

---

*Araştırma Makalesi / Research Article*

---

## **Bıyık Deseni Üretiminde Çekişmeli Üretici Ağların Performans Karşılaştırması**

Emrullah ŞAHİN<sup>1\*</sup>, Muhammed Fatih TALU<sup>1</sup>

<sup>1</sup>*İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Türkiye  
(ORCID: [0000-0002-3390-6285](https://orcid.org/0000-0002-3390-6285)) (ORCID: [0000-0003-1166-8404](https://orcid.org/0000-0003-1166-8404))*

---

### **Öz**

Bu çalışmada görüntüden görüntüye dönüşüm yapan çekişmeli üretici ağ mimarilerinin performans incelemesi yapıp, sentetik görüntü üretimindeki başarımları değerlendirilmiştir. Bu modellerin kaliteli bir başarımlı değerlendirilmesi için standartlaştırılmış veri kümeleri yerine gerçek iş alanından toplanılan denim2bıyık veri kümesi kullanılmıştır. Denim kumaşları üzerine çizilen bıyık desenleri lazer cihazıyla oluşturulmaktadır. Bu cihazın istenilen bıyık desenini oluşturabilmesi için uzmanlaşmış bir personel tarafından görsel düzenleme programları ile yaklaşık 2-3 saat süren bir çalışma yapılması gerekir. Önerilen yaklaşımla otomatik bir bıyık üretim işlemi gerçekleştirilecek, manuel üretimdeki hatalar ve zamansal kayıplar elimine edilecektir. Yaptığımız literatür araştırması neticesinde denim ürün görsellerinin üretken ağlar ile üretilmesi ile ilgili farklı bir çalışma bulunmamaktadır. Bu durum yapılan çalışmanın akademik özgün değerini yükseltmektedir. Çalışmada kullanılan ÇÜA mimarileri Pix2Pix, CycleGAN, DiscoGAN ve AttentionGAN'dır. Her bir mimarinin denim2bıyık veri kümesindeki eğitim ve test verileri üzerinde bıyık deseni üretim başarımlı değerlendirilmesi ve maliyet analizi yapılmıştır. Yapılan çalışmalar sonucunda, bıyık desen görseli üretim hızı bir saniyenin altına düşerken, üretim doğruluğu %86 seviyelerine çıktığı görülmektedir.

**Anahtar kelimeler:** Çekişmeli üretici ağlar, Sentetik görüntü üretimi, Bıyık deseni, Denim kumaş.

---

## **Performance Comparison of Generative Adversarial Networks in Mustache Pattern Production**

### **Abstract**

In this study, performance analysis of generative adversarial network architectures that transform from image to image is made and its performance in synthetic image generation is evaluated. For a quality performance evaluation of these models, the denim2bıyık dataset collected from the real-world area was used instead of standardized datasets. Mustache patterns drawn on denim fabrics are created with a laser device. For this device to create the desired mustache pattern, it is necessary to work with visual editing programs for approximately 2-3 hours by specialized personnel. With the proposed approach, an automatic mustache production process will be realized, errors and time losses in manual production will be eliminated. As a result of our literature research, there is a no different study on the production of denim product images with productive networks. This situation increases the academic original value of the study. GAN architectures used in the study are Pix2Pix, CycleGAN, DiscoGAN, and AttentionGAN. Mustache pattern production performance evaluation and cost analysis were performed on the training and test data in the denim2bıyık dataset of each architecture. As a result of the studies, it is seen that the production speed of the mustache pattern image drops below one second, while the production accuracy reaches 86%.

**Keywords:** GANs, Synthetic image generation, Mustache pattern, Denim fabric.

---

---

\*Sorumlu yazar: [essahin950@gmail.com](mailto:essahin950@gmail.com)

Geliş Tarihi: 22.08.2021, Kabul Tarihi: 17.11.2021

## 1. Giriş

Teknolojinin hızlı gelişmesiyle birlikte yapay öğrenme alanındaki çalışmalar da ilerlemektedir. Günümüzde teknolojik yaşamın bir parçası haline gelen bu alan; dil çevirileri, konuşma botları, yüz tanıma, ses tanıma, hastalık teşhisi, veri iyileştirme, otonom araçlar ve veri üretimi gibi birçok alanda kullanılmaktadır. Bu alan birçok öğrenme yöntemi barındırmasına rağmen günümüzde en popüler olanı Derin öğrenme mimarileridir ve gün geçtikçe gelişerek popülerliğini arttırmaktadır [1].

Derin öğrenme olarak adlandırılan evrişimli öğrenme metodolojisi LeCun tarafından 1989'da önerilmiştir. Çalışmasında el yazısı ile yazılan posta kodlarını tanımlayan bir algoritma geliştirmiştir. Bu algoritmanın en önemli özelliği evrişimsel (konvölüsyon) katmanları içermesi ve geri beslemeli olarak eğitilebilmesidir [2].

Evrişimsel ağların çeşitlenmesiyle birlikte sentetik görüntü üretiminde devrimsel özellik taşıyan ÇÜA mimarisi 2014 yılında Goodfellow ve ekibi tarafından ortaya çıkmıştır [3]. Min-max algoritmasına dayanarak geliştirilen bu mimari, iki evrişimsel modelin karşıt çalışmasıyla öğrenmenin gerçekleştiği bir sistemdir [3].

ÇÜA mimarilerinin sentetik veri üretiminde kullanılmasıyla birlikte 2016 yılında Radford ve diğerleri tarafından DCGAN mimarisi geliştirilmiştir. Bir veri kümesindeki örneklere benzeyen sentetik görüntüler üretmek için basit bir gauss sinyalinin ters evrişim işlemlerinden geçirek görüntüye dönüştüren bir yapıya sahiptir [4].

Denetimli veri üzerinde eşleşmiş görüntü çiftleri arasında görüntüden görüntüye dönüşüm işlemini yapan ve koşullu çekişmeli bir ağ yapısına sahip olan Pix2Pix mimarisi 2016 yılında Isola ve ekibi tarafından önerilmiştir. Bu mimari ÇÜA maliyet fonksiyonuna ek olarak L1 metriği içermektedir [5].

2017 yılında Zhu ve diğerleri tarafından denetimsiz veri kümeleri üzerinde görüntüden görüntüye dönüşüm ve çift taraflı doğrulama işlemi yapan CycleGAN mimarisi geliştirilmiştir [6].

2018 yılında yüksek çözünürlüğe sahip kaliteli yapay görüntüler üretmek için Progressive GAN mimarisi Karras ve diğerleri tarafından geliştirilmiştir. Bu mimari, 4x4'lük görüntüler ile eğitime başladıktan sonra zamanla gerçekçi görüntüler üretmeyi öğrenerek 1024x1024'lük görüntüler sentezleyebilmektedir [7].

Huang ve diğerleri tarafından gerçek zamanlı stil aktarımı için 2017 yılında sinirsel stil aktarımının temellerini oluşturan bir mimari geliştirilmiştir. Bu mimari içerik ve stil görüntüsü ile beslenerek dönüşmüş görüntüyü elde etmektedir. Bu çalışmaya ilaveten 2018 yılında Karras ve diğerleri tarafından çekişmeli üretici ağlar için stil tabanlı bir üretici olan StyleGAN mimarisi geliştirilmiştir [8,9].

Wang ve diğerleri tarafından 2018 yılında anlamsal haritalar ile yüksek çözünürlüklü sentetik görüntü üretmek için Pix2Pix mimarisinin geliştirilmiş sürümü olan Pix2PixHD sürümü geliştirilmiştir [10].

Nvidia'da araştırmacılar tarafından anlamsal görüntü sentezlemek için 2019 yılında mekansal uyarlanabilir normalleştirme özelliğine sahip SPADE mimarisi geliştirilmiştir [11].

2020 yılında anlamsal görüntü sentezlemenin yetersiz kalmasından dolayı panoptik harita bazlı görüntü sentezlemek için Dundar ve diğerleri tarafından "Panoptic-based Image Synthesis" adlı çalışma sunulmuştur [12].

2021 yılında Karras ve diğerleri tarafından, 2018 yılında geliştirilen StyleGAN mimarisindeki öteleme ve döndürmeden kaynaklı gizli alandan gelen sinyal probleminin çözümü için bu mimariye Fourier tabanlı özellikler ve çeşitli değişimler dahil edilerek, StyleGAN3 mimarisi geliştirilmiştir [13].

