YAPAY ZEKA VE DERİN ÖĞRENME TEKNİKLERİ KULLANILARAK YÜZ FOTOĞRAFLARI İÇEREN SAHTE FOTOĞRAF VE VİDEO SENTEZİ

Mustafa Salih BAHAR

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Ercan BULUŞ

2021

T.C.

TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

YAPAY ZEKA VE DERİN ÖĞRENME TEKNİKLERİ KULLANILARAK YÜZ FOTOĞRAFLARI İÇEREN SAHTE FOTOĞRAF VE VİDEO SENTEZİ

Mustafa Salih BAHAR

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN: Doç. Dr. Ercan BULUŞ

TEKİRDAĞ-2021

ÖZET

Yüksek Lisans Tezi

YAPAY ZEKA VE DERİN ÖĞRENME TEKNİKLERİ KULLANILARAK YÜZ FOTOĞRAFLARI İÇEREN SAHTE FOTOĞRAF VE VİDEO SENTEZİ

Mustafa Salih BAHAR

Tekirdağ Namık Kemal Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Ercan BULUŞ

Günümüzde sosyal medya ve Web aracılığıyla fotoğraf paylaşımı oldukça arttı. Neredeyse hemen her çekildiğimiz fotoğrafı, özel olsun olmasın, Web' e bir şekilde yüklüyoruz. Ama bu fotoğrafların kötü niyetli insanlar tarafından da görülebileceği ve kullanılabileceği konusunda pek bilinçli değiliz. Bu çalışmanın amacı da bu noktada ortaya çıkıyor. Çalışmada bu amaçla; bir kişinin gerçekte var olan kısa bir videosundan veya birkaç fotoğrafından bile, o kişinin yüz fotoğrafını içeren sahte videolar oluşturulabileceği kanıtlanmaktadır. Bu birkaç fotoğraf veya kısa bir video alınıp derin öğrenme teknikleriyle eğitilerek sahte fotoğraflar ve videolar oluşturulabilir. Sahte videolarda kişinin yüzüyle başka bir kişinin yüz değişimi (Face swapping) uygulanabilir veya kişinin yüzüne yeniden canlandırma (hareketlendirme) (Face reenactment) yapılabilir. Yeniden canlandırmada ise kaynak bir kişinin yüzüne başka bir kişinin videosundaki yüz hareketleri uygulanabilir. Hatta StyleGAN gibi teknikler ile gerçek insan yüz fotoğraflarından oluşan bir fotoğraf kümesi kullanılarak var olmayan insan fotoğrafları bile üretilebilir. Yaygın olarak Derin sahtelik (Deepfake) teknolojisi olarak bilinen bu teknikler, bu çalışmada yüzde kullanılan çeşitleri ve yapıları ile birlikte ele alınmıştır. Bu teknikler, eski dönemlerde yaşamış bilim adamlarının, ünlülerin var olan fotoğraflarına yeniden canlandırma yapılıp konuşturularak çocuklar için eğitim amaçlı kullanılabilir. Kuklacılıkta bu yöntem kullanılabilir. Oyuncuların (Aktörlerin-Aktrislerin) yerine sahnelerde, onların fotoğraflarıyla bilgisayarda yeniden canlandırma yapılabilir. Portreler canlandırılabilir (hareketlendirilebilir). Bilgisayar oyunlarındaki karakterlerde de bu yöntem kullanılabilir. Son zamanlarda özellikle bu teknolojilerle oluşturulan Kemal Sunal başta olmak üzere birçok oyuncu ve sanatçının videoları, reklam filmlerinde ve sosyal medyada artmıştır. Bu çalışmanın benzer araştırmalardan ayrıldığı nokta ise eğitim için kullanılacak olan verinin diğer çalışmalardaki verilerden daha az olması ve sahte video oluşturma çeşitlerinin, yapılarının birlikte ele alınmasıdır. Veri eğitimi için kullanılan materyal GPU ve veri seti olarak ise VoxCeleb veri seti, birkaç kısa video ve birkaç fotoğraftan oluşmaktadır. Kullanılan yöntem ise Çekişmeli üretici ağlar ve Otomatik kodlayıcılar gibi üretken ağlardır. Yapılan çalışma kullanılan video ve fotoğraflarda yüzün karşıya (öne) dönük veya hafif sağa ya da hafif sola dönük iken, yüz hareketinin belirli bir alanda sınırlı olduğunda ve yüzün yavaş hareket ettiğinde yapay zekayı daha iyi eğittiği ve bu eğitim verileri kullanılarak oluşturulan sahte videoların daha başarılı olduğunu göstermiştir.

Anahtar kelimeler: Yüz değiştirme, Yüz canlandırma, Derinsahte video, Derin öğrenme.

2021, 46 sayfa

ABSTRACT

MSc. Thesis

SYNTHESIS OF FAKE PHOTO AND VIDEO CONTAINING FACE PHOTOGRAPHS USING ARTIFICIAL INTELLIGENCE AND DEEP LEARNING TECHNIQUES

Mustafa Salih BAHAR

Tekirdağ Namık Kemal University

Graduate School of Natural and Applied Sciences

Department of Computer Science

Supervisor: Assoc. Prof. Dr. Ercan BULUŞ

Sharing photos from social media and the Web has increased considerably these days. We upload almost every photograph we take, private or not, to the Web in some way. But we are not very conscious that these photos can also be seen and used by malicious people. The purpose of this study emerges at this point. For this purpose, in the study; It is proved that even from a short video or a few photos of a person that actually exists, fake videos containing a face photo of that person can be created. Fake photos and videos can be created by taking these few photos or a short video and training them with deep learning techniques. In fake videos, face swapping of another person can be applied with the face of the person or face reenactment can be applied to the person's face. In re-enactment, facial movements of another person's video can be applied to the face of a source person. Even non-existent human photographs can be produced using techniques such as StyleGAN using a set of photographs of real human faces. These techniques, commonly known as deepfakes technology, are discussed in this study together with the types and structures used on the face. These techniques can be used for educational purposes for children by animating existing photographs of scientists and celebrities who lived in ancient times. This method can be used in puppetry. Instead of actors (Actors-Actresses), scenes can be animated with their photos on the computer. Portraits can be animated. This method can also be used in characters in computer games. Recently, the videos of many actors and artists, especially “Kemal Sunal”, who were created with these technologies, have increased in commercials and social media. The difference of this study from similar studies is that the data used for training is less than the data in other studies, and the types and structures of fake video creation are considered together. The material used for data training is the GPU and the dataset consists of VoxCeleb dataset, several short videos and several photos. The method used is generative networks such as Generative adversarial networks (GAN) and Auto-encoders. The study has shown that in the videos and photos used, when the face is turned forward or slightly to the right or slightly to the left, when the facial movement is limited in a certain area and the face moves slowly, it trains the artificial intelligence better. It has shown that the fake videos created using this training data are more successful.

Key words: Face swapping, Face reenactment, Deepfake video, Deep learning.

2021, 46 pages

İÇİNDEKİLER

ÇİZELGE DİZİNİ

ŞEKİL DİZİNİ

SİMGELER VE KISALTMALAR

AE : Auto-Encoder (Otomatik Kodlayıcı)

CNN : Convolutional Neural Network (Evrişimli Sinir Ağı)

CONV : Convolutional Layer (Evrişim Katmanı)

ConvNet : Convolutional Neural Network (Evrişimli Sinir Ağı)

ÇÜA : Çekişmeli Üretici Ağ

FC : Fully-Connected Layer (Tam Bağlı Katman)

GAN : Generative Adversarial Network (Çekişmeli Üretici Ağ)

Mask R-CNN : Maske Bölgesel Konvolüsyonel Sinir Ağı

MLP : Multilayer Perceptron (Çok Katmanlı Algılayıcı)

OK : Otomatik Kodlayıcı

RNN : Recurrent Neural Network (Tekrarlayan Sinir Ağı)

1. GİRİŞ

Görüntü işleme bir şekilde elde edilmiş, ölçülmüş olan görüntü verilerinden yararlı bilgiler çıkarmak için kullanılan yöntemdir. Derin öğrenme ise bilgisayarın bir yapay sinir ağı ve birçok algoritma ile var olan verilerden yeni veriler elde etmesidir. Derin öğrenme, Yapay zekanın alt alanı olan Makine öğrenmesinin bir alt alanıdır. Makine öğrenmesinde özniteliklerden model kurulurken derin öğrenmede direkt veriden model kurulur. Sahte fotoğraf ve video sentezi, derin öğrenme ve görüntü işleme alanlarının kesişiminde bulunan bir konudur. Yüz fotoğrafları içeren sahte fotoğraf ve video sentezi bir kişinin yüz özelliklerini kullanarak, o kişiye ait olmayan fotoğraf ve video oluşturmakla ilgilidir. Aynı zamanda hiç var olmayan bir kişi yüz özellikleri de oluşturularak fotoğraf ve video sentezlenebilir. Kaynak veriden alınan bu yüz özellikleriyle çeşitli işlemler yapılarak sahte fotoğraflar ve videolar oluşturulabilir.

Derin öğrenme, son yıllarda bilgisayar görüşü alanını güçlendirdiğinden, dijital görüntünün manipülasyonu, özellikle de insan portreleri görüntüsünün manipülasyonu, hızla gelişti ve çoğu durumda fotogerçekçi sonuç elde etti. Yüz değiştirme, kaynağın yüz hareketlerini ve ifade deformasyonlarını korurken hedefe bir kaynak yüzü aktararak sahte içerik oluşturmada göze çarpan bir görevdir. Yüz manipülasyon tekniklerinin arkasındaki temel motivasyon Çekişmeli Üretici Ağlar' dır (GAN' lar) (Goodfellow vd., 2014). StyleGAN (Karras, Laine ve Aila, 2019a), StyleGAN2 (Karras vd., 2019b) tarafından sentezlenen daha fazla yüz, giderek daha gerçekçi hale geliyor ve insan gözünden tamamen ayırt edilemez hale geliyor. GAN tabanlı yüz değiştirme yöntemleriyle sentezlenen çok sayıda sahte video Youtube ve diğer video web sitelerinde yayınlanıyor. Genel olarak sahte video oluşturma Deepfake (Derin sahtelik) olarak adlandırılsa da birçok yöntem ve bu yöntemlerde kullanılan birçok yapı vardır.

Kullanılan yöntemler Çekişmeli Üretici Ağlar (Generative Adversarial Networks, GAN) ve Otomatik kodlayıcılar (Auto-encoders)‘ dır. Çekişmeli üretici ağlar iki ana birimden oluşur: Generator (Üretici) ve Discriminator (Ayırt Edici). Generator (Üretici) aldığı bir işaretten (gürültüden) sanal resimler oluşturmaktadır. Discriminator (Ayırt Edici)‘ a ise gerçek resimler ve sanal resimler bir arada verilip ayırt etmesi istenir. Discriminator, bu resimlere belirli değerler verir. Verdiği bu değerlere göre doğru ya da yanlış ayırt ettikleri bildirilir ve bu şekilde eğitilir. Generator ise oluşturduğu sanal resimlerle, tekrar bu yapıdan geçirilerek eğitilir. Bu iki yapı birbiri ile yarışarak, eğitilerek daha başarılı bir model ortaya çıkar. Otomatik kodlayıcı ise aldığı görüntü verilerinden gizli bir uzay vektörü oluşturur. Bu vektörü kullanarak

daha sonra görüntünün temsilini oluşturur. Otomatik kodlayıcı, Çekişmeli üretici ağdaki Generator' a benzetilebilir. Burada gerçek veri kullanılarak bir çıktı oluşturulduğu için Discriminator' a ihtiyaç yoktur. ÇÜA' da ise Generator alınan bir gürültüden (ses, video vb.) çıktı oluşturduğu için bunun doğruluğunu-yanlışlığını kontrol eden Discriminator' a ihtiyaç vardır.

Genel olarak 3 çeşit sahte fotoğraf ve video sentezi vardır. Bunlar: Yüz Değiştirme (Face Swapping), Yüz Hareketlendirme, Canlandırma (Face Re-enactment), Yüz Oluşturma (Face Generation). Bu tekniklerde de kullanılan genel olarak 2 çeşit yapı vardır. Bunlar: Otomatik Kodlayıcı (Auto- Encoder), Çekişmeli Üretici Ağ (GAN).

1.1. Konvolüsyonel Sinir Ağları

Konvolüsyonel Sinir Ağları (LeCun, Bengio ve Hinton, 2015) (CNN veya ConvNet) bir çeşit çok katmanlı algılayıcıdır (MLP). Görme merkezi hücreleri görselin tümünü içerecek biçimde alt alanlara bölünmüştür. Basit hücreler, kenara benzeyen özelliklerde odaklanırken, karmaşık hücreler geniş alıcılar yardımıyla, görselin bütününe odaklanır. Konvolüsyonel sinir ağları, bilgisayarlı görü alanında başarısı ispatlanmış binlerce değişik problem için tasarlanan yüzlerce modeline rastlanabilecek, derin öğrenmenin bir konusudur. Örneğin, robot ya da otonom araçların görü sistemlerinde; trafik işaretleri, nesne ve yüz tanıma vb. alanlarda faydalanılır. Bir ileri beslemeli sinir ağı olan CNN, hayvanların görme merkezi ilham alınarak ortaya çıkmıştır. Burada yapılan matematiksel konvolüsyonel işlem, nöronun uyarı bölgesinden uyaranlara verdiği cevaptır. Şekil 1.1' deki gibi CNN görüntüleri farklı katmanlarla işler. Bu katmanlar ve amaçları aşağıda belirtilmiştir:

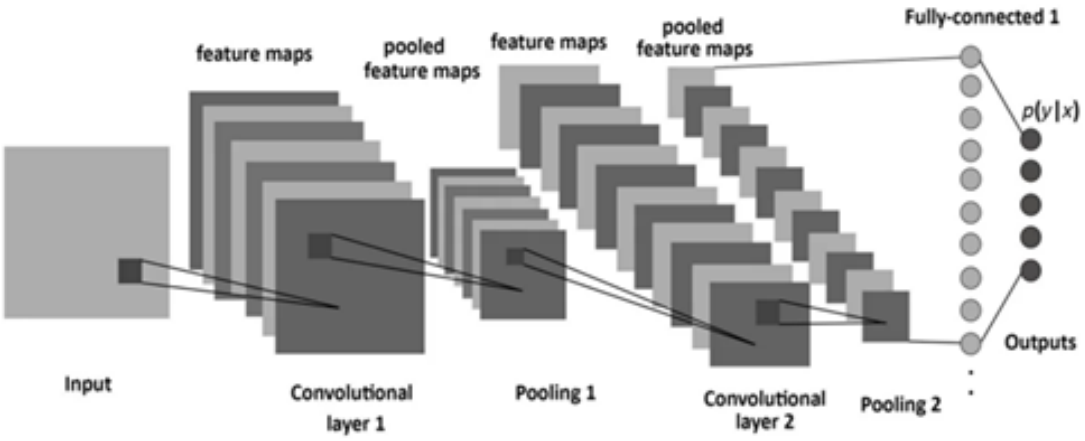
Evrişim katmanı (Convolutional layer): Özelliklerin saptanmasında yararlanılır.

Doğrusal olmayan katman (Non-linearity layer): Doğrusal olmayanlığın sisteme verilmesi amacıyla kullanılır.

Ortaklama katmanı (Pooling layer): Ağırlıkların sayılarını düşürür ve uygunluk kontrolü yapar.

