**Histogram Eşitleme (Histogram Equalization)**

Photo by [Giorgio Tomassetti](https://unsplash.com/@gtomassetti?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

Görüntü ön işleme süreci aslında diğer makine öğrenmesi alanlarında olduğu gibi en uzun süren ve en zahmetli kısımdır.

 Kısaca bahsetmek gerekirse ,

* **Verileri normalize etme ,**
* **Belli bir domaine kaydırma,**
* **Hatalıları ayıklama,**
* **Aykırıları (Outlier) ayıklama,**
* **Boyut düşürme,**
* **Sonuca en fazla etki eden kısımları seçme,**

ve daha onlarcası…

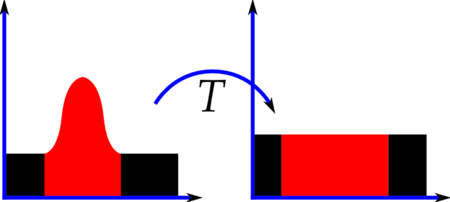
Görüntü işlemede bu ön işleme operasyonlarına ek olarak ,

* **Boyut sabitleme ,**
* **Histogram eşitleme ,**
* **Rastgelelik katma,**
* **Boyut indirgeme ,**
* **Kanal belirleme (3 Kanal , Siyah-Beyaz Format vs),**
* **Veri çoğaltma** gibi operasyonlar gerekmektedir.

Bu bölümde de saydıklarım arasında da geçen **histogram eşitlemesi** konusunu inceleyeceğiz.

**İlk Olarak Görüntüde Histogram Nedir ?**

Histogram görüntüdeki piksel değerlerinin gruplandırarak dağılımının sütun grafiğiyle gösterimidir.Görüntünün histogramına bakılarak o görüntünün parlaklık, kontrast, yoğunluk dağılımı gibi bilgilerini elde etmek mümkün.

(https://tr.wikipedia.org/wiki/Histogram )

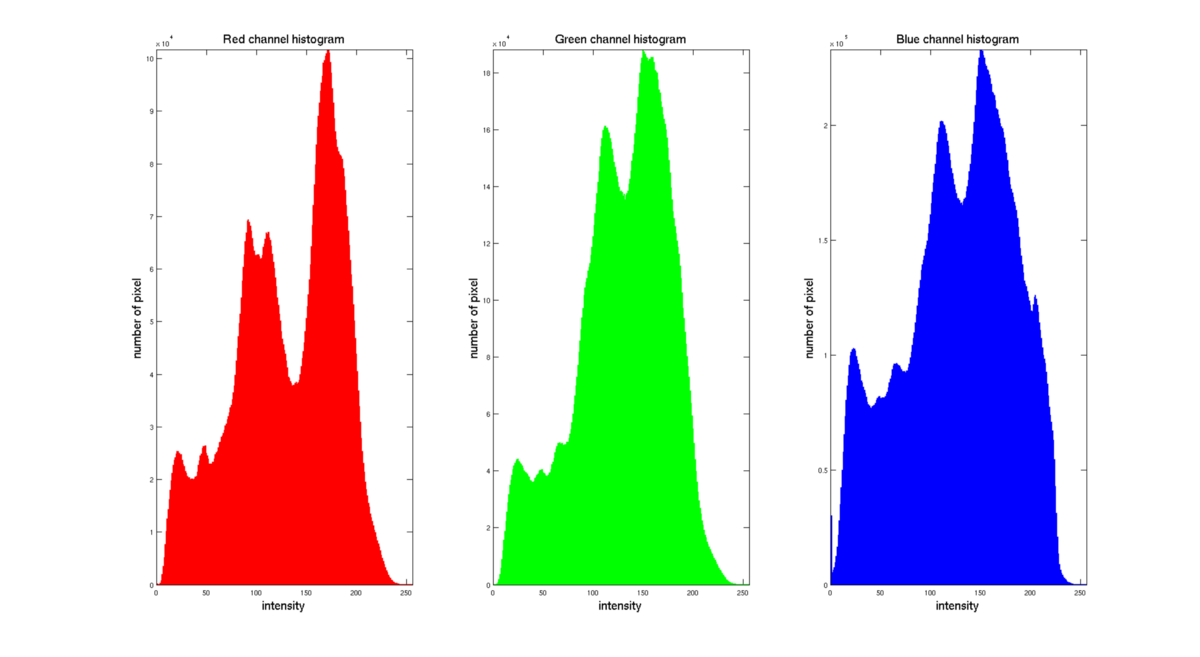
**Görüntü ile olan ilgisinde derine inersek,**

Görüntüde var olan pixel değerlerini, ki bu normalize edilmemiş ise 0 -255, normalize edilmiş ise 0–1 arasında oluyor genelde.