Real-ESRGAN mimarisi 2021 yılında Wang ve diğerleri tarafından düşük kaliteli görüntüleri yüksek kalitede sentezleyebilmek için geliştirilmiştir. Bu mimarinin ana özelliği bir görüntü düşük çözünürlük, bulanıklık, sıkıştırma veya gürültü benzeri birden fazla problem alanı içerse bile aynı anda birden fazla hedefe odaklanabilir olmasıdır [14].

Bir görüntüdeki istenmeyen nesnelere veya bölgelerin, görüntünün ana akışını bozmadan, kaldırılmış imajını sentezleyebilmek için Suvorov ve diğerleri tarafından 2021 yılında Fourier konvölüsyon tabanlı LaMa mimarisi geliştirilmiştir [15].

İnsan yüzüne ait eski veya kalitesiz görsellerin yüksek kalitede gürültüsüz ve renkli olarak sentetik bir şekilde üretilebilmesi için 2021 yılında Wang ve diğerleri tarafından uzamsal özellik dönüşümlü GFP-GAN mimarisi geliştirilmiştir [16].

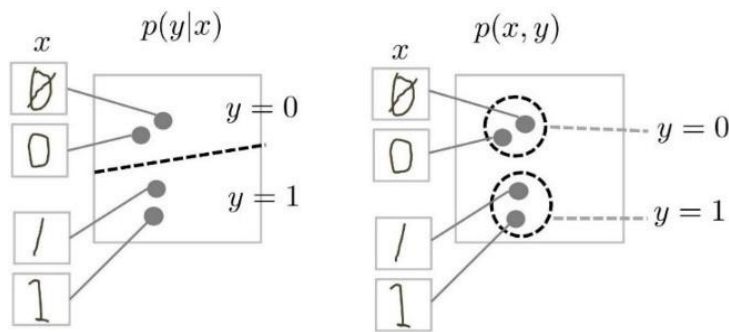
Bu çalışmada, ÇÜA'lar tekstil alanındaki bir problemin çözümünde kullanılmaktadır. Bu problem, denim kot kumaşları üzerindeki çizgi desen modelinin (bıyık) çıkarılmasıdır. Bıyık çıkarımı olarak bilinen bu işlem müşterinin örnek bir kot numunesi vermesiyle başlar. Örnek numune kotlar fotoğraflandıktan sonra uzman bir personel Photoshop gibi bir program kullanarak 2-3 saat süren bir çalışmayla bıyık motiflerini çıkarır. Klasik yaklaşım işletme maliyetlerini arttırmakta, üretim sürecinin uzamasına neden olmaktadır. Önerilen yaklaşımda, müşterilerden temin edilen denim kumaş numunelerindeki bıyık desenlerinin otomatik üretilmesi söz konusudur. Bunun için görüntüden görüntüye çeviri yapan ÇÜA mimarileri kullanılmaktadır. Bu çalışmada görüntü sentezleme alanındaki ÇÜA mimarileri detaylı incelenmekte ve performansları karşılaştırılmaktadır. Bölüm 2.1'de ÇÜA mimarilerinin genel yapısı hakkında bilgilendirme yapılmıştır. Çalışmada kullanılan mimarilerin sırasıyla Pix2Pix, CycleGAN, DiscoGAN ve AttentionGAN denim2bıyık veri kümesi üzerine uygulanması ve temel yapılarının incelenmesi bölüm 2.2-2.5'e kadar olan kısımlarda anlatılmıştır. Bölüm 2.6'da veri kümesinin oluşturulması hakkında bilgi verilmiştir. Bölüm 2.7'de sonuçların karşılaştırmasında kullanılan ölçüm metrikleri hakkında bilgi verilmiştir. Bölüm 3'de bu çalışmada elde edilen çıktılar ve bu çıktılarının değerlendirilmesi ile ilgili sonuçlar verilmektedir.

## 2. Materyal ve Metot

Bu bölümde çekişmeli üretici ağlar ve görüntüden görüntüye dönüşüm işlemi yapan mimarilerin temel yapısı anlatılmaktadır.

### 2.1. Çekişmeli Üretici Ağlar

Çekişmeli üretici ağlar gerçek dünyadaki veri kümeleri ile karmaşık örüntüleri öğrenen enerji bazlı bir mimaridir. Çekişmeli üretici ağların temel yapısı, ünlü matematikçi John Nash'in oyun teorisini baz alır ve birbirleriyle karşıt çalışarak öğrenen, iki evrişimli modülden oluşur. Bu modüllerden biri üretici (generator) ağ, diğeri ise ayırıcı (discriminator) ya da maliyet (cost) ağı olarak adlandırılır. Üretici ağ, gittikçe gerçeğe daha fazla benzeyen yapay (sahte) görüntüler sentezlemeye çalışan bir üreticidir, ayırıcı ağ da gerçek ve sahte görüntüleri birbirinden ayırmak isteyen basit bir ikili sınıflandırıcıdır. Üretici, ayırıcıyı kandırabilmek için sahte görüntüler oluştururken ayırıcı da zamanla öğrenerek sahte ile gerçek görüntü arasında aldanmamaya çalışır [3,17,18]. Ayırıcı ve üretici modüllerinin örnek veri üzerindeki çalışma mekanizması **Şekil 1**'de gösterilmektedir. Bu şemada ki sembollerden,  $X$  veri örneklerini,  $Y$  örnek çıktılarının gerçek veya sahte olduğunu,  $\mathcal{P}(X, Y)$  ortak olasılık yoğunluk fonksiyonunu,  $\mathcal{P}(X|Y)$  koşullu olasılık dağılımını belirtmektedir.

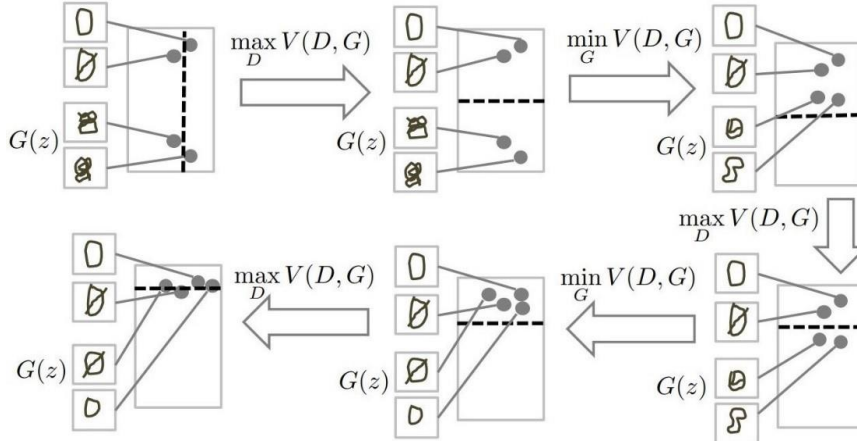


**Şekil 1.** ÇÜA mimarisi (1. sütun ayırıcı modülü, 2. sütun üretici modülü) [19]

Buna göre **Şekil 1**'in 1. sütunundaki ayırıcı ağ günlük hayatta kullanılan gerçek veriyi ( $Y = 1$ ) ile üretici ağdan elde edilen sentetik görüntünün ( $Y = 0$ ) bir düzlem üzerindeki izdüşümünü en iyi şekilde 1 ya da 0 olacak şekilde sınıflandırmaya çalışarak maliyeti yükseltmek ister, 2. sütundaki üretici ağ ise düzlem üzerinde gerçek ve üretilen veri kümeleri arasında ki mesafeyi azaltmak için gerçek görüntülere benzer görüntüler üretmeye çalışarak maliyeti azaltmak ister. Bu işlem bir oyundaki iki rakibin birbirini yenmek için rakip tarafın yapabileceği en iyi (max) hamleyi düşünerek hasarı en aza (min) indirmeye çalışmak istemesidir. Literatürde bu yöntem min-max algoritması olarak geçmektedir. Çekişmeli üretici

ağlar da matematiksel olarak min-max yöntemini kullanarak eğitimi gerçekleştirir [3,19]. Bu yöntemin eğitim işleminde aşamalı olarak gösterimi **Şekil 2'**de verilmiştir.

