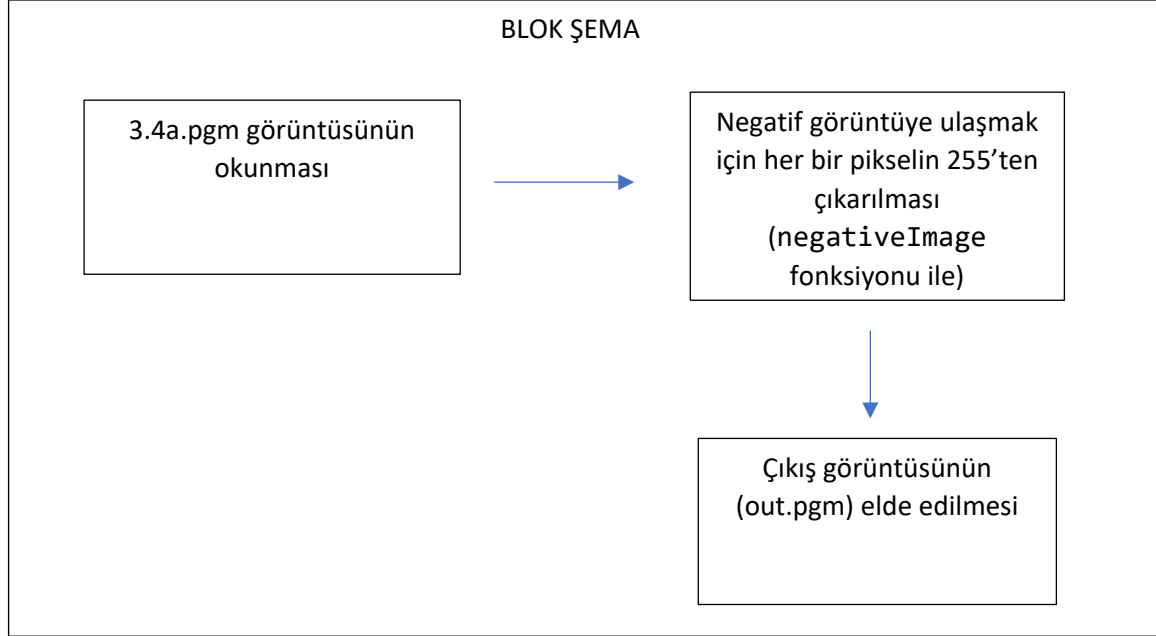


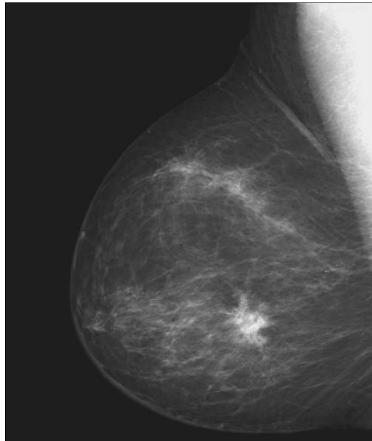
LAB1 RAPORU

1. Adım 1 – Soru 1 (Figure 3.4) :

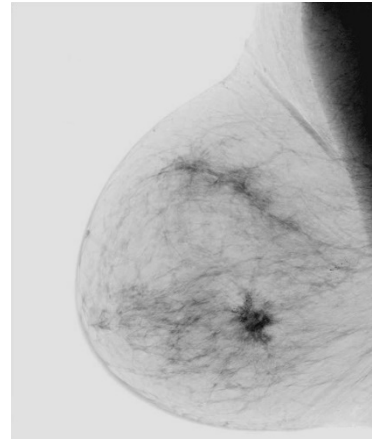
Bu kısımda yapılan işlemler ve ilgili işlemler sonucunda elde edilen çıktıların verildiği blok şema Şekil 1’de görüldüğü gibidir.