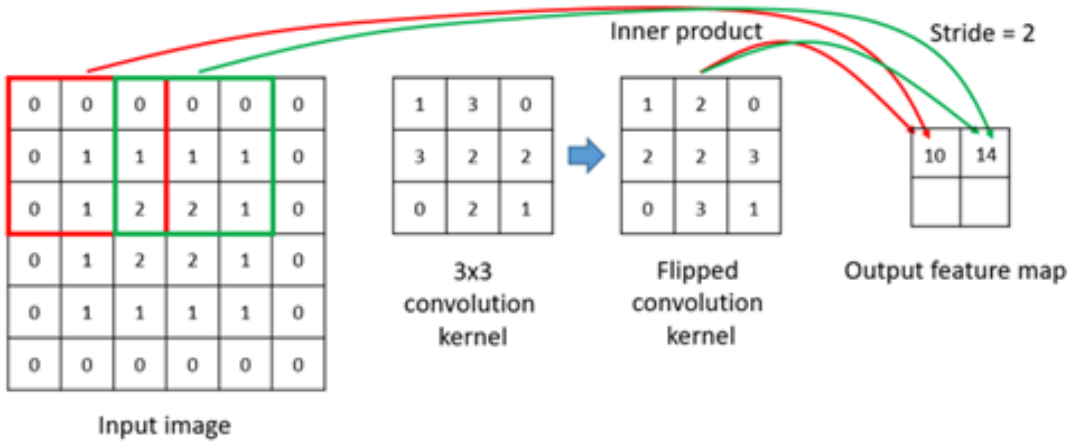
Düzleştirme katmanı (Flattening layer): Temel sinir ağına veri hazırlar.

Tam bağlı katman (Fully-connected layer): Sınıflama amacıyla yararlanılan sinir ağıdır.



Şekil 1.1. Konvolüsyonel sinir ağlarının genel mimarisi (LeCun vd., 2015)

Şekil 1.2' deki gibi Konvolüsyon katmanı (CONV) konvolüsyon işlemini yerine getiren filtreleri, I girişini boyutuna göre tarar iken yararlanır. Hiper parametreleri F filtre boyut ve S adımlarını kapsar. Oluşan çıktı O, öznitelik haritası ya da aktivasyon haritası biçiminde isimlendirilir. Buradaki katman CNN'nin yapıtaşıdır. Resmin özelliklerinin algılar. Buradaki katman, görüntülerdeki düşük ve yüksek dereceli özelliklerin çıkarılması amacıyla görüntülere filtre uygulamaktadır. Mesela, burada filtre kenarlar için bir algılayıcı kullanılabilir. Buradaki filtre genel olarak çok boyutlu olmakta ve piksel değerlerini içermektedir. (5x5x3) Buradaki matrisin yüksekliği 5 , genişliği 5 ve derinliği ise 3' tür.



Şekil 1.2. İki boyutlu evrişim işlemi (LeCun vd., 2015)

$$G[m,n] = (f * h)\,[m,n] = \sum_j \sum_k h[j,k]\, f[m-j, n-k] \quad (1.1)$$

Özellik haritası m. satır n. sütun çıktı elemanları, f girdi imajı matrisi ve h konvolüsyon çekirdeği matrisi olarak tanımlandığında Denklem 1.1' deki gibi bulunabilir.

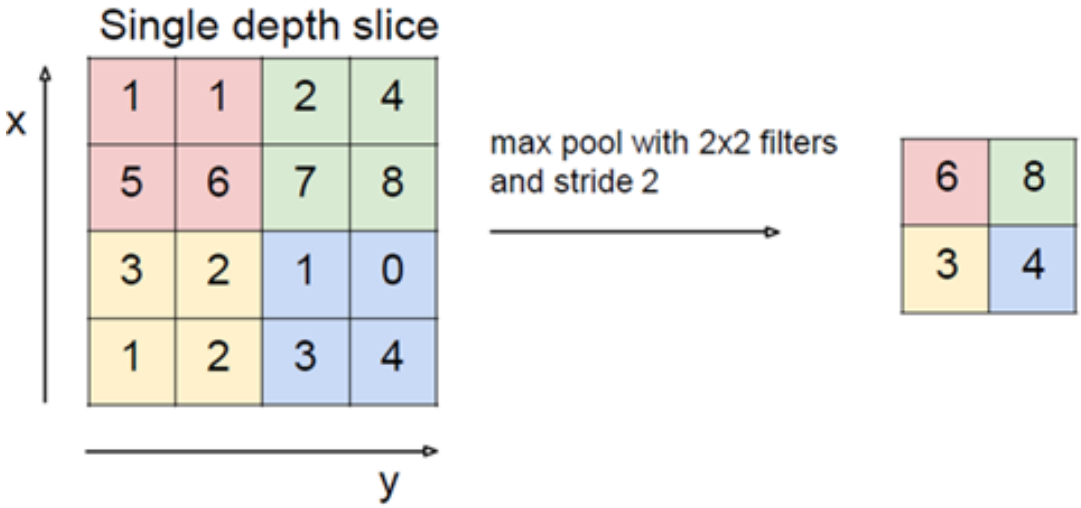
İki boyut bilgide kullanılacak filtrede x ve y eksenlerine göre simetri uygulanır. Bütün değerler matriste eleman olarak çarpılır ve bütün değerler toplamı çıkış matrisinde ilgili eleman olarak belirlenir. Bu çapraz korelasyon ilişkisi olarak da tanımlanabilir. Giriş verileri (mesela; fotoğraf) tek kanallıyken bu işlem basit olarak yapılabilir. Fakat giriş verileri değişik formatta ve kanal sayılarında bulunabilir. Renkli görüntüler, Kırmızı-Yeşil-Mavi (RGB) 3 kanaldan oluşmaktadır. Bu şartta konvolüsyon işlemi 3 kanalda uygulanır. Çıkış işaretlerinin kanal sayıları da uygulanmış filtre kanalları/sayılarıyla eşit hesaplanmaktadır. Burada hesaplanmış işlem sinir ağında bir katman olarak farz edilirse, giriş görüntüleri ve filtreler de devamlı geri yayılım ile güncellenmiş ağırlıkların matrisleridir. Aktivasyon fonksiyonları uygulanmış çıkış matrislerine son bir skaler b (bias) değeri katılır.

Bütün Konvolüsyonel katmanlardan hemen sonra genel olarak doğrusal olmayanlık (Non-Linearity) katmanı uygulanır. Doğrusal olmayanlık katmanı, aktivasyon fonksiyonlarından birini kullandığı için aktivasyon katmanı (Activation layer) biçiminde de isimlendirilir. Önceden, sigmoid ya da tanh benzeri doğrusal olmayanlık fonksiyonları kullanılırdı, fakat sinir ağındaki eğitim hızında daha iyi sonuç veren Rectifier (ReLu) fonksiyonu olduğundan şimdi bu fonksiyon kullanılmaktadır.

Kenar bilgisi, görüntülerden alınan özniteliklerin içerisinde en fazla ihtiyaç duyulan bir bilgidir. Giriş bilgilerinin yüksek frekans bölgelerini simgeler. Buradaki özniteliklerin oluşturulması için dikey ve yatay olarak filtreler ayrı ayrı uygulanır. Temel yöntemlerde- Sobel, Prewitt, Gabor benzeri filtreler- filtre, görüntüler üzerindeki konvolüsyonel işleme dahildir. Oluşturulan çıkış, görüntülerin kenar bilgisini ifade eder.

Ortaklama katmanı (POOL), bir miktarda uzamsal olarak değişkenlik göstermekte olan bir konvolüsyonel katmandan hemen sonra uygulanmış örnekleme işlemi olarak ifade edilir. Özel olarak, maksimum ve ortalama ortaklama, sırayla maksimum ve ortalama değerlerin hesaplandığı özel ortaklama çeşitleridir.

Buradaki katman, Konvolüsyonel Sinir Ağındaki art arda konvolüsyonel katmanlar arasına genellikle katılan bir katmandır. Buradaki katmanın amacı; gösterim kayma boyutunu, ağ içi parametreleri ve hesaplamayı düşürmek içindir. Bu şekilde ağ içindeki uyumsuzluk kontrol edilir. Çok fazla ortaklama işlemi bulunur, ama en popüler maksimum ortaklamadır. Benzer prensip ile çalışan ortalama ortaklama ve L2-norm ortaklama algoritmaları da bulunur.



Şekil 1.3. Maksimum ortaklama işlemi (LeCun vd., 2015)

Bu katmanda genellikle Şekil 1.3'teki gibi maksimum ortaklama çeşidi kullanılmaktadır. Ağdaki bu kısımdaki katmanda öğrenilmiş parametre bulunmaz. Giriş matrislerinin kanal sayılarını sabit bırakarak yükseklik ve genişlik gibi bilgilerini düşürür. Hesaplamadaki karmaşıklığı azaltmak amacıyla uygulanan adımdır. Fakat Hinton' un kapsül teorisine göre verilerdeki değerli bazı bilgilerin de yok olmasına neden olduğundan başarım kısmından taviz vermektedir.

Genellikle konum bilgilerinin pek önemli olmadığı problemler için bile oldukça iyi sonuç verir. Seçilmiş ortaklama boyutu içinden piksellerin en büyüğü çıkış kısmına aktarılır. Yukarıda Şekil 1.3' teki örnekte 2x2 maksimum ortaklama işlemi 2 basamak (piksel) kaydırılıp uygulandığı görülmektedir. İlgili dört elemanın bulunduğu alanın en büyük değeri çıkış kısmına aktarılır. Çıkış kısmına dörtte bir boyutlu bir veri oluşur.

Tam bağlantı katman (FC), her girişin bütün nöronlara bağlandığı bir giriş üzerinde çalışmaktadır. Varsa, FC katmanı özellikle CNN mimarisinde sona doğru bulunmaktadır ve sınıf skorları vb. hedefleri optimize amacıyla kullanılır.

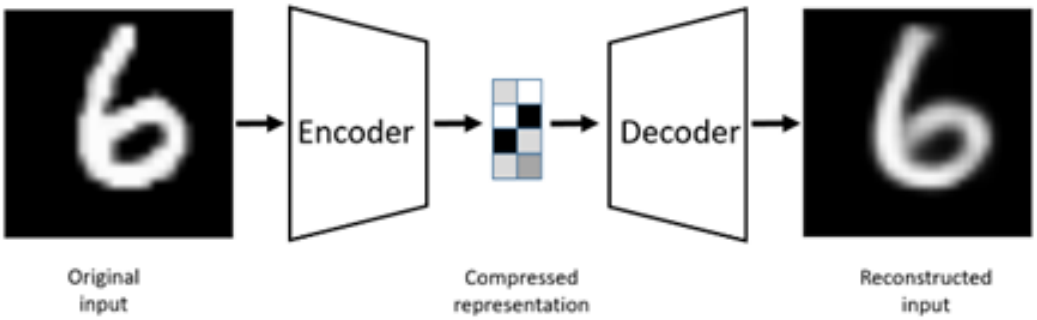
Buradaki katman Konvolüsyonel Sinir Ağının en önemli ve son katmanıdır. Verileri, Düzleştirme işleminden alır ve sinir ağını kullanarak öğrenme işlemi gerçekleştirmektedir.

Bilgisayarın filtre değerini (ya da ağırlıklarını) ayarlayabilmenin yolu, geri yayılım (back propagation) (Rumelhart, Hinton ve Williams, 1986) olarak isimlendirilen bir eğitim kısmıdır. Geri yayılım, asıl ve istenen çıktıyı hesaba katarak sinir ağının ağırlıklarını güncellemek amacıyla uygulanır. Ağırlıklara göre kaybın derecesini hesaplamak amacıyla kaybı yeniden geri yönde yayar.

1.2. Kullanılan Yapılar

1.2.1. Otomatik kodlayıcı

Otomatik kodlayıcı (Bank, Koenigstein ve Giryes, 2020), Encoder ve Decoder olmak üzere 2 yapıdan oluşur. Encoder yapısında veri alınır ve bu veriden gizli bir uzay vektörü oluşturulur. Decoder kısmında da bu gizli uzay vektörü kullanılarak giriş verisi yeniden yapılandırılarak bir çıkış verisi oluşturulur. Otomatik kodlayıcı, verilerin verimli bir şekilde nasıl sıkıştırılacağını ve kodlanacağını öğrenen, ardından veriyi indirgenmiş kodlanmış gösterimden orijinal girdiye mümkün olduğunca yakın bir temsile nasıl yeniden yapılandırılacağını öğrenen, denetimsiz bir yapay sinir ağıdır. Amacı giriş verisinden özellikleri çıkarıp bu özellikleri kullanarak yeniden inşa edilmiş çıkış verisi oluşturmaktır. Boyut azaltma, görüntü ve seste gürültü giderme, anormallik – aykırı değer algılama, resim boyama ve bilgi alma gibi konularda kullanılabilir. ÇÜA' ya göre oldukça küçük bir uzay vektörü oluşur ve eğitimi daha kolaydır. ÇÜA' lar gibi üretken modellerdir. Yapısında Konvolüsyonel Sinir Ağları (CNN) bulunabilir.



Şekil 1.4. Bir otomatik kodlayıcı örneği (Bank vd., 2020)

Şekil 1.4' teki gibi bir otomatik kodlayıcıda girdi görüntüsü sıkıştırılmış bir gösterime kodlanır ve sonra kodu çözülür.

Bir otomatik kodlayıcı, girdinin sıkıştırılmış bir temsilini geliştirmek için veri içindeki yapıyı keşfedebilen bir sinir ağı mimarisidir. Denetimsiz bir öğrenme tekniğidir. Genel otomatik kodlayıcı mimarisinin birçok farklı varyantı, sıkıştırılmış gösterimin orijinal veri girişinin anlamlı özelliklerini temsil etmesini sağlamak amacıyla mevcuttur; otomatik kodlayıcılarla çalışırken tipik olarak en büyük zorluk, modelinizin anlamlı ve genelleştirilebilir bir gizli alan temsilini gerçekten öğrenmesini sağlamaktır. Otomatik kodlayıcılar, eğitim sırasında verilerden keşfedilen özniteliklere (yani, giriş özelliği vektörü arasındaki korelasyonlara) dayalı olarak verileri nasıl sıkıştıracaklarını öğrendikleri için, bu modeller tipik olarak yalnızca modelin eğitim sırasında gözlemlediği gözlem sınıfına benzer verileri yeniden yapılandırabilir.

Otomatik kodlayıcıda ağırlıkları güncellemek için kullanılan loss fonksiyonu Denklem 1.2'deki gibi encoder ve decoder kısmı için tanımlanan θ ve φ parametrelerine bağlıdır. Encoder G_φ ile temsil edilirken, decoder F_θ ile temsil edilir ve bunlar yalnızca sinir ağının ağırlıklarını ve bias değerlerini ifade eder.

$$L(\theta, \varphi) = \frac{1}{n}\sum_{i=1}^{n}(x^i - f_\theta(g_\varphi(x^i)))^2 \tag{1.2}$$

1.2.2. Çekişmeli üretici ağ

ÇÜA (GAN) (Goodfellow vd., 2014), Üetici (Generator) ve Ayırt edici (Discriminator) olmak üzere 2 yapıdan oluşur. Üretici ağ bir gürültüden (random sayılar) sahte imajlar üreten ağdır. Ayırt edici ağ ise gerçek ve sahte imajları alarak bunları ayırt eden ağdır. Üretici ağ hiçbir şekilde gerçek verileri göremez. Üretici ağ sürekli yeni veriler üretmeyi öğrenirken, ayırt edici ağ ise girdi olarak kabul edilen veri seti ile üretilen verileri ayırt etmeyi öğrenir, bu süreçte her iki ağ da ne üretip ne ayırt edeceğini kuralsız olarak kendi kendine keşfettiğinden dolayı bu bir gözetimsiz öğrenme tipidir. Her imaja 0 ile 1 arasında değer verilir. Bu değerlere göre Geri Yayılım (Backpropagation) ile her tekrar (epoch)' da ağlar, ağırlık (weight) ve kayıp (loss) değerlerini günceller ve birbiriyle yarışarak gelişir. Ağların eğitimi oto-kodlayıcılara göre daha karmaşıktır. Görüntü işlemenin birçok alanında, ses üretiminde vb. birçok konuda kullanılabilir. Yapısında Konvolüsyonel Sinir Ağları (CNN) bulunur. Hem Generator hem Discriminator birer sinir ağından oluşur. Sinir ağları aldığı veriyi öğrenebilen yapılardır.

