**Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 14(4), 1397-1407, 2024**
***Journal of the Institute of Science and Technology, 14(4), 1397-1407, 2024***
**ISSN: 2146-0574, eISSN: 2536-4618**
**Bilgisayar Mühendisliği / *Computer Engineering*** **DOI: 10.21597/jist.1533471**
**Araştırma Makalesi / *Research Article***
**Geliş tarihi / *Received*: 14.08.2024** **Kabul tarihi / *Accepted*: 10.09.2024**

**Atıf İçin:** Kılıç, E., Şahin, Ö. ve Toprak, A. N., (2024). Çekişmeli Saldırıların Derin Öğrenme Tabanlı Yüz Sahteciliği Önleme Sistemlerine Etkisi. *Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 14(4), 1397-1407.

**To Cite:** Kılıç, E., Şahin, Ö. & Toprak, A. N., (2024). The Impact of Adversarial Attacks on Deep Learning-based Face Anti-Spoofing Systems. *Journal of the Institute of Science and Technology*, 14(4), 1397-1407.

# Çekişmeli Saldırıların Derin Öğrenme Tabanlı Yüz Sahteciliği Önleme Sistemlerine Etkisi

Ersin KILIÇ[1,2*], Ömür ŞAHİN[2], Ahmet Nusret TOPRAK[2].

**Öne Çıkanlar:**
- Yüz sahteciliği önleme sistemleri
- Çekişmeli saldırı üretimi

**Anahtar Kelimeler:**
- Yüz sahteciliği önleme
- Çekişmeli saldırı
- Derin öğrenme

**ÖZET:**

Günümüzde yüz tanıma ve doğrulama sistemleri akıllı telefonlardan güvenlik sistemlerine, ödeme uygulamalarından uzaktan sağlık uygulamalarına kadar birçok alanda yüksek doğrulukla kullanılmaktadır. Yüz tanıma sistemlerini yanıltmanın en yaygın yolu kişinin sisteme kendi yüzü yerine maske, yazıcı çıktısı ya da monitör gibi araçlarla başka birinin yüzünü sunmasıdır. Son yıllarda birçok bilgisayarla görme uygulamasında olduğu gibi yüz tanıma sistemlerinde görüntü sahteciliğini önleme amacıyla da oldukça başarılı derin sinir ağı modelleri geliştirilmiştir. Bu modellerin farklı sahtecilik yöntemlerini hassas bir şekilde tespit edebilmesi ve girdi görüntülere uygulanacak saldırılara karşı dayanıklı olması beklenmektedir. Bu çalışmada güncel ve başarılı bir görüntü sahteciliği tespit modeli olan DGUA-FAS'ın çekişmeli saldırılara karşı dayanıklılığı araştırılmaktadır. Bu amaçla MIO algoritmasına dayalı kara kutu çekilmeli saldırı üretme yöntemi ile DGUA-FAS modelini yanıltmak amacıyla çekişmeli örnek görüntüler üretilmektedir. Deneysel çalışmalar, DGUA-FAS modelinin saldırı uygulanan tüm görüntüleri hatalı şekilde sınıflandırdığını göstermektedir. Elde edilen sonuçlar, yüz sahteciliği tespit modellerinin çekişmeli saldırılara karşı daha dayanıklı hale gelmesi gerektiğini göstermektedir.

# The Impact of Adversarial Attacks on Deep Learning-based Face Anti-Spoofing Systems

**Highlights:**
- Face anti-spoofing system
- Adversarial attack generation

**Keywords:**
- Face anti-spoofing
- Adversarial attack
- Deep learning

**ABSTRACT:**

Face recognition and verification systems are widely employed in various applications, from smartphones and security systems to payment and remote healthcare services, demonstrating high accuracy. However, a common method to spoof these systems involves presenting a different person's face using tools such as masks, printouts, or monitors instead of the actual user's face. In recent years, similar to advancements in other computer vision tasks, deep neural networks have been developed to effectively combat image forgery in face recognition systems. These models are expected to accurately detect diverse forgery techniques and be resilient to adversarial attacks on input images. This study investigates the robustness of DGUA-FAS, a state-of-the-art image forgery detection model, against adversarial attacks. Adversarial examples are generated using a black-box adversarial attack generation method based on MIO algorithm, to mislead the DGUA-FAS model. Experimental results demonstrate that the DGUA-FAS model misclassifies all attacked images. The findings highlight the necessity for developing face forgery detection models that are more resilient to adversarial attacks.

[1] Ersin KILIÇ (**Orcid ID:** 0000-0002-0924-9246), ArkSigner Yazılım ve Donanım San. A.Ş., Bilkent Cyberpark, Ankara, Türkiye
[2] Ömür ŞAHİN (**Orcid ID:** 0000-0003-1213-7445), Ahmet Nusret TOPRAK (**Orcid ID:** 0000-0003-4841-9508), Erciyes Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Kayseri, Türkiye

***Sorumlu Yazar/*Corresponding Author**: Ersin KILIÇ, e-mail: ersinkilic@erciyes.edu.tr

## GİRİŞ

Yapay zekâ ve biyometri teknolojilerindeki hızlı gelişmeler, yüz tanıma sistemlerinin her alanda daha yaygın olarak kullanılmasına yol açmaktadır. Akıllı telefonlardan güvenlik sistemlerine, ödeme uygulamalarından uzaktan sağlık uygulamalarına kadar yüz tanıma, kimlik doğrulama için tercih edilen yöntemlerden biri haline gelmiştir. Ancak, bu teknolojinin yaygınlaşmasıyla birlikte, güvenlik açıkları ve potansiyel tehditler de ortaya çıkmaktadır. Bu tehditlerden en önemlilerinden biri, yüz tanıma sistemini kullanan kişinin kendine ait olmayan bir yüzle sistemi kandırmasıdır. Bu amaçla en yaygın kullanılan sahtecilik yöntemleri arasında iki boyutlu ve üç boyutlu maskeler, başkasına ait yüze ait yazıcı çıktısı ya da ekran görüntüsü sayılabilir.

Yüz sahteciliği önleme (Face Anti-Spoofing) yöntemleri, yüz doğrulama sistemlerinde, sistemle etkileşime geçen kişinin gerçek ve canlı olduğundan emin olmak amacıyla kullanılan bir güvenlik mekanizmasıdır (Yu ve ark., 2022). Bu yöntemler, özellikle güvenliğin kritik olduğu ve kimlik doğrulamanın kesinlik gerektirdiği finansal işlemler, güvenlik sistemleri sağlık sektörü gibi alanlarda büyük önem taşımaktadır. Günümüzde birçok alanda olduğu gibi canlılık kontrolü için de derin sinir ağları oldukça başarılı sonuçlar üretmektedir (Yu ve ark., 2022).

