

Histogram Eşitleme ve Histogram Benzetme

Emre Can Kuran

İçindekiler

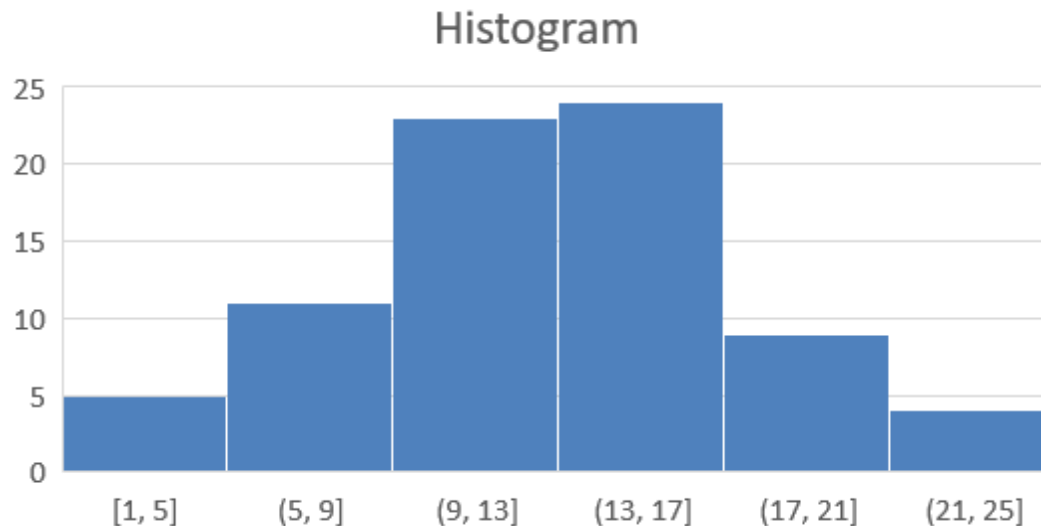
- İstatistik: Histogram, PDF, CDF, TF
- Görüntü işleme: Histogram, PDF, CDF, TF
- Histogram Eşitleme (Histogram Equalization)
- Histogram Benzetme(Histogram Matching/Specification)
- Renkli Görüntülerde Histogram Eşitleme ve Histogram Benzetme

İstatistik

- İstatistiksel işlemler birçok alanda olduğu gibi görüntü işlemede de kullanılır.
- İstatistik, bir olayın karakteristik özelliklerini analiz etmemizi sağlar
- Görüntünün özelliklerini tespit etmek, görüntüyü işlemek açısından önemlidir
 - Bu sebeple görüntü işlemede de istatistikten faydalanılır.

Histogram

- Histogram, bir verinin dağılımını gösterir. Genelde ilgili kategori x ekseninde, karşılık gelen sayılar y ekseninde verilir. Kısacası her bir gruptan ya da tek varlıktan ne kadar olduğu gösterilir.



PDF

- Probability density function veya probability mass function (olasılık yoğunluk fonksiyonu), bir verinin belli bir değeri % kaç olasılıkla alacağını belirler.
- Örneğin bir zarın 1,2,3,4,5 veya 6 olan yüzlerinden birinin gelme olasılığı yaklaşık %16.67 ($1/6$ - toplam 6 olay) olarak hesaplanabilir
- PDF bize, «kesin bir değer için ne kadar olasılık olabilir?» sorusunun yanıtı verir. Mesela 1 gelmesinin olasılığı %16.67'dir. 2 gelmesinin olasılığı da %16.67' dir

CDF

- Cumulative distribution function(kümülatif dağılım fonksiyonu), bir verinin belli bir değere eşit ya da küçük bir değeri % kaç olasılıkla alacağını belirler.
- Örneğin bir zarın 1,2,3,4,5 veya 6 olan yüzlerinden birinin gelme olasılığı yaklaşık %16.67 ($1/6$ - toplam 6 olay) olarak hesaplanabilir
- CDF bize, «belli bir değere kadar ne kadar olasılık olabilir?» sorusunun yanıtı verir. Mesela 1 gelmesinin kümülatif olasılığı %16.67'dir. 2 gelmesinin kümülatif olasılığı $16.67 + 16.67 = \%33.34$. 3 gelmesinin kümülatif olasılığı %49.68 olarak verilebilir.

