







Evrişimli Sinir Ağlarında Transfer Öğrenmesi ile GAN tarafından Üretilen Sahte Görüntü Tespiti

GAN-Generated Fake Image Detection with Transfer Learning in Convolutional Neural Networks

¹Ece ECEMİŞ , ²Kemal GÜNER , ³Umut KURAN , ⁴Emre Can KURAN 

^{1,2,3}Harran Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Haliliye/Şanlıurfa, Türkiye

⁴Bandırma Onyediy Eylül Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Yazılım Mühendisliği Bölümü, Bandırma/Balıkesir, Türkiye

¹eeecemis2@gmail.com, ²kguner@harran.edu.tr,

³ukuran@harran.edu.tr, ⁴ekuran@bandirma.edu.tr

Araştırma Makalesi/Research Article

ARTICLE INFO

Article history

Received : 27 February 2023

Accepted : 22 March 2023

Keywords:

Fake Face Image Detection, Generative Adversarial Networks, Transfer Learning, BFKL Layer

ABSTRACT

The Generative Adversarial Network (GAN) is known as a generative deep learning model. It consists of generator and discriminator structures. Very successful examples of GAN model outputs known as synthetic data are known. It is a very difficult problem to detect synthetic data, which can be used for different purposes, in case of successful generation. In this study, a Laplace filter and a new dissimilarity-based Convolutional Neural Network (CNN) layer is proposed in order to distinguish synthetic and real images, in which different and popular CNN models are used as feature extractors. The success results of the GAN model on different models have been determined. Thus, it has been understood that using CNN models is a suitable alternative for the detection of synthetic data that cannot be distinguished by the naked eye. The best success was achieved with DenseNet with an accuracy rate of 98.75%.

© 2023 Bandırma Onyediy Eylül University, Faculty of Engineering and Natural Science. Published by Dergi Park. All rights reserved.

MAKALE BİLGİSİ

Makale Tarihleri

Gönderim : 27 Şubat 2023

Kabul : 22 Mart 2023

Anahtar Kelimeler:

Sahte Yüz Görüntüsü Tespiti, Çekişmeli Üretken Ağlar, Transfer Öğrenmesi, BFKL Katmanı

ÖZET

Çekişmeli Üretken Ağ (GAN), üretken bir derin öğrenme modeli olarak bilinir. Üretici (generator) ve ayırt edici (discriminator) yapılarından oluşmaktadır. Sentetik veri olarak bilinen GAN modeli çıktılarının oldukça başarılı örnekleri bilinmektedir. Farklı amaçlar ile kullanılabilen sentetik verilerin, başarılı bir şekilde üretilmesi durumunda insan gözü ile tespit edilebilmesi oldukça güç bir problemdir. Bu çalışmada farklı ve popüler Evrişimli Sinir Ağı (CNN) modellerinin öznelik çıkarıcı olarak kullanıldığı, sentetik ve gerçek görüntüleri ayırt eden bu problem için Laplace filtresi ve benzemezlik tabanlı yeni bir CNN katmanı önerilmiştir. GAN modelinin farklı modeller üzerindeki başarı sonuçları tespit edilmiştir. Böylece, gözle ayırt edilemeyen sentetik verilerin tespiti için CNN modellerinden yararlanmanın uygun bir alternatif olduğu anlaşılmıştır. En iyi başarı %98.75 doğruluk oranıyla DenseNet ile elde edilmiştir.

© 2023 Bandırma Onyediy Eylül Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi. Dergi Park tarafından yayınlanmaktadır. Tüm Hakları Saklıdır.

ORCID: ¹0009-0002-8214-4499

²0000-0002-7506-6114

³0000-0001-5508-1870

⁴0000-0002-0987-3866

1. GİRİŞ

Çekişmeli üretici ağlar (Generative Adversarial Networks, GAN) sentetik veri üretimi amaçlı geliştirilmiş modellerdir. GAN yöntemlerinin kullanımının yaygınlaşması ile birlikte oldukça farklı sahalarda çok başarılı sentetik veriler üretebilmektedir. Özellikle görüntü üretimi konusunda şaşırtıcı örnekleri mevcuttur. Bu verilerin göz ile ayırt edilebilmesi çoğu durumda mümkün olmamaktadır [1]. Sentetik veri üretimi yeterli veri bulunmayan alanlarda çalışma yapmak isteyen araştırmacılar için önemli bir alternatiftir. Örneğin derin öğrenme modelleri eğitimleri için büyük miktarda veriye ihtiyaç duymaktadır, bazı problemler için oldukça sınırlı miktarda veri bulunabilmektedir. Bu öznelikleri içeren verilerin uzmanlar tarafından dahi oluşturulabilmesi mümkün olmayabilmektedir. [2]. Ayrıca eğlence amaçlı ya da insanları yanıltma amaçlı olarak da sentetik bilgilerin kullanımı mümkün olabilmektedir [3]. Sentetik veri üretimi için kullanılan GAN yalnızca yeterli miktarda eğitim verisine ihtiyaç duyar. Bu verilerin etiketli olmalarına dahi gerek duyulmamaktadır. Gerçekleşen öğrenme işleminde eğitim verileri üzerinden oldukça karmaşık öznelikler, istatistikler çıkarılmaktadır [4].

GAN modellerinin yaptıkları iş temel olarak bir üretken modelleme sistemi oluşturmaktadır. Üretken modelleme girdi verilerinden öğrendiği verilerinde yer alan kalıpları, örüntüleri öğrenip bu verilere benzer sentetik veriler üreten bir stratejidir. Bir derin sinir ağı olan GAN modelleri, bunlardan farklı olarak üretici (Generative, G) ve ayırıcı (discriminator, D) olmak üzere iki farklı derin ağ içermektedir [5]. İsmi bu iki ağı çekişmeli olarak çalışmasından almaktadır. GAN modelleri ile literatürde oldukça farklı sahalarda başarılı çalışmalar gerçekleştirilmiştir. Stoll vd. tarafından önerilen çalışmada GAN modelleri kullanılarak işaret dili verileri üretilmiştir [6]. Nazeri, GAN modelleri ile gri seviyeli görüntülerin renklendirilmesi amacıyla bir yöntem geliştirmiştir [7]. Fabbri ise GAN modellerinin su altı görüntülerinin iyileştirilmesi için kullanılan bir çalışma gerçekleştirmiştir [8]. Antipov ise GAN modellerini yüz yaşlandırma uygulaması oluşturma amacıyla kullanmıştır [9].