Derin öğrenme tekniklerinin bilgisayarla görme sahasındaki başarısı ile yüz sahteciliği tespiti amacıyla da birçok derin sinir ağı modeli önerilmiştir. Yang ve arkadaşları, özellik gösterimi için 8 katmanlı sığ CNN kullanan ilk uçtan uca derin yüz sahteciliği tespiti yöntemini önermiştir. Ancak, verisetlerinin sınırlı sayıda ve çeşitlilikte görüntüler içermesi sebebiyle, yüz sahteciliği tespiti amacıyla geliştirilen CNN tabanlı modeller aşırı uyumlanma (overfitting) problemi yaşayabilmektedir (Yang ve ark., 2014). Literatürdeki bazı çalışmalar yüz sahteciliği tespiti amacıyla girdi görüntüden ürettikleri yapay derinlik haritalarını kullanmaktadır. Böyle bir çalışmada Atoum ve arkadaşları geliştirdikleri evrişimli sinir ağı modelini (DepthNet) ürettikleri yapay derinlik haritaları ile eğitmiştir (Atoum ve ark., 2017). Yu ve arkadaşları DepthNet modelindeki evrişim katmanını merkezi fark evrişim katmanı ile değiştirerek Merkezi Fark Evrişim Ağı (Central Difference Convolution Network, CDCN) mimarisini oluşturmuşlardır (Yu ve ark., 2020). Derinlik tahminine dayanan bir diğer model olan Mekansal Gradyan ve Zamansal Derinlik (Spatial Gradient and Temporal Depth, FAS-SGTD) modeli ise daha ayırt edici ayrıntıları tespit etmeye ve uzaysal-zamansal bilgileri kullanmaya odaklanmaktadır (Wang ve ark., 2020). Jourabloo ve arkadaşları yüz sahteciliği tespitini sahte gürültü modelleme problemi olarak yeniden tanımlayarak, girdi görüntüden sahteciliği ele veren örüntüleri çıkartacak için bir kodlayıcı-kod çözücü mimarisi önermiştir (Jourabloo ve ark., 2018). Son zamanlarda yapılan bir çalışmada ise Hong ve arkadaşları alan genelleştirilmiş saldırıları tespit etmek için bir Transformatör tabanlı özellik çıkarıcı ve Sentetik Bilinmeyen Saldırı Örnek Üreteci (synthetic unknown attack sample generator, SUASG) oluşan yeni bir yöntem olan DGUA-FAS'ı önermişlerdir (Hong ve ark., 2023). Bu çalışmada kullanılan SUASG ağları, özellik çıkarıcının eğitimine yardımcı olmak için bilinmeyen saldırı örneklerini simüle etmektedir.

Derin sinir ağlarının ürettiği bu başarılı sonuçlara rağmen Szegedy ve ark. belirli küçük bozulmaların kasıtlı olarak girdi görüntülere eklenmesiyle derin sinir ağlarının yanıltılabileceğini ve görüntülerin hatalı şekilde sınıflandırılmasına neden olunabileceğini göstermiştir (Szegedy ve ark. 2013). Çekişmeli saldırı (adversarial attack) olarak adlandırılan bu yöntem, bir yapay zekâ modelini yanıltmak amacıyla tasarlanan, genellikle insan gözünün fark edemeyeceği kadar küçük değişikliklerin uygulandığı saldırılar olarak tanımlanır (Taşyürek ve Gül, 2023). Literatürde bu saldırıların, özellikle görüntü tanıma, doğal dil işleme ve ses tanıma gibi alanlarda kullanılan yapay zekâ modellerine karşı gerçekleştirildiği çalışmalar bulunmaktadır (Akhtar ve ark, 2021).

Çekişmeli saldırı üretme yöntemleri arasından en bilinenlerinden olan hızlı gradyan işaret yöntemi (Fast Gradient Sign Method, FGSM) kayıp fonksiyonunun gradyan değerini kullanarak giriş görüntüsünü modeli en çok yanıltan yönde değiştirmeyi amaçlar (Goodfellow ve ark, 2014). Kurakin ve arkadaşları ise FGSM'nin yinelemeli bir uygulaması olan Yinelemeli Hızlı Gradyan İşaret Yöntemi'ni (Iterative Fast Gradient Sign Method, I-FGSM) önermiştir (Kurakin ve ark, 2018). Dong ve arkadaşları I-FGSM'ye bir momentum mekanizması ekleyerek bu yöntemi genişletmiş ve Momentum Yinelemeli Hızlı Gradyan İşaret Yöntemi'ni (Momentum Iterative Fast Gradient Sign Method, MI-FGSM) önermişlerdir. Madry ve arkadaşları ise yine FGSM'ye dayalı bir çekişmeli saldırı üretme yöntemi olan Yansıtılmış Gradyan İnişi (Projected Gradient Descent, PGD) geliştirmiştir (Madry ve ark, 2017). Jacobian Tabanlı Belirginlik Haritası Saldırısı (Jacobian-based Saliency Map Attack, JSMA) görüntüye eklenen bozucu etkeni (perturbation) görüntünün daha küçük bölgeleriyle sınırlandırarak diğer yöntemlerden ayrışmaktadır (Papernot ve ark., 2016).

Bu çalışmada, Çoklu Bağımsız Hedef (Many Independent Objective, MIO) algoritmasına dayalı bir kara kutu çekişmeli saldırı üretme yöntemi (Sahin, 2024) yüz sahteciliği önleme sistemlerinin dayanıklılığını test etmek amacıyla uygulanmıştır. Gerçekleştirilen çalışmada son dönemde önerilmiş başarılı bir yüz sahteciliği tespiti ağı olan DGUA-FAS ağının (Hong ve ark., 2023) MIO tabanlı çekişmeli saldırı ürütme yöntemine karşı dayanıklılığı araştırılmıştır. Bu amaçla iBeta canlılık tespiti veri setinden beş farklı sahtecilik yöntemi ile üretilmiş dörder görüntü rastgele seçilerek kullanılmıştır. Kullanılan çekişmeli saldırı üretme yönteminin tutarlılığını göstermek yöntem seçilen görüntüler üzerinde on kez birbirinden bağımsız olarak çalıştırılmıştır. Elde edilen sonuçlar önerilen yöntemin tüm denemelerde ağı yanıltarak sahte görüntüleri gerçek olarak sınıflandırmasını sağlandığını göstermektedir.

## MATERYAL VE METOT

Bu bölümde öncelikle çalışmanın temel aldığı problem tanımlanmakta, kullanılan yüz sahteciliği önleme modeli ve bu modeli yanıltmak için kullanılan örneklerin üretildiği çekişmeli saldırı üretme yöntemi tanıtılmaktadır.

