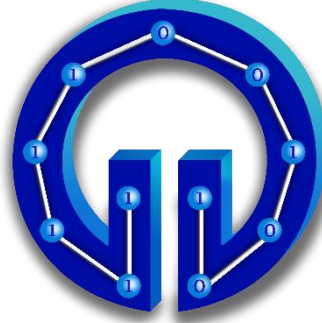


**KARADENİZ TEKNİK ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**İNSAN SURATINDAKİ SAKALIN SAKALSIZ CİLT İLE  
DEĞİŞTİRİLEREK SİLİNMESİ**

**MÜHENDİSLİK TASARIMI**

**Tunahan AYVAZ  
Semih ALİYAZICIOĞLU  
Muhammed Berk AYVAZ**

**2019-2020 GÜZ DÖNEMİ**

**KARADENİZ TEKNİK ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**İNSAN SURATINDAKİ SAKALIN SAKALSIZ CİLT İLE  
DEĞİŞTİRİLEREK SİLİNMESİ**

**MÜHENDİSLİK TASARIMI**

**Tunahan AYVAZ  
Semih ALİYAZICIOĞLU  
Muhammed Berk AYVAZ**

**2019-2020 GÜZ DÖNEMİ**



## **IEEE Etik Kuralları IEEE Code of Ethics**



Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriyi kabul etmek ve eleştiriyi yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayrımcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

## ÖNSÖZ

Üniversite hayatından ancak araştırma ve kendini geliştirme yollarından yüründüğü takdirde en iyi verim alınabilir. Bu yolları üniversite yıllarımız boyunca bizlere gösteren, öğrencilik hayatımız süresince bizlere rehber olan, bizleri araştırmaya ve öğrenmeye sürekli teşvik eden ve “İyi bir mühendis nasıl olur?” sorusunu cevaplamamızı sağlayan kıymetli hocamız **Prof. Dr. Vasif NABİYEV**’e derin saygı ve sevgilerimizi sunarız. Ayrıca projemiz süresince bizden yardımlarını esirgemeyen tüm arkadaşlarımıza ve hocalarımıza teşekkürü borç biliriz.

Tunahan AYVAZ  
Semih ALİYAZICIOĞLU  
Muhammed Berk AYVAZ  
Trabzon 2019

## İÇİNDEKİLER

IEEE ETİK KURALLARI .....	II
ÖNSÖZ.....	III
İÇİNDEKİLER.....	IV
ÖZET.....	V
1. GENEL BİLGİLER.....	1
1.1 Giriş.....	1
2. PROJE TASARIMI.....	2
2.1 Gereksinim Analizi.....	2
2.2 Mimari Tasarım.....	2
2.2.1 Python.....	2
2.2.2 Otsu Algoritması.....	2
2.2.3 Haar Cascade.....	3
2.2.4 Morfolojik İşlemler.....	7
2.2.4.1 Giriş.....	7
2.2.4.2 Renkli Morfolojik Medyan.....	7
2.2.4.3 Algoritma.....	9
2.2.4.4 Morphing Sekansı.....	10
2.2.5 GAN Metodu.....	12
2.2.5.1 Yöntem.....	12
2.2.5.2 Algoritma.....	13
2.2.5.3 GAN Blokları.....	14
2.2.6 İnpaint .....	14
3. UML NESNE MODELİ.....	18

4. KAYNKALAR.....	19
5. STANDARTLAR ve KISITLAR FORMU.....	20

## ÖZET

Bu projede öncelikle Haar Cascade ile kişinin yüz ve göz tespiti sağlanmıştır. Sonrasında ise GAN ve İnpainting metotları yardımıyla insan yüzündeki istenmeyen bir bölgenin bir diğeriyle değiştirilmesi (estetik operasyonlar, sakal silme vs.) hedeflenmiştir. Bu dönüşüm aşamalarında kullanılacak olan morfolojik işlemler ve dönüşümü sağlayacak olan interpolasyon dizisinin oluşturulması hakkında bilgi verilmiştir.

## 1. GENEL BİLGİLER

### 1.1. Giriş

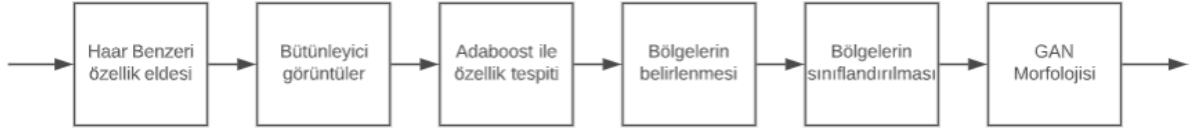
İnsan yüzünün bilgisayar sistemleriyle algılanması 20. yüzyılın ortalarına dayanmaktadır. Yüz algılamanın temelleri 1960’larda Amerikalı bilgisayar bilimcisi ve matematikçi Woody Bledsoe tarafından atılmıştır. O dönemlerde kişilerin eşleştirilmesi dakikalar sürerken günümüz teknolojisi ile bu süre saniyelere indirgenmiştir. Günümüzde güvenlik oldukça önemli bir hal almıştır. Örnek olarak özellikle gümrük, havaalanları, sınır kapıları gibi insan trafiğinin yoğun olduğu noktalarda güvenlik oldukça büyük önem arz etmektedir. Bu projede, bahsedilen bölgelerde kolluk kuvvetleri tarafından aranan kişilerin tanınmamak için çeşitli yollara başvurmuş olmasına rağmen kimlik tespitinin yapılabilmesi amaçlanmıştır. Kimlik tespitinin yapılabilmesi için öncelikle kişinin yüzünün bilgisayar ortamında algılanması gerekmektedir. Bunun için kişinin yüzü üzerinde çeşitli işlemler gerçekleştirilir. Öncelikle kişinin fotoğrafı, fotoğrafın RGB değerlerinin ortalaması alınarak griye (graytone) çevrilir. Sonrasında ise elde edilen bu gri fotoğraf Otsu algoritması kullanılarak ikili (binary) görüntüye dönüştürülür ve söz konusu fotoğrafın histogramı oluşturulur. Elde edilen histogramdaki ilgili değerler Haar Cascade aracılığıyla yorumlanarak kişinin yüz tespiti yapılır. Tespit edilen yüz üzerinde GAN metodu yardımıyla morfolojik işlemler gerçekleştirilerek kişinin yüzündeki tanınmayı engelleyen faktörler (sakal, bıyık, peçe, maske vb.) arındırıldıktan sonra aranan kişinin kimliği kabul edilebilir hata payı vasıtasıyla elde edilir.

Anahtar kelimeler: Haar cascade, morfolojik medyan, GAN, yapay sinir ağları, iç boyama.



## 2. PROJE TASARIMI

Bu projede zengin görüntü işleme kütüphanelerine sahip olan Python programlama dili tercih edilmiştir. Görüntü işleme amacıyla ilgili fonksiyonları içeren OpenCV kütüphanesi kullanılmıştır. Histogram oluşturmak için Otsu algoritmasından, yüz tespiti için Haar Cascade yönteminden yararlanılmıştır. Kişinin yüzünün tanınmasına engel olan faktörlerden arındırmak için ise morfoloji metodu olarak GAN kullanılacaktır. Proje şeması aşağıda gösterildiği gibidir.



### 2.1 Gereksinim Analizi

Projenin gerçekleştirilmesi için temel programlama bilgisi gerekmektedir. Ortadaki sorunun anlaşılabilmesi ve çözülebilmesi adına algoritma ve yapay sinir ağları bilgisi gerekmektedir. İnsan yüzündeki belirli bölgelerin değiştirilmesini sağlayacak olan metotlar için morfolojik işlemlere hakim olunması söz konusudur. Bu işlemleri gerçekleştirmek için gereken veri setlerinden her başlık altında bahsedilmiştir.

### 2.2 Mimari Tasarım

#### 2.2.1 Python

Projenin gerçekleştirilmesi için kullanılan kod PyCharm tümleşik yazılım geliştirme ortamında yazılmış ve derlenip çalıştırılmıştır.