Marra vd. başarılı GAN sentetik verilerinin özellikle görüntülerinin insan gözü ile gerçek görüntülerden ayırmaları nedeniyle sentetik veri tespiti yapmak için de çalışmalar gerçekleştirilmiştir. GAN görüntülerinin tespiti problemi için derin öğrenme tabanlı klasik yöntemlerin performansları karşılaştırılmıştır. Sahte görüntülerin ve videoların sosyal medyada yayılmasının hızla büyüyen bir sorun olmasından kaynaklı olarak sahte görüntülerin tespit edilmesinin oldukça önemli olduğu vurgulanmıştır. GAN üretimi görüntülerin tespit edilmesi için Marra tarafından önerilen bir çalışmada, sosyal medya üzerinde yer alan görüntüler kullanılmaktadır. Bu yöntemde gerçek görüntüler GAN modeli aracılığı ile görüntüden görüntüye çeviri işlemine maruz bırakılmıştır. Daha sonra GAN üretimi ile gerçek görüntüler CNN modelleri vasıtasıyla sınıflandırılmıştır. [10]. Yine, model olarak Cozzolino 2017 [11] kullanılarak yapısal kısaltmalar kaldırılmış ve ağı ince ayar yapılarak performansı optimizasyonu yapılmıştır. Diğer bir model olan Bayar2016 [12] ile evrimsel katmanda yüksek geçirgen filtreler kullanılan bir CNN modeli kullanmışlardır ve

Nataraj vd. GAN ürettiği sentetik ile gerçek verileri tespit etmek için ortak oluşum ve derin öğrenmenin bir kombinasyonu kullanılarak GAN tarafından oluşturulan sahte görüntülerin tespit etmek için yöntem önerilmiştir. Pksel alanındaki üç renk kanalında eş oluşum matrislerini (co-occurrence matrix) çıkarılmış ve derin bir CNN modeli kullanılarak bir model eğitilmiştir. Görüntünün RGB kanallarından ortak oluşum matrisleri hesaplanır. Bu ortak oluşum matrisleri derin öğrenme modeli vasıtasıyla sınıflandırılarak GAN üretimi olan görüntüler tespit edilmektedir. Bu çalışmada görüntüler ortak oluşum matrislerine dönüştürülerek eğitilmekte önerilen yöntem ile benzer olarak ikili sınıflandırma şeklinde çalışan bir yöntem olup, yine benzer şekilde CNN tabanlı yöntem fakat bizim çalışmamızda ise görüntüler doğrudan verilmiştir. Bu modelin bu alanda bilinen iki veri kümesinde de başarılı sonuçlar ürettiği görülmüştür [13].

McCloskey ve Albright, bir GAN modelinin üretici ağı yapısını analiz ederek ağı işleminin gerçek kamera görüntüsünün yapısından farklı olduğunu iki farklı doğrulama şekli ile (renk ve doygunluk) ile GAN üretimi görüntüleri tespit etmişlerdir. Bunu da görüntülerin GAN üretimi olup olmadığının tespitinde kullanmışlardır. GAN'lerden gelen üretici ağlarının öznelik temsillerini kırmızı, yeşil ve mavi piksel yoğunluklarına dönüştürme şekli ile ilgili farklı özelliğin etkinliği belirlenmiş ve değerlendirilmiştir. Özellikle aşırı pozlanmış piksellerin frekansına dayanan nispeten basit bir adli bilişimin, bağımsız olarak oluşturulmuş bir sorgulama veri seti ile deneyler yoluyla GAN tarafından oluşturulan ve kamera görüntüleri arasında iyi bir ayırım sağlandığı gösterilmiştir. Üretici ağı özneliklerini kullanarak belirlenmiş iki özel öznelik bilgisi kullanılarak sınıflandırma işlemi gerçekleştirilmiştir [14].

Bu çalışmada GAN modelleri vasıtasıyla üretilen sahte insan görüntülerini tespit edebilmek için CNN tabanlı bir görüntü sınıflandırma yöntemi önerilmiştir. CNN modellerinin hangisinin bu problem için daha başarılı performans sergilediğini saptayabilmek amacıyla ayrıca sentetik ve gerçek görüntüler içeren yeni ve özgün bir veri kümesi oluşturulmuş ve önerilen yöntemin eğitim ve test işlemlerinde kullanılmıştır. Diğer çalışmalarda farklı yaklaşımlar kullanılmıştır, fakat görülmüştür ki sentetik görüntüler ve gerçek görüntüler arasındaki bazı düşük seviyeli özneliklerin sınıflandırmaya dahil edilmesi daha başarılı sonuçlara sebep olacaktır. Bu sebeple yeni bir özelleştirilmiş katman önerilmiştir. Çalışmanın literatüre ana katkıları şunlardır:

- İnternet ortamında hazır bulunan görüntülerden yeni bir veri seti derlenmiştir.
- Hafif bir blok eğitilmiş modellerin sonuna eklenerek yüksek bir performans elde edilmiştir.

- Yüksek geçiren Laplace filtreye ve bensemezliğe (dissimilarity) dayalı yeni bir katman önerilmiştir. Önerilen bu katman, sentetik görüntüler ve gerçek görüntülere karşı ağırlık farklı tepki vermesini sağlayarak ağırlık başarısını artırmıştır.

Bu çalışmanın organizasyonu şu şekildedir: ikinci bölümde CNN ve kullanılan CNN modelleri hakkında genel bilgiler verilmektedir. Üçüncü bölümde önerilen yöntem ve kullanılan veri kümesi ile ilgili bilgiler verilmektedir. Dördüncü bölümde ise deneysel sonuçlar ve yorumlamalar yer almaktadır. Beşinci bölümde ise deneysel sonuçlar yorumlanmış, önerilen yöntemin avantajlarından ve dezavantajlarından bahsedilmiştir. Ayrıca gelecekteki çalışmalara yönelik önerilerde bulunulmuştur.

2. EVRİŞİMLİ SİNİR AĞLARI (CNN)

Evrışimli sinir ağları genellikle görüntü verileri üzerinde kullanılmaktadır. Evrişim ya da konvolüsyon (convolution) adı verilen filtreler aracılığıyla giriş görüntüsünden öznelik haritaları çıkarılmaktadır. Problem türüne göre öznelik haritalarının boyutu büyütüp küçültülebilme işlemleri de gerçekleştirilen CNN modellerinin temel bileşenleri evrişim katmanı, aktivasyon fonksiyonu, havuzlama katmanı, tam bağlı katman, seyreltme katmanı ve yığın normalizasyonu katmanı olarak listelenebilir. Aşağıda bu bileşenler hakkında kısa bilgiler aktarılmıştır [15]–[17].

Evrışim Katmanı (convolutional layer): CNN modellerinin en önemli bileşenidir. Bu katmanın görevi kendisine gelen girişten sahip olduğu filtre sayısı, filtre büyüklüğü gibi parametreler ölçüsünde öznelik çıkarmak ve bu öznelikleri kendinden sonraki katmana iletmektir.

Aktivasyon Fonksiyonu (activation function): CNN modellerinde oldukça önemli bir göreve sahiptir. Çıkış sinyalinin doğrusal, basit bir fonksiyon olmasını engellemek için kullanılan aktivasyon fonksiyonları modelin öğrenme performansının artması için kullanımına ihtiyaç duyulmaktadır. En yaygın kullanılan aktivasyon fonksiyonları ReLu ve sigmoid fonksiyonlarıdır.

Havuzlama Katmanı (pooling layer): Veri boyutunu ve parametre sayısını azaltmak amacıyla kullanılan havuzlama katmanı en büyüğe havuzlama ve ortalamaya havuzlama olmak üzere iki çeşiti bulunmaktadır.

Tam Bağlı Katman: Klasik bir yapay sinir ağıdır. Genellikle CNN modellerinin çıkışında kullanılarak sınıflandırıcı olarak kullanılmaktadır.

Yığın Normalizasyonu (Batch Normalization): CNN modelinde yer katmanların aynı anda öğrenmesine yardımcı olduğu için öğrenmeyi artırdığı, eğitimi hızlandırdığı bilinmektedir. Bu nedenle yaygın olarak kullanılmaktadır.

