

BLM 5113- Sayısal Video İşleme

Kenar Belirleme İşlemi

Metin Uslu

235B7014

1. Özet

Bu çalışmada bizimle paylaşılan 3 adet PGM (Portable Gray Map) uzantılı görüntü nesneleri üzerinde Gaussian ve Sobel filtreleri ve bu filtreler için parametre değerleri ile değiştirilerek görüntü üzerinde “Kenar Belirleme İşlemi” çalışması yapılmıştır. İlk olarak PGM formatındaki görseller Python programlama dili ile 2 boyutlu bir dizi olarak okunmuştur. Ardından bu görsellerin üzerine ilk olarak Gaussian Filter Kernel Size [3x3, 5x5, 7x7] ve Sigma [1, 5] (standart sapma) değerleri seçilerek her bir görüntünün 6 (KernelSize x Sigma) farklı blurlanmış hali elde edilmiştir. Ardından 3 adet PGM görseline Sobel Filtresi uygulanarak ve orijinal görsellerin kenarlarının tespit edilmesi sağlanmıştır. Çalışmada son olarak Gaussian filtresi uygulanarak en bulanık halde bulunan görseller için Sobel Filtresi uygulanmıştır. Devamında orijinal görseller ile en bulanık görsellerin sonuçları değerlendirilmiştir.

2. Giriş

Görüntü işleme süreçlerinde, dijital görüntüler üzerinde çeşitli işlemler uygulanarak analizler yapılır ve görüntülerin yorumlanması sağlanır. Bu işlemler arasında görüntü azaltılması/giderilmesi, kenar tespiti, keskinleştirme ve nesne tanıma gibi birçok farklı görev bulunmaktadır. Bu çalışmada ise Kenar Tespiti için Sobel Filtresi, görüntüdeki görüntüler azaltılması için ise Gaussian Filtresi kullanılacaktır.

Çalışmada ilk olarak PGM formatında verilen P2 ASCII Text dosya ve P5 Binary dosya formatında bulunan 3 adet görüntünün okunması ve bunların 2 boyutlu bir dizi olarak return edilmesi sağlanmıştır. Burada ASCII Text dosyası satır satır okunmuş, Binary dosyası ise byte olarak okunmuştur.

Devamında ön işleme adımı olarak görüntülerin azaltılması, ortadan kaldırılması için Gaussian Filtresi 3 farklı Filtre büyüklüğü ve 2 farklı Sigma değeri olmak üzere 6 farklı varyasyon ile denenmiştir. Buradaki filtre büyüklükleri ve sigma değerleri varsayılan olarak çalışmada belirtilmiştir. Ardından orijinal görüntü üzerine ve blurlanmış görüntüler üzerine Sobel Filtresi uygulanma adımına geçilmiştir.

Kenar bulma işlemi için genellikle Sobel ve Gauss filtreleri kullanılmaktadır. Sobel filtresi, görüntüdeki kenarları gradyanlar vasıtasıyla ayırmaya çalışır. Bu amaçla, dikey ve yatay kenarları algılayabilmek için iki farklı filtre matrisi kullanılır:

Dikey Kenarlar için

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

Yatay Kenarlar için

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

Gauss filtresi ise görüntülerdeki görüntüleri gidermek için kullanılır. Bulanıklaştırma veya görüntü azaltma işlemi, piksellerin ortalama yoğunluk değerlerini hesaplayarak ve komşu piksellerin değerlerini birbirine yaklaştırarak gerçekleştirilir.

Gauss filtresinin iki hiper parametresi vardır. Birincisi, filtrelerin kaç kaçlık olacağını belirleyen filtre boyutudur. Daha büyük filtre boyutları, daha fazla bulanıklaştırma ve daha az görüntü anlamına gelir, ancak çok büyük filtre boyutları detay kaybına neden olabilir. İkinci hiper parametre ise sigma olarak adlandırılır ve Gauss filtresinin varyansını temsil eder. Daha yüksek sigma değerleri, daha fazla bulanıklaştırma ve daha az görüntü anlamına gelir, ancak aşırı yüksek sigma değerleri de görüntünün aşırı bulanıklaşmasına neden olabilir. Sigma, kenar piksellerin Gauss dağılımına göre ne kadar katkı

yapacağını temsil eder. Yüksek bir varyans, en yakın kenar piksellerin daha fazla etkileşimde bulunmasına yol açarken, düşük bir varyans ile en yakın kenar piksellerin etkisi daha az olacaktır.

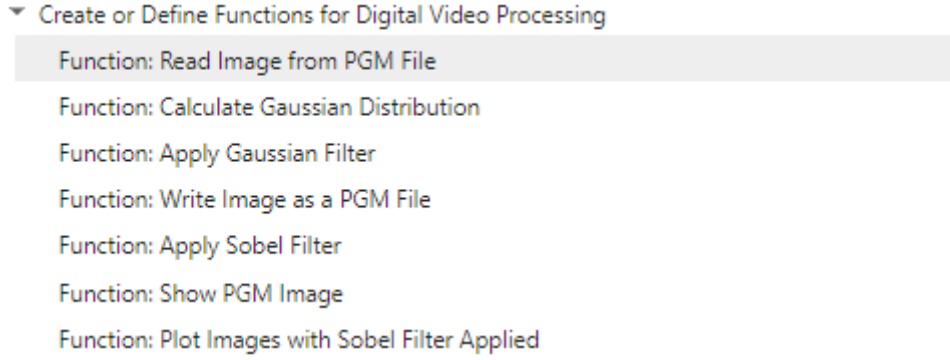
3. Yöntem

Bu problemin çözümünde 3 ayrı adımdan oluşmaktadır.



Şekil 1: Adım 1 Görsel

Adım 1: Proje de kullanılan library import edilmesi, bazı genel configlerin set edilmesi gibi projenin genel akışını belirleyen kısımlardan oluşmaktadır. Uygulama Python dili kullanılarak ve built-in ve standart kütüphanelerin (numpy) yardımı ilgili fonksiyonlar yazılarak çözümlenmiştir.



Şekil 2: Adım 2 Görsel

Adım 2: Bu problem çözümünde ilk olarak genel fonksiyonların yazılması sağlanmıştır. Burada toplamda 8 tane fonksiyon yazılmıştır.

1. Read Image from PGM File (P2 ve P5 ayrı ayrı olmak üzere)

Bu kısımda 2 tane fonksiyon yazılmıştır. Sebebi ise P2 formatında yani ASCII Text File olan görseller ile P5 Binary File olan görsellerin birbirinden farklı yapıda bulunmalarından kaynaklanmaktadır. Text File'lar satır satır okunarak, Binary File'lar byte byte okunarak 2 boyutlu bir görüntü nesnesi (img_arr) ve görüntüye ait width, height max_val değerleri return edilmiştir.

2. Calculate Gaussian Distribution

Gaussian Filter uygulanabilmesi için gerekli olan Gaussian Distribution ait olasılık yoğunluk fonksiyonu aşağıdaki şekilde kodlanmıştır.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

3. Apply Gaussian Filter

Görüntü nesnesi üzerinden uygulanacak olan Gaussian Filter burada kodlanmıştır. Bu fonksiyon 3 tane parametre almaktadır. Bunlardan ilki `img_arr`, diğerleri `sigma` ve `size` değerleridir. `Sigma` değeri uygulama esnasında 1 ve 5 değerleri almaktadır. `Sigma`, Gaussian dağılımının genişliğini dolayısıyla yapılacak filtreleme sürecinin etkisi kontrol eder.

`Size` ise Kernel Size değerini ifade etmektedir ve 3, 5 ve 7 değerleri almaktadır. Bu değer küçük yada büyük olması işlem gücü doğrudan etkilemektedir. Örneğin 3x3 filtre için 8 pixel kullanılarak bir değer oluşturuluyorken, 5x5 filtre 24 pixel kullanılarak yeni bir değer oluşturulur.

Fonksiyon görevi `img_arr` nesnesi üzerinde belirlenen Kernel Size filtreleri ve `Sigma` değerleri alarak resmin üzerindeki gürültülerin azaltılması/yok edilmesi/pürüzsüzleştirilmesi sağlamaktadır.

4. Write Image as a PGM File

Gaussian Filtresi ve Sobel Filtresi uygulandıktan sonra elde edilen nesnenin diske persist edilmesi sağlamak için kullanılmaktadır.

5. Apply Sobel Filter

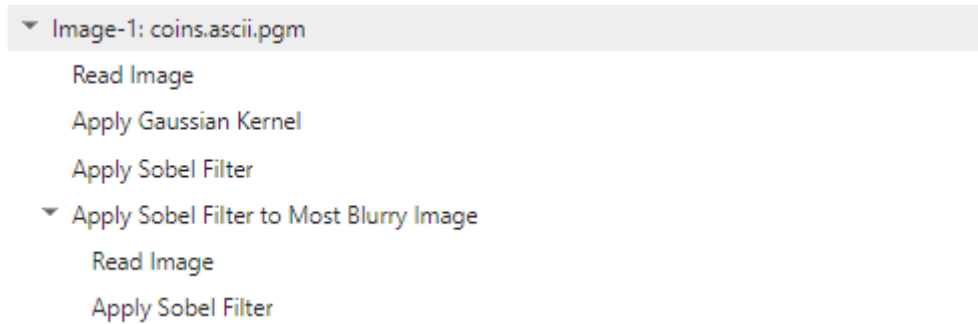
Görüntü nesneleri üzerinde kenar algılama için yaygın olarak kullanılan klasik filtrelerden bir tanesidir. Dikey ve Yata yönleri olmak üzere 2 ayrı filtre kullanarak kenarların tespiti sağlanır. Görüntü üzerine Sobel filtresini uygulamak için yazılmıştır.

6. Show PGM Image

Raw PGM ve ardından filtrelerden geçirilerek elde edilen görüntülerin görselleştirilmesi için yazılmıştır.

7. Plot Images with Sobel Filter Applied

Sobel filtre uygulandıktan sonra elde edilen Gradient (X ve Y) ve Magnitude nesnelerinin görselleştirilmesi için yazılmıştır.