Şekil 1



Şekil 2

#### 2.2.2 Otsu Algoritması

Bir fotoğrafın ikili (binary) görüntüye dönüştürülmesi için geleneksel yöntem bir eşik değeri belirlenip bu değerin altında kalan değerlerin siyaha, üstündekilerin ise beyaza dönüştürülmesini önerir. Fakat her fotoğraf aynı niteliklere sahip olmadığından dolayı geleneksel yöntem görüntünün renk dağılımına uygun bir ikili görüntü oluşturulmasında yetersiz kalmaktadır. Bu sorunun giderilmesi için farklı bir yöntem olan Otsu algoritmasından yararlanılır. Otsu, geleneksel yöntem gibi renkli fotoğraf üzerinde değil, gri seviyedeki

görüntü üzerinde kullanılmaktadır. Bu yöntemin gerçekleştirilmesi için görüntünün ön ve arka plan olmak üzere iki renk sınıfından oluştuğu varsayımında bulunulur.

Tüm eşik değerleri için bu iki sınıfın varyans değerleri hesaplanır. Bu değerlerin en küçük olmasını sağlayan eşik değeri, optimum olarak kabul edilir. Yöntem gri seviyedeki görüntüler üzerinde uygulanır ve görüntü üzerinde hangi renklerin kaç kez kullanıldığının hesaplanmasını sağlar. Bu amaçla öncelikle görüntünün renk histogramı hesaplanır ve gerekli tüm işlemler bu histogram üzerinden gerçekleştirilir.

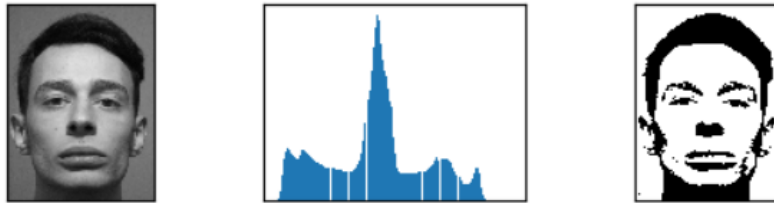
$$\sigma^2 = \sum_{i=1}^N (x_i - \bar{x}_i)^2 Pr(x_i) \quad (1)$$

Varyans, bir sayı dizisinin, bu dizinin aritmetik ortalaması etrafındaki dağılımının yorumlanmasına yardımcı olan bir ölçüdür ve (1)'deki formülle hesaplanır. Otsu algoritmasında sınıf içi varyansın minimum, sınıflar arası varyansın ise maksimum olması beklenir. Sınıflar arası varyansı hesaplarken kullanılacak olan örnek uzayındaki eleman sayısının az olmasından dolayı işlem yapmak daha kolay olacaktır. Bu sebeple işlemler gerçekleştirilirken sınıflar arası varyans değerinden yararlanılır. 0-255 aralığındaki her bir eşik değeri için sınıflar arası varyans değeri hesaplandıktan sonra en büyük sınıflar arası varyans değeri en uygun değer olarak kabul edilir.

```
ret,th = cv.threshold(img,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
```

Tablo 1

Bu projede Tablo 1'de görülen kod parçası ile OpenCV kütüphanesinde bulunan Otsu eşikleme fonksiyonu kullanılarak elde edilen ikili görüntü ve bu görüntüye ait histogram Şekil 1'de belirtilmiştir. Kodda görülen *ret* değişkeni görüntüde en yoğun olarak bulunan değeri, *th* ise Otsu eşiklemeye uğramış ikili görüntüyü temsil etmektedir.

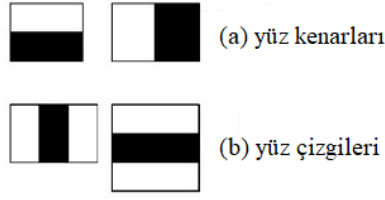


Şekil 3 - Otsu ile histogram ve ikili görüntü eldesi

### 2.2.3 Haar Cascade

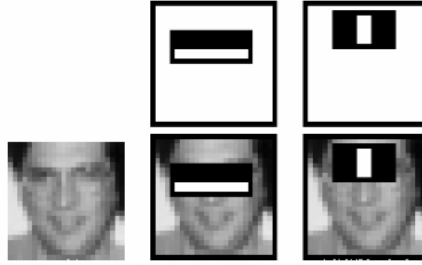
Haar Cascade yöntemi, makine öğrenmesi aracılığıyla fotoğraf içerisindeki cisimlerin algılanmasını sağlayan bir yöntemdir. Algoritma aşağıda belirtilen 4 adımdan oluşmaktadır.

1) İlk adım Haar özelliklerini seçimidir. Haar özelliği, belirli bir konumdaki bitişik dikdörtgen bölgelerle çalışır, her bölgedeki piksel yoğunluklarını toplar ve bu toplamalar arasındaki farkı hesaplar. Şekil 4'te bu işlemin sonucu olarak elde edilen görüntü gösterilmiştir.



Şekil 4

2) Bütünleyici görüntüler bu işlemin çok daha hızlı gerçekleşmesini sağlar. Ancak hesaplanan bu özelliklerin birçoğu aslında işe yaramazdır. Örnek olarak Şekil 5'teki görüntü ele alınırsa, üst sıra iki iyi özellik gösterir. İlk özellik, göz bölgesinin genellikle burun ve yanak bölgelerinden daha koyu olduğudur. İkinci özellik ise gözlerin burun köprüsünden daha koyu olma özelliğine dayanır. Ancak aynı filtreleme yanaklara veya başka herhangi bir yere uygulandığında bir anlam ifade etmez.



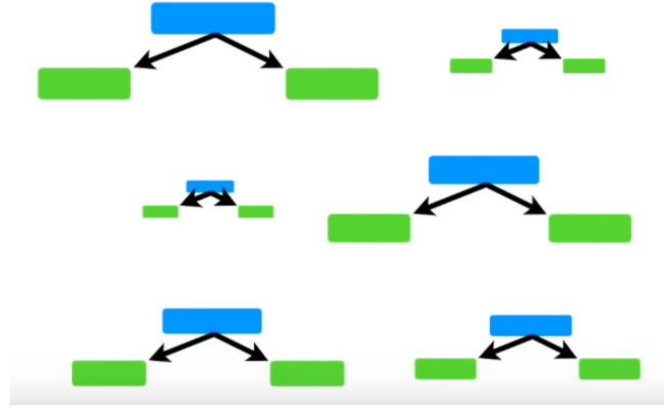
Şekil 5 - Bütünleyici görüntü

3) İşe yarayacak olan özelliklerin tespit edilmesi için 3. adım olarak AdaBoost algoritmasından yararlanılır. Bu algoritma, yüz tespiti için yetersiz olan sınıflandırıcıların doğrusal bir kombinasyon sonucu daha verimli bir sınıflandırıcı oluşturulmasını sağlar. AdaBoost algoritmasının temelini oluşturan 3 yapıtaşını açıklamak için karar ağaçları yapısı ele alındığında, AdaBoost geliştirilmesindeki en temel eleman “kütük” (stump) olarak adlandırılır.



Şekil 6 - Kütük yapısı

Bu kütükler bir köke bağlı iki alt düğüme (child) sahip olan elemanlardır. İki alt düğüm olması bu kütüklerin ilgili sorulara yalnızca evet veya hayır cevabı verebilmesine, dolayısıyla özellik belirlenmesinin yetersiz olmasına yol açmaktadır. Özellik belirlemesini güçlendirmek için bu kütükler bir arada gruplandırılarak “orman” (forest) adı verilen yapılar oluşturur. Bu şekilde karmaşık sorulara daha isabetli cevaplar verilebilir. Şekil 7’de “orman” yapısı örneği gösterilmiştir.



Şekil 7 - Orman yapısı

Ağaçlarla oluşturulan bir ormanda karar verilirken her ağaç diğer ağaçlarla eşit söz hakkına sahiptir. Dolayısıyla elde edilen veriler daha isabetli sonuçlar üretilmesini sağlayacağı halde bu eşit söz hakkı sebebiyle olumsuz sonuçlar elde edilebilir. Kütüklerden oluşan bir ağaçta ise bazı kütükler diğerlerine göre daha fazla söz hakkına sahiptir. Yeni bir kütük oluşturulurken önceki kütüklerde yapılan hatalar da hesaba katılır ve daha doğru bir özellik sınıflandırması elde edilmiş olur. Bu sayede tek başına bir anlam ifade etmeyen veri setlerinin bir araya getirilmesiyle istenilen sınıflandırma elde edilir.

