



BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

**T.C.
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANA BİLİM DALI**

**X-RAY GÖRÜNTÜLERİNDEN COVID-19 TESPİTİ İÇİN DERİN
ÖĞRENME TABANLI MODELLERİN GELİŞTİRİLMESİ**

GİZEM ÖTER

YÜKSEK LİSANS TEZİ

DANIŞMAN

DOÇ.DR. EMRAH HANÇER

BURDUR – 2023

**T.C.
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANA BİLİM DALI**

**X-RAY GÖRÜNTÜLERİNDEN COVID-19 TESPİTİ İÇİN DERİN
ÖĞRENME TABANLI MODELLERİN GELİŞTİRİLMESİ**

GİZEM ÖTER

YÜKSEK LİSANS TEZİ

DANIŞMAN: DOÇ.DR. EMRAH HANÇER

JURİ ÜYESİ: DR. ÖĞR. ÜYESİ MELİKE ŞİŞECİ ÇEŞMELİ

JURİ ÜYESİ: DR.ÖĞR. ÜYESİ TAYYİP ÖZCAN

T.C.
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ

ETİK BEYAN

Burdur Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği'ne göre hazırlamış olduğum “X-ray Görüntülerinden Covid-19 Tespiti İçin Derin Öğrenme Tabanlı Modellerin Geliştirilmesi” adlı tezin hazırlanması sürecinde akademik ve etik kuralları ihlal etmediğimi, tezimin özgün olduğunu, başvurduğum kaynakları metin içinde, dipnotlarda ve kaynakçada eksiksiz ve bilimsel kurallara uygun olarak gösterdiğimi taahhüt ederim.

GİZEM ÖTER

Tarih ve İmza

30.01.2023

TEŞEKKÜR

Yüksek lisans eğitimim boyunca yardımlarını ve desteklerini benden esirgemeyen, bu konuda çalışmak için beni cesaretlendiren, bana her daim sabır ve hoşgörü ile yaklaşan saygıdeğer danışman hocam Doç. Dr. Emrah HANÇER'e teşekkürü bir borç bilirim.

Maddi ve manevi her zaman yanımda olan, tüm zorlukları aşmamda bana destek olan ve bana olan inancını hiçbir zaman kaybetmeyen kıymetli annem Aysel ÖTER'e ve babam Yılmaz ÖTER'e teşekkür ederim.

Burdur, 2023

(ÖTER, Gizem, “X-ray Görüntülerinden Covid-19 Tespiti İçin Derin Öğrenme Tabanlı Modellerin Geliştirilmesi”, Yüksek Lisans Tezi, Burdur, 2023)

ÖZET

Şiddetli akut solunum yolu sendromu koronavirüs 2'nin (SARS-CoV-2) neden olduğu Covid-19 salgını tüm dünyada hayatı durma noktasına getirdi ve binlerce can aldı. Bu ölümcül salgın ile mücadelede en önemli unsur erken teşhistir. Erken teşhisle enfekte olan kişi karantinaya alınarak bulaş hızı ve ölüm oranı düşürülebilir. Günümüzde Covid-19'un teşhisinde PCR (Polymerase Chain Reaction- Polimeraz Zincir Reaksiyonu) testlerinden ve akciğer X-ray görüntülerinden faydalanılmaktadır. PCR testlerinin sonuçlanması zaman alan bir süreç olması sebebiyle erken teşhiste X-ray görüntüleri hayati bir öneme sahiptir. Bu çalışmada Covid-19 teşhisinde uzmanlara yardımcı olabilmek amacıyla X-ray görüntülerinden bir teşhis metodolojisi tasarlanmıştır. Metodolojinin ilk aşamasında ön eğitilmiş SqueezeNet mimarisi ile X-ray görüntülerinden oluşan veri setinden öznitelikler çıkarılmıştır. İkinci aşamada çıkarılan özniteliklere en yakın bileşen analizi (neighborhood component analysis (NCA)) uygulanarak mevcut öznitelik setinden gürültülü olanlar elimine edilmiştir. Son aşamada ise, destek vektör makineleri ile gürültüden arındırılmış öznitelik seti üzerinde Covid-19 tespiti gerçekleştirilmiştir. Önerilen metodolojinin performans analizi 5 farklı veri seti üzerinde birden çok derin öğrenme mimarileri ve sınıflandırıcılar ile karşılaştırılarak yapılmıştır. Elde edilen sonuçlar incelendiğinde önerilen metodolojinin daha başarılı sonuçlar aldığı görülmüştür.

Anahtar Kelimeler: Derin Öğrenme, SqueezeNet, Covid-19, NCA.

(ÖTER, Gizem, “Development of Deep Learning Based Models for Covid-19 Detection from X-ray Images”, Master Thesis, Burdur, 2023)

ABSTRACT

Thousands of people died as a result of the Covid-19 epidemic, which was caused by severe acute respiratory syndrome Coronavirus 2 (SARS-CoV-2). The most critical element in combating this deadly epidemic is early diagnosis. Through early diagnosis and quarantine, it is possible to reduce the rate of transmission and death. Today, PCR (Polymerase Chain Reaction) tests and lung X-ray images are used in the diagnosis of Covid-19. X-ray images are of vital importance in early diagnosis, as the conclusion of PCR tests is a time-consuming process. The overall goal of this study is to develop a diagnostic methodology based on X-ray images in order to assist experts in diagnosing Covid-19. In the first stage of the methodology, features are extracted from the dataset consisting of X-ray images by using a pre-trained SqueezeNet architecture. In the second stage, nearest component analysis (NCA) is applied to the extracted feature set, and noisy ones are eliminated from this feature set. In the final stage, Covid-19 detection is performed on the noise-free feature set with support vector machines. The performance analysis of the proposed methodology is made by comparing it with a variety of deep pre-trained architectures and classifiers on 5 different data sets. According to the results, the proposed methodology achieves better results than others.

Keywords: *Deep Learning, SqueezeNet, Covid-19, NCA.*

İÇİNDEKİLER

ETİK BEYAN.....	i
TEŞEKKÜR.....	ii
ÖZET.....	iii
ABSTRACT	iv
İÇİNDEKİLER.....	v
KISALTMALAR.....	viii
TABLO DİZİNİ.....	ix
ŞEKİLLER DİZİNİ	x
GİRİŞ.....	1
BİRİNCİ BÖLÜM.....	4
KORONAVİRÜS HASTALIĞI (COVID-19)	4
1. 1. Tanım	4
1. 2. Koronavirüs'ün Bulaşma Yolları	4
1. 3. Koronavirüs' ün Teşhis Yöntemleri	5
1. 4. Literatür Taraması	7
İKİNCİ BÖLÜM	16
YAPAY ZEKA TEKNİKLERİ.....	16
2. 1. Yapay Zeka	16
2. 2. Makine Öğrenmesi	17
2. 2. 1. Makine Öğrenmesi Teknikleri	18
2. 2. 1. 1. Gözetimsiz Öğrenme	19
2. 2. 1. 1. 1. K-ortalamlar (K-means)	20
2. 2. 1. 1. 2. Apriori Algoritması	21
2. 2. 1. 2. Gözetimli Öğrenme	21
2. 2. 1. 2. 1. K-en yakın komşu (kNN)	22
2. 2. 1. 2. 2. Destek Vektör Makinesi (SVM)	23

2. 2. 1. 2. 3. Naive Bayes.....	24
2. 2. 1. 2. 4. Bagging	25
2. 2. 1. 2. 5. Yapay Sinir Ağları	26
2. 2. 1. 3. Pekiştirmeli Öğrenme.....	27
2. 3. Derin Öğrenme	28
2. 3. 1. Derin Öğrenme Algoritmaları Kullanılarak Neler Yapıldı?.....	28
2. 4. Öğrenme Aktarımı (Transfer Learning)	29
2. 5. Derin Öğrenme Algoritmaları	30
2. 5. 1. Evrişimli Sinir Ağları (CNN).....	30
2. 5. 2. ResNet	30
2. 5. 3. MobileNetV2.....	32
2. 5. 4. VGG16	32
2. 5. 5. GoogLeNet	33
2. 5. 6. SqueezeNet.....	34
2. 6. Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenme Kavramlarının Karşılaştırılması	35
ÜÇÜNCÜ BÖLÜM	36
MATERYAL VE METOT	36
3. 1. Önerilen Metodoloji	36
3. 2. Verilerin Toplanması.....	36
3. 2. 1. Çalışmada Kullanılan Veri Setleri.....	36
3. 3. SqueezeNet ile Öğrenme Aktarımı	39
3. 4. Öznitelik Seçme	42
3. 4. 1. Komşuluk Bileşen Analizi (NCA)	42
3. 5. Destek Vektör Makineleri (SVM) ile Sınıflandırma.....	44
DÖRDÜNCÜ BÖLÜM	48
BULGULAR	48
4. 1. Değerlendirme Metrikleri.....	48
4. 2. Deneysel Sonuçlar	49
4. 2. 1. Öznitelik Seçme ile Elde Edilen Sonuçların Karşılaştırılması.....	49

4. 2. 2. Önerilen Metodolojinin Diğer Derin Öğrenme Mimarileri ile Karşılaştırılması	53
SONUÇLAR	55
KAYNAKÇA	56
EK 1. KOD DİZİNİ	64
ÖZGEÇMİŞ	67



KISALTMALAR

CNN: Convolutional neural network

FFB: Feature Fusion-Based

KNN: K-Nearest Neighbors

MVF: Multi-view fusion

NCA: Neighbourhood Components Analysis

PCR: Polimeraz Zincir Reaksiyonu

ResNet: Residual Network

RF: Random Forest

SLB: Single layer-based

SVM: Support vector machine

WHO: World Health Organization

TABLO DİZİNİ

Tablo 1. Literatür Taraması	13
Tablo 2. Makine Öğrenmesi Teknikleri	18
Tablo 3. Dataset 1	37
Tablo 4. Dataset 2	37
Tablo 5. Dataset 3	38
Tablo 6. Dataset 4	38
Tablo 7. Dataset 5	38
Tablo 8. Çekirdek Fonksiyonu	47
Tablo 9. Deneysel Çalışma Sonuçları.....	51
Tablo 10. Sınıflandırıcıların Karşılaştırılması	52
Tablo 11. Öznitelik Çıkarımı Yapılan Katmanlar	53
Tablo 12. Deneysel Sonuçların Karşılaştırılması	54

ŞEKİLLER DİZİNİ

Şekil 1. Hayvanlardan İnsana Covid-19'un Bulaşma Süreci	5
Şekil 2. PCR Testi	6
Şekil 3. X-Ray Görüntülerinde Opasifikasyon Görünümler	6
Şekil 4. Örnek X-ray Görüntüleri	6
Şekil 5. Makine Öğrenmesi Teknikleri	19
Şekil 6. Gözetimsiz Öğrenme	20
Şekil 7. Gözetimli Öğrenme	22
Şekil 8. K-En Yakın Komşu Algoritması	23
Şekil 9. Bagging Yöntemi	26
Şekil 10. Yapay Sinir Ağları Modeli	27
Şekil 11. Pekiştirmeli Öğrenme	27
Şekil 12. Derin Öğrenme Modeli	28
Şekil 13. Geleneksel Öğrenme ve Öğrenme Aktarımı Öğrenim Süreci	29
Şekil 14. Evrişimli Sinir Ağı Modeli	30
Şekil 15. ResNet Modeli	31
Şekil 16. MobileNetV2'nin Darboğaz Kalıntı Katmanı	32
Şekil 17. VGG16 Modeli	33
Şekil 18. GoogLeNet Mimarisi	34
Şekil 19. Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenme İlişkisinin Şematik Gösterimi	35
Şekil 20. Çalışmanın Metodolojisi	36
Şekil 21. Ateşleme Modülündeki Evrişim Filtrelerinin Organizasyonu. Bu örnekte; $s1 \times 1 = 3$ $e1 \times 1 = 4$ ve $e3 \times 3 = 4$ olarak ayarlanmıştır	40
Şekil 22. Sol: SqueezeNet, Orta: Basit atlama SqueezeNet, Sağ: Karışık Atlama SqueezeNet	41

Şekil 23. Doğrusal Ayrılabilen Destek Vektör Makinesi	45
Şekil 24. Doğrusal Ayrılamayan Destek Vektör Makinesi.....	46



GİRİŞ

Günümüzde sağlık sorunlarının çözümünde sıklıkla bilişim sistemlerinden ve farklı teknolojilerden yararlanılmaktadır. Sağlık hizmetlerinin belirli bir düzeyde ve etkin bir şekilde yönetilebilmesi amacıyla veri oluşturulması ve oluşturulan verilerin etkili bir şekilde kullanılabilmesi için her türlü yazılım, donanım vb. sistemler sağlık bilişim sistemleri olarak ifade edilmektedir. Hastanelerin bilgi sistemlerinden elde edilen hastane, hasta ve tıbbi bilgiler veri bilimciler tarafından derlenerek analiz edilebilir veriler haline getirilebilmektedir. Sağlık bilişim sistemleri ile erken teşhis, tedavi ve hataları en minimum seviyeye indirmek mümkündür (Yücel vd., 2018: 152).

Son dönemde sağlık bilişim sistemleri alanındaki çalışmalar küresel sağlık krizi olarak tanımlanan Covid-19 üzerine yoğunlaşmıştır. Her ülke virüsle mücadele sürecinde sokağa çıkma kısıtlamaları, seyahat kısıtlamaları, sosyal mesafe, maske zorunluluğu ve hijyen bilincini arttırmaya yönelik çalışmalar yapmış ve yapmaya devam etmektedir. Ancak, virüsün etkileri tam anlamıyla bitirilebilmiş değildir (Chowdhury vd., 2020: 132665-132669).

Kuşkusuz Covid-19 ile mücadelede en önemli hususlardan birisi hızlı tanı ve hızlı müdahaledir. Günümüzde yapılan tanı yöntemlerinden en yaygını PCR (Polimeraz Zincir Reaksiyon) testleridir. Yapılan bu testlerin sonuçlanması zaman almaktadır ve bu da ilgili süreci olumsuz etkilemektedir. Bu test yönteminin kitlelere bağımlı olması ve kırsal alanlarda erişim sıkıntısı olması gibi eksiklikleri de söz konusudur. Bu testlerin temel hedefinde, sonucu daha hızlı bir şekilde verebilmek olmasına rağmen maliyetleri fazladır ve test sonucu verme süresi X-ray görüntülerinden alınan sonuçlara oranla daha uzundur.

Covid-19 virüsünün tedavisi için ilaç ve aşı geliştirme çabaları devam etmektedir. Ancak hala tedavisi için onay alınmış spesifik bir ilaç söz konusu değildir. Aşı geliştirmede ise birçok araştırma grupları tarafından farklı teknolojiler ve yöntemlerle çalışmalar yapılmış ve yapılmaya devam etmektedir. Bu aşılarından bazıları Dünya Sağlık Örgütü (WHO) tarafından acil kullanım onayı almıştır. Her ne kadar Covid-19 için aşı üretilmiş olsa da nakliye, depolama, üretim yetersizliği ve ülkeler arası gelişmişlik düzeyinin farklı olması gibi

olumsuz etkenlerden dolayı aşılama tam anlamıyla her ülkede gerçekleşmemiştir. Virüs mutasyon geçirdiğinden dolayı aşılama sağladığı bağışıklığı düşürebilmektedir. Bunun gibi olumsuz nedenlerden dolayı tüm dünyada Covid-19'un etkileri uzun bir süre daha devam edebilir (Calina vd., 2020: 3-5).

Röntgen bir diğer adıyla X-ray bir nesnenin içini görselleştirmek için belirli radyasyon türlerini kullanan bir görüntüleme tekniğidir. Tıpta X-ray görüntülemeleri oldukça erişilebilir ve akciğer koşullarını değerlendirmede kullanılan yaygın bir yöntemdir. Göğüs röntgeni ise göğüs bölümünde bulunan kemik, doku ve organların görüntülenme şeklidir. Covid-19, X-ray görüntülerinde opasifikasyon (Buzlu cam görünümü) şeklinde belirtilen beyaz lekeler şeklinde görüntülenebilir (Ardakani vd., 2020:1-2). Buna bağlı olarak X-ray görüntüleri, Covid-19'un tespitinde kullanılabilir. Buradan yola çıkılarak son zamanlarda bu görüntüler üzerinde derin öğrenme algoritmaları kullanılarak Covid-19 tespiti üzerinde çalışmalar yoğunlaşmıştır.

Bu çalışmanın amacı, mevcut X-ray görüntülerinden Covid-19 hastalığı tespitine yönelik olarak geliştirilmiş bir metodoloji ortaya koymaktır. Bu amacı gerçekleştirmek için öncelikle üç aşamalı bir derin öğrenme tabanlı Covid-19 tespit metodolojisi geliştirilmiştir. İlk aşamada ön eğitilmiş SqueezeNet mimarisi (Iandola, 2016) ile X-ray görüntülerinden özniteliklerin çıkarılması sağlanır. İkinci aşamada çıkarılan özniteliklerden komşuluk bileşen analizi (neighborhood component analysis (NCA)) algoritması (Goldberger, 2004) ile ilgili olan özniteliklerin seçilmesi sağlanır. Son aşamada ise seçilen öznitelik alt seti üzerinde destek vektör makineleri (SVM) algoritması ile bir sınıflandırma modeli oluşturularak Covid-19 tespiti gerçekleştirilir. Geliştirilen metodoloji spesifik bir veri setine göre dizayn edilmeyip 5 farklı Covid-19 ile ilişkin veri seti üzerinde uygulanmıştır.

Çalışmanın bundan sonrası dizaynı şu şekildedir. Bölüm 1'de Covid-19 virüsü hakkında bilgiler ve literatür taraması yer alacaktır. Bölüm 2'de yapay zeka tekniklerine değinilecektir. Bölüm 3'te kullanılan veri setleri, önerilen metodoloji, metodolojide kullanılan algoritmaların parametre ayarları ve değerlendirme için

kullanılan metrikler verilecektir. Son bölümde ise elde edilen deneysel sonuçlar tartışılacak ve çalışmanın genel değerdendirmesi yapılacak olup ileride yapılması muhtemel çalışmalara değinilecektir.



BİRİNCİ BÖLÜM

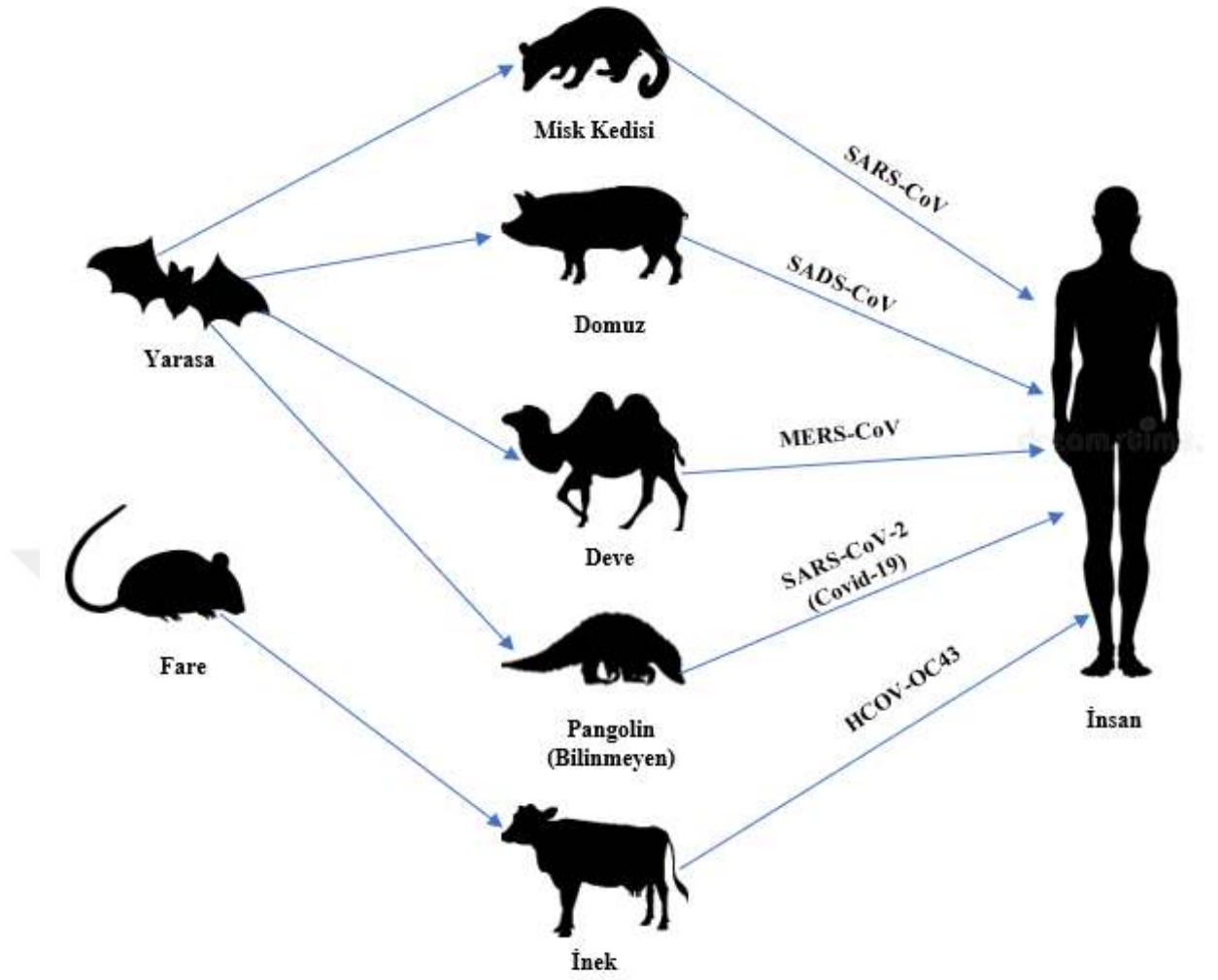
KORONAVİRÜS HASTALIĞI (COVID-19)

1. 1. Tanım

2019'un aralık ayında Çin'in Wuhan kentinde bilinmeyen zatürre (pnömoni) vakalarının bildirilmesiyle beraber ortaya çıkan SARS-CoV-2 'nin sebep olduğu bulaşıcı bir hastalıktır. Salgın ilk olarak bölgedeki deniz ürünleri ve hayvan pazarında bulunanlarda tespit edilmiştir (T.C. Sağlık Bakanlığı., t.y.). Dünya Sağlık Örgütü (WHO) tarafından 11 Mart 2020 tarihinde bu virüsün pandemiye dönüştüğü duyurulmuştur. Ülkemizdeki ilk Covid-19 vakası 11 Mart 2020'de bildirilmiştir. Semptomlar her ne kadar çok farklılık gösterse de yüksek ateş, öksürük, nefes darlığı, yorgunluk, koku ve tat alamama semptomlarına sıklıkla rastlanmıştır. Covid-19 virüsü Türkiye'de 10 Ekim 2022 tarihine kadar geçen süre zarfında; 16.896.522 vaka sayına ulaşmış ve 101.179 kişinin ölümüne neden olmuştur (T.C. Sağlık Bakanlığı., t.y.).

1. 2. Koronavirüs'ün Bulaşma Yolları

Hasta olan birey öksürdüğünde, konuştuğunda ya da hapşırdığında havaya ve etrafa damlacıklar saçılır. Bu damlacıkları sağlıklı bireylerin soluması ya da temas etmesi halinde bulaş gerçekleşebilir. Bundan dolayı hasta bireylerle yakın temas halinde bulunmak virüsün bulaşmasına neden olabilmektedir. Koronavirüs ailesinin üyeleri alfa-CoV (α), beta-CoV (β), gama-CoV (δ) ve delta-CoV (γ)'dan oluşmaktadır. SARS-CoV-2 virüsünün beta-CoV (β) ailesi grubunda yer aldığı tespit edilmiştir (Chang vd., 2020: 76-77). Beta-koronavirüsler (beta-CoV (β)), Şekil 1'de gösterildiği gibi ilk olarak fare, yarasa gibi vahşi hayvanlardan, misk kedisi, domuz, deve, pangolin, inek gibi evcil hayvanlara akabinde insanlara bulaşmıştır. Pangolin canlılarında SARS-CoV-2'e yakın bir virüsün bulunması sebebi ile koronavirüslerinin olası konakçıları oldukları düşünülmektedir. Hayvanlardan koronavirüs bulaşımının durdurulabilmesi için vahşi hayvanlardan ve vahşi hayvan pazarlarından uzak durulması gerekmektedir (Lam vd., 2020: 282-283; Loey vd., 2020: 2).

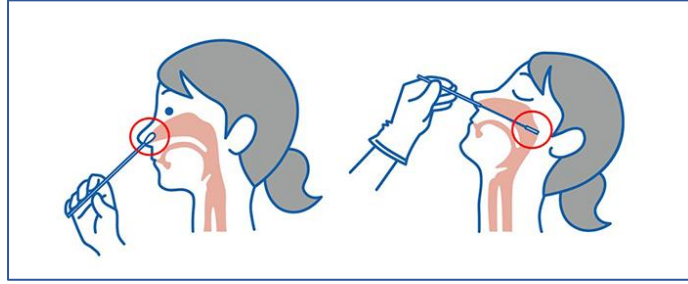


Şekil 1. Hayvanlardan İnsana Covid-19'un Bulaşma Süreci (Loey vd., 2020: 2)

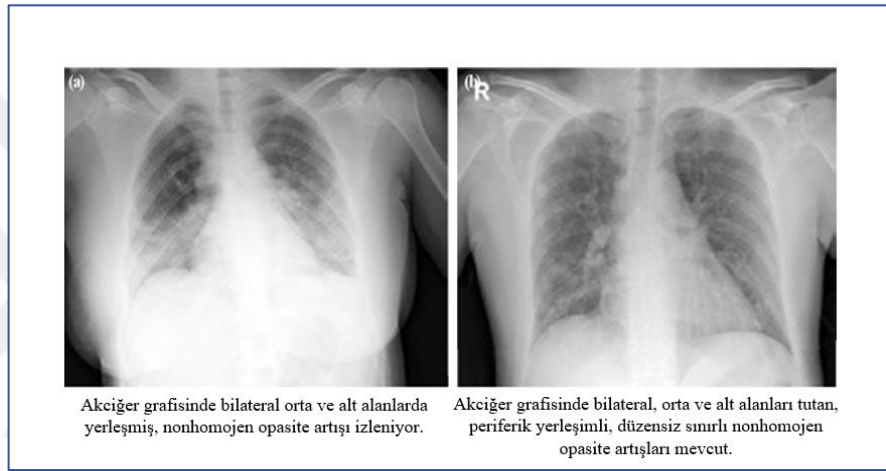
1. 3. Koronavirüs' ün Teşhis Yöntemleri

PCR, kuşkusuz Covid-19 ile mücadelede en önemli hususlardan birisi hızlı teşhis ve hızlı müdahaledir. Günümüzde yapılan teşhis yöntemlerinden en yaygın olanı PCR (Polimeraz Zincir Reaksiyon) testleridir. PCR testi, burun ve boğazdan ince bir çubuk yardımı ile sürüntü örneği alınan testlerdir (Şekil 2). Bunun yanı sıra öksürük dolayısı ile çıkan balgamdan da örnek alınabilmektedir. Ancak hastalığın ilerleyen safhasında havayollarından örnek alınması çok doğru olmayabilir. Bunun sebebi burun ve boğazda virüs kalıntılarının kalmamış olabileceğidir. (Tang vd.,2020: 3-5) Salgının başladığı ilk zamanlarda testlerin bazı yerlerde yapılamaması, test sayısındaki yetersizlik ve test sonuçların alınma

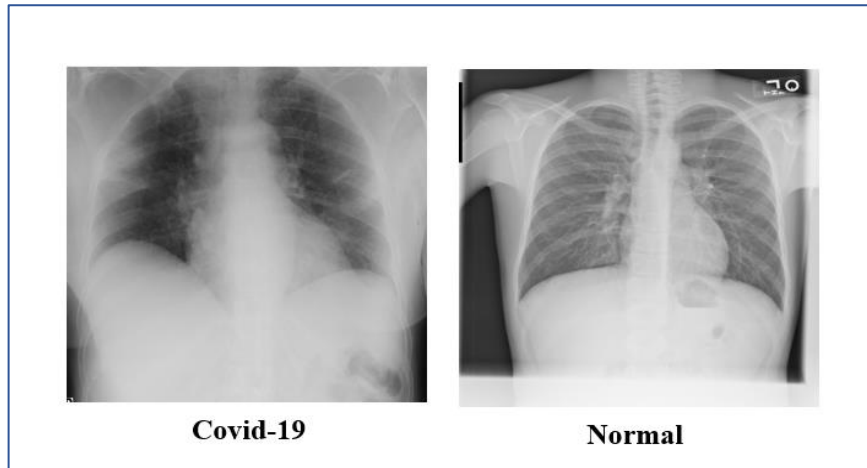
sürelerinin uzaması gibi durumlardan dolayı Covid-19 teşhisinde X-ray görüntüleri de kullanılmıştır.



Şekil 2. PCR Testi



Şekil 3. X-Ray Görüntülerinde Opasifikasyon Görünümler (Cömert, Kırıl, 2020: 17)



Şekil 4. Örnek X-ray Görüntüleri

X-ray kavramı radyoloji biliminin en temel unsurlarından biridir. Röntgen bir diğer adıyla X-ray bir nesnenin içini görselleştirmek için belirli radyasyon

türlerini kullanan ilk görüntüleme tekniğidir. X-ray cihazlarının temel amacı, x ışınlarını tüm vücuttan geçirerek bir alanda görüntü oluşturmaktır. Günümüzde tıbbi görüntüleme alanında birçok yeni teknoloji geliştirilmiştir. Ancak, X-ray teknolojisi tek görüntüyle en fazla bilgiyi barındırması, uzaysal çözünürlüğün yüksek olması, cerrahi işlem gerektirmemesi, basit ve ucuz olması gibi nedenlerden dolayı hala sıklıkla tercih edilen radyolojik görüntüleme tekniğidir. Şekil 3'te ifade edildiği üzere Covid-19 X-ray görüntülerinde buzlu cam opasitesi şeklinde belirtilen beyaz lekeler formunda görüntülenebilmektedir. Buzlu cam opasitesi, Covid-19 tespitinde en sık rastlanan X-ray bulguları arasında yer alır. Genellikle hastalığın başlangıcında ya da hafif geçirildiği durumlarda oluşur. Tek taraflı ya da bilateraldir. Akciğer ödemi (pulmoner) ve hyalin membran oluşumuna bağlı gelişme göstermektedir. Küçük konsolidasyonlarla birlikte saptanması durumunda akciğer hasarına bağlı olarak zatürrenin (pnömoniye) ilerlediği gözlemlenebilmektedir (Yurtsever, 2021: 199-200; Kaya, 2017: 1). Şekil 4'te ise çalışmada kullanılan veri setlerinden alınan Covid-19 ve normal (hasta olmayan) bireylere ait örnek X-ray görüntüleri yer almaktadır.

1. 4. Literatür Taraması

Covid-19 tüm dünyayı etkisi altına alan güncel bir salgın olduğundan dolayı birçok bilimsel çalışmanın konusu olmuş ve olmaya devam etmektedir. Yapılan çalışmalarda genel olarak Covid-19'un teşhisinde kullanılan radyolojik görüntüler ile derin öğrenme algoritmaları kullanılarak daha kısa sürede teşhis yapabilen sistemler tasarlanmıştır.

Toğaçar vd. (2019) evrişimsel sinir ağlarını kullanarak zatürre hastalığını erken tespit etmeye çalışmışlardır. Çalışmada kullanılan veri seti 1-5 yaş arası çocukların göğüs X-ray görüntülerinden oluşmaktadır. AlexNet mimarisi kullanılarak öznitelik çıkarımı yapılmıştır. Elde edilen öznitelikler karar ağaçları, destek vektör makinesi, lineer regresyon, doğrusal ayırmacılık analizi ve en yakın komşu kullanılarak sınıflandırılmıştır. Çalışmada en iyi sonuçların destek vektör makinesi ile alındığı gözlemlenmiştir.

Minaee vd. (2020) evrişimsel sinir ağlarını kullanarak Covid-19 tespitine yardımcı olmayı amaçlamışlardır. Birden fazla kaynaktan toplanan 5000 X-ray

görüntüsünden yeni veri seti oluşturulmuştur. Veri seti SqueezeNet, ResNet50, ResNet18 ve DenseNet161 mimarileri kullanılarak eğitilmiştir. Sınıflandırma sonuçlara göre ROC eğrisi çıkarılmıştır. ROC-AUC skorları 0.97 ve 0.99 arasında bulunmuştur. Kullanılan her bir mimarinin duyarlılık değeri yaklaşık olarak %98, özgüllük değeri ise %90 civarında bulunmuştur. Sonuçlara bakıldığında çok az bir farkla da olsa SqueezeNet mimarisi ile en başarılı sonuçlar elde edilmiştir.

Rahman vd. (2020) bakteriyel zatürre ve zatürrenin otomatik olarak tespit edebilmesine yönelik bir çalışma ortaya koymuşlardır. Çalışmada 5247 X-ray görüntüsünden oluşan veri seti ve AlexNet, ResNet18, SqueezeNet, DenseNet201 mimarileri kullanılmıştır. DenseNet201 mimarisi ile en başarılı sonuçlar elde edilmiştir.

Wu vd. (2020), tomografi görüntüleri kullanarak hastaların tanısının hızlı bir şekilde konulabilmesi için derin öğrenmeye dayalı model tasarlanmıştır. Çin'deki üç hastaneden 495 hastanın göğüs tomografisini alınmıştır. Veri setinin 368 görüntüsü Covid-19 hastasına diğer 127 tane görüntü ise zatürre hastalarına aittir. Veri seti %80 eğitim, %10 test ve %10 doğrulama şeklinde ayrılmıştır. Görüntülerin her biri 256x256 olarak yeniden boyutlandırılmıştır. Tasarlanan modelde ResNet50 ağı kullanılmıştır. Çalışmada hem multi-view fusion (MVF) ile hemde single-view fusion (SVF) analizi yapılmıştır, ancak MVF ile daha başarılı sonuçlar elde edilmiştir.

Özcan, (2020) GoogLeNet, ResNet18 ve ResNet50 mimarilerini kullanarak Covid-19 tespitine yardımcı olmaya çalışmıştır. Çalışmada Grid Search ve öğrenme aktarımı destekli evrimsel sinir ağına dayalı bir yöntem önerilmiştir. Kullanılan veri seti %50 eğitim, %20 doğrulama ve %30 test olacak şekilde ayrılmıştır. Sonuçlar karşılaştırıldığında, en başarılı sonucun Grid Search ve ResNet50 kullanılarak sağlandığı gözlemlenmiştir. Bir başka çalışmada, (Özcan, 2021) X-ray görüntülerinden çıkarılan derin öznitelikler kullanılarak Covid-19 vakalarının tespit edilmesinde yeni bir birleşik model yaklaşımı önerilmiştir. Yaklaşımın single layer-based (SLB) ve feature fusion-based (FFB) olmak üzere iki varyansı vardır. SLB modeli, ön işleme, derin öznitelik çıkarımı, son işleme ve sınıflandırma aşamalarından oluşmaktadır. FFB modeli ise ön

işleme, derin öznitelik çıkarımı, öznitelik birleştirme, son işleme ve sınıflandırma aşamalarından oluşmaktadır. Öznitelik çıkarma aşamasında kullanılan katmanların sayısı ve ikili kombinasyonuna göre dört farklı SLB ve altı farklı FFB modeli geliştirilmiştir. Her model için ikili ve çoklu sınıflandırma işlemi yapılmıştır. FFB3 modeli ile daha başarılı sonuçlar elde edilmiştir.

Alqudah vd. (2020) Covid-19'un erken saptanması için yapay zeka tekniklerini kullanmışlardır. Covid-19 veya hasta olmayanların (normal) sınıflandırılmasında en iyi performansı elde etmek için evrişimli sinir ağı (CNN), destek vektör makinesi (SVM) ve rastgele orman (RF) gibi farklı makine öğrenimi algoritmaları kullanılarak gerçekleştirilmiştir. Çıkarılan en önemli iki öznitelik, eğitim ve parametre testi için kullanılmıştır. Tasarlanan modelin performans sonuçlarına bakıldığında CNN, 9 test doğruluğu ile diğer sınıflandırıcılardan daha iyi performans göstermiştir.

Jaiswal vd. (2020) Covid-19, normal ikili sınıflandırmasını yapabilmek için VGG16, InceptionResNet, ResNet152V2 ve DenseNet201 mimarilerinden yararlanmıştır. Çalışmada 1.262 adet Covid-19 ve 1.230 adet normal bilgisayarlı tomografi görüntüleri kullanılmıştır. Veri seti %68 eğitim, %15 test ve %17 doğrulama olacak şekilde ayrılmıştır. Deneysel sonuçlara bakıldığında ise en başarılı modelin DenseNet201 olduğu saptanmıştır.

Wang vd. (2020)'nin yaptıkları çalışmada Covid-19, zatürre ve normal X-ray görüntüleri ile üçlü sınıflandırma yapılmıştır. Beş farklı açık kaynak veri setinden elde ettikleri görüntü verilerini birleştirerek COVIDx adında yeni bir veri seti oluşturulmuştur. Bu görüntü verileri ile eğittikleri derin evrişimsel sinir ağına ise COVID-Net adını vermişlerdir. Çalışmada önerilen COVID-Net mimarisi, VGG19 ve ResNet50 ile karşılaştırılmıştır. Elde edilen sonuçlar neticesinde COVID-NET' in diğer mimarilerinden daha iyi performans sergilediği gözlemlenmiştir.

Hemdan vd. (2020), Covid-19 teşhisi için CovidX-Net'i önermişlerdir. Önerilen metodoloji yedi farklı mimari (VGG19, DenseNet201, InceptionV3, MobileNetV2, ResNetV2, Xception ve InceptionResNetV2) ile sınıflandırılmıştır. Çalışmada kullanılan veri seti 25 Covid-19 ve 25 normal X-ray

görüntüsünden oluşmaktadır. Sınıflandırma sonuçlara bakıldığında en iyi performansların VGG19 ve DenseNet201 ile elde edildiği gözlemlenmiştir.

Sethy ve Behera (2020) Covid-19 tespiti için 25 Covid-19, 25 normal görüntüden olan veriler ile derin öznitelik çıkarımı yapmışlardır. Xception, ResNet-18, ResNet-50, ResNet-101, InceptionV3, InceptionResNetV2, GoogLeNet ve DenseNet-201 ağlarında 1000 öznitelik çıkarmışlardır. VGG16, VGG19 ve AlexNet ağlarında ise 4096 öznitelik çıkarılmıştır. Elde edilmiş olan öznitelikler SVM kullanılarak sınıflandırılmış ve sonuçlarının karşılaştırmalı analizi yapılmıştır. Çalışmadan elde edilen sonuçlara bakıldığında ResNet-50 ile en başarılı sonuçlar elde edilmiştir.

Nayak vd. (2021) Covid-19 teşhisine yardımcı olabilmek adına Inception-V3, AlexNet, ResNet-34, VGG-16, GoogleNet, MobileNet-V2, ResNet-50 ve SqueezeNet gibi derin öğrenme mimarilerini kullanarak karşılaştırmalı analiz yapmışlardır. Çalışmada kullanılan veri seti 256 (143 Covid-19, 143 Normal) eğitim, 120 (Covid-19 60, Normal 60) test olarak ayrılmıştır. Deneysel bulgulara bakıldığında ise en başarılı sonuçların ResNet-34 ile elde edildiği gözlemlenmiştir.

Eryılmaz ve Karacan (2021) Covid-19 ve diğer akciğer hastalıklarına ait X-ray görüntüleriyle çoklu sınıflandırma yapmayı amaçlamıştır. Çalışmada kullanılan veri setini daha dengeli hale getirmek amacıyla normal ve opasiteye ait görüntü sayısı en değerli veri olan Covid-19 X-ray görüntü sayısına indirilmiştir. Daha başarılı sınıflandırma sonuçları elde edebilmek için eğitim setinde veri artırım yöntemleri kullanılmıştır. Veri setinde yer alan X-ray görüntüleri MobileNetV2, NASNetMobile, Xception ve DenseNet121 mimarileri kullanılarak sınıflandırmış ve sonuçların karşılaştırmalı analizi yapmıştır. Çalışmada en iyi sonuçlar DenseNet121 ve MobileNet mimarileri kullanılarak elde edilmiştir.

Ismael ve Şengür (2021) Covid-19 tespitini yapabilmek adına Covid-19 ve sağlıklı bireylerin X-ray görüntülerini sınıflandırmıştır. Çalışmada 380 X-ray görüntüsünden oluşan veri seti kullanılarak ResNet18, ResNet50, ResNet101, VGG16 ve VGG19 modelleri ile öznitelik çıkarımı yapılmıştır. Özniteliklerin

sınıflandırılması için destek vektör makinesi, lineer, Gauss ve Cubic gibi çekirdek fonksiyonlarından yararlanılmıştır. Çalışmadan elde edilenlere bakıldığında en başarılı sonuçların ResNet50 modeli ve lineer çekirdek fonksiyonlu destek vektör makinesi sınıflandırıcısı ile sağlanmıştır.

Alhudhaif vd. (2021) çalışmalarında 368 Covid-19 ve 850 diğer hastalık etkenlerinin sebep olduğu zatürre hastalarının X-ray görüntüleri ResNet18, SqueezeNet ve DenseNet201 ağırları ile ikili sınıflandırma yapmışlardır. Veri setinin %90'ı eğitim için ve %10'u test (görünmez) aşamalı bir beş katlamalı çapraz doğrulama (k-fold cross validation) yaklaşımını kullanılmışlardır. Çalışmada en başarılı sonuçlar DenseNet201 ile elde edilmiştir.

Bozkurt (2021) çalışmasında 11 farklı derin öğrenme modeli kullanarak akciğer X-ray görüntülerini Covid-19, normal ve viral pnömoni (zatürre) bireyler olarak sınıflandırabilen bir model tasarlamıştır. Çalışmada 11 farklı derin öğrenme algoritması (AlexNet, VGG16, VGG19, InceptionV3, DenseNet121, ResNet50, InceptionV3, Derin evrişimli ağ modeli, Sığ-evrişimli ağ modeli, Derin-tam bağlı ağ modeli ve sığ tam bağlı ağ modeli) üzerinde çalışılmış ve en başarılı sonuç DenseNet121 ile elde edilmiştir.

Narin vd. (2021) Covid-19 hastalarını otomatik olarak tahmin etmek için normal, Covid-19, bakteriyel zatürre ve zatürre hastalarından elde edilen görüntü verilerini kullanarak önceden eğitilmiş beş evrişimli sinir ağı (CNN) tabanlı model (ResNet50, ResNet101, ResNet152, InceptionV3 ve Inception-ResNetV2) önerilmiştir. Beş katlamalı çapraz doğrulama (k-fold cross validation) kullanılarak dört sınıf ile üç farklı ikili sınıflandırma gerçekleştirilmiştir. İlk ikili sınıflandırma Covid-19-normal, ikinci ikili sınıflandırma Covid-19-zatürre ve üçüncü ikili sınıflandırma ise Covid-19 bakteriyel zatürre şeklindedir. Çalışmadan elde edilen sonuçlara bakıldığında, tüm sınıflandırmalarda en başarılı sonuçlar ResNet50 ile alınmıştır.

Sünneci vd. (2021), 137 Covid-19, 90 sağlıklı ve 90 zatürre bireylerin X-ray görüntülerini AlexNet mimarisi kullanarak öznelik çıkarımını yapmıştır. Tam bağlantılı katmanlardan her bir görüntü için 1000 öznelik elde edilmiştir. Elde edilen öznelikler kullanılarak Medium Tree, Bagged Trees Ensemble,

Kernel Naive Bayes, Cosine K-Nearest Neighbor, Linear Discriminant, Quadratic SVM, Linear SVM ve Cubic SVM sınıflandırıcıları eğitilmiştir. En başarılı sonuçlar Cubic SVM sınıflandırıcısı ile elde edilmiştir.

Demir ve Yılmaz (2021) akciğer radyolojik görüntülerinden yararlanarak Covid-19'un hızlı ve doğru teşhisi amacıyla derin öğrenme temelli bir yaklaşım kullanmışlardır. Çalışmada 3616 Covid-19 ve 10192 normal hasta bilgilerini içeren veri seti kullanılmıştır. Basic-CNN ve VGG16 mimarileri kullanılarak sınıflandırma işlemi yapılmıştır. Elde edilen deneysel sonuçlara bakıldığında VGG16 mimarisi ile daha başarılı sonuçlar elde edilmiştir.

Moulaei vd. (2022) Covid-19 hastalarının ölüm riskini ve yüksek riskli hastaları tespit etmişlerdir. Çalışmada Ayatollah Taleghani hastanesinden alınan 1500 hasta verisi kullanılmıştır. Öznitelik tanımlama ve seçimi yapıldıktan sonra iki aşamalı Delphi araştırması ile Covid-19 hastalarının ölüm riskini belirlemek için olası 54 klinik öz niteliği belirlenmiştir. Özniteliklerin önem derecesi genetik algoritma kullanılarak yapılmıştır. Önem düzeyi en yüksek olan öz nitelik nefes darlığı en düşük olan ise trombosit sayımı olarak bulunmuştur. Covid-19 hastalarının ölüm riskini belirlemek için random forests, XGBoost, K-en yakın komşu, çok katmanlı algılayıcılar, lojistik regresyon, J-48 karar ağacı ve Naive Bayes algoritmaları kullanılmıştır. Deneysel sonuçlara bakıldığında en başarılı sonuçların random forests algoritması ile elde edildiği gözlemlenmiştir.

Bahsi geçen tüm çalışmalar özetlenerek ayrıca Tablo 1'de de sunulmuştur. Literatürdeki çalışmalardan görüleceği üzere, X-ray görüntülerinden Covid-19 başta olmak üzere çeşitli hastalıkların otomatik olarak tespit edilmesine yönelik çalışmalar hızla devam etmektedir. Ancak mevcut çalışmaların çoğunluğu belirli bir veri seti üzerinde yoğunlaşmakta ve bundan kaynaklı olarak farklı veri setine uygulandığında istenen performansı göstermeyebilmektedir. Dolayısıyla farklı veri setleri üzerinde uygulanarak geliştirilmiş hastalık tespit metodolojilerine ihtiyaç vardır. Literatürdeki çalışmalarda görülen bir başka husus da evrimsel sinir ağı mimarilerinin kullanımına ilişkindir. Her ne kadar ilgili derin öğrenme mimarileri resimlerden ilgili öz niteliklerin çıkarılmasında çok iyi yöntemler olsa da elde edilen öz niteliklerin tamamının kullanılması, oluşturulacak olan

sınıflandırma modelinin performansına ve verimliliğine olumlu katkı sağlaması mümkün olmayabilmektedir. Bir diğer söylemle çıkarılan özneliliklerde faydalı olanlar ve gereksiz veya alakasız olanlarla beraber bulunabilmektedir. Bu sebepten ötürü gereksiz veya alakasız olan öznelilikleri elimine eden öznelilik seçme algoritmalarının kullanılmasına ihtiyaç duyulmaktadır.

Tablo 1. Literatür Taraması

Yazarlar	Sınıf Sayısı	Kullanılan Yöntemler	En Başarılı Yöntem
Toğaçar vd. (2019)	2 (Zatürre, Normal)	AlexNet, Karar Ağaçları, Destek Vektör Makinesi, Lineer Regresyon, Doğrusal Ayrımçılık Analizi, En Yakın Komşu	AlexNet-SVM
Minaee vd. (2020)	2 (Covid-19, Normal)	SqueezeNet, ResNet50, ResNet18, DenseNet-161	SqueezeNet
Rahman vd. (2020)	3 (Bakteriyel Zatürre, Zatürre, Normal)	AlexNet, ResNet18, Squeezenet, DenseNet201	DenseNet201
Wu vd. (2020)	2 (Covid-19, Zatürre)	ResNet50, Multi-View Fusion (MVF), Single-View Fusion (SVF)	ResNet50- multi-view fusion (MVF)
Özcan, (2020)	4 (Covid-19, Zatürre, Bakteriyel Zatürre, Normal)	GoogLeNet, ResNet18, ResNet50	ResNet50- Grid Search
Özcan, (2021)	3 (Covid-19, Zatürre, Normal)	Single Layer-Based (SLB), Feature Fusion-Based (FFB)	Feature Fusion-Based (FFB)
Alqudah vd. (2020)	2 (Covid-19, Normal)	Evrişimli Sinir Ağı (CNN), Destek Vektör Makinesi (SVM), Rastgele Orman (RF)	Evrişimli Sinir Ağı (CNN)
Jaiswal vd. (2020)	2(Covid-19, Normal)	VGG-16, InceptionResNet, ResNet-152V2, DenseNet-201	DenseNet-201
Wang vd. (2020)	3 (Covid-19, Zatürre, Normal)	VGG19, ResNet50, COVID-Net	COVID-Net
Hemdan vd. (2020)	2(Covid-19, Normal)	VGG19, DenseNet201, InceptionV3, MobileNetV2, ResNetV2, Xception, InceptionResNetV2, CovidX-Net	CovidX-Net -VGG19, CovidX-Net -DenseNet201

Sethy ve Behera (2020)	2 (Covid-19, Normal)	Xception, ResNet-18, ResNet-50, ResNet-101, InceptionV3, InceptionResNetV2, GoogLeNet, DenseNet-201 VGG16, VGG19, AlexNet, Destek Vektör Makinesi (SVM)	ResNet50-SVM
Nayak vd. (2021)	2(Covid-19, Normal)	Inception-V3, AlexNet, ResNet34, VGG-16, GoogleNet, MobileNet-V2, ResNet50, SqueezeNet	ResNet34
Eryılmaz ve Karacan (2021)	4 (Covid-19, Zatürre, Normal, Opasite)	MobileNetV2, NASNetMobile, Xception, DenseNet121	DenseNet121
Ismael ve Şengür (2021)	2 (Covid-19, Normal)	ResNet18, ResNet50, ResNet101, VGG16, VGG19	ResNet50-Linear Çekirdek Fonksiyonlu SVM
Alhudhaif vd. (2021)	2 (Covid-19, Zatürre)	ResNet18, SqueezeNet, DenseNet201	DenseNet201
Bozkurt (2021)	3 (Covid-19, Zatürre, Normal)	AlexNet VGG16, VGG19, InceptionV3, DenseNet121, ResNet50, InceptionV3, Derin evrişimli ağ modeli, Sığ-evrişimli ağ modeli, Derin-tam bağlı ağ modeli ve sığ tam bağlı ağ modeli	DenseNet121
Narin vd. (2021)	2 (Covid-19-Normal), 2 (Covid-19-Zatürre), 2(Covid-19-Bakteriyel zatürre)	ResNet50, ResNet101, ResNet152, InceptionV3, Inception-ResNetV2	ResNet50
Sünnetci vd. (2021)	3 (Covid-19, Zatürre, Normal)	AlexNet	AlexNet- Cubic Destek Vektör Makinesi
Demir ve Yılmaz (2021)	2 (Covid19, Normal)	Basit-CNN, VGG16	VGG16

Moulaei vd. (2022)	2 (Hayatta Kalan, Ölüm)	Rastgele Ormanlar, XGBoost, K-en yakın komşu, Çok Katmanlı Algılayıcılar, Lojistik Regresyon, J-48 karar ağacı, Naive bayes	Rastgele Ormanlar
-----------------------	----------------------------	---	-------------------



İKİNCİ BÖLÜM

YAPAY ZEKA TEKNİKLERİ

2. 1. Yapay Zeka

Yapay zeka, makinelerin insan zekasını taklit edebilme sürecini kapsamaktadır. Başka bir ifade ile insanoğlunun zekasını kullanarak yaptığı işlerin otomatikleştirilme çabasıdır. Yapay zeka multidisipliner bir yaklaşım olup içerisinde bilgisayar bilimleri, matematik, sinir bilimi, felsefe ve dilbilim gibi disiplinleri barındırır. Yapay zeka kavramı ilk kez Alan Turing'in "Computing Machinery and Intelligence" makalesini yayınlaması ile gündeme gelmiştir. Turing bu çalışmasında "Makineler düşünebilir mi?" sorusuna dikkat çekmiştir. Alan Turing bunun yanı sıra Turing testini geliştirmiştir. Testin işleyişi şu şekildedir. Testi yapan kişi iki farklı katılımcıya sorular yöneltir ve katılımcılardan alınan cevapların makine veya insana ait olduğu ayrımı yapılmaya çalışılır. Eğer testi yapan kişi bu ayrımı yapamıyorsa yapay zeka testi geçmiş olarak kabul edilmektedir. Yapay zeka kavramını zayıf yapay zeka ve güçlü yapay zeka olarak ikiye ayırmak mümkündür. Zayıf yapay zeka, kodlanan programın dışına çıkmayan sistemleri ifade eder. Güçlü yapay zeka ise insan beynini ve hareketlerini birebir taklit edebilen sistemleri ifade etmektedir (Pirim, 2006: 82-83).

Yapay zekanın tarihçesindeki önemli olaylar kısaca şunlardır:

- 1950- Turing testi
- 1956- Dartmouth Konferansında "Yapay Zeka" kavramının ilk defa kullanılması
- 1957- İlk yapay sinir ağının geliştirilmesi
- 1966- İlk mobil robot Shakey
- 1989- Evrimsel sinir ağlarının geliştirilmesi
- 1997- Dünya satranç şampiyonu Garry Kasparov, IBM tarafından geliştirilen Deep Blue'a yenildi. Bu olay bir makinenin insan zekasına karşı ilk meydan okumasıdır.
- 2005- İlk otonom araç

- 2011- IBM, bilgi yarışmasına katılacak bilgisayar olan Watson'u geliştirmiştir. Watson yarışmanın en başarılı katılımcısı iki kişi ile yarışmış ve iki yarışmacıyı da yenmiştir. İnsanın bir makineye karşı olan ilk yarışı olarak tarihe geçmiştir.
- 2011- İlk sesli kişisel asistan Siri
- 2016- Google tarafından geliştirilen AlphaGo Go şampiyonlarını yenilgiye uğratmıştır.
- 2016- Sophia isimli ilk insansı robot geliştirilmiştir.

Yapay zeka günümüzde hala gelişmeye ve bir çok alanda kullanılmaya devam etmektedir. Yapay zekanın kullanım alanlarını özetleyecek olursak eğitim, sağlık, bankacılık işlemleri, otonom araçlar, siber güvenlik sistemleri, öneri sistemleri, kişisel sesli asistanlar vb. olarak sıralayabiliriz.

2. 2. Makine Öğrenmesi

Bir problemin makinede çözümlenebilmesi için algoritmasının oluşturulması gerekir. Algoritma girilen girdilerin çıktıya dönüştürülme sürecini ifade etmektedir. Örneğin, girilen her bir sayıyı 100'e bölecek olan algoritmanın girdisi sayı kümesi, çıktısı da girilen her bir sayının 100'e bölündüğü bir dizi olacaktır.

Problemler algoritmalar yardımı ile çözülebilir ancak bazı problemlerin çözümü için belirli bir algoritma yoktur. Bu aşamada devreye "makine öğrenmesi" kavramı girmektedir. Örneğin, hasta birey ve sağlıklı bireylerin ayrımının yapılmasında belirli bir algoritma yoktur. Bireylerin bilgileri girdi verileridir ve bu girdilere bağlı olarak çıktı belirlenmektedir. Programlanması zor ya da mümkün olmayan problemler verilerden öğrenerek çözüm sunmaktadır.

Makine öğrenmesi kavramı, ilk kez Arthur Samuel tarafından 1959'da ele alınmıştır (Samuel, 1959). Yapay zeka kavramının alt kümesi olarak değerlendirilen makine öğrenmesi, insanoğlunun çevresindekileri gözlemleyerek mantıksal muhakeme yapabilme ve öğrenebilme yeteneklerinin makineler tarafından taklit edilmesidir. Makine öğrenmesi yöntemlerini Tablo 2'de görüldüğü üzere gözetimli öğrenme (supervised learning), gözetimsiz öğrenme (unsupervised learning) ve pekiştirmeli öğrenme (reinforcement learning) olarak

ayırmak mümkündür. Gözetimli öğrenmede sınıflandırma, regresyondan ve varsayımdan, gözetimsiz öğrenmede ise kümeleme ve tahminden yararlanılmaktadır. Pekiştirmeli öğrenmede ise karar verme yöntemlerinden yararlanılmaktadır.

Tablo 2. Makine Öğrenmesi Teknikleri (Qiu vd., 2016:3)

Öğrenme Tipi	Veri İşleme Yöntemleri	Uzaklık Ölçütü	Öğrenme Algoritmaları
Gözetimli Öğrenme	Sınıflandırma Regresyon Varsayım	Hesaplamalı sınıflandırıcılar	Destek vektör makinesi
		İstatistiksel sınıflandırıcılar	Naive bayes, Hidden markov modeli, Bayes networks
		Bağılantılı sınıflandırıcılar	Sinir Ağları
Gözetimsiz Öğrenme	Kümeleme Tahmin Birliktelik Kuralları Analizi	Parametrik	K-means, Gaussian mixture modeli
		Non-parametrik	X-Means, Dirichlet process mixture modeli
Pekiştirmeli Öğrenme	Karar Verme	Model-free	Q öğrenme, R öğrenme
		Model tabanlı	TD öğrenme, Sarsa öğrenme

2. 2. 1. Makine Öğrenmesi Teknikleri

Makine öğrenmesi teknikleri Şekil 5'ten görüleceği üzere temel olarak üç grupta incelenmektedir. Bu gruplar alt bölümlerde izah edilecektir.

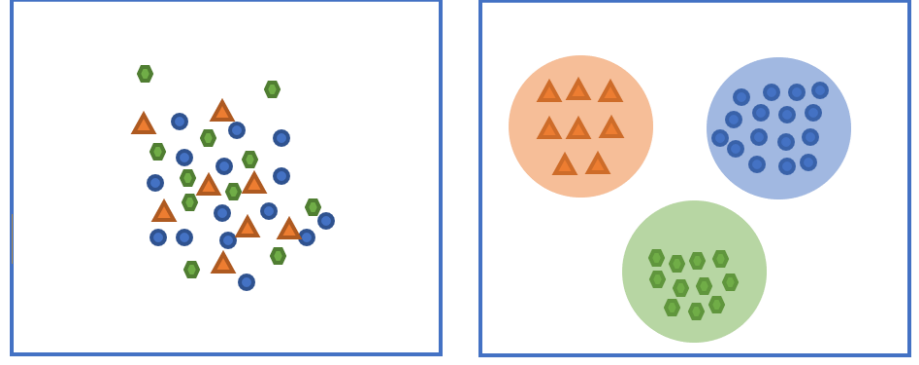


Şekil 5. Makine Öğrenmesi Teknikleri

2. 2. 1. 1. Gözetimsiz Öğrenme

Makine öğrenmesi temelli tekniklerden biri olan gözetimli öğrenme, ham etiketlenmemiş verileri kullanarak aralarındaki gizli bağlantıyı ortaya çıkartma sürecini kapsamaktadır. Çıktı verilerini kullanmadan sadece girdi verileri ile öğrenme gerçekleştirilmektedir. Başka bir ifade ile bağımlı değişkenler kullanılmadan bağımsız değişkenlerle öğrenmenin gerçekleşmesidir (Kutlugün, 2017: 31).

Kümeleme ve birliktelik analizi algoritmaları gözetimsiz öğrenme içerisinde yer almaktadır. Kümeleme, veri setindeki birbirine benzeyen verilerin gruplara ayrılması işlemidir (Şekil 6). Kümeleme için kullanılan bazı algoritmalar K-ortalamlar (K-means), K-ortanca (K-medians), OPTICS ve Mean Shift olarak ifade edebiliriz. Birliktelik analizi algoritmaları ise, büyük veri setlerindeki ilginç ilişkilerin bulunmasını ve verilerin beraber bulunma durumlarını irdeler. Birliktelik analizi ile geçmiş verilerden elde edilen birlikte bulunma durumları doğrultusunda geleceğe yönelik çalışma yapılmasına olanak sağlanmaktadır. Birliktelik analizi için kullanılan bazı algoritmalar Apriori algoritması, GRI, Carma ve Eclat olarak söylenebilir.



Şekil 6. Gözetimsiz Öğrenme

2. 2. 1. 1. 1. K-ortalamalar (K-means)

K-ortalamalar en eski kümeleme algoritmaları içerisinde yer almaktadır ve 1967 yılında MacQueen tarafından geliştirilmiştir (MacQueen, 1967). K-means algoritması her bir verinin yalnızca tek bir kümeye ait olabilmesine imkan sağlamaktadır. Bundan dolayı keskin (partitional) bir kümeleme algoritmasıdır. Genel mantık şu şekildedir; n adet veriden oluşan veri seti, belirlenen k adet kümeye bölünmektedir. K-ortalamalar algoritması rastgele belirlenen k adet küme merkez noktası ile başlar ve küme içerisindeki her veri kendisine en yakın olan küme merkezine atanır. Küme merkezlerinin uzaklığı hesaplanırken Öklid, Minkowski ve Manhattan gibi yöntemlerden faydalanılmaktadır. Küme merkezleri kendine ait noktaların ortalaması alınarak hesaplanmaktadır. Bu işlem merkezler değişmeyinceye kadar tekrar edilir. Gerçekleştirilen kümeleme sonucunda küme içi benzerliğin yüksek, küme dışı benzerliğin ise düşük olması beklenir (Sarıman, 2014: 195).

K-ortalamalar algoritmasının adımları şu şekildedir:

Adım 1: Küme sayısı “ k ” belirlenir.

Adım 2: Küme merkezleri belirlenir.

Adım 3: Veri setindeki her bir elemanın küme merkezine olan uzaklığı hesaplanır ve en yakın küme merkezine atanır.

Adım 4: Her bir kümedeki elemanların ortalamaları hesaplanarak yeni küme merkezleri güncellenir.

Adım 5: 3. ve 4. adımlarda durdurma kriteri sağlanıncaya kadar işlemler tekrar eder.

2. 2. 1. 1. 2. Apriori Algoritması

Apriori algoritması, 1994 yılında Agrawal ve Srikant (Agrawal ve Srikant, 1994) tarafından geliştirilmiştir. Apriori algoritmasında bilgiler bir önceki adımdan alındığı için önceki anlamında kullanılan “apriori” kelimesinden türetilmiştir (Agrawal vd., 1993: 490). Veri seti içindeki yaygın olarak kullanılan verilerin keşfedilmesini sağlar. Genel olarak veri setindeki verilerin birbirleriyle olan bağlantıları ve ilişkilerini ortaya çıkaran bir algoritmadır. Veri setindeki sıkça geçen verilerin bulunabilmesi için birçok kez veri setinin taranması gerekir.

Apriori algoritmasının adımları şu şekildedir:

Adım 1: Minimum destek sayısı ve minimum güven değerinin belirlenmesi,

Adım 2: Öğe kümeler içerisindeki her bir ögenin destek değerinin bulunması,

Adım 3: Minimum destek değerinden daha düşük desteğe sahip olan öğelerin devre dışı bırakılması,

Adım 4: Elde edilen tekli birliktelikler dikkate alınarak ikili birlikteliklerin oluşturulması,

Adım 5: Minimum destek değerinden düşük olan öğe kümelerin çıkartılması,

Adım 6: Üçlü birlikteliklerin oluşturulması,

Adım 7: Üçlü birlikteliklerden minimum destek değerini geçenlerin dışındakilerin çıkarılması, ve

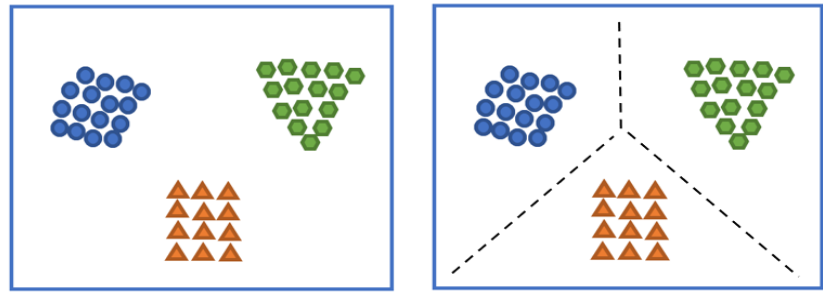
Adım 8: Üçlü birlikteliklerden birliktelik kurallarının çıkarılması.

2. 2. 1. 2. Gözetimli Öğrenme

Makine öğrenmesi temelli tekniklerden biri olan gözetimli öğrenmenin eğitim setinde etiket bilgisinin yer aldığı veriler vardır. Makinelerin etiketlenmiş olan verilerden öğrenmesi ve bundan yola çıkarak genelleme yapabilmesi beklenir (Şekil 7). Bir başka deyişle bağımlı ve bağımsız değişkenlerin bir arada bulunması durumudur. Bağımlı değişken, amacımızı temsil eder. Örneğin, ev fiyatlarını tahmin eden bir sistem tasarladığımızı düşünürsek, bağımlı değişkenimiz ev

fiyatları ve bağımsız değişkenler ise ev fiyatlarına etki eden binanın yaşı, oda sayısı, bulunduğu kat, balkon sayısı gibi faktörler olacaktır.

Sınıflandırma ve regresyon algoritmaları gözetimli öğrenme içerisinde yer alır. Sınıflandırma, verinin daha önceden belirlenmiş sınıfların hangisine ait olduğunun belirlenmesini sağlayan algoritmalar. En sık kullanılan sınıflandırma algoritmaları olarak destek vektör makinesi (SVM), K-en yakın komşu (kNN), karar ağaçları, Naive Bayes ve yapay sinir ağları şeklinde söyleyebiliriz. Regresyon ise sürekli ya da gerçek bir değer olan çıktı değişkenini tahmin etmek için kullanılır. Başka bir ifade ile aldığı girdilere bağlı olarak sürekli çıktı üretmektedir. Örneğin, kişilerin boyunu tahmin etmek, soğuk ya da sıcaklık tahmini ve ev fiyat tahmini gibi problemlerin çözümünde tercih edilebilir. Regresyon için kullanılan bazı algoritmaları ifade edecek olursak basit doğrusal, Lasso, karar ağacı, polinomsal ve lineer regresyon olarak söyleyebiliriz.



Şekil 7. Gözetimli Öğrenme

2. 2. 1. 2. 1. K-en yakın komşu (kNN)

K-en yakın komşu algoritması ilk olarak 1951'de Fix ve Hodges (Fix ve Hodges, 1951) daha sonrasında ise Cover ve Hart (Cover ve Hart, 1967) tarafından geliştirilen parametrik olmayan gözetimli öğrenme yöntemleri içerisinde yer alır. Hem sınıflandırma hem de regresyon problemlerinin çözümünde kullanılabilmesinden dolayı sıklıkla tercih edilmektedir. kNN algoritmasında sınıflandırma yapılırken çıktı bir sınıf üyeliğidir. Eğer $k=1$ ise, veri en yakın komşunun sınıfına atanır. kNN algoritmasında regresyon yapılırken çıktı verinin özellik değeridir. Bu değer k tane en yakın komşunun değerinin ortalamasıdır. Eğitim veri setindeki verilerin tamamını kullanır. Eğitim veri setinden ayırt edici bir işleve sahip olmadığı için tembel bir öğrencidir. Eğitim

verilerini ezberler bundan dolayı da öğrenme aşaması yoktur. Genel olarak, benzer verileri aradığımız problemlerin çözümünde kullanıldığını söyleyebiliriz.

K-en yakın komşu algoritmasının adımları şu şekildedir:

Adım 1: Komşuların k sayısını seçilir.

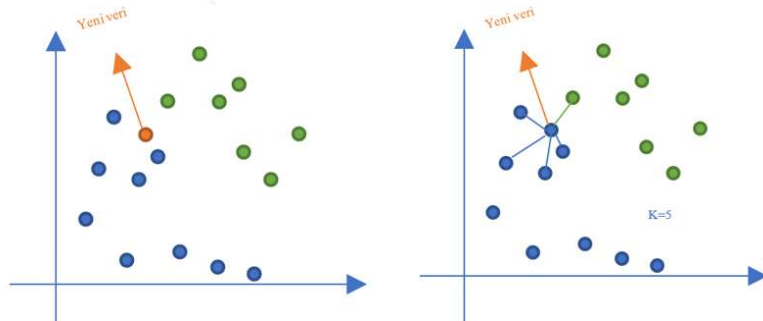
Adım 2: k sayıda komşunun Öklid mesafesi hesaplanır.

Adım 3: Hesaplanan Öklid mesafesine göre en minimum uzaklığa bağlı olarak k en yakın komşular alınır.

Adım 4: En yakın komşu sınıfındakilerin sayısı belirlenir.

Adım 5: Yeni veriler, belirlenen komşu sayısının maksimum olduğu sınıfa atanır.

Örnek verilecek olursa, Şekil 8’de görüldüğü üzere $k=5$ seçtiğimizi varsayalım. Yeni verinin en yakın komşuları uzaklık mesafelerine göre belirlenir ve daha sonrasında en yakın komşu sayısı hangi sınıfta fazla ise o sınıfa atanır. Şekil 8’de görüleceği üzere mavi sınıftan 4 ve yeşil sınıftan 1 komşu olduğu için yeni veri mavi sınıfa atanmıştır.



Şekil 8. K-En Yakın Komşu Algoritması

2. 2. 1. 2. 2. Destek Vektör Makinesi (SVM)

Destek vektör makinelerinde temel amaç, 2 ya da daha fazla sınıfı en iyi şekilde ayırabilmektir. Hem doğrusal hem de doğrusal olmayan problemlerin çözümünde kullanılabilmektedir. Doğrusal problemlerin çözümünde hiper düzlem ile sınıflandırma yapılabilmektedir. Doğrusal olmayan problemlerin çözümünde ise 2 boyutlu düzlemde boyut artırma işlemi yapılarak sınıflandırma yapılabilmesi mümkündür. Destek vektör makinesi algoritmasından Bölüm 3 ‘te daha detaylı olarak bahsedilecektir.

2. 2. 1. 2. 3. Naive Bayes

Naive Bayes algoritmasının temeli Thomas Bayes tarafından oluşturulan Bayes teoremine (Bayes, 1763) dayanmaktadır. Sınıflandırma için kullanılan basit, hızlı ve olasılıkçı bir algoritmadır. Naive Bayes ile veri setindeki sınıflandırılmış veriler kullanılarak, veri setine yeni bir veri dahil olduğunda mevcut sınıflardan herhangi birine dahil olma olasılığı hesaplanır. Bayes teoremi, ön bilgiler ile bir hipotezin olasılığı belirlemek için kullanılır ve koşullu olasılığa bağlıdır. Başka bir ifade ile önceki olayla ilgili bilgi verildiğinde, olayların gerçekleşme olasılığını bulabilme diyebiliriz.

$P(y|X)$; X olayının gerçekleşmesi koşulu ile y olayının gerçekleşme olasılığı şu şekilde hesaplanır:

$$d_w P(y|X) = \frac{P(y|X) * P(y)}{P(X)} \quad (1)$$

Burada $P(X|y)$ y olayının gerçekleşmesi koşulu ile X olayının gerçekleşme olasılığıdır, $P(y)$ y olayının önsel olasılığıdır, ve $P(X)$; X olayının önsel olasılığıdır.

Veri setindeki öznitelikler Denklem 2’de gösterildiği şekilde ifade edilmektedir.

$$X = (x_1, x_2, \dots, x_n) \quad (2)$$

Denklemdaki y ifadesi hedef değişkeni yani sınıf/etiketi temsil etmektedir. Denklem 1’e X yerleştirildiğinde elde edilen formül Denklem 3’te ifade edilmektedir. Başka bir ifade ile X’ e göre y’nin sınıfı Denklem 3’te ifade edildiği gibi hesaplanmaktadır.

$$P(y|(x_1, x_2, \dots, x_n)) = \frac{P(x_1|y)P(x_2|y)\dots P(x_n|y)P(y)}{P(x_1)P(x_2)\dots P(x_n)} \quad (3)$$

Bir verinin hangi sınıfa ait olduğunu belirleyebilmek adına veri setindeki her bir sınıf için Denklem 4’te ifade edildiği şekilde hesaplama yapılır.

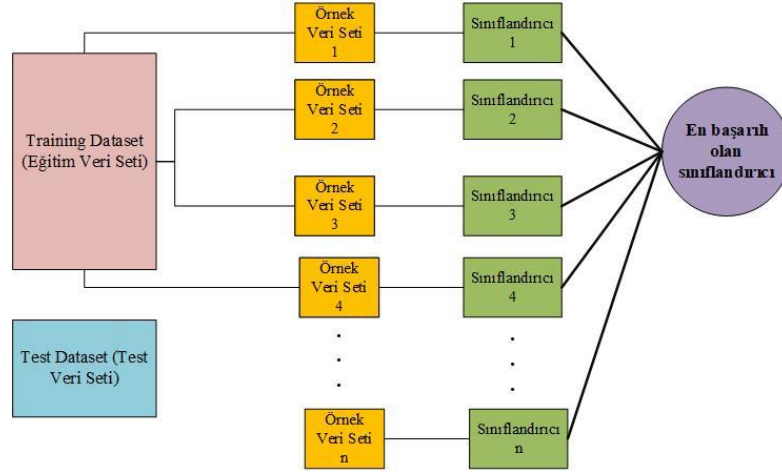
$$P(y|x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i|y) \quad (4)$$

Denklem 5 kullanılarak öznitelikleri verilen bir verinin hangi sınıfa ait olduğu bulunabilmektedir.

$$y = \operatorname{argmax}_y P(y) \prod_{i=1}^n P(x_i|y) \quad (5)$$

2. 2. 1. 2. 4. Bagging

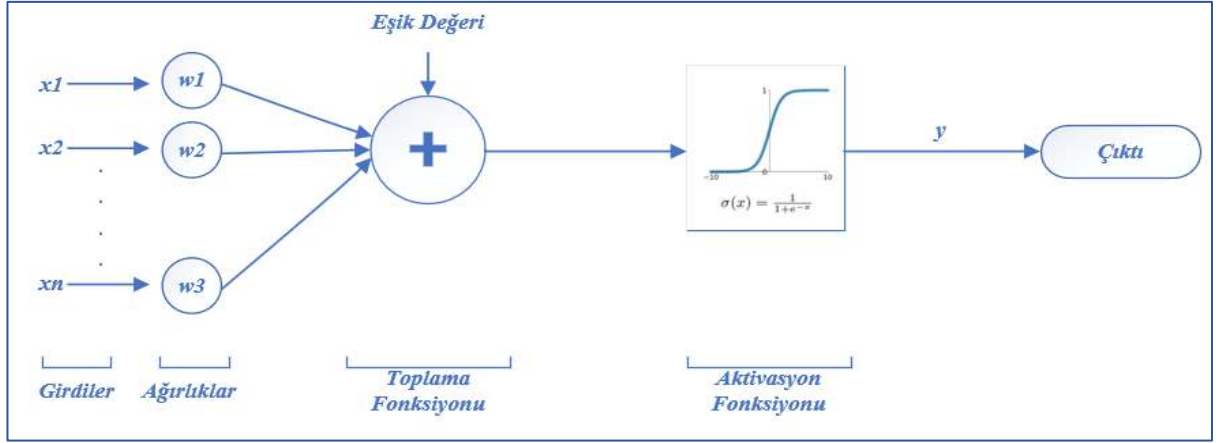
Bagging, (Breiman, 2001) tarafından geliştirilmiş bir algoritmadır. Orijinal veri seti üzerinden alt kümeler oluşturma mantığına göre çalışmaktadır. Bu alt kümelerin her birine de temel model oluşturmaktadır. Tüm modellerden gelen sonuçlar doğrultusunda nihai bir sonuç üretmektedir. Başka bir şekilde ifade edecek olursak, orijinal veri setinden rastgele alınan örneklerden oluşturulan yeni eğitim setleri kullanılarak toplu öğrenme kümesinde bulunan tüm sınıflandırıcılar teker teker eğitilir. Akabinde en başarılı olan sınıflandırıcı seçilir (Şekil 9) (Breiman, 2001: 7-21; Çölkesen, 2015: 63). Bagging yöntemi, topluluk öğrenmesi (Ensemble Learning) yöntemleri içerisinde yer alır. Model başarısının ensemble yöntem olmasından kaynaklı olduğunu söylemek mümkündür. Ensemble yöntemlerin mantığı ise, belirli bir problemi çözebilmek için birçok sınıflandırıcı arasından en uygun olanın seçilmesidir. Daha sonrasında optimizasyon yapılarak ve sınıflandırıcılardan alınan sonuçlar birleştirilerek en yüksek doğruluk değerine ulaşılması sağlanmaktadır. Sınıflandırıcıların birleştirilmesi işlemi ile hata oranının en optimum seviyeye getirilebilmesi amaçlanmaktadır. Bagging yöntemi bu gibi yeteneklerinden dolayı tek sınıflandırıcılı algoritmalara kıyasla daha iyi sınıflandırma doğruluğu elde edebilmektedir. Diğer önemli özellikleri ise, düzensiz veri setlerinde dahi başarılı sonuçlar elde edebilmesi ve aşırı öğrenmenin (overfitting) önüne geçebilmeyi amaçlayan bir yöntem olmasıdır.



Şekil 9. Bagging Yöntemi

2. 2. 1. 2. 5. Yapay Sinir Ağları

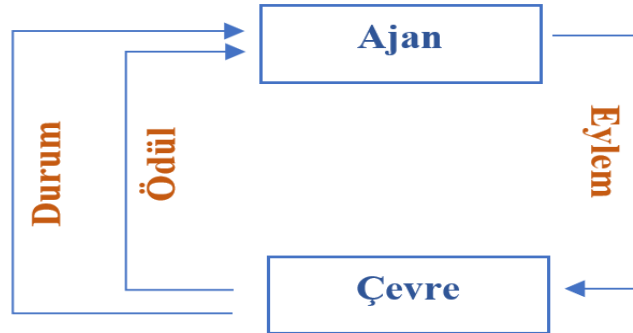
İlk yapay sinir ağı, sinir hekimi Warren McCulloch ve matematikçi Walter Pitts tarafından 1943 yılında geliştirilmiştir. Böylece yapay sinir ağlarının temelleri atılmıştır (McCulloch, Pitts, 1943). McCulloch ve Pitts' in yaptığı çalışma, yapay sinir hücreleri ile her türlü mantıksal olayın formüle edilebilmesinin mümkün olduğunu kanıtlamıştır. Yapay sinir ağları, insanoğlunun bilgi türetebilmesi, yeni bilgiler üretebilmesi ve keşfedebilmesi gibi yeteneklerini otomatikleştirmek amacıyla geliştirilen bir algoritmadır. Bu yeteneklerinden dolayı geleneksel programlama ile çözümü zor ya da çözümü mümkün olmayan problemlerin çözümlenmesinde kullanılan algoritmalarlardır. Şekil 10'da görüldüğü üzere yapay sinir ağlarının yapısında giriş katmanı, ara katman ve çıkış katmanı bulunur. Giriş katmanı, dışarıdan gelen verilerin bulunduğu katmandır. Ara katmanda girdi katmanından gelen bilgilerin işlenmesi yapılmaktadır. Bu yapılırken toplam fonksiyonu ve aktivasyon fonksiyonu kullanılabilir. Çıkış katmanında, ara katmandan gelen bilgiler işleyerek girdi setine karşılık çıktılar oluşturulur (Öztemel, 2003: 45-58).



Şekil 10. Yapay Sinir Ağları Modeli

2. 2. 1. 3. Pekiştirmeli Öğrenme

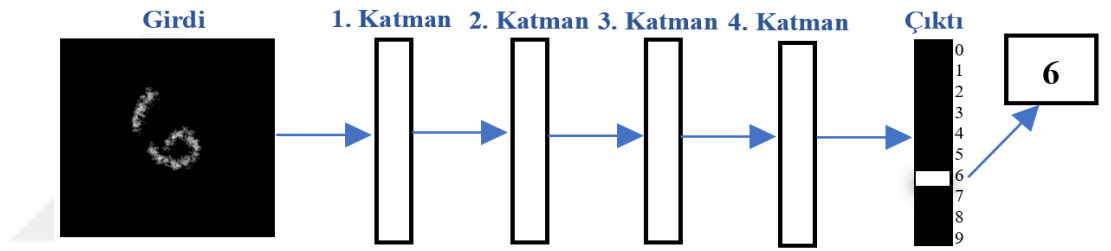
Pekiştirmeli öğrenme, herhangi bir öğreticiye gerek duyulmadan sistemin deneme yanılma yoluyla hedefine ulaşması için en uygun olan hareketleri öğrenip uygulayabilmesi temeline dayanan bir yaklaşımdır. Pekiştirmeli öğrenmede ödül ve ceza kavramları vardır. Sistemin herhangi bir olay karşısındaki hareketi ödüllendirilir ya da cezalandırılır (Karadoğan, 2014: 13-15). Şekil 11’de ifade edildiği üzere, ajan ve çevre ilişkisi gösterilmektedir. Çevre hakkında önceden bilgisi olmayan ajan, içinde bulunduğu duruma en uygun olan eylemi yapar. Bu eylem çevre tarafından değerlendirilir. Ajan aldığı karar doğrultusunda çevre tarafından cezalandırılır ya da ödüllendirilir. Ajan bir önceki eylemden aldığı ödül ya da cezayı kendi içerisinde değerlendirerek yeni durum için eylem oluşturur. Ajanın deneme yanılma yolu ile öğrenmesi gerçekleşene kadar bu döngü devam eder.



Şekil 11. Pekiştirmeli Öğrenme

2. 3. Derin Öğrenme

Derin öğrenme, art arda birbirini takip eden katmanlarda veriler işlenirken verilerin daha anlamlı hale gelmesini sağlayan makine öğrenmesinin bir alt dalıdır. Derin söylemi ile kastedilen derin öğrenmeden elde edilen derin bilgiler değil, art arda birbirini takip eden katmanlardır. Bir modelin ne kadar derin olduğu katman sayısına bağlı bir unsurdur. Modern derin öğrenme algoritmaları onlarca hatta yüzlerce katmandan meydana gelir. Şekil 12’de dört katmandan oluşan bir modelin girdi görüntüsünün hangi rakama ait olduğunu tanımlayan modelin gösterimi sunulmuştur. Modelin girdisi katmanlardan geçtikçe ya da model derinleştikçe çıktı hakkındaki bilgisi de artmaktadır. Sonuç olarak, derin öğrenme algoritmalarında çok aşamalı öğrenme gerçekleşir ve basit olan bilgi derinleştikçe daha anlamlı bilgi haline dönüşür (Chollet, 2019: 3-9).



Şekil 12. Derin Öğrenme Modeli

2. 3. 1. Derin Öğrenme Algoritmaları Kullanılarak Neler Yapıldı?

Derin öğrenme, makine öğrenmesinin oldukça eski bir alt dalı olmasına rağmen popülerite kazanması 2010 yılını bulmuştur. Makine öğrenmesi algoritmaları ile çözülmesi zor olan problemler için derin öğrenme algoritmaları kullanılmıştır:

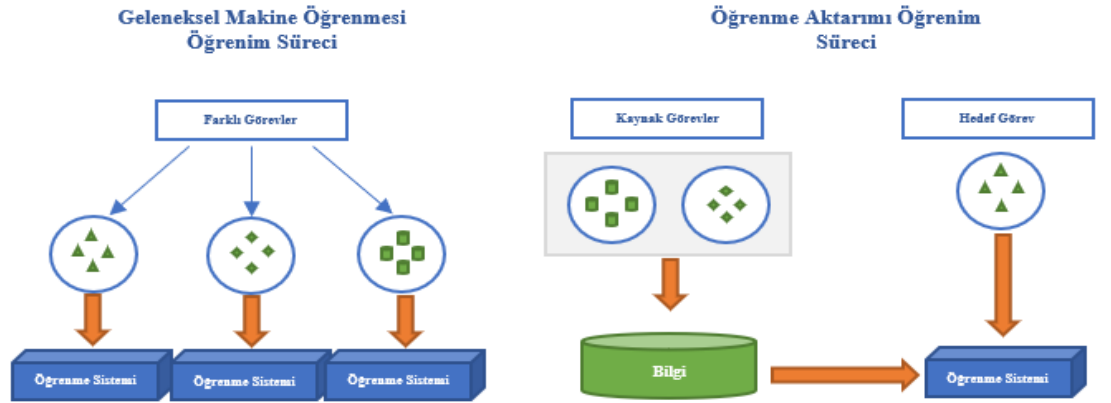
- İnsan seviyesine yakın ses tanımlama,
- İnsan seviyesine yakın görüntü sınıflandırma,
- İnsan seviyesine yakın otonom araç sürüşü,
- İnsan seviyesine yakın el yazısı,
- Amazon Alexa ve Google Now sanal asistanları,
- Sorulan sorulara yanıt verebilme yeteneği olan sistemler,
- Dijital reklamlar için daha iyi hedef kitle belirleyebilen sistemler,

- Go oynamada insan üstü seviyeye ulaşmıştır.

Ama hala derin öğrenme kullanılarak neler yapılabileceği araştırılmaktadır. Derin öğrenmenin, makine öğrenmesi ve doğal dil işlemenin ötesinde akıl yürütme gibi daha spesifik konularda kullanılabilmesi için hala araştırmalar devam etmektedir (Chollet, 2019: 12).

2. 4. Öğrenme Aktarımı (Transfer Learning)

Öğrenme aktarımı, makine öğrenmesi algoritmalarının eğitim aşamasında öğrendiği bilgiyi farklı ya da benzer sorunların çözümünde kullanabilmesidir. İnsanoğlunun doğasında öğrenme aktarımı yaklaşımı vardır. Bilinçli ya da bilinçsiz şekilde geçmişte yaşanan olaylardan yapılan çıkarımlar doğrultusunda problemler çözülmeye çalışılır. Öğrenme aktarımı tıpkı insanoğlunun zekasını kullanarak geçmiş deneyimlerinden yararlanma kabiliyetinin makine öğrenimi ile gerçekleştirilmesidir (Koçer, 2012: 9-10). Şekil 13'te görüleceği üzere makine öğrenmesi teknikleri ile öğrenim sürecinde her görev teker teker öğrenilmektedir. Öğrenme aktarımında ise, öğrenmeye sıfırdan başlamak yerine daha önceden elde edilen öğrenimler bir diğerine aktarılır. Geleneksel makine öğrenimi algoritmalarının belirgin dezavantajlarından birisi de sıfırdan oluşturulan modeller için büyük veri kümelerine ihtiyaç vardır. Bu kümelerin oluşturulması ve eğitimi zaman açısından maliyetlidir. Transfer öğrenimi yaklaşımı ile önceden eğitilmiş modellere ince ayar (fine-tuning) yapılarak daha küçük veri kümelerinden daha yüksek performans alınabilir ayrıca eğitim süreleri düşürülebilir.

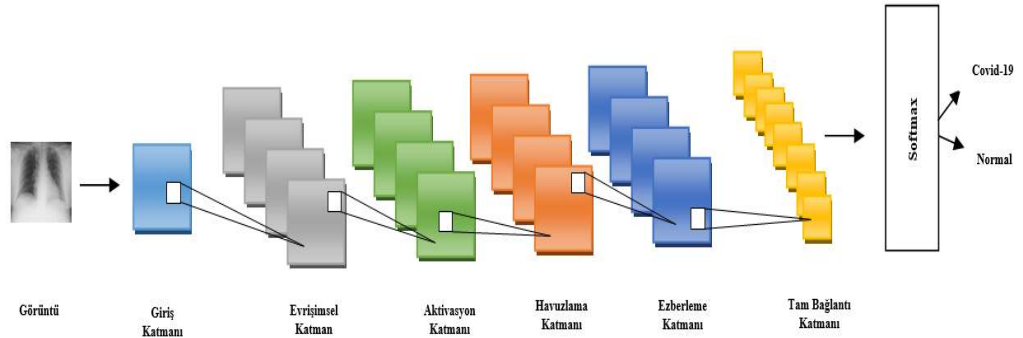


Şekil 13. Geleneksel Öğrenme ve Öğrenme Aktarımı Öğrenim Süreci

2. 5. Derin Öğrenme Algoritmaları

2. 5. 1. Evrişimli Sinir Ağları (CNN)

Klasik bir evrişimli sinir ağı modeli giriş, evrişim katmanı, aktivasyon, havuzlama, ezberleme ve tam bağlantı katmanlarından oluşmaktadır (Şekil 14). Giriş katmanı, verilerin gerekli standartlaştırma işlemleri yapılarak sinir ağına sunulabilir hale gelmesini sağlar. Evrişimsel katman, giriş verileri üzerinde dolaşacak olan filtreleme işlemlerinin yapıldığı katmandır. Aktivasyon katmanı, elde edilen değerlerin belirli bir aralığa çekilebilmesi ya da indirgenebilmesi için kullanılan katmandır. Sıklıkla tercih edilen fonksiyonlar ise ReLU, Sigmoid ve Step fonksiyonlarıdır. Havuzlama katmanı ile verilerin daha küçük boyutlara indirgenmesi sağlanır. Bu indirgemelere bağlı olarak veri kaybı yaşanabilmektedir. Sıklıkla kullanılan havuzlama fonksiyonları ise en büyük değer (max pooling), en küçük değer (min pooling) ve ortalama değer (average pooling) fonksiyonlarıdır. Ezberleme katmanı, bazı durumlarda model eğitim sırasında kullanılan verileri ezberleyebilir. Bu durumu çözebilmek için ezberleme katmanı ağı ezberlediği bu verileri ağı unutturmayı sağlamaktadır. Tam bağlantı katmanı ise verilerin tek boyutlu vektöre dönüşmesini sağlayarak sınıflandırma katmanına göndermektedir (Metlek ve Çetiner, 2021: 7-17).

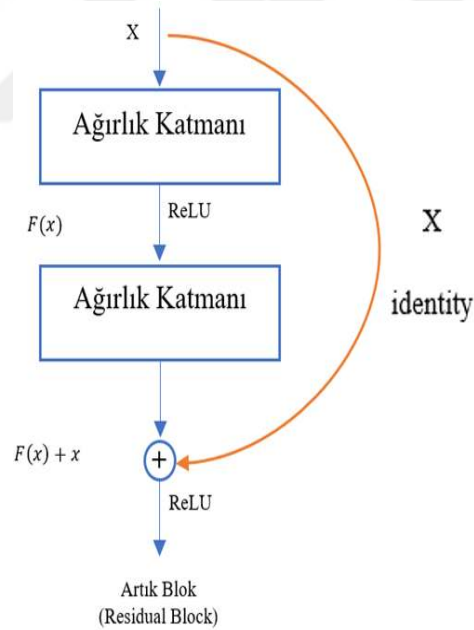


Şekil 14. Evrişimli Sinir Ağı Modeli

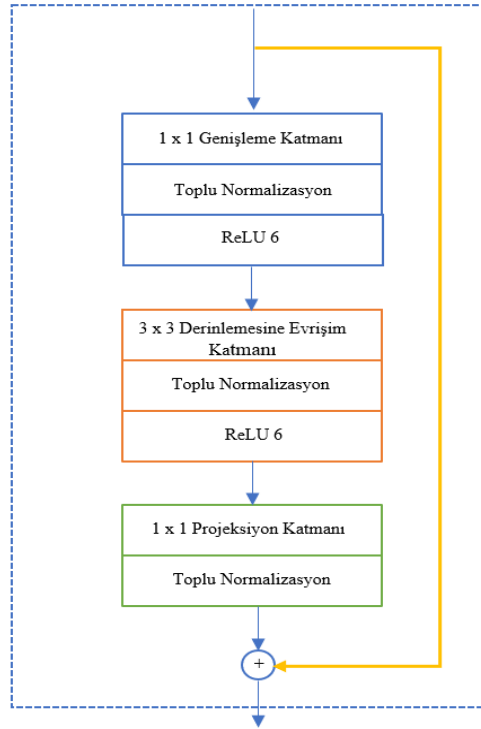
2. 5. 2. ResNet

ResNet mimarisi He, Zhang, Ren ve Sun tarafından 2016 (He vd., 2016) yılında tanıtılmıştır. ResNet, kendinden önceki ağlardan çok daha derin ve yüzlerce katmanı olan ilk ileri beslemeli sinir ağı olma özelliğine sahiptir. ImageNet veri setinde VGG mimarilerine nazaran 8 kat daha derin ağ yapısı vardır

ve 152 katmana kadar derinleşebilmektedir. ResNet mimarisinin 18, 34, 50, 101 ve 152 gibi versiyonları bulunmaktadır. Bütün ResNet modelleri aynı mantığı takip eder; tek fark ağdaki katman sayılarıdır. Şekil 15'te ifade edildiği üzere, x girdisini “+” ifadesine kadar taşıyan yapıya kısayol bağlantısı ya da artık bağlantı denmektedir. Kısayol bağlantısı, bir ya da birden fazla katmanı atlayarak ResNet mimarisinde daha kolay yayılım sağlamaktadır. ResNet kısa yol bağlantısı kullanması sebebi ile VGG mimarilerinden daha az karmaşıklığa ve daha az filtreden oluşmasıdır. Bunun yanı sıra kısa yol bağlantısı kullanılması ile expading/vanishing gradient problemlerinin önüne geçilebilir. Kısa yol bağlantısı kullanılmadan önce girdiler (x) ağırlık katmanları ile çarpılır sonrasında $F(x)$ aktivasyon fonksiyonundan geçerek çıktı $H(x)$ olarak alınır. Kısa yol bağlantısına $H(x)=F(x)+x$ diyecek olursak, son çıktı olan $F(x)$ 'i doğrudan x olarak öğrenir. Bundan dolayı $F(x)$ artık ağ (residual network) olarak adlandırılır (He vd., 2016: 771-773).



Şekil 15. ResNet Modeli



Şekil 16. MobileNetV2'nin Darboğaz Kalıntı Katmanı (Endachev vd., 2019:5299)

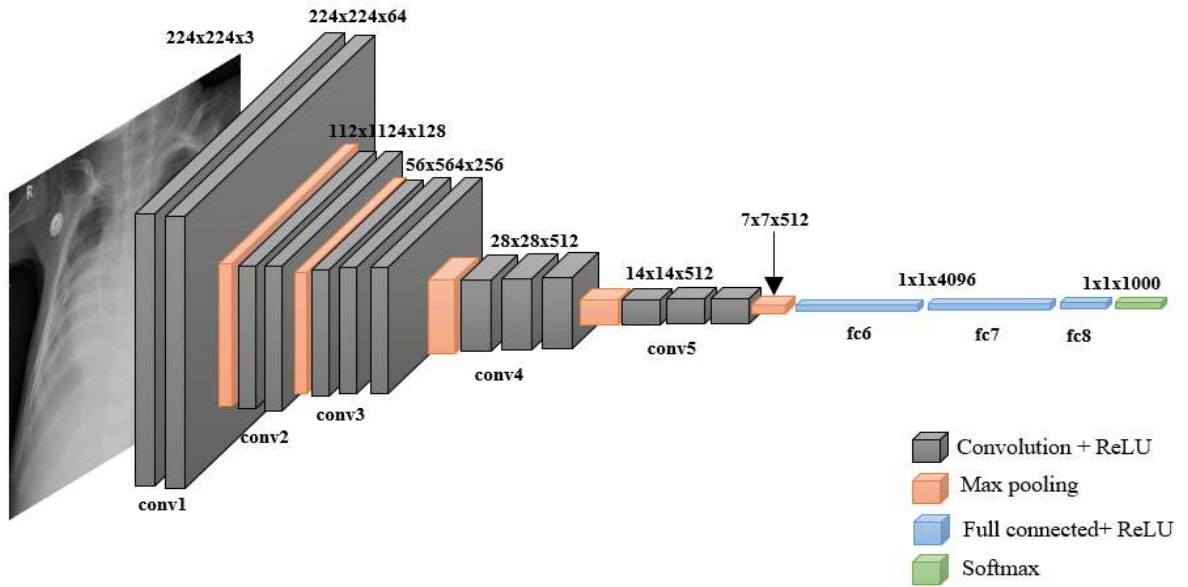
2. 5. 3. MobileNetV2

MobileNetV2 modeli Sandler, Howard, Zhu, Zhmoginov ve Chen tarafından 2018 yılında oluşturulmuştur (Sandler vd., 2018). MobileNet modelinin ilk olarak V1 versiyonu tanıtılmıştır. MobileNetV1, mobil cihazlar ve düşük hesaplama yeteneği olan tüm cihazlara uyum sağlayan derinlemesine ayrılabilir bir modeldir. MobileNetV1 modeline tersine çevrilmiş artık yapılar ve darboğazlar getirilerek MobileNetV2 versiyonu elde edilmiştir. Şekil 16'da ifade edildiği üzere MobileNetV2'nin darboğaz kalıntı katmanı genişleme, derinlemesine evrişim ve projeksiyon katmanlarının art arda dizilmesiyle oluşmaktadır. MobileNetV2 mimarisinin yapısı, giriş, evrişim, yedi adet darboğaz kalıntı katmanları, ardından iki adet evrişim ve tam bağlı katmanlardan oluşmaktadır (Sandler vd., 2018: 4510-4519; Yaşar, 2022: 60).

2. 5. 4. VGG16

VGG16 algoritması, 2014 yılında Karen Simonyan ve Andrew Zisserman (Simonyan ve Zisserman, 2014) tarafından "Büyük Ölçekli Görüntü Tanıma için Çok Derin Evrişimli Ağlar" makalesiyle tanıtılmıştır. ILSVRC-2014 yarışmasında daha iyi sonuçlar elde edebilmek için 13 evrişim 3 tam bağlı katmanlardan ve yaklaşık 138 milyon parametreden oluşan evrişimsel bir sınır

ağıdır. Girdi katmanındaki görüntüler 224x224x3 boyutundadır. Son katmanında ise sınıflandırıcı katmanı bulunmaktadır. ImageNet veri setinden 100000 görüntü ile eğitilmiştir. VGG16 modeli ImageNet veri setinde %89 oranında başarı elde etmiştir. Şekil 17’de ifade edildiği üzere, model 13 evrişim(convolution), 5 maksimum havuzlama (max pooling) ve 3 tamamen bağlı (full connected) katmanlarından oluşmaktadır. Ayarlanabilir parametre katman sayısı (convolution, full connected) 16 olduğundan modelin adı VGG16’dır. İlk bloktaki filtre sayısı 64 iken sonraki bloklarda bu sayı ikiye katlanarak 512’ye kadar ulaşmaktadır. Model birbirine tam bağlı iki gizli katman ve bir çıkış katmanı ile tamamlanmaktadır. Tamamen bağlı iki katmanda 4096, çıkış katmanında ise 1000 nöron bulunmaktadır.



Şekil 17. VGG16 Modeli

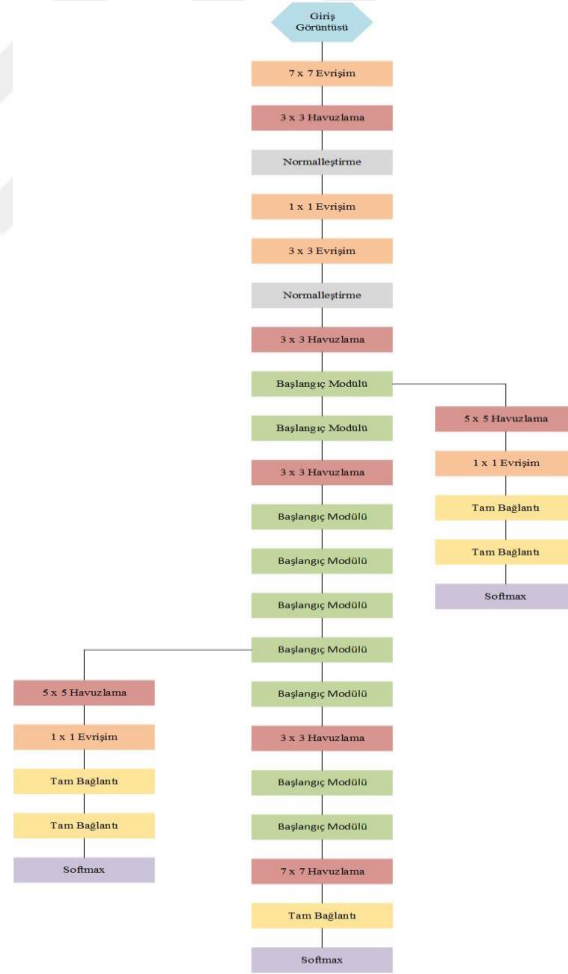
2. 5. 5. GoogLeNet

GoogLeNet ya da InceptionV1 Szegedy, Liu, Jia, Sermanet, Reed, Anguelov ve Rabinovich tarafından geliştirilen Inception Network'ün bir çeşidi olan 22 katmanlı derin evrişimli bir sinir ağıdır (Szegedy vd., 2015). Diğer mimarilerden farklı olarak art arda dizilmiş katmanlar yerine paralel katmanlar şeklinde devam eden yapısı vardır. Paralel yapı ile aşırı öğrenme (overfitting) sorununun önüne geçebilmek, hesaplama ve bellek maliyetlerini azaltabilmek amaçlanmıştır. Bunun yanı sıra AlexNet mimarisinden 12 kat daha az işlem

yüküne sahiptir. GoogLeNet mimarisi softmaxlayer, fullconnected layer, konvolüsyon, maxpooling, giriş katmanı ve çıkış katmanı gibi 144 katman içermektedir. Giriş katmanı görüntü boyutu 224x224x3'tür. Konvolüsyon katmanında 1x1, 3x3 ve 5x5 filtreler bulunmaktadır. Mimaride 3x3 boyutunda havuzlama kullanılmaktadır. GoogleNet mimarisinin yapısı Şekil 18'de ifade edildiği gibidir (Szegedy vd., 2015: 2-6; Doğan ve Türkoğlu, 2018: 15).

2. 5. 6. SqueezeNet

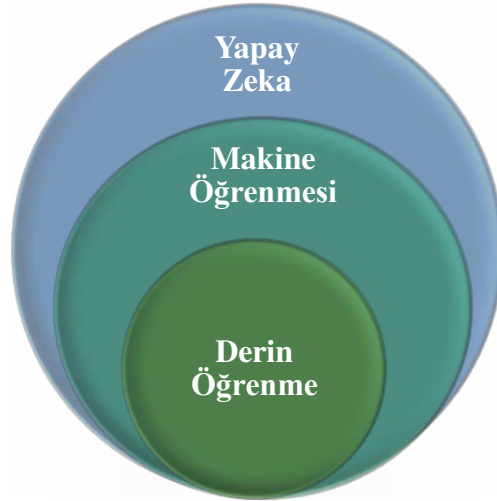
Iandola ve arkadaşları evrişimli sinir ağları mimarileri içerisinde yer alan SqueezeNet modelini 2016 yılında önermişlerdir. SqueezeNet önerilen ateşleme modüllerinden oluşmaktadır. Ateşleme modülünde genişletilmiş ve sıkıştırılmış katmanlar yer almaktadır. SqueezeNet mimarisinden Bölüm 3'te daha detaylı olarak bahsedilecektir.



Şekil 18. GoogLeNet Mimarisi (Szegedy vd., 2015: 6; Doğanalp 2022: 36)

2. 6. Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenme Kavramlarının Karşılaştırılması

Yapay zeka Şekil 19’da görüldüğü üzere makine öğrenmesi ve derin öğrenmeyi kapsayan genel bir kavramdır. Yapay zeka, makineye belirli durumlarda belirli davranışları yapabilme kabiliyetini kazandırır. Makine öğrenmesi veriden öğrenerek gizli bilginin çıkarılmasını sağlar. Derin öğrenme ise makine öğrenmesi algoritmalarının geliştirilerek son halini aldığı yapıdır. Makine öğrenmesi algoritmalarına bakıldığında bir ya da iki katmandan oluşan sığ öğrenmeli modeller yer alır. Derin öğrenme algoritmalarında ise onlarca hatta yüzlerce katmanla derin öğrenme sağlanır. Makine öğrenmesinin adımlarından biri olan öznitelik çıkarımı derin öğrenme ile birlikte otomatikleştirilmiştir. Derin öğrenmenin sağladığı bu avantaj problemlerin çözümüne daha hızlı ulaşabilmeyi mümkün kılmaktadır. Yapay zeka 1950’lerde, makine öğrenmesi 1980’lerde derin öğrenme ise 2010’lu senelerde popülerite kazanmıştır. Bu kavramlar birbirlerinin yerine kullanılsa da aynı şeyi ifade etmemektedir. Yapay zeka en genel kavram olup bunun alt sınıfları makine öğrenmesi ve derin öğrenmedir. Her makine öğrenmesi ve derin öğrenme tekniği yapay zeka kapsamında değerlendirilebilir. Ancak her yapay zeka tekniği, makine öğrenmesi ve derin öğrenme olarak nitelendirilemez.



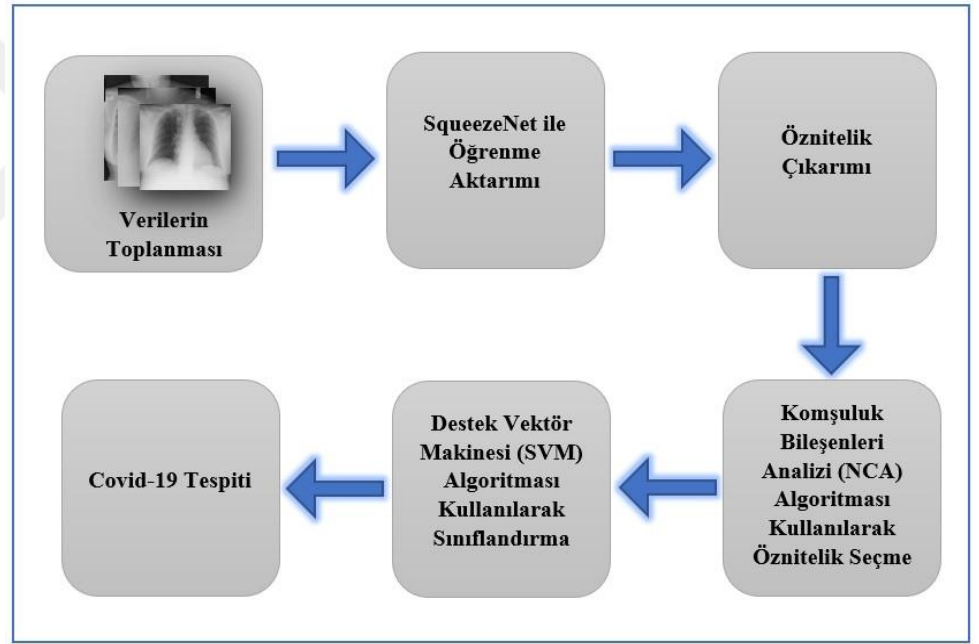
Şekil 19. Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenme İlişkisinin Şematik Gösterimi

ÜÇÜNCÜ BÖLÜM

MATERYAL VE METOT

3. 1. Önerilen Metodoloji

Önerilen derin öğrenme tabanlı Covid-19 tespit metodolojisi Şekil 20’de sunulmuştur. Şekil 20’den de görüleceği üzere öncelikli olarak veri toplama işlemi gerçekleştirilmiştir. Ardından SqueezeNet ile öğrenme aktarımı (transfer learning) işlemi uygulanarak öz niteliklerin çıkartılması sağlanmıştır. Çıkarılan öz nitelik setine daha sonra NCA öz nitelik seçme algoritması uygulanarak gürültülü öz niteliklerin elimine edilmesi sağlanmıştır. Son olarak geriye kalan alt öz nitelik seti ile SVM algoritması kullanılarak bir Covid-19 tespit modeli oluşturulmuştur.



Şekil 20. Çalışmanın Metodolojisi

3. 2. Verilerin Toplanması

3. 2. 1. Çalışmada Kullanılan Veri Setleri

Dataset 1 (Covid chestxray dataset):

Dataset 1, GitHub’tan (Cohen, t.y.) alınmıştır. Pozitif olan veya şüphelenilen hastaların akciğer röntgen görüntüleri farklı hastanelerden alınan veriler toplanarak veri seti oluşturulmuştur. Veri seti Montreal Üniversitesi Etik

Kurulu tarafından onaylanmıştır. Veri setinde Covid-19, normal (hasta olmayan) ve zatürre bireylerin X-ray görüntüleri yer almaktadır (Tablo 3).

Dataset 2 (Curated chest X-ray image dataset for Covid-19):

Dataset 2, Kaggle'dan (Sait, t.y.) alınmıştır. Veri seti 15 farklı kaynaktan alınan verilerin harmanlanmasıyla oluşturulmuştur. Veri setinde Covid-19, zatürre, bakteriyel zatürre ve normal (hasta olmayan) bireylerin X-ray görüntüleri yer almaktadır (Tablo 4).

Dataset 3 (Covid-19 radiography database):

Dataset 3, Kaggle'dan (Rahman vd., t.y.) alınmıştır. Katar Üniversitesi, Dakka Üniversitesi, Bangladeş'ten bir araştırma ekibi, Pakistan ve Malezya'dan işbirlikçiler ve doktorların yardımıyla 43 farklı kaynaktan yararlanılarak bu veri seti oluşturulmuştur. Veri setinde Covid-19 ,zatürre, akciğer opaklığı ve normal (hasta olmayan) bireylerin X-ray görüntüleri yer almaktadır (Tablo 5).

Tablo 3. Dataset 1

Sınıf Etiketi	Sayı
Covid-19	125
Normal	500
Zatürre	500
Toplam	1125

Tablo 4. Dataset 2

Sınıf Etiketi	Sayı
Covid-19	1281
Bakteriyel Zatürre	3001
Normal	3270
Zatürre	1656
Toplam	9208

Tablo 5. Dataset 3

Sınıf Etiketi	Sayı
Covid-19	3616
Akciğer Opaklığı	6012
Normal	10192
Zatürre	1345
Toplam	21165

Tablo 6. Dataset 4

Sınıf Etiketi	Sayı
Covid-19	60
Bakteriyel Zatürre	650
Normal	880
Zatürre	412
Toplam	2002

Tablo 7. Dataset 5

Sınıf Etiketi	Sayı
Covid-19	1790
Normal	1301
Toplam	3091

Dataset 4 (Covid-19 detection X-ray dataset):

Dataset 4, Kaggle'dan (Deshpande, t.y.) alınmıştır. Farklı kaynaklardan alınan veriler derlenerek hazırlanmış bir veri setidir. Veri setinde Covid-19, bakteriyel zatürre, zatürre ve normal (hasta olmayan) bireylerin X-ray görüntüleri yer almaktadır (Tablo 6).

Dataset 5 (Covid X-ray dataset):

Dataset 5, Kaggle'dan (Hemateja, t.y.) platformundan alınmıştır. Farklı kaynaklardan alınan veriler derlenerek hazırlanmış bir veri setidir. Veri setin de

Covid-19 ve normal (hasta olmayan) bireylerin X-ray görüntüleri yer almaktadır (Tablo 7).

3. 3. SqueezeNet ile Öğrenme Aktarımı

SqueezeNet Iandola ve arkadaşları tarafından 2016 yılında önerilmiştir. SqueezeNet mimarisi, AlexNet'e kıyasla 50 kat daha az parametre kullanmasına rağmen aynı işlem kapasitesine ve aynı sınıflandırma doğruluğuna sahiptirler. Bunun yanı sıra model sıkıştırma teknikleri ile SqueezeNet 0,5 MB'den daha az veya AlexNet'ten 510 kat daha küçük olacak şekilde sıkıştırılabilmektedir. Böylece bellekte daha az yer kaplayan ve AlexNet'e göre daha kısa öğrenme süresi olan bir mimaridir. Günümüzde bu gibi olumlu kazanımlarından dolayı sıklıkla kullanılmaktadır (Iandola vd., 2016: 1-4; Polat ve Özerdem, 2021: 194).

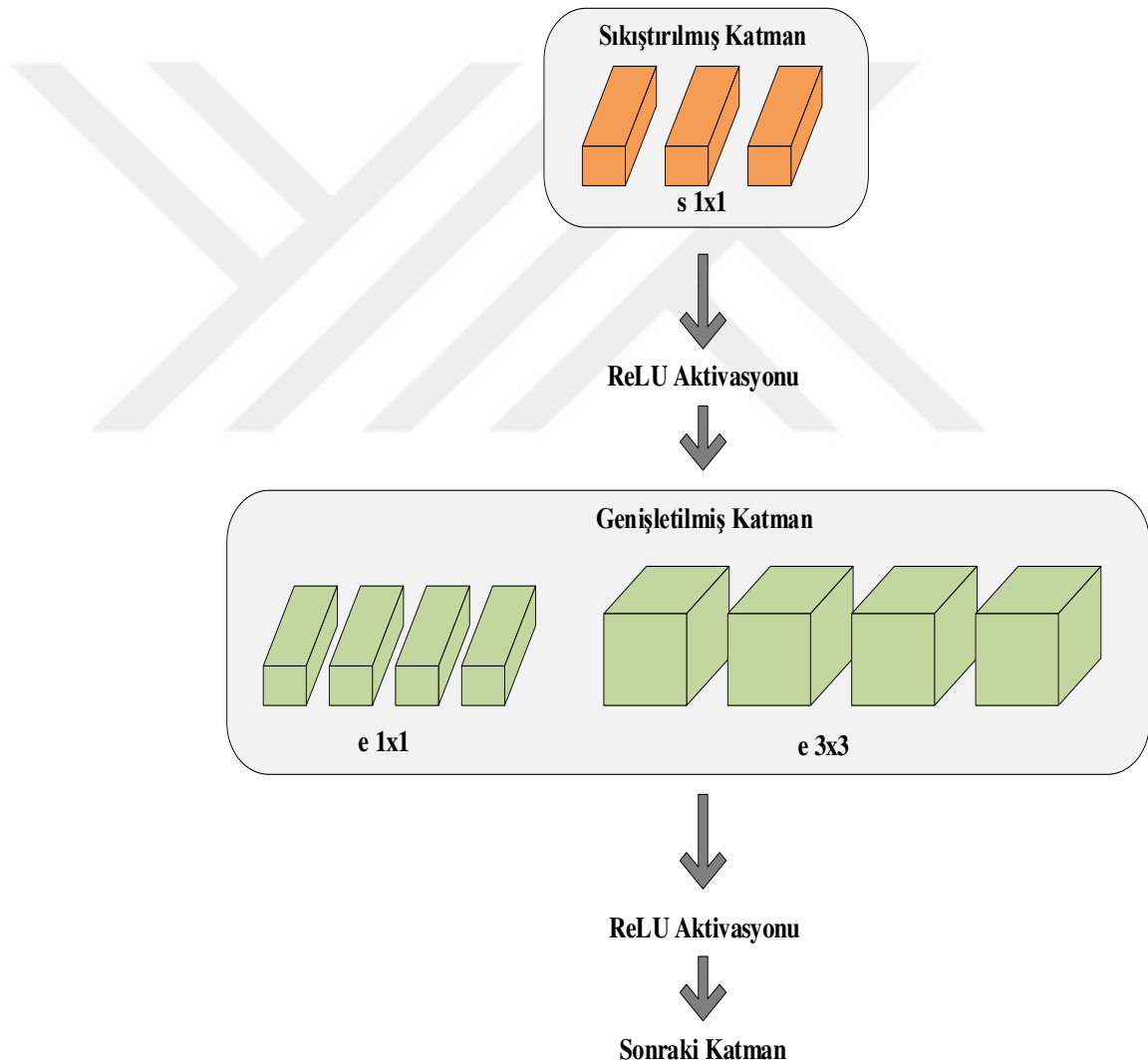
SqueezeNet mimarisinin tasarımında 3 strateji üzerinde durulmaktadır. Bunlar;

- **3x3 filtrelerin 1x1 filtrelerle değiştirilmesi:** Belirli sayıda evrişim kullanılması gerektiğinde, 1x1 filtre 3x3 filtreden 9 kat daha az parametreye sahip olduğundan, bu filtrelerin çoğu 1x1 seçilebilmektedir (Iandola vd., 2016: 3).

- **Giriş kanallarının sayısının 3 x 3 filtrelere indirgenmesi:** Tamamen 3x3 filtrelerden oluşan evrişim katmanı düşünelim. Bu katmandaki toplam parametre sayısı (giriş kanal sayısı) * (filtre sayısı) * (3x3) şeklindedir. Bu nedenle, bir evrişimli sinir ağında az sayıdaki toplam parametreyi korumak için, yalnızca 3x3 filtrelerin sayısını indirmek değil, aynı zamanda giriş kanallarının sayısını 3x3 filtrelere indirmek de önemlidir. Sıkıştırma katmanları kullanılarak giriş katmanlarının sayısı 3x3 filtrelere indirgenmektedir (Iandola vd., 2016: 3).

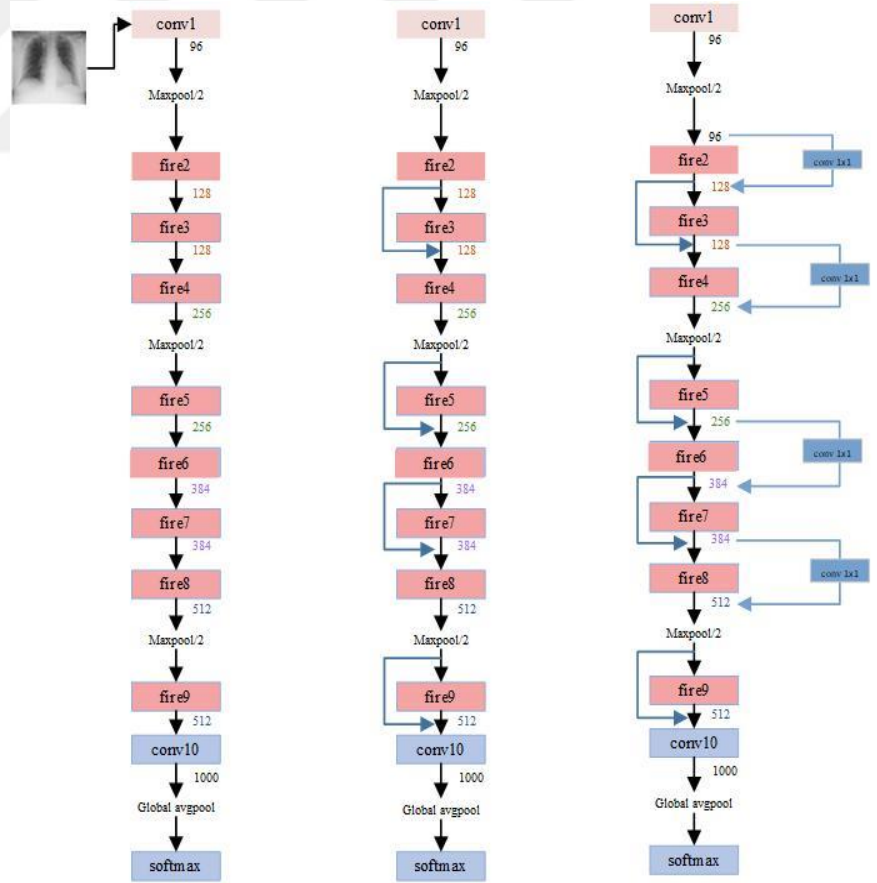
- **Evrişimsel katmanlarının büyük aktivasyon haritalarına sahip olması için ağda geç örnekleme yapılması:** Bir CNN ağında, her evrişimsel katmanı en az 1x1 olmak üzere etkinleştirme haritası çıktısı üretmektedir. Bu etkinleştirme haritalarının yüksekliği ve genişliği giriş verilerinin boyutu ve indirgenen katman seçimi ile kontrol altına alınabilir. İndirgeme genel olarak evrişimsel veya havuzlama katmanlarında iterasyon (adım) sayısının 1'den büyük seçilmesi ile yapılır. Eğer ilk katmanlarda, ağdaki iterasyon (adım) sayısı büyük seçilmiş olursa çoğu katmanın çıkışındaki etkinleştirme haritası daha küçük olur.

Eğer ilk katmanlarda, ağıdaki iterasyon (adım) sayısı 1'den küçük seçilmiş ise ve ilerleyen katmanlarda 1'den büyük iterasyon sayısı seçilirse etkinleştirme haritalarının boyutu daha geç indirgenmektedir. Sonuç itibariyle gecikmeli alt örneklemeden dolayı büyük etkinleştirme haritaları daha yüksek sınıflandırma doğruluğuna ulaşmaktadır. Birinci ve ikinci stratejilerde doğruluğu koruyup, parametre sayısını azaltmakla ilgiliyken, üçüncü strateji sınırlı bir parametre kullanıldığında doğruluğu en üst düzeye çıkarma ile ilgilenir. Bu üç stratejinin düzgün bir şekilde uygulandığı modüle ise yangın modülü (fire module) denmektedir (Kapucuoğlu, 2022: 32-33).



Şekil 21. Ateşleme Modülündeki Evrişim Filtrelerinin Organizasyonu. Bu örnekte; $s1x1=3$ $e1x1=4$ ve $e3x3 = 4$ olarak ayarlanmıştır.

SqueezeNet önerilen ateşleme modüllerinden oluşmaktadır. Ateşleme modülünde genişletilmiş ve sıkıştırılmış katmanlar yer almaktadır. Genişletilmiş katman, $1 \times 1(e_{1 \times 1})$ $3 \times 3(e_{3 \times 3})$ filtrelerinden oluşmaktadır. Sıkıştırılmış katman ise $1 \times 1(s_{1 \times 1})$ filtresinden oluşmaktadır. SqueezeNet modelinde genişletilmiş katmandaki toplam filtre sayısı, sıkıştırılmış katmandaki toplam filtre sayısından büyük olarak ayarlanmıştır ($\text{num} [((e_{1 \times 1}) + (e_{3 \times 3}))] > \text{num} [(s_{1 \times 1})]$) (Özerdem, 2021: 194). Genişletilmiş katmandaki toplam filtre sayısı, sıkıştırılmış katmandaki toplam filtre sayısından büyük olabilmesi için giriş kanal sayısının 3×3 filtrelere sınırlandırılması gerçekleştirilmiştir (Şekil 21). SqueezeNet mimarisi 8 yangın modülü, başta ve sonda 2 adet evrimsel katmanı içermektedir. Giriş görüntü verisi olarak $224 \times 224 \times 3$ kullanır. Etkinleştirme fonksiyonu olarak ise ReLU kullanmaktadır. Fire9 modülünden sonra %50 oranında seyreltme (dropout) kullanılmaktadır (Kapucuoğlu, 2022: 33). Genel SqueezeNet mimarisi Şekil 22’de ifade edildiği gibidir.



Şekil 22. Sol: SqueezeNet, Orta: Basit atlamalı SqueezeNet, Sağ: Karışık Atlamalı SqueezeNet

3. 4. Öznitelik Seçme

Öznitelik seçme, belirli bir veri kümesindeki veriye ait öznitelik setinden belirli kriterlerin baz alınarak uygun olanların seçilmesi işlemidir. Özellik seçimi, orijinal özniteliklerin orijinal anlamını koruduğu ve böylece daha iyi okunabilirlik ve yorumlanabilirlik sağladığı için çeşitli veri madenciliği ve makine öğrenmesi tekniklerinde yaygın olarak kullanılmaktadır. Genel olarak bir özellik seçme yaklaşımı seçim, değerlendirme, durdurma kriteri ve doğrulama olmak üzere dört aşamadan oluşmaktadır. İlk aşamada, arama stratejisi kullanılarak öznitelik alt kümesi seçilir. İkinci aşamada, seçilen öznitelik alt kümesi belirli bir kritere göre değerlendirilir. Üçüncü aşamada, eğer durdurma kriteri sağlanıyorsa tüm olası alt kümelerden en iyi olan seçilir. Son aşamada ise seçilen alt kümenin doğrulama metrikleri kullanılarak doğrulama işlemi yapılır (Hançer vd., 2020: 4520-4522).

3. 4. 1. Komşuluk Bileşen Analizi (NCA)

Öznitelik seçme yöntemlerinden biri olan Neighborhood Component Analysis (Komşuluk Bileşen Analizi), sınıflandırma doğruluğunu maksimize edecek şekilde öznitelik ağırlıklandırma vektörünü öğrenen en yakın komşu tabanlı öznitelik ağırlıklandırma algoritmasıdır. Eğitim veri setindeki her bir örneğin kendi bulunduğu sınıftaki örneklerle komşu olabilme ihtimali artırılmaktadır. Komşuluk bileşen analizinde özniteliklerin arasındaki mesafe ölçümleri ve ağırlıkları baz alınarak yapılmaktadır (Goldberger vd., 2004; İsmailoğlu, 2021: 1743-1744). NCA algoritması non-parametrik ve denetimli öğrenme algoritmaları içerisinde yer almaktadır.

$T = \{(x_1, c_1), \dots, (x_i, c_i), \dots, (x_N, c_N)\}$, bir eğitim kümesi olduğu varsayılırsa, x_i öznitelik vektörünü, $c_i \in \{1, \dots, C\}$ sınıf etiketini ve N örnek sayısını ifade eder. NCA algoritmasında amaç, en yakın komşu algoritmasını optimize eden öznitelik vektörü x_i 'ye karşılık gelen ağırlık (w) vektörünün bulunmasıdır. NCA, bütün örneklerden örnek x_i için bir referans örnek noktası olarak x_j seçilir. x_j 'nin x_i için referans noktası olarak seçilebilme olasılığı, iki örnek arasındaki uzaklığın yakınlığına bağlı olarak daha yüksektir. Bu iki örnek arasındaki ağırlıklı uzaklık mesafesi Denklem 6 ile hesaplanabilmektedir (Yang vd., 2012: 161-163).

$$d_w(x_i, x_j) = \sum_{m=1}^r w_m^2 |x_{im} - x_{jm}| \quad (6)$$

Buradaki w_m , m'inci öz niteliğin atanan ağırlığıdır. P_{ij} , olasılığı ve d_w ağırlıklı mesafesi arasındaki ilişkiyi, küçük d_w için büyük değerler döndürebilen bir çekirdek fonksiyonu ile kurabilmek mümkündür. P_{ij} Denklem 7'deki gibi tanımlanır:

$$P_{ij} = \frac{k(d_w(x_i, x_j))}{\sum_{j=1, j \neq i}^n d_w(x_i, x_j)} \quad (7)$$

Eğer $i = j$ ise $P_{ij} = 0$; aksi takdirde 1 değerini alır. Çekirdek fonksiyonu $k = (z) \exp(-\frac{z}{\sigma})$ olarak tanımlanır, burada σ ve k parametreleri çekirdek genişliğini ifade eder ve bu parametreler x_j 'nin referans noktası olarak seçilme ihtimalini etkiler. x_i 'nin doğru olarak sınıflandırma olasılığı Denklem 8 ile ifade edilmektedir:

$$P_{ij} = \sum_{j=1, j \neq i}^n P_{ij} Y_{ij} \quad (8)$$

NCA algoritması sadece sınıflandırılma problemlerinde değil regresyon problemlerinin çözümünde de tercih edilmektedir. Regresyon ve sınıflandırma arasındaki fark ise y yanıt değerinin sürekli olmasıdır. Regresyon modellerinde x için eğitim setinden rastgele bir şekilde elde edilen $ref(x)$ noktası toplanır. $ref(x)$ referans noktasının yanıt değerine eşit x 'teki yanıt değeri ayarlanmaktadır. x_j noktasının eğitim setinde x için referans noktası olarak seçilebilme ihtimali $P(ref(x) = x_j | S)$ Denklem 9 ile ifade edilir:

$$P(ref(x) = x_j | S) = \frac{k(d_w(x_i, x_j))}{\sum_{j=1}^n k(d_w(x_i, x_j))} \quad (9)$$

Regresyonda çapraz doğrulamada, yani S^{-i} deki veriler kullanılarak eğitim veri setinde (x_i, y_i) noktası dahil edilmeyerek x_i için yanıt tahmininde bulunduğu

x_j noktasının x_i için referans noktası olarak seçilebilme ihtimali Denklem 10 ile hesaplanır:

$$P_{ij} = P(ref(x_i) = x_j | S^{-i}) = \frac{k(d_w(x_i, x_j))}{\sum_{j=1, j \neq i}^n k(d_w(x_i, x_j))} \quad (10)$$

$\tilde{y}_i$, regresyonun tahmin ettiği yanıt değerini y_i ise x_i için gerçek yanıt olduğunda iki değer arasındaki aykırılığı ölçen kayıp fonksiyonu $l(\tilde{y}_i, y_i)$ olarak Denklem 11’ deki gibi tanımlanmaktadır:

$$l_i = E(l(\tilde{y}_i, y_i) | S^{-i}) = \sum_{j=1, j \neq i}^n P_{ij} l(\tilde{y}_i, y_i) \quad (11)$$

NCA’nın aşırı öğrenmesinin önüne geçebilmek adına nihai amaç fonksiyonuna düzeltme parametresi olarak λ eklenmiştir. Böylece amaç fonksiyonu Denklem 12’deki gibi hesaplanmaktadır:

$$A = \sum_{i=1}^n P_i - \lambda \sum_{m=1}^r w_m^2 \quad (12)$$

NCA’nın nihai hedefi, amaç fonksiyonu olan A’yı maksimize etmektir. Bunu yapabilmek için A fonksiyonu, eşlenik gradyan yaklaşımı kullanılarak çözülebilmektedir.

3. 5. Destek Vektör Makineleri (SVM) ile Sınıflandırma

Destek Vektör Makinelerinin (SVM) temeli Vladimir Vapnik ve Alexey Chervonenkis (Vanpik ve Chervonenkis, 1964) tarafından atılmıştır. SVM’nin temeli her ne kadar 1960’lı yıllarda atılmış olsa da Vladir Vapnik, Isabelle Guyon Bernhard Boser (Boser vd., 1992) ve Cortes (Cortes ve Vapnik, 1995) tarafından geliştirilmiş bir algoritmadır. İlk olarak sınıflandırma problemlerine çözüm olarak ortaya çıkarılmış daha sonra regresyon problemlerine uyarlanmıştır. Temel gayesi iki sınıfın ayrımını en iyi şekilde belirleyecek hiper-düzlemi bulmaktır (Cortes ve Vapnik, 1995: 274-280). SVM algoritmalarında verilerin türüne bağlı olarak çekirdek fonksiyonlarda kullanılabilmektedir. Bu sayede hem doğrusal hem de doğrusal olmayan sınıflandırmayı yapabilmektedir. Eğer tam ayrıştırılabilen veriler kullanılırsa, hiper düzlem ile sınıflandırma yapılmaktadır. Tam

ayrıştırılamayan veriler kullanılırsa da aynı boyutta tek bir düzlem ile sınıflandırma yapılmaktadır. SVM algoritmaları, işlem basamaklarını azaltmasından dolayı çözümlemeyi hızlı bir şekilde yapmaktadır. Bu özeliğinden dolayı büyük verilerin analizinde sıklıkla tercih edilmektedir (Ayhan ve Erdoğan, 2014: 177-179).

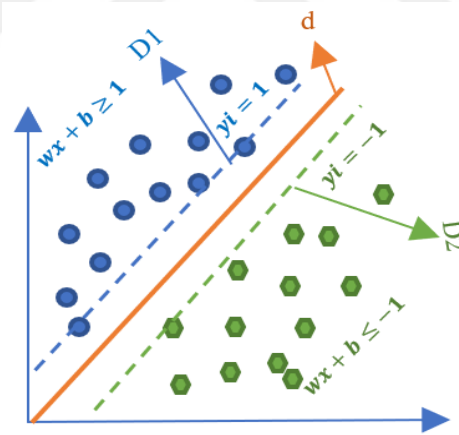
Şekil 23'te sunulan d doğrusu, Denklem 13 ile ifade edilmektedir. Denklem 13'te kullanılan w, x, b terimleri sırasıyla ağırlığı, giriş verisini ve eşik değerini ifade eder.

$$w \times x + b = 0 \quad (13)$$

D1 doğrusu Denklem 14 ile, D2 doğrusu ise Denklem 15 ile hesaplanmaktadır:

$$w \times x + b = 1 \quad (14)$$

$$w \times x + b = -1 \quad (15)$$



Şekil 23. Doğrusal Ayrılabilen Destek Vektör Makinesi

D1 ve D2 doğrusu arasında d doğrusu aralarındaki uzaklığı (marjin) ifade etmektedir. Marjin ise Denklem 16 ile hesaplanmaktadır:

$$d = \frac{2}{\|w\|} \quad (16)$$

Denklem 16’da ifade edilen $||w||$ değeri Denklem 17 kullanılarak hesaplanmaktadır:

$$d||w|| = \sqrt{w_1^2 + w_2^2} \quad (17)$$

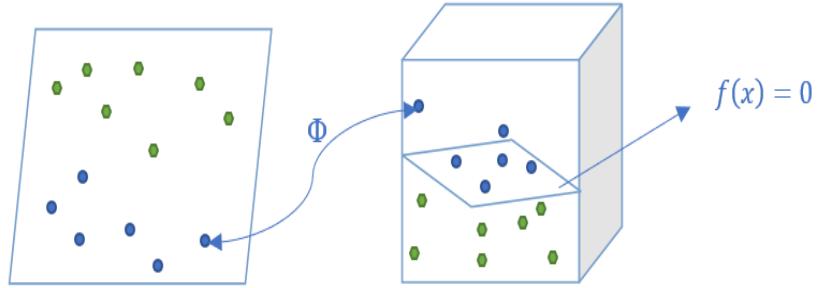
Destek vektör makinelerinde temel amaç sınıflar arasındaki uzaklığın maksimum değerini elde etmek olduğu için $||w||$ ya da $||w^2||$ ’nin minimum değerinin elde edilmesi gerekmektedir (Gürhan vd., 2014: 88-89). Denklem 18’de görüldüğü üzere Şekil 23’te yer alan mavi sınıf $y_i=1$ ile yeşil sınıf ise $y_i=-1$ ile ifade edilmektedir.

$$w \times x + b \geq 1 \text{ ise } y_i = 1 \quad (18)$$

$$w \times x + b \leq -1 \text{ ise } y_i = -1$$

$$y_i \times (w \times x) + b \geq 1 \quad (19)$$

Denklem 18’in düzenlenmesi sonucunda Denklem 19 elde edilmektedir. Kullanılan veri setlerine bağlı olarak, bazı veriler Şekil 23’te olduğu gibi doğrusal olarak ayıramamaktadır. Bu gibi durumda Şekil 24’te gösterildiği gibi Φ çekirdek fonksiyonları kullanılmaktadır.



Şekil 24. Doğrusal Ayıramayan Destek Vektör Makinesi

Tablo 8’de sunulan çekirdek fonksiyonları ile ikinci boyutta bulunan veriler üçüncü bir boyuta aktarılır. Aktarılan bu verilerin arasından $f(x)$ düzlemi geçirilerek veriler birbirinden ayrıştırılmaktadır (Şekil 24).

Tablo 8. Çekirdek Fonksiyonu

Çekirdek	Matematiksel ifadesi
Doğrusal	$x^t + c$
Gaussian	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$
Quadratic	$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
Cubic	$ax^3 + bx^2 + cx + d = 0$

Destek Vektör Makinesinin Avantajları:

- Aşırı öğrenme sorunu bulunmamaktadır.
- Çok fazla sayıdaki bağımsız değişkenler ile çalışabilmektedir.
- Karar verme aşamasında bir alt eğitim noktası kümesi (support vectors) kullandıklarından dolayı daha az bellek miktarı kullanmaktadırlar.
- Yüksek boyutlu uzaylarda başarılı sonuçlar elde edilebilir.
- Boyut sayısının, örneklem sayısından fazla olduğu durumlarda başarılı sonuçlar elde edilebilir.
- Karar fonksiyonu için çok farklı çekirdek fonksiyonlarını kullanılabilmektedir.

Yukarıdaki ve benzer avantajları dolayısı ile önerilen metodolojide sınıflandırıcı olarak destek vektör makinesi tercih edilmiştir. Bunun yanı sıra dördüncü bölümdeki deneysel sonuçlara bakıldığında da sınanan diğer sınıflandırıcılardan daha yüksek başarı elde ettiği tespit edilmiştir.

DÖRDÜNCÜ BÖLÜM

BULGULAR

4. 1. Değerlendirme Metrikleri

Çalışma kapsamında geliştirilen metodolojinin karşılaştırmalı performans analizi için kullanılan değerlendirme metrikleri şunlardır:

Doğruluk (Accuracy): Doğruluk değeri, doğru tahmin edilmiş verilerin toplam veri setine oranını ifade etmektedir:

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN} \quad (20)$$

Burada TP gerçek pozitif (true positive) ve TN gerçek negatifi (true negative) temsil ederken, FP ve FN yanlış pozitif (false positive) ve yanlış negatifi (false negative) temsil etmektedir.

Kesinlik (Precision): Doğru tespit edilen pozitif sınıfların toplam pozitif değerlere oranıdır:

$$Precision = \frac{TP}{TP + FP} \quad (21)$$

Duyarlılık (Recall): Doğru tespit edilen pozitif sınıfların gerçek pozitif değerlere oranıdır:

$$Recall = \frac{TP}{TP + FN} \quad (22)$$

F1-Skoru (F1-Score): F1-skoru, kesinlik ve duyarlılık ölçütlerinin harmonik ortalamasıdır:

$$F1 - Score = 2 * \frac{(Recall * Precision)}{(Recall + Precision)} \quad (23)$$

Cohen'in Kappa Katsayısı (Cohen Kappa Score): Cohen Kappa, iki ya da ikiden fazla gözlemci arasındaki değişimleri hesaplayarak uyumu ölçen bir yöntemdir. Hesaplanırken P_0 ve P_c kullanılmaktadır. P_0 , uyumların toplamının oranını ifade

etmektedir. P_c ise uyumların şansa bağlı olarak ortaya çıkma olasılığını ifade etmektedir. Kappa -1 ile +1 arasında değerler almaktadır. Değer +1 ise uyumun tam olduğu, 0 ise uyumun şansa bağlı olduğu, -1 ise uyumsuzluğun tam olduğu ifade edilmektedir (Denklem 24).

$$k = \frac{P_0 - P_c}{1 - P_c} \quad (24)$$

4. 2. Deneysel Sonuçlar

4. 2. 1. Öznitelik Seçme ile Elde Edilen Sonuçların Karşılaştırılması

Deneysel sonuçlar Doğruluk, F1-Skor, Kappa, Duyarlılık, Kesinlik ve Boyut üzerinden Tablo 9, 10 ve 11’de sunulmuştur. İlgili tablolarda “Boyut” elde edilen öznitelik sayısını temsil etmektedir. Çalışmada ilk olarak SqueezeNet ile öznitelik çıkarımı yapılmıştır. SqueezeNet’in 66. katmanı olan “pool10” kullanılmış ve katmandan alınan öznitelikler destek vektör makinesi (SVM) algoritması kullanılarak sınıflandırılmıştır. Veri setleri %80 eğitim %20 test olacak şekilde ayrılmıştır. Başarıyı arttırmak içinde SqueezeNet ile öznitelik çıkarımı yapıldıktan sonra NCA algoritması kullanılarak öznitelik seçme yapılmıştır. NCA algoritmasının parametreleri eşik değeri 0.02 ve iterasyon sayısı 100 olarak belirlenmiştir. Öznitelik seçme ile elde edilen öznitelikler SVM algoritması kullanılarak sınıflandırılmıştır.

Tablo 9’da iki metodoloji bulunmaktadır: “SqueezeNet” mevcut özniteliklerin ilgili derin öğrenme algoritmaları ile çıkarılarak öznitelik seçme yapılmadan elde edilen sonuçları ve “SqueezeNet-NCA” ise, önerilen metodoloji ile elde edilen sonuçları ifade etmektedir. Bir diğer söylemle derin öğrenme algoritmaları ile çıkarılan özniteliklerin, NCA ile öznitelik seçme işleminden alınan sonuçları ifade etmektedir. Mevcut metodolojilerin birbirinden tek farkı, birinin öznitelik seçme aşamasını uygulamamasıdır.

Tablo 9’a göre, önerilen metodoloji diğer metodolojiye oranla daha iyi sonuçlar elde etmiştir. Üç ve daha fazla sınıf içeren ilk dört veri setinde sınıflandırma performansı artarken veri setinin boyutu da önemli ölçüde azaltılmıştır. Örneğin, Dataset 1 ve Dataset 2’de boyut yaklaşık olarak 5 buçuk kat azalmasına rağmen sınıflandırma doğruluk oranı %6-%7 oranında artmıştır.

Diğer iki sınıf bilgisi içeren Dataset 5’te ise yine önemli ölçüde öznitelik sayısı azaltılmış ve sınıflandırma performansı genel olarak korunmuştur. Her ne kadar Dataset 5’te ufak bir performans kaybı olsa da öznitelik sayısındaki yaklaşık olarak 7 kat azalma oranı bu durumu kabul edilebilir kılmaktadır.

Tablo 10’da önerilen metodoloji ile KNN, Naive Bayes ve SVM sınıflandırıcıları kullanılarak alınan sonuçlar sunulmuştur. Tabloda ifade edilen “boyut” yani özniteliklerin tüm sınıflandırıcılarda eşit olmasının sebebi, hepsinde öznitelik çıkarımı olarak SqueezeNet’in 66. katmanı olan “pool10” katmanının kullanılmış olmasından kaynaklıdır. Veri setlerinin KNN sınıflandırıcısı ile üç ve daha fazla sınıf içeren veri setlerinde ortama doğruluk oranı %79.30, Naive Bayes’te %68.30, Bagging’de %79.72 ve SVM ‘de %83.49 gibi oranlar bulunmuştur. İki sınıflı veri setinde ise bu oranlar şu şekildedir: KNN’de %92.56, Naive Bayes’te %88.19, Bagging’de %93.37 ve SVM’de %93.85 olarak bulunmuştur. Çalışmadan elde edilen sonuçlara baktığımızda kullanılan tüm veri setlerinde en başarılı sonuçların SVM ile alınmış olduğu gözlemlenmektedir. KNN ile çoklu sınıflandırmada yaklaşık olarak %4.16, ikili sınıflandırma için ise %1.29 oranında SVM’den daha düşük sonuçlar elde edilmiştir. Bagging için çoklu sınıflandırmada yaklaşık olarak %3.77, ikili sınıflandırmada ise %0.48 gibi bir oranla SVM’den daha düşük sonuçlar elde edilmiştir. Naive Bayes için ise bu oranlar çoklu sınıflandırma için %15.19 ve ikili sınıflandırma için ise %5.66 oranında SVM’den daha düşük sonuçlar elde edilmiştir. Her ne kadar ikili sınıflandırmada doğruluk oranı biraz iyileştirilmiş olsa da genel çerçevede değerlendirildiğinde Naive Bayes sınıflandırıcısının SVM, Bagging ve KNN’nin gerisinde kaldığını söylemek mümkündür. Genel olarak değerlendirdiğimizde ise gerek ikili gerekse çoklu sınıflandırma başarılı sonuçların SVM sınıflandırıcısı ile alındığını gözlemlenmektedir. Bundan dolayı önerilen metodolojide SVM algoritmasının sınıflandırıcı olarak seçilmesinin yerinde olduğunu söyleyebiliriz.

Tablo 9. Deneysel Çalışma Sonuçları

Dataset	Yöntem	Doğruluk	F1-Skor	Kappa	Duyarlılık	Kesinlik	Boyut
Dataset 1	SqueezeNet	74.22	78.16	55.61	72.67	84.55	1000
	SqueezeNet -NCA	80.44	84.15	67.33	85.33	83	184
Dataset 2	SqueezeNet	79.74	78.98	71.96	80.32	77.68	1000
	SqueezeNet -NCA	86.22	88.77	76.87	89.67	87.89	184
Dataset 3	SqueezeNet	75.40	79.31	64.22	81.30	77.41	1000
	SqueezeNet -NCA	85.78	85.83	77.78	84.68	87.02	548
Dataset 4	SqueezeNet	80.25	75.86	69	68.45	85.06	1000
	SqueezeNet -NCA	81.50	80.33	71.36	78.69	82.04	214
Dataset 5	SqueezeNet	95.15	95.02	90.04	95.02	95.02	1000
	SqueezeNet- NCA	93.85	93.69	87.39	93.69	93.69	149

Tablo 10. Sınıflandırıcıların Karşılaştırılması

Dataset	Sınıflandırma Yöntemi	Doğruluk	F1-Skor	Kappa	Duyarlılık	Kesinlik	Boyut
Dataset 1	K- en yakın komşu (KNN)	77.78	81.70	62.12	79.33	84.21	184
	Naive Bayes	68.89	72.95	48.40	75.67	70.43	
	Bagging	79.56	83.82	65.33	82.67	85	
	Destek Vektör Makinesi (SVM)	80.44	84.15	67.33	85.33	83	
Dataset 2	K- en yakın komşu (KNN)	77.78	81.70	62.12	79.33	84.21	184
	Naive Bayes	68.89	72.95	48.40	75.67	70.43	
	Bagging	79.56	83.82	65.33	82.67	85	
	Destek Vektör Makinesi (SVM)	86.22	88.77	76.87	89.67	87.89	
Dataset 3	K- en yakın komşu (KNN)	83.13	83.54	73.55	81.22	85.99	548
	Naive Bayes	66.68	67.06	51.56	71.27	63.31	
	Bagging	82.99	82.62	73.22	79.54	85.86	
	Destek Vektör Makinesi (SVM)	85.78	85.83	77.78	84.68	87.02	
Dataset 4	K- en yakın komşu (KNN)	78.50	72.70	66.67	66.95	79.53	214
	Naive Bayes	68.75	66.20	53.70	72.50	60.91	
	Bagging	76.75	69.88	63.82	63.25	78.07	
	Destek Vektör Makinesi (SVM)	81.50	80.33	71.36	78.69	82.04	
Dataset 5	K- en yakın komşu (KNN)	92.56	92.45	84.83	92.68	92.21	149
	Naive Bayes	88.19	88.02	75.96	88.28	87.77	
	Bagging	93.37	93.19	86.38	93.17	93.21	
	Destek Vektör Makinesi (SVM)	93.85	93.69	87.39	93.69	93.69	

4. 2. 2. Önerilen Metodolojinin Diğer Derin Öğrenme Mimarileri ile Karşılaştırılması

Tablo 12’ e göre, SqueezeNet ile oluşturulan Covid-19 tespit metodolojisinin diğer derin öğrenme mimarileri ile karşılaştırılması verilmiştir. Önerilen metodolojiyi karşılaştırabilmek için GoogLeNet, ResNet18, MobileNetV2 ve VGG16 kullanılmıştır. Bu mimarilerle öznitelik çıkarımı yapılırken kullanılan katmanlar Tablo 11 ‘de ifade edildiği şekilde kullanılmıştır. Kullanılan bu katmanlar, 1000 tam bağlı katmanlardır (Full connected). Önerilen metodoloji ile genel olarak diğer derin öğrenme mimarilerinden daha iyi sonuçlara ulaşmıştır. Sadece son veri setinde SqueezeNet-NCA, GoogLeNet’in yaklaşık olarak %1 oranında gerisinde kalmıştır. Elde edilen öznitelik sayılarına bakıldığında ise, bir metodolojinin diğerine keskin bir üstünlüğü söz konusu olmayıp genel olarak tüm metodolojiler gürültülü öznitelikleri elimine etmekte başarılı bir performans sergilemiştir. Sonuç olarak önerilen Covid-19 tespit metodolojisi üç ve daha fazla sınıf içeren veri setlerinde ortalama %83.49 gibi bir oran elde ederken, bu oran iki sınıflı veri setinde %93,85’e kadar çıkmıştır. Bu durum öznitelik seçmenin derin öğrenmede ne kadar hayati bir öneme sahip olduğunu ve önerilen metodolojinin doğru bir şekilde dizayn edildiğini göstermektedir.

Tablo 11. Öznitelik Çıkarımı Yapılan Katmanlar

Yöntem	Katman
SqueezeNet	66. katman (“pool10”)
GoogLeNet	142. katman (“loss3-classifier”)
ResNet18	69. katman (“pool10”)
MobileNetV2	125. katman(“Logits”)
VGG16	39. katman (“fc8”)

Tablo 12. Deneysel Sonuçların Karşılaştırılması

Dataset	Yöntem	Doğruluk	F1-Skor	Kappa	Duyarlılık	Kesinlik	Boyut
Dataset 1	SqueezeNet-NCA	80.44	84.15	67.33	85.33	83	184
	GoogLeNet-NCA	71.11	72.13	49.61	65.33	80.50	183
	ResNet18-NCA	79.11	78.41	64.57	77.33	79.52	228
	MobileNetV2-NCA	75.56	74.81	58.44	73.67	75.99	229
	VGG16-NCA	77.33	79.59	61.36	77	82.36	166
Dataset 2	SqueezeNet-NCA	86.22	88.77	76.87	89.67	87.89	184
	GoogLeNet-NCA	81.21	79.32	73.48	79.11	79.52	291
	ResNet18-NCA	83.60	81.76	76.71	80.44	83.12	366
	MobileNetV2-NCA	65.78	70.38	40.32	63.33	79.20	345
	VGG16-NCA	53.83	70.12	40.71	61.68	81.23	294
Dataset 3	SqueezeNet-NCA	85.78	85.83	77.78	84.68	87.02	548
	GoogLeNet-NCA	80.86	81.32	69.41	77.44	85.60	458
	ResNet18-NCA	78.76	81.01	69.11	84.16	78.09	582
	MobileNetV2-NCA	81.78	81.35	70.50	74.83	89.13	557
	VGG16-NCA	80.51	81.14	68.57	75.74	87.37	474
Dataset 4	SqueezeNet-NCA	81.50	80.33	71.36	78.69	82.04	214
	GoogLeNet-NCA	78.75	76.57	67.52	73.72	73.66	210
	ResNet18-NCA	81.25	78.59	71.21	75.60	81.82	256
	MobileNetV2-NCA	78.75	76.33	66.51	68.61	86.01	249
	VGG16-NCA	77.75	73.97	65.52	75.80	72.23	188
Dataset 5	SqueezeNet-NCA	93.85	93.69	87.39	93.69	93.69	149
	GoogLeNet-NCA	95.15	95.03	90.05	95.07	94.98	146
	ResNet18-NCA	93.37	93.22	86.42	93.33	93.11	148
	MobileNetV2-NCA	94.50	94.36	88.72	94.41	94.32	179
	VGG16-NCA	94.34	94.29	88.47	94.59	94	145

SONUÇLAR

Günümüzün küresel sağlık krizi olan Covid-19 tüm dünyayı sosyal, ekonomik, sağlık ve daha birçok alanda olumsuz olarak etkilemiştir. Dolayısıyla Covid-19'un teşhisi de önemli bir unsundur. Her ne kadar farklı teşhis yöntemleri söz konusu olsa da bu yöntemlerden sonuç alınması ve teşhis konulması hastane ve hasta yoğunluğuna bağlı olarak uzayabilen bir süreçtir.

Bu tez çalışmasında X-ray görüntülerinden Covid-19 tespiti yapabilmek için uzmanlara yardımcı olabilecek geliştirilmiş bir metodoloji önerilmiştir. Önerilen metodolojinin spesifik bir veri seti yerine X-ray görüntülerinden oluşturulmuş 5 farklı veri seti üzerinde, 5 farklı metodoloji ile karşılaştırmalı performans analizi yapılarak değerlendirmesi yapılmıştır. Çalışmada önerilen “SqueezeNet-NCA” metodolojisinin çeşitli veri setleri, farklı mimariler ve farklı sınıflandırıcılar ile karşılaştırılması sonucunda başarılı sonuçların elde edildiği gözlemlenmiştir. Literatürde her ne kadar bu alanla ilgili çalışmalar yapılmış olsa da birden fazla veri setini ele alan ve öznetelik seçme işlemine değinilen çalışmaların son derece az olduğu görülmüştür. Bu tez çalışması ile bir nebze de olsa bu değinilmeyen konular ele alınarak eksiklerin giderilmesi amaçlanmıştır. Bundan sonraki çalışmalarda sadece X-ray görüntüleri ile değil bilgisayarlı tomografi görüntülerinin de yer aldığı geliştirilmiş bir Covid-19 veri seti oluşturulması ve üzerinde çalışılması düşünülmektedir.

KAYNAKÇA

- Agrawal, R., Imielinski, T. ve Swami, A., (1993), Mining association rules between sets of items in large databases, *Acm Sigmod Conference On Management Of Data*, Washington, 207-216. doi: <https://doi.org/10.1145/170035.170072>
- Agrawal, R. ve Srikant, R. (1994) Fast Algorithms for Mining Association Rules in Large Databases. *Proceedings of the 20th International Conference on Very Large Data Bases*, Santiago de Chile, 12-15 September 1994, 487-499
- Alhudhaif, A., Polat, K. ve Karaman, O. (2021). Determination of Covid-19 pneumonia based on generalized convolutional neural network model from chest X-ray images. *Expert Systems with Applications*, 180(115141). <https://doi.org/10.1016/j.eswa.2021.115141>
- Alqudah, A. M., Qazan, S., Alquran, H., Qasmieh, I. A. ve Alqudah, A. (2020). Covid-19 detection from x-ray images using different artificial intelligence hybrid models. *Jordan Journal of Electrical Engineering*, 6(2), 168-178. doi: 10.5455/jjee.204-1585312246
- Ardakani, A. A., Kanafi, A. R., Acharya, U. R., Khadem, N. ve Mohammadi, A. (2020). Application of deep learning technique to manage Covid-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks. *Computers in Biology and Medicine*, 121(103795), 1-9. doi: <https://doi.org/10.1016/j.compbiomed.2020.103795>
- Ayhan, S. ve Erdoğan, Ş. (2014). Destek vektör makineleriyle sınıflandırma problemlerinin çözümü için çekirdek fonksiyonu seçimi. *Eskişehir Osmangazi Üniversitesi İktisadi ve İdari Bilimler Dergisi*, 9(1), 175-201.
- Boser, B. E., Guyon, I. M. ve Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *In Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144-152. doi: <https://doi.org/10.1145/130385.130401>
- Bozkurt, F. (2021). Derin öğrenme tekniklerini kullanarak akciğer X-ray görüntülerinden Covid-19 tespiti. *Avrupa Bilim ve Teknoloji Dergisi*, (24), 149-156. doi: 10.31590/ejosat.898385
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32. doi: <https://doi.org/10.1023/A:1010933404324>
- Calina, D., Docea, A. O., Petrakis, D., Egorov, A. M., Ishmukhametov, A. A., Gabibov, A. G., Shtilman, M. I., Kostoff, R., Carvalho, F., Vinceti, M., Spandidos, D. A. Ve Tsatsakis, A. (2020). Towards effective Covid-19 vaccines: Updates, perspectives

- and challenges (Review). *International Journal of Molecular Medicine*, 46(1), 3–16. doi: <https://doi.org/10.3892/ijmm.2020.4596>
- Chang, L., Yan, Y. ve Wang, L. Coronavirus disease 2019: Coronaviruses and blood safety. *Transfusion Medicine Reviews*, 34(2), 75–80. doi: <https://doi.org/10.1016/j.tmr.2020.02.003>
- Chollet, F. (2019). Python ile Derin Öğrenme. çev. Aksoy, B. (1.Baskı) Buzdağı Yayınevi.
- Chowdhury, M. E., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., Islam, K. R., Khan, M. S., Al-Emadi, N., Reaz, M. R ve Islam, T. I (2020). Can AI help in screening viral and Covid-19 pneumonia?. *IEEE Access*, 8, 132665-132676. doi: 10.1109/ACCESS.2020.3010287.
- Cohen, J. P. (6 Ağustos 2020). *Covid chestxray dataset*. Erişim Tarihi: 7 Ağustos 2022, <https://github.com/ieee8023/covid-chestxray-dataset>
- Cortes, C. ve Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297. doi: <http://dx.doi.org/10.1007/BF00994018>
- Cover, T.M. ve Hart, P.E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21–27.
- Cömert, S. Ş., & Kırıl, N. (2020). Covid-19 pnömonisinin radyolojik bulguları. *Southern Clinics of Istanbul Eurasia*.
- Çölkesen, İ. (2015). *Yüksek çözünürlüklü uydu görüntüleri kullanarak benzer spektral özelliklere sahip doğal nesnelerin ayırt edilmesine yönelik bir metodoloji geliştirme* (Yayın Nu. 389359) [Doktora tezi, İstanbul Teknik Üniversitesi]. <http://hdl.handle.net/11527/12528>
- Demir, F. B. ve Yılmaz, E. (2021). X-ray görüntülerinden Covid-19 tespiti için derin öğrenme temelli bir yaklaşım. *Avrupa Bilim ve Teknoloji Dergisi*, 32, 627-632 . doi: 10.31590/ejosat.1039522
- Deshpande, D. (t.y.). *Covid-19 detection X-ray dataset*. Erişim Tarihi: 7 Ağustos 2022, <https://www.kaggle.com/datasets/darshan1504/covid19-detection-xray-dataset>
- Doğanalp, T. (2022). *Tümör tanılı farklı sekans manyetik rezonans görüntülerinin derin öğrenme modelleri ile sınıflandırılması* (Yayın Nu. 721163) [Yüksek lisans tezi, Necmettin Erbakan Üniversitesi]. <https://acikerisim.erbakan.edu.tr/xmlui/handle/20.500.12452/8218>

- Doğan, F. ve Türkoğlu, İ. (2018). Derin öğrenme algoritmalarının yaprak sınıflandırma başarımlarının karşılaştırılması. *Sakarya University Journal of Computer and Information Sciences*, 1(1), 10-21.
- Endachev, D. V., Vasin, P. A. ve Shadrin, S. S. (2019). Applicability of computer vision architectures and their influence on traffic safety of autonomous vehicles. *International Journal of Engineering and Advanced Technology*, 8(6), 5295-5301. doi: 10.35940/ijitee.F91540981119
- Eryılmaz, F. ve Karacan, H. (2021). Akciğer X-ray görüntülerinden Covid-19 tespitinde hafif ve geleneksel evrişimsel sinir ağ mimarilerinin karşılaştırılması. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 9(6), 26-39. doi: 10.29130/dubited.1011829
- Fix, E. ve Hodges, J.L. (1951) Discriminatory analysis, nonparametric discrimination: Consistency properties. *Technical Report 4, USAF School of Aviation Medicine, Randolph Field*.
- Hançer, E., Xue, B. Ve Zhang, M. (2020). A survey on feature selection approaches for clustering. *Artificial Intelligence Review*, 53(6), 4519-4545. doi: <https://doi.org/10.1007/s10462-019-09800-w>
- He, K., Zhang, X., Ren, S. ve Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 770-778. doi: 10.1109/CVPR.2016.90 770-778.
- Hemateja, A. (t.y.). *Covid X-ray dataset*. Erişim Tarihi: 7 Ağustos 2022, <https://www.kaggle.com/datasets/ahemateja19bec1025/covid-xray-dataset>
- Hemdan, E. E. D., Shouman, M. A. ve Karar, M. E. (2020). Covidx-net: A framework of deep learning classifiers to diagnose Covid-19 in X-ray images. *arXiv preprint arXiv:2003.11055*.
- Goldberger, J., Hinton, G. E., Roweis, S. Ve Salakhutdinov, R. R. (2004). Neighbourhood components analysis. *Advances in Neural Information Processing Systems* 17 (NIPS).

- Güran, A., Uysal, M. ve Doğrusöz, Ö. (2014). Destek vektör makineleri parametre optimizasyonunun duygu analizi üzerindeki etkisi. *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, 16 (48), 86-93.
- İsmailoğlu, F. (2021). Metrik öğrenmesi kullanarak çeşitli kanser dokularına ait mikro dizi gen verilerinin sınıflandırılması. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 9 (5), 1739-1753. doi: 10.29130/dubited.886353
- Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J. ve Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.
- Ismael, A. M. ve Şengür, A. (2021). Deep learning approaches for Covid-19 detection based on chest X-ray images. *Expert Systems with Applications*, 164 (114054). doi: 10.1016/j.eswa.2020.114054
- Jaiswal, A., Gianchandani, N., Singh, D., Kumar, V., ve Kaur, M. (2020). Classification of the Covid-19 infected patients using DenseNet201 based deep transfer learning. *Journal of Biomolecular Structure and Dynamics*, 39(15), 5682-5689. doi: <https://doi.org/10.1080/07391102.2020.1788642>
- Kapucuoğlu, K. (2022). *Derin öğrenme ağları kullanılarak doğal ortamda hastalıklı domateslerin belirlenmesi* (Yayın Nu. 737018) [Yüksek lisans tezi, İstanbul Teknik Üniversitesi].
- Karadoğan, A. (2014). *Takviyeli öğrenme için yapay atom algoritması (a3) kullanımı* (Yayın Nu. 390631) [Yüksek lisans tezi, İnönü Üniversitesi]. <http://hdl.handle.net/11616/5929>
- Kaya, T. (2017). Radyografinin temel prensipleri ve radyografik yorumda temel ilkeler, *Türk Radyoloji Seminerleri*, Türk Radyoloji Derneği, Eskişehir.
- Koçer, B. (2012). *Transfer öğrenmede yeni yaklaşımlar* (Yayın Nu. 315854) [Selçuk Üniversitesi, Doktora Tezi]. <http://hdl.handle.net/123456789/1429>
- Kutlugün, M. A. (2017). *Gözetimli makine öğrenmesi yoluyla türe göre metinden ses sentezleme* (Yayın Nu. 489610) [Yüksek lisans tezi, İstanbul Sabahattin Zaim Üniversitesi]. <https://openaccess.izu.edu.tr/xmlui/handle/20.500.12436/569>

- Lam, T. T. Y., Jia, N., Zhang, Y. W., Shum, M. H. H., Jiang, J. F., Zhu, H. C. ve Cao, W. C. (2020). Identifying SARS-CoV-2-related coronaviruses in Malayan pangolins. *Nature*, 583(7815), 282-285. doi: <https://doi.org/10.1038/s41586-020-2169-0>
- Loey, M., Smarandache, F. ve M. Khalifa, N. E. (2020). Within the lack of chest Covid-19 X-ray dataset: a novel detection model based on GAN and deep transfer learning. *Symmetry*, 12(651). doi: <https://doi.org/10.3390/sym12040651>
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281-297.
- McCulloch, W. S. ve Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133. doi: <https://doi.org/10.1007/BF02478259>
- Metlek, S., Çetiner, H. (2021). Matlab Ortamında Derin Öğrenme. İksad Yayınevi.
- Minaee, S., Kafieh, R., Sonka, M., Yazdani, S. ve Jamalipour Soufi, G. (2020). Deep-Covid: Predicting Covid-19 from chest X-ray images using deep transfer learning. *Medical image analysis*, 65 (101794). doi: <https://doi.org/10.1016/j.media.2020.101794>
- Moulaei, K., Shanbehzadeh, M., Mohammadi-Taghiabad, Z. ve Kazemi-Arpanahi, H. (2022). Comparing machine learning algorithms for predicting Covid-19 mortality. *BMC Medical Informatics and Decision Making*, 22(1), 1-12. doi: <https://doi.org/10.1186/s12911-021-01742-0>
- Narin, A., Kaya, C. ve Pamuk, Z. Automatic detection of coronavirus disease (Covid-19) using X-ray images and deep convolutional neural networks (2021). *Pattern Analysis and Applications* 24, 1207–1220. doi: <https://doi.org/10.1007/s10044-021-00984-y>
- Nayak, S. R., Nayak, D. R., Sinha, U., Arora, V. ve Pachori, R. B. (2021). Application of deep learning techniques for detection of Covid-19 cases using chest X-ray images: A comprehensive study. *Biomedical Signal Processing and Control*, 64(102365). <https://doi.org/10.1016/j.bspc.2020.102365>

- Özcan, T. (2020). A deep learning framework for coronavirus disease (COVID-19) Detection in X-Ray Images. *Research Square*. doi:10.21203/rs.3.rs-26500/v1.
- Özcan T. (2021). A new composite approach for Covid-19 detection in X-ray images using deep features. *Applied Soft Computing*, 111, 107669. doi: <https://doi.org/10.1016/j.asoc.2021.107669>
- Öztemel, E. (2003).Yapay sinir ağları.(1.Baskı) Papatya Yayıncılık.
- Pirim, A. G. H. (2006). Yapay Zeka. *Yaşar Üniversitesi E-Dergisi*, 1 (1), 81-93.
- Polat, H. ve Özerdem, M. S. (2021). Derin transfer öğrenimi yaklaşımı ile kamusal alanda medikal maske kullanımının otomatik kontrolü. *Türk Doğa ve Fen Dergisi*, 10 (2), 191-198. doi: 10.46810/tdfd.948098
- Qiu, J., Wu, Q., Ding, G., Xu, Y. ve Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 67, 1-16. doi: <https://doi.org/10.1186/s13634-016-0355-x>
- Rahman, T., Chowdhury, M. ve Khandakar, A. (t.y.). *Covid-19 radiography database*. Erişim Tarihi: 7 Ağustos 2022, <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>
- Rahman, T., Chowdhury, M. E., Khandakar, A., Islam, K. R., Islam, K. F., Mahbub, Z. B., M.A., Kadir ve Kashem, S. (2020). Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray. *Applied Sciences*, 10(9), 3233. doi: <https://doi.org/10.3390/app10093233>
- Sait U. (t.y.). *Curated chest X-ray image dataset for Covid-19*. Erişim Tarihi: 7 Ağustos 2022, <https://www.kaggle.com/datasets/unaisait/curated-chest-xray-image-dataset-for-covid19>
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3), 210–229. doi:10.1147/rd.33.0210
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. ve Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, USA, 4510-4520. doi: 10.1109/CVPR.2018.00474.

- Sarıman, G. (2014). Veri madenciliğinde kümeleme teknikleri üzerine bir çalışma: K-means ve K-medoids kümeleme algoritmalarının karşılaştırılması. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 15 (3), 192-202.
- Sethy, P.K. ve Behera, S.K. (2020) Detection of coronavirus disease based on deep features. *Preprints*. doi: 10.20944/preprints202003.0300.v1
- Simonyan, K. ve Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 1-14.
- Sünneci, K. M., Alkan, A. ve Tar, E. (2021). Göğüs X-ray görüntülerinin AlexNet tabanlı sınıflandırılması. *Computer Science, 5th International Artificial Intelligence and Data Processing Symposium*, 375-384. doi: 10.53070/bbd.989192
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. ve Rabinovich, A. (2015). Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 1-9. doi: 10.1109/CVPR.2015.7298594
- Tang, Y. W., Schmitz, J. E., Persing, D. H., ve Stratton, C. W. (2020). Laboratory diagnosis of Covid-19: Current issues and challenges. *Journal of Clinical Microbiology*, 58(6). doi: <https://doi.org/10.1128/JCM.00512-20>
- T.C. Sağlık Bakanlığı. (t.y.). *Covid-19*. Erişim Tarihi: 10 Ekim 2022, <https://covid19.saglik.gov.tr>
- T.C. Sağlık Bakanlığı. (t.y.). *Covid-19 nedir*, Erişim Tarihi: 8 Eylül 2022, <https://covid19.saglik.gov.tr/TR-66300/Covid-19-nedir-.html>.
- Bayes, T. (1763). Essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53, 370–418.
- Toğaçar, M., Ergen, B. ve Sertkaya, M. E. (2019). Zatürre hastalığının derin öğrenme modeli ile tespiti. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 31 (1), 223-230.
- Vapnik, V. ve Chervonenkis, A. Y. (1964). A class of algorithms for pattern recognition learning. *Avtomat. i Telemekh*, 25(6), 937-945.

- Wang, L., Lin, Z. Q. ve Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific reports*, 10(1), 1-12. doi: <https://doi.org/10.1038/s41598-020-76550-z>
- Wu, Xiangjun; Hui, Hui; Niu, Meng; Li, Liang; Wang, Li; He, Bingxi; Yang, Xin; Li, Li; Li, Hongjun; Tian, Jie; Zha, Yunfei (2020). Deep learning-based multi-view fusion model for screening 2019 novel coronavirus pneumonia: A multicentre study. *European Journal of Radiology*, 128, 109041. doi: 10.1016/j.ejrad.2020.109041
- Yang, W., Wang, K. ve Zuo, W. (2012). Neighborhood component feature selection for high-dimensional data. *Journal of Computers*, 7(1), 161-168.
- Yaşar, H. (2022). *Covid-19 hastalığının derin öğrenme yöntemleri kullanılarak tespiti* (Yayın Nu. 747494) [Doktora tezi, Konya Teknik Üniversitesi].
- Yazıcı, K. (2021). *Makine öğrenmesi yöntemleri kullanılarak kısa dönem rüzgar gücü tahmini* (Yayın Nu. 688438) [Yüksek lisans tezi, Sakarya Üniversitesi]. <https://hdl.handle.net/20.500.12619/97232>
- Yurtsever, İ. (2021). Covid-19 pandemisinde hastalık/hastane yönetimi. (1. Baskı) Bezmialem Vakıf Üniversitesi Yayınları.
- Yücel, Y. B., Aytekin, A., Ayaz, A. ve Tüminçin, F. (2018). Bilişim sistemlerinin sağlık sektörü açısından önemi. *Avrasya Sosyal ve Ekonomi Araştırmaları Dergisi*, 5 (8), 147-155.

EK 1. Kod Dizini

Çalışmada Kullanılan Veri Setleri Sırasıyla:

1. Veri Seti: <https://github.com/ieee8023/covid-chestxray-dataset>
2. Veri Seti: <https://www.kaggle.com/datasets/unaissait/curated-chest-xray-image-dataset-for-covid19>
3. Veri Seti: <https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>
4. Veri Seti: <https://www.kaggle.com/datasets/darshan1504/covid19-detection-xray-dataset>
5. Veri Seti: <https://www.kaggle.com/datasets/ahemateja19bec1025/covid-xray-dataset>

Çalışmanın GitHub platformuna eklenmiş hali:

<https://github.com/Gizemoter/Development-of-Deep-Learning-Based-Models>

1. Aşama olan Transfer Learning işleminin kodu aşağıdaki gibidir. Kod dizininde bulunan “SqueezeNet” mimarisi kullanılmış olmalıdır. Ancak diğer mimarilerde de benzer kod dizini olup kullanılan mimarilerin özelliklerine göre değiştirilmesi gerekmektedir.

```
clear,clc, close all
datapath='Data';%Veri seti yüklendi
imds=imageDatastore(datapath, ...
    'IncludeSubfolders',true, ...
    'LabelSource','foldernames');
All_samples=countEachLabel(imds)
net =squeezeNet(); %Kullanacağımız ağ mimarisi
net.Layers(1)%İlk katmanın girdi boyutlarını verir
net.Layers(end)%Son katmanın girdi boyutlarını verir
numel(net.Layers(end).ClassNames) %Çıkış katmanında bulunan
toplam sınıf sayısı
[testSet,trainingSet] = splitEachLabel(imds, 0.2,
    'randomize');%Veri seti %20 test %80 eğitim olarak ayrıldı
imageSize = net.Layers(1).InputSize;
augmentedTrainingSet = augmentedImageDatastore(imageSize,
    trainingSet, 'ColorPreprocessing', 'gray2rgb');
%Yeniden boyutlandırmak için eğitim ve test kümesinden
%augmentedImageDatastore oluşturuldu
augmentedTestSet = augmentedImageDatastore(imageSize, testSet,
    'ColorPreprocessing', 'gray2rgb');
featureLayer = 'pool10';
% " SqueezeNet'in sondan 3. olan "pool10" katmanı kullanılarak
öznitelik çıkarımı yapıldı
testFeatures = activations(net, augmentedTestSet, featureLayer,
    ...
    'MiniBatchSize', 16, 'OutputAs', 'columns');% CNN
kullanarak test özniteliklerini çıkartıldı
testLabels = testSet.Labels; % Sınıf etiketleri alındı.
```

2. Aşama olan “NCA” kullanılarak öz nitelik seçilmesi için kullanılan kod dizini aşağıdaki gibidir. Ancak diğer mimarilerde de benzer kod dizini olup kullanılan mimarinin “öz nitelik çıkarımı” alınmalıdır.

```
clear,clc, close all
datapath='C:\Users\gizem\Desktop\Seminer Covid-19\Dataset_8';
%SqueezeNet ile çıkarılan öz niteliklerin kaydedildiği yerin
dosya yolu
feature_extractor='squeezenet.mat';
load(feature_extractor, 'trainingFeatures')
load(feature_extractor, 'trainingLabels')
load(feature_extractor, 'testFeatures')
load(feature_extractor, 'testLabels')
%Gerekli parametrelerin yüklenmesi
traindata = trainingFeatures';
testdata = testFeatures';
%trainingFeatures ve testFeaturesların transpozesi alındı.
y_train = trainingLabels;
y_test = testLabels;
ncaMdl =
fscnca(traindata,y_train,'FitMethod','exact','Verbose',1, ...
'Solver','lbfgs','IterationLimit',100);
tol = 0.02;% Eşik değeri belirlendi
subset = find(ncaMdl.FeatureWeights >
tol*max(1,ncaMdl.FeatureWeights));
%Öznitelik ağırlıkları ve eşik değeri kullanılarak NCA ile
öz nitelik seçilmesi
trainingFeatures = traindata(:,subset);
testFeatures = testdata(:,subset);
```

3. Aşama olan elimine edilen öz niteliklerin sınıflandırılmasının yapılması için kullanılan kod dizini aşağıdaki gibidir.

Sınıflandırma algoritmalarını uygulamadan önce performans metrikleri fonksiyonunu oluşturmamız gerekir. Fonksiyonun kod dizini aşağıdaki gibidir.

```
function [acc,overall_precision,overall_recall,f1_score] =
calculatemetrics(actual,predicted)

cm = confusionmat(actual, predicted);
cmt = cm';

diagonal = diag(cmt);
sum_of_rows = sum(cmt, 2);

acc = sum(diagonal)/sum(sum(cmt));%Doğruluk
precision = diagonal ./ sum_of_rows;%Kesinlik
overall_precision = mean(precision);

sum_of_columns = sum(cmt, 1);

recall = diagonal ./ sum_of_columns';
overall_recall = mean(recall);%Duyarlılık
```

```
f1_score =
2*((overall_precision*overall_recall)/(overall_precision+overall
_recall));%F1 skor
```

```
end
```

```
function kappa = cohensKappa(y, yhat)
    C = confusionmat(y, yhat);
    n = sum(C(:));
    C = C./n;
    r = sum(C,2);
    s = sum(C);
    expected = r*s;
    po = sum(diag(C));
    pe = sum(diag(expected));
    kappa = (po-pe)/(1-pe); %Cohen kappa
```

```
end
```

Sınıflandırma algoritmaları için kullanılan kod dizini:

```
%SVM
Mdl = fitcsvm(trainingFeatures, y_train);
%2'li sınıflandırmalarda "fitcsvm" çoklu sınıflarda "fitcecoc"
%kullanılmıştır
predicted = predict(Mdl, testFeatures);
[acc_svm,precision_svm,recall_svm,f1_svm]=calculatemetrics(y_test,
predicted);
[kappa_svm]=kappa(y_test,predicted);

%KNN
Mdl= fitcknn(trainingFeatures, y_train,'NumNeighbors',5);
predicted = predict(Mdl, testFeatures);
[acc_knn,precision_knn,recall_knn,f1_knn]=calculatemetrics(y_test,
predicted);
[kappa_knn]=kappa(y_test,predicted);

%BAYES
Mdl= fitcnb(trainingFeatures, y_train)
predicted = predict(Mdl, testFeatures);
[acc_cnb,precision_cnb,recall_cnb,f1_cnb]=calculatemetrics(y_test,
predicted);
[kappa_cnb]=kappa(y_test,predicted);

%bagging
rng('default');
Mdl =
TreeBagger(50,trainingFeatures,y_train,'Method','classification',
'NumPredictorsToSample',75);
predicted = predict(Mdl, testFeatures);
predicted = str2double(predicted);
[acc_bagger,precision_bagger,recall_bagger,f1_bagger]=calculatemetrics(y_test,
predicted);
[kappa_bagger]=kappa(y_test,predicted);
```

ÖZGEÇMİŞ

Kişisel Bilgiler :

Adı ve Soyadı : Gizem ÖTER

Eğitim Durumu :

Lisans Öğrenimi :

Burdur Mehmet Akif Ersoy Üniversitesi/ Yönetim Bilişim Sistemleri (2016-2020)

Anadolu Üniversitesi/ Uluslararası İlişkiler (Açıköğretim) (2017-Devam ediyor)

Yüksek Lisans Öğrenimi :

Burdur Mehmet Akif Ersoy Üniversitesi/Sosyal Bilimler Enstitüsü/ Yönetim Bilişim Sistemleri (Tezli) (2021-2023)

Yabancı Dil ve Düzeyi :

İngilizce: Orta

İş Deneyimi :

Bilişim Destek Hizmetleri- Stajyer (2014)

Demircioğlu Group- Stajyer (2020)

Bilimsel Yayınlar ve Çalışmalar :

Öter G., Hançer E. (2022). İslami Kaynakların Metin Madenciliği Yöntemleriyle İncelenmesi. İslam Hukuku Araştırmalarına Zemin Oluşturması Açısından “Yapay Zeka” Sempozyumu, Nisan 15-17, Burdur.

Öter G., Hançer E. (2023). X-ray Görüntülerinden Covid-19 Tespiti için Hibrit Bir Derin Öğrenme Yaklaşımı. *Avrupa Bilim ve Teknoloji Dergisi*. (Hakemde)