



KARADENİZ TEKNİK ÜNİVERSİTESİ
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

RAPOR

Sayısal İşaret İşleme İlk Ödev

Ad-Soyad	<u>Uğurcan YILMAZ</u>
Numara	383242
Ders Sorumlusu	<u>Doç.Dr. Sedat GÖRMÜŞ</u>

İÇİNDEKİLER

- 1- [Gri seviye karesel resmin histogramını bulma işlemi](#)
- 2- [Bulunan histogram değerlerinden PMF hesaplama işlemi](#)
- 3- [Resmin piksellerinden ortalama yoğunluk değeri, varyans ve standart sapma hesaplama işlemi](#)
- 4- [Resmin normal dağılım grafiğini hesaplama işlemi](#)
- 5- [Normal dağılım grafiği ve PMF grafiği neden farklıdır?](#)
- 6- [Resmin dağılımını normal dağılıma nasıl yakınsatabiliriz?](#)
- 7- [Program Hakkında Bilgiler](#)

Gri seviye karesel resmin histogramını bulma işlemi

Gri seviye karesel bir resmin histogramını bulabilmek için tüm pikseller üzerinde gezinmeli ve hangi gri renk tonundan kaç tane olduğunu bir dizi içerisinde saklamalıyız. Oluşan dizi histogramı içerecektir.

Bu projede resim formatı olarak C Programlama Dili'nde, üzerinde daha kolay işlem yapılabilirdiği için “.pgm” formatını tercih ettim.

pgm dosyasının başlık kısmı aşağıda görünen şekildedir. 3 satırdan oluşur. pgm dosyasının türü, resmin boyutu ve maksimum gri seviye bilgilerini içerir.

```
image.pgm
1 P5
2 256 256
3 255
```

pgm dosyasının başlık bilgilerini okumak için aşağıdaki kod bloğunu yazdım.

```
// PGM dosyasının başlığını oku
int genislik, uzunluk, maksDeger;
char baslik[3];
fscanf(fp, "%s\n%d %d\n%d\n", baslik, &genislik, &uzunluk, &maksDeger);
```

Piksellerin seviyelerini okuma işlemi için aşağıdaki kod bloğu gerekiyor.

```
// Piksel verisini oku
fread(pikseller, sizeof(unsigned char), genislik * uzunluk, fp);
```