Şekil 1. Blok Şema



Şekil 2. Figure 3.4a

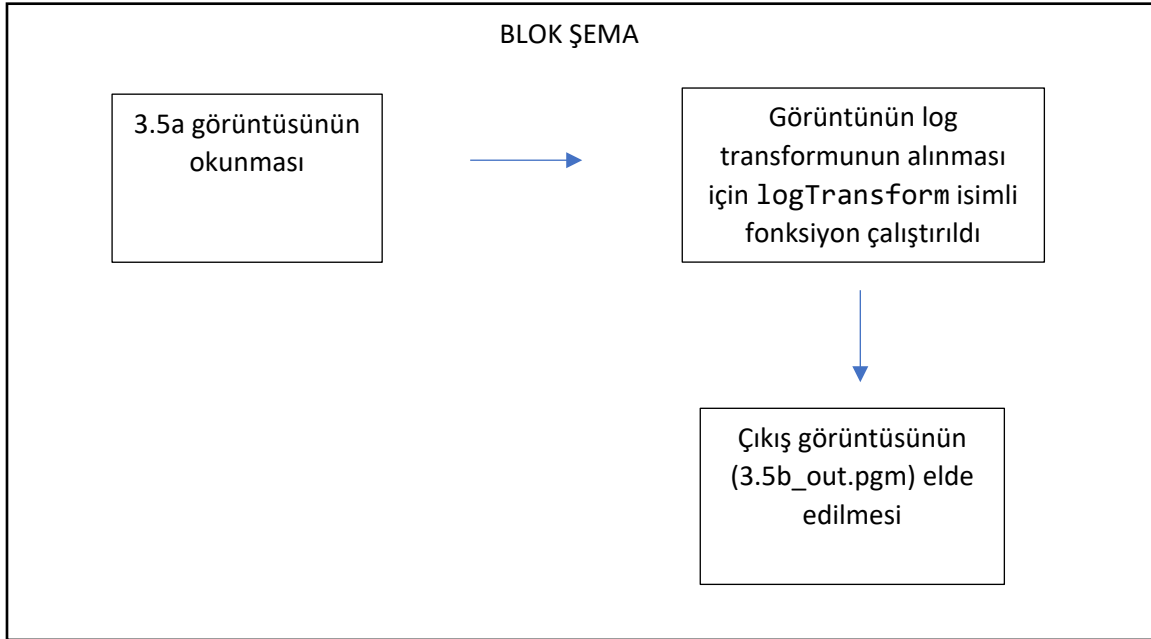


Şekil 3. Figure 3.4b_out

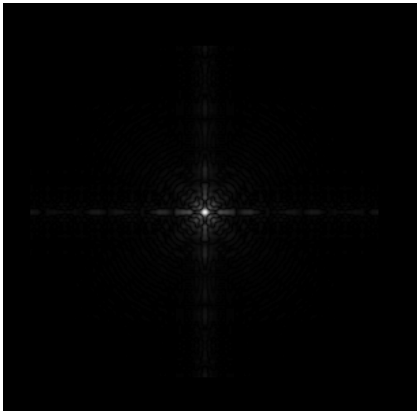
Şekil 3'teki görüntü okunduktan sonra `negativeImage` isimli fonksiyon çalıştırılmıştır. 8 bitlik bir görüntü için maksimum piksel değerinin 255 olduğu göz önünde bulundurulduğunda, her bir piksel değerinin 255'ten çıkarılması halinde ilgili piksel değerinin negatifi elde edilecektir. Bu işlem bütün piksel değerleri için yapılırsa görüntünün negatifi elde edilmiş olur.

Adım 1 – Soru 2 (Figure 3.5) :

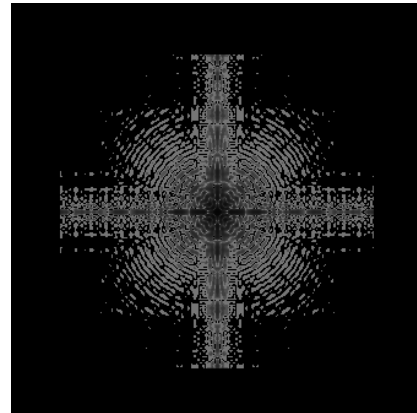
Bu kısımda yapılan işlemler ve ilgili işlemler sonucunda elde edilen çıktıların verildiği blok şema Şekil 1’de görüldüğü gibidir.



