



T.C  
DOKUZ EYLÜL ÜNİVERSİTESİ  
FEN FAKÜLTESİ

# **STAJ RAPORU**

Ağustos / 2020

İZMİR



T.C  
DOKUZ EYLÜL ÜNİVERSİTESİ  
FEN FAKÜLTESİ

**STAJ  
RAPORU**

Hüseyin Cem ARAS

Ağustos / 2020

İZMİR

## RAPOR DEĞERLENDİRME FORMU

Hüseyin Cem ARAS tarafından Öğr. Gör. Dr. Alper VAHAPLAR yönetiminde hazırlanan “Python ile Görüntü İşleme” başlıklı rapor tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Staj Raporu olarak kabul edilmiştir.

Öğr. Gör. Dr. Alper VAHAPLAR

## TEŞEKKÜR

Bu pandemi döneminde uzaktan staj çalışması boyunca bana yardımcı olan danışmanım Öğr. Gör. Dr. Alper VAHAPLAR'a teşekkür ederim.

Hüseyin Cem ARAS

# İÇİNDEKİLER

İÇİNDEKİLER.....	5
ŞEKİLLER LİSTESİ.....	6
BÖLÜM BİR GÖRÜNTÜ İŞLEMEYE GİRİŞ .....	7
1.1. Görüntü İşleme Kütüphaneleri.....	7
1.1.1. Numpy Kütüphanesi.....	7
1.1.2. Matplotlib Kütüphanesi.....	8
1.1.3. OpenCV Kütüphanesi.....	8
BÖLÜM İKİ GÖRÜNTÜ İŞLEMENİN TEMELLERİ.....	9
2.1. Görüntünün Tanımı.....	9
2.1.1. Dijital Görüntü .....	9
2.1.2. Çözünürlük .....	10
2.2. Görüntü İşleme.....	10
2.3. Görüntü Ön İşleme.....	11
2.3.1. Ölçeklendirme .....	11
2.3.1.1. Küçültme.....	11
2.3.1.2. Büyütme.....	12
BÖLÜM ÜÇ UYGULAMA .....	14
3.1. Küçültme-Piksel Değiştirme Metodu.....	14
3.2. Küçültme-İnterpolasyon Metodu .....	15
3.3. Büyültme-Piksel Değiştirme Metodu.....	16
3.4. Büyültme-Piksel İnterpolasyonu Metodu .....	17
BÖLÜM DÖRT SONUÇ .....	19
BÖLÜM BEŞ KAYNAKÇA.....	20

## ŞEKİLLER LİSTESİ

Şekil 2.1 Görüntü Küçültme İşlemi.....	12
Şekil 2.2 Görüntü Küçültme İşlemi.....	13
Şekil 3.1 Orijinal Görüntü .....	15
Şekil 3.2 1.25 Kat Piksel Değiştirme ile Küçültülmüş Görüntü .....	15
Şekil 3.3 Orijinal Görüntü .....	16
Şekil 3.4 1.25 Kat Piksel İnterpolasyonu ile Küçültülmüş Görüntü .....	16
Şekil 3.5 Orijinal Görüntü .....	17
Şekil 3.6 1.25 Kat Piksel Değiştirme ile Büyültülmüş Görüntü .....	17
Şekil 3.7 Orijinal Görüntü .....	18
Şekil 3.8 1.25 Kat Piksel İnterpolasyonu ile Büyültülmüş Görüntü .....	18

# BÖLÜM BİR

## GÖRÜNTÜ İŞLEMEYE GİRİŞ

Görüntü işleme, verilerin kamera, tarayıcı vb. cihazlar yardımıyla görüntülerin alınıp değerlendirme işleminden sonra, bilgisayarın veya başka bir aracın okuyabileceği bir şekilde dönüştürülme işlemine tabi tutulması veya dijital bir ortamdan başka bir dijital ortama aktarılma işlemidir. (Rush, 2016) Görüntü işleme temelde üç adımda incelenmektedir. Bunlar; çeşitli cihazlarla görüntünün elektronik ortama aktarılması, görüntünün analiz edilerek işlenmesi ve işlenen görüntünün veri raporu ve çıktısının alınmasıdır. (Neuman, Sapirstein, Shwedyk, & Bushuk, 1989)