Bu şemada ilk başta iki gerçek ve iki üretici ağdan ( $G(z)$ ) elde edilmiş  $z$  olarak adlandırılan rastgele üretilmiş ya da bir veriye ait öznelik barındıran örneklerin izdüşümü düzlem üzerine aktarılmaktadır. Çekişmeli üretici ağın ana fonksiyonu  $V$  olarak tanımlanır. Ayırıcı ağ ( $D$ ) gerçek ve sentetik verilerin arasına bir set çekerek maliyeti maximize etmektedir. Ardından üretici ağ ( $G$ ) sentetik verilerin izdüşümünü gerçek verilerin izdüşümüne yakınlaştırmaya çalışarak maliyeti minimize etmektedir. Bu süreç ayırıcı ağın üretici ağdan gelen yapay görüntü ile gerçek görüntüyü ayırt edemeyecek duruma gelmesine kadar devam eder.



**Şekil 2.** Üretici ve ayırıcı ağın min-max yöntemiyle aşamalı öğrenimi [19]

Literatürdeki GAN mimarileri incelendiğinde sinyal-sinyal, sinyal-görüntü ve görüntü-görüntü dönüşümünü yapan farklı mimarilerin olduğu görülmektedir [17]. Bu çalışmada örnek denim kumaş görselinden bıyık desen görselinin elde edilmesi amaçlandığı için görüntüden görüntü üretebilen mimarilere odaklanılmıştır.

## 2.2. Pix2Pix

Pix2Pix, görüntüden görüntüye dönüşüm işlemi yapan bir mimaridir. Bu mimari, 2016 yılında Berkeley'den araştırmacılar tarafından geliştirilmiştir. Görüntü işleme ve bilgisayarlı görü alanlarındaki sorunların çoğu, bir giriş görüntüsüne karşılık gelen bir çıktı görüntüsü şeklindedir. Örneğin, bir siyah beyaz görüntünün renklendirilmesi, kenar haritasından anlamsal etiket haritasının elde edilmesi gibi işlemlerde bir görüntü alanından diğer alana dönüşüm işlemi yapmak zordur. Bu tarz problemlerin çözümünde Pix2Pix mimarisi ile önemli bir adım atıldı. Pix2Pix, koşullu (condition) GAN mimarilerinin özel bir versiyonudur. Bu model koşullu GAN mimarisine ek olarak  $L_1$  maliyet değeri içermektedir.  $L_1$  metriği üretilen görüntü ile hedef görüntü arasında benzerlik karşılaştırmasıdır [5].

Klasik koşullu GAN mimarilerindeki maliyet değeri Denklem (1) de ifade edilmektedir. Denklemdeki  $x$  girdisi  $y$  çıktısına bağlıdır. Ayırıcı ağ gerçek  $x$  ve  $y$  girdilerin öznelik çıktısı ile gerçek  $x$  ve  $z$  özneliğinin üretici ağa verilerek elde edilen  $G(x, z)$  çıktısı arasındaki maliyeti hesaplar.

$$\mathcal{L}_{CGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log D(x, G(x, z))] \quad (1)$$

Buna göre üretici  $G$  ağı bu amaç fonksiyonunu minimize etmek isterken, ayırıcı  $D$  ağı maksimize etmek ister. Böylece optimal üreticinin maliyeti Denklem (2) gibi tanımlanır:

$$\arg \min_G \max_D \mathcal{L}_{CGAN}(G, D) \quad (2)$$

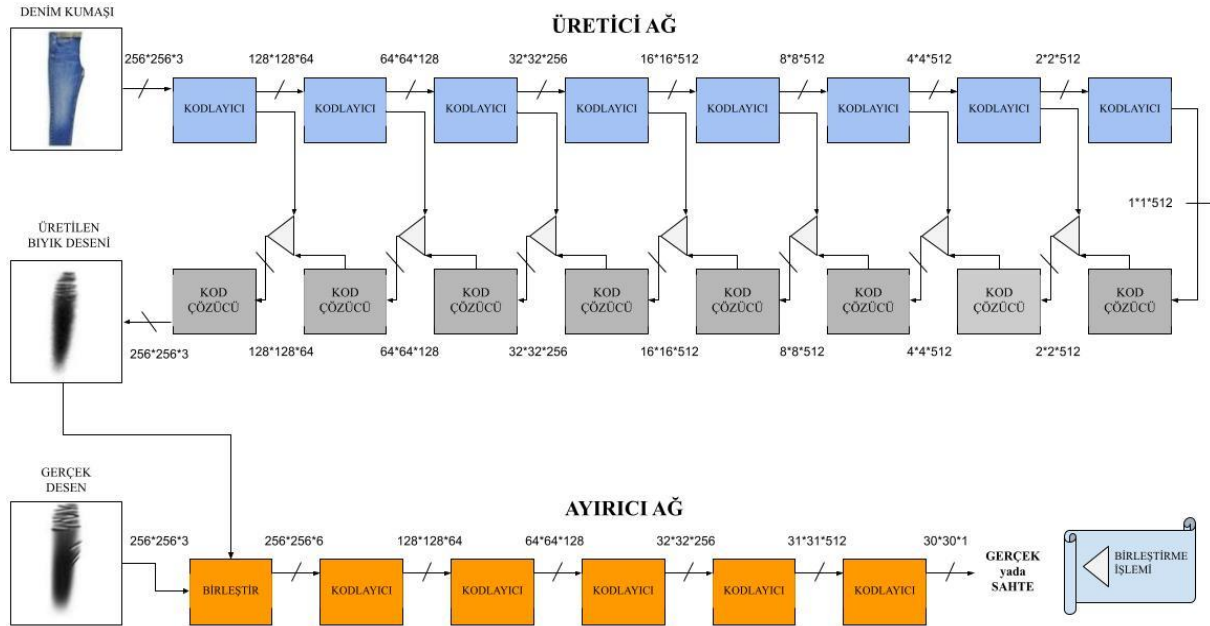
Denklem (1) de ifade edilen maliyet fonksiyonuna birde gerçek görüntü ile üretilen sahte görüntü arasındaki benzerlik maliyeti eklenerek Denklem (3) gibi ifade edilir. Bu maliyet değeri  $L_1$  uzaklık metriği olarak adlandırılan ortalama mutlak hatadır (MAE) [20].

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z} [\|y - G(x, z)\|_1] \quad (3)$$

$L_1$  maliyetinin eklenmesi sonucunda Pix2Pix mimarisinin maliyet fonksiyonu Denklem (4) gibi tanımlanır. Denklemdeki ana maliyet klasik ÇÜA maliyet sonucu ve  $L_1$  değerinin  $\lambda = 100$  değeri ile çarpılıp eklenmesiyle hesaplanır.

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \quad (4)$$

Pix2Pix modeli üretici ağ olarak U-Net [21] mimarisi kullanmaktadır. Bu model  $256 \times 256 \times 3$  ebatlarında renkli görüntü almaktadır. Giriş görüntüleri kodlayıcı (Encode) bloklardan geçerek  $1 \times 1 \times 512$  ebatlarındaki öznelik haritası vektörüne dönüşür. Kodlayıcı bloklar evrişim (Conv2d), Leaky Relu ve Batch Normalization katmanlarını içermektedir. Elde edilen öznelik vektörü ve her kodlayıcı bloğun çıktısı kendi karşısındaki kod çözücü (Decode) blokların çıktısı ile birleştirilerek üst kod çözücü bloğa giriş olarak verilir. Bu işlemle dönüşmüş görüntü elde edilir. Kod çözücü bloklar kodlayıcı bloklardan farklı olarak aktivasyon fonksiyonunda Relu kullanmaktadır. Ayırıcı ağ gerçek ve üretilen sentetik görüntüleri kodlayıcı bloklarda işleyerek  $30 \times 30 \times 1$  ebatlarına dönüştürür. Elde edilen bu öznelik haritası gerçek veya sahte olarak sınıflandırılmaya çalışır [5]. Pix2Pix mimarisi ile denim kumaşından bıyık desen üretim işleminin şeması Şekil 3'de gösterilmiştir.



