



DERİN ÖĞRENME YÖNTEMLERİ İLE

GÖRÜNTÜ SAHTECİLİĞİ TESPİTİ

NIHAT EREN ÖZMEN

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. ERCAN BULUŞ

2022

T.C.
TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



DERİN ÖĞRENME YÖNTEMLERİ İLE GÖRÜNTÜ SAHTECİLİĞİ TESPİTİ

NIHAT EREN ÖZMEN
ORCID: 0000-0002-0053-3865

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
YÜKSEK LİSANS TEZİ
Danışman: Doç. Dr. ERCAN BULUŞ

HAZİRAN - 2022

Her hakkı saklıdır.

ÖZET

DERİN ÖĞRENME YÖNTEMLERİ İLE GÖRÜNTÜ SAHTECİLİĞİ TESPİTİ

Nihat Eren ÖZMEN

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. Ercan BULUŞ

Günümüzde her geçen gün kullanımı artan sosyal medya ve internet sayesinde neredeyse çekilen her fotoğraf internet ortamına yüklenmektedir. Bu sebeple görüntü elde etmek çok daha kolay hale gelmiştir. Görüntülerin kolaylıkla elde edilmesi, gelişen teknolojiler sayesinde görüntü üzerinde değişiklik yapılmasını teşvik edici hale gelmiştir. Görüntüler üzerinde kişinin kendi isteği ile yapmış olduğu değişikliklerin yanı sıra kötü niyetli kişiler tarafından yapılan değişiklikler ise bir takım sorunlara yol açmakta olup istenmeyen durumlar yaratabilmektedir. Yapılan değişiklikler ile kişisel verilerin ihlalden herhangi bir resmi belge üzerinde sahteciliğe kadar geniş bir alanı etkilemektedir. Sahtecilik yöntemleri farklı şekilde yapılmaktadır. Kopyala-taşı yöntemi, mevcut görüntüdeki bir alanın aynı görüntü üzerinde farklı bir alana kopyalanması ile yapılmaktadır. Görüntü birleştirme, farklı iki görüntünün istenilen alanları birleştirilmesi ile oluşturulmaktadır. Görüntü rötuşlama, görüntü üzerinde istenmeyen bölgelerin etkisi azaltarak ya da istenilen alan üzerinde iyileştirme yapılarak görüntü daha ilgi çekici hale getirilmektedir. Çalışmada bu düşünceden yola çıkılarak görüntüler üzerinde değişiklik olup olmadığının tespit edilmesi eğer değişiklik var ise o alanın işaretlenmesi için derin öğrenme yöntemi kullanılmıştır. Derin öğrenme yöntemleri sayesinde yüksek boyutlu verilerde daha hızlı ve daha yüksek başarı elde edilmiştir. Çalışmada CASIA ve CoMoFoD veri setleri üzerinde uygulama gerçekleştirilmiştir. Veri setleri üzerinde yapılan testler sonucunda ortalama %76,86 oranından doğruluk elde edilmiştir.

Anahtar Kelimeler: Derin öğrenme, Görüntü sahteciliği, Fotomontaj tespiti

ABSTRACT

PHOTOMONTAGE DETECTION WITH DEEP LEARNING METHODS

Nihat Eren ÖZMEN

Department of Computer Engineering

MSc. Thesis

Supervisor: Assoc. Prof. Dr. Ercan BULUŞ

Today, thanks to the increasing use of social media and the internet almost every photo taken is uploaded to the internet. For this reason, it has become much easier to obtain images. Easily obtaining images has become an encourage to make changes on the image thanks to the developing technologies. In addition to the changes made on the images by the person's own will, the changes made by malicious people cause some problems and can create unwanted situations. With the changes made, it affects a wide area from the violation of personal data to the forgery of any official document. Forgery methods are carried out in different ways. Copy-move method is done by copying an area in the current image to a different area on the same image. Image fusion is created by combining the desired areas of two different images. Image retouching makes the image more interesting by reducing the effect of unwanted areas on the image or by improving the desired area. Based on this idea, in the study, deep learning method was used to determine whether there is a change on the images, and if there is, to mark the area. Thanks to deep learning methods, faster and higher success has been achieved in high-dimensional data. In the study, an application was carried out on the CASIA and CoMoFoD data sets. As a result of the tests performed on the data sets, an average accuracy of 76.86% was obtained.

Keywords: Deep learning, Image forgery, Photomontage detection

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER.....	iii
ÇİZELGELER DİZİNİ.....	v
ŞEKİLLER DİZİNİ.....	vi
KISALTMALAR DİZİNİ.....	vii
TEŞEKKÜR.....	ix
1. GİRİŞ	1
1.1 Literatür Özeti	2
1.1.1 Derin Öğrenme Tanımı	2
1.1.2 Derin Öğrenme Tarihiçesi	3
1.1.3 Derin Öğrenme Yaklaşımları	8
1.1.3.1 Denetimli Öğrenme	8
1.1.3.2 Yarı Denetimli Öğrenme	8
1.1.3.3 Denetimsiz Öğrenme	8
1.1.3.4 Derin Pekiştirmeli Öğrenme	9
1.1.4 Derin Öğrenme Mimarileri.....	9
1.1.4.1 Evrişimli Sinir Ağları	9
1.1.4.2 Tekrarlayan Sinir Ağları	11
1.1.4.3 Uzun Kısa Vadeli Hafıza Ağları.....	12
1.1.4.4 Sınırlı Boltzmann Makineleri	13
1.1.4.5 Derin İnanç Ağları	14
1.1.4.6 Derin Oto-Kodlayıcılar	15
1.1.5 Derin Öğrenme Uygulamaları	16
1.1.5.1 Görüntü/Video İşleme.....	16
1.1.5.2 Doğal Dil İşleme	18
1.1.5.3 Diğer Uygulamalar	19
1.1.6 Derin Öğrenme Alanındaki Çalışma Ekipleri ve Kullanılan Kütüphaneler.....	21
1.1.6.1 Derin Öğrenme Alanındaki Çalışma Ekipleri.....	21
1.1.6.2 Derin Öğrenme Kütüphaneleri.....	21
1.1.7 Fotomontaj Tespiti Hakkında	23
1.1.7.1 Fotomontaj Tespiti Nedir?	23
1.1.7.2 Görüntü Sahteciliği Yöntemleri.....	23

1.1.7.2.1 Kopyala-Taşı Sahteciliği	24
1.1.7.2.2 Görüntü Birleştirme Sahteciliği	24
1.1.7.2.3 Görüntü Rötuşlama Sahteciliği	25
1.1.7.3 Görüntü Sahteciliği Algılama Yaklaşımları	25
1.1.7.3.1 Kopyala-Taşı Sahtecilik Algılama Teknikleri.....	25
1.1.7.3.2 Görüntü Birleştirme Algılama Teknikleri.....	26
1.1.7.3.3 Görüntü Rötuşlama Algılama Teknikleri	27
1.1.7.4 Derin Öğrenme Yöntemleri ile Görüntü Sahteciliği.....	28
1.2 Çalışmanın Amacı ve Kapsamı.....	31
2. MATERYAL VE YÖNTEM.....	32
2.1 Tensoflow	32
2.2 Keras	34
2.3 Numpy.....	37
2.4 Matplotlib.....	38
2.5 Anaconda	40
2.6 Akış Diagramı	41
3. UYGULANAN MODEL VE BİLGİLERİ.....	42
3.1 Model Bilgileri.....	42
3.1.1 Manipülasyon Algılama Dalı	43
3.1.2 Benzerlik Algılama Dalı.....	43
3.1.3 BusterNet Fusion.....	44
3.2 Model Eğitim Bilgileri.....	44
3.1.1 CASIA Veri Seti.....	44
3.1.2 CoMoFoD Veri Seti	45
3.3 Model Özeti	45
3.3 Uygulama Özet Kod Gösterimleri	46
3.5 Uygulama Çıktıları.....	47
4. SONUÇ VE ÖNERİLER.....	49
KAYNAKLAR.....	51
ÖZGEÇMİŞ	56

ÇİZELGELER DİZİNİ

Çizelge 1.1. Derin öğrenme alanında çalışan ekipler	21
Çizelge 1.2. Derin öğrenme kütüphaneleri hakkında bilgiler.....	22
Çizelge 1.3. Görüntü sahteciliği tespiti için derin öğrenme tabanlı yöntemlerin karşılaştırması	29
Çizelge 2.1. Tensorflow modelleri hakkında bilgiler	32
Çizelge 2.2. Tensorflow’da yer alan bazı veri setleri ve çalışma alanları	34
Çizelge 2.3. Keras api örnekleri ve özelliği.....	36
Çizelge 2.4. Çizim tipleri, basit kod yapısı ve örnek gösterimi	38
Çizelge 4.1. BusterNet doğruluk oranları	49



ŞEKİLLER DİZİNİ

Şekil 1.1. Ivankhnenko tarafından eğitilen ilk derin ağ mimarisi	4
Şekil 1.2. Hopfield ağları görseli.....	5
Şekil 1.3. Önde gelen teknoloji şirketleri tarafından yapılan yapay zekâ başlangıç firmalarının satın alımlarının toplam sayısı.....	7
Şekil 1.4. Ülke bazındaki yapay zekâ başlangıç firma sayıları ve bilgileri.....	7
Şekil 1.5. Evrişimsel ağ mimarisi.....	11
Şekil 1.6. Tekrarlayan sinir ağı mimarisi	12
Şekil 1.7. Sınırlı boltzmann makineleri mimarisi.....	14
Şekil 1.8. Oto-Kodlayıcılar mimarisi	15
Şekil 1.9. Derin Oto-Kodlayıcılar mimarisi	16
Şekil 1.10. Görüntü sahteciliği yöntemleri.....	23
Şekil 1.11. Görüntü üzerinde kopyala-taşı ve rötuşlama etkisi	24
Şekil 1.12. İki farklı görüntü kullanarak görüntü birleştirme.....	25
Şekil 1.13. Kopyala-Taşı sahteciliği tespitinin geliştirilmiş yapısı.....	26
Şekil 1.14. Görüntü birleştirme tespitinin geliştirilmiş yapısı	27
Şekil 2.1. Sıralı model gösterimi	35
Şekil 2.2. Takımların birincil kullandığı makine öğrenmesi yapısı	37
Şekil 2.3. Akış diagramı	41
Şekil 3.1. BusterNet mimarisi.....	42
Şekil 3.2. Maske kod çözücü ağ gösterimi, BN-Inception modülü parametreleri.....	43
Şekil 3.3. Model özeti.....	45
Şekil 3.4. CASIA veri seti çıktıları.....	47
Şekil 3.5. CoMoFoD veri seti çıktıları	48

KISALTMALAR DİZİNİ

AE	Oto-kodlayıcılar – Autoencoder
API	Uygulama Programlama Arayüzü - Application Programming Interface
ASIC	Özel Tasarımlı Uygulamaya Özel Tümlşik Devre - Application Specific Integrated Circuit
CNN	Evrişimsel Sinir Ağları – Convolutional Neural Networks
COCO	Common Objects In Context
CPU	Merkezi İşlem Birimi - Central Process Unit
DAE	Derin oto-kodlayıcılar - Deep Autoencoder
DBN	Derin İnanç Ağları – Deep Belief Network
DCT	Ayrık Kosinüs Dönüşümü - Discrete Cosine Transform
DEFT	Derin Araştırma ve Metin Filtreleme - Deep Exploration and Filtering of Text
DNA	Deoksiribonükleik Asit - Deoxyribonucleic Acid
DNN	Derin Sinir Ağları – Deep Neural Networks
DRL	Derin Pekişmeli Öğrenme – Deep Reinforcement Learning
DWT	Ayrık Dalga Dönüşümü - Discrete Wavelet Transform
GAN	Çekişmeli Üretici Ağlar – Generative Adversarial Networks
GPS	Küresel Konumlama Sistemi - Global Positioning System
GPU	Merkezi İşlem Birimi - Graphics Processing Unit
GRU	Geçitli Tekrarlayan Birimler – Gated Recurrent Unit
ICDAR	The International Conference on Document Analysis and Recognition
ICPR	Indian Council of Philosophical Research
IDE	Entegre Geliştirme Ortamı - Integrated Development Environment
ILSVRC	The ImageNet Large Scale Visual Recognition Challenge
LSTM	Uzun Kısa Vadeli Hafıza – Long Short-Term Memory
Mani-Det	Manipülasyon Algılama - Manipulation Detection
MNIST	Modified National Institute of Standards and Technology
MRI	Manyetik Rezonans Görüntüleme - Magnetic Resonance Imaging
NMT	Nöral Makine Çevirisi – Neural Machine Translation
ONEIROS	Açık Uçlu Nöro-Elektronik Akıllı Robot İşletim Sistemi - Open-ended Neuro-Electronic Intelligent Robot Operating System
RBM	Sınırlandırılmış Boltzmann Makineleri – Restricted Boltzmann Machines

RL	Pekişmeli Öğrenme - Reinforcement Learning
RNA	Ribonükleik Asit - Ribonucleic Acid
RNN	Tekrarlayan Sinir Ağları – Recurrent Neural Networks
Simi-Det	Benzerlik Algılama - Similarity Detection
SSIM	Yapısal Benzerlik Endeks Metriği – The Structural Similarity Index Measure
TPU	Tensör İşleme Birimi - Tensor Processing Unit



TEŞEKKÜR

Yüksek lisans eğitiminde ve tez çalışmamda göstermiş olduğu ilgi, destek, yardımları ve akıl hocalığından dolayı kıymetli danışmanım Doç. Dr. Ercan BULUŞ'a en içten duygularıyla teşekkürlerimi sunarım. Yüksek lisans eğitimim de bana yardımları dokunan Tekirdağ Namık Kemal Üniversitesi Çorlu Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümündeki tüm hocalarıma ve tüm çalışma arkadaşlarıma teşekkür ederim. Eğitim hayatım süresince benden maddi ve manevi olarak yardımlarını esirgemeyen anneme, babama, babaanneme, halama ve kuzenime kalbi teşekkürlerimi sunarım.

Nihat Eren ÖZMEN

Araştırma Görevlisi

1. GİRİŞ

Son yıllarda hızla gelişen teknoloji ile birlikte verilerin sağlıklı bir şekilde elde edilmesi ve elde edilen verilerin korunması büyük önem kazanmaktadır. Verilerin elde edilip korunmasının yanı sıra elde edilen verilerin özgün olmasının ayrı bir önem teşkil etmektedir. Bu sebeple steganografi alanında çeşitli algoritmalar ve teknikler geliştirilmiştir.

Steganografi, gizlenmek istenen verinin bir başka nesne içerisine gömülmesi ve saklanmasına dayanmaktadır. Steganografi kelimesinin kökeni Yunancadan gelmektedir. “στεγανος” ve “γραφειν” kelimelerinden türetilmiş olup kelime anlamı tam olarak “kaplanmış yazı” demektir. Steganaliz, bir nesne içerisinde yer alan herhangi bir bilgi veya değişiklik olup olmadığını bulmaya çalışan, eğer bir değişiklik söz konusu ise bu değişikliğin nerede olduğunu tespit etmek için steganografi tekniklerini kullanan yöntemler bütünüdür. Steganaliz yapan kişiye steganalist denilmektedir. Bu kişi steganaliz yapabilmesi için geliştirilmiş olan metotlar hakkında yeterince bilgi sahibi olmalıdır (Şahin, Buluş, Sakallı, Buluş, 2007).

Bir görüntü üzerinde yapılan değişiklikler de steganografi konusunun başlıca ilgi alanlarından biridir. Görüntü üzerinde çeşitli teknikleri ile değişiklikler yapılmaktadır. Bu tekniklerden biri de fotomontaj yöntemidir. Fotomontaj, görüntüyü kesme, yapıştırma, düzenleme veya üst üste bindirme yöntemleri kullanarak yeni bir görüntünün elde edilmesidir. Günümüzde popülerliği her geçen gün artan sosyal medya mecraları ve hızla gelişen teknoloji sayesinde dijital görüntü sayısı sürekli artmaktadır. Görüntü sayısındaki artış ve gelişen teknoloji ile birlikte elde edilen görüntülere fotomontaj yaparak insanları manipüle etmek daha kolay hale gelmiştir. Bu sebeple görüntülere duyulan şüphe artmış olup, kullanılan görüntülerin özgünlüğü büyük önem kazanmıştır. Görüntünün özgünlüğü, mahkeme kanıtları, haber çekimleri gibi çeşitli alanlarda dikkatle ele alınmaktadır. Bu sebeple, görüntünün orijinalliğin tespit edilmesi veya bozulmuş bir görüntü var ise görüntü üzerinde yapılmış olan değişikliklerin olduğu bölgelerin işaret edilmesi için güvenilir bir yöntem ihtiyacı duyulmaktadır (Jarusek, Volna, Kotyrba, 2019).

Derin öğrenme, çoklu işleme katmanlarına sahip verilerin temsili öğrenme yöntemi ile bir araya getirilmiş çoklu sayıdaki soyutlama yapısından oluşmaktadır. Yöntemler kullanılarak nesne tanıma, nesne algılama, ses tanıma gibi birçok alanda sıkça kullanılmaktadır. Derin öğrenme, öğretim yapılan makinenin öğrenimi yapılan modelde yer alan her katmanındaki temsili öğrenme parametrelerinin önceki katmanlardaki parametreler ile nasıl değişeceğini

hesaplayan geri beslemeli algoritmaları kullanmaktadır. Derin evrişimsel ağlar sayesinde büyük verilerde istenilen sonuçlar daha hızlı bir şekilde alınmakta olup ayrıca evrişimsel ağlar sayesinde görüntü ve video yer alan nesnelere algılanması işlenmesi konusunda büyük ilerleme kat edilmesini sağlamaktadır (LeCun, Bengio, Hinton, 2015).

Derin öğrenme yöntemleri sayesinde elde edilen ilerleme bu yöntemlerin farklı alanlarda kullanımının da yolunu açmıştır. Bu alanlardan biri de fotomontaj tespitidir. Derin evrişimsel ağlar ile görüntüdeki nesnelere algılanması ve tanınması ile görüntü üzerinde bozulma olup olmadığı eğer bozulma varsa görüntünün hangi bölgelerinde yer aldığı tespit edilebilmektedir.