### Problem Tanımı

Geleneksel çekişmeli saldırı üretimi yöntemleri genellikle modelin gradyan bilgisine dayalı saydam kutu olarak üretilir. Ancak bu metotlar modelin iç yapısına erişim gerektirdiği için gerçek dünya problemlerine uygun değildir. Bu durum kara kutu yöntemlerin geliştirilmesine neden olmuştur. Bartlett ve arkadaşları, arama algoritmaları kullanarak çekişmeli saldırı üretme yöntemi önermiştir. Bu çalışmada önerilen bu yöntem kullanılmıştır (Bartlett ve ark., 2023). Bu metoda göre bir görüntü sınıflandırıcı takip eden eşitlik ile sınıflandırılmaktadır:

$$f: I \rightarrow L \times \mathbb{R}^n \tag{1}$$

burada *f* fonksiyonu, girdi olarak, $i \in I$ görüntüsünü alır ve $l \in L$ etiketi ile güven vektörünü ($conf \in \mathbb{R}^n$) çıktı olarak döndürür. Güven vektörü ($conf \in \mathbb{R}^n$) her bir etiket değerinin olasılığını azalan sırada göstermektedir. En yüksek olasılıklı güven vektörü $conf_1$ iken diğer etiketlerin güven değerleri $conf_1$, $conf_2$, …, $conf_n$ olarak sıralanmaktadır. Bu çalışma için *f* fonksiyonu, girdi olarak verilen yüz görüntüsünü sahte ya da gerçek olarak sınıflandıran bir derin sinir ağı modelidir. Ayrıca $m: I \rightarrow I$, girdi olarak $i \in I$ alan bir değişim fonksiyonudur. Buradaki temel amaç mutasyona uğramış ve $f(m(i)) \neq f(i)$ olan mutasyona uğramış $m(i)$ değerini bulmaktır. Bu noktada da hedefli ve hedefsiz olmak üzere iki farklı strateji uygulanmaktadır. Sonucu herhangi bir etikete çevirmenin yettiği durumlar hedefsiz,

spesifik olarak belirlenen bir etikete çevirmeye çalışılan yöntemler hedefli olarak adlandırılmaktadır. Burada ağın üretmiş olduğu güven değeri uygunluk değeri olarak kullanılmaktadır. Verilen sınıflandırma modeli $f: I \rightarrow L \times \mathbb{R}$ fonksiyonuna verilen temel bir *i* görüntüsü ve bu görüntünün mutantı *m(i)* kullanılarak Eşitlik 2'de sunulan uygunluk fonksiyonu kullanılmıştır.

$$f = \begin{cases} f(m(i))_1^{conf} - f(m(i))_2^{conf} & eğer\ f(i)_1^l = f(m(i))_1^l \\ -f(m(i))_1^{conf} & aksi\ takdirde \end{cases} \quad (2)$$

Bu denklemdeki temel amaç en olası ikinci güven değerini artırırken en yüksek güven değerini de düşürmektir. Uygunluk değeri [-1, 1] aralığında tasarlanmıştır ve negatif değerler etiketin değiştirildiğini göstermektedir.

**Bilinmeyen Saldırılar ile Alan Genelleştirilmiş Yüz Sahteciliği Tespiti (DGUA-FAS) Modeli**

DGUA-FAS modeli, bilinen ve bilinmeyen saldırıların üstesinden gelmek için bir transformatör ağı ve ek bir sentetik örnek üreteci kullanarak etkili bir şekilde yüz sahteciliği tespiti yapmayı hedeflemektedir. Yöntem, farklı alanlardan (veri setlerinden) gelen aynı saldırı türü örneklerinin benzer, farklı türdeki örneklerin ise farklı olacak şekilde tespitini amaçlamaktadır. Sınıflandırma kaybı ($L_{\text{cls}}$) ve ilişkilendirme kaybı ($L_{\text{assoc}}$) olmak üzere iki temel kayıp fonksiyonuna dayanmaktadır. $L_{\text{cls}}$, çok sınıflı çapraz entropi kaybı fonksiyonudur. Gerçek yüz örneklerinin, farklı veri setlerinden gelse bile, benzer ve kompakt özellik vektörüne sahip olmasını sağlamak amacıyla, $L_{\text{assoc}}$ kaybı, gerçek yüz örneklerinin özelliklerini orijine yakınlaştırarak konsantre olmalarını sağlamaktadır. DGUA-FAS, bu kayıpların yanı sıra başka ek kayıp terimleriyle eğitilmektedir. Model, omurga ağı olarak bir transformatör ağı kullanılmaktadır.

Eğitim sırasında yalnızca bilinen saldırılarla ilgili verilere sahip olunsa da modelin bilinmeyen saldırıları da tanıyabilmesi amaçlanmaktadır. Bunun için, eğitim sürecinde simüle edilmiş bilinmeyen saldırı örnekleri kullanılmaktadır. Bu simülasyonlar, ağın farklı katmanları kullanılarak çeşitli zorluk seviyelerinde üretilmekte ve hem eğitim veri dağılımının içinde hem de dışında yer alan örneklerden oluşmaktadır. Bu amaçla, "Sentetik Bilinmeyen Saldırı Örnek Üreteci (SUASG)" adı verilen bir yapı tasarlanmıştır. SUASG, eğitimin ilk aşamasında sınıflandırma kaybı ve taklit kaybı (imitation loss, $L_{\text{imi}}$) kullanılarak eğitilmektedir. İkinci aşamada ise, SUASG sabitlenmekte ve Transformer tabanlı özellik çıkarıcı, simüle edilmiş bilinmeyen saldırı örnekleriyle eğitilmektedir. Eğitimin sonunda, sadece Transformer tabanlı özellik çıkarıcı ve son sınıflandırma katmanı test aşamasında kullanılmaktadır.

$$L_{assoc} = \frac{1}{|F_{real}|} \sum_{f \epsilon F_{real}} \|f\|_1, \forall f \epsilon F_{real} \quad (3)$$

$$L_{imi} = \frac{1}{N} \sum_{i=1}^{N} \sum_{g=1}^{2} \left\| f_{SUASG}^{g,i} - f_{extract}^{g,i} \right\|_1 \quad (4)$$

**Çoklu Bağımsız Hedef Algoritması**

Çoklu bağımsız hedef algoritması (Many Independent Objective, MIO) ilk olarak Arcuri tarafından test senaryosu üretiminde kullanılmak üzere önerilmiştir (Arcuri, 2018). Daha sonra bu algoritma Sahin tarafından (Sahin, 2024) çekişmeli saldırı üretimi için uygun hale getirilmiştir. MIO algoritması, belirli piksel değerlerinin ağa daha fazla zarar verebileceğini varsayarak çekişmeli saldırı üretmektedir. Bu şekilde, derin ağın en az sayıda değişiklikle yanlış cevabı vermesi sağlanmaktadır. Önerilen algoritmanın ayrıntıları Algoritma 1'de verilmiştir.