- $(x_1, y_1), \dots, (x_n, y_n)$  eğitim kümesini,  $y_1 = \{0, 1\}$  ise sırasıyla negatif ve pozitif örnekleri ifade etmektedir.
- $y_1 = 0, 1$  için ilk ağırlık değerleri sırasıyla  $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$  olmak üzere  $m$  ve  $l$  negatif ve pozitif örnek sayılarını ifade etmektedir.
- For  $t = 1, \dots, T$ :

1. Ağırlıklar normalize edilir  $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$

1. Ağırlıklı hataya göre en iyi zayıf sınıflandırıcı seçilir:

$$\varepsilon_t = \min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - y_i|$$

3.  $h_t(x) = h(x, f_t, p_t, \theta_t)$  tanımlanır.  $f_t$ ,  $p_t$  ve  $\theta_t$ , minimum  $\varepsilon_t$ 'yi sağlayan değerleri ifade etmektedir.

4. Ağırlıklar güncellenir

$$w_{t+1,i} = w_{t,i} \beta^{1-\varepsilon_i}$$

Eğer  $x_i$  doğru sınıflandırılırsa  $\varepsilon_i = 0$ , aksi takdirde  $\varepsilon_i = 1$  olmakta ve

$$\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t}$$

- Final güçlü sınıflandırıcı:

$$C(x) = \begin{cases} 1, & \text{eğer } \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0, & \text{aksi durumda} \end{cases}$$

ve  $\alpha_t = \log \frac{1}{\beta_t}$ .

Şekil 8 - AdaBoost algoritması

Yüzün algılanması aşamasında, hedef boyutta bir pencere giriş görüntüsü üzerine taşınır ve görüntünün her bir alt bölümü ve Haar özellikleri hesaplanır. Bu fark daha sonra, nesneler ile nesne olmayanları birbirinden ayırmayı sağlayacak olan bir eşik değeri ile karşılaştırılır. Her Haar özelliği yukarıda bahsedildiği gibi yetersiz bir sınıflandırıcı olduğundan bir nesneyi yeterli doğrulukla tespit edebilmek için çok sayıda Haar özelliği gereklidir.

4) Son aşamada ise sınıflandırıcının her aşaması, bulunduğu bölgeyi pozitif veya negatif olarak etiketler. Pozitif etiket ilgili bölgede bir nesnenin bulunduğunu ve negatif ise hiçbir nesnenin bulunmadığını gösterir. Etiket negatifse, bu bölgenin sınıflandırması tamamlanır ve belirleyici pencere bir sonraki bölgeye kaydırılır. Etiket pozitifse, sınıflandırıcı bölgeyi bir sonraki aşamaya geçirir. Belirleyici, son aşama bölgeyi pozitif olarak sınıflandırdığında geçerli konumda bulunan nesneyi çıktı olarak rapor eder. Aşamalar, negatif örnekleri mümkün olduğunca hızlı bir şekilde eleyecek şekilde tasarlanmıştır. Taranacak bölge bir pencere olarak ele alındığında pencerenin olabildiğince büyük olması tercih edilir. Bu sayede nesne içermeyen bölgeler daha çabuk elimine edilir. Kısaca, yüzde bulunmak istenen bölgeler istenmeyenlere göre daha az olduğundan dolayı süreci gereksizleri elemek yerine gereklileri bulacak şekilde oluşturmak vakit kaybına sebep olacaktır. Bir pozitif örnek doğru olarak sınıflandırıldığında doğru bir pozitif değer oluşur. Bir negatif örnek hatalı sınıflandırıldığında bir hatalı pozitif değer oluşur. Bir pozitif örnek hatalı sınıflandırıldığında ise bir hatalı negatif değer oluşur. Algoritmanın doğru çalışması için, Cascade'in her aşamasının düşük sayıda hatalı negatif orana sahip olması gerekir. Eğer bir aşamada bir nesneyi yanlışlıkla negatif olarak etiketlerse, sınıflandırma durur ve hatayı düzeltme imkânı bulunamaz. Bununla birlikte, her aşama yüksek sayıda hatalı pozitif orana sahip olabilir. Belirleyici, nesne olmayan bir görüntüyü yanlışlıkla pozitif olarak etiketlese bile, sonraki aşamalarda hatayı düzeltme imkânı vardır. Daha fazla aşama eklemek hatalı pozitif değer oranını azaltır, ancak aynı zamanda doğru pozitif oranı azaltır. Projede yüz ve gözleri tespit etmek için kullanılan kod parçası Tablo 2’te verilmiştir.

```
Face_cascade = cv.CascadeClassifier('frontalface.xml')
Eye_cascade = cv.CascadeClassifier('eye.xml')
Faces = face_cascade.detectMultiScale(blur,3.3,4)
Gray = blur[y:y+h, x:x+w]
Eyes = eye_cascade.detectMultiScale(gray)
```

Tablo 2

Burada *detectMultiScale* fonksiyonuna parametre olarak gönderilen ilk değer olan *blur* değişkeni, üzerindeki yüz ve gözleri tespit etmek istediğimiz görüntüyü, ikinci parametre ölçeklendirme değerini ve üçüncü parametre iterasyon sayısını temsil etmektedir. Bu program aracılığıyla elde edilen çıktı görüntüsü şekil 9’da gösterilmiştir.



Şekil 9

## 2.2.4 Morfolojik İşlemler

### 2.2.4.1 Giriş

Bu projede lineer filtreleme ile birleştirilmiş morfolojik interpolasyona dayalı görüntü dönüşümü yöntemi kullanılacaktır. Bu yöntem aracılığıyla, önerilen algoritma kullanılarak art arda morfolojik medyan üretimi ile verilen girdi görüntüsünü farklı bir görüntüye dönüştüren interpolasyon dizi üretilir. Algoritma ardışık iki kare (frame) arasındaki benzerlik oranını kullanır. Algoritmanın iki versiyonu önerilmektedir. Birincisinde; gerekli parametre son dizinin karelerinin sayısı, ikincisinde ise ardışık iki karenin arasındaki kabul edilebilir maksimum hatadır.

Uygulamanın en popüler kullanım alanlarından biri görüntü interpolasyonu yani verilen görüntüyü ikinci görüntüye dönüştüren otomatik bir sekans oluşturma yöntemidir. Bu tür işlemler bilgisayar grafiklerinde “morphing” adı altında bulunur. Morfolojik medyan iki girdi görüntüsünün arasına yeni bir görüntü eklenmesiyle elde edilir. İteratif medyan görüntülerinin üretilmesi ilk görüntüyü sonuncuya dönüştüren interpolasyon dizisinin üretilmesini sağlar. Bu diziye morphing dizisi denir.

### 2.2.4.2 Renkli Morfolojik Medyan

Morfolojik medyan ikili, renkli ve gri görüntüler için üretilebilir ve aşınma (erosion), genişleme (dilation) işlemlerine dayanır. Aşınma her görüntü pikselinin komşuları arasına minimum değer atayan bir operatör olarak tanımlanır. Girdi görüntüsü olan  $F$ ’nin aşınması (1)’de tanımlanmıştır.

$$G = \varepsilon(F) \Leftrightarrow \forall p \in P: G(p) = \min_{q \in N(p)} \{F(p + q)\} - h \quad (1)$$

G çıktı (output) görüntüsünü, epsilon ( $\epsilon$ ) aşınma operatörünü, P görüntünün domainini, N(p) bir p pikselinin en yakın komşuluğunu ve h yapı (structuring) elamanlarının yüksekliğini temsil eder.

Genişleme işlemi komşu pikseller arasındaki maksimum değere dayanır ve (2) ile elde edilir. Burada sigma ( $\delta$ ) genişleme operatörünü temsil eder. Diğer semboller ise yukarıda tanımlandığı gibidir.

$$G = \delta(F) \Leftrightarrow \forall p \in P: G(p) = \max_{q \in N(p)} \{F(p+q)\} + h \quad (2)$$

Medyan görüntüsü (3)'te tanımlanmıştır.