2.1. Kullanılan CNN Modelleri

Önerilen yöntemde ön-eğitilmiş olarak kullanılan, görüntü sınıflandırma konusunda oldukça popüler CNN modelleri Resnet18, Resnet50, VGG19, AlexNet, DenseNet ve SqueezeNet denenmiştir. Literatürde bu ağlarla ilgili çalışmalar oldukça yaygındır [18] Bu kısımda bu modellerle ilgili kısa bilgiler verilmektedir.

1) Resnet18 [19]: Residual Neural Network kısaca adı resnet olarak bilinen ağ mimarisinin 18 derin katmana sahip olan evrişimli sinir ağıdır. Tipik olarak evrişim, yığın normalizasyonu ve ReLu aktivasyon fonksiyonu içeren bloklardan meydana gelmektedir. Girişte 224x224x3 boyutlarında RGB görüntüler alır, çıkışta ise 512 uzunluğunda bir öznelik vektörü üretilmektedir. Yaklaşık olarak 11 milyon parametre eğitilmektedir [20].

2) Resnet50 [19]: Yine bir resnet versiyonu olan resnet50'de adından da anlaşılacağı gibi 50 derin katman yer almaktadır. Yaklaşık olarak 23 milyon eğitilebilir parametre içermektedir. Girişte 224x224x3 boyutlarında RGB görüntüler alır, çıkışta ise 2048 uzunluğunda bir öznelik vektörü üretilmektedir. Yaklaşık 23 milyon eğitilebilir parametre içermektedir. ResNet 50 ile yeni görüntüler sınıflandırılabilir.

3) VGG19 [21]: 16 evrişim ve 3 tam bağlı katmandan oluşmaktadır. Yapısında iki adet iki evrişim bir en büyüğe havuzlama bloğu, üç adet de dörtlü evrişim ve bir en büyüğe havuzlama bloğu içermektedir. VGG19 modeli girişte 224x224x3 boyutlarında RGB görüntü alıp, çıkışta 7x7x512 boyutlarında öznelik vektörleri üretmektedir. Yaklaşık 144 milyon eğitilebilir parametresi vardır.

4) AlexNet [22]: Alex Krizhevsky tarafından tasarlanan bir CNN modelidir. Toplamda 5 evrişim katmanı ve 3 tam bağlı katman içermektedir. Diğer yöntemlere kıyasla daha sade bir mimarisi olduğu söylenebilir. Bu model girişte 227x227x3 boyutlarında RGB görüntüler alır, çıkışta ise 9216 elemanlı bir öznelik vektörü elde edilir. Yaklaşık 60 milyon parametre içermektedir.

5) DenseNet [23]: DenseBlock adı verilen evrişim katmanları kümeleri içermektedir. Geçiş katmanı (Transition Layer) adı verilen yapılar vasıtasıyla bileşenler arası veri alışverişini sağlayan bu model karmaşık bir mimaride tasarlanmıştır. 224x224 boyutundaki girişlere karşılık 1024 elemanlı bir öznelik vektörü üretir. Yaklaşık 7 milyon eğitilebilir parametre içermektedir.

6) SqueezeNet [24]: 3x3 boyutlarında filtreler tercih edilen bu modelde eğitilebilir parametre sayısı ve çıkış boyutları diğer modellere göre daha az miktardadır. Maliyeti düşürme amacıyla üretilmiş bir modeldir. 227x227 boyutlarında giriş görüntülerine karşılık, 1000 elemanlı bir öznelik vektörü üretmektedir. Yaklaşık 1.2 milyon eğitilebilir parametre içermektedir.

7) GoogleNet [25]: Inception adı verilen derin öğrenme katmanları temel alınarak önerilmiş bir evrişimli sinir ağı modelidir. Bu mimarideki temel motivasyon, ağ içindeki işlem kaynaklarının gelişmiş bir biçimde kullanımınıdır.

Yani hesaplama bütçesi sabitken ağdaki derinlik ve genişliğin artırılabilmesine imkan sağlamıştır. 22 katmanlı bir yapıya sahiptir.

3. MATERYAL VE YÖNTEM

Bu bölümde kullanılan materyal ve yöntemlerden bahsedilmiştir. Bölüm 3.1’ de veri setine değinilmiş, özellikleri verilmiştir. Bölüm 3.2’de uygulama detaylarından ve çalıştırma ortamında bahsedilmiştir. Bölüm 3.3 ve Bölüm 3.4’te kullanılan tekniklerin temelleri hakkında bilgiler verilmiş, Bölüm 3.5 ve Bölüm 3.6’ da ise önerilen yöntem detaylarıyla birlikte açıklanmıştır.

3.1. Veri Seti

Bu çalışma kapsamında kullanılan veri kümesi sentetik ve gerçek insan görüntülerinden oluşmaktadır. Görüntüler çeşitli kaynaklardan hazır olarak elde edilmiştir. Sentetik görüntüler thispersondoesnotexist [26] web sitesinden elde edilmiş olup, gerçek görüntüler Flickr [27] platformunda yer alan lisanssız fotoğraflar toplanarak elde edilmiştir. Kullanılan veri kümesinde 1000 sentetik, 1000 gerçek görüntü olmak üzere toplamda 2000 görüntü bulunmaktadır. Bu görüntülerin %80’i (1600 adet) eğitim, %20’si (400 adet) de test için kullanılmıştır. Görüntüler PNG formatında olup 3 kanallı RGB (Renkli) görüntülerdir. Boyutları 256x256 olarak ayarlanmıştır ve her bir kanal 8-bit derinliğe sahiptir. Her kanal için piksel değerleri [0-255] dahili aralığında değişmektedir. Görüntülerin etiketleri dosya adlarından çıkarılmaktadır. Gerçek görüntüler için dosya adında “g”, sahte görüntüler için ise “s” kullanılmıştır. Veri seti daha sonra dosyalardan görüntüler okunarak hazır hale getirilmiştir. Şekil 1’de veri kümesi içerisinde örnek görüntüler gösterilmiştir.



Şekil 1. Veri setinden örnek görüntüler: (a-c): Gerçek Görüntüler, (d-f): Sentetik GAN görüntüleri.

3.2. Uygulama Detayları

Kullanılan makine Nvidia RTX3050 Ti grafik işleme birimine, 16 GB RAM belleğe ve 2.90 GHz işlemciye sahip olup, eğitim ve test işlemleri Cuda ile gerçekleştirilmiştir. Epoch sayısı 10 olarak tercih edilmiştir. Kayıp fonksiyonu olarak ikili sınıflandırma yapıldığı için İkili Çapraz Entropi (İÇE-Binary Cross Entropy) [28] tercih edilmiş olup, ağ eğitimi için Stokastik Gradyan İnişi (SGİ-Stochastic Gradient Descent) yöntemi tercih edilmiştir. Momentum değeri 0.9 ve öğrenme oranı 0.005 olarak eğitimler gerçekleştirilmiştir. Uygulama Python dilinde, PyTorch kullanılarak ve Windows 10 işletim sisteminde gerçekleştirilmiştir. Uygulama detayları daha ayrıntılı olarak Tablo 1’de verilmiştir.

Tablo 1. Uygulama detayları.