**Algoritma 1** Çekişmeli saldırı üretiminde kullanılan MIO algoritmasının adımları

```
1:  MFE ← Maksimum uygunluk fonksiyonu değerlendirme sayısı
2:  M1 ← Bir mutasyonu oranı
3:  M0 ← Sıfır mutasyonu oranı
4:  apc ← AdaptifParametreKontrolu(apc_start, apc_end, apc_time, apc_threshold)
5:  arsiv ← Arşiv
6:  while !maksimumDegerlendirme(MFE) ve !degisti do
7:  if arsiv boş ise then
8:  ornek ← çözüm_üret(apc)
9:  else
10: ornek ← arşivden_örnekle(arsiv, apc)
11: end if
12: gerekli_ise_arşive_ekle(ornek, arsiv)
13: end while
14: arşivi_küçült(arsiv)
```

Algoritmanın MFE, $M_0$, $M_1$, $apc_{start}$, $apc_{end}$, $apc_{time}$, $apc_{threshold}$ olmak üzere yedi farklı kontrol parametresi bulunmaktadır ve ilk olarak kontrol parametreleri atamaları yapılmaktadır (satır 1-4). Yeni birey üretimi yalnızca mutasyon operatörü ile yapılmaktadır. Çalışmada üç farklı mutasyon operatörü kullanılmaktadır ve bu operatörler $M_0$ ve $M_1$ parametreleri ile kontrol edilmektedir. $M_0$ parametresi RGB değerlerinin 0 olarak atanma ihtimalini, $M_1$ parametresi ise 255 olarak atanma ihtimalini belirtmektedir. Üçüncü parametre ise bir Gaussian mutasyondur. Bu mutasyon ile 0-255 aralığında bir değer atanabilmektedir. Bu mutasyonun olasılığı ise 1- ($M_0$+$M_1$) formülü ile hesaplanmaktadır. Bu değer hesaplanırken kullanılan standart sapma ($\sigma$) değeri adaptif parametre kontrolü (*apc*) mekanizması ile belirlenmektedir. 0-255 aralığındaki yeni piksel değeri, mevcut piksel değeri ve standart sapma $\sigma$ değeri kullanılarak rastgele normal dağılım ile hesaplanmaktadır. *apc* mekanizmasıyla birlikte gelen $\sigma$ değerindeki adaptiflik sayesinde keşif/sömürü arasındaki denge sağlanılmaktadır. Bu değer Eşitlik 5'teki gibi hesaplanmaktadır:

$$\sigma = \begin{cases} apc_{start} & eğer\ budget_{used} < apc_{time} \\ apc_{end} & eğer\ budget_{used} > apc_{threshold} \\ apc_{start} + (apc_{end} - apc_{start}) * \frac{budget_{used} - apc_{time}}{apc_{threshold} - apc_{time}} & aksi\ takdirde \end{cases} \quad (5)$$

Bu eşitlikte $apc_{start}$, $apc_{end}$, $apc_{time}$ ve $apc_{threshold}$ olmak üzere dört farklı parametre bulunmaktadır. İlk olarak algoritma kullanılan değerlendirme sayısı ($budget_{used}$ olarak gösterilmektedir) $apc_{time}$ olana kadar $apc_{start}$ değerini döndürmektedir. Bu süreçte algoritma daha büyük adımlar daha fazla atarak keşif sürecini gerçekleştirmektedir. $apc_{start}$ ve $apc_{end}$ değer aralığında ise ölçeklenmiştir bir değer döndürmektedir. Burada keşif yavaş yavaş azalmakta ve sömürü sürecini başlatmaktadır. Son olarak $budget_{used}$ değeri $apc_{threshold}$ değerini aştığı anda $apc_{end}$ değerini döndürmektedir.

Ayrıca algoritmada içerisinde bir veya birden fazla çözüm barındıran bir arşiv mekanizması bulunmaktadır. Güven değerini düşüren herhangi bir çözüm doğrudan arşive eklenmektedir (satır 12). Her bir çözüm satır, sütun, değişiklik yapılacak RGB değeri ve bir kalite göstergesi $c_k$ sayaç değeri barındırmaktadır. Arşivde hiçbir çözüm bulunmaması durumunda algoritma rastgele bir çözüm ile başlar (satır 8). Bu çözüm güven değerini düşüren bir çözüm ise arşive eklenmektedir. Arşive ilk çözüm eklenne kadar bu süreç devam eder. İlk çözüm eklendikten sonra ise arşivden örnekleyerek arama devam eder (satır 10). Arşivdeki her bir çözüme bir kalite göstergesi $c_k$ değeri atanmaktadır. Atanan bu değer arşivden örnekleme yapıldıktan sonra üretilen çözümün kalitesine bağlı olarak değiştirilmektedir. Eğer bu üretilen çözüm güven değerini daha aşağı çekebilen bir çözüm ise $c_k$ değeri 0 olarak atanır, değilse 1 artırılır. Arşivden örnekleme yaparken $c_k$ değeri en düşük değer seçilir. Böylelikle arama sürecinde

sömürülebilir bir hedef piksel bulduysa bu pikselin komşularında aramaya devam etmektedir. Son olarak arama, $budget_{used}$ değeri MFE değerine ulaştığı anda arama sonlandırılır ve arşiv küçültme işlemi başlar. Arşiv küçültme eyleminde arşivdeki her bir birey tek tek değerlendirilir ve çevrilen etiketin güven değeri 0.5 üstünde kalacak şekilde bir budama işlemi yapılır. Burada maksimum değerlendirme sayısının MFE'yi aşmaması için $budget_{used}$ değeri Eşitlik 6'daki gibi hesaplanır:

$$budget_{used} = (eval_{current} + archive_{size})/MFE \quad (6)$$

burada $eval_{current}$ mevcut değerlendirme sayısını verirken; $archive_{size}$ arşivdeki birey sayısını vermektedir.

## BULGULAR VE TARTIŞMA

Bu bölümde öncelikle çalışma kapsamında yapılan deneylerde kullanılan veri seti tanıtılmakta, ardından deneysel kuruluma dair bilgiler verilmekte ve son olarak deneylerde elde edilen sayısal ve görsel sonuçlar sunulmaktadır.

### Veri Seti

Deneyler için kullanılan görüntüler iBeta canlılık tespiti veri setinden[1] seçilmiştir. Görüntüler, maske, 3B maske, fotoğraf, 3B fotoğraf ve monitör olmak üzere beş farklı saldırı tipinden seçilmiştir. Her bir saldırı tipinden dört adet olmak üzere toplam yirmi adet görüntü seçilmiştir.

### Deneysel Kurulum

Bu çalışmadaki deneyler Çekişmeli Örnek Üreteci (Adversarial Example Generator) isimli araç kullanılarak gerçekleştirilmiştir. Deneyler Intel i9 işlemci ve 64 GB RAM içeren bir bilgisayar üzerinde gerçekleştirilmiştir. Her bir deney on kez bağımsız olarak tekrarlanmıştır. Deneylerde kullanılan kontrol parametre değerleri algoritmanın önerildiği çalışmada (Sahin, 2024) olduğu gibi seçilmiştir. $M_0$ ve $M_1$ değerleri sırasıyla 0.4 ve 0.3 iken; $apc_{start}$, $apc_{end}$, $apc_{time}$, $apc_{threshold}$ değerleri sırasıyla 40, 30, 0.4 ve 0.6 olarak alınmıştır. Son olarak *MFE* değeri 50000 seçilmiştir ancak algoritmanın geçerli bir atak ürettiği anda durması nedeniyle hiçbir koşmada 50000 değerine ulaşılmamıştır.

