

DİJİTAL GÖRÜNTÜ İŞLEME ÖDEVİ

Görüntü İşleme Ödevi 2: Temel Görüntü Operasyonları ve İnterpolasyon

Özet

Ödevde; görüntü boyutunu büyütme, küçültme, zoom in, zoom out, döndürme gibi temel görüntü işleme operasyonlarını gerçekleştirilmiştir. Ancak, bu operasyonları gerçekleştirirken herhangi bir dış kütüphane kullanma yasağı bulunmaktadır. Ayrıca, bu operasyonları yaparken bilinear, bicubic, average gibi farklı interpolasyon yöntemlerini de kullanılmıştır.

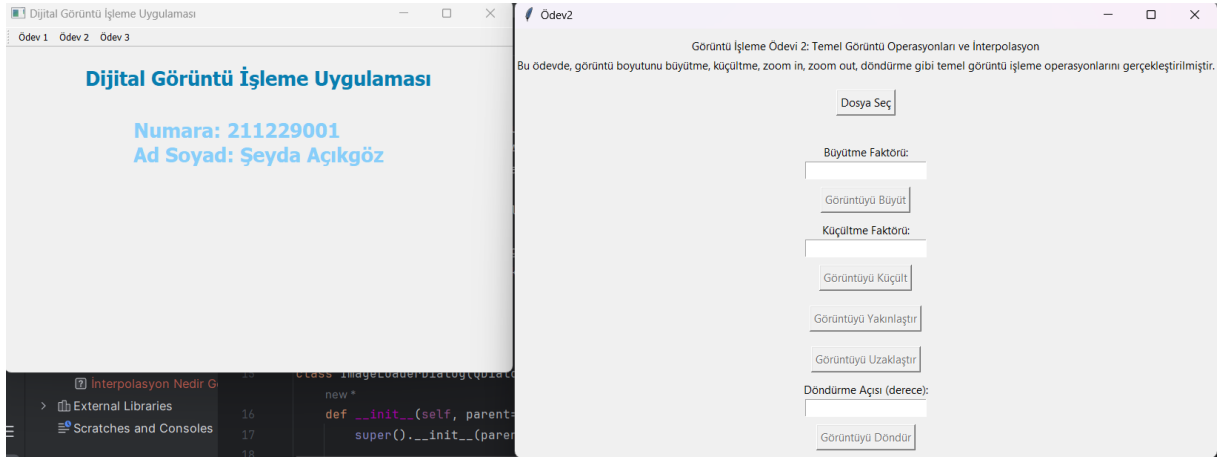
Ödevin Bulunduğu Github Linki

<https://github.com/SeydaAcikgoz/DijitalGoruntuIsleme/tree/main/GUI/Odev2>

Şeyda AÇIKGÖZ
211229001

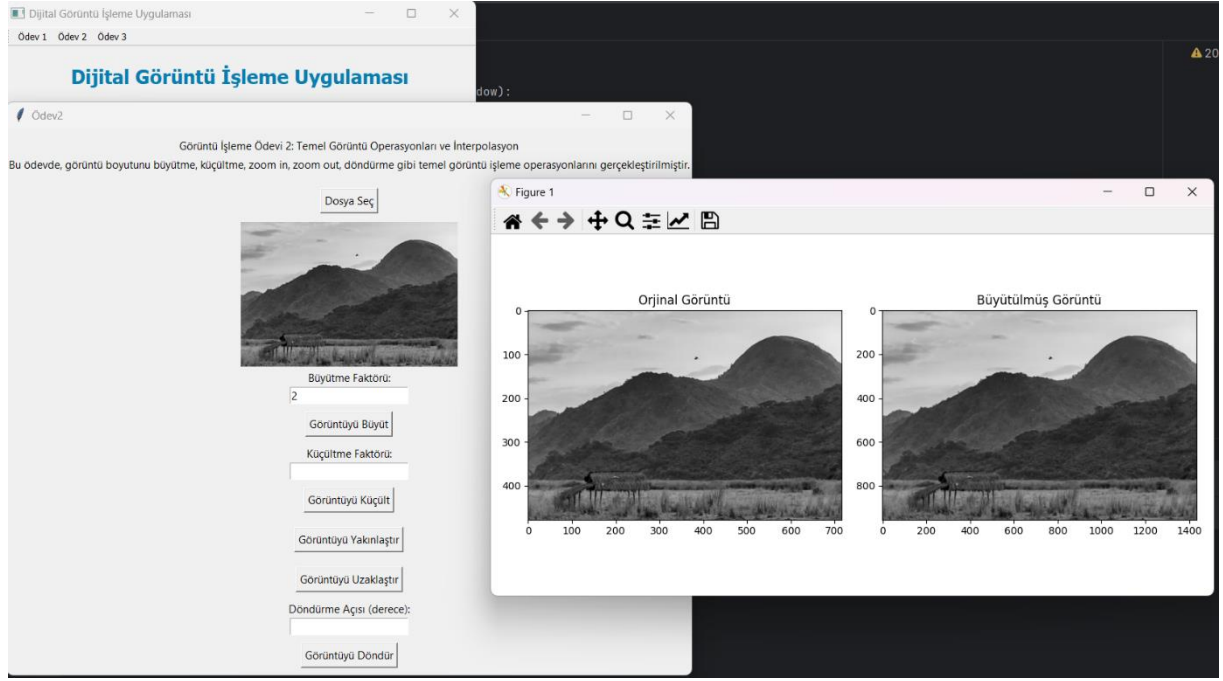
Bu ödevde, görüntü boyutunu büyütme, küçültme, zoom in, zoom out, döndürme gibi temel görüntü işleme operasyonlarını gerçekleştirilmiştir. Ayrıca average, bilinear ve bicubic interpolasyon yöntemlerini kullanarak görüntüleri işlemek için fonksiyonlar içerir. Her işlem sonucunda, kullanıcıya orijinal ve işlenmiş görüntülerin yan yana gösterildiği bir grafik penceresi sunulmuştur.