Görüntü işleme, bir görüntüyü sayısal hale dönüştürdükten sonra elde edilen verilere matematiksel işlemler ve farklı yöntemler uygulanarak çeşitli çıkarımlar oluşturma ya da yeni görüntüler üretme tekniğidir. Görüntü işleme tekniği, görüntünün netliğini artırma, ışık değerlerini değiştirme ve görüntü üzerinde bulunan herhangi bir nesnenin tanımlanabilmesi gibi birçok amaçla kullanılmaktadır. Görüntü işleme uygulamaları son yıllarda ciddi bir artış göstermiştir. Özellikle araç içi otomasyon sistemlerinden, robot uygulamalarında, askeri uygulamalardan tarım uygulamalarına kadar birçok alanda yaygın olarak kullanılmaktadır. (SAMTAŞ & GÜLESİN, 2011)

### 1.1. Görüntü İşleme Kütüphaneleri

Python programlama dilinde görüntü işlemede genellikle Numpy, OpenCV, Matplotlib, Pillow, Scipy, Scikit-Image kütüphaneleri kullanılmaktadır.

#### 1.1.1. Numpy Kütüphanesi

Numpy, bilimsel hesaplama işlemleri, çok boyutlu diziler, çeşitli türetilmiş nesneler dahil olmak üzere diziler üzerinde hızlı işlemler yapılabilmesi için kullanılan önemli kütüphanelerden birisidir. Numpy, veri tipi olarak n boyutlu dizinin tanımlanmasını sağlar ve diziler üzerinde yeniden boyutlandırma, indeksleme gibi temel işlemlerin gerçekleştirilmesinde kullanılmaktadır. Görüntü işleme ve yapay zeka uygulamalarında, minimum kod yazarak, hesaplama işlemleri ve dönüşüm işlemlerinde sıkça kullanılmaktadır. Görüntü işleme ve yapay zeka uygulamalarında,

minimum kod yazarak, hesaplama işlemleri ve dönüşüm işlemlerinde sıkça kullanılmaktadır. (Numpy Kütüphanesi Nedir? Nasıl Kullanılır?, 2020)

### **1.1.2. Matplotlib Kütüphanesi**

Matplotlib, çeşitli basılı formatlarda ve interaktif ortamlarda yüksek kalitede görüntüler üretebilen iki boyutlu çizim kütüphanesidir. Matplotlib, grafiksel olarak zor işlerin gerçekleştirilmesini mümkün kılmaktadır. Kütüphanenin önemli avantajlarından birisi de çok fazla kod yazmadan sadece birkaç satırlık kodla grafikler, histogramlar, çubuk grafikleri oluşturabilmesini sağlamasıdır. (matplotlib.org, 2020)

### **1.1.3. OpenCV Kütüphanesi**

OpenCV (Open Source Computer Vision), bilgisayarlı görü uygulamaları ve ticari ürünlerde makine öğrenme algısını hızlandırmak için hazırlanmış bir kütüphanedir. OpenCV açık kaynak kodlu ve platformdan bağımsız bir kütüphane olduğu için Windows, Linux, FreeBSD, Android, Mac OS ve iOS platformlarında çalışabilmektedir. OpenCV kütüphanesi içerisinde yer alan 500'den fazla fonksiyon ile görüntü analiz işlemlerini hızlı bir şekilde gerçekleştirebilmektedir. OpenCV, Bilgisayar Görüsü ve Makine Öğrenimi ile ilgili çok sayıda algoritmayı desteklemekte ve günden güne genişlemektedir. Bu algoritmalar ile yüz tanıma, nesneleri ayırt etme, nesne sınıflandırma, plaka tanıma, üç boyutlu görüntü üzerinde işlem yapabilme, insan hareketlerini tespit edebilme gibi birçok alanda başarı ile uygulanmaktadır. İsrail'de video kamera kullanılarak izinsiz girişlerin belirlenmesi, Çin Halk Cumhuriyeti'ndeki maden takımlarının izlenmesi, Avrupa'da yüzme havuzundaki boğulma olaylarının belirlenmesi gibi birçok uygulamada OpenCV kütüphanesi kullanılmaktadır. Ayrıca Microsoft, Toyota, Honda, Google, Intel, IBM, Yahoo, Sony gibi büyük şirketler OpenCV kütüphanesini kullanmaktadır. (OpenCv.org, 2020)