Generator' ın gerçekçi resimler üretmesi, Discriminator' ın gerçek ve gerçek olmayan resimleri ayırt etmesi istenir. Bu ancak sinir ağları ile oluşturulabilir.

Normal yapay sinir ağları bir şeyleri tahmin etmede, bir şeyleri sınıflandırmada, problem çözmede oldukça iyidir. Ancak kendi başına yeni bir veri yaratmakta pek iyi değildir. Yapay zeka araştırmacıları uzun zamandır kendi kendine yeni ve anlamlı veriler üreten bir yapay zeka üretmeyi hayal etmiştir. Gans ise tam olarak bunu sağlıyor. Gans resimleri öğrenip neredeyse gerçekle bir farkı olmayan yeni resimler oluşturabilir. Gan' ler kullanılan nesneleri, doğadaki hayvanları, insan yüzlerini tanıyarak aslında gerçek olmayan ama ilk bakışta gerçek olmadığı fark edilemeyecek yeni resimler üretebiliyor.

Gans eğitilirken iki Network de sıfırdan eğitilmeye başlanıyor. Yani önceden eğitilmiş gerçek ve sahte resim ayırt edebilecek bir Discriminator yoktur. Hem Generator hem Discriminator sıfırdan beraber eğitilecektir. Eğitim esnasında bu ikisi birbirleriyle bağlantılı olacaktır. Mesela gemi resimleri üretmek isteniyor olsun. Eğitim yapılırken Generator resim üretecektir. Daha sonra üretilen resim Discriminator' a verilir. Aynı zamanda Discriminator' a gerçek gemi resimleri de verilmesi gerekir. Discriminator üretilen resim gerçek resme ne kadar benziyor buna karar verecektir. Daha sonra Discriminator, Generator' a neden ürettiği resmin gemi olmadığını söyleyecektir. Bu şekilde döngü devam edecek ve zaman içerisinde Generator oldukça gerçekçi resimler üretebilecektir. Kısacası beraber öğrenip gelişeceklerdir. Bu gerçek

hayattan örnekle en iyi şu şekilde açıklanabilir: Mesela sahte para üreten bir kişi olsun. İlk ürettiği para pek gerçekçi değildir ve polis bunu hemen yakalar. Aynı zamanda polis sahte para üreten kişiye nerede hata yaptığını da söylüyor olsun. Sahte para üreten kişi yeniden dener. Daha gerçekçi bir para üretir. Polis yine yakalar ve yine nerede hata yaptığını söyler. Bu döngü böyle devam eder. Sahte para üreten kişi sürekli gerçeğe yaklaşır. Aynı zaman da polis de gerçek paradaki her ayrıntıyı zaman içerisinde öğrenerek sahte parayı gerçek paradan ayırt etmeye çalışır. Tabii bu örnekte polisin sahte para basan kişiyi tutuklamadığı farz ediliyor. İkisi de birbirinden öğrenerek zaman içerisinde kendilerini geliştiriyorlar. Artık bir noktadan sonra üretilen sahte parayı normal bir insan ayırt edemeyecek duruma geliyor. Hem sahte para üreticisi hem de polis birbirine karşı yarışıyor. Bu sayede ikisi de zaman içerisinde yaptıkları işi daha iyi yapıyorlar. Gans mantıksal olarak kabaca bu şekildedir.

Şekil 1.5'teki gibi üretici ağ sürekli yeni veriler üretmeyi öğrenirken, ayırt edici ağ ise girdi olarak kabul edilen veri seti ile üretilen verileri ayırt etmeyi öğrenir, bu süreçte her iki ağ da ne üretip ne ayırt edeceğini kuralsız olarak kendi kendine keşfettiğinden dolayı bu bir gözetimsiz öğrenme tipidir.

Şekil 1.5. Bir çekişmeli üretici ağ örneği (Goodfellow vd., 2014)

Dicriminator (Ayırt edici) D, Generator (Üretici) G, θ_d discriminator' ın parametreleri, θ_g generator'ın parametreleri, $P_z(z)$ giriş gürültü dağılımı, $P_{data}(x)$ original veri dağılımı, $P_g(x)$ üretilen dağılım olarak tanımlandığında ağırlıkları güncellemek için kullanılan loss fonksiyonu denklemleri denklem 1.3, 1.4, 1.5 ve 1.6 gibi bulunabilir. Denklem 1.3 discriminator için,

Denklem 1.4 generator için loss denklemini göstermektedir. İki ağ için birleşik loss denklemi de denklem 1.5' te gösterilmiştir.

$$L^{(D)} = \max[\log(D(x)) + \log(1 - D(G(z)))] \tag{1.3}$$

$$L^{(G)} = \min[\log(D(x)) + \log(1 - D(G(z)))] \tag{1.4}$$

$$L = \underbrace{\min}_{G}\ \underbrace{\max}_{D}\ [\log(D(x)) + \log(1 - D(G(z)))] \tag{1.5}$$

Denklem 1.5' teki birleşik loss denklemi tek bir veri için geçerlidir. Tüm veri seti düşünüldüğünde birleşik loss hesabı için Denklem 1.6 geçerli olur.

$$\underbrace{\min}_{G}\ \underbrace{\max}_{D}\ V(D,G) = \underbrace{\min}_{G}\ \underbrace{\max}_{D}\ (E_{x\sim P_{data}(x)}[\log D(x)] + E_{z\sim P_z(z)}[\log(1 - D(G(z)))]) \tag{1.6}$$

1.3. Kullanılan Çeşitler

1.3.1. Yüz değiştirme

Bir bilgisayar görüşü (computer vision) konusu olan yüz değiştirme genel olarak imaj ya da videodaki yüzleri tanıma, hizalama, maskeleme, yüz segmentasyonu ve başka bir kişinin yüzü ile değiştirme gibi alt işlemlerin bütünüdür. Mevcut medyadaki kişinin yüzünün kaynak medyadaki kişinin yüzü üzerinde birleştirilmesi ve üst üste konması ile sahte medya üreten bir metottur. Yapay sinir ağlarını kullanır. Kullanılanacak model- mimariye göre OK veya ÇÜA kullanabilir.

1.3.2. Yüz hareketlendirme, canlandırma

Bir bilgisayar görüşü konusu olan yüz canlandırma genel olarak kaynak yüzün şeklini hedef yüze aktarırken hedef yüzün görünümün ve kimliğinin de korunmasını sağlar. Yani kısaca hareketin aktarılması da denebilir. Örneğin kaynak videodaki yüzde göz kırpılıyorsa hedef yüzdeki imaja da göz kırpma uygulanır. Genellikle kaynak yüz için bir video, hedef yüz için bir veya birkaç imaj kullanılır. Genellikle ÇÜA kullanılır. Portre ve tabloların canlandırılmasında kullanılabilir. Yüz şeklini tahmin etmek için çeşitli yüz işaret algılayıcıları kullanılabilir.

1.3.3. Yüz oluşturma

Görüntü oluşturma (image generation)' nın bir alt alanı olan yüz oluşturma, var olan bir veri kümesinden eğitilerek yeni yüzler oluşturma ile ilgilidir. ÇÜA' ları kullanır. Bu sayede gerçekte hiç var olmayan gerçekçi insan yüz resimleri üretilebilir. Yüz oluşturmanın en bilinen örneği Nvidia araştırmacılarının tanıttığı StyleGAN (Karras vd., 2019a)' dir. StyleGAN (Karras vd., 2019a) Nvidia' nın CUDA (NVIDIA, Vingelmann ve Fitzek, 2020) yazılımını, GPU donanımlarını ve Tensorflow (Abadi vd., 2016) kütüphanesini kullanır. Konuşan kafa oluşturma (Talking head generation) ve Konuşan yüz oluşturma (Talking face generation) gibi alt alanlara ayrılır. Konuşan kafa oluşturma, bir kişinin bir dizi görüntüsünden konuşan bir yüz oluşturma görevidir. Konuşan yüz oluşturma, verilen konuşma semantiğine karşılık gelen bir dizi yüz görüntüsü sentezlemeyi amaçlamaktadır.

2. KAYNAK ÖZETLERİ

"DeepFaceLab: A simple, flexible and extensible face swapping framework" adlı çalışmada Perov vd. (2020) tarafından oluşturulan yüz değiştirme için yapılmış kolay kullanımlı bir framework anlatılmaktadır. Eğitim paradigmasına göre OK veya ÇÜA kullanılabilir. Burada kullanılan Deepfakes (t.y.) modeli, yüz çiftleri üzerinde eğitim gerektirir. Genel olarak Extraction, Training ve Conversion gibi aşamalarla sahte video sentezi anlatılmaktadır. Extraction da Face Detection, Face Alignment ve Face Segmentation olmak üzere üçe ayrılır. Yani Extraction kısmında resimleri veya videoyu frame' lerine ayırır. Frame' lerdeki yüzleri bulur. Yüzleri hizalar. Yüzleri diğer kısımlardan ayırarak yüz segmentini oluşturur. Training kısmında iki farklı kişinin yüzünü alarak birbiriyle değiştirecek şekilde

eğitir. Conversion kısmında da eğittiği modeli kullanarak istenilen videoda yüz değiştirme uygular.

"FSGAN: Subject Agnostic Face Swapping and Reenactment" adlı çalışma Nirkin, Keller ve Hassner (2019) tarafından yüz değiştirme ve yüz canlandırma için oluşturulmuş modeldir. ÇÜA yapısını ve RNN tekrarlayan sinir ağını kullanır. Yüzler üzerinde eğitim gerektirmeden yüz çiftlerine uygulanabilir. Model; Face reenactment and segmentation, Face inpainting, Face blending olmak üzere üç kısımdan oluşur. Face reenactment and segmentation kısmında kaynak ve hedef yüzleri yeniden hareketlendirerek yüz segmentlerini alır. Hedef görüntünün yüz ve saç segmentasyon maskesini tahmin eder. Face inpainting kısmında kaynak yüzün eksik kısımlarını boyar. Face blending kısmında segmentasyon maskesini kullanarak iki yüzü harmalanma yapar.

"Few-Shot Adversarial Learning of Realistic Neural Talking Head Models" adlı çalışma Zakharov, Shysheya, Burkov ve Lempitsky (2019) tarafından Samsung Yapay Zeka Laboratuvarlarında yapılmış bir çalışmadır. ÇÜA yapısını kullanır. Yüz üretimi konusunun bir alt dalı olan Konuşan Kafa Üretimi (Talking Head Generation) modelidir. Bir kişinin Konuşan Kafa modelinin; birkaç görüntüsünden, hatta tek bir görüntüsünden öğrenilmesi için geniş bir video veri kümesi üzerinde uzun süre meta- öğrenme gerçekleştirir. VoxCeleb (Nagrani, Chung ve Zisserman, 2017) veri setini kullanır. Few-shot learning (R. Zhang, Che, Ghahramani, Bengio ve Song, 2018) ve one-shot learning öğrenme metotlarını kullanır. Tabloların, resimlerin canlandırılmasında kullanılabilir. Modelde bir Embedder yardımıyla gerçek resimlerde renk ve yüz şekillerini, çok katmanlı algılayıcılarla Adain parametreleri olarak Generator'a verir. Generator da gürültü olarak alacağı yüz şekli input' undan sahte resimler üretir. Sahte resim, yüz şekli ve gerçek resim Discriminator' a verilerek loss değerleri hesaplanarak ağırlıklar güncellenir.

"First Order Motion Model for Image Animation" adlı çalışma Siarohin, Lathuilière, Tulyakov, Ricci ve Sebe (2020a) tarafından görüntü animasyonu için oluşturulmuş bir çalışmadır. ÇÜA yapısını kullanır. Görüntü animasyonu, bir kaynak görüntüdeki bir nesnenin başka bir videonun hareketine göre canlandırılmasıyla bir video dizisi oluşturmayı amaçlar. Aynı kategorideki nesneleri (ör. Yüzler, insan vücutları) gösteren bir dizi video üzerinde eğitildikten sonra, yöntem bu sınıftaki herhangi bir nesneye uygulanabilir. Karmaşık hareketleri desteklemek için, yerel afin dönüşümleri (affine transform) (Berger, 1987) ile birlikte bir dizi öğrenilmiş anahtar nokta' (key points)' dan oluşan bir temsil kullanır. Bir Generator ağı, hedef

hareketler sırasında ortaya çıkan tıkanıklıkları modeller ve kaynak görüntüden çıkarılan görünümü ve hareket videosundan türetilen hareketi birleştirir. Monkey-Net (Siarohin, Lathuilière, Tulyakov, Ricci ve Sebe, 2019) altyapısını kullanır. Yüz değiştirme ve yüz canlandırma alanlarında kullanılabilir.

"Motion-supervised Co-Part Segmentation" adlı çalışma Siarohin vd. (2020b) tarafından yüzün bir kısmı için değiştirme yapılması için oluşturulmuş bir çalışmadır. Saç, sakal, gözler ve dudaklar için değiştirme yapılabilir. ÇÜA yapısını kullanır. Co-part segmentasyon metotları çoğunlukla, eğitim için büyük miktarda açıklamalı veri gerektiren denetimli (supervised) bir öğrenme ortamında çalışır. Bu sınırlamanın üstesinden gelmek için kendi kendine denetlenen (self-supervised) bir derin öğrenme yöntemi oluşturmuşlardır. Segment motion bölümüyle yüzün bir kısmında veya kişinin duruşunda değişiklik yaparak hareketlendirme yapılır. Yüz değiştirme ve yüz hareketlendirme alanlarında kullanılabilir.

"One-shot Face Reenactment" adlı çalışma Y. Zhang vd. (2019) tarafından yüz canlandırma modeli oluşturmak için yapılan çalışmadır. OK yapısını kullanır. One-shot learning öğrenme metodunu kullanır. "ReenactGAN: Learning to Reenact Faces via Boundary Transfer" adlı çalışma Wu, Y. Zhang, Li, Qian ve Loy (2018) tarafından yüz canlandırma modeli oluşturmak için yapılan çalışmadır. OK ve ÇÜA yapılarını kullanır.

"X2Face: A network for controlling face generation by using images, audio, and pose codes" adlı çalışma Wiles, Koepke ve Zisserman (2018) tarafından konuşan kafa üretimi (talking head generation) modeli oluşturmak için yapılan çalışmadır. OK kullanır. Model, Embedding Network ve Driving Network olmak üzere iki kısımdan oluşur. Kaynak videoyu, sürüş videosunun hareketleriyle yeniden oluşturur. "A Style-Based Generator Architecture for Generative Adversarial Network" adlı çalışma Karras vd. (2019a) tarafından Nvidia' nın yüz üretimi için yaptığı çalışmadır. ÇÜA yapısını kullanır. Style Based Generator kullanır.

3. MATERYAL VE YÖNTEM

Materyal olarak iki çeşit GPU kullanılmıştır. Bunlar: Tesla P40 ve Nvidia Geforce GTX 1050' dir.