TF

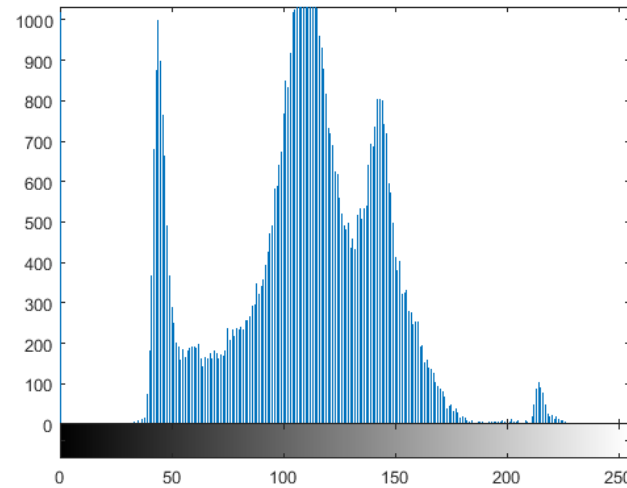
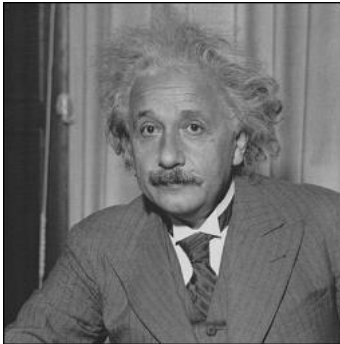
- Transformation function (dönüşüm fonksiyonu) bir veri kümesindeki her bir noktaya matematiksel bir dönüşümü uygular.
- Bu dönüşüm; artırma, azaltma vb. dönüşümler olabilir.

Görüntü

- Görüntü, iki boyutlu bir fonksiyondur. Her $I(x,y)$ koordinatı tek bir değere işaret eder. Görüntüdeki her bir nokta bir renk/ton değerine işaret eder.
- Görüntüdeki bir nokta 1-bit, 2-bit, 3-bit, 4-bit, 8bit, 16-bit, 24- bit ile temsil edilebilir.
- Bit sayısı (ya da derinliği) arttıkça görüntüdeki bir nokta daha çok değer alabilir. Böylece daha fazla renk/ton kullanılabilir.
- Bit sayısına göre noktanın kaç değer alabileceği (kaç farklı renk tonu olduğu- gri ya da renkli) n bit sayısı olmak üzere, 2^n ile belirlenir.
 - $2^3 = 8$ seviye
 - $2^8 = 256$ seviye
 - $2^{16} = 65536$ seviye
 - Böylece görüntüdeki bir noktanın alabileceği en küçük değer 0, en büyük değer ise $(2^n - 1)$ olarak belirlenir.
- RGB görüntünün, gri görüntüden farkı, her bir kanalının n -bit ile kodlanmış olmasıdır. Aslında, R bir gri görüntü, G bir gri görüntü ve B başka bir gri görüntü olarak düşünülebilir. Her biri ilgili rengin şiddetini gösterir. Örneğin gri görüntü tek kanallı 8-bit iken RGB görüntü her kanalı 8-bit ile kodlanmış 3 kanallı görüntü içerir. Bu 3 kanalın her biri parlaklık matrisidir.
- OpenCV görüntü kanallarının kodlanmasında, genelde RGB değil BGR sırasını takip eder. Yani görüntüye ait olan 0. kanal mavi, 1.kanal yeşil ve 3.kanal kırmızıdır.

Görüntü İşleme: Histogram

- Histogram istatistikte olduğu gibi görüntü işlemede de kullanılır. Her bir seviyeye «gri seviye» ya da «bin» denir.
- Bir görüntüde histogram, her bir renk tonundan kaç adet bulunduğunu gösterir. Tüm seviyelerdeki bu adetlerin toplamı aynı zamanda toplam piksel sayısına eşittir.
- Örneğin: 8 bitlik bir gri görüntü 0-255 arası değer alır (256) seviye. 8 bit gri bir görüntü için histogram örneği aşağıda verilmiştir.

