

# Telea ve Navier Stokes Algoritmaları Kullanılarak Görüntülerdeki Bozulmaları Düzeltme

## Using Telea and Navier Stokes Algorithms Fixing Distortions on Images

Engin AKKOCA  
Namık Kemal Üniversitesi,  
Bilgisayar Mühendisliği Bölümü, Tekirdağ  
Türkiye  
enginakkoca@hotmail.com  
ORCID: 0000-0001-7005-6489

Ercan BULUŞ  
Namık Kemal Üniversitesi,  
Bilgisayar Mühendisliği Bölümü, Tekirdağ  
Türkiye  
ercanbulus@nku.edu.tr  
ORCID: 0000-0002-0902-1903

### Öz

Görseller üzerindeki bozulmaları düzeltmek veya görsel üzerindeki istenilmeyen bazı kısımları, görselin orijinal halini bilmeyen kişilerin algılayamayacağı şekilde kaldırmak veya değiştirmek insanların çok uzun zamandır talep ettiği işlemlerdir. Bilgisayarların bu işlemler için kullanılması hem işlemin kalitesini arttırmış hem de işlemi kolaylaştırmıştır, fakat bilgisayar ortamında yapılıyor olsa da görsel üzerindeki işlemler halen manuel olarak yapılmaktadır. Görüntü boyama (Image Inpainting) yöntemi ile bu işlem hem daha hızlı yapılmaya başlanmış hem de işlem otomatikleştirilmiştir. Open CV kütüphanesi için geliştirilen inpaint\_telea ve inpaint\_ns sınıfları ile görsel üzerinde görüntü boyama işlemi yapılabilmektedir.

**Anahtar Sözcükler:** Görüntü Boyama, Telea Algoritması, Navier Stokes Algoritması

### Abstract

People have long desired editing the defects on images and, removing or altering the corrupt regions of image data in a way that would not be detected by the neutral observer unaware of the original copy. Utilizing computers for such procedures have not only improved the quality of the work, but also eased the processes. However, notwithstanding in a computing environment, the procedures are still conducted manually. Image Inpainting method has enabled a

Gönderme ve kabul tarihi: 24.09.2020 - 10.09.2021

Makale türü: Derleme

faster and an automatic process. Video Inpainting on images can be done by using inpaint\_telea and inpaint\_ns classes, which are both developed for Open CV Library.

**Keywords:** Image Inpainting, Telea Algorithm, Navier Stokes Algorithm

### 1. Giriş

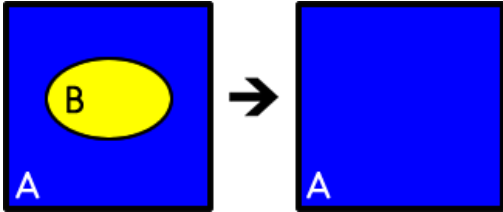
Görseller üzerinden istenilmeyen bir nesne veya zarar görmüş bir bölgenin kaldırılması için farklı yöntemler kullanılabilir. Yaygın olarak kullanılan yöntemde aynı görsel veya farklı bir görsel üzerinden, işlem yapmak istenilen alanın çevresi ile uyumlu bir bölge kopyalanır ve kaldırılmak istenilen bölgenin üzerinde yapıştırılarak istenilmeyen alan yok edilmeye çalışılır. Bu yöntem görüntü işleme programları ile manuel olarak çalışma yapan kullanıcılar tarafından basitliği sebebiyle çok sık kullanılır fakat yüz görüntüleri veya karmaşık görüntüler için uygun bir yöntem değildir. Ayrıca yapılan işlemin kalitesi kullanıcının görsel yeteneği ile doğru orantılıdır. Diğer bir yöntem ise görüntü boyama tekniğidir ki kullanıcının yetenek düzeyini aradan çıkarıp bahse konu işlemleri otomatik ve hızlı yapmasının yanında karmaşık görüntülerde çok daha iyi sonuç vermektedir. Görüntü boyama, görüntünün hem doku hem de yapı bileşenleri üzerinde sayısal hesaplamalar yaparak kaldırılmak istenilen nesneyi görselden çıkartabilir.

Görüntü boyama işlemi aslen restoratorleri zarar görmüş tarihi eserleri restore ederken uzun zamandan beri kullandıkları bir teknik olmasına rağmen, görüntü işleme tekniklerindeki son gelişmeler ve işlemin bilgisayar ortamında daha hızlı ve kaliteli

yapılabilmesi nedeniyle daha da popülerlik kazanmıştır. Görüntü işleme araçlarının iyileştirilmesi ve dijital görüntü düzenlemenin esnekliği ile otomatik görüntü boyama, bilgisayar görüşünde önemli uygulamalar bulmuş ve görüntü işlemede önemli bir araştırma konusu olmuştur.

