

Çekişmeli üretken ağ modellerinin görüntü üretme performanslarının incelenmesi

Gaffari ÇELİK^{1,*}, Muhammed Fatih TALU²

¹Ağrı İbrahim Çeçen Üniversitesi, Meslek Yüksekokulu, Bilgisayar Tek. Böl., Ağrı.
²İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Malatya.

Geliş Tarihi (Received Date): 22.04.2019
Kabul Tarihi (Accepted Date): 10.07.2019

Öz

Derin öğrenme alanında yaşanan en önemli gelişmelerden biri, hiç şüphesiz çekişmeli üretken ağ (Generative adversarial network-GAN) modelleridir. GAN olarak anılan bu modeller, görüntü veri kümesinin genişletilmesinde (image augmentation), resim/karikatür boyamada (painting), yüksek çözünürlüğe sahip süper görüntü elde etmede, bir görüntüdeki doku/desenin başka bir görüntüye transferinde kullanılan en modern yaklaşımlar olarak karşımıza çıkmaktadır. Bu çalışmada literatürde yaygın olarak kullanılan GAN modellerinin (cGAN, DCGAN, InfoGAN, SGAN, ACGAN, WGAN-GP, LSGAN), gerçek görüntülere çok benzeyen sentetik görüntüleri üretmedeki performansları incelenmiştir. Çalışmanın orijinalliği, cGAN ve DCGAN'ın avantajlarını barındıran hibrit bir GAN modeli (cDCGAN) geliştirilmesi ve GAN yöntemlerinin performansları, derin öğrenme tabanlı evrimsel sinir ağları(CNN) ile kıyaslamalı olarak değerlendirmesidir. Kodlanan modellerle veri kümelerindeki görüntülere benzer sentetik görüntüler üretilmiştir. Üretilen sentetik görüntülerin mevcut görüntülere benzerliklerini hesaplamak, böylece model performansını değerlendirebilmek için fréchet başlangıç mesafesi (FID) metriği ve CNN kullanılmıştır. Yapılan deneysel çalışmalarda, tüm modellerin zamana bağlı görüntü üretim performansları değerlendirilmiştir. Sonuç olarak, LSGAN modeliyle üretilen görüntülerin yüksek sınıflandırma başarımları sağladığı, ancak DCGAN ve WGANGP ile daha gürültüsüz net görüntüler ürettiği gözlemlenmiştir.

Anahtar kelimeler: Çekişmeli üretken ağlar, GAN, derin öğrenme, görüntü üretimi.

*Gaffari ÇELİK, gcelik@agri.edu.tr, <http://orcid.org/0000-0001-5658-9529>
Muhammed Fatih TALU, fatihtalu@gmail.com, <http://orcid.org/0000-0003-1166-8404>

Investigation of generative adversarial network models' image generation performance

Abstract

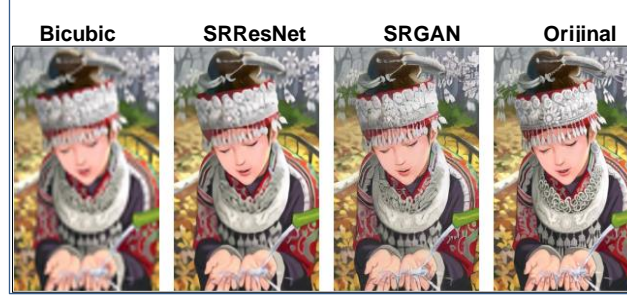
One of the most important developments in the field of deep learning is the generative adversarial network (GAN) models. These models, known as GAN, are the most modern approaches used in image editing, image/cartoon painting, high resolution super image acquisition, and the transfer of the texture/pattern in another image to another image. In this study, the performances of GAN models (cGAN, DCGAN, InfoGAN, SGAN, ACGAN, WGAN-GP, LSGAN), which are commonly used in the literature, in producing synthetic images very similar to the real images were investigated. The originality of the study is the development of a hybrid GAN model (cDCGAN) that incorporates the advantages of cGAN and DCGAN and evaluates the performances of GAN methods in comparison with deep learning based convolutional neural networks (CNN). Synthetic images similar to the images in the data sets were generated with the encoded models. Fréchet inception distance (FID) metric and CNN were used to calculate the similarity of the produced synthetic images to the existing images so as to evaluate the model performance. In the experimental studies, time-based image production performances of all models were evaluated. As a result, it was observed that the images produced by the LSGAN model provide a high classification performance rate, but with DCGAN and WGAN-GP, it produces clearer noise images.

Keywords: *Generative adversarial network, GAN, deep learning, image production.*

1. Giriş

Son yıllarda derin öğrenme teknikleri yapay öğrenme alanında bir çığır açmıştır [1]. Derin öğrenme, gerçek dünyadaki veri kümelerinde karmaşık yapıların öğrenildiği bir makine öğrenmesi yaklaşımı olarak ortaya çıkmıştır. Büyük ölçekli veriler kullanılarak eğitilen derin sinir ağları, görüntülerin anlamsal olarak anlaşılmasıyla ilgili olarak klasik görüntü işleme tekniklerinden önemli ölçüde daha iyi performans göstermiştir [2]. Derin öğrenme, ayrıca görüntü üretiminde büyük bir yetenek sergilemiştir [1]. Derin ağların görüntü üretebilen ilk modeli 2014 yılında Goodfellow [3] tarafından önerilmiştir. Çekişmeli üretken ağlar olarak adlandırılan bu üretken modeller İngilizce yazılışının baş harfleriyle “GAN” anılmaktadır. Klasik derin ağ mimarilerinden farklı olarak bir üretici (generative, G) ve bir ayırıcı (discriminator, D) olmak üzere iki farklı derin ağa sahiptir ve bu iki ağın çekişmeli olarak çalışmasıyla öğrenme işlemini gerçekleştirir. Ayırıcı derin ağ, sentetik olarak üretilen görüntüler (sahte) ile veri tabanındaki görüntüleri (gerçek) birbirinden ayırt etmeye çalışırken; üretici derin ağ, giriş olarak aldığı bir gürültü sinyalinden veri tabanındaki görüntülere benzer görüntüler üretmeye çalışmaktadır. Ancak üretilen görüntüler gerçekmiş gibi ayırıcıyı kandırmaya çalışır. Her iki ağ da paralel olarak rekabetçi bir şekilde eğitilir. Bu işlemler belli bir tekrar aşamasından sonra üretici ağ gerçek görüntülere benzer yeni görüntüler üretir [3]. GAN yaklaşımları, görüntü sentezi, çözünürlük artırma, doku sentezi, video üretimi, görüntü düzenleme gibi alanlarda etkileyici sonuçlar vermiştir [1]. Örnek olarak çözünürlük artırma alanında Ledig [4]’in önerdiği SRGAN’ın klasik yaklaşımlara olan üstünlüğü açıkça görülmüştür (bak Şekil 1). Bu yaklaşım, girişte aldığı düşük