### Deneysel Sonuçlar

Bu bölümde, MIO algoritmasına dayalı kapalı kutu çekişmeli saldırı üretme yöntemi kullanılarak elde edilen görüntüler için DGUA-FAS modelinin ürettiği sonuçlar sunulmaktadır. Deneylerde iBeta canlılık tespiti veri setinden seçilen maske, 3B maske, çıktı, 3B çıktı ve monitör sınıflarından dörder görüntü kullanılmıştır. Seçilen tüm görüntüler, saldırı uygulanmadan önce DGUA-FAS modeli tarafından *sahte* olarak sınıflandırılmıştır.

Çizelge 1, MIO algoritması tabanlı yöntemin ürettiği sonuçları göstermektedir. Elde edilen sonuçlara göre bütün sınıflarda ve bütün görüntülerde saldırının başarılı olduğu, *sahte* olarak sınıflandırılması gereken görüntülerin DGUA-FAS modeli tarafından *gerçek* olarak sınıflandırıldığı görülmektedir. Bütün saldırılar sınıf tabanlı incelendiğinde Maske sınıfında değişim sağlanabilmesi için ortalama 194.7 piksel değişikliği gerekmiştir. Algoritmanın uygun saldırıyı üretmek için ihtiyaç duyduğu ortalama zaman ise 24.7-149.0 dakika aralığında değişmektedir. Tablodan anlaşıldığı üzere bu zaman gerekli olan piksel değişikliği ile doğru orantıda artış göstermektedir. 3B Maske sınıfı incelendiğinde ise ortalama 145.4 piksel değişikliği ile sınıf değişimi gerçekleştirilmiştir. Özellikle mask3d_01 isimli görüntü ortalama 3.9 saniyede ve 22 piksel değişimi ile gerçek sınıfı çıktısı üretmiştir. Monitör sınıfı incelendiğinde ise ortalama 181.8 piksel değişimi ile başarılı sonuç elde edilmiştir. Çıktı

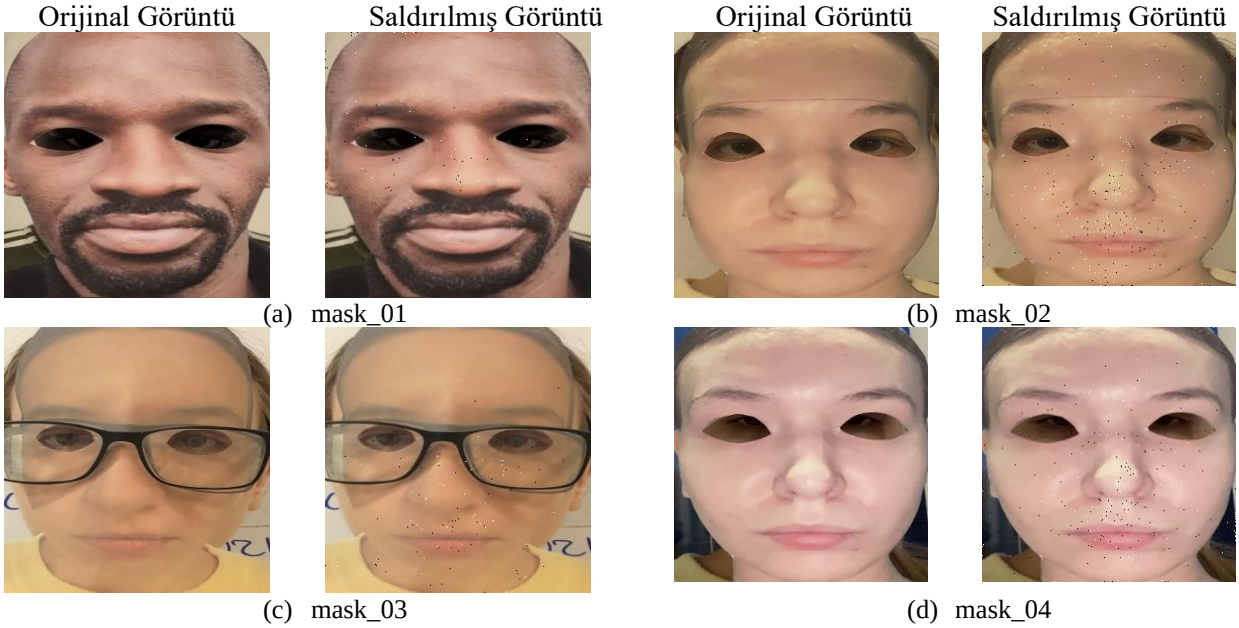
[1] https://www.kaggle.com/datasets/trainingdatapro/ibeta-level-1-liveness-detection-dataset-part-1

sınıfına geldiğimizde ortalama 83 piksel değişimi ile sonuca ulaşılmıştır. Buradaki piksel değişikliği ve ihtiyaç duyulan zaman diğer sınıflara göre daha azdır. Son olarak 3B çıktı sınıfı incelendiğinde ortalama 163,1 piksel değişikliği ile sonuca ulaşmıştır. Genel olarak bakıldığında algoritma Çıktı sınıfında en az zorlanırken monitör sınıfını gerçek etiketine çevirmesi diğer sınıflara kıyasla daha zor gerçekleşmiştir.

**Çizelge 1.** MIO algoritmasına dayalı kapalı kutu çekişmeli saldırı üretme yöntemi ile DGUA-FAS modeli üzerinde gerçekleştirilen saldırı sonuçları

| Sınıf | Görüntü | Görüntü Boyutu | Değiştirilen Piksel Sayısı | Başarı Oranı | İşlem Süresi |
|---|---|---|---|---|---|
| Maske | mask_01 | 256×256 | 72.8 | 100.0 | 24.7 |
| | mask_02 | 256×256 | 335.0 | 100.0 | 149.0 |
| | mask_03 | 256×256 | 112.8 | 100.0 | 45.3 |
| | mask_04 | 256×256 | 258.2 | 100.0 | 84.3 |
| 3B Maske | mask3d_01 | 256×256 | 22.0 | 100.0 | 3.9 |
| | mask3d_02 | 256×256 | 53.8 | 100.0 | 14.2 |
| | mask3d_03 | 256×256 | 205.2 | 100.0 | 84.9 |
| | mask3d_04 | 256×256 | 300.7 | 100.0 | 120.3 |
| Monitör | monitor_01 | 256×256 | 210.4 | 100.0 | 72.4 |
| | monitor_02 | 256×256 | 111.1 | 100.0 | 30.2 |
| | monitor_03 | 256×256 | 151.5 | 100.0 | 95.7 |
| | monitor_04 | 256×256 | 254.2 | 100.0 | 125.8 |
| Çıktı | outline_01 | 256×256 | 68.8 | 100.0 | 22.3 |
| | outline_02 | 256×256 | 64.6 | 100.0 | 16.1 |
| | outline_03 | 256×256 | 137.0 | 100.0 | 61.7 |
| | outline_04 | 256×256 | 61.5 | 100.0 | 23.3 |
| 3B Çıktı | outline3d_01 | 256×256 | 64.9 | 100.0 | 18.0 |
| | outline3d_02 | 256×256 | 119.5 | 100.0 | 33.3 |
| | outline3d_03 | 256×256 | 204.4 | 100.0 | 123.2 |
| | outline3d_04 | 256×256 | 263.6 | 100.0 | 198.7 |
| ORTALAMA: | | | 153.6 | 100.0 | 67.4 |