## BÖLÜM İKİ

# GÖRÜNTÜ İŞLEMENİN TEMELLERİ

### 2.1. Görüntünün Tanımı

Görüntü işleme, bir görüntüyü sayısal forma dönüştürerek nesne tanıma, görüntü iyileştirme gibi işlemleri gerçekleştirebilmek için kullanılan bir yöntemdir. Ayrıca görüntü işleme yöntemleri ile iyileştirilmiş veya değiştirilmiş görüntüler elde edilerek görüntülerden özellik ve anlamlı bilgiler alınabilmektedir. İki boyutlu bir sahneyi temsil eden veri olarak tanımlanabilen görüntü ultrason, elektro mikroskop ve bilgisayar sayesinde girdi olarak alınan çeşitli içeriklere sahip olabilir. Farklı içeriklere sahip olan görüntüler ile görüntüler çevre, mekân, kentsel planlama coğrafi analizler yapılmaktadır. Yapılan analiz işlemleri ile ilgili bugüne kadar birçok yaklaşım ve yöntem sunulmuştur. Bu yöntemler sayesinde otoyoldan geçen araç sayısını belirlemek, aracın plakasını tespit etmek, farklı şekle sahip cisimleri ayırmak, farklı renklerdeki cisimleri algılamak, bilgisayarlı tomografi görüntüsünün netleştirilmesi, resim veya videoda doğru diyafram ayarlarının belirlenmesi, uygun renk ve ışık kalitesini ayarlanması, güvenlik taramaları için renklerin zenginleştirilmesi, fotoğraf netliğinin artırılması, görüntünün verimli bir şekilde depolanması ve aktarılması, görüntü boyut/renk/tonlama niteliklerinin düzenlenmesi, barkod okuma, yazı karakterinin metne dönüştürülmesi, ortamdaki kişi sayısının tespiti, üç boyutlu görüntülerin hazırlanması, gerçek ile sanal dünya görüntülerinin birleştirilmesi ve medikal görüntülerin analizi gibi birçok alanda görüntü işleme teknolojilerinden yararlanılmaktadır. (BAYRAM, 2019)

#### 2.1.1. Dijital Görüntü

Görüntü  $x$  ve  $y$  uzamsal koordinatlarında  $I(x,y)$  şeklinde gösterilen bir fonksiyondur. Dijital görüntü, iki boyutlu tam sayı dizisinden oluşmaktadır ve dizideki her elemana piksel adı verilmektedir.

Görüntü boyutu bilinmeyen bir görüntünün dizi boyutunu öğrenmek için Python programlama dilinde *shape* fonksiyonu kullanılmaktadır.

### 2.1.2. Çözünürlük

Çözünürlük, görüntüdeki her bir inç içinde bulunan nokta veya piksel sayısıdır. İnç başına düşen nokta/piksel sayısı ile ifade edilmektedir.

Görüntünün çözünürlüğü dört farklı tipe ayrılmaktadır;

- Düşük çözünürlük (ing. Low Resolution (LR)),
- Orta çözünürlük (Medium Resolution (MR)),
- Yüksek çözünürlük (High Resolution (HR)),
- Süper çözünürlük (Super Resolution (SR)).