1.1 Literatür Özeti

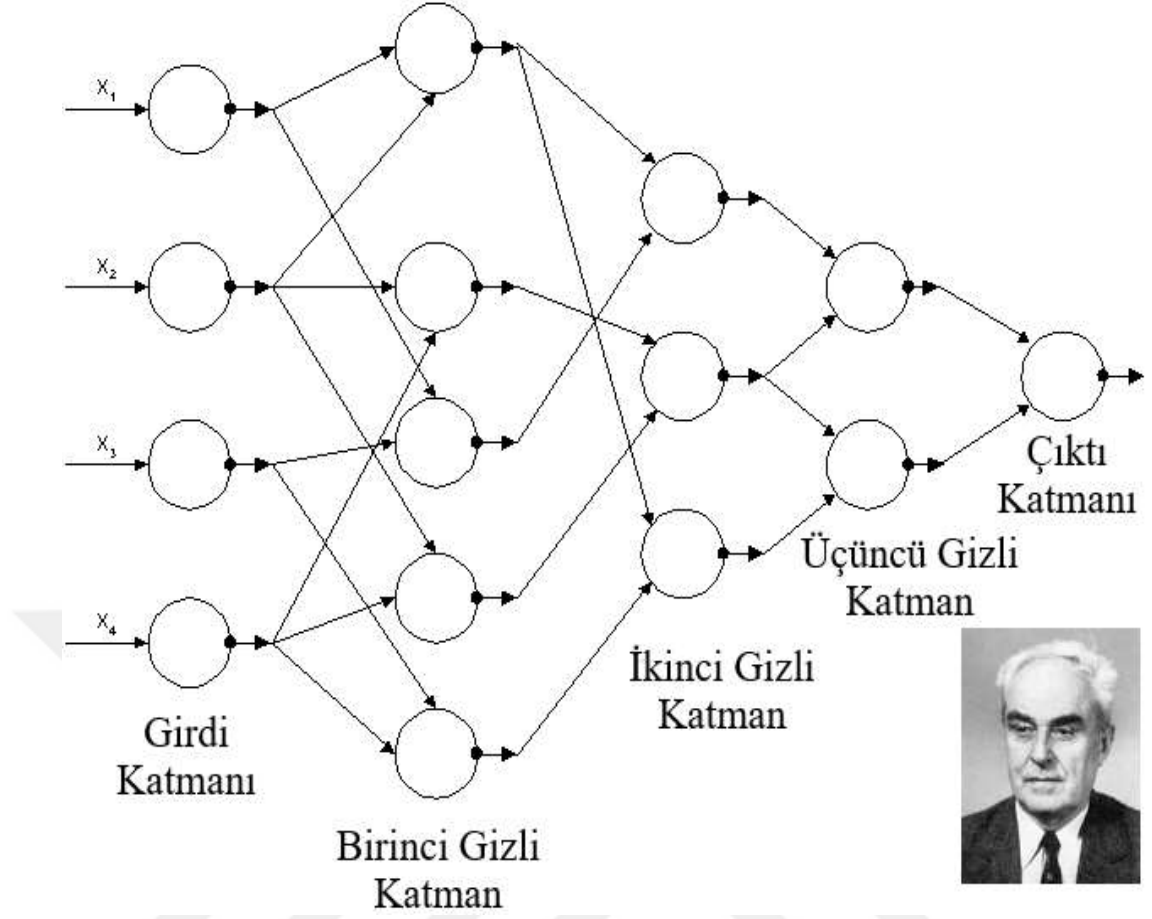
1.1.1 Derin öğrenme Tanımı

Makine öğrenmesi web üzerinden yapılan aramalarda, sosyal ağlarda ve web sitelerinde yer alan önerilerden tüketicilerin kullandığı akıllı telefonlara kadar geniş bir ürün ağına kullanılmaktadır. Makine öğrenmesindeki temel amaç görüntüler üzerindeki nesnelere tanımak, ses verilerindeki konuşmayı metne dönüştürmek ve kullanıcıların ilgi alanlarına uygun ürünler ile kullanıcıları eşleştirmektir. Bu tarz uygulamalar her geçen gün artan bir şekilde derin öğrenme adı verilen makine öğrenmesinin alt sınıfı olan bir teknik sayesinde gerçekleştirilmektedir. Derin öğrenme veriler üzerinden özellik çıkarımı yaparken temsil öğrenme yöntemini kullanmaktadır. Temsil öğrenme veriler üzerinde algılama ve sınıflandırma gibi işlemler için gerekli olan gösterimleri otomatik olarak makinenin yapmasını sağlayan bir dizi yöntemler tümüdür. Derin öğrenme yöntemleri, her temsili işlenmemiş veriden başlayarak daha yüksek düzeydeki temsile dönüştüren bu dönüştürme işlemi basit parçalar oluşturarak yapan çoklu temsil düzeylerine sahip öğrenme yöntemlerinin tümüdür. Örnek olarak ele alınan bir görüntü ilk olarak bir piksel değerleri biçiminde gelir. İlk temsil katmanında öğrenilen özellikler görüntüdeki kenarların varlığını kontrol eder. İkinci katmanda ise kenar konumlarındaki düzenleri tespit ederek desenleri oluşturur. Üçüncü katmanda ise belirlenen desenlerdeki parçalara karşılık gelen daha büyük kombinasyonları birleştirir ve sonraki katmanlarda ise görüntüdeki nesnelere bu kombinasyonların birleştirilmesi ile algılar. Bu katmanların her biri uygulayıcı tarafından değil öğrenme yöntemleri kullanılarak verilerden çıkartılarak gerçekleşir (LeCun, Bengio, Hinton, 2015).

Derin öğrenme kapsamında denetimli veya denetimsiz özellik çıkarma, çıkarılan özellikleri dönüştürme ve bu özellikler üzerinde desen analizi ve sınıflandırma yapılabilir. Bunun yanında karmaşık veriler arasındaki ilişkileri modellemek için katmanlar kullanır ve bu katmanlar bir önceki katmandaki çıktıyı girdi olarak kabul etmektedir. Bu özellik sayesinde derin mimari yapısına sahiptir. Derin öğrenme yardımı ile görüntü, ses ve metin gibi verilerde özellik çıkarma ve anlamlandırma işlemleri yapılmaktadır (Deng, Yu, 2014). Çıkarılan bu özelliklerden bazıları veriyi daha iyi temsil etmektedir. Derin öğrenme yöntemleri sayesinde çıkarılan bu özellikler hiyerarşik olarak ele alınmakta olup, veriyi daha iyi temsil eden özellikler seçilmektedir.

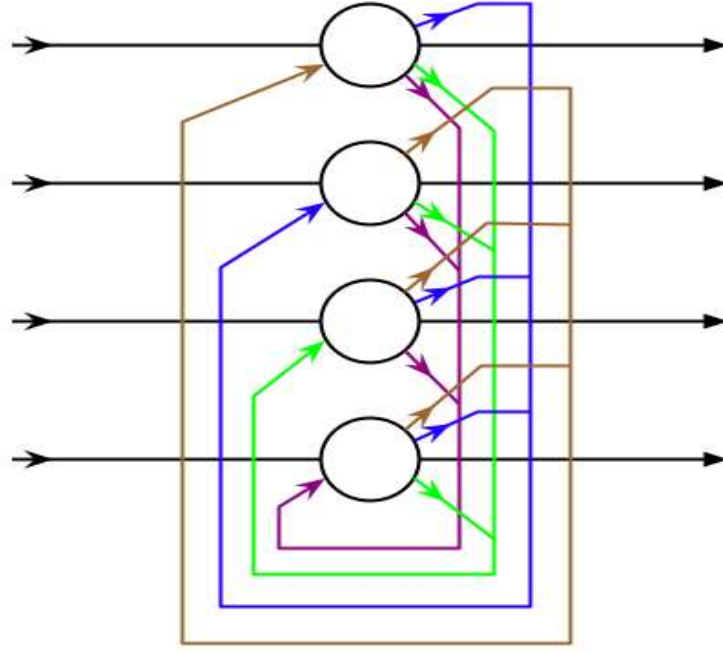
1.1.2 Derin Öğrenme Tarihçesi

Denetimli derin ileri beslemeli çok katmanlı algılayıcılar (perceptronlar) için ilk çalışan öğrenme algoritması 1965 yılında Ivakhnenko ve Lapa tarafından yayınlanmıştır. Regresyon ve ayrı bir doğrulama seti kullanarak dahili temsilleri öğrenmek için ağ katmanlarını kademeli olarak eğitmiş ve eğitilmiş katmanları sonrasında budayarak optimize etmiştir (Ivakhnenko, Lapa,1966). Şekil 1.1’de ilk derin ağ mimarisini gösterilmektedir.



Şekil 1.1. Ivankhnenko tarafından eğitilen ilk derin ağ mimarisi (Şeker, Diri, Balık, 2017).

1979 yılında Kunihiro Fukushima tarafından “Neokognitron” adı verilen yapay sinir ağı önerilmiştir. Bu yöntem el yazısı karakter ve başka desen tanıma, öneri sistemleri oluşturma da hatta dil işleme alanlarında kullanılmıştır. Neokognitron, hayvanlarda bulunan görsel korteks sisteminden ilham alarak geliştirilen ilk evrimsel sinir ağıdır. 1982 yılında John Hopfield tarafından tasarlanan Hopfield Ağları adreslenebilir içeriğe sahip bir bellek sistemine uygun olarak çalışan tekrarlayan sinir ağını oluşturmuştur. Şekil 1.2’de Hopfield ağları görseli yer almaktadır.



Şekil 1.2. Hopfield ağları görseli (Hopfield Ağı, 2022)

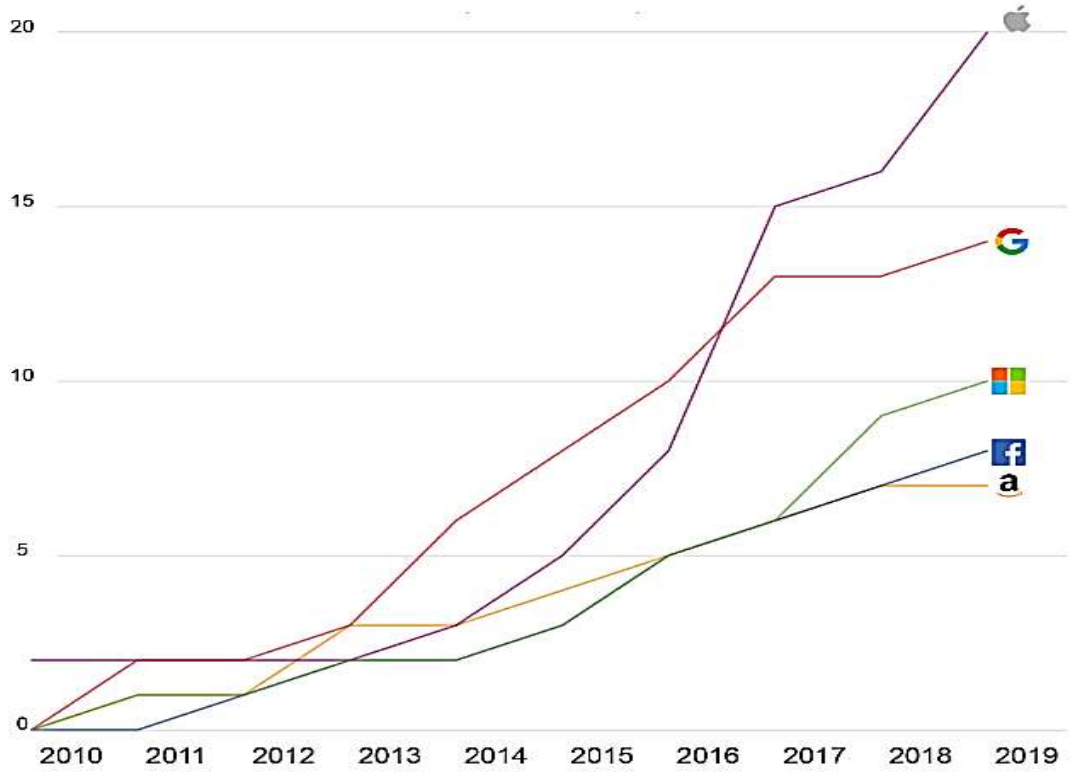
1989 yılında Yann LeCun ve arkadaşları tarafından geliştirilen geri yayılım algoritması ile posta kutusu yazıları üzerinde başarılı bir derin sinir ağı uygulaması oluşturulmuştur. Ancak geliştirilen bu uygulamanın eğitim aşaması 3 gün sürdüğünden pratikte pek uygun olmadığı görülmüştür (LeCun vd., 1989). 1997 yılında Jürgen Schmidhuber ve Sepp Hochreiter tarafından tekrarlayan sinir ağı olan Uzun Kısa Vadeli Hafıza(LSTM) önerilmiştir. Bu yöntem ile birlikte uzun süreli bağımlılık sebebi ile ortaya çıkan tekrarlayan sinir ağları sorunları giderilmiş olup hem verimlilik artmıştır hem de uygulanması daha pratik hale gelmiştir. LSTM ağları bilgileri daha uzun süre hatırlayabilmektedir (Hochreiter, Schmidhuber, 1997). 1998 yılında Yann LeCun ve arkadaşları tarafından “Belge Tanıma da Uygulanan Gradyan Tabanlı Öğrenme” adlı yayını ile geri yayılım algoritması ile birlikte gradyan tabanlı öğrenme algoritması bir arada kullanarak başarılı bir derin öğrenme yaklaşımı sunmuştur (LeCun, Bottou, Bengio, Haffner,1998). Sinir ağlarında ortaya çıkan bu gelişmelere rağmen hesaplama maliyetleri sebebi ile kullanılmamış alternatif olarak ele alınan probleme özgün çözüm sunan elle hazırlanan ve basit modeller olan destek vektör makineleri tercih edilmiştir (Cortes, Vapnik, 1995).

Sonraki yıllarda gelişen teknoloji ile bilgisayarların çalışma hızlarının artması ve grafik işlemci birimlerinin de hesaplamalara dahil edilmesi ile birlikte yapay sinir ağları destek vektör makineleri karşısında tercih edilmeye başlamıştır. Yapay Sinir Ağları kapsamında “Derin

öğrenme” ifadesi ilk defa 2000 yılında Aizenberg ve arkadaşları tarafından tanımlanmıştır. 2006 yılında ise Geoffrey Hinton tarafından çok katmanlı ileri beslemeli sinir ağı ile her iterasyon ile modelde yer alan her katmanın etkin bir şekilde nasıl eğitilebileceği ve bu eğitimler sonrasında geri yayılım algoritması ile oluşturulan modelde ufak değişiklikler yaparak iyileştirme yapılabileceği gösterilmiştir (Hinton, 2007).

Yapay Sinir Ağı algoritmalarında ortaya çıkan bu gelişmeler ile birlikte 2009 yılında Fei-Fei Li tarafından ImageNet ortaya çıkarılmıştır. Bu veri setinde 14 milyondan fazla etiketlenmiş görüntü bulunmaktadır. ImageNet veri setinden araştırmacılar, eğitimciler ve öğrenciler ücretsiz şekilde yararlanabilmektedir. 2011 yılında Alex Krizhevsky tarafından geliştirilen AlexNet ile makine öğrenmesi ve derin öğrenme alanlarında büyük başarılar elde etmiştir. AlexNet başlangıç olarak sadece 8 katman içermekteydi. Bu 8 katmanın 3 tanesi tam bağımlı katmandan oluşmaktaydı. AlexNet ile eğitimi daha hızlı getirmek için doymamış nöronlar ile verimli bir GPU uygulaması oluşturulmuştur. Tamamen bağı katmanlardaki aşırı öğrenmeyi (overfitting) engellemek için bırakma (dropout) yöntemi kullanılmıştır. Bu yöntem ile ILSVRC-2012 ImageNet yarışmasında olağanüstü sonuçlar elde edilmiştir ve geçerliliği kanıtlanmıştır (Krizhevsky, Sutskever, Hinton, 2012). 2014 yılında Ian Goodfellow ve ekibi tarafından Çekişmeli üretici ağlar (GAN) tanıtılmıştır. GAN yapısında aynı anda iki model eğitilmiştir. İlk model olan G verileri alır ve ayırt edilemez örnekler yaratmaya çalışırken ikinci ağ olan D ise hem verileri hem de oluşturulan örnekleri alır ve her verinin orijinal mi yoksa üretilip üretilmediğini belirlemeye çalışmaktadır. Modeldeki ağlar eşzamanlı bir şekilde eğitilerek birbiri ile rekabet etmesini sağlar ve birbirlerini daha akıllı ve daha hızlı hale getirmeye zorlamaktadır (Goodfellow vd., 2014).

Derin öğrenme alanının yaşanan gelişmeler Apple, Google, Microsoft gibi önde gelen teknoloji firmaların ve ülkelerin bu alana yatırım yapmasına sebep olmuştur. Şekil 1.3’de 2010 yılından itibaren önde gelen firmaların yapay zekâ başlangıç firmalarını satın alma sayıları verilmiştir. Şekil 1.4’de ülkelerde yer alan yapay zekâ başlangıç firmalarının sayılarını gösteren dünya haritası yer almaktadır. Ayrıca ilk 20 ülkede yer alan firma sayıları ile dünya üzerindeki merkezi 20 bölge isimleri ve firma sayıları gösterilmektedir.



Şekil 1.3. Önde gelen teknoloji şirketleri tarafından yapılan yapay zekâ başlangıç firmalarının satın alımlarının toplam sayısı (Yapay zekâ, 2022).



Şekil 1.4. Ülke bazındaki yapay zekâ başlangıç firma sayıları ve bilgileri (Girişim, 2022).

1.1.3 Derin Öğrenme Yaklaşımları

Makine öğreniminde olduğu gibi, derin öğrenme yaklaşımları da kategorize edilebilir. Bu yaklaşımlar denetimli (supervised), yarı denetimli (semi-supervised) veya kısmen denetimli ve denetimsiz (unsupervised) olarak kategorize edilebilir. Ek olarak, yarı denetimli veya denetimsiz öğrenmenin altında değerlendirilecek olan pekiştirmeli öğrenme (RL veya DRL) yaklaşımı da mevcuttur (Alom vd., 2018).

1.1.3.1 Denetimli Öğrenme

Denetimli öğrenme, etiketli verileri kullanan bir öğrenme tekniğidir. Denetimli derin öğrenme yaklaşımlarında, ortamda bir dizi giriş ve karşılık gelen çıkışlar bulunur. Örneğin, girdi için çıktı değerini tahmin ederse, aracı bir kayıp değeri alacaktır. Aracı daha sonra istenen çıktıların daha iyi yaklaştırılması için ağ parametrelerini yinelemeli olarak değiştirecektir. Başarılı bir eğitimden sonra, temsilci çevreden gelen sorulara doğru cevapları alabilecektir. Derin Sinir Ağları (DNN), Evrişimsel Sinir Ağları (CNN), Uzun Kısa Vadeli Hafıza (LSTM) dahil Tekrarlayan Sinir Ağları (RNN) ve Geçitli Tekrarlayan Birimler (GRU) dahil olmak üzere derinlemesine eğitim için farklı denetimli öğrenme yaklaşımları bulunmaktadır (Alom vd., 2018).

1.1.3.2 Yarı Denetimli Öğrenme

Yarı denetimli öğrenme, kısmen etiketlenmiş veri setlerine (genellikle pekiştirmeli öğrenme olarak da adlandırılır) dayalı olarak gerçekleşen öğrenmedir. Bazı durumlarda, Derin pekiştirmeli öğrenme ve Çekişmeli Üretici Ağlar (Generative Adversarial Networks-GAN) yarı denetimli öğrenme teknikleri olarak kullanılır. Ek olarak, Uzun Kısa Vadeli Hafıza ve Geçitli Tekrarlayan Birimler içeren tekrarlayan sinir ağları, yarı denetimli öğrenme için de kullanılmaktadır (Alom vd., 2018).

1.1.3.3 Denetimsiz Öğrenme

Denetimsiz öğrenme yaklaşımları, veri etiketleri olmadan öğrenme yapabilen sistemlerdir. Bu durumda aracı, girdi verilerindeki bilinmeyen ilişkileri veya yapıyı keşfetmek için dahili temsili veya önemli özellikleri öğrenmektedir. Genellikle kümeleme, boyut azaltma ve üretken teknikler denetimsiz öğrenme yaklaşımları olarak kabul edilmektedir. Derin öğrenme ailesinin, Oto-kodlayıcılar (AE), Sınırlandırılmış Boltzmann Makineleri (RBM) ve

yakın zamanda geliştirilen Çekişmeli Üretici Ağlar dahil olmak üzere kümelemede ve doğrusal olmayan boyutluluk azaltmada iyi olan birkaç üyesi bulunmaktadır. Ek olarak, Uzun Kısa Vadeli Hafıza ve Pekiştirmeli Öğrenme gibi Tekrarlayan Sinir Ağların gibi birçok uygulama alanında denetimsiz öğrenme için kullanılmaktadır (Alom vd., 2018).

