

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



İnsan Yüzünde Sakal İçeren Bölgenin Sakalsız Cilt ile Değiştirilmesi

BİTİRME PROJESİ

Mehmet Çağrı Yonca



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, eleştiriye kabul etmek ve eleştiriye 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

Bana sorgulamayı öğreten değerli hocam Vasif V. Nabiyev'e ve araştırmayı öğreten sevgili hocam Berkin Malkoç'a teşekkürlerimi borç bilirim.

Mehmet Çağrı Yonca

Trabzon 2018

İÇİNDEKİLER

	Sayfa No
IEEE ETİK KURALLARI	II
ÖNSÖZ	III
İÇİNDEKİLER	IV
ÖZET	V
1. GENEL BİLGİLER	1
1.1. Giriş	1
1.2. Kullanılan Programlar	1
1.3. Kullanılan Kütüphaneler	1
1.4. Kullanılan Verisetleri	1
1.5. Kullanılan Mimariler	1
2. Yapay Sinir Ağları	2
2.1. Konvolüsyonel Ağlar	2
2.2. Çekişmeli Üretken Ağlar	3
2.2.1. Olasılık Dağılımı	3
2.2.2. Olasılık Yoğunluk Fonksiyonu	3
2.2.3. Normal Dağılım	3
3. Görüntü İçi Boyama	4
3.1. Bilgisayarlı Görüntü İçi Boyama	4
3.1.1. Yapısal İç Boyama	4
3.1.2. Dokusal İç Boyama	4
3.1.3. Yapısal ve Dokusal İç Boyama	4
4. Veri Kümesine Benzer Veri Üretimi	5
4.1. Veri Setinin İstatistiksel Olarak Yorumlanması	5
4.2. Derin Öğrenme ile Benzer Resim Üretimi	7
4.2.1. GAN Blokları	7
4.2.2. DCGAN Mimarisini	7
4.2.2.1. Bağlamsal Kayıp	8
4.2.2.2. Algısal Kayıp	8
4.2.2.3. Toplam Kayıp	8
5. Proje Şeması	9
5.1. Sakalın Tespiti	9
5.2. Tespit Edilen Sakal Yerine Cilt İmplantı	10
6. Proje Aşamaları	11
6.1. İnsan Yüzünün Tespiti	11
6.2. Sakalın Tespiti	12
6.3. Çenenin Görüntüden Çıkartılması	14
6.4. Görüntünün Eksik Parçasının GAN ile Doldurulması	15
KAYNAKÇA	16
STANDARTLAR VE KISITLAR FORMU	17

ÖZET

Projede bir insan yüzü üzerinde sakal olup olmadığının tespiti ve sakal bulunması halinde görüntüden sakalın kaldırılıp yerine sakalsız görüntünün getirilmesi amaçlanmıştır. Bu sayede kimliğini gizlemek için kılık değiştiren insanların tespitinin kolaylaştırılacaktır.

Bu tez içerisinde öncelikle gereksinimler açıklanmıştır. Ardından sinir ağları ve üretken ağlar hakkında bilgi verilmiştir. Son olarak projenin şablonu sunulmuştur.

1. GENEL BİLGİLER

1.1. Giriş

Projede sakallı bir erkek görüntüsü üzerinde sakalın tespit edilip resimden çıkartılarak yerine olması muhtemel cilt kaplamalarının yapılması amaçlanmıştır. Bu sayede kılık değiştirerek gizlenmeye çalışan insan yüzleri ortaya çıkarılmaktadır.

Proje Ubuntu işletim sistemi üzerinde Python dilinde geliştirilmiştir. Dlib, OpenCV, PyTorch kütüphaneleri kullanılmıştır. Görüntü oluşturmada DCGAN mimarisi kullanılmıştır.

Görüntüde sakalın tespiti ve sakalsız görüntü oluşturmak için 2 veri seti kullanılmıştır.

1.2. Kullanılan Programlar

Projeyi bulut ortamında kolayca kontrol edip düzenleyebilmek için Vim Text Editor kullanılmıştır. Bu sayede terminal üzerinde kolayca dosya içerisindeki metinler üzerinde değişiklik yapılabilir. Bulut bağlantısı SSH ile sağlanmıştır. Buluttaki sunucu için Google Cloud Engine üzerinde sunucu kiralanmıştır. Veri setini etiketlemek için labelImg adlı açık kaynak projesi kullanılmıştır.

1.3. Kullanılan Kütüphaneler

İnsan yüzünün tespit edilebilmesi için C ve C++ dilleriyle yazılmış BSD lisanslı görüntü işleme kütüphanesi olan OpenCV kütüphanesi kullanılmıştır. Burada tespit edilen insan yüzü üzerinde çene kısmının ayırt edilebilmesi için Dlib kütüphanesi kullanılmıştır. Aynı şekilde Dlib kütüphanesi de kullanılması tamamen serbest olan açık kaynak kodlu bir projedir, içerisinde bulundurduğu face_landmarks fonksiyonu ile yüz üzerinde göz, çene, ağız, burun gibi noktaların tespit edilmesini sağlar. Görüntülerin sınır ağlarında kullanılabilmesi için Numpy kütüphanesi kullanılmıştır. Modellerin oluşturulabilmesi için PyTorch kütüphanesi kullanılmıştır. Numpy sayesinde matrisel olarak işlenmeye hazır hale gelen görüntü PyTorch kütüphanesiyle oluşturulan modele sokulabilir.