çözünürlükteki bir görüntünün netliğini bozmadan dört kat büyülterek çıkışta üretebilmiştir.



Şekil 1. Çözünürlük artırma yöntemlerinin sonuçları [4].

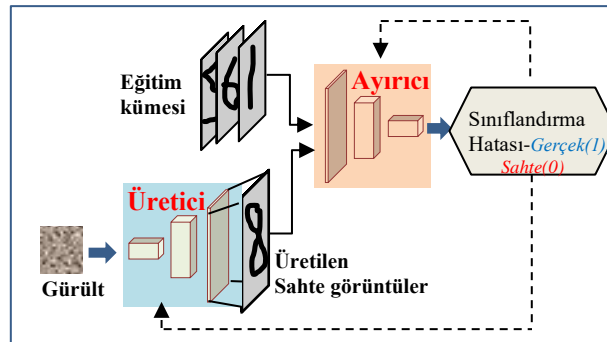
Bu çalışmada sunulan ana katkıların bir özeti aşağıda listelenmiştir:

- En sık kullanılmakta olan GAN modellerinin (cGAN, DCGAN, InfoGAN, SGAN, ACGAN, WGAN-GP ve LSGAN) Python ortamında uygulamaları gerçekleştirilmiştir. Bu modellerin iki farklı görüntü veri tabanı görüntülerine benzer yeni görüntüler üretme performansları incelenmiştir.
- Hibrit bir GAN modeli (cDCGAN) önerilmiştir. Böylece cGAN ve DCGAN modellerinin avantajları tek bir mimaride toplanmıştır.
- Klasik GAN çalışmalarından farklı olarak bu çalışmada toplu eğitimle birlikte sınıflar ayrıştırılarak da eğitim gerçekleştirilmiş ve sonuçlar irdelenmiştir.

2. GAN modeli ve güncel GAN modelleri

2.1. GAN modeli

GAN modellerinin genel mimarisinde biri üretici diğeri ayırıcı olmak üzere iki farklı derin ağ bulunmaktadır. Çekişmeli olarak çalışan bu ağlardan, üretici ağ, giriş olarak aldığı gürültü vektörü eğitim kümesindeki gerçek görüntülere benzetmeyi amaçlarken, ayırıcı ağ, sentetik üretilen görüntülerle (sahte) ve gerçek eğitim görüntüleri birbirinden ayırt etmeye çalışır [5]. Şekil 2'de geleneksel GAN mimarisi gösterilmektedir.



Şekil 2. Geleneksel GAN mimarisi [6,7].

Geleneksel GAN yönteminin sözde kodu aşağıdaki gibi verilmiştir [3, 8]:

Giriş: $x^{(i)}$ gerçek görüntü örneği, $z^{(i)}$ gürültü vektörü

Çıkış: $G(z^i)$ sahte görüntü örneği

1. **For** *iterasyon_sayısı* **do**
2. **For** *adım_sayısı(k)* **do**
3. Sahte görüntüler üret $\{G(z^{(1)}), \dots, G(z^{(m)})\}$
4. Gerçek görüntüler al $\{x^{(1)}, \dots, x^{(m)}\}$
- D**'yi eği (ağırlıkları güncelle): $\nabla_{\theta_d} = \frac{1}{m} \sum_{i=1}^m [\log D(x^i) + \log (1 - D(G(z^i)))]$
5. **end for**
6. Sahte görüntüler üret $\{G(z^{(1)}), \dots, G(z^{(m)})\}$
- G**'yi eği (ağırlıkları güncelle): $\nabla_{\theta_g} = \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(z^i)))$
7. **end for**

Algoritmada kullanılan *iterasyon_sayısı* değişkeni G ve D ağlarının kaç defa eğitileceğini belirtmektedir. *adım_sayı* değişkeni *ToplamÖrüntüSayısı/GrupÖrüntüSayısı* ile hesaplanmaktadır. Bu sayede eğitim kümesindeki görüntüler bütün olarak kullanılmaz ve her pakette *GrupÖrüntüSayısı* kadar görüntü olacak şekilde eğitim gerçekleştirilir.

Ayırıcı derin ağ (D), gerçek ve sahte görüntüleri birbirinden ayıran ikili sınıflandırıcıdır (binary classifier). Bu nedenle “logistic regression” olarak bilinen ikili sınıflandırıcının maliyet fonksiyonu kullanır. Üretici ağ G sembolüyle gösterilmektedir ve $G(z^i)$ nütasyonu ile üretilen sahte görüntüler temsil edilmektedir. Üretici derin ağ, öyle görüntüler üretmek ister ki, ayırıcı üretilen sahte görüntüyü gerçeğinden ayırt edemesin [7]. Bu nedenle üreticinin ürettiği sahte görüntü gerçekmiş gibi ayırıcıya verilir ve elde edilen yüksek hata değeriyle üreticinin eğitilmesi istenir [7].