Cihaz Özellikleri	Ekran Kartı	Nvidia RTX3050 Ti
	İşlemci	AMD Ryzen 7 4800H with Radeon Graphics
	RAM	16 GB
Hiper parametreler	Epoch (Adım) Sayısı	10
	Öğrenme Katsayısı	0.005
	Momentum	0.9
	Toplu Boyut (Batch Size)	4
İşletim Sistemi Bilgileri	İşletim Sistemi	Windows 10
	Sistem Türü	64-bit
Kullanılan Kütüphane / Programlama Dili	PyTorch / Python	
Kayıp Fonksiyonu	İkili Çapraz Entropi	
Optimize Edici Fonksiyon	Stokastik Gradyan İnişi	
Aktivasyon Fonksiyonu	Son Katman	Sigmoid
	Ara Katmanlar	ReLU (ReLU katmanları)

Bölüm 4'te verilen çalışma süreleri, Tablo 1'de verilen donanım özelliklerinden farklı özellikleri sahip bir cihazda değişebilir. Daha yüksek doğruluk düzeyleri için hiper parametreler değiştirilebilir, özellikle adım sayısı değiştirilerek ağına daha uzun bir süre eğitilmesi sağlanabilir. Öğrenme katsayısı küçültülerek veya artırılarak daha farklı yakınsamalar gerçekleştirilebilir. Görece çok büyük öğrenme katsayıları, optimum ağırlıkları kaçırmaya sebep olabilir. Momentum, SGI için dalgalanmaları engelleme yardımcı olup, modelin daha tutarlı olmasını sağlar. Bu çalışmada 0.9 seçilmiştir, daha düşük momentumlar iterasyonlar boyunca model başarısında dalgalanmalara sebep olabilir. İkili Çapraz Entropi, iki sınıflı bir sınıflandırma yapıldığı için tercih edilmiştir. Benzer şekilde Sigmoid fonksiyonu da son katmanda bu sınıflandırma için yeterlidir.

3.3. Gri Düzey Eş Oluşum Matrisi (GDEOM) Öznitelikleri

GDEOM, görüntü işleme ve analizinde kullanılan matematiksel bir araçtır. İlk olarak 1973'te Haralick, Shanmugam ve Dinstein tarafından önerilmiştir [29] ve o zamandan beri görüntü doku analizleri için popüler bir yöntem haline geldi. GDEOM, bir görüntüdeki piksel çiftleri arasındaki uzamsal ilişkiyi tanımlayan bir matristir. Spesifik olarak, belirli bir yoğunluk değerine ve uzamsal ilişkiye sahip piksel çiftlerinin bir görüntüde ne sıklıkta meydana geldiğini tutar. Bu matrisi analiz ederek, görüntü sınıflandırma, segmentasyon vb. uygulamalar için yararlı olan bir dizi doku özelliği çıkarılabilir. Önce yatay, dikey veya köşegen gibi bir mesafe ve yön seçilerek ve ardından belirli bir yoğunluk değerine ve bu mesafe ve yönde uzamsal ilişkiye sahip piksel çiftlerinin sayısı sayılarak hesaplanır. Bu işlem, yoğunluk değeri ve uzamsal ilişkinin olası her bir kombinasyonu için tekrarlanır ve her öğenin belirli bir piksel değeri çiftinin oluşma sıklığını temsil ettiği bir kare matrisle sonuçlanır. Matris elde edildikten sonra kontrast, enerji, homojenlik, benzemezlik ve entropi gibi bir dizi doku öznitelikler matristen çıkarılabilir. Bu farklı öznitelikler, görüntünün farklı bölgelerini ayırt etmek veya görüntüyü sınıflandırmak için kullanılabilir. Bu çalışmada, 0 derece, 45 derece, 90 derece ve 180 derece olan açılar kullanılarak GDEOM elde edilmiştir, oluşan matristen Denklem 1 ile verilen benzemezlik özniteliği kullanılmıştır. Benzemezlik, görüntüdeki iki bitişik piksel çiftinin arasındaki kontrast farkını ölçer, ve böylece birbirine komşu olan iki pikselin ne kadar farklılık gösterdiğini anlayabiliriz.

$$\text{Benzemezlik} = (1/N) \times \sum_{i,j=0}^{N-1} P_{i,j} |i-j| \quad (1)$$

Denklem 1'de N, görüntüdeki toplam piksel sayısıdır, $|i-j|$ iki pikselin gri seviye değerleri arasındaki mutlak farktır ve $P(i, j)$, belirli bir mesafe ve yönde gri seviye değerleri olan i ve j 'nin birlikte olma olasılığıdır. Burada Benzemezlik, iki bitişik pikselin farklı gri seviye değerlerine sahip olma sayısının görüntüdeki toplam piksel sayısına bölünmesiyle ölçülür.

3.4. Laplace Filtresi

Laplace filtresi [30], görüntü işleme ve bilgisayar görüşünde yaygın olarak kullanılan bir görüntü filtresi türüdür. Adını 19. yüzyılın başlarında Laplace operatörünü tanıtan matematikçi Pierre-Simon Laplace'dan almıştır. Laplacian filtresi, bir görüntüdeki kenarları ve ince ayrıntıları vurgulamak için geliştirilmiştir ve bu da onu görüntü keskinleştirme, kenar algılama ve öznitelik çıkarma gibi çeşitli uygulamalar için kullanışlı bir araç haline getirir. Laplace filtresi, bir görüntüyü 3x3 veya 5x5 sayı matrisi olan Laplace filtresiyle evriştirerek (konvolüsyon işlemi) çalışır. Filtre, farklı nesnelere arasındaki kenarlar veya sınırlar gibi yoğunluğun hızla değiştiği görüntü alanlarını vurgulamak için tasarlanmıştır. Laplace filtresi, yoğunluk gradyanındaki değişim oranının bir ölçüsü olan görüntü yoğunluğunun ikinci türevini hesaplayarak çalışır. Sonuçlar kullanılan renk uzayına bağlı olarak değişebilir de, Laplace filtresi hem gri tonlamalı hem de renkli görüntülere uygulanabilir. Bu çalışmada kullanılan Laplace matrisi, aşağıda Denklem 2 ile verilmiştir.

$$\text{LAP} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2)$$

3.5. Benzemezlik Filtrelili Konvolüsyonel Laplace (BFLK) Katmanı

Bu çalışma kapsamında geliştirilen yöntemde yeni ve özelleştirilmiş bir katman olan BFLK katmanı önerilmiştir. BFLK katmanı, ağına, sentetik yüz görüntüleri ve gerçek yüz görüntüleri arasındaki ayrımı yapabilme yeteneğini geliştirmek için tasarlanmıştır. Gatys, Ecker ve Bethge tarafından yapılan bir çalışmada [31], doğal görüntülerin üstel kanun dağılımını [32] takip ettiği görülmüştür. Bu da doğal görüntülerin düşük frekanslı bileşenlere ve daha az yüksek frekanslı bileşenlere sahip olduklarını gösterir. Buna karşılık, sinir ağları vb. tarafından oluşturulan sentetik görüntüler, doğal görüntülerle aynı fiziksel kısıtlamalara sahip olmadıkları için daha yüksek frekanslı bileşenlere sahiptir. Her ne kadar doğal görüntüler taklit edilse de doğal görüntülerden farklı bir dokuya veya görünüme sahip sentetik görüntüler ortaya çıkabilmektedir. Yapılan bir çalışmada [33], GAN ile üretilen görüntülerin tespiti için GDEOM kullanılmıştır. GDEOM, sahte olan görüntülerdeki istatistiksel öznitelikleri elde etmek için faydalı bulunmuştur. Bu çalışmada, GDEOM ve Laplace filtresi birleştirilerek doğal olan görüntülerin ve sahte görüntülerin kolayca ayırt edilmesini sağlayan bir katman oluşturulmuştur.