1.4. Kullanılan Verisetleri

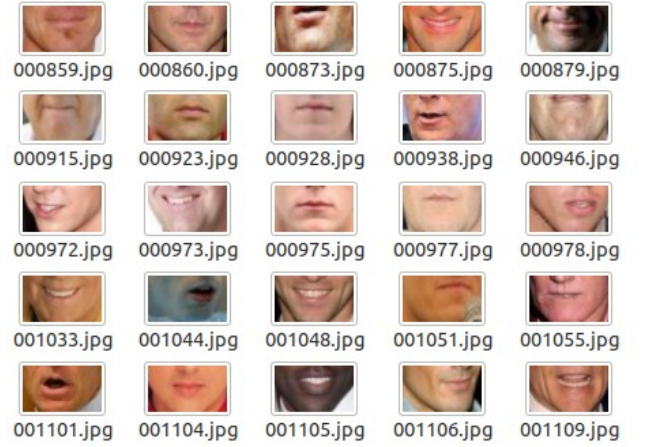
İnsan yüzünde sakalın ve bıyığın tespit edilmesi için modelin bu özellikleri bir veri setinden öğrenmesi gerekmektedir. Bu veri setinden maksimum verim alınabilmesi için ise yüz görüntülerinin gözler, burun, ağız ve çene olarak aynı hizada ve aynı boyutta olması gerekir.

Bu görevi yerine getirebilmek için toplamda 202.599 görüntü içeren CelebA veri setinin erkek yüzü içeren görüntüleri kullanılmaktadır. Erkek yüzünde sakalın olup olmadığının tespit edilebilmesi için sakallı ve sakalsız olmak üzere iki tip veri kullanılmaktadır. 2500 sakallı, 2500 sakalsız erkek yüzü kullanılmıştır.



Sakallı ve bıyıklı erkek veri seti

veri seti



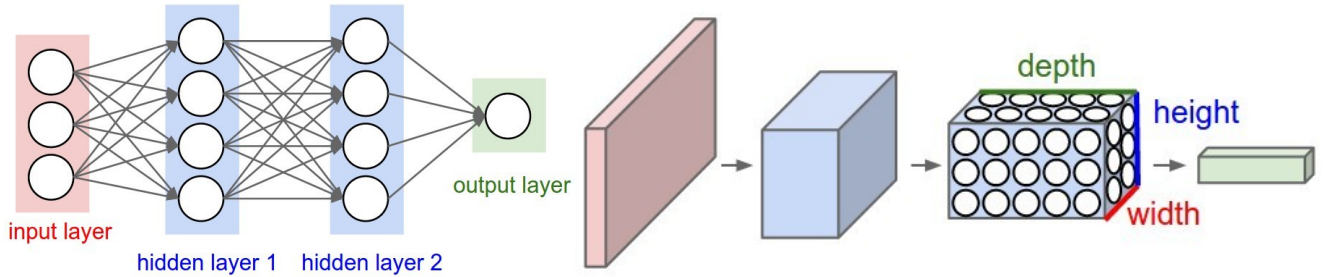
Sakalsız ve bıyıksız erkek

Alınan çene görüntüsünün sakallı mı sakalsız mı olduğunu tespit edebilmek için gereken sınıflandırıcının eğitiminde 2000 sakallı, 2000 sakalsız eğitim verisi kullanılmıştır. Ayrıca sınıflandırıcının ezberle öğrenmemesi için 500 sakallı, 500 sakalsız test verisi kullanılmıştır.

2. Yapay Sinir Ağları

Canlıların davranışlarını inceleyip, matematiksel olarak modelleyip, benzer yapay modellerin üretilmesine sibernetik denir. Eğitilebilir, adaptif ve kendi kendine organize olup öğrenebilen ve değerlendirme yapabilen yapay sinir ağları ile insan beyninin öğrenme yapısı modellenmeye çalışılmaktadır. Aynı insanda olduğu gibi yapay sinir ağları vasıtasıyla makinelerin eğitilmesi, öğrenmesi ve karar vermesi amaçlanmaktadır.

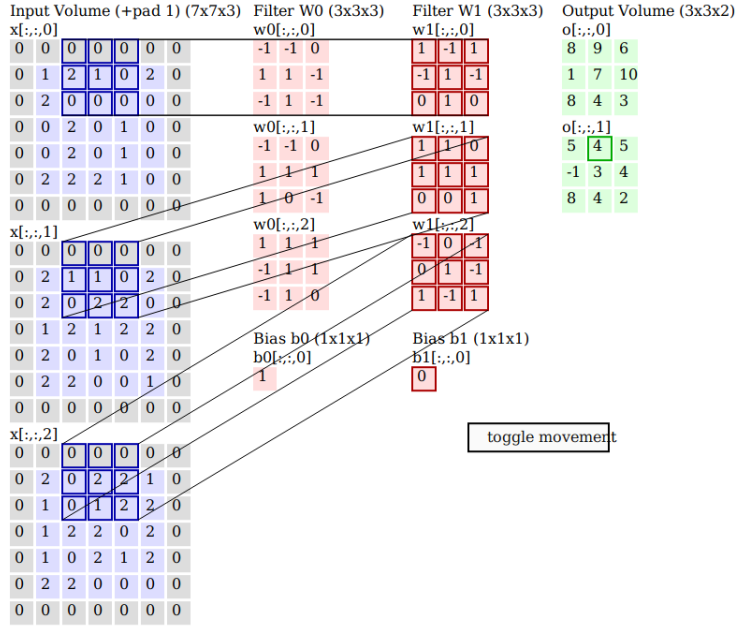
İnsan beynindeki bir nöron yeterli seviyede uyarıcıya ulaştığında diğer nörona iletim sağlamaktadır. Yapay sinir ağları da aynı bu şekilde çalışır. Çok sayıda nöronun bağlı olduğu diğer nöronlarla aralarındaki bağlantılar ağırlıklandırılır ve bu ağırlıklar, uç nörondaki çıkış verisinin istenilen veri olasılığı kadar değiştirilir. Bu işlem istenilen hata payından az değişim oluncaya kadar tekrarlanır ve bu sayede model eğitilmiş olur.



2.1. Konvolüsyonel Sinir Ağları

Temel yapay sinir ağlarında nöronlar birbirine giriş katmanı, gizli katman ve çıkış katmanı olmak üzere 3 katmanda bağlıdır. Giriş katmanına gelen görüntü gizli katmana sokulur ve gizli katmanın son katmanından gelen veri çıkış katmanında anlamlandırılır.