1.1.3.4 Derin Pekiştirmeli Öğrenme

Derin Pekiştirmeli Öğrenme, bilinmeyen ortamlarda kullanılan bir öğrenme tekniğidir. Derin Pekiştirmeli Öğrenme, 2013 yılında Google Deep Mind ile başlamıştır (Mnih, vd., 2015; Mnih, vd., 2013). Pekiştirmeli Öğrenme 'de, basit bir kayıp fonksiyonumuz yoktur, bu nedenle öğrenmeyi geleneksel denetimli yaklaşımlara kıyasla daha zor hale getirmektedir. Pekiştirmeli öğrenme ile denetimli öğrenme arasındaki temel farklar şunlardır: birincisi, optimize etmeye çalıştığınız işleve tam erişiminiz yoktur; onları etkileşim yoluyla sorgulamalısınız ve ikincisi, durum tabanlı bir ortamla etkileşim halinde olduğu için girdi eylemlere bağlıdır (Alom vd., 2018).

1.1.4 Derin Öğrenme Mimarileri

1.1.4.1 Evrişimli Sinir Ağları

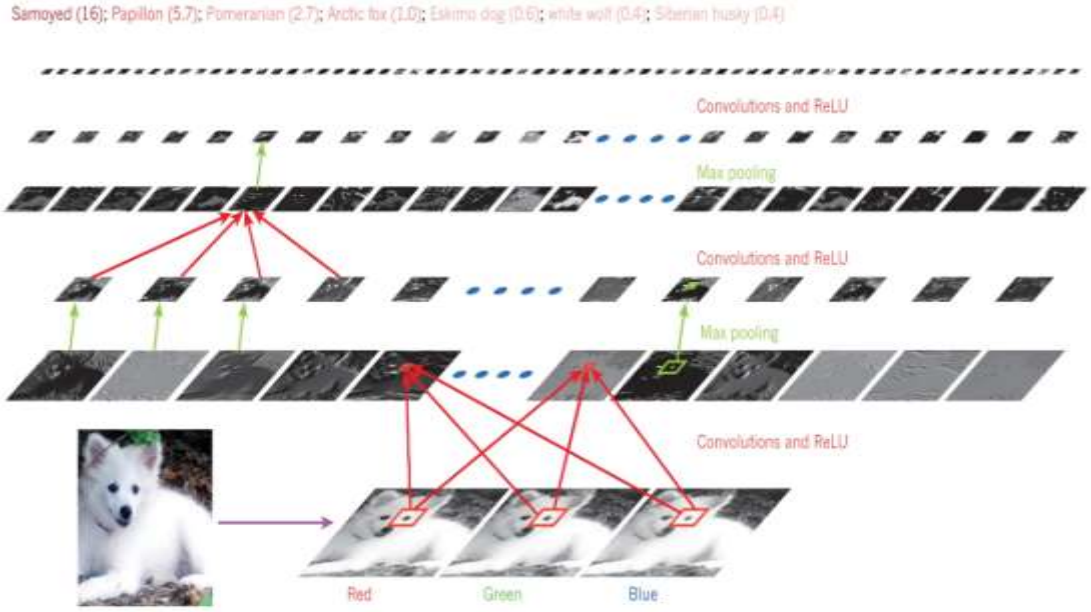
CNN çok katmanlı algılayıcıların kullanıldığı bir türdür. Evrişimsel sinir ağlarının temel tasarım ilkelerden bazıları sinirbiliminden alınmıştır. Hayvan beynindeki görme merkezinin çalışma mantığı araştırılarak ortaya konulmuştur. Görme merkezinde, görüntü öncelikle görüntünün tümünü kapsayacak şekilde alt bölgelere ayrılır. Basit hücreler ile görüntüde yer alan nesnelerin kenar özellikleri gibi yerel özelliklerin belirlenmesi sağlamaktadır. Karmaşık hücreler ise basit hücreler tarafından algılananlara benzer özellikler ile cevap verir ancak karmaşık hücreler cevap verilen özelliğin konumundaki küçük kaymalar ile değişmez ve tüm görsele odaklanmaktadır (LeCun vd., 2015; Goodfellow, Bengio, Courville, 2016).

Evrişimsel Sinir Ağları mimarisi birden çok katmandan ve bir dizi aşamadan oluşmaktadır. İlk birkaç aşama iki farklı katmandan oluşur: Evrişimsel katmanlar (convolutional layers) ve havuz katmanları (pooling layers). Evrişimli katmandaki birimler, her birimin ağırlık setleri filtre bankası adı verilen bir yapı aracılığıyla bir önceki katmanın özellik haritalarındaki yerel yamalara bağlandığı özellik haritaları tarafından düzenlenmektedir. Bu

yerel ağırlıklı toplamların sonucu daha sonra ReLU fonksiyonunda geçirilir. ReLU, doğrusal olmayan bir fonksiyondur.

$$f(z) = \max(z, 0) \quad (1.1)$$

Negatif değerler için sıfır değerini çıktı olarak verirken pozitif değer kendisini çıktı olarak vermektedir. ReLU birçok katmana sahip ağlarda çok daha hızlı öğrenmektedir. Özellik haritasındaki tüm birimler aynı filtre kümelerini paylaşmaktadır. Katmandaki farklı özellik haritaları farklı filtre kümelerini kullanmaktadır. Böyle bir mimariye sahip olamamasının sebebi iki yönlü olmasıdır. İlk olarak görüntüler dizi verilerinde tutulur ve yerel değerler genellikle yüksek oranda ilişkilidir ve kolayca tespit edilebilen ayırt edici desenlerden oluşur. İkinci olarak görüntülerin ve diğer girdilerin yerel istatistikleri konuma göre değişmez. Diğer bir deyişle görüntünün bir bölümünde bir desen görünebilirse herhangi bir yerde de görünebilir dolayısıyla farklı konumlardaki birimlerin aynı ağırlıkları paylaşır ve dizinin farklı bölümlerinde aynı modeli algılar. Evrişimsel katmanın rolü bir önceki katmandaki özelliklerin yerel bağlantıları tespit olsa da havuz katmanının(pooling layer) rolü benzer özellikleri bir araya getirmektir. Tipik havuz katmanında bir özellik haritasında veya birkaç özellik haritasında en fazla yer alan yamaları hesaplar. Komşu havuz birimleri, birden fazla satır veya birden fazla sütun tarafından kaydırılan yamaları girdi olarak alır. Böylece temsilin boyutunu azaltır ve böylece küçük kaymalara ve bozulmalara karşı değişmezlik yaratır. İki veya üç evrişim katmanı, doğrusal olmama ve havuzlama aşamaları arka arkaya dizilir ve sonrasında bunları daha çok evrişimsel katman ve tam bağlantılı katmanlar izler (LeCun, Bengio, Hinton, 2015). Şekil 1.5'te görüldüğü gibi evrişimsel ağ mimarisi bir dizi aşamadan meydana gelmektedir.

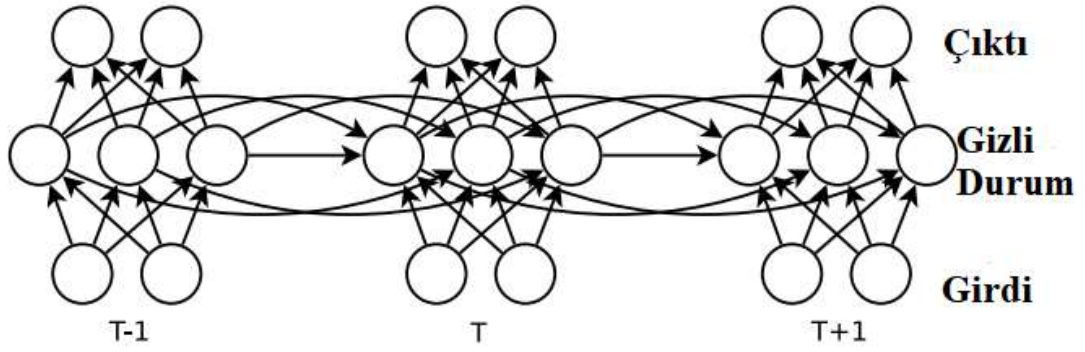


Şekil 1.5. Evrişimsel ağ mimarisi (LeCun, Bengio, Hinton, 2015).

Evrişimsel Sinir Ağları, birçok sinyalin daha düşük seviyeli özellikleri daha üst seviyedeki özelliklerin elde edildiği hiyerarşilerdir. Konuşma tanıma ve belge okuma için 1990'ların başından itibaren çok sayıda CNN uygulaması oluşturulmuştur. 1990'ların başından itibaren ayrıca görüntü ve ses işleme gibi alanlar başta olmak üzere biyomedikal, doğal dil işleme gibi birçok farklı alanda uygulanmıştır. Özellikle görüntü işleme alanında başarılı sonuçlar elde edilmiştir. 2014 yılında ImageNet yarışmasında milyonlarca resim ve yüzlerce nesne sınıflandırması ve algılanmasında elde edilen başarılarla kullanılan algoritmalarda CNN algoritmaları kullanılmıştır (LeCun, Bengio, Hinton, 2015; Şeker vd., 2017).

1.1.4.2 Tekrarlayan Sinir Ağları

Tekrarlayan Sinir Ağı sıralı verilerin modellenmesini sağlamak için standart ileri beslemeli sinir ağlarının basit bir uyarlamasıdır. İlk olarak Jeff Elman tarafından tasarlanmıştır. Tekrarlayan Sinir Ağı bir girdiyi alır, gizli durumunu günceller ve bir tahmin yapar. Şekil 1.6'da tekrarlayan sinir ağı mimarisi yer almaktadır.



Şekil 1.6. Tekrarlayan sinir ağı mimarisi (Sutskever, Martens, Hinton, 2011).

Elman tarafından oluşturulan simülasyonda kullanılan kelimeler için örüntü kümeleme sonucunda göre fiil ve isim olarak başarılı bir şekilde kategorize edilmiştir. İsim olarak kategorize edilen kelimeler ise kendi arasında canlı ya da cansız olmasına göre, cansız olarak kategorize edilen kelimeleri yenilebilir ve yenilemez olarak, canlı olarak kategorize edilen kelimeleri ise insan ve hayvan olarak alt kategorilere kadar ayırt etmeyi başarılı bir şekilde yapmıştır (Elman, 1990). Tekrarlayan sinir ağı mimarisinde sıralı bilgilerin kullanılması ile sonuçlar elde edilir. Sıralı bilgiler kullanılırken kullanım alanına göre girdiler ve çıktılar birbiri ile bağımlı ya da bağımsız oldukları varsayılarak mimari oluşturulur. Örneğin görüntü üzerinde işlem yapılırken girdilerin ya da çıktılar birbirinden bağımsız olduğu varsayımı üzerine mimari kurulmaktadır; ancak zaman değişkeni olan problemler için mesela cümle içerisindeki bir sonraki kelimeyi tahmin etmek için mimari oluşturulurken mevcut kelimedenden önce gelen kelimeyi bilmek gerekmektedir.

1.1.4.3 Uzun Kısa Vadeli Hafıza Ağları

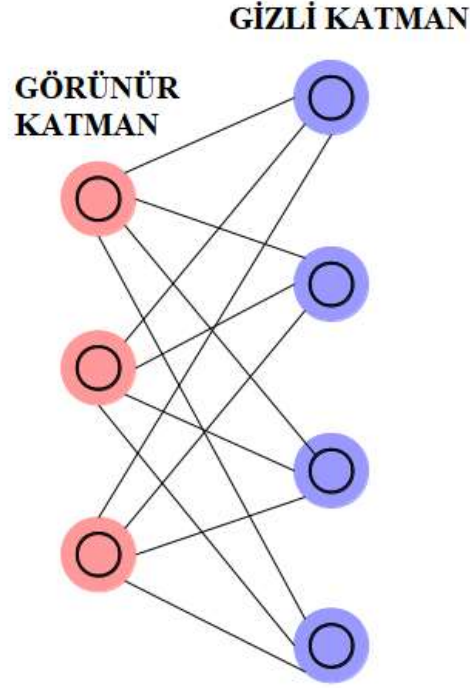
Tekrarlayan sinir ağı mimarisinde daha önce girdi olarak kullanılan bilgilerin kullanımına dayalı bir yaklaşım mevcuttur. Bu yaklaşım ile basit cümlelerde verimli bir şekilde çalışmaktadır. Ancak birbiri ile bağlamı olan cümleler arasındaki boşluk arttığında tekrarlayan sinir ağının bu bilgiyi kullanması güçleşmektedir. Uzun vadeli bağımlılıklar sinir ağı uygulanırken problemlere yer açtığı gözlenmiştir. Bu probleme çözüm olarak Schmidhuber ve Hochreiter tarafından 1997 yılında uzun vadeli bağımlılıkları öğrenebilen bir tekrarlayan sinir ağı türü olan uzun kısa vadeli hafıza (LSTM) önerilmiştir. Uzun vadeli bağımlılık sorununa çözüm bulunarak tekrarlayan sinir ağlarının hem verimliliğinin artması sağlanmıştır hem de uygulanabilirliğini arttırmışlardır. Uygulama esnasında gerekli olan bilgi tekrarlayan sinir ağlarında çok "geride" kaldığında ve "kaybolduğunda" bile uzun kısa vadeli hafıza ağları bu

bilgiyi daha uzun bir süre "hatırlayabilir" (Hochreiter, Schmidhuber, 1997). Uzun kısa vadeli hafıza ağları mimarisinde giriş kapısı, unutma kapısı ve çıkış kapısı olmak üzere 3 adet kapı bulunmaktadır. 3 kapı hata döngüsü, çıkış aktivasyon fonksiyonu ve gözetleme bağlantılarına sahiptirler. Bloкта çıkan veri bloğun girişine ve bloktaki tüm kapılara tekrar tekrar bağlanmaktadır. Gözetleme bağlantılar ile birlikte unutma kapısı geliştirilip öne sürülen ilk mimaride yer almamaktadır. Bloğun ve ağın kendi durumunu sıfırlamak amacı ile unutma kapısı eklenmiştir. Gözetleme bağlantıları ise zamanlamayı öğrenmeyi kolaylaştırmak amacı ile mimariye sonradan eklenmiştir (Gers, Schmidhuber, Cummin, 2000; Gers, Schmidhuber, 2000).

Uzun kısa vadeli hafıza ağı mimarisin en çok kullanıldığı alanlar konuşma ve metin işleme uygulamalarıdır. 2 alanda yapılan çalışmalarda başarılı sonuçlar elde edilmiştir. Ayrıca bu mimari yapısı düzensiz dillerde öğrenme, çevrimdışı el yazısı tanınması, protein homolojisinin algılanması gibi birçok farklı alanda başarılı bir şekilde kullanılmıştır.

1.1.4.4 Sınırlı Boltzmann Makineleri

Sınırlı boltzmann makineleri, 1986 yılında Harmonium ismi ile ilk olarak öne atılmıştır (Smolensky, 1986). Uzun yıllar sonra 2006 yılında Geoffrey Hinton ve Ruslan Salakhutdinov tarafından hızlı bir öğrenme algoritması olarak öne sürülmüştür (Hinton, Salakhutdinov, 2006). Sınırlı boltzmann makineleri girdi setinin üzerinde yer alan olasılık dağılımlarını öğrenebilen rastgele bir yapay sinir ağıdır. Sınırlı boltzmann makineleri derin inanç ağlarının yapı taşı oluşturmaktadır. Mimarisinde iki katmanlı sinir ağı bulunmaktadır. İlk katmanı olan görünür katman ve ikinci katmanı olan gizli katmandan oluşmaktadır. Şekil 1.7'de sınırlı boltzmann makineleri mimarisi yer almaktadır.



Şekil 1.7. Sınırlı boltzmann makineleri mimarisi

Her düğüm sinir ağındaki bir nöronu temsil eder ve hesaplamaların gerçekleştirildiği yerdir. Katmanlar arasındaki düğümler birbiri ile bağlanır ancak aynı katmanda yer alan düğümler birbiri ile bağlanmazlar (Salakhutdinov, Hinton, 2009). Bu sebeple katmanlar arasında iletişim bulunmamaktadır ki mimarinin adı da buradan gelmektedir. Her düğümde girdi olarak alınan veriler hesaplanır ve sonraki düğüme iletilip ileilmeyeceği rastgele kararlar verir. Sınırlı Boltzmann Makineleri, özellik öğrenme, konu modelleme gibi birçok farklı alandaki konular için kullanışlı bir algoritmadır.

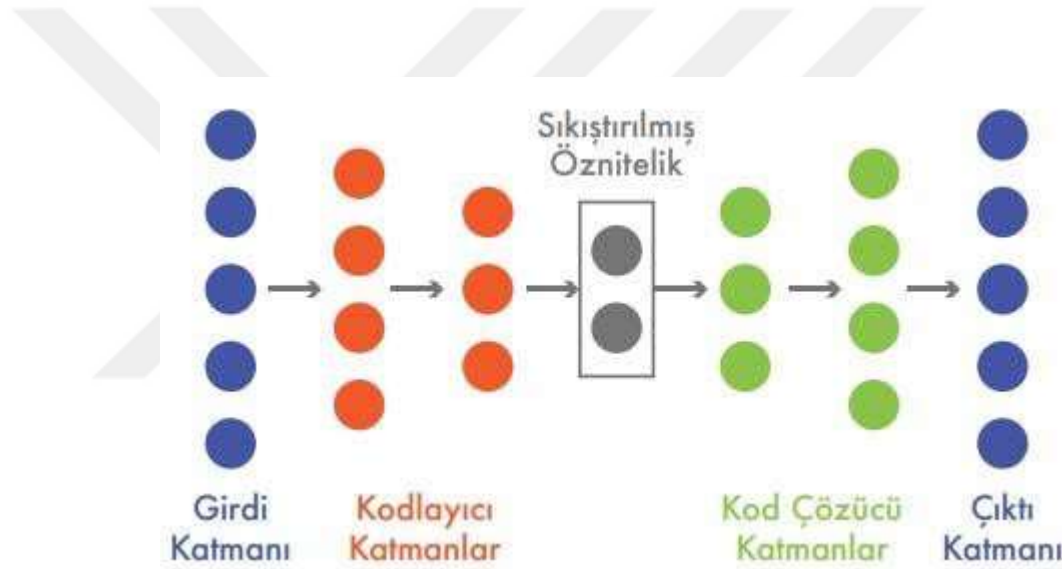
1.1.4.5 Derin İnanç Ağları (DBN)

Geoffrey Hinton ve Ruslan Salakhutdinov tarafından öne sürülen ve sınırlı boltzmann makinelerini yapı taşı olarak alan bu mimarinin yığını olarak tanımlanmaktadır. Sınırlı boltzmann makinelerinde katmanlarda arasında bir iletişim mevcuttur fakat aynı katmanda yer alan düğümler arasında iletişim yoktur. Derin inanç ağları mimarisinde en katman olarak bir softmax katmanı ile sınıflandırma yapılmaktadır ya da denetimsiz öğrenme yapılabilmesi için

kümeleme özelliğine sahiptir (Hinton, Salakhutdinov, 2006). Derin inanç ağılar mimarisi görüntü tanıma ve görüntü üretme konularında kullanılmaktadır.

