

MATLAB TABANLI GÖRÜNTÜ İŞLEME UYGULAMALARI VE FİLTRELEME TEKNİKLERİ

MATLAB BASED IMAGE PROCESSING APPLICATIONS AND FILTERING TECHNIQUES

Arda Can İREN

Zonguldak Bülent Ecevit Üniversitesi Elektrik Elektronik Mühendisliği Zonguldak, Türkiye

Acan.iren@mf.karaelmas.edu.tr

ÖZET

Dijital Görüntü İşleme, dijital bir görüntünün bilgisayar aracılığıyla işlenerek iyileştirilmesini amaçlar. Bu amaç uğruna geliştirilmiş bir takım filtreleme teknikleri bulunmaktadır. Bu çalışmada görüntüyü iyileştirmek adına sık kullanılan filtrelerin başlangıçtaki görüntüyü nasıl değiştirdiği açıklanmış ve Matlab ortamında elde edilen çıktılar paylaşılmıştır. Bu sayede insanların elde etmeyi hedefledikleri görüntü için filtreler hakkında kazandığı bilgi birikimi sayesinde seçme konusunda karar verebilecek yetkinliğe ulaşabilmesi amaçlanmıştır.

ANAHTAR KELİMELEER: Dijital görüntü işleme, Matlab, Filtreleme

ABSTRACT

Digital Image Processing aims to improve a digital image by processing it through a computer. There are a number of filtering techniques developed for this purpose. In this study, it is explained how frequently used filters change the initial image in order to improve the image, and the outputs obtained in the Matlab environment are shared. In this way, it is aimed that people can reach the competence to decide on the choice thanks to the knowledge they have gained about the filters for the image they aim to achieve.

KEYWORDS: Digital image processing, Matlab, Filtering

1.GİRİŞ

Dijital Görüntü İşleme, dijital görüntünün bir dijital bilgisayar aracılığıyla işlenmesi anlamına gelir. Gelişmiş görüntü elde etmek veya bazı yararlı bilgileri çıkarmak için bilgisayar algoritmalarının kullanılması olduğunu da söyleyebiliriz. Dijital görüntü işleme, dijital görüntüleri işlemek ve analiz etmek için algoritmaların ve matematiksel modellerin kullanılmasıdır. Dijital görüntü işlemenin amacı, görüntülerin kalitesini artırmak, görüntülerden anlamlı bilgiler çıkarmak ve görüntü tabanlı görevleri otomatik hale getirmektir. Dijital görüntü işleme sürecinde önemli bir takım adımlar vardır. Bunlar;

1.Görüntü alma: Bu, bir dijital kamera veya tarayıcı kullanarak bir görüntünün yakalanmasını veya mevcut bir görüntünün bir bilgisayara aktarılmasını içerir.

2. Görüntü geliştirme: Bu, kontrastı artırma, gürültüyü azaltma ve yapay bozuklukları giderme gibi bir görüntünün görsel kalitesini iyileştirmeyi içerir.

3.Görüntü restorasyonu: Bu, görüntüdeki bulanıklık, gürültü ve bozulma gibi bozulmaların giderilmesini içerir.

4.Görüntü bölütleme: Bu, bir görüntüyü, her biri görüntüdeki belirli bir nesneye veya özelliğe karşılık gelen bölgelere veya parçalara ayırmayı içerir.

5.Görüntü temsili ve açıklaması: Bu, bir görüntünün bir bilgisayar tarafından analiz edilebilecek ve manipüle edilebilecek şekilde temsil edilmesini ve bir görüntünün özelliklerinin kompakt ve anlamlı bir şekilde açıklanmasını içerir.

6.Görüntü analizi: Bu, nesneleri tanıma, kalıpları algılama ve özellikleri niceleme gibi bir görüntüden bilgi çıkarmak için algoritmaların ve matematiksel modellerin kullanılmasını içerir.

7.Görüntü sentezi ve sıkıştırma: Bu, depolama ve iletim gereksinimlerini azaltmak için yeni görüntüler oluşturmayı veya mevcut görüntüleri sıkıştırmayı içerir. Dijital görüntü işleme, tıbbi görüntüleme, uzaktan algılama,

bilgisayar görüşü ve multimedya gibi çeşitli uygulamalarda yaygın olarak kullanılmaktadır.