Konvolüsyonel sinir ağlarında ise görüntü genişlik, yükseklik ve derinlik olmak üzere 3 katmanda gelir. Görüntünün genişliği ve yüksekliğinin yanında RGB değerlerinin derinlik olarak alınmasıyla piksellerin birbiriyle ilişkisi irdelenir. Gelen görüntü üzerinde bir filtre gezdirilir ve bu filtre piksellerin durumuna göre görüntüyü encode eder. Encode etme işlemi filtre değerlerinin giriş verisindeki değerlerle çarpılıp toplanmasıyla elde edilmektedir. Bu işlem görüntünün özütünü çıkarmak olarak yorumlanabilir. Konvolüsyonel ağlarda istenilen sayıda filtre kullanılabilir.



Giriş verisinin üzerinde gezdirilen Filter W0 ve Filter W1 çerçeveleri yanda gösterilmektedir.

7x7x3 olan bir giriş verisi 3x3x2 olarak encode ediliyor.

Encode edilmiş haldeki çıkış verisi yapay sinir ağına sokularak anlamlandırılabilir.

2.2. Çekişmeli Üretken Ağlar

Çekişmeli Üretken Ağ (Generative Adversarial Networks) mimarisi 2014 yılında Ian Goodfellow tarafından tasarlanmıştır. Burada GAN olarak bahsedilecektir. GAN'lar bir karar verme mekanizması ve bir üretim mekanizması içerir.

Karar verme mekanizması (Discriminator) girişine gelen verinin sahte olup olmadığını kontrol eder.

Üretim mekanizması (Generator) ise giriş olarak bir gürültü vektörü alır ve bu vektörü upsample ederek görüntüye çevirir.

Üretken ağların eğitimi minimax algoritması ile sağlanır.

$$\min_G \max_D E_{x \sim p_{data}} \log D(x) + E_{z \sim p_z} \log(1 - D(G(z)))$$

2.2.1. Olasılık Dağılımı

Olasılık dağılımı rassal bir olayın ortaya çıkabilmesi için değerleri ve olasılıkları tanımlar. Değerler olay için mümkün olan tüm sonuçları kapsamalıdır ve olasılıkların toplamı bire eşit olmalıdır. Sayılabilir şekilde ayrı ayrı sonuçlar ve bunlara bağlı olan pozitif olasılıklar ayrık olasılık dağılımı olarak temsil edilmektedir.

2.2.2. Olasılık Yoğunluk Fonksiyonu

Olasılık kuramı ve istatistik bilim dallarında bir rassal değişken X için olasılık yoğunluk fonksiyonu bir reel sayılı sürekli fonksiyon olup f ile ifade edilir ve şu özellikleri olması gereklidir:

- R üzerinde pozitif veya sıfır değerleri alır
- R üzerinde integral değeri bulunabilir
- $\int_R (f(x)) dx = 1$ koşuluna uyar, yani eğri altındaki tüm alan bire eşittir. X 'in a ve b değerleri arasındaki olasılık, yani $P(a < X \leq b)$ şu ifade kullanılarak hesaplanır:

$$P(a < X \leq b) = \int_a^b (f(x)) \cdot dx$$

2.2.3. Normal Dağılım

Gauss dağılımı olarak da isimlendirilen normal dağılım birçok alanda pratik uygulaması olan, çok önemli bir sürekli olasılık dağılımı ailesidir. Bu dağılım ailesinin her bir üyesi konum gösteren ortalama (μ , aritmetik ortalama) ve ölçek gösteren varyans (σ^2 , yayılım) olmak üzere iki parametreyle tam olarak tanımlanabilir.

Doğa ve davranış bilimleri içinde bulunan birçok fenomenin niceliksel modeli yapılmasında normal dağılımın kullanılmasına neden merkezi limit teoreminin uygulanmasından doğmaktadır. Birçok psikolojik ölçüm ve fiziksel fenomen normal dağılım kullanılarak çok iyi yaklaşık olarak açıklanmaktadır. Bu fenomenlerin altında yatan mekanizmalar çoğu zaman bilinmemekte fakat normal dağılım modelinin açıklamada uygulanmaktadır. Bunun pratik yaklaşımın teorik olarak savunması ise her bir reel gözlemin oluşması için geri planda çok sayıda birbirinden bağımsız etkilerin ayrı ayrı toplam olarak katkıda bulundukları varsayımıdır.

3. Görüntü İçi Boyama

Görüntü içi boyama (Image Inpainting) bir görüntü içerisindeki kaybolan veya istenmeyen alanların görüntünün aslına benzer şekilde yeniden üretilmesidir. İnsanlar ve bilgisayarlar tarafından olmak üzere iki farklı şekilde gerçekleştirilebilir. Burada bilgisayarlı iç boyama anlatılacaktır.

3.1. Bilgisayarlı Görüntü İçi Boyama

Dijital kamera ve eski fotoğrafların sayısal ortama taşınmaya başladığından beri, otomatik iç boyama dijital görüntülerde gerçekleştirilen popüler bir uygulama haline gelmiştir. Yapısal iç boyama, dokusal iç boyama ve bu ikisinin kombinasyonu olmak üzere 3 farklı yaklaşıma sahiptir.

3.1.1. Yapısal İç Boyama

Yapısal iç boyama, boyanacak kısımların nasıl doldurulacağına dair geometrik yaklaşımları kullanır. Bu yaklaşım ana görüntünün geometrik yapısını bozmadan görüntünün doldurulmasına odaklanır.

3.1.2. Dokusal İç Boyama

Her yaklaşım gibi yapısal iç boyama yönteminin de avantaj ve dezavantajları bulunmaktadır. Yapısal iç boyama yöntemlerinin ana problemi dokuyu uygun şekilde üretememeleridir. Doku tekrar eden bir örüntüye sahiptir, bu da eksik olan kısmın seviye çizgileri içerisinde yenilenememesine sebep olur.