1.1.4.6 Derin Oto-Kodlayıcılar

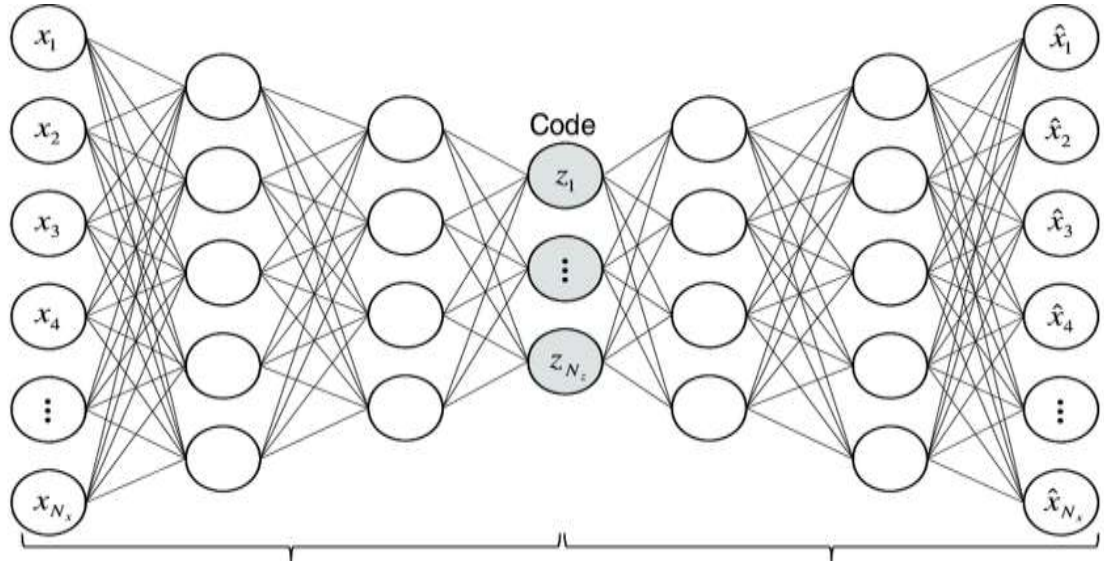
Oto-kodlayıcılar (AE), diablo ağı olarak adlandırılan ve denetimsiz öğrenme gerçekleştirmek için kullanılmakta olan hususi yapay sinir ağıdır (Bengio, 2009). Oto-kodlayıcılar girdi katmanında kullanılan değerleri çıktı katmanına kopyalayan bir sinir ağıdır. Sinir ağına verilen girdiyi çıktı katmanında tekrar oluşturmaktadır. Oto-kodlayıcılar girdinin sıkıştırılmış halindeki en iyi özelliklerin öğrenilmesini amaçlayan ileri beslemeli bir sinir ağıdır (Krizhevsky, Hinton, 2011). Şekil 1.8’de Oto-Kodlayıcılar mimarisi gösterimi yer almaktadır.



Şekil 1.8. Oto-Kodlayıcılar mimarisi (Kayaalp, Süzen, 2018).

Şekil 1.8’de gösterildiği üzere mimari girdiyi şifreleyip daha sonra şifre çözme işlemlerinden sonra çıktı olarak girdiyi sağlayana kadar ağırlıklarını değiştirir. Bu hedefe ulaşana kadar ağırlıklarda değişiklikler yapılır.

Derin oto-kodlayıcılar (Deep Autoencoder - DAE), katmanda oluşturulan çıktının mevcut katmanı takip eden sonraki katmana girdi olarak bağlandığı birden çok oto-kodlayıcı katmanından oluşan sinir ağıdır (Şeker, Yüksek, 2017). Şekil 1.9’da derin Oto-Kodlayıcılar mimarisi yer almaktadır.



Şekil 1.9. Derin Oto-Kodlayıcılar mimarisi (Canchumuni, Emerick, Pacheco, 2019).

Şekil 1.9’da gösterildiği gibi girdilerden oluşan dizi ilk gizli katmandaki şifrelenmiş dizi ile oluşturulmuştur. Sonraki katmana girdi olarak verilen dizi bir sonraki gizli katmandaki şifreleme sonucu oluşturulmuştur. Bu aşamaların tekrar edilerek bitmesi sonucunda kodlama işlemi gerçekleştirilmiş olmaktadır. Şifreleme işleminden sonra şifreyi çözme aşaması yer almaktadır. Şifreleme aşamasında yapılan işlemlerin tersi yapılarak şifresi çözülen dizi bir sonraki katmana girdi olarak verilmekte ve şifre çözme işlemi bitene kadar devam etmektedir. Derin Oto-Kodlayıcılar, görüntü tanıma, özellik sınıflandırma gibi birçok farklı alanda kullanılmaktadır.

1.1.5 Derin Öğrenme Uygulamaları

1.1.5.1 Görüntü/Video İşleme

Görüntü işleme derin öğrenme uygulamaları arasında oldukça popüler bir uygulama alanıdır. Görüntü işleme alanında karşılaşılan en büyük problemlerden biri ise görüntü sınıflandırmasıdır. Bu problem ile yapılan çalışmalarda ortak olarak MNIST veri seti kullanılmaktadır. MNIST veri seti eğitim için 60000 veriye ve test için ise 10000 el yazısı ile yazılmış rakamlardan oluşmaktadır. Yapay sinir ağları da dâhil olmak üzere birden fazla farklı alanda bu veri seti kullanılmıştır (LeCun, Cortes, Burges, t.y). MNIST veri seti ile bugüne kadar yapılmış en başarılı sonuç alanın çalışmalardan biri Cireşan ve arkadaşlarının 2012 yılında ortaya koyduğu %0,23 hata ile evrişimli sinir ağları modelinin kullanıldığı çalışmadır (Cireşan, Meier, Schmidhuber, 2012).

Ciresan ve arkadaşları evrişimli sinir ağı modeli kullanarak 2011 yılında görsel desen tanıma yarışmasında süper insan performansı elde eden ilk ekip olmayı başarmışlardır (Cireşan, Meier, Gambardella, Schmidhuber, 2012). Ciresan ve ekibi ICDAR Çin el yazısı yarışmasında ve 2012’de resim bölümlene yarışmasını da kazanmışlardır (Cireşan, Guisti, Gambardella, Schmidhuber, 2012a). Alex Krizhevsky, Geoffrey Hinton ve ekibinin oluşturmuş oldukları sistem ImageNet yarışmasını diğer makine öğrenme yöntemlerine karşı büyük üstünlük elde edip, yarışmayı kazanmışlardır (Krizhevsky, Sutskever, Hinton, 2012). Oluşturulan benzer bir sistem ise ICPR ve MICCAI yarışmalarında kanser tespiti için büyük biyomedikal görüntülerin işlenmesi konusunda başarı kazanmışlardır (Cireşan, Guisti, Gambardella, Schmidhuber, 2012b).

Bu çalışmaların yanı sıra derin öğrenme mimarisi temelli denetimsiz öğrenme algoritmaları etiketsiz verilerden sınıflara özgü özelliklerin çıkarımı konusunda sıkça kullanılmaktadır. 2011 yılında Quoc V. Le tarafından gerçekleştirilen çalışmada havuzlama ve normalizasyon ile birlikte 9 katmanlı derin oto-kodlayıcı kullanılarak internetten indirilen 10 milyon görüntü içeren bir veri kümesi eğitilmiştir. 1000 makine ile gerçekleştirilen bu çalışmada paralel bir küme ile eğitim 3 günde gerçekleştirilmiştir. Uygulanan bu model ile birlikte yüz etiketleme yapmadan yüz tanıma sistemi gerçekleştirilmiştir. Hatta yapılan bu çalışma ile birlikte insan vücudunun parçası ve kedi yüzü tanımlamak da mümkün olmuştur (Le, 2013).

Derin öğrenme algoritmaları nesne tanıma problemleri üzerinde oldukça başarılı sonuçlar alınmasını sağlamaktadır. Derin kalıntı ağı kullanarak oluşturulan mimari ILSVRC 2015, ImageNet algılama ve yerelleştirme, COCO segmentasyon yarışmalarında 1. olmuştur (He, Zhang, Ren, Sun, 2016). Görüntüde yer alan nesnelerin etiketlenmesi problemi mevcut görüntünün ham piksellerin ait olduğu nesnenin kategorisine dahil edilmesiyle çözülebilir. Farabet ve ekibinin yapmış olduğu çalışmada çoklu ölçekli evrişimsel sinir ağı mimarini sahip bir model önermiştir. Geliştirilen bu model Barcelona, Stanford arka plan verileri üzerinde daha hızlı ve başarılı sonuçlar elde edilmesini sağlamıştır (Farabet, Couprie, Najman, LeCun, 2012).

Biyomedikal alanındaki görüntü işleme problemlerinin çözümü de aynı zamanda görüntü işlemenin bir alt dalı olarak kabul edilebilir. Bu alanda da derin öğrenme yöntemleri başarılı sonuçlar ortaya koymuştur (Shen, Wu, Suk, 2017). Beyin tümörü ile ilgili çalışmalarda

görüntünün segmentasyonuna ihtiyaç duyulmaktadır. Bu ihtiyacı gidermek için derin öğrenme yöntemleri başarılı bir şekilde segmentasyon işlemini başarılı bir şekilde yapmaktadır. Beyin tümörü için büyük miktardaki MRI görüntülerinde otomatik segmentasyon işleminin gerçekleştirilmesinde evrimsel sinir ağları başarılı sonuçlar ortaya koymuştur (Havaei vd., 2017; Urban, Bendszus, Hamprecht, Kleesiek, 2014). Biyomedikal alanında sadece beyin tümörü segmentasyonunda değil meme kanseri, akciğer kanseri tespiti gibi birçok farklı alanda derin öğrenme yöntemleri kullanılmıştır (Zhang vd., 2016; Sun, Zheng, Qian, 2017).

1.1.5.2 Doğal Dil İşleme

Doğal dil işleme alanında oldukça başarılı sonuçlar veren derin öğrenme algoritmaları mevcuttur. Derin oto-kodlayıcılar, uzun kısa vadeli hafızalar ve evrimsel sinir ağı mimarileri oldukça başarılı sonuçlar vermiştir.

2000'li yıllardan başlayarak, sinir ağları, önceki kelimeleri verilen bir metinde sonraki kelimeyi tahmin etmeyi amaçlayan dil modellemesi için kullanılmaya başlandı. 2003 yılında Bengio ve arkadaşları tek gizli katmanlı ileri beslemeli sinir ağından oluşan ilk sinir dili modelini önerdi. Model, öğrenilen bir tabloya bakılan önceki n kelimeyi girdi vektörün temsilleri olarak aldı. Vektörler, çıktılar daha sonra dizinin bir sonraki kelimesini tahmin eden bir softmax katmanına sağlanan gizli bir katmana beslendi (Bengio, Ducharme, Vincent, 2000). Bu çalışma ağının genel yapı taşları günümüzde çoğu sinirsel dilde ve kelime gömme modellerinde hala bulunmaktadır.

Collobert ve Weston, doğal dil işleme için sinir ağlarına birden fazla öğrenme görevinin aynı anda çözüldüğü bir makine öğrenimi alt alanı olan çok görevli öğrenmeyi uyguladılar. Tek bir evrimsel sinir ağı mimarisi kullandılar ve bir cümle verildiğinde, konuşma bölümü etiketleri, adlandırılmış varlık etiketleri ve anlamsal roller gibi birçok dil işleme tahminini çıktı olarak verdiler (Collobert, Weston, 2008).

2013 yılında Mikolov ve arkadaşları en popüler kelime gömme modelini tanıttılar: Word2Vec. Kelimelerin yoğun vektör temsilleri 2003 gibi erken bir tarihte kullanılmış olsa da, çalışmalarında önerilen ana yenilik, gizli katmanı kaldırarak ve hedefe yaklaşarak eğitim prosedürünün verimli bir şekilde iyileştirilmesiydi. Daha yeni birçok gelişmeye rağmen, Word2Vec hala popüler bir seçimdir ve günümüzde yaygın olarak kullanılmaktadır (Mikolov, Chen, Corrado, Dean, 2013; Mikolov, Sutskever, Chen, Corrado, Dean, 2013).

2014 yılında Sutskever ve arkadaşları tarafından önerilen sekans-sekans öğrenme, bir sinir ağı kullanarak bir sekansı diğerine eşlemek için genel bir uçtan uca yaklaşımdır. Yöntemlerinde, bir kodlayıcı sinir ağı, bir cümle sembolünü sembole göre işler ve onu bir vektör temsiline sıkıştırır. Daha sonra, bir kod çözücü sinir ağı, kodlayıcı durumuna ve her adımda girdi olarak alınan önceden tahmin edilen sembollere dayalı olarak çıkış dizisini sembol ile tahmin etmektedir (Sutskever, Vinyals, Le, 2014).

2015 yılında Bahdanau ve arkadaşları nöral makine çevirisindeki (NMT) temel yeniliklerden biri olan dikkat ilkesini ve nöral makine çevirisi modellerinin klasik cümle tabanlı makine çevirisi sistemlerinden daha iyi performans göstermesini sağlayan anahtar fikrini tanıttılar. Temel olarak, kaynak dizinin tüm içeriğini sabit boyutlu bir vektöre sıkıştırma gereksinimi olan diziden diziye öğrenmenin ana darboğazını hafifletmektedir. Kod çözücünün daha sonra ağırlıklı ortalamayla birleştirilen ve kod çözücüye ek girdi olarak sağlanan kaynak dizisi gizli durumlarına geri bakmasına izin vermektedir. Dikkat, girdinin belirli kısımlarına dayalı kararlar almayı gerektiren herhangi bir görev için potansiyel kullanıma uygundur (Bahdanau, Cho, Bengio, 2014).

Doğal dil işleme alanındaki en son büyük yenilik, şüphesiz büyük önceden eğitilmiş dil modelleridir. İlk olarak 2015'te Dai ve Le tarafından önerilmiş olsalar da, çok çeşitli görevlerde son teknoloji yöntemlere göre büyük bir gelişme sağladığı ancak son zamanlarda gösterilmektedir. Önceden eğitilmiş dil modeli yerleştirmeleri, bir hedef modelde özellikler olarak kullanılmaktadır. Önceden eğitilmiş dil modellerinin temel avantajı, büyük açıklamasız metin külliyatından kelime temsillerini öğrenme yeteneklerinden gelir; bu, özellikle etiketli verilerin kıt olduğu düşük kaynak dilleri için yararlıdır.

1.1.5.3 Diğer Uygulamalar

Görüntü ve doğal dil işleme alanındaki birçok farklı alanda derin öğrenme mimarileri problem çözümleri için kullanılmıştır. İlaç endüstrisinde aday ilaçların olumsuz sonuçlarını tespiti için derin öğrenme mimarileri uygulanmıştır. Böylelikle kimyasal bileşendeki oluşmuş olası olumsuz sonuçların tespiti başarılı bir şekilde ele alınmıştır (Arrosmith, Miller, 2013). Genetik bilgilerin ortaya çıkarılması alandaki önemli problemlerden biridir. Bu problemin ortadan kaldırılması için derin oto-kodlayıcı mimarisi ile oluşturulmuş bir sistem önerilmiştir. Önerilen bu sistem başarılı sonuçlar ortaya koymuştur (Chicco, Sadowski, Baldi, 2014). DNA ve RNA dizilimlerini incelemek ve bilgi sahibi oluşan hastalık sebeplerini ve değişkenlerini

ortaya koymak için önem arz etmektedir. DNA ve RNA dizilimlerini özelliklerinin desen keşfini yapan ve esnek mimariye sahip bir yaklaşım sunan derin öğrenme modeli ortaya konulmuştur (Alipanahi, DeLong, Weirauch, Frey, 2015).

İnsanlar üzerinde hareket tespiti konusunda derin öğrenme mimarileri uygulanmıştır (Vinyals, Toshev, Bengio, Erhan, 2015; Arrowsmith, Miller, 2013). Pazarlama alanında müşteriye uygun pazarlama stratejilerini hesaplamak için yapay sinir ağı kullanılmıştır (Tkachenko, 2015). Akıllı trafik planlanması, oluşturulması hedeflenen akıllı şehirler için önemli bir etkiye sahiptir. Trafik ağındaki rastgele durumların ve özelliklerin üstesinden gelmek büyük bir zorluğa sahiptir. Niu ve ekibi DeepSense adını verdikleri derin öğrenme mimarisini kullandıkları model ile taksilerde yer alan GPS bilgilerini veri olarak alarak ve eğiterek trafik akışında tahminde bulunmuşlardır. Bu yöntem ile önerilen mevcut diğer yöntemlere göre %5 daha başarılı sonuçlar elde etmişlerdir (Niu, Zhu, Zhang, 2014).

DARPA tarafından, doğal dil işleme alanında başarılı sonuçlar vermiş yöntemler kullanarak Derin Araştırma ve Metin Filtreleme (Deep Exploration and Filtering of Text-DEFT) programı meydana getirilmiştir. Oluşturulan bu program ile büyük boyutlu verilerde başarılı bir şekilde yöntemlerin uygulanması sağlanması hedeflenmektedir. Ayrıca program ile belgelerdeki örüntülerin yani tekrar eden bilgilerin ortaya çıkarılması amaçlanmaktadır (Onyshkevych, 2022). Kapalı alanlarda yönlendirme ise karşılaşılan başka bir problemdir. Bu problemin çözümü için Kim ve Chen tarafından drone için otonom olarak kapalı bir mekanı dolaşabilen ve belirlenen bir hedefi bulmaya yarayan bir sistem önerilmiştir. Önerilen bu sistemde evrimsel sinir ağı mimarisi ile pilotlarında davranışları drone tarafından taklit edilmesi yapılan deneyler sonucunda ortaya konmuştur (Kim, Chen, 2015).

1.1.6 Derin Öğrenme Alanındaki Çalışma Ekipleri ve Kullanılan Kütüphaneler

1.1.6.1 Derin Öğrenme Alanındaki Çalışma Ekipleri

Derin öğrenme alanının her geçen gün popülerliği artması sebebi ile birçok üniversite ve büyük çaplı firmalar bu alanda araştırma yapan çalışma ekipleri oluşturmuşlardır. Çizelge 1.1'de bu alanda çalışma gösteren başlıca ekipler hakkında bilgiler yer almaktadır.

Çizelge 1.1. Derin öğrenme alanında çalışan ekipler (Şeker vd., 2017)

Araştırma Ekipleri	Ekip Üyeleri	Kütüphane	Çalışma Alanları
Toronto Üniv. - Makine Öğrenmesi	Geofrey Hinton	Torch	Doğal Dil İşleme
Montréal Üniv. - MILA Lab.	Yoshua Bengio	Theano	Doğal Dil İşleme
New York Üniversitesi - CILVR Lab.	Yann Lecun	Theano	Doğal Dil İşleme
Stanford Üniv. - SAIL ve SVL	Andrew Ng	TensorFlow	Bilgisayarlı Görü
Kaliforniya Üniv. - BAIR	Pieter Abbeel	Caffe	Bilgisayarlı Görü
Koç Üniv. - AI Lab.	Deniz Yuret	KNET	Doğal Dil İşleme
Google Research	Jeff Dean	TensorFlow	Görüntü İşleme
Facebook - FAIR	Yann Lecun	Caffe2	Doğal Dil İşleme
Twitter - Cortex	Hugo Larochelle	Torch	Doğal Dil İşleme
Microsoft - DLTC	Xiaodong He	Caffe	Doğal Dil İşleme
İsviçre U. B. Üniv. - IDSIA	Jurgen Schmidhuber	--	Görüntü İşleme
Baidu - Derin Öğrenme Enstitüsü	Lin Yuanqing Wei Xu	--	Doğal Dil İşleme

1.1.6.2 Derin Öğrenme Kütüphaneleri

Derin öğrenme alanında geliştirilen çok farklı kütüphaneler bulunmaktadır. Çizelge 1.2’de kütüphanelerin bilgileri ve bazı kullanım özellikleri yer almaktadır.