Şekil 4. Blok Şema



Şekil 5. Figure 3.5a

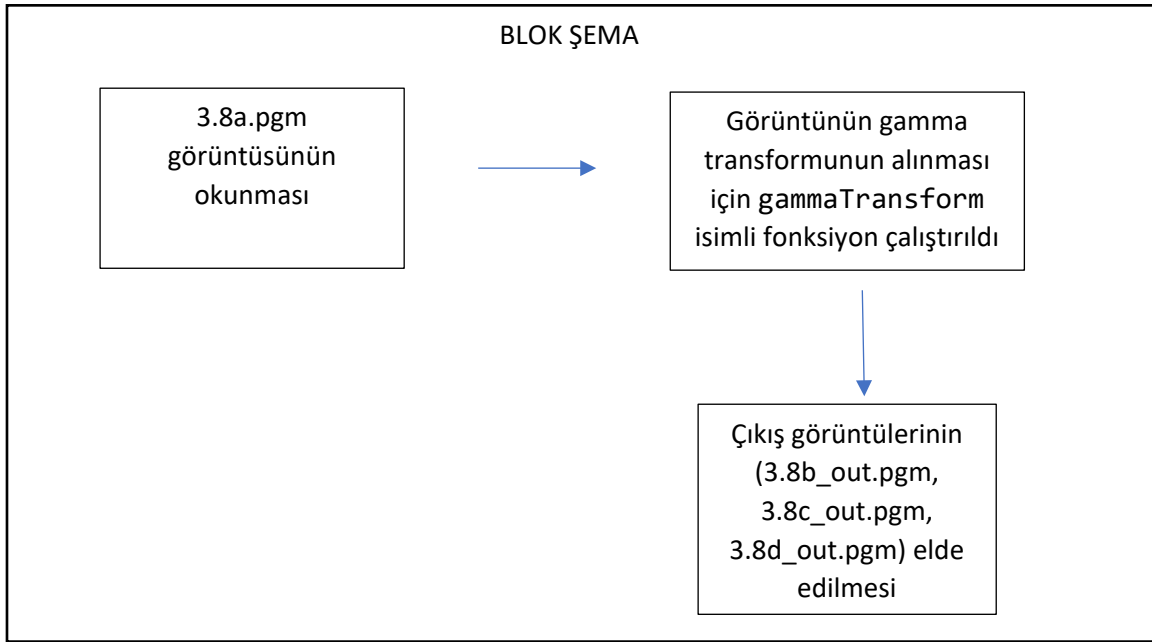


Şekil 6. Figure 3.5b_out

Şekil 5’teki görüntü okunduktan sonra logTransform isimli fonksiyon çalıştırılmıştır.

$s = c * \log(1 + r)$ formülü doğrultusunda giren r piksel değerine göre çıkış s piksel değeri her bir piksel için ayrı ayrı hesaplanır. Burada dikkat edilmesi gereken nokta, logaritma alma kısmıdır. Logaritması alınan bir sayı küçülecektir. Giriş görüntüsü pikseli r max 255 değeri alırsa, çıkış görüntüsü pikseli s ’de 2,4 değeri görülür. Bu yüzden 0-255 aralığının örneklenmesi gerekmektedir. Bahsedilen formülün (max piksel / okunan piksel) değeri ile çarpılmasıyla istenen örneklenme gerçekleşmiş olur ve beklenen görüntü doğru bir şekilde çıkışta görülür.

Adım 1 – Soru 3 (Figure 3.8) :



Şekil 7. Blok Şema



Şekil 8. Figure 3.8a



Şekil 9. Figure 3.8b_out



Şekil 10. Figure 3.8c_out



Şekil 11. Figure 3.8d_out

Şekil 8'deki görüntü okunduktan sonra gammaTransform isimli fonksiyon çalıştırılmıştır. $s = c * (r + \epsilon)^\gamma$ formülüne göre çıkış pikseli hesaplanır. Gamma transformu ile görüntünün beyazlık-siyahlık değerlerinde ayarlamalar yapılabilir. Bu ayarın seviyesini formülde üste yer alan gama (γ) değeri belirlemektedir. Gama değeri küçüldükçe görüntü beyaza yaklaşır. Giriş görüntüsü Şekil 8'de görüldüğü gibidir. Şekil 9, 10 ve 11'deki görüntülere uygulanan gammaTransform fonksiyonundaki gama değerleri sırasıyla 0.6, 0.4 ve 0.3 şeklindedir. Gama değeri küçüldükçe görüntüdeki beyazlık artar.

Adım 1 – Soru 4 (Figure 3.9) :

Bu sorudaki blok şema, Şekil 7'deki blok şemayla neredeyse aynıdır. Tek fark, girişte 3.9a.pgm görüntüsünün okunup çıkışta 3.9b_out.pgm, 3.9c_out.pgm, 3.9d_out.pgm görüntülerinin elde ediliyor olmasıdır.