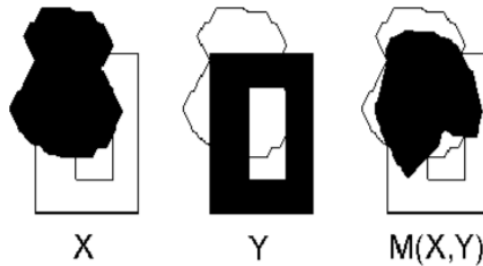
$$M(F, G) = \sup_{\forall \lambda} \{ \inf [ \delta^{(\lambda)}(\inf(F, G)), \epsilon^{(\lambda)}(\sup(F, G)) ] \} \quad (3)$$

Burada  $\lambda=1,2,\dots$  giderek artan tamsayı değerleri ifade eder. Sup ve inf değerleri ise şu şekilde tanımlanmıştır. Supremum: Üst sınırların en küçüğü, infimum: Alt sınırların en büyüğü olmaktadır ve (4) ve (5)'teki formüllerle elde edilir.

$$G = \sup(F_1, F_2) \Leftrightarrow \forall p \in P \quad G(p) = \max \{F_1(p), F_2(p)\} \quad (4)$$

$$G = \inf(F_1, F_2) \Leftrightarrow \forall p \in P \quad G(p) = \min \{F_1(p), F_2(p)\} \quad (5)$$

İkili görüntülerde yapılandırma elemanının yüksekliği  $h=0$  olmalıdır. Bununla birlikte ikili görüntülerde ek bir koşul olarak girdi görüntülerinin kesişme noktası boş küme olamaz



Şekil 10 - İkili görüntünün medyanı

Gri görüntülerin morfolojik medyanını hesaplamak için yapılandırma elemanının yüksekliği  $h>0$  olmalıdır. Öte yandan renkli görüntülerde yükseklik;  $h_1, h_2, h_3 > 0$  olmak üzere  $h = [h_1, h_2, h_3]$  vektörüyle temsil edilir. Bu  $h_1, h_2, h_3$  değerleri RGB renk uzayında sırasıyla kırmızı, yeşil ve mavi bileşenleri temsil ederler. Renkli görüntüler söz konusu olduğunda (1) ve (5)'teki min değerleri ve (2) ve (4)'teki max değerleri bu vektör alanından elde edilen değerler üzerinden hesaplanmalıdır. Her kartezyen renk uzayında, çalışmayı gerçekleştiren kişi pikselleri karşılaştırırken her zaman bu üç vektör elemanını karşılaştırması gerekir. Bu karşılaştırmanın en popüler yollarından birisi leksografik sıralamadır fakat RGB renk uzayı söz konusu olduğunda önemli bir dezavantajı vardır. Burada R bileşeni G bileşeninden daha önemli olarak kabul edilirken B bileşeni daha az önemli kabul edilir. Bu problemi çözmek için yeni bir renk uzayı oluşturulmaktadır. İlk RGB renk uzayı  $3 \times 3$ 'lük dönüşüm matrisi oluşturularak karşılaştırmalı uzaya dönüştürülür.

Dönüşüm renk bileşenlerinin insan algısı için görsel önemine dayanmaktadır. Karşılaştırmalı uzaya dönüşüm, renk bileşenlerini sıralar ve doğrusal kombinasyonları ile bunları birleştirir böylece ardışık karşılaştırmalar bileşenlerin görsel önemine göre sıralanır. İlk renk uzayının dönüşüm matrisi olan  $M$ , renk bileşenlerinin vektörlerini karşılaştırmalı olarak çarpar. Dönüşüm işlemi (6)'da tanımlanmıştır.

$$M = \begin{bmatrix} 0.3 & 0.6 & 0.1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}^T = M \cdot \begin{bmatrix} r & g & b \end{bmatrix}^T \quad (6)$$

Yukarıda sunulan yaklaşım renklerin insan görüşü için önemini göz önüne alarak RGB renk uzayının vektörlerini karşılaştırmayı sağlar. Sonuç olarak renkli görüntülerin morfolojik medyanı hesaplanmış olur.

#### 2.2.4.3 Algoritma

Denklem (3) medyan görüntü hesaplarının yinelemeli algoritmasını oluşturmak için kullanılmaktadır. İlk görüntü çiftinden başlayarak (7)'ye eşit olan 3 yardımcı görüntü olarak  $Z$ ,  $W$ ,  $M$ 'yi incelenir.

$$Z_0 = \inf(F, G); W_0 = \sup(F, G); M_0 = \inf(F, G) \quad (7)$$

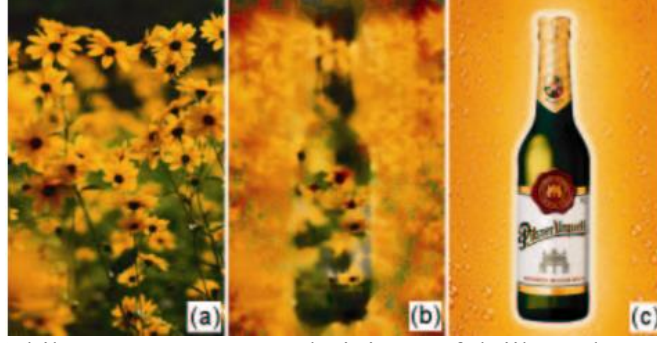
$$Z_i = \delta(Z_{i-1}); W_i = \varepsilon(W_{i-1}); M_i = \sup[\inf(Z_i, W_i), M_{i-1}] \quad (8)$$

İndeksler iterasyon sayısını belirtir.  $i$ . iterasyon sonucu elde edilen değerler (8)'deki kurallar kullanılarak hesaplanır.  $M$  aralığına kadar iterasyonlara devam edilir.

$$M(F, G) = M_i$$

Burada  $M(F, G)$ ,  $F$  ve  $G$  görüntülerinin morfolojik medyanıdır.  $i$  ise  $M_i = M_{i+1}$  olacak şekilde en düşük iterasyon sayısıdır. Yukarıdaki algoritma yakınsaktır. Bu, denklem (3) ve dijital görüntülerin ayırık doğası ile garantilenmiştir. Yakınsama için teorik olarak tek tehlike  $M_i = M_{i+2} \neq M_{i+1} = M_{i+3}$  gibi osilasyonlardır. Ancak  $M_{i+1} \geq M_i$  olması durumunda bu salınımlar oluşmaz. Bu algoritmayı kullanarak elde edilen renkli medyan görüntülerin örnekleri Şekil 11 ve Şekil 12'de sunulmaktadır.





Şekil 11 - a ve c görüntülerinin morfolojik medyanı b



Şekil 12

Şekil 11 çoğunlukla yeşil ve sarı renk tonu içeren benzer renk paletine sahip iki görüntünün morfolojik medyanını gösterir. Şekil 12 baskın renk tonlarına, farklı renk paletlerine sahip 2 görüntünün medyanını içerir. Medyan görüntünün her iki giriş görüntüsündeki nesnelerin şekil deformasyonu ile elde edilen nesneleri içerdiği açıkça görülmektedir.