Çizelge 1.2. Derin öğrenme kütüphaneleri hakkında bilgiler (Derin Öğrenme Karşılaştırma, 2022)

Kütüphane	Yazıldığı Dil	Geliştirici	Önemli Özellikleri
TensorFlow	Python	Google	-Birden fazla CPU'ya ya da GPU'ya dağıtma olanağı.
Torch	Lua	Ronan Collobert, Clement Farabet, ...	-Kullanışlı arayüz -Algoritma oluştururken esnek ve hızlı bir yapıya sahip olması
Theano	Python	MILA Lab	-GPU desteği -API'ler sayesinde matematiksel hesapların kolaylıkla yapılması
Caffe	Python	Berkeley Vision and Learning Center (BVLC)	-GPU desteği -Caffe Model Zoo ile önceden eğitilmiş ağların kullanıma hazır olarak yer alması.
Digits	C++	NVIDIA	-Görselleştirme performansı sayesinde gerçek zamanlı izleme olanağı -Çoklu GPU desteği
DeepLearning	Java	Adam Gibson	- JVM tabanlı
Microsoft Cognitive Toolkit	C++	Microsoft Research	-Aynı zamanda birden çok GPU ve sunucu arasında paralel kullanım
DeepLearning4J	Java	Alex D. Black Adam Gibson	-Eğitimi hızlandırmak için dağıtılmış CPU ve GPU kullanımı

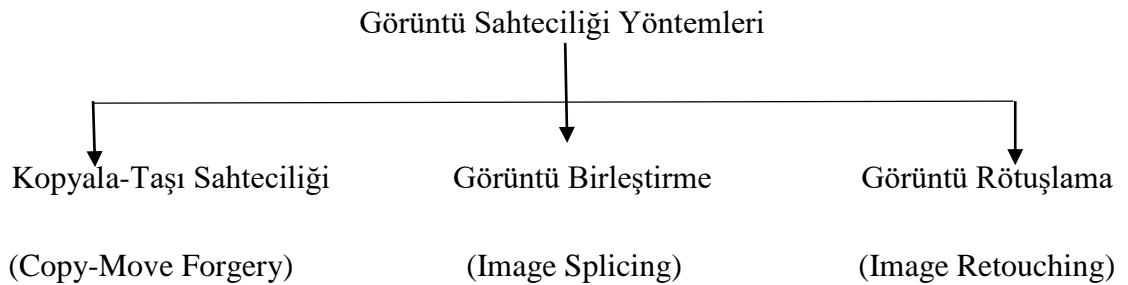
1.1.7 Fotomontaj Tespiti Hakkında

1.1.7.1 Fotomontaj Tespiti Nedir?

Fotomontaj diğer bir deyişle görüntü sahteciliği, fotoğrafın icadıyla birlikte geldiği için modern bir kavram değildir. Ancak, görüntü düzenleme yazılım araçlarıyla desteklenen kolay erişilebilir dijital kameraların icat edilmesiyle günümüzde ilgi odağı haline gelmiştir. Günümüzde dijital görsel medya, kullanımı kolay ve ucuz cihazların artması nedeniyle bilgi alışverişinde öne çıkan tekniklerden birini temsil etmektedir. Dahası, görsel medya, mevcut medyanın herhangi birinden daha fazla ifade potansiyeline sahiptir. Karışık sahneleri karmaşık olmayan bir şekilde anlatıyor, hangisi farklı bir şekilde kopyalamak oldukça zor olabilir. Kamuoyunun algısını değiştirmek adına aldatmak amacıyla dijital görüntülerin kötü niyetli olarak değiştirilmesi Dijital Görüntü Sahteciliği olarak adlandırılır. Değişiklik, neredeyse hiç görsel olarak saptanabilir iz bırakmayacak şekilde yapılır. Bir görüntünün sahte olup olmadığı, zarar verene kadar ya da vermedikçe ilgilenilmez. Bu görüntüler neredeyse herkes tarafından ve her yerde doğruluk belgesi olarak kabul edilmektedir. Bu nedenle, bir görüntünün gerçekliğinin doğrulanması gerekir. Böyle bir doğrulama, görüntü sahteciliği tespit teknikleri yardımıyla yapılır. Bu yöntemler, görüntülerin gerçekliğini doğrulamayı amaçlamaktadır. Bugüne kadar maruz bırakılan çeşitli görüntü sahteciliği türleri ve buna bağlı olarak sahtecilik tespit teknikleri bulunmaktadır (Deep Kaur, Kanwal, 2019).

1.1.7.2 Görüntü Sahteciliği Yöntemleri

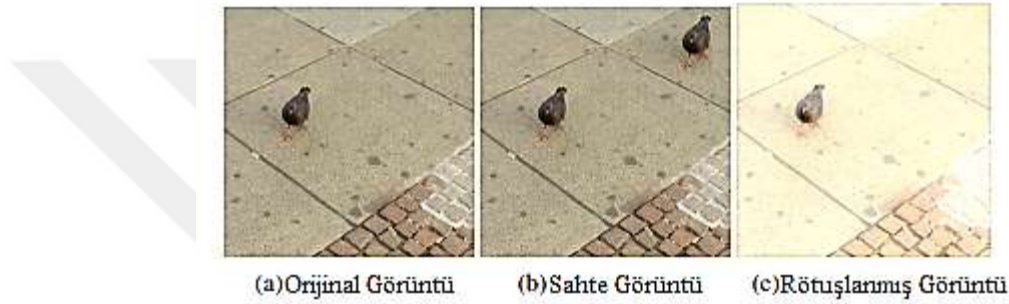
Görüntü, orijinal görüntüdeki bazı bölgeleri ekleyerek, kaldırarak veya değiştirerek, görsel olarak tespit edilebilir bir iz bırakmadan tek bir şey göz önünde bulundurularak taklit edilebilmektedir. Görüntüde, birkaç yöntem kullanılarak bozulma gerçekleştirilebilir. Bu yöntemler genellikle Şekil 1.10'daki gibi kategorize edilebilir:



Şekil 1.10. Görüntü sahteciliği yöntemleri

1.1.7.2.1 Kopyala-Taşı Sahteciliği

Kopyala-Taşı Sahteciliği, bir görüntünün bir kısmının çoğaltılmasını ve ardından aynı görüntüdeki başka bir alana yapıştırılmasını içermektedir. Amaç, orijinal görüntüdeki bazı bilgileri gizlemektir. Bir görüntüyü taklit etmek için en çok kullanılan yöntemdir. Kopyalanan parça aynı görüntüde kaldığından, gözle görülür önemli değişiklikler yoktur (Deep Kaur, Kanwal, 2019). Şekil 1.11’da görüntü üzerinde kopyala-taşı ve rötuşlama etkisinin gösterimine yer verilmiştir.



Şekil 1.11. Görüntü üzerinde kopyala-taşı ve rötuşlama etkisi

1.1.7.2.2 Görüntü Birleştirme Sahteciliği

Bir veya daha fazla görüntünün bileşimidir. Görüntüler, değiştirilmiş bir görüntü oluşturmak için birleştirilir. Kes / kopyala ve yapıştır işlemleri kullanılır. Bir görüntüden bir parça alınır ve başka bir görüntüye yapıştırılır. Bir görüntünün kesilmiş / kopyalanmış bölümünü başka bir görüntüyle tamamen birleştirmek için bazı işlem sonrası işlemlere ihtiyaç duyar. Yapıştırılan bölüm görüntünün modelini bozar. Böylece, görüntü deseninin analizi, görüntü sahteciliğinin saptanmasına yardımcı olur (Deep Kaur, Kanwal, 2019). Şekil 1.12’de iki farklı görüntü kullanarak görüntü birleştirme örneği yer almaktadır.



Şekil 1.12. İki farklı görüntü kullanarak görüntü birleştirme

1.1.7.2.3 Görüntü Rötüslama Sahteciliği

Bunda, görüntü temelde değişmez, ancak Şekil 1.11'de gösterildiği gibi orijinal görüntünün belirli bir özelliğinde bir gelişme, azalma vardır. Yumuşak, yıkıcı bir görüntü sahtekârlığıdır. Fotoğrafları daha çekici hale getirmek için dergi fotoğraf editörleri tarafından yaygın olarak kullanılır.

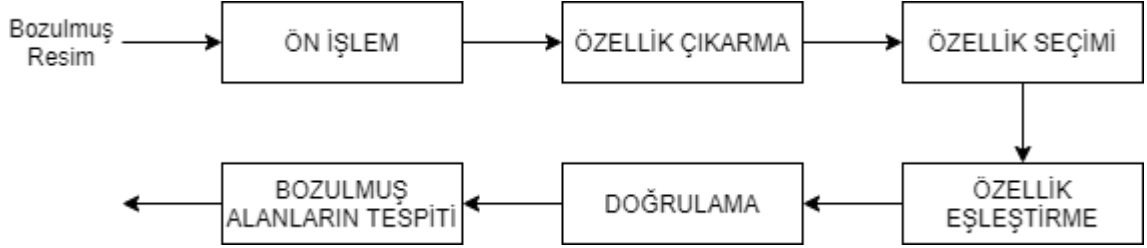
1.1.7.3 Görüntü Sahteciliği Algılama Yaklaşımları

Görüntü sahteciliği, sahteciliğinin tespiti için piksel tabanlı yöntemlerin geniş bir şekilde kullanılmasına yol açan basitliği nedeniyle genellikle piksel düzeyinde yapılmaktadır.

1.1.7.3.1 Kopyala-Taşı Sahtecilik Algılama Teknikleri

Kopyala-taşı sahteciliğinde, bir görüntünün farklı bölümleri kopyalanır ve aynı görüntüde farklı konumlara taşınır. Bir görüntünün farklı bölümleri, özellikleri açısından güçlü bir şekilde ilişkilidir. Tutarsız özellikler, bir görüntüyü üst üste binen bloklara veya ayrı blokla bölerek veya görüntünün tamamı için yerel anahtar noktalarını hesaplayarak hesaplanır. Bu özellikler kopya-hareket sahteciliğinin tespitinde önemli bir rol oynar. Genelleştirilmiş yapı ve ardından her kopya-hareket sahteciliği algılama tekniği Şekil 1.13'de gösterilmiştir. Kırpma, RGB görüntünün gri tonlamaya dönüştürülmesi, Ayrık Kosinüs Dönüşümü (DCT) veya Ayrık Dalga Dönüşümü (DWT) dönüşümü gibi işlemlerin tümü, sınıflandırma performansını artırmak için Ön İşleme tarafından yönetilir. Özellik Çıkarma ve Özellik Seçimi, bir görüntünün bir dizi özelliğinden manipülasyona duyarlı ve en bilgilendirici özelliklerin çıkarılmasını içerir. Özellik Eşleştirme, herhangi bir benzerlik bulmak için her bloğun seçilen özelliklerini diğeriyle karşılaştırır. Sahtecilik, bir görüntüdeki benzer blokların vurgulanmasıyla yerelleştirilir. Belirgin araştırmacılar, farklı özellik türlerinden

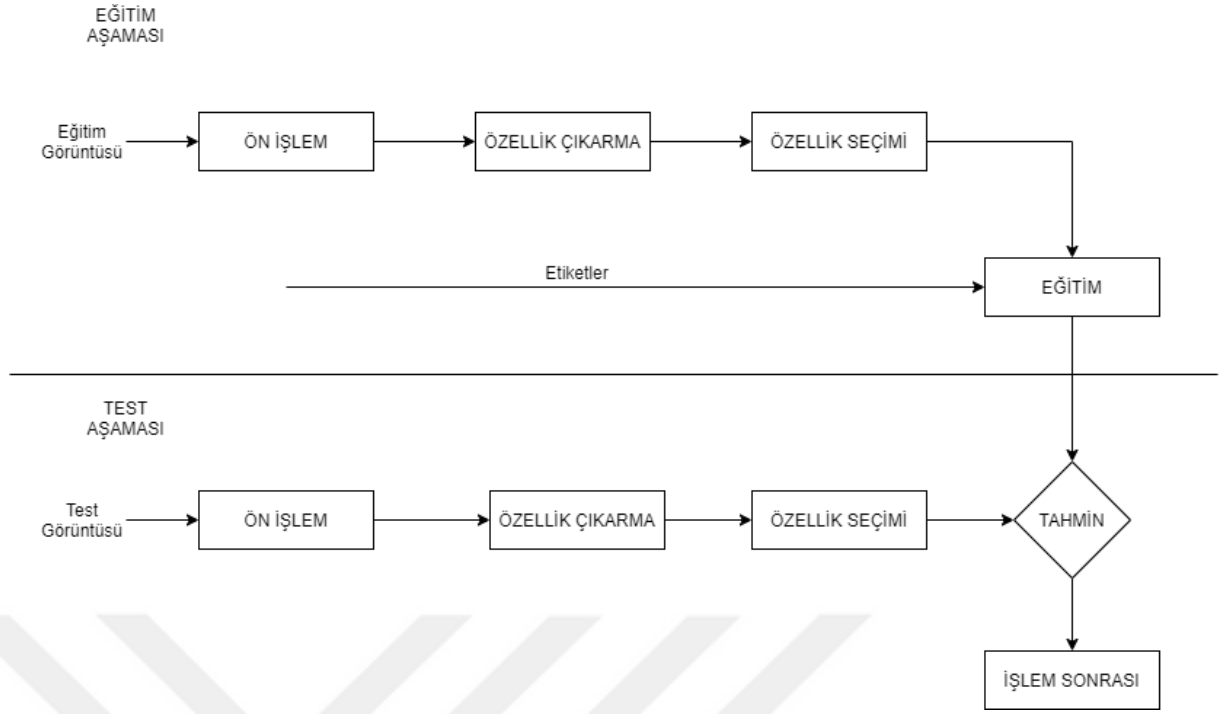
yararlanmaktadır (Deep Kaur, Kanwal, 2019). Şekil 1.13’de kopyala-Taşı sahteciliği tespitinin geliştirilmiş yapısına yer verilmiştir.



Şekil 1.13. Kopyala-Taşı sahteciliği tespitinin geliştirilmiş yapısı

1.1.7.3.2 Görüntü Birleştirme Algılama Teknikleri

Görüntü Birleştirme Algılama teknikleri, kesik ve yapıştırılan bölgeler nedeniyle görüntüde mevcut olan, eklemeye duyarlı özellikleri ve görüntü desenlerindeki herhangi bir bozukluğu bulmayı amaçlar. Her algılama yöntemi, Şekil 1.14'da gösterildiği gibi geliştirilmiş bir yapıyı takip eder. Bu geliştirilmiş yapıda, Ön işleme, kategorize performansını artırmak için düzenleme, RGB resmini gri tonlamaya, DCT veya DWT değişikliğine çevirme gibi etkinlikleri yönetir. Özellik Ayıklama ve Özellik Seçimi, bir görüntünün bir dizi özelliğinden en bilgilendirici özelliğin çıkarılmasını içerir. Bu modeldeki sınıflandırıcı, mevcut farklı veri kümelerinin çeşitli görüntülerinin çıkarılan özelliklerinin yardımı ile bir görüntünün orijinal veya müdahale edilmiş olup olmadığını ayırt etmek için eğitilir. Görüntü piksel korelasyonunda ve birleştirme ile tetiklenen tutarlılıkta bir süreksizlik olarak görüntü uzunluk tasvirini ve keskin görüntü niteliklerini inceler. Görüntü çalıştırma uzunluğu tasviri ve özellikler gibi görüntü kenarı istatistikleri, içindeki eklemeyi tanımak için kullanılır (Deep Kaur, Kanwal, 2019). Şekil 1.14'de görüntü birleştirme tespitinin geliştirilmiş yapısının gösterimi verilmiştir.



Şekil 1.14. Görüntü birleştirme tespitinin geliştirilmiş yapısı

1.1.7.3.2 Görüntü Rötüşlama Algılama Teknikleri

Görüntü Rötüş Tespit yöntemleri, boyama sahteciliğini tespit etmeyi amaçlar. Kopyalanan bilgilerin sürekli olmadığı sahte bir türdür. Dijital fotoğraf düzenleme tekniklerinin yardımıyla bir görüntüye yapılan geometrik ve fotometrik değişiklikleri tahmin ederek fotoğrafları bir metrik (1-5) üzerinden derecelendirir. Minimum metrik değer, bir görüntüye yapılan en az rötüşlamayı temsil eder ve maksimum değer, büyük miktarda rötüşlamayı temsil eder. Geometrik değişiklikler 4 ölçümle hesaplanır: nesnenin gövdesi ve yüzü için bağımsız olarak hesaplanan hareket büyüklüğünün ortalama ve standart sapması ve fotometrik değişiklikler 4 ölçüm kullanılarak tahmin edilir: keskinleştirme / düzleştirme filtrelerinin uzamsal sınırlarının ortalama ve standart sapması ve yapısal Benzerlik Endeks Metriği (SSIM). İç boyama tespitinin yapıldığı bir yöntem önerilmiştir. Yamalardan ve üç genel parametre eşliğinden yararlanır. İlk adımda, benzerlik ölçüsü, mesafe ölçüsü ve kardinalite ölçüsünü kullanarak tüm benzer yama çiftlerini aranmaktadır. İkinci adımda, eşleşen yamaların lokalizasyonu ile kurcalama maskesi oluşturulur. Yanlış tespit edilen yamaları azaltmak için üçüncü adımda filtreleme şemasından yararlanmaktadır. Yüz, göz, göz bebeği, kaşlar, burun, dudak, çene lokalizasyonu gibi farklı özelliklerden yararlanarak insan güzelliği. Makyaj, saç kesimi, gözlüklerin varlığı gibi pek çok süresiz özellik, insanın güzellik konusundaki yargılarını da aynı şekilde etkiler. Bu derecelendirme, makine öğrenimi ve derin öğrenme

tekniklerini kullanan çeşitli yüz geometrik özellikleri ve doku özellikleri karışımlarıyla gerçekleştirilir. Etkili bir sahtecilik algılama algoritması, benzer karma değerlere sahip şüpheli blokları bulmak için bir hızlandırma yöntemi olan merkezi piksel eşlemeyi entegre etmektedir. En büyük sıfır bağlanabilirlik bileşen etiketleme şüpheli blok çiftlerinde ve parçalarda kurcalanmış pikselleri işaretler. Varsayılan blok çiftlerindeki değiştirilmiş pikselleri belirten ekleme algılama ve görüntü değişikliğini saptamak için değiştirilmiş bölgeleri en iyi eşleşen alanlarından tanıyan ve konumlandırılan parça ekleme saptama yöntemi kullanılmaktadır. Diğer bir teknik iki prosedür içermektedir. Şüpheli bölgeleri keşfetmek için benzerlik blokları üzerinden bakan şüpheli bir alan konumlandırma işlemi, yanlış pozitifleri azaltmak için karşılaştırılabilirlik vektörünü kullanır. Kurcalanmış alanları ayırt etmek için çok bölgeli gerçekleştirilmeden (MRR) faydalanan sahte bölge tanımlama işlemini kullanılmaktadır (Deep Kaur, Kanwal, 2019).

1.1.7.4 Derin Öğrenme Yöntemleri ile Görüntü Sahteciliği

GPU teknolojilerinin geliştirilmesi ve bilgisayarla görme alanında derin öğrenme tekniklerinin başarısı, görüntü sahteciliği tespiti için derin öğrenme modellerinin uygulanması popülerliğini her geçen gün artmıştır. Derin öğrenme, özellikleri (düşük / orta / yüksek seviye) çıkarma ve sınıflandırma aşamalarını birleştirir. Teknik, veriye dayalıdır ve kurcalanmış bölgeleri tanımlamak için gerekli olan soyut ve karmaşık özellikleri otomatik olarak öğrenebilir. Dahası, kurcalanmış görüntülerden el yapımı özellikleri bulmak için gereken zamandan ve enerjiden tasarruf sağlar. Bununla birlikte, derin öğrenme modellerinin eğitimi zordur ve yüksek hesaplama gücü ve çok büyük miktarda veri gerektirir. Evrişimli Sinir Ağları (CNN), Derin Sinir Ağı (DNN) ve Tekrarlayan Sinir Ağı (RNN) gibi birçok derin öğrenme modeli görüntü sahteciliğinde kullanılmaktadır. Evrişimli Sinir Ağları, bu derin öğrenme modelleri arasında popülerdir (Barad, Goswami, 2019). Çizelge 1.3'de görüntü sahteciliği tespiti için derin öğrenme yöntemlerinin karşılaştırması yer almaktadır.