Şekil 3. Pix2Pix mimarisinin bıyık desen çıkarımında kullanımı

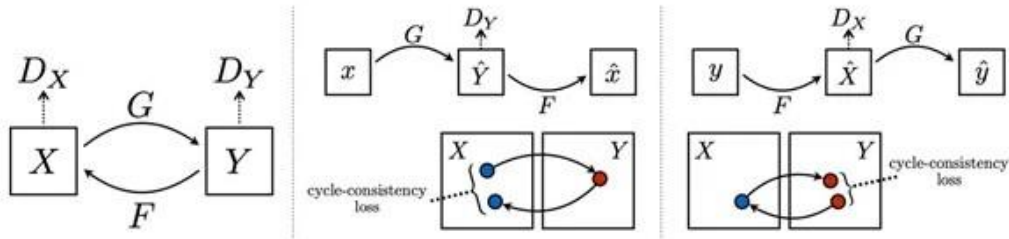
### 2.3. CycleGAN

CycleGAN mimarisi eşleştirilmiş görüntü çiftlerine ihtiyaç duymadan bir görüntünün X alanından Y alanına dönüşüm yapmasını sağlar. Bu mimari 2016 yılında yayımlanan Pix2Pix mimarisinde yapılan güncellemeler sonucunda 2017 yılında geliştirilmiştir. Pix2Pix mimarisi eşleştirilmiş görüntü çiftlerine ihtiyaç duyarken CycleGAN mimarisiyle bu gereksinim ortadan kaldırılmıştır. Başka bir deyişle, kaynak ve hedef alan arasında bire bir eşleştirme olmadan bir alandan diğerine dönüşüm yapılabilir. CycleGAN mimarisi bir görüntü kümesinden diğer görüntü kümesine dönüşüm işlemi için döngü tutarlılık kaybı kullanır. Genel yapısı Pix2Pix mimarisine benzerdir ve aynı zamanda aynı üretici ağ mimarisini kullanmaktadır. Pix2Pix mimarisi bir üretici ve ayırıcı ağ yapısı kullanırken CycleGAN mimarisi iki üretici ve iki ayırıcı ağ yapısı kullanmaktadır [5,6].

Mimaride eğitimde kullanılan iki üretici ( $G$  ve  $F$ ) ve iki ayırıcı ( $D_X$  ve  $D_Y$ ) ağı çalıştırma mekanizması:

- Üretici  $G$  ağı  $X$  alanındaki görüntüyü  $Y$  alanındaki görüntüye dönüştürmeyi öğrenir. ( $G: X \rightarrow Y$ )
- Üretici  $F$  ağı  $Y$  alanındaki görüntüyü  $X$  alanındaki görüntüye dönüştürmeyi öğrenir. ( $F: Y \rightarrow X$ )
- Ayırıcı  $D_X$  ağı,  $X$  alanındaki gerçek görüntü ile  $F$  ağının sonucunda üretilen  $\hat{X}(F(Y))$  görüntüsünü gerçek ya da sahte olarak sınıflandırmaya çalışır.
- Ayırıcı  $D_Y$  ağı,  $Y$  alanındaki gerçek görüntü ile  $G$  ağının sonucunda üretilen  $\hat{Y}(G(X))$  görüntüsünü gerçek ya da sahte olarak sınıflandırmaya çalışır.

Bu adımların görsel şeması Şekil 4’de verilmiştir.



Şekil 4. CycleGAN mimarisinin ağ ve maliyet hesaplama şeması [6]

CycleGAN mimarisinde gerçek görüntü ile yeniden inşa edilen (reconstructed) görüntü arasındaki benzerlik maliyeti için döngü tutarlılık kaybı (cycle consistency loss) kullanılır. Bu maliyet işlemi Denklem (5) gibi ifade edilir. Denklem de  $X$  ile yeniden inşa edilen  $\hat{X}$  ve  $Y$  ile yeniden inşa edilen  $\hat{Y}$  arasında  $\mathcal{L}_{cyc}$  olarak adlandırılan ortalama karesel hata (MSE) [22] metriği ile maliyet hesaplanır.

$$\mathcal{L}_{cyc} = |F(G(X)) - X| + |G(F(Y)) - Y| \quad (5)$$

Mimarideki iki üretici ( $G, F$ ) ve iki ayırıcı ( $D_X, D_Y$ ) ağın maliyetleri Denklem (1)’deki fonksiyon ile hesaplanır ve ek olarak döngü tutarlılık kaybı olduğu için toplam maliyeti üç ayrı kayıptan oluşur. Genel maliyet Denklem (6) gibi tanımlanır. Bu denklemdeki  $\lambda$  değeri 10 alınarak hesaplama yapılır.

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{cGAN}(G, D_Y, X, Y) + \mathcal{L}_{cGAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F) \quad (6)$$

Optimizasyonun bir parçası olarak, Denklem (7)’deki kayıp işlevi optimize edilir.

$$G^*, F^* = \arg \min_{G, F} \max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y) \quad (7)$$

Üretici  $G, F$  mimarilerinin kayıp değeri minimize etmeye çalışırken, ayırıcı  $D_X, D_Y$  ağlarını maksimize etmeye çalışır. Üretici ağların maliyet değerinin düşmesi ağın daha gerçekçi görüntüler üretmesini sağlar ve ayırıcı ağlardaki maliyetin artması üretilen sentetik görüntü ile gerçek görüntü arasındaki ayırım yapmasını zorlaştırır. Bu maliyetin artması ayırıcı ağın gerçek ve sentetik görüntüyü sınıflandırmaya çalışırken daha dikkatli olmasını sağlar. Böylece üretici ve ayırıcı ağlar birbirini besleyerek giriş görüntülerine en uygun dönüşüm görüntüsünü elde etmeye çalışır.

## 2.4. DiscoGAN

DiscoGAN mimarisi birbirleri ile bağlantısı olmayan veri kümeleri arasındaki ilişkiyi keşfetmek için 2017 yılında geliştirilmiş koşullu GAN mimarisidir. Denetimsiz veriler üzerinde çalışan DiscoGAN mimarisinin temel konsepti CycleGAN mimarisine benzer: Her ikisi de iki ayrı dönüşüm işlevini öğrenir, biri  $X$  alanından  $Y$  alanına bir dönüşümü öğrenirken diğeri ters bir eşleşmeyi öğrenir. CycleGAN gibi, DiscoGAN da yeniden yapılandırma kaybı temeli üzerine inşa edilmiştir [6,23]. Buradaki fikir, bir görüntü bir alandan diğerine ve ardından orijinal alana geri

dönüştürüldüğünde, oluşturulan görüntünün orijinali kadar benzer olması gerektiğidir. Bu durumda nicel fark, yeniden yapılandırma kaybı olarak değerlendirilir ve eğitim sırasında model bu kaybı en aza indirmeye çalışır. İki mimari arasındaki temel fark, DiscoGAN mimarisinin iki yeniden yapılandırma kaybı kullanmasıdır CycleGAN mimarisi tek yeniden yapılandırma kaybını belirli bir  $\lambda$  parametresi ile çarparak eğitimi gerçekleştirirken; DiscoGAN mimarisi iki yapılandırma kaybının tamamını kullanarak işlemi gerçekleştirir [23].

DiscoGAN mimarisinin yeniden yapılandırma kaybı Denklem (8) gibi tanımlanır. DiscoGAN mimarisi CycleGAN mimarisi gibi döngü tutarlılık kaybı için ortalama karesel hata (MSE) metriğini kullanır. Bu işlem giriş görüntüsü  $X_A$  ile iki üretici ( $G_{BA}, G_{AB}$ ) ağdan geçtikten sonra elde edilen yeniden inşa edilen görüntü arasında hesaplanır.