Şekil 12. Figure 3.9a_out



Şekil 13. Figure 3.9b_out



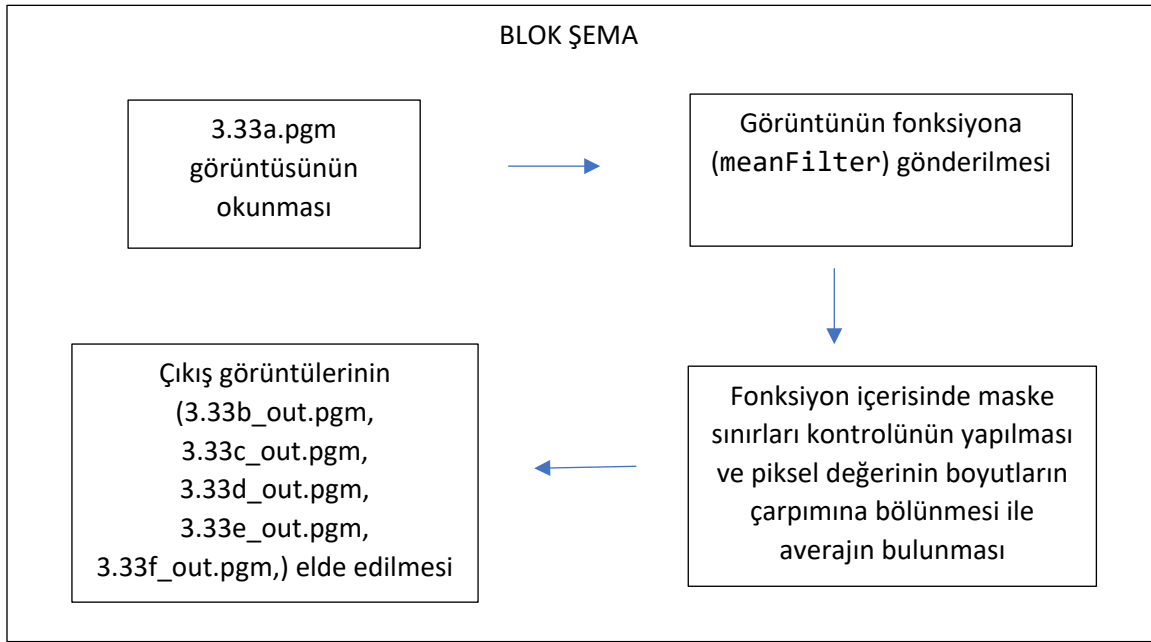
Şekil 14. Figure 3.9c_out



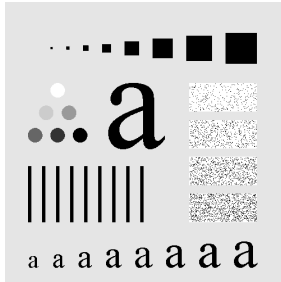
Şekil 15. Figure 3.9d_out

Bir önceki soruyla bire bir aynı işlemlerin yapıldığı, ancak gama değerlerinin bu sefer büyütülerek giriş görüntüsüne uygulandığı kısımdır. Şekil 12'deki giriş görüntüsünden elde edilen çıkış görüntüleri Şekil 13, 14 ve 15'te görülmektedir. Gama değerleri sırasıyla 3.0, 4.0 ve 5.0 olarak uygulanmıştır. Bir önceki soruda görülenin aksine, gama değeri büyüdükçe görüntü daha çok kararır.

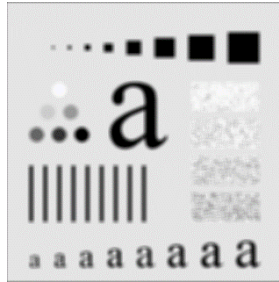
Adım 1 – Soru 5 (Figure 3.33) :