Şekil 3: Adım 3 Görsel

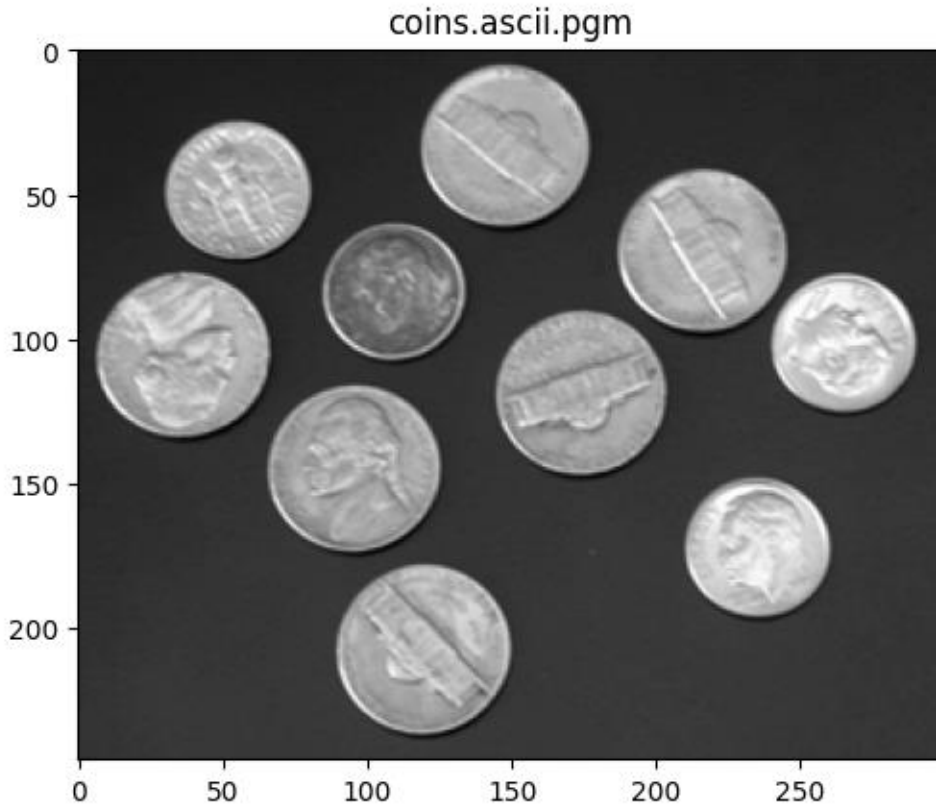
Adım 3: Ardından her bir PGM P2 ve P5 formatlı görseli üzerinde öncelikle “Read Image”, “Apply Gaussian Kernel (farklı Kernel Size ve Sigma değerleri ile)”, “Apply Sobel Filter” ve “Apply Sobel Filter to Most Blurry Image” adımları uygulanmıştır. Daha fazlası için Notebook (.ipynb ve .html) inceleyebilir ve çıktılara ulaşabilirsiniz.

4. Uygulama

Uygulama kısmında her 3 görseli için;

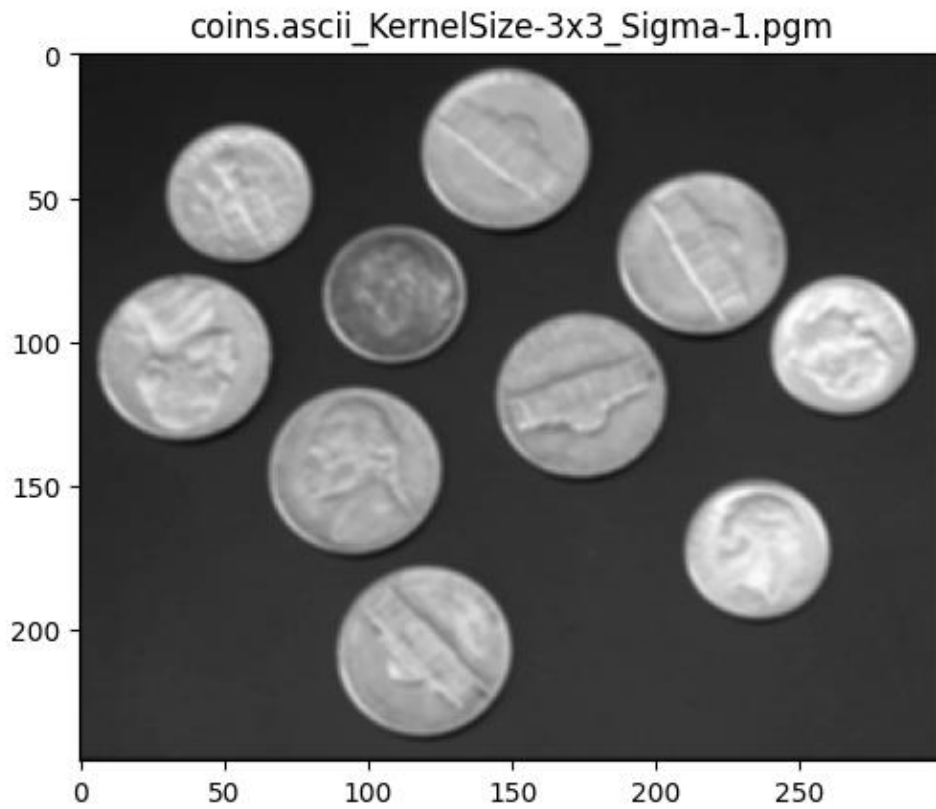
- ❖ Orijinal hali,
- ❖ Gaussian filtresi kullanılarak (KernelSize:{3,5,7} x Sigma{1,5}) 6 adet Blurlanmış halleri,
- ❖ Orijinal görüntü üzerine uygulana Sobel filtresi uygulanmış hali,
- ❖ Gaussian filtre kullanılarak elde edilen en bulanık görsele üzerine Sobel filtresi uygulanmış hali yer almaktadır.

