Özet

Çekişmeli üretken ağlar, üretici ağın karar verici ağ ile rekabet etmesi gereken bir oyun teorisi senaryosuna dayanmaktadır. Çekişmeli üretken ağlar (GAN) eğitim koleksiyonunu inceleyip olasılık dağılımlarını öğrenirler ve bu olasılık dağılımlarından yola çıkarak daha fazla örnek üretebilirler. GAN'lar arka planların maskelenmesi, model aktarımı ve var olmayan görüntülerin oluşturulması vb. gibi uygulamalara olanak sağlamaktadır. Ancak, oluşturulan görüntülerin giderek daha gerçekçi hale gelmeleri insanların mahremiyeti endişesini artırmaktadır. Bu yazıda yeni görüntülerin oluşturulmasında; görüntüler üzerinde daha başarılı sonuçlar üreten Derin Evrişimli Üretken Ağlar (DCGAN) kullanılmıştır.

Anahtar Kelimeler: Derin Evrişimli Ağlar, Çekişmeli Üretken Ağlar, Derin Öğrenme, Derin Sahte Görüntüler

1.GİRİŞ

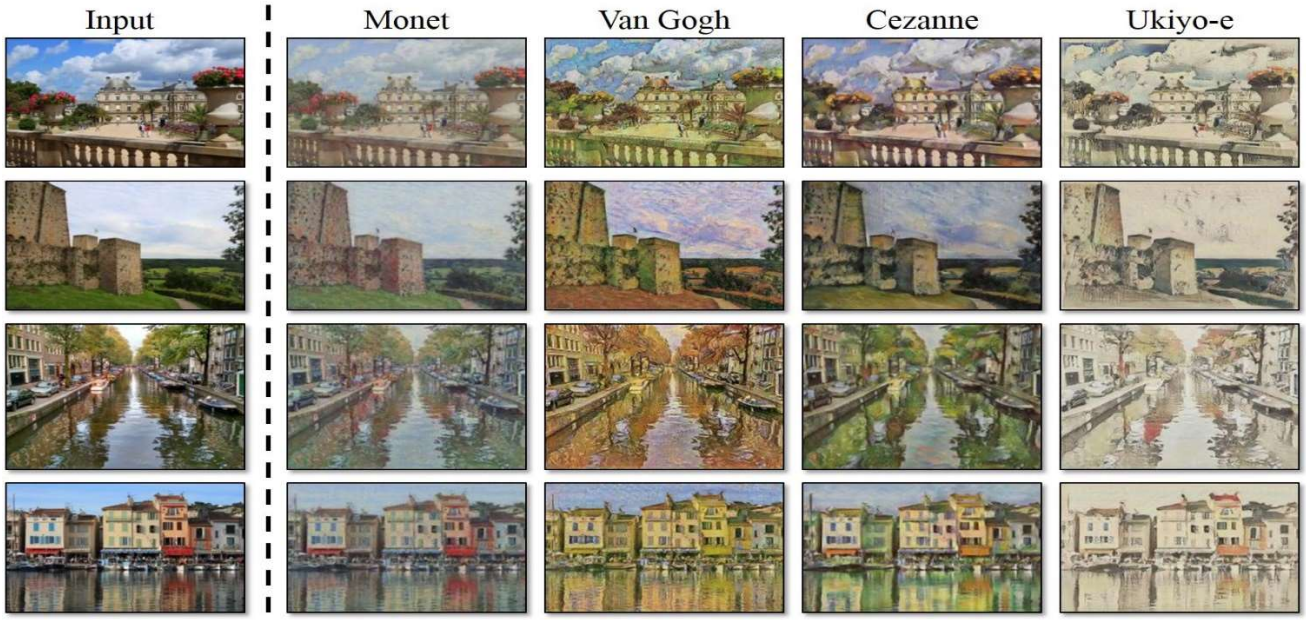
Çekişmeli üretici/üretken ağlar, 2014 yılında Ian Goodfellow ve çalışma grubu tarafından NIPS konferansında tanıtılmıştır. Çekişmeli üretken ağlar olarak adlandırılan bu üretken modeller İngilizce yazılışının baş harfleriyle "Generative Adversarial Networks, GAN" olarak anılmaktadır. Üretken modelleme, girdi verilerindeki kalıpları, orijinal veri kümesinden uygun bir şekilde çıkararak yeni örnekler oluşturabilmek için otomatik olarak keşfetmeyi ve öğrenmeyi içeren denetimsiz öğrenme türüdür. Klasik derin ağ mimarilerinden farklı olarak bir üretici/üretken (Generative, G) ve bir ayırıcı/ayırt edici (Discriminator, D) olmak üzere iki farklı derin ağa sahiptir ve bu iki ağın çekişmeli olarak çalışmasıyla öğrenme işlemini gerçekleştirir.