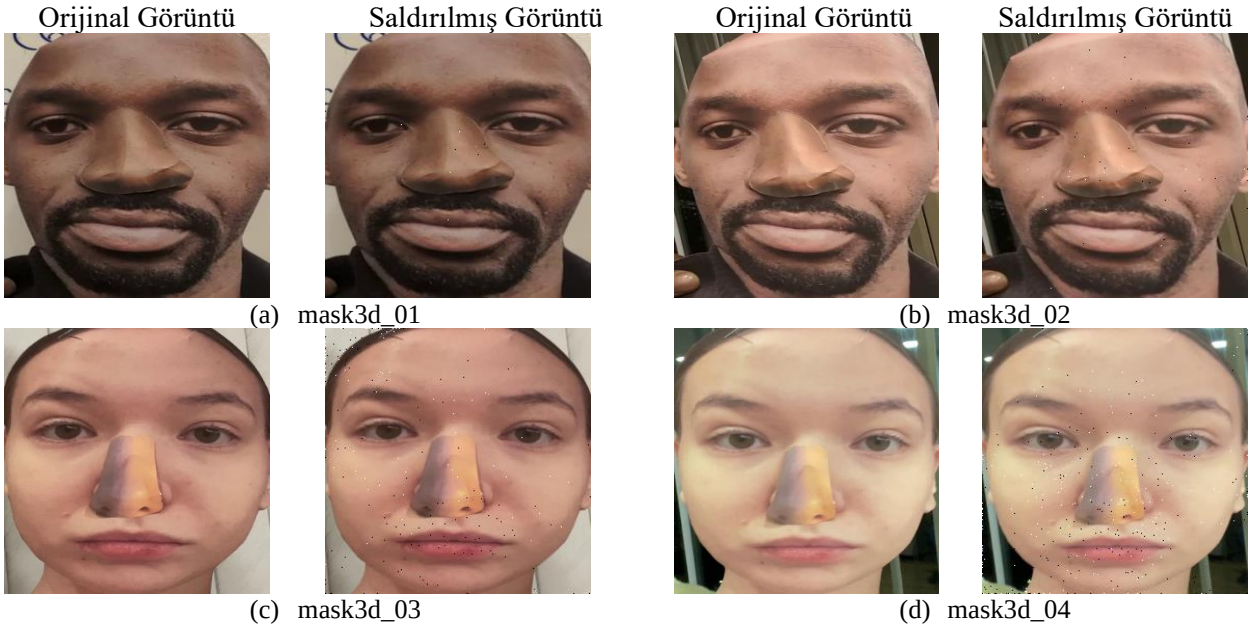
Deney kapsamında kullanılan maske görüntüleri ve MIO algoritması tabanlı kapalı kutu çekişmeli saldırı üretme yönteminin ürettiği saldırılar uygulanarak elde edilen görüntüler Şekil 1'de görülmektedir. Şekil 1 (a) ve (c)'de görüntülerden anlaşılacağı üzere özellikle mask_01 ve mask_03 görüntülerinde oldukça sınırlı sayıda piksel değişikliğe uğratılarak modelin yanıltılması mümkün olmuştur.




**Şekil 1.** Maske Sınıfına ait Görsel Sonuçlar

3B maske sınıfına ait orijinal ve saldırı gerçekleştirilmiş görüntüler ise Şekil 2'de verilmektedir. Özellikle mask2d_01 (Şekil 2 (a)) ve mask3d_02 (Şekil 2 (b)) görüntüleri için gözle ayırt etmenin dahi zor olduğu sınırlı sayıda pikselin değişikliğe uğradığı görülmektedir. Diğer taraftan mask3d_04 (Şekil 2

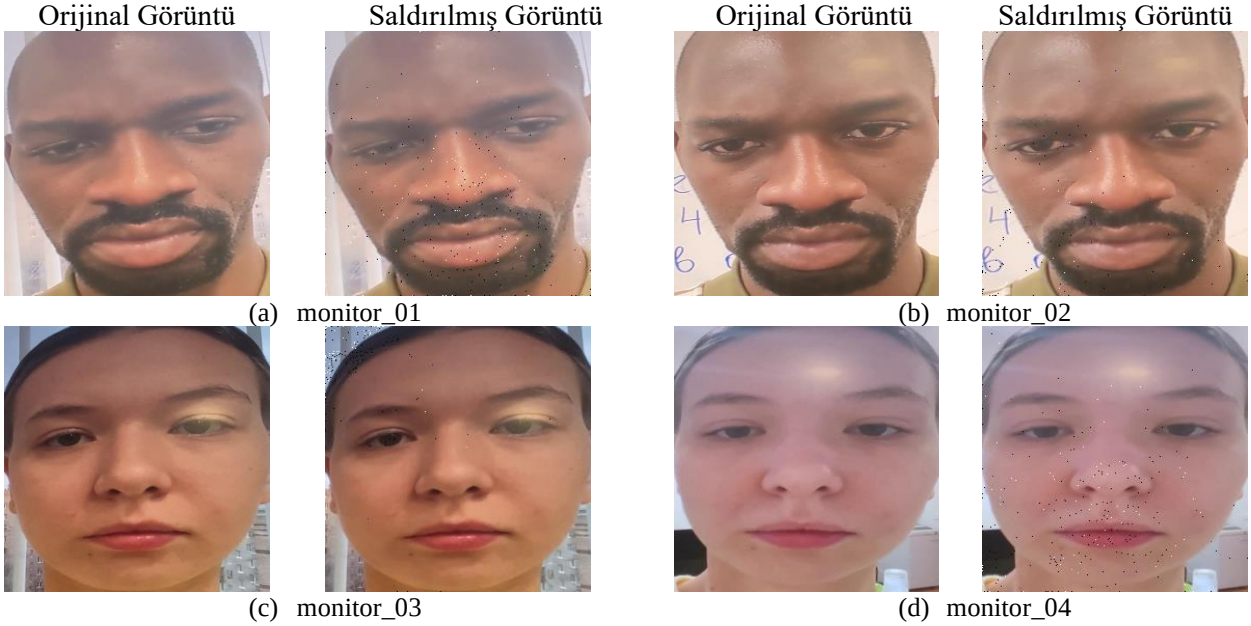
(d)) görüntüsünün hatalı olarak sınıflandırılması için daha fazla sayıda piksel değerinin değiştirilmesi gerekmiştir.