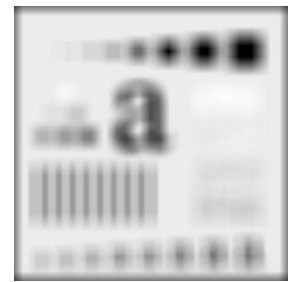
Şekil 16. Blok Şema



Şekil 17. Figure 3.33a



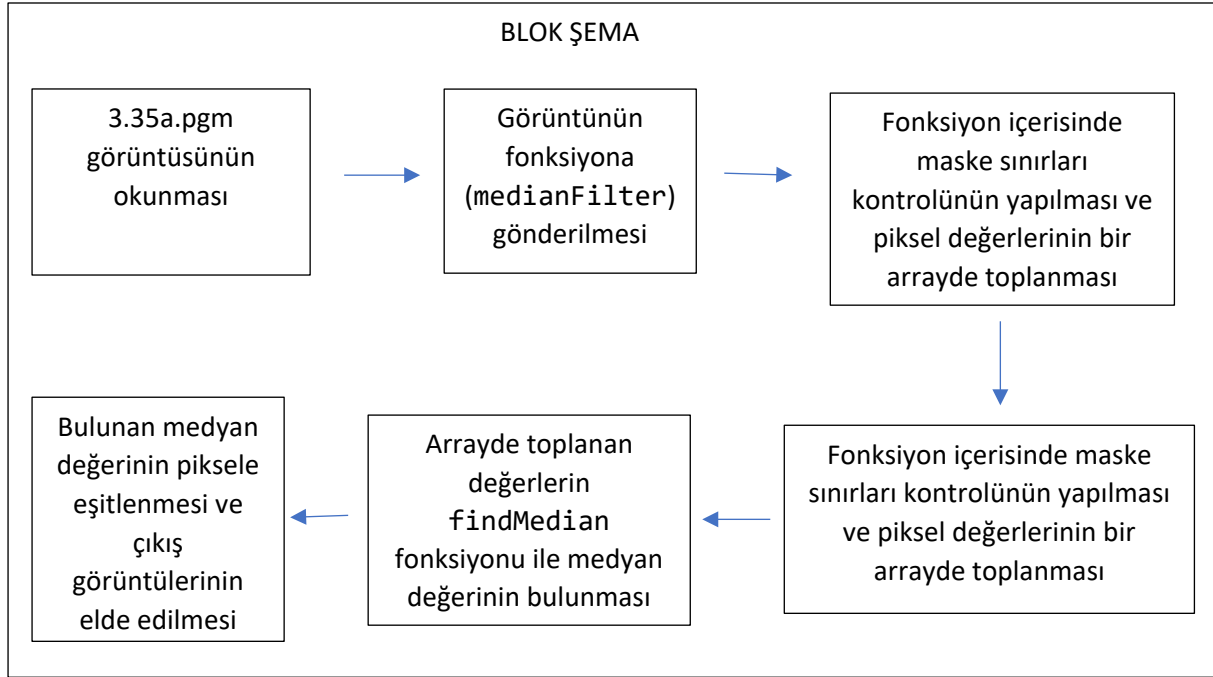
Şekil 18. Figure 3.33d_out



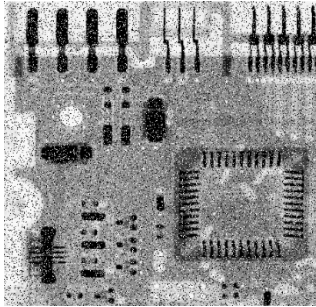
Şekil 19. Figure 3.33f_out

Şekil 17’deki görüntü okundan sonra, meanFilter isimli fonksiyon ile bulanıklaştırılma işlemine tabi tutulmuştur. Önceki yazılan fonksiyonlarda, görüntünün yani matrisin içinde gezildiği için iç içe iki for döngüsü yeterli oluyordu. Ancak bu sefer maskeleme işlemi söz konusu olduğundan dolayı matris içindeki maskede gezmek için ek olarak bir içe içe for döngüsü daha gereklidir. Önemli noktalardan biri, en içteki for döngüsünün içinde maskelerin sınır kontrolünün yapılması gerektiğidir. Bu kontrol yapılmadığı takdirde kod derlendikten sonra oluşan exe dosyası çalıştırıldığında stackdump hatası almak çok olası bir durumdur. Maske dışına taşma durumunda ilgili piksellere direkt 0 atanmıştır. Averaj hesaplaması, maske içerisindeki her bir pikselin toplamının maske boyutları çarpımına bölünmesi ile bulunur. Elde edilen değer aynı zamanda maskeleme sonucu görüntünün ilgili pikseline atanacak olan değeri ifade eder. Maske boyutu arttıkça, bulanıklaştırma oranı artar. Şekil 18 ve 19’da sırasıyla 9x9 ve 15x15’lik averaj maskelemesi sonucu oluşan görüntüler bulunmaktadır. Bütün şıklara ait görüntüler kod çalıştırıldığında görülmektedir.