4.1. Image-1: coins.ascii.pgm

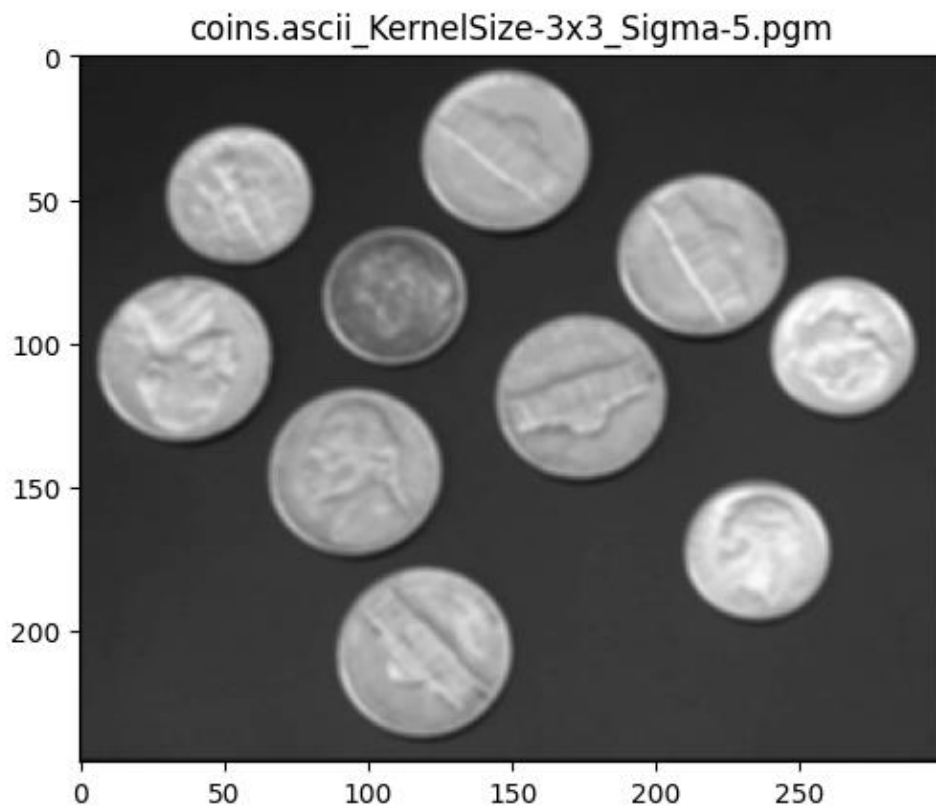


4.1.1. *Gaussian Filter*

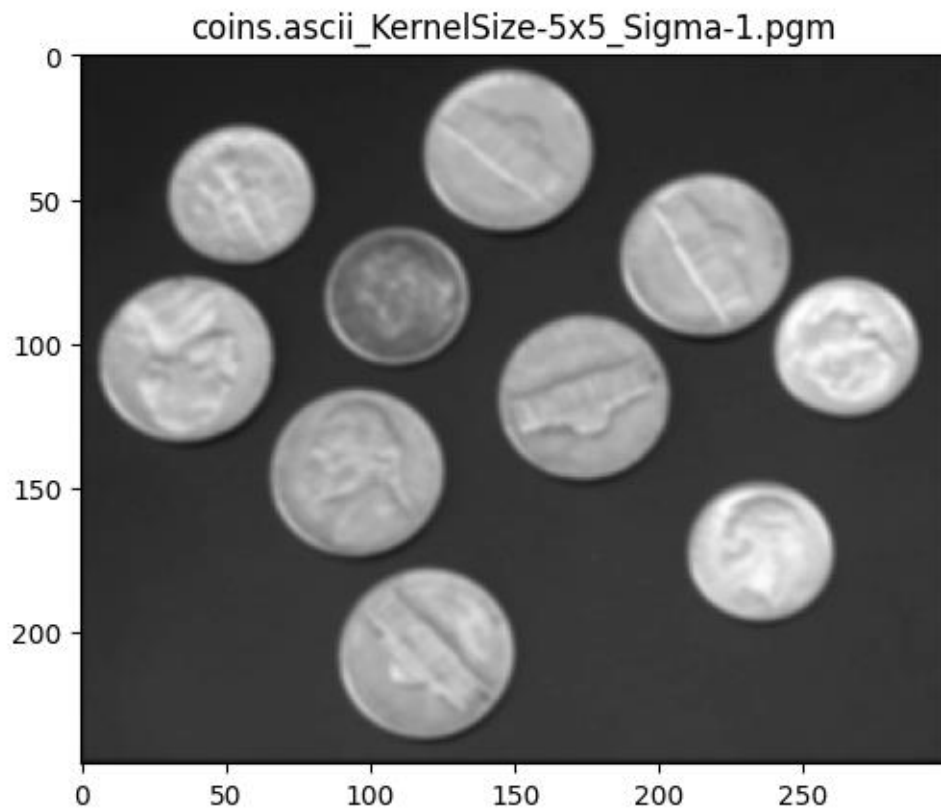
Kernel Size: 3x3 & Sigma: 1



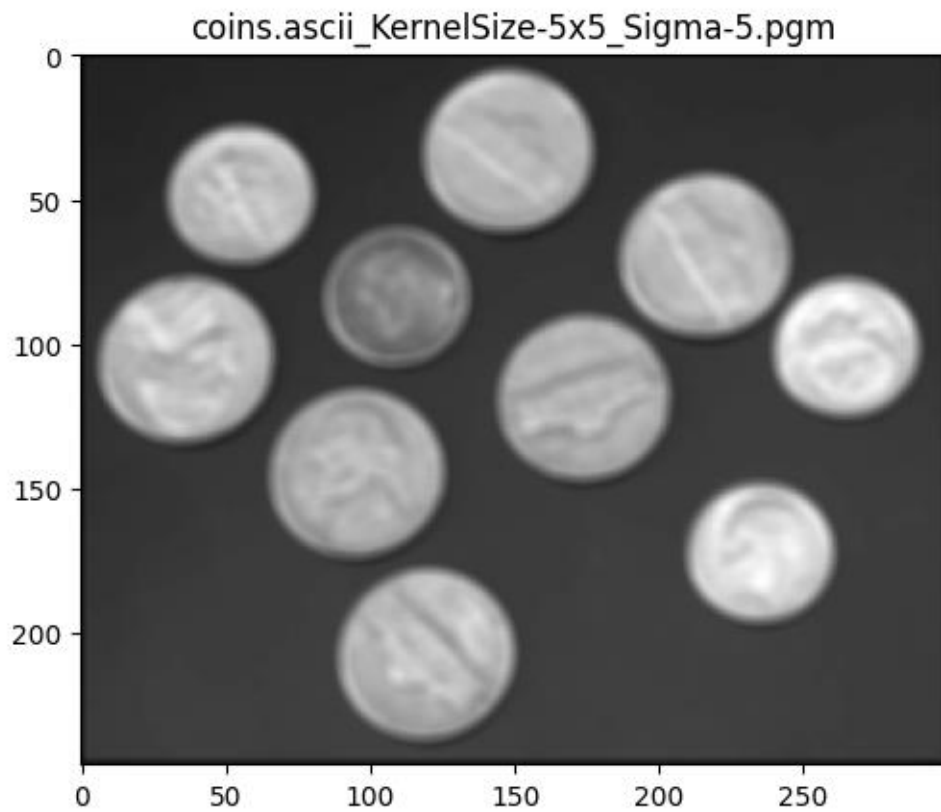
Kernel Size: 3x3 & Sigma: 5



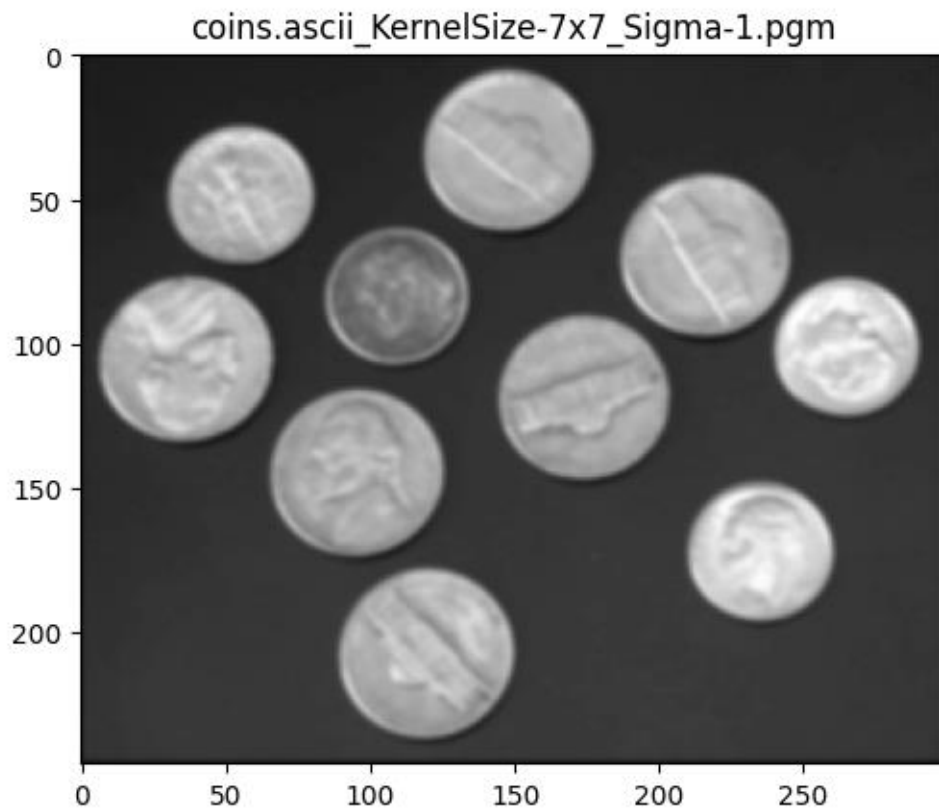
Kernel Size: 5x5 & Sigma: 1



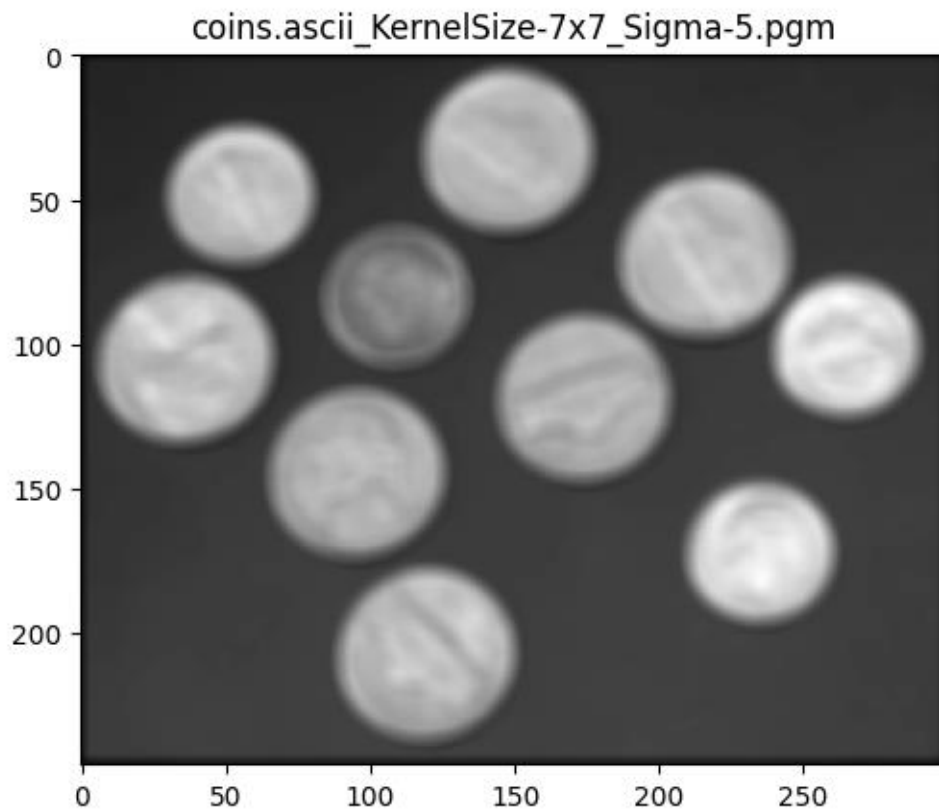
Kernel Size: 5x5 & Sigma: 5



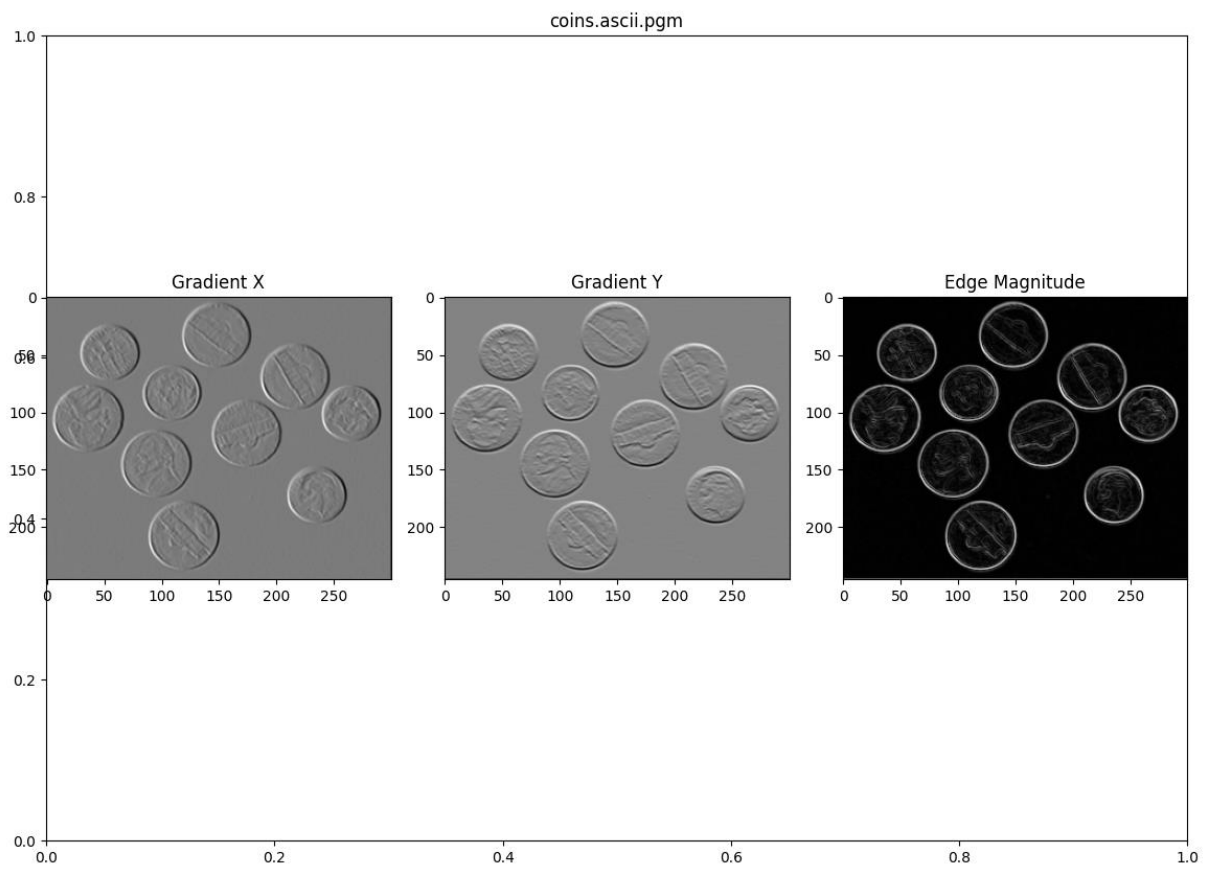
Kernel Size: 7x7 & Sigma: 1



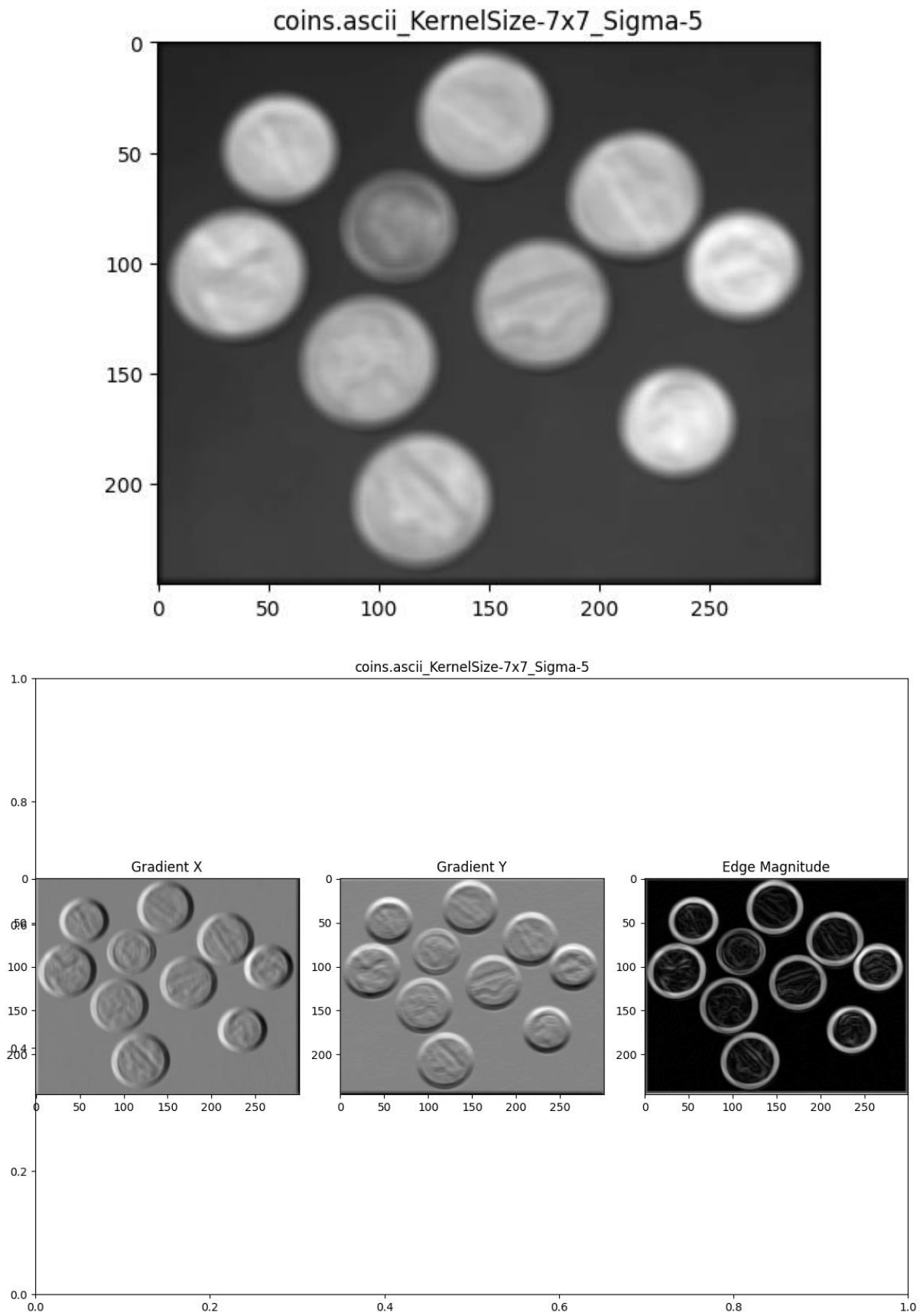
Kernel Size: 7x7 & Sigma: 5



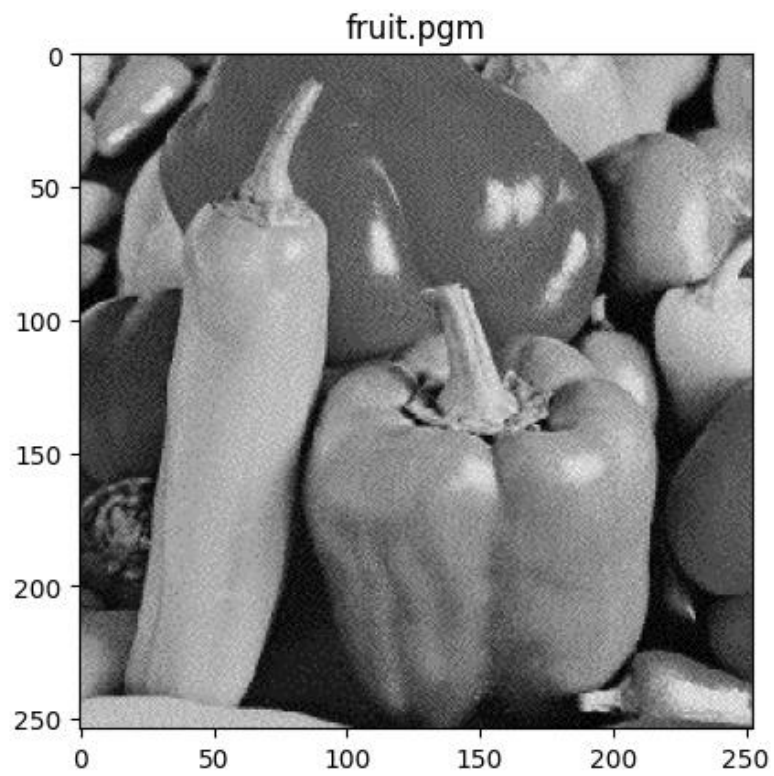
4.1.2. Sobel Filter



4.1.3. Sobel Filter to Most Blurry Image

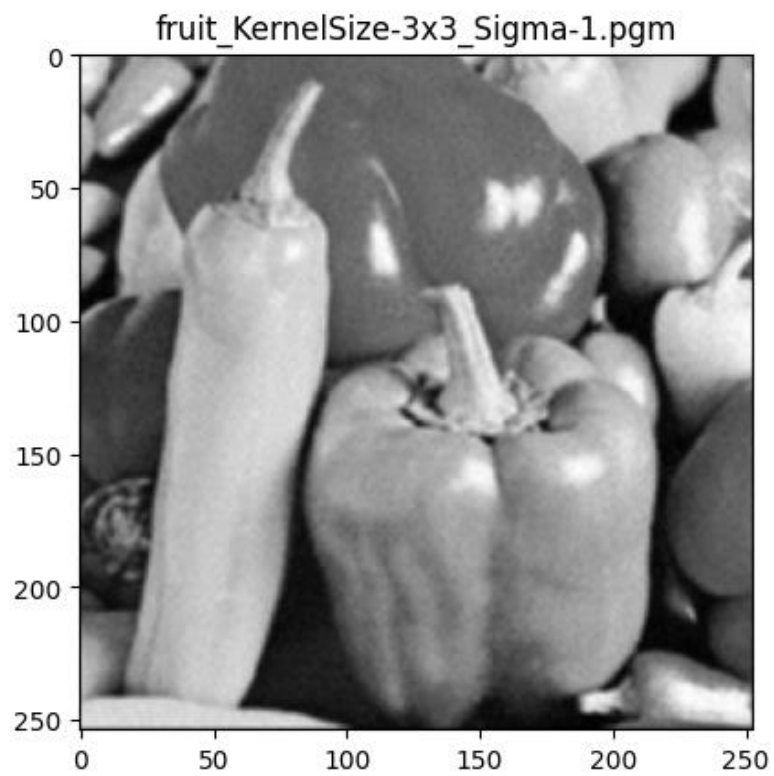


4.2. Image-2: fruit.pgm

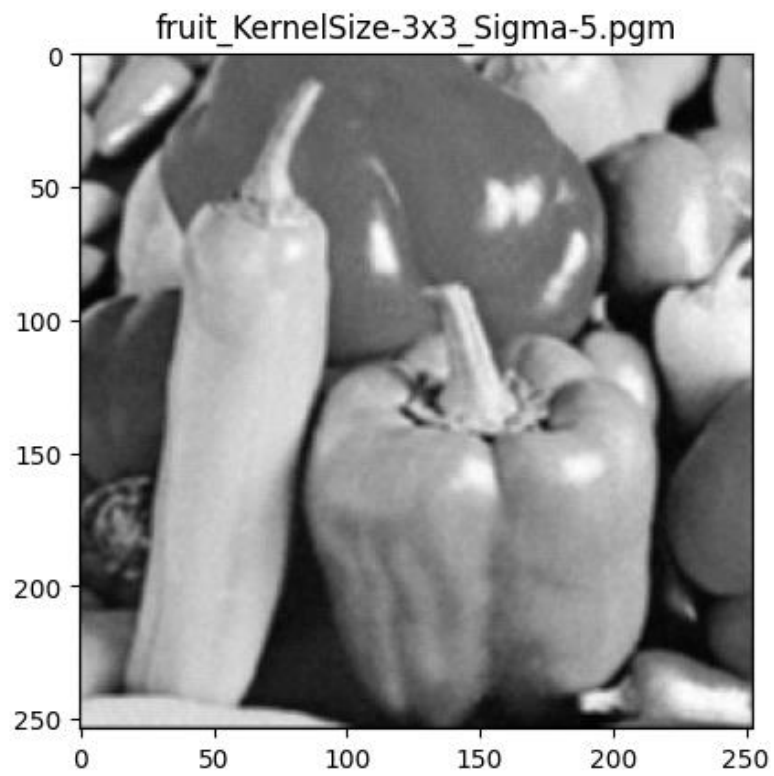


4.2.1. Gaussian Filter

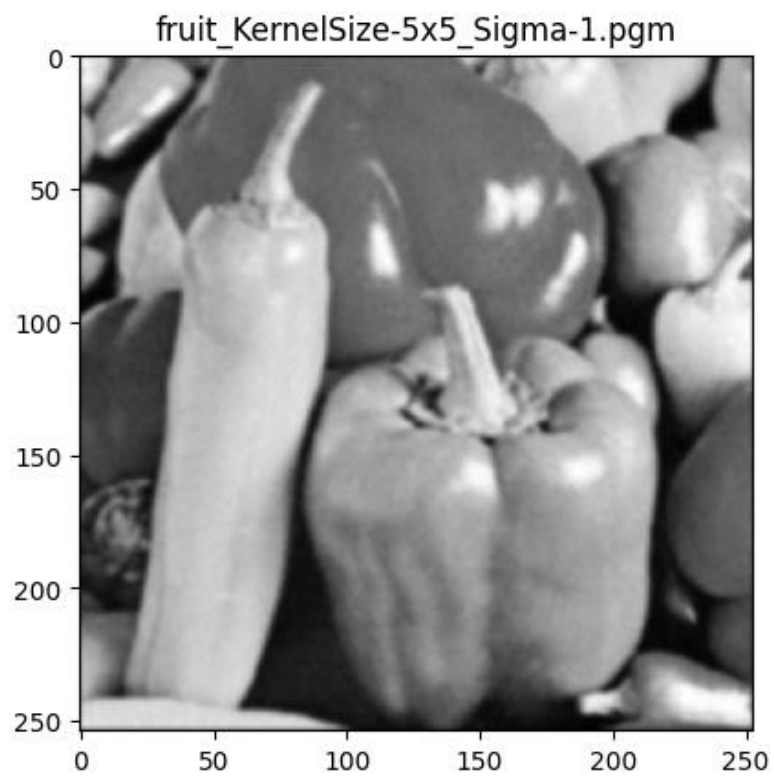
Kernel Size: 3x3 & Sigma: 1



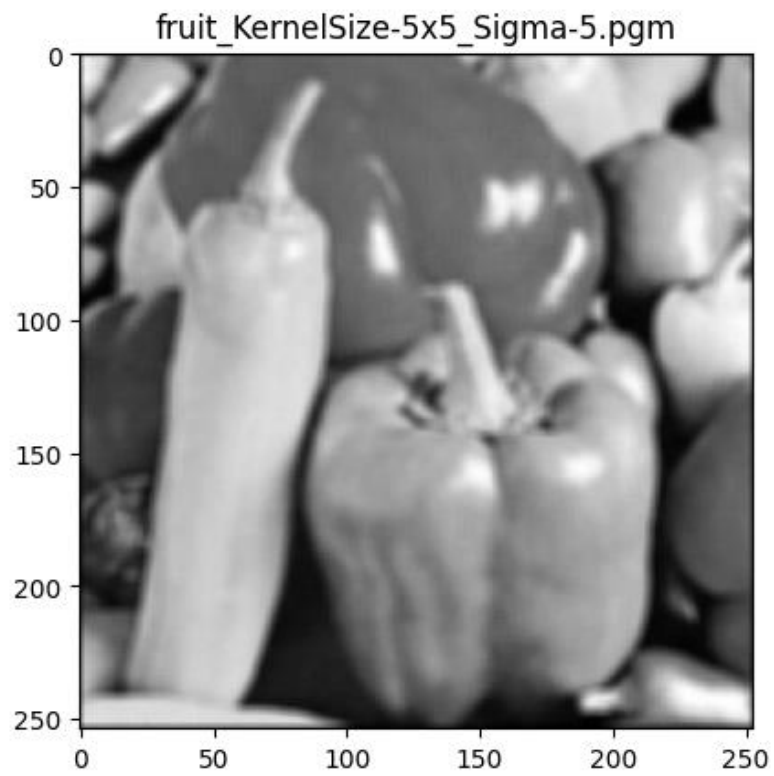