Veri seti olarak çeşitli videolar kullanılmıştır. VoxCeleb (Nagrani vd., 2017) ve CelebA-HQ (Karras, Aila, Laine ve Lehtinen, 2017) veri setlerinin eğitilmiş modeller üzerindeki

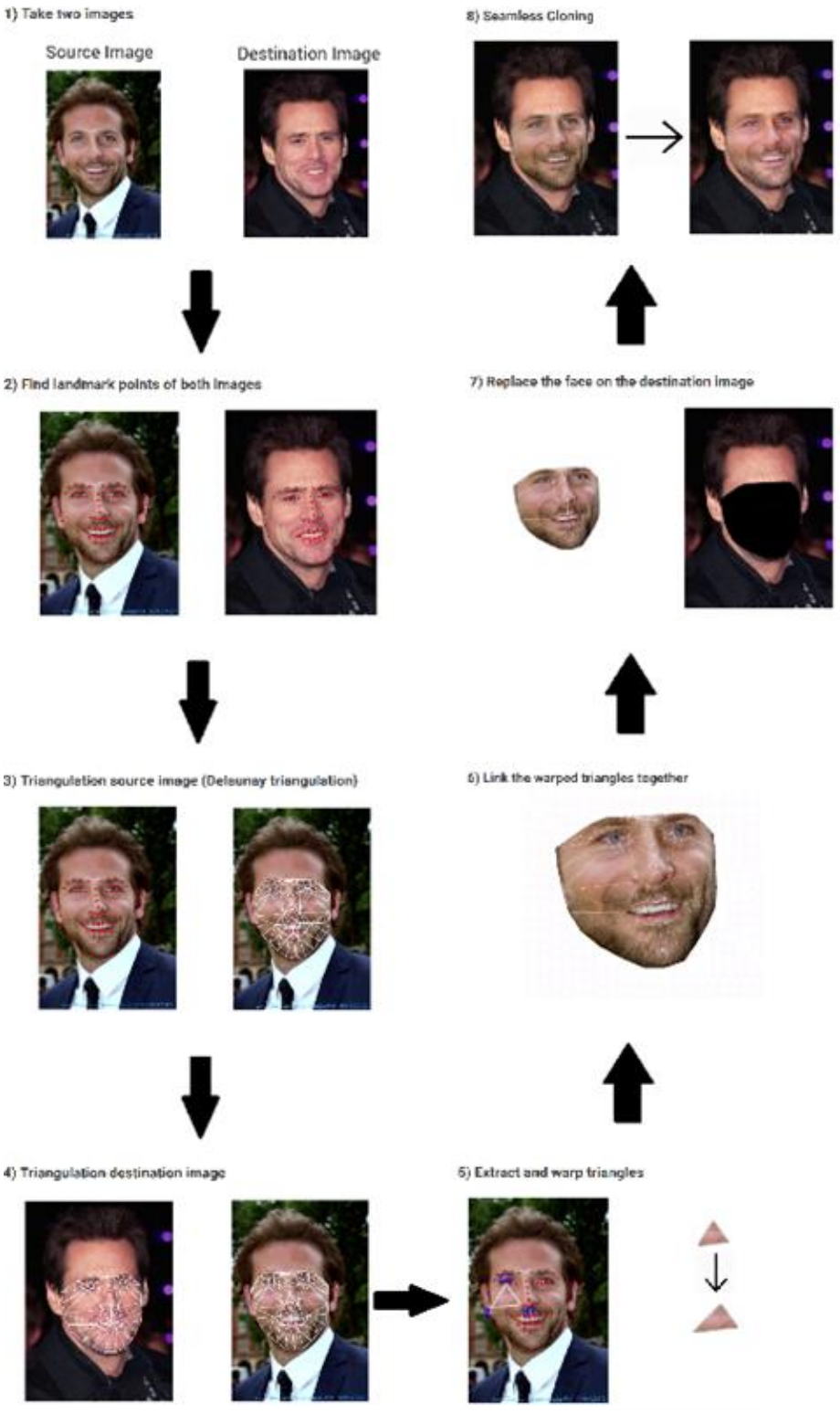
ağırlıkları kullanılmıştır. Yöntem olarak yüz değiştirme, yüz canlandırma ve yüz üretimi modelleri kullanılmıştır.

Yüz değiştirme modeli olarak Deepfakes (t.y.), yüz canlandırma modeli olarak First order motion model (Siarohin vd., 2020a), yüz üretimi modeli olarak StyleGAN (Karras vd., 2019a) kullanılmıştır.

Deepfakes modelinde çeşitli videolar (Örneğin, Şekil 3.8' deki çıktılar için 79 ve 195 saniyelik iki video kullanılarak eğitilmiştir.) kullanılmıştır. First Order Motion Model (Siarohin vd., 2020a)' de Voxceleb (Nagrani vd., 2017) veri setinin eğitilmiş modeldeki ağırlıkları kullanılmıştır. StyleGAN (Karras vd., 2019a) modelinde de CelebA-HQ (Karras vd., 2017) veri setinin eğitilmiş modeldeki ağırlıkları kullanılmıştır.

Genel olarak dört uygulama gerçekleştirildi. İlk olarak OK veya ÇÜA kullanılmadan basit bir yüz değiştirme işlemi gerçekleştirildi. İkinci olarak OK kullanılarak bir yüz değiştirme işlemi gerçekleştirildi. Üçüncü olarak da ÇÜA kullanılarak bir yüz canlandırma işlemi gerçekleştirildi. Son olarak yine ÇÜA kullanılarak bir yüz üretimi işlemi gerçekleştirildi.

3.1. Üretken Modeller Kullanılmadan Basit Bir Yüz Değiştirme



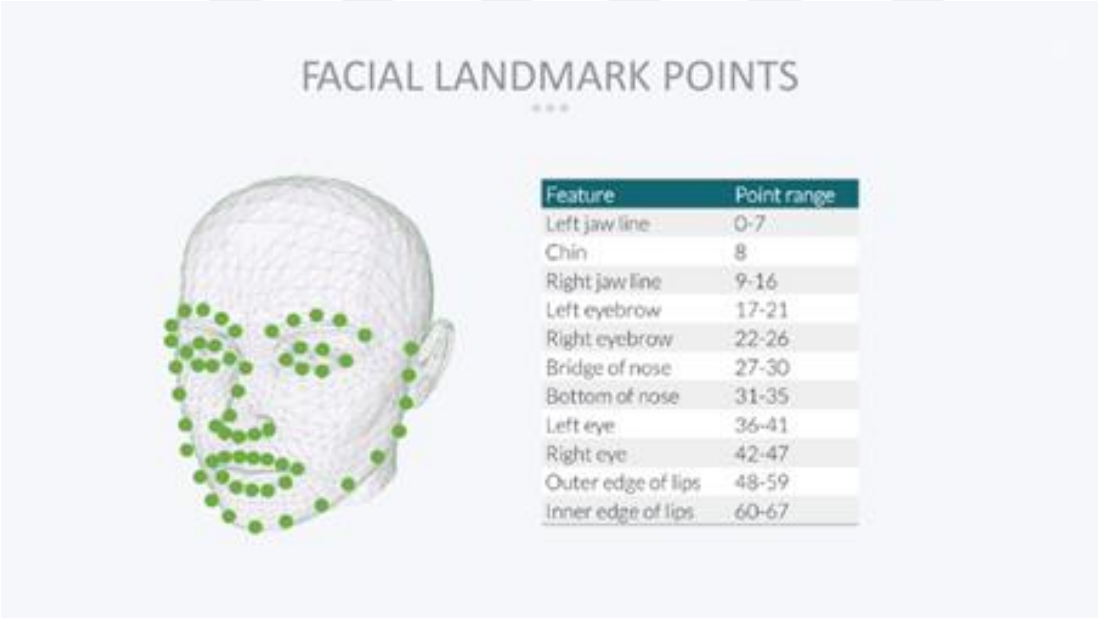
Şekil 3.1. Üretken modeller kullanılmadan basit bir yüz değiştirme metodunun aşamaları (Canu, 2019)

Şekil 3.1’ de aşamaları gösterilen bu metot (Canu, 2019)’ ta OpenCV (Bradski, 2000), Dlib (King, 2009) kütüphaneleri ve Python ortamı kullanılmıştır. Dlib kütüphanesi, çok çeşitli makine öğrenimi algoritmaları içerir. OpenCV (Bradski, 2000) kütüphanesi de gerçek zamanlı bilgisayarlı görü uygulamalarında kullanılan açık kaynaklı bir kütüphanedir. Bu metot sekiz adımdan oluşur.

Metodun ilk adımında kaynak görüntü ve hedef görüntü olmak üzere yüz çiftleri alınarak görüntüler dosyalarda biriktirildi. "Kaynak görüntü", yüzü aldığımız görüntüdür.

Metodun ilk adımında kaynak görüntü ve hedef görüntü olmak üzere yüz çiftleri alınarak görüntüler dosyalarda biriktirildi. "Kaynak görüntü", yüzü aldığımız görüntüdür ve "hedef görüntü", kaynak görüntüden çıkarılan yüzü koyduğumuz yerdir. Alınan yüz çiftlerine daha kolay işlem yapabilmek için ikisi de siyah-beyaz (Gray) görüntüye dönüştürülmüştür.

İkinci adımda Dlib kütüphanesini kullanılarak siyah-beyaz görüntülerdeki yüzler bulunmuştur. Dlib shape predictor kullanılarak Şekil 3.2’ de gösterilen yüzü tanımlayan 68 önemli nokta (face landmarks) kaynak görüntü ve hedef görüntüde bulunmuştur.



Şekil 3.2. Yüzü tanımlayan 68 önemli nokta

Daha sonra bu noktalar kullanılarak yüz bu noktaların bileşiminden oluşan üçgenlere (Delaunay triangulation) ayrılmaktadır. Bu işlem üçgenleme veya nirengi olarak adlandırılmaktadır. Bu adım metottaki yüz değişiminin temelini oluşturur. Çünkü hedef

görüntüdeki üçgenlere karşılık gelen kaynak görüntü üçgenler yer değiştirecektir. Yüzün üçgenlere ayrılmasındaki nedenler şunlardır: Farklı boyut ve perspektife sahip oldukları için kaynak görüntüden yüz kesilip aynı şekilde hedef görüntüye konulamaz. Ayrıca yüz orijinal orantılarını kaybedeceği için boyutu ve perspektifi hemen değiştirilemez. Bunun yerine, yüz üçgenlere ayrılırsa, her üçgen kolayca değiştirilebilir ve bu şekilde orantılarını korur; ayrıca gülümseme, göz kapatma ve ağzını açma gibi yeni yüzün ifadeleriyle eşleşir. Nirengi ya da üçgenleme, haritalarda uzunluk, yükseklik, koordinat gibi bilinmeyen değerlerin bulunması için bir alanın üçgenlere bölünerek bilinmeyenlerin hesaplanmasıdır.

Dördüncü adım olarak hedef görüntünün üçgenlemesinin, kaynak görüntünün üçgenlemesinin aynı modellerine sahip olması gerekir. Bu, noktaların bağlantısının aynı olması gerektiği anlamına gelir. Kaynak görüntünün üçgenlemesi yapıldıktan sonra, bu üçgenlemeden yüzü tanımlayan 68 dönüm noktasının dizinleri alınır, böylece hedef görüntüde aynı üçgenleme kopyalanabilir.

Beşinci adımda; her iki yüzün üçgenlerini elde ettikten sonra, kaynak yüzün üçgenleri alınır ve çıkarılır. Ayrıca, hedef yüzdeki eşleştirme üçgeni ile aynı boyut ve perspektife sahip olması için kaynak yüzün üçgenlerinin çarpıtılabilmesi için hedef yüzün üçgenlerinin koordinatları da alınmalıdır. Aşağıda Şekil 3.3' teki kod, kaynak görüntünün üçgenlerinin afin dönüşüm (Berger, 1987) kullanılarak nasıl çarpıtıldığını gösterir.

```
 # Warp triangles
 points = np.float32(points)
 points2 = np.float32(points2)
 M = cv2.getAffineTransform(points, points2)
 warped_triangle = cv2.warpAffine(cropped_triangle, M, (w,
h))
 warped_triangle = cv2.bitwise_and(warped_triangle,
warped_triangle, mask=cropped_tr2_mask)
```

Şekil 3.3. Üçgenleri çıkarma ve çarpıtma (Canu, 2019)

Altıncı adım olarak tüm üçgenler kesilip büküldüğünde, onların birbirine bağlaması gerekir. Bu sefer çarpık üçgenin konulmasındaki tek farkla, nirengi deseni kullanılarak yüz basitçe yeniden oluşturulur.

Yedinci adımda yüz artık değiştirilmeye hazırdır. Yeni yüze yer açmak için hedef görüntünün yüzü kesilir. Bu nedenle yeni yüz ve yüzsüz hedef fotoğraf alınıp birbirine bağlanır.

Sekizinci adımda son olarak, yüzler doğru bir şekilde değiştirilir ve kaynak görüntünün hedef görüntüye uyması için renkleri ayarlama zamanıdır. OpenCV (Bradski, 2000) kütüphanesinde bu işlemi otomatik olarak yapan "seamlessClone" adlı yerleşik bir işlev vardır. Yeni yüz (6. adımda oluşturulmuş) alınmalıdır, orijinal hedef resim alınmalı ve yüzü kesmek için maske olmalıdır. En son olarak da cilt tonu yüzün ortasından alınmalı ve son yüze uygulanmalıdır.

Bu işlemlerdeki adımlar görüntüler ve webcam üzerinde denenmiştir. Bu işlemlerle yapılmış bir yüz değiştirme aşağıda Şekil 3.4' te gösterilmiştir. Mona Lisa' nın yüzü alınarak oyuncu Taner Ölmez' in yüzüne aktarılmıştır (sağdaki görüntü).

Şekil 3.4. Basit bir yüz değiştirme işlemi

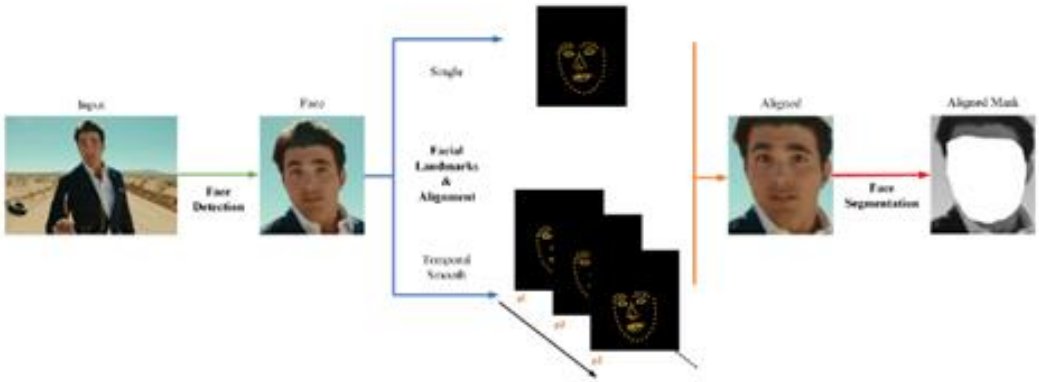
3.2. Otomatik Kodlayıcı Kullanarak Yüz Değiştirme

Bu metotta kullanılan DeepFaceLab (Perov vd., 2020), Ivan Perov ve arkadaşları tarafından oluşturulan açık kaynaklı bir deepfake sistemidir. Deepfakes (t.y.) modeli genel hattıyla yüz hizalama modülü, yüz ayrıştırma modülü, yüz harmanlama modülünden oluşur. Bu şekilde fotogerçekçi yüz değiştirme sonuçları elde etmeyi sağlar.