BFLK katmanı, basitçe iki ayrı öznitelikten ve ağırlık yapılandırmasından oluşur: girdiye tamamen duyarlı olan konvolüsyon ve girdiye duyarlı olan konvolüsyon. Her iki konvolüsyon işlemi için konvolüsyon katmanı

filtrelerinin ağırlıkları ayırdır. Girdiye duyarlı olan konvolüsyon için Denklem 3 ile verilen delta matrisi kullanılır ve Denklem 4 ile hesaplanır:

$$M_{\Delta} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (3)$$

$$\text{Çıktı}(i,j) = \sum_{k,l} (\text{girdi}(i+k-1, j+l-1) * M_{\Delta}(k,l)) \quad (4)$$

Denklem 4 ile verilen çıktı değeri, konvolüsyon sonucu oluşan çıktıdır. Girdi ve M_{Δ} arasında konvolüsyon işlemi yapılır ve ağırlıklar güncellenir. Girdiye duyarlı konvolüsyon işlemi ise Denklem 5 ile verilebilir:

$$\text{Çıktı}(i,j) = \sum_{k,l} (\text{girdi}(i+k-1, j+l-1) * \text{LAP}(k,l)) \quad (5)$$

Denklem 5 ile verilen çıktı değeri, girdi ile Denklem 2 ile verilen Laplace matrisinin evriştirilmesi sonucu elde edilmektedir. Böylece duyarlı olan ağırlıklar güncellenir. Bu iki ayrı işlem bu şekilde açıklanabilir. Duyarlılık, her parça (batch) ile model eğitildikçe hesaplanmaktadır. Her parça geldikçe konvolüsyonel katmanda hesaplamalar yapılır. Hesaplama için Denklem 1 ile verilen Benzemezlik metriği kullanılır. Parçadaki her görüntü için 0 derece, 45 derece, 90 derece ve 135 derece ile ilgili olan Benzemezlik metriklerinin ortalaması alınır. Daha sonra tüm parça için Benzemezlik ortalaması alınır. Bu ortalama belli bir eşik değerinin üstündeyse, duyarlı konvolüsyon aksi halde duyarlı konvolüsyon uygulanır ve ağırlıklar güncellenir. Böylece sahte görüntülere daha duyarlı bir katman oluşturulmuş olur. Benzemezlik filtresi, B_s parça boyutu, G_0 , G_{45} , G_{90} , G_{135} sırasıyla 0 derecelik, 45 derecelik, 90 derecelik ve 135 derecelik GDEOM olmak üzere, Denklem 6 ile verilebilir:

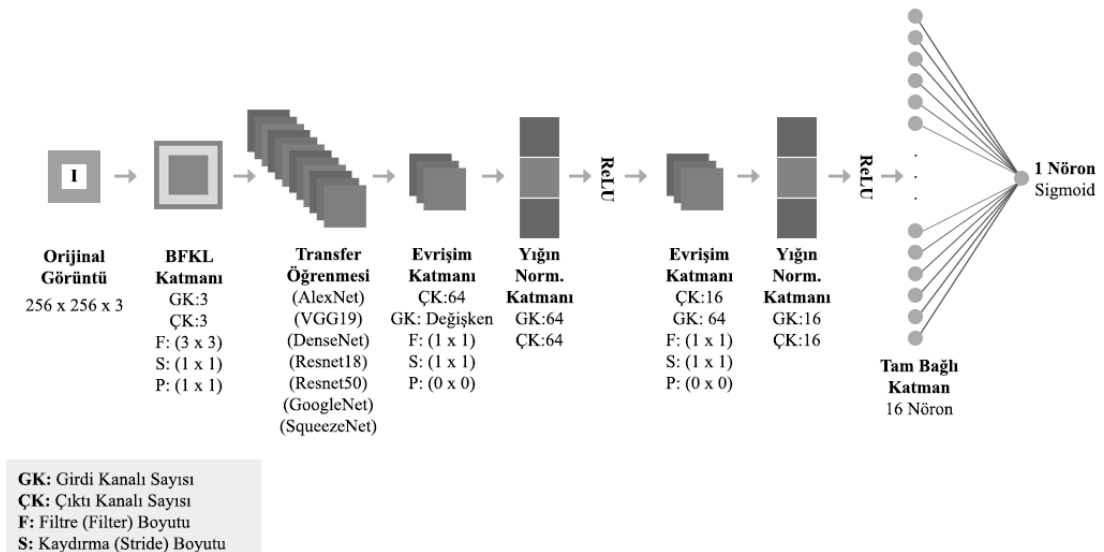
$$\text{Çıktı}(i,j) = \left(\frac{1}{B_s}\right) \times \sum_{i=1}^{B_s} \left(\frac{1}{4}\right) \times \text{Benzemezlik}(G_0, G_{45}, G_{90}, G_{135}) \quad (6)$$

Ağ, tek bir örnek üzerinde çalıştırılmak istenildiğinde benzer şekilde sadece bu örnek için 4 farklı açıda benzemezlik hesaplanır ve katmandan geçirilir.

3.6. Transfer Öğrenme ile BFKL Katmanının Birleştirilmesi

Transfer öğrenme, eğitilmiş bir modelin tekrar kullanılmasını hedefler ve böylece oluşan model yeni örnekleri daha iyi sınıflandırır [34]. Önceden eğitilmiş bir model kullanırken, eğitim sırasında güncellenmelerini önlemek için modelin bazı veya tüm katmanlarını dondurmak popüler bir uygulamadır. Bu katmanlar dondurularak eğitim süreci hızlandırılabilir ve veriyi ezberleme önlenir. Bu çalışmada, sadece son katman dondurularak eğitilmiş modellerden öznetelikler çıkartılmıştır. Böylece binlerce veya milyonlarca veri üzerinde eğitilen bir modelin çıkardığı özneteliklerden fayda sağlanabilmektedir.

Şekil 2 incelendiğinde soldan ikinci sırada yer alan, en büyük boyutlu bileşen olarak göze çarpan ön-eğitilmiş model, Bölüm 2.1’de belirtilen ön eğitilmiş CNN modellerini temsil etmektedir. Her bir çalışmada öneğitilmiş model yerine sırasıyla kullanılmıştır. Ön-eğitilmiş modelin çıkışı sırasıyla evrişim, yığın normalizasyonu, ReLU aktivasyon fonksiyonundan oluşan iki bloktan geçerek tam bağlı katmana iletilmektedir. Yığın normalizasyonu modelin daha hızlı yakınsamasını sağlar [35]. Aynı zamanda modelin veriyi ezberlemesini engeller. ReLU ağı lineer olmayan bir yapı kazandırır. Tam bağlı katman aracılığıyla sınıflandırması gerçekleştirilen görüntüler için ağı ürettiği çıkışlar, sigmoid aktivasyon fonksiyonu ile 0-1 aralığına ölçeklenmiştir. Bu sayede çıkış için 0.5’ten küçük değerler gerçek, 0.5’ten büyük değerler ise sentetik görüntüler anlamına gelmektedir.



Şekil 2. Önerilen yöntemin mimarisi.