1.1 Görüntü Kavramı

Dijital görüntü işlemi uygulamalarına geçmeden önce görüntünün ne olduğunun iyice anlaşılması gerekmektedir. Bir görüntü iki boyutlu bir fonksiyon $F(x,y)$ olarak tanımlanır Burada x ve y uzamsal koordinatlarıdır ve F 'nin herhangi bir koordinat çiftindeki (x,y) genliği o görüntünün o andaki yoğunluğu olarak adlandırılır. F 'nin x,y ve genlik değerleri sonlu olduğunda buna sayısal görüntü denir. Başka bir deyişle, bir görüntü, satırlar ve sütunlar halinde özel olarak düzenlenmiş iki boyutlu bir dizi ile tanımlanabilir. Dijital Görüntü, her biri belirli bir konumda belirli bir değere sahip olan sınırlı sayıda öğeden oluşur. Bu öğelere resim öğeleri, görüntü öğeleri ve pikseller denir.

1.2 Görüntü Tipleri

1. İkili Görüntü (Binary Image): Adından da anlaşılacağı gibi ikili görüntü, yalnızca iki piksel öğesi içerir, yani 0 ve 1, burada 0 siyahı ve 1 beyazı ifade eder. Bu görüntü aynı zamanda Tek Renkli olarak da bilinir.

2. Siyah Beyaz Görüntü (Black and White Image): Sadece siyah ve beyaz renkten oluşan görüntüye SİYAH BEYAZ GÖRÜNTÜ denir.

3. 8 Bit Renk Formatı: En ünlü görüntü formatıdır. İçinde 256 farklı renk tonu vardır ve yaygın olarak Gri Tonlamalı Görüntü olarak bilinir. Bu formatta 0 Siyahı, 255 beyazı ve 127 griyi temsil eder.

4.16 Bit Renk Formatı: Renkli bir görüntü formatıdır. İçinde 65.536 farklı renk barındırır. Yüksek Renk Formatı olarak da bilinir. Bu formatta renk dağılımı, Gri Tonlamalı görüntü ile aynı değildir. 16 bitlik bir format aslında Kırmızı, Yeşil ve Mavi olmak üzere üç ayrı formata bölünmüştür.

Bu çalışmanın literatürde bulunan diğer araştırmalara göre başarılı olmasını sağlayacak yanı bu çalışmaya bakan herkesin filtreleme konusunda edineceği bilgi sayesinde hedefledikleri görüntüye ulaşmada doğru seçimleri yapabilmelerini sağlayacak bilinci kazanmaları amaçlanmıştır. [1]

2. BENZER ÇALIŞMALAR

Bu çalışmaya benzer literatürdeki çalışmalardan birisi Dr E. Karthikeyan ve arkadaşlarının 2019 yılında yapmış oldukları

“Comparative Analysis of Various Filtering Techniques In Image Processing” [2] adlı makalesidir. Karthikeyan ve arkadaşlarının bu çalışmasında görüntüleri gürültüden arındırmak adına en iyi filtre seçimini yapmayı hedeflemişlerdir ve bu konuda karşılaştırmalarda bulunmuşlardır. Yapmış olduğumuz çalışmada ise karşılaştırma arzumuz benzer taraflarımıza örnek verilebilir. Fakat bu çalışma ile birlikte hedeflediğimiz amacımız diğer çalışmanın aksine gürültüyü giderecek en iyi yöntemi bulmak değil, güzel bir görüntü hedefleyen kişilere kolayca filtrelerin farklılıklarını karşılaştırabilmelerini sağlayarak en doğru filtreyi seçmelerinde yardımcı olmaktır.

3. METOD VE YÖNTEMLER

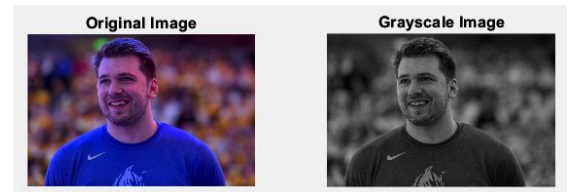
Bu bölümde farklı farklı filtreler seçilip her birinin Matlab ortamında görüntü işleme sayesinde iyileştirilmiş halleri paylaşılacaktır. Bu sayede en çok beğenilen filtrenin bulunması hedeflenmektedir.

3.1.Binary (İkili) Görüntü: Yalnızca iki benzersiz piksel yoğunluğu değerine sahip görüntüler - 0 (siyahı temsil eder) ve 1 (beyazı temsil eder) ikili görüntüler olarak adlandırılır. Bu tür görüntüler genellikle renkli bir görüntünün ayırt edici bir bölümünü vurgulamak için kullanılır. Burada örnek olarak seçtiğimiz basketbolcu resminin değişimi şekil 1’de gösterilmektedir. [3]