$$\mathcal{L}_{CONST_A} = |G_{BA}(G_{AB}(X_A)) - X_A| \quad (8)$$

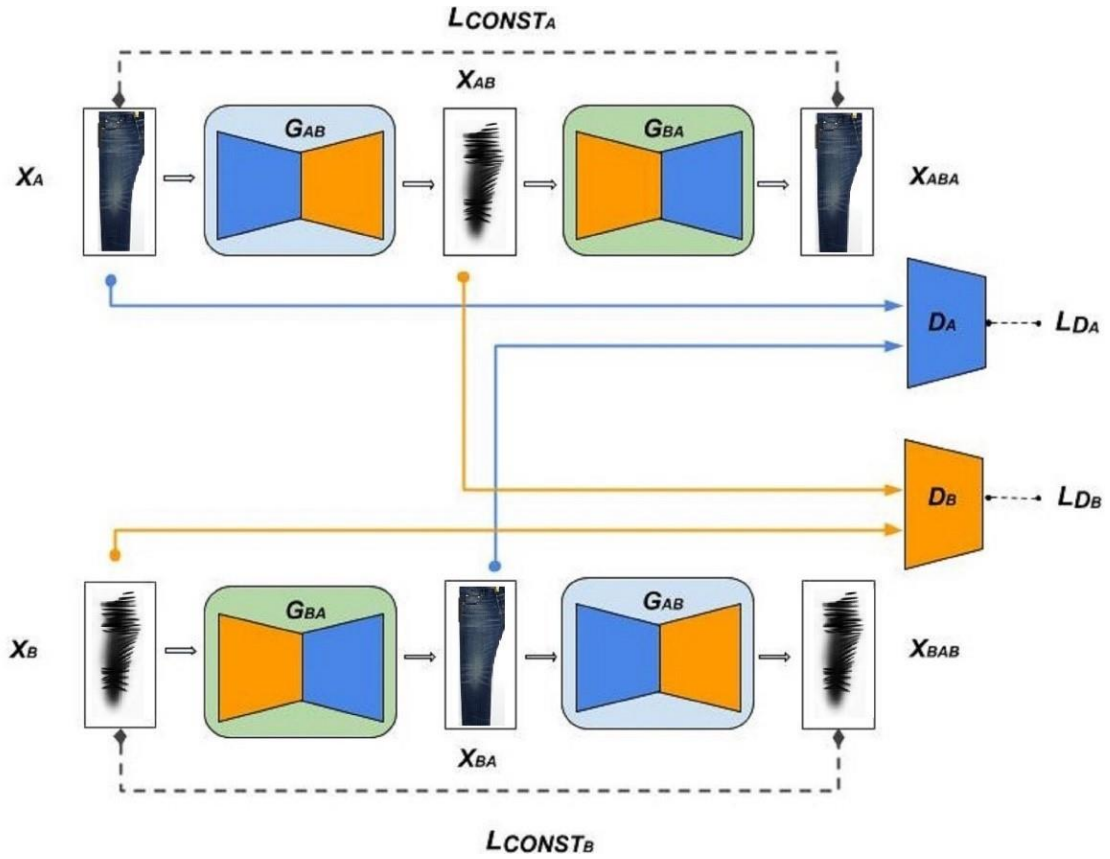
Mimarinin bir alandaki eğitim işlemi için bir üretici ( $G_{AB}$ ) ve ayırıcı ( $D_B$ ) ağın maliyeti Denklem (1)'deki maliyet fonksiyonunun sonucu ile yeniden inşa edilen görüntü ile orijinali arasındaki döngü tutarlılık kaybının toplamıdır. Bu maliyet fonksiyonu Denklem (9) gibi ifade edilir.

$$\mathcal{L}_{G_{AB}} = \mathcal{L}_{CGAN(G_{AB}, D_B)} + \mathcal{L}_{CONST_A} \quad (9)$$

İki üretici ve iki ayırıcı ağ için toplam maliyet fonksiyonu Denklem (10) gibi tanımlanır.

$$\mathcal{L}(G_{AB}, G_{BA}, D_A, D_B) = \mathcal{L}_{G_{AB}} + \mathcal{L}_{G_{BA}} \quad (10)$$

Dolayısıyla model,  $G_{AB}$  ve  $G_{BA}$  üretici ağları ve  $D_A$  ve  $D_B$  ayırıcı ağlardan oluşmaktadır. DiscoGAN mimarisinin çalışma prensibinin örnek denim kumaşından büyük desen üretiminde kullanımı Şekil 5'deki görselde verilmiştir. Bu görselde alanlar arası ilişkiyi yönleri açısından öğrenmeye çalışır. Bir görüntünün yeniden yapılandırılmasından sonra, yön orijinal görüntünün yönüyle aynı olmalıdır. Şemada ki  $L$  sembolü maliyet fonksiyonu göstergesidir.



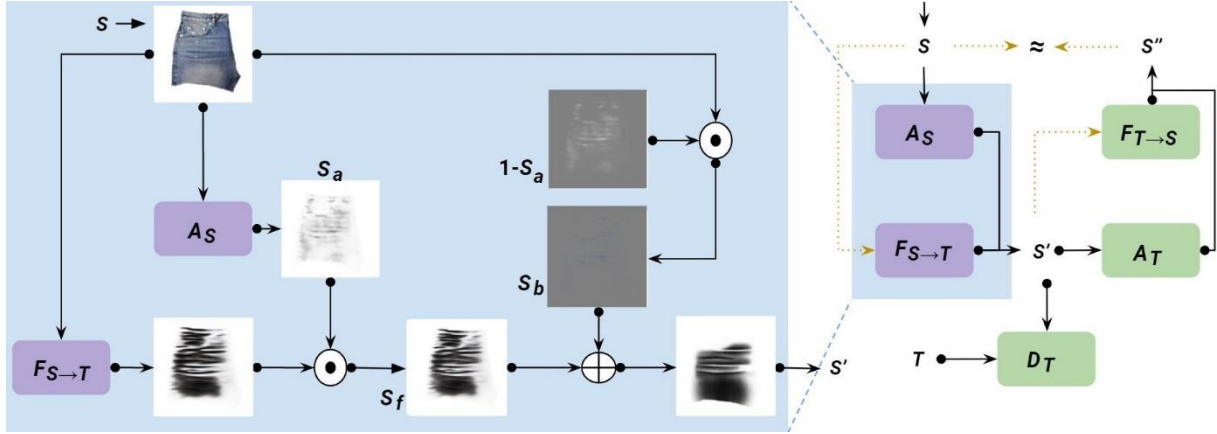
Şekil 5. DiscoGAN mimarisinin büyük desen çıkarımında kullanımı

## 2.5. AttentionGAN

AttentionGAN denetimsiz veri üzerinde görüntüden görüntüye dönüşüm sağlayan dikkat güdümlü bir mimaridir. Başlangıç noktası olarak CycleGAN mimarisi referans alınıp sıfır ile bir aralığında değerler içeren dikkat haritaları oluşturacak şekilde dikkat ağına mimariye entegrasyonu ile AttentionGAN mimarisi elde edildi [6,24]. Klasik görüntüden görüntüye çeviri yapan mimarilerin üretici ve ayırıcı ağları sadece kodlayıcı ve kod çözücü yapısını barındırmasından dolayı görüntü üzerinde belirli bir alana dikkat yoğunlaştırmaları zordur. Bu yüzden görüntünün odaklanması gereken bölümüne yeterli önem verilememektedir. AttentionGAN mimarisi görüntü içeriğini üretici ağ yapısıyla iletirken aynı zamanda görüntüde odaklanması gereken kısımların dikkat haritasını da çıkarıp ikisini birleştirir [24]. Böylece ağ değişiminin olduğu yerlere odaklanıp arka plan veya ön plan içeriğine gereğinden fazla odaklanmasını engeller. Bu mimari dikkat haritalarından dolayı eşleşmemiş veri kümelerindeki görüntü dönüşümlerinin daha gerçekçi olmasını sağlar.

AttentionGAN mimarisinin akış diyagramı Şekil 6'da gösterilmiştir. Mimari;  $A_S$ ,  $A_T$  dikkat,  $F_{S \rightarrow T}$ ,  $F_{T \rightarrow S}$  üretici ve  $D_S$ ,  $D_T$  ayırıcı ağlarından oluşmaktadır. Giriş görüntüsü ( $S$ ), bu görüntünün dönüştürülmek istendiği görüntü ( $T$ ) olarak ifade edilir. Giriş görüntüsü üretici  $F_{S \rightarrow T}$  ve dikkat  $A_S$  ağına verilerek elde edilen dikkat haritası ( $S_a$ ) ve içerik görüntüsü 3 kanalda (RGB) noktasal olarak çarpılır. Elde edilen çıktı  $S_f$  ön plan (foreground) görüntüsü olarak adlandırılır. Arka plan (background) içeriği içinde  $S$  giriş görüntüsü ve  $S_a$  dikkat haritasının tersi alınarak noktasal olarak çarpılır. Ön plan ve arka plan görüntüleri toplanarak giriş görüntüsünün dönüşmüş hali  $S'$  elde edilir. Çift taraflı doğrulama işlemi için  $S'$  görüntüsü  $F_{T \rightarrow S}$  ve  $A_T$  ağına verilerek aynı işlem tekrar edilir. Elde edilen  $S''$  görüntüsü giriş görüntüsünün yeniden inşa edilmiş imajıdır. Son olarak  $S$  ve  $S''$  görüntüsü arasında döngü tutarlılık maliyeti hesaplanır [24].