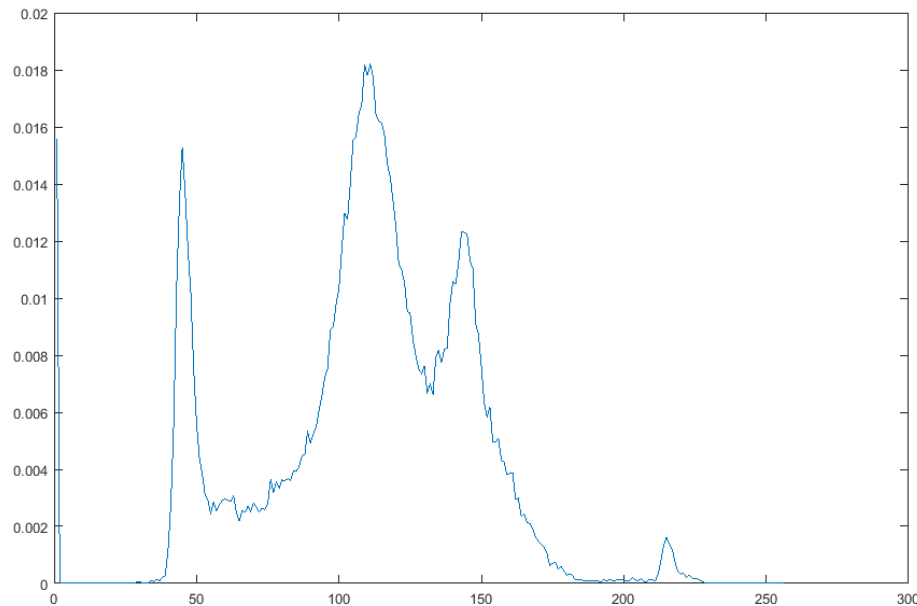


$$H(i) = \text{say } (I(x, y) == i),$$
$$i = 0, 1, \dots, L - 1, \quad \text{görüntü } I \text{ için}$$

Görüntü İşleme: PDF

- Histogramdaki her bir gri seviyenin histogramdaki değerini toplam piksel sayısına bölersek PDF hesaplamış oluruz.

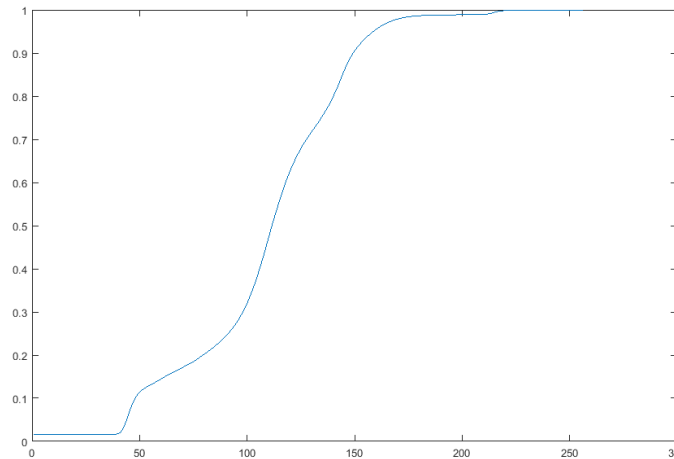
$$PDF(i) = (hist(i) / M * N), \quad i = 0, 1, \dots, L - 1,$$



Görüntü İşleme: CDF

- Histogramdaki her bir gri seviyenin ilgili gri seviyeye kadar olan toplam (kümülatif) PDF değerini hesaplırsak CDF değerini buluruz.
- Kural gereği, son seviye her zaman 1' e eşittir (Tüm olasılıklar toplamı %100 olması gerektiğinden). Daha önceki seviyeler de 1' e eşit olabilir, eğer o seviyelerin değeri 0 ise.

$$CDF(i) = \sum_{j=0}^i PDF(j), \quad j = 0, 1, \dots, L-1$$
$$i = 0, 1, \dots, L-1$$



Görüntü İşleme: TF

- CDF ile hesapladığımız değerleri çıktı görüntüye uygulayabilmemiz için, aşağıdaki gibi bir eşitlik kullanırız:

$$Y(I(x, y) == i) = CDF(i) \times (L_{maks.} - L_{min.}) + L_{min.}$$

$$i = 0, 1, \dots, L - 1$$

- Y çıktı görüntü, X girdi görüntü, $L_{maks.}$ Gri seviyenin alabileceği en yüksek değer (255) ve $L_{min.}$ Gri seviyenin alabileceği en düşük değerdir (0). Dolayısıyla:

$$Y(I(x, y) == i) = CDF(i) \times 255$$

$$i = 0, 1, \dots, L - 1$$

- Olur. Yani çıktı görüntünün gri seviye limitlerini aşması engellenir. Çünkü $CDF(i)$ 'nin alabileceği en yüksek değer 1'dir.