Deepfakes (t.y.) modeli üç ana aşamadan oluşur: Yüzleri bularak çıkarma ve biriktirme aşaması, yüz çiftlerinin birbiriyle karşılıklı değiştirilecek şekilde eğitilmesi aşaması ve yüzleri eğitim sonuçlarını kullanarak dönüştürme aşaması. Yüzleri çıkarma aşaması da yüz algılama, yüz hizalama ve yüz bölümlemeyi içerir. Bu modelde ilk olarak yüz çıkarma işlemi gerçekleştirilir. Bilindiği gibi video, görüntülerden (frame) oluşur. Bilgisayar oyunlarında da

bilindiği gibi daha gerçekçi görüntü için saniyedeki görüntü sayısı (fps: frame per second) fazla olmalıdır. Modelin yüz çıkarma işleminde saniyede çıkarılması istenen sayıda görüntü çıkarılır ve görüntüdeki yüzler algılanır.

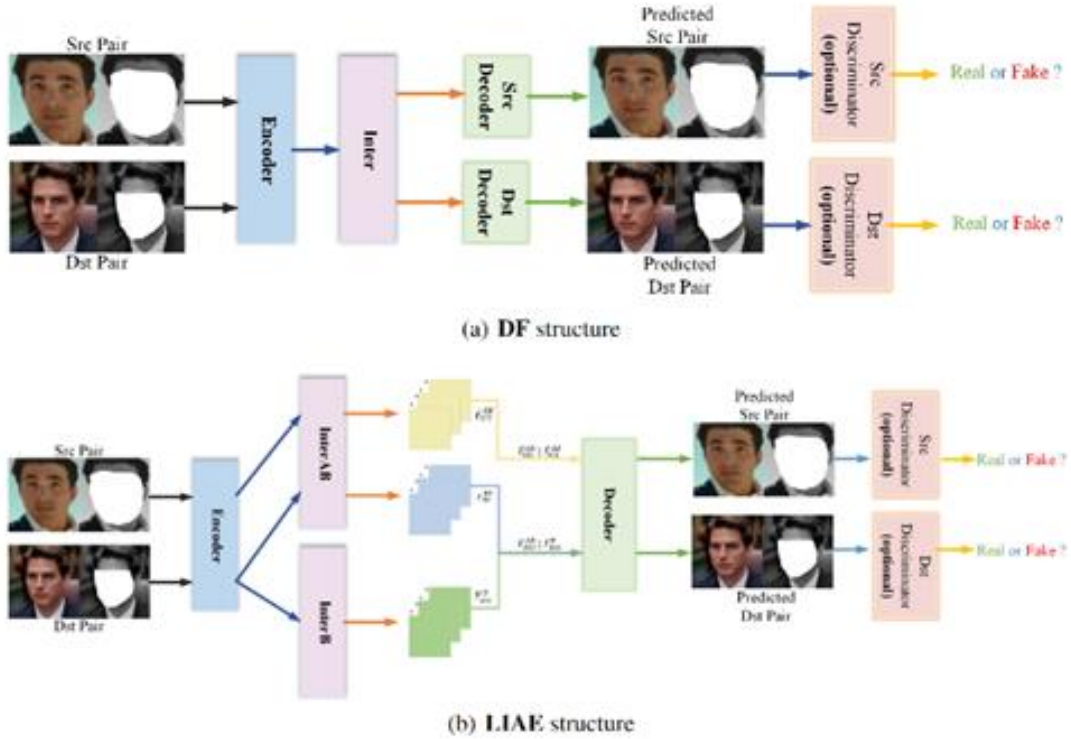
Yüz çıkarma işleminin ilk adımı yüz algılamanın amacı, verilen klasörlerde hedef yüzü bulmaktır. Belirtilen hedef için S3FD (S. Zhang vd., 2017), RetinaFace (Deng vd., 2019), MTCNN (K. Zhang, Z. Zhang, Li ve Qiao, 2016) yüz algılama algoritmalarından herhangi birisi kullanılabilir. Yüz çıkarma işlemi Şekil 3.5 ile gösterilmektedir.



Şekil 3.5. Deepfakes modelinde yüz çıkarımına genel bakış (Perov vd., 2020)

İkinci adım olarak yüz hizalama, yüz yer işaretleri algoritması ile yüzlerin hizalamasıdır. Bunu çözmek için çıkarma algoritması: (a) ısı haritası tabanlı yüz çevreleme algoritması 2DFAN (Bulat ve Tzimiropoulos, 2017) (normal duruşa sahip yüzler için) ve (b) PRNet (Feng, Wu, Shao, Wang ve Zhou, 2018) 3D yüz öncül bilgileri (geniş Euler açılı yüz için (sapma, eğim, rulo), örneğin, geniş açılı bir yüz, yüzün bir tarafının görüş alanı dışında olduğu anlamına gelir) kullanılabilir.

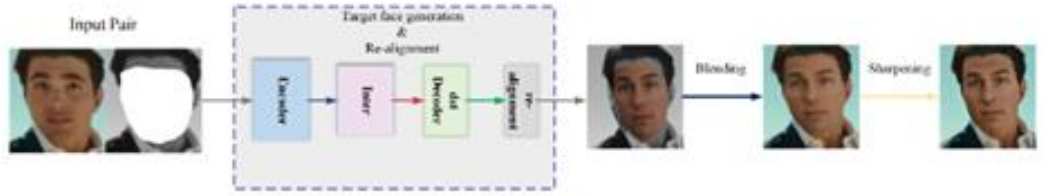
Yüz segmentasyonunda yüz hizalamadan sonra elde edilen, standart ön / yandan görünüm hizalı kesite sahip veri klasörü kullanılır. Hizalanmış klasörün üstünde bir Yüz Segmentasyon ağı (TernausNet (Iglovikov ve Shvets, 2018)) kullanılır; bu ağ üzerinden saç, parmak veya gözlük içeren bir yüz tam olarak bölümlere ayrılabilir. Eğitim sürecinde ağı ellere, gözlüklere ve yüzleri bir şekilde kaplayabilecek diğer nesnelere karşı sağlam tutmak için düzensiz tıkanıklıkları gidermek için tasarlanmış, isteğe bağlı olarak ancak yararlıdır.



Şekil 3.6. Deepfakes modelinde eğitim aşamasına genel bakış (Perov vd., 2020)

Şekil 3.6' da gösterildiği gibi eğitim aşamasında otomatik kodlayıcıdaki gizli uzay vektörleri birleştirilmektedir. İstenirse otomatik kodlayıcı ağ mimarisine ayırt edici ağ eklenerek Deepfakes (t.y.) modeli için ÇÜA kullanılabilir.

Eğitim aşaması Deepfakes (t.y.) modelinin fotogerçekçi yüz değiştirme sonuçlarına ulaşmada en önemli rol oynamaktadır.



Şekil 3.7. Deepfakes modelinde yüz dönüştürme aşamasına genel bakış (Perov vd., 2020)

Son olarak yüz dönüştürme aşamasında Şekil 3.7' de görüldüğü gibi eğitim sonuçları kullanılarak çeşitli işlemlerle (harmanlama, bileme vb.) yüz diğer bir yüze dönüştürülmektedir. Burada eğitim sonuçları karşılıklı kullanılarak Umeyama' nın (Umeyama, 1991) tersine çevrilebilirliği kullanılarak ters dönüştürme de

gerçekleştirilebilir. Örneğin; A yüzü B yüzüne dönüştürülebileceği gibi, B yüzü de A yüzüne dönüştürülebilir. Bu aşamada çeşitli renk aktarım algoritmaları kullanılır (Reinhard renk aktarımı: RCT (Reinhard, Ashikhmin, Gooch ve Shirley, 2001), yinelemeli dağıtım aktarımı: IDT (Pitié, Kokaram ve Dahyot, 2007) ve vb.). Bunun da ötesinde, harmanlamanın sonucu iki görüntü birleştirilerek elde edilebilir. Herhangi bir karışım, özellikle sınırlandırılmış bölge ile farklı cilt tonları, yüz şekilleri ve aydınlatma koşulları arasındaki bağlantılarda hesaba katılmalıdır. Burada Poisson-harmanlama (Pérez, Gangnet ve Blake, 2003) optimizasyonu kullanılır. Ardından, son işlem olan, bileme (keskinleştirme) işlemi gerçekleştirilir. Karıştırılmış yüzü keskinleştirmek için önceden eğitilmiş bir süper çözünürlüklü yüz sinir ağı (FaceEnhancer olarak ifade edilir) kullanılır. Sahte görüntüyü, oluşturulan yüzü sorunsuz bir şekilde hedef yüzün tasarlanmış kısmına yerleştirir ve bu arada oluşturulan yüzün cilt tonunu hedef yüze göre ayarlar, ardından Yüz çıkarma aşamasında kaydedilen koordinatlarına göre orijinal resme geri yerleştirir.

Yüz dönüştürme aşaması için hedef imaj I_t, Mt maskesinden oluşturulan yüz I_t^r, olarak tanımlanırsa; Maske M_t' nin dış çevresi boyunca hedef görüntü ile kusursuz bir şekilde uyum sağlar. Kalan kısımları hedef imaj I_t' e daha fazla uyarlanabilir hale getirmek için beş renk aktarım algoritması daha kullanılır. Üstelik, I_t^r ve I_t olmak üzere iki görüntü birleştirilerek harmanlamanın sonucu Denklem 3.1'deki gibi elde edilebilir.

$$\mathrm{Ioutput} = \mathrm{Mt} \odot \mathrm{I_t^r} + (1-\mathrm{Mt}) \odot \mathrm{It} \tag{3.1}$$

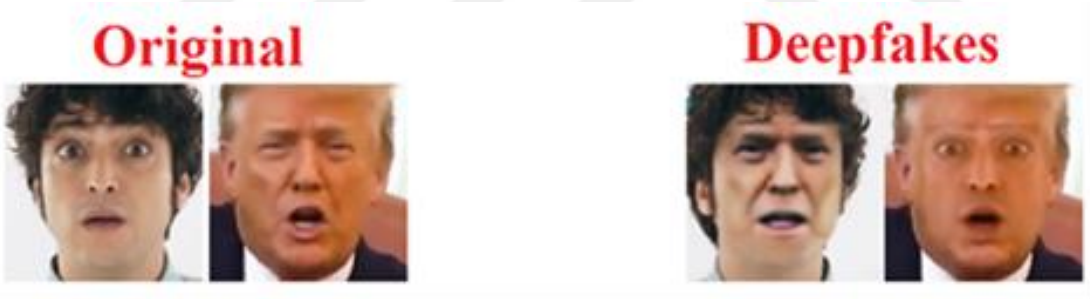
Herhangi bir harmanlama işleminde, özellikle sınırlandırılmış bölge ile I_t^r ve I_t arasındaki bağlantılarda, farklı cilt tonları, yüz şekilleri ve aydınlatma koşullarını hesaba katılmalıdır. Burada Poisson harmanlama (Pérez vd., 2003) optimizasyonu Denklem 3.2' deki gibi şu şekilde tanımlanır:

$$\mathrm{P(It;\, I_t^r; Mt)} = \begin{cases} ||\nabla \mathrm{It(i,j)} - \nabla \mathrm{I_t^r(i,j)}||_2^2, & \forall \mathrm{Mt(i,j)} = 0 \\ \min||\nabla \mathrm{f(i,j)} - \nabla \mathrm{I_t^r(i,j)}||_2^2, & \forall \mathrm{Mt(i,j)} = 1 \end{cases} \tag{3.2}$$

Çeşitli kısa videolar alınarak Deepfakes modelinde GTX 1050 GPU ile Windows ortamında model eğitilerek yüzler birbiriyle değiştirilmiştir. Örneğin, Şekil 3.8' deki çıktılar için 79 ve 195 saniyelik iki video kullanılmıştır.

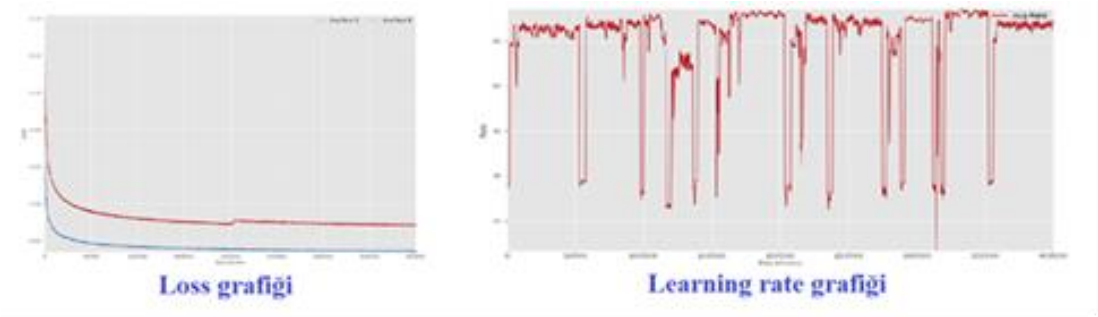
Videolardan frame' ler çıkarılarak Yüz tanıma kısmında bu frame' lerde yüzler S3FD (S. Zhang vd., 2017) yüz tanıma kütüphanesi kullanılarak bulunmuştur. Bulunan yüzler 2DFAN (Bulat ve Tzimiropoulos, 2017) yüz çıkarma kütüphanesi kullanılarak hizalanmış ve yüz segmentasyonu çıkarılmıştır. Deepfakes modeli kullanılarak elde edilen verilerle iki yüz birbiriyle değişecek şekilde 12 batch (yığın) olarak yaklaşık 54 saat 400.000 iterasyon eğitilmiştir. Eğitilen modelle videolar yüzler değiştirilerek yeniden oluşturulmuştur. Oluşturulan videodaki örnek frame' ler Şekil 3.8'de sağda gösterilmiştir.

Bu görüntülerde yüzün hangi kısımlardan itibaren değiştirildiğini daha net görebilmek için yüzü ve kafası birbirine benzemeyen iki kişi tercih edilmiştir. Örneğin birinin saçı siyah iken diğerinin sarı, ten renkleri farklı ve yüz özellikleri farklıdır Normalde birini kandırmak amaçla veya kötü bir niyetle yapılan yüz değiştirmede yüz ve kafa özellikleri benzeyen kişiler tercih edilebilir.



Şekil 3.8. Deepfakes (t.y.) modeli kullanarak yüzlerin eğitilerek birbiriyle değiştirilmesi

Learning rate ve loss değerleri grafikleri oluşturulmuştur. Burada 200.000 iterasyondan sonra bazı görüntülerin yapay zekayı iyi eğitemediği gözlendiğinden bu veriler çıkarılmıştır ve Şekil 3.9' daki gibi loss değerinde anlık bir artış olmuştur ve sonra loss değeri azalmaya devam etmiştir. Buradaki loss değerlerini hesaplamak için Denklem 1.2 kullanılabilir.



Şekil 3.9. Loss ve learning rate grafikleri

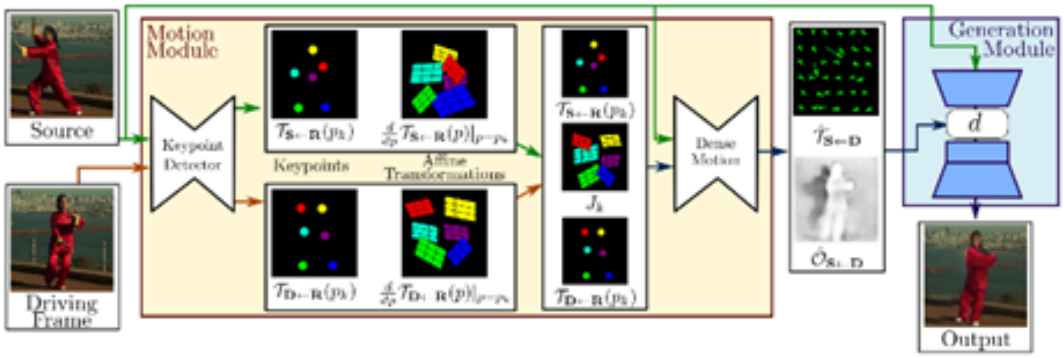
3.3. Çekişmeli Üretici Ağ Kullanarak Yüz Canlandırma

Bu metotta kullanılan First Order Motion Model (Siarohin vd., 2020a) ile görüntü canlandırma, bir kaynak görüntüdeki bir nesnenin sürüş videosunun hareketine göre canlandırılması için bir video sekansı oluşturmayı içerir. Bu modelde, canlandırılacak belirli nesne ön bilgi kullanmadan bu sorunu ele alır. Aynı kategorideki nesneleri (ör. Yüzler, insan vücutları) gösteren bir dizi video üzerinde eğitildikten sonra, yöntem bu sınıftaki herhangi bir nesneye uygulanabilir. Bunu başarmak için, kendi kendini denetleyen bir formülasyon kullanarak görünüm ve hareket bilgileri ayrılmaktadır. Karmaşık hareketleri desteklemek için, yerel afin dönüşümleriyle birlikte bir dizi öğrenilmiş anahtar noktadan oluşan bir temsil kullanılmaktadır. Bir üretici ağı, hedef hareketleri sırasında ortaya çıkan tıkanıklıkları modeller ve kaynak görüntüden çıkarılan görünümü ve sürüş videosundan türetilen hareketi birleştirir.