Şekil 6. AttentionGAN mimarisinin çalışma prensibi

AttentionGAN, CycleGAN mimarisi gibi gerçek görüntü ile yeniden inşa edilen görüntü arasındaki benzerlik maliyeti için döngü tutarlılık kaybı kullanılır. Bu maliyet fonksiyonu Denklem (11) gibi tanımlanır. Denklemdeki  $S$  ifadesi giriş görüntüsü ve  $S''$  ifadesi de yeniden inşa edilen giriş görüntüsüdür.

$$\mathcal{L}_{cyc}^S(S, S'') = \|S - S''\| \quad (11)$$

Kaynak ve hedef alan için toplam maliyet fonksiyonu Denklem (12)'de  $\lambda_{cyc} = 10$  alınarak hesaplanır.

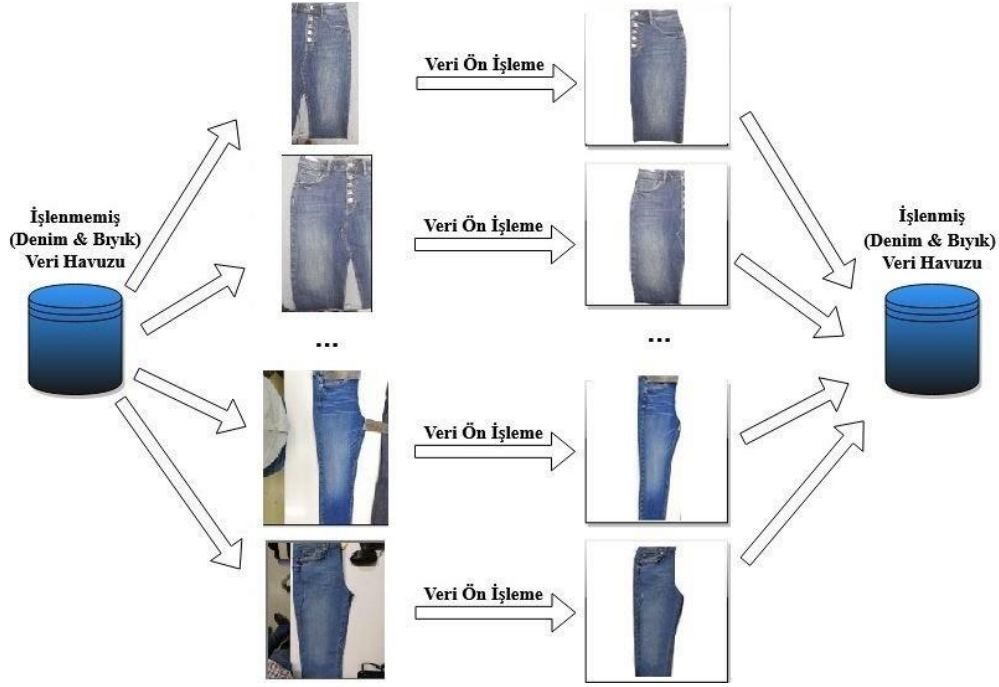
$$\mathcal{L}(F_{S \rightarrow T}, F_{T \rightarrow S}, A_S, A_T, D_S, D_T) = \mathcal{L}_{cGAN}^S(F_{S \rightarrow T}, A_S, D_S) + \mathcal{L}_{cGAN}^T(F_{T \rightarrow S}, A_T, D_S) + \lambda_{cyc}(\mathcal{L}_{cyc}^S + \mathcal{L}_{cyc}^T) \quad (12)$$

Optimizasyonun bir parçası olarak Denklem (13)'de maliyet değeri optimize edilmiş olur.

$$F_{S \rightarrow T}^*, F_{T \rightarrow S}^*, A_S^*, A_T^*, D_S^*, D_T^* = \underset{F_{S \rightarrow T}, F_{T \rightarrow S}, A_S, A_T}{\operatorname{argmin}} \left( \underset{D_S, D_T}{\operatorname{argmax}} \mathcal{L}(F_{S \rightarrow T}, F_{T \rightarrow S}, A_S, A_T, D_S, D_T) \right) \quad (13)$$

## 2.6. Denim2bıyık Veri Kümesi

Bu çalışma da kullanılan veri kümesi, Baykan Denim Konfeksiyon A.Ş.'nin Malatya'daki üretim fabrikasından elde edilmiştir. Veri kümesinin inşasında, firmanın şimdiye kadar elde ettiği denim kumaş numune görselleri ve bunlara bakılarak görsel düzenleme programları ile manuel olarak çizilen bıyık görüntüleri kullanılmıştır. Toplamda 950 adet denim2bıyık görüntü ikilisi elde edilmiştir. Bu görüntülerin 900 tanesi eğitim 50 tanesi test işlemi için ayrılmıştır. Veri kümesi, pantolon, etek ve şort olmak üzere üç farklı denim kumaş türünün ön yüz, arka yüz ve paça gibi farklı bölgelerine ait görseller içermektedir. Firma çalışanlarının cep telefonlarıyla elde edilen bu görüntülerinin standart bir forma sahip olmadığı görülmüştür. Daha açık bir ifadeyle, kumaş görsellerinin arkaplan görünümü, ışık dağılımları, boyutları ve perspektiflerinin farklı olduğu görülmüştür. Verinin standart bir forma dönüştürülebilmesi için her bir denim kumaş görseli üzerinde PhotoShop programı ile manuel bir temizleme çalışması yapılmıştır. Bu işlemin görsel şeması Şekil 7'de gösterilmektedir.



Şekil 7. Denim2bıyık veri kümesi ön işleme şeması

Veri kümesindeki her bir görüntü çiftinin ön plan nesnesi (kumaş görüntüsü) arka plandan ayıklanmış ve görüntü merkezde olacak şekilde beyaz bir zemin üzerine aktarılmıştır. Benzer işlem bıyık desenlerinde de yapılarak girdi ve çıktı boyutlarının benzer olması (256x256x3) sağlanmıştır. Ayrıca görseller üzerinde normalizasyon yapılarak ışık dağılımından kaynaklanan farklılıkların elimine edilmesi sağlanmıştır. Ön işlemler sonucu veri kümesinde yer alan birkaç denim kumaş görseli ve bunlara karşılık elle hazırlanan bıyık desen görüntüleri Şekil 8'de gösterilmektedir.



Şekil 8. Ön işlemler sonucu elde edilen denim2bıyık veri kümesindeki görüntü örnekleri

## 2.7. Ölçüm Metrikleri

Bu bölümde gerçek (Ground True) ve üretilen sentetik görüntülerin karşılaştırmasında kullanılan ölçüm metrikleri anlatılmaktadır.

### 2.7.1. SSIM

Yapısal benzerlik metriği (SSIM) parlaklık, maskeleme ve kontrast gibi önemli algısal özelliklerle birlikte görüntüde algılanan değişikliği de analiz eden bir metriktir. Bu metrik görüntüdeki pikseller arasında bulunan konum farklılığını göz ardı ederek, görüntü çiftleri arasındaki benzerliğin hesaplanmasında ortalama ( $\mu$ ) ve standart sapma ( $\sigma$ ) parametreleri üzerine inşa edilen istatistiksel bir ölçümdür [25].

İstatistiksel olarak gerçek ( $x$ ) ve yapay ( $y$ ) görüntüsü arasındaki yapısal benzerlik hesaplaması Denklem 14'te gösterilmektedir. Bu denklemdeki gerçek ve yapay görüntünün  $\mu_x$  ve  $\mu_y$  piksel

ortalamasını,  $\sigma_x^2$  ve  $\sigma_y^2$  varyansını belirtirken  $\sigma_{xy}$  gerçek ve yapay görüntü arasındaki kovaryansı belirtmektedir. Ek olarak  $c_1 = (\mathbf{k}_1\mathbf{L})^2$  ve  $c_2 = (\mathbf{k}_2\mathbf{L})^2$  sabit değerleri, piksel aralığını belirten  $L$  değeri 255. alınarak hesaplanır.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (14)$$

### 2.7.2. MSE

İki veri örneği arasındaki benzerlik maliyetini hesaplamak için noktasal farkın karesel ortalamasının alındığı bir karşılaştırma metriğidir [22]. Bu metriğin ana formülü Denklem 15'te gösterilmiştir. Denklem de gerçek ( $x$ ) ve yapay ( $y$ ) görüntüsü arasındaki her bir piksel farkının karesi alınıp toplanır. Toplam sonuç piksel sayısına ( $n$ ) bölünür.

$$MSE(x, y) = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad (15)$$

Ölçüm değerinin ne kadar düşük olduğu yapay görüntü ve gerçek arasındaki benzerliğin o kadar yüksek olduğunu gösterir.