Örnek olarak bir pixel i ele alalım ,

Siyah-Beyaz görüntüde siyah renk için değeri 0 , beyaz renk için pixel değerinin 255 olması lazım. Arasındaki değerlerde (0 ile 255 arasında) siyah ile beyaz arasındaki renk tonunu veriyor.

RGB yani 3 boyutlu vede renkli görüntüler için görüntü 3 tane kanala sahip ve kanalların hepsi yine aynı şekilde 0 ile 255 arası sayılar ile dolu .

<http://www.sci.utah.edu/~acoste/uou/Image/project1/Arthur_COSTE_Project_1_report.html>

Histogramda bu sayıları guruplandırma görevini üstleniyor. Yani

**0 değerine sahip pixel toplamı → 0'lar**

**1’ e değerine sahip pixel toplamı → 1'ler**

**..**

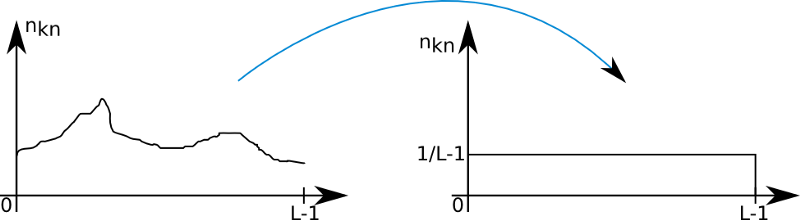
**..**

**255’ e değerine sahip pixel toplamı → 255'ler**

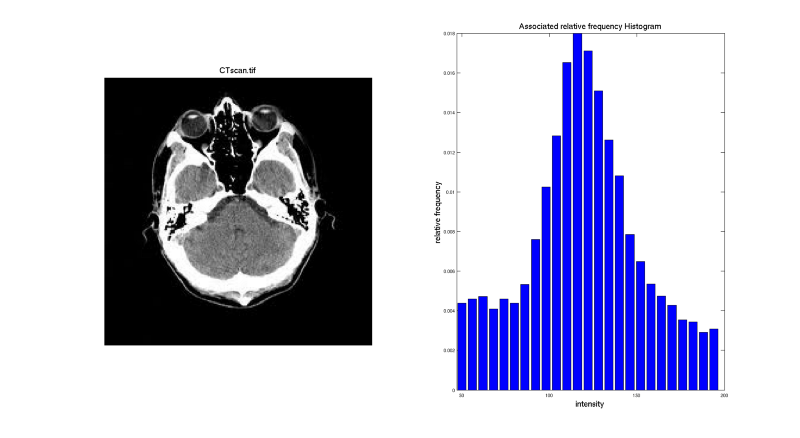
**Peki bizim guruplandırmamızdaki amacımız ne ?**

Kontrastı eşitlemek.Çünkü görüntünün **çok aydınlık** yada **çok karanlık** olması istediğimiz bir durum değil.Hedefimiz bir resimdeki renk değerlerinin belli bir yerde kümelenmiş olmasından kaynaklanan renk dağılımı bozukluğunu gidermek. Çeşitli ayrıntıları kaçırmamız çok olası bu sebeble belli bir tarafta yoğunlamaşyı 0 ile 255 arası tüm aralıklara yaymak görüntünün iyileşmesi açısından çok önemli .Bu operasyona da **histogram eşitlemesi** diyoruz.

**Burda ayrıca güzel bir görüntü nedir onun tanımını yapmak lazım .Güzel bir görüntü eşit miktarda koyu ve açık renk tonların olması olarak geçiyor. (Yani her pixelden eşit sayıda olması)**

<http://www.sci.utah.edu/~acoste/uou/Image/project1/Arthur_COSTE_Project_1_report.html>

Gördüğüm başka kaynaklarda da normal dağılımın daha iyi olacağına dair görüşlerde mevcut .Burda da ortanca değelerin çok olması ve çok aydınlık ve çok karanlık olan renklerin ortanca değere göre daha az olması .

<http://www.sci.utah.edu/~acoste/uou/Image/project1/Arthur_COSTE_Project_1_report.html>

Tabi burda görselinizin hangi ile daha iyi olduğuna sizin karar vermeniz gerekecek ki karmaşık durumlara kapı açan bir yorum aynı zamanda.