3.1.3. Yapısal ve Dokusal İç Boyama

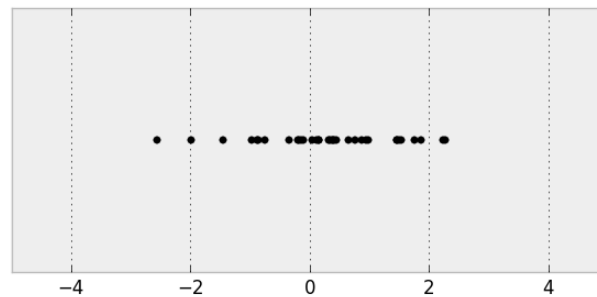
Yapısal ve dokusal iç boyama yaklaşımları eksik bölgeyi aynı anda dokusal ve yapısal boyamaya çalışır. Bir görüntünün çoğu parçası doku ve yapı içerir. Farklı dokular iç içe girdiğinde görüntü bölgeleri arasındaki sınırlar karmaşık bir bilgi olan yapısal bilgiyi taşır. Son teknoloji iç boyama yöntemleri yapısal ve dokusal bilgileri aynı anda değerlendirerek boyamaya çalışır.

Model tabanlı boyama ise görüntü içerisindeki kayıp bilgiyi en etkili şekilde doldurmak için Bayesyen iç boyama yaklaşımını kullanır. Bu proje içerisinde Bayesyen iç boyama yaklaşımı kullanılmıştır.

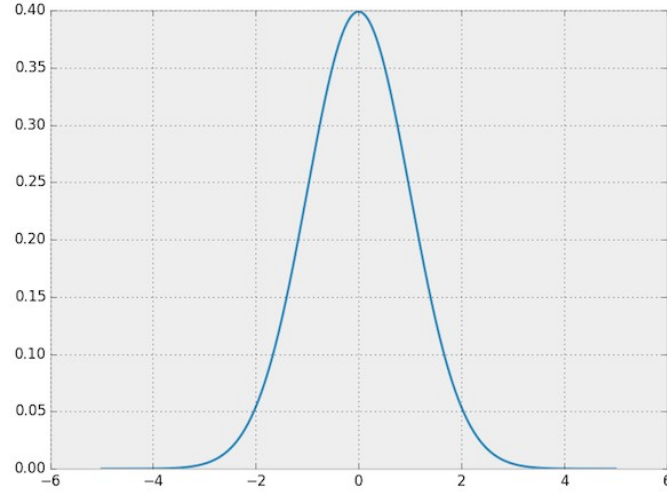
4. Bir Veri Kümesine Benzer Veri Üretimi

4.1. Veri Setinin İstatistiksel Olarak Yorumlanması

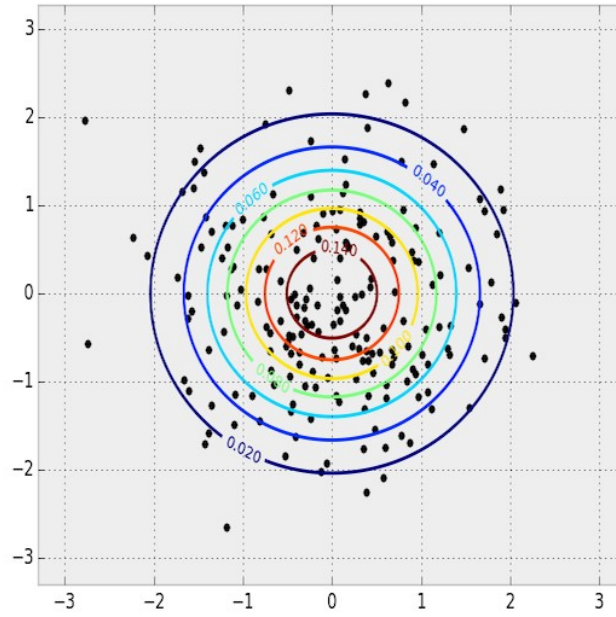
Aşağıda $[-4, 4]$ aralığında örnek bir normal dağılım görmektesiniz, bu grafik tek boyuttadır:



Burada ise yukarıdaki normal dağılımın olasılık yoğunluk fonksiyonunu görmektesiniz, bu grafik ise 2 boyutta bilgi içerir:



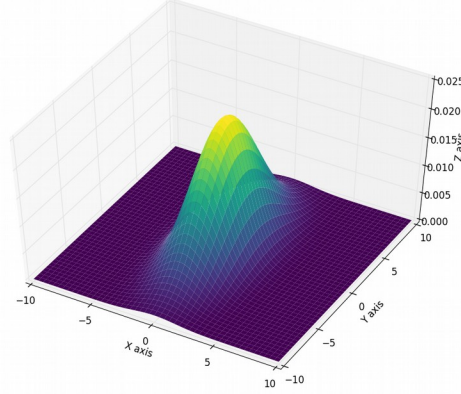
Görüldüğü gibi 0 etrafında yoğunlaşan veriler yoğunluk fonksiyonunda daha sık olarak gösteriliyor. Bu bize normal dağılımda temsil edilen verilerin $[-4, 4]$ aralığında olabilecekleri koordinatların olma ihtimalini vermektedir.



Yukarıda ise bu verilerin 3 boyutta olasılık yoğunluk fonksiyonunu görmektesiniz. İlk grafikte tek boyutlu bir veri temsil edilirken bu grafikte 2 boyutlu bir verinin (x, y) olasılık dağılımını incelenmektedir.

Görüntüler olasılık dağılımları olarak bu şekilde ifade edilebilmektedir. Bir fotoğraf makinesi ile fotoğraf çekildiğinde ışık sensörün üzerine düştüğü an bir olasılık dağılımı

örneklenmiş olur. Kırmızı, yeşil ve mavi renk değerlerine sahip 3 kanallı bir görüntünün olasılık dağılım grafiği aşağıdaki gibi bir grafikte temsil edilebilir:



64 genişlik ve 64 yüksekliğe sahip 3 kanallı bir görüntü 12288 ilişkiye (boyuta) sahiptir.

Olasılık dağılımının yoğunluk fonksiyonu bize resmin ne kadar sık geldiğini söyler, buna göre veri setindeki tüm görüntüler bir grafik üzerinde temsil edilirse veri setinde olmayan bir verinin veri setindeki verilere istatistiksel olarak ne kadar benzediği söylenebilir. GAN'ların çalışma mantığı da bu şekildedir. Veri setindeki verileri dağılım grafiğine sokarken aynı zamanda veri setindeki verilere benzer görüntü üretilmeye çalışılmaktadır.