Ancak, herhangi bir iz bırakmadan bir görüntüyü düzenleyebilen düzenleme uygulamaları, güvenlik için bir sorun oluşturmaktadır. Bu nedenle, mevcut olan gerçek görüntüyü tespit etmek ve çıkarmak için otomatik bir sisteme ihtiyaç duyulması acil bir taleptir. Görüntü boyama işlemi yapılan görüntülerin işlemden sonra kullanılabilirliği büyük ölçüde orijinal görüntünün çıkarılma mekanizmasına bağlıdır, bu nedenle görüntülerden nesne kaldırılması bilgi güvenliği için oldukça önemli bir konudur [1]. Bir görsel üzerinde herhangi bir teknik ile oynama yapıp yapılmadığının insan gözü ile anlaşılacağı noktalarda otomatik olarak işlemin tespit edilebilmesi amacı ile yapılan çalışmalar literatürde mevcuttur.

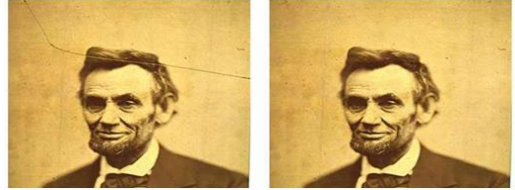
Görüntü boyama; görüntünün istenmeyen bir bölümünü, bu bölüm dışında kalan görüntü bölgesinden sağlanan uygun imgeciklerle doldurulup kişinin istekleri doğrultusunda yeni bir görüntünün elde edilmesi işlemidir. Görüntü boyama yapılırken özgün görüntüde yapılan değişikliklerin görüntünün aslını bilmeyen bir kişinin algılayamayacağı kadar bu özgün görüntüyle uyumlu olması amaçlanır [2]. Görüntü üzerindeki istenilmeyen bölüm görüntünün bir parçası olabileceği gibi eski ve zarar görmüş bir resmin üzerinde bulunan zarar görmüş bölümler de olabilir. Görüntü boyama ilk olarak sanat dünyasında eski tabloların onarılması amacıyla ortaya çıkmıştır. Görüntü fotoğrafçılığına yönelik dijital fotoğrafçılıkla standart bir araç haline gelmiştir [3].



**Şekil-1:** Görüntü boyama işleminin şematik gösterimi

Şekil 1.'de yer alan mavi renkli A görseli üzerinde bulunan sarı renkli B bölgesi kişinin görsel üzerinde istemediği bir nesne olabilir veya A görselinin zarar görmesi ile oluşan istenilmeyen bir leke olabilir.

Görüntü boyama işlemi sonucunda sarı renkli bölgenin görseli daha önceden görmeyen bir kişinin algılayamayacağı şekilde görselden çıkartılması hedeflenmektedir. Şekil 2.'de eski bir resimdeki deformasyonların görüntü boyama işleminden sonra nasıl giderildiğinin bir örneği görülmektedir.



**Şekil-2:** Görüntü boyama örneği

Görüntü boyama işlemi sadece resim veya fotoğraf gibi görseller üzerinde yapılan işlem olarak düşünülmemelidir. Eski video görüntüleri üzerindeki bozulmaları gidermek, video görüntüleri üzerinden istenmeyen nesnelere veya logoları kaldırmak veya görüntüleri sıkıştırmak için de yaygın olarak kullanılmaktadır.

## 2. Literatür Özeti

Görüntü boyama teknolojisi üzerine yapılan araştırmalar esas olarak üç gruba ayrılabilir. Birinci grup çalışmalar, bazı küçük görüntü çiziklerini onarmak için varyasyonel kısmi fark denklemlerine dayanarak boyama işlemi yapılan çalışmaları içerir. İkinci grup çalışmalar, bazı büyük görüntü bölgelerini doldurmak için doku sentez teknolojisine dayanarak boyama yapılan çalışmaları içerir ve üçüncü grup çalışmalar ise önceki iki grup çalışmanın birleşimi olarak düşünülebilir [2].

Son on yılda, matematikçiler veya bilgisayar bilimcileri tarafından birçok boyama yöntemi yaygın olarak geliştirilmiş konudur. Genel olarak bakılırsa, boyama yöntemleri iki kategoriye ayrılır: geometrik yöntemler (doku olmayan) ve doku tabanlı yöntemler [4]. Geometrik yöntemler, yalnızca sınır piksellerinin kullanılabilirliğini gösteren hasarlı alanları sınırdan merkeze geri yükler. Geometrik yöntemlerin kısıtlamaları şunlardır:

- Yumuşak restorasyon sonuçları.
- Geniş alanlarda zayıf performans.
- Uzun hesaplama süresi.