**Ne demek istiyorum ?**

Örneğin görüntüdeki bir obje siyah renkli görünüyorsa akla şu soru gelebilir. Gerçekten siyah mı yoksa görüntü alınırken çevresel etmenlerden dolayı mı siyah çıktı?

Şayet gerçekten siyah bir obje ise histogram eşitlemesi uygulamak bu objenin rengini açmak anlamına geleceği için dikkatli olmakta fayda var.Burasıda bizi histogram türlerine getiriyor.

Histogram eşitlemesinde genelleştirirsek 2 ye ayırabiliriz. **Eski** ve **modern** şeklinde .

Eski yöntem tüm resmin alınıp kontrastın eşitlenmesine dayanıyordu.

Faydalı olduğu görüntüler olmakla birlikte bozduğu görüntülerde oldukça çok .Ayrıca görüntünün yumuşamasına sebeb oluyordu sanki görüntü de alçak filtre uygulanmış gibi.

Bu yöntemde histogramı çıkarılan görüntüde ki değerleri eşitleyebilmek adına yuvarlama teknikleri kullanılıyor ve görüntü bir aralıktan çıkıp tüm aralık boyunca dağıtılıyor. Örnek olarak 55–100 arasında iken 0 ile 255 arasına çekiliyor.

Fakat histogram eşitlemede tüm görüntünün yoğunluk dağılımı kullanıldığından bu bazı görüntülerde ortalama yoğunluk orta seviyeye getirildiğinde solmuş etkiye sebep olabilir.

Ve dar bir bölgede kalabalık bir yoğunluk dağılımına sahip görüntülerde ise bir çokgürültü pikselinin oluşmasına neden olabilir.

Modern yöntem diyebileceğimiz **CLAHE** ise adaptif bir eşitleme tekniği uyguluyor.

Kernel diyeceğimiz küçük 2 boyutlu diziler ile görüntü üzerinde gezinerek bölgesel eşitleme uyguluyor.

Bu sayede görüntüdeki objelerin renkleri birbirlerini etkilememiş oluyor.

**Görüntü Okumada İlk Yapılması Gerekenler**

Görüntüyü alırken **LAB** image formatında almakta fayda var. Sonrasında ayrıştırarak **l, a ve b** ‘ye dönüştürüyoruz.

LAB görüntüsünün l’si ışık yoğunluğunu (lightness ) temsil etmekte ve diğer a ve b kanalları ile de renkler oluşturuluyor.

Temelde parlaklık bizim bu bölümde uğraştığımız parametre olduğuna göre renk değerlerin aradan çıkararsak hem renkleri korumuş oluruz hemde parlaklığı dengeli bir seviyede tutmuş oluruz.

RGB formatında okunursa görüntüler her renk tonu için ayrı ayrı histogram eşitlemesi yapmak gerekecek.

**Bu bölümde kodu ile birlikte 4 farklı sistemi inceleyeceğiz.**

* Birincisinde uzun bir şekilde ve basamak basamak ne yapıldığı ile ilgili olacak .
* 2,3 ,4. kısımlarda bu basamaklar çeşitli modüller tarafından birleştirildiği için tek sayırlık kod parçaları ile göstereilcektir.
* En sonunda da karşılaştırılıp en iyi olan belirtilecektir.

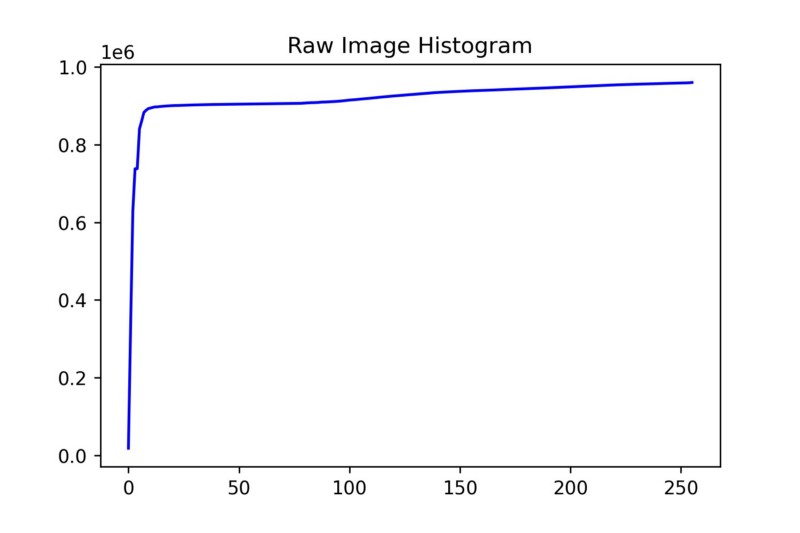
**Hadi kodlamaya geçelim,**

İlk başta görüntü l, a be b parçalarına ayrılıyor.