Kernel Size: 3x3 & Sigma: 5



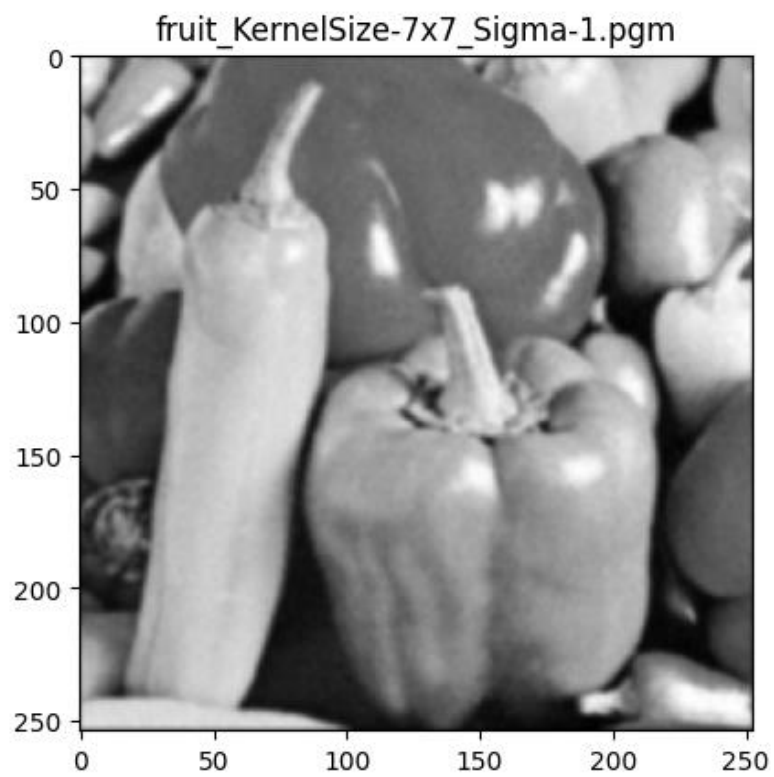
Kernel Size: 5x5 & Sigma: 1



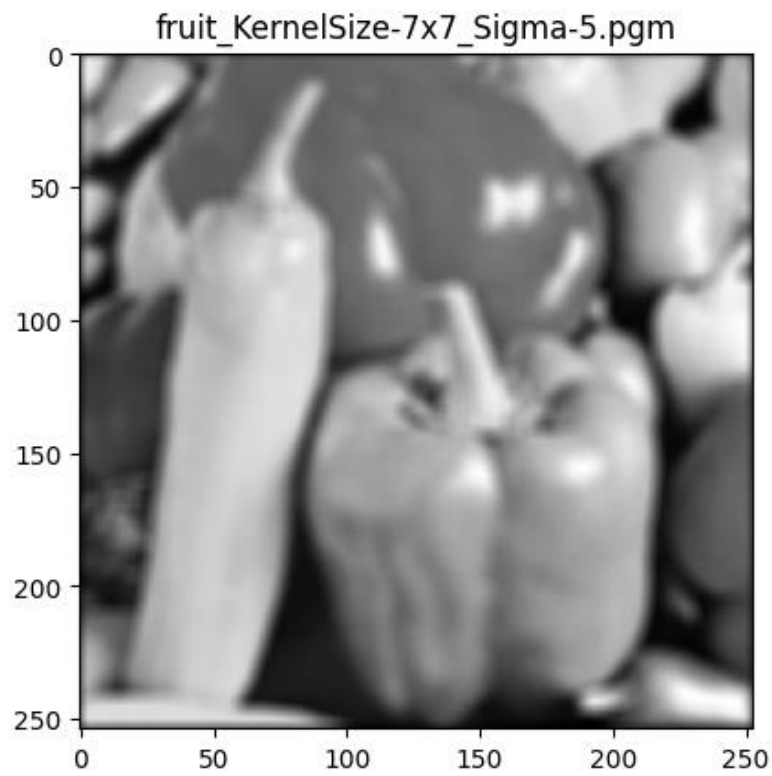
Kernel Size: 5x5 & Sigma: 5



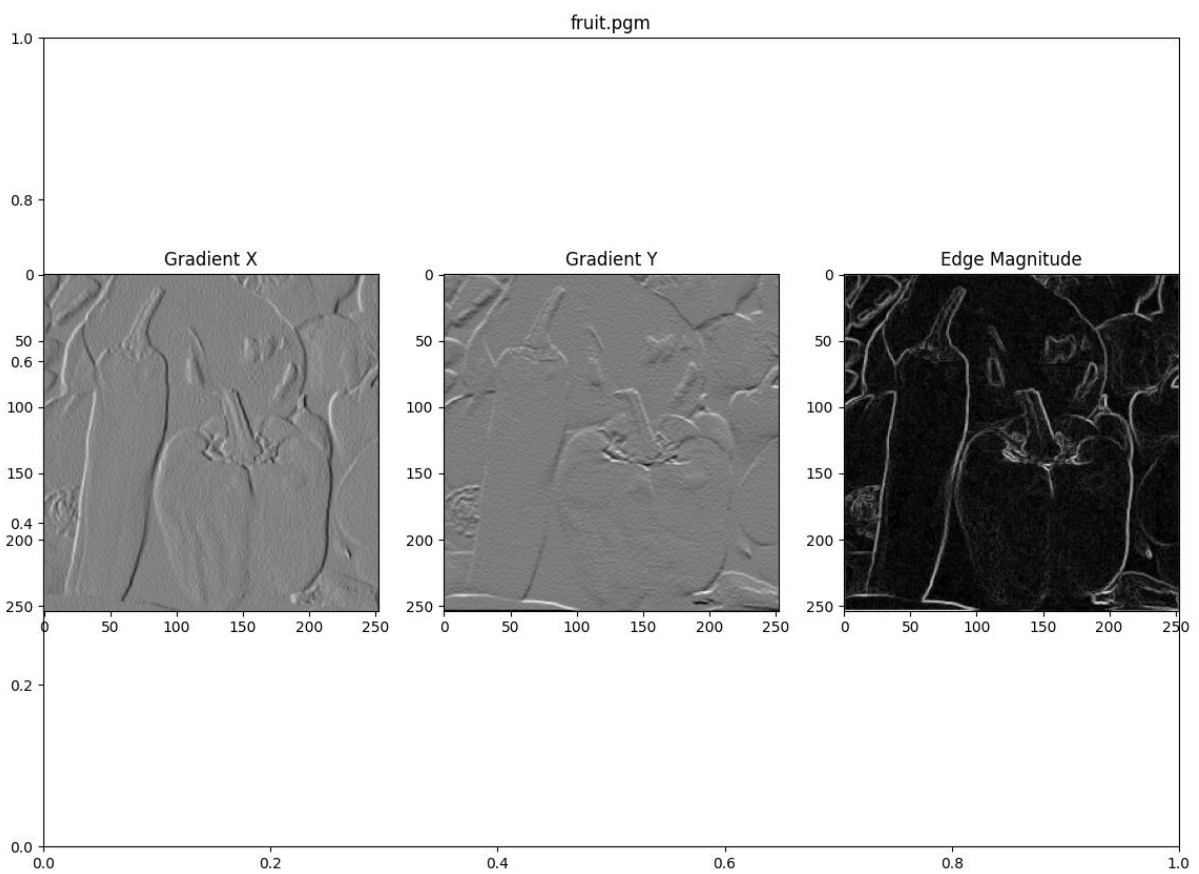
Kernel Size: 7x7 & Sigma: 1



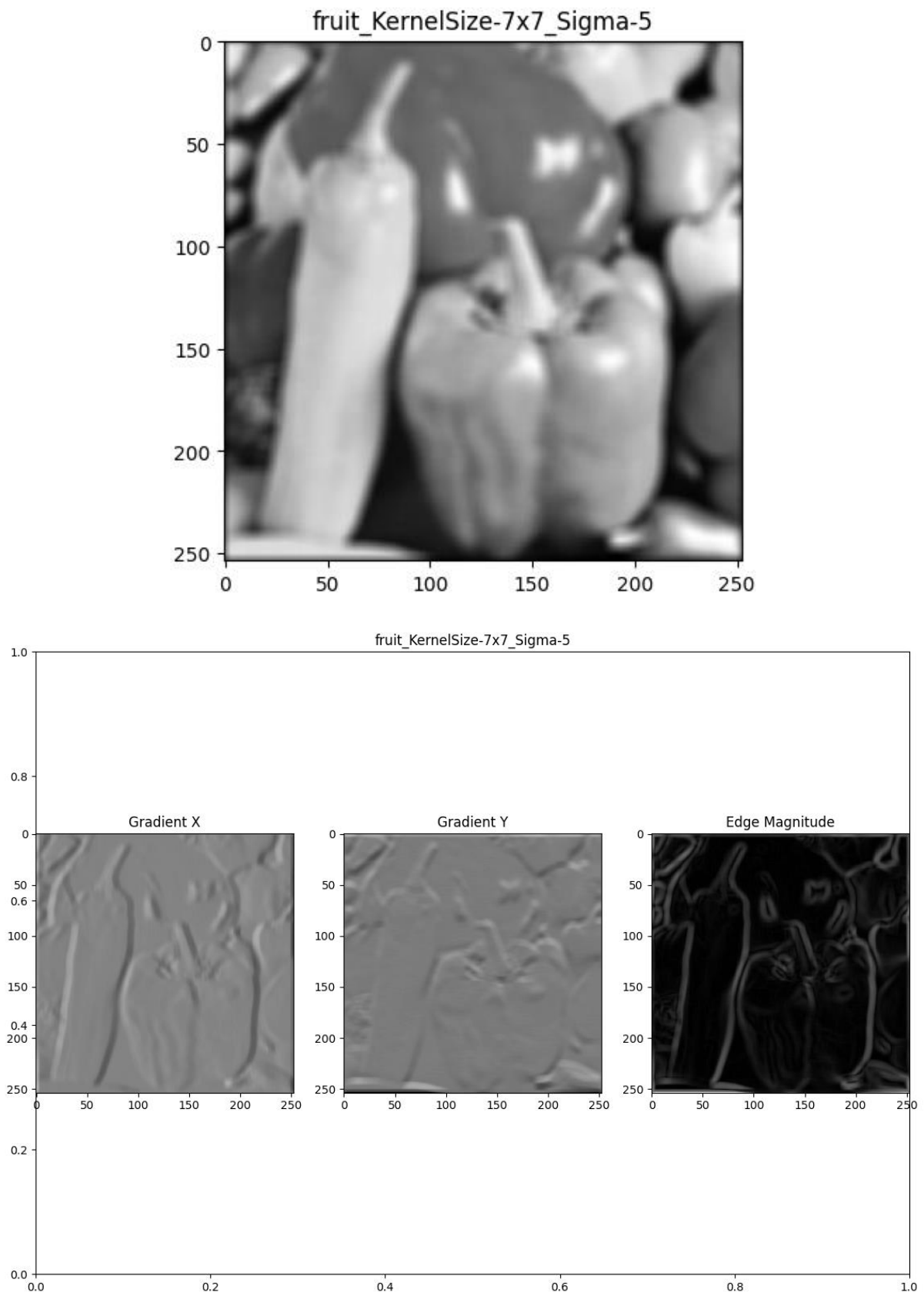
Kernel Size: 7x7 & Sigma: 5



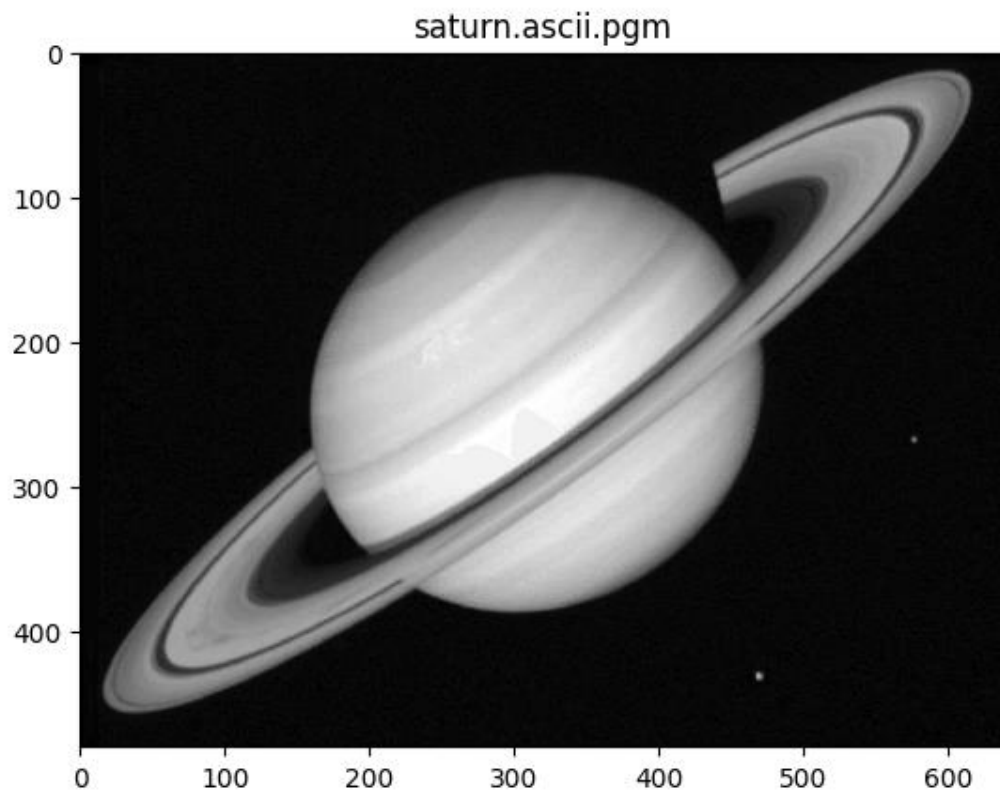
4.2.2. Sobel Filter



4.2.3. Sobel Filter to Most Blurry Image

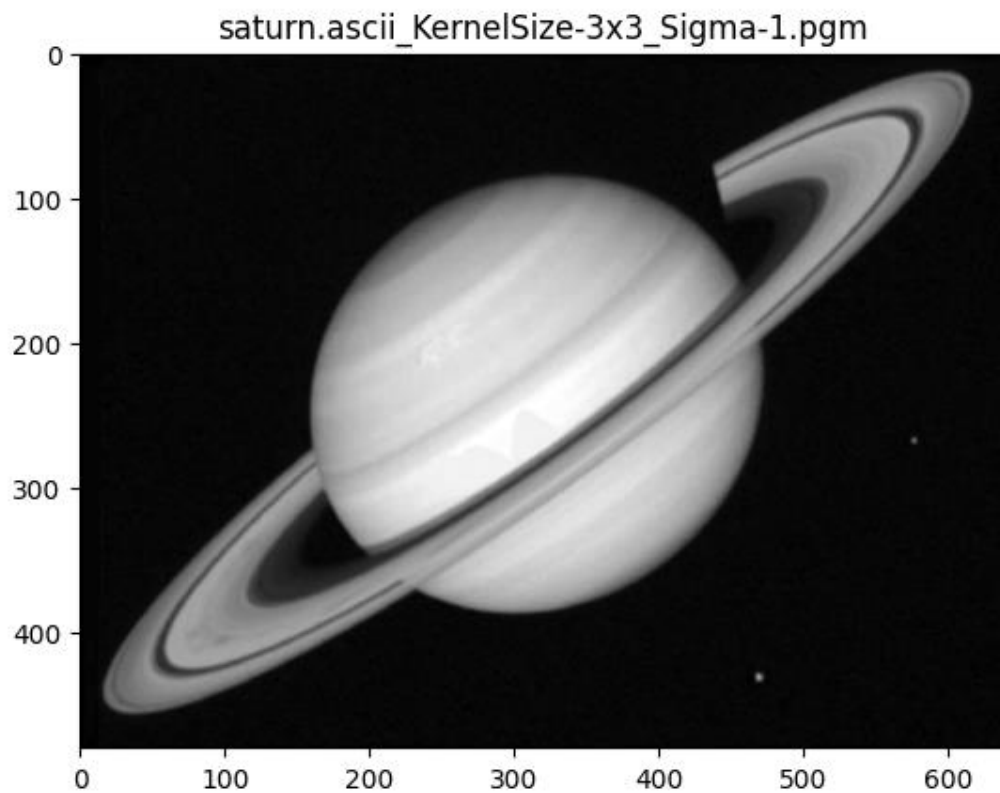


4.3. Image-3: saturn.ascii.pgm

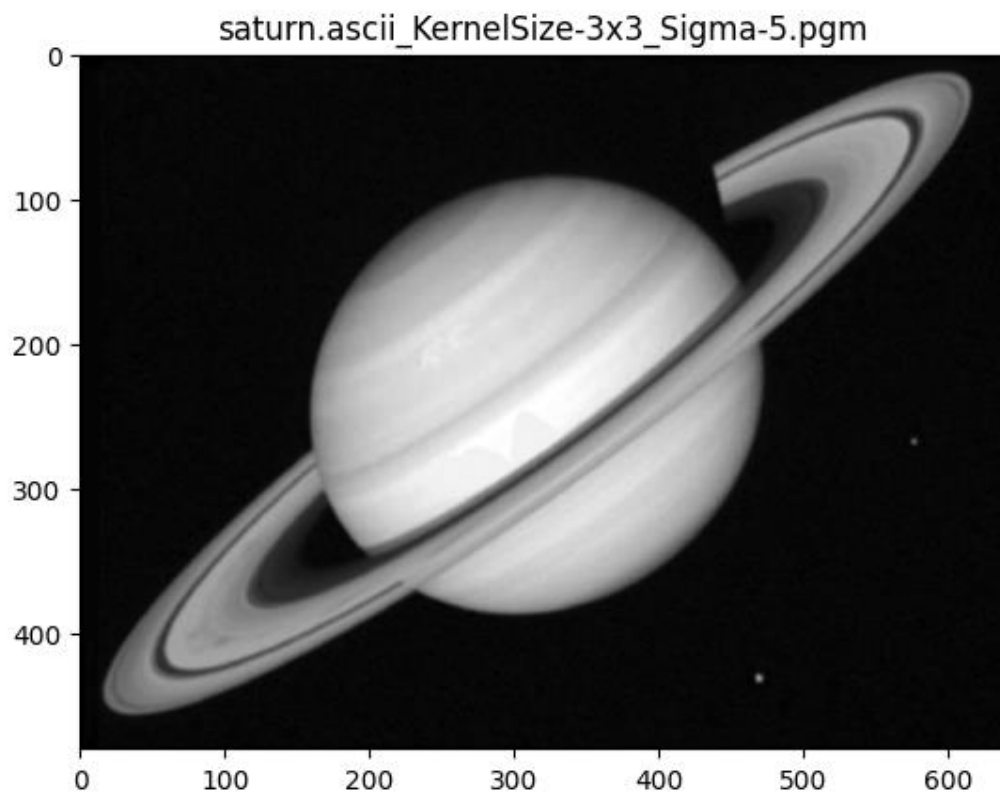


4.3.1. Gaussian Filter

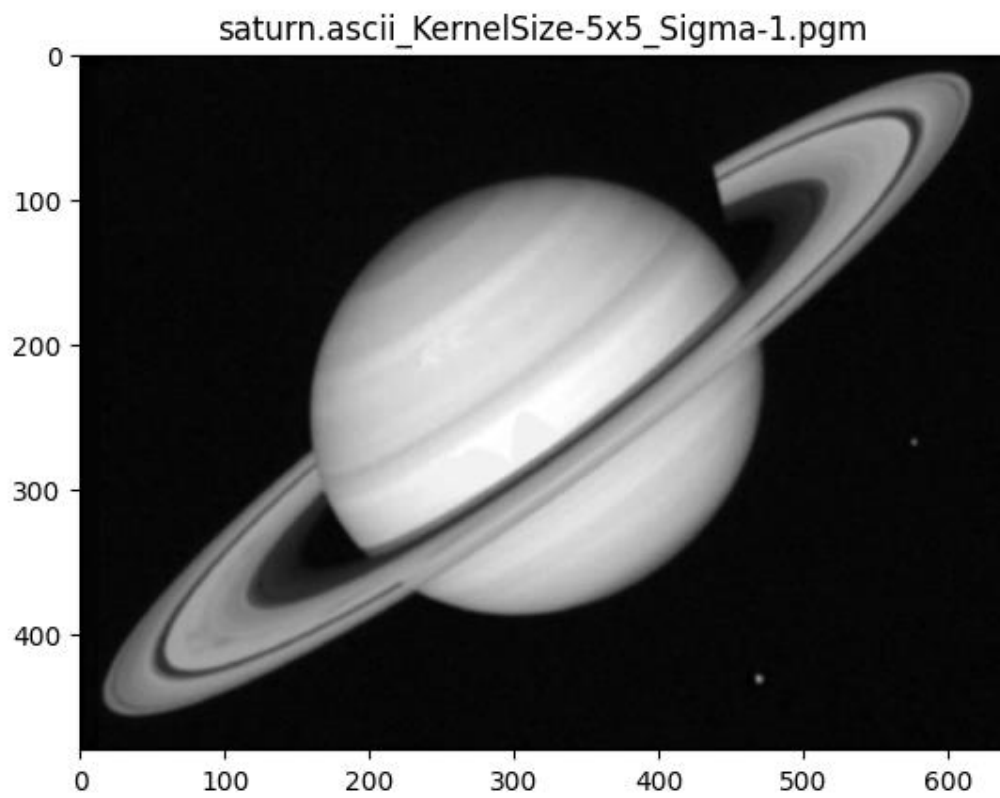
Kernel Size: 3x3 & Sigma: 1



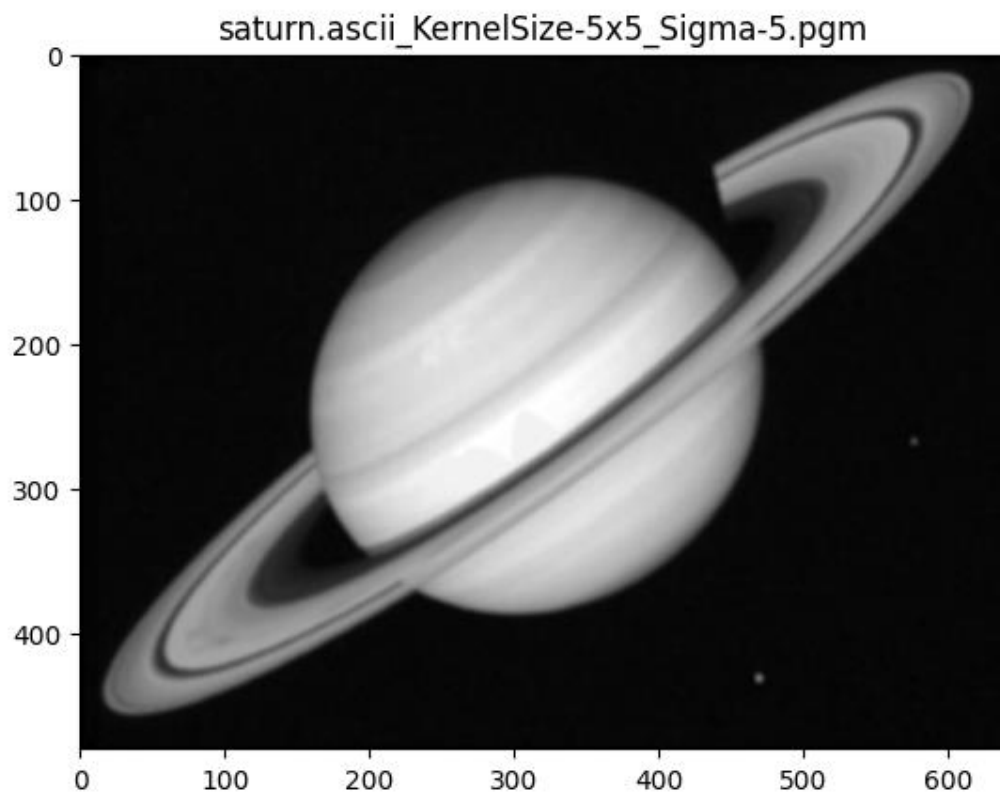
Kernel Size: 3x3 & Sigma: 5



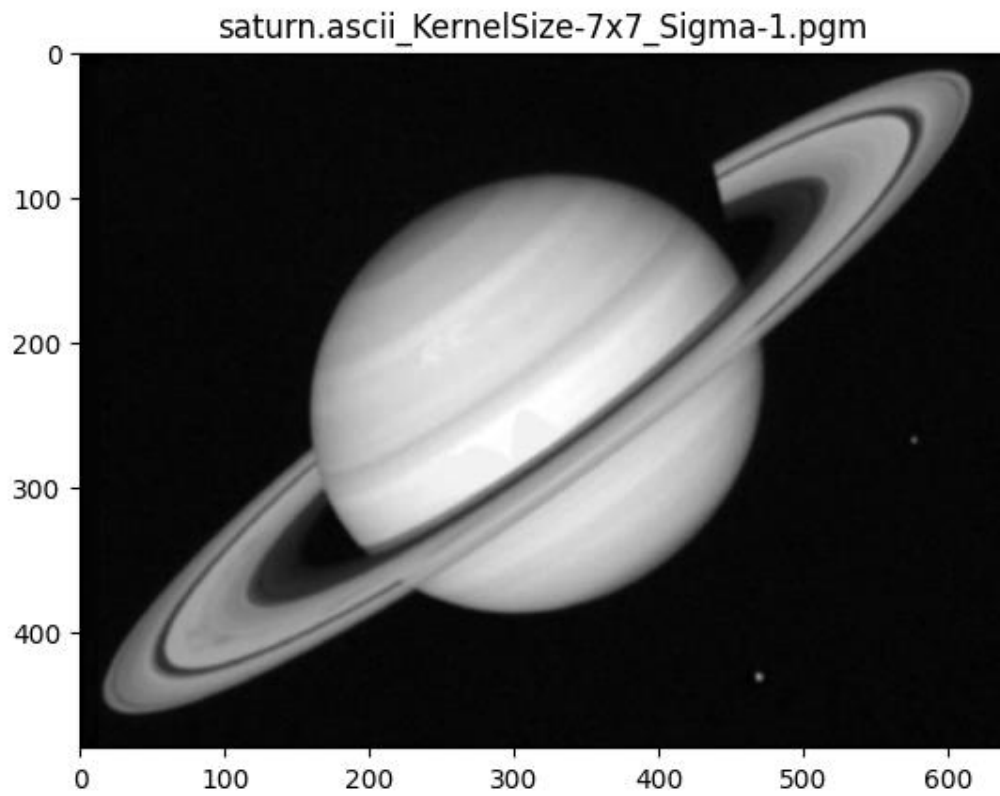
Kernel Size: 5x5 & Sigma: 1



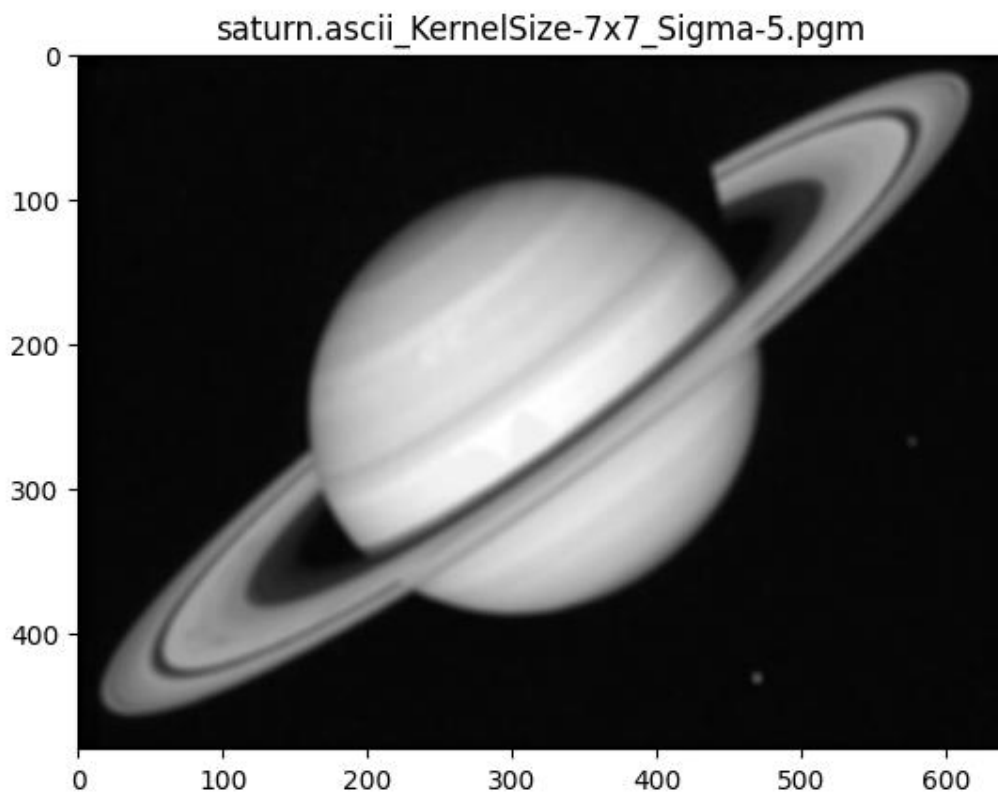
Kernel Size: 5x5 & Sigma: 5