Histogram Eşitleme-Mantığı ve Kullanımı

- Histogram eşitleme, diğer frekans alanı uygulamalarına göre daha kolaydır- çünkü uzaysal alanda direk görüntü pikselleri üzerinde işlem yapar.
- Uygulaması oldukça basit ve hızlı olan bir algoritmadır.
- Histogram eşitleme, CDF kullandığından çok yüksek olasılığa sahip olan parlaklık değerlerini daha geniş alana yayar.
- Histogram eşitleme kontrastı ya da parlaklığı iyileştirmeyi amaçlar. Tüm seviyelerin mümkün olduğunca eşit olarak dağılımını hedefler.

Histogram Eşitleme-Mantığı ve Kullanımı

- Parlaklık: Göreceli bir kavramdır, görüntünün parlak ya da karanlık olduğunu ifade eder.
- Kontrast: Bir görüntüdeki maksimum parlaklık değerinin minimum parlaklık değerinden çıkartılması sonucu elde edilir. 256 seviyeli bir gri görüntüde, maksimum değer 150 minimum değer ise 50 ise kontrast $150 - 50 = 100$ olarak hesaplanabilir. Genelde kontrast arttıkça detaylar artar, çünkü aralık genişler. Bu da daha çok ton kullanıldığı anlamına gelebilir.

Histogram Eşitleme - Algoritması

- Histogram eşitleme, aşağıdaki adımlardan oluşur:
 1. Görüntünün histogramını hesapla
 2. Histogramdan PDF hesapla
 3. PDF' den CDF hesapla
 4. TF' yi oluşturulan, CDF ile çıktı görüntüye uygula

Histogram Eşitleme - Pseudocode

```
//I girdi görüntü, M ve N görüntünün x eksenini ve y eksenini boyutları, L maksimum seviye sayısı
compute_hist(I, M, N, L){
    Hist = array(1, L); //histogram için 1 x 256' lık bir dizi
    for i=0;i<M
        for j=0;j<N
            hist[(i, j)]++; //histogram hesapla
        end for
    end for
    return hist;
}

//hist histogram, M ve N görüntünün x eksenini ve y eksenini boyutları, L maksimum seviye sayısı
compute_pdf(hist, M, N, L){
    pdf = array(1, L);
    for i=0;i<L
        pdf[i] = hist[i] / (M*N); //pdf için 1 x 256' lık bir dizi
    end for
    return pdf;
}

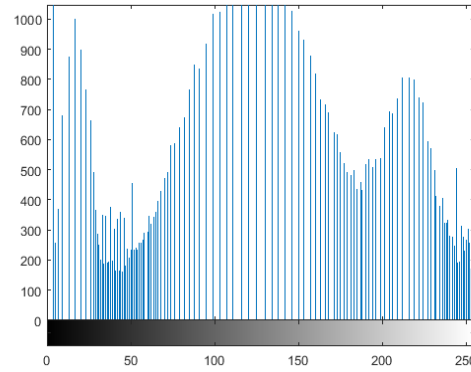
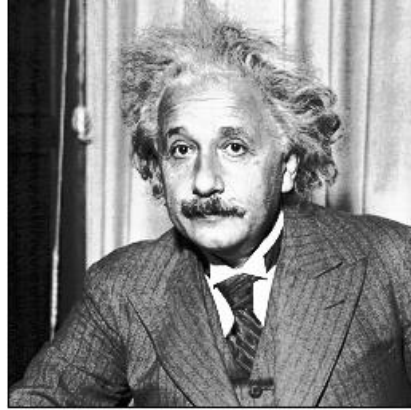
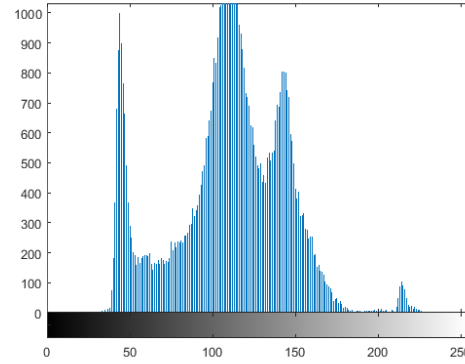
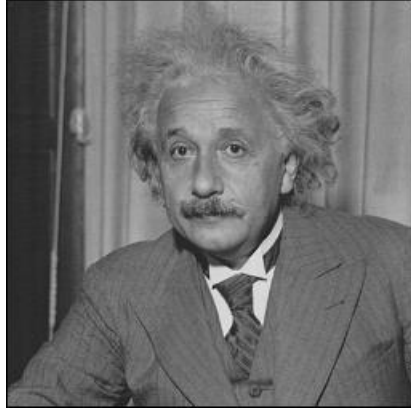
//pdf ol. yoğun. fonk., M ve N görüntünün x eksenini ve y eksenini boyutları, L maksimum seviye sayısı
compute_cdf(pdf, M, N, L){
    cdf = array(1, L);
    sum_pdf = 0;
    for i=0;i<L
        sum_pdf += pdf[i];
        cdf[i] = sum_pdf; //pdf için 1 x 256' lık bir matris
    end for
    return cdf;
}

//cdf küm dağı. fonk., M ve N görüntünün x eksenini ve y eksenini boyutları, L maksimum seviye sayısı
compute_tf(cdf, M, N, L){
    tf = array(1, L); //tf için 1 x 256' lık bir matris
    for i=0;i<L
        tf[i] =cdf[i] *(L-1); //pdf için 1 x 256' lık bir matris
    end for
    return tf;
}

//I görüntü, tf dönüşüm fonk., M ve N görüntünün x eksenini ve y eksenini boyutları, L maksimum seviye sayısı
apply_tf(I, tf, M, N, L){
    out= array(M, N); //çıktı görüntü için M x N matris
    for i=0;i<M
        for j=0;j<N
            out[(i, j)] = tf[(i, j)]; //dönüşüm hesapla
        end for
    end for
    return out;
}

//I girdi görüntü, M ve N görüntünün x eksenini ve y eksenini boyutları, L maksimum seviye sayısı
h_eqv(I, M, N, L){
    hist = array(1, L), pdf = array(1, L), cdf = array(1, L), tf = array(1, L), out= array(M, N);
    hist = compute_hist(I,M,N,L);
    pdf = compute_pdf(hist,M,N,L);
    cdf = compute_cdf(pdf,M,N,L);
    tf = compute_tf(cdf,M,N,L);
    out = apply_tf(I,tf,M,N,L);
    return out;
}
```