2.AMAÇ

Elinizde yeterli sayıda örnek olduğu sürece aklınıza gelen herhangi bir veri GAN ile üretilebilir. En çok görülen örnekler resim üretmek üzerinedir. İnsan yüzü, hayvanlar, doğa fotoğrafları, haritalar, araç tasarımları, kıyafetler vb. Sadece yeni veri üretmek için değil, olan bir veriyi başka bir konsepte dönüştürmek için de bu algoritma kullanılabilir.

Resim 1'de GAN ile elde edilmiş bir görüntü sunulmuştur. Sistemi aynı ressamın tabloları ile eğitmek, sistemin o ressamın tarzını öğrenmesine ve sanki o ressam yapmış gibi yeni tablolar üretmesine imkan sağlayacaktır. Böylelikle sanatçıları ölümsüz hale getirebilmek mümkün olacaktır. Burada yapabilecek şeyler sadece resimler değil yeni

şarkılar, ilaçlar, heykeller, mobilyalar, binalar vb. olabilir. Bunların hepsi zaten üzerinde çalışılan ve birçoğu gerçekleştirilmiş uygulamalardır.



Resim 1-GAN ile Elde Edilmiş Görüntüler

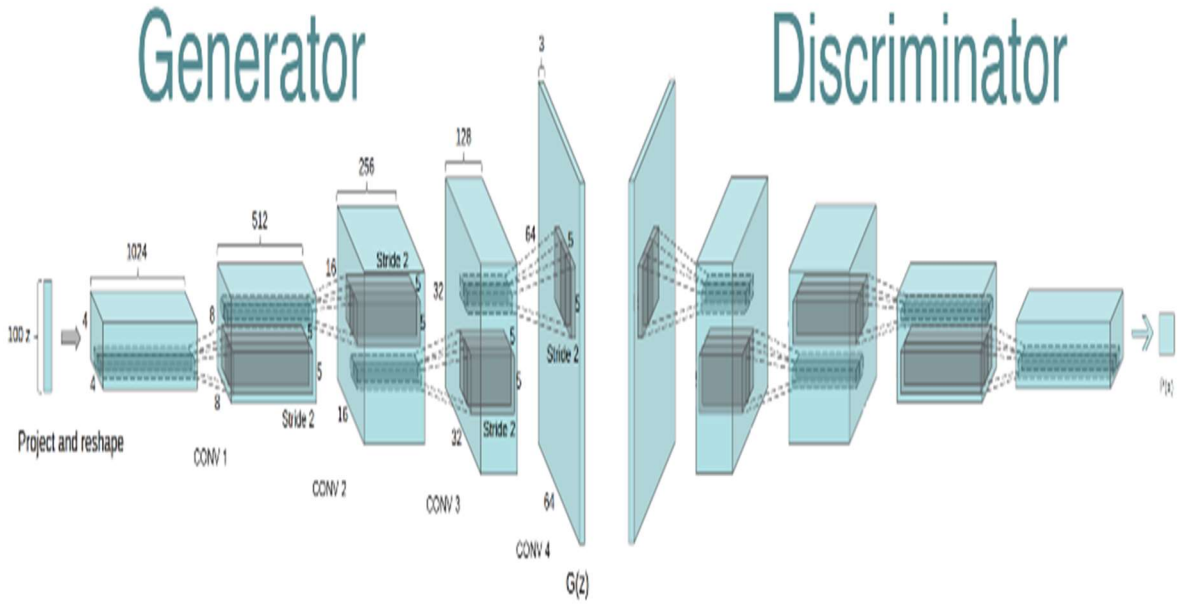
3.YÖNTEM

Üretken görüntü modelleri, üzerinde daha önce birçok uygulama geliştirilen ve iki kategoriye ayrılan, parametrik ve parametrik olmayan modellerdir. Belirli bir forma sahip olan ve örneğin θ gibi bir dizi parametre ile tanımlanan fonksiyonlarla ya da verilere bağlı olarak herhangi bir fonksiyonel formu öğrenmekte özgür olan modeller üretken modelleri temsil ederler. İlk model sınıfına Parametrik Modeller adı verilir, örneğin Gauss modeli (μ ve σ ile parametrelendirilmiş), ikinci model kategorisi ise genellikle Parametrik Olmayan Modeller olarak adlandırılır ve k-En Yakın Komşu bunun popüler bir örneğidir [1]. Parametrik olmayan modeller genellikle mevcut görüntülerin bir veri tabanından eşleştirme yaparlar; görüntü parçalarını eşleştirme ve doku sentezi, süper çözünürlüklü görüntüler ve boyamada kullanılmışlardır [2].

3.1.MODEL MİMARİSİ

Kurulan ağ yapısında kademeli evrişimli ağlar kullanılarak ağın kendi uzaysal alt örneklemesini öğrenmesine imkan verilmiştir. Girdi olarak düzgün dağılımı kullanarak oluşturulan gürültülü görüntü GAN'ın ilk katmanına sunulur, sonuç 4 boyutlu bir tensör olarak yeniden şekillendirilir ve evrişim yığınının başlangıcı olarak kullanılır. Ayırıcı için, son evrişim