Şekil 2 ile verildiği üzere, girdi görüntüsü girdi katmanına verilir. Daha sonra BFKL katmanı ile filtreleme yapılır. Bu yapıldıktan sonra, ilgili transfer öğrenme mimarisine BFKL katmanında uygulanan konvolüsyon sonucu iletilir. Burada boyutlar spesifiktir, mimariden mimariye değişmektedir. Transfer öğrenmesi modelinden sonra gelen ve 2 defa tekrarlayan (sırasıyla evrişim katmanı, yığın normalizasyonu katmanı ve ReLU katmanı) katmanlar bulunmaktadır. Evrişimli katmana, hazır modelden çıkartılan öznetelikler girdi verilir. Evrişim katmanının girdisi, modelin çıktısına göre değişmektedir (256, 512 vb.). İlk evrişimli katmanın 64 çıktı kanalı vardır. Bu katmandan yığın normalizasyonu katmanına 64 girdi (öznetelik) boyutuna sahip bir girdi verilir. Daha sonra bu öznetelikler ReLU katmanına iletilir. 2. Blokta benzer şekilde, Evrişim katmanı için 64 girdi kanalı ve 16 çıktı kanalı bulunurken, yığın normalizasyonu katmanı için ise 16 girdi boyutu mevcuttur. En sonunda tüm öznetelikler 16 nöronlu tam bağlı katmana iletilir ve sigmoid aktivasyon fonksiyonu ile ağın çıktısı belirlenir.

4. DENEYSEL SONUÇLAR VE TARTIŞMA

Önerilen yöntem üzerinde farklı öznetelik çıkarıcı yöntemler transfer öğrenmesi mimarileri ile denenmiş ve ikili sınıflandırma doğruluğuna ilişkin sayısal değerler Tablo 2’de paylaşılmıştır. Sonuçlar değerlendirilirken Doğruluk (Accuracy), Hassasiyet (Sensitivity), Kesinlik (Precision), F1 skoru olarak bilinen dört metrikten yararlanılmıştır. Doğruluk, Denklem 7’de verilmiştir. Hassasiyet ve Kesinlik ise sırasıyla Denklem 8 ve Denklem 9 ile verilmiştir. Ayrıca, öznetelik dengesi veri setlerine karşı daha güvenilir bir metrik olan [36] ve Denklem 10 ile verilen F1 skoru kullanılmıştır.

$$\text{Doğruluk} = \frac{TP+TN}{TP+FP+TN+FN} \quad (7)$$

$$\text{Hassasiyet} = \frac{TP}{TP+FN} \quad (8)$$

$$\text{Kesinlik} = \frac{TP}{TP+FP} \quad (9)$$

$$F1 = 2 \times \frac{\text{Kesinlik} \times \text{Hassasiyet}}{\text{Kesinlik} + \text{Hassasiyet}} \quad (10)$$

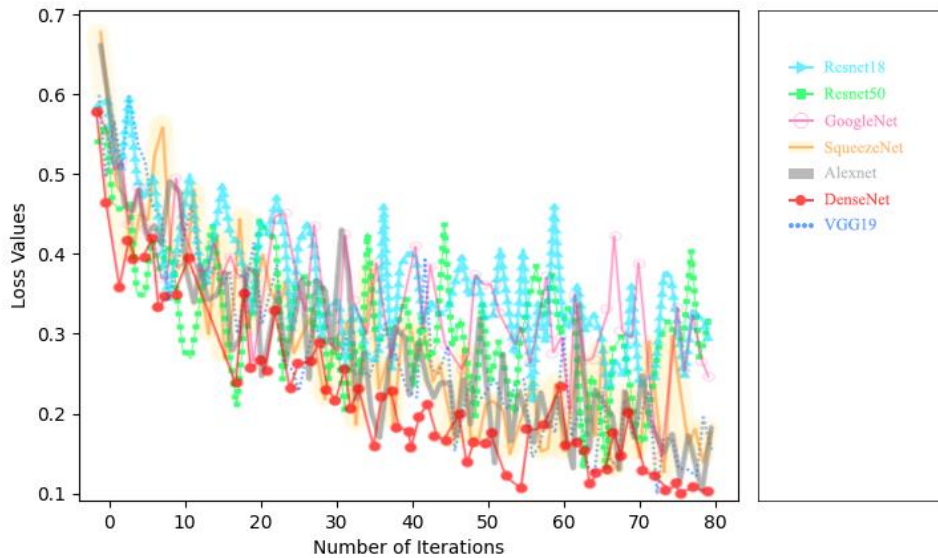
Burada, TP 1 sınıfına ait olup 1 olarak tahmin edilen görüntü sayısını, TN 0 sınıfına ait olup 0 olarak tahmin edilen görüntü sayısını, FP 0 sınıfına ait olup 1 olarak tahmin edilen görüntü sayısını, FN ise 1 sınıfına ait olup 0 olarak tahmin edilen görüntü sayısını temsil eder. 7-9 arasındaki denklemlerde doğru tahmin edilen görüntü sayısının tüm görüntü sayısına oranı hesaplanmaktadır. Şekil 3 ile modellerin kayıp grafikleri verilmiştir. Tablo 2’de farklı modellerin doğruluk açısından elde edilen en iyi sonuçlarının kıyaslamaları, Doğruluk, Hassasiyet, Kesinlik, F1 Skoru ve Çalışma Süresi (Sn.) metrikleri ile yapılmıştır. Tablo 2 incelendiğinde, eşit şartlarda, SqueezeNet dışında hemen hemen tüm CNN modellerinin oldukça başarılı performanslar gösterdiği görülmüştür. Resnet18 ile %95.50 doğruluk oranı ve %95.60 F1 skoru, Resnet 50 ile %94.75 doğruluk oranı ve %94.48 F1 skoru, VGG19 ile 92.50 doğruluk oranı ve %91.97 F1 skoru, AlexNet ile %94.75 doğruluk oranı ve %94.57 F1 skoru, DenseNet ile %98.75 doğruluk oranı ve %98.75 F1 skoru, SqueezeNet ile %93.75 doğruluk oranı ve %93.36 F1 skoru, GoogleNet ile ise %93.50 doğruluk oranı ve %93.71 F1 skoru elde edilmiştir. Çalışma süresi bakımından en yavaş model VGG19 olurken, en hızlı model ise SqueezeNet olmuştur. Çalışma süresi önemsenmediğinde her açıdan göreceli olarak en yüksek performansa sahip olan modelin DenseNet olduğu açıkça görülmektedir. Ayrıca SqueezeNet modelinin başarı açısından performansı açık ara artarken, çalışma süresi de düşük olduğu için tercih edilebilir bir modeldir. BFKL katmanı olmadan her ne kadar modeller daha hızlı çalışmış olsa da, model eğitildikten sonra kullanılması daha hızlıdır. Ayrıca iterasyon sayısı vb. parametreler değiştirilerek BFKL katmanlarından daha fazla verim alınabilir. Bu çalışmada minimal bir yaklaşımla modellerin başarısı gözlemlenmiştir. Çalışma süresi kullanılan donanıma bağlı olarak değişebilir, diğer sonuçlar ise ağı en iyi durumu göz önüne alındığında ortalama olarak yaklaşık \pm %1.22 sapma gösterebilir (10 defa çalıştırılma sonucu elde edilmiştir).

Tablo 2. Farklı modellerin performans açısından kıyaslaması.