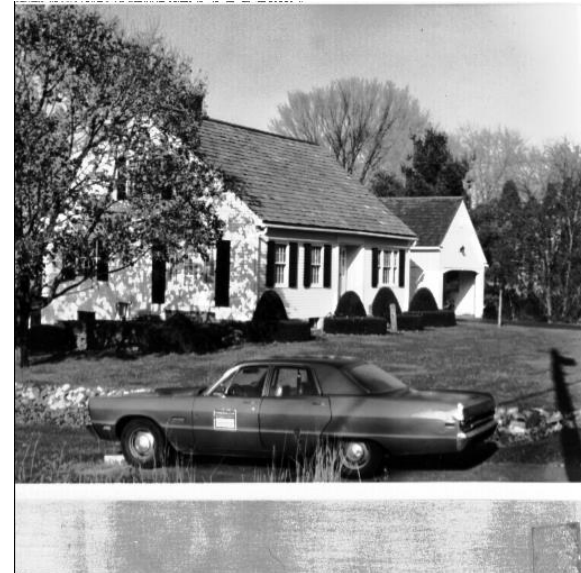
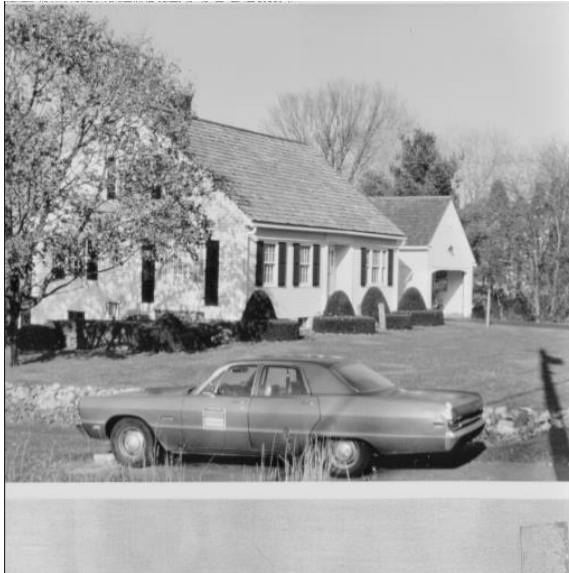

Histogram Eşitleme-Mantığı ve Kullanımı



- Yukarıdaki şekillerde görüldüğü gibi aralık genişlemiş ve görüntü iyileştirilmiştir.

Histogram Eşitleme-Dezavantajları

- Detay kaybına ve bozulmalara sebep olabilmektedir.
- Düşük bit derinliğine sahip görüntüler için tercih edilmesi daha çok bozulmaya sebep olabilir.
- Daha gelişmiş teknikler mevcuttur.