Görüntü işleme yöntemlerinde genellikle yüksek çözünürlüklü görüntüler üzerinde analiz işlemleri gerçekleştirilmek istenmektedir. Ancak sensör ve optik araçlardan alınan görüntülerde istenilen çözünürlükte görüntüler elde edilemeyebilir. Bu sorunu ortadan kaldırmak için farklı görüntü işleme yöntemleri kullanılarak düşük çözünürlüklü görüntülerden daha yüksek çözünürlüklü bir görüntü elde edilmektedir. Yüksek çözünürlüklü görüntü elde etmek için 2017 yılında tek düşük çözünürlüklü görüntüyü yüksek frekans ayrıntıların kaybetmeden ve görüntü kalitesini düşürmeden üç katmanlı Evrişimli Sinir Ağı yöntemi (ing. Convolutional Neural Network (CNN) ile gerçekleştirilmiştir. (BAYRAM, 2019)

### 2.2. Görüntü İşleme

Görüntü işleme, gerçek hayattan kamera, tarayıcı vb. cihazlar yardımıyla yakalanan görüntülerin bilgisayar veya başka bir cihazın okuyabileceği sayısal hâle dönüştürüldükten sonra özelliklerinin ve niteliklerinin değiştirilmesi işlemidir. Görüntü işleme teknolojisi; görüntünün netliğini artırma, görüntü üzerinde bulunan herhangi bir nesnenin çıkarımının yapılabilmesi ya da nesnelerin tanımlanabilmesi gibi birden fazla amaç için kullanılabilir. (Young, Gerbrands, & Vliet, 2007)

Herhangi bir resmin yazılım aracılığıyla kullanılabilmesi için sayısallaştırılması gerekmektedir. Sayısallaştırma; resimde bulunan renkleri sayısal veriye dönüştürme işlemidir. Sayısal görüntü, her birinin özel bir konumu ve değeri olan sonlu sayıdaki piksel olarak adlandırılan alt birimlerden oluşmaktadır.

Görüntü işleme teknolojisi ise, kamera, video gibi araçlar ile görüntünün elektronik ortama aktarılması ile görüntünün analiz edilerek istenilen şekilde işlenmesi, görüntünün analizinden sonra oluşan veri raporu ve sonucun alınması olarak üç aşamadan oluşmaktadır.

Görüntü işleme teknolojisi için analog ve dijital olmak üzere iki tür yöntem kullanılmaktadır. Fotokopiler ve fotoğraflar gibi basılı kopyalar için analog görüntü işleme teknikleri kullanılırken, görüntü üzerinde iyileştirme ya da düzeltme yapılması gerektiğinde dijital görüntü işleme tekniği tercih edilmektedir. Ek olarak dijital görüntü işleme için verilerin geçmesi gereken ön işleme, geliştirme ve görüntüleme,

bilgi çıkarım gibi temel aşamalar mevcuttur. Bu aşamalardan sonra ilgili veriden amaca bağlı olarak istenilen sonuçlar elde edilebilmektedir.

Görüntü işleme teknolojisinin kullanım amaçları beş aşamada incelenmektedir;

- GÖrselleştirme: Görüntüde zor ayırt edilebilen nesneleri gözlemlemek,
- Görüntü keskinleştirme ve restorasyon: Gürültülü veya bozuk görüntüleri iyileştirmek,
- Görüntü alımı: Yüksek çözünürlüklü görüntü arama,
- Desen Tanıma: Herhangi bir görüntüdeki çeşitli nesneleri tanımlama,
- Görüntü Tanıma: Görüntü içerisindeki nesneleri ayırt etme.

### **2.3. Görüntü Ön İşleme**

Görüntü işleme aşamalarının ilki görüntü ön işlemedir. Kamera ya da diğer görüntüleme cihazlarıyla elde edilen görüntüleri doğrudan analiz etmek ve yorum yapmak mümkün olmamaktadır. Bunun nedeni kaynak olarak alınan görüntülerin gürültü, ışık yetersizliği gibi olumsuzluklar içermesidir. Bu sorunları gidermek için kaynak görüntüyü iyileştirmeye yönelik filtreler uygulanarak görüntünün işlenebilmesi görüntü ön işleme aşamalarından geçirilmesi gerekmektedir.