Okunan piksel seviye değerleri ile histogram hesaplama işlemi

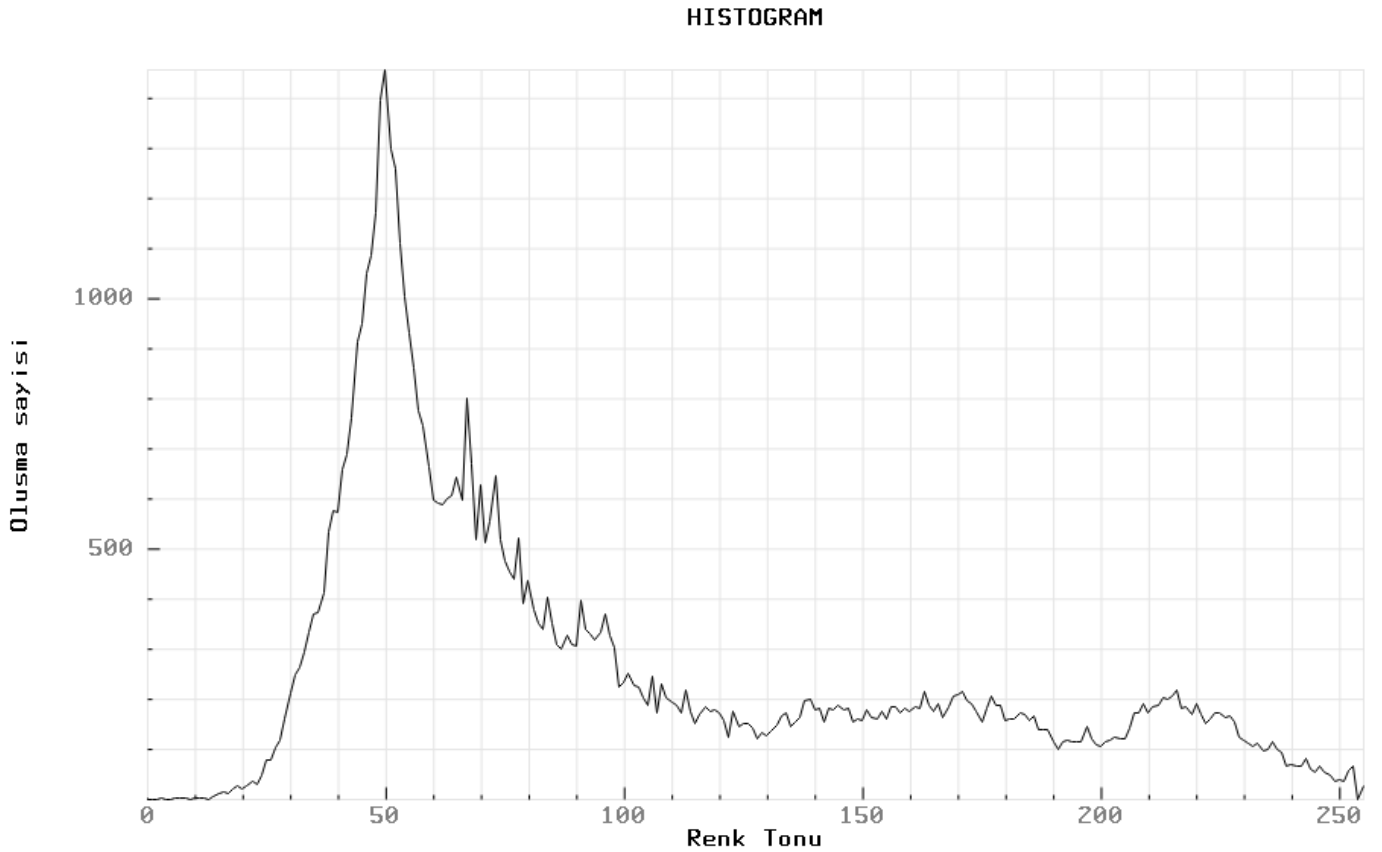
```
// Resmin histogramını hesapla
int histogram[maksDeger + 1];
for (int i = 0; i <= maksDeger; i++)
{
    histogram[i] = 0;
}
for (int i = 0; i < genislik * uzunluk; i++)
{
    histogram[pikseller[i]]++;
}
```

Örnek olarak aşağıdaki resmin histogramını bulalım.



(256x256)

Oluşturulan histogram grafiği:



Not: Grafik oluşturmak için harici bir kütüphane olan “pbPlots” kütüphanesini kullandım.

Bulunan histogram değerlerinden PMF hesaplama işlemi

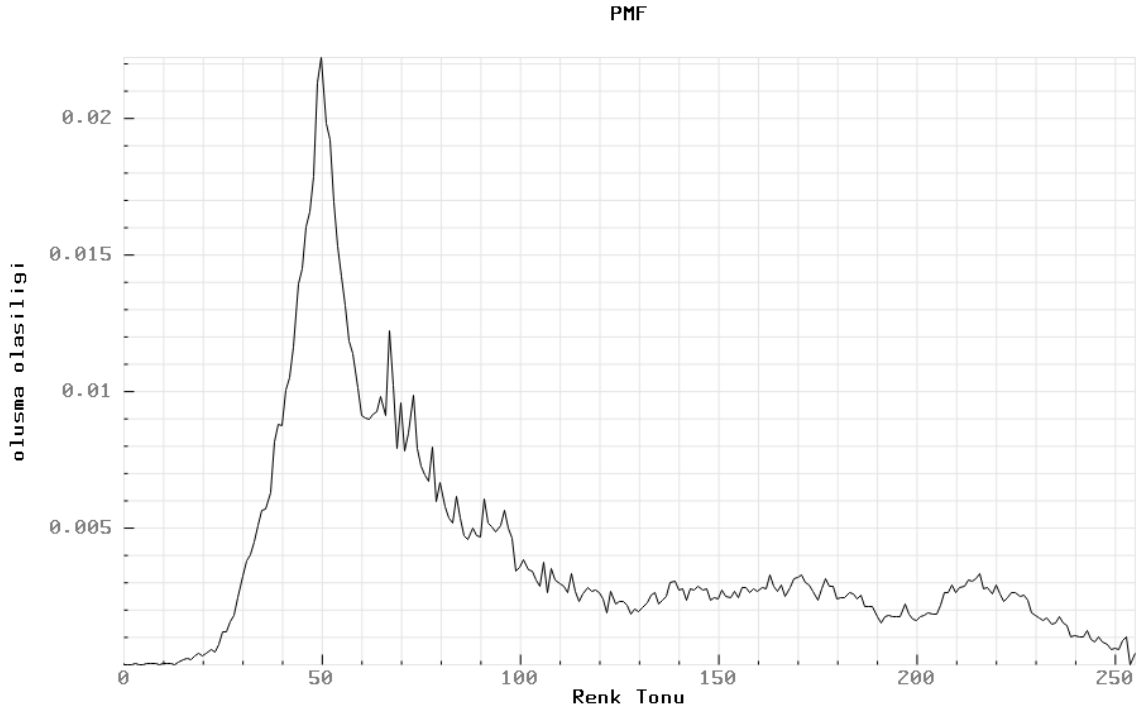
Histogram değerlerinden PMF hesaplayabilmek için histogram dizisi içerisindeki her değeri resmin toplam piksel sayısına bölmek gerekiyor. Bulunan olasılık değeri o renk tonunun oluşma olasılığını temsil edecektir.