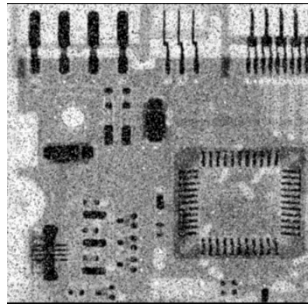
Adım 1 – Soru 6 (Figure 3.35) :



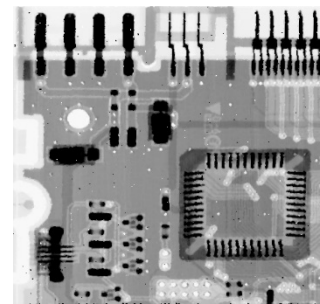
Şekil 20. Blok Şema



Şekil 21. Figure
3.35a



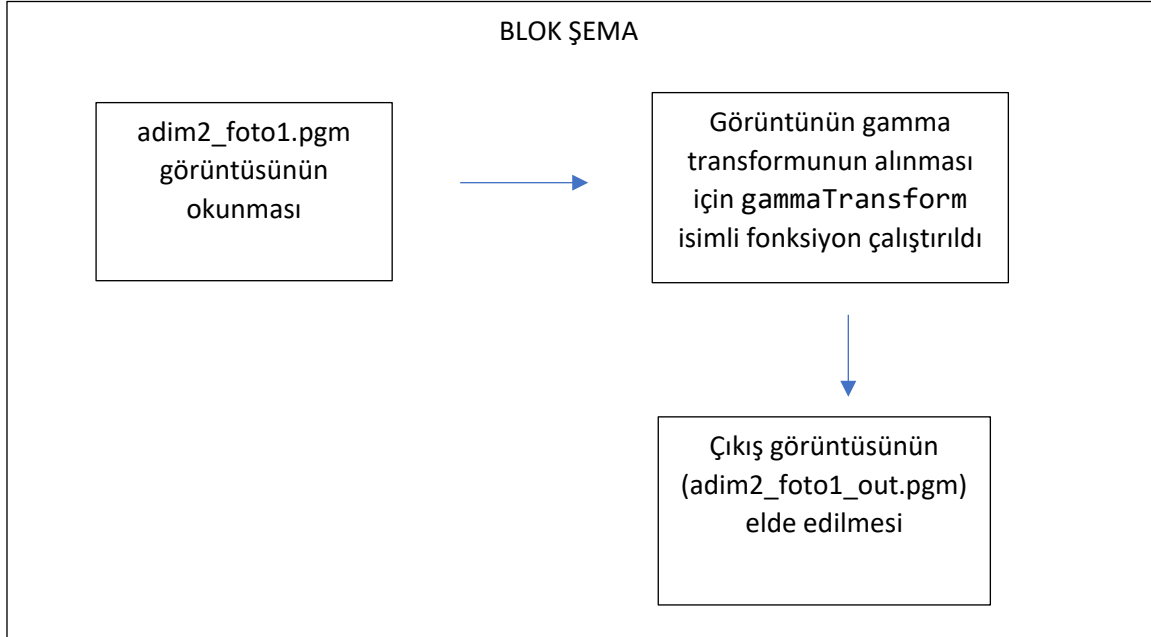
Şekil 22. Figure
3.35b_out



Şekil 23. Figure
3.35c_out

Şekil 21'deki görüntü okunup hem meanFilter hem de medianFilter isimli fonksiyonlar ile üzerindeki gürültü giderilmeye çalışılmıştır. Şekil 22 averaj filtresi, Şekil 23 ise medyan filtresi uygulandığında çıkış görüntüsüne karşılık gelmektedir. Medyan filtresi üzerinden konuşulduğunda, maskeleme işlemi söz konusu olduğu için kodda dört adet for döngüsü yer almaktadır (ikisi görüntü ikisi maske için). Stackdump hatası almamak için yapılan maske sınır kontrolünde, eğer sınır taşması durumu olursa direkt olarak diğer piksele geçilip o pikselin kontrolü yapılır. Taşma olmaması durumunda maskeye ait bütün pikseller bir arrayde tutulur. Bu array findMedian isimli fonksiyona gönderilir. Bu fonksiyonda önce bütün sayılar küçükten büyüğe sıralanır, sonrasında ise bulunan medyan değeri geri döndürülür. Bulunan değer aynı zamanda piksel değeridir. Bu değerler ile yeni görüntü elde edilir. Gürültü giderme işlemi için averaj ve medyan filtresi karşılaştırıldığında, en iyi sonucu medyan filtresinin verdiği görülecektir.