Görüntü ön işleme; görüntüleme cihazlarından elde edilen sayısal görüntüleri analiz ve yorumlama (anlamlandırma) işlemlerine geçmeden görüntü üzerinde yapılan ölçeklendirme, renk uzayındaki geçişler, eşikleme (ing. thresholding), histogram işlemleri, filtreleme ve morfolojik dönüşüm gibi teknikler kullanılarak gerçekleştirilmektedir. Görüntü ön işleme görüntü üzerindeki nesneleri daha belirgin ve kolay işlenebilir hâle getirmek için uygulanmaktadır. (Aksoy, 2020)

#### **2.3.1. Ölçeklendirme**

Ölçeklendirme işlemleri görüntüler üzerinde belirli bir alanda ya da görüntünün tamamında görüntünün büyütülmesi veya küçültülmesi için kullanılmaktadır. (Scaling (geometry), 2020) Büyültme ve küçültme işlemleri; bir grup matris piksel değerlerini daha küçük ya da daha büyük matrislerle ifade ederek gerçekleştirilmektedir.

##### **2.3.1.1. Küçültme**

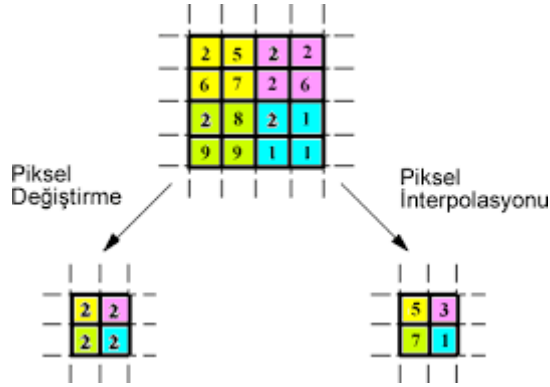
Görüntünün küçültülmesinde piksel değiştirme ve piksel interpolasyonu olmak üzere iki yöntem kullanılmaktadır.

### Piksel Değiştirme

Piksel değiştirme yönteminde, bir grup komşu piksel değerinin tek bir piksel değeri ile ifade edilerek küçültme işlemi gerçekleştirilmektedir. Yapılan işlem hesaplama açısından kolay iken pikseller arasında farklılık fazla ise kötü sonuçlar verebilmektedir. (Scaling (geometry), 2020)

### Piksel İnterpolasyonu

Piksel interpolasyonu istatistiksel yöntemlerden yararlanarak komşu pikselleri temsil edecek piksel değerini hesaplamak için kullanılan bir yöntemdir. Şekil 2.1'de görüntü küçültme yöntemlerinin örneği verilmiştir. Piksel değiştirme yönteminde görüntü üzerinde dörtlü komşu piksellerden sol üst köşedeki pikseller seçilerek küçültme işlemi gerçekleştirilmiştir. Piksel interpolasyonunda ise dörtlü komşu piksellerin ortalamaları alınarak küçültme işlemi gerçekleştirilmiştir. (Çayıroğlu, 2020)



Şekil 2.1 Görüntü Küçültme İşlemi

#### 2.3.1.2. Büyütme

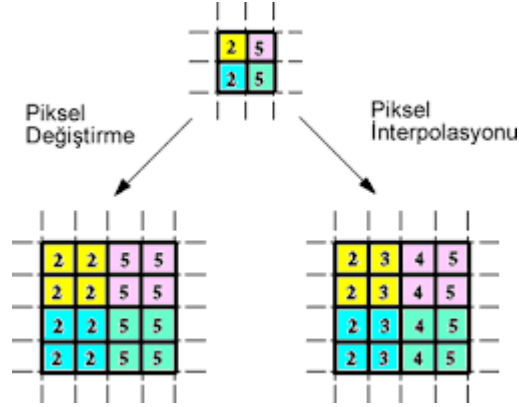
Görüntünün belirli bir alanı veya tamamını büyütmek için kullanılan yöntemdir. Görüntü büyültmede piksel değiştire veya piksel interpolasyonu olmak üzere iki farklı yöntem bulunmaktadır.