Öznetelik Çıkarıcı	Doğruluk	Hassasiyet	Kesinlik	F1 Skoru	Eğitim ve Test Aşamalarındaki Toplam Çalışma Süresi (Sn.)
Resnet18	94.50	90.00	98.90	94.24	196.215
Resnet18 (BFKL ile)	95.50	98.00	93.33	95.60	337.160
Resnet50	94.50	90.00	98.00	94.24	313.239
Resnet50 (BFKL ile)	94.75	90.00	99.44	94.48	497.189
VGG19	91.00	84.00	97.67	90.32	526.983
VGG19 (BFKL ile)	92.50	86.00	98.85	91.97	736.976
AlexNet	86.00	72.00	100.00	83.72	173.426
AlexNet (BFKL ile)	94.75	91.50	97.86	94.57	330.110
DenseNet	98.00	97.5	98.48	97.98	283.756
DenseNet (BFKL ile)	98.75	99.50	98.02	98.75	586.547
SqueezeNet	79.50	60.00	99.17	74.76	151.027
SqueezeNet (BFKL ile)	93.75	88.00	99.43	93.36	319.526
GoogleNet	96.50	99.00	94.28	96.58	227.067
GoogleNet (BFKL ile)	93.50	97.00	90.65	93.71	383.121

DenseNet modeli, %98,00 doğruluk ve %97,5 hassasiyet elde ederek genel olarak en iyi performansı göstermiştir. BFKL ile birleştirildiğinde modelin performansı daha da iyileşerek %98,75 doğruluk ve %99,50 hassasiyet elde edilmiştir. Bu durum BFKL katmanının iyileştirilmiş performans için derin öğrenme modellerinde etkili bir yöntem olabileceğini düşündürmektedir. Resnet18 ve Resnet50 modelleri de BFKL ile sırasıyla %95,50 ve %94,75 doğruluk oranı elde ederek iyi performans göstermiştir. Resnet50 modeli %99,44 gibi yüksek bir kesinlik elde etmiştir. VGG19 modeli, BFKL ile uygulandığında %92,50'lik bir doğruluk elde etmiştir; bu durum, BFKL olmadan %91,00 olan doğruluğa göre bir gelişmedir. AlexNet modeli, bu çalışmada değerlendirilen diğer modellere kıyasla nispeten düşük olan %86,00'lik bir doğruluk elde etmiştir. Ancak BFKL ile birlikte, modelin doğruluğu %94,75'e yükselmiştir ve bu da önemli bir gelişmedir. Bu, BFKL katmanının başlangıçta daha az doğruluk oranına sahip olan bazı modellerin performansını iyileştirmek için etkili bir yöntem olabileceğini düşündürmektedir. SqueezeNet modeli ile, BFKL katmanı olmadan %79,50 ile en düşük doğruluk ve %60,00 ile en düşük hassasiyet elde edilmiştir. Yine bu model de BFKL katmanı ile birleştirildiğinde, modelin doğruluğu ve hassasiyeti önemli ölçüde iyileşerek %93,75 doğruluk ve %88,00 hassasiyet elde edilmiştir. Hassasiyetin artması, modelin pozitif sınıfa ait olan örnekleri (gerçek yüzler) daha iyi sınıflandırabileceğini göstermektedir. GoogleNet modeli, BFKL olmadan %96,50 doğruluk ve %94,28 kesinlik elde etmiştir. BFKL ile uygulandığında, modelin doğruluğu biraz düşerek %93,50 olurken, kesinliği %90,65'e düşmüştür. Bu, BFKL katmanının her zaman iyilştirilmiş bir performansa yol açmayabileceğini ve başarının belirli modele veya veri kümesine bağlı olabileceğini düşündürmektedir. F1 skorları incelendiğinde doğruluk oranlarına paralel olduğu, başarı modeller için de yüksek olduğu görülmektedir.

Genel olarak, bu çalışmanın sonuçları, DenseNet, AlexNet ve Resnet (18 ve 50) modellerinin göreceli olarak sahte yüz görüntülerini sınıflandırmak için etkili olduğunu ve BFKL katmanının derin öğrenme modellerinin performansını iyileştirmek için etkili bir yöntem olabileceğini göstermektedir. Modellerin büyük çoğunluğunda başarı oranı BFKL ile artmıştır. Bununla birlikte, bazı modeller için BFKL ile sağlanan verim, belirli modele veya veri kümesine bağlı olarak değişebilir. Nitekim bu durum, GoogleNet için gözlemlenmiştir.



Şekil 3. Model kayıp grafikleri (BFKL ile).

Şekil 3'te verilen kayıp grafikleri incelendiğinde, modellerin genel olarak dalgalanma ile birlikte iterasyonlar boyunca kayıplarının da azaldığı görülmektedir. Bu dalgalanma, yüz görüntüsü verilerinin gürültülü olmasından dolayı (farklı arka planlar, farklı yüzler vb.) ağız öğrenirken karşılaştığı bir zorluk olarak değerlendirilebilir. Yine kayıp grafikleri incelendiğinde, DenseNet modelinin özellikle son 10 iterasyonda kayıp değerinin ve kayıptaki dalgalanmalarının azaldığı, daha stabil duruma geldiği gözlemlenmiştir. DenseNet modelinin kaybının son iterasyonda, mevcut diğer kayıplardan daha düşük olması bu modelin hedefe daha hızlı yakınsadığını göstermektedir. Bir tarafta Resnet50, GoogleNet ve Resnet18 birbirine yakın bir performans izlerken, diğer taraftan AlexNet, SqueezeNet ve VGG19 birbirine yakın bir performans izlemiştir. DenseNet modelinin daha hızlı yakınsadığı ve daha başarılı olduğu söylenebilir, çünkü kaybın minimum olması beklenir.

5. SONUÇ

Bu çalışma kapsamında elde edilen bulgular göz önüne alındığında, sentetik verilerin tespiti için evrişimli sinir ağlarının kullanımının makul bir tercih olduğu gözlemlenmiştir. İnsan gözü ile tespit edilemeyen sentetik görüntülerin evrişimli sinir ağı modelleri aracılığıyla çok yüksek oranda başarı sağlayarak tespit edilebileceği gösterilmiştir. Laplace filtresi ve benzemezlik, yeni özelleştirilmiş katmanda kullanıldığında yüksek frekanslı öğelerin tespit edilmesini ve ağız benzer özelliklerini değerlendirmesini sağlamış, böylece sahte görüntülerin ayırt

edilmesini kolaylaştırmıştır. Bunun yanında farklı evrişimli sinir ağlarının öznelik çıkarıcı olarak kullanımına ilişkin bir karşılaştırma gerçekleştirilmiş olup, mimari açıdan karmaşık bir yapıya sahip olan DenseNet modelinin göreceli olarak en başarılı performansı sergilediği gözlemlenmiştir. DenseNet için %98 olan doğruluk oranı, %98.75'e yükselerek modelin başarısını artırmıştır. Ayrıca başka dikkat çekici bir sonuç olan SqueezeNet modelinin başarısının %79.50 doğruluk oranından %93.75 doğruluk oranına çıkması, oldukça önemli bir iyileşmedir. Bu durum sonuç olarak BFKL katmanının farklı modellerin yapısına entegre edilerek ilgili modelin başarısını artırabileceğini vurgulamaktadır. Bununla birlikte, GoogleNet için aynı şey geçerli değildir. GoogleNet modelinin doğruluk oranı %3 kadar düşmüştür. Model ve veri seti seçilirken, bu sebepten dolayı dikkatli olunması gerekmektedir. Fakat modellerin çoğunluğu için konuşursak, BFKL katmanı iyileştirici bir performans göstermiştir. Gelecek çalışmalarda veri setindeki görüntü sayısı artırılabilir gibi, sentetik görüntüler üzerinde farklı türde çalışmalar yapılabileceği düşünülmektedir. Özellikle farklı yüksek geçiren filtrelerin uygulanması tavsiye edilmektedir.