Bu yöntemin gerçek hayattaki görüntülerde hesaplanması karmaşık olduğundan ve çok iş gerektirdiğinden derin öğrenme yöntemleri ile gerçekleştirilmesi amaçlanmaktadır.

4.2. Derin Öğrenme ile Benzer Resim Üretimi

Fikir aynı anda iki ağın beraber eğitilmesidir. İlk ağ, karar verme mekanizması giriş olarak bir görüntü alır ve çıkış olarak sahte olup olmadığına karar verir. Karar verme mekanizması $D(Y)$ olarak ele alınacaktır. Y veri setindeki verilere benzediğinde 0'a yakınsar, benzemediğinde ise 1'e yakınsar.

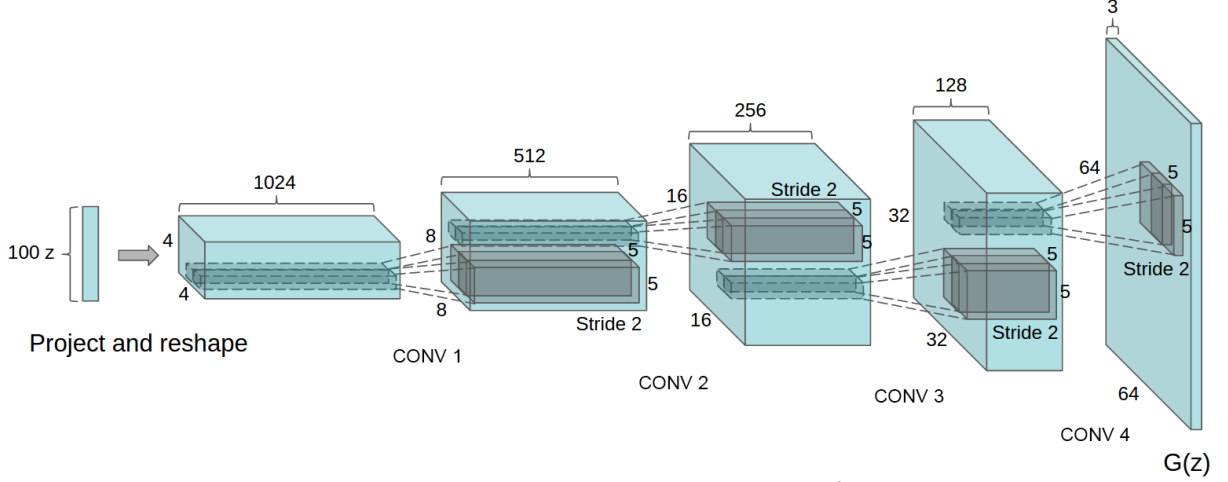
İkinci ağ olan üretim mekanizması ise giriş olarak bir gürültü vektörü alır ve çıkış olarak bir görüntü üretir. Üretim mekanizması $G(Z)$ olarak ele alınacaktır. Z verisi rastgele değerler ile başlatılmıştır. $G(Z)$ ile üretilen veri $D(G(Z))$ olarak discriminator'e girer. Discriminator'den gelen sonuca göre Z vektörü daha benzer bir görüntü üretecek şekilde değiştirilir. Discriminator'ün görevi $D(Y)$ 'yi maximize etmeye çalışmak iken Generator'ün görevi $D(G(Z))$ minimize etmeye çalışmaktır.

4.2.1. GAN Blokları

Tanımlanacak yapı basitçe, bilinen bir dağılım oluşturup onu p_z olarak göstermektir. Sonraki kullanımlardan p_z 'yi $[-1, 1]$ arasında bir sürekli dağılım olarak düşünelim. Örneğimiz bu dağılımın içerisinde bir numara olarak düşünülebilir.

Örneğin p_z 5 boyutlu ise $[0.77, 0.93, -0.18, -0.69, 0.34]$ olabilir. Şimdi z adında bir gürültü verimiz var. Bu veriden görüntü üretecek bir generator fonksiyona ihtiyacımız var. Bu projede görüntü üretmek için 2016 yılında Alec Radford ve arkadaşları tarafından tasarlanan DCGAN mimarisi kullanılmıştır.

4.2.2. DCGAN Mimarisi



Burada 100 boyutlu bir gürültü vektörü oluşturulmuştur. İlk olarak 100 boyutlu tek düze dağılım olan Z , $4 \times 4 \times 1024$ 'lük bir veri dizisine upsample edilmektedir. Ardından bu işlem 3 kez daha tekrarlanır ve sonunda 4 kez upsample edilmiş olan gürültü vektörü $G(Z)$ $64 \times 64 \times 3$ boyutunda bir veri olarak elde edilmektedir.

Generator ile üretilen görüntüler, esas veri setindeki görüntülerle beraber discriminator'e sokulur. Discriminator girişindeki verilerin generator ile üretilen verilerin veri setindeki gerçek verilerden olup olmadığına karar vermeye çalışır.

Generator modeli ürettiği verilerin discriminator'den gerçek görüntü olarak geçmesini istediği için eğittikçe ürettiği veriler gerçek veri setindeki verilere yakınsar. Bu şekilde eğitim sonunda gerçek veri setimizdeki verilere benzeyen sahte bir veri setimiz oluşmuş olur.

4.2.2.1. Bağlamsal Kayıp

DCGAN mimarisinde giriş görüntüsündeki temayı kaybetmemek için üretilen görüntü giriş görüntüsünde benzemediğinde cezalandırılmalıdır. İşlemsel olarak bu işlem y giriş görüntüsünün piksellerinin $G(z)$ 'den çıkartılmasıyla yapılabilir.

$$L_{\text{baglamsal}}(z) = \|M \odot G(z) - M \odot y_{\text{giris}}\|$$

İdeal şartlarda y ve $G(z)$ görüntülerinde bilinen pikseller birbirine eşittir.