Doku tabanlı yöntemler ise kendi kendine tutarlılık özelliklerini keşfetmekle başlar ve daha sonra en iyi doku yamasını restorasyon sözlüğünden eksik alana kopyalar [4].

D. Calvetti ve arkadaşlarının yaptığı çalışmada gerçek görüntü keskin kenarlar içermiyorsa, boyamanın izotropik pürüzsüzlük ile oldukça tatmin edici bir şekilde yapılabileceği gösterilmiştir[5].

S.S. Wang ve S.L. Tsai görüntünün değiştirilen bölgesinin tam otomatik bir şekilde algılanıp kurtarılabilceği yeni bir otomatik görüntü doğrulama ve kurtarma şeması önermişlerdir. Yüksek kaliteli görüntüleri kurtarmak için görüntüdeki önemli bölgeyi seçip fraktal kodlama gerçekleştirmişler ve sonrasında fligranlı görüntüyü elde etmek için fligranın kodlarını orijinal görüntüye eklemişlerdir. Bu yöntemle fligranlı görüntü değiştirilmiş ise otomatik olarak görüntüyü fligran eklemişlerdir [6].

H. Li ve arkadaşları sahne dönüşümü ve renk transferine dayalı olarak stc-inpainting adlı bir görüntü boyama algoritması önermişlerdir. İlk olarak, hasarlı görüntüye en çok benzeyen görünümü içeren bir kaynak görüntü, doku, renk ve yapısal bilgi gibi özelliklerle karşılaştırılarak görüntü veritabanında aranmış, ardından, kaynak görüntüden uygun nesne seçilerek hasarlı bölgedeki sahne dolgusu tamamlanmıştır [7].

R. Köhler ve arkadaşları eksik piksellerden dolayı bozulmuş görüntü yamalarından tam görüntü yamalarına doğrudan, büyük bir görüntü veri seti üzerinde otomatik olarak eğitilip yapay sinir ağı olarak temsil edilen bir eşleme önermişlerdir. Özellikler yaygın olarak göz ardı edilen eksik bölgelerin, yani maskelerin şekil bilgilerinin kullanılmasının faydalı olup olmadığını değerlendirmişlerdir [8]. Bu çalışma görüntü boyama tekniklerine ek olarak yapay zeka tekniklerinin kullanılması ile ön plana çıkmaktadır.

### 3. Gerçekleştirilen Çalışma

Bu çalışmada Python programlama dilinde OpenCV kütüphanesi kullanılarak görüntü boyama işleminin nasıl gerçekleştirildiği anlatılmış ve görüntü boyama işlemi için kullanılan Telea ve Navier Stokes algoritmaları ile aynı görsel üzerinde uygulama yapılarak iki algoritma karşılaştırılmıştır. OpenCV kütüphanesinin kullanıcılarına sunduğu Telea ve Navier Stokes algoritmalarına değinmeden önce OpenCV kütüphanesi hakkında genel bir bilgi vermekte fayda vardır.

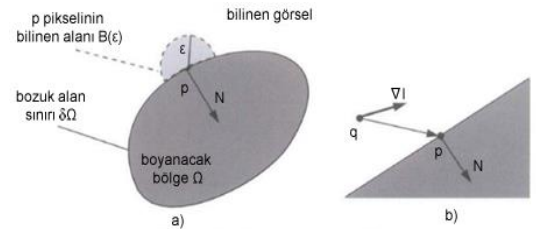
#### 3.1 OpenCv

OpenCV (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesidir [13].

1999 yılında İntel tarafından geliştirilmeye başlanmış daha sonra Itseez, Willow, Nvidia, AMD, Google gibi şirket ve toplulukların desteği ile gelişim süreci devam etmiştir. İlk sürüm olan OpenCV alfa 2000 yılında piyasaya çıkmıştır. İlk etapta C programlama dili ile geliştirilmeye başlanmış ve daha sonra birçok algoritması C++ dili ile geliştirilmiştir. Açık kaynak kodlu bir kütüphanedir ve BSD lisansı altında geliştirilmektedir. BSD lisansına sahip olması bu kütüphaneyi istediğiniz projede ücretsiz olarak kullanabileceğiniz anlamına gelmektedir. OpenCV kütüphanesi 2500den fazla optimize edilmiş algoritmaya sahiptir. OpenCV kütüphanesinde yer alan görüntü işleme algoritmaları ile nesne algulama ve tanıma, hareket yakalama gibi birçok uygulama gerçekleştirilebilmektedir [14]. OpenCV kütüphanesi görüntü boyama işlemleri için “INPAINT\_TELEA” ve “INPAINT\_NS” isimli iki adet algoritmaya sahiptir.