2.2. Güncel GAN modelleri

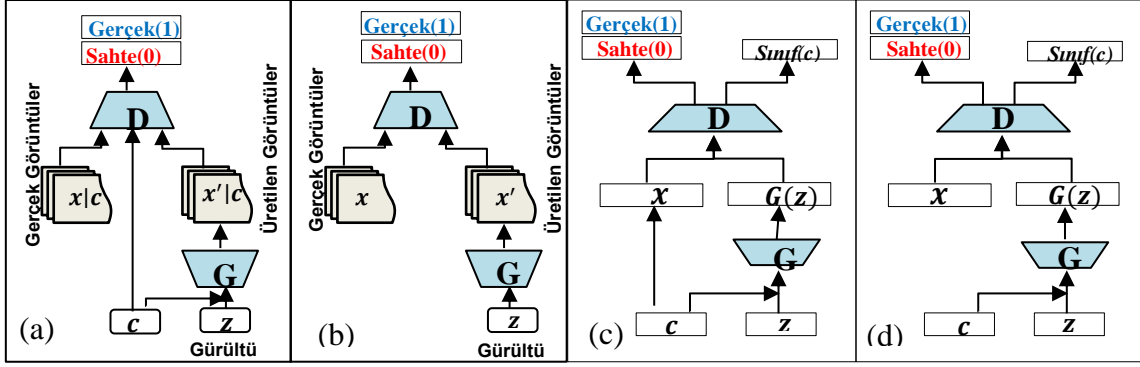
Bu bölümde çekişmeli üretken ağlar ile ilgili literatürde en sık kullanıldığını düşündüğümüz yedi farklı GAN modelinden (cGAN, DCGAN, InfoGAN, SGAN, ACGAN, WGAN-GP, LSGAN) bahsedilmektedir. Modellerin mimarilerindeki farklılıklar ön plana çıkarılmakta ve maliyet fonksiyonları ifade edilmektedir.

2.2.1. Koşullu (Conditional) GAN (cGAN)

Mimarisi Şekil 3 (a)'da gösterilen bu GAN modelinde farklı olarak üretici ve ayırıcı derin ağların girişine ek bilgi (c) verildiği görülmektedir. Böylece ayırıcı, girişindeki gerçek veya sahte görüntünün hangi sınıfa ait olduğunu bilmektedir. Ayrıca üretici gürültü verisiyle birlikte sınıf bilgisini bilerek üretme işlemine başlar [9]. Böylece amaç fonksiyonu aşağıdaki gibi güncellenmiştir.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}} [\log D(x|c)] + \mathbb{E}_{z \sim p_z} [\log (1 - D(G(z|c)))] \quad (1)$$

Burada c , sınıf etiketleri gibi yardımcı bilgi olabilir. $D(x|c)$ ve $G(z|c)$, sırasıyla c etiketi ile verilen görüntüyü ayırt etmemizi ve bu etikette görüntü üretilmesini sağlayan fonksiyonları ifade etmektedir.



Şekil 3. Farklı GAN mimarileri (a) cGANs [1], (b) DCGAN, (c) ACGAN [10], (d) InfoGAN [11].

2.2.2. Derin konvolüsyonel GAN (DCGAN)

Radford ve arkadaşları [12] tarafından geliştirilen DCGAN mimarisi (bak Şekil 3 (b)) üretici ve ayırıcıya bazı kısıtlamalar getirdiler. Bu kısıtlamalar:

- Havuzlama işlemi (pooling) ayırıcı ve üretici ağda kullanılmamaktadır. Görüntü daraltma işlemi ayırıcı ağda konvolüsyon işlemiyle yapılmaktadır
- Hem üretici hem ayırıcı için toplu normleştirme kullanılır
- Daha derin mimariler için tamamen bağlı gizli katmanlar kullanılmaz
- G'de çıkış katmanı hariç diğer katmalarda RELU aktivasyon fonksiyonu kullanılır
- D'nin tüm katmanlarında LeakyReLU aktivasyon fonksiyonu kullanılır.

Yapılan bu kısıtlamalar ile DCGAN'ların eğitimde daha kararlı olduğu ve daha kaliteli görüntüler üretebildikleri, bu nedenle birçok uygulamada yaygın olarak kullanıldığı görülmüştür [1].

2.2.3. Yardımcı sınıflandırıcı GAN (ACGAN)

Bu modelin ayırıcı ve üretici ağlarında log-olabilirlik dağılımı kullanılmaktadır. G giriş olarak z gürültü vektörüne ek olarak c sınıf bilgisi alır ve sahte görüntü ($X_{sahte} = G(z, c)$) üretir. ACGAN mimarisi Şekil 3 (c)'de verilmiştir. Ayırıcı $L_C - L_S$ 'yi, üretici ise $L_C + L_S$ 'yi maksimum yapmak için eğitilmektedir [13].

$$L_S = E[\log P(S = gercek | X_{gercek})] + E[\log P(S = sahte | X_{sahte})] \quad (2)$$

$$L_C = E[\log P(C = c | X_{gercek})] + E[\log P(C = c | X_{sahte})] \quad (3)$$

L_S , imgeler üzerindeki log-olabilirlik dağılımını gösterirken, L_C ise sınıflar üzerindeki log-olabilirlik dağılımını göstermektedir.

2.2.4. Bilgi GAN (InfoGAN)

Üreticinin girişine verilen gürültü vektörüne gizli kod veya ek bilgi (c) eklenmektedir. InfoGAN mimarisi Şekil 3 (d)'de verilmiştir. Üretici, gürültüye ve c 'ye dayalı örnekler üretir. Üretici fonksiyonu $G(z)$ yerine $G(z, c)$ kullanılır. Maliyet fonksiyonu c 'ye bağlı olarak, üretilen sentetik görüntü arasında maksimum karşılıklı bilgiyi (I) elde etmeye dayanır [11]:

$$\min_G \max_D V_{\text{InfoGAN}} = V_{\text{GAN}}(D, G) - \lambda I(c; G(z, c)) \quad (4)$$

$$V_{\text{GAN}}(D, G) \equiv \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x|c)] + \mathbb{E}_{x \sim p_z} [\log (1 - D(G(z|c)))] \quad (5)$$