katmanı düzleştirilir ve ardından tek bir *Sigmoid* çıktıya beslenir. Örnek bir model mimarisinin görselleştirilmesi Resim 2'de, kurulan katman yapıları ise uygulama bölümünde gösterilmiştir.



Resim 2-Model Mimarisi

Toplu normalleştirme, sinir ağlarının optimizasyonunu büyük ölçüde geliştirir ve DCGAN'lar için oldukça etkili olduğu gösterilmiştir [2]. Üretici ağ ve karar verici ağda her bir katmanda toplu normalleştirme için momentum katsayısı 0,8 olarak belirlenmiştir. Her bir adımda öğrenme algoritması için *LeakyRelu* tercih edilmiş ve öğrenme katsayısı 0,2 olarak alınmıştır.

Üretken düşman ağları, farklı modellerin performansını karşılaştırmayı zorlaştıran bir objektif işlevden yoksundur. İnsan yorumcuların numunelerin görsel kalitesini yargılamasını sağlayarak sezgisel bir performans ölçütü elde edilebilir [3]. Modelin değerlendirilmesinde ikili çapraz entropi kullanılmıştır. İkili çapraz entropi, tahmin edilen olasılıkların her birini 0 veya 1 olabilen gerçek sınıf çıktısıyla karşılaştırır. Ardından, beklenen değerden uzaklığa dayalı olarak olasılıkları cezalandıran puanı hesaplar. Bu, gerçek değere ne kadar yakın veya uzak olduğu anlamına gelir. İkili Çapraz Entropi, düzeltilmiş tahmin edilen olasılıkların logaritmasının negatif ortalamasıdır ve matematiksel olarak ifade edilmiştir (1).

$$Logg\ loss = \frac{1}{N}\sum_{i=1}^{N} -(y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i)) \quad (1)$$

Burada p_isınıf 1'in olasılığını, (1-p_i) ise sınıf 0'ın olasılığını temsil etmektedir. Gözlem sınıf 1'e ait olduğunda, formülün ilk kısmı aktif hale gelir ikinci kısım sıfıra eşit olur ve tam tersi durumda gözlemin gerçek sınıfı 0'dır. Son evrişim bloğunda seçilen *Sigmoid* fonksiyonu ikili çapraz entropi için 0 ve 1 değerlerini üretecektir. Ek olarak, üreteç, çıkış katmanında hiperbolik tanjant (*tanh*) aktivasyon fonksiyonunu kullanır ve üreteç ve ayırıcıya girişler [-1, 1] aralığına ölçeklenir.