### Piksel Değiştirme

İstenilen matris büyüklüğüne göre ele alınan piksel değeri değiştirilmeden çoğaltılarak büyütme işlemi gerçekleştirilmektedir.

### Piksel İnterpolasyonu

Piksel interpolasyonu yönteminde istenilen matris büyüklüğüne bağlı kalınarak pikseller arası fark oluşturacak şekilde yeni piksel değerleri belirlemektedir. Böylece daha az keskinliğe sahip piksel geçişleri oluşmaktadır. Piksel interpolasyonunda pikseller arasındaki keskinliği azaltmak için piksel değerleri arasında minimum fark olacak şekilde değerler seçilmiştir. Piksel değiştirmede ise istenilen matris büyüklüğü kadar piksel çoğaltılmıştır. (Çayıroğlu, 2020)



Şekil 2.2 Görüntü Küçültme İşlemi

## BÖLÜM ÜÇ

### UYGULAMA

#### 3.1. Küçültme-Piksel Değiştirme Metodu

Giriş görüntüsünü x ve y yönünde birer piksel atlanarak okunursa ve okunan değerleri yeni görüntüde sırayla yerleştirilirse  $1/2x$ 'lik bir küçültme sağlamış olunur. Döngüdeki n değeri küçültme katsayısı olacaktır.

```
import numpy as np
import cv2

# Gösterme Fonksiyonu
def goster(resim): cv2.imshow("resim", resim);cv2.waitKey(0); cv2.destroyAllWindows();
cv2.waitKey(1)
img= cv2.imread("indir.png")
goster(img)

# Çözünürlük
r= np.shape(img)[0]
c= np.shape(img)[1]
print(r,c)
n= 1.25
h= round(r/n)
w= round(c/n)

ims= np.zeros((h, w, 3), np.uint8) #Boş Görüntü oluşturma
for i in range(0,h):
    for j in range(0,w):
        ims[i,j]=img[round(i*n),round(j*n)]

cv2.imwrite("ims1.png",ims)
goster(ims)
```



Şekil 3.1 Orijinal Görüntü



Şekil 3.2 1.25 Kat Piksel Değiştirme ile Küçültülmüş Görüntü

### 3.2. Küçültme-İnterpolasyon Metodu

Giriş resmini x ve y yönünde küçültme katsayısı kadar atlanarak okunursa ve her atladığında aradaki piksellerin ortalama değerini alınırsa ve bu ortalama değerleri çıkış resmine aktarılırsa  $1/x$ 'lik bir küçültme sağlamış olunur. Döngüdeki n değeri küçültme katsayısı olacaktır.

```
# Küçültme İnterpolasyon
n= 1.25
h= round(r/n)
w= round(c/n)

ims= np.zeros((h, w, 3))

# Fonksiyon
def bgrmean(i,j,n,img):
    total = [[0,0,0]]
    for m in range(round(i*n),round((i+1)*n)):
        for n in range(round(j*n),round((j+1)*n)):
            total=np.vstack(total, img[m,n])
    return(np.round(np.mean(total,axis=0)))

# Döngü
for i in range(0,h):
    for j in range(0,w):
        ims[i,j]= bgrmean(i,j,n,img)

cv2.imwrite("ims2.png",ims)
goster(ims)
```