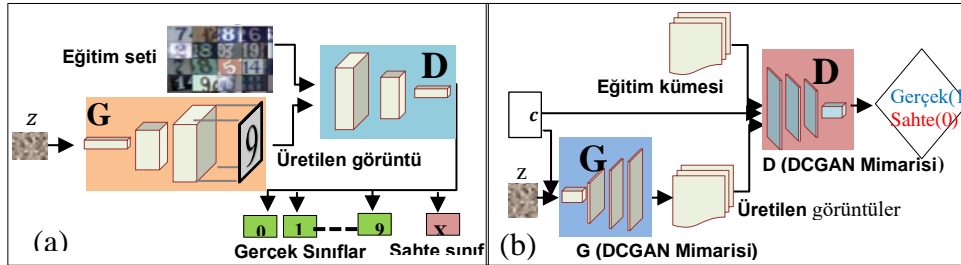
Klasik GAN modelinde G , ek bilgi(c)'ye bağlı olarak istenilen bilgideki görüntüyü üretemezler. Böyle bir problemin üstesinden gelmek için c ile G 'nin üreteceği görüntü($G(z, c)$) arasında karşılıklı bilgi(I)'ye ihtiyaç vardır. $I(c; G(z, c))$ değerinin yüksek çıkması istenir. I değeri *entropi*(H) kullanarak tahmin edilebilir [11].

$$I(c; G(z, c)) = H(c) - H(c|G(z, c)) \quad (6)$$

$I(c; x)$, c biliniyorsa, x 'i ne kadar bildiğimizi ölçer. x tahmin edilen c ile tamamen alakasızsa, $I(c; x)$, 0'a eşittir. D , c 'yi doğru tahmin ederse, I değeri yüksek çıkacağından maliyet düşürecektir. InfoGAN ekstra bir hiper-parametre(λ) getirirse de, basit gizli kodlar için 1 değeri verilmelidir.

2.2.5. Yarı eğitici GAN (SGAN)

Bu modelde ayırıcı ağın son katmandaki aktivasyon fonksiyonu değiştirilir. Klasik ayırıcı ağ, giriş olarak aldığı görüntüden bir olasılık değeri üreten, yani tek bir sigmoid fonksiyonla biten bir ağıdır. Ancak SGAN'ın çıkış katmanı gerçek ve sahte sınıfların tamamı için çıkışlar üreten bir softmax katmanına sahiptir[14]. SGAN mimarisi Şekil 4(a)'da gösterilmektedir.



Şekil 4. GAN mimarileri (a) SGAN [15], (b) cDCGAN

2.2.6. Wasserstein GAN'ların geliştirilmiş eğitimi (WGAN-GP)

Arjovsky ve arkadaşları [16] tarafından geliştirilen bu modelin ayırıcı ve üretici ağlarının maliyet fonksiyonları aşağıdaki gibi güncellenmiştir. Ayırıcı ağın sigmoid katmanı yerine ağırlık kırpma yönteminin kullanılması, olasılık mesafesini ölçmek için Earth-Mover (EM) metriği kullanılması ve maliyet fonksiyonunda log kullanılmaması göze çarpan güncellemeler olarak karşımıza çıkmaktadır [16]:

$$L_D^{WGAN} = E[D(x)] - E[D(G(z))] \quad (7)$$

$$L_G^{WGAN} = E[D(G(z))] \quad (8)$$

WGAN, bazen sadece kötü örnekler üretebildiği görülmüştür [17]. Bu olumsuz duruma ağırlık kırpma fonksiyonunun neden olduğu anlaşılmıştır. Daha sonra ağırlık kırpma yöntemine alternatif olarak yeni bir yöntem WGAN-GP önerilmiştir. Buna göre maliyet fonksiyonu aşağıdaki şekilde güncellenmiştir:

$$L_D = \lambda E\left[\left(\left|\nabla D\left(ax - (1 - aG(z))\right)\right| - 1\right)^2\right] \quad (9)$$

$$L_D^{WGAN-GP} = L_D^{WGAN} + L_D$$

$$L_G^{WGAN-GP} = L_G^{WGAN} \quad (10)$$

Burada a ve λ hiper parametrelerdir. Bu parametrelerin varsayılan değerler olarak $a = 0.0001$, $\lambda=10$ verilmelidir.

2.2.7. En az kareler GAN (LSGAN)

Ayırıcıyı bir sınıflandırıcı olarak gören GAN'larda hata fonksiyonu olarak Sigmoid Cross Entropy (SCE) fonksiyonu kullanılmaktadır. Bu fonksiyon yardımıyla üretici güncellenirken, bazı örnekler için gradyan bilgisi yok olmaktadır. Çözüm olarak SCE yerine en az kareler yaklaşımı önerilmiştir [18]. Buna göre maliyet fonksiyonları aşağıdaki gibi güncellenmektedir.

$$\min_D V_{LSGAN}(D) = \frac{1}{2} \mathbb{E}_{x \sim p_{data}(x)} [(D(x) - b)^2] + \frac{1}{2} \mathbb{E}_{z \sim p_z(z)} [(D(G(z)) - a)^2] \quad (11)$$

$$\min_G V_{LSGAN}(G) = \frac{1}{2} \mathbb{E}_{z \sim p_z(z)} [(D(G(z)) - c)^2] \quad (12)$$

Burada c , üreticinin ayırıcıyı sahte verilere inanmasını istediği değeri belirtir. a ve b , sahte ve gerçek veriler için kullanılan etiketlerdir [18].

2.2.8. Koşullu derin konvolüsyonel GAN (cDCGAN)

Geliştirilen hibrit GAN mimarisidir. cGANs ve DCGAN'nın avantajlarını içermektedir. Yapılan uygulamalarda cGAN ile üretilen görüntüler sınıflandırılabilen ancak görüntünün düşük kalite çözünürlüğe sahip olmaktadır. cGAN'nın aksine DCGAN daha kaliteli görüntüler üretebilmektedir, ancak belirli bir sınıf görüntünün üretimi gerçekleştirilememektedir. Her iki modelin avantajları birleştirilerek belirli bir sınıfta yüksek çözünürlükte görüntü üretebilme fikri cDCGAN hibrit modelini ortaya çıkarmıştır. Geliştirilen modelinin mimarisi Şekil 4(b)'de verilmiştir.

cDCGAN maliyet fonksiyonu aşağıdaki gibi verilmiştir:

$$\max_D V_{cDCGAN}(D) = \mathbb{E}_{x \sim p_{data}} [\log D(x|c)] + \mathbb{E}_{x \sim p_z} [\log (1 - D(G(z|c)))] \quad (13)$$

$$\min_G V_{cDCGAN}(G) = \mathbb{E}_{x \sim p_z} [\log (1 - D(G(z|c)))] \quad (14)$$

Burada c , sınıf etiketi(ek bilgi). $D(x|c)$, c etiketi ile verilen görüntüyü ayırt etmemizi, $G(z|c)$, bu etikette görüntü üretilmesini sağlayan fonksiyonlardır. cDCGAN modeli aynı zamanda DCGAN modelinde yapılan kısıtlamalar göz önünde bulundurularak inşa edilmiştir.