Açılan ödev2 sayfasında ödevle ilgili temel bilgilerin bulunduğu labelar ve işlem yapılacak butonlar bulunmaktadır. Dosya seç butonu ile bilgisayardan görüntü seçilmektedir. Bu görüntü seçilmeden işlem yapacak diğer butonlar inaktif durumdadır. (Resim1)



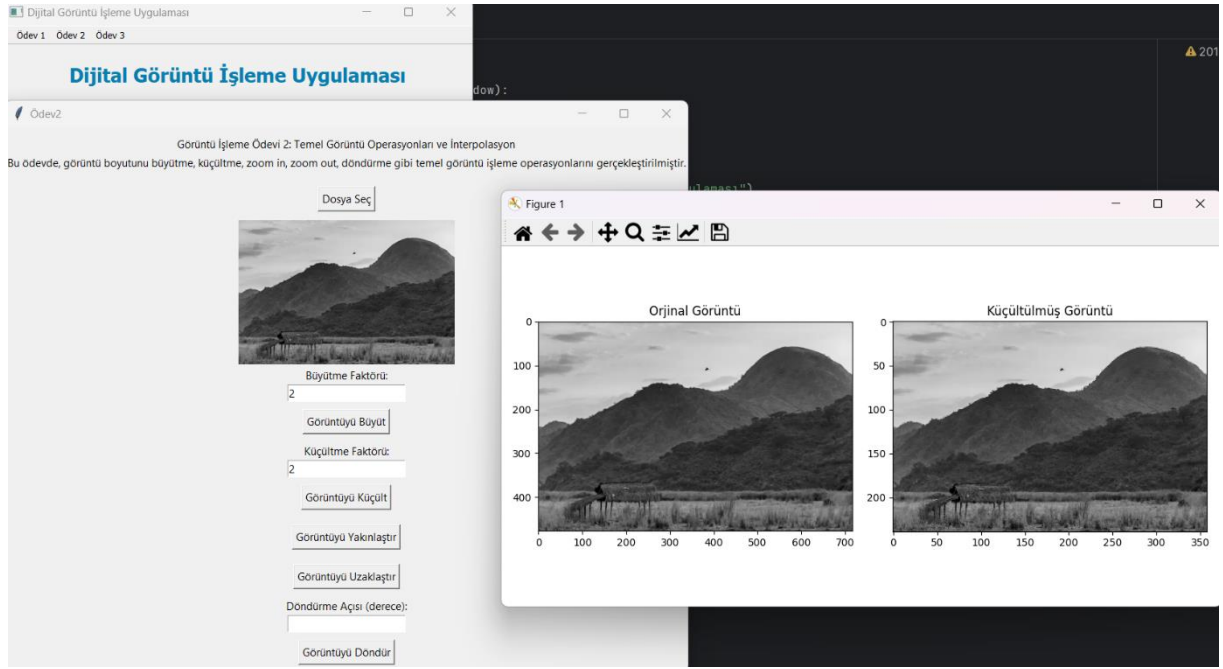
Resim1 Ödev2 sayfası

Görüntü büyütme kısmında büyütme faktörünü alabilecek bir label bulunmaktadır. Kullanıcıdan alınan değere göre bilinear interpolasyon ile büyütme işlemi gerçekleştirilmiştir. İlk olarak, seçilen görüntü **Image.open** fonksiyonuyla açılır ve **image** adlı bir değişkende saklanır. Ardından, görüntü bir NumPy dizisine dönüştürülür. Bu, görüntü işleme işlemlerini gerçekleştirmek için NumPy kütüphanesinin kullanılmasını sağlar. Görüntünün yükseklik (**height**), genişlik (**width**) ve kanal sayısı (**channels**) değerleri alınır. Kanal sayısı, görüntünün renk kanallarının sayısını temsil eder. Yeni boyutlar hesaplanır. Önceki adımlarda alınan yükseklik ve genişlik değerleri, kullanıcının belirlediği büyütme faktörüyle çarpılarak yeni boyutlar elde edilir. Yeni boyutlarda bir NumPy dizisi oluşturulur. Bu dizi, büyütülmüş görüntünün piksel değerlerini saklayacaktır. Büyütme işlemi gerçekleştirilir. İki iç içe döngü kullanılarak, her bir piksel için orijinal görüntüdeki konumu hesaplanır. Bu konumlar, kullanıcının belirlediği büyütme faktörüne göre yeniden hesaplanır. Ardından, bu orijinal konumlara karşılık gelen değerler, bilinear interpolasyon kullanılarak yeni boyuttaki görüntüye aktarılır. (Resim2)