2. Adım 2 – Foto 1 :



Şekil 24. Blok Şema



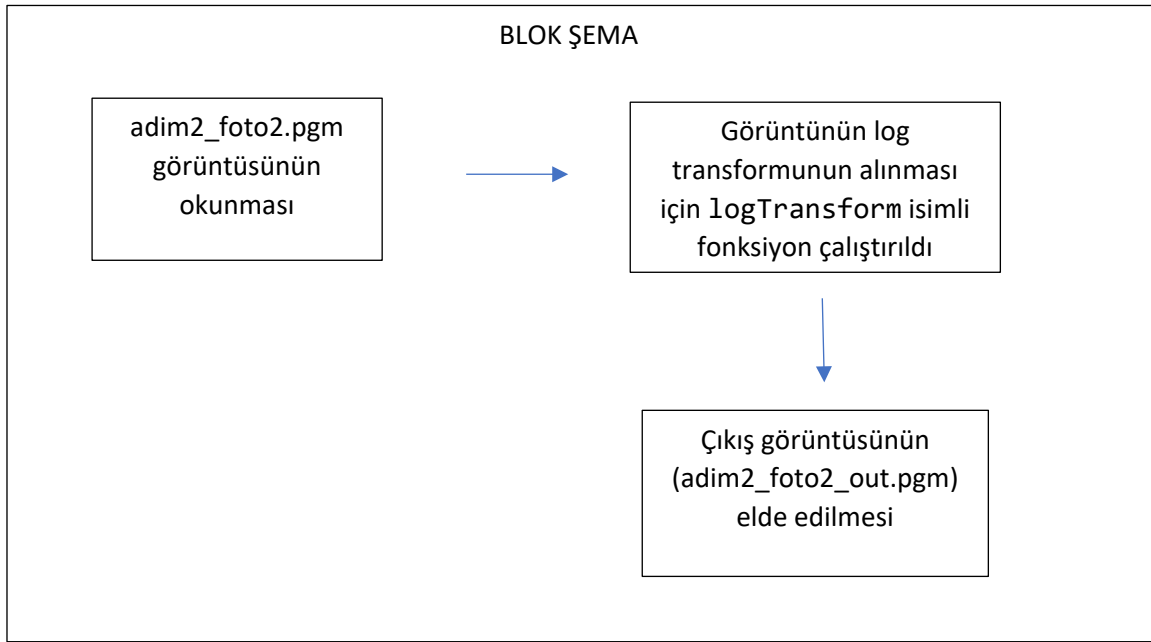
Şekil 25. adim2_foto1.pgm



Şekil 26. adim2_foto1_out.pgm

Şekil 25'te düşük kontrastlı bir görüntü vardır. Bu görüntünün iyileştirilmiş hali, Şekil 26'da görülebilmektedir. Bunu yapabilmek için gammaTransform fonksiyonu kullanılmıştır. Kodda $s = c * (r + \epsilon)^\gamma$ formülünde yer alan gama (γ) yerine 3.0 değeri yazılmıştır. Böylelikle düşük kontrast sorunu ortadan kalkmış ve görüntü iyileşmiştir.

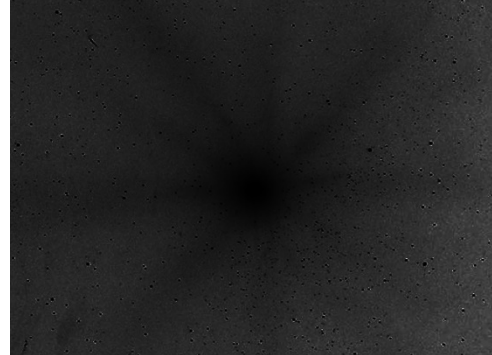
Adım 2 – Foto 2 :