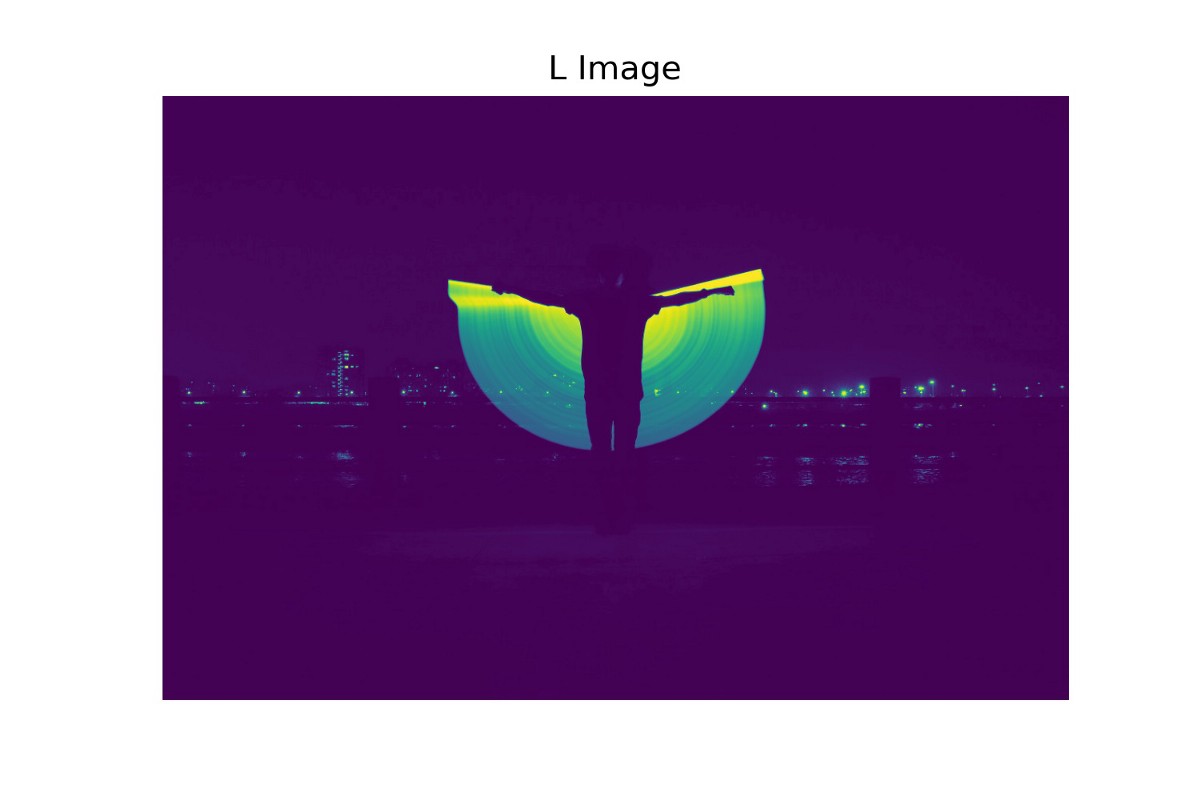
<https://gist.github.com/mntalha/f35efa1c1772437dceef23e318698ea7>

Ayrılmış olan görüntünün düzeltme öncesi görüntüsü şu şekilde,

Photo by [Universal Kalakar](https://unsplash.com/@universalkalakar?utm_source=medium&utm_medium=referral) on [Unsplash](https://unsplash.com?utm_source=medium&utm_medium=referral)

Histogram Eşitleme öncesi kümülatif gösterim

Görüldüğü gibi kümülatif olarak gösterilen grafikte piksellerin nerdeyse tamamının düşük değere sahip olduğu yani siyah ağırlıklı bir görüntü olduğu söylenebilir.Ayrıca belli bir aralıkta tüm resmin yoğunlaşması bizim histogram eşitlemesi uygulamasını tavsiye etmekte.