#### 3.1.1 Telea Algoritması

Telea algoritması 2004 yılında Alexandru Telea tarafından yazılan “An Image Inpainting Technique Based on the Fast Marching Method” (Hızlı Yürüyüş Metoduna Dayalı Görüntü Boyama Tekniği) makalesi temel alınarak geliştirilmiştir[9]. Algoritmanın çalışma mantığı kısaca şöyle açıklanabilir: Resimdeki boyanacak bölgenin sınırından başlayarak, önce sınırdaki alanlar yavaş yavaş doldurularak bölgenin içine doğru girilir. Boyanacak piksel etrafında bilinen tüm piksellerin normalleştirilmiş ağırlıklı toplamıyla değiştirilir. Algoritmadaki en önemli kısım nokta seçimi konusudur. Noktanın yakınında, sınırın normaline yakın olan piksellere ve sınır konturlarında yatan piksellere daha fazla ağırlık verilir. Bir piksel boyandığında, aynı yöntem kullanılarak bir sonraki en yakın piksele geçilir. Hızlı yürüyüş metodu önce bilinen piksellerin yakınındaki piksellerin boyanmasını sağlar, böylece sadece manuel bir sezgisel işlem gibi çalışır.



Şekil-3: Görüntü Boyama Prensipli[9]

Telea algoritmasını daha iyi anlamak için Şekil 3.

incelenabilir. Şekil 3. a)'da görülen koyu gri alan boyanmak istenilen alandır ve bu alanın sınıır bölgesi üzerinde bulunan p pikselini boyamak için, görselin boyanmayacak alanından, p pikseli için B(ε) alanı belirlenir. Daha sonra bu alan içinde bulunan her piksel için denklem (1) ile ağırlık değerleri hesaplanır.

$$I(p) = \frac{\sum_{q \in B_\epsilon(p)} w(p,q)[I(q) + \nabla I(q)(p-q)]}{\sum_{q \in B_\epsilon(p)} w(p,q)} \quad (1)$$

B(ε) alanındaki bütün pikseller için ağırlık değerleri hesaplandıktan sonra Şekil 4 ' de verilen algoritma ile p pikseli boyanır. Algoritmada bulunan I(i, j) (1)'de verilen denkleme göre yapılır. Görselin gradyanı, *gradI*, algoritmada bulunan matematiksel işlem ile tahmin edilir. Boyanmak istenilen piksel ile alakalı olarak *dir*, *dst*, *lev* değişkenleri hesaplandıktan sonra bu değişkenler kullanılarak, pürüzsüz bölgeleri ve kesin görüntü ayrıntılarını boyanmış bölgeye yaymak için kullanılacak olan ağırlık değeri olan, w değişkeni hesaplanır. Kodda yer alan *dir* değişkeni boyanmak istenilen piksele yakın olan piksellerin işleme katkısının daha fazla olmasını sağlamak için kullanılırken, *dst* değişkeni ise piksellerin katsayısını geometrik olarak ayarlamak için kullanılır. Boyanmak istenilen pikselin rengine ve yoğunluğuna yakın piksellerin daha çok dikkate alınması için ise *lev* hesaplaması kullanılır. Daha öncede bahsedildiği gibi bu üç değere bağlı olarak ağırlık hesaplaması yapılır.

```
void inpaint(int i,int j)
{
  for (all (k,l) in B_ε(i,j) such that f(k,l)!=KNOWN)
  {
    r = vector from (i,j) to (k,l);
    dir = r * gradT(i,j)/length(r);
    dst = 1/(length(r)*length(r));
    lev = 1/(1+fabs(T(k,l)- T(i,j)));
    w = dir*dst*lev;
    if (f(k+1,l)==KNOWN && f(k-1,l)==KNOWN &&
        f(k,l+1)==KNOWN && f(k,l-1)==KNOWN)
      gradI = (I(k+1,l)- I(k-1,l), I(k,l+1)- I(k,l-1));
    Ia += w * (I(k,l) + gradI * r);
    s += w;
  }
  I(i,j) = Ia/s;
}
```

Şekil-4: p pikselinin boyanması algoritması[9]

### 3.1.2 Navier\_Stokes Algoritması

Navier-Stokes algoritması 2001 yılında Bertalmio, Marcelo, Andrea L. Bertozzi ve Guillermo Sapiro'nun "Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting" (Navier-Stokes, Akışkanlar Dinamiği ve Görüntü ve Video Boyama) makalesine

dayanmaktadır (Bertalmio ve diğerleri, 2001). Bu algoritma akışkan dinamiğini ve kullanımlarını temel almaktadır. İlk olarak bilinen bölgelerden bilinmeyen bölgelere kenarlar boyunca ilerler. Boyayan bölgenin sınırındaki gradyan vektörleri eşleştirilirken aynı yoğunluktaki noktaları birleştirerek devam eder. Bir sıvının girdabına kapılmış nesne gibi dıştan içe doğru dairesel hareketlerle boyama işlemi yapılır. Algoritma, profesyonel restoratörler tarafından kullanılan temel yaklaşımları taklit etmeye çalışır. Algoritma ayrıca görüntünün hem degrade yönünü (geometri) hem de gri değerlerini (fotometri) doldurulacak deliği çevreleyen bir bantta yaymanın önemini ortaya koymaktadır [10].