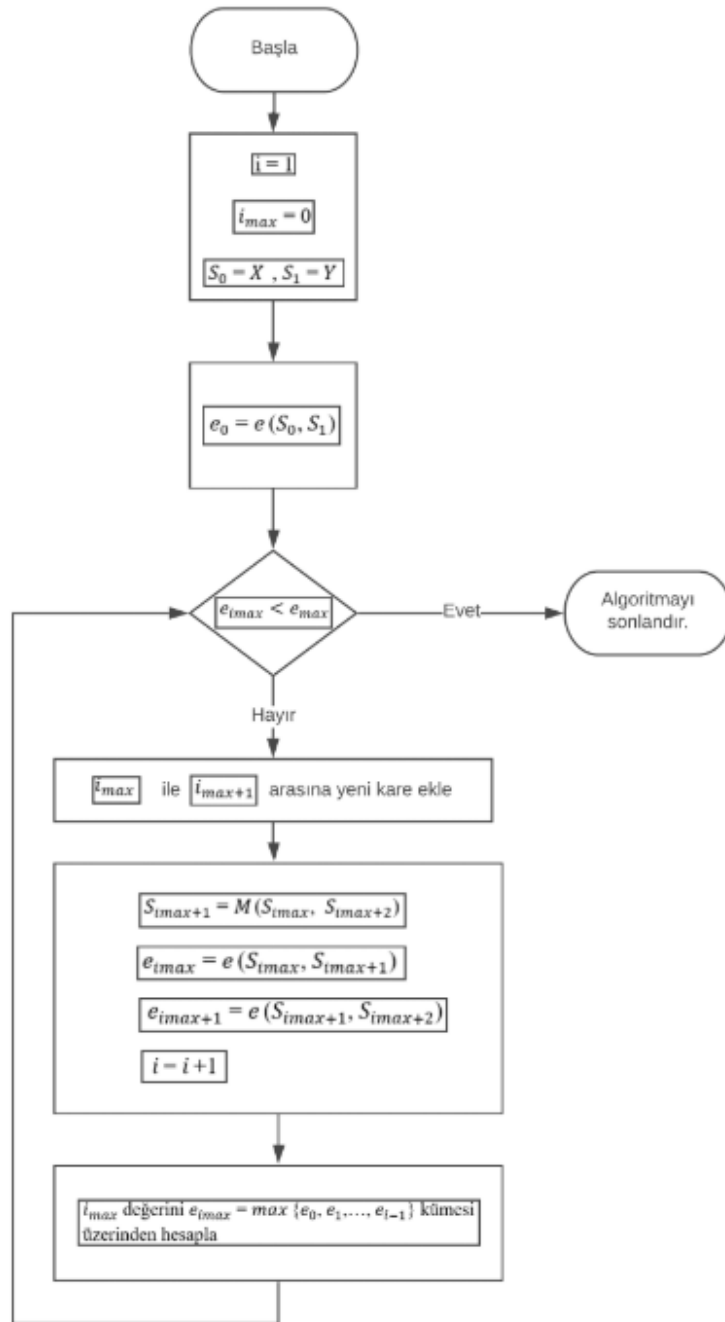
#### 2.2.4.4 Morphing Sekansı

Önceki bölümde tek interpolasyonu bir görünümün üretilmesi gözlemlenmiştir. Bu bölümde bütün bir dizinin nasıl üretileceği açıklanmaktadır. İnterpolasyon dizisinin üretimi geleneksel metoda göre daha önce üretilmiş çiftlerin arasına art arda yeni medyanların üretimine dayalıdır. Fakat böyle bir yaklaşımın bir dezavantajı vardır. Görüntünün içeriğini dikkate almaz. Ayrıca istenilen herhangi bir uzunluktaki bir dizinin üretimine izin vermez. Burada işlem başına bir kare üreten iteratif bir interpolasyon dizisinin elde edilmesi ele alınmıştır.

Algoritma, iki komşu görüntü arasındaki farka bağlı olarak yeni medyan oluşturulmasına gerek olup olmadığına karar verir. Bu yöntemde öncekinin aksine seviyelerin dağılımı yerine ardışık her bir kare çifti arasındaki ölçü dağılımı optimize edilir. Benzerlik ölçüsü “e” hatasına eşittir. Bu da P ve Q karelerine ait piksellerin parlaklık değerleri arasındaki ortalama kare hatası (MSE: Mean Square Error) olarak hesaplanır.

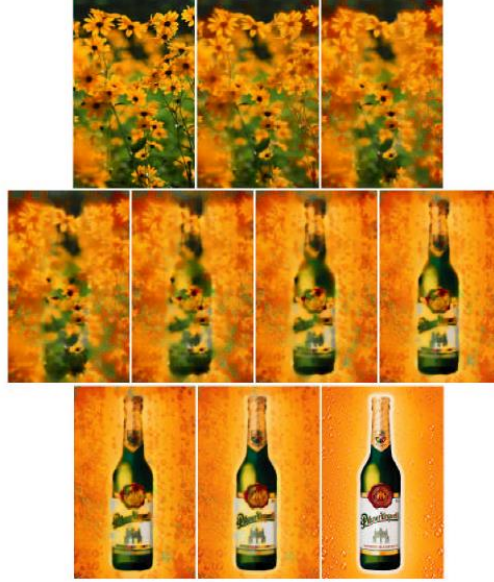
$$e(P, Q) = \frac{1}{x_{\max} \cdot y_{\max}} \cdot \sum_{i=0}^{x_{\max}-1} \sum_{j=0}^{y_{\max}-1} [lum(P(i, j)) - lum(Q(i, j))]^2 \quad (9)$$

Burada “lum” fonksiyonu argümanın parlaklığını,  $X_{\max}$  ve  $Y_{\max}$  görüntünün boyutlarını temsil eder. Algoritma iki vektör kullanır. Birincisi dizi karelerinin  $S = [S_0, S_1 \dots]$  şeklindeki bir vektörüdür. İkincisi ise ardışık her kare çifti arasındaki “e” benzerlik ölçülerinin bir vektörüdür. Her iki vektör de aynı sayıda elemana sahiptir. Giriş görüntüleri  $X$  ve  $Y$ ’dir. Sıra değeri olan geçici değişken “i” önceden üretilmiş kare sayısına eşittir.



Şekil 13 - Morphing dizisi üretim algoritması

Final dizisi olan  $S = [S_1, S_2, \dots, S_{i-1}]$  morfolojik interpolasyonlu dizidir. Öyle ki;  $S_0 = X$ ,  $S_{i-1} = Y$  ve  $S_1, S_2, \dots, S_{i-2}$  kareleri interpolasyonlu görüntüleri temsil eder. Son dizinin kare sayısı karmaşıklığa ve giriş görüntülerinin arasındaki farka bağlı olarak değişebilir. Hata,  $e_{max}$ 'tan küçük olduğu sürece iterasyonlara devam edilir.



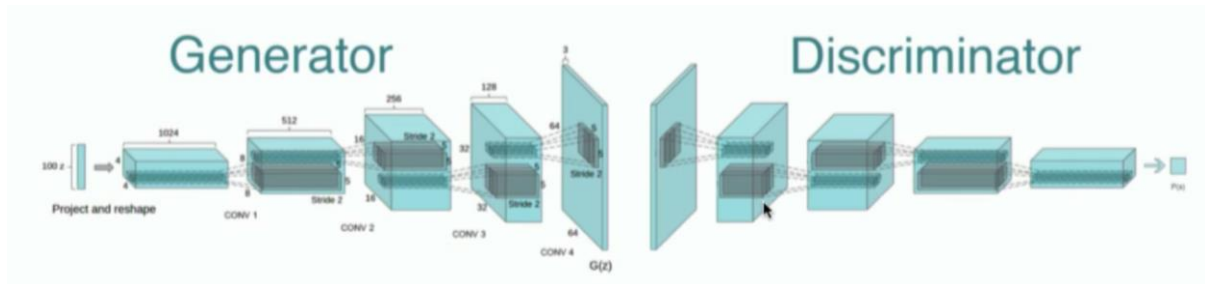
Şekil 14 - Morphing sekansı sonucu elde edilen interpolasyon dizisi

## 2.2.5 GAN (Generative Adversarial Network) Metodu

GAN metodu aldığı girdileri makine öğrenmesi üzerinden uygun şekilde birleştirerek ortaya daha önce üretilmemiş fakat aldığı girdilerle aynı istatistik değerlerine sahip yeni bir çıktı verisi üretmek için kullanılır. Moda tasarımı, oyunlar, fotoğrafçılık, grafik tasarım GAN'ın uygulama alanlarından bazılarıdır.

### 2.2.5.1 Yöntem

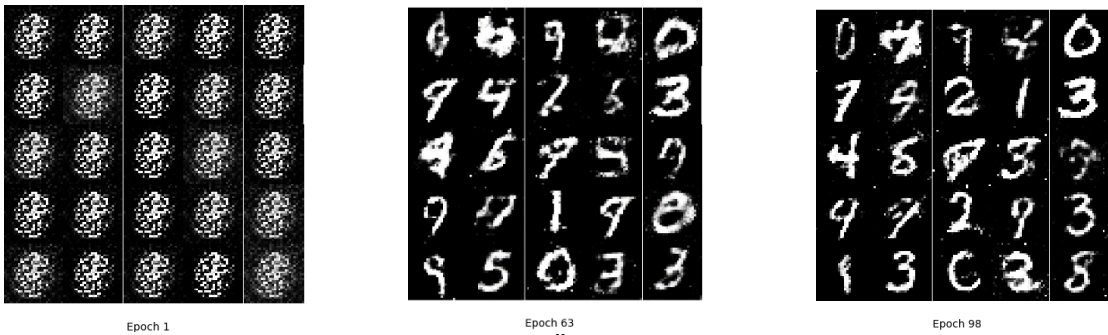
2014 yılında Google araştırmacısı Ian Goodfellow tarafından icat edilen bu network tipi birbirine zıt şekilde çalışan 2 yapay sinir ağından oluşmaktadır.