Şekil 3.10’ da görüldüğü gibi çerçeve iki ana modülden oluşmaktadır: hareket tahmin modülü ve görüntü oluşturma modülü. Hareket tahmin modülünün amacı, yoğun bir hareket alanını tahmin etmektir. Soyut bir referans çerçevesi olduğunu varsayılmaktadır ve bağımsız olarak iki dönüşüm tahmin edilmektedir: referanstan kaynağa ve referanstan sürüşe. Bu seçim, kaynağı ve sürüş çerçevelerini bağımsız olarak işlememeye olanak tanır. Bu, test zamanında model, görsel olarak çok farklı olabilen farklı bir videodan örneklenmiş kaynak görüntü ve sürüş çerçevelerinin çiftlerini aldığından istenir.

İlk adımda, kendi kendini denetleyen bir şekilde öğrenilen anahtar noktaları kullanarak elde edilen seyrek yörünge setlerinden her iki dönüşümü de tahmin edilmektedir. Yerel afin dönüşümleri kullanarak her bir kilit noktanın komşuluğundaki hareketi modellenir. Yalnızca ana nokta yer değiştirmelerini kullanmakla karşılaştırıldığında, yerel afin dönüşümler, daha büyük bir dönüşüm ailesini modellemeye izin verir. İkinci adım sırasında, yoğun hareket ağı,

ortaya çıkan yoğun hareket alanını elde etmek için yerel yaklaşımları birleştirir. Ayrıca, yoğun hareket alanına ek olarak, bu ağ, kaynak görüntünün eğilmesi ile hangi görüntü parçalarının yeniden yapılandırılabileceğini ve hangi parçaların boyanması gerektiğini (bağlamdan çıkarılır) gösteren bir kapatma maskesi çıkarır. Son olarak, oluşturma modülü, sürücü videoda sağlandığı gibi hareket eden kaynak nesnenin bir görüntüsünü işler. Burada, kaynak görüntüyü yoğun harekete göre büken ve kaynak görüntüde tıkanan görüntü parçalarını boyayan bir generator ağı kullanılır.



Şekil 3.10. First order motion model ’ in çekişmeli üretici ağ yapısı (Siarohin vd., 2020a)

Geometride Afin dönüşüm (Berger, 1987), afin uzayların aralarındaki noktalar, düz çizgiler ve düzlemler için oranların korumasını sağlayan eşlemedir. Ayrıca, paralel olan çizgi kümeleri afin dönüşümün sonrasında paralel olarak kalır. Afin dönüşümde aynı doğru üzerindeki noktaların aralarındaki mesafelerin oranları sabit kalmasına rağmen, çizgilerin aralarındaki açı ile noktaların arasında bulunan mesafeler sabit kalmayabilir.

Bu modelde kilit noktalara ve yerel afin dönüşümlere dayalı yeni bir görüntü animasyonu yaklaşımı kullanılır. Yeni matematiksel formülasyon, iki çerçeve arasındaki hareket alanını tanımlar ve birinci dereceden bir Taylor genişleme yaklaşımı türetilerek verimli bir şekilde hesaplanır. Bu şekilde hareket, bir dizi anahtar nokta yer değiştirmeleri ve yerel afin dönüşümler olarak tanımlanır.

Hareket tahmin modülü, sürücü çerçeve D'den kaynak çerçeve S'ye geriye doğru optik akışı $T_{S\leftarrow D}$ tahmin eder. Anahtar nokta konumlarının bir komşuluğundaki birinci dereceden Taylor genişlemesi ile $T_{S\leftarrow D}$' yi yaklaşık olarak tahmin etmek modelde önerilir. Soyut bir referans çerçevesi R olduğunu varsayılır. Bu nedenle, $T_{S\leftarrow D}$ tahmini $T_{S\leftarrow R}$ ve $T_{R\leftarrow D}$ tahminini içerir. Ayrıca, bir X çerçevesi verildiğinde, öğrenilen anahtar noktaların

komşuluğundaki her bir $T_{X\leftarrow R}$ dönüşümü tahmin edilmektedir. Biçimsel olarak, bir $T_{X\leftarrow R}$ dönüşümü verildiğinde, K anahtar noktalarındaki p_1, ... p_K birinci dereceden Taylor açılımları dikkate alınır. Burada, p_1, ... p_K, referans çerçevesi R'deki anahtar noktaların koordinatlarını belirtir. Her yerel dönüşümün nerede tuttuğunu gösteren K + 1 maskeleri Mk, k = 0, ... K tahmin edilmektedir. Monkey-Net (Siarohin vd., 2019) ve U-Net (Ronneberger, Fischer ve Brox, 2015) altyapısı kullanılarak oluşturulan nihai yoğun hareket tahmini $\check{T}_{S\leftarrow D}(z)$ Denklem 3.3' teki gibi verilmektedir.

$$\check{T}S\leftarrow D(z) = M0z + \sum_{k=1}^{K} Mk\ (TS\leftarrow R(pk) + Jk(z - TD\leftarrow R(pk))) \tag{3.3}$$

Sistem çeşitli loss değerleri birleştirilerek uçtan uca bir şekilde eğitilir. İlk olarak, Johnson ve arkadaşlarının (Johnson, Alahi ve Fei-Fei, 2016) perceptual loss hesabına dayanan rekonstrüksiyon loss hesabı kullanılmaktadır. Önceden eğitilmiş VGG-19 ağını ana loss hesabı Wang vd. (Wang vd., 2018)' nin çalışmalarına dayanmaktadır. Giriş sürüş çerçevesi D ve karşılık gelen yeniden yapılandırılmış çerçeve $\check{D}$ ile gösterildiğinde yeniden yapılandırma loss hesabı Denklem 3.4' deki gibi yapılır:

$$L_{rec}(\check{D}, D) = \sum_{i=1}^{I} |N_i(\check{D}) - N_i(D)| \tag{3.4}$$

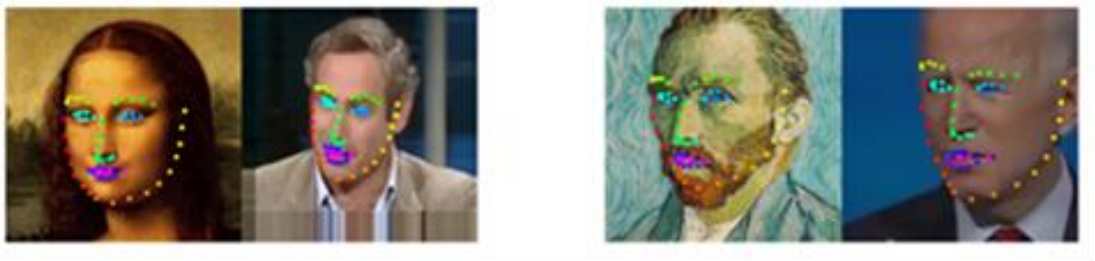
Bağıl hareket transferi aşamasında amaç, sürüş videosu D_1, ... D_T' yi kullanarak bir S_1 kaynak karesindeki bir nesneyi canlandırmaktır. Her kare Dt, St.' yi elde etmek için bağımsız olarak işlenir. $T_{S1\leftarrow Dt}(p_k)$ ' de kodlanmış hareket S'ye aktarılmak yerine, göreceli hareket D_1 ve Dt arasında S_1' e aktarılmaktadır. Başka bir deyişle, her p_k anahtar noktasının komşuluğuna bir $T_{Dt\leftarrow D1}(p)$ dönüşümü uygulanarak Denklem 3.5 ve Denklem 3.6 elde edilmektedir.

$$T_{S1\leftarrow St}(z) \approx T_{S1\leftarrow R}(p_k) + J_k(z - T_{S\leftarrow R}(p_k) + T_{D1\leftarrow R}(p_k) - T_{Dt\leftarrow R}(p_k)) \tag{3.5}$$

$$J_k = \left(\frac{d}{dp} T_{D1\leftarrow R}(p)|_{p=pk}\right) \left(\frac{d}{dp} T_{Dt\leftarrow R}(p)|_{p=pk}\right)^{-1} \tag{3.6}$$

VoxCeleb (Nagrani vd., 2017) veri setinin First Order Motion Model (Siarohin vd., 2020a) üzerinde eğitilmiş ağırlıkları kullanılarak çeşitli tablolar, portreler, resimler (elimizde bir pozu bulunan resimler); alınan başka videolardaki yüz hareketleri kullanılarak Google Colab ortamında Tesla P4 GPU ile ve Windows ortamında GTX 1050 GPU ile canlandırılmıştır. Bu işlemler alınan görüntüler ve webcam üzerinde denenmiştir. Bu modelde kullanılan kaynak görüntüler yüz eşleşmelerinde solda ve driving video (hareketleri kaynak görüntüye aktarılacak videolar) sağda gösterilmiştir. Şekil 3.11' de de gösterildiği gibi sol taraflarda bulunan Mona Lisa ve Van Gogh tabloları, hemen sağlarında bulunan kişilerin yüz hareketleri transfer edilerek canlandırılmıştır. Burada yüz çiftlerinin en uygun yüz hizası eşleşmeleri gösterilmektedir.

Buradaki eşleşen noktalarının hareketine göre tablolar üretken modellerle yeniden oluşturularak hareketlendirilmektedir. Aynı zamanda burada wav2lip modeli (Prajwal, Mukhopadhyay, Namboodiri ve Jawahar, 2020) ile arka plandaki konuşmaya göre (söylenen sözcüklere göre) dudak hareketleri de verilmiştir. Şekil 16' daki görseldeki çıktılar Tesla P4 GPU kullanılarak elde edilmiştir.



Şekil 3.11. Tabloların canlandırılması

3.4. Çekişmeli Üretici Ağ Kullanarak Yüz Oluşturma

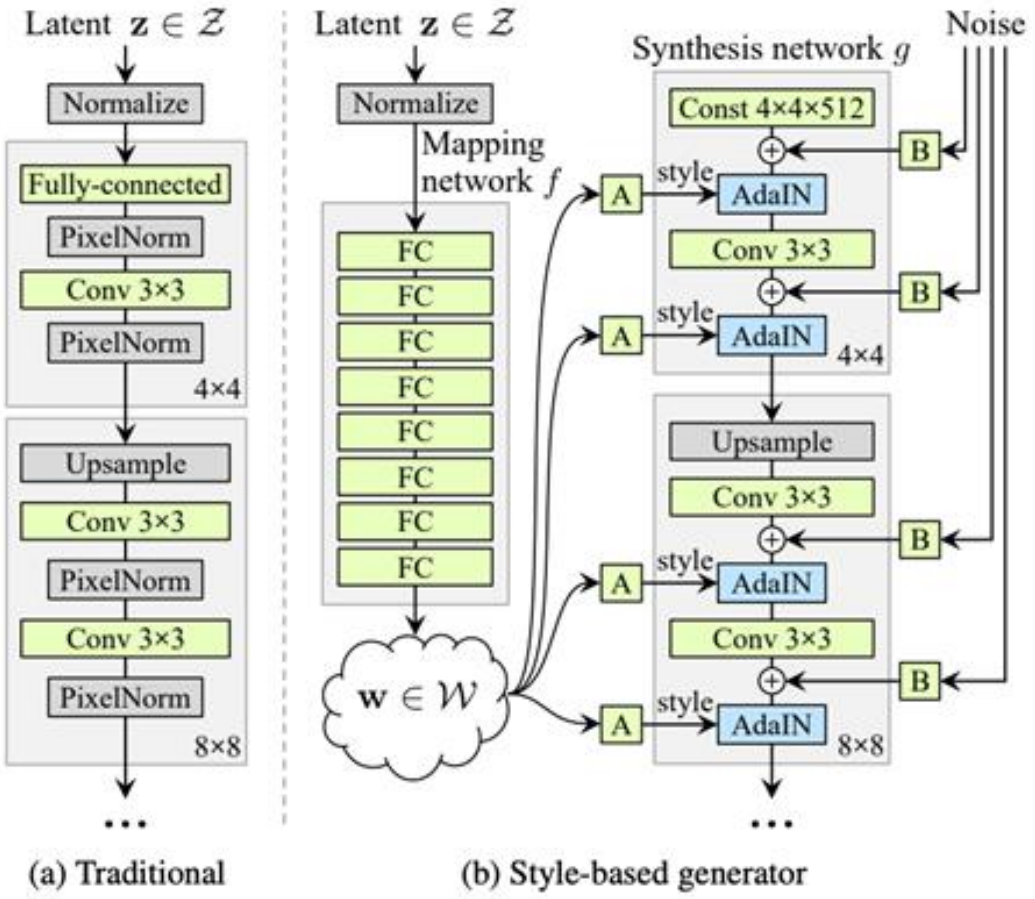
Bu metotta kullanılan StyleGAN (Karras vd., 2019a), bir tür çekişmeli üretici ağdır. Stil transferi (Gatys, Ecker ve Bethge, 2015) literatüründen alıntı yaparak, özellikle, uyarlanabilir örnek normalizasyonunun kullanımı ile ÇÜA için Şekil 3.12' deki gibi alternatif bir üreteç mimarisi kullanır. Aksi takdirde, giderek artan bir eğitim rejimi kullanırken Aşamalı GAN (Progressive GAN) (Karras vd., 2017)' i izler. Diğer farklılıklar arasında, normal GAN' lerde (Goodfellow vd., 2014) olduğu gibi stokastik olarak oluşturulmuş gizli değişkenler olmayan sabit değerli bir tensörden üretildiği gerçeği yer alır. Stokastik olarak üretilen gizli değişkenler, 8 katmanlı ileri besleme ağı tarafından dönüştürüldükten sonra her çözünürlükte uyarlanabilir

örnek normalizasyonunda stil vektörleri olarak kullanılır. Son olarak, eğitim sırasında iki stil gizli değişkeni karıştıran karıştırma düzenlileştirme adı verilen bir düzenleme biçimi kullanır.

Stil tabanlı generator, geleneksel generator' den farklı olarak öğrenilmiş bir sabitten işleme başlar. Her bir evrişim katmanında uyarlanabilir örnek normalizasyonu (AdaIN) aracılığıyla generator kontrol edilir. Doğrusal olmama durumu değerlendirilmeden önce, her evrişimden sonra Gauss gürültüsü eklenir. Burada "A", öğrenilmiş afin dönüşümü ifade eder ve "B", gürültü girişine kanal başına öğrenilen ölçeklendirme faktörlerini uygular. Afin dönüşümleri öğrenilen w, sentez ağının her bir evrişim katmanından sonra uyarlanabilir örnek normalizasyonu (Adaptive Instance Normalization) (AdaIN) (Huang ve Belongie, 2017; Dumoulin, Shlens ve Kudlur, 2016; Ghiasi, Lee, Kudlur, Dumoulin ve Shlens, 2017; Dumoulin vd., 2018) işlemlerini kontrol eden y = (ys, yb) stillerine özelleşir AdaIN işlemi Denklem 3.7' deki gibi tanımlanır.

$$\mathrm{AdaIN}(x_i, y) = y_{s,i} \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}, \tag{3.7}$$

Burada her özellik haritası xi ayrı normalleştirilir ve ardından y stilindeki karşılık gelen skaler bileşenler kullanılarak ölçeklenir.