Çizelge 1.3. Görüntü sahteciliği tespiti için derin öğrenme tabanlı yöntemlerin karşılaştırması (Barad, Goswarni, 2019)

Çalışma	Sahtecilik Türü	Model	Veri Kümesi	Sonuç
Bayar ve ark.	Gauss bulanıklığı, Medyan filtreleme, Yeniden örnekleme	CNN	12 farklı kamera modelinden toplanmıştır	Doğruluk % 99.10
Zhang ve ark.	Kes yapıştır, Kopyala-taşı	Derin oto-kodlayıcı	CASIA v1.0, CASIA v2.0, Columbia	Doğruluk % 91.09
Salloum ve ark.	Görüntü Birleştirme	Çok görevli tam evrişimli ağ (MFCN)	Columbia, CASIA v1.0, CASIA v2.0, Carvalho	0.54(CASIA v1.0) 0.61(Columbia)
Amerini ve ark.	Çift JPEG sıkıştırma, Kesme Yapıştır	Çok alanlı CNN	UCID (1338 görüntü)	Doğruluk % 95
Chen ve ark.	Medyan filtreleme, Kes yapıştır	CNN	15352 görüntü (NRCS Photo Gallery, BOSSbase 1.01, UCID, Dresden, BOSS RAW)	Doğruluk % 85.14

Çizelge 1.3. Görüntü sahteciliği tespiti için derin öğrenme tabanlı yöntemlerin karşılaştırması (devamı)

Bondi ve ark.	Kes yapıştır	CNN	Dresden	Tespit etme
			Veri tabanı	Doğruluk
			(26 farklı	% 81
Cozzolino ve Verdoliva	Kes yapıştır	Oto-kodlayıcı	6 akıllı telefon ve bir kameradan görüntüler	F-Measure 0,41 (temel) 0.37 (son işleme)
Rao and Ni	Kopyala-taşı, Kes yapıştır	CNN	CASIA v1.0, CASIA v2.0, Columbia gri DVMM	Doğruluk % 98.04 (CASIA v1.0) % 97,83 (CASIA
Wu ve ark.	Kopyala-taşı	BusterNet (Derin Sinir Ağ)	CASIA v2.0, CoMoFoD dataset	
Bi ve ark.	Kes yapıştır	Ringed Residual U-Net (RRU-Net)	CASIA, COLUMB	Doğruluk (Resim) % 76
Wang ve ark.	Kopyala-taşı, Kes yapıştır	Mask R-CNN	Cover, Columbia	% 93 (Cover) % 97 (Columbia)

1.3 Çalışmanın Amacı ve Kapsamı

Tez çalışmasının amacı bozulmuş görüntüler üzerinde fotomontaj tespiti yapılmış alanları belirlemek ve bu alanları görüntü üzerinde işaretlemeyi amaçlamaktadır. Bu doğrultuda derin evrişimsel ağlar kullanılmaktadır. Derin evrişimsel ağlar ile görüntü içerisinde yer alan nesnelere algılanıp tespit edilecek ve piksel bazında görüntü üzerinde işaretlenecektir. İşaretleme sonrasında işaretlenen alanların bozulmaya uğratılmış alanlar olup olmadığı tespiti yapılacaktır. Derin evrişimsel ağlar kullanılarak fotomontaj tespitinde daha büyük verilerde ve daha karmaşık görüntülerde bozulma tespitleri yapılabilecektir.



2. MATERYAL VE YÖNTEM

2.1. Tensorflow

TensorFlow, büyük verilerde ve homojen olmayan ortamlarda çalışan makine öğrenimi sistemidir. Tensor-Flow, hesaplamayı, paylaşılan durumu ve bu durumu değiştiren işlemleri temsil etmek için veri akışı grafiklerini kullanır. Bir veri akışı grafiğinin düğümlerini birçok makinede eşlemektedir. Küme ve bir makine içinde, çok çekirdekli Merkezi İşlem Birimi (CPU), genel amaçlı Grafik İşlem Birimi (GPU) ve Tensör İşleme Birimleri (TPU) olarak bilinen özel tasarımlı Uygulamaya Özel Tümlşik Devre (ASIC) dahil olmak üzere birden çok hesaplama cihazında tarafından kullanılmaktadır. Bu mimari, uygulama geliştiricisine esneklik sağlar: oysa önceki "parametre sunucusunda" paylaşılan durum yönetimini sistemde yerleşik olarak tasarlar. TensorFlow, uygulama geliştiricilerin yeni optimizasyonlar ve eğitim algoritmaları ile denemeler yapmasına olanak tanımaktadır. TensorFlow, derin sinir ağlarında eğitim ve çıkarım odaklı çeşitli uygulamaları destekler. Birkaç Google hizmeti, üretimde TensorFlow kullanır, açık kaynaklı bir proje olarak yayınlanmıştır ve makine öğrenmesi için yaygın olarak kullanılmaya başlanmıştır (Abadi vd., 2016).

Tensorflow çok farklı çalışma alanında kullanılmaktadır. TensorFlow resmi modelleri, TensorFlow'un üst düzey API'lerini kullanan bir model koleksiyonudur. Bakımlı, test edilmiş ve en son TensorFlow API ile güncel tutulmaları amaçlanmıştır. Bu modeller uçtan uca testler olarak kullanılır ve modellerin her yeni TensorFlow derlemesinde aynı veya geliştirilmiş hız ve performansla çalışmasını sağlamaktadır. Çizelge 2.1'de tensorflow modelleri hakkında bilgiler yer almaktadır.

Çizelge 2.1. Tensorflow modelleri hakkında bilgiler

Çalışma Alanı	Özelleştirilmiş Çalışma Alanı	Modelin Adı
Bilgisayarla Görü	Görüntü Sınıflandırma	ResNet
		ResNet-RS
		EfficientNet
		Vision Transformer

Çizelge 2.1. Tensorflow modelleri hakkında bilgiler (devamı)

Bilgisayarla Görü	Nesne Algılama ve Segmentasyon	RetinaNet
		Mask R-CNN
		SpineNet
		Cascade RCNN-RS ve RetinaNet-RS
	Video Sınıflandırması	Mobile Video Networks
Doğal Dil İşleme		ALBERT
	Önceden Eğitilmiş Dil Modeli	BERT
		ELECTRA
	Nörol Makine Çevirisi	Transformer
	Doğal Dil Üretimi	NHNet (News Headline generation)
	Bilgi Damıtma	MobileBERT
	Öneri	DLRM(Derin Öğrenme Öneri
		DCN v2 (Derin ve Çapraz Ağ - Deep & Cross Network)
	NCF(Nöral İşbirlikçi Filtreleme - Neural Collaborative Filtering)	

Tensorflow çok sayıda veri setine de sahiptir. Çizelge 2.2’de tensorflow da yer alan bazı veri setleri ve çalışma alanları yer almaktadır.

Çizelge 2.2. Tensorflow’da yer alan bazı veri setleri ve çalışma alanları

Çalışma Alanı						
Ses	Görüntü	Görüntü Sınıflandırma	Nesne Tanıma	Soru Cevaplama	Yapılandırılmış	Metin
Veri Seti Adı						
accentdb	arc	beans	coco	ai2_arc	covid19	glue
fuss	clevr	cifar10	kitti	coqa	dart	lambada
gtzan	dsprites	cmaterdb	open_images_challenge2019_detection	mlqa	iris	math_qa
librispeech	flic	imagenet2012	voc	squad	radon	race
nsynth	pass	mnist	wider_face	web_questions	titanic	wikipedia

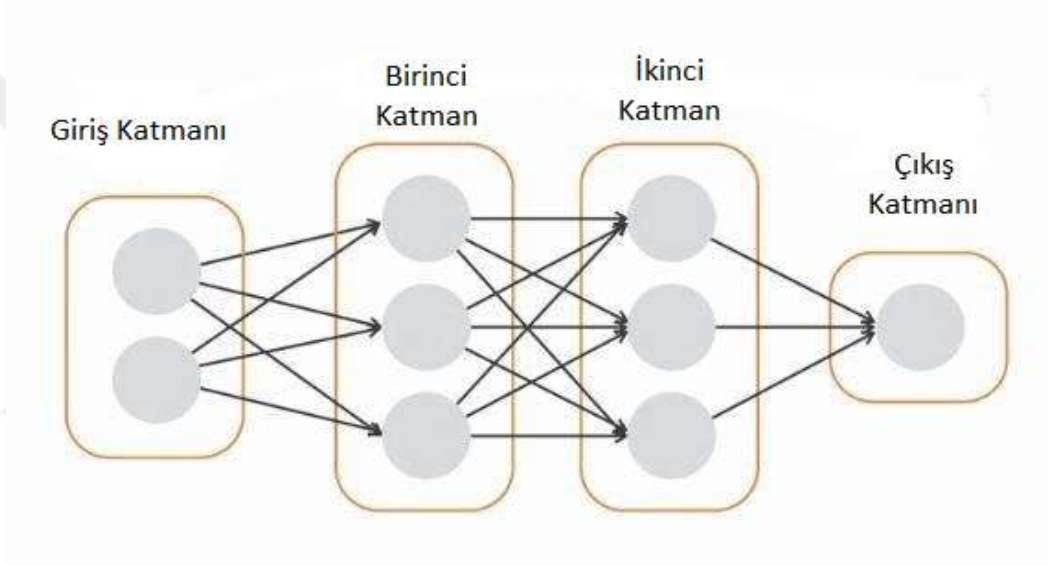
2.2 Keras

Keras, TensorFlow üzerinde çalışan bir derin öğrenme api'sidir. Yapılacak olan denemelerin daha hızlı olmasını mümkün kılmak amacıyla geliştirilmiştir. Keras, basit, esnek ve güçlü bir yapıya sahiptir. Açık Uçlu Nöro-Elektronik Akıllı Robot İşletim Sistemi (ONEIROS) araştırma çabasının bir parçası olarak geliştirilmiştir ve birincil yazarı ve bakımıcısı bir Google mühendisi olan François Chollet'tir. Keras dört temel özelliği birleştirmektedir. Bunlar:

- CPU, GPU veya TPU üzerinde gerçekleştirilecek olan düşük seviyeli tensör işlemlerini verimli bir şekilde ele almak.

- Keyfi türevlenebilir ifadelerin gradyanını hesaplama.
- Hesaplamayı çok sayıda GPU'dan oluşan kümeler halinde birçok makinede ölçeklendirme.
- Programları sunucular, tarayıcılar, mobil ve gömülü cihazlar gibi harici çalışma zamanlarına aktarma.

Keras'ın temel veri yapıları katmanlar ve modellerdir. En basit model türü, doğrusal bir katman yığını olan Sıralı modeldir. Daha karmaşık mimariler için, isteğe bağlı katman grafikleri oluşturmaya veya alt sınıflar aracılığıyla tamamen sıfırdan modeller yazmaya izin vermektedir. Şekil 2.1'de sıralı model gösterimi yer almaktadır.



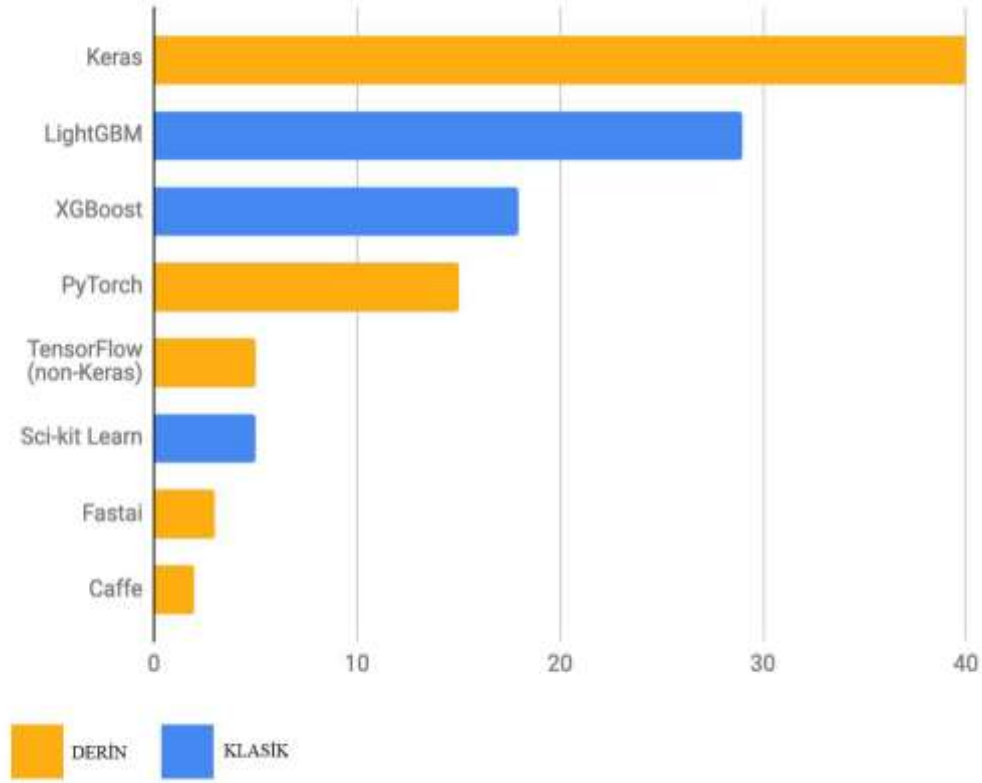
Şekil 2.1. Sıralı model gösterimi

Keras, Yunanca boynuz anlamına gelmektedir. İlk olarak Odyssey'de bulunan, rüya ruhlarının hayalperestleri yanlış vizyonlarla aldatanlar, fildişi bir kapıdan Dünya'ya gelenler arasında bölüdüğü antik Yunan ve Latin edebiyatından edebi bir imgeye göndermedir ve gerçekleşecek bir geleceği ilan edenler, bir boynuz kapısından gelenler anlamını taşımaktadır.

Çizelge 2.3. Keras api örnekleri ve özelliği

Özelliği	Adı
Model API	Model Sınıfı
	Sıralı Sınıf
Katman API	Evrışim Katmanı
	Normalizasyon Katmanı
Callbacks API	TensorBooard
	CSVLogger
Optimizasyon	SGD
	Adam
Metrikler	Doğruluk metrikleri
	Regresyon metrikleri
Küçük Veri Setleri	MNIST sayı sınıflandırma veri seti
	CIFAR100 küçük görüntü sınıflandırma veri seti
Keras Uygulamaları	Xception
	ResNet ve ResNetV2

2019'un başlarında, önceki iki yılda herhangi bir Kaggle yarışmasında ilk 5'te yer alan takımlar arasında bir anket yapılmıştır. Takımların ilk 5'e girdikleri yarışmada kullandıkları birincil makine öğrenmesi yapısı istenmiştir. Şekil 2.2'de anket sonuçları yer almaktadır.



Şekil 2.2. Takımların birincil kullandığı makine öğrenmesi yapısı

2.3 Numpy

NumPy, Python ile sayısal hesaplamayı etkinleştirmeyi amaçlayan açık kaynaklı bir projedir. 2005 yılında, Numeric ve Numarray kitaplıklarının ilk çalışmaları üzerine inşa edilmiştir. NumPy her zaman %100 açık kaynaklı yazılım olacaktır, herkesin kullanması ücretsiz olacaktır. NumPy, NumPy ve daha geniş bilimsel Python topluluğunun fikir birliği yoluyla GitHub'da açık olarak geliştirilmiştir (Numpy, 2022).

Dizi programlama, vektörler, matrisler ve yüksek boyutlu dizilerdeki verilere erişmek, bunları işlemek ve bunlar üzerinde çalışmak için güçlü, kompakt ve anlamlı bir sözdizimi sağlamaktadır. NumPy, Python dili için birincil dizi programlama kitaplığıdır. Fizik, kimya, astronomi, yerbilimi, biyoloji, psikoloji, malzeme bilimi, mühendislik, finans ve ekonomi gibi çeşitli alanlarda araştırma analizi boru hatlarında önemli bir role sahiptir. Örneğin, astronomide

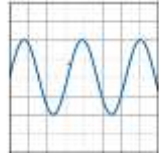
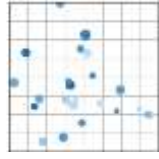
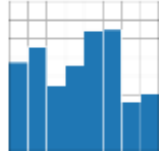
NumPy, yerçekimi dalgalarının keşfinde ve bir kara deliğin ilk görüntülenmesinde kullanılan yazılım yığınının önemli bir parçasıydı. NumPy, bilimsel Python üzerine inşa edilmiştir. O kadar yaygındır ki, özel ihtiyaçları olan kitleleri hedefleyen birçok proje kendi NumPy benzeri ara yüzlerini ve dizi nesnelerini geliştirmiştir. Ekosistemdeki merkezi konumu sayesinde NumPy, bu tür dizi hesaplama kitaplıkları arasında giderek artan bir şekilde birlikte çalışabilirlik katmanı görevi görür ve API ile birlikte bilimsel ve endüstriyel analizleri desteklemek için esnek bir çerçeve sağlamaktadır (Harris vd., 2020).

2.4 Matplotlib




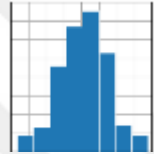
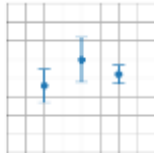

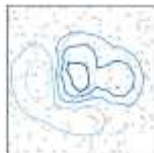
Matplotlib, uygulama geliştirme, etkileşimli komut dosyası oluşturma ve kullanıcı arayüzleri ve işletim sistemleri arasında yayın kalitesinde görüntü oluşturma için Python için kullanılan bir 2B (2 Boyutlu) grafik paketidir (Hunter, 2007).

Matplotlib, farklı çizim tipleri içermektedir. Bunların bazılarının isimleri ve örnekleri Çizelge 2.4'de yer almaktadır.

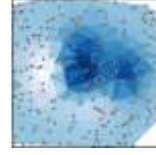
Çizelge 2.4. Çizim tipleri, basit kod yapısı ve örnek gösterimi (Matplotlib, 2022).