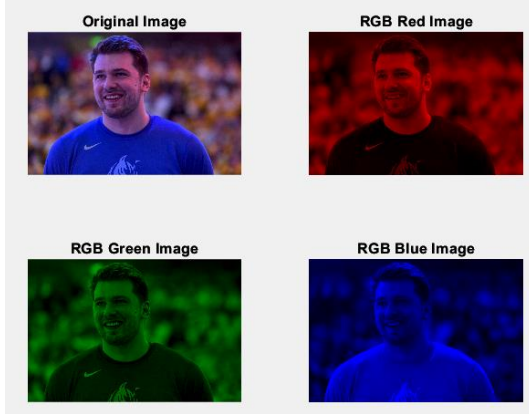
Şekil 1(Binary Image)

3.2 Gri Seviyeli(Grayscale) Görüntü: Dijital fotoğrafçılıkta kullanılan bir görüntü dönüştürme tekniğidir. Her türlü renk bilgisini ortadan kaldırır ve yalnızca grinin farklı tonlarını bırakır; en parlakı beyaz ve en koyusu siyahtır. Burada örnek olarak seçmiş olduğumuz basketbolcu resminin değişimi şekil 2’de gösterilmektedir. [4]



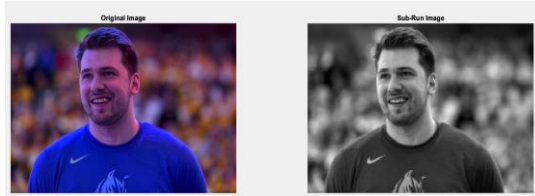
Şekil 2(Grayscale Image)

3.3 RGB Görüntü: Bazen gerçek renkli görüntü olarak adlandırılan bir RGB görüntüsü, her bir piksel için kırmızı, yeşil ve mavi renk bileşenlerini tanımlayan bir m'ye n'ye 3 veri dizisi olarak depolanır. Bu görüntünün elde edilmesinde resmin önce gri seviyeli görüntüye dönüştürülmesi önemlidir. Basketbolu fotoğrafının 3 farklı rgb renkteki görünümü şekil 3'de gösterilmektedir. [5]



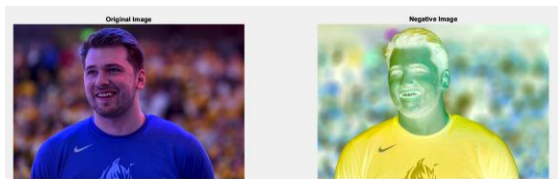
Şekil 3 (RGB Red, Green and Blue Images)

3.4 Alt Örnekleme(Sub Run) Görüntü: Sayısal imgede örnekleme yapılarak imgenin çözünürlüğü değiştirilebilir. Şekil 4'de örnek olarak aldığımız resmin gri seviyeli tonunun alt örnekleme sonucu pixel sayısının az hali gösterilmektedir. [6]

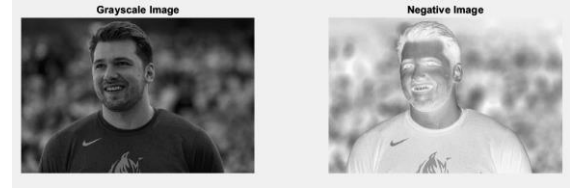


Şekil 4 (Sub-Run Image)

3.5 Negatif Görüntü: Bir görüntünün negatifi, orijinal görüntüdeki 'i' yoğunluğunun 'i-1' ile değiştirilmesiyle elde edilir, yani en koyu pikseller en parlak ve en parlak pikseller en koyu olur. Görüntü negatifi, her bir pikselin maksimum yoğunluk değerinden çıkarılmasıyla üretilir. Şekil 5'da ve şekil 6'da buna örnek olarak basketbolcu resminin renkli halinin ve gri tonlamalı halinin negatifi gösterilmektedir. [7]



Şekil 5 (Negative Image-Colored)



Şekil 6(Negative Image-Colorless)

3.6 Kontrastı Değiştirilmiş Görüntü: Bu kısımda görüntünün farklı kontrast seviyesindeki değişimleri gösterilecektir. Şekil 7'de yüksek kontrasta, şekil 8'da ise düşük kontrasta sahip görüntüler gösterilmektedir. [8]



Şekil 7(High Contrast Image)



Şekil 8 (Low Contrast Image)

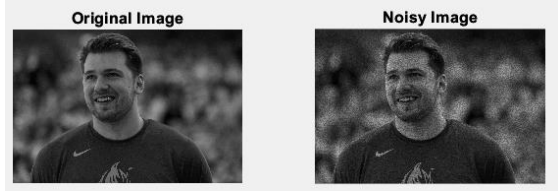
3.7 Tuz ve Biber Gürültülü (Salt and Pepper Noise) Görüntü: Darbe gürültüsü olarak da bilinen tuzlu ve biberli gürültü, bazen dijital görüntülerde görülen bir gürültü biçimidir. Bu gürültü, görüntü sinyalindeki keskin ve ani bozulmalardan kaynaklanabilir. Kendisini seyrek olarak oluşan beyaz ve siyah pikseller olarak sunar. Şekil 9'da bu görüntüye örnek gösterilmiştir. [9]