Şekil 15 - Üretici ve Ayırt Edici Yapısı

Şekil 14'te görüldüğü üzere görüldüğü üzere, soldaki yapay sinir ağı, üretici (Generator) ilk başta rasgele sayılardan gerçeğe benzer resimler üretmeye başlar. Sağdaki yapay sinir ağı ayırt edici (Discriminator) ise üretilen resimlerden veya gerçek resimleri girdi olarak alıp, alınan girdinin sahte mi gerçek mi olduğunu algılar ve ikili sistemde bu değeri geri döndürür.

Her bir işlem adımından sonra yine geri yayılım (back-propagation) ile geriye doğru *gradyanlar* hesaplanır. Böylece rasgele sayılardan üretilen resimlerin gerçek olanları ile arasındaki farkın hesaplanması ve üreticinin her bir işlem adımında kendisini güncellemesi söz konusudur. Üreticinin amacı ayırt edicinin daha fazla hata yapmasını sağlayacak fotoğraflar üretmek iken ayırt edicinin amacı gerçek fotoğrafları sahte olanlardan daha iyi ayırt etmektir. Bu yapay sinir ağı yeterince eğitildikten sonra üreticinin gittikçe gerçeğe benzer fotoğraflar ürettiği görülecektir.



Şekil 16 - 1, 63 ve 98. Aşamada Üretilen Yapay Görüntüler

### 2.2.5.2 Algoritma

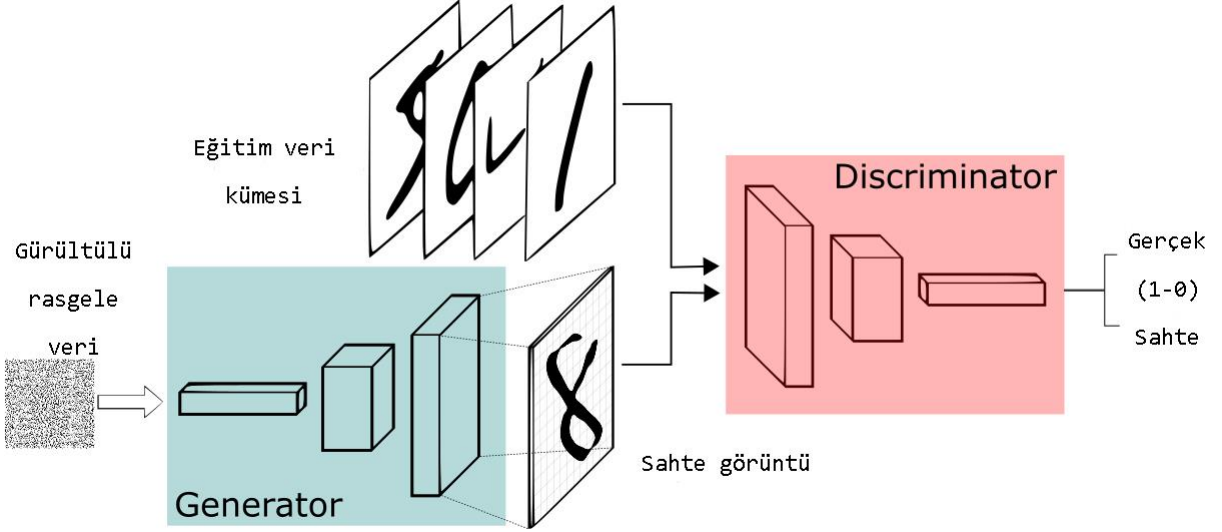
Üretici algoritmalar girdi verilerini sınıflandırmaya çalışır; yani, bir veri örneğinin özellikleri göz önüne alındığında, bu verilerin ait olduğu bir etiketi veya kategoriye öngörürler. Örneğin, bir e-postadaki tüm kelimeler göz önüne alındığında, ayırt edici bir algoritma iletinin spam olup olmadığını tahmin edebilir. Burada etiketler spam ve not\_spam'dır. E-postadan toplanan sözcükler giriş verilerini oluşturan özelliklerdir. Bu sorun matematiksel olarak ifade edildiğinde, etiket  $y$  olarak adlandırılır ve özellikler  $x$  olarak adlandırılır.  $P(y | x)$  formülasyonu, " $x$ 'in  $y$  olasılığı" anlamına gelir, bu durumda "içerdiği kelimeler göz önüne alındığında bir e-postanın spam olma olasılığı" anlamına gelir.

Ayırt edici algoritmalar ise özellikleri etiketlerle eşleştirir. Sadece bu korelasyonla ilgilenirler. Üretici algoritmalar ise bu işlemin tersini yapmaktadır. Belirli özellikler verilen bir etiketi tahmin etmek yerine, belirli bir etiket verilen özellikleri tahmin etmeye çalışırlar.

Üretici bir algoritmanın cevaplamaya çalıştığı soru şudur: Bu e-postanın spam olduğunu varsayarsak, bu özelliklerin olasılığı nedir? Ayırt edici modeller  $y$  ve  $x$  arasındaki ilişkiyi önemsemekle birlikte, üretici modeller "Nasıl  $x$  elde edersiniz?" konusuna önem verir.  $P(x | y)$  formülasyonu, " $y$ 'in  $x$  olasılığı" anlamına gelir. Bununla birlikte, üretici algoritmalar sınıflandırıcılar olarak da kullanılabilir

### 2.2.5.3 GAN Blokları

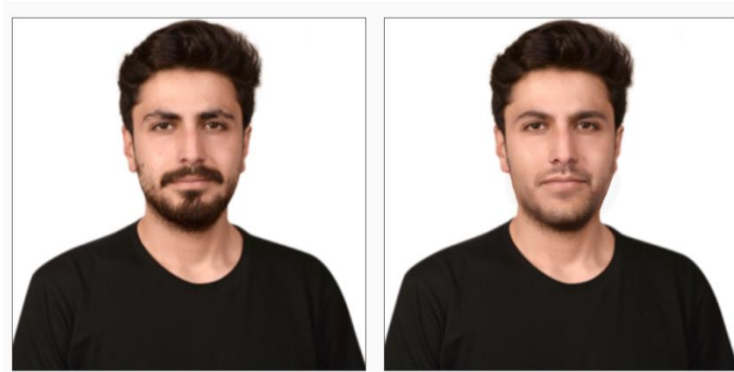
Üretici rastgele sayılar alır ve bir görüntü döndürür. Bu oluşturulan görüntü, gerçek-doğruluk veri kümesinden alınan bir görüntü akışı ile birlikte ayırt edici gönderilir. Üretici hem gerçek hem de sahte görüntüler alır ve 0 ile 1 arasında bir sayı olan olasılıkları döndürür; 1 özgünlük tahminini temsil eder 0 ise sahte anlamına gelir.



Şekil 17 - Üretici ve Ayırt edici Katman Yapısı

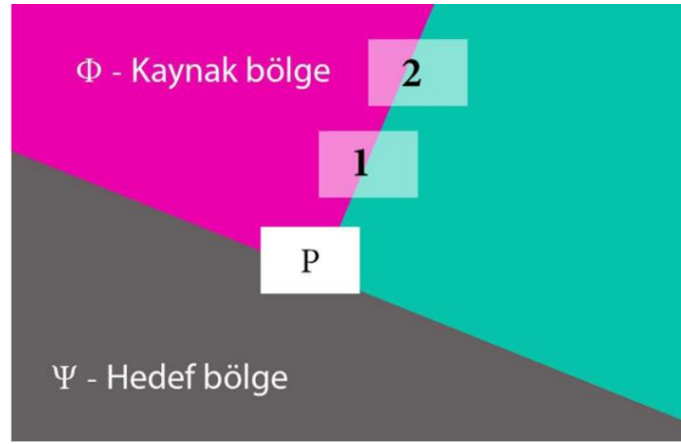
### 2.2.6 Inpainting

Inpainting (iç boyama); görüntünün istenmeyen bir bölümünü, bu bölüm dışında kalan görüntü bölgesinden elde edilen uygun piksellerle doldurulup istenilen görüntüyü oluşturma işlemidir. İç boyama yapıldığında orijinal görüntüde yapılan değişikliklerin görüntünün aslını bilmeyen kişinin algılayamayacağı kadar özgün görüntüyle uyumlu olması amaçlanır.



Şekil-18 - Çene bölgesine uygulanmış inpaint örneği.