Şekil 3.12. Geleneksel üretici ağ (a) ve stil tabanlı üretici ağ (b)' ın karşılaştırılması (Karras vd., 2019a)

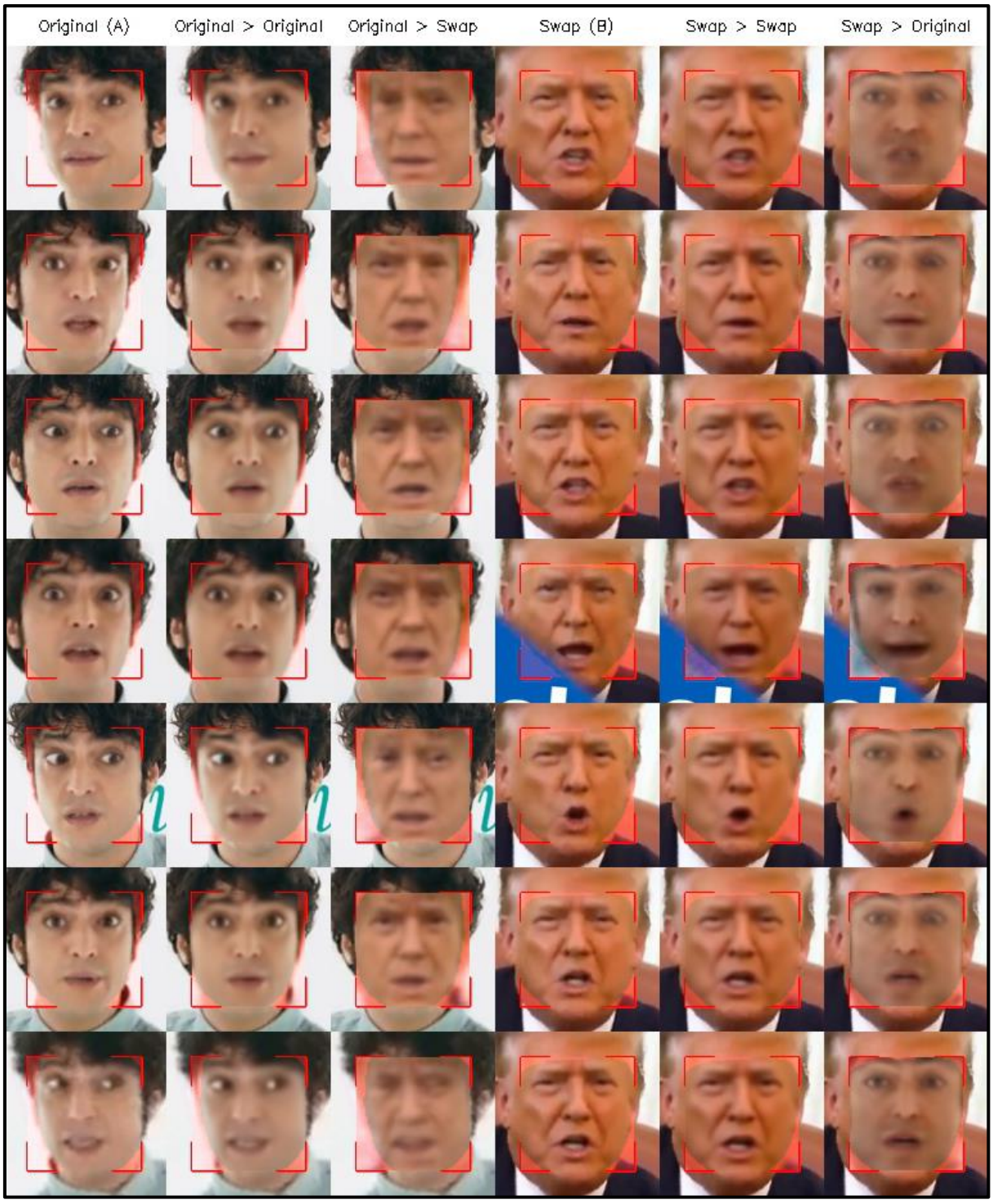
Sinirsel Stil Transferi (Gatys vd., 2015), bir içerik imajı ve bir stil imajını girdi olarak alma ve içerik imajının içeriğine ve stil imajının stiline sahip bir animasyon çıktılama problemidir. Nöral stil aktarımını mümkün kılan anahtar teknik, evrişimli sinir ağlarıdır (CNN) (Krizhevsky, Sutskever ve Hinton, 2012).

StyleGan (Karras vd., 2019a) üzerinde CelebA-HQ (Karras vd., 2017) veri seti ağırlıkları kullanılarak önceden eğitilmiş modelinde Google Colab ortamında Tesla P4 GPU ile birkaç yüz görüntüsü oluşturma işlemi denenmiştir. Bu model ile üretilen birkaç görüntü aşağıda Şekil 3.13'te gösterilmiştir.

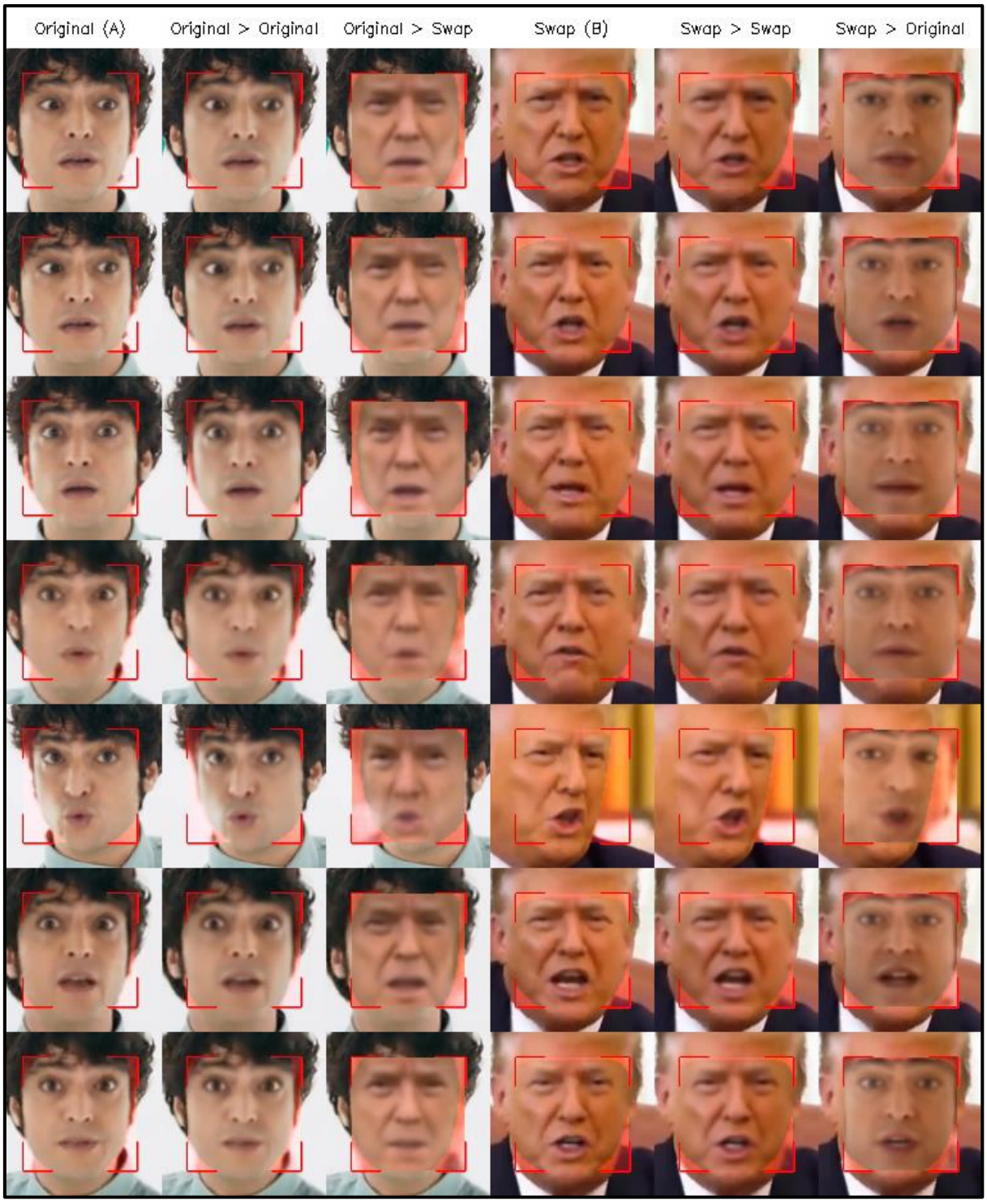
Şekil 3.13. Gerçekte var olmayan insan yüz resimleri üretimi

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

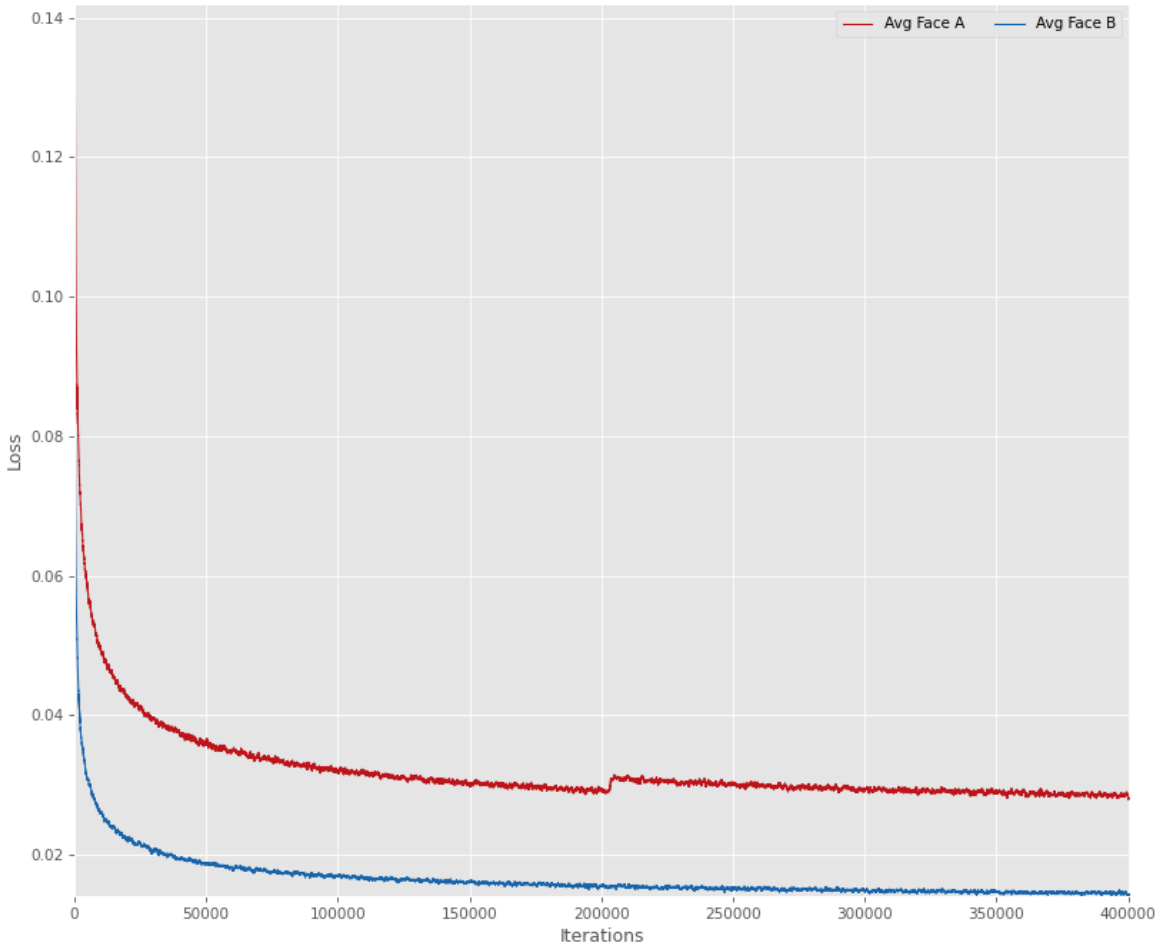
Yapılan çalışma; kullanılan video ve fotoğraflarda yüzün karşıya (öne) dönük veya hafif sağa ya da hafif sola dönük iken, yüz hareketinin belirli bir alanda sınırlı olduğunda ve yüzün yavaş hareket ettiğinde yapay zekayı daha iyi eğittiği ve bu eğitim verileri kullanılarak oluşturulan sahte videoların daha başarılı olduğunu göstermiştir. Aynı zamanda çalışma, amaca uygun olarak çeşitli yöntem ve modellerle bir resim, birkaç resim ya da kısa bir videodan o kişiye ait sahte videolar oluşturulabileceğini göstermektedir. Deepfakes (t.y.) modeli için Şekil 4.1' de görüldüğü gibi videolardan çıkarılan bazı görüntülerin yapay zekayı iyi eğitemediği, bulanık bir yüzün çıkmasına neden olduğu görülmüştür. Bu görüntüler çıkarılarak ayı süre ile eğitim devam ettirildiğinde Şekil 4.2' deki gibi daha başarılı çıktılar elde edilmiştir. Kalan görüntülerin yapay zekayı daha iyi eğittiği görülmüştür. Bu şekillerde kişilerin orijinal görüntüleri kişilerin solunda, yüz değiştirme uygulanmış görüntüler ise kişileri sağında bulunmaktadır. Hesaplanan loss değerleri Şekil 4.3' te gösterilmiştir.