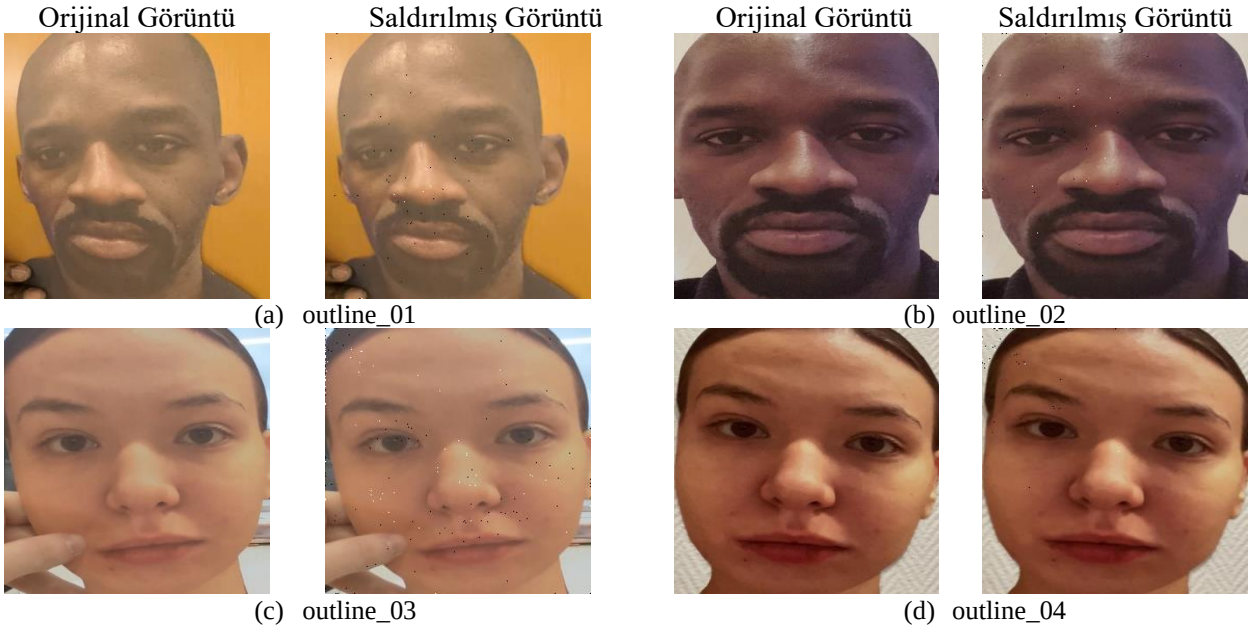

**Şekil 2.** 3B Maske Sınıfına ait Görsel Sonuçlar

Şekil 3, monitör sınıfına ait orijinal ve saldırı uygulanmış görüntüleri göstermektedir. Monitör sınıfından seçilen monitör_02 (Şekil 3 (b)) ve monitör_03 (Şekil 3 (c)) görüntülerinde az sayıda piksel değişikliğe uğratılarak modelin yanıltıldığı görülmektedir. Diğer taraftan monitör_01 (Şekil 3 (a)) ve monitör_04 (Şekil 3 (d)) görüntüleri için daha fazla pikselin değiştirilmesi gerekmiştir.

Çıktı sınıfına ait orijinal ve saldırı uygulanmış görüntüler Şekil 4'te verilmiştir. Çıktı sınıfındaki tüm görüntüler için modelin yanıltılması için gereken piksel sayısı oldukça sınırlı kalmıştır. Diğer sınıflara ait saldırı uygulanmış görüntülerle beraber değerlendirildiğinde DGUA-FAS modelinin çıktı sınıfındaki görüntüleri için daha kolay yanıltılabildiği görülmüştür. Buradan çıktı sınıfındaki görüntülerin kamera karşısındaki gerçek bir kişiyi gösteren görüntülere daha fazla benzediği çıkarımına da varılabilmektedir.



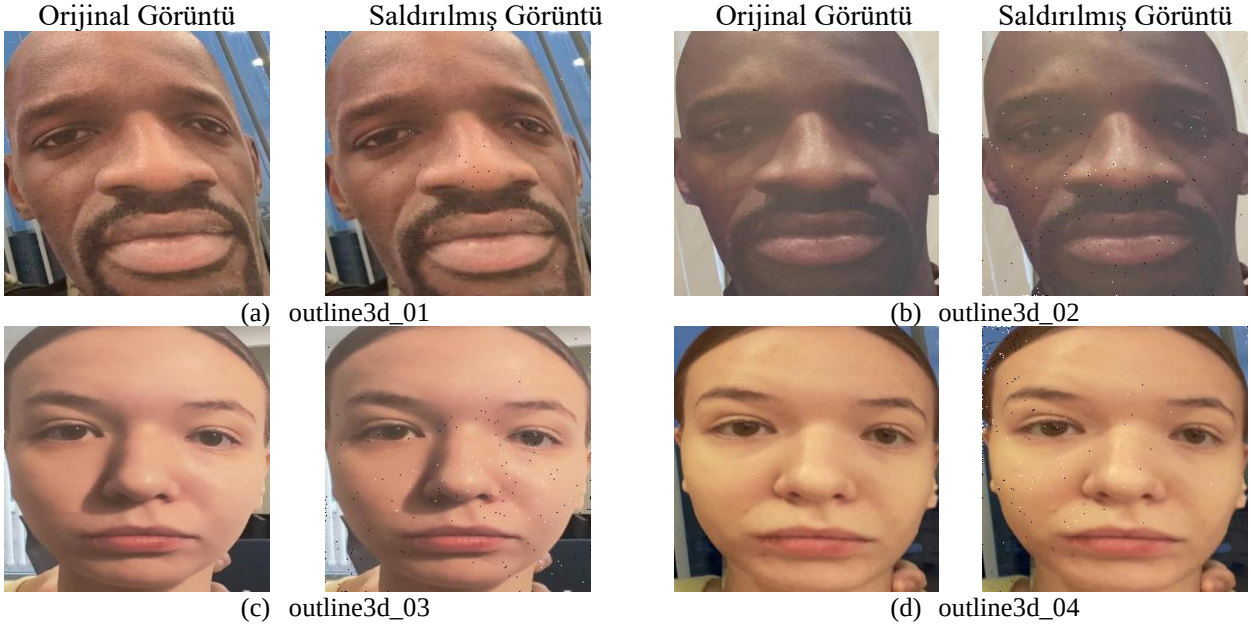

**Şekil 3.** Monitör Sınıfına ait Görsel Sonuçlar




**Şekil 4.** Çıktı Sınıfına ait Görsel Sonuçlar

Veri setinin son sınıfı olan 3B çıktı sınıfından orijinal ve saldırı uygulanmış görüntüler Şekil 5'te görülmektedir. Özellikle Şekil 5 (a)'da görülen outline3d_01 görüntüsü için modelin yanıltılması için az sayıda pikselin değiştirilmesinin yeterli olduğu anlaşılmaktadır. Buna karşın, Şekil 5 (d) ile verilen outline3d_04 görüntüsü için modelin yanıltılması için daha fazla sayıda pikselin değeri değiştirilmiştir.