Şekil 9 (Salt and Pepper Noised Image)

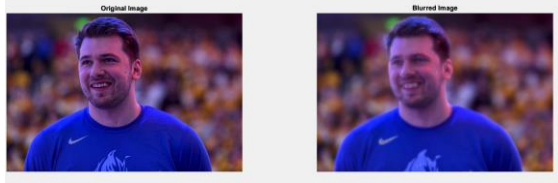
3.8 Benekli Görüntü: Benek görüntüleme, atmosferik türbülansın varyasyonunu donduran çok sayıda kısa pozlamanın analizine dayanan bir dizi yüksek çözünürlüklü astronomik görüntüleme tekniği içerir. Kaydır ve ekle ('görüntü istifleme') yöntemine ve benek interferometri yöntemlerine ayrılabilirler. Bu teknikler, yer tabanlı teleskopların

çözünürlüğünü önemli ölçüde artırabilir, ancak parlak hedeflerle sınırlıdır. Benekli görüntüye örnek görüntü şekil 10'da gösterilmiştir. [10]

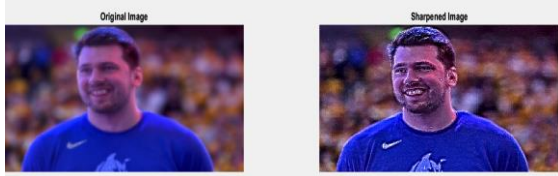


Şekil 10(Benekli-Speckle Gürültülü Görüntü)

3.9 Bulanıklaştırılmış ve Netleştirilmiş (Blurred and Sharpened) Görüntü: Bulanıklaştırma bir şeyi daha az net veya belirgin hale getirmektir. Bu, görüntü analizi bağlamında oldukça geniş bir şekilde yorumlanabilir. Bir görüntünün ayrıntılarını azaltan veya bozan her şey geçerli olabilir. Buna örnek olarak şekil 11'de görüntü gösterilmiştir. Ayrıca bulanıklaştırılmış görüntünün netleştirilmesinin gösterildiği görüntü de şekil 12'ye eklenmiştir. [11]



Şekil 11 (Blurred Image)



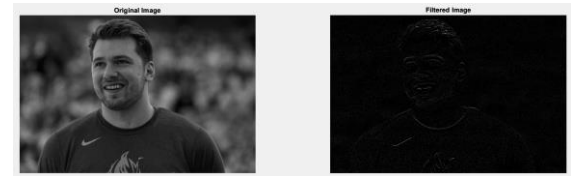
Şekil 12 (Sharpened Image)

3.10 İdeal Alçak Geçiren Filtreli Görüntü: Alçak geçiren filtre çoğu yumuşatma yönteminin temelidir. Bir görüntü, yakındaki piksellerin ortalaması alınarak piksel değerleri arasındaki eşitsizlik azaltılarak yumuşatılır. Düşük geçiş filtresi kullanmak, yüksek frekans bilgisini azaltırken, görüntü içindeki düşük frekans bilgisini koruma eğilimindedir. Şekil 13'de basketbolcu fotoğrafına ideal alçak geçiren filtre uygulandığındaki değişimini gösteren görüntüler paylaşılmıştır. [12]



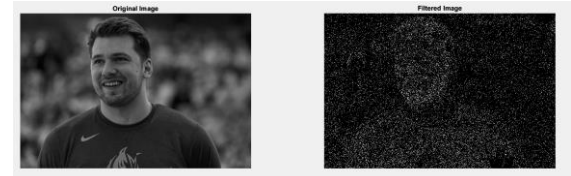
Şekil 13(Ideal Low Pass Filtered Image)

3.11 Sobel Filtreli Görüntü: Sobel filtresi, kenar tespiti için kullanılır. Görüntü içindeki her pikselde görüntü yoğunluğunun gradyanını hesaplayarak çalışır. Işıktan karanlığa doğru en büyük artışın yönünü ve bu yöndeki değişim oranını bulur. Şekil 14'de sobel filtreli görüntüye örnek gösterilmiştir. [13]



