

Makro - Resim : yumusatırıcı emmeyle kollarılır.

* Veri tabanındaki resimlerin ortakları olan ve birebir resimlerden farklılıkların ortakları olsalar. Bu işlemi buna göre etiketleme. Sharpening

Original - smooth = detail { Data keskinlerinde edilsin.
original + detail = new



GLCM dokumaları

Hog, LBP, ...

* Her algoritmanın öz karakteristigini açıklar.

Low level

Mid Level feature ekler. Resim ile ilgili bilgi edilebilir.

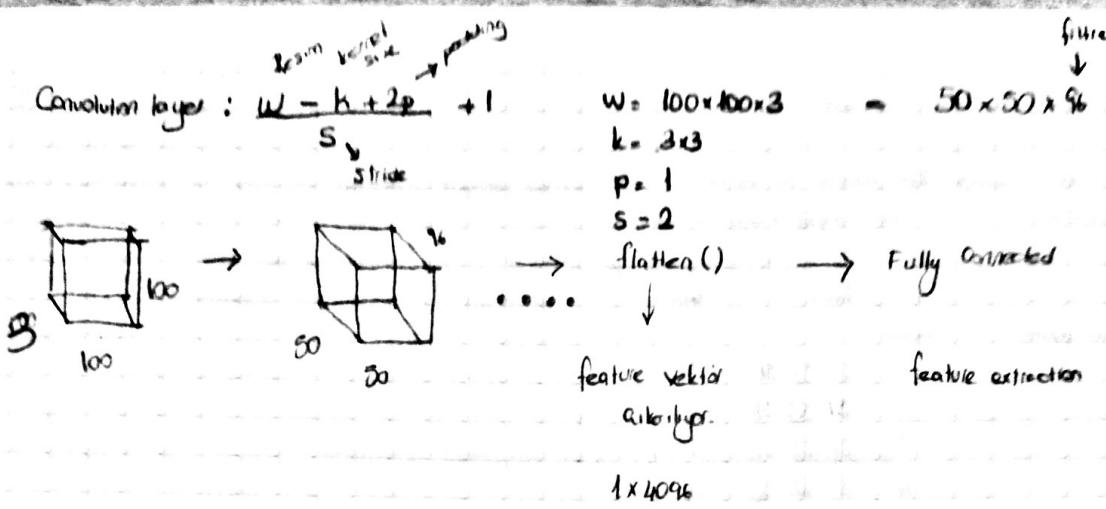
High level

Convolution

ReLU

Max Pooling

Kernel filter padding
100x100x3 3x3 46 → 96 tane resim olur.



Pre-trained Networks

- AlexNet
- GoogLeNet
- LeNet
- VGG16, VGG19, RCNN, Mask-RCNN, VGG-VD-m, VGG-VD-f, VGG-VD-s, etc
- SegNet, FCN, UNet (RCNN gibi teknikler)

Object Detection

- * Matlab
- * Keras - TensorFlow (backward)
- * Keras - Tensorflow (backward)

CNN eğitilecek Imagenet

- Veri kümeleri dergesi Basketball jeu
- Validation elle seçildi.

Performance Evaluation

Veri setinin

- Bir kümeli de test bir kümeli de train edilebilir.
- De test etmek için o test etmek için basıncı sağlanır.

Hastane yöntemleri

2/3 train

1/3 test iyi gürültü.

Stratified sample → A2 olan data → hastane data setinde calısırken hastaların örnekləri.

1 hast 2/3 1/3

2 sağlam 2/3 1/3
train test

Random subsampling

Birkaç farklı şəkildə yepitilir.

Test ve train tətbiq etməq qədəmələr.

O tətbiq etmək dərman istəniləndə.

• Eldeki vəzi ziş train ve test olab yepitilir.

shuffle edilər yepitilir.

$$F = \frac{1}{k} \cdot \sum_{i=1}^k \epsilon_i$$

Cross validation (k-fold cross validation)

- Data set olumsuz da tüm data hem train hem test kullanılır.

Örnek: leave one out \rightarrow her örnekte bir testset

Cross validation test olgularını oluşturur.

generation set ea

train

test

ve bunu leave one out yapılı.

1	S1 S2 S3 S4	S1
2	S1 S2 S3 S4	S2
3	S1 S2 S3 S4	S3
4	S1 S2 S3 S4	S4

Örnek 2: 10-fold cross validation

100 örnek varsa 90 test set

10 test set için kullandır.

özetleme varsa da kullandır.



Iteration 1



Iteration 2

Stratified cross validation: Testleri terteola.

Bootstrap: N train örneği rastgele seç.

Bu örneklere iade edilebilir bir örnektir birden fazla kez de tekrar.

Data seti Train ve validation türkleri test etmeli.

Ideal durumda

train	validation	test
-------	------------	------

↓

parameter

tuning

Sistem için farklı yöntemler denenir. \rightarrow en iyi basarı için seçilir.

Seçilen yöntem için farklı parametreler ayıklanır, en iyi sonucun parametreleri seçilir.

Data sonunda test ederek validationa göre basarı ölçülür.

Data eger her sey bittiken sonra tüm örneklere test, validation, train özellikleri sisteme teker teker verileceğidir.

Performance Evaluation

Confusion matrix

Predicted class

Predicted class		
Yes	No	
Yes	TP	FN
No	FP	TN

maximize diagonal

TP = True positive

FN = False negative

FP = False positive

TN = True negative

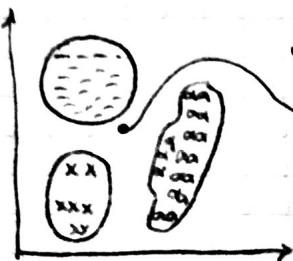
Test Nasıl Yapılır?

Sorunlar olusturulur. Once en problematik olusumlar testler.

Örnek: Sınıfla yuvarlak, once sau okyanus, adasına bir insanın yuvarlanmeye çalışması.

Daha sonra diğerlerine genel olumlu ihan ettilerler.

→ last problem



overfitting: Edebi, veri seti için çok fazla öğrenme. (cezberlenme)

Farklı bir data test için kullanıldığından önceki bulutlu ve bulutluysa, öğrenmeyecektir. Sistem başarısız olacak.

Chancı olur. Genel karakteristiklerin yedeklenebilmesinden

Basarı Nasıl Ölçürtür?

Accuracy: is a ratio of correctly predicted observations to the total observations

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

TP + TN

1000 → 990 sağlam
10 hasta

→ sistem herkeste doğru olduğunu

Acc: $\frac{990}{1000} \rightarrow$ basarılı değil. Veri uniform olmadığının tespiti gibi düşüyor.
1000

Precision: Doğru bildiklerinin kacı olduğunu

$$Precision = \frac{TP}{TP + FP}$$

Recall (Sensitivity): Ratio of correctly predicted positive observations to the all observations in actual class

$$Recall = \frac{TP}{TP + FN} \rightarrow hasta olanların kaçının hasta dedi.$$

F1 score: Precision ve Recall ortalaması

$$F1 score: 2 * (Recall * Precision) / (Recall + Precision)$$