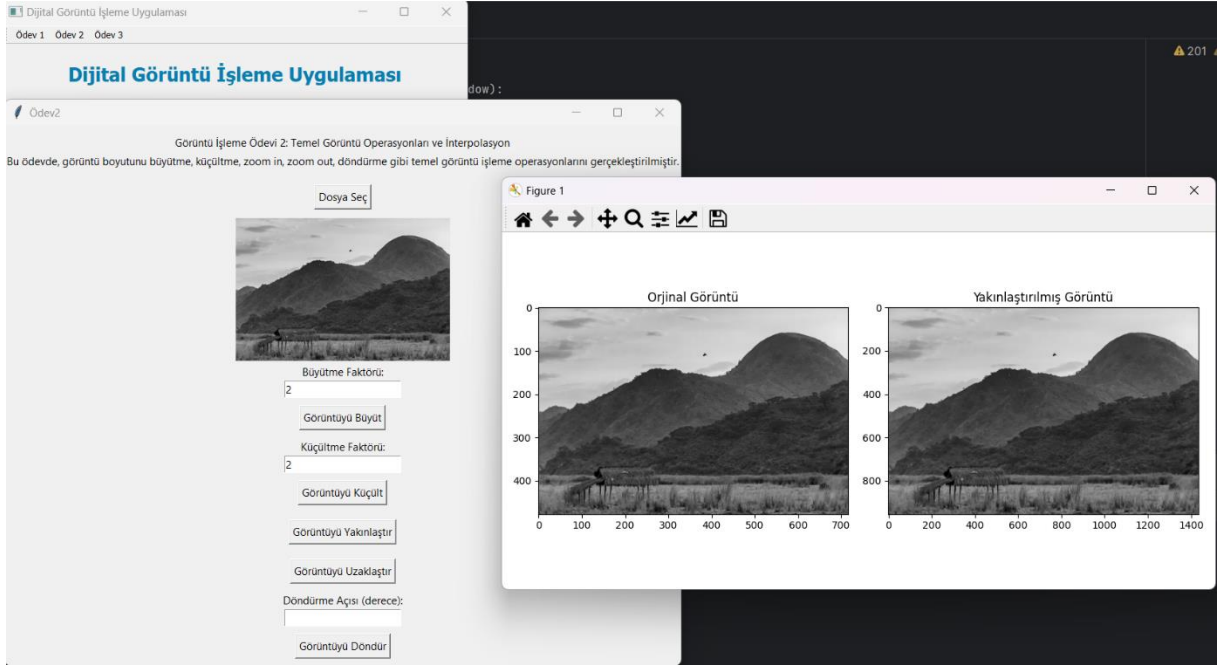
Resim2 Büyütme işlemi

Görüntü küçültme kısmında öncelikle küçültme faktörünü alabilecek bir label bulunmaktadır. Kullanıcıdan alınan değere göre bilinear interpolasyon ile büyütme işlemi gerçekleştirilmiştir. İlk olarak, seçilen görüntü **Image.open** fonksiyonuyla açılır ve **image** adlı bir değişkende saklanır. Ardından, görüntü bir NumPy dizisine dönüştürülür. Bu, görüntü işleme işlemlerini gerçekleştirmek için NumPy kütüphanesinin kullanılmasını sağlar. Görüntünün yükseklik (**height**), genişlik (**width**) ve kanal sayısı (**channels**) değerleri alınır. Kanal sayısı, görüntünün renk kanallarının sayısını temsil eder. Yeni boyutlar hesaplanır. Önceki adımlarda alınan yükseklik ve genişlik değerleri, kullanıcının belirlediği küçültme faktörüyle bölünerek yeni boyutlar elde edilir. Yeni boyutlarda bir NumPy dizisi oluşturulur. Bu dizi, küçültülmüş görüntünün piksel değerlerini saklayacaktır. Küçültme işlemi gerçekleştirilir. İki iç içe döngü kullanılarak, her bir piksel için orijinal görüntüdeki konumu hesaplanır. Bu konumlar, kullanıcının belirlediği küçültme faktörüne göre yeniden hesaplanır. Ardından, bu orijinal konumlara karşılık gelen değerler, bilinear interpolasyon kullanılarak yeni boyuttaki görüntüye aktarılır. (Resim3)