Önceki GAN çalışması, eğitimi hızlandırmak için ivmeyi kullanırken, bu çalışmada ayarlanmış hiperparametrelerle Adam optimize ediciyi kullanıldı. Önerilen 0,001 öğrenme oranını çok yüksek bulduğumuz için bunun yerine 0,0002'yi kullandık. Ek olarak, β1 momentum terimini önerilen 0,9 değerinde bırakmanın, eğitimin değişkenliği ve kararsızlığı ile sonuçlanırken 0,5'e düşürülmesi eğitimin dengelenmesine yardımcı olduğunu bulduk [2]. Bu çalışmada da hem üretici ağ için hem ayırt edici ağ için *Adam* optimize edici kullanılmış ve parametre olarak öğrenme oranı her ikisi içinde 1.5e-4, momentum 0,5 olarak belirlenmiştir.

Karar verici ağ yapısında oluşturulan katman yapısında 2 ve üsleri şeklinde artacak (64, 128, 256, 512…) filtre tercihlerinin; literatürde genellikle en iyi sonuçları ürettiği önerilmiştir.

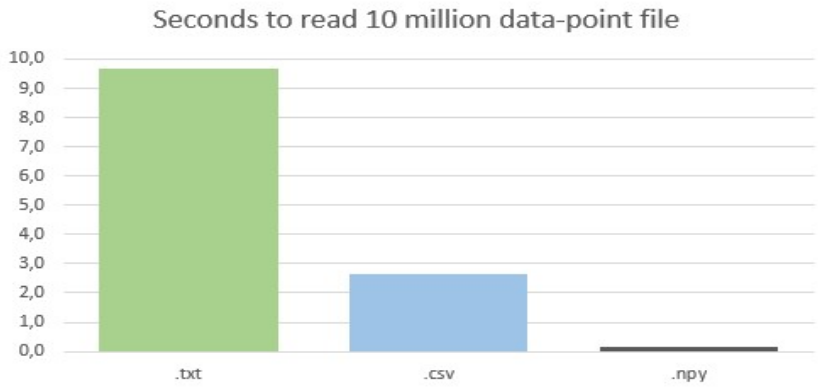
4.UYGULAMA

Çalışmada https://github.com/kiteco/python-youtube-code/tree/master/Deepfake-detection kaynağından alınan 4259 adetlik görüntü setinin, süre ve hafıza kısıtları nedeni ile 490 tanesi kullanılmıştır. Uygulama kodları COLAB üzerinde çalıştırılmıştır.

Resim 3-Görüntü Setinden Bazı Örnekler

4.1.Veri Setinin Hazırlanması ve Mimari

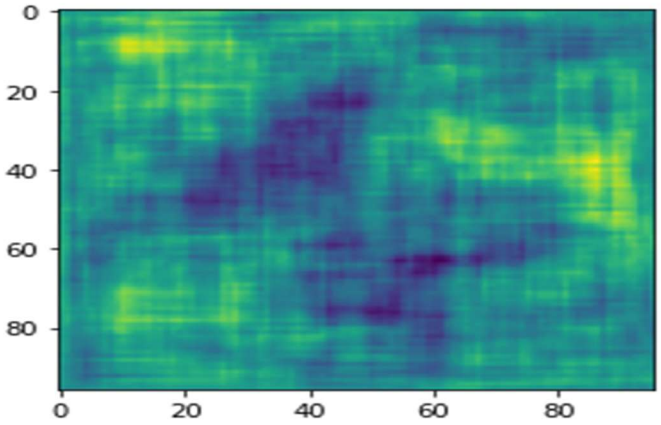
Veri setindeki her bir görsel okunma ve işlenme kolaylığı açısından 96x96 piksel boyutunda olmak üzere boyutlandırılmıştır. *NUMPY* kütüphanesi kullanılarak oluşturulan hazırlanan veri seti elemanları [-1,1] aralığında normalize edilmiştir.



Resim 4-Verilerin Okunma Hızları [4]

Her bir adımda üretici ağın oluşturacağı görüntünün vektör boyutu 100 ve kanal sayısı 3 olarak belirlenmiştir.