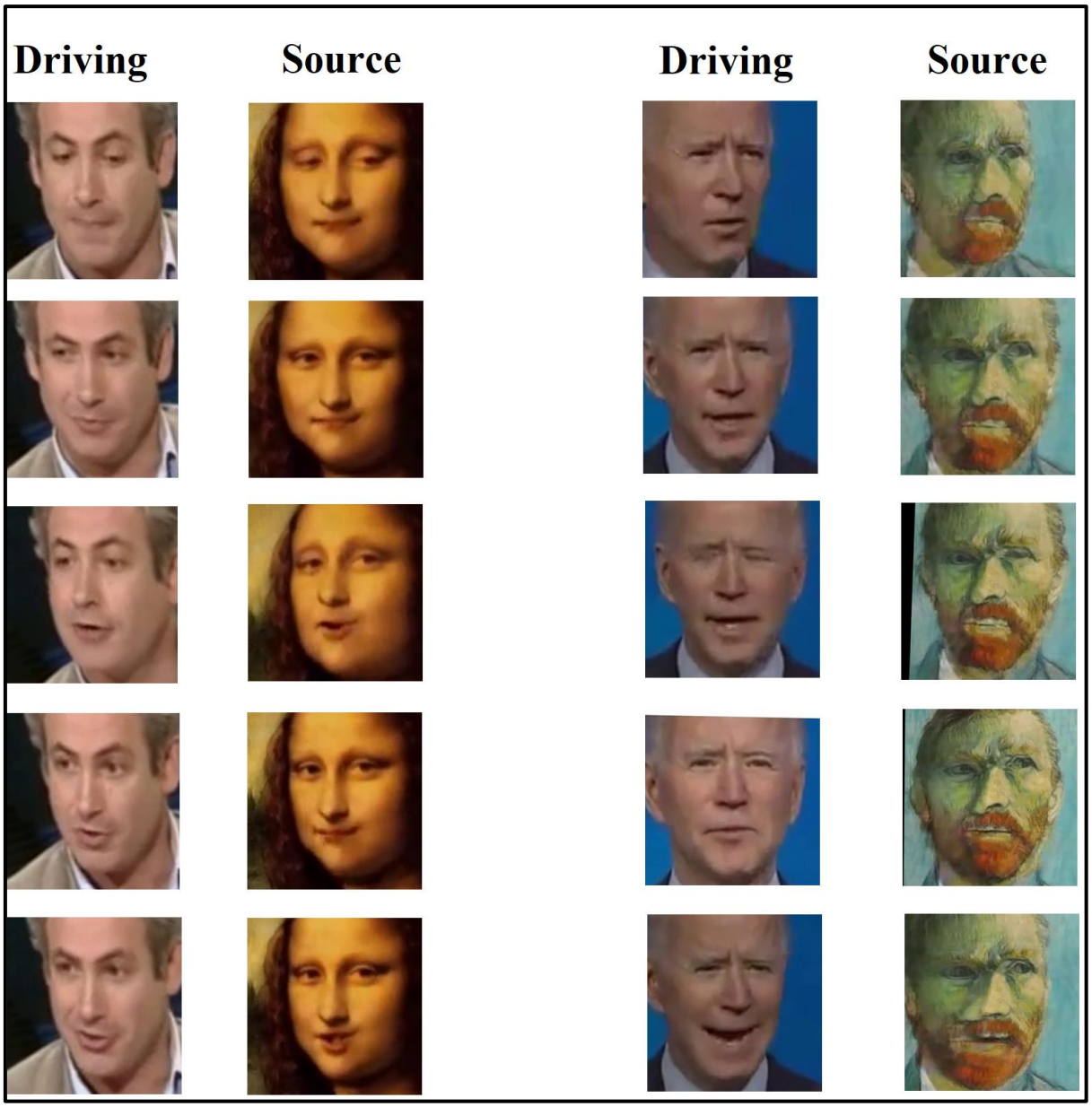
Şekil 4.1. Deepfakes modeli için yaklaşık 200.000 iterasyondaki eğitim çıktıları (Deepfakes, t.y.)



Şekil 4.2. Deepfakes modeli için 400.000 iterasyondaki eğitim çıktıları (Deepfakes, t.y.)



Şekil 4.3. Deepfakes modeli için loss değerleri grafiği (Deepfakes, t.y.)



Şekil 4.4. First Order Motion Model ile tabloların bir videonun hareketi kullanılarak canlandırılması (Siarohin vd., 2020a)

First Order Motion Model (Siarohin vd., 2020a) ile hareketin alınacağı bir video (driving) kullanılarak kaynak (source) tablolar canlandırılmış ve oluşturulan videolardan bazı görüntüler Şekil 4.4' te gösterilmiştir. Ayrıca Mona Lisa tablosuna arka plandaki konuşmayı dudak hareketlerine çeviren Wav2lip (Prajwal vd., 2020) modeli uygulanmış ve Van Gogh tablosunda uygulanmamıştır.

5. SONUÇ VE ÖNERİLER

Çizelge 5.1. Deepfakes modeli için hesaplanan loss değerleri (Deepfakes, t.y.)

Deepfakes modeli	**200.000 iterasyon loss değeri**	**400.000 iterasyon loss değeri**
Avg Face A	0,0301418	0,0288525
Avg Face B	0.0160740	0,0146704

Çizelge 5.1' de Deepfakes (t.y.) modelinin A kişisi ve B kişisi için hesaplanan ortalama loss değerleri gösterilmiştir.

Çizelge 5.2. First Order Motion Modeli için hesaplanan loss değeri (Siarohin vd., 2020a)

First Order Motion Modeli	**VoxCeleb veri seti için**
Hesaplanan loss değeri	0,0431123

Çizelge 5.2' de First Order Motion Modeli (Siarohin vd., 2020a) için hesaplanan loss değeri gösterilmiştir.

Sahte videoların kötü niyetin haricinde çok çeşitli amaçlarla kullanabileceği görülmüştür. Bu durumda sahte video oluşturma teknolojisinin daha başlangıç aşamasında bu kadar geliştiği bir dönemde güvenilmeyen insanların görebileceği, bulabileceği bir şekilde video ve resim paylaşmaktan kaçınılmalıdır.

Deepfake videoları henüz ipuçlarıyla kendi başına keşfedilebilecek bir aşamadadır. Deepfake videolarda aşağıdaki özellikler ile videonun sahteliği anlaşılabilir:

- Titreşim hareketleri
- Kareler arası akışta ışıksal farklılıklar
- Cilt rengi değişimleri
- Göz hareketlerinde gariplik veya gözün kırpılmaması
- Dudakların konuşma ile uyumsuz olması
- Görüntüdeki yapay dijital unsurlar

Fakat Deepfake teknolojisi geliştikçe videolardaki sahteliğin insan gözü ile ayırt edilmesinde zorluklar olacağından gelişmiş siber güvenlik programlarından yardım alınmaya ihtiyaç duyulacaktır.

Ayrıca Deepfake videolar Deepfake tespit algoritmaları kullanılarak da belirlenebilir.

Videolarının gerçekliğini doğrulayabilmek için şifreleme algoritmaları, videoya belirli aralıklarla hash değeri eklemekte faydalanılabilir. Video değiştirilince hash değeri de farklılaşacaktır. Yapay zeka ile blokzincir, videoya değiştirilmesi imkansız sayısal bir iz ekleyebilir.

Deepfake girişimlerini engellemek için çözüm olarak, yüz tespit yazılımlarının yararlandığı piksel düzenlerinin gizlenmesi amacıyla videoya özel dijital "yapay unsurlar" ekleyebilecek bir program kullanılabilir. Bu unsurlar, algoritmaların hızını azaltabilir ve kalitesi düşük sonuçlar ortaya çıkabilir. Bu sayede Deepfake girişimlerinin başarılı olma olasılığı düşürülür.

Başka bir çözüm olarak da Deepfake tespit algoritmaları kullanılabilir. Örneğin yüzün değiştirildiği bazı modellerde sobel filtresi yardımıyla maske bölgesel konvolüsyonel sinir ağı (Mask R-CNN) ile fotomontaj tespit algoritması kullanılabilir (Özmen ve Buluş, 2020).

Ayrıca kaliteli haber kaynaklarını kullanmalı ve emin olmadan hiçbir içeriğe güvenmemeliyiz.

Daha sonraki çalışmalarda yüz tanımanın alt alanları olan bu teknolojiler (yüz değiştirme, yüz hareketlendirme) kullanılarak iki özelliği de uygulayabilen, otomatik kodlayıcı veya çekişmeli üretici ağ tabanlı bir üretici ağın geliştirilmesi tasarlanmaktadır.

KAYNAKLAR

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M. vd. (2016). TensorFlow: A System for Large-Scale Machine Learning. OSDI (p./pp. 265--283).

Bank, D., Koenigstein, N. ve Giryes, R. (2020). Autoencoders. CoRR, abs/2003.05991.

Berger, Marcel. (1987). Geometry I. Berlin: Springer, ISBN 3-540-11658-3.

Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools, 120; 122-125.

Bulat, A. ve Tzimiropoulos, G. (2017). How Far are We from Solving the 2D & 3D Face Alignment Problem? (and a Dataset of 230, 000 3D Facial Landmarks). ICCV (p./pp. 1021-1030), : IEEE Computer Society. ISBN: 978-1-5386-1032-9.

Canu, S. (2019). Face swapping (explained in 8 steps) - Opencv with Python. Pysource. 11 Mayıs 2021, Erişim adresi https://pysource.com/2019/05/28/face-swapping-explained-in-8-steps-opencv-with-python/

Deepfakes (t.y..). Deepfakes/faceswap. 10 Mart 2021, Erişim adresi https://github.com/deepfakes/faceswap

Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I. ve Zafeiriou. S. (2019) Retinaface:Single-stage dense face localisation in the wild.arXiv preprint arXiv:1905.00641.

Dumoulin, V., Perez, E., Schucher, N., Strub, F., Vries, H., d., Courville, A. ve Bengio, Y. (2018). Feature-wise transformations. 11 Mayıs 2021, Erişim adresi https://distill.pub/2018/feature-wise-transformations.

Dumoulin, V., Shlens, J. ve Kudlur, M. (2016). A Learned Representation For Artistic Style. CoRR, abs/1610.07629.

Feng, Y., Wu, F., Shao, X., Wang, Y. ve Zhou, X. (2018). Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network (cite arxiv:1803.07835 Comment: 18 pages, 10 figures).

Gatys, L. A., Ecker, A. S. ve Bethge, M. (2015). A Neural Algorithm of Artistic Style (cite arxiv:1508.06576).

Ghiasi, G., Lee, H., Kudlur, M., Dumoulin, V. ve Shlens, J. (2017). Exploring the structure of a real-time, arbitrary neural artistic stylization network. CoRR, abs/1705.06830.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C. ve Bengio, Y. (2014). Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence ve K. Q. Weinberger (eds.), NIPS (p./pp. 2672-2680).

Huang, X. ve Belongie, S. J. (2017). Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. CoRR, abs/1703.06868.

Iglovikov, V. ve Shvets, A. (2018). TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation. CoRR, abs/1801.05746.

Johnson, J., Alahi, A. ve Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In B. Leibe, J. Matas, N. Sebe ve M. Welling (eds.), ECCV (2) (p./pp. 694-711), : Springer. ISBN: 978-3-319-46474-9.

Karras, T., Aila, T., Laine, S. ve Lehtinen, J. (2017). Progressive Growing of GANs for Improved Quality, Stability, and Variation. CoRR, abs/1710.10196.

Karras, T., Laine, S. ve Aila, T. (2019a). A Style-Based Generator Architecture for Generative Adversarial Networks. CVPR (p./pp. 4401-4410), : Computer Vision Foundation / IEEE.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J. ve Aila, T. (2019b). Analyzing and Improving the Image Quality of StyleGAN. CoRR, abs/1912.04958.

King, D. E. (2009). Dlib-ml: A Machine Learning Toolkit.. J. Mach. Learn. Res., 10, 1755-1758.

Krizhevsky, A., Sutskever, I. ve Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems, 25, 1106--1114.

LeCun, Y., Bengio, Y. ve Hinton, G. (2015). Deep Learning. Nature, 521, 436--444. doi: 10.1038/nature14539.

Nagrani, A., Chung, J. S. ve Zisserman, A. (2017). VoxCeleb: A Large-Scale Speaker Identification Dataset. In F. Lacerda (ed.), INTERSPEECH (p./pp. 2616-2620), : ISCA.

Nirkin, Y., Keller, Y. ve Hassner, T. (2019). FSGAN: Subject Agnostic Face Swapping and Reenactment. ICCV (p./pp. 7183-7192), : IEEE. ISBN: 978-1-7281-4803-8.

NVIDIA, Vingelmann, P., ve Fitzek, F. H. P. (2020). CUDA, release: 10.2.89. 11 Mayıs 2021 Erişim adresi https://developer.nvidia.com/cuda-toolkit

Özmen, N. E. ve Buluş, E. (2020). Derin Sinir Ağları Yardımıyla Fotomontaj Tespiti. Mühendislik Bilimleri ve Tasarım Dergisi, 8(5), 236-240.

Pérez, P., Gangnet, M. ve Blake, A. (2003). Poisson image editing. ACM Trans. Graph., 22, 313-318.

Perov, I., Gao, D., Chervoniy, N., Liu, K., Marangonda, S., Umé, C., Dpfks, M., Facenheim, C. S., RP, L., Jiang, J., Zhang, S., Wu, P., Zhou, B. ve Zhang, W. (2020). DeepFaceLab: A simple, flexible and extensible face swapping framework. CoRR, abs/2005.05535.

Pitié, F., Kokaram, A. C. ve Dahyot, R. (2007). Automated colour grading using colour distribution transfer. Comput. Vis. Image Underst., 107, 123-137.

Prajwal, K. R., Mukhopadhyay, R., Namboodiri, V. P. ve Jawahar, C. V. (2020). A Lip Sync Expert Is All You Need for Speech to Lip Generation In the Wild. In C. W. Chen, R. Cucchiara, X.-S. Hua, G.-J. Qi, E. Ricci, Z. Zhang ve R. Zimmermann (eds.), ACM Multimedia (p./pp. 484-492), : ACM. ISBN: 978-1-4503-7988-5.

Reinhard, E., Ashikhmin, M., Gooch, B. ve Shirley, P. (2001). Color Transfer between Images. IEEE Computer Graphics and Applications, 21, 34-41.

Ronneberger, O., Fischer, P. ve Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells ve A. F. Frangi (eds.), Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III (pp. 234--241). Springer International Publishing. ISBN: 978-3-319-24574-4.

Rumelhart, D. E., Hinton, G. E. ve Williams, R. J. (1986). Learning Representations by Back-propagating Errors. Nature, 323, 533--536. doi: 10.1038/323533a0.

Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E. ve Sebe, N. (2019). Animating Arbitrary Objects via Deep Motion Transfer. CVPR (p./pp. 2377-2386), : Computer Vision Foundation / IEEE.

Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E. ve Sebe, N. (2020a). First Order Motion Model for Image Animation. CoRR, abs/2003.00196.

Siarohin, A., Roy, S., Lathuilière, S., Tulyakov, S., Ricci, E. ve Sebe, N. (2020b). Motion-supervised Co-Part Segmentation. CoRR, abs/2004.03234.

Umeyama, S. (1991). Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. IEEE Trans. Pattern Anal. Mach. Intell., 13, 376-380.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Yakovenko, N., Tao, A., Kautz, J. ve Catanzaro, B. (2018). Video-to-Video Synthesis. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi ve R. Garnett (eds.), NeurIPS (p./pp. 1152-1164).

Wiles, O., Koepke, A. S. ve Zisserman, A. (2018). X2Face: A Network for Controlling Face Generation Using Images, Audio, and Pose Codes. In V. Ferrari, M. Hebert, C. Sminchisescu ve Y. Weiss (eds.), ECCV (13) (p./pp. 690-706), : Springer. ISBN: 978-3-030-01261-8.

Wu, W., Zhang, Y., Li, C., Qian, C. ve Loy, C. C. (2018). ReenactGAN: Learning to Reenact Faces via Boundary Transfer. In V. Ferrari, M. Hebert, C. Sminchisescu ve Y. Weiss (eds.), ECCV (1) (p./pp. 622-638), : Springer. ISBN: 978-3-030-01246-5.

Zakharov, E., Shysheya, A., Burkov, E. ve Lempitsky, V. S. (2019). Few-Shot Adversarial Learning of Realistic Neural Talking Head Models. ICCV (p./pp. 9458-9467), : IEEE. ISBN: 978-1-7281-4803-8.

Zhang, K., Zhang, Z., Li, Z. ve Qiao, Y. (2016). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Process. Lett., 23, 1499-1503.

Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X. ve Li, S. Z. (2017). S3FD: Single Shot Scale-invariant Face Detector. CoRR, abs/1708.05237.

Zhang, R., Che, T., Ghahramani, Z., Bengio, Y. ve Song, Y. (2018). MetaGAN: An Adversarial Approach to Few-Shot Learning. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi ve R. Garnett (eds.), NeurIPS (p./pp. 2371-2380).

Zhang, Y., Zhang, S., He, Y., Li, C., Loy, C. C. ve Liu, Z. (2019). One-shot Face Reenactment. CoRR, abs/1908.03251.