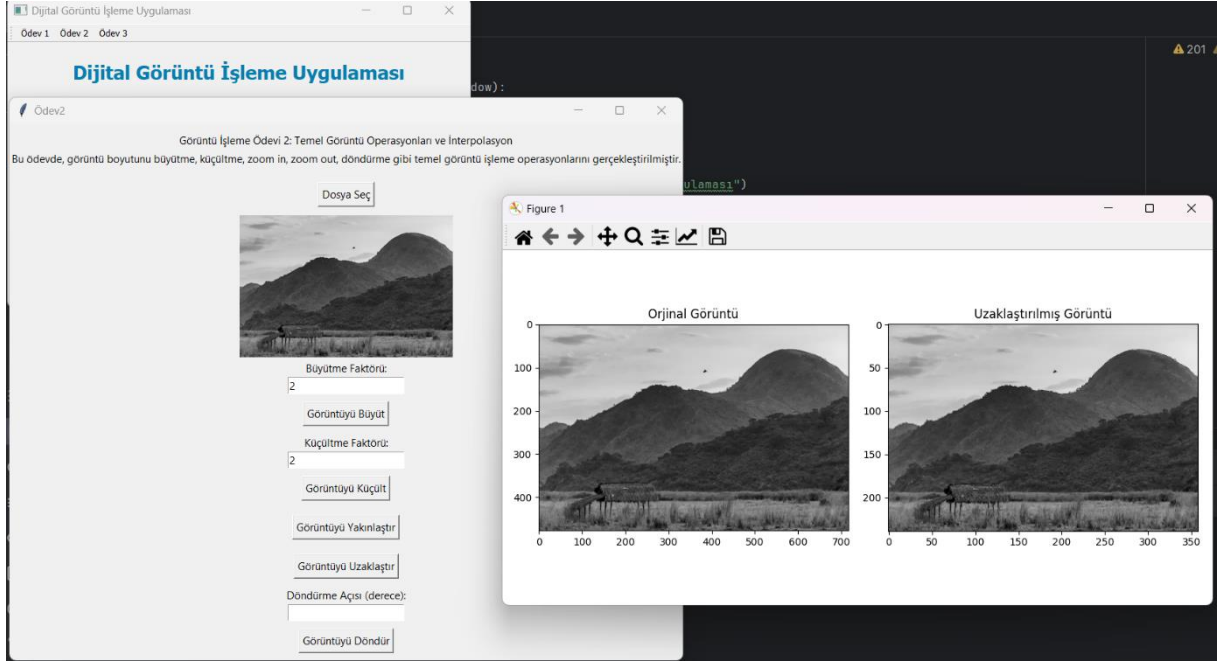
Resim3 Küçültme işlemi

Görüntü yakınlaştırma kısmında,average interpolasyon yöntemiyle belirlenen ölçüde yakınlaştırma işlemi yapılmıştır. İlk olarak, seçilen görüntü **Image.open** fonksiyonuyla açılır ve **image** adlı bir değişkende saklanır.Ardından, görüntü bir NumPy dizisine dönüştürülür. Bu, görüntü işleme işlemlerini gerçekleştirmek için NumPy kütüphanesinin kullanılmasını sağlar.Görüntünün yükseklik (**height**), genişlik (**width**) ve kanal sayısı (**channels**) değerleri alınır. Kanal sayısı, görüntünün renk kanallarının sayısını temsil eder.Yeni boyutlar hesaplanır. Önceki adımlarda alınan yükseklik ve genişlik değerleri, kullanıcının belirlediği yakınlaştırma faktörüyle çarpılarak yeni boyutlar elde edilir.Yeni boyutlarda bir NumPy dizisi oluşturulur. Bu dizi, yakınlaştırılmış görüntünün piksel değerlerini saklayacaktır.Yakınlaştırma işlemi gerçekleştirilir. İki iç içe döngü kullanılarak her piksel için orijinal görüntüdeki koordinatlar hesaplanır. Bu koordinatlar, yakınlaştırma faktörüne göre orijinal görüntünün boyutlarına uygun hale getirilir. Daha sonra, bu orijinal koordinatlara karşılık gelen piksel değerlerinin ortalaması alınarak yeni boyuttaki görüntüye aktarılır.Bu yöntem, her bir pikselin yakınlaştırılmasında orijinal piksellerin etrafındaki bilgilerin ortalama değerlerini kullanır. Bu, görüntünün daha yumuşak bir şekilde yakınlaştırılmasını sağlar.(Resim4)