Her bir adımda farklı görüntüler üretecek olan üretici ağ için gürültü faktörü normal dağılım kullanarak 0 ile 1 arasında rastgele sayılar ve 100 vektör boyutunda yeni görüntüler üretecektir.

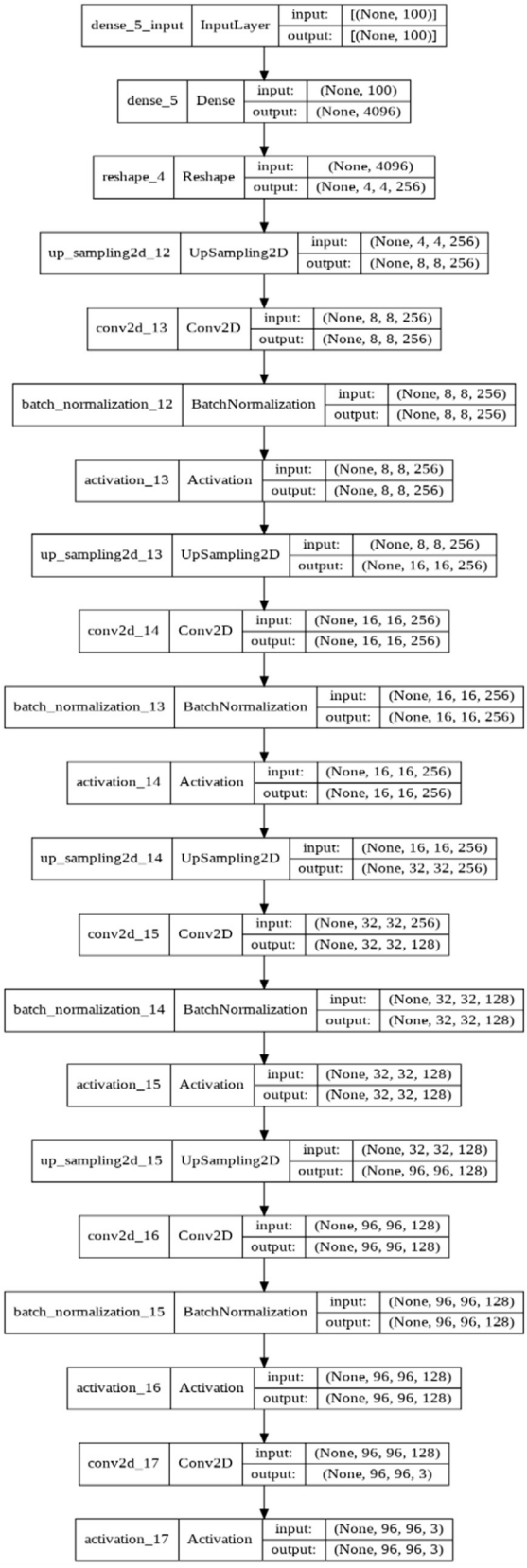


Resim 5-Üreticinin İlk Kez Üreteceği Muhtemel Görüntü

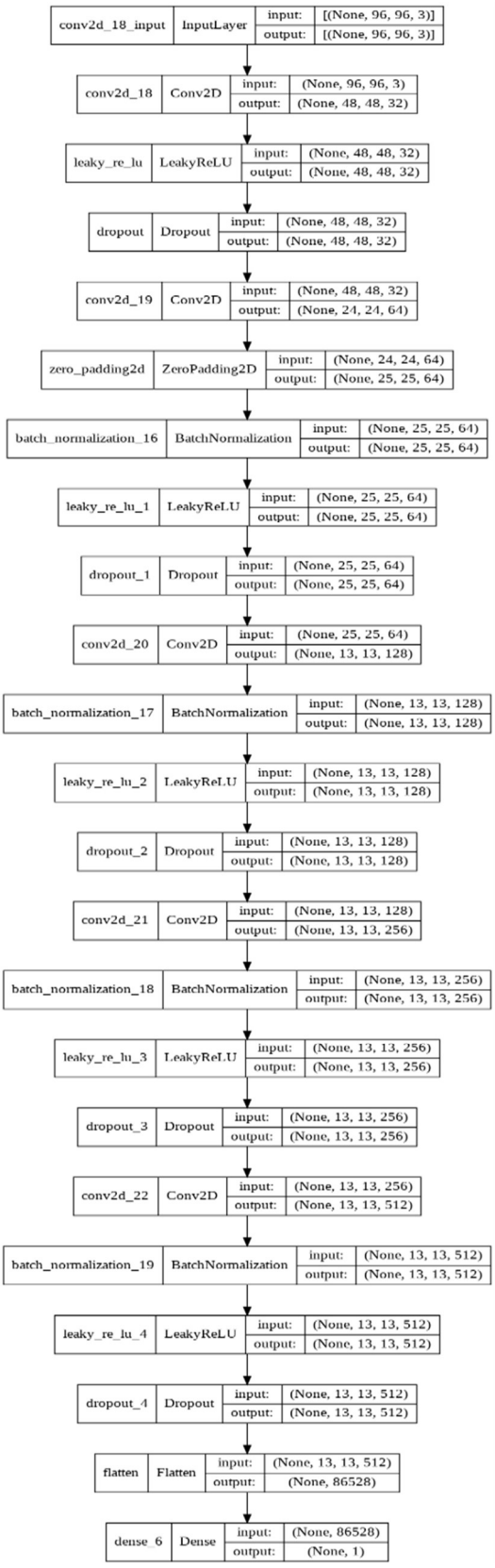
Resim 7 için ayırt edici ağın oluşturacağı çıktı aşağıda gösterilmiştir. Ayırt edici gerçek görüntüleri 1, sahte görüntüleri ise 0 olarak sınıflandıracaktır.

```
tf.Tensor([[0.49972102]], shape=(1, 1), dtype=float32)