### 2.7.3. PSNR

Bir sinyalin olası maksimum gücü ile temsilinin doğruluğunu etkileyen bozucu gürültünün gücü arasındaki oran için logaritmik tabanlı kullanılan bir metriktir [26]. Sinyal gerçek görüntü veya veri olarak kabul edilir, gürültü ise verideki sıkıştırma veya bozulmadan kaynaklanan hatadır. İki görüntü arasındaki bu oran desibel olarak hesaplanır. Bu metriğin ana formülü Denklem 16'da gösterilmektedir. Denklemdaki gerçek ( $x$ ) ve yapay ( $y$ ) görüntüsü arasındaki ölçüm için  $MSE(x, y)$  değeri ve gerçek görüntünün en büyük piksel değeri ( $MAX_x^2$ ) hesaplamaya dahil edilir.

$$PSNR(x, y) = 10 * \log_{10} \left( \frac{MAX_x^2}{MSE} \right) \quad (16)$$

Ölçüm değerinin ne kadar yüksek olduğu sinyal kalitesinin o kadar iyi olduğunu gösterir, yani gerçek ile yapay veri arasındaki maliyetin düşük olduğunu gösterir.

## 3. Bulgular ve Tartışma

Çekişmeli üretici ağ modellerinin performans karşılaştırması denim kumaşından sentetik bıyık deseni üretim veri kümesi üzerinde yapılmıştır. Bu çalışmada bölüm 2'de anlatılan 4 farklı mimarinin eğitim ve test işlemleri PyTorch derin öğrenme kütüphanesiyle RTX 2080 ekran kartına sahip bir sunucu üzerinde 150 iterasyon (epoch) boyunca part boyutu (batch size) 1 alınarak gerçekleştirilmiştir. Mimarielerin temel yazılımsal kodları github: “<https://github.com/hee9joon/Image-to-Image-Translation>” adresli kaynaktan alınmıştır. Bu kaynaktan görüntüden görüntüye dönüşüm yapan temel mimarilerin kodları bulunmaktadır. Temel kodların üzerinde hiperparametre, iterasyon (epoch), görüntü boyutları gibi ayarlamalar ve düzenlemeler yapılmıştır. Böylece bu modellerin uygun koşullar altında başarımlarını performans karşılaştırması için gerekli ortam hazırlanmıştır. Başarımlarını performans kriteri olarak görüntü üretim kalitesi ve bir adet görüntünün üretimi için gereken zamansal maliyet hesaplamaları yapılmıştır. Sentetik bıyık desen üretim kalitesi, test kümesindeki görüntü çıktılarının 3 farklı metrik ile karşılaştırılmasıyla elde edilmiştir. Gerçek ve üretilen görüntü arasındaki benzerliği tam değerlendirebilmek için SSIM metriği ile görüntülerin yapısal benzerliği, PSNR yöntemi ile görüntü piksellerindeki maksimum sinyal gücü ve MSE yöntemi ile iki görüntü arasındaki piksel farklılık

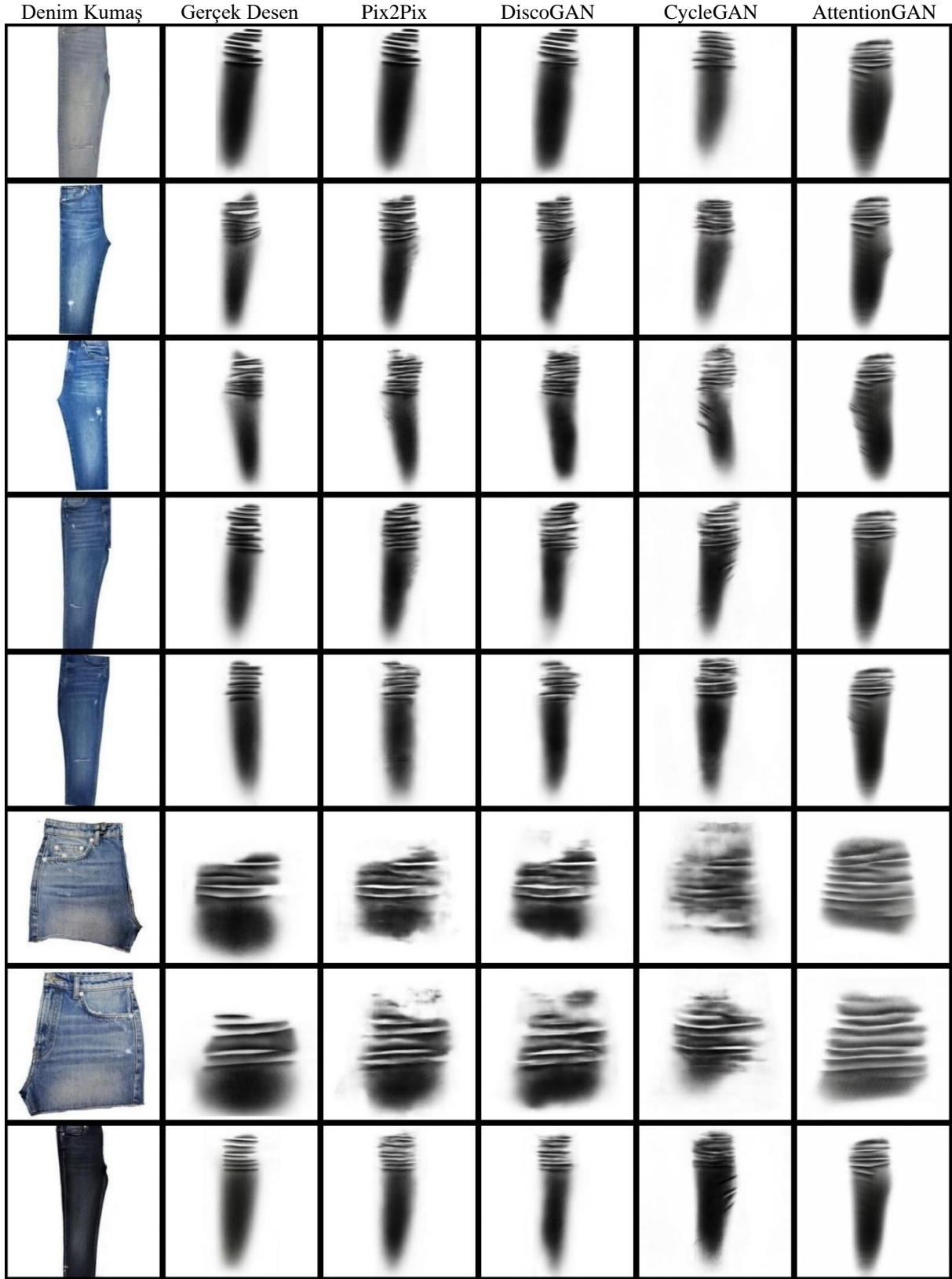
maliyeti ölçülmüştür. Modellerin test veri kümesinde bulunan 50 görüntü çifti üzerindeki başarımların performansları **Tablo 1**'de gösterilmektedir.

Modellerin sonuçları incelendiğinde Pix2Pix mimarisinin sentetik görüntü üretim aşamasında yapısal (SSIM) olarak yüzde 86 oranında başarımlar göstererek en iyi sonuç verdiği gözlemlenmektedir. DiscoGAN mimarisi eğitim gerçekleştirirken çift taraflı dönüşüm ve doğrulama işlemi kullandığından Pix2Pix mimarisine yakın kalitede görüntüler üretmektedir. Pix2Pix mimarisi aynı zamanda zamansal maliyet karşılaştırmasında da daha iyi sonuç verdiği görülmektedir. Bu mimarinin ağ yapısının denetimli ve daha basit olmasından dolayı kısa sürede kaliteli görüntü ürettiği görülmektedir.