## 4. Uygulamalar

OpenCV kütüphanesi ile görüntü boyama işlemi yaparken girdi olarak kullanmak için iki adet görsele ihtiyaç vardır:

1. Üzerinde deformasyon olan ve onarmak istediğimiz görsel veya üzerindeki bir nesnenin ve bölgenin kaldırılmasını istediğimiz görsel. Yani görselin orijinal hali.
2. Orijinal görsel üzerinde hasarın veya kaldırılmak istenilen bölgenin nerede olduğunu gösteren maske görüntüsü. Maske görüntüsü orijinal görsel ile aynı genişlik ve yükseklik değerlerine sahip olmalıdır. Maske görüntüsündeki sıfır olmayan yani siyah olmayan pikseller boyanması gereken alanları gösterirken sıfır olan pikseller yani siyah alanlar ise orijinal görsel üzerindeki boyama işlemi yapılmayacak alanları gösterir.

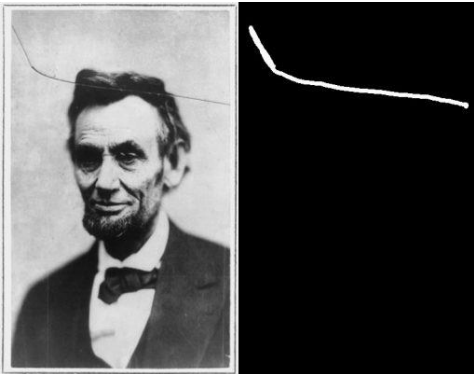
Şekil 5.'de orijinal girdi görseli ve maske örneği görülmektedir. Sol taraftaki orijinal görselde yer alan katlama veya yırtılmaya dayalı deformasyon açık bir şekilde görülmektedir. Sağ tarafta yer alan maske görselinde ise bozuk olmayan alanlara ait pikseller sıfır olarak ayarlanmış, bozuk alana ait pikseller ise sıfırdan farklı olarak ayarlanmıştır. Dolayısı ile bozuk olmayan alanlar maskede siyah olarak görülmektedir.

### 4.1 Maske Oluşturma

Maskeyi oluşturmak için photoshop, fireworks gibi görüntü işleme programları kullanılabileceği gibi literatürde maskeyi otomatik oluşturma üzerine çalışmalar mevcuttur. Maskeyi görüntü işleme programları ile oluşturmak manuel bir işlem olduğu için hem zaman almaktadır hem de zaman zaman istenilen hassasiyette maske oluşturulamamaktadır.

Maskenin istenilen hassasiyette oluşturulamaması, oluşturulan maske istenilen maske görüntüsünden çok farklı değil ise problem yaratmayacaktır. Çünkü kaliteli bir boyama işlemi yapılacağından görselin boyanmaması gereken alanları boyansa bile yabancı bir göz bunu anlayamayacaktır ama yine de maskenin boyanmak istenilen alana en uygun biçimde hazırlanması önem arz etmektedir. Her ne sebeple olursa olsun boyanacak alanın büyümesi işlemin doğallık seviyesini düşürecektir. Maske oluştururken manuel yöntemler kullanmak yerine literatürde bulunan otomatik yöntemler kullanılabilir. M. Isogawa ve arkadaşları insan emeğini azaltmak ve iyi boyama sonuçlarının otomatik olarak elde edilebilmesi amacıyla maskelenmiş bölge optimizasyonu için bir yöntem önermişlerdir. Boyasız görüntülerdeki tüm süper pikseller için boyamanın doğallığını tahmin edip orijinal piksel maskesini süper piksel temelinde yeniden şekillendirmişlerdir ve böylece boyamanın sonucunun doğallığını artırmışlardır. Bu yaklaşımın etkinliği görüntü boyama algoritmalarına bağlı değildir, bu nedenle her görüntü boyama yöntemi için bir eklenti olarak uygulanabilir[11].