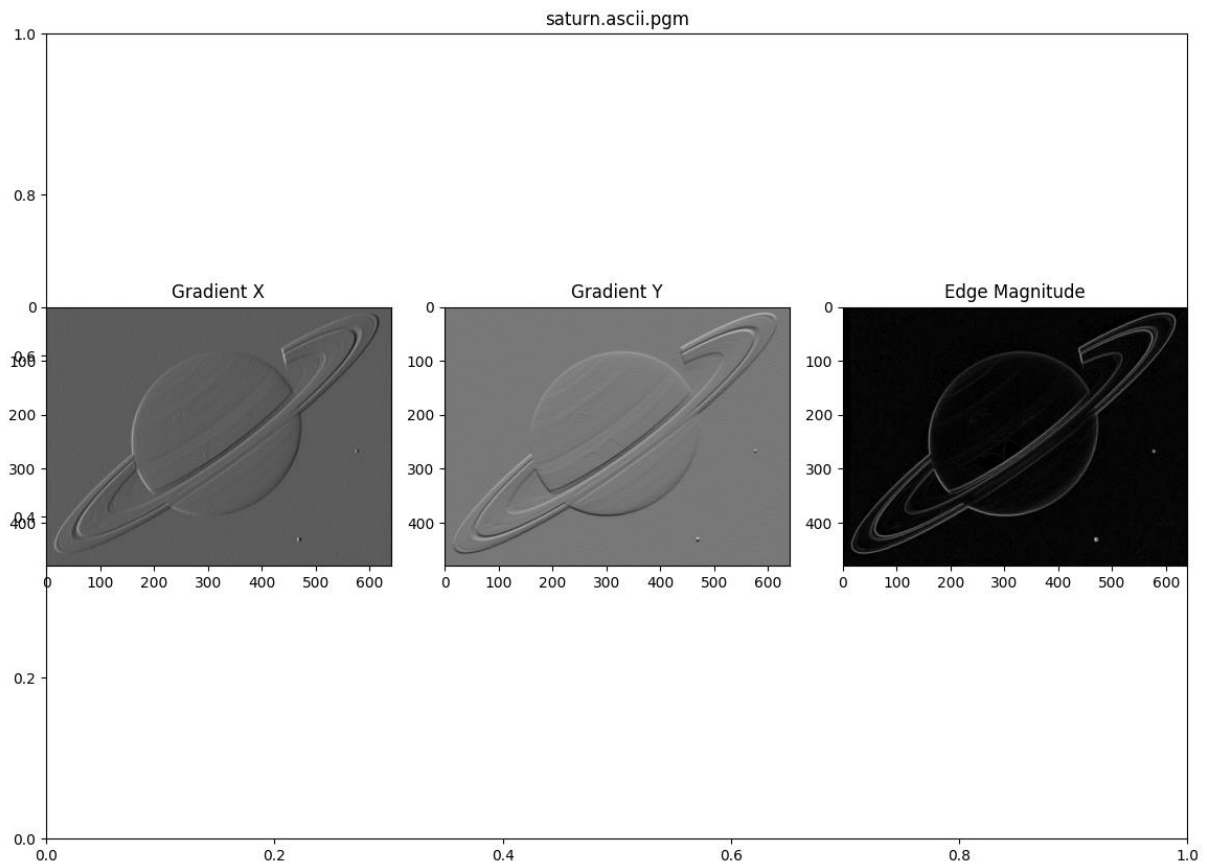
Kernel Size: 7x7 & Sigma: 1



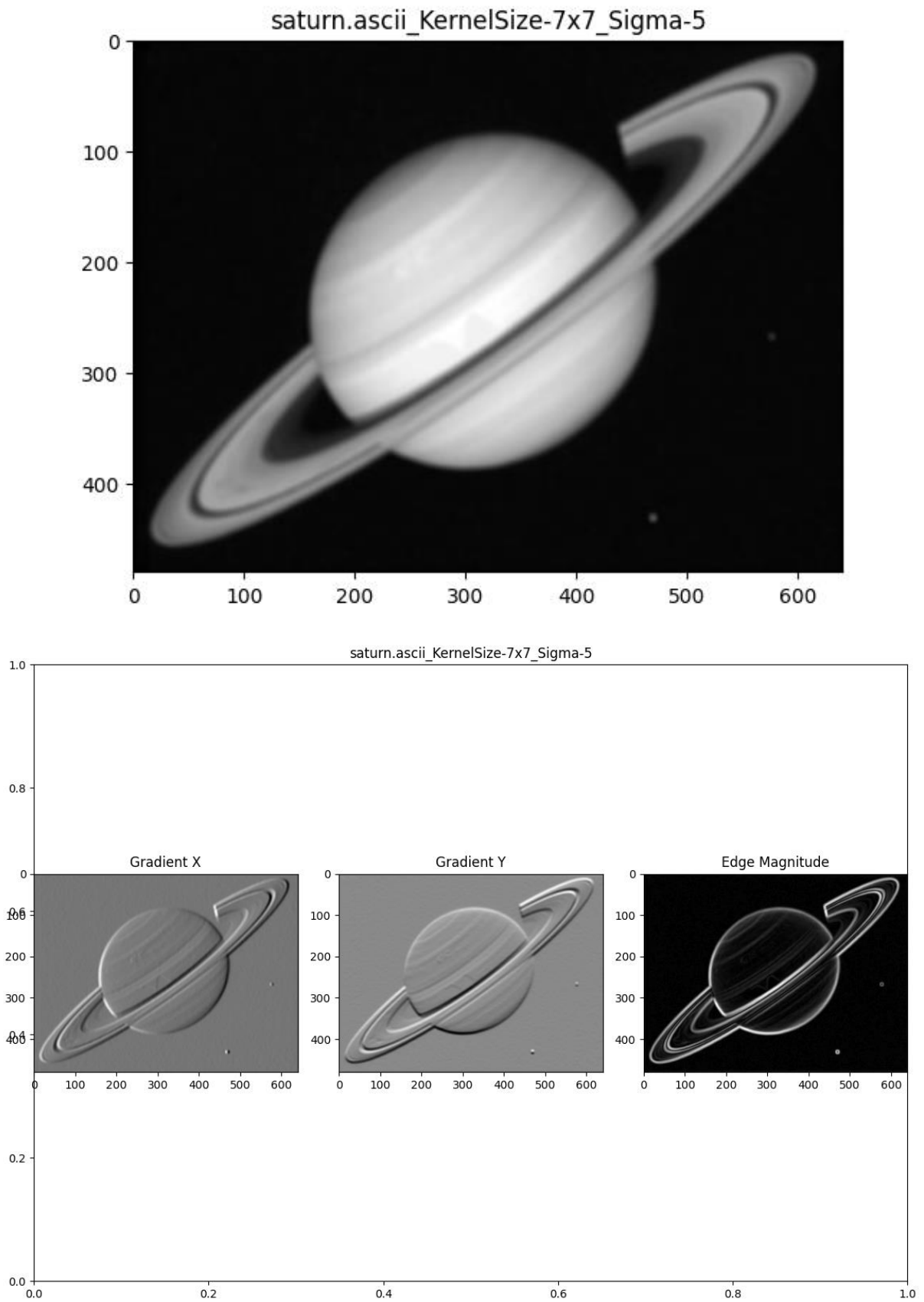
Kernel Size: 7x7 & Sigma: 5



4.3.2. Sobel Filter



4.3.3. Sobel Filter to Most Blurry Image



5. Sonuç

Bu çalışmada, gri renklerdeki 3 adet resme kenar belirleme işlemi uygulanmıştır. Kenar belirleme işlemi için öncelikle Gaussian Filter ile görüntü üzerinde bulanıklaştırma işlemi gerçekleştirilmiş, ardından Sobel Filter ile kenar algılama filtreleri kullanılarak kenarlar belirlenmiştir. Yapılan deneyler sonucunda filtre boyutları (3x3, 5x5 ve 7x7 olmak üzere) ve sigma (1 ve 5 standart sapma) değerleri olmak üzere 6 farklı gözlem yapılarak kenar belirleme işlemine olan etkileri incelenmiştir.

5.1. Bulanıklaştırma Filtre Boyutlarının Etkisi