Şekil 27. Blok Şema



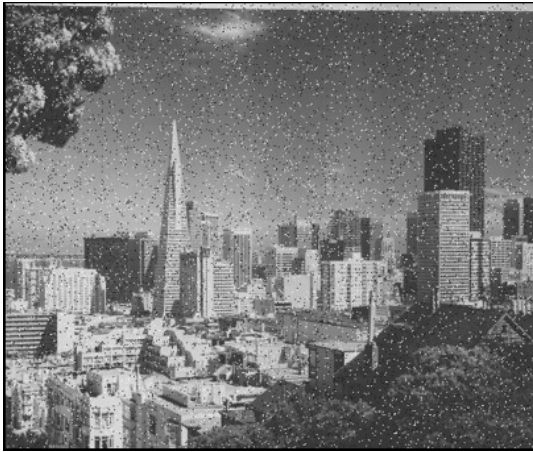
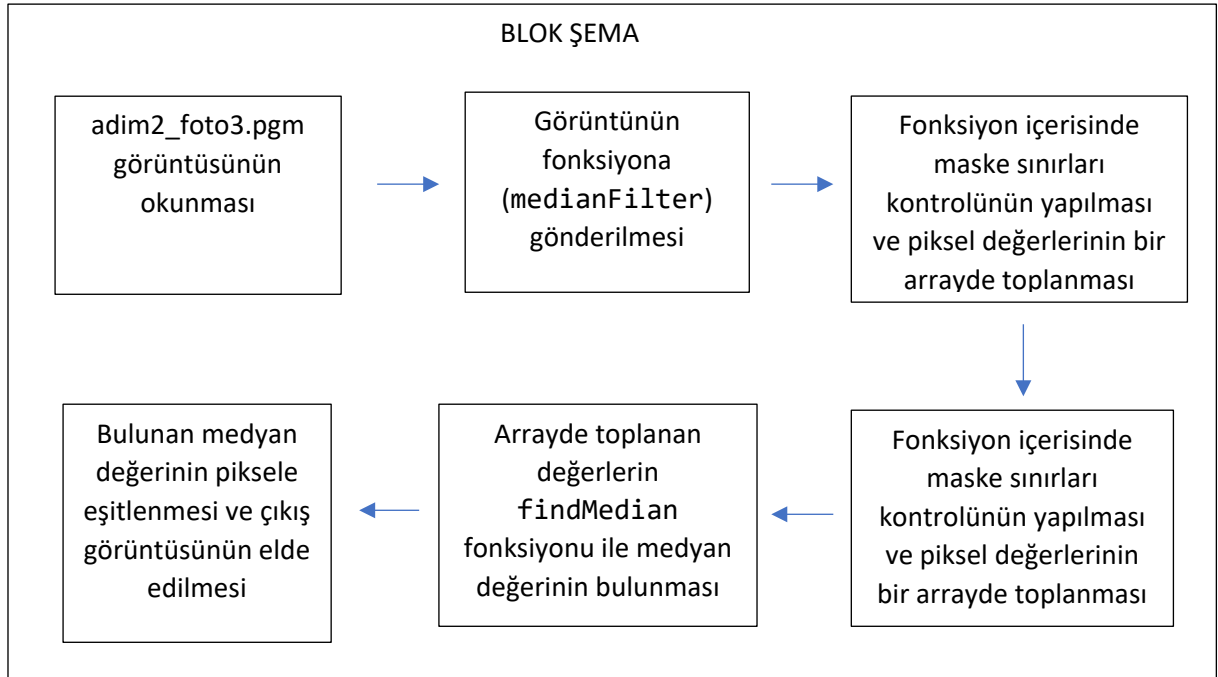
Şekil 28. adim2_foto2.pgm



Şekil 29. adim2_foto2_out.pgm

Şekil 28’de görülen görüntü, log transformu uygulanarak Şekil 29’daki hale getirilmiştir. Görüntünün iyileştirilmesi, amaca göre değişen bir durum olduğu için yukarıdaki şekillerde görüldüğü gibi bir iyileştirme görüntü üzerinde uygulanabilir.

Adım 2 – Foto 3 :



Şekil 30. adim2_foto3.pgm



Şekil 31. adim2_foto3_out.pgm

Şekil 30’da görülen fotoğraf medyan filtresi uygulanarak Şekil 31’deki hale getirilmiştir. Gürültü gidermedeki en iyi filtrenin medyan filtresi olduğu daha önce bahsedilmişti. Bu örnekte görüldüğü gibi de bu durum bir kez daha kanıtlanmıştır.

Referanslar:

1. Ders notları
2. Digital Image Processing by Rafael C. Gonzalez, Richard E. Woods
3. <http://lawrencetrujillophotography.blogspot.com/2013/02/high-and-low-contrast.html> (Şekil 27)
4. <https://www.space.com/57-stars-formation-classification-and-constellations.html> (Şekil 28)
5. <https://www.sciencedirect.com/topics/engineering/pepper-noise> (Şekil 30)