Şekil 3.3 Orijinal Görüntü



Şekil 3.4 1.25 Kat Piksel İnterpolasyonu ile Küçültülmüş Görüntü

### 3.3. Büyültme-Piksel Değişirme Metodu

Giriş görüntüsünü x ve y yönündeki her bir piksel sırayla okunursa ve okunan her bir değer yeni görüntüde birer piksel atlanarak yerleştirilip ve ardından alınan değerler yeni görüntüdeki boş piksellere atanırsa 2x'lik bir büyültme sağlamış olunur. Döngüdeki n değeri büyültme katsayısı olacaktır.

```
# Okuma
img= cv2.imread("ims1.png")
goster(img)

# Çözünürlük
r= np.shape(img)[0]
c= np.shape(img)[1]
print(r,c)

n= 1.25
h= round(r*n)-1
w= round(c*n)-1

#Boş Görüntü oluşturma
imb= np.zeros((h, w, 3), np.uint8)

for i in range(0,h):
    for j in range(0,w):
        imb[i,j]=img[round(i/n),round(j/n)]

cv2.imwrite("imb1.png",imb)
goster(imb)
```





Şekil 3.5 Orijinal Görüntü



Şekil 3.6 1.25 Kat Piksel Değiştirme ile Büyültülmüş Görüntü

### 3.4. Büyültme-Piksel İnterpolasyonu Metodu

Giriş görüntüsünü x ve y yönündeki her bir piksel sırayla okunursa ve okunan her bir değer yeni görüntüde birer piksel atlanarak yerleştirilip ve ardından alınan değerler ile interpolasyon yapıp yeni görüntüdeki orijinal görüntüden gelen pikseller arasında kalan boş piksellere atandıktan sonra yeni atanmış piksellerin interpolasyonu ile bu pikseller arasında kalan pikseller atanırsa 2x'lik bir büyültme sağlamış olunur. Döngüdeki n değeri büyültme katsayısı olacaktır.

```
# Okuma
img= cv2.imread("ims1.png")
goster(img)

# Çözünürlük
r= np.shape(img)[0]
c= np.shape(img)[1]
print(r,c)

n= 1.25
h= round(r*n)
w= round(c*n)

# Aynı Değerleri Atama ve Orijinal Noktalar Arası İnterpolasyonu Döngüsü
for i in range (0,h):
    for j in range (0,w):
        if (i%n==0) and (j%n==0):
            imn[i,j]=img[(i//n),(j//n)]
        elif (i%n==0):
            aa = np.multiply((n-(j%n))/n,img[(i//n),(j-(j%n))//n])
            bb = np.multiply((j%n)/n,img[(i//n),(j+n-(j%n))//n-1])
            imn[i,j]=np.sum((aa,bb),axis=0)
        elif (j%n==0):
            aa = np.multiply((n-(i%n))/n,img[(i-(i%n))//n,j//n])
            bb = np.multiply((i%n)/n,img[(i+n-(i%n))//n-1,j//n])
            imn[i,j]=np.sum((aa,bb),axis=0)
```

```
# Yeni Atanmış Noktalar Arası İnterpolasyonu Döngüsü
```

```
for i in range(0,h):
```

```
    if (i%n!=0):
```

```
        for j in range (0,w):
```

```
            if (j%n != 0):
```

```
                u=np.multiply((n-i%n)/2/n,img[(i-i%n)//n-1,j//n])
```

```
                d=np.multiply((i%n)/2/n,img[(i+n-i%n)//n-1,j//n])
```

```
                l=np.multiply((n-j%n)/2/n,img[i//n,(j-j%n)//n-1])
```

```
                r=np.multiply((j%n)/2/n,img[i//n,(j+n-j%n)//n-1])
```

```
                imn[i,j]=np.sum([u,d,l,r], axis=0)
```

```
goster(imn)
```



Şekil 3.7 Orijinal Görüntü



Şekil 3.8 1.25 Kat Piksel İnterpolasyonu ile  
Büyültülmüş Görüntü

## BÖLÜM DÖRT

### SONUÇ

Python programla dili ile “Görüntü İşleme” projesi kapsamında, Numpy, OpenCV, Matplotlib, Pillow kütüphaneleri kullanılmıştır. OpenCV kütüphanesi yardımıyla Python üzerinde üçgen ve daire gibi geometrik şekiller for döngüsü yardımıyla çizilmiştir. Bu çizilen şekiller üzerinde grayscale dönüşüm yapılarak edge detection uygulanmıştır. Ayrıca elektronik bir ortamda çizilen renkli bir görüntü, dijital olarak Python üzerinde okutulmuştur. Bu okutulan görüntü Kırmızı, Yeşil ve Mavi (RGB) renklerinden oluştuğu için çift katlı array veri yapısında saklanmıştır. Bu görüntü üzerinde Numpy kütüphanesi yardımıyla küçültme ve büyültme yöntemlerini uygulamak için çarpma, toplama gibi matris işlemleri yapılmıştır.