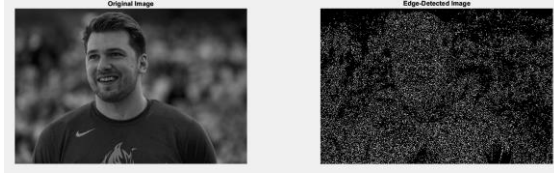
Şekil 14(Sobel Filtered Image)

3.12 Marr Hildreth Filtreli Görüntü: Bir başka kenar belirleme filtresi de marr hildreth filtresidir. Laplacian of Gaussian (LoG) filtresi olarak da bilinen Marr-Hildreth filtresi, görüntü işlemede kullanılan bir kenar algılama filtresidir. Bir görüntüdeki kenarları tanımlamak için görüntü yumuşatma ve kenar algılama kavramlarını birleştirir. Şekil 15'de bu filtresinin kullanımına bir örnek gösterilmiştir. [14]



Şekil 15(Marr Hildreth Edge Detection Image)

3.13 Canny Kenar Tespit Edici Filtre: Canny kenar algılama algoritması, görüntülerdeki kenarları algılamak için popüler bir yöntemdir. 1986 yılında John Canny tarafından geliştirilmiştir ve bilgisayarlı görü ve görüntü işleme uygulamalarında yaygın olarak kullanılmaktadır. Canny kenar algılama algoritmasının çıktısı, kenar piksellerin beyaz (veya 1) olarak işaretlendiği ve kenar olmayan piksellerin siyah (veya 0) olarak işaretlendiği ikili bir görüntüdür. Şekil 16'da bu filtresinin uygulanmasına bir örnek gösterilmiştir. [15]



Şekil 16(Canny Edge Detection Image)

3.14. Boyutu Değiştirilmiş (Resized)

Görüntü: Görüntü boyutları, piksel genişliği ve yüksekliği olarak ifade edilir. Resizing işlemi, bir görüntünün boyutlarını orantılı olarak büyütme veya küçültme için kullanılır. Görüntü yeniden boyutlandırma işlemi sırasında, piksel değerleri veya içeriği değiştirilmez, sadece boyutlar değiştirilir. Bu işlem, görüntünün farklı bir ekranda veya çıktı ortamında doğru boyutta görüntülenmesini sağlamak veya görüntü işleme algoritmalarının uygulanması için uygun boyutları sağlamak amacıyla yapılır. Şekil 17'de bu filtrenin uygulanmasına bir örnek gösterilmiştir. [16]



Şekil 17(Resized Image)

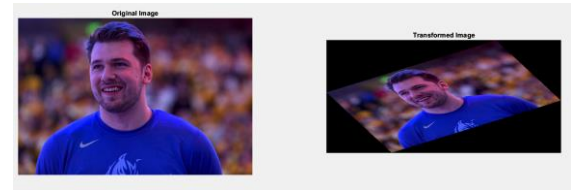
3.15 Ayna filtreli Görüntü: Bu filtre, bir görüntünün yatay veya dikey olarak aynalanmasını sağlar, yani görüntünün sol ve sağ yönlerini veya üst ve alt yönlerini birbirine yansıtır. Ayna filtresi, her pikselin değerini, yatay veya dikey yönde simetrik bir konumda bulunan pikselin değeriyle değiştirir. Yatay ayna filtresi uygulandığında, görüntüyü yatay olarak yansıtırken, dikey ayna filtresi uygulandığında görüntüyü dikey olarak yansıtır. Bu filtreler, görüntünün simetrisini kırmadan manipülasyon yapmak veya görsel etkiler oluşturmak için kullanılabilir. Şekil 18'de bu filtrenin normal resme uygulandığındaki etkisi gösterilmektedir. [17]



Şekil 18(Mirrored Image)

3.16 Uzamsal Dönüşüme Uğramış Görüntü:

Uzamsal dönüşüme uğramış bir görüntü, görüntü işleme alanında bir dönüşüm işlemine tabi tutulmuş bir görüntüyü ifade eder. Bu dönüşüm, görüntünün konumunu, boyutunu veya yönelimini değiştirir. Uzamsal dönüşümler, bir görüntünün orijinal koordinat düzlemi üzerinde farklı bir konuma, boyuta veya yöne taşınmasını sağlar. Bu dönüşümler, geometrik dönüşümler olarak da adlandırılabilir ve bir görüntünün transformasyonunu temsil eder. Bu dönüşüm şekil 19'da gösterilmektedir. [18]