Bulanıklaştırma aşamasında filtre boyutlarının değişmesinin kenar belirleme işlemine önemli bir etkisi olduğu gözlemlenmiştir. Küçük filtre boyutları (bu çalışmada en küçük 3x3) kullanıldığında, resimdeki küçük detaylar ve gürültüler daha iyi korunmuş, ancak kenarlar daha belirsiz ve pürüzlü hale gelmiştir. Öte yandan yine bu çalışmada kullanılan en büyük 7x7'lik filtre boyutları kullanıldığında, kenarlar daha belirgin hale gelmiş, ancak küçük detaylar kaybolmuştur. Bu durum, filtre boyutunun kenar belirleme işleminde bir denge sağlaması gerektiğini göstermektedir. İncelenen görüntülerde, 7x7 filtre boyutunun diğerlerine göre daha iyi olduğu ve kenarları belirginleştirdiği gözlemlenmiştir.

5.2. Bulanıklaştırma Sigma (s) Değerinin Etkisi

Bulanıklaştırma işlemi sırasında önemli bir diğer parametrenin sigma değeri olduğu ve kenar belirleme işlemine önemli bir etkisi olduğu tespit gözlemlenmiştir. Bu çalışmada 1 ve 5 sigma değerleri ile hareket edilmiş ve 1 sigma değeri için, görüntüdeki küçük detaylar ve gürültüler daha iyi korunmuş, ancak kenarlar daha az belirgin hale gelmiştir. Yine bir diğer 5 sigma değeri için ise kenarları belirginleştirmiş, ancak küçük detayları azaltmıştır. Bu çalışma da 2 ön tanımlı Sigma değeri ile gözlemler yapılmış ve 5 sigma değerinin 1 sigma değerine göre daha iyi bir denge sağladığı ve kenarların belirginleştirdiği raporlardaki görseller ile gözlemlenmiştir.

Sonuç olarak “Kenar Belirleme İşlemi” için bir ön adım olarak değerlendirilebilecek olan Bulanıklaştırma işleminin de filtre boyutlarının ve sigma değerlerinin önemli olduğunu gözlemlenmiştir. Burada bizim varsayılan olarak hareket ettiğimiz filtre boyutları ve sigma değerleri çalışma bazlı olarak denenip çıktılar üzerinde gözlem yapılarak seçilmesi uygun olacaktır.

5.3. Orijinal görüntüye Sobel filtresi uygulanması ile bulanıklaştırılmış görüntülere Sobel filtresi uygulanması sonucu arasındaki farkı değerlendiriniz.

Bu çalışmada, orijinal görüntüye uygulanan Sobel kenar belirleme filtresi ile önceden bulanıklaştırılmış görüntülere uygulanan Sobel filtresi sonuçları karşılaştırılmıştır. Sobel filtresi, kenarları belirlemek için sıklıkla kullanılan bir operatördür.

Bu çalışmada kullandığımız;

- ❖ Image-3: “saturn.ascii.pgm” görseli için orijinal görüntüye Sobel Filtresi uygulanan çıktı ile önce Gaussian Filter (Kernel Size:7x7 ve Sigma:5) ardından Sobel Filtresi uygulandıktan sonraki çıktıdaki kenarların keskinliği fark edilir şekilde olduğunu söyleyebilirim. (Bknz: 4.3.2 Sobel Filter ile 4.3.3 Sobel Filter to Most Blurry Image)
- ❖ Fakat Image-1: “coins.ascii.pgm” ve Image-2: “fruit.pgm” görsellerinde ise bunu net bir şekilde söylememiz mümkün değildir.