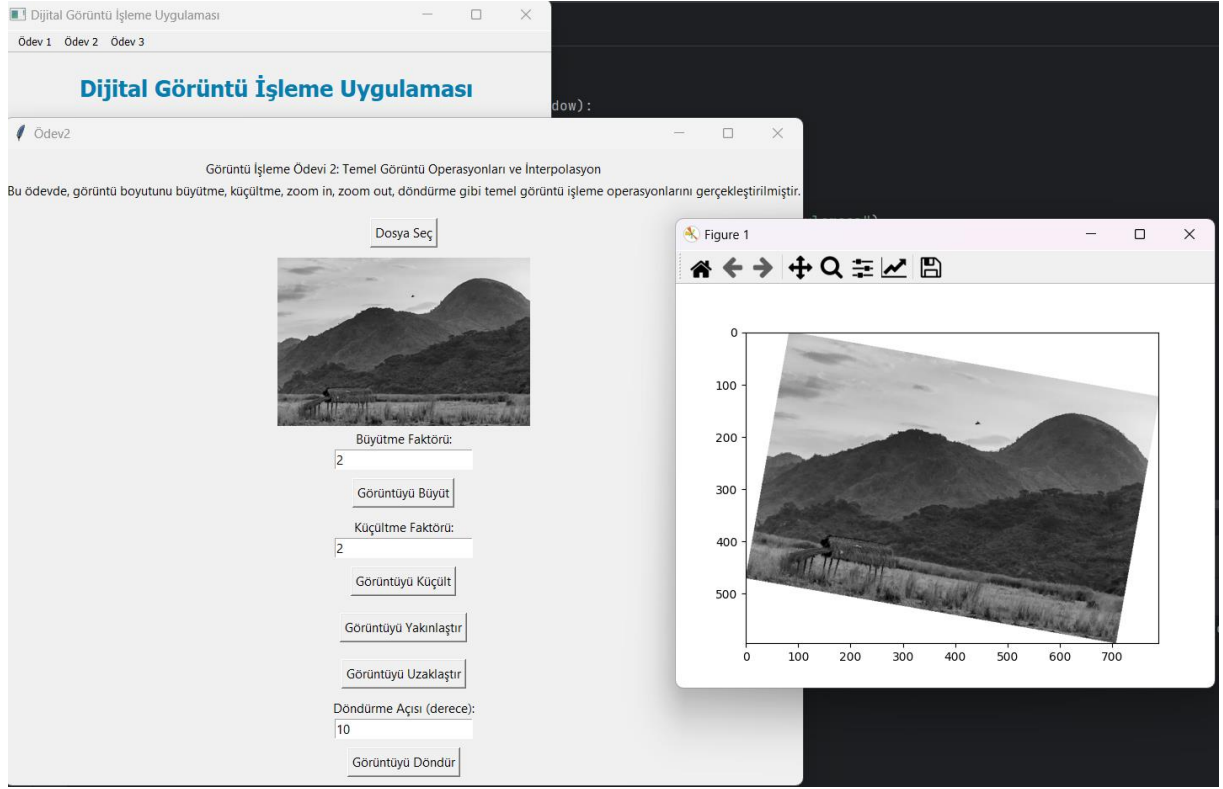
Resim4 Yakınlaştırma işlemi

Görüntü uzaklaştırma kısmında, bicubic interpolasyon yöntemiyle belirlenen ölçüde uzaklaştırma işlemi yapılmıştır. Bicubic interpolasyon, her bir hedef pikselin orijinal görüntüdeki bir bölgesinden alınan değerlerin ağırlıklı ortalamasını hesaplar. Bu yöntem, piksel değerlerini düzgün bir şekilde dağıtarak daha pürüzsüz sonuçlar elde etmeyi sağlar. İlk olarak, verilen görüntü dosyası **self.görüntü_yolu** kullanılarak açılır ve bir **PIL.Image** nesnesine yüklenir. Ardından, **PIL.Image** nesnesi bir NumPy dizisine dönüştürülür. Bu dönüşüm, görüntü işleme işlemlerinin NumPy dizileri üzerinde daha etkili bir şekilde gerçekleştirilmesini sağlar. Görüntünün yüksekliği (**height**), genişliği (**width**) ve kanal sayısı (**channels**) alınır. Uzaklaştırılmış görüntünün boyutu, orijinal boyutun belirlenen uzaklaştırma faktörüne bölünerek hesaplanır. Yeni boyutlarda bir NumPy dizisi oluşturulur. Bu dizide, her bir piksel için kanal başına bir değer saklanır. İki iç içe döngü kullanılarak, her bir hedef piksel için orijinal görüntüdeki koordinatlar hesaplanır. Ancak, bu koordinatlar doğrudan hedef pikselin koordinatlarından türetilmez. Bunun yerine, hedef pikselin konumu, uzaklaştırma faktörü ile orijinal görüntünün boyutuna oranlanarak hesaplanır. Bu şekilde, hedef pikselin koordinatları doğrudan orijinal görüntüdeki piksellerin konumlarına karşılık gelmez, ancak bir ara değer olarak kullanılır. Her bir hedef piksel için, orijinal görüntüdeki bir bölge seçilir ve bu bölgedeki piksel değerlerinin bicubic interpolasyonu hesaplanır. Bicubic