Çizim Tipi	Çizim Kodu	Örnek Gösterim
Basit	<code>plot(x,y)</code>	
	<code>scatter(x,y)</code>	
	<code>bar(x, height /barh(y, width)</code>	

Çizelge 2.4. Çizim tipleri, basit kod yapısı ve örnek gösterimi (devamı)

	<code>imshow(Z)</code>	
Array ve Alan	<code>contour(X, Y, Z)</code>	
	<code>barbs(X, Y, U, V)</code>	
	<code>hist(x)</code>	
İstatistik	<code>errorbar(x, y, yerr, xerr)</code>	
	<code>pie(x)</code>	
Yapılandırılmamış Koordinatlar	<code>tricontour(x, y, z,)</code>	

Yapılandırılmamış tripcolor(x, y, z)
Koordinatlar



2.5 Anaconda

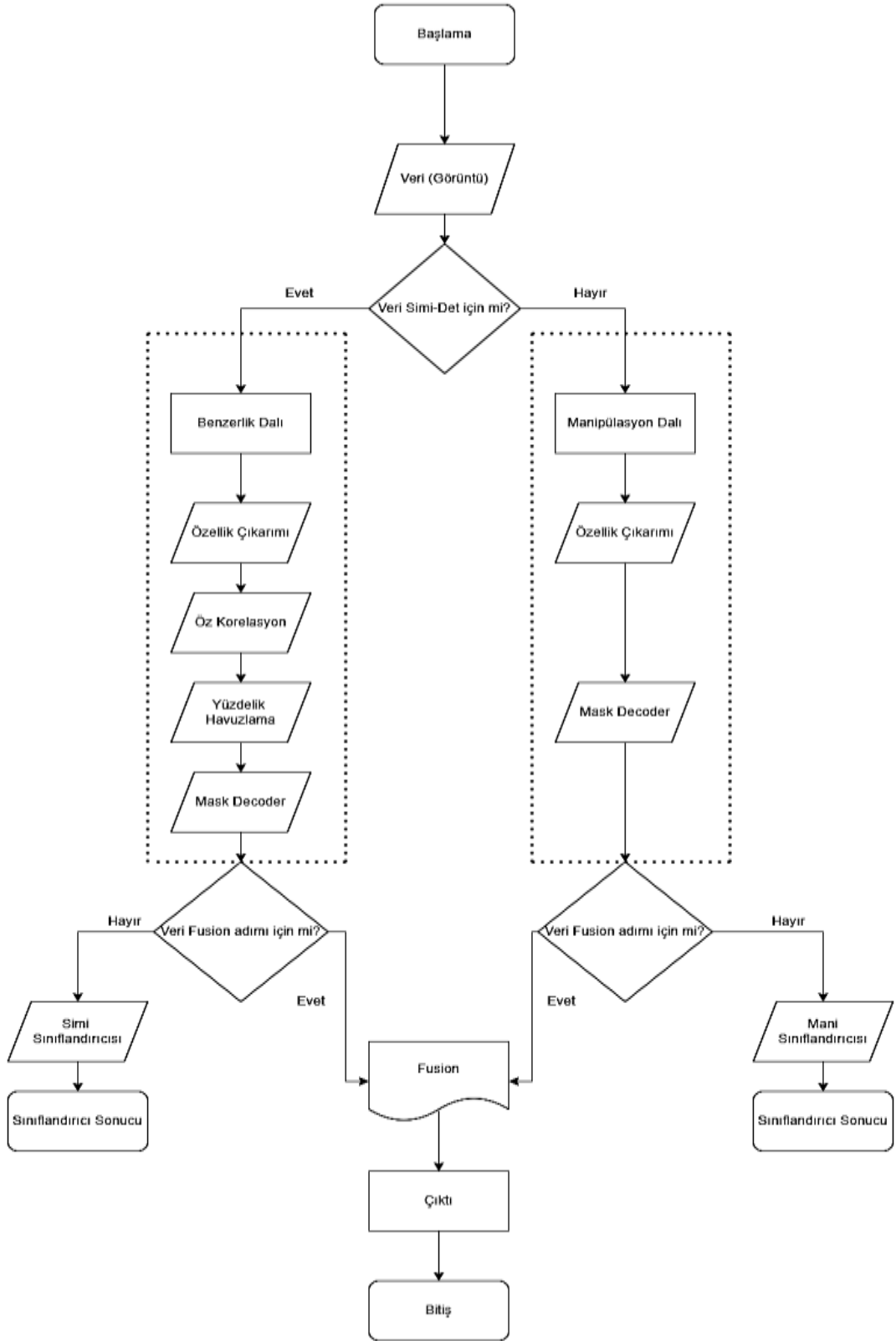
Python, son yıllarda araştırma için en popüler programlama dillerinden biri haline gelmiştir. Ücretsiz, açık kaynaklı doğası ve geniş çevrimiçi topluluğu, başarısının arkasındaki nedenlerden bir kaçıdır. Python sayesinde artan araştırma üretkenliğinin sayısız örneği, veri bilimi, yapay zekâ ve bilimsel araştırma dahil olmak üzere çok sayıda çevrimiçi alanda bulunmaktadır. Anaconda yazılımının amacı kullanıcıları ilk betiklerini yazarak Python'u kullanmaya başlamasına yardımcı olmaktır.

Anaconda, araştırma ve bilim için özel olarak hazırlanmış bir araç takımı sağlayan ücretsiz bir yazılımdır. Anaconda'yı yüklemek, Python veya R'de kod yazmanıza izin veren farklı ortamlara erişimi sağlamaktadır. Entegre geliştirme ortamları (IDE) olarak da bilinen bu ortamlar, kod geliştirmeyi büyük ölçüde kolaylaştıran platformlar veya uygulamalardır. Metin yazmak için Microsoft Word ve Google Doc gibi metin işlemcilerine benzer bir rol üstlenmektedir, ancak gerçekte çok daha fazlasıdır. IDE'ler, kod yazmak, düzenlemek ve hata ayıklamak, verileri görselleştirmek ve incelemek, değişkenleri depolamak, sonuçları sunmak ve projeler üzerinde işbirliği yapmak için birçok kullanışlı özellik içermektedir (Rolon-Mérette, Ross, Rolon-Mérette, Church, 2016).

Bir terminal penceresine komut yazmanıza gerek kalmadan farklı IDE'lere kolayca erişip başlatabileceğiniz Anaconda'nın "menü" sayfasıdır. Python programlama için özel olarak tasarlanmış birçok farklı IDE bulunmaktadır ve Anaconda ile otomatik olarak yüklenmektedir.

2.6 Akış Diagramı

Şekil 2.3’de akış diagramı yer almaktadır.



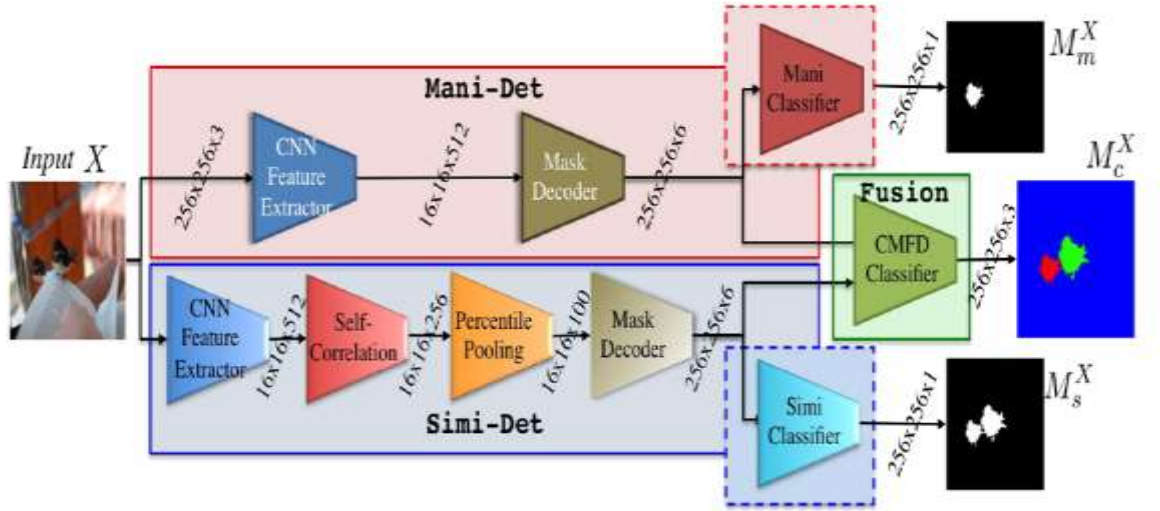
Şekil 2.3. Akış diagramı

3. UYGULANAN MODEL VE BİLGİLERİ

Görüntü sahteciliği tespiti alanında birçok yöntem geliştirilmiştir ve kullanılmıştır. Bu tez çalışmasında derin öğrenme tabanlı yöntem kullanılarak görüntü sahteciliği tespiti yapılmıştır.

3.1 Model Bilgileri

Görüntü sahteciliğinde derin öğrenme tabanlı mimari tasarlamak için, elle ayarlanmış parametreleri ve/veya karar kurallarını içermeyen, farklı kaynak ve hedef manipülasyon maskeleri üretebilen uçtan uca eğitilebilir bir derin sinir ağı hattı tasarlanması gerekmektedir. BusterNet, yukarıda bahsi geçen hedeflere ulaşmak için, geçerli bir derin sinir ağı çözümü olarak iki özelliği aynı anda elde etmektedir. Kaynak ve hedef özelliklerini, kaynağı hedeften ayırt edecek kadar farklı olması ile bozulmamış bölgelerdekilerden daha benzer özelliklerin elde edilmesidir. Şekil 3.1'de BusterNet mimarisinin hattı gösterilmektedir. (Wu, Abd-Almageed, Natarajan, 2018).



Şekil 3.1. BusterNet mimarisi

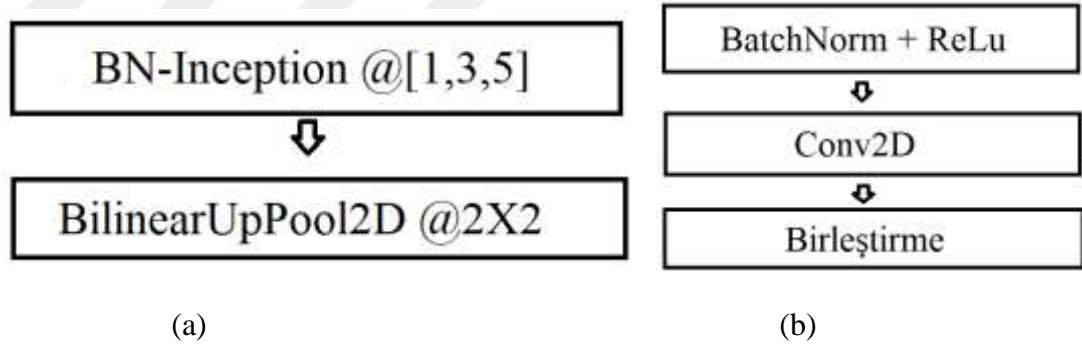
Mani-Det (Manipülasyon Algılama) dalını manipüle edilmiş bölgeleri algılamak için tasarlarlarken, Simi-Det (Benzerlik Algılama) dalı ise klonlanmış bölgeleri algılamak ve son olarak Fusion'da her iki özelliği kullanmak için tasarlanmıştır. Bozulmamış, kaynak kopya ve hedef kopya sınıflarını farklılaştıran ise piksel düzeyinde kopyala-taşı maskelerini tahmin etmektir. Bu iki dalın istenen işlevselliği elde etmesini sağlamak için, Şekil 3.1'de noktalı bloklarla gösterildiği gibi her bir dalı bir yardımcı görev tanımlanmaktadır. Mani-Det dalının ve Simi-Det dalının görevleri, bir ikili manipülasyon maskesi tahmin etmektir. Görüntüler

256×256×3 boyutundadır. Ancak BusterNet diğer boyutlardaki görüntüleri işleyebilmektedir. (Wu, vd., 2018).

3.1.1 Manipülasyon Algılama Dalı

Mani-Det, amacı manipüle edilen bölgeleri segmentlere ayırmak olan özel bir segmentasyon ağı olarak belirtilmiştir. Daha doğrusu, girdi görüntüsü X'i almaktadır, CNN özellik çıkarıcı kullanarak öznitelikleri çıkarmaktadır, Mask Decoder kullanarak öznitelik haritalarını orijinal görüntü boyutuna yükseltmektedir ve yardımcı görevi yerine getirmek için Binary Classifier'ı uygulamaktadır ve sonuç olarak bir manipülasyon maskesi üretmektedir (Wu, vd., 2018).

Herhangi bir CNN, CNN Özellik Çıkarıcı olarak hizmet verebilmektedir. VGG16 mimarisinin ilk dört bloğunu kullanılmaktadır. Ortaya çıkan CNN özelliği, manipülasyon maskesinin çözünürlüğü gerektirdiğinden çok daha düşük olan 16×16×512 boyutundadır. Bu nedenle, bu özelliğin kodunu çözmemiz ve orijinal çözünürlüğü, Şekil 3.2'te gösterildiği gibi, BN-Inception ve BilinearUpPool2D'yi alternatif bir şekilde uygulayan ve sonunda bir tensör üreten Maske Kod Çözücü aracılığıyla geri yüklemek için kullanılmaktadır (Wu, vd., 2018).



Şekil 3.2. (a) Maske kod çözücü ağ gösterimi

(b) BN-Inception modülü parametreleri

3.1.2 Benzerlik Algılama Dalı

Simi-Det, bir X girdi görüntüsü almaktadır, CNN özellik çıkarıcı kullanarak öznitelikleri çıkarmaktadır. Öz Korelasyon modülü aracılığıyla öznitelik benzerliğini hesaplamaktadır. Yüzdelerle Havuzlama yoluyla faydalı istatistikler toplar, üst örnekler özelliği, Mask decoder kullanarak orijinal görüntü boyutuna eşlemektedir. Yardımcı görevi yerine getirmek için Binary Classifier'ı uygular, yani aynı X çözünürlüğünde bir kopyala-taşı maskesi

üretmektedir. CNN özellik çıkarıcı, yalnızca ağ mimarisini paylaşır, ağırlıkları paylaşmamaktadır.

Mani-Det dalı gibi, Simi-Det dalı da CNN özellik çıkarıcı aracılığıyla özellik gösterimi ile başlamaktadır. Yine $16 \times 16 \times 512$ boyutunda bir özellik tensörü üretmektedir, bu aynı zamanda 16×16 yama benzeri özellikler olarak da ele alınmaktadır. Amaç, potansiyel kopyala-taşı bölgelerini kurtarmak olduğundan, hangi yama benzeri özelliklerin eşleştiğine karar vermek için faydalı bilgiler çıkarmamız gerekmektedir. Bunu yapmak için, önce Öz Korelasyon kullanarak tüm özellik benzerlik puanını hesaplanmaktadır ve Yüzdeler Havuzlama yoluyla eşleşen yamaları belirlenmektedir (Wu, vd., 2018).

3.1.3 BusterNet Fusion

Şekil 3.1'de gösterildiği gibi, Fusion modülü, Maske kod çözücü özelliklerinin girdilerini her iki daldan almaktadır ve bu iki dalı birlikte ele alır ve nihai görüntü sahteciliği tahminini yapmaktadır. Özniteliğini birleştirmektedir, parametre seti ile BN-Inception kullanarak 3×3 çekirdek boyutunda bir filtre ve ardından softmax aktivasyonu ile bir Conv2D kullanarak tahmin yapmaktadır (Wu, vd., 2018).

3.2 Model Eğitim Bilgileri

3.2.1 CASIA Veri Seti

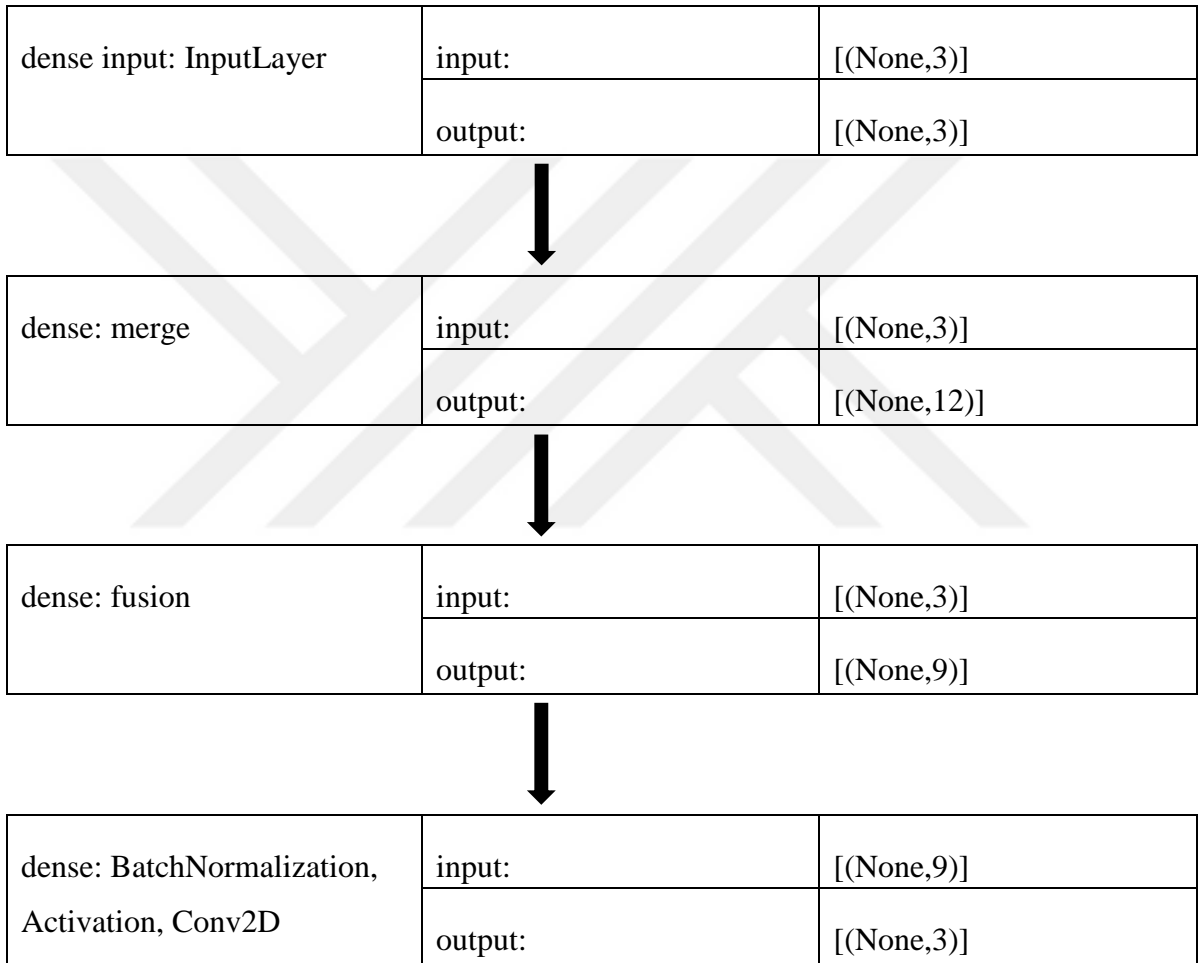
Modelin değerlendirilmesi için iki standart veri seti kullanılmaktadır. İlk veri seti, tüm manipülasyonların manuel olarak oluşturulduğu, kamuya açık en büyük görüntü sahteciliği tespit kıyaslaması olan CASIA TIDEv2.0 veri setidir (Casia, 2022). 7491 orijinal ve 5123 tahrif edilmiş renkli görüntü içermektedir. Ancak, hangi görüntülerin kopyala-taşı şeklinde manipüle edildiğini belirtilmemektedir. Bu nedenle, tahrif edilmiş 5123 numuneden 1313'ünün kopyala-taşı sahtekarlığı olduğunu doğrulanmıştır. Bu 1313 kopyala-taşı sahtekarlığı numunesi ve bunların orijinal karşılıkları ile birlikte test veri seti toplam 2626 numune oluşmaktadır.

3.2.2 CoMoFoD Veri Seti

İkinci veri seti, 200 temel sahte görüntü ve 25 kategoride toplam 5000 görüntü içeren CoMoFoD veri setidir. Her kategori, sahte ipuçlarını gizlemek için temel kategori görüntülerine işlem uygulanarak yapılmaktadır (Tralic, vd., 2013).

3.3 Model Özeti

Gerçekleştirilen modelin özet bilgileri Şekil 3.3’de yer almaktadır.



Şekil 3.3. Model özeti

3.4 Uygulama Özet Kod Gösterimleri

Modeli gerçekleştirirken kullanılan kod parçacıkları özet halinde aşağıda verilmiştir.

- Kullanılacak olan kütüphanelerin çağırılması

```
import keras
import tensorflow as tf
import numpy as np
from matplotlib import pyplot
```

- VGG16 blok kodlamasını örnekleri. Bu blok kodlamaları Simi- Det ve Mani- Det için tekrar yapılmıştır.

```
x1 = Conv2D(64, (3, 3), activation='relu', padding='same',)
x1 = MaxPooling2D((2, 2), strides=(2, 2),)
```

- HDF5MATRIX kullanımı (Veri seti temsili için kullanılmıştır.)

```
from keras.utils.io_utils import HDF5Matrix
X = HDF5Matrix('CASIA-CMFD-Pos.hd5', 'X' )
Y = HDF5Matrix('CASIA-CMFD-Pos.hd5', 'Y' )
```

- Ağ yapısının oluşturulması. Bu yapı Simi- Det ve Mani- Det için tekrar yapılmıştır.

```
f64a = BnInception()
f64b = BilinearUpSampling2D()
f64 = Concatenate()([f64a, f64b])
dx64 = BnInception
```

- Görselleştirme adımında görüntülerin array olarak alınması ve tipinin düzenlenmesi.