Yazar Katkıları

Yazarlar çalışmaya eşit oranlı katkı sunmuşlardır.

Çıkar Çatışması

Makale yazarları aralarında herhangi bir çıkar çatışması olmadığını beyan ederler.

KAYNAKÇA

- [1] I. Goodfellow vd. "Generative adversarial networks", Commun. ACM, vol. 63, no. 11, pp. 139-144, 2020.
- [2] S. Motamed, P. Rogalla, ve F. Khalvati "Data augmentation using Generative Adversarial Networks (GANs) for GAN-based detection of Pneumonia and COVID-19 in chest X-ray images", Informatics Med. Unlocked, vol. 27, p. 100779, 2021.
- [3] N. Sharma, R. Sharma, ve N. Jindal "Comparative analysis of CycleGAN and AttentionGAN on face aging application", Sādhana, vol. 47, no. 1, p. 33, 2022.
- [4] R. Huang, L. Ma, J. He, ve X. Chu "T-GAN: A deep learning framework for prediction of temporal complex networks with adaptive graph convolution and attention mechanism", Displays, vol. 68, p. 102023, 2021.
- [5] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, ve A.A. Bharath "Generative Adversarial Networks: An Overview", IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53-65, 2018.
- [6] S. Stoll, N. C. Camgoz, S. Hadfield, ve R. Bowden "Text2Sign: Towards Sign Language Production Using Neural Machine Translation and Generative Adversarial Networks", Int. J. Comput. Vis., vol. 128, no. 4, pp. 891-908, 2020.
- [7] K. Nazeri, E. Ng, ve M. Ebrahimi "Image colorization using generative adversarial networks", içinde Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), LNCS, vol. 10945 pp. 85-94, 2018.
- [8] C. Fabbri, M.J. Islam, ve J. Sattar "Enhancing Underwater Imagery Using Generative Adversarial Networks", içinde Proceedings - IEEE International Conference on Robotics and Automation, pp. 7159-7165, 2018.
- [9] G. Antipov, M. Baccouche, ve J.L. Dugelay "Face aging with conditional generative adversarial networks", içinde Proceedings - International Conference on Image Processing, ICIP, pp. 2089-2093, 2018.
- [10] F. Marra, D. Gragnaniello, D. Cozzolino, ve L. Verdoliva "Detection of GAN-Generated Fake Images over Social Networks", IEEE 1st Conference on Multimedia Information Processing and Retrieval, MIPR, pp. 384-389.
- [11] D. Cozzolino, G. Poggi, ve L. Verdoliva "Recasting residual-based local descriptors as convolutional neural networks: An application to image forgery detection", Proceedings of the 2017 ACM Workshop on Information Hiding and Multimedia Security, pp. 159-164, 2017.
- [12] B. Bayar ve M.C. Stamm "A deep learning approach to universal image manipulation detection using a new convolutional layer", Proceedings of the 2016 ACM Information Hiding and Multimedia Security Workshop, pp. 5-10, 2016.
- [13] L. Nataraj vd. "Detecting GAN generated Fake Images using Co-occurrence Matrices", Electron. Imaging, vol. 31, no. 5, pp. 532-1-532-7, 2019.
- [14] S. McCloskey ve M. Albright "Detecting GAN-Generated Imagery Using Saturation Cues", International Conference on Image Processing, ICIP, pp. 4584-4588.
- [15] Y. LeCun ve Y. Bengio "Convolutional networks for images, speech, and time series", Handb. brain theory neural networks, vol. 3361, pp. 255-258, 1995, [Çevrimiçi]. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.9297&rep=rep1&type=pdf>.
- [16] T. Rahman vd. "Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection Using Chest X-ray", Appl. Sci., vol. 10, no. 9, p. 3233, 2020.
- [17] S. Raghu, N. Sriraam, Y. Temel, S.V. Rao, ve P.L. Kubben "EEG based multi-class seizure type classification using convolutional neural network and transfer learning", Neural Netw., vol. 124, pp. 202-212, 2020.

- [18] A.M. Ismael ve A. Şengür “Deep learning approaches for COVID-19 detection based on chest X-ray images”, *Expert Syst. Appl.*, vol. 164, 2021.
- [19] K. He, X. Zhang, S. Ren, ve J. Sun “Deep Residual Learning for Image Recognition”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [20] E. Cengil ve A. Cinar “Multiple Classification of Flower Images Using Transfer Learning”, *International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1–6, 2019.
- [21] K. Simonyan ve A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, 2015.
- [22] A. Krizhevsky, I. Sutskever, ve G. E. Hinton “2012 AlexNet”, *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.
- [23] G. Huang, Z. Liu, L. Van Der Maaten, ve K. Q. Weinberger “Densely connected convolutional networks”, *IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.
- [24] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, ve K. Keutzer “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size”, *arXiv Prepr. arXiv1602.07360*, 2016.
- [25] C. Szegedy vd. “Going deeper with convolutions”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ss. 1–9, 2015.
- [26] “This Person Does Not Exist”. <https://thispersondoesnotexist.com/>.
- [27] “Flickr”. <https://www.flickr.com/>.
- [28] Usha Ruby Dr. A “Binary cross entropy with deep learning technique for Image classification”, *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 5393–5397, 2020.
- [29] R.M. Haralick, K. Shanmugam, ve I. Dinstein “Textural Features for Image Classification”, *IEEE Trans. Syst. Man. Cybern.*, c. SMC-3, vol. 6, pp. 610–621, 1973.
- [30] R.C. Gonzalez, R.E. Woods, ve B.R. Masters “Digital Image Processing, Third Edition”, *J. Biomed. Opt.*, vol. 14, no. 2, p. 029901, 2009.
- [31] L.A. Gatys, A.S. Ecker, ve M. Bethge “Image Style Transfer Using Convolutional Neural Networks”, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2414–2423, 2016.
- [32] D.L. Ruderman “Origins of scaling in natural images”, *Vision Res.*, vol. 37, pp. 3385–3398, 1996.
- [33] M. Goebel, L. Nataraj, T. Nanjundaswamy, T.M. Mohammed, S. Chandrasekaran, ve B.S. Manjunath “Detection, Attribution and Localization of GAN Generated Images”, *Electron. Imaging*, vol. 33, no. 4, pp. 276-1-276–11, 2021.
- [34] F. Zhuang vd. “A Comprehensive Survey on Transfer Learning”, *IEEE*, vol. 109, no. 1. pp. 43–76, 2021.
- [35] C. Garbin, X. Zhu, ve O. Marques “Dropout vs. batch normalization: an empirical study of their impact to deep learning”, *Multimed. Tools Appl.*, vol. 79, no. 19–20, pp. 12777–12815, 2020.
- [36] Z. DeVries vd. “Using a national surgical database to predict complications following posterior lumbar surgery and comparing the area under the curve and F1-score for the assessment of prognostic capability”, *Spine J.*, vol. 21, no. 7, pp. 1135–1142, 2021.