```

Model mimarisi bölümünde açıklanan parametreler kullanılarak üretici ve ayırt edici ağ yapıları oluşturulmuş ve mimarileri Resim 6 ve Resim 7'de gösterilmiştir.



Resim 6-Üretici Ağ Katman Yapısı



Resim 7-Ayırt Edici Ağ Katman Yapısı

4.3.Elde Edilen Sonuçlar

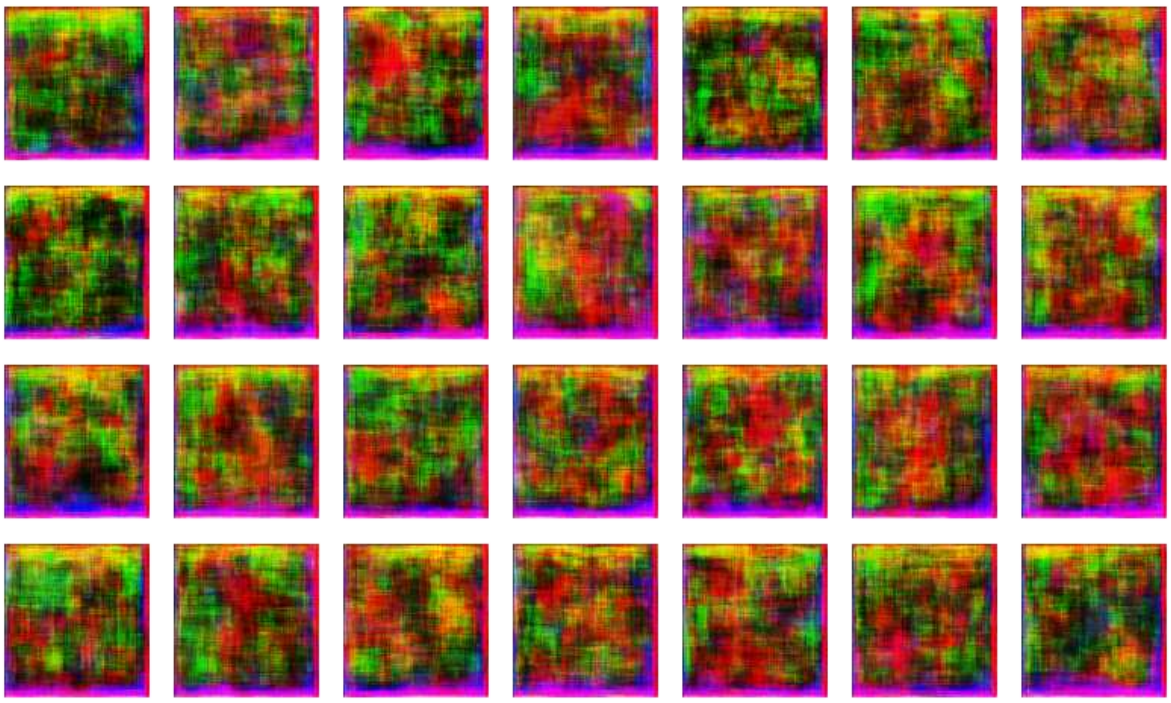
Oluşturulan ağ yapısına durdurma sınırı olarak verilen 50 adım sonunda ağ eğitimi tamamlanmış ve sonuçlar kaydedilmiştir. 490 adetlik veri seti ile eğitimin tamamlanma süresi 2:51:38.95 olarak gerçekleşmiştir.