Aşağıdaki kod bloğu ile PMF hesaplama işlemi yapılmaktadır.

```
// PMF hesapla
printf("-----PMF-----\n");
double pmf[maksDeger + 1];
for (int i = 0; i <= maksDeger; i++)
{
    pmf[i] = (double)histogram[i] / (double)(genislik * uzunluk);
    if (pmf[i] == 0)
        continue;

    printf("%d: %f\n", i, pmf[i]);
}
```

Örnek olarak öncesinde bulduğumuz histogram değerlerinden PMF grafiğini oluşturalım.



Resmin piksellerinden ortalama yoğunluk değeri, varyans ve standart sapma hesaplama işlemi

İlk olarak ortalama yoğunluk değerini hesaplamamız gerekiyor çünkü varyans hesaplamak için ortalama değerine ihtiyacımız var ve aynı şekilde standart sapma hesaplamak için de varyans değerine ihtiyacımız var.

Ortalama yoğunluk değeri aşağıdaki formül ile hesaplanır.

EQUATION 2-1

Calculation of a signal's mean. The signal is contained in x_0 through x_{N-1} , i is an index that runs through these values, and μ is the mean.

$$\mu = \frac{1}{N} \sum_{i=0}^{N-1} x_i$$

Bu formülün C kodundaki karşılığını aşağıdaki şekilde yaptım.

```
// Ortalama yoğunluk değerini hesapla
double ortalama = 0.0;
for (int i = 0; i <= maksDeger; i++)
{
    ortalama += (double)i * (double)histogram[i];
}
ortalama /= (double)(genislik * uzunluk);
```

Varyans ve standart sapma değerlerini hesaplamak için aşağıdaki formülü uygulamamız gerekiyor.

EQUATION 2-2

Calculation of the standard deviation of a signal. The signal is stored in x_i , μ is the mean found from Eq. 2-1, N is the number of samples, and σ is the standard deviation.

$$\sigma^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \mu)^2$$

```
// varyans hesapla
double varyans = 0.0;
for (int i = 0; i <= maksDeger; i++)
{
    varyans += pow((double)histogram[i] - ortalama, 2.0);
}
varyans /= (double)(genislik * uzunluk);

// Standart sapma hesapla
double standartSapma = sqrt(varyans);
```

Yukarıda grafiğini verdiğimiz gri-seviye resmin ortalama, varyans ve standart sapma değerleri aşağıdaki gibidir.

```
Ortalama Yogunluk Degeri: 99.090652  
Varyans: 363.467577  
Standart Sapma: 19.064826
```

Resmin normal dağılım grafiğini hesaplama işlemi

Normal dağılımın formülü aşağıdaki şekildedir.