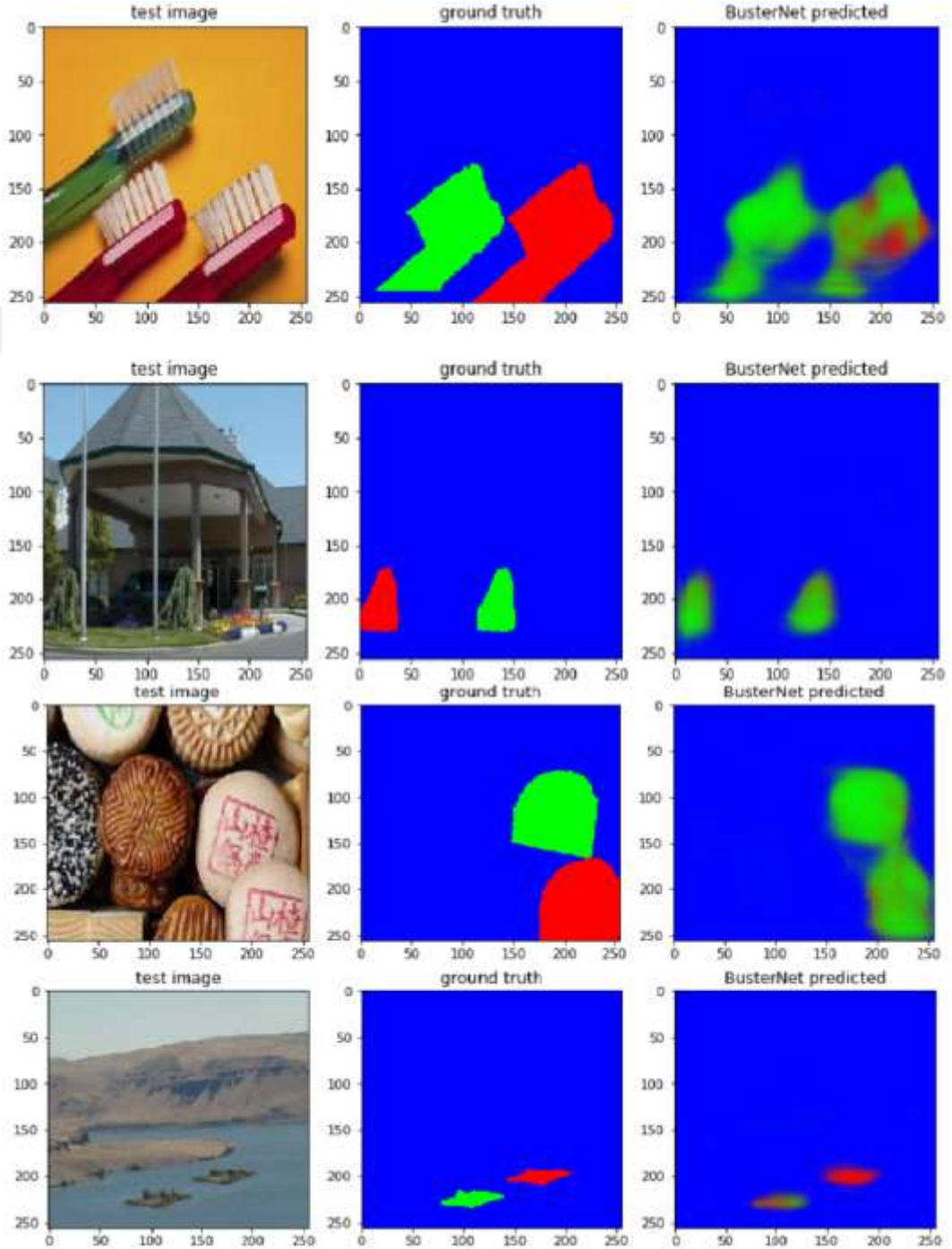
```
x = np.array() + np.array([]).reshape([])
x = np.round(x).astype('uint8')[]
```

- Rastgele seçilen örneklerin gösterimi

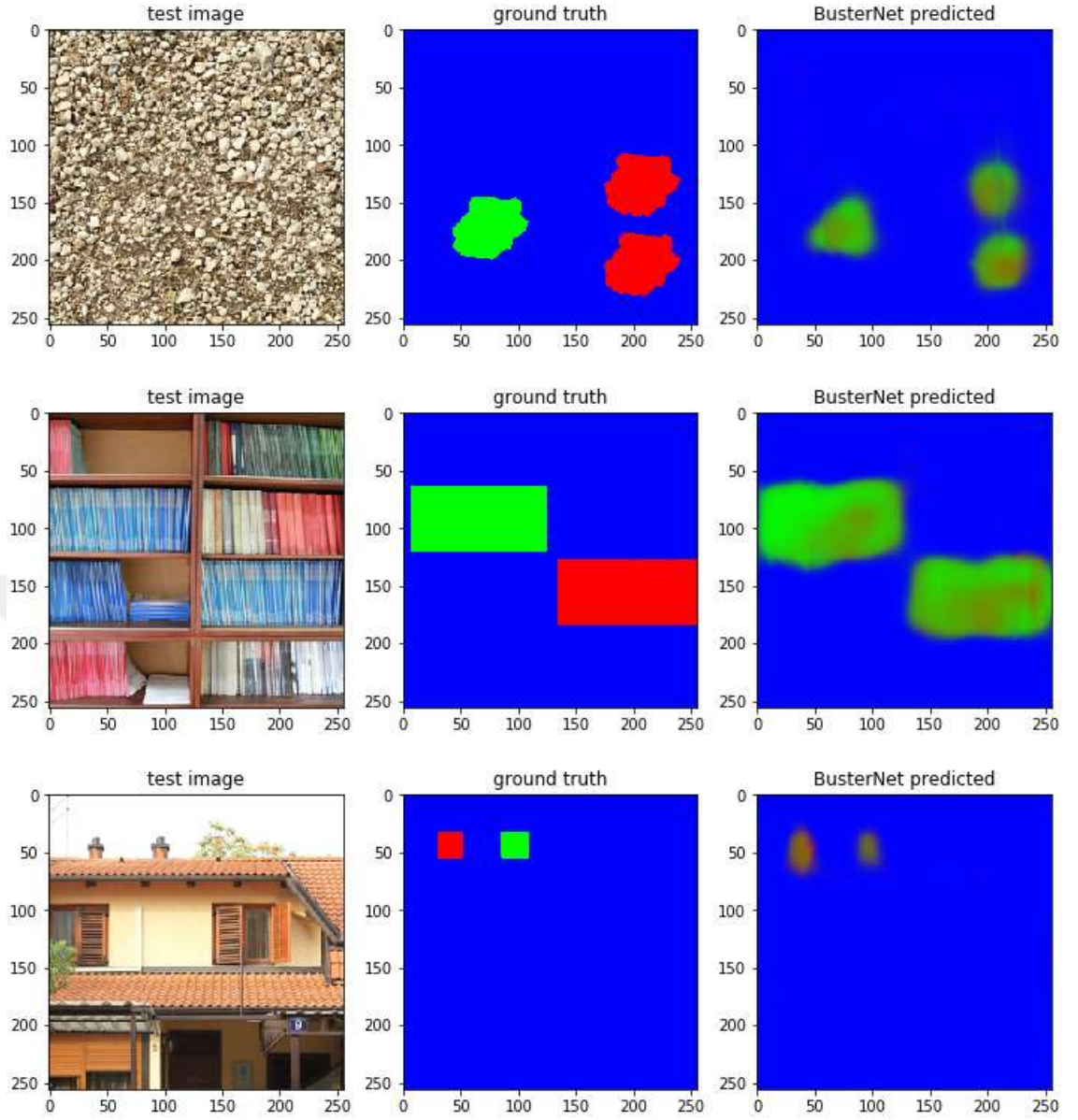
```
visualize_random_samples( X, Y, Z,)
```

3.5 Uygulama Çıktıları

Yapılan çalışma, çeşitli yöntemler ile gerçekleştirilmiş olan görüntü sahteciliği örneklerinin yer aldığı veri setleri üzerinde gerçekleştirilmiş olup başarılı sonuçlar elde edilmiştir. Şekil 3.4 ve şekil 3.5’de elde edilen çıktılar gösterilmiştir.



Şekil 3.4. CASIA veri seti çıktıları



Şekil 3.5. CoMoFoD veri seti çıktıları

4. SONUÇ VE ÖNERİLER

Çizelge 4.1’de BusterNet mimarisinin CASIA ve CoMoFoD veri setlerindeki doğruluk oranları verilmiştir.

Çizelge 4.1. BusterNet doğruluk oranları

Veri seri	Görüntü Sayısı	Doğruluk Oranı
CASIA	1313	74.52%
CoMoFoD	200	79.21%
Toplam	1513	76.86%

Çalışmada oluşturulan model ile kaynak ve hedef görüntülerin belirlenmesi gerçekleştirilmiştir. Görüntü üzerindeki değişiklik yapılmış alan belirlenmiştir. Model ve kullanılan veri setleri ile alınan sonuçlara göre CoMoFoD veri setinden daha yüksek başarı oranı elde edilmiştir. İki veri setinden alınan sonuçlar %76.86 olarak belirlenmiştir.

Görüntülerde yapılan sahteciliğin tespiti, kötü niyetli yapılmış olan ve istenmeyen sonuçlar ile karşılaşılacak durumların önüne geçmekte yardımcı olmaktadır. Bu sebeple kişiler güvenmediği kişilere görüntü göndermemelidir ya da güvenmediği kişiler tarafından gösterilen görüntülere hemen inanıp istenmeyen olayların yaşanmasına engel olmalıdır.

Görüntü sahteciliğinde, kişiler sahteciliği görüntü üzerinde çalışma yapmadan anlaması zor olsa da bazı noktalar da şüpheye düşebilirler ve görüntüyü sorgulayabilirler. Aşağıda bazı görüntü özellikleri ile sahtecilikten şüphelenilebilir hatta sahte olduğu kararını verebilirler:

- Görüntüde tekrar eden unsurlar
- Görüntünün belli noktalarında meydana gelen renk geçişleri ve değişimi
- Görüntüdeki yapay unsurlar
- Görüntü üzerinde resmin bütünlüğüne bozan eklemeler
- Görüntüde yer alan nesnelerin orantısızlığı

Teknoloji ilerledikçe görüntülerdeki sahteciliğin insan gözü ile ayırt edebilmesi daha zor olacağından mevcut sahtecilik tespiti yöntemlerinin geliştirilmesine ihtiyaç duyulacaktır. Yöntemler sadece görüntü üzerinde değil video ve canlı yayın akışlarında da uygulanarak

önerilen yöntemlerin eksikleri belirlenip, bu eksikler üzerinde çalışma yapılmasına olanak sağlayabilir.

Mevcut veri setlerinin yanında görüntü sahteciliği için özel oluşturulmuş ve düzenlenmiş veri seti oluşturulabilir. Oluşturulan veri seti ile yöntemler test edilebilir ve yetersiz kaldığı noktalar belirlenebilir.

Ayrıca elde edilen görüntülerin kaynakları doğrulanmalı ve kaynağından emin olunmayan görüntülere güvenilmemelidir.

Sonraki çalışmalarda farklı yöntemler ve farklı veri seti ile iyileştirme yapılabilir. Klasik görüntü işleme yöntemleri ile derin öğrenme metotları birleştirilerek doğrulama oranları daha yüksek seviyelere çekilebilir.

KAYNAKLAR

- Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology*, 33(8), 831-838.
- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... & Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*.
- Arrowsmith, J., & Miller, P. (2013). Trial watch: phase II and phase III attrition rates 2011-2012. *Nature reviews. Drug discovery*, 12(8), 569.
- Arrowsmith, J., & Miller, P. (2013). Trial watch: phase II and phase III attrition rates 2011-2012. *Nature reviews. Drug discovery*, 12(8), 569.
- B. Onyshkevych, "Deep Exploration and Filtering of Text (DEFT)." [Online]. Erişim adresi: <http://www.darpa.mil/program/deep-exploration-andfiltering-of-text>. Erişim tarihi: Mayıs 2022
- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Barad, Z. J., & Goswami, M. M. (2020, March). Image forgery detection using deep learning: A survey. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 571-576). IEEE.
- Bengio, Y. (2009). *Learning deep architectures for AI*. Now Publishers Inc.
- Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. *Advances in Neural Information Processing Systems*, 13.
- Canchumuni, S. W., Emerick, A. A., & Pacheco, M. A. C. (2019). Towards a robust parameterization for conditioning facies models using deep variational autoencoders and ensemble smoother. *Computers & Geosciences*, 128, 87-102.
- Casia, (t.y.). Erişim adresi http://forensics.idealtest.org/casiav2#/ Erişim tarihi: Nisan 2022
- Chicco, D., Sadowski, P., & Baldi, P. (2014, September). Deep autoencoder neural networks for gene ontology annotation predictions. In *Proceedings of the 5th ACM conference on bioinformatics, computational biology, and health informatics* (pp. 533-540).
- Ciregan, D., Meier, U., & Schmidhuber, J. (2012, June). Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 3642-3649). IEEE.
- Ciresan, D., Giusti, A., Gambardella, L., & Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in neural information processing systems*, 25.

- Cireşan, D. C., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2013, September). Mitosis detection in breast cancer histology images with deep neural networks. In *International conference on medical image computing and computer-assisted intervention* (pp. 411-418). Springer, Berlin, Heidelberg.
- Cireşan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2012). Deep big multilayer perceptrons for digit recognition. In *Neural networks: tricks of the trade* (pp. 581-598). Springer, Berlin, Heidelberg.
- Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning* (pp. 160-167).
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.
- Deep Kaur, C., & Kanwal, N. (2019). An analysis of image forgery detection techniques. *Statistics, Optimization & Information Computing*, 7(2), 486-500.
- Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends in signal processing*, 7(3-4), 197-387.
- Derin Öğrenme Karşılaştırma, (t.y.). Erişim adresi https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software Erişim tarihi: Nisan 2022
- Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2), 179-211.
- Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2012). Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1915-1929.
- Gers, F. A., & Schmidhuber, J. (2000, July). Recurrent nets that time and count. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium* (Vol. 3, pp. 189-194). IEEE.
- Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10), 2451-2471.
- Girişim, (t.y.). Erişim adresi <https://asgard.vc/wp-content/uploads/2018/05/Artificial-Intelligence-Strategy-for-Europe-2018.pdf> Erişim tarihi: Mayıs 2022
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.
- Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362.

- Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., ... & Larochelle, H. (2017). Brain tumor segmentation with deep neural networks. *Medical image analysis*, 35, 18-31.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Hinton, G. E. (2007). Learning multiple layers of representation. *Trends in cognitive sciences*, 11(10), 428-434.
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786), 504-507.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- Hopfield Ağı. (2022). 12, Mayıs 2022, tarihinde Wikipediadan alındı https://en.wikipedia.org/wiki/Hopfield_network
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9(03), 90-95.
- Ivakhnenko, A. G., & Lapa, V. G. (1966). *Cybernetic predicting devices*. PURDUE UNIV LAFAYETTE IND SCHOOL OF ELECTRICAL ENGINEERING.
- Jarusek, R., Volna, E., & Kotyrba, M. (2019). Photomontage detection using steganography technique based on a neural network. *Neural Networks*, 116, 150-165.
- Kayaalp, K., Süzen, A. A., “DERİN ÖĞRENME.” DERİN ÖĞRENME VE TÜRKİYE’DEKİ UYGULAMALARI, Adıyaman, Türkiye: İKSAD Yayınevi, 2018, böl. 2, ss. 25-28.
- Keras. (t.y.). Erişim adresi <https://keras.io/about/> Erişim tarihi: Mayıs 2022
- Kim, D. K., & Chen, T. (2015). Deep neural network for real-time autonomous indoor navigation. *arXiv preprint arXiv:1511.04668*.
- Krizhevsky, A., & Hinton, G. E. (2011, April). Using very deep autoencoders for content-based image retrieval. In *ESANN* (Vol. 1, p. 2).
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Le, Q. V. (2013, May). Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 8595-8598). IEEE.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.

- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4), 541-551.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Matplotlib, (t.y.). Erişim adresi <https://matplotlib.org/> Erişim tarihi: Mayıs 2022
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540), 529-533.
- Niu, X., Zhu, Y., & Zhang, X. (2014, December). DeepSense: A novel learning mechanism for traffic prediction with taxi GPS traces. In *2014 IEEE global communications conference* (pp. 2745-2750). IEEE.
- Numpy, (t.y.). Erişim adresi <https://numpy.org/> Erişim tarihi: Mayıs 2022
- Rolon-Mérette, D., Ross, M., Rolon-Mérette, T., & Church, K. (2016). Introduction to Anaconda and Python: Installation and setup. *Python for research in psychology*, 16(5), S5-S11.
- Salakhutdinov, R., & Hinton, G. (2009). International Conference on Artificial Intelligence and Statistics.
- Shen, D., Wu, G., & Suk, H. I. (2017). Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19, 221-248.
- Smolensky, P. (1986). *Information processing in dynamical systems: Foundations of harmony theory*. Colorado Univ at Boulder Dept of Computer Science.
- Sun, W., Zheng, B., & Qian, W. (2017). Automatic feature learning using multichannel ROI based on deep structured algorithms for computerized lung cancer diagnosis. *Computers in biology and medicine*, 89, 530-539.
- Sutskever, I., Martens, J., & Hinton, G. E. (2011, January). Generating text with recurrent neural networks. In *ICML*.
- Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

- Şahin, A., Buluş, E., Sakallı, M. T., & Buluş, H. N. (2007). Resim içerisindeki gizli bilginin RQP steganaliz yöntemiyle sezilmesi. *Akademik Bilişim 2007*.
- Şeker, A., & YÜKSEK, A. G. (2017). Stacked autoencoder method for fabric defect detection. *Cumhuriyet Üniversitesi Fen Edebiyat Fakültesi Fen Bilimleri Dergisi*, 38(2), 342-354.
- Şeker, A., Diri, B., & Balık, H. H. (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. *Gazi Mühendislik Bilimleri Dergisi (GMBD)*, 3(3), 47-64.
- Tkachenko, Y. (2015). Autonomous CRM control via CLV approximation with deep reinforcement learning in discrete and continuous action space. *arXiv preprint arXiv:1504.01840*.
- Tralic, D., Zupancic, I., Grgic, S., & Grgic, M. (2013, September). CoMoFoD—New database for copy-move forgery detection. In *Proceedings ELMAR-2013* (pp. 49-54). IEEE.
- Urban, G., Bendszus, M., Hamprecht, F., & Kleesiek, J. (2014). Multi-modal brain tumor segmentation using deep convolutional neural networks. *MICCAI BraTS (brain tumor segmentation) challenge. Proceedings, winning contribution*, 31-35.
- Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156-3164).
- Wu, Y., Abd-Almageed, W., & Natarajan, P. (2018). Busternet: Detecting copy-move image forgery with source/target localization. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 168-184).
- Y. LeCun, C. Cortes, and C. Burges, “MNIST handwritten digit database.” [Online].
- Yapay Zekâ. (t.y.). Erişim adresi <https://www.cbinsights.com/research/top-acquirers-ai-startups-ma-timeline/> Erişim tarihi: Mayıs 2022
- Zhang, Q., Xiao, Y., Dai, W., Suo, J., Wang, C., Shi, J., & Zheng, H. (2016). Deep learning based classification of breast tumors with shear-wave elastography. *Ultrasonics*, 72, 150-157.