4.2.2.2. Algısal Kayıp

Görüntünün gerçeksi olması isteniyorsa discriminatorun geçmesine izin verdiği görüntülerin uygunluğundan emin olunmalıdır. Bu durum DCGAN'da kullanılan notasyonla sağlanmaktadır.

$$L_{\text{algisal}}(z) = \log(1 - D(G(Z)))$$

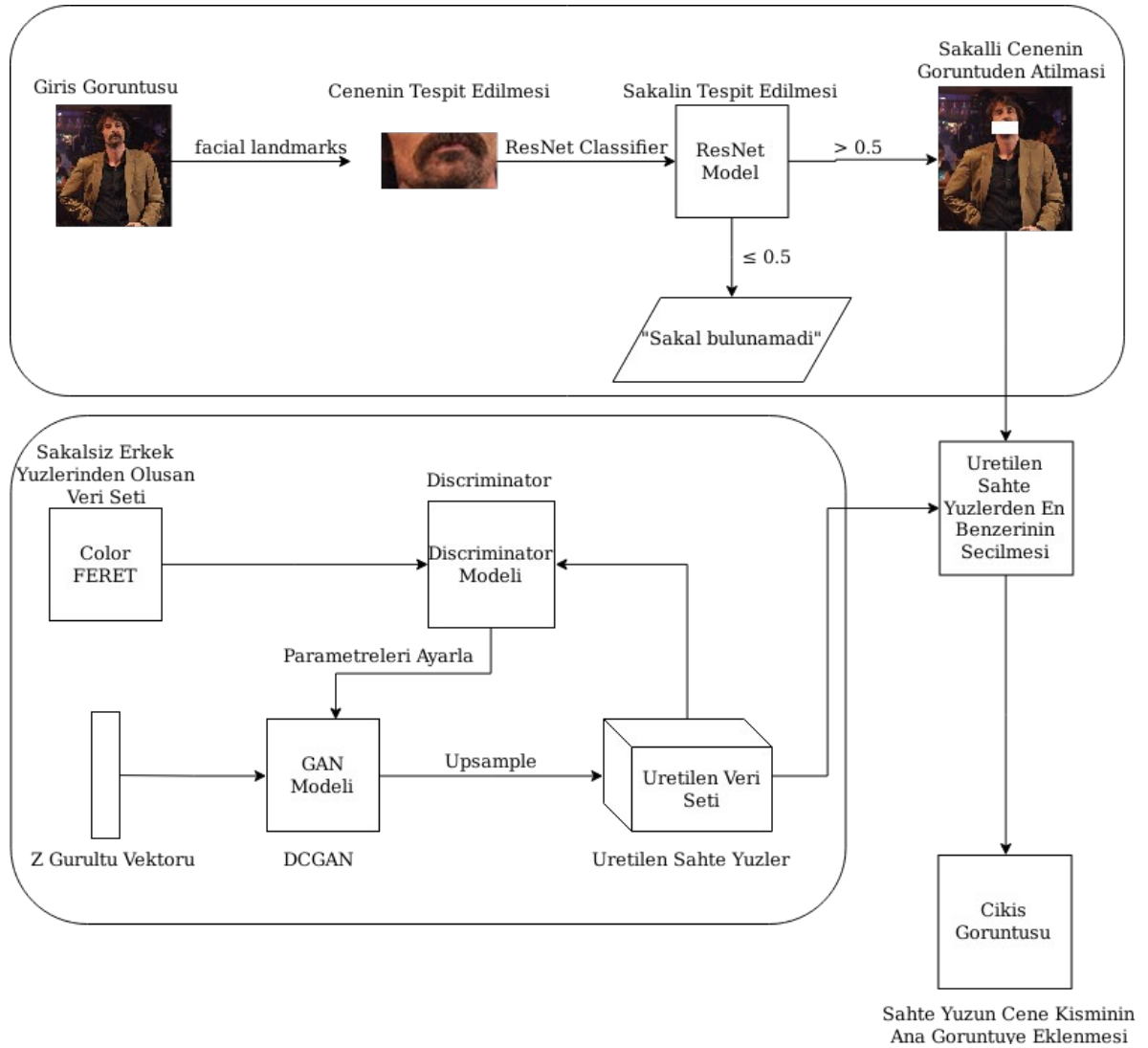
4.2.2.3. Toplam Kayıp

Doldurulmak istenilen görüntünün benzerliğinin sağlanması için bağlamsal ve algısal kayıplar beraber kullanılabilir.

$$L(z) = L_{\text{baglamsal}}(z) + \lambda L_{\text{algisal}}(z)$$

λ parametresi bağlamsal kaybın algısal kayba olan bağlılığını ayarlar.

5. Proje Şeması



5.1. Sakalın Tespiti

Projede sakal tespiti için öncelikle insan yüzünün çene kısmı tespit edilecektir. Bunun için dlib kütüphanesinin yüz ve yüzün bölümlerinin tespiti için aşağıdaki gibi detector ve predictor kullanılacaktır.

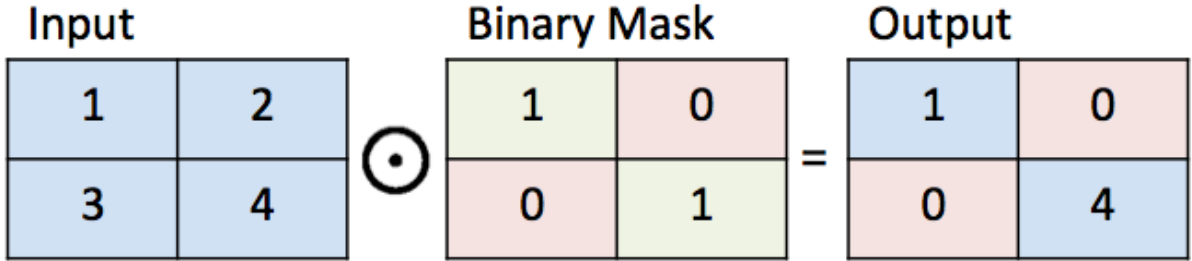
```
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(args["shape_predictor"])
```

Bu şekilde yüzün çene kısmının tespit edilmesinin ardından çenede sakal yoğunluğunun belirlenmesi için görüntü sınıflandırıcıya sokulacaktır. Sınıflandırıcı çıktı olarak görüntünün sakal içerme ihtimalini verir ve bu ihtimalin %50'den büyük olduğu durumda yüzde sakal var olarak kabul edilir.