3. Deneysel çalışmalar

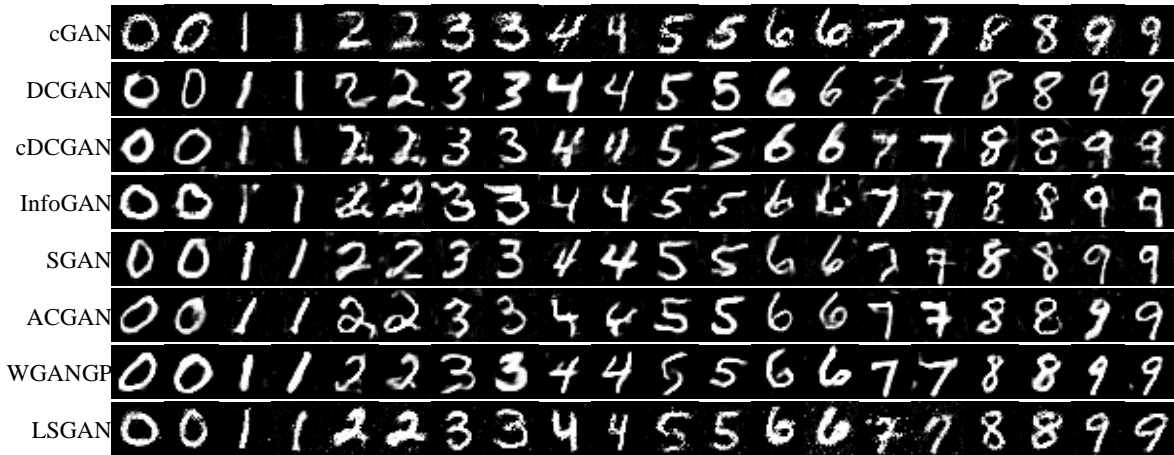
GAN modellerinin görüntü üretim performanslarını analiz edebilmek için MNIST ve Fashion-MNIST veri kümeleri kullanılmıştır. Her iki veri kümesinde 10 sınıf ve her

sınıfta yaklaşık 6.000 olmak üzere toplam 60.000 görüntü bulunmaktadır. Üretilen sentetik görüntülerin gerçek görüntülere ne kadar benzediğini hesaplayabilmek için FID benzerlik metriği ve CNN tabanlı bir sınıflandırma modeli kullanılmıştır. FID benzerlik metriği aşağıdaki gibidir:

$$FID(x, g) = \left\| \mu_x - \mu_g \right\|_2^2 + Tr \left(C_x + C_g - 2(C_x C_g)^{\frac{1}{2}} \right) \quad (15)$$

Burada (μ_x, C_x) ve (μ_g, C_g) , sırayla gerçek ve sentetik görüntünün ortalama ve kovaryans değerlerini göstermektedir [19]. Sentetik görüntü gerçeğine çok yakınsa FID metriği 0'a yakın, çok uzaksa 1'e yakın çıkmaktadır. Gerçek/sentetik görüntü benzerliğini hesapladığımız diğer yöntemde ise MNIST veri kümesiyle eğitilmiş (doğruluk %99,1) bir CNN ağı kullanılmaktadır. CNN, bilgisayar görme alanındaki yüksek boyutlu verileri öğrenmeye yönelik bir derin öğrenme yöntemidir [20]. Buna göre yeni üretilen sentetik bir görüntü eğitilmiş CNN ağına verilerek çıkış kümesindeki doğruluk oranına bakılmaktadır. Üretilen bir sentetik görüntünün herhangi bir sınıfa yüksek doğrulukla atanması üretim işleminin kalitesini ortaya koymaktadır.

MNIST ve Fashion-MNIST veri kümeleri kullanılarak yapılan iki uygulamada da adadelta(lr=1.0, rho=0.95) optimizasyon yöntemi kullanılmıştır. Bu yöntem, zaman içerisinde sadece birinci dereceden bilgiyi kullanarak dinamik olarak adapte olur ve rastgele dereceli inişin(stochastik gradient descent) ötesinde minimum hesaplama ek yüküne sahiptir [21]. Eğitim işlemi 128 örnekli(batch) olarak yapılmıştır. İlk uygulamada, GAN modellerinin MNIST görüntülerini üretebilme kabiliyeti incelenmiştir. GAN modelleri her sınıf için ayrı ayrı eğitilmiştir. Eğitim süresi 12 dk. tutulmuştur. Eğitim sonrası her modelin sınıfların her biri için 100 görüntü üretmesi istenmiştir. Üretilen sentetik görüntü örnekleri Şekil 5'te gösterilmektedir.



Şekil 5. GAN modelleriyle üretilen MNIST görüntü örnekleri.

Üretilen 100 adet sentetik görüntüyle, eğitim kümesindeki 100 adet gerçek görüntü arasındaki FID benzerlik değerleri Tablo 1. (sol)'da verilmiştir. Bu sonuçlarının daha iyi anlaşılabilmesi için tablonun son satırında gerçek görüntüler (100 adet) arasındaki FID benzerlik oranları verilmiştir. Bu satırdaki FID değerlerinin yüksek çıkması veri kümesi içerisindeki veri çeşitliliğinin bir göstergesidir. Normalde FID benzerlik oranının düşük çıkması görüntüler arasındaki benzerliğin yüksek olduğunu anlamına gelir. Ancak gerçek görüntülerin bile kendi aralarında bir benzerlik hatası vermesi nedeniyle bu hata değeri (offset) genel FID sonuçlarından çıkarılmış ve Tablo 1 (sağ)'da

ki sonuçlar elde edilmiştir. Elde edilen sonuçlar içerisinde maksimum değere sahip olan modelin en benzer görüntü ürettiği yargısına varılmıştır. Buna göre, (1, 4, 7, 9) sınıflarında LSGAN, (2, 8) sınıflarında cGAN, (3, 5) sınıflarında cGAN ve LSGAN, (0) sınıfında InfoGAN ve cGAN en benzer görüntüleri üretmiştir.

