Soru2)

Soru 2’nin ilk kısmında RGB bir resme 3x3’lük bir Sobel filtresi uyguladık.

Girdi görüntüsü:



Sonuç ise şu şekilde(SobelOut.jpg):



ApplySobelFilter metodu ile resme Sobel filtresi uyguladık. Önce Gx ve Gy matrislerini tanımladık. Daha sonra bu filtreleri sırayla resmimize uyguladık. Bunun sonucunda 2 adet matris elde ettik. Biri Gx uyguladığımız, diğeri ise Gy. Sonrasında ise her bir (i, j) pikseli için Gx(i, j) ve Gy(i, y) değerlerinin karelerini alıp topladık. Elde ettiğimiz değerin kare kökü ise bize o pikselin Sobel değerini verdi.

Soru 2’nin ikinci kısmında ise LoG filtresi uyguladık. Öncelikle, verilen parametrelere göre bir log filtresi oluşturduk. Bu işlemi create\_log\_filter.m dosyasında yaptık. Bunu yaparken, matrisin tam ortasındaki değere parametre olarak (0, 0) verilmesine dikkat ettik.

Sonrasında ise oluşturduğumuz filtreyi ApplyLogFilter metodu ile resme uyguladık.

İnput resmimiz:



Sonuç ise (LogOut.jpg):



Soru3)

Bu soruda ise Seam Carving algoritmasını implement ettik.

Bunun için 4 adet fonksiyon yazdık. genislikAzalt(), genislikArtir(), yukseklikAzalt() ve yukseklikArtir().

genislikAzalt() fonksiyonunu inceleyelim. Öncelikle bir adet for loop tanımladık. Bu loop pikselSayisi parametresi ile verilen değer kadar dönecek ve her iterasyonda bir adet seam’i resimden çıkaracak. Sonrasında ise enerji fonksiyonu seçimi için bir if-else yazdık. Bu kısımlar 4 adet fonksiyonda da ortak. (Sadece artırma-azaltma kısımları farklı.)

Sonrasında Path’i hesapladık. Burada, dersteki algoritmayı kullanarak, bir Path hesapladık. Artırma fonksiyonlarında maximum yoğunluğa sahip değerlerden oluşan ve toplamı max olan bir path, azaltma fonksiyonlarında ise minimum toplama sahip bir path oluşturduk. Path’i oluştururken de komşulukları kullandık. Bir pikselden diğerine giderken komşuluklara bakarak gideceğimiz yere karar verdik.

Bu işlemlerden sonra artık yoğunluk değerleri toplama işlemi ile hesaplanmış oldu. Şimdi ise Path’i bulmamız gerekiyor. En alt satırdaki maximum değer bizim path’imizin son pikseli olacak. Path’i bulmak için geldiğimiz yoldan geri gideceğiz. Yine komşuluklara bakarak en üst satırdaki piksele ulaşmaya çalışacağız.

Burada yaptığımız işlem ise şu: Path’in o satırdaki hangi piksele karşılık geleceğini buluyoruz. Daha sonra o pikselin sağındaki tüm pikselleri bir sola kaydırıyoruz. Böylece o pikselin üzerine yazılmış ve piksel silinmiş oluyor. Bu işlemi bütün satırlar için tekrar ediyoruz.

Son olarak da, en sağdaki kolon artık gerekli değil. Bu kolonu resimden çıkarmamız lazım. Resmin kolon sayısını bir azaltarak bu işlemi gerçekleştiriyoruz.

Örnek girdi:



Çıktı resmi ise(IstanbulGenislikAzalt.jpg):

50 adet seam sildik, sonuç güzel görünüyor.

Bir başka örnek girdi:



Bunun çıktısı ise:



Yine 50 adet seam sildik.

Son örnek girdimiz:



Bunun çıktısı:



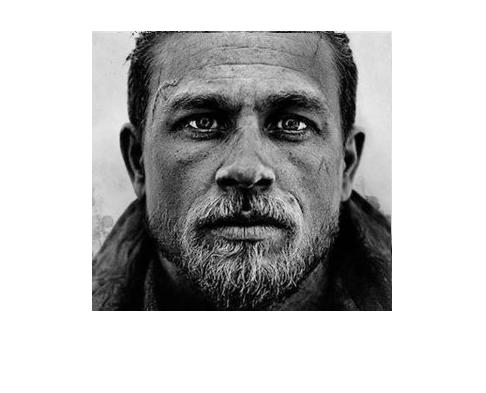
Bu resim daha küçük olduğu için 20 adet seam sildik.

yukseklikAzalt() fonksiyonunu inceleyecek olursak, burada genislikAzalt() fonksiyonundan farklı yaptığımız tek şey, resimde yukarıdan aşağı gitmek yerine sağdan sola gitmek. İmplementasyonda farklılık gösteren tek kısım bu. Onun dışında yaptığımız işlemler ve kullandığımız yöntemler aynı.

Örnek girdi:



Bu resmin çıktısı:



İkinci örnek girdi:



Bu resmin çıktısı ise:



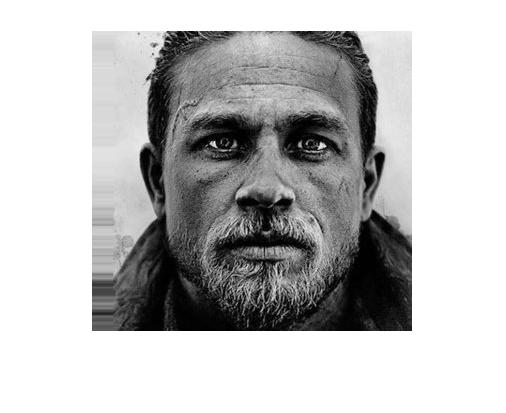
Azaltma işlemleri için hep minimum değerli pikselleri seçmiştik. Şimdi ise artırma işlemini yapacağız.

Önce genislikArtir() fonksiyonunu inceleyelim. Burada önceki fonksiyonlardan farklı olarak, komşuluklarda max değeri aradık. En son piksele vardığımızda ise yine max değeri aradık. Dönerken de komşuluklarda aradığımız şart yine max olma durumu oldu. Bu işlemler sonucu max toplam değere sahip bir path elde ettik. Bu Path’i bir yana kaydırdık. Bu işlemi artırmak istediğimiz piksel sayısı kadar tekrar ettik. Bu şekilde önceki resmin boyutlarını artırmış olduk.

Örnek resim:



Çıktısı:



Bir başka örnek resim:



Çıktı:



yukseklikArtir() fonksiyonunda ise, genişlikArtir() fonksiyonundan farklı olarak, işlemi yukarıdan aşağı değil de, sağdan sola yaptık.

Örnek resim(ocean.jpg):

Çıktısı(OceanYukseklikArtir):



Bu resimde göze güzel gelen bir sonuç elde ettik.

Bir başka örnek resim:



Çıktı:



Bu çıktı ise çok güzel değil

Artırma işlemlerinin sonucuna baktığımızda hep kenardaki kolon ya da satırın seçilmiş olduğunu gördük. Bunun sebebini ise, resme yaptığımız zero padding olarak düşündüm. Siyah renkle yapılmış bir çerçeve, kenardaki piksellerin yoğunluğunu artıracak ve max path hep çerçeve yanındaki pikseller olacak. Max değerinin hiç değişmemesinden dolayı sürekli aynı seam tekrar edildi ve resmin boyutu aynı seam ile arttı.