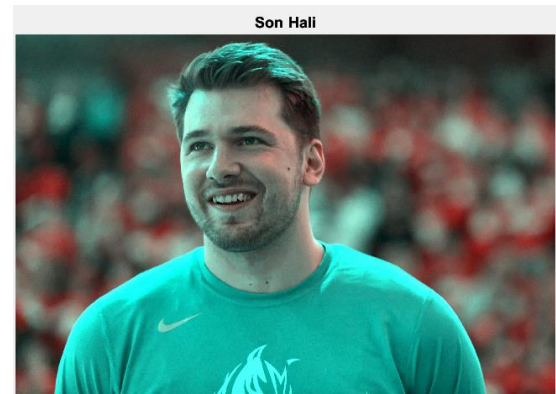


Şekil 19(Transformed Image)

4. SONUÇLAR

Bu çalışmada farklı filtrelerin özellikleri ve normal fotoğrafa yaptıkları değişiklikler gösterilmiştir. Bu çalışmadaki amaç kesin doğruyu bulmak yerine insanların kendi doğrularını seçmelerinde yardımcı olmaktır.

Bu çalışma sonucu kendi ayarlarım ile birlikte resmin tamamını gri yapmadan oyuncunun üzerindeki tişörtün renginin değiştirilmesi amaçlanmıştır. Bu amaçla ortaya çıkan görüntü şekil 20'de gösterilmektedir.



Şekil 20 (Son Hali)

YORUM

Bu çalışmanın görüntü işleme uygulamalarında filtrelerin karşılaştırılması adına yararlı ve faydalı olduğu düşünülmektedir. Her bir filtrenin tanıtımının yapılmış ve sonuçlarının da paylaşılmış olması kendine uygun bir filtre

tasarlamak isteyen kişiler için çok faydalı bir kılavuz olma niteliği kazanmıştır.

EKLER

1. İkili Görüntü (Binary Image)

```
originalImage =  
imread('basketbolcu.jpg');  
if size(originalImage, 3) > 1  
    originalImage =  
    rgb2gray(originalImage);  
end  
threshold = graythresh(originalImage);  
binaryImage =  
imbinarize(originalImage, threshold);  
subplot(1, 2, 1);  
imshow(originalImage);  
title('Original Image');  
subplot(1, 2, 2);  
imshow(binaryImage);  
title('Binary Image');
```

3.2 Gri Seviyeli(Grayscale) Görüntü

```
originalImage =  
imread('basketbolcu.jpg');  
grayImage = rgb2gray(originalImage);  
subplot(1, 2, 1);  
imshow(originalImage);  
title('Original Image');  
subplot(1, 2, 2);  
imshow(grayImage);  
title('Grayscale Image');
```

3.3 RGB Görüntü

```
originalImage =  
imread('basketbolcu.jpg');  
redImage = originalImage;  
redImage(:, :, 2:3) = 0;  
greenImage = originalImage;  
greenImage(:, :, [1 3]) = 0;  
blueImage = originalImage;  
blueImage(:, :, 1:2) = 0;  
subplot(2, 2, 1);  
imshow(originalImage);  
title('Original Image');  
subplot(2, 2, 2);  
imshow(redImage);  
title('RGB Red Image');  
subplot(2, 2, 3);  
imshow(greenImage);  
title('RGB Green Image');
```

```
subplot(2, 2, 4);  
imshow(blueImage);  
title('RGB Blue Image');
```

3.4 Alt Örnekleme(Sub Run) Görüntü

```
originalImage =  
imread('basketbolcu.jpg');  
subSize = 16;  
  
grayImage = rgb2gray(originalImage);  
  
numSubRows = floor(size(grayImage, 1)  
/ subSize);  
numSubCols = floor(size(grayImage, 2)  
/ subSize);  
subRunImage = zeros(numSubRows,  
numSubCols);  
for row = 1:numSubRows  
    for col = 1:numSubCols  
        subRegion = grayImage((row-  
1)*subSize+1 : row*subSize, (col-  
1)*subSize+1 : col*subSize);  
        subRunImage(row, col) =  
        mean(subRegion(:));  
    end  
end  
subplot(1, 2, 1);  
imshow(originalImage);  
title('Original Image');  
subplot(1, 2, 2);  
imshow(subRunImage, []);  
title('Sub-Run Image');
```

3.5 Negatif Görüntü

% Renkli için

```
originalImage =  
imread('basketbolcu.jpg');  
  
negativeImage = 255 - originalImage;  
subplot(1, 2, 1);  
imshow(originalImage);  
title('Original Image');  
subplot(1, 2, 2);  
imshow(negativeImage);  
title('Negative Image');
```

.....