Tablo 1. GAN modellerinin görüntü üretme sonuçları (FID benzerlik değerleri).

	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
cGAN	4.6	4.1	4.0	3.0	3.2	3.1	3.5	3.8	3.4	3.2	0,5	0,2	0,9	0,3	0,6	0,2	0,0	0,3	0,3	0,2
DCGAN	3.9	3.9	2.9	2.8	2.7	2.8	2.8	3.5	2.9	2.8	0,1	0,0	0,1	0,2	0,1	0,0	0,7	0,0	0,0	0,1
cDCGAN	3.9	3.7	3.1	2.6	3.5	2.7	3.4	4.2	2.8	3.0	0,1	0,1	0,0	0,0	0,8	0,1	0,1	0,7	0,2	0,0
InfoGAN	3.5	3.4	2.9	2.4	2.6	2.6	3.1	3.6	2.8	2.6	0,5	0,3	0,1	0,1	0,0	0,2	0,4	0,1	0,2	0,4
SGAN	3.6	3.6	2.8	2.6	2.6	2.7	3.2	3.5	2.7	2.7	0,4	0,1	0,2	0,0	0,0	0,1	0,3	0,0	0,3	0,2
ACGAN	4.1	3.8	2.9	2.4	2.9	2.7	3.3	3.3	3.0	2.8	0,0	0,0	0,1	0,1	0,3	0,1	0,1	0,1	0,0	0,1
WGAN-GP	3.9	4.0	2.9	2.8	3.0	2.9	3.3	3.6	2.9	3.1	0,1	0,2	0,1	0,1	0,4	0,0	0,2	0,1	0,1	0,1
LSGAN	4.0	4.4	3.2	3.0	3.5	3.1	3.6	4.3	3.2	3.5	0,0	0,6	0,1	0,3	0,9	0,2	0,0	0,8	0,1	0,5
MNIST	4.0	3.8	3.1	2.6	2.6	2.8	3.5	3.5	3.0	3.0	0	0	0	0	0	0	0	0	0	0

FID benzerlik metriği ile elde edilen bu sonuçların sağlamlasını yapmak için CNN tabanlı bir benzerlik ölçme metriği kullanılmıştır. Üretilen her bir görüntü eğitilmiş CNN ile sınıflandırılmak istenmiştir. Sınıflandırmanın doğru veya yanlış olmasına göre Tablo 2’de verilen sonuçlar elde edilmiştir. Başarı oranı denklem 16’ya göre hesaplanmıştır:

$$Başarı(GAN_i) = \left(1 - \frac{Hata(GAN_i)}{\sum_j Hata(GAN_j)}\right) * 100 \quad (16)$$

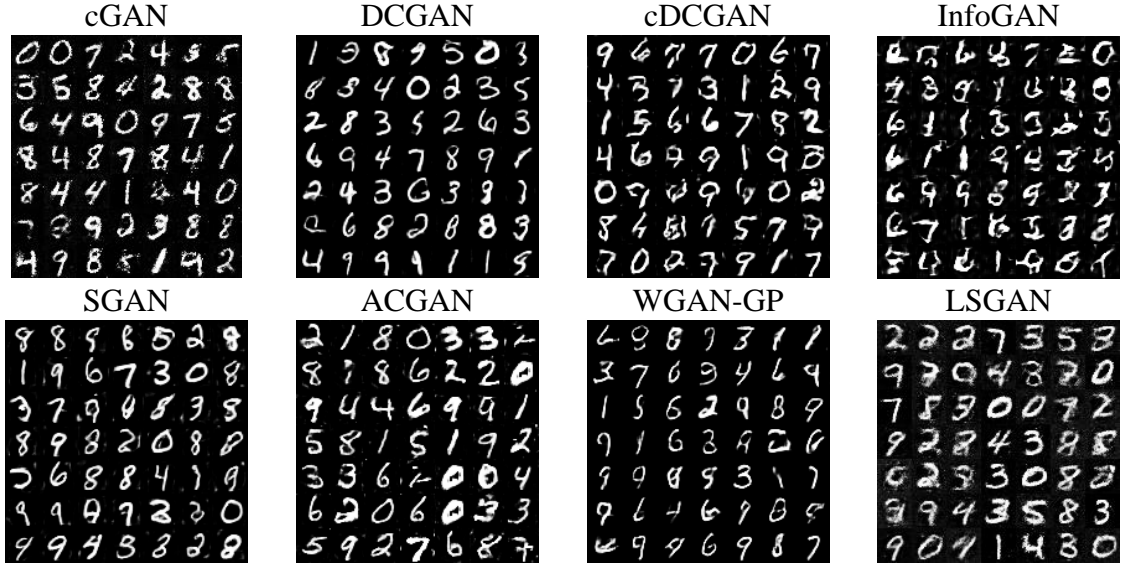
Tablo 2’de 0-9 sütunlarındaki rakamlar sınıflandırma işleminde başarısız olan görüntü sayısını ifade etmektedir. Buna göre, LSGAN kullanılarak üretilen 10 sınıf görüntünün 8 sınıfı hatasız sınıflandırılabilmiştir. cDCGAN modelinin ise 5 sınıfta hatasız ve bazı sınıflarda (8 gibi) diğer modellerden daha iyi sonuç verdiğini görülmüştür.

Tablo 2. Üretilen görüntülerin CNN kullanılarak sınıflandırılması (rakamlar yanlış sınıflandırılan görüntü sayısını göstermektedir).