Piksel interpolasyonu ile küçültme yönteminde Numpy kütüphanesinde yer alan `np.mean()` fonksiyonu ile orijinal görüntüdeki piksel değerlerinin RGB renk ortalamaları alınmıştır. Bu alınan ortalama değerler yeni görüntüye atanmıştır. Piksel interpolasyonu ile büyültme yönteminde ise orijinal görüntüdeki pikseller yeni görüntüye aktarılmış, yatay ve dikey olarak bu piksellerin arasındaki boş piksellere uzaklıklarına göre bilineer interpolasyon uygulanarak atama işlemi yapılmış ve ardından bu piksellerin arasında kalan boş pikseller de yeni piksellerin bilineer interpolasyonu ile atanmıştır. Bu sayede tek boyut üzerinde çalışan bilineer interpolasyonu iki kez uygulanarak iki boyutlu olan görüntülerdeki uygulaması sağlanmıştır.

Python üzerinde OpenCV gibi kütüphaneler kullanarak görüntü işleme kapsamında anlık kamera görüntüleri üzerindeki canlı ve cansız nesnelerin tespiti, bu nesnelerin davranışları veya durumları tespit edilebilir. Amazon firmasının bazı satış mağazalarında çalışan olmadan sadece görüntü işleme yardımı ile gelen müşterilerin aldıkları ürünleri tespit edilmektedir. Ziraat alanında bitki üzerinde hastalık olup olmadığı olan hastalığın ne olduğunun tespitinde de yine görüntü işleme kullanılmaktadır. Bu gibi projeler detaylandırılabilir ve üzerinde çalışılabilir.

## BÖLÜM BEŞ

### KAYNAKÇA

- (2020, Ağustos 1). matplotlib.org: <https://matplotlib.org/> adresinden alındı
- (2020). OpenCv.org: <https://opencv.org/about/> adresinden alındı
- Aksoy, B. (2020). *Python ile İmgeden Veriye Görüntü İşleme ve Uygulamaları*. NOBEL Akademik Yayıncılık.
- BAYRAM, P. B. (2019, Mart 5). Yıldız Teknik Üniversitesi Akademik Veri Yönetim Sistemi: <https://avesis.yildiz.edu.tr/bayram> adresinden alındı
- Çayıroğlu, İ. (2020). *Görüntü İşleme (3.Hafta)*. <http://www.ibrahimcayiroglu.com/> adresinden alındı
- Neuman, M., Sapirstein, H., Shwedyk, E., & Bushuk, W. (1989). *Wheat grain colour analysis by digital image processing II. Wheat class discrimination*. Journal of Cereal Science.
- Numpy Kütüphanesi Nedir? Nasıl Kullanılır?* (2020, Ağustos 1). teknoloji.org: <https://teknoloji.org/numpy-kutuphanesi-nedir-nasil-kullanilir/> adresinden alındı
- Rush, J. C. (2016). *The Image Processing Handbook*. CRC Press.
- SAMTAŞ, G., & GÜLESİN, M. (2011). Sayısal Görüntü İşleme ve Farklı Alanlardaki Uygulamaları. *Electronic Journal of Vocational Colleges*, 85-97.
- Scaling (geometry)*. (2020, Haziran 15). Wikipedia: [https://en.wikipedia.org/wiki/Scaling\\_\(geometry\)](https://en.wikipedia.org/wiki/Scaling_(geometry)) adresinden alındı
- Young, I., Gerbrands, J., & Vliet, L. (2007). *Fundamentals of Image Processing*.