Benzer şekilde W. Zhang ve arkadaşları hasar bölgesini etkili bir şekilde bölümlere ayırmak için hasarlı bölgenin sınırını doğru bir şekilde bulabilen ve daha sonra görüntü boyamanın nihai etkisini büyük ölçüde geliştirebilen bir algoritma önermişlerdir. Deneysel sonuçlar, özellikle göze çarpmayan sınırlara sahip hasarlı bölge için, boyama yönteminin oluşturulması aşamasında önerilen yöntemin en gelişmiş yöntemlerden üstün olduğunu göstermektedir[12]



**Şekil-5:** Görüntü boyama işlemi için orijinal girdi görseli ve maske görseli

```

1. import cv2
2. gorsel = cv2.imread("gorseller/foto01.png")
3. maske = cv2.imread("gorseller/maske01.png")
4. maske = cv2.cvtColor(maske, cv2.COLOR_BGR2GRAY)
5. cikti_gorseli = cv2.inpaint(gorsel, maske, 3, flags=cv2.INPAINT_TELEA)
6. cv2.imshow("Orijinal Gorsel", gorsel)
7. cv2.imshow("Maske", maske)

```

**Şekil-6:** Telea Algoritması ile görüntü boyama örneği

## 4.2 OpenCV ve Python İle Boyama İşleminin Uygulanması

Şekil 6’da verilen, Python dilinde yazılmış kod bloğu ile üzerinde istenilmeyen alanlar bulunan bir görselden Telea algoritması ile bu istenilmeyen alanlar çıkartılabilir.

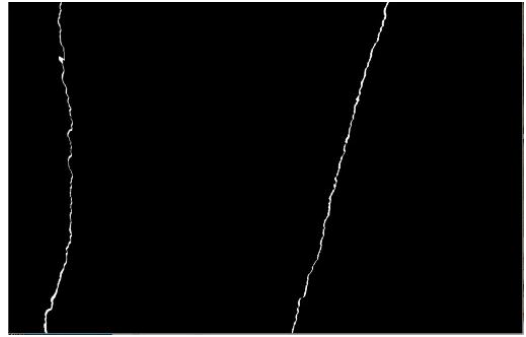
Şekil 6’da verilen program parçasının 1. satırında import cv2 komutu ile OpenCV kütüphanesi programa eklenmiş, 2. ve 3. satırlarında OpenCV kütüphanesinin imread metodu ile görüntü boyama yapılacak görsel ve görüntü boyama işleminde kullanılacak maske iki ayrı nesneye alınmıştır. 4. Satırda OpenCV kütüphanesinin cvtColor metodu ile görüntü boyama işleminde kullanılacak maske gri tonlamalı hale dönüştürülmüştür. Görünü boyama işlemi kod bloğunun

5. satırında OpenCV kütüphanesinin inpaint metodu ile gerçekleştirilmektedir. OpenCV kütüphanesinin inpaint metodu 4 parametre alır. İlk parametre işlem

yapılacak olan orijinal görsel, ikinci parametre görüntü boyama işleminde kullanılacak olan maskedir ki bu iki görsel program parçasının 2. ve 3. satırlarında iki ayrı nesne içerisine alınmıştır. Daha önce de bahsedildiği üzere maske manuel olarak photoshop, fireworks gibi görüntü işleme programları veya [11]'daki veya benzer yöntemler ile otomatik olarak da oluşturulabilir. Üçüncü parametre boyanacak olan pikselin en fazla ne kadar uzağındaki görüntü piksellerinin hesaplamaya katılacağını belirten yarıçap bilgisidir ve varsayılan olarak 3 olarak ayarlanır, fakat görselin üzerindeki kaybedilmek istenilen cisim veya zarar görmüş olanın boyutuna göre farklı değerler kullanılabilir. Son olarak görüntü boyama işleminin hangi algoritma ile yapılacağı bilgisi parametre olarak gönderilir. Eğer Telea algoritması ile görüntü boyama yapılmak isteniliyorsa son parametre olarak `flags=cv2.INPAINT_TELEA`, Navier-Stokes algoritması kullanılmak isteniliyorsa son parametre olarak `flags=cv2.INPAINT_NS` bilgisi girilir. Şekil 6' da kodları verilen görüntü boyama işlemi Telea algoritması ile yapılmıştır. Şekil 7'de görüntü boyama işleminde kullanılan görselin zarar görmeden önceki orijinal hali (a), boyama işlemi için programa girdi olarak gönderilen zarar görmüş hali (b) ve boyama işleminde kullanılacak olan maske görseli (c) görülmektedir.



(b)



(c)

**Şekil-7:** Görüntü boyama işleminde kullanılan görselin zarar görmeden önceki orijinal hali, deforme olmuş hali ve boyama işleminde kullanılan maske.