interpolasyon, seçilen bölgedeki piksel değerlerinin bir polinom kullanılarak ağırlıklı ortalamasını alır. Bu polinom genellikle bicubic spline fonksiyonları kullanılarak belirlenir. Son olarak, elde edilen yeni piksel değerleri, yeni boyuttaki görüntünün ilgili piksel konumlarına yerleştirilir. Böylece, orijinal görüntünün her bir pikseli, yeni boyuttaki görüntüdeki piksel değerlerinin doğru bir şekilde atanmasıyla uzaklaştırma işlemi gerçekleştirilir. (Resim5)



Resim5 Uzaklaştırma işlemi

Görüntü döndürme kısmında öncelikle döndürme açısını alabilecek bir label bulunmaktadır. Alınana aç değeri göre döndürme işlemi yapılmaktadır. İlk olarak, verilen görüntü dosyası **self.görüntü_yolu** kullanılarak açılır ve bir **PIL.Image** nesnesine yüklenir. Görüntünün genişliği (**width**) ve yüksekliği (**height**) alınır. Dönüş açısı (**aci**) radyan cinsinden hesaplanır. Çünkü **math** modülü trigonometrik fonksiyonlarla çalışırken radyan cinsinden açılar kullanır. Döndürülmüş görüntünün yeni boyutları, dönüş açısına göre hesaplanır. Bu, dönüşüm matrisi kullanılarak gerçekleştirilir. Dönüşüm matrisi, döndürülmüş görüntünün sınırlarını içerecek şekilde hesaplanır. Yeni boyutlarda boş bir görüntü oluşturulur. Bu görüntü, döndürülmüş piksellerin tutulacağı alanı temsil eder. Her bir pikselin koordinatları döngüler aracılığıyla yeni dönüşüm matrisine göre hesaplanır. Bu, döndürülmüş görüntüdeki her bir pikselin orijinal görüntüdeki konumunu belirler. Eğer pikselin orijinal konumu orijinal görüntünün sınırları içindeyse, orijinal pikselin değeri döndürülmüş görüntüdeki ilgili konuma yerleştirilir. (Resim6)



Resim6 Döndürme işlemi