EQUATION 2-8

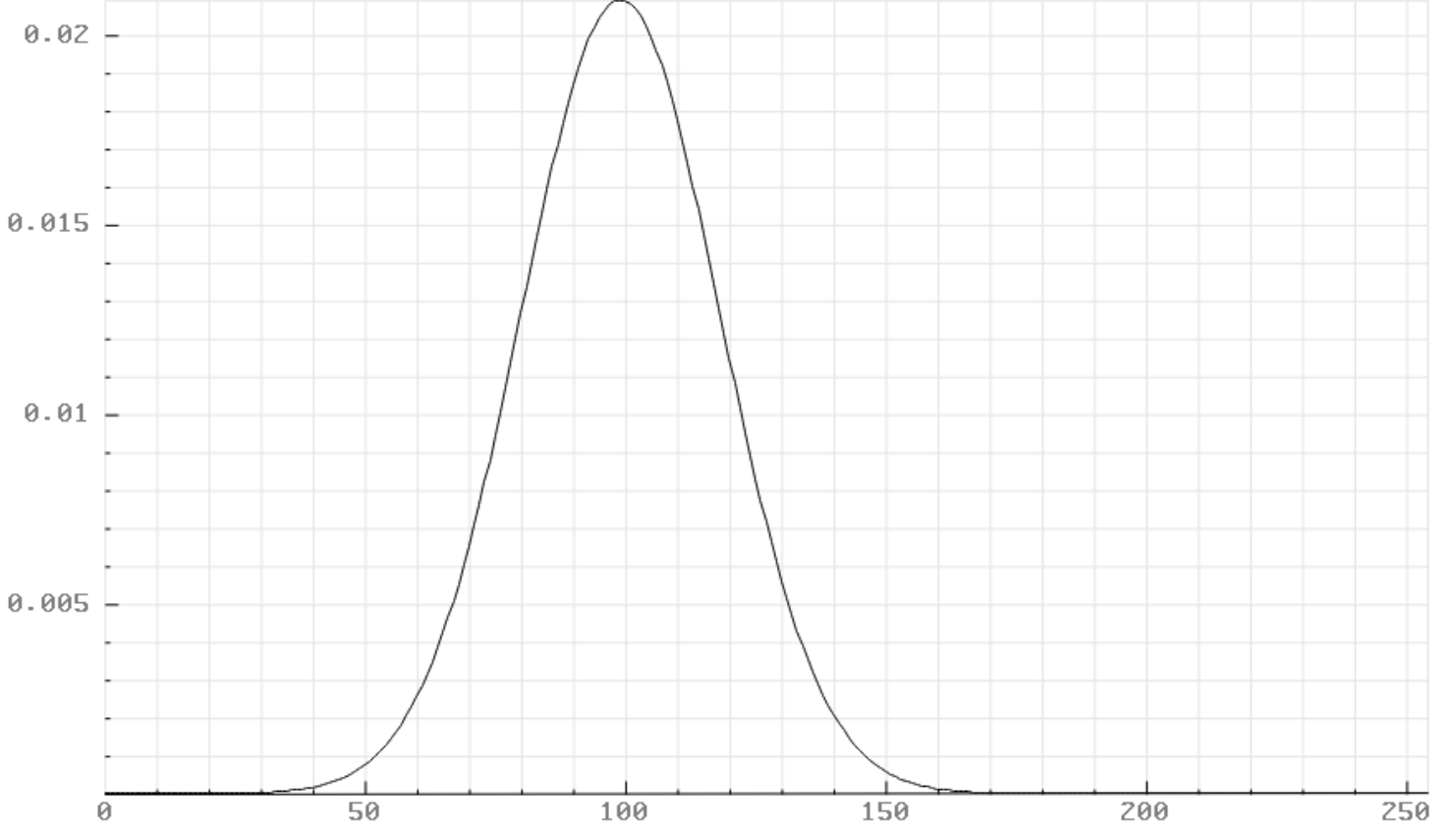
Equation for the *normal distribution*, also called the *Gauss distribution*, or simply a *Gaussian*. In this relation, $P(x)$ is the probability distribution function, μ is the mean, and σ is the standard deviation.