Adım 1'de üretici ağın ürettiği görüntü Resim 12'de, son adımda ürettiği görüntü ise Resim 13'te gösterilmiştir.

Modelin ürettiği sonuçlar kaydedilmiş ve Tablo 1'de gösterilmiştir.

Adım (Epoch)	Üretici Ağ Kaybı (Generator Loss)	Ayırt Edici Kaybı (Discriminator Loss)
Epoch 1	1.7913249731063843	0.7969334721565247
Epoch 2	2.639806032180786	0.5276483297348022
Epoch 3	3.0682923793792725	0.5465590357780457
Epoch 4	2.840125560760498	0.5931566953659058
Epoch 47	2.279351234436035	1.0442893505096436
Epoch 48	2.106844902038574	0.9658728241920471
Epoch 49	2.2078237533569336	1.1645305156707764
Epoch 50	2.1807661056518555	0.8403931260108948

Tablo 1-Her Bir İterasyonda Hesaplanan Kayıplar



Resim 8-Adım 1'de Üretici Ağ Çıktısı

Resim 9-Adım 50'de Üretici Ağ Sonuçları

5.SONUÇ

Çalışmada elde edilen görüntüler DCGAN yapısının tatmin edici seviyelerde iyi sonuçlar ürettiğini göstermiştir. Bu yönü ile çok çeşitli alanlarda kullanılabilecek olan GAN yapıları çeşitli sorunları da beraberinde getirmekte. Örneğin bir giyim markasının ürünleri kullanılarak sisteme yeni elbiseler ürettirilirse, giyim markası hak talep edebilir mi?

Bir başka sorun ise gerçeğinden ayırt edilemeyen sahte veriler kullanılarak yapılabilecek illegal aktiviteler olabilir. Örneğin hızlıca sahte bir yüz resmi üretip, otomatik metin üreticilerle milyonlarca mail atılabilir veya sosyal medya hesabı açılabilir. *DeepFake* gibi sistemleri kullanılarak bir veya birden fazla kişinin hiç söylemedikleri şeylerin veya hiç bulunmadıkları yerlerin videolarını üretilebilir. Bir kişinin bütün tweetleri veya sosyal medya yazıları kullanılarak onun gibi davranan sahte sosyal medya hesapları meydana getirilebilir.

Üretilen sahte resimlerin hem insanlar hem de yapay zeka sistemleri tarafından gerçeğinden ayırt edilememesi bir çok sorunu ve etik tartışmayı da beraberinde getiriyor. Bu modelin bir web sitesine dönüştürülmüş versiyonu da var. thispersondoesnotexist.com sitesine her girdiğinizde sizin için yeni bir sahte yüz resmi üretecektir.

Bunların hepsi çözülmesi gereken etik ve hukuki sorunlar olarak bekliyor.

KAYNAKÇA

[1] Viswanath Pramod, Lecture Notes, Generative Models, Moitreya Chatterjee, Tao Sun, WZ, Sep 14, 2017

[2] Alec Radford, Luke Metz, Soumith Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

[3] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. arXiv preprint arXiv:1506.05751, 2015.

[4] https://towardsdatascience.com/what-is-npy-files-and-why-you-should-use-them-603373c78883