Bu bölümde verilen sayısal ve görsel sonuçlar beraberce değerlendirildiğinde MIO algoritması tabanlı çekişmeli saldırı üretme yöntemi ile üretilen saldırıların maske, 3B maske, monitör, çıktı ve 3B çıktıya ait *sahte* görüntülere eklenmesiyle elde edilen tüm görüntüler için DGUA-FAS modelinin yanıltıldığı ve bu görüntülerin kamera karşısındaki gerçek bir kişiyi ifade eden *gerçek* sınıfına ait olarak etiketlendiği görülmüştür.




**Şekil 5.** 3B Çıktı Sınıfına ait Görsel Sonuçlar

## SONUÇ

Günümüzde akıllı telefonlardan güvenlik sistemlerine, ödeme uygulamalarından uzaktan sağlık uygulamalarına yaygın bir kullanım alanı olan yüz tanıma ve doğrulama sistemleri için yüz sahteciliğinin

önlenmesi büyük bir öneme sahiptir. Literatürde yüz sahteciliği önleme amacıyla geliştirilmiş çok sayıda derin sinir ağı modeli bulunmaktadır. Bu yöntemlerin farklı sahtecilik yöntemlerini hassas bir şekilde tespit edebilmesi ve girdi görüntülere uygulanacak saldırılara karşı dayanıklı olması beklenmektedir. Bu çalışmada, güncel ve başarılı bir yüz sahteciliği tespit modeli olan DGUA-FAS modeli için MIO algoritması tabanlı kapalı kutu çekişmeli saldırı üretme yöntemi ile saldırılar üretilmiştir. Çalışma kapsamında test edilen çekişmeli örneklerin büyük kısmında DGUA-FAS modelinin görüntüleri hatalı şekilde *gerçek* olarak sınıflandırdığı ve başarısız olduğu gözlemlenmiştir. Elde edilen sonuçlar, yüz sahteciliği tespit modellerinin çekişmeli saldırılara karşı daha dayanıklı hale gelmesi gerektiğini göstermiştir.

Gelecek çalışmalarda kapalı kutu çekişmeli saldırı yöntemlerinin daha fazla yüz sahteciliği tespiti modeline uygulanması ve bu modellerin belirlenen çekişmeli örneklerle yeniden eğitilerek daha dayanıklı hale getirilip getirilemeyeceğinin araştırılması planlanmaktadır.

## TEŞEKKÜR

Bu çalışma, Erciyes Üniversitesi Bilimsel Araştırma Projeleri (ERU-BAP) Koordinasyon Birimi tarafından FBA-2024-13536 numaralı proje kapsamında desteklenmektedir. Bu çalışmada yer alan tüm nümerik hesaplamalar TÜBİTAK ULAKBİM, Yüksek Başarım ve Grid Hesaplama Merkezi'nde (TRUBA kaynaklarında) gerçekleştirilmiştir.

### Çıkar Çatışması

Yazarlar aralarında herhangi bir çıkar çatışması olmadığını beyan eder.

### Yazar Katkısı

Yazarlar makaleye eşit oranda katkı sağlamış olduklarını beyan eder.

## KAYNAKLAR

Akhtar, N., Mian, A., Kardan, N., & Shah, M. (2021). Advances in adversarial attacks and defenses in computer vision: A survey. IEEE Access, 9, 155161-155196.

Arcuri, A. (2018). Test suite generation with the Many Independent Objective (MIO) algorithm. Information and Software Technology, 104, 195-206.

Atoum, Y., Liu, Y., Jourabloo, A., & Liu, X. (2017, October). Face anti-spoofing using patch and depth-based CNNs. In 2017 IEEE international joint conference on biometrics (IJCB) (pp. 319-328). IEEE.

Bartlett, A., Liem, C. C., & Panichella, A. (2023, May). On the Strengths of Pure Evolutionary Algorithms in Generating Adversarial Examples. In 2023 IEEE/ACM International Workshop on Search-Based and Fuzz Testing (SBFT) (pp. 1-8). IEEE.

Dong, Y., Liao, F., Pang, T., Hu, X., & Zhu, J. (2017). Discovering adversarial examples with momentum. arXiv preprint arXiv:1710.06081, 5.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Hong, Z. W., Lin, Y. C., Liu, H. T., Yeh, Y. R., & Chen, C. S. (2023, October). Domain-Generalized Face Anti-Spoofing with Unknown Attacks. In 2023 IEEE International Conference on Image Processing (ICIP) (pp. 820-824). IEEE.

Jourabloo, A., Liu, Y., & Liu, X. (2018). Face de-spoofing: Anti-spoofing via noise modeling. In Proceedings of the European conference on computer vision (ECCV) (pp. 290-306).

Kurakin, A., Goodfellow, I. J., & Bengio, S. (2018). Adversarial examples in the physical world. In Artificial intelligence safety and security (pp. 99-112). Chapman and Hall/CRC.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016, March). The limitations of deep learning in adversarial settings. In 2016 IEEE European symposium on security and privacy (EuroS&P) (pp. 372-387). IEEE.

Sahin, O. (2024). Generation of black-box adversarial attacks using many independent objective-based algorithm for testing the robustness of deep neural networks. Applied Soft Computing, 164, 111969.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

Taşyürek, M., & Gül, E. (2023). Nesne Tespitinde En Uygun Modelin Seçimi İçin Görüntüler Üzerinde Evrişimli Sinir Ağları ile Çekişmeli Saldırı Tespiti. Journal of the Institute of Science and Technology, 13(4), 2353-2363.

Wang, Z., Yu, Z., Zhao, C., Zhu, X., Qin, Y., Zhou, Q., ... & Lei, Z. (2020). Deep spatial gradient and temporal depth learning for face anti-spoofing. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5042-5051).

Yang, J., Lei, Z., & Li, S. Z. (2014). Learn convolutional neural network for face anti-spoofing. arXiv preprint arXiv:1408.5601.

Yu, Z., Zhao, C., Wang, Z., Qin, Y., Su, Z., Li, X., ... & Zhao, G. (2020). Searching central difference convolutional networks for face anti-spoofing. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5295-5305).

Yu, Z., Qin, Y., Li, X., Zhao, C., Lei, Z., & Zhao, G. (2022). Deep learning for face anti-spoofing: A survey. IEEE transactions on pattern analysis and machine intelligence, 45(5), 5609-5631.