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$$

Programda bu formülü aşağıdaki gibi uyguladım.

```
// Normal dağılım hesapla  
double normalDagilim[256];  
for (int i = 0; i < 256; i++) {  
    normalDagilim[i] = (1.0 / (standartSapma * sqrt(2 * M_PI))) *  
exp(-pow((i - ortalama), 2) / (2 * pow(standartSapma, 2)));  
}
```

Ortaya çıkan dizinin grafiğine baktığımızda aşağıdaki gibi bir şekil oluştuğunu görebiliyoruz.



Normal dağılım grafiği ve PMF grafiği neden farklıdır?

Normal dağılım grafiğinin şekline baktığımızda PMF grafiği ile benzemediğini gözlemleyebiliyoruz. Bu farklılığın nedeni resmimizdeki örneklem büyüklüğünün yeterli seviyede olmamasıdır.

Örnek resmimizde örneklem büyüklüğü arttıkça resmin dağılımının normal dağılıma benzediği görülecektir.

Resmin dağılımını normal dağılıma nasıl yakınsatabiliriz?

Merkezi limit teoreminden yararlanarak bu işlemi gerçekleştirebiliriz.

Merkezi limit teoremine göre bir dağılımın normal dağılıma benzemesi için aşağıdaki şartları karşılaması gerekir:

- 1- Örneklem boyutunun yeterince büyük olması gerekir. Bu koşul genellikle, örneklem büyüklüğü $n \geq 30$ ise karşılanır.
- 2- Örnekler bağımsız ve aynı şekilde dağıtılmış (i.i.d.) rastgele değişkenler olması gerekir. Örnekleme rastgele ise bu koşul genellikle karşılanır.
- 3- Popülasyon dağılımının sonlu varyansı olması gerekir. Merkezi limit teoremi, sonsuz varyansa sahip dağılımlar için geçerli değildir.

Resmin dağılımını normal dağılıma benzetebilmemiz için resmin PMF değerlerinden rastgele 10 tane alıp bu 10 değerın ortalamasını almak gerektiğine karar verdim. Bu şekilde 256 defa ortalama değeri bulunduğunda bu ortalama değeriğlerinin grafiğı normal dağılım grafiğıne çok benzer bir grafik oluşturacaktır.

Kod yazma kısmında bu yapmayı amaçladığım işlemi gerçekleştiremedim.

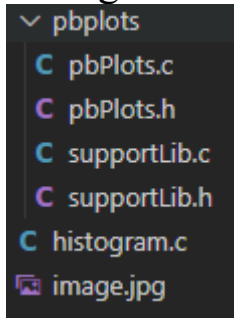
Program Hakkında Bilgiler

Gerekli kütüphaneler ve yazılımlar:

- 1- [pbPlots](#) (C dilinde grafik çizimleri için)
- 2- [ImageMagick](#) (.pgm dosyalarını düzgünce görüntüleyebilmek için ve png, jpg gibi formatlardan pgm formatına dönüşüm sağlamak için)
- 3- [Netpbm](#) (ImageMagick için gerekli.)

Programın kullanımı:

- 1- Dosyalar aşağıdaki gibi olmalıdır. Üzerinde test yapmak istediğiniz resminiz aynı konumda olmalıdır.



- 2- Programda pgm formatı kullandığım için jpg dosyasını pgm dosyasına aşağıdaki komut ile çevirmemiz gerekiyor.

```
convert image.jpg image.pgm
```

- 3- C kodunu derleme işlemi için aşağıdaki şekilde komut satırından gcc derleyicisini argüman vermek gerekiyor.

```
gcc histogram.c pbplots/pbPlots.c pbplots/supportLib.c -o histogram.exe
```

- 4- Daha sonra aşağıdaki şekilde exe dosyasına resmi argüman şeklinde belirterek çalıştırmak gerekiyor.

```
histogram image.pgm
```

5- Exe dosyası çalıştıktan sonra terminalde Histogram değerleri ve PMF değerleri basılacaktır. Ayrıca ortalama, varyans ve standart sapma değerleri de basılacaktır. Bunun dışında dosya konumuna histogram grafiği (histogram.png), PMF grafiği (pmf.png) ve normal dağılım grafiği (normalDagilim.png) eklenecektir.