5.2. Tespit Edilen Sakal Yerine Cilt İmplantı

Sakalı olduğu tespit edilen yüz görüntüsünün çene bölümü görüntüden atılır, sakalsız insan yüzleri ile eğitilen generatif modelin ürettiği verilerden en benzer olanının çene bölgesi alınır ve ana görüntüye eklenir.

Sakallı bölgenin ana görüntüden çıkartılması için binary maskeleme işlemi yapılmalıdır:



Giriş olarak verilen görüntüde çıkartılmak istenilen bölge 0, kalması istenilen bölgeye 1 ile noktasal çarpım uygulanırsa atılmak istenilen bölgeyi 0 olarak olarak setlenmiş olmaktadır.

Çıkartılan bölgeye eklenmek istenilen sahte görüntünün de çene bölgesi benzeri şekilde ana görüntüye giydirilebilir. (ž eksik parça olsun)

$$X_{\text{üretilen}} = X_{\text{input}} \odot M_{\text{binary}} + (1 - M) \odot G(\check{z})$$

6. Proje Aşamaları

Öncelikle proje şemasında belirtildiği gibi projeye insan yüzünün tespiti ve sonrasında sakal olup olmadığının tespit edilmesi ile başlanmaktadır.

6.1. İnsan Yüzünün Tespiti

Görüntü üzerinde insan yüzünün tespit edilebilmesi için dlib kütüphanesi kullanılmaktadır.

```
// Dlib kutuphanesinin get_frontal_face_detector fonksiyonu bir yuz
dedektörü tanımlar, ardından yüzdeki bölge bilgisini sağlayabilmek için ön eğitilmiş
bir model predictor olarak kullanılır, predictor grayscale resimden göz, burun, ağız
ve çene bilgisini üretir
```

```
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(args["shape_predictor"])
shape = predictor(gray, rect)
shape = face_utils.shape_to_np(shape)
```

```
// Yüzün yönelimini bulup hizalayabilmek için iki göz arasındaki eğim ele
alınır ve OpenCV kütüphanesinin getRotationMatrix2D fonksiyonu ile rotasyon
matrisi oluşturulur
```

```
center_eyes = shape[27].astype(np.int)
eyes_d = np.linalg.norm(shape[36] - shape[45])
face_size_x = int(eyes_d * 2)

d = (shape[45] - shape[36]) / eyes_d
a = np.rad2deg(np.arctan2(d[1], d[0]))
scale_factor = float(256) / float(face_size_x * 1.3)
```

```

M = np.append(cv2.getRotationMatrix2D((center_eyes[0], center_eyes[1] +
15), a, scale_factor),[[0,0,1]], axis=0)

M1 = np.array([[1, 0, -center_eyes[0] + 128.],
               [0, 1, -center_eyes[1] + 128.], [0, 0, 1]])

M = M1.dot(M)[:2]

// Ardından bu matris görüntüye uygulanır ve tespit edilip hizalanmış yüz
görüntüsü dosyaya yazılır

face = cv2.warpAffine(image, M, (256, 256))

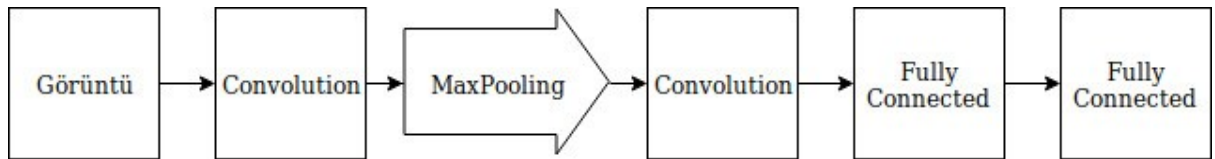
```

Bu sayede görüntü üzerinde yüz tespit edilip hizalanmış olur. Ardından yüzün çene kısmı seçilip çenenin sakallı olup olmadığı tespit edilir. Bu işlem için öncelikle sakal olup olmadığını anlayabilecek bir sınıflandırıcı eğitilmelidir.

6.2. Sakalın Tespiti

Sakallı ve sakalsız çene olarak 2 ayrı veri seti bulunmaktadır. 2500 sakallı çene görüntüsünün ve 2500 sakalsız çene görüntüsünün 500'er tanesi eğitimin başarılı olup olmadığını test edebilmek için ayrılmıştır.

PyTorch ile model eğitimi için öncelikle bir model tanımlanmalıdır. Modelin amacının 2 sınıfı birbirinden ayırabilmek olması nedeniyle derin bir mimariye ihtiyaç duyulmamıştır.



Model eğitildikten sonra veri集中的 verileri parça parça modele verip eğitime başlayabilmek için data loader gerekmektedir, bu iş için PyTorch'un DataLoader fonksiyonu kullanılmıştır. Ayrıca eğitimin hızlı bir şekilde sağlanabilmesi için model ve veri seti cuda fonksiyonu ile GPU'ya gönderilmiştir.