(a)

Kod bloğunun 6. 7. ve 8. Satırlarında görüntü boyama işlemi için kullanılan görseller ve işlem sonucunda oluşan görsel OpenCV kütüphanesinin imshow metodu ile ekranda gösterilmektedir. Inpaint metodunda ilgili parametre `INPAINT_NS` olarak değiştirilerek aynı görsel üzerinde aynı maske kullanılarak Navier-Stokes algoritması ile boyama işlemi yapılabilir. Aynı zarar görmüş görsel ve aynı maske kullanılarak iki algoritma ile de boyama işlemi gerçekleştirilmiştir. Şekil 8' de Telea algoritması, Şekil 9'da ise Navier-Stokes algoritması ile gerçekleştirilen boyama işlemleri görülmektedir.



**Şekil 8.** Talea algoritması kullanılarak görüntü boyama işlemi sonucunda oluşan görsel.



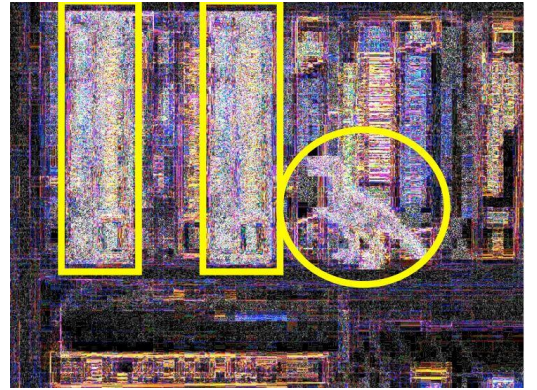
**Şekil-9:** Navier-Stokes algoritması kullanılarak görüntü boyama işlemi sonucunda oluşan görsel.

## 5. Sonuçlar

OpenCV kütüphanesi açık kaynak kodlu olduğu için sürekli gelişmekte ve içerdiği algoritmalar ile görüntü işleme uygulamalarını oldukça kolaylaştırmaktadır. Bu çalışmada, üzerinde yırtılma sonucu deformasyon izleri oluşmuş olan görsel, OpenCV kütüphanesinde yer alan Telea ve Navier Stokes algoritmaları ile ayrı ayrı işlenerek istenmeyen yırtılma izleri otomatik olarak giderilmeye çalışılmıştır. Her iki algoritmada önceki başlıklar altında anlatılan yöntemlerle görsel üzerinde istenilmeyen alanın yerine boyama işlemi yapılmıştır ve birbirlerine yakın sonuçlar elde etmişlerdir. Burada önemli olan algoritmalara maske yardımı ile resim üzerinde boyama işleminin yapılacağı alanın bildirilmesinden sonra boyama işleminin otomatik olarak yapılmasıdır.

Bir görsel üzerinde herhangi bir görüntü işleme programı ile oynama yapıp yapılmadığının tespiti için Hata Seviyesi Analizi (Error Level Analysis, ELA) kullanılabilir. Hata Seviyesi Analizi, farklı

sıkıştırma seviyelerine sahip bir görüntünün parçalarını tanımlamak için adli bir yöntemdir ve JPEG görüntülerin sıkıştırılmasında geçen bir tekniktir [15]. Bu teknik, bir görüntünün dijital olarak değiştirilip değiştirilmediğini belirlemek için kullanılabilir. Şekil 10' da bir görsel üzerine sonradan görüntü işleme yöntemleri ile eklenen nesnelerin Hata Seviyesi Analizi yöntemi ile nasıl tespit edildiği görülmektedir. Görseller dikkatle incelenirse oynama yapılan alanların Hata Seviyesi Analizinde diğer alanlara göre daha çok renklerle işaretlendiği anlaşılmaktadır.



**Şekil 10.** Hata Seviyesi Analizi (Error Level Analysis, ELA) ile görseldeki oynamaların tespiti [15]

Hata Seviyesi Analizi görüntünün kalitesi belirli oranda düşürüp, orijinal görüntü ile farkını alarak görüntü üzerindeki oynamaları tespit eder. Görüntü üzerinde oynama yapılan alanların, orijinal görüntü ile farkları daha fazla çıkacağı için Şekil 10' da görüldüğü gibi oynama yapılan alanlar gözle görülecek kadar belirgin olur. Şekil 11' de Talea

algoritması ile boyama yapılan görselin, Şekil 12' de Navier-Stokes algoritması ile boyama yapılan görselin Hata Seviyesi Analizi sonucunda oluşan çıktıları görülmektedir.



**Şekil 11.** Talea algoritması ile boyama yapılan görselin Hata Seviyesi Analizi çıktısı



**Şekil 12.** Navier Stokes algoritması ile boyama yapılan görselin Hata Seviyesi Analizi çıktısı

Şekil 11 ve Şekil 12 incelendiğinde, Talea ve Navier Stokes algoritmaları ile gerçekleştirilen boyama işlemlerinde, görsel üzerinde yapılan işlemin tespit edilmesinin çok mümkün olmadığı görülmektedir. Hata Seviyesi Analizi çıktılarında sadece kıyafeti farklı olan öğrencinin üzerindeki renkli desenler tespit edilmiş gibi durmaktadır, fakat görselin zarar görmüş hali incelendiğinde bu alanda herhangi bir deforme olmadığı görülmektedir. Yani bu alana boyama işlemi yapılırken dokunulmamıştır, orijinal hali ile kalmıştır. Bu durum Hata Seviyesi Analizi metodunun görsel içindeki renklerden genel olarak farklı olan renklerinde zaman zaman oynama yapılmış gibi tespit edilebileceği anlamına gelmektedir. Bu tarz durumlara karşı görseller dikkatle incelenmelidir.