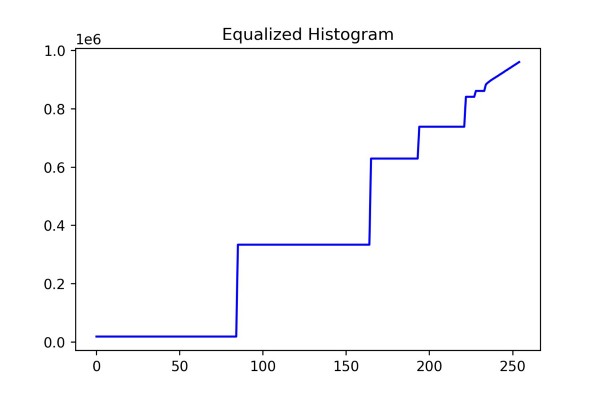
L Görüntüsü

Yukarıdaki resim LAB görüntünün aktarılması sonucu L kısmının ayrışması sonucu olucan görüntü.Çok parlak olan noktalar resimde yüksek değerli pikselleri yani beyaz rengi işaret etmekte.

**1-Geleneksel Yöntem (Uzun)**

<https://gist.github.com/mntalha/e90ffb4146fd437a8fa085f47495fe40>Eşitlenmiş L görüntüsü

Geleneksel yöntem kullanıldıktan sonra parlak renkler tüm görüntüye dağıtıldı.

Eşitlenmiş kümülatif histogram — Eşitlenmiş Görüntü

Histogram’ı eşitledikten sonra yukarı da görüleceği gibi 0 ile 255 arasında eşit bir şekilde görüntü yayılmış . Bunun sonucunda sağda oluşan resim aslında konunun başında bahsettiğimiz soluklaşma problemini yansıtmış.

**2-Geleneksel Yöntem (OpenCv ile)**

<https://gist.github.com/mntalha/1c76d6623da6599c4970f5f9bdc116fd>

EqualizeHist ile Eşitlenmiş Görüntü

Aşağı yukarı geleneksel yöntemlerin hızlı bir uygulaması ama sonuç nerdeyse aynı.

**3-Adaptive Eşitleme Yöntemi (Skimage ile)**

<https://gist.github.com/mntalha/36bc680295c8b974a1a6a9d6c21722de>

Adaptif Eşitlenmiş Görüntü

**3-Kontrast Genişletme Yöntemi**

<https://gist.github.com/mntalha/7339ddb71a5c6d3dd4ab71cc810f8816>

Kontrast Genişletme Yöntemi ile Eşitlenmiş Görüntü

**4-CLAHE Yöntemi (OpenCv ile )**

<https://gist.github.com/mntalha/75a4174c46f0905c5999ba9e90bc4033>

CLAHE Görüntüsü

CLAHE ile görüldüğü gibi görüntüyü soluklaştırmadan en güzel eşitlemeyi sağladık .İlk resme dönüp bakarsanız korkuluklar nerdeyse belli değil Clahe görüntüsünde ise oldukça belirgin.

Dezavantaj kısmında ise kernel boyutunun ve de kesme limitinin bir çeşit hiperparametre olması yani sizler birazcık deneyerek en uygun resmi seçmeye çalışmalısınız.

**Yazının sonuna geldiğimizde , CLAHE’nin büyük bir farkla diğer yöntemlere karşı önde olduğunu görüyoruz.Eşitlerken aynı zamanda lokal keskinliği de koruması tercih edilesi bir uygulama haline getiriyor kendisini.**

Kodların toplu ve en son halini github sayfamda paylaşıyorum.Bakmak isterseniz aşağıdaki linkten ulaşabilirsiniz.

[**GitHub - mntalha/Histogram\_Equalization: show the best way among the given technics by…**  
*show the best way among the given technics by investigating the comparisons. - GitHub - mntalha/Histogram\_Equalization…*github.com](https://github.com/mntalha/Histogram_Equalization)

Bu konuyla ilgili herhangi bir sorunuz varsa, linked-in yoluyla bana kolayca ulaşabilirsiniz

[**Muhammed Nur Talha KILIÇ - İstanbul Teknik Üniversitesi - Istanbul, Istanbul, Turkey | LinkedIn**  
*View Muhammed Nur Talha KILIÇ's profile on LinkedIn, the world's largest professional community. Muhammed Nur Talha has…*www.linkedin.com](https://www.linkedin.com/in/mntalhakilic/)