Histogram Eşleştirme (Histogram Matching)

- Histogram eşitlemeye göre avantajları vardır.
- Bir görüntünün parlaklık dağılımını referans alınan başka bir görüntüye benzetmek için kullanılır.
- Örneğin, eldeki bir görüntünün parlaklığını direk olarak diğer görüntüye benzetmek istersek kullanabiliriz. Bu sebeple histogram eşitlemede olduğu gibi görüntü bozulmaya uğramaktan kurtarılabilir.

Histogram Eşleştirme (Histogram Matching) Mantığı

- İki CDF hesapladığımızı kabul edelim. HM ile çıktı görüntünün CDF' sini referans görüntünün CDF' sine TAM OLARAK benzetmeye çalışırız.

$$CDF_1(G_1) = CDF_2(G_2)$$

- Fakat bu her zaman mümkün olmayabilir, çünkü CDF dağılımları farklılık gösterebilir. Bu sebeple yaklaşık şu şekilde hesaplanabilir:

$$TF(G_1) = \min |CDF_1(G_1) - CDF_2(G_2)|$$

- Yani, transformasyon fonksiyonu girdi görüntü ile çıktı görüntü arasındaki gri seviyelerden minimum CDF farkına sahip olan seviyeyi baz alır.

Histogram Eşleştirme (Histogram Matching) Mantığı

- Histogram eşleştirme, aşağıdaki adımlardan oluşur:
 1. Girdi ve referans görüntünün histogramını hesapla
 2. Girdi ve referans görüntünün PDF' lerini hesapla
 3. Girdi ve referans görüntünün CDF' lerini hesapla
 4. Girdi görüntünün TF' sini hesapla
 5. Girdi görüntüye, referans görüntüden hesaplanan TF'yi uygula

Histogram Eşleştirme (Histogram Matching) - Pseudocode

//I girdi görüntü, ref referans görüntü, M ve N girdi görüntünün x eksen ve y eksen boyutları, W ve H M ve N referans görüntünün x eksen ve y eksen boyutları, L maksimum seviye sayısı

h_match(I, ref, M, N, W, H L){

hist1 = array(1, L), hist2 = array(1, L);

pdf1 = array(1, L), pdf2 = array(1, L);

cdf1 = array(1, L), cdf2 = array(1, L);

hist1 = compute_hist(I,M,N,L); hist2 = compute_hist(ref,W,H,L);

pdf1 = compute_pdf(hist1,M,N,L); pdf2 = compute_pdf(hist2,W,H,L);

cdf1 = compute_cdf(pdf1,M,N,L); cdf2 = compute_cdf(pdf2,W,H,L);

tf = array(1, L);

out= array(M, N);

for i=0:i<L

min_dif = 1.2;

ind = 0;

for j=0:j<L

cur_dif = **abs**(cdf1[i] - cdf2[j]);

if(cur_dif < min_dif)

min_dif = cur_dif;

ind = j;

end for

tf(i) = ind;

end for

out= array(M, N);

//çıktı görüntü için M x N matris

for i=0:i<M

for j=0:j<N

out[i, j] = tf[I(i, j)];

//dönüşüm hesapla

end for

end for

return out;

}

RGB (Renkli) Görüntülerde Histogram Eşitleme

- Histogram eşitleme gri görüntüler için tasarlandığından, bu yöntemi RGB görüntülere uygulamak için farklı teknikler kullanılabilir. Gri görüntü tek kanallıdır (parlaklık kanalı-luminance channel/matrix), RGB görüntü ise kırmızı-yeşil-mavi olmak üzere üç kanaldan oluşur.
- İlk teknik: RGB görüntünün her bir kanalına histogram eşitleme uygula ve eşitlenmiş kanalları birleştir.
 - Fakat şekilde görüldüğü gibi renk bozulmasına sebep olur.



RGB (Renkli) Görüntülerde Histogram Eşitleme

- İkinci teknik: Daha çok tercih edilmektedir, görüntüyü RGB renk uzayından HSV, YCrCb gibi farklı bir renk uzayına çevir ve parlaklık matrisi üzerinde işlem yap. Daha sonra parlaklık matrisinin histogramını eşitle ve V uzayını tekrardan bu matrise ata. En son ise görüntüyü HSV uzayından RGB uzayına geri çevir. Daha uygun sonuçlar vermektedir.
 - Şekilde görüldüğü gibi sadece görüntünün parlaklığı iyileştirilmiştir.