Görüntü iç boyama için geliştirilen metotlar genel olarak, germeye dayalı, dalgacık tabanlı, örüntü tekrarına dayalı ve örnek tabanlı olan metotlardır. Bu yöntemlerin en önemli olanı örnek tabanlı olan metottur. Örnek tabanlı metotta temel olarak örnek tabanlı görüntü iç boyama yaklaşımı vardır. Bu yaklaşımın da temelinde izotop sürücü görüntü örnekleme işlemi bulunmaktadır. Bu metotta izotopları ele almak için ek bir sentez mekanizmasına ihtiyaç duyulur. Bu metotta çalışma kolaylığı olması açısından standart notasyonlar kullanılmıştır. Kullanılan bazı notasyonlar verilecek olursa; doldurulacak görüntü bölgesi yani hedef bölge “ $\Omega$ ”, kullanılacak yama blok boyutu “P”, Kaynak Bölge “ $\Phi$ ” ve hedef bölge sınırı da “ $\partial\Omega$ ” ile gösterilmiştir. Algoritma ilerledikçe “ $\partial\Omega$ ” içe doğru yayılmakta ve değişmektedir. Bu sebeple her iterasyonda onun değişmiş hali olan “ön doldurulacak sınır” referans olarak alınır. Kaynak Bölge ( $\Phi$ ) algoritma boyunca sabit kalır ve doldurma işleminde kullanılacak gerekli örnekleri sağlar.



Şekil 19 - Örnek tabanlı inpaint örneği

Doldurulacak sınır üzerindeki her piksele bir öncelik değeri ataması yapılarak o piksel merkezli yamanın doldurulmasıyla, lineer yapıyı bozmadan, doldurma yapılması sağlanır. Öncelikli doldurulacak yamayı belirleyen iki faktör vardır. Bu faktörlerden birisi güven terimi (biri yama eğer kaynak bölgeden daha fazla piksel içeriyorsa), diğeri ise veri terimidir.

P noktasındaki pikselin çevresindeki güvenilir bilginin bir ölçüsünü veren değere güven terimi denir ve sembolik olarak  $C(p)$  ile gösterilir.

Veri terimi olarak gösterilen  $D(p)$  ise P noktasındaki parlaklık gradyanı ile o noktada sınıra dik olan normal vektörünün çarpımıyla hesaplanan bir değer olup sınırın önüne doğru uzanan izotopun keskinliğinin bir ölçüsünü verir.

P merkezli yamaya verilen öncelik değeri ise bu iki değerın çarpımı sonucunda elde edilir.  $P(p)$  olarak gösterilir. Hesaplanan  $C(p)$  değeri,  $D(p)$  değeri ile çarpılması sonucu öncelik değeri oluşacağından normalize edilmiş bu değerın etkisini dengede tutmasını sağlar. (10) eşitliğinin açık ifadesi (11) eşitliği olarak da verilebilir.

$$P(p) = C(p) * D(p) \quad (10)$$



$$C(p) = \frac{\sum_{q \in \Psi_p \cap \Phi_n} C(q)}{A \Psi_p} \quad (11)$$

(11) eşitliğinde  $C(p)$ ,  $p$  noktası merkezli  $\Psi_p$  yamasının güven değerini göstermektedir. Eşitlikte görülen  $A_{\Psi_p}$ ,  $\Psi_p$  yamasının alanını,  $C(q)$  ise daha önceden  $\Psi_p$  yamasının atanmış güven değerini göstermektedir. Buradaki  $q$ ,  $p$  merkezli yamanın kaynak bölgesindeki piksel sayısını gösterir.  $\Psi_p$  yamasının başlangıç güven değerini sembolize eden  $C(q)$ ; algoritma başında aşağıdaki atamalar yapılarak oluşturulur;

$$C(q) = \begin{cases} 1, & \forall q \in \Phi \\ 0, & \forall q \in \Omega \end{cases} \quad (12)$$

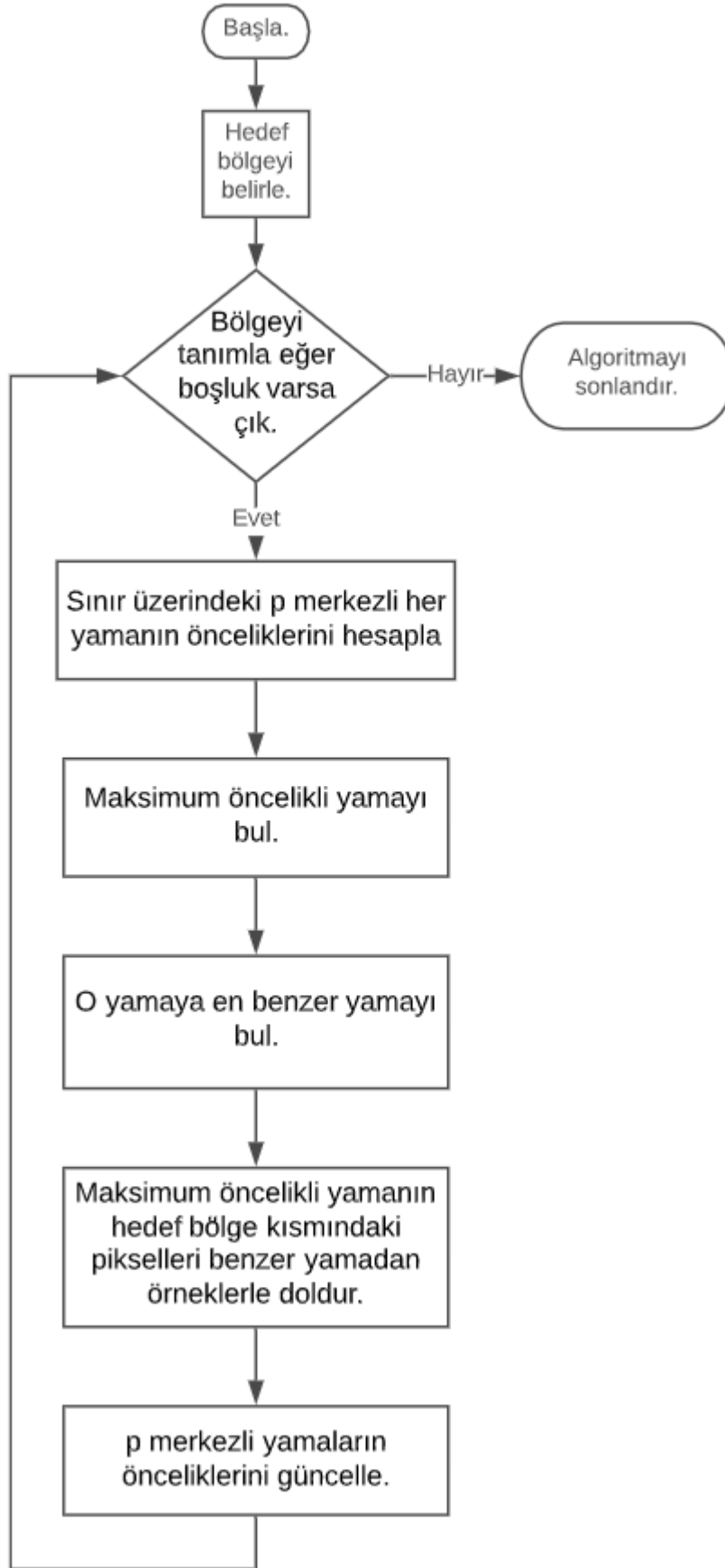
Eşitlik (10)'da verilen güven teriminin hesaplanmasından sonra, bu güven terimiyle çarpılacak olan veri terimi ( $D(p)$ ) değeri ise (13) formülüyle hesaplanır.

$$D(p) = |\nabla I_p^\perp \cdot n_p| / \alpha \quad (13)$$

En yüksek olasılıklı yama  $\Psi_p^\wedge$  sembolü ile gösterilmektedir. Bulunan  $\Psi_p^\wedge$  yaması daha sonra kaynak bölgesinden ( $\Phi$ ) elde edilen verilerle doldurulur. Doldurulacak blokla yerine koyulacak blok arasındaki farkın en az olması esasına dayanarak kopyalama işlemi yapılır.