% Gri seviyeli için

```
originalImage =  
imread('basketbolcu.jpg');  
grayImage = rgb2gray(originalImage);  
negativeImage = 255 - grayImage;  
subplot(1, 3, 1);  
imshow(originalImage);  
title('Original Image');  
subplot(1, 3, 2);  
imshow(grayImage);  
title('Grayscale Image');  
subplot(1, 3, 3);  
imshow(negativeImage);  
title('Negative Image');
```

3.6 Kontrastı Değiştirilmiş Görüntü

% Yüksek kontrast için(karşıtlık oranı yüksek)

```
originalImage =  
imread('basketbolcu.jpg');  
highContrastImage =  
imadjust(originalImage, [], [], 2);  
subplot(1, 2, 1);  
imshow(originalImage);  
title('Original Image');  
subplot(1, 2, 2);  
imshow(highContrastImage);  
title('High Contrast Image');
```

.....
% Düşük kontrast için(karşıtlık oranı düşük)

```
originalImage =  
imread('basketbolcu.jpg');  
lowContrastImage =  
imadjust(originalImage, [], [], 0.5);  
subplot(1, 2, 1);  
imshow(originalImage);  
title('Original Image');  
subplot(1, 2, 2);  
imshow(lowContrastImage);  
title('Low Contrast Image');
```

3.7 Tuz ve Biber Gürültülü (Salt and Pepper Noise) Görüntü

```
originalImage =  
imread('basketbolcu.jpg');  
if size(originalImage, 3) > 1  
    originalImage =  
    rgb2gray(originalImage);  
end  
noiseDensity = 0.1;  
noisyImage = imnoise(originalImage,  
    'salt & pepper', noiseDensity);  
subplot(1, 2, 1);  
imshow(originalImage);  
title('Original Image');  
subplot(1, 2, 2);  
imshow(noisyImage);  
title('Noisy Image');
```

3.8 Benekli Görüntü

```
originalImage =  
imread('basketbolcu.jpg');  
if size(originalImage, 3) > 1  
    originalImage =  
    rgb2gray(originalImage);  
end  
noiseAmount = 0.02;  
noisyImage = imnoise(originalImage,  
    'speckle', noiseAmount);
```

```
subplot(1, 2, 1);  
imshow(originalImage);  
title('Original Image');  
subplot(1, 2, 2);  
imshow(noisyImage);  
title('Speckle Image');
```

3.9 Bulanıklaştırılmış (Blurred) Görüntü

% Blurlama

```
originalImage =  
imread('basketbolcu.jpg');
```

```
blurRadius = 15;  
blurredImage =  
imgaussfilt(originalImage,  
    blurRadius);  
blurAmount = 2.0;  
blurredImage = originalImage +  
    blurAmount * (blurredImage -  
    originalImage);  
subplot(1, 2, 1);  
imshow(originalImage);  
title('Original Image');  
subplot(1, 2, 2);  
imshow(blurredImage);  
title('Blurred Image');
```

.....
% Netleştirme

```
originalImage = imread('bulanık.jpg');  
sharpenedImage =  
imsharpen(originalImage, 'Amount', 4,  
    'Radius', 4);  
subplot(1, 2, 1);  
imshow(originalImage);  
title('Original Image');  
subplot(1, 2, 2);  
imshow(sharpenedImage);  
title('Sharpened Image');
```

3.10 İdeal Alçak Geçiren Filtreli Görüntü

```
originalImage =  
imread('basketbolcu.jpg');  
if size(originalImage, 3) > 1  
    originalImage =  
    rgb2gray(originalImage);  
end  
fftImage =  
fft2(double(originalImage));  
cutoffFrequency = 20;  
[M, N] = size(originalImage);  
u = 0:(M-1);  
v = 0:(N-1);  
idx = find(u > M/2);  
u(idx) = u(idx) - M;  
idy = find(v > N/2);  
v(idy) = v(idy) - N;  
[V, U] = meshgrid(v, u);
```



```

D = sqrt(U.^2 + V.^2);
idealFilter = double(D <=
cutoffFrequency);
filteredImage = real(ifft2(fftImage .*
idealFilter));
filteredImage = uint8(255 *
mat2gray(filteredImage));
subplot(1, 2, 1);
imshow(originalImage);
title('Original Image');
subplot(1, 2, 2);
imshow(filteredImage);
title('Filtered Image');

```

3.11 Sobel Filtreli Görüntü

```

originalImage =
imread('basketbolcu.jpg');
if size(originalImage, 3) > 1
    originalImage =
    rgb2gray(originalImage);
end
sobelFilter = fspecial('sobel');
filteredImage =
imfilter(originalImage, sobelFilter);
filteredImage = uint8(255 *
mat2gray(filteredImage));
subplot(1, 2, 1);
imshow(originalImage);
title('Original Image');
subplot(1, 2, 2);
imshow(filteredImage);
title('Filtered Image');