**Tablo 1.** Mimarilerin denim2bıyık test kümesindeki başarımların sonuçları

Model Adı	Metrik Adı (Benzerlik)			Zaman (Saniye)
	SSIM	MSE	PSNR	
AttentionGAN	0.757	2293.70	32.72	Bir görüntü üretim süresi 0,865
CycleGAN	0.754	2465.01	32.55	0,024
Pix2Pix	<b>0.868</b>	<b>808.37</b>	<b>33.32</b>	<b>0,022</b>
DiscoGAN	0.834	1628.74	32.90	0,848

Eğitim tamamlandıktan sonra 4 mimari ile test kümesinde bulunan görüntülerin sentetik bıyık desen çıktısı elde edilmiştir. **Şekil 9**'da test kümesinden birkaç denim kumaş görseline ait sonuçlar gösterilmektedir. Buna göre Pix2Pix ve DiscoGAN mimarisiyle elde edilen görsellerin yüksek oranda gerçek sonuçlara (Ground True) benzediği görülmüştür. CycleGAN ve AttentionGAN mimarilerinin ise bıyık desen üretiminde beklenen seviyede kaliteli görüntüler üretmediği görülmektedir. Ancak hedeflenen gerçek bıyık desen görüntüsünün henüz istenilen benzerlikte üretilemediği görülmektedir. Bunun en büyük nedeni, veri kümesindeki görsellerin belirli bir standartta çekilmemiş olmasıdır. Bir sonraki çalışmada bu standartlaştırma işlemi üzerinde durulacaktır.



Şekil 9. Mimarilerin denim2byık test kümesindeki görüntü üretim sonuçları

#### 4. Sonuç ve Öneriler

Bu makalede, görüntüden görüntüye dönüşüm yapan çekişmeli üretici ağ mimarilerinin performans karşılaştırması yapılmıştır. Mimarilerin karşılaştırmasında kullanılan veri kümesi Malatya organize sanayinde faaliyet gösteren Baykan Denim firmasından elde edilmiştir. Veri kümesi 950 denim2bıyık desen çifti görüntüsünü içermektedir. Geleneksel sistemde, denim kumaşından sentetik bıyık desen üretimi çeşitli görüntü düzenleme programları ile manuel çizilerek elde edilir. Bu işlem ortalama 2-3 saat sürmekte ve işlemi yapan uzmanın kabiliyetine göre farklı kalitede bıyık motifleri elde edilebilmektedir. Bu çalışmanın temel katkısı, bahsedilen dezavantajları ortadan kaldıran, bıyık üretim süresini yaklaşık 1 saniyeye düşüren, personel maliyetlerini azaltan ve farklı kalitede bıyık motiflerinin çıkmasını engelleyen otonom bir sistem önerilmiştir. Literatürde üretken ağ mimarilerinin denim ürünlerine uygulamasını içeren farklı bir çalışmanın olmadığı görülmektedir. Bu durum yapılan çalışmanın akademik özgün değerini yükseltmektedir. Yapılan deneysel çalışmalarda Pix2Pix mimarisinin zamansal ve doğruluk olarak en iyi sonuçları ürettiği görülmüştür. Oldukça ümit verici sonuçların elde edildiği bu çalışmada veri kümesinin standardizasyonu ve endüstriyel ortamda gerçek zamanlı çalışabilecek gömülü bir sistemin üretilme aşaması bir sonraki hedeflerdir.

#### Teşekkür

Bu çalışma İnönü Üniversitesi Bilimsel Araştırma ve Koordinasyon birimi tarafından “FKP-2021-2144” proje numarası ile finanse edilmiştir.

Çalışmada kullanılan veri kümesini erişime açan Baykan Denim Konfeksiyon A. Ş’ye ve İnönü Üniversitesi’ne teşekkürlerimizi sunarız.

#### Yazarların Katkısı

Yazarların makaleye olan katkıları eşit orandadır.

#### Çıkar Çatışması Beyanı

Yazarlar arasında herhangi bir çıkar çatışması bulunmamaktadır.

#### Araştırma ve Yayın Etiği Beyanı

Yapılan çalışmada araştırma ve yayın etiğine uyulmuştur.

#### Kaynaklar

- [1] Das S., Dey A., Pal A., Roy N. 2015. Applications of artificial intelligence in machine learning: review and prospect. *International Journal of Computer Applications*, 115 (9), 31–41.
- [2] LeCun Yann, et al. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation*.
- [3] Goodfellow I J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y. 2014. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 2672–2680.
- [4] Radford A., Metz L., Chintala S. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *4th International Conference on Learning Representations, ICLR 2016- Conference Track Proceedings*, 1–16.
- [5] Isola P., Zhu J Y., Zhou T., Efros A A. 2017. Image-to-Image Translation with Conditional Adversarial Networks. *Proceedings- 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 5967–5976.
- [6] Zhu J Y., Park T., Isola P., Efros A A. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2242–2251.

- [7] Karras T., Aila T., Laine S., Lehtinen J. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. 6th International Conference on Learning Representations, ICLR 2018- Conference Track Proceedings, 1–25.
- [8] Huang X., Belongie S. 2017. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. Proceedings of the IEEE International Conference on Computer Vision, 1510–1519.
- [9] Karras T., Laine S., Aila T. 2019. A Style-based Generator Architecture for Generative Adversarial Networks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 4396–4405.
- [10] Wang T C., Liu M Y., Zhu J Y., Tao A., Kautz J., Catanzaro B. 2018. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 8798–8807.
- [11] Park T., Liu M Y., Wang T C., Zhu J Y. 2019. Semantic Image Synthesis with Spatially-adaptive Normalization. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2332–2341.
- [12] Dundar A., Sapra K., Liu G., Tao A., Catanzaro B. 2020. Panoptic-based Image Synthesis. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 8067–8076.
- [13] Karras T., Aittala M., Laine S., Härkönen E., Hellsten J., Lehtinen J., Aila T. 2021. Alias-free generative adversarial networks. NeurIPS, 2106.12423.
- [14] Wang X., Xie L., Dong C., Shan Y. 2021. Real-ESRGAN: Training Real-world Blind Super-resolution with Pure Synthetic Data. Proceedings of the IEEE/CVF International Conference on Computer Vision, 1905-1914.
- [15] Suvorov R., Logacheva E., Mashikhin A., Remizova A., Ashukha A., Silvestrov A., Kong N., Goka H., Park K., Lempitsky V. 2021. Resolution-robust large mask inpainting with fourier convolutions. 2109.07161.
- [16] Wang X., Li Y., Zhang H., Shan Y. 2021. Towards real-world blind face restoration with generative facial prior. 2101.04061.
- [17] Huang H., Yu P S., Wang C. 2018. An Introduction to Image Synthesis with Generative Adversarial Nets. 1–17.
- [18] Goodfellow I. 2016. Generative Adversarial Networks. NIPS 2016 Tutorial.
- [19] Lazarou C. 2021. Generative Adversarial Networks. <https://www.slideshare.net/ckmarkohchang/generative-adversarial-networks>. (Erişim Tarihi: 20.04.2021)
- [20] Ghosh A., Kumar H., Sastry P S. 2017. Robust Loss Functions under Label Noise for Deep Neural Networks. 31st AAAI Conference on Artificial Intelligence, AAAI 2017, 1919–1925.
- [21] Ronneberger O., Fischer P., Brox T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI.
- [22] Mihelich M., Dognin C., Shu Y., Blot M. 2020. A Characterization of Mean Squared Error for Estimator with Bagging. ArXiv, abs/1908.02718.
- [23] Kim T., Cha M., Kim H., Lee J. K., Kim J. 2017. Learning to Discover Cross-domain Relations with Generative Adversarial Networks. 34th International Conference on Machine Learning, ICML 2017, 4, 2941–2949.
- [24] Mejjati Y A., Richardt C., Tompkin J., Cosker D. 2018. Unsupervised attention-guided image-to-image translation. NeurIPS 2018, 1–11.
- [25] Nilsson J., Akenine-Möller T. 2020. Understanding SSIM. ArXiv, abs/2006.13846.
- [26] Fardo F A., Conforto V H., Oliveira F C., Rodrigues P. 2016. A formal evaluation of PSNR as quality measurement parameter for image segmentation algorithms. ArXiv, abs/1605.07116.