$$\psi_{\hat{q}} = \arg \min_{\psi_q \in \Phi} d(\psi_{\hat{p}}, \psi_q)$$

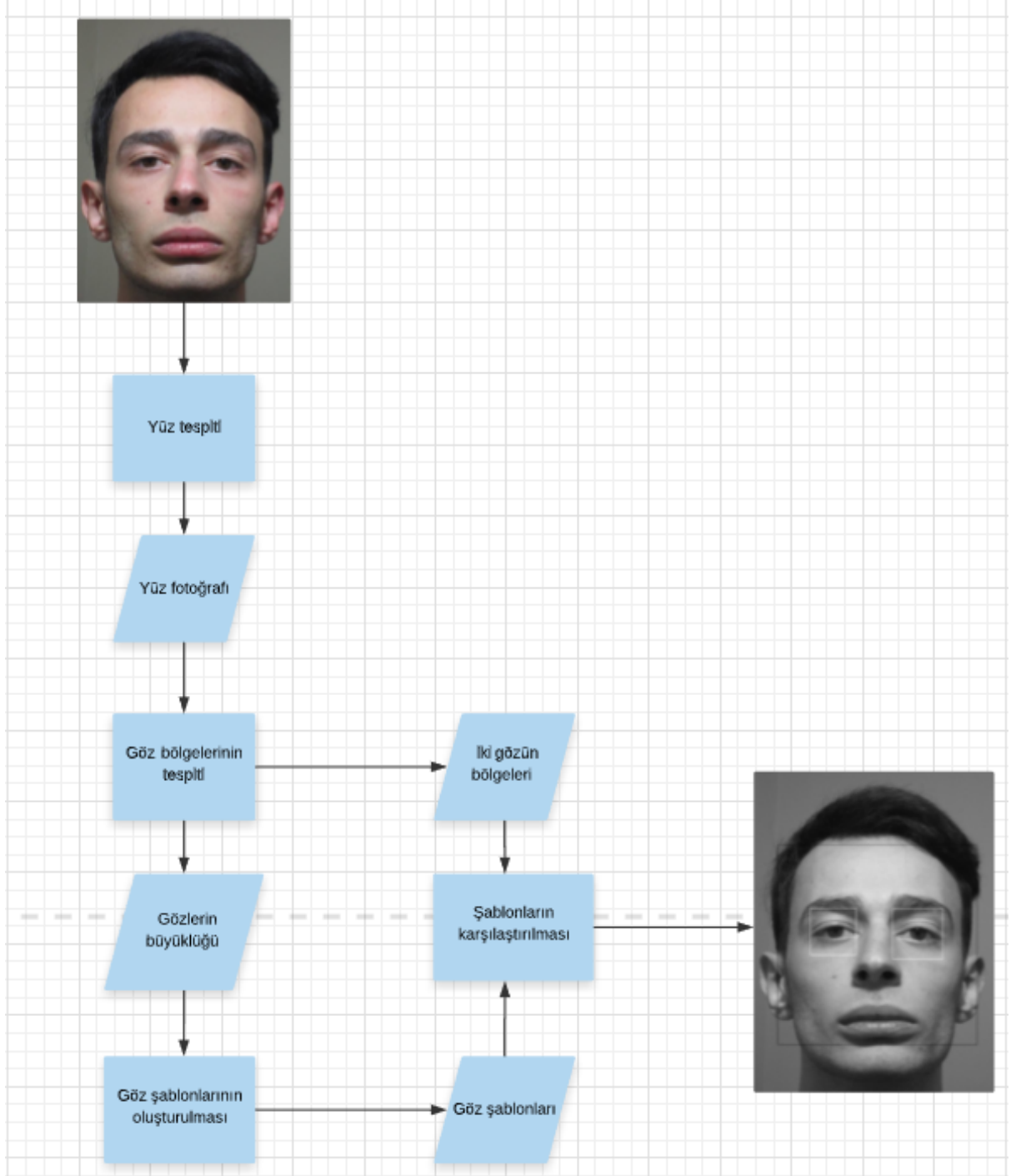
Bütün bu işlemler yapıldıktan sonra güven terimi,  $\Psi_p^\wedge$  tarafından sınırlandırılmış alanda güncellenir. Güncelleme kuralı ise  $\forall q \in \Psi_p^\wedge \cap \Omega$  için  $C(q) = C(p^\wedge)$  ataması yapılmasıdır. Bu güncelleme kuralı doldurulacak ön sınır üzerindeki yamaların bağlı güven değerlerinin özel görüntü parametreleri olmaksızın ölçülebilmesini sağlar. Doldurma işlemi yapılırken güven değeri bozulmaya başlar. Bu ise hedef bölge merkez yakınındaki piksellerin parlaklık değerinden daha az olunduğunun bir göstergesidir. Algoritması ise aşağıdaki Şekil (20)'da gösterilmiştir.



Şekil 20 - Örnek tabanlı doldurma algoritması



### 3. UML NESNE MODELİ



#### 4. KAYNAKLAR

- 1) <https://www.python.org/downloads> Python
- 2) <https://www.jetbrains.com/pycharm/download/#section=windows> PyCharm
- 3) <https://opencv.org/releases> OpenCV
- 4) [https://en.wikipedia.org/wiki/Thresholding\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing)) Geleneksel eşikleme.
- 5) [https://en.wikipedia.org/wiki/Otsu%27s\\_method](https://en.wikipedia.org/wiki/Otsu%27s_method) Otsu eşikleme.
- 6) [https://docs.opencv.org/3.4/db/d28/tutorial\\_cascade\\_classifier.html](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html) Haar Cascade
- 7) <https://ieeexplore.ieee.org/abstract/document/7960323> Haar Cascade - “Seröz Efüzyon Sitolojisinde Makine Öğrenmesine Dayalı Otomatik Çekirdek Algılama”, Murat Ekinci, Elif BAYKAL
- 8) [https://www.researchgate.net/publication/221546601\\_Image\\_Morphing\\_Based\\_on\\_Morphological\\_Interpolation\\_Combined\\_with\\_Linear\\_Filtering](https://www.researchgate.net/publication/221546601_Image_Morphing_Based_on_Morphological_Interpolation_Combined_with_Linear_Filtering) Morphing - “Image Morphing Based on Morphological Interpolation with Linear Filtering”, Marcin Iwanowski
- 9) <https://bair.berkeley.edu/blog/2018/03/13/mcgan> GAN Metodu
- 10) <https://medium.com/@ankankumarbhunia/recurrent-font-gan-4b5ba27ad138> GAN Metodu
- 11) <https://medium.com/@devnag/generative-adversarial-networks-gans-in-50-lines-of-code-pytorch-e81b79659e3f> GAN Metodu
- 12) [https://github.com/togheppi/vanilla\\_GAN](https://github.com/togheppi/vanilla_GAN) GAN Metodu
- 13) <https://www.youtube.com/watch?v=0VPQHbMvGzg> GAN Metodu
- 14) <https://jhui.github.io/2017/03/05/Generative-adversarial-models/> GAN Metodu
- 15) <https://dergipark.org.tr/tr/download/article-file/207196> İç Boyama - “İstenmeyen Cisimlerin Resimden Kaldırılması”, Vasif V. Nabyev, Ahmet Taşçı

**STANDARTLAR ve KISITLAR FORMU**

Projenin hazırlanmasında uyulan standart ve kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

1. Projenizin tasarım boyutu nedir? (Yeni bir proje midir? Var olan bir projenin tekrarı mıdır? Bir projenin parçası mıdır? Sizin tasarımınız proje toplamının yüzde olarak ne kadarını oluşturmaktadır?)

Proje konumuz üzerine daha önce farklı yöntemler ile çalışılmıştır. Benzer metotların kullanıldığı, daha verimli bir proje KTÜ bünyesinde bulunmamaktadır.

2. Projenizde bir mühendislik problemini kendiniz formüle edip, çözdünüz mü? Açıklayınız.

Hayır. Mevcut formüllerle sonuca ulaştık.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Algoritmalar ve Görüntü İşleme derslerinde edindiğimiz bilgiler sorunun anlaşılması ve çözüm geliştirilmesi noktalarında etkili olmuştur.

4. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir? (Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız).

5. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? Lütfen boşlukları uygun yanıtlarla doldurunuz.

a) Ekonomi

Herhangi bir kısıt yoktur.

b) Çevre sorunları:

Herhangi bir kısıt yoktur.

c) Sürdürülebilirlik:

Herhangi bir kısıt yoktur.

d) Üretilebilirlik:

Herhangi bir kısıt yoktur.

e) Etik:

Herhangi bir kısıt yoktur.

f) Sağlık:

Herhangi bir kısıt yoktur.

g) Güvenlik:

Herhangi bir kısıt yoktur.

h) Sosyal ve politik sorunlar:

Herhangi bir kısıt yoktur.