```

3.12 Marr Hildreth Filtreli Görüntü

```

originalImage =
imread('basketbolcu.jpg');
if size(originalImage, 3) > 1
    originalImage =
    rgb2gray(originalImage);
end
filteredImage = edge(originalImage,
'log');
subplot(1, 2, 1);
imshow(originalImage);
title('Original Image');
subplot(1, 2, 2);
imshow(filteredImage);
title('Filtered Image');

```

3.13 Canny Kenar Tespit Edici Filtre

```

originalImage =
imread('basketbolcu.jpg');
if size(originalImage, 3) > 1
    originalImage =
    rgb2gray(originalImage);
end

```

```

edgeImage = edge(originalImage,
'Canny');
subplot(1, 2, 1);
imshow(originalImage);
title('Original Image');
subplot(1, 2, 2);
imshow(edgeImage);
title('Edge-Detected Image');

```

3.14. Boyutu Değiştirilmiş (Resized) Görüntü

```

originalImage =
imread('basketbolcu.jpg');
outputSize = [500, 500];
resizedImage = imresize(originalImage,
outputSize);
subplot(1, 2, 1);
imshow(originalImage);
title('Original Image');

subplot(1, 2, 2);
imshow(resizedImage);
title('Resized Image');

```

3.15 Ayna filtreli Görüntü

```

originalImage =
imread('basketbolcu.jpg');
mirroredImage = flip(originalImage,
2);
subplot(1, 2, 1);
imshow(originalImage);
title('Original Image');
subplot(1, 2, 2);
imshow(mirroredImage);
title('Mirrored Image');

```

3.16 Uzamsal Dönüşüme Uğramış Görüntü

```

inputImage =
imread('basketbolcu.jpg');
theta = 30;
scaleX = 1.5;
scaleY = 0.8;
translationX = 50;
translationY = -20;
T = affine2d([
    scaleX*cosd(theta) -
    scaleY*sind(theta) 0;
    scaleX*sind(theta)
    scaleY*cosd(theta) 0;
    translationX      translationY
1
]);
transformedImage = imwarp(inputImage,
T);
subplot(1, 2, 1);

```



```
imshow(inputImage);
title('Original Image');
subplot(1, 2, 2);
imshow(transformedImage);
title('Transformed Image');
```

Sonuçlar Kısmı Kendi Filtrem

```
originalImage =
imread('basketbolcu.jpg');
originalImage =
im2double(originalImage);
redChannel = originalImage(:, :, 3);
minValue = min(redChannel(:));
maxValue = max(redChannel(:));
range = maxValue - minValue;
lowContrastRed = 1.1 * range *
redChannel + minValue;
lowContrastImage = originalImage;
lowContrastImage(:, :, 2) =
lowContrastRed;
figure;
subplot(1, 2, 1);
imshow(originalImage);
title('Original Resim');
subplot(1, 2, 2);
imshow(lowContrastImage);
title('Son Hali');
```

REFERANSLAR

[1]<https://www.geeksforgeeks.org/digital-image-processing-basics/>

[2]<https://www.ijstr.org/final-print/sep2019/Comparative-Analysis-Of-Various-Filtering-Techniques-In-Image-Processing.pdf>

[3]<https://www.geeksforgeeks.org/how-to-convert-rgb-image-to-binary-image-using-matlab/>

[4]<https://www.geeksforgeeks.org/matlab-rgb-image-to-grayscale-image-conversion/>

[5]<https://www.youtube.com/watch?v=c2VMpu0Q4UU>

[6]<https://www.mathworks.com/matlabcentral/answers/894162-converting-image-to-subimages>

[7]<https://www.delftstack.com/howto/matlab/negative-of-image-matlab/>

[8]<https://www.mathworks.com/matlabcentral/answers/497066-low-contrast-and-high-contrast>

[9]<https://www.imageeprocessing.com/2011/10/add-salt-and-pepper-noise-to-image.html>

[10]<https://www.youtube.com/watch?v=H2EqWgAdMaY>

[11]<https://www.youtube.com/watch?v=5bxh5UgOZRk>

[12]<https://www.geeksforgeeks.org/matlab-ideal-lowpass-filter-in-image-processing/>

[13]<https://www.geeksforgeeks.org/matlab-image-edge-detection-using-sobel-operator-from-scratch/>

[14]<https://www.youtube.com/watch?v=cToG83MLkqw>

[15]<https://www.mathworks.com/matlabcentral/answers/539287-image-processing-canny-edge-detection>

[16]<https://www.mathworks.com/help/matlab/ref/imresize.html>

[17]<https://www.geeksforgeeks.org/create-mirror-image-using-matlab/>

[18]<https://www.mathworks.com/help/images/rf/imeregcorr.html>