Loss fonksiyonunun hesaplanabilmesi için PyTorch içerisinde gelen Cross Entropy fonksiyonu kullanılmıştır. Loss fonksiyonunu optimize edebilmek için Adam optimizer'ı kullanılmıştır.

```
// Eğitim verisini ve etiketi alınır ve net adlı modele verilir,
```

```
train_x, train_y = data
```

```
y_hat = net.forward(train_x.cuda())
```

```
train_y = torch.Tensor(np.array(train_y)).cuda()
```

```
// Loss fonksiyonu hesaplanır
```

```
loss = criterion(y_hat, train_y.long())
```

```
epoch_loss += loss.item()
```

```
optimizer.zero_grad()
```

Eğitilen modeli kullanabilmek için görüntüde seçilen insan yüzünün çene kısmının alınması gerekmektedir.

```
// Daha önce oluşturulan detector ve predictor kullanarak yüzün alt
```

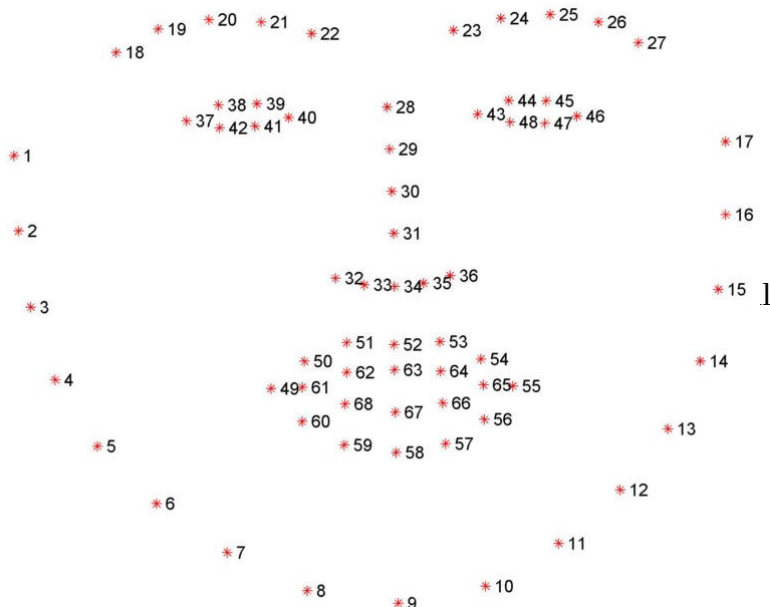
```
bölümleri tespit edilir ve çene görüntüden alınır
```

```
yuz = detector(gray, 1)
```

```
shape = predictor(gray, yuz[0])
```

```
shape = face_utils.shape_to_np(shape)
```

```
cene = image[shape[2][1] : shape[8][1], shape[2][0] : shape[14][0]].copy()
```



edilen noktaların

Ardından elde edilen çene koordinatlarını kapsayan bölge giriş görüntüsünden çıkartılır. Bu sayede GAN modeline verilen eksik görüntü istenilen şekilde sakallı cildi sakalsız cilt ile değiştirecektir.

6.3. Çenenin Görüntüden Çıkartılması

Bu aşamada çeneyi kapsayan bölgede ağız bölgesi de olduğundan çene bölgesinden ağız bölgesi çıkartılmalıdır.

```
// Tum görüntüden çene görüntüsünü çıkartabilmek için görüntüyü AND
kapısı ile AND'leyebileceğimiz bir maske oluşturmamız gerekmektedir

mask_cene = np.zeros(image.shape[:2], np.uint8)

cv2.drawContours(cene, [cene_koordinatlari], -1, (255, 255, 255), -1,
cv2.LINE_AA)

cenesiz_goruntu = cv2.bitwise_and(cene, cene, mask=mask_cene)
```



Yandaki görselde çene çıkartılmış görüntü gösterilmektedir.

Sakal bölgesi silinen görüntüden sakalsız insan yüzü oluşturabilmek için GAN modeli gerekmektedir. GAN modeli olarak önceden CelebA veriseti ile eğitilmiş bir model kullanılmıştır.

6.4. Görüntünün Eksik Parçasının GAN ile Doldurulması

Generatif ağlar bir veriye benzer veri üretmek için kullanılır. Generatif ağların eğitimindeki loss fonksiyonu ise üretilen verinin değişimini yönlendirir. Bu projede loss fonksiyonu görüntünün eksik bölgesinin çevresindeki piksellerle arasındaki ilişkiye bakarak hesaplanmaktadır.

Göz bebeklerinin konumu göre hizalanmış veri setiyle eğitilen bir generatif model kendi içerisinde yüz üzerindeki piksellerin featurelerini öğrenmiş olur. Bu sayede yüz görüntüsü üzerinde silinen bölge modelin veri setinden öğrendiği verilere göre en benzer alt görüntü adaylarının eklenmesiyle doldurulmuş olur.



GAN ile sol üstteki orijinal görüntüden oluşturulan benzer görüntüler

KAYNAKÇA

<https://www.draw.io/>
<https://github.com/goodfeli/adversarial>
<http://papers.nips.cc/paper/5423-generative-adversarial>
<https://www.quora.com/What-are-some-recent-and-potentially-upcoming-breakthroughs-in-deep-learning/answer/Yann-LeCun?srid=nZuy>
https://en.wikipedia.org/wiki/Statistical_model
https://en.wikipedia.org/wiki/Probability_distribution
https://en.wikipedia.org/wiki/Probability_density_function
https://en.wikipedia.org/wiki/Normal_distribution
<https://www.tensorflow.org/>
<https://arxiv.org/abs/1511.06434>
<https://arxiv.org/abs/1607.07539>
<http://bamos.github.io>
<http://cs231n.github.io/convolutional-networks/>

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 yeni bir projedir. Herhangi bir projenin tekrar değildir ve tamamen kendi tasarımımıdır.

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

Projede herhangi bir problemi formüle etmedim. Olan araçları kendi tasarımımında kullanabilmek için ufak değişiklikler uyguladım.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Yapay zeka dersinde görmüş olduğum yapay sinir ağlarını ve olasılık dersinde öğrendiğimiz olasılık dağılımı ve yoğunluk fonksiyonlarını kullandım.

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

Minimax algoritmasına bağlı kalındı.

$$\min_G \max_D E_{x \sim p_{data}} \log D(x) + E_{z \sim p_z} \log(1 - D(G(z)))$$

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

a) Ekonomi

Kısıt yoktur.

b) Çevre sorunları:

Kısıt yoktur.

c) Sürdürülebilirlik:

Kısıt yoktur.

d) Üretilirlik:

Kısıt yoktur.

e) Etik:

Kısıt yoktur.

f) Sağlık:

Kısıt yoktur.

g) Güvenlik:

Kısıt yoktur.

h) Sosyal ve politik sorunlar:

Kısıt yoktur.