Hata Seviyesi Analizi sonuçları bilgisayar ortamında

yapılan otomatik görüntü işleme çalışmalarının, manüel çalışmalardan daha hızlı ve daha güvenilir olduğu gerçeğini ortaya koymuştur. Ancak bu durumun adli vakalarda kötüye kullanılabileceği göz ardı edilmemelidir. Özellikler kriminal incelemelerde tek bir inceleme ile hızlı karar verilmemelidir.

## Kaynakça

- [1] R.T. Pushpalwar ve S.H. Bhandari, Image Inpainting Approaches - A Review. IEEE 6th International Conference on Advanced Computing, 2016(6), 340-345. doi : 10.1109/IACC.2016.70
- [2] V.V. Nabiyev ve A.Taşçı, İstenmeyen Cisimlerin Resimden Kaldırılması, Türkiye Bilişim Vakfı Bilgi. Bilim. ve Mühendisliği Derg., vol. 4, no. 1, Jun. 2016, Accessed: Jun. 14, 2020.
- [3] J.Wang, K.Lu, D.Pan, N.He ve B.Bao, Robust object removal with an exemplar-based image inpainting approach, Neurocomputing, 2014, 123(2014), 150-155 doi:10.1016/j.neucom.2013.06.022.
- [4] M.Wang, B.Yan ve K.N. Ngan, An efficient framework for image/video inpainting, Signal Process. Image Commun., vol. 28, no. 7, pp. 753-762, 2013, doi:10.1016/j.image.2013.03.002.
- [5] D.Calvetti, F.Sgallari ve E.Somersalo, Image inpainting with structural bootstrap priors, Image Vis. Comput., vol. 24, no.7, pp.782-793, 2006, doi:10.1016/j.imavis.2006.01.015.
- [6] S.S.Wang ve S.L.Tsai, Automatic image authentication and recovery using fractal code embedding and image inpainting, Pattern Recognit., vol. 41, no. 2, pp. 701-712, 2008, doi: 10.1016/j.patcog.2007.05.012.
- [7] H.Li, S.Wang, W.Zhang ve M.Wu, Image inpainting based on scene transform and color transfer, Pattern Recognit. Lett., vol. 31, no. 7, pp. 582-592, 2010, doi:10.1016/j.patrec.2009.12.012.
- [8] K.Rolf, C.Schuler, B.Sch ve S.Harmeling, Mask-Specific Inpainting with Deep Neural Networks, pp. 523-534, 2014, doi: 10.1007/978-3-319-11752-2.
- [9] A.Telea, An Image Inpainting Technique Based on the Fast Marching Method An Image Inpainting Technique Based on the Fast Marching Method, Journal of Graphics Tools, vol.7651, 2004, doi:10.1080/10867651.2004.10487596.
- [10] M.Bertalmio ve G. Sapiro, Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.355-362, 2001. doi:10.1109/CVPR.2001.990497
- [11] M.Isogawa, D.A.N.Mikami, ve D.Iwai, H. Kimahata, K.Sato, Mask Optimization for Image Inpainting, IEEE Access, vol. PP, no. c, p. 1, 2018, doi: 10.1109/ACCESS.2018.2877401.



- [12] W.Zhang, Y. Ru, H. Meng, M. Liu, X. Ma, L. Wang ve B. Jiang, A Precise-Mask-Based Method for Enhanced Image Inpainting vol. 2016, no. 1, doi: 10.1155/2016/6104196
- [13] E.Erişti, Görüntü İşlemede Yeni Bir Soluk, OPENCV, Akademik Bilişim'10 - XII. Akademik Bilişim Konferansı Bildirileri, pp. 223-229, 2010.
- [14] M.Tenekeci, A.Gümüşcü ve E.Aslan, Görüntüden OpenCV ile Duygu Analizi, Akademik Bilişim'14 - XVI. Akademik Bilişim Konferansı Bildirileri, pp. 861-865, 2014
- [15] Y.M. Djaksana ve A.K. Rivai, Analisis Manipulasi Citra (Image Forgery) Menggunakan Integrasi Metode Error Level Analysis Dan Block Matching. Jurnal Teknologi Informasi, 12(2018), 83-89.