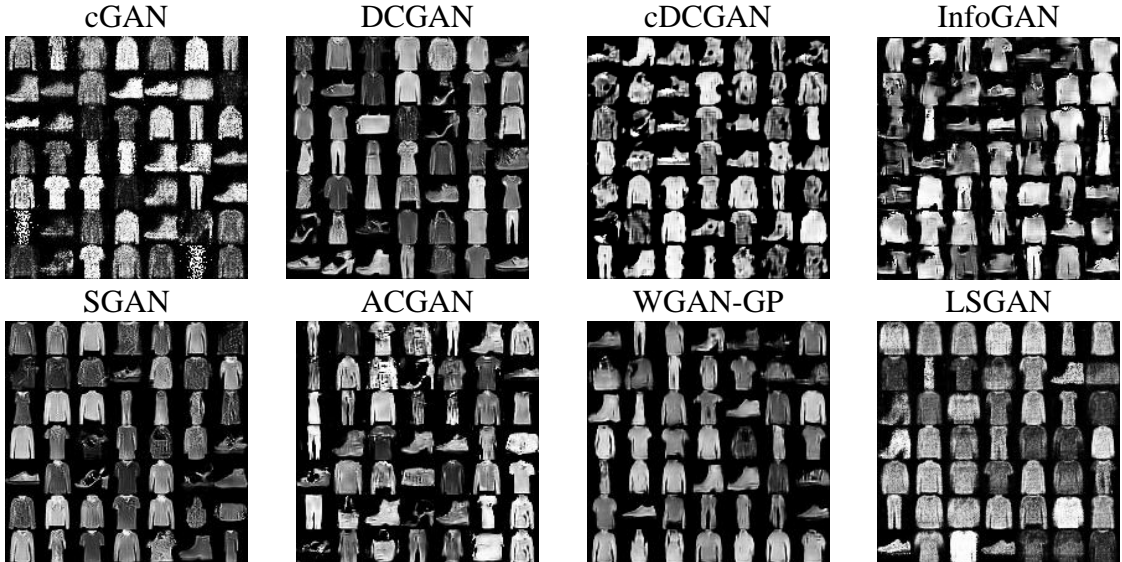
	0	1	2	3	4	5	6	7	8	9	Hata	Başarı
cGAN	0	0	3	2	2	0	2	0	3	2	14	90,60
DCGAN	0	0	0	3	0	2	0	1	2	1	9	93,96
cDCGAN	0	0	2	4	0	3	0	5	0	15	29	80,54
InfoGAN	0	0	4	1	3	2	2	1	15	12	40	73,15
SGAN	0	1	1	2	4	0	1	3	2	6	20	86,58
ACGAN	1	0	1	2	2	1	0	1	0	4	12	91,95
WGAN-GP	1	0	2	1	4	2	1	3	2	5	21	85,91
LSGAN	0	0	2	0	0	0	0	0	2	0	4	97,32
	Toplam Hata										149	

İkinci uygulamada üretken modellerin toplu eğitim yapıldığındaki sonuçları incelenmiştir. Önceki uygulamada her bir sınıf için eğitim işlemi ayrı yapılırken, bu uygulamada sınıfların tamamı toplu bir şekilde eğitilmiştir. Eğitim işlemi MNIST ve Fashion MNIST veri kümelerinde gerçekleştirilmiş ve toplam 90 dk. sürmüştür. Eğitim sonrası modeller tarafından üretilen sentetik görüntüler Şekil 6 ve Şekil 7’de

gösterilmiştir. Şekil 6'daki sonuçlar incelendiğinde, DCGAN, SGAN ve WGANGP'nin net görüntüler ürettiği, InfoGAN, cGAN ve LSGAN'ın gürültülü görüntüler ürettiği görülmektedir. Ancak FID benzerlikleri ve CNN tanıma sonuçlarına bakıldığında LSGAN'ın en başarılı sonuçlar verdiği görülmektedir. Yine DCGAN ve SGAN'ın karakterleri oldukça düzgün üretebildiği görülmektedir. Şekil 7'de görüldüğü gibi, modellerin Fashion-MNIST nesne görüntülerini üretme performansları MNIST sonuçlarıyla paralellik arz etmektedir. DCGAN ve WGANGP tarafından üretilen nesnelere oldukça net olmasına karşılık diğer GAN modellerinde gürültü seviyesi ve nesne bozulma oranı yüksektir.



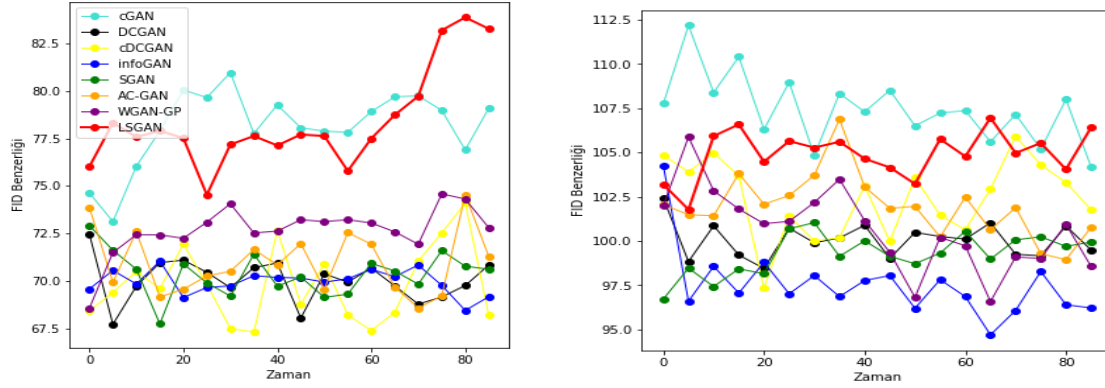
Şekil 6. Modeller tarafından üretilen sentetik MNIST görüntüleri.



Şekil 7. Modeller tarafından üretilen sentetik Fashion-MNIST görüntüleri.

Eğitim süresinin model doğruluğuna etkisini analiz etmek için 90 dk. süren eğitim işleminin 5 dk.'lık periyotlarında (toplam 18 zaman diliminde) FID benzerlik değerleri elde edilmiştir. Her bir dilim için ağırlıklar kaydedilmiş ve eğitim sonrası bu ağırlıklar kullanılarak görüntü üretilmiştir. Üretilen görüntülerin FID benzerlikleri hesaplanarak

Şekil 8’de gösterilmiştir. Şekil 8(sol) sonuçları incelendiğinde LSGAN modelinin FID benzerlik oranı zamanla daha istikrarlı bir şekilde arttığı ve 90 dk. sonunda en yüksek değere ulaştığı görülmektedir. Aynı şeklin sağ grafiğindeki sonuçlarda da LSGAN modelinin 90 dk. sonunda en yüksek benzerlik değeri verdiği görülmüştür. Sonuç olarak yapılan iki deneysel çalışmada (sınıfsal ve toplu eğitim) LSGAN modelinin eğitim kümesindeki görüntüleri üretme performansının en iyi olduğu görülmüştür.



Şekil 8. Modellerin farklı eğitim sürelerindeki üretme başarıları (sol) MNIST (sağ) Fashion-MNIST.

4. Sonuçlar ve tartışma

Bu çalışmada önerdiğimiz model ile birlikte sekiz farklı üretken ağ modelinin görüntü üretimindeki performansları değerlendirilmiştir. Modellerin ayrı ve toplu eğitimler sonucundaki performansları ayrı ayrı incelenmiştir. Tüm üretken modeller Python ortamında kodlanmıştır. Görüntü üretimi için MNIST ve Fashion-MNIST veri kümeleri seçilmiştir. Üretilen sentetik görüntülerin gerçek görüntülere benzerliklerinin hesaplanabilmesi için literatürde en sık kullanılan FID benzerlik metriği ve CNN kullanılmıştır. Elde edilen sonuçlar incelendiğinde LSGAN modeliyle üretilen sentetik görüntülerin CNN tabanlı eğitilmiş bir derin ağla sınıflandırılması sonucunda %97.32 oranında bir başarımla elde edilmiştir. Önerilen cDCGAN modeli ise, LSGAN modelinden sonra DCGAN ile birlikte 5 sınıfta en iyi sonucu verdiği görülmüştür, ancak 9 nolu görüntü üretimindeki başarısızlığı sınıflandırma başarısını olumsuz etkilemiştir. Sonuç olarak, LSGAN modeliyle üretilen görüntülerin yüksek sınıflandırma başarımla sağladığı, ancak DCGAN ve WGAN-GP ile daha gürültüsüz net görüntüler ürettiği gözlemlenmiştir.

Kaynaklar

- [1] Wu, X., Xu, K. ve Hall, P., A survey of image synthesis and editing with generative adversarial networks, **Tsinghua Sci. Technol.**, 22, 6, 660–674,(2017).
- [2] Wason, R., Deep learning: Evolution and expansion, **Cognitive Systems Research**, 52, 701-708, (2018).
- [3] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. ve Bengio, Y, Generative Adversarial Networks, in **Proc. 27th Int. Conf. Neural Information Processing Systems**, 2672–2680, Montreal, (2014).

- [4] Ledig, C., Theis, L., Husz'ar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z. ve Twitter, W. S., Photo-realistic single image super-resolution using a generative adversarial network, **017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, 105–114, Honolulu, (2017).
- [5] Goodfellow, I., NIPS 2016 Tutorial: Generative Adversarial Networks, **arXiv:1701.00160**, (2016).
- [6] Silva, T. A, Beginner's Guide to Generative Adversarial Networks (GANs), <https://skymind.ai/wiki/generative-adversarial-network-gan>, (28.12.2018).
- [7] Langr, J. ve Bok, V., GANs in Action, **MEAP edition Manning Publications**, 350, Newyork USA, (2018).
- [8] Hua, G., Jégou, H , **Adversarial Training for Sketch Retrieval in** Creswell, A. ve Bharath, A.A., *Computer Vision – ECCV 2016 Workshops*, Springer International Publishing, 798-809, Switzerland, (2016).
- [9] Mirza, M. ve Osindero, S., Conditional Generative Adversarial Nets, **CoRR**, abs/1411.1784, 1–7, (2014).
- [10] Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J.ve Greenspan, H., GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification, **Neurocomputing**, 321, 321–331, (2018).
- [11] Chen, X., Duan, Y., Houthoof, R., Schulman, J., Sutskever, I. ve Abbeel, P., InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, **In Proceedings of the 30th International Conference on Neural Information Processing Systems**, 2172–2180, Barcelona, (2016).
- [12] Radford, A., Metz, L. ve Chintala, S., Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, **arXiv:1511.06434**, (2016)
- [13] Odena, A., Olah, C. ve Shlens, J., Conditional Image Synthesis With Auxiliary Classifier GANs., **In Proceedings of the International Conference on Machine Learning**, 2642–2651, Sydney, (2017).
- [14] Odena, A, Semi-Supervised Learning with Generative Adversarial Networks, **arXiv:1606.01583**, (2016).
- [15] Silva, T, Semi-supervised learning with Generative Adversarial Networks (GANs), (2018). <https://towardsdatascience.com/semi-supervised-learning-with-gans-9f3cb128c5e>, (25.12.2018).
- [16] Arjovsky, M., Chintala, S. ve Bottou, L., Wasserstein GAN, **arXiv:1701.07875**, (2017).
- [17] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. ve Courville, A.C., Improved Training of Wasserstein Gans, **In Proceedings of the Advances in Neural Information Processing Systems**, 5769–5779, Long Beach, (2017).
- [18] Mao, X., Li, Q., Xie, H., Lau, R. Y.K., Wang, Z. ve Smolley, S. P., Least Squares Generative Adversarial Networks, **2017 IEEE International Conference on Computer Vision** , 2813–2821, Venice, (2017).
- [19] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. ve Hochreiter, S., GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, **31st Conference on Neural Information Processing Systems (NIPS 2017)**, 6626-6637, Long Beach, (2017).
- [20] Krizhevsky, A., Sutskever, I. ve Hinton, G. E., ImageNet Classification with Deep Convolutional Neural Networks, **Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12)**, 1097–1105, Lake Tahoe, (2012).
- [21] Zeiler, M. D., ADADELTA: An Adaptive Learning Rate Method